



三汇系列语音卡

程序员手册

Version 5.4.3.0

杭州三汇信息工程有限公司

www.synway.cn

目 录

目 录	i
版权声明	liii
SynCTI 驱动程序授权协议	liv
版本修订记录	lv
前 言	lvii
1 Synway 板卡编程基础	1
1.1 Synway 板卡的分类 (CTI 系列)	1
1.2 Synway 板卡的分类 (REC 系列)	3
1.3 SynCTI 支持的 CODEC	4
1.3.1 编码格式转换的工具函数	4
1.3.2 板卡支持的 CODEC	6
1.4 SynCTI 驱动程序支持的操作系统	9
1.5 SynCTI 驱动程序的体系结构 (CTI 系列)	9
1.6 SynCTI 驱动程序的体系结构 (REC 系列)	11
1.7 SynCTI 驱动程序的配套工具软件	12
1.7.1 系统配置工具软件 ShCtiConfig.exe	12
1.7.2 信号音分析工具软件 ShTA.exe	13
1.7.3 SS7 信令服务器配置程序 SS7Cfg.exe	13
1.7.4 SS7 信令服务器 SS7monitor.exe	13
1.7.5 MSU 解码工具软件 MsuDecode.exe	13
1.8 基本概念	13
1.8.1 通道 (CTI 系列)	13
1.8.2 通道 (REC 系列)	14
1.8.3 通道的逻辑编号	14
1.8.4 数字中继线	15
1.8.4.1 数字中继线的帧同步	15
1.8.4.2 数字中继线的时钟	15
1.8.4.3 数字中继线的配置	15
1.8.4.4 数字中继线中的信令时隙	16
1.8.4.5 数字中继线的编号	16
1.8.4.6 设置 B 信道的语音编码格式	16
1.8.5 板卡的用户授权号	16
1.8.6 模拟电话线的电压变化	16
1.8.7 模拟电话线上的铃流信号	19
1.8.8 模拟电话线上的闪断信号	19
1.8.9 模拟电话线上的主叫号码	19
1.9 配置应用系统的时钟	20

1.10 语音处理	21
1.10.1 语音播放 (CTI 系列)	21
1.10.1.1 设置放音数据音量及去向	22
1.10.1.1.1 CTI 系列	22
1.10.1.1.2 REC 系列	22
1.10.1.2 设置播放任务的终止条件	22
1.10.1.3 播放预加载文件	23
1.10.1.4 播放单个文件	24
1.10.1.5 播放文件序列	25
1.10.1.6 播放单个内存	25
1.10.1.7 播放内存序列	25
1.10.1.8 播放乒乓内存	26
1.10.1.9 终止播放任务的通用函数	26
1.10.1.10 获取播放任务的相关信息	27
1.10.2 录音操作	27
1.10.2.1 设置录音信号源	27
1.10.2.2 设置录音任务的终止条件	27
1.10.2.3 设置预录音功能	27
1.10.2.4 设置录音文件截尾功能	28
1.10.2.5 设置录音 AGC	28
1.10.2.6 获取录音操作的信息	29
1.10.2.7 录音函数概述	30
1.10.2.7.1 文件录音	30
1.10.2.7.2 内存录音	30
1.10.2.7.3 乒乓缓冲区录音	31
1.10.3 DTMF 检测器(DTMF Detector)	31
1.10.3.1 DTMF 滤波器 (DTMF Filter)	32
1.10.3.2 DTMF 脉宽滤波器	33
1.10.3.3 WaitDtmf 任务	34
1.10.3.4 Callback 方式编程	34
1.10.3.5 检测脉冲方式的按键信号	34
1.10.4 DTMF 发生器(DTMF Generator) (CTI 系列)	34
1.10.5 信号音检测器(Tone Detector)	35
1.10.5.1 FFT 模块	38
1.10.5.2 噪声滤波器(Noise Filter)	38
1.10.5.3 呼叫进程音检测器(Call Progress Tone Detector) (SHT 系列 only)	39
1.10.5.3.1 拨号音检测器 (Dial Tone Detector)	40
1.10.5.3.2 回铃音检测器	40
1.10.5.3.3 忙音检测器	41
1.10.5.3.3.1 常规功能描述	41
1.10.5.3.3.2 背靠背忙音检测功能	42
1.10.5.3.4 自定义信号音检测器 (User-defined Tone Detector)	43
1.10.5.4 传真进程音检测器(CTI 系列)	43
1.10.5.5 简易忙音检测器(Simplified Busy Tone Detector)	44

1.10.5.6	机器应答检测器 (Answering Machine Detector)	45
1.10.5.7	增强信号音检测器(Enhanced Tone Detector)	45
1.10.6	信号音发生器(<i>Tone Generator</i>) (CTI 系列)	47
1.10.6.1	设置信号音发生器的工作参数	48
1.10.6.2	产生信号音	48
1.10.7	回波抵消器 (CTI 系列)	48
1.10.8	Barge in 检测器 (Barge in Detector)	49
1.10.9	TDM 交换总线	50
1.10.10	分布式电话会议(CTI 系列)	52
1.10.10.1	基本概念	52
1.10.10.2	管理会议室	53
1.10.10.3	管理会议通道	54
1.10.10.4	给会议室播放背景音乐	54
1.10.10.5	录制会议室的声音	55
1.11	FSK 数据收发器 (CTI 系列)	55
1.11.1	FSK 数据发送器	55
1.11.2	FSK 数据接收器	56
1.12	传真 (CTI 系列)	56
1.12.1	传真资源通道的数量	56
1.12.2	传真操作支持的速率	57
1.12.3	传真操作支持的文件格式	59
1.12.4	设置传真操作的工作参数	60
1.12.5	获取传真操作的信息	60
1.12.6	发送传真	61
1.12.7	接收传真	62
1.13	板载音频功率放大器	62
1.14	编写基于 Synway 板卡的应用程序	62
1.14.1	设置驱动程序的事件输出模式	62
1.14.1.1	事件等待方式的编程示例	63
1.14.1.2	事件回调方式的编程示例	65
1.14.2	输出事件的数据结构	67
1.14.2.1	MESSAGE_INFO	67
1.14.2.2	SSM_EVENT	77
1.14.2.3	IPR_SessionInfo	79
1.14.2.4	Reg_Info	80
1.14.2.5	RegResp	81
1.14.3	事件的过滤	81
1.14.4	使用自定义事件	81
1.14.5	常见编程平台下的编程要点	81
1.14.5.1	MS VC/C++	81
1.14.5.2	VB	81
1.14.5.3	C++ BUILDER	82
1.14.5.4	Delphi	82
1.14.5.5	PB6.5	83

1.14.5.6 其它.....	83
1.14.5.7 Linux 操作系统.....	83
1.14.5.8 字符串的使用	84
1.14.6 配套演示程序 (CTI 系列)	84
1.14.7 配套演示程序 (REC 系列)	85
1.14.8 使用驱动程序提供的定时器.....	85
1.14.9 使用驱动程序提供的调试功能.....	85
1.15 SHT 系列板卡 (CTI 系列)	86
1.15.1 SHT 系列板卡简介	86
1.15.2 SHT 系列板卡的工作原理框图.....	86
1.15.3 模拟中继线通道.....	89
1.15.3.1 在模拟线路上产生闪断信号	89
1.15.3.2 检测极性反转信号	89
1.15.3.3 模拟中继线通道的状态转移	89
1.15.3.4 远端摘机检测器(Remote Pickup Detector) (SHT 系列 only)	92
1.15.3.4.1 普通的远端摘机检测器	93
1.15.3.4.2 增强的远端摘机检测器	94
1.15.4 坐席通道.....	95
1.15.4.1 状态迁移图	95
1.15.4.2 检测话机的摘机/挂机动作	96
1.15.4.3 检测话机的闪断信号.....	96
1.15.4.4 向电话机发送振铃信号	96
1.15.4.5 在电话线上产生极性反转信号	96
1.15.4.6 检测到话机摘机后自动送拨号音	96
1.15.5 联合模块.....	97
1.15.6 磁石电话机通道.....	97
1.15.6.1 概述.....	97
1.15.6.2 状态迁移图	98
1.15.7 EM 模块	98
1.15.8 使用未安装业务模块的通道	98
1.16 SHD 系列板卡 (CTI 系列)	99
1.16.1 SHD 系列板卡简介	99
1.16.2 SHD 系列板卡的工作原理框图.....	100
1.16.3 设置来话呼叫的收号规则	103
1.16.4 SS7 协议的编程	103
1.16.4.1 OPC 和 DPC	103
1.16.4.2 时隙分配	103
1.16.4.3 设置地址信号中备用值的对应字符	105
1.16.4.4 SS7 信令应用系统的配置实例	106
1.16.4.5 TUP 协议的编程与应用	107
1.16.4.5.1 TUP 通道的状态转移	107
1.16.4.5.2 通道的闭塞与解除.....	112
1.16.4.5.2.1 通道闭塞的概念.....	112
1.16.4.5.2.2 通道闭塞的作用	113

1.16.4.5.2.3	通道闭塞的途径	113
1.16.4.5.2.4	实现通道闭塞的函数列表	113
1.16.4.5.3	电路的复原	114
1.16.4.5.4	“锁定主叫用户”功能	114
1.16.4.5.5	TUP 协议的配置	114
1.16.4.6	ISUP 协议的编程与应用	116
1.16.4.6.1	ISUP 协议的配置	116
1.16.4.6.2	ISUP 通道的状态转移	118
1.16.4.6.3	ISUP 通道工作于汇接局方式	123
1.16.4.6.4	不使用 SynCTI 提供的 ISUP 协议状态机	124
1.16.4.7	SS7 的高级编程	124
1.16.4.7.1	使用 MTP3 层提供的服务	124
1.16.4.7.2	使用 SS7 服务器的客户端软件编程接口	125
1.16.4.7.3	使用 SS7 服务器的虚电路编程接口	125
1.16.4.7.4	使用 SCCP 协议的编程接口	125
1.16.5	ISDN 通道的编程	125
1.16.5.1	基本概念	125
1.16.5.1.1	网络侧模式与用户侧模式	125
1.16.5.1.2	TEI 值	126
1.16.5.1.3	通路识别表示方法	126
1.16.5.1.4	端局方式与汇接局方式	126
1.16.5.1.5	ISDN 号码与 ISDN 地址	126
1.16.5.2	ISDN 通道的状态转移图	127
1.16.5.3	ISDN 协议的配置	131
1.16.5.4	ISDN 高级编程接口	134
1.16.6	SS1 通道编程指南	134
1.16.6.1	选择信令协议	134
1.16.6.2	设置接续过程的基本工作参数	134
1.16.6.3	设置状态机中通道的工作方式	135
1.16.6.4	中国 1 号信令的状态机描述	135
1.16.6.4.1	来话呼叫时启用号码拦截功能	135
1.16.6.4.2	与 Dialogic 公司 SS1 通道对接	135
1.16.6.4.3	中国 1 号信令的状态转移	136
1.16.6.4.4	输出呼叫过程中的调试信息	140
1.16.6.5	Line side 协议的状态机描述	141
1.16.6.6	SS1 通道的高级编程接口	142
1.16.6.7	No.1 信令的配置	143
1.16.7	连接到远端坐席接入模块	144
1.17	SHV 系列板卡 (CTI 系列)	145
1.17.1	SHV 系列板卡简介	145
1.17.2	SHV 系列板卡的典型应用原理框图	145
1.17.3	有关变声功能的函数一览表	145
1.18	SHN 系列板卡 (CTI 系列)	146
1.18.1	SHN 系列板卡简介	146

1.18.2	SHN-32A-CT/PCI 板卡的工作原理框图.....	146
1.18.3	SHN B 系列板卡工作原理框图.....	148
1.18.4	SHN C 系列板卡工作原理框图.....	149
1.18.5	SIP 通道的编程.....	149
1.18.5.1	SIP 编程基本概念.....	149
1.18.5.2	SIP 通道的状态转移图.....	151
1.18.5.3	SIP 通道的配置.....	154
1.18.6	IP 资源卡专用参数和结构说明.....	156
1.18.6.1	IP 资源卡 RTP 收发模式定义.....	156
1.18.6.2	IP 资源卡专用结构说明.....	156
1.18.7	IPC 型卡专用配置项说明.....	156
1.18.8	IPC 型卡的环境与使用.....	157
1.18.9	IPC 型卡的工作原理.....	158
1.19	DST 系列板卡 (REC 系列)	159
1.19.1	DST 系列板卡简介	159
1.19.2	DST 系列板卡支持的交换机型号	160
1.19.3	DST 系列板卡的工作原理框图	160
1.19.4	DST 系列板卡的配置	163
1.19.5	DST 系列板卡的语音操作	163
1.19.6	DST 系列板卡的信令编程	164
1.19.6.1	使用 D 信道的原始信令消息	164
1.20	ATP 系列板卡 (REC 系列)	165
1.20.1	ATP 系列板卡简介	165
1.20.2	ATP 系列板卡的工作原理框图	165
1.20.3	模拟中继线录音通道的状态迁移图	167
1.20.4	检测摘机/挂机状态	168
1.21	DTP 系列板卡 (REC 系列)	168
1.21.1	DTP 系列板卡简介	168
1.21.2	DTP 系列板卡的工作原理框图	169
1.21.3	术语介绍	171
1.21.4	DTP 系列板卡的语音功能	172
1.21.4.1	获取 DTMF	172
1.21.4.2	语音录制	172
1.21.5	驱动程序实现的状态机	173
1.21.5.1	数字中继线录音通道的通用状态机模型	173
1.21.5.2	获取呼叫信息	173
1.21.6	DTP 系列板卡的应用及配置实例	174
1.21.6.1	使用 No.1 信令/ISDN PRI 协议的系统	174
1.21.6.2	使用 SS7 信令的系统	175
1.22	SHF 系列板卡 (CTI 系列)	176
1.22.1	SHF 系列板卡简介	176
1.22.2	SHF 系列板卡的工作原理框图	176
1.23	IPR 系列产品 (REC 系列)	178
1.23.1	IPR 系列产品简介	178

1.23.2	<i>IPR</i> 系列产品的系统模型	179
1.23.3	<i>SynIPAnalyzer</i> 的相关概念	180
1.23.4	<i>SynIPAnalyzer</i> 的通道状态迁移图	180
1.23.5	<i>SynIPRecorder</i> 的相关概念	181
1.23.6	<i>SynIPRecorder</i> 的通道状态迁移图	183
1.23.7	<i>IPR</i> 系列产品的语音操作	183
1.23.8	<i>IPR</i> 系列产品的信令编程	184
1.23.8.1	使用 D 信道的原始信令消息	184
1.23.9	<i>IPR</i> 系列产品支持的协议类型及相关参数	184
1.23.9.1	IPR 通用的宏定义和结构体	184
1.23.9.2	支持的协议	185
1.23.9.3	协议配置结构体	185
1.23.9.3.1	SIP 协议的配置结构体	185
1.23.9.3.2	CISCO SCCP 协议的配置结构体	186
1.23.9.3.3	AVAYA H323 协议的配置结构体	186
1.23.9.3.4	SHORTEL MGCP 协议的配置结构体	186
1.23.9.3.5	H323 协议的配置结构体	186
1.23.9.3.6	PANASONIC MGCP 协议的配置结构体	187
1.23.9.3.7	TOSHIBA MEGACO 协议的配置结构体	187
1.23.9.3.8	Siemens H323 协议的配置结构体	187
1.23.9.3.9	Alcatel 协议的配置结构体	187
1.23.9.3.10	Mitel 协议的配置结构体	188
1.23.9.3.11	LG Nortel 协议的配置结构体	188
1.23.9.3.12	Samsung 协议的配置结构体	188
1.23.9.3.13	Tadicom MGCP 协议的配置结构体	188
1.23.9.3.14	Zenitel 协议的配置结构体	188
1.23.9.3.15	Nortel UNIStim 协议的配置结构体	189
1.23.9.4	使用 <i>SynIPAnalyzer</i> 与 <i>SynIPRecorder</i> 搭建的 VoIP 录音系统编程举例	189
1.24	HMP 系列产品 (CTI 系列)	195
1.24.1	HMP 系列产品简介	195
1.24.2	HMP 系列产品的系统架构	196
1.24.3	HMP 的配置与使用	196
1.24.4	HMP Client 专用配置项	197
1.24.5	HMP Server 专用配置项	197
1.24.6	HMP Server 与 HMP Client 配置使用范例	197
2	SynCTI 开发包函数说明	200
2.1	系统函数	200
2.1.1	驱动平台初始化函数	200
2.1.1.1	SsmStartCti	200
2.1.1.2	SsmStartCtiEx	201
2.1.1.3	SsmCloseCti	203
2.1.2	设置 API 和事件输出方式的函数	203
2.1.2.1	SsmSetLogEnable	203
2.1.2.2	SsmSetLogAttribute	205

2.1.2.3 SsmSetApiLogRange.....	205
2.1.2.4 SsmGetLogAttribute.....	206
2.1.3 有关定时器的函数	207
2.1.3.1 SsmStartTimer	207
2.1.3.2 SsmStopTimer	207
2.1.3.3 StartTimer	208
2.1.3.4 ElapseTime	208
2.1.4 获取系统错误信息的函数	209
2.1.4.1 SsmGetLastErrCode	209
2.1.4.2 SsmGetLastErrMsg	210
2.1.4.3 SsmGetLastErrMsgA	210
2.1.4.4 fBmp_GetErrMsg	210
2.1.5 获取板卡信息的函数.....	211
2.1.5.1 获取安装板卡的信息.....	211
2.1.5.1.1 SsmGetMaxCfgBoard	211
2.1.5.1.2 SsmGetMaxUsableBoard.....	211
2.1.5.1.3 GetTotalPciBoard	212
2.1.5.1.4 ReloadPciBoardInfo	212
2.1.5.2 获取板卡授权号.....	213
2.1.5.2.1 SsmGetAccreditId	213
2.1.5.2.2 SsmGetAccreditIdEx	213
2.1.5.3 获取板卡型号	213
2.1.5.3.1 SsmGetBoardModel	213
2.1.5.3.2 GetPciBoardModel	216
2.1.5.3.3 SsmGetBoardName	219
2.1.5.4 获取板卡序列号	222
2.1.5.4.1 SsmGetPciSerialNo	222
2.1.5.4.2 GetPciBoardSerialNo	222
2.1.5.5 获取文件的版本信息.....	223
2.1.5.5.1 SsmGetDllVersion	223
2.1.6 获取通道属性的函数.....	224
2.1.6.1 SsmGetMaxCh.....	224
2.1.6.2 SsmGetChType.....	224
2.1.6.3 SsmGetChHdInfo.....	225
2.1.6.4 SsmGetAppChId	226
2.1.7 设置通道属性的函数.....	226
2.1.7.1 SsmSetFlag	226
2.1.7.2 SsmGetFlag	228
2.2 有关事件方式编程的函数	229
2.2.1 获取事件的函数	229
2.2.1.1 SsmWaitForEvent	229
2.2.1.2 SsmWaitForEventA	229
2.2.1.3 SsmGetEvent.....	230
2.2.1.4 SsmGetEventA	230

2.2.2 过滤输出事件.....	231
2.2.2.1 SsmSetEvent	231
2.2.3 获取编程模式.....	235
2.2.3.1 SsmGetEventMode.....	235
2.2.4 通过驱动程序输出应用程序的自定义事件.....	236
2.2.4.1 SsmPutUserEvent.....	236
2.2.4.2 SsmPutUserEventA	236
2.3 有关呼叫状态机的函数.....	237
2.3.1 有关去话呼叫的函数 (CTI 系列)	237
2.3.1.1 搜索空闲通道	237
2.3.1.1.1 SsmSearchIdleCallOutCh	237
2.3.1.2 发起去话呼叫	238
2.3.1.2.1 SsmAutoDial	238
2.3.1.2.2 SsmAutoDialEx	238
2.3.1.2.3 SsmAppendPhoNum.....	241
2.3.1.2.4 SsmChkAutoDial	242
2.3.1.2.5 SsmGetAutoDialFailureReason	243
2.3.1.2.6 SsmSetTxCallerId	244
2.3.1.2.7 SsmGetTxCallerId.....	245
2.3.1.2.8 SsmSetTxOriginalCallerID	246
2.3.1.2.9 SsmSetTxRedirectingNum	247
2.3.1.2.10 SsmSetWaitAutoDialAnswerTime	247
2.3.1.2.11 SsmSetWaitAutoDialAnswerTimeEx	248
2.3.1.2.12 SsmGetWaitAutoDialAnswerTime	249
2.3.1.2.13 SsmGetKB	249
2.3.1.3 SS1 通道的专用函数	250
2.3.1.3.1 SsmSetKA.....	250
2.3.1.3.2 SsmSetKD.....	251
2.3.2 有关来话呼叫的函数.....	252
2.3.2.1 获取主叫用户信息	252
2.3.2.1.1 SsmChkOpCallerId	252
2.3.2.1.2 SsmGetCallerId.....	252
2.3.2.1.3 SsmGetCallerIdA	252
2.3.2.1.4 SsmGetCallerIdEx.....	252
2.3.2.1.5 SsmGetCallerName	253
2.3.2.1.6 SsmClearCallerId	254
2.3.2.1.7 SsmClearCallerIdEx	254
2.3.2.1.8 SsmGetKA	255
2.3.2.1.9 SsmGetKD	256
2.3.2.2 获取被叫用户信息	256
2.3.2.2.1 SsmGetPhoNumStr.....	256
2.3.2.2.2 SsmGetPhoNumStrA	256
2.3.2.2.3 SsmGetPhoNumLen	256
2.3.2.3 获取第 1 被叫号码信息	257

2.3.2.3.1 SsmGet1stPhoNumLen	257
2.3.2.3.2 SsmGet1stPhoNumStr	257
2.3.2.3.3 SsmGet1stPhoNumStrA	258
2.3.2.4 设置来话接续过程参数	259
2.3.2.4.1 SsmEnableAutoSendKB	259
2.3.2.4.2 SsmGetAutoSendKBFlag	259
2.3.2.4.3 SsmSetKB	260
2.3.3 通道状态迁移事件	262
2.3.3.1 SsmGetChState	262
2.3.3.2 SsmGetChStateKeepTime	264
2.3.4 获取 PendingReason	265
2.3.4.1 SsmGetPendingReason	265
2.3.5 通道摘机函数 (CTI 系列)	267
2.3.5.1 SsmPickup	267
2.3.5.2 SsmPickupANX	268
2.3.6 通道挂机函数 (CTI 系列)	269
2.3.6.1 SsmHangup	269
2.3.6.2 SsmHangupEx	269
2.3.7 设置通道的呼叫方向	270
2.3.7.1 SsmSetAutoCallDirection	270
2.3.7.2 SsmGetAutoCallDirection	271
2.3.8 获取释放原因	272
2.3.8.1 SsmGetReleaseReason	272
2.3.9 TUP 通道专用函数	272
2.3.9.1 SsmSetCalleeHoldFlag	272
2.3.9.2 SsmGetTupFlag	273
2.3.9.3 SsmSetTupParameter	274
2.3.9.4 SsmGetTupParameter	274
2.3.10 ISUP 通道专用函数	275
2.3.10.1 SsmSetISUPCAT	275
2.3.10.2 SsmSetIsupParameter	276
2.3.10.3 SsmGetIsupParameter	277
2.3.10.4 SsmSetIsupFlag	278
2.3.10.5 SsmGetIsupFlag	280
2.3.10.6 SsmGetIsupUPPara	280
2.3.10.7 SsmSetIsupUPPara	281
2.3.10.8 SsmSendIsupMsg	282
2.3.10.9 SsmGetRedirectionInfReason	283
2.3.10.10 SsmGetRedirectionInfNum	283
2.3.10.11 SsmlsHaveCpg	284
2.3.10.12 SsmGetCpg	284
2.3.10.13 SsmlsupGetUsr	285
2.3.10.14 SsmlsupSendUsr	285
2.3.11 ISDN 通道专用函数	286

2.3.11.1	设置 Setup 消息的参数.....	286
2.3.11.1.1	SsmISDNSetDialSubAddr	286
2.3.11.1.2	SsmISDNSetDialSubAddrEx	286
2.3.11.1.3	SsmISDNSetTxSubAddr	287
2.3.11.1.4	SsmISDNSetTxSubAddrEx	287
2.3.11.1.5	SsmISDNSetCallerIdPresent.....	287
2.3.11.1.6	SsmSetIsdnParameter	288
2.3.11.1.7	SsmSetIsdnParameterA	289
2.3.11.2	获取来话呼叫信息	290
2.3.11.2.1	获取子地址信息	290
2.3.11.2.1.1	SsmISDNGetSubAddr	290
2.3.11.2.1.2	SsmISDNGetCallerSubAddr	290
2.3.11.2.2	获取用户自定义主叫号码	291
2.3.11.2.2.1	Ssm GetUserCallerId	291
2.3.11.2.3	获取 SETUP 消息指定字段	292
2.3.11.2.3.1	SsmGetIsdnParameter	292
2.3.11.3	设置挂机原因	292
2.3.11.3.1	SsmISDNSetHangupRzn	292
2.3.11.4	其它函数.....	293
2.3.11.4.1	SsmISDNGetDisplayMsg	293
2.3.11.4.2	SsmISDNGetTxCallerSubAddr.....	294
2.3.11.4.3	SsmSetNumType.....	294
2.3.11.4.4	SsmGetNumType	295
2.3.11.4.5	SsmSetCharge	296
2.4	有关信号音发生器的函数（CTI 系列）	296
2.4.1	设置信号音发生器的工作参数	296
2.4.1.1	SsmSetTxTonePara	296
2.4.1.2	SsmQueryOpSendTone	297
2.4.1.3	SsmGetTxTonePara.....	297
2.4.2	发送信号音	298
2.4.2.1	SsmSendTone.....	298
2.4.2.2	SsmSendToneEx	298
2.4.2.3	SsmStopSendTone	299
2.4.2.4	SsmChkSendTone	299
2.5	有关信号音检测器的函数	300
2.5.1	有关信号音检测器的常用函数	300
2.5.1.1	设置工作状态	300
2.5.1.1.1	SsmStartToneAnalyze	300
2.5.1.1.2	SsmCloseToneAnalyze	301
2.5.1.1.3	SsmQueryOpToneAnalyze	301
2.5.1.2	获取检测结果	302
2.5.1.2.1	SsmGetToneAnalyzeResult.....	302
2.5.1.3	清除检测结果	303
2.5.1.3.1	SsmClearToneAnalyzeResult.....	303

2.5.2 有关 FFT 的函数.....	303
2.5.2.1 设置工作参数	303
2.5.2.1.1 SsmSetPeakFrqDetectBW	303
2.5.2.2 获取工作参数	304
2.5.2.2.1 SsmQueryOpPeakFrqDetect.....	304
2.5.2.2.2 SsmGetPeakFrqDetectBW.....	305
2.5.2.3 获取检测结果	305
2.5.2.3.1 SsmGetPeakFrq.....	305
2.5.2.3.2 SsmGetPeakFrqEnergy	306
2.5.2.3.3 SsmGetOverallEnergy	306
2.5.2.3.4 SsmGetOverallEnergyAllCh	306
2.5.3 设置噪声滤波器的参数	307
2.5.3.1 SsmSetMinVocDtrEnergy.....	307
2.5.3.2 SsmGetMinVocDtrEnergy	308
2.5.4 有关呼叫进程音检测器的函数	308
2.5.4.1 设置第 2 呼叫进程音检测器的工作状态	308
2.5.4.1.1 SsmStart2ndToneAnalyzer	308
2.5.4.1.2 SsmGet2ndToneAnalyzerState	309
2.5.4.2 获取第 2 呼叫进程音检测器的结果	309
2.5.4.2.1 SsmGet2ndToneAnalyzeResult.....	309
2.5.4.2.2 SsmClear2ndToneAnalyzeResult	310
2.5.4.3 设置频率检测器的参数	311
2.5.4.3.1 SsmSetTonePara.....	311
2.5.4.3.2 SsmSet2ndTonePara	311
2.5.4.3.3 SsmGetTonePara	312
2.5.4.3.4 SsmGet2ndTonePara	312
2.5.4.4 设置拨号音检测器的参数.....	313
2.5.4.4.1 SsmSetIsDialToneDtrTime	313
2.5.4.4.2 SsmSet2ndIsDialToneDtrTime	313
2.5.4.4.3 SsmGetIsDialToneDtrTime	313
2.5.4.4.4 SsmGet2ndIsDialToneDtrTime	313
2.5.4.5 设置回铃音检测器的参数.....	314
2.5.4.5.1 SsmSetRingEchoTonePara	314
2.5.4.5.2 SsmSet2ndRingEchoTonePara	314
2.5.4.5.3 SsmGetRingEchoToneTime	315
2.5.4.5.4 SsmGetRingEchoTonePara	315
2.5.4.5.5 SsmGet2ndRingEchoTonePara	315
2.5.4.6 设置忙音检测器的参数	316
2.5.4.6.1 SsmSetBusyTonePeriodEx	316
2.5.4.6.2 SsmSetBusyTonePeriod	316
2.5.4.6.3 SsmSet2ndBusyTonePeriod	316
2.5.4.6.4 SsmSetIsBusyToneDtrCnt	317
2.5.4.6.5 SsmSet2ndIsBusyToneDtrCnt	317
2.5.4.6.6 SsmGetBusyTonePeriodEx	318

2.5.4.6.7 SsmGetBusyTonePeriod	318
2.5.4.6.8 SsmGet2ndBusyTonePeriod	318
2.5.4.6.9 SsmGetIsBusyToneDtrCnt	319
2.5.4.6.10 SsmGet2ndIsBusyToneDtrCnt	319
2.5.4.7 获取检测结果	319
2.5.4.7.1 SsmGetBusyToneLen	319
2.5.4.7.2 SsmGet2ndBusyToneLen	320
2.5.4.7.3 SsmGetBusyToneCount	320
2.5.4.7.4 SsmGet2ndBusyToneCount	320
2.5.4.8 获取采用背靠背忙音信号检测算法的检测结果	321
2.5.4.8.1 SsmGetBusyToneEx	321
2.5.4.9 获取 Selcall Tone 信号检测算法的检测结果	321
2.5.4.9.1 SsmGetSelcallToneStr	321
2.5.4.9.2 SsmGetSelcallToneLen	322
2.5.4.9.3 SsmClearSelcallToneBuf	323
2.5.5 有关传真进程音检测器的函数	323
2.5.5.1 设置工作参数	323
2.5.5.1.1 SsmSetVoiceFxPara	323
2.5.5.1.2 SsmGetVoiceFxPara	324
2.5.5.2 获取检测结果	325
2.5.5.2.1 SsmGetVocFxFlag	325
2.5.6 有关机器应答检测器的函数	325
2.5.6.1 获取与处理机器应答检测结果	325
2.5.6.1.1 SsmGetAMDResult	325
2.5.6.1.2 SsmClearAMDResult	326
2.5.6.1.3 SsmControlAMD	326
2.5.6.1.4 SsmSetAMDPara	327
2.6 有关 DTMF 检测器的函数	328
2.6.1 启动/关闭 DTMF 检测器	328
2.6.1.1 SsmEnableRxDtmf	328
2.6.2 设置回调函数	329
2.6.2.1 SsmSetRxDtmfHandler	329
2.6.3 获取 DTMF 字符	329
2.6.3.1 SsmClearRxDtmfBuf	329
2.6.3.2 SsmGetDtmfStr	330
2.6.3.3 SsmGetDtmfStrA	330
2.6.3.4 SsmGetRxDtmfLen	330
2.6.3.5 SsmGet1stDtmf	331
2.6.3.6 SsmGet1stDtmfClr	331
2.6.3.7 SsmGetLastDtmf	332
2.6.4 使用 WaitDtmf 任务	332
2.6.4.1 SsmSetWaitDtmf	332
2.6.4.2 SsmSetWaitDtmfEx	332
2.6.4.3 SsmSetWaitDtmfExA	332

2.6.4.4 SsmSetWaitDtmfExB	332
2.6.4.5 SsmCancelWaitDtmf	333
2.6.4.6 SsmChkWaitDtmf.....	334
2.7 有关 DTMF 发生器的函数 (CTI 系列)	335
2.7.1 设置 DTMF 发生器的参数	335
2.7.1.1 SsmSetTxDtmfPara	335
2.7.1.2 SsmQueryTxDtmf	335
2.7.1.3 SsmGetTxDtmfPara	336
2.7.2 启动 DTMF 发生任务.....	336
2.7.2.1 SsmTxDtmf	336
2.7.2.2 SsmStopTxDtmf	337
2.7.2.3 SsmChkTxDtmf.....	338
2.8 有关 Barge-in 检测器的函数	338
2.8.1 设置 Barge-in 检测器的参数	338
2.8.1.1 SsmSetBargeInSens.....	338
2.8.1.2 SsmSetIsBargeInDtrmTime	339
2.8.1.3 SsmSetVoiceEnergyMinValue.....	339
2.8.1.4 SsmSetNoSoundDtrmTime.....	340
2.8.1.5 SsmGetBargeInSens	340
2.8.1.6 SsmGetIsBargeInDtrmTime	341
2.8.1.7 SsmGetVoiceEnergyMinValue	341
2.8.1.8 SsmGetNoSoundDtrmTime	342
2.8.2 获取 Barge-in 检测器的检测结果.....	342
2.8.2.1 SsmDetectBargeIn.....	342
2.8.2.2 SsmDetectNoSound	343
2.8.2.3 SsmGetNoSoundTime	343
2.9 有关放音操作的函数	344
2.9.1 设置放音数据的去向.....	344
2.9.1.1 SsmSetPlayDest.....	344
2.9.2 设置放音的音量	345
2.9.2.1 SsmSetPlayVolume	345
2.9.2.2 SsmSetPlayGain.....	345
2.9.3 设置播放任务的终止条件	345
2.9.3.1 SsmSetDtmfStopPlay.....	345
2.9.3.2 SsmSetDTMFStopPlay CharSet	346
2.9.3.3 SsmGetDtmfStopPlayFlag	347
2.9.3.4 SsmGetDTMFStopPlay CharSet.....	347
2.9.3.5 SsmSetBargeinStopPlay	348
2.9.3.6 SsmGetBargeinStopPlayFlag.....	348
2.9.3.7 SsmSetHangupStopPlayFlag.....	349
2.9.4 播放单个文件的专用函数	349
2.9.4.1 SsmPlayFile	349
2.9.4.2 SsmStopPlayFile.....	351
2.9.4.3 SsmPausePlay.....	351

2.9.4.4	SsmRestartPlay	352
2.9.4.5	SsmFastFwdPlay	352
2.9.4.6	SsmFastBwdPlay	353
2.9.4.7	SsmSetPlayTime	353
2.9.4.8	SsmSetPlayPrct	354
2.9.4.9	SsmGetDataBytesToPlay	354
2.9.4.10	SsmGetDataBytesPlayed	355
2.9.4.11	SsmGetPlayedTimeEx	355
2.9.4.12	SsmGetPlayingFileInfo	356
2.9.4.13	SsmPlayFileW	356
2.9.5	播放文件序列的函数	357
2.9.5.1	SsmClearFileList	357
2.9.5.2	SsmAddToFileList	357
2.9.5.3	SsmPlayFileList	358
2.9.5.4	SsmStopPlayFileList	359
2.9.6	播放单个内存的函数	359
2.9.6.1	SsmPlayMem	359
2.9.6.2	SsmStopPlayMem	361
2.9.6.3	SsmSetStopPlayOffset	361
2.9.6.4	SsmGetPlayOffset	362
2.9.7	播放指定内存序列的函数	362
2.9.7.1	SsmAddToPlayMemList	362
2.9.7.2	SsmClearPlayMemList	363
2.9.7.3	SsmPlayMemList	363
2.9.7.4	SsmStopPlayMemList	364
2.9.8	播放预加载语音的函数 (CTI 系列)	365
2.9.8.1	预加载语音信息的初始化函数	365
2.9.8.1.1	SsmSetMaxIdxSeg	365
2.9.8.1.2	SsmGetTotalIndexSeg	365
2.9.8.1.3	SsmLoadIndexData	366
2.9.8.1.4	SsmFreeIndexData	367
2.9.8.1.5	SsmLoadChIndexData	367
2.9.8.1.6	SsmFreeChIndexData	368
2.9.8.2	播放预加载语音的函数	369
2.9.8.2.1	SsmPlayIndexString	369
2.9.8.2.2	SsmPlayIndexList	369
2.9.8.2.3	SsmStopPlayIndex	370
2.9.9	播放乒乓缓存的函数 (CTI 系列)	370
2.9.9.1	SsmPlayMemBlock	370
2.9.9.2	SsmStopPlayMemBlock	373
2.9.10	终止播放任务的通用函数	373
2.9.10.1	SsmStopPlay	373
2.9.11	获取播放任务的相关信息	374
2.9.11.1	SsmQueryPlayFormat	374

2.9.11.2 SsmGetPlayType	374
2.9.11.3 SsmCheckPlay.....	375
2.9.11.4 SsmGetPlayedPercentage.....	376
2.9.11.5 SsmGetPlayedTime	376
2.10 有关录音操作的函数.....	377
2.10.1 设置录音操作的参数	377
2.10.1.1 设置信号源及其音量.....	377
2.10.1.1.1 SsmSetRecVolume	377
2.10.1.1.2 SpySetRecVolume	377
2.10.1.1.3 SsmSetRecMixer	378
2.10.1.1.4 SpySetRecMixer.....	378
2.10.1.1.5 SsmQueryOpRecMixer	379
2.10.1.1.6 SsmGetRecMixerState	379
2.10.1.1.7 DTRSetMixerVolume.....	380
2.10.1.1.8 DTRGetMixerVolume	381
2.10.1.1.9 SsmSetRecBack	381
2.10.1.1.10 SsmSetNoModuleChBusRec	382
2.10.1.2 设置录音任务的终止条件	383
2.10.1.2.1 SsmSetDTMFStopRecCharSet	383
2.10.1.2.2 SsmGetDTMFStopRecCharSet.....	383
2.10.1.2.3 SsmSetHangupStopRecFlag.....	384
2.10.1.3 设置预录音功能（REC 系列）	385
2.10.1.3.1 SsmSetPrerecord.....	385
2.10.1.3.2 SsmGetPrerecordState	385
2.10.1.4 设置录音文件截尾功能	386
2.10.1.4.1 SsmSetTruncateTail	386
2.10.1.4.2 SsmGetTruncateTailTime	387
2.10.1.5 设置录音 AGC.....	387
2.10.1.5.1 SsmSetRecAGC	387
2.10.1.5.2 SsmGetRecAGCSwitch.....	388
2.10.1.6 设置录音立体声开关.....	389
2.10.1.6.1 SsmSetRecStereo.....	389
2.10.2 获取录音操作的信息	389
2.10.2.1 SsmQueryRecFormat	389
2.10.2.2 SsmGetRecType.....	390
2.10.2.3 SsmGetRecTime.....	391
2.10.2.4 SsmCheckRecord	391
2.10.3 录制到文件的函数	392
2.10.3.1 SsmRecToFile	392
2.10.3.2 SsmRecToFileA	392
2.10.3.3 SsmRecToFileB	392
2.10.3.4 SsmRecToFileEx.....	392
2.10.3.5 SpyRecToFileA.....	392
2.10.3.6 SpyRecToFileB.....	392

2.10.3.7	SpyRecToFile	392
2.10.3.8	SsmStopRecToFile.....	396
2.10.3.9	SpyStopRecToFile	396
2.10.3.10	SsmPauseRecToFile	397
2.10.3.11	SsmRestartRecToFile	397
2.10.3.12	SsmChkRecToFile	398
2.10.3.13	SsmGetDataBytesToRecord	399
2.10.3.14	SsmRecStereoToFile	399
2.10.3.15	SsmStopRecStereoToFile	400
2.10.3.16	SsmChkStereoToFile	400
2.10.3.17	SsmPauseRecToFileEx	401
2.10.3.18	SsmRecToFileW	401
2.10.3.19	SsmRecToFileAW	401
2.10.3.20	SsmRecToFileBW	402
2.10.3.21	SsmRecToFileExW	402
2.10.4	录制到指定内存的函数.....	402
2.10.4.1	SsmRecToMem.....	402
2.10.4.2	SpyRecToMem.....	402
2.10.4.3	SsmStopRecToMem	403
2.10.4.4	SpyStopRecToMem.....	403
2.10.4.5	SsmGetRecOffset	404
2.10.5	录制到指定乒乓缓存的函数 (CTI 系列)	405
2.10.5.1	SsmRecordMemBlock	405
2.10.5.2	SsmStopRecordMemBlock	407
2.11	有关交换总线的函数	408
2.11.1	双向连接函数 (CTI 系列)	408
2.11.1.1	建立双向连接	408
2.11.1.1.1	SsmTalkWith.....	408
2.11.1.1.2	SsmTalkWithEx.....	408
2.11.1.2	拆除双向连接	409
2.11.1.2.1	SsmStopTalkWith.....	409
2.11.2	单向连接函数	409
2.11.2.1	建立单向连接	409
2.11.2.1.1	SsmListenTo.....	409
2.11.2.1.2	SsmListenToEx.....	410
2.11.2.1.3	SsmLinkFrom	410
2.11.2.1.4	SsmLinkFromEx	410
2.11.2.1.5	SsmLinkFromAllCh.....	410
2.11.2.2	拆除单向连接	412
2.11.2.2.1	SsmStopListenTo.....	412
2.11.2.2.2	SsmStopLinkFrom	412
2.11.2.2.3	SsmUnLinkFromAllCh	412
2.11.3	获取总线信息函数	413
2.11.3.1	SsmGetChBusInfo	413

2.11.4 高级总线操作函数 (CTI 系列)	414
2.11.4.1 强制拆除通道的所有总线连接	414
2.11.4.1.1 SsmClearChBusLink	414
2.11.4.2 通道到总线时隙的单向连接	415
2.11.4.2.1 SsmLinkToBus	415
2.11.4.2.2 SsmUnLinkToBus	415
2.11.4.3 总线时隙到通道的单向连接	416
2.11.4.3.1 SsmLinkFromBus	416
2.11.4.3.2 SsmLinkFromBusEx	416
2.11.4.3.3 SsmUnLinkFromBus	417
2.11.4.3.4 SsmListenToPlay	417
2.11.4.3.5 SsmUnListenToPlay	418
2.12 有关板载音频功率放大器的函数	418
2.12.1 SsmQueryOpPowerAmp	418
2.12.2 SsmSetPowerAmpVlm	419
2.12.3 SetVolume	419
2.13 有关板载扬声器的函数	420
2.13.1 SsmSetLine0OutTo	420
2.14 有关回波抵消器 (Echo Canceller) 的函数 (CTI 系列)	421
2.14.1 设置回波抵消器的工作状态	421
2.14.1.1 SsmQueryOpEchoCanceller	421
2.14.1.2 SsmSetEchoCanceller	421
2.14.1.3 SsmGetEchoCancellerState	422
2.14.2 设置回波抵消器的工作参数	422
2.14.2.1 SsmSetEchoCancelDelaySize	422
2.14.2.2 SsmGetEchoCancelDelaySize	423
2.15 有关电话会议的函数 (CTI 系列)	423
2.15.1 会议室初始化函数	423
2.15.1.1 SsmCreateConfGroup	423
2.15.1.2 SsmFreeConfGroup	424
2.15.2 会议室成员操作函数	425
2.15.2.1 SsmJoinConfGroup	425
2.15.2.2 SsmExitConfGroup	426
2.15.2.3 SsmSetListenVlmInConf	426
2.15.2.4 SsmSetContactInConf	427
2.15.3 获取会议室信息的函数	427
2.15.3.1 SsmGetConfCfgInfo	427
2.15.3.2 SsmGetTotalConfGroup	428
2.15.3.3 SsmGetConfGrpCfgInfo	428
2.15.3.4 SsmGetConfGrpInfo	429
2.15.3.5 SsmGetConfGrpId	429
2.15.3.6 SsmValidateGrpId	430
2.15.4 获取会议室成员信息的函数	430
2.15.4.1 SsmGetConfGrpMmbrId	430

2.15.4.2 SsmGetConfGrpMmbrInfo	431
2.15.4.3 SsmGetConfChInfo.....	431
2.16 有关 FSK 收发器的函数 (CTI 系列)	432
2.16.1 FSK 数据发送器.....	432
2.16.1.1 SsmSetFskPara	432
2.16.1.2 SsmGetFskPara.....	433
2.16.1.3 SsmTransFskData	433
2.16.1.4 SsmStartSendFSK	434
2.16.1.5 SsmCheckSendFsk	435
2.16.1.6 SsmStopSendFsk	435
2.16.2 FSK 数据接收器.....	436
2.16.2.1 SsmStartRcvFSK	436
2.16.2.2 SsmStartRcvFSK_II	437
2.16.2.3 SsmStartRcvFSK_III	438
2.16.2.4 SsmCheckRcvFSK	440
2.16.2.5 SsmStopRcvFSK	441
2.16.2.6 SsmClearRcvFSKBuf.....	441
2.16.2.7 SsmGetRcvFSK.....	442
2.17 有关电压检测器的函数	442
2.17.1 忽略电压检测结果.....	442
2.17.1.1 SsmSetIgnoreLineVoltage.....	442
2.17.1.2 SsmGetIgnoreLineVoltage	443
2.17.2 设置检测摘机/挂机动作的门限电压.....	444
2.17.2.1 SsmSetDtrmLineVoltage	444
2.17.2.2 SsmGetDtrmLineVoltage.....	444
2.17.3 获取电压检测器的输出结果.....	445
2.17.3.1 SsmGetLineVoltage	445
2.17.4 有关极性反转检测器的函数.....	445
2.17.4.1 SsmQueryOpPolarRvrs	445
2.17.4.2 SsmGetPolarRvrsCount.....	446
2.17.5 有关铃流信号检测器的函数.....	446
2.17.5.1 SsmGetRingCount	446
2.17.5.2 SsmGetCallBackRingCount	447
2.17.5.3 SsmClearRingCount	447
2.17.5.4 SsmGetRingFlag.....	448
2.18 有关数字中继线信息的函数	449
2.18.1 SsmGetMaxPcm.....	449
2.18.2 SsmGetPcmInfo.....	449
2.18.3 SsmGetPcmLinkStatus	450
2.18.4 SsmPcmTsToCh	452
2.18.5 SsmChToPcmTs	452
2.18.6 SsmSetPcmClockMode	453
2.18.7 SsmGetInboundLinkSet.....	453
2.18.8 SsmGetCbChStatus.....	454

2.18.9 SsmSetPcmPowerDown.....	454
2.19 SHT 系列板卡专用函数 (CTI 系列)	455
2.19.1 设置联合模块工作模式.....	455
2.19.1.1 SsmSetUnimoduleState	455
2.19.2 模拟中继通道专用函数.....	456
2.19.2.1 在模拟线路上产生闪断信号.....	456
2.19.2.1.1 SsmTxFlash	456
2.19.2.1.2 SsmSetTxFlashCharTime	456
2.19.2.1.3 SsmGetTxFlashCharTime	457
2.19.2.1.4 SsmChkTxFlash.....	457
2.19.2.1.5 SsmTxFlashEx	458
2.19.2.2 通道状态的强制迁移.....	458
2.19.2.2.1 SsmSetChState	458
2.19.2.3 去话接续时检测被叫摘机的函数.....	459
2.19.2.3.1 SsmStartPickupAnalyze	459
2.19.2.3.2 SsmSetCalleeHookDetectP	460
2.19.2.3.3 SsmGetPickup	460
2.19.2.4 检测通道摘机执行情况的函数	461
2.19.2.4.1 SsmCheckActualPickup	461
2.19.2.5 检测通道是否为模拟中继转成录音通道的函数	461
2.19.2.5.1 SsmGetIsAnalogToRec	461
2.19.3 坐席通道专用函数.....	462
2.19.3.1 向电话机发送振铃信号	462
2.19.3.1.1 SsmStartRing	462
2.19.3.1.2 SsmStartRingWithCIDStr.....	462
2.19.3.1.3 fPcm_ConvertFskCID.....	463
2.19.3.1.4 SsmStopRing	463
2.19.3.1.5 SsmCheckSendRing	464
2.19.3.1.6 SsmSetRingPeriod	465
2.19.3.2 检测到话机摘机后自动送拨号音	465
2.19.3.2.1 SsmSetASDT	465
2.19.3.2.2 SsmGetASDT	466
2.19.3.3 检测话机的闪断信号	466
2.19.3.3.1 SsmSetLocalFlashTime	466
2.19.3.3.2 SsmGetFlashCount.....	467
2.19.3.3.3 SsmClearFlashCount	467
2.19.3.4 检测话机的摘机/挂机动作	468
2.19.3.4.1 SsmGetHookState	468
2.19.3.5 在电话线上产生极性反转信号	469
2.19.3.5.1 SsmSetPolarState	469
2.19.3.5.2 SsmGetPolarState	469
2.19.4 有关远端摘机检测器的函数 (SHT 系列 <i>only</i>)	470
2.19.4.1 有关普通远端摘机检测器的函数.....	470
2.19.4.1.1 设置工作参数	470

2.19.4.1.1.1 SsmSetVoiceOnDetermineTime	470
2.19.4.1.2 获取工作参数.....	471
2.19.4.1.2.1 SsmGetVoiceOnDetermineTime	471
2.20 SHD 系列板卡的高级编程 API (CTI 系列)	471
2.20.1 通道的闭塞控制 (CTI 系列)	471
2.20.1.1 闭塞本端的函数.....	471
2.20.1.1.1 闭塞单个通道.....	471
2.20.1.1.1.1 SsmBlockLocalCh	471
2.20.1.1.1.2 SsmUnblockLocalCh	472
2.20.1.1.1.3 SsmQueryLocalChBlockState.....	473
2.20.1.1.2 闭塞整条数字中继线	473
2.20.1.1.2.1 SsmBlockLocalPCM	473
2.20.1.1.2.2 SsmUnblockLocalPCM	474
2.20.1.1.2.3 SsmQueryLocalPCMBLOCKSTATE	474
2.20.1.2 闭塞对端的函数.....	475
2.20.1.2.1 闭塞单个通道.....	475
2.20.1.2.1.1 SsmQueryOpBlockRemoteCh	475
2.20.1.2.1.2 SsmBlockRemoteCh.....	476
2.20.1.2.1.3 SsmUnblockRemoteCh	476
2.20.1.2.1.4 SsmGetRemoteChBlockStatus	477
2.20.1.2.2 闭塞整条数字中继线	478
2.20.1.2.2.1 SsmBlockRemotePCM	478
2.20.1.2.2.2 SsmUnblockRemotePCM	479
2.20.1.2.2.3 SsmGetRemotePCMBLOCKSTATE	480
2.20.2 基于 SS7-MTP3 的高级编程 API	480
2.20.2.1 SsmSendSs7Msu	480
2.20.2.2 SsmGetSs7Msu	481
2.20.2.3 SsmGetMtp3State	482
2.20.2.4 SsmGetMtp3StateEx	482
2.20.2.5 SsmGetMtp2Status	482
2.20.2.6 SsmSendSs7MsuEx	483
2.20.2.7 SsmGetMaxSs7link	484
2.20.2.8 SsmGetSs7Mtp2Msu	484
2.20.2.9 SsmSendSs7Mtp2Msu	485
2.20.2.10 SsmSs7Mtp2CmdCtrl	485
2.20.2.11 SsmGetDecodeSs7Msu	486
2.20.3 有关 ISDN 信令的高级函数	487
2.20.3.1 SsmGetIsdnMsu	487
2.20.3.2 SsmSendIsdnMsu	488
2.20.3.3 SsmCheckIsdnMsu	488
2.20.3.4 SsmISDNGetStatus	489
2.20.3.5 SsmGetUserInfo	489
2.20.3.6 SsmSetUserInfo	490
2.20.3.7 SsmISDNGetProgressMsg	490

2.20.3.8 SsmGetLsdnMsuEx	491
2.20.3.9 SsmSendLsdnMsuEx.....	491
2.20.4 有关中国1号信令的高级函数回	492
2.20.4.1 控制ABCD信令比特.....	492
2.20.4.1.1 SsmGetCAS.....	492
2.20.4.1.2 SsmSendCAS	493
2.20.4.1.3 SsmGetSendingCAS.....	493
2.20.4.1.4 SsmSetSendCASFlag.....	494
2.20.4.1.5 SsmGetSendCASFlag.....	495
2.20.4.2 控制R2信号收发器	495
2.20.4.2.1 SsmSetRxR2Mode.....	495
2.20.4.2.2 SsmGetR2.....	496
2.20.4.2.3 SsmSendR2	496
2.20.4.2.4 SsmSendR2Ex.....	497
2.20.4.2.5 SsmStopSendR2.....	498
2.20.4.2.6 SsmGetSendingR2	498
2.21 SHV系列板卡的专用函数(CTI系列)	499
2.21.1 SsmGetMaxVCh.....	499
2.21.2 SsmGetMaxFreeVCh.....	499
2.21.3 SsmBindVCh	500
2.21.4 SsmUnBindVCh.....	501
2.21.5 SsmSetVoiceEffect	501
2.21.6 SsmGetVoiceEffect	502
2.21.7 SsmSetVoiceEffectEx	502
2.21.8 SsmGetVoiceEffectEx.....	503
2.22 SHN系列板卡的专用函数(CTI系列)	504
2.22.1 SIP通道的专用编程接口.....	504
2.22.1.1 SsmIpGetSessionCodecType	504
2.22.1.2 SsmIpSetForwardNum.....	504
2.22.1.3 SsmIpInitiateTransfer	505
2.22.1.4 SsmIpGetMessageField	505
2.22.1.5 SsmIPGetMsgFieldStr.....	506
2.22.1.6 SsmSipMsgSetHeader.....	507
2.22.1.7 SsmSipMsgSetHeaderA	508
2.22.1.8 SsmSipStackRegister	508
2.22.1.9 SsmSipBindChWithRegInfo	509
2.22.1.10 SsmSipUnBindChWithOutRegInfo.....	510
2.22.1.11 SsmSipStackUnRegister	510
2.22.1.12 SsmSipStackRemoveRegister	511
2.22.1.13 SsmSipGetRegInfo	511
2.22.1.14 SsmSipSetTxUserName	512
2.22.1.15 SsmSetIpFlag	513
2.22.1.16 SsmSipGetReferStatus	513
2.22.1.17 SsmSipSendRequest.....	514

2.22.1.18	SsmSipSendRequestA	514
2.22.1.19	SsmSetHangupReason	515
2.22.1.20	SsmSipGetBoardRegStatus.....	516
2.22.1.21	SsmSipGetChRegStatus.....	516
2.22.1.22	SsmSetRcvRegisterCallBack.....	517
2.22.1.23	SsmIpGetBoardMacAddress	517
2.22.1.24	SsmSipRegister	518
2.22.1.25	SsmSipSetConnectionInforOfSDP.....	519
2.22.1.26	SsmSipSetMsgFieldParameter	520
2.22.1.27	SsmSipChHold	520
2.22.1.28	SsmSipChTransfer.....	521
2.22.1.29	SsmSipSetConnectionInforOfSDPEX	521
2.22.1.30	SsmSipSetContactSection.....	522
2.22.1.31	SsmSipChEnableRtpStun	522
2.22.1.32	SsmSipOutCallSendOptions.....	523
2.22.1.33	SsmSipChEnableSrtp	523
2.22.1.34	SsmSipSetMultiNetIP	524
2.22.1.35	SsmSipOutCallSendNotify	524
2.22.1.36	SsmSipSetTransportProtocol.....	525
2.22.1.37	SsmSipChHoldA	525
2.22.1.38	SsmSipChTransferA	525
2.22.2	<i>IP 资源卡的专用函数接口和说明</i>	526
2.22.2.1	SsmLockMediaCh.....	526
2.22.2.2	SsmGetMediaChParam	527
2.22.2.3	SsmOpenMediaCh.....	527
2.22.2.4	SsmCloseMediaCh	528
2.22.2.5	SsmUnlockMediaCh	528
2.22.2.6	SsmUpdateMediaCh.....	529
2.22.2.7	SsmCheckMediaChRTPTimeOut.....	529
2.22.3	<i>NAT 穿透专用接口函数</i>	530
2.22.3.1	SsmIPGetStunPublicIP	530
2.22.4	<i>SHN B 系列卡的专用函数</i>	531
2.22.4.1	SsmCheckBoardIcmp	531
2.22.4.2	SsmSipStackAddWhiteOrBlackList	531
2.22.4.3	SsmSipStackRemoveWhiteOrBlackList	532
2.22.5	<i>IP 卡注册服务系统专用函数</i>	532
2.22.5.1	SsmAutoDialAgent.....	532
2.22.5.2	SsmSipRegResponse	533
2.22.5.3	SsmSipGetUserInfoByIndex	534
2.22.5.4	SsmGetCallNum	535
2.22.5.5	SsmSipStackSetUserAgent.....	535
2.23	<i>ATP 系列录音卡专用函数（REC 系列）</i>	536
2.23.1	<i>麦克风通道的输入增益控制</i>	536
2.23.1.1	SsmQueryOpMicGain	536

2.23.1.2 SsmSetMicGain	536
2.23.1.3 SsmGetMicGain.....	537
2.23.2 有关 MF 检测器的函数.....	538
2.23.2.1 SsmEnableRxMF	538
2.23.2.2 SsmClearRxMFBuf	538
2.23.2.3 SsmGetMFStr	539
2.24 DTP 系列板卡的专用函数（REC 系列）	539
2.24.1 基于状态机编程模式的函数.....	539
2.24.1.1 获取基本信息的函数.....	539
2.24.1.1.1 SpyGetMaxCic	539
2.24.1.1.2 SpyChToCic.....	540
2.24.1.2 获取呼叫进展的函数.....	541
2.24.1.2.1 SpyGetState	541
2.24.1.3 获取呼叫方向的函数.....	541
2.24.1.3.1 SpyGetCallInCh.....	541
2.24.1.3.2 SpyGetCallOutCh.....	542
2.24.1.4 取得主叫方号码和被叫方号码的函数.....	542
2.24.1.4.1 SpyGetCallerId	542
2.24.1.4.2 SpyGetCalleeId	542
2.24.1.5 获取呼叫挂机信息的函数	543
2.24.1.5.1 SpyGetHangupInfo.....	543
2.24.1.6 取得主叫方号码类型和被叫方号码类型的函数	544
2.24.1.6.1 SpyGetCallerType	544
2.24.1.6.2 SpyGetCalleeType	544
2.24.1.7 获取数字中继线状态的函数	544
2.24.1.7.1 SpyGetLinkStatus	544
2.24.1.8 获取 SS7 信令呼叫状态机同步输出的原始信令消息.....	545
2.24.1.8.1 SsmGetSs7SpyMsu	545
2.24.1.9 取得连接号码的函数.....	546
2.24.1.9.1 SpyGetConId.....	546
2.24.1.10 获取 ISDN 第 2 层（L2）监控原始消息的函数	546
2.24.1.10.1 SsmGetIsdnL2SpyMsu.....	546
2.25 DST 系列录音卡的专用函数（REC 系列）	547
2.25.1 SsmGetDstChSNRofUplink	547
2.25.2 SsmGetDstChSNRofDownlink.....	547
2.25.3 SsmGetDstChVoltageState	548
2.26 PCM1280E 录音卡的专用函数（REC 系列）	548
2.26.1 SsmGetPcm32LineState	548
2.27 IPR 系列录音产品的专用函数（REC 系列）	549
2.27.1 SynIPRRecorder 的专用函数	549
2.27.1.1 SsmIPRSetRecVolume	549
2.27.1.2 SsmIPRSetMixerType	550
2.27.1.3 SsmIPRGetMixerType	550
2.27.1.4 SsmIPRGetRecSlaverCount.....	551

2.27.1.5 SsmIPRGetRecSlaverList.....	551
2.27.1.6 SsmIPRGetRecSlaverInfo.....	552
2.27.1.7 SsmIPRStartRecSlaver	552
2.27.1.8 SsmIPRCloseRecSlaver	553
2.27.1.9 SsmIPRActiveSession	554
2.27.1.10 SsmIPRDeActiveSession.....	555
2.27.1.11 SsmIPRActiveAndRecToFile	555
2.27.1.12 SsmIPRDeActiveAndStopRecToFile	556
2.27.1.13 SsmGetUSBKeySerial	557
2.27.1.14 SsmlsBoardIPR	557
2.27.1.15 SsmIPRConnectToSlaver	558
2.27.2 相关函数: SynIPAnalyzer 的专用函数.....	558
2.27.2.1 SsmIPRAddProtocol	558
2.27.2.2 SsmIPRRmvProtocol	559
2.27.2.3 SsmIPRGetProtocol	560
2.27.2.4 SsmIPRGetStationCount.....	560
2.27.2.5 SsmIPRGetStationList	561
2.27.2.6 SsmIPRGetStationInfo	561
2.27.2.7 SsmIPRGetStationInfoEx.....	562
2.27.2.8 SsmIPRSendSession.....	562
2.27.2.9 SsmIPRStopSendSession	563
2.27.2.10 SsmIPRGetSessionInfo	563
2.27.2.11 SsmIPRGetMonitorType	564
2.27.2.12 SsmIPRSetMonitorType.....	564
2.27.2.13 SsmIPRAddStationToMap.....	565
2.27.2.14 SsmIPRAddStationToMapEx	566
2.27.2.15 SsmIPRRmvStationFromMap	566
2.27.2.16 SsmIPRRmvStationFromMapEx	567
2.27.2.17 SsmIPRChkForward	568
2.27.2.18 SsmIPRGetCallInfo.....	568
2.27.2.19 SsmIPRGetMessageField.....	569
2.27.3 IPR 系列录音卡用到的结构体	569
2.27.3.1 IPR_Addr	569
2.27.3.2 IPR_SLAVERADDR	570
2.27.3.3 IPR_SessionInfo	570
2.27.3.4 COMPUTER_INFO	570
2.27.3.5 IPR_MONITOR_CFGS	570
2.27.3.6 STATION_LIST	570
2.27.3.7 IPR_CALL_INFO	571
2.27.3.8 IPR_CALL_INFOEX.....	571
2.27.3.9 StationInfoEx.....	571
2.28 有关语音编码格式转换的函数	572
2.28.1 fPcm_Mp3ConvertALaw	572
2.28.2 fPcm_Mp3ConvertULaw.....	572

2.28.3	fPcm_AdpcmToAlaw.....	573
2.28.4	fPcm_AdpcmToGsm.....	573
2.28.5	fPcm_AdpcmToMp3.....	573
2.28.6	fPcm_AlawToUlaw.....	574
2.28.7	fPcm_UlawToAlaw.....	574
2.28.8	fPcm_MemAdpcmToALAW	574
2.28.9	fPcm_MemAdpcmToULAW	574
2.28.10	fPcm_ALawConvertPcm16.....	575
2.28.11	fPcm_ALawConvertPcm16_16K.....	575
2.28.12	fPcm_ALawConvertPcm8.....	575
2.28.13	fPCM_AlawConvertGC8.....	575
2.28.14	fPcm_ALawConvertMp3.....	575
2.28.15	fPcm_ULawConvertMp3.....	576
2.28.16	fPcm_GetDtmfAndMskFromMp3.....	577
2.28.17	fPcm_ALawToVox.....	578
2.28.18	fPcm_MemAlawToPcm8.....	579
2.28.19	fPcm_MemAlawToPcm16.....	579
2.28.20	fPcm_MemAlawToPcm16_16K.....	579
2.28.21	fPcm_MemGSMToPcm16.....	580
2.28.22	fPcm_MemGSMToPcm8	581
2.28.23	fPcm_MemGSMToUlaw	581
2.28.24	fPcm_Pcm16ConvertALaw.....	582
2.28.25	fPcm_Pcm16_16KConvertALaw	582
2.28.26	fPcm_MemPcm8ToAlaw.....	583
2.28.27	fPcm_MemAlawToUlaw.....	584
2.28.28	fPcm_MemUlawtoAlaw.....	584
2.28.29	fPcm_MemPcm16ToAlaw.....	584
2.28.30	fPcm_MemPcm16_16KToAlaw	585
2.28.31	fPcm_GC8Convert.....	586
2.28.32	fPcm_Vox6KTo8K.....	586
2.28.33	fPcm_Vox8KTo6K.....	586
2.28.34	fPcm_ULawConvertGSM.....	586
2.28.35	fPCM_Pcm16ConvertG729A.....	587
2.28.36	fPcm_Pcm8ConvertGSM.....	588
2.28.37	fPcm_GSMToMp3.....	588
2.28.38	fPcm_G729AConvert.....	589
2.28.39	fPCM_MemULawToG729A	589
2.28.40	fPcm_MemMp3ToPcm16	590
2.28.41	fPcm_MemG729AToPcm8.....	591
2.28.42	fPcm_MemG729AToPcm16.....	591
2.28.43	fPcm_GetLastErrMsg	591
2.28.44	fPcm_Close	592
2.28.45	fPcm_NotchFilter_ULAW.....	592
2.28.46	fPcm_ALawToAdpcm.....	593

2.28.47 fPcm_InitEx.....	594
2.28.48 fPcm_MemPcm16ToGSMEx	594
2.29 有关传真操作的函数（CTI 系列）	595
2.29.1 设置传真操作的工作参数	595
2.29.1.1 SsmFaxSetChSpeed.....	595
2.29.1.2 SsmFaxSetMaxSpeed	595
2.29.1.3 SsmFaxSetID.....	596
2.29.2 发送传真.....	596
2.29.2.1 SsmFaxStartSend	596
2.29.2.2 SsmFaxStartSendEx	597
2.29.2.3 SsmFaxSendMultiFile	597
2.29.2.4 SsmFaxSendMultiFileEx	597
2.29.2.5 SsmFaxAppendSend	599
2.29.3 接收传真.....	599
2.29.3.1 SsmFaxStartReceive	599
2.29.4 终止传真任务	600
2.29.4.1 SsmFaxStop.....	600
2.29.5 获取传真操作的信息	601
2.29.5.1 SsmFaxGetSpeed.....	601
2.29.5.2 SsmFaxCheckEnd	601
2.29.5.3 SsmFaxGetDcnTag	602
2.29.5.4 SsmFaxGetChStateMsg.....	602
2.29.5.5 SsmFaxGetPages	603
2.29.5.6 SsmFaxGetFailReason	603
2.29.5.7 SsmFaxGetID	604
2.29.5.8 SsmFaxGetMode	605
2.29.5.9 SsmFaxGetAllBytes	606
2.29.5.10 SsmFaxGetSendBytes	606
2.29.5.11 SsmFaxGetRcvBytes	606
2.29.5.12 SsmFaxGetCodeMode	607
2.29.6 .tif 文件的相关操作	607
2.29.6.1 fBmp_ValidateFaxFile	607
2.29.6.2 fBmp_SetHeaderFormat	608
2.29.6.3 fBmp_AddTxtToTif.....	609
2.29.6.4 fBmp_AddTxtToTif_Big	609
2.29.6.5 fBmp_UniteTif	610
2.29.6.6 fBmp_CutTifHeader	611
2.29.6.7 fBmp_GetFileAllPage.....	611
2.29.6.8 fBmp_SetHeaderFormatA.....	612
3 SynCTI 驱动程序的配置	613
3.1 系统配置文件 ShConfig.ini.....	613
3.1.1 必须配置项	613
3.1.1.1 设置板卡总数	613
3.1.1.1.1 TotalBoards	613

3.1.1.1.2 WhoSupplySysClock.....	613
3.1.1.2 设置通道编号	614
3.1.1.2.1 TotalAppCh.....	614
3.1.1.2.2 AppCh	614
3.1.1.3 设置板卡信息	615
3.1.1.3.1 BoardModel.....	615
3.1.1.3.2 BoardSerialNumber.....	615
3.1.1.4 SHD/DTP 系列板卡	615
3.1.1.4.1 设置板卡复位功能.....	615
3.1.1.4.1.1 ResetBoardOnClose.....	615
3.1.1.4.2 设置数字中继线的工作参数	616
3.1.1.4.2.1 PcmNumber.....	616
3.1.1.4.2.2 PcmSSx.....	616
3.1.1.4.2.3 PcmClockMode	616
3.1.1.4.2.4 PcmLinkType	616
3.1.1.4.2.5 IsdnAutoBuildLink.....	616
3.1.1.4.2.6 CRC-4.....	616
3.1.1.4.2.7 loopback	618
3.1.1.4.2.8 SpySyncAndCCS.....	619
3.1.1.4.2.9 framing.....	621
3.1.1.4.2.10 coding.....	621
3.1.1.4.2.11 syncc.....	621
3.1.1.4.3 设置 PCM 的逻辑编号	622
3.1.1.4.3.1 TotalPcm.....	622
3.1.1.4.3.2 Pcm	622
3.1.1.4.3.3 2PcmIn1Line.....	623
3.1.1.4.4 设置 SpyPCM 编号 (DTP 系列)	624
3.1.1.4.4.1 TotalSpyPcm.....	624
3.1.1.4.4.2 SpyPcm	624
3.1.1.4.4.3 SpyT1TransE1Line	625
3.1.1.4.4.4 SpyDefaultGetCallerIdType	625
3.1.1.4.5 设置 SpyCIC 编号 (DTP 系列)	626
3.1.1.4.5.1 TotalAppSpyCIC	626
3.1.1.4.5.2 AppSpyCIC	626
3.1.1.4.6 SS7 信令的必须配置项 (DTP 系列)	627
3.1.1.4.6.1 TotalSpyLinkSet	627
3.1.1.4.6.2 SpyLinkSet	627
3.1.1.4.6.3 TotalSpyLinkPcm.....	627
3.1.1.4.6.4 SpyLinkPcm.....	627
3.1.1.4.6.5 SpyCICPcm	627
3.1.1.4.6.6 SpySpCodeLen.....	627
3.1.1.4.6.7 SpyIsupCICPcm	627
3.1.1.4.7 SS7 信令的必须配置项 (SHD 系列)	629
3.1.1.4.7.1 设置使用 ISUP 协议的数字中继线.....	629

3.1.1.4.7.1.1	TotalIlsupPcm.....	629
3.1.1.4.7.1.2	IlsupPcm.....	629
3.1.1.4.8	ISDN 信令必须配置项 (SHD 系列)	630
3.1.1.4.8.1	设置运行模式	630
3.1.1.4.8.1.1	UseISDNMode	630
3.1.1.4.8.2	设置使用 ISDN 信令(用户侧)的数字中继线	630
3.1.1.4.8.2.1	TotalUserLinker	630
3.1.1.4.8.2.2	UserPcmLink.....	630
3.1.1.4.8.2.3	UserSendEstablish.....	630
3.1.1.4.8.2.4	UserTEIValue	630
3.1.1.4.8.3	设置使用 ISDN 信令(网络侧)的数字中继线	631
3.1.1.4.8.3.1	TotalNetLinker	631
3.1.1.4.8.3.2	NetPcmLink.....	631
3.1.1.4.8.3.3	NetSendEstablish.....	631
3.1.1.4.8.3.4	NetTEIValue	631
3.1.1.4.9	SS1 信令的必须配置项 (SHD 系列)	632
3.1.1.4.9.1	ProtocolType.....	632
3.1.1.5	SHN 系列板卡	633
3.1.1.5.1	SHN 系列卡通用配置.....	633
3.1.1.5.1.1	LocalSipIp	633
3.1.1.5.1.2	LocalSipPort	633
3.1.1.5.1.3	SipSetConnectionIp.....	633
3.1.1.5.2	SHN A 系列卡必须配置项	634
3.1.1.5.2.1	设置协议类型	634
3.1.1.5.2.1.1	ProtocolType	634
3.1.1.5.2.2	SIP 信令的必须配置项	634
3.1.1.5.2.2.1	RTPRange	634
3.1.1.5.2.2.2	AudioCodecList.....	634
3.1.1.5.2.2.3	SendDtmfType	634
3.1.1.5.2.2.4	RecvDtmfType	635
3.1.1.5.3	SHN B 系列卡必须配置项.....	636
3.1.1.5.3.1	SIP 信令的必须配置项	636
3.1.1.5.3.1.1	BoardIP	636
3.1.1.5.3.1.2	SubMask	636
3.1.1.5.3.1.3	Gateway	636
3.1.1.5.3.1.4	DNS	636
3.1.1.5.3.1.5	AudioCodecList.....	636
3.1.1.5.3.1.6	RTPRange	636
3.1.1.5.3.1.7	SendSipMsg.....	636
3.1.1.5.3.1.8	SendDtmfType	636
3.1.1.5.3.1.9	RecvDtmfType	637
3.1.1.5.3.1.10	TelephoneEventsPt	638
3.1.1.5.3.1.11	Send180After183	638
3.1.1.5.4	SHN C 系列卡必须配置项	638

3.1.1.5.4.1 SHN C 系列卡专有配置项	638
3.1.1.5.4.1.1 OctMac0.....	638
3.1.1.5.4.1.2 DhcpServer	639
3.1.1.5.4.1.3 ProcessorCtrlMac.....	639
3.1.1.5.4.1.4 FilterMacRange	639
3.1.1.5.4.1.5 BootFileName	639
3.1.1.5.4.1.6 LogLevel	640
3.1.1.5.4.1.7 DHCPRange	640
3.1.1.5.4.2 SIP 信令的必须配置项	640
3.1.1.5.4.2.1 BoardIP	640
3.1.1.5.4.2.2 SubMask	640
3.1.1.5.4.2.3 Gateway	640
3.1.1.5.4.2.4 DNS	640
3.1.1.5.4.2.5 AudioCodecList.....	640
3.1.1.5.4.2.6 RTPRange	641
3.1.1.5.4.2.7 SendSipMsg.....	641
3.1.1.5.4.2.8 SendDtmfType	641
3.1.1.5.4.2.9 RecvDtmfType	642
3.1.1.5.4.2.10 TelephoneEventsPt	643
3.1.1.6 DST 系列板卡 (REC 系列 only)	643
3.1.1.6.1 PBXType	643
3.1.1.6.2 PhoneType	643
3.1.1.6.3 DstRecRawData.....	644
3.1.1.6.4 SetAnalogCtrlEnable.....	644
3.1.1.6.5 AnalogCtrl	645
3.1.1.6.6 SetVoxChSelectEnable	646
3.1.1.6.7 VoxChSelect.....	646
3.1.1.6.8 SetFilterSwitchEnable	647
3.1.1.6.9 FilterSwitch.....	647
3.1.1.6.10 SetVoltageReferenceEnable	647
3.1.1.6.11 VoltageReference	648
3.1.1.7 IPR 系列产品必须配置项 (REC 系列 only)	648
3.1.1.7.1 SynIPRecorder 系列产品 Master 端的必须配置项	648
3.1.1.7.1.1 RecMasterIP	648
3.1.1.7.1.2 RecMasterPort	649
3.1.1.7.1.3 RTPTimeOut	649
3.1.1.7.2 SynIPAnalyzer 系列产品的必须配置项	649
3.1.1.7.2.1 MonitorNIC	649
3.1.1.7.2.2 ForwardIP	649
3.1.1.7.2.3 ForwardPort	650
3.1.1.7.2.4 MonitorType	650
3.1.1.7.2.5 RTPTimeOut	650
3.1.1.7.2.6 RtpFwdCtrl.....	651
3.1.1.7.2.7 ThreadPairs	651

3.1.1.7.2.8	IGMPEventEnable	651
3.1.1.8	HMP 系列产品必须配置项 (CTI 系列 only)	652
3.1.1.8.1	HMP 系列产品 Server 端的必须配置项	652
3.1.1.8.1.1	TotalMediaForward	652
3.1.1.8.1.2	LocallIP[n]	652
3.1.1.8.1.3	TotalCh[n]	652
3.1.1.8.1.4	LocalPort[n]	653
3.1.1.8.1.5	RemotelP[n]	653
3.1.1.8.1.6	RemotePort[n]	653
3.1.1.8.1.7	RtpIP[n]	653
3.1.1.8.1.8	MaxMediaThread	654
3.1.2	常用配置项	654
3.1.2.1	设置 SHD/DTP 系列板卡的协议	654
3.1.2.1.1	CardType	654
3.1.2.2	设置驱动程序抛出的事件	655
3.1.2.2.1	DefaultEventOutput	655
3.1.2.2.2	OvrlEnrgEventOut	655
3.1.2.3	设置驱动程序内部事件队列的深度	655
3.1.2.3.1	MaxEventPerChannel	655
3.1.2.3.2	MaxUserEventSize	656
3.1.2.4	设置状态机的通用参数	656
3.1.2.4.1	MaxWaitAutoDialAnswerTime	656
3.1.2.4.2	RingToPending	657
3.1.2.4.3	AutoClearCallerIdBufOnHangup	657
3.1.2.4.4	AutomaticAisGeneration	657
3.1.2.4.5	AutomaticRaiGereration	658
3.1.2.4.6	IgnoreBlockInGra	658
3.1.2.4.7	AllowTimeoutInSpyISDN	659
3.1.2.5	设置数字线路上的语音编码格式	659
3.1.2.5.1	DefaultVoiceFormat	659
3.1.2.6	设置信号音检测器	660
3.1.2.6.1	设置数字卡的信号音检测开关	660
3.1.2.6.1.1	DefaultToneCheckState	660
3.1.2.6.2	设置噪声滤波器的工作参数	660
3.1.2.6.2.1	MinimumVoiceDetermineEnergy	660
3.1.2.6.3	设置脉宽滤波器的工作参数 T	660
3.1.2.6.3.1	ToneHighFilterPoint	660
3.1.2.6.3.2	ToneLowFilterPoint	661
3.1.2.6.4	设置第 1 呼叫进程音检测器	661
3.1.2.6.4.1	设置频率检测器的工作参数	661
3.1.2.6.4.1.1	TonePara	661
3.1.2.6.4.2	设置拨号音检测器的工作参数	662
3.1.2.6.4.2.1	IsDialToneDetermineTime	662
3.1.2.6.4.3	设置回铃音检测器的工作参数	662

3.1.2.6.4.3.1	RingEchoTonePara	662
3.1.2.6.4.4	设置忙音检测器的工作参数.....	662
3.1.2.6.4.4.1	BusyTonePeriod	662
3.1.2.6.4.4.2	IsBusyToneDetermineCount	662
3.1.2.6.4.5	设置 Kewl Start 功能的工作参数.....	663
3.1.2.6.4.5.1	EnableKewlStart.....	663
3.1.2.6.4.5.2	KSVoltageThreshold	663
3.1.2.6.4.5.3	KSKeepTime	664
3.1.2.6.4.6	设置自定义信号音检测器的工作参数	664
3.1.2.6.4.6.1	AppointedToneAnalyzerSwitch.....	664
3.1.2.6.4.6.2	IsAppointedToneDetermineTime	664
3.1.2.6.4.6.3	AppointedTonePara.....	665
3.1.2.6.4.6.4	IsAppointedToneDetermineCount	665
3.1.2.6.5	设置第 2 呼叫进程音检测器	665
3.1.2.6.5.1	设置工作状态	665
3.1.2.6.5.1.1	Enable2ndToneAnalyzer	665
3.1.2.6.5.1.2	Check2ndToneOnAutoDial	666
3.1.2.6.5.2	设置频率检测器的工作参数.....	666
3.1.2.6.5.2.1	2ndTonePara.....	666
3.1.2.6.5.3	设置拨号音检测器的工作参数	666
3.1.2.6.5.3.1	2ndIsDialToneDetermineTime	666
3.1.2.6.5.4	设置回铃音检测器的工作参数	667
3.1.2.6.5.4.1	2ndRingEchoTonePara	667
3.1.2.6.5.5	设置忙音检测器的工作参数.....	667
3.1.2.6.5.5.1	2ndBusyTonePeriod	667
3.1.2.6.5.5.2	2ndIsBusyToneDetermineCount	667
3.1.2.6.5.5.3	MaxBsTnOffTime	668
3.1.2.6.5.6	设置自定义信号音检测器的工作参数	668
3.1.2.6.5.6.1	2ndAppointedToneAnalyzerSwitch.....	668
3.1.2.6.5.6.2	2ndIsAppointedToneDetermineTime	668
3.1.2.6.5.6.3	2ndAppointedTonePara.....	669
3.1.2.6.5.6.4	2ndIsAppointedToneDetermineCount	669
3.1.2.6.6	设置传真进程音检测器	669
3.1.2.6.6.1	VoiceFreqF1Para	669
3.1.2.6.6.2	VoiceFreqF2Para	670
3.1.2.6.7	设置简易忙音检测器	670
3.1.2.6.7.1	设置工作状态	670
3.1.2.6.7.1.1	ToneAnalyzeAtRcvFsk	670
3.1.2.6.7.2	设置噪声滤波器的工作参数.....	671
3.1.2.6.7.2.1	TAARFDetermineEnergy	671
3.1.2.6.8	设置回铃音检测方式	671
3.1.2.6.8.1	设置回铃音检测方式	671
3.1.2.6.8.1.1	IsEchoQuickDetect.....	671
3.1.2.6.8.2	设置检测回铃音高电平时允许的误差	671

3.1.2.6.8.2.1 EchoOnTolerance.....	671
3.1.2.6.8.3 设置检测回铃音低电平时允许的误差	672
3.1.2.6.8.3.1 EchoOffTolerance.....	672
3.1.2.6.9 设置 CCIR 信号音检测器	672
3.1.2.6.9.1 CCIREnableDetector	672
3.1.2.6.10 设置 SelcallTone 信号音检测器.....	673
3.1.2.6.10.1 SelcallToneDetectMode	673
3.1.2.6.10.2 SelcallToneType	673
3.1.2.6.10.3 SelcallTonePara.....	674
3.1.2.6.11 设置背靠背忙音检测器工作参数	674
3.1.2.6.11.1 ExToneLevel	674
3.1.2.7 设置机器应答检测器.....	675
3.1.2.7.1 EnableAMD	675
3.1.2.7.2 AMDNoSoundAfterDialTime.....	675
3.1.2.7.3 AMDNoSoundTime	675
3.1.2.7.4 AMDTimeOut.....	675
3.1.2.7.5 AMDToneCount.....	675
3.1.2.7.6 AMDTOn	676
3.1.2.7.7 AMDTOff	676
3.1.2.7.8 AMDTimeA.....	676
3.1.2.7.9 AMDTimeB.....	676
3.1.2.7.10 AMDTimeC.....	676
3.1.2.7.11 AMDTimeD	677
3.1.2.7.12 AMDSilentEnergy	677
3.1.2.7.13 EnableAMDBeep.....	677
3.1.2.7.14 AMDTOffEx	677
3.1.2.8 设置增强信号音检测器	677
3.1.2.8.1 ToneDetectorMode	677
3.1.2.8.2 VoiceOffDetermineTime	678
3.1.2.8.3 MaxToneDetectorItem	678
3.1.2.8.4 ToneDetectorItem.....	678
3.1.2.9 设置信号音发生器(CTI 系列)	681
3.1.2.9.1 DefaultSendToneFrequence	681
3.1.2.9.2 DefaultSendToneVolume	681
3.1.2.10 设置电话会议参数(CTI 系列)	682
3.1.2.10.1 ClearInVoiceOnRxDtmf	682
3.1.2.10.2 ClearInVoiceOnRx450Hz	682
3.1.2.10.3 ConfMaxGroup	683
3.1.2.10.4 ConfDefaultMaxGroupMember	683
3.1.2.10.5 ConfDefaultMaxGroupSpeaker	683
3.1.2.10.6 ConfDefaultMaxGroupSpeaking	684
3.1.2.10.7 ConfJoinedbyEnergy	684
3.1.2.10.8 ConfMaxListener	684
3.1.2.10.9 ConfDefaultMaxSilenceTime	685

3.1.2.10.10 BackgroundVoicePriority	685
3.1.2.10.11 PlayVoiceIsListen	685
3.1.2.10.12 QuitDynKeepingInConf	686
3.1.2.11 设置放音操作	686
3.1.2.11.1 设置播放任务的终止条件	686
3.1.2.11.1.1 DefaultDtmfStopPlay	686
3.1.2.11.1.2 DtmfStopPlayCharSet	686
3.1.2.11.1.3 HangupStopPlay	687
3.1.2.11.1.4 DefaultBargeInStopPlay	688
3.1.2.11.2 设置驱动程序内部的放音缓冲区	688
3.1.2.11.2.1 PlayBufSize	688
3.1.2.11.3 设置文件序列播放任务的参数	689
3.1.2.11.3.1 MaxPlayFileList	689
3.1.2.11.4 设置预加载文件播放任务的参数	689
3.1.2.11.4.1 MaxPlayIndexList	689
3.1.2.11.5 设置内存序列播放任务的参数	689
3.1.2.11.5.1 MaxPlayMemList	689
3.1.2.11.6 设置文件播放任务的参数	690
3.1.2.11.6.1 FastPlayTime	690
3.1.2.11.7 设置播放操作的音量	690
3.1.2.11.7.1 DefaultPlayVolume	690
3.1.2.11.8 设置播放操作的编码格式	690
3.1.2.11.8.1 DefaultPlayFormat	690
3.1.2.11.9 设置播放操作的其它参数	691
3.1.2.11.9.1 DefaultPausePlayOnRxDtmf	691
3.1.2.11.9.2 PlayFilterFlag	692
3.1.2.12 设置录音操作	692
3.1.2.12.1 设置录音任务的终止条件	692
3.1.2.12.1.1 DtmfStopRecCharSet	692
3.1.2.12.1.2 HangupStopRec	693
3.1.2.12.2 设置录音信号源	693
3.1.2.12.2.1 DefaultRecordVolume	693
3.1.2.12.2.2 DefaultRecordMixerVolume	693
3.1.2.12.3 设置文件录音的截尾功能	694
3.1.2.12.3.1 TruncateTailOnRecordToFile	694
3.1.2.12.4 设置录音操作的编码格式	695
3.1.2.12.4.1 DefaultRecordFormat	695
3.1.2.12.5 设置驱动程序内部的录音缓冲区	695
3.1.2.12.5.1 RecordBufSize	695
3.1.2.12.6 设置录音文件为 wav 格式	695
3.1.2.12.6.1 RecAsWavFormat	695
3.1.2.12.6.2 Mp3IsOnlyRead	696
3.1.2.12.7 设置 AGC 的工作参数	696
3.1.2.12.7.1 AGCMAXGAIN	696

3.1.2.12.7.2 AGCMINGAIN	696
3.1.2.12.7.3 AGCMAXGAINUPLINK	696
3.1.2.12.7.4 AGCMINGAINUPLINK.....	696
3.1.2.12.7.5 AGCMAXGAINDOWNLINK.....	696
3.1.2.12.7.6 AGCMINGAINDOWNLINK	697
3.1.2.12.7.7 AGCMAXLEVEL.....	697
3.1.2.12.7.8 AGCMINLEVEL	697
3.1.2.12.7.9 AGCMAXLEVELUPLINK.....	697
3.1.2.12.7.10 AGCMINLEVELUPLINK	698
3.1.2.12.7.11 AGCMAXLEVELDOWNLINK.....	698
3.1.2.12.7.12 AGCMINLEVELDOWNLINK.....	698
3.1.2.12.7.13 AGCDOWNRATIO.....	699
3.1.2.12.7.14 AGCUPRATIO	699
3.1.2.12.7.15 AGCDOWNRATIOUPLINK	699
3.1.2.12.7.16 AGCUPRATIOUPLINK	699
3.1.2.12.7.17 AGCDOWNRATIOODOWNLINK.....	699
3.1.2.12.7.18 AGCUPRATIOODOWNLINK.....	699
3.1.2.12.7.19 AGCKEEPTIME.....	699
3.1.2.12.7.20 OpenRecEnAndPlayEnOnIdle	700
3.1.2.12.7.21 AutoRecAgcSwitch	700
3.1.2.12.8 设置录音操作的其它参数	701
3.1.2.12.8.1 RecEliDTDurTalking	701
3.1.2.12.9 设置 MP3 格式录音的比特率	701
3.1.2.12.9.1 Mp3EncodingBitRate	701
3.1.2.13 设置录音/放音的共用参数	702
3.1.2.13.1 设置乒乓缓冲区的最小值	702
3.1.2.13.1.1 RecordAndPlayUseAsIP	702
3.1.2.13.2 设置乒乓缓存放音的模式	702
3.1.2.13.2.1 BlockPlayType	702
3.1.2.13.3 设置软件编码的 GSM 格式	702
3.1.2.13.3.1 GsmCodecEnable	702
3.1.2.14 设置 DTMF 检测器	703
3.1.2.14.1 启动/关闭 DTMF 检测器	703
3.1.2.14.1.1 AlwaysEnableRxDtmf	703
3.1.2.14.1.2 RcvDtmfOnIdle	703
3.1.2.14.2 设置 DTMF 检测器的缓冲区大小	704
3.1.2.14.2.1 RxDtmfBufSize	704
3.1.2.14.3 设置 DTMF 滤波器的参数	704
3.1.2.14.3.1 DualAndAllFreqEnScale	704
3.1.2.14.3.2 HighAndLowFreqEnScale.....	705
3.1.2.14.3.3 DtmfEnergy.....	705
3.1.2.14.3.4 DTMFFreqOffset.....	706
3.1.2.14.3.5 DtmfModel	706
3.1.2.14.4 设置 DTMF 脉宽滤波器的参数	707

3.1.2.14.4.1 ReceiveDtmfSensitive.....	707
3.1.2.14.4.2 OmitABCD	708
3.1.2.14.4.3 DtrmOnDtmfHighLevel.....	708
3.1.2.15 设置 DTMF 发生器(CTI 系列).....	709
3.1.2.15.1 设置发送缓冲区的大小	709
3.1.2.15.1.1 TxDtmfBufSize.....	709
3.1.2.15.2 设置 DTMF 发生器的参数.....	709
3.1.2.15.2.1 TxDtmfTimePara.....	709
3.1.2.15.2.2 DefaultTxDelayTime	710
3.1.2.15.2.3 TxDtmfAmp.....	710
3.1.2.16 设置回波抵消器(CTI 系列)	710
3.1.2.16.1 设置工作状态	710
3.1.2.16.1.1 EnableEchoCancellor	710
3.1.2.16.2 设置工作参数.....	711
3.1.2.16.2.1 EchoCancelDelaySize	711
3.1.2.17 设置 Barge-in 检测器 (CTI 系列).....	711
3.1.2.17.1 设置 Barge in 检测器的启动方式	711
3.1.2.17.1.1 AlwaysDetectBargeln	711
3.1.2.17.2 设置 Barge in 检测器的工作参数	712
3.1.2.17.2.1 VoiceEnergyMinValue.....	712
3.1.2.17.2.2 BargelnSensitive	712
3.1.2.17.2.3 BargelnDtrmTime	713
3.1.2.17.2.4 IsNoSoundDtrmTime	714
3.1.2.17.2.5 DefaultDtmflsSound.....	714
3.1.2.18 设置 TDM 总线连接.....	715
3.1.2.18.1 InVoiceToBus	715
3.1.2.18.2 LinkFromStopPlayAndTone	715
3.1.2.18.3 DefaultSpeakVolume	716
3.1.2.18.4 EnableCommonTimeSlot	716
3.1.2.19 设置模拟电话线的 Caller ID 检测器 (SHT+ATP 系列)	716
3.1.2.19.1 设置 Caller ID 检测器的工作模式	716
3.1.2.19.1.1 CallerIdStyle	716
3.1.2.19.2 设置 FSK 制式的工作参数	717
3.1.2.19.2.1 CloseCallerIdOnReceived	717
3.1.2.19.2.2 FSKCallerIdDtrmTime.....	717
3.1.2.19.2.3 FilterInvalidCID	718
3.1.2.19.2.4 FSKMinGate	718
3.1.2.19.2.5 FskNoPhase	719
3.1.2.19.3 设置 DTMF 制式的工作参数	719
3.1.2.19.3.1 DtmfCallerIDStyleLength	719
3.1.2.19.3.2 DtmfCallerIDInterTimeOut	719
3.1.2.20 设置铃流信号检测器 (SHT+ATP 系列)	720
3.1.2.20.1 RingDetectFilterPara.....	720
3.1.2.20.2 CallBackRingDetectFilterPara.....	720

3.1.2.20.3	RingEndDtrTime	721
3.1.2.20.4	AlwaysToRingingOnRingCntX	721
3.1.2.20.5	ChToRingingOnRingCnt	722
3.1.2.21	设置 FSK 数据收发器(CTI 系列)	722
3.1.2.21.1	设置 FSK 数据发送器的工作参数	722
3.1.2.21.1.1	FreqBit0	722
3.1.2.21.1.2	FreqBit1	722
3.1.2.21.1.3	Baudrate	722
3.1.2.21.1.4	MdlAmp	723
3.1.2.22	设置 FSK 数据接收器(CTI 系列)	723
3.1.2.22.1	FskMarkSignal	723
3.1.2.22.2	FskFrameMode	724
3.1.2.22.3	FskEchoCancelDelay	724
3.1.2.23	设置传真资源通道的工作参数	725
3.1.2.23.1	MaxSpeed	725
3.1.2.23.2	RcvDisTime	725
3.1.2.23.3	ResendForRTN	725
3.1.2.23.4	ResetRcvForRTN	726
3.1.2.23.5	KeepPageForRTN	726
3.1.2.23.6	PercentageForRTN	726
3.1.2.23.7	FaxCodeMode	727
3.1.2.23.8	FaxSendDisTime	727
3.1.2.24	设置电压检测器	727
3.1.2.24.1	忽略电压检测器的检测结果	727
3.1.2.24.1.1	IgnoreLineVoltage	727
3.1.2.24.2	设置振铃检测门限值	728
3.1.2.24.2.1	JudgeLineVoltage	728
3.1.2.24.3	设置摘机/挂机判决条件	728
3.1.2.24.3.1	IsHangupDtrmVoltage	728
3.1.2.24.3.2	LineOncnt	729
3.1.2.24.4	设置断线监测的判定条件	729
3.1.2.24.4.1	OffLineSet	729
3.1.2.24.4.2	OffLineDetermineTime	729
3.1.2.24.5	设置极性反转信号是否影响通道的状态迁移	730
3.1.2.24.5.1	DisablePolarReverse	730
3.1.2.24.6	设置极性反转的线路干扰忽略电压	730
3.1.2.24.6.1	PolarIgnore	730
3.1.2.25	设置 USB 录音盒/语音盒中的内置板载扬声器	731
3.1.2.25.1	USBLine0Output	731
3.1.2.26	SHT 系列板卡常用配置项(CTI 系列)	731
3.1.2.26.1	模拟中继通道专用配置项	731
3.1.2.26.1.1	设置模拟中继通道为录音通道	731
3.1.2.26.1.1.1	SetAnalogChToRecCh	731
3.1.2.26.1.2	设置去话呼叫的参数	732

3.1.2.26.1.2.1	MaxWaitDialToneTime	732
3.1.2.26.1.3	设置模拟中继通道是否开启校准.....	732
3.1.2.26.1.3.1	AdjustImmediately.....	732
3.1.2.26.2	设置闪断信号及保持时间.....	732
3.1.2.26.2.1	DefaultTxFlashChar.....	732
3.1.2.26.2.2	DefaultTxFlashTime.....	733
3.1.2.26.2.3	设置闪断后是否保持通道状态.....	733
3.1.2.26.2.3.1	AfterFlashNotAffectChState	733
3.1.2.26.2.4	远端摘机检测器.....	733
3.1.2.26.2.4.1	设置普通的远端摘机检测器的工作参数	733
3.1.2.26.2.4.1.1	WaitAfterDialTime	733
3.1.2.26.2.4.1.2	MaxWaitVocAfterEcho	734
3.1.2.26.2.4.1.3	VoiceOnDetermineTime.....	734
3.1.2.26.2.4.1.4	EchoNoVoiceToTalking	734
3.1.2.26.2.4.2	设置增强的远端摘机检测器	735
3.1.2.26.2.4.2.1	RelativeEngyHookDetect.....	735
3.1.2.26.2.4.2.2	HookEngyConfigMulti	735
3.1.2.26.2.4.2.3	HookValidEngyCnt.....	735
3.1.2.26.3	坐席通道专用配置项	736
3.1.2.26.3.1	AutoSendDialTone.....	736
3.1.2.26.3.2	StopSendDialToneOnDtmf.....	737
3.1.2.26.3.3	MaxLocalFlashTime.....	737
3.1.2.26.3.4	UserOnHookFilterTime	738
3.1.2.26.3.5	UserChGenerateRingMode	738
3.1.2.26.3.6	UserSendPolar	738
3.1.2.26.3.7	LocalHookFilterTime	739
3.1.2.26.4	传真通道专用配置项	739
3.1.2.26.4.1	MaxFaxChannel	739
3.1.2.26.4.2	DSP3WORKMODE	740
3.1.2.26.5	联合模块的专用配置项	740
3.1.2.26.5.1	UnimoduleState	740
3.1.2.26.6	无业务模块通道的专用配置项	741
3.1.2.26.6.1	NoModuleChBusRec	741
3.1.2.26.7	磁石通道	741
3.1.2.26.7.1	IsMagnetModule	741
3.1.2.27	SHD 系列板卡常用配置项(CTI 系列)	742
3.1.2.27.1	设置来话呼叫的收号规则	742
3.1.2.27.1.1	DefaultRcvPhoNumLen	742
3.1.2.27.1.2	DefaultRcvCallerID	742
3.1.2.27.1.3	RcvPhoNumCfgLen	742
3.1.2.27.1.4	MaxPhoNumRule	743
3.1.2.27.1.5	Rule	743
3.1.2.27.1.6	MfcLenCtrlPos	744
3.1.2.27.1.7	MfcLengthTable	744

3.1.2.27.1.8 CallerIdEnTable	744
3.1.2.27.2 设置“自动应答来话呼叫”功能	744
3.1.2.27.2.1 AutoSendKB	744
3.1.2.27.2.2 AutoSendACM	744
3.1.2.27.2.3 UserSideAutoSendAck	745
3.1.2.27.2.4 NetSideAutoSendAck	745
3.1.2.27.3 设置去话呼叫的主叫号码	745
3.1.2.27.3.1 TxCallerId	745
3.1.2.27.3.2 CalloutCallerId	745
3.1.2.27.3.3 SetSTSsignal	746
3.1.2.27.4 设置去话呼叫的被叫号码	746
3.1.2.27.4.1 SetCalledSTSsignal	746
3.1.2.27.5 连接到远端坐席接入模块	747
3.1.2.27.5.1 UsageMode	747
3.1.2.27.5.2 CBProtocolType	749
3.1.2.27.5.3 CBChannelType	749
3.1.2.27.5.4 CBChangeChannelType	750
3.1.2.27.6 设置板卡工作模式	750
3.1.2.27.6.1 RunAsSpy	750
3.1.2.27.7 SS1 信令的高级设置	751
3.1.2.27.7.1 选择国家	751
3.1.2.27.7.1.1 mfcr2_Protocol	751
3.1.2.27.7.2 设置接续过程的 R2 参数	753
3.1.2.27.7.2.1 tonesgroupA	753
3.1.2.27.7.2.2 tonesgroupB	753
3.1.2.27.7.2.3 Tonesendofinfo	754
3.1.2.27.7.2.4 Tonesanswer	754
3.1.2.27.7.2.5 Tonesrepeatrequest	754
3.1.2.27.7.2.6 TonesAnswerA	755
3.1.2.27.7.3 设置接续过程的基本工作参数	755
3.1.2.27.7.3.1 TxCas_CD	755
3.1.2.27.7.3.2 RxCASFilterTime	756
3.1.2.27.7.3.3 MaxWaitMfcTime	756
3.1.2.27.7.3.4 RxR2FilterTime	756
3.1.2.27.7.3.5 LSRingTimeout	757
3.1.2.27.7.4 设置状态机中通道的工作方式	757
3.1.2.27.7.4.1 EnableAutoCall	757
3.1.2.27.7.4.2 AutoCallInTimeSlot	757
3.1.2.27.7.5 设置中国 1 号信令状态机的参数	758
3.1.2.27.7.5.1 MfcKB	758
3.1.2.27.7.5.2 MaxWaitSetKBTime	758
3.1.2.27.7.5.3 MaxWaitKDTIME	758
3.1.2.27.7.5.4 PcmSyncMask	758
3.1.2.27.7.5.5 PhoNumHoldup	759

3.1.2.27.7.5.6 A1ToA3pWaitTime.....	759
3.1.2.27.7.5.7 A3pTime.....	759
3.1.2.27.7.5.8 设置状态机中去话呼叫的参数.....	760
3.1.2.27.7.5.8.1 MaxWaitOccupyAckTime	760
3.1.2.27.7.5.8.2 MfcKD	760
3.1.2.27.7.5.8.3 MfcKA	761
3.1.2.27.7.5.8.4 MaxWaitKBTime	761
3.1.2.27.7.5.9 与 Dialogic 的 SS1 通道对接.....	761
3.1.2.27.7.5.9.1 ToRingingDelayTime.....	761
3.1.2.27.7.5.9.2 RepeatPhoNumOn1stR2bwdsA5	762
3.1.2.27.7.5.10 应用程序退出时是否向闭塞对端	762
3.1.2.27.7.5.10.1 IsBlockSS1In	762
3.1.2.27.7.5.11 输出调试信息.....	762
3.1.2.27.7.5.11.1 MfcR2ToRxCallerIdBuf.....	762
3.1.2.27.7.6 设置 LineSide 协议的参数.....	763
3.1.2.27.7.6.1 LSWaitPickupTime	763
3.1.2.27.7.6.2 Ss1SendIdleState.....	763
3.1.2.27.7.6.3 LSTxCas	764
3.1.2.27.8 SS7 信令的高级设置	764
3.1.2.27.8.1 设置数字中继线第 16 时隙的属性	764
3.1.2.27.8.1.1 UseTS16AsCircuit.....	764
3.1.2.27.8.1.2 Ss7SignalingTS.....	764
3.1.2.27.8.2 设置地址信号中备用值的对应字符	766
3.1.2.27.8.2.1 AddressSignal.....	766
3.1.2.27.8.3 设置 SS7 信令服务器的 IP 地址	766
3.1.2.27.8.3.1 Ss7ServerIP	766
3.1.2.27.8.3.2 SecondServerIP	766
3.1.2.27.8.4 设置本机的 IP 地址	767
3.1.2.27.8.4.1 LocalIP	767
3.1.2.27.8.5 SS7 信令服务器输出语音信道	767
3.1.2.27.8.5.1 LoadShp_a3AsSIU.....	767
3.1.2.27.8.6 设置数字中继线 SS7 电路	767
3.1.2.27.8.6.1 Ss7CircuitMap[pcm]	767
3.1.2.27.8.7 TUP 协议高级配置项	768
3.1.2.27.8.7.1 设置电路群消息的电路范围	768
3.1.2.27.8.7.1.1 SendGRMRange	768
3.1.2.27.8.7.1.2 HangupRingSendCBK	768
3.1.2.27.8.7.2 来话呼叫: 定制 ACM 消息	769
3.1.2.27.8.7.2.1 DefaultACM	769
3.1.2.27.8.7.3 来话呼叫: 定制 GRQ 消息	769
3.1.2.27.8.7.3.1 ReqTypeIndicators.....	769
3.1.2.27.8.7.4 去话呼叫: 定制 IAI/IAM 消息	770
3.1.2.27.8.7.4.1 ConnectReqMsg.....	770
3.1.2.27.8.7.4.2 CalloutIAM_CAT	770

3.1.2.27.8.7.4.3 CalloutIAM_MsgPntr.....	771
3.1.2.27.8.7.4.4 CallingIndicatorBit.....	772
3.1.2.27.8.7.4.5 OriginalCalleeAddrInd	772
3.1.2.27.8.7.4.6 CallerAddrInd.....	773
3.1.2.27.8.7.5 去话呼叫: 设置应答 GRQ 消息的模式	774
3.1.2.27.8.7.5.1 AutoSendGSM.....	774
3.1.2.27.8.7.6 不使用 SynCTI 提供的 TUP 协议状态机	774
3.1.2.27.8.7.6.1 AutoHandleTup.....	774
3.1.2.27.8.7.7 输出调试信息.....	774
3.1.2.27.8.7.7.1 DebugViewTupCallProc.....	774
3.1.2.27.8.8 ISUP 协议高级配置项	775
3.1.2.27.8.8.1 来话呼叫: 定制本端接受来话呼叫时应答的消息类型.....	775
3.1.2.27.8.8.1.1 DefaultCalledPickupMsg	775
3.1.2.27.8.8.2 来话呼叫: 定制后向呼叫表示语	775
3.1.2.27.8.8.2.1 DefaultBackwardCallInd	775
3.1.2.27.8.8.3 来话呼叫: 定制 REL 消息	776
3.1.2.27.8.8.3.1 DefaultHangupRELIInd.....	776
3.1.2.27.8.8.3.2 DefaultCauseInd.....	777
3.1.2.27.8.8.4 来话呼叫: 设置其它参数.....	778
3.1.2.27.8.8.4.1 SaveRGNTo1stPhoNumStr	778
3.1.2.27.8.8.5 去话呼叫: 定制 IAM 消息.....	778
3.1.2.27.8.8.5.1 DefaultNatureOfConnectionInd.....	778
3.1.2.27.8.8.5.2 DefaultIAM_ForwardCallInd.....	778
3.1.2.27.8.8.5.3 DefaultIAM_CAT	779
3.1.2.27.8.8.5.4 DefaultIAM_TransmissionMediumRequirment	779
3.1.2.27.8.8.5.5 DefaultIAM_CalleeParam	780
3.1.2.27.8.8.5.6 DefaultIAM_CallerParam	780
3.1.2.27.8.8.5.7 DefaultIAM_OriginalCalleeParam.....	781
3.1.2.27.8.8.5.8 DefaultIAM_RedirectingNumber.....	782
3.1.2.27.8.8.5.9 bSubscriberSI	782
3.1.2.27.8.8.5.10 SubscriberSI	782
3.1.2.27.8.8.5.11 bOptionalFCI	782
3.1.2.27.8.8.5.12 OptionalFCI	783
3.1.2.27.8.8.5.13 Usr2UsrInfo	783
3.1.2.27.8.8.5.14 LocationNumber	784
3.1.2.27.8.8.6 去话呼叫: 设置应答 INF 消息的模式.....	784
3.1.2.27.8.8.6.1 AutoSendINF	784
3.1.2.27.8.8.7 去话呼叫: 设置等待 ACM 消息的超时时间	784
3.1.2.27.8.8.7.1 MaxWaitACMTime	784
3.1.2.27.8.8.8 设置挂起状态自动迁移到空闲状态的超时时间.....	785
3.1.2.27.8.8.8.1 MaxWaitPendingToldleTime	785
3.1.2.27.8.8.9 设置驱动 WaitRlc 状态修改为 Pending 状态	785
3.1.2.27.8.8.9.1 WaitRlcToPending	785
3.1.2.27.8.8.10 不使用 SynCTI 提供的 ISUP 协议状态机	785

3.1.2.27.8.8.10.1 AutoHandleIsup	785
3.1.2.27.8.8.10.2 CircuitReset	786
3.1.2.27.8.9 SCCP 协议的高级配置项	786
3.1.2.27.8.9.1 AutoHandleScpp	786
3.1.2.27.8.10 设置 SS7 MSU 输出方式	786
3.1.2.27.8.10.1 GetMsuOnAutoHandle	786
3.1.2.27.8.11 设置 SS7 MTP2 MSU 处理方式	787
3.1.2.27.8.11.1 AppHandleMtp2Msu	787
3.1.2.27.8.12 SS7 高负荷系统相关配置项	787
3.1.2.27.8.12.1 RefreshWatchDogInSys	787
3.1.2.27.8.12.2 ReplySLTAInSys	787
3.1.2.27.8.13 SS7 协议类型	788
3.1.2.27.8.13.1 Ss7Type	788
3.1.2.27.9 ISDN 信令的高级设置	788
3.1.2.27.9.1 设置接续模式	788
3.1.2.27.9.1.1 AutoHandleIsdn	788
3.1.2.27.9.2 设置用户侧/网络侧数字中继线的工作参数	789
3.1.2.27.9.2.1 UserCrcMode	789
3.1.2.27.9.2.2 NetCrcMode	789
3.1.2.27.9.2.3 UserChIdentify	789
3.1.2.27.9.2.4 NetChIdentify	789
3.1.2.27.9.2.5 UserVoiceFormat	790
3.1.2.27.9.2.6 NetVoiceFormat	790
3.1.2.27.9.2.7 UserRestartTime	790
3.1.2.27.9.2.8 NetRestartTime	790
3.1.2.27.9.2.9 UserEstablishTime	791
3.1.2.27.9.2.10 NetEstablishTime	791
3.1.2.27.9.3 设置去话呼叫的工作参数	791
3.1.2.27.9.3.1 UserCalledNoSet	791
3.1.2.27.9.3.2 NetCalledNoSet	791
3.1.2.27.9.3.3 UserCallingNoSet	791
3.1.2.27.9.3.4 NetCallingNoSet	792
3.1.2.27.9.3.5 UserNumIsFull	792
3.1.2.27.9.3.6 NetNumIsFull	793
3.1.2.27.9.3.7 UserChPreference	793
3.1.2.27.9.3.8 NetChPreference	793
3.1.2.27.9.3.9 UserHighLayerCompatible	793
3.1.2.27.9.3.10 NetHighLayerCompatible	794
3.1.2.27.9.3.11 UserLowLayerCompatible	794
3.1.2.27.9.3.12 NetLowLayerCompatible	794
3.1.2.27.9.3.13 UserDialTime	794
3.1.2.27.9.3.14 NetDialTime	795
3.1.2.27.9.3.15 TransferCapability	795
3.1.2.27.9.3.16 PresentNumber	795

3.1.2.27.9.3.17 UserT303	796
3.1.2.27.9.3.18 NetT303	796
3.1.2.27.9.3.19 UserT304	796
3.1.2.27.9.3.20 NetT304	796
3.1.2.27.9.3.21 UserWaitAfterCallProceeding.....	796
3.1.2.27.9.3.22 NetWaitAfterCallProceeding.....	797
3.1.2.27.9.3.23 UserTxCallingPartyNum.....	797
3.1.2.27.9.3.24 NetTxCallingPartyNum.....	797
3.1.2.27.9.4 设置来话呼叫的工作参数	798
3.1.2.27.9.4.1 UserSideDefaultAckTimer	798
3.1.2.27.9.4.2 UserSideDefaultAck	798
3.1.2.27.9.4.3 NetSideDefaultAckTimer	798
3.1.2.27.9.4.4 NetSideDefaultAck	798
3.1.2.27.9.4.5 UserIsReceivePhoNum.....	798
3.1.2.27.9.4.6 NetIsReceivePhoNum.....	799
3.1.2.27.9.4.7 UserIsSendChIdentify	799
3.1.2.27.9.4.8 NetIsSendChIdentify	799
3.1.2.27.9.4.9 UserT302	799
3.1.2.27.9.4.10 NetT302	800
3.1.2.27.9.4.11 UserT313.....	800
3.1.2.27.9.4.12 ProgressExt	800
3.1.2.27.9.5 设置定时器.....	801
3.1.2.27.9.5.1 UserT305	801
3.1.2.27.9.5.2 NetT305	801
3.1.2.27.9.5.3 NetT306	801
3.1.2.27.9.5.4 UserT308	802
3.1.2.27.9.5.5 NetT308	802
3.1.2.27.9.6 设置其它参数	802
3.1.2.27.9.6.1 WaitHangupTime.....	802
3.1.2.27.9.6.2 UserStatusReason	802
3.1.2.27.9.6.3 NetStatusReason	803
3.1.2.27.9.6.4 EnAutoAlert02.....	803
3.1.2.27.9.6.5 EnAutoAlert03.....	803
3.1.2.27.9.6.6 GetRedirectionReason.....	804
3.1.2.27.9.7 NetCallingNoSet	804
3.1.2.27.10 设置 DSP 芯片的工作方式.....	805
3.1.2.27.10.1 LoadFskBin.....	805
3.1.2.27.11 设置信号录制模式	805
3.1.2.27.11.1 ShdDEToneRec	805
3.1.2.27.12 设置底噪过滤门限.....	806
3.1.2.27.12.1 NoiseFilteringMinGate	806
3.1.2.27.13 设置指定 PCM 工作状态	806
3.1.2.27.13.1 PcmPowerDown	806
3.1.2.28 SHN 系列板卡常用配置项(CTI 系列)	807

3.1.2.28.1 SHN 高级通用项配置.....	807
3.1.2.28.1.1 EnableSIPStun	807
3.1.2.28.1.2 SIPStunServerIP.....	807
3.1.2.28.1.3 LogLevel.....	807
3.1.2.28.1.4LogFile.....	807
3.1.2.28.1.5 EventThreadNum	807
3.1.2.28.1.6 SipCallCheckInterval	807
3.1.2.28.1.7 SendOptionsOutCallIP.....	807
3.1.2.28.1.8 SendOptionsOutCallInterval	807
3.1.2.28.1.9 RemoteCrashCheckInterval.....	807
3.1.2.28.1.10 IgnoreUserName	807
3.1.2.28.1.11 AutoDetectRemoteRTPAddress.....	807
3.1.2.28.1.12 HeartInterval	807
3.1.2.28.1.13 UseReport	809
3.1.2.28.1.14 SipTransportProtocol	810
3.1.2.28.1.15 RetransmitLost200OK	810
3.1.2.28.1.16 RegisterTimeOutSpace.....	810
3.1.2.28.1.17 EnableSetUsername.....	810
3.1.2.28.1.18 UserName	811
3.1.2.28.1.19 RegPassword	811
3.1.2.28.1.20 Domain	811
3.1.2.28.1.21 RegRealm.....	811
3.1.2.28.1.22 RegExpires	811
3.1.2.28.1.23 RegStartCh	811
3.1.2.28.1.24 RegEndCh	811
3.1.2.28.1.25 SipTotalMultiRegNum	812
3.1.2.28.1.26 UseGroupByIp	814
3.1.2.28.1.27 UseSipKB	814
3.1.2.28.1.28 SipProxyAddr.....	815
3.1.2.28.1.29 SipProxyPort.....	815
3.1.2.28.1.30 ims	815
3.1.2.28.1.31 SipNewRegisterMode	816
3.1.2.28.1.32 SipBuildBusyReplaceForbidden	816
3.1.2.28.1.33 SipFindRingChFromPreRingCh.....	816
3.1.2.28.1.34 SipDomain	817
3.1.2.28.1.35 SipSearchChInRegisterChannel	817
3.1.2.28.1.36 SipRegNumberAreaLen.....	817
3.1.2.28.1.37 SipAddRTPChkSum	818
3.1.2.28.1.38 StunMaskAddr	818
3.1.2.28.1.39 SipMsgHeaderName	818
3.1.2.28.1.40 SipMsgHeaderValue	818
3.1.2.28.1.41 SipAutoAdaptRtpSrc.....	818
3.1.2.28.1.42 SipSpecialReg	819
3.1.2.28.1.43 SipTotalWhiteListNum.....	819

3.1.2.28.1.44	SipTotalBlackListNum	820
3.1.2.28.1.45	SipWhiteList.....	820
3.1.2.28.1.46	SipBlackList	820
3.1.2.28.1.47	SipSrtp.....	820
3.1.2.28.1.48	SipSessionExpiresMin	820
3.1.2.28.1.49	SipSessionExpires.....	821
3.1.2.28.1.50	SipNetMultiIP	821
3.1.2.28.1.51	SipTransferRTPFromHost	821
3.1.2.28.1.52	SDPAnswerSpecified	822
3.1.2.28.1.53	SipMultilpOn	822
3.1.2.28.1.54	SipRestartAudioFor183	822
3.1.2.28.1.55	SipSingleProtocol	823
3.1.2.28.1.56	SipSendRequestToSrcIP	823
3.1.2.28.1.57	SipExpiresPercent	823
3.1.2.28.1.58	UserAgent.....	824
3.1.2.28.1.59	LocalTLSPort.....	824
3.1.2.28.1.60	SipRegRefreshTime	824
3.1.2.28.1.61	SipRegMode.....	825
3.1.2.28.1.62	SipDscp	825
3.1.2.28.1.63	SipListOnlyForOptions.....	825
3.1.2.28.1.64	SipTCPConnectTime	826
3.1.2.28.1.65	SipCallInTickCountLimit.....	826
3.1.2.28.1.66	SipExpiresPercent	826
3.1.2.28.1.67	SipLearnNetFromVia	826
3.1.2.28.1.68	ContactSection	827
3.1.2.28.2	SHN A 系列卡	827
3.1.2.28.2.1	SHN A 系列卡的高级设置	827
3.1.2.28.2.1.1	RTP 的 NAT 穿透及设置注册服务器	827
3.1.2.28.2.1.1.1	EnableRTPStun	827
3.1.2.28.2.1.1.2	StunServerIP	827
3.1.2.28.2.1.1.3	Register	827
3.1.2.28.2.1.1.4	UserName	827
3.1.2.28.2.1.1.5	RegPassword	827
3.1.2.28.2.1.1.6	Domain	828
3.1.2.28.2.1.1.7	RegExpires	828
3.1.2.28.2.1.1.8	MapIP	828
3.1.2.28.2.1.1.9	RegRealm.....	828
3.1.2.28.3	SHN B/ SHN C 系列卡	829
3.1.2.28.3.1	SHN B/ SHN C 系列卡的高级设置	829
3.1.2.28.3.1.1	RTP 的 NAT 穿透及设置注册服务器	829
3.1.2.28.3.1.1.1	EnableRTPStun	829
3.1.2.28.3.1.1.2	StunServerIP	829
3.1.2.28.3.1.1.3	Register	829
3.1.2.28.3.1.1.4	UserName	829

3.1.2.28.3.1.1.5 RegPassword	830
3.1.2.28.3.1.1.6 Domain	830
3.1.2.28.3.1.1.7 RegExpires	830
3.1.2.28.3.1.1.8 MapIP	830
3.1.2.28.3.1.1.9 RegRealm	830
3.1.2.28.3.1.1.10 DscpFlag	831
3.1.2.28.3.1.1.11 DscpValue	831
3.1.2.28.3.1.2 设置 RTP 语音包负载大小	832
3.1.2.28.3.1.2.1 SizeG711A	832
3.1.2.28.3.1.2.2 SizeG711U	832
3.1.2.28.3.1.2.3 SizeG729	832
3.1.2.28.3.1.2.4 JitterTime	832
3.1.2.28.3.1.3 开启 OCT 语音复位功能	833
3.1.2.29 DST 系列板卡常用配置项(REC 系列)	833
3.1.2.29.1 SupplyBoardClockLine	833
3.1.2.29.2 DEventUpdates	833
3.1.2.30 ATP 系列板卡常用配置项(REC 系列)	834
3.1.2.30.1 设置输入端的增益	834
3.1.2.30.1.1 MicGain	834
3.1.2.30.2 设置空闲通道的能量检测	834
3.1.2.30.2.1 EnableIdleChTA	834
3.1.2.30.3 选择多种录音格式	835
3.1.2.30.3.1 DspCoder	835
3.1.2.30.4 设置预录音功能	835
3.1.2.30.4.1 PrerecordEnable	835
3.1.2.30.4.2 PrerecordMode	836
3.1.2.30.4.3 PrerecordInsertTime	836
3.1.2.30.4.4 PrerecordCodec	837
3.1.2.30.5 检测脉冲方式的按键信号	837
3.1.2.30.5.1 EnablePulseKeyDetect	837
3.1.2.30.6 设置闪断信号的最大保持时间	837
3.1.2.30.6.1 MaxRecChFlashFilterTime	837
3.1.2.30.7 设置振铃信号计数器的清零保护时间	838
3.1.2.30.7.1 RecChClearRingDelayTime	838
3.1.2.30.8 设置软交换功能	838
3.1.2.30.8.1 BusPlayListen	838
3.1.2.30.9 设置 MF 检测器	838
3.1.2.30.9.1 AlwaysEnableRxMF	838
3.1.2.30.9.2 MFDualAndAllFreqEnScale	839
3.1.2.30.9.3 MFFreqOffset	839
3.1.2.31 DTP 系列板卡常用配置项(REC 系列)	839
3.1.2.31.1 将监控消息保存到文件	839
3.1.2.31.2 SS7 信令状态机同步输出原始信令消息	840
3.1.2.31.2.1 bOpenSpySS7Mstu	840

3.1.2.31.3 ISDN 监控支持 CorNet 协议	840
3.1.2.31.3.1 ISDNProtocolType	840
3.1.2.31.4 E1 SS1 监控支持无 R2 信令链路	840
3.1.2.31.4.1 SpySs1NoR2	840
3.1.2.31.5 链路不同步到通道不可用状态需要的时间	841
3.1.2.31.5.1 SpyWaitUnavailableTime	841
3.1.2.31.6 设置板卡消息解码方式	841
3.1.2.31.6.1 UseHdIc	841
3.1.2.31.7 SS1 监控支持 RBS 协议	842
3.1.2.31.7.1 RBSType	842
3.1.2.31.8 ISDN 监控支持输出第 2 层 (L2) 原始消息事件	842
3.1.2.31.8.1 bOpenSpyIsdnL2	842
3.1.2.32 IPR 系列产品常用配置项(REC 系列)	843
3.1.2.32.1 SynIPRecorder 系列产品 Master 端的高级通用项配置	843
3.1.2.32.1.1 RcvTimeSpan	843
3.1.2.32.1.2 CodecInBufferSize	843
3.1.2.32.1.3 SlaverLogLevel	843
3.1.2.32.1.4 SlaverLogType	844
3.1.2.32.1.5 RecorderMixerType	844
3.1.2.32.1.6 RecorderJitterBuffer	844
3.1.2.32.1.7 RecorderVolume	845
3.1.2.32.2 SynIPAnalyzer 系列产品的 IP 交换机监控端口项配置	845
3.1.2.32.2.1 SIP	845
3.1.2.32.2.2 Cisco SCCP	846
3.1.2.32.2.3 Avaya H323	847
3.1.2.32.2.4 Shortel MGCP	848
3.1.2.32.2.5 H323	848
3.1.2.32.2.6 Panasonic MGCP	849
3.1.2.32.2.7 Toshiba MEGACO	850
3.1.2.32.2.8 Siemens H323	850
3.1.2.32.2.9 Alcatel	851
3.1.2.32.2.10 Mitel	851
3.1.2.32.2.11 LG Nortel	851
3.1.2.32.2.12 Samsung	852
3.1.2.32.2.13 Tadicom MGCP	852
3.1.2.32.2.14 Zenitel	853
3.1.2.32.2.15 Nortel UNIStim	853
3.1.2.32.3 SynIPAnalyzer 系列产品高级通用项配置	854
3.1.2.32.3.1 SynIPAnalyzer 声控录音	854
3.1.2.32.3.1.1 RTPCtrlRec	854
3.1.2.32.3.2 SynIPAnalyzer 分析日志	854
3.1.2.32.3.2.1 IPALogType	854
3.1.2.32.3.2.2 IPALogLevel	854
3.1.2.32.3.3 SynIPAnalyzer 数据抓包	855

3.1.2.32.3.3.1	DumpIPData.....	855
3.1.2.32.3.3.2	DumpPackNumPerFile.....	855
3.1.2.32.3.3.3	DumpFileReserveNum.....	855
3.1.2.32.3.3.4	DumpFilePath	855
3.1.2.33	SHV 系列板卡常用配置项	855
3.1.2.33.1	OldVar	855
3.1.3	高级配置项	856
3.1.3.1	驱动程序支持多个进程	856
3.1.3.1.1	MultiCardMultiProcess	856
3.1.3.2	输出 API 函数调用的调试信息	856
3.1.3.2.1	ApiLogEnable.....	856
3.1.3.2.2	ApiLogSetEventRange	857
3.1.3.2.3	ApiLogSetChRange	857
3.1.3.2.4	ApiLogCreateMode	858
3.1.3.2.5	Mask_SsmGetNoSoundTime	858
3.1.3.2.6	Mask_SsmGetChStateKeepTime.....	858
3.1.3.2.7	Mask_SsmGetPlayedTime.....	858
3.1.3.2.8	Mask_SsmGetPlayedPercentage	858
3.1.3.2.9	Mask_SsmGetPlayOffset	858
3.1.3.2.10	Mask_SsmGetRecTime	859
3.1.3.2.11	Mask_SsmGetRecOffset	859
3.1.3.2.12	Mask_SsmGetChState	859
3.1.3.3	设置 ISDN 接续日志	859
3.1.3.3.1	IsdnLogEnable	859
3.1.3.3.2	DecodeIsdnMsg	860
3.1.3.3.3	IsdnDebugLog.....	860
3.1.3.4	设置 Ss1 接续日志	861
3.1.3.4.1	Ss1LogEnable.....	861
3.1.3.4.2	Ss1LogCreateMode	861
3.1.3.5	设置 Ss1 监控日志	861
3.1.3.5.1	SpySs1LogEnable.....	861
3.1.3.5.2	SpySs1CreateMode	862
3.1.3.6	设置 Ss7 监控日志	862
3.1.3.6.1	SpySs7LogEnable.....	862
3.1.3.7	设置 ISDN 监控日志	862
3.1.3.7.1	SpyIsdnLogEnable	862
3.1.3.8	设置系统信息监控日志	863
3.1.3.8.1	SystemInfoLogEnable	863
3.1.3.9	设置数字电话监控信令日志	863
3.1.3.9.1	DSTLogEnable.....	863
3.1.3.9.2	DSTLogCreateMode	863
3.1.3.10	设置传真交互命令日志	864
3.1.3.10.1	FaxLogEnable	864
3.1.3.10.2	FaxLogCreateMode	864

3.1.3.11 设置 Oct 模块调试信息和错误信息日志	864
3.1.3.11.1 OctLogEnable.....	864
3.1.3.12 设置创建日志相关的控制信息	865
3.1.3.12.1 LogCreatePeriod.....	865
3.1.3.12.2 LogMaxKeep.....	865
3.1.3.12.3 LogMaxPeriod.....	866
3.1.3.12.4 LogFilePath.....	866
3.1.3.12.5 LogOverWrite	867
3.1.3.12.6 CreateDumpWhenCrash.....	867
3.1.3.13 设置 DSP 四字节访问功能	868
3.1.3.13.1 DspAddr4	868
3.1.3.14 设置 DSP 的回波抵消效果	868
3.1.3.14.1 DspEc.....	868
3.1.3.15 设置 CPU 多核分担功能.....	868
3.1.3.15.1 EnableSetDpc	868
3.1.3.16 设置过滤直流偏离	868
3.1.3.16.1 DcOnOffValue	868
3.1.3.16.2 BusDcOnOffValue	869
3.1.3.17 设置驱动事件处理方式	869
3.1.3.17.1 MultiBoardClock	869
3.2 预加载语音配置文件 ShIndex.ini (CTI 系列)	869
3.2.1 [System] 节	869
3.2.1.1 MaxIndexSeg	869
3.2.2 [SegNo=x] 节	870
3.2.2.1 FileName.....	870
3.2.2.2 Alias	870
3.2.2.3 CodecFormat	870
3.2.2.4 StartOffset	871
3.2.2.5 Length	871
3.3 SynIPRecorder Slaver 端配置文件 record_slaver.ini (SynIPR 产品 only)	871
3.3.1 Slaver 端的必须配置项	872
3.3.1.1 ActiveConnect.....	872
3.3.1.2 SlaverIP	872
3.3.1.3 SlaverPort	872
3.3.1.4 MasterIP.....	873
3.3.1.5 MasterPort	873
3.3.1.6 SlaverListenIP	873
3.3.1.7 SlaverListenPort.....	873
3.3.2 Slaver 端的高级通用项配置	874
3.3.2.1 KeepAliveTime	874
3.3.2.2 RTPPortRange.....	874
3.4 HMP Client 端配置文件 HMPCodec.ini (SynHMP 产品 only)	874
3.4.1 HMP 系列产品 Client 端的必须配置项	874
3.4.1.1 NICNum	874

3.4.1.2	LocalIP	875
3.4.1.3	RemoteIP	875
3.4.1.4	LocalPort.....	875
3.4.1.5	RemotePort.....	875
3.4.1.6	LogType	876
3.4.1.7	LogLocation	876
3.4.1.8	MaxRtpThread	876
4	SS7 信令方式 (CTI 系列)	878
4.1	SS7 信令方式的基本概念	878
4.1.1	OPC 和 DPC	878
4.1.2	直联和准直联.....	878
4.1.3	信令链路和链路组	879
4.2	Synway 板卡的 SS7 信令应用系统	879
4.3	SS7 信令服务器的配置和说明	880
4.3.1	设置信令服务器的工作参数.....	882
4.3.1.1	OPC	882
4.3.1.2	SpCodeLen	883
4.3.1.3	TEJ	883
4.3.2	设置信令服务器的 IP 地址.....	884
4.3.2.1	ServerIP	884
4.3.2.2	SecondServerIP	884
4.3.2.3	Port	884
4.3.3	设置 MTP3 的工作模式和工作参数	884
4.3.3.1	ConfigMtp3AsSTP	884
4.3.3.2	SendSNT	885
4.3.3.3	SubServicefield	885
4.3.3.4	SubServicefield	886
4.3.4	设置客户端的信息	887
4.3.4.1	MaxSs7Client.....	887
4.3.4.2	IP	887
4.3.5	设置信令链路的物理位置	888
4.3.5.1	MaxSs7Pcm.....	888
4.3.5.2	Ss7PcmLink	888
4.3.6	设置信令链路组	889
4.3.6.1	MaxLinkSet	889
4.3.6.2	LinkSet	889
4.3.7	设置 DPC	890
4.3.7.1	MaxDPC.....	890
4.3.7.2	DPC	890
4.3.8	设置用户层消息的目的点码.....	890
4.3.8.1	MaxUP_DPC.....	890
4.3.8.2	UP_DPC	890
4.3.9	设置 TUP/ISUP 消息的分发路由	891
4.3.9.1	UP_DPC	891

4.3.9.2	CIC_PCM.....	891
4.3.9.3	DPC	892
4.3.10	设置服务器的启动特性.....	892
4.3.10.1	ConfigAsGateway	892
4.3.11	设置服务器的显示特性.....	893
4.3.11.1	RcvMsuListMaxItem.....	893
4.3.11.2	TxMsuListMaxItem	893
4.3.12	Windows 7 号信令服务器专用配置项.....	893
4.3.12.1	EnDisplayL32Msu	893
4.3.12.2	EnDisplayL23Msu	894
4.3.12.3	ShowSNT.....	894
4.3.12.4	ShowSNM.....	894
4.3.12.5	AutoTranslate.....	895
4.3.12.6	WriteToFile	895
4.3.12.7	ShowDPCOPC.....	895
4.3.12.8	LogToPcap	895
4.3.12.9	HeartTimeOut.....	896
4.3.12.10	HandleWithoutCic	896
4.3.12.11	OutputTimerStateToLog	896
4.3.13	Linux 7 号信令服务器专用配置项.....	897
4.3.13.1	EnDisplayL32Msu	897
4.3.13.2	EnDisplayL23Msu	897
4.3.13.3	ShowMSU	897
4.3.13.4	ShowStatus	897
4.3.13.5	ShowSNT	898
4.3.13.6	ShowSNM	898
4.3.13.7	AutoTranslate.....	898
4.3.13.8	LogFileSize	899
4.3.13.9	TotalLogFile.....	899
4.3.13.10	LogPath	899
4.3.13.11	LogToPcap	899
4.3.13.12	WriteToFile	900
4.3.13.13	ShowDPCOPC	900
4.3.14	一个 Ss7Monitor 程序支持多个 Client 客户端的应用	900
4.3.14.1	客户端与服务器在同一台机器上.....	900
4.3.14.2	客户端与服务器在不同的机器上.....	902
4.4	SS7 信令服务器的应用及配置实例	902
4.4.1	单 OPC/单 DPC.....	902
4.4.1.1	单机单卡的应用系统.....	902
4.4.1.2	单机多卡的应用系统.....	903
4.4.1.3	多机系统.....	904
4.4.1.4	使用专用信令服务器的多机系统.....	905
4.4.2	单 OPC/多 DPC.....	907
4.4.3	多 OPC 的 SS7 应用系统	908

4.4.4	高可靠性的应用系统.....	909
4.4.5	为第三方板卡提供 SS7 信令服务	912
4.4.6	准直联的应用系统	913
附录 1	ISDN 通道释放原因值表	915
附录 2	ISUP 可选参数列表和部分参数介绍	918

版权声明

本文档是杭州三汇信息工程有限公司（以下简称三汇公司）“Synway 板卡驱动软件”产品的组成部分，三汇公司拥有该软件以及本文档的一切版权，受中华人民共和国法律的保护。未经本公司书面授权，任何人不得复制、传播、摘抄、修改本文档的全部或部分内容。使用本文档，即视为接受后面的“软件授权协议”。

三汇公司保留对本文档进行修改而不另行通知之权利。

三汇公司对本文档进行了仔细校对，力求文档内容准确、可靠，但并不保证绝无错误。

请在使用本产品前，自行确定所使用的相关技术文件及规格为最新有效之版本。若因贵公司使用本公司之文件或产品，而需要第三方之产品、专利或者著作等与其配合时，则应由贵公司负责取得第三方同意及授权。关于上述同意及授权，非属本公司应为保证之责任。

注：本文中提到的 Windows, Windows 2000, Windows XP 等是微软的注册商标，Dialogic 是英特尔的注册商标。

SynCTI 驱动程序授权协议

- 1、三汇公司（以下简称本公司）拥有“本软件及所有附属产品、文件和相关文档”（以下简称本产品）的完全版权。任何单位和个人在使用前须获得本公司的授权。
- 2、本公司授权符合以下全部条件的单位和个人免费使用本产品：
 - A、本产品与通过合法销售渠道购买的本公司硬件产品配合使用时；
 - B、不私自向第三方传播本产品和本产品的任何一部分。
- 3、除符合第2条以外的其他单位和个人需要使用本产品时，必须获得本公司的书面授权。
- 4、获得授权的单位和个人不得转让其获得的授权。
- 5、使用本产品，即表明完全理解本协议的所有条款并全部接受。

版本修订记录

版本号	发布日期	修订内容
Ver2.0	2001.12	创建本文档。
Ver3.0.0.0	2003.07	重大修订。
Ver4.7.3.0	2006.12	重大修订。
Ver4.8.0.0	2007.07	重大修订。
Ver4.8.0.1	2007.08	无。
Ver4.9.0.0	2007.10	重大修订。
Ver5.0.0.0	2007.12	重大修订。
Ver5.0.1.0	2008.4	重大修订。
Ver5.0.2.0	2008.8	重大修订。
Ver5.0.3.0	2009.1	重大修订。
Ver5.0.4.0	2009.5	重大修订。
Ver5.0.5.0	2009.7	重大修订。
Ver5.1.0.0	2009.11	重大修订。
Ver5.1.1.0	2009.12	重大修订。
Ver5.2.0.0	2010.6	重大修订。
Ver5.2.0.1	2010.8	重大修订。
Ver5.2.0.2	2010.8	重大修订。
Ver5.2.0.3	2010.10	重大修订。
Ver5.3.0.0	2010.12	重大修订。
Ver5.3.0.1	2011.1	重大修订。
Ver5.3.0.2	2011.3	重大修订。
Ver5.3.0.3	2011.4	重大修订。
Ver5.3.0.4	2011.4	重大修订。
Ver5.3.1.0	2011.7	重大修订。

Ver5.3.1.1	2011.10	重大修订。
Ver5.3.1.2	2011.12	重大修订。
Ver5.3.1.3	2012.2	重大修订。
Ver5.3.1.4	2012.4	重大修订。
Ver5.3.1.5	2012.7	重大修订。
Ver5.3.2.0	2012.10	重大修订。
Ver5.3.2.1	2012.12	重大修订。
Ver5.3.2.2	2013.3	重大修订。
Ver5.3.2.3	2013.6	重大修订。
Ver5.3.2.4	2013.8	重大修订。
Ver5.3.2.5	2013.11	重大修订。
Ver5.3.2.6	2014.3	重大修订。
Ver5.3.2.7	2014.7	重大修订。
Ver5.3.3.0	2014.12	重大修订。
Ver5.3.3.1	2015.4	重大修订。
Ver5.3.3.2	2015.11	重大修订。
Ver5.3.4.0	2016.3	重大修订。
Ver5.4.0.0	2016.8	重大修订。
Ver5.4.1.0	2017.1	重大修订。
Ver5.4.2.0	2017.5	重大修订。
Ver5.4.3.0	2018.8	重大修订。

注：此处只记载针对文档本身的主要修订记录，由于 SynCTI 驱动程序升级而对文档进行修订的详细信息请参考对应版本的《驱动程序升级手册》。

前言

欢迎使用 **Synway** 板卡。**SynCTI** 是 **Synway** 板卡的驱动程序，本文档是 **SynCTI** 驱动程序的帮助文件，主要阅读对象是使用 **SynCTI** 驱动程序的应用程序开发接口（API）开发应用软件的软件工程师。本文档还可以作为使用 **Synway** 产品构建 CTI 应用系统的安装维护技术人员、产品销售人员的参考书。

Synway 所有的板卡产品，都使用一个称为 **SynCTI** 的统一的驱动程序及开发平台作为硬件和应用软件之间的桥梁。**SynCTI** 为不同型号的硬件板卡提供统一的 API 接口，因此，基于某一种型号的 **Synway** 板卡开发的应用系统，可以很方便地扩展到各种不同的环境下。

本文档由四章组成。第 1 章讲述语音卡编程的基础知识，其中包括 **Synway** 板卡和 **SynCTI** 驱动程序的基本组成、系统架构，实现各种主要功能的操作方法和注意事项。阅读完这一章，读者可以知道 **Synway** 板卡能做什么，以及如何使用 API 来实现。对于首次使用 **Synway** 产品的读者，建议您在编程之前首先详细阅读这一章，当然，在阅读时可以跳过与您的应用程序无关的内容。例如，如果您的应用系统只是用来处理模拟电话线，那么您可以跳过数字中继线部分内容。

第 2 章详细描述 **SynCTI** 驱动程序的 API 函数、驱动程序抛出的事件，并对相关的数据结构进行了详细的描述。

第 3 章详细描述 **SynCTI** 驱动程序的全部配置项。本章的内容按照功能分类进行排列，每一功能的相关配置项按照分为必须配置项和常用/高级配置项，高级配置项通常用于支持少量特殊环境和特殊用户的需求。

第 4 章描述如何配置和实现使用 SS7 号信令的应用系统。

虽然 **Synway** 公司对本书进行了仔细的校对，但其中仍难免有错误和遗漏的地方，我们对由此给读者带来的不便深表歉意。同时，如果您能对此提出改进意见，我们将会非常感谢。

1 Synway 板卡编程基础

1.1 Synway 板卡的分类 (CTI 系列)

Synway 板卡分为 2 大类: CTI 系列和 REC 系列。CTI 系列板卡主要作为通讯的一方进行电话呼叫, REC 系列为录音卡, 主要用于高阻复接方式的并线录音。CTI 系列板卡和 REC 系列板卡的驱动程序都包含在 SynCTI 驱动程序中。

CTI 系列包括五个子系列:

子系列名称	描述
SHT 系列	采用模块化结构, 模拟线路
SHD 系列	数字中继线 (E1/T1/J1)
SHV 系列	变声卡
SHN 系列	VoIP 卡
SHF 系列	传真卡
HMP 系列	主机媒体处理

各子系列所包含板卡型号的详细列表:

系列	总线形式	板卡型号	备注
SHT	USB	SHT-2A/USB	简称“USB 语音盒”
		SHT-4A/USB	
		SHT-2B/USB	
		SHT-4B/USB	
		SHT-16A-CT/PCI	
		SHT-120A-CT/PCI	已停产
		SHT-8B/PCI	
		SHT-8B/PCI/FAX	
		SHT-16B-CT/PCI	
		SHT-16B-CT/PCI/FAX	
		SHT-16B-CT/PCI/MP3	
		SHT-8C/PCI/FAX	
		SHT-8C/PCI/EC	
		SHT-16C-CT/PCI/FAX	
		SHT-16C-CT/PCI/EC	
	PCIe	SHT-16D-CT/PCIe	
	cPCI	SHT-120A-CT/cPCI	已停产
		SHT-16B-CT/cPCI	
		SHT-16B-CT/cPCI/FAX	
		SHT-16B-CT/cPCI/MP3	
SHD	PCI	SHD-30A-CT/PCI/SS1	支持 SS1
		SHD-30A-CT/PCI/ISDN	支持 SS1、ISDN
		SHD-30A-CT/PCI/SS7	支持 SS1、ISDN、SS7
		SHD-60A-CT/PCI/SS1	支持 SS1
		SHD-60A-CT/PCI/ISDN	支持 SS1、ISDN
		SHD-60A-CT/PCI/SS7	支持 SS1、ISDN、SS7
		SHD-120A-CT/PCI/SS1	支持 SS1
		SHD-120A-CT/PCI/ISDN	支持 SS1、ISDN
		SHD-120A-CT/PCI/SS7	支持 SS1、ISDN、SS7
		SHD-30B-CT/PCI/SS7/FAX	支持 SS1、ISDN、SS7

		SHD-60B-CT/PCI/SS7/FAX	支持 SS1、ISDN、SS7
		SHD-30C-CT/PCI	支持 E1 SS1、ISDN、SS7 和 T1 ISDN
		SHD-30C-CT/PCI/FAX	支持 E1 SS1、ISDN、SS7 和 T1 ISDN
		SHD-60C-CT/PCI	支持 E1 SS1、ISDN、SS7 和 T1 ISDN
		SHD-60C-CT/PCI/FAX	支持 E1 SS1、ISDN、SS7 和 T1 ISDN
		SHD-120D-CT/PCI	支持 E1 SS1、ISDN、SS7 和 T1 ISDN
		SHD-120D-CT/PCI/EC	支持 E1 SS1、ISDN、SS7 和 T1 ISDN
		SHD-120D-CT/PCI/CAS	支持 SS1
		SHD-240D-CT/PCI	支持 E1 SS1、ISDN、SS7 和 T1 ISDN
		SHD-240D-CT/PCI/EC	支持 E1 SS1、ISDN、SS7 和 T1 ISDN
		SHD-240D-CT/PCI/CAS	支持 SS1
		SHD-30E-CT/PCI(SSW)	支持 E1 SS1、ISDN、SS7 和 T1 ISDN
		SHD-30E-CT/PCI/FAX(SSW)	支持 E1 SS1、ISDN、SS7 和 T1 ISDN
		SHD-30E-CT/PCI/EC(SSW)	支持 E1 SS1、ISDN、SS7 和 T1 ISDN
		SHD-60E-CT/PCI(SSW)	支持 E1 SS1、ISDN、SS7 和 T1 ISDN
		SHD-60E-CT/PCI/FAX(SSW)	支持 E1 SS1、ISDN、SS7 和 T1 ISDN
		SHD-60E-CT/PCI/EC(SSW)	支持 E1 SS1、ISDN、SS7 和 T1 ISDN
		SHD-120E-CT/PCI(SSW)	支持 E1 SS1、ISDN、SS7 和 T1 ISDN
		SHD-120E-CT/PCI/FAX(SSW)	支持 E1 SS1、ISDN、SS7 和 T1 ISDN
		SHD-120E-CT/PCI/EC(SSW)	支持 E1 SS1、ISDN、SS7 和 T1 ISDN
		SHD-240E-CT/PCI(SSW)	支持 E1 SS1、ISDN、SS7 和 T1 ISDN
		SHD-240E-CT/PCI/FAX(SSW)	支持 E1 SS1、ISDN、SS7 和 T1 ISDN
		SHD-240E-CT/PCI/EC(SSW)	支持 E1 SS1、ISDN、SS7 和 T1 ISDN
	PCIe	SHD-30E-CT/PCIe	支持 E1 SS1、ISDN、SS7 和 T1 ISDN
		SHD-30E-CT/PCIe/FAX	支持 E1 SS1、ISDN、SS7 和 T1 ISDN
		SHD-30E-CT/PCIe/EC	支持 E1 SS1、ISDN、SS7 和 T1 ISDN
		SHD-60E-CT/PCIe	支持 E1 SS1、ISDN、SS7 和 T1 ISDN
		SHD-60E-CT/PCIe/FAX	支持 E1 SS1、ISDN、SS7 和 T1 ISDN
		SHD-60E-CT/PCIe/EC	支持 E1 SS1、ISDN、SS7 和 T1 ISDN
		SHD-120E-CT/PCIe	支持 E1 SS1、ISDN、SS7 和 T1 ISDN
		SHD-120E-CT/PCIe/FAX	支持 E1 SS1、ISDN、SS7 和 T1 ISDN
		SHD-120E-CT/PCIe/EC	支持 E1 SS1、ISDN、SS7 和 T1 ISDN
		SHD-240E-CT/PCIe	支持 E1 SS1、ISDN、SS7 和 T1 ISDN
		SHD-240E-CT/PCIe/FAX	支持 E1 SS1、ISDN、SS7 和 T1 ISDN
		SHD-240E-CT/PCIe/EC	支持 E1 SS1、ISDN、SS7 和 T1 ISDN
		SHD-240E-CT/PCIe/VAR	支持 E1 SS1、ISDN、SS7 和 T1 ISDN，支持 120 路变声资源通道
	cPCI	SHD-30A-CT/cPCI/SS7	支持 SS1、ISDN、SS7
		SHD-60A-CT/cPCI/SS7	支持 SS1、ISDN、SS7
		SHD-120A-CT/cPCI/SS7	支持 SS1、ISDN、SS7
		SHD-240A-CT/cPCI	支持 SS7
		SHD-480A-CT/cPCI	支持 SS7
		SHD-30B-CT/cPCI/SS7/FAX	支持 SS1、ISDN、SS7
		SHD-60B-CT/cPCI/SS7/FAX	支持 SS1、ISDN、SS7
		SHD-240S-CT/cPCI	支持 SS7
		SHD-480S-CT/cPCI	支持 SS7
SHV	PCI	SHV-120A-CT/PCI	
		SHV-240A-CT/PCI	
	cPCI	SHV-240A-CT/cPCI	
SHN	PCI	SHN-32A-CT/PCI	支持 SIP 协议
		SHN-8B-CT/PCI+	支持 SIP 协议
		SHN-16B-CT/PCI+	支持 SIP 协议
		SHN-32B-CT/PCI+	支持 SIP 协议
		SHN-60B-CT/PCI+	支持 SIP 协议
		SHN-120B-CT/PCI+	支持 SIP 协议

	PCIe	SHN-60B-CT/PCIe+	支持 SIP 协议
		SHN-120B-CT/PCIe+	支持 SIP 协议
		SHN-120B-CT/PCIe/VAR	支持 SIP 协议
		SHN-480C-CT/PCIe	支持 SIP 协议
SHF	PCI	SHF-2D/PCI	支持 V.34、V.17、V.29、V.27ter、V.8
		SHF-4D/PCI	支持 V.34、V.17、V.29、V.27ter、V.8
	PCIe	SHF-4D/PCIe	支持 V.34、V.17、V.29、V.27ter、V.8
HMP		SYNWAY-HMP	支持 SIP 协议

1.2 Synway 板卡的分类 (REC 系列)

Synway 板卡分为 2 大类: CTI 系列和 REC 系列。CTI 系列板卡主要作为通讯的一方进行电话呼叫, REC 系列为录音卡, 主要用于高阻复接方式的并线录音。CTI 系列板卡和 REC 系列板卡的驱动程序都包含在 SynCTI 驱动程序中。

REC 系列包括五个子系列:

子系列名称	描述
ATP 系列	采用模块化结构, 适用于对模拟电话线路或其它类型模拟语音信号进行高阻复接录音
DTP 系列	适用于对数字中继线同时进行信令链路和语音信道的高阻复接录音。支持 SS1/ISDN/SS7
DST 系列	适用于对数字话机线路 (2B+D) 进行高阻复接录音
SHF 系列	用于对录制成声音文件的传真数据进行解码, 从而获得实际的传真图像, 通常作为录音系统的配套产品使用, 其中 D 型卡是支持 33.6k 高速传真收/发功能的板卡
IPR 系列	适用于对 VoIP 线路上的话机或交换机进行录音

各子系列所包含板卡型号的详细列表:

系列	总线形式	板卡型号	备注
ATP 系列	USB	SHT-2A/USB	简称“USB 录音盒”
		SHT-4A/USB	
		SHT-2B/USB	
		SHT-4B/USB	
	PCI	SHT-8A/PCI	支持 MP3 格式的硬件编码
		SHT-8B/PCI	
		SHT-16A-CT/PCI	
		SHT-16B-CT/PCI	
		SHT-16B-CT/PCI/MP3	
		ATP-24A/PCI	
	PCIe	ATP-24A/PCI+	支持 GSM、MP3 或 G.729A 格式的硬件编码
		ATP-24A/PCIe	
DTP 系列	PCI	ATP-24A/PCIe+	支持 GSM、MP3 或 G.729A 格式的硬件编码
		SHT-16B-CT/cPCI	
		SHT-16B-CT/cPCI/MP3	支持 MP3 格式的硬件编码
		SHD-30A-CT/PCI/FJ	板卡内置高阻接入电路
		SHD-60A-CT/PCI/FJ	板卡内置高阻接入电路
		SHD-30B-CT/PCI/FJ	板卡内置高阻接入电路
		SHD-60B-CT/PCI/FJ	板卡内置高阻接入电路
		DTP-30C/PCI	板卡内置高阻接入电路
		DTP-30C/PCI+	板卡内置高阻接入电路
	PCIe	DTP-60C/PCI	板卡内置高阻接入电路
	PCI	DTP-60C/PCI+	板卡内置高阻接入电路
		DTP-120C/PCI	板卡内置高阻接入电路
	PCIe	DTP-120C/PCI+	板卡内置高阻接入电路
		DTP-30C/PCIe	板卡内置高阻接入电路

	DTP-30C/PCIe+	板卡内置高阻接入电路
	DTP-60C/PCIe	板卡内置高阻接入电路
	DTP-60C/PCIe+	板卡内置高阻接入电路
	DTP-120C/PCIe	板卡内置高阻接入电路
	DTP-120C/PCIe+	板卡内置高阻接入电路
DST 系列	PCI	SHR-16DA-CT/PCI 支持 PCI-X 总线
		SHR-24DA-CT/PCI 支持 PCI-X 总线
		DST-24B/PCI
		DST-24B/PCI(SSW)
		DST-24B/PCI+ 支持 GSM、MP3 或 G.729A 格式的硬件编码
		DST-24B/PCI+(SSW) 支持 GSM、MP3 或 G.729A 格式的硬件编码
PCIe	DST-24B/PCIe	
	DST-24B/PCIe+	支持 GSM、MP3 或 G.729A 格式的硬件编码
IPR 系列	SynIPAnalyzer	支持 USB-KEY 或 NTP 产品
	SynIPRecorder	支持 USB-KEY 或 NTP 产品

1.3 SynCTI 支持的 CODEC

Synway 板卡支持的语音编码/解码格式(CODEC)格式及特性如下表所示:

CODEC	简称	采样率	码率	字节/帧	字节/秒	CODEC 引擎	编码值
PCM8 无符号	PCM8	8000 Hz	64 Kbps	1	8000	主机 CPU	1
16 位线性 PCM	PCM16	8000 Hz	128 Kbps	2	16000	主机 CPU	-2
A-law	A-law	8000 Hz	64 Kbps	1	8000	板载 DSP	6
μ-law	μ-law	8000 Hz	64 Kbps	1	8000	板载 DSP	7
IMA ADPCM	IMA ADPCM	8000 Hz	32 Kbps	256	4055	板载 DSP	17
VOX	VOX	8000 Hz	32 Kbps	256	4000	板载 DSP	23
MP3	MP3	8000 Hz	8 Kbps	72	1000	板载 DSP 主机 CPU	85
GSM 6.10	GSM	8000 Hz	13 Kbps	65	1625	主机 CPU	49
G.729A	G.729A	8000 Hz	8 Kbps	10	1000	板载 DSP	131

注:

- (1) 如果板卡型号中带有“/MP3”标记, CODEC 引擎使用板载 DSP 芯片; 否则使用主机 CPU。
- (2) VOX 格式是指 Dialogic 的 ADPCM 编码格式。
- (3) 由于历史原因, G729A 曾经使用过的语音编码参数值 65411 在低版本依然有效, 版本向下兼容。如果使用 G.729A 格式录音, 建议采用 131。

1.3.1 编码格式转换的工具函数

ShPcmHandle.dll 提供了语音编码格式转换的工具函数, 这些工具函数在使用时, 无须先调用 [SsmStartCti](#) 函数初始化 SynCTI 驱动程序, 但使用结束后务必调用 [fPcm_Close](#) 函数以释放资源。语音编码格式转换的函数包括:

函数名称	功能概述
fPcm_Mp3ConvertALaw	将 Synway 板卡录制的 MP3 格式文件转换为 A 率格式文件
fPcm_Mp3ConvertULaw	将 MP3 格式文件转换为 μ 率格式文件
fPcm_AdpcmToALaw	将 Synway 板卡录制的 IMA ADPCM 格式文件转换为 A 率格式文件

fPcm_AdpcmToGsm	将 IMA ADPCM 格式文件转换为 GSM 6.10 格式文件
fPcm_AdpcmToMp3	将 IMA ADPCM 格式文件转换为 MP3 格式文件
fPcm_AlawToUlaw	将 Synway 板卡录制的 A-Law 格式文件转换为 μ-Law 格式文件
fPcm_UlawToAlaw	将 Synway 板卡录制的 μ-Law 格式文件转换为 A-Law 格式文件
fPcm_MemAdpcmToALAW	将使用 Synway 板卡录制的、保存在缓冲区中的 IMA ADPCM 格式数据转换成 A-law
fPcm_MemAdpcmToULAW	将使用 Synway 板卡录制的、保存在缓冲区中的 IMA ADPCM 格式数据转换成 μ-law
fPcm_AlawConvertPcm16	将 A-law 格式文件转换为 16 Bit PCM 格式
fPcm_ALawConvertPcm8	将 A-law 格式文件转换为 8 Bit 无符号 PCM 格式文件
fPCM_AlawConvertGC8	将 A-law 格式文件转换为 G.729A 格式文件
fPcm_ALawConvertMp3	将 A-law 格式文件转换为 MP3 格式文件
fPcm_ULawConvertMp3	将 μ-Law 格式文件转换为 MP3 格式文件
fPcm_ALawToVox	将 A-Law 格式文件转换为 vox 格式文件
fPcm_MemAlawToPcm8	将保存在缓冲区中的 A-law 格式数据转换为 8 Bit 无符号 PCM 格式
fPcm_MemAlawToPcm16 fPcm_MemMp3ToPcm16	将保存在缓冲区中的 A-law/MP3 格式数据转换为 16 Bit 无符号 PCM 格式
fPcm_MemGSMToPcm16	将保存在缓冲区中的 GSM 格式数据转换成 16 Bit PCM 格式
fPcm_MemGSMToPcm8	将保存在缓冲区中的 GSM 格式数据转换成 8 Bit 无符号 PCM 格式
fPcm_MemGSMToUlaw	将保存在缓冲区中的 GSM 格式数据转换成 μ-law 格式
fPcm_Pcm16ConvertALaw	将 16 Bit PCM 格式文件转换为 A-law 格式的文件
fPcm_MemPcm8ToAlaw	将保存在缓冲区中的 8 Bit 无符号 PCM 格式数据转换成 A-law 格式
fPcm_MemAlawToUlaw	将保存在缓冲区中的 A-Law 格式数据转换成 μ-Law 格式
fPcm_MemUlawtoAlaw	将保存在缓冲区中的 μ-Law 格式数据转换成 A-Law 格式

fPcm_MemPcm16ToAlaw	将保存在缓冲区中的 16 Bit 无符号 PCM 格式数据转换成 A-law 格式
fPcm_GC8Convert	将 G.729A 格式的文件转换成其它语音编码格式的文件
fPcm_Vox6KTo8K fPcm_Vox8KTo6K	实现 VOX 格式的语音文件在 6K 与 8K 采样率之间进行转换
fPcm_ULawConvertGSM	将 μ-Law 格式文件转换为 GSM 格式的文件
fPCM_Pcm16ConvertG729A	将 PCM16 格式文件转换为 G.729A 格式文件
fPcm_Pcm8ConvertGSM	将 PCM8 格式文件转换为 GSM 格式的文件
fPcm_GSMToMp3	将 GSM 格式文件转换为 MP3 格式
fPcm_G729AConvert	将 G.729A 格式的文件转换成其它编码格式语音的文件
fPCM_MemULawToG729A	将保存在缓冲区中的 μ-Law 格式数据转换成 G.729A 格式
fPcm_MemG729AToPcm8 fPcm_MemG729AToPcm16	将保存在缓冲区中的 G729A 格式数据转换成 pcm16 或 pcm8 格式
fPcm_NotchFilter_ULAW	将 Synway 板卡录制的 U-Law 格式文件通过点阻滤波算法过滤掉声音文件中的特定频率
fPcm_ALawToAdpcm	将 Synway 板卡录制的 A-Law 格式文件转换为 IMA ADPCM 格式文件
fPcm_MemPcm16ToGSME	将保存在缓冲区中的 pcm16 格式数据转换成 GSM 格式

1.3.2 板卡支持的 CODEC

Synway 板卡的放音操作和录音操作所支持的 CODEC 如下表所示：

系列	板卡型号	PCM8		PCM16		A-law		μ-law		IMA ADPCM		VOX		MP3		GSM		G.729A	
		DEC	COD	DEC	COD	DEC	COD	DEC	COD	DEC	COD	DEC	COD	DEC	COD	DEC	COD	DEC	COD
SHT	SHT-2A/USB	☆	☆	☆	☆	√	√	√	√	—	—	√	√	★	★	★	★	★	—
	SHT-4A/USB	☆	☆	☆	☆	√	√	√	√	—	—	√	√	★	★	★	★	★	—
	SHT-2B/USB	☆	☆	☆	☆	√	√	√	√	—	—	√	√	★	★	★	★	★	—
	SHT-4B/USB	☆	☆	☆	☆	√	√	√	√	—	—	√	√	★	★	★	★	★	—
	SHT-16A-CT/PCI	☆	☆	☆	☆	√	√	√	√	√	★	—	★	★	★	★	★	★	—
	SHT-120A-CT/PCI	☆	☆	☆	☆	√	√	√	√	√	★	—	★	★	★	★	★	★	—
	SHT-8B/PCI	☆	☆	☆	☆	√	√	√	√	√	★	—	★	★	★	★	★	★	—
	SHT-8B/PCI/FAX	☆	☆	☆	☆	√	√	√	√	√	★	—	★	★	★	★	★	★	—
	SHT-16B-CT/PCI	☆	☆	☆	☆	√	√	√	√	√	★	—	★	★	★	★	★	★	—
	SHT-16B-CT/PCI/FAX	☆	☆	☆	☆	√	√	√	√	√	★	—	★	★	★	√	√	★	—
	SHT-16B-CT/PCI/MP3	☆	☆	☆	☆	√	√	√	√	√	★	—	★	★	★	√	★	★	—

	SHD-30A-CT/cPCI/SS7	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	★	★	★	★	-	
	SHD-60A-CT/cPCI/SS7	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	★	★	★	★	-	
	SHD-120A-CT/cPCI/SS7	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	★	★	★	★	-	
	SHD-240A-CT/cPCI	☆	☆	☆	☆	☆	√	√	√	√	√	√	-	-	√	√	√	√	√	-	
	SHD-480A-CT/cPCI	☆	☆	☆	☆	☆	√	√	√	√	√	√	-	-	√	√	√	√	√	-	
	SHD-30B-CT/cPCI/SS7/FAX	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	★	★	★	★	-	
	SHD-60B-CT/cPCI/SS7/FAX	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	★	★	★	★	-	
	SHD-240S-CT/cPCI	☆	☆	☆	☆	☆	√	√	√	√	√	√	-	-	√	√	√	√	√	-	
	SHD-480S-CT/cPCI	☆	☆	☆	☆	☆	√	√	√	√	√	√	-	-	√	√	√	√	√	-	
SHN	SHN-32A-CT/PCI	☆	☆	☆	☆	☆	√	√	√	√	√	√	-	-	☆	-	★	★	★	★	-
	SHN-8B-CT/PCI+	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	★	★	★	★	-	
	SHN-16B-CT/PCI+	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	★	★	★	★	-	
	SHN-32B-CT/PCI+	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	★	★	★	★	-	
	SHN-60B-CT/PCI+	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	★	★	★	★	-	
	SHN-120B-CT/PCI+	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	★	★	★	★	-	
	SHN-60B-CT/PCle+	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	★	★	★	★	-	
	SHN-120B-CT/PCle+	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	★	★	★	★	-	
	SHN-120B-CT/PCle/VAR	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	★	★	★	★	-	
	SHN-480C-CT/PCle	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	★	★	★	★	-	
ATP	SHT-2A/USB	☆	☆	☆	☆	☆	√	√	√	√	√	√	-	-	√	√	★	★	★	★	-
	SHT-4A/USB	☆	☆	☆	☆	☆	√	√	√	√	√	√	-	-	√	√	★	★	★	★	-
	SHT-2B/USB	☆	☆	☆	☆	☆	√	√	√	√	√	√	-	-	√	√	★	★	★	★	-
	SHT-4B/USB	☆	☆	☆	☆	☆	√	√	√	√	√	√	-	-	√	√	★	★	★	★	-
	SHT-8A/PCI	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	★	★	★	★	-	
	SHT-16A-CT/PCI	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	★	★	★	★	-	
	ATP-24A/PCI	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	★	★	★	★	-	
	ATP-24A/PCI+	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	√	★	√	★	-	
	ATP-24A/PCle	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	★	★	★	★	-	
	ATP-24A/PCle+	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	√	★	√	★	√	
	SHT-8B/PCI	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	★	★	★	★	-	
	SHT-16B-CT/PCI	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	★	★	★	★	-	
	SHT-16B-CT/PCI/MP3	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	√	★	√	★	-	
	SHT-16B-CT/cPCI	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	★	★	★	★	-	
	SHT-16B-CT/cPCI/MP3	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	√	★	√	★	-	
DTP	SHD-30A-CT/PCI/FJ	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	★	★	★	★	-	
	SHD-60A-CT/PCI/FJ	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	★	★	★	★	-	
	SHD-30B-CT/PCI/FJ	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	★	★	√	★	-	
	SHD-60B-CT/PCI/FJ	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	★	★	√	★	-	
	DTP-30C/PCI	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	★	★	★	★	-	
	DTP-30C/PCI+	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	√	★	√	★	-	
	DTP-60C/PCI	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	★	★	★	★	-	
	DTP-60C/PCI+	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	√	★	√	★	-	
	DTP-30C/PCle	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	★	★	★	★	-	
	DTP-30C/PCle+	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	√	★	√	★	-	
	DTP-60C/PCle	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	★	★	★	★	-	
	DTP-60C/PCle+	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	√	★	√	★	-	
	DTP-120C/PCle	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	★	★	★	★	-	
	DTP-120C/PCle+	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	√	★	√	★	-	
DST	SHR-16DA-CT/PCI	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	★	★	★	★	-	
	SHR-24DA-CT/PCI	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	★	★	★	★	-	
	DST-24B/PCI	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	★	★	★	★	-	
	DST-24B/PCI(SSW)	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	★	★	★	★	-	
	DST-24B/PCI+	☆	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	√	★	√	★	-	

	DST-24B/PCI+(SSW)	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	√	★	√	★	√
	DST-24B/PCIe	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	★	★	★	★	-
	DST-24B/PCIe+	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	√	★	√	★	√
SHF	SHF-2D/PCI	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	★	★	★	★	-
	SHF-4D/PCI	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	★	★	★	★	-
	SHF-4D/PCIe	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	★	★	★	★	-
IPR	SynIPRecorder	-	☆	-	☆	☆	☆	☆	☆	☆	☆	-	-	-	-	☆	-	☆	☆
HMP	SYNWAY-HMP	☆	☆	☆	☆	√	√	√	√	√	√	☆	-	★	★	★	★	★	-

图例: COD: Coder (编码器)

√: 硬件支持

DEC: Decoder (解码器)

-: 不支持

☆: 软件支持, 由驱动程序通过软件算法实现

★: 软件支持, 通过外挂 ACM 程序实现

注意: 对于 DTP 系列板卡, 更多的说明请特别参考 1.21.4.2 [语音录制](#) 章节。

1.4 SynCTI 驱动程序支持的操作系统

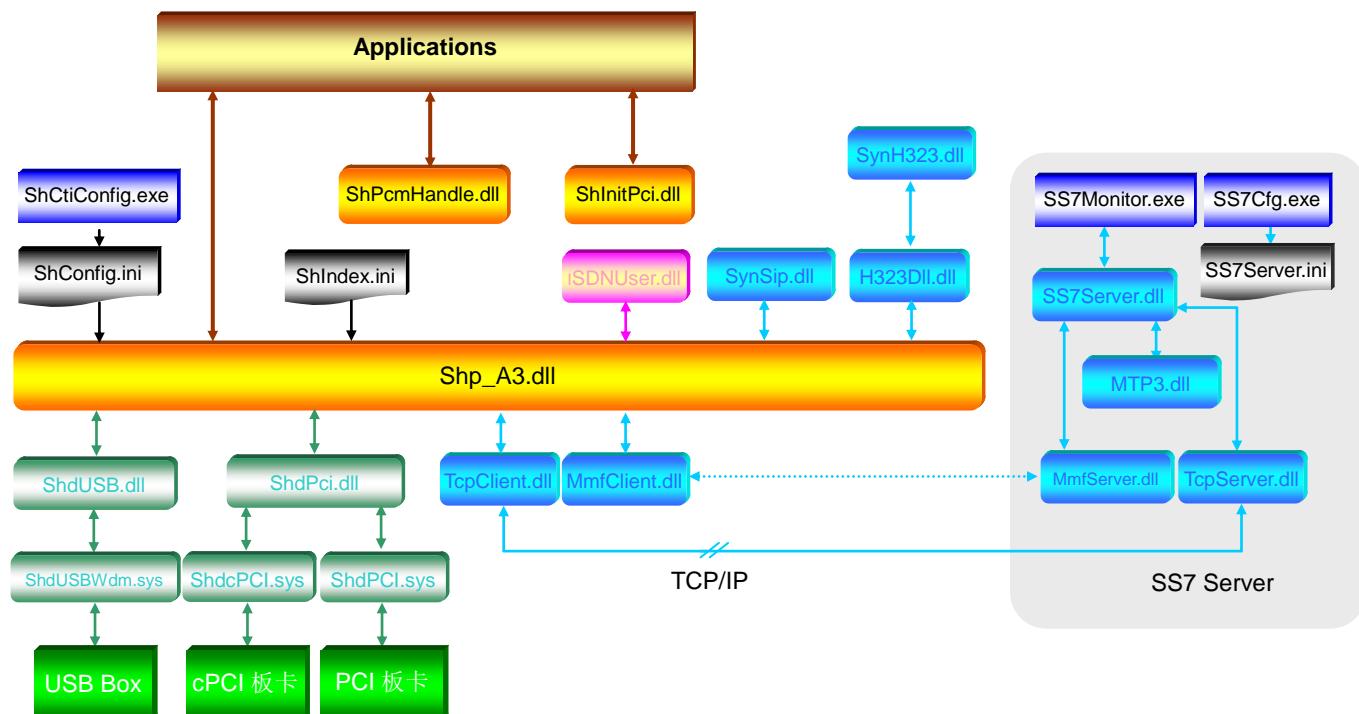
SynCTI 驱动程序支持 WINDOWS 和 LINUX 两大操作系统, 具体如下:

- WINDOWS 操作系统: 支持所有 5.0 及以上版本的 Windows 系统, 支持 32 位及 64 位的 Windows 系统。
包括 Windows 2000、Windows XP 、Windows 2003 server、Vista、Windows 2008 server、Win7
- LINUX 操作系统: 包括 CentOS、Debian、FC4、FC6、FC7、Gentoo、Red Hat A4 U4、Red Hat AS3、RH9.0、Suse、Tribox2.0、Ubuntu

注意: 对于采用 Compact PCI 总线的语音卡, 必须使用 Windows 2000 以上操作系统。

1.5 SynCTI 驱动程序的体系结构 (CTI 系列)

WINDOWS 操作系统下, SynCTI 驱动程序的体系结构如下图所示:

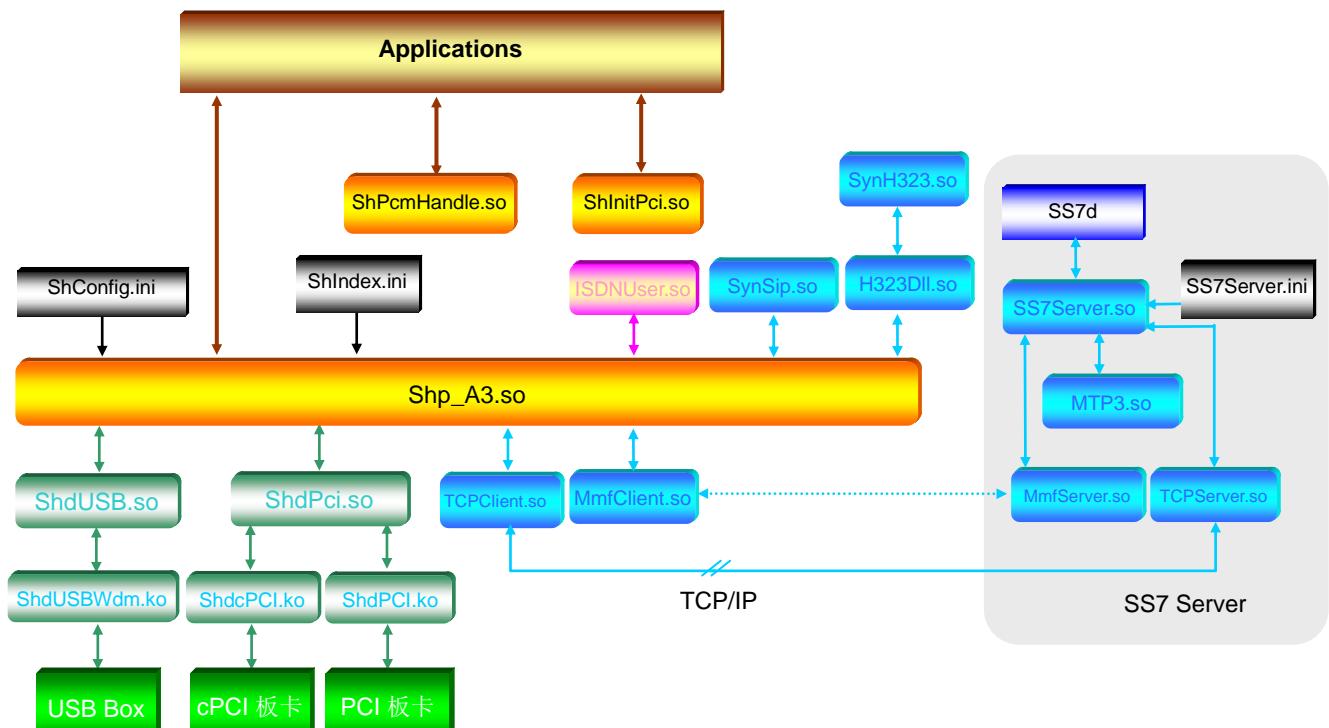


上图中，各模块的功能如下表所示：

模块名称	功能描述
Applications	用户应用程序
ShPcmHandle.dll	为应用程序提供 API 函数，实现各种音频格式的相互转换
ShInitPci.dll	为应用程序提供接口，便于用户获得板卡的基本信息
ShIndex.ini	内存索引语音对照表配置文件
ISDNUser.dll	ISDN 消息处理模块。通过接口函数，对 A3 传递的命令或数据，依照协议进行分析和处理，再将所得结果通过接口函数传递给 A3
ShCtiConfig.exe	自动配置程序
ShConfig.ini	系统配置文件，与 ShCtiConfig.exe 相对应，详细配置信息参见驱动程序安装目录下的 ShConfig 高级选项样本.ini 文件
Shp_A3.dll	为应用程序提供接口 API 函数，帮助实现板卡的各种功能
ShdUSB.dll ShdPci.dll	在 A3 层与 SYS 层间进行数据、状态和命令的传递
ShdUSBWdm.sys ShdcPCI.sys ShdPCI.sys	系统 SYS 层。完成对硬件设备的添加、删除，负责和 DSP 程序进行通信等功能，三个模块分别处理 USB、cPCI、PCI 三种接口形式
SS7Monitor.exe	7 号信令服务器
SS7Cfg.exe	7 号信令配置程序
SS7Server.ini	7 号信令服务器配置文件
SS7Server.dll	7 号信令服务器调度模块，实现 SS7 的信令传递及路由控制等各项功能
MTP3.dll	7 号信令 MTP3 层功能模块
MmfServer.dll	7 号信令单机版服务器端模块，与 SS7Server.dll 及 MmfClient.dll 进行信令和数据的传递
MmfClient.dll	7 号信令单机版客户端模块，与 MmfServer.dll 及 A3 层进行信令和数据的传递
TcpServer.dll TcpClient.dll	7 号信令网络版服务器端模块及客户端模块，两模块间依据 TCP/IP 协议进行数据通信

USB Box、cPCI 板卡、PCI 板卡	接口分别为 USB、cPCI、PCI 形式的语音卡
SynSip.dll	SIP 信令处理模块
H323Dll.dll	专为处理 H.323 协议的软件适配层
SynH323.dll	H.323 消息处理模块

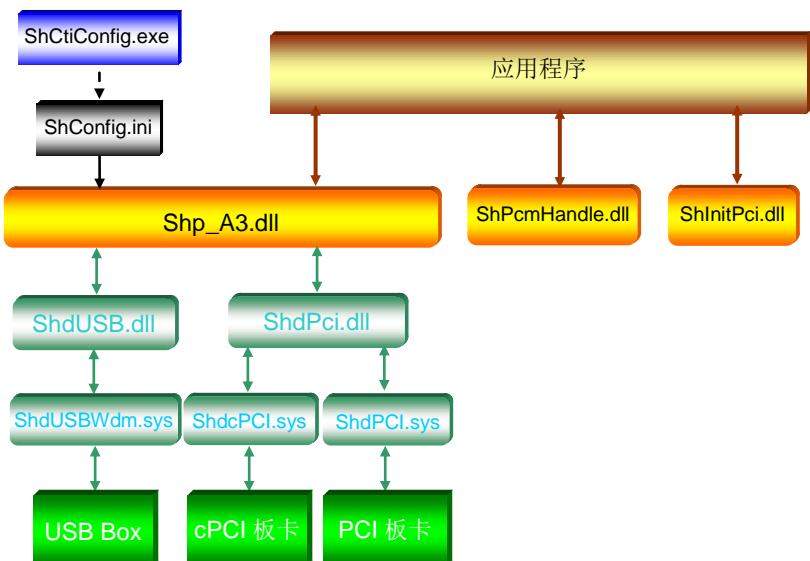
Linux 操作系统下，SynCTI 驱动程序的体系结构如下图所示：



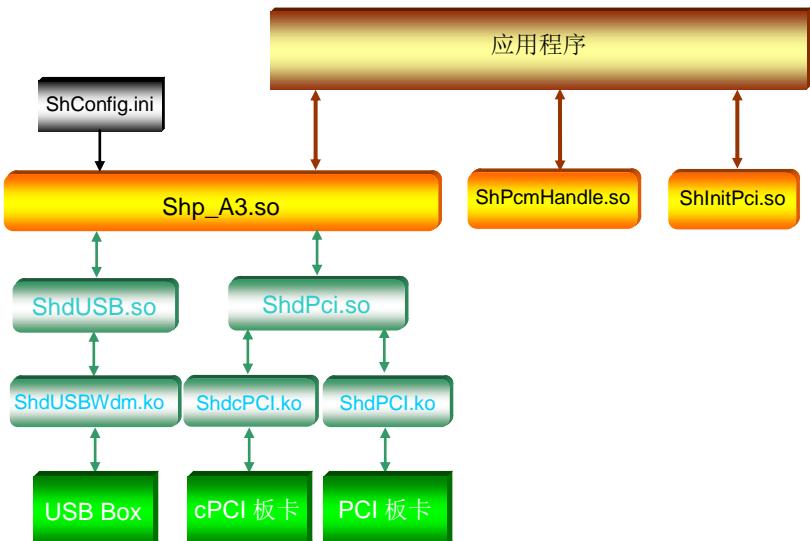
注：图中各模块的功能与 WINDOWS 操作系统下驱动程序体系结构图中的相应模块的功能一致。

1.6 SynCTI 驱动程序的体系结构 (REC 系列)

WINDOWS 操作系统下，SynCTI 驱动程序的体系结构如下图所示：



Linux 操作系统下，SynCTI 驱动程序的体系结构如下图所示：



注：REC 系列驱动程序体系结构图中的各模块功能与 CTI 系列的相应模块的功能一致。

1.7 SynCTI 驱动程序的配套工具软件

1.7.1 系统配置工具软件 ShCtiConfig.exe

ShCtiConfig.exe 是对 ShConfig.INI 文件进行可视化配置的系统配置工具软件。ShCtiConfig.exe 支持对 Synway 所有型号板卡的常用参数配置。在完成 SynCTI 驱动程序的安装后，如果您不想通过手动编辑来取得有效的配置文件，只需运行 ShCtiConfig.exe，也可以进行参数设置。配置程序能够自动从板卡的固件中读取板卡的型号、序列号等信息，并可以自动计算通道数量等基本信息，并提供相应的控制按钮，将配置项的缺省值写入到 ShConfig.INI 文件中。

如果要把驱动安装整合到程序中，同时程序里面要实现缺省配置板卡，可以在启动 ShCtiConfig.exe 的时候添加一个命令行参数：1 为缺省配置->应用->关闭配置程序；2 为缺省配置->确定->关闭配置程序，使用方法：
`CreateProcess("C:\shcti\ShCtiConfig.exe", "1", NULL, NULL, FALSE, 0, NULL, "C:\shcti", &si, &pi);`

ShCtiConfig.exe 的主要功能包括：

- 对 PCI 板卡常用参数的可视化配置。
- 对 USB 语音盒常用参数的可视化配置。
- 对 cPCI 板卡常用参数的可视化配置和热插拔管理。

1.7.2 信号音分析工具软件 ShTA.exe

在使用SHT系列、ATP系列板卡时，如果在模拟中继线上出现呼叫不成功、无法检测到对端挂机等情况，往往是由于呼叫进程音检测器的工作参数与交换机侧的参数不匹配所导致。某些小型交换机的信号音频率、周期与电信局用交换机的标准可能有差别，而Synway板卡上信号音检测器的工作参数是按照标准的局用交换机参数进行设置，因此，需要正确设置相关参数，如信号音的频率，拨号音、回铃音、忙音的占空比等，可以使用信号音分析工具软件ShTA.exe来对交换机侧的呼叫进程音的参数进行检测与分析，获得正确的设置值。

ShTA.exe 的具体使用方法请参考信号音分析工具软件用户手册，即 SynCTI 驱动程序安装目录下的 ShTA_UserManual_en.chm 文件（英文）或 ShTA_UserManual_cn.chm 文件（中文）。

1.7.3 SS7 信令服务器配置程序 SS7Cfg.exe

SS7Cfg.exe 是用于设置 SS7 信令服务器程序的配置文件 Ss7server.ini 的工具软件，具体的使用方法请参见文档《SS7 信令服务器配置程序用户手册》，即 SynCTI 驱动程序安装目录下的 SS7Cfg_UserManual.chm 文件（英文）或 SS7Cfg_UserManual_cn.chm 文件（中文）。

1.7.4 SS7 信令服务器 SS7monitor.exe

Ss7Monitor.exe 实现 SS7 信令的 MTP3 协议，同时具有信令消息分发和系统监视的作用。如果应用系统需要使用 SS7 信令，必须首先运行 SS7 信令服务器。

1.7.5 MSU 解码工具软件 MsuDecode.exe

MsuDecode.exe 是用来对 SS7 信令服务器的日志文件中的消息（MSU）进行翻译的工具软件。Ss7Monitor.exe 在运行时，可以根据用户的要求生成扩展名为.msu 的日志文件，将需要保存的消息以二进制的方式保存到文件中。MSU 解码工具软件可以将二进制形式的消息翻译成容易理解的文本形式的消息文件（.txt），可以直接进行阅读。有关 MSU 解码工具软件的具体使用方法请参考《MSU 解码工具软件用户手册》，即 SynCTI 驱动程序安装目录下的 MsuDecode_UserManual.chm 文件（英文）或 MsuDecode_UserManual_cn.chm 文件（中文）。

1.8 基本概念

1.8.1 通道（CTI 系列）

指板卡上用于处理一路电话的、具有完整的语音处理能力的物理电路实体。根据线路接口类型和信令的不同，通道的分类如下表所示：

板卡系列	业务模块 / 信令协议	通道类型	类型编码
SHT	外线模块	模拟中继线通道	0
	坐席模块	坐席通道	2
	录音模块	录音通道	3
	麦克风模块	录音通道	3
	磁石模块	磁石通道	10
	未安装任何业务模块	无模块通道	20

	传真资源通道	传真通道	9
SHD	SS1 信令	SS1 通道	4
	ISDN 信令（用户侧）	ISDN 通道（用户侧）	7
	ISDN 信令（网络侧）	ISDN 通道（网络侧）	8
	SS7 信令-TUP 协议	TUP 通道	6
	SS7 信令-ISUP 协议	ISUP 通道	11
	传真资源通道	传真通道	9
SHV		变声资源通道	14
SHN	SIP 信令	SIP 类型 IP 通道	16
SHF	外线模块	模拟中继线通道	0
	坐席模块	坐席通道	2
	传真资源通道	传真通道	9
HMP	SIP 信令	SIP 类型 IP 通道	16

注：类型编码可以通过函数 [SsmGetChType](#) 获取。

1.8.2 通道 (REC 系列)

指板卡上用于处理一路电话的、具有完整的语音处理能力的物理电路实体。根据线路接口类型和信令的不同，通道的分类如下表所示：

板卡系列	业务模块 / 信令协议	通道类型	类型编码
ATP	录音模块	模拟中继线录音通道	3
	麦克风模块	麦克风录音通道	3
DST		数字电话录音通道	12
DTP	SS1	SS1 监控通道	4
	ISDN	ISDN 监控通道	7 或 8
	SS7-TUP	TUP 监控通道	6
	SS7-ISUP	ISUP 监控通道	11
	DASS2	DASS2 监控通道	19
	DPNSS	DPNSS 监控通道	27
IPR	IPRA_CH	Analyzer 通道	26
	IPRR_CH	Recorder 通道	25

注：类型编码可以通过函数 [SsmGetChType](#) 获取。

1.8.3 通道的逻辑编号

驱动程序中，每个通道都有 2 个编号：物理编号和逻辑编号。

物理编号（简称 BCh）是指通道在所在板卡上的编号。通道物理编号由驱动程序自动分配，假设 N 表示板卡上的总通道数，M 表示总传真通道数，则板卡上通道的物理编号规则为：

板卡系列	通道的物理编号规则
SHT 系列	型号中没有 “/FAX” 标记：0~N-1 型号中包含 “/FAX” 标记：语音通道为 0~N-M-1，传真通道为 N-M~N-1
SHD 系列	0~N-1
SHV 系列	0~N-1
SHN 系列	0~N-1
ATP 系列	0~N-1
DTP 系列	CIC 编号：0~N-1 通道编号：0~N * 2 - 1
DST 系列	0~N-1
SHF 系列	语音通道为 0~N-M-1，传真通道为 N-M~N-1
IPR 系列	0~N-1

逻辑编号是指在应用系统中，为所有板卡上的全部通道进行统一编号。配置项 [TotalAppCh](#) 用于设置系统中的通道总数。本驱动程序中大部分函数使用的参数 ch 均指通道的逻辑编号。

通道的逻辑编号与物理编号之间是一一对应关系，由配置项 [AppCh](#) 决定。

注：在本手册中，如果不是特别说明，通道编号就是指通道的逻辑编号。

1.8.4 数字中继线

将用户终端设备（如用户交换机、集团电话、呼叫中心等）接入公用电话网络的电话交换机的线路称为中继线。数字中继线是为用户提供数字线路联入电信网的中继线，通常为一对引自程控交换机的同轴电缆线或双绞线，一条用于发送，另一条用于接收。如果在电缆线上数据传输速率是 2.048 Mbps（简称 2M 线），中继线与用户终端设备之间的接口就称为 E1 接口；如果传输速率是 1.544Mbps，则称为 T1 接口。

数字中继线采用时分复用的工作方式，由多个时隙（Time Slot）组成，每个时隙的传输速率为 64 Kbps。E1 中继线具有 32 个时隙，T1 中继线具有 24 个时隙。

对于 E1 中继线，32 个时隙中的第 0 时隙被用作帧同步信息，因此通常只有 31 个有效时隙，其中，第 16 时隙通常被用于传输信令信息以及复帧同步信息，称为 D 信道；其余 30 个时隙被用作语音数据的传输时隙，称为 B 信道。

在本手册中，每一对 E1/T1 数字中继线均称为一条数字中继线。

1.8.4.1 数字中继线的帧同步

数字中继线的时隙 0 通常用来传递帧同步信息，因此不能作为语音时隙使用。SHD 系列和 DTP 系列的每条数字中继线端口都安装了帧同步指示灯，用于指示链路层的帧同步情况，灯亮表示帧同步正常，灯灭或闪烁表示帧同步失败。

如果使用 CAS 信令，复帧同步信息也在 0 时隙中传送。

当数字中继线的帧同步状态发生改变后，驱动程序会向应用程序抛出 [E_CHG_PcmLinkStatus](#) 事件。应用程序也可以随时调用函数 [SsmGetPcmLinkStatus](#) 获取数字中继线的当前状态。

1.8.4.2 数字中继线的时钟

每一条数字中继线都需要时钟才能正常工作，更多信息请参见本章中“[配置应用系统的时钟](#)”部分内容。

1.8.4.3 数字中继线的配置

本端的每条物理数字中继线必须与对端交换机的数字中继线的配置保持一致才能正常工作。

对数字中继线的设置是以板卡为基础，在**[BoardId=x]** 节中进行设置。包括下列配置项：

[BoardId=x]

.....

PcmNumber=M //设置板卡内数字中继线的总数

PcmSSx[0]=1 //设置板卡内第 0#数字中继线：选择信令方式

PcmClockMode[0]=j //选择时钟的工作模式

PcmLinkType[0]=k //选择线路的物理传输类型

.....

1.8.4.4 数字中继线中的信令时隙

数字中继线中的信令时隙与协议类型有关，如下表所示：

协议类型	信令时隙
No.1 信令	第 16 时隙用来传送 ABCD 信令码
ISDN PRI	第 16 时隙用来传输信令消息
SS7 信令	通常将第 16 时隙用作信令链路。通过修改配置项 UseTS16AsCircuit 和 Ss7SignalingTS 也可以使用其它的时隙来承载信令链路。更多信息请参见本章中“ 时隙分配 ”部分内容

1.8.4.5 数字中继线的编号

数字中继线有 2 个编号：物理编号和逻辑编号。

PCM 的物理编号是指该数字中继线在板卡内部的编号。板卡上的数字中继线从 0 开始编号，最大编号为板卡的数字中继线总数减 1。数字中继线的逻辑编号，是指在整个应用系统内，将所有板卡上数字中继线进行统一编制的号码。当应用程序内只有一片板卡时，数字中继线的逻辑编号与物理编号相同。

数字中继线的物理编号与逻辑编号之间的映射关系可以通过配置项 [TotalPcm](#) 和 [Pcm](#) 进行指定。

注意：在本手册中，如果不是特别说明，“PCM 编号”是指数字中继线的逻辑编号。

1.8.4.6 设置 B 信道的语音编码格式

配置项 [DefaultVoiceFormat](#) 用来设置数字用户线上的语音编码格式。

1.8.5 板卡的用户授权号

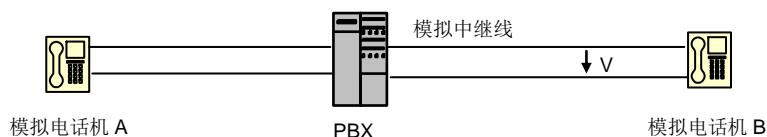
Synway 板卡上设计了具有加密功能的授权号电路，除了生产厂家之外，任何人不可能修改这一授权号。利用授权号功能，用户可以将应用程序与具体板卡绑定，相当于给用户软件装上了软件狗。使用了授权号后，您使用的 Synway 板卡就相当于您向三汇公司定做的一个特殊型号板卡。采用授权号保护用户的软件有以下优点：

- 操作简便：用户只要先向三汇公司申请一个授权号，然后由三汇公司将授权号写入提供给该用户的产品上，用户软件只要检查授权号是否符合即可，不需要任何算法，并且用户软件每次安装完毕不必作任何设置，不像采用系列号进行软件保护时需要进行复杂的加密计算，并且每次安装还要根据序列号进行一系列设置。
- 可靠性高：采用序列号进行软件保护时，如果加密算法外泄，则保护失效。而采用授权号方式，三汇公司在技术上保证除了生产厂家之外，没有任何人可以改写授权号，因此，对于申请了授权号的用户，即使授权号是公开的，其软件也绝对不可能运行在从其它渠道获得或者是三汇公司提供给其它用户的语音卡产品上。

函数 [SsmGetAccreditId](#) 或 [SsmGetAccreditIdEx](#) 可以用来获取板卡上的用户授权号。

1.8.6 模拟电话线的电压变化

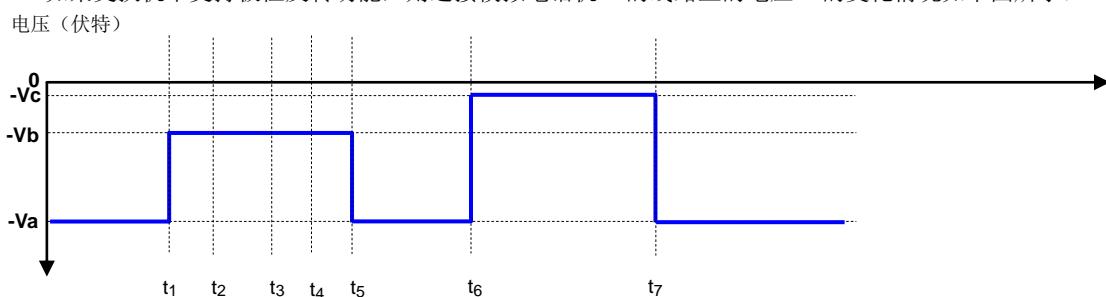
模拟电话线上的直流工作电压为-48V，由交换机提供。Synway 的 SHT/ATP 系列板卡的每个通道上都具有电压检测器，用于检测模拟中继线上工作电压。下图是一个典型的电话连接示意图：



如果在模拟电话机 B 上进行如下操作：

- (1) 在 t_1 时刻，B 摘机；
- (2) 在 t_2 时刻，B 完成拨号，交换机向 A 振铃，向 B 送回铃音；
- (3) 在 t_3 时刻，A 摘机，双方进入通话；
- (4) 在 t_4 时刻，通话结束，A 挂机，交换机开始给 B 送忙音；
- (5) 在 t_5 时刻，B 挂机；
- (6) 在 t_6 时刻，B 与交换机的线路被断开；
- (7) 在 t_7 时刻，B 与交换机的线路恢复连接。

如果交换机不支持极性反转功能，则连接模拟电话机 B 的线路上的电压 V 的变化情况如下图所示：



上图中，

V_a : 电话机处于挂机状态时的线路电压，一般为-48V。

V_b : 电话机处于摘机状态时的线路电压，对于 Synway 板卡，一般为-15V 左右。

V_c : 电话机的连线断开时，理论值为 0V。

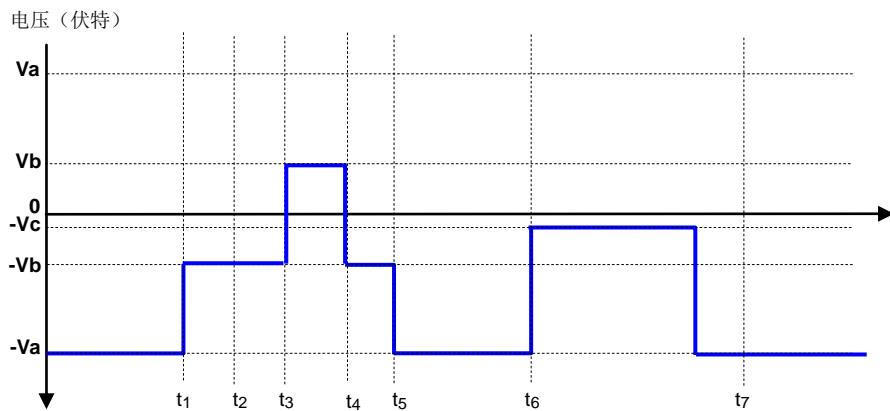
电压检测器利用线路上的实际电压值（假设为 V_{in} ）与 V_a 、 V_b 、 V_c 之间的关系来判断线路及话机的状态，判定标准可以表述为：

- ◊ 摘机：如果 $V_c < V_{in} < V_b$ ，并且持续一定时间 $t_{offhook}$ 。
- ◊ 挂机：如果 $V_b \leq V_{in}$ ，并且持续一定时间 t_{onhook} 。
- ◊ 断线：如果 $V_{in} \leq V_c$ ，并且持续一定时间 t_{off} 。

SynCTI 驱动程序中，用于设置上述参数的配置项或函数如下表所示：

参数名称	相关配置项或函数
V_c	配置项： OffLineSet
t_{off}	配置项： OffLineDetermineTime
V_b	配置项： IsHangupDtrmVoltage 函数： SsmSetDtrmLineVoltage
$t_{offhook}$	配置项： LineOncnt
t_{onhook}	配置项： MaxRecChFlashFilterTime

如果交换机支持极性反转功能，则连接模拟电话机 B 的线路上的电压 V 的变化情况如下图所示：



从上图可以看出，如果交换机支持极性反转功能，当被叫摘机或挂机时，交换机会变换主叫方的线路电压的极性，以向主叫方传递被叫的摘机/挂机信息。对于模拟线路，利用线路上电压的极性反转信号来判断被叫是否摘机是最准确的，但需要交换机的支持。

Synway 的 SHT 和 ATP 系列板卡均内置了电压检测器。每当线路上的电压发生变化，并且变化的幅度超过预设值时，驱动程序会向应用程序抛出 [E_CHG_LineVoltage](#) 事件。电压变化的预设值为 5V，可以通过调用函数 [SsmSetEvent](#)（携带 E_CHG_LineVoltage 参数）进行设置。应用程序也可以通过函数 [SsmGetLineVoltage](#) 获取线路的电压值。

电压检测器的作用：

- 监测本端与交换机端的线路连接是否正常。

ATP 系列的模拟中继线录音通道均支持断线监测功能。当线路电压低于配置项 [OffLineSet](#) 的设定值，并且保持时间大于配置项 [OffLineDetermineTime](#) 的设定值时，驱动程序会判定发生了断线故障，会将通道的状态迁移到“断线”状态。

- 判断线路上的摘机和挂机动作。

ATP 系列板卡的录音通道是根据线路电压(绝对值)在 V_a 与 V_b 之间的变化情况，来判断话机是否摘机或挂机。

在缺省情况下，模拟录音通道会根据电压检测器的输出来判断话机的摘机/挂机状态，但通过修改函数 [SsmSetIgnoreLineVoltage](#) 或配置项 [IgnoreLineVoltage](#) 的设置，可以让驱动程序忽略电压检测器的检测结果，使录音通道一直保持在摘机状态。

对于麦克风录音模块，由于线路上没有馈电电压，因此，应该将配置项 [IgnoreLineVoltage](#) 设置为 1，这样可以使录音通道一直保持在“摘机”状态。

- 检测极性反转信号

模拟电话线上的馈电电压由交换机提供，通常为负极性，交换机可以改变馈电电压的极性（比如将 $-V_b$ 改变为 V_b ），通过极性的变化向电话机传递一些信息。

利用模拟电话线的极性反转信号可以准确地判断被叫的摘机和挂机动作，但需要交换机的配合。ATP 系列的模拟电话线录音通道和 SHT 系列的模拟中继线通道均支持极性反转检测功能。每当通道上检测到极性反转后，驱动程序会向应用程序抛出 [E_CHG_PolarRvsCount](#) 事件，应用程序也可以通过函数 [SsmGetPolarRvsCount](#) 获取极性反转信息。

在有些情况下，由于交换机或者线路上的干扰比较大，导致驱动程序将干扰信号误以为极性反转信号，为了避

免这种情况发生，可以通过设置配置项 [PolarIgnore](#) 来忽略线路上较大的干扰信号。

对于模拟中继线通道，当检测到极性反转信号后，通道的状态是否发生迁移，取决于配置项 [DisablePolarReverse](#) 的设置。

1.8.7 模拟电话线上的铃流信号

当交换机通过模拟电话线叫话机时，会向话机发送铃流信号，使话机开始响铃。铃流信号通常是 1 秒高电平、4 秒低电平的周期性信号。

ATP 系列板卡的模拟中继线录音通道和 SHT 系列板卡的模拟中继线通道都具有铃流信号检测器。配置项 [RingDetectFilterPara](#) 用于设置铃流信号高电平和低电平的最小保持时间。配置项 [RingEndDtrTime](#) 用于设置一个铃流信号高低电平的最大保持时间。每当铃流信号的电平发生变化时，驱动程序会向应用程序抛出 [E_CHG_RingFlag](#) 事件。

每当驱动程序检测到一次完整的铃流信号，都会向应用程序抛出 [E_CHG_RingCount](#) 事件。当铃流信号的累计个数到达指定值 N 时，通道状态会从“空闲”迁移到“振铃”。N 可以通过配置项 [AlwaysToRingingOnRingCntX](#) 或 [ChToRingingOnRingCnt](#) 进行设置。实际的 N 按照下列规则计算：

- (1) 如果没有在配置项 [ChToRingingOnRingCnt](#) 中指定或通过函数 [SsmSetFlag](#)（携带 [F_ChToRingingOnRingCnt](#) 参数）设置，以配置项 [AlwaysToRingingOnRingCntX](#) 的设定值为准。
- (2) 如果没有在配置项 [AlwaysToRingingOnRingCntX](#) 中指定，由驱动程序自动确定，此时 N 为 2。

应用程序也可以调用函数 [SsmGetRingCount](#) 来获取铃流信号的个数。

当满足下列条件时，驱动程序会自动将铃流信号的计数值清零：

- ◆ 通道从“振铃”状态迁移到“空闲”状态。
- ◆ 通道从“振铃”状态迁移到“摘机”状态，并且在“摘机”状态的保持时间超过配置项 [RecChClearRingDelayTime](#) 的设定值。

应用程序可以调用函数 [SsmClearRingCount](#) 将计数值清零。

1.8.8 模拟电话线上的闪断信号

闪断信号是指电话机进入通话后，通过快速拍打模拟电话机的叉簧而产生的短时挂机信号，通常用于与交换机配合实现转分机等操作。在话机上拍下叉簧的时间应该足够短，以免交换机将闪断信号误判为挂机信号。

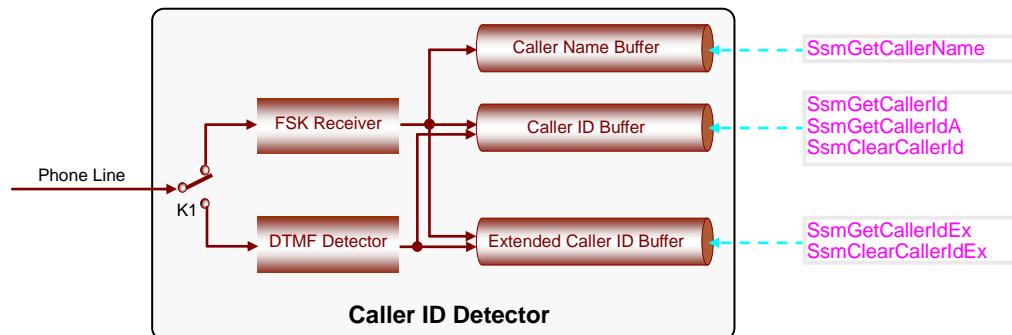
1.8.9 模拟电话线上的主叫号码

电话机的来电显示（Caller ID，主叫号码）是电信等运营商提供的一项付费业务，通常需要向其申请后才会获得这项功能。交换机向电话机传输主叫号码的方式有两种：

- ◆ FSK（频移键控）制式：Caller ID 是由交换机在给电话机的第一声振铃和第二声振铃之间进行传送。
- ◆ DTMF（双音多频）制式：在多数情况下，Caller ID 是由交换机在给电话机发送第一声振铃信号之前进行传送，但也有少数交换机在第一声振铃信号和第二声振铃信号之间向电话机发送。

一般的经验是，市局级交换机给模拟话机提供的来电显示号码是 FSK 制式传输的，小型交换机可能会以 DTMF 方式来传输。**注意：**对于应用程序而言，如果需要完整地接收主叫号码，必须第二声铃声之后才能摘机。模拟外线通道在第几声振铃信号后才进入“振铃”状态，可以在配置文件中进行配置。

Synway 的 SHT / ATP 系列板卡都具有 Caller ID 检测器, 用于检测模拟电话线上的 FSK 制式/DTMF 制式的 Caller ID 信号, 其原理框图如下图所示:



驱动程序内部为每个通道都设置了 2 个存放 Caller ID 的缓冲区: 主叫号码缓冲区 (Caller ID Buffer) 和主叫号码扩展缓冲区 (Extended Caller ID Buffer)。主叫号码扩展缓冲区用于存放 FSK 制式的主叫号码的原始数据, 里面包含了主叫号码、呼叫日期和时间、主叫方姓名等信息。驱动程序对原始信息进行下述处理:

- ◆ 分离出主叫方号码, 并将其保存到主叫号码缓冲区中。
- ◆ 分离出主叫方姓名, 并将其保存到主叫方姓名缓冲区中。

有关 Caller ID 检测器的函数、事件和配置项如下表所示:

类别	名称	描述
配置项	CallerIdStyle	设置 Caller ID 检测器的工作模式 (FSK 制式 / DTMF 制式), 即上图中的 k1 开关
	CloseCallerIdOnReceived FSKCallerIdDtrmTime FilterInvalidCID	设置 Caller ID 检测器工作在 FSK 制式时的工作参数
	DtmfCallerIDStyleLength DtmfCallerIDInterTimeout	设置 Caller ID 检测器工作在 DTMF 制式时的工作参数
函数	SsmSetFlag (携带 F_CALLERIDTYPE 参数)	设置 Caller ID 检测器的工作模式
	SsmGetCallerId SsmGetCallerIdEx SsmGetCallerIdA	获取主叫方号码
	SsmGetCallerName	获取主叫方姓名
	SsmClearCallerId SsmClearCallerIdEx	清空驱动程序内部保存主叫号码的缓冲区
事件	E_CHG_CIDExBuf	Caller ID 检测器工作在 FSK 制式时, 每当驱动程序完成一次主叫号码的接收后, 会向应用程序抛出此事件

1.9 配置应用系统的时钟

每片 Synway 板卡都需要提供时钟信号才能正常工作。在设置基于 Synway 板卡的应用系统的时钟信号源时, 需要考虑下列因素遵循下列原则:

- ◆ 如果板卡之间通过 CT-BUS 进行互联:
 - ◆ 只能由其中一片板卡提供唯一的时钟信号源, 称为主时钟(Master Clock), 提供主时钟的板卡称为主卡 (Master Board), 其它板卡称为从卡(Slave Board), 从卡的时钟信号通过 CT-BUS 从主卡获得, 称为从时钟 (Slave Clock)。

- ◆ 如果应用系统中至少包括一片 SHD 系列板卡，必须选择一片 SHD 系列板卡作为主卡；如果不包含任何 SHD 系列板卡，但包含 DST 系列或 DTP 系列板卡，应选择一片 DST 系列或 DTP 系列板卡作为主卡。
- ◆ 如果主卡是 SHD 系列，主时钟信号只能由主卡上的一条数字中继线提供，主卡上的其余数字中继线必须工作在从时钟模式。
- ◆ SHD 系列、DTP 系列板卡的时钟信号必须与公共电信网络保持一致，否则可能出现通话中有杂音、信令差错率高等不正常现象。
- ◆ 如果板卡没有通过 CT-BUS 与其它板卡进行互联，该卡必须设置为主卡。
- ◆ 通过 CT-BUS 与第三方板卡互联：如果将第三方板卡设置为主卡，需要将 Synway 板卡全部设置为从卡。

主卡提供的主时钟信号可以有 2 种来源方式：线路同步方式和自振荡方式。线路同步方式是指时钟信号提取自对端交换机连接到本端的输入信号，适用于 SHD 系列、DTP 系列和 DST 系列板卡，以保证与对端交换机的时钟信号保持同步；自振荡方式是指由板卡上内置的时钟振荡电路产生时钟信号，适用于 SHT 系列、ATP 系列板卡。如果使用 SHD 系列板卡来搭建一个测试系统，与实际的应用系统相连，则应该将测试系统的主卡的主时钟设置为自振荡模式。

SHV 系列板卡不能作为主卡，必须工作在从时钟模式。

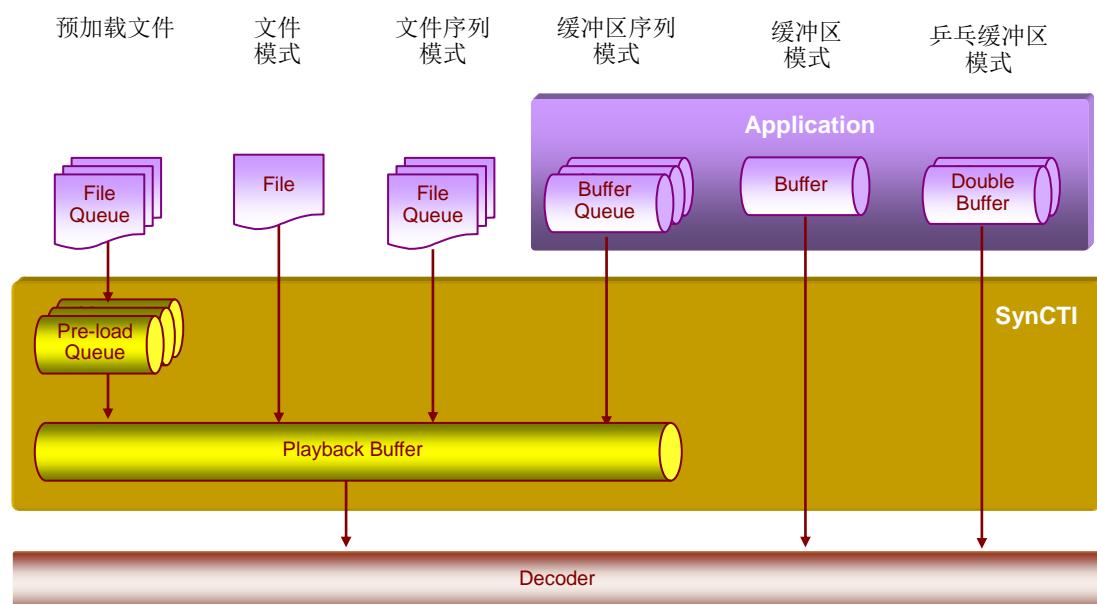
与时钟信号相关的配置项和函数如下表所示：

类别	名称	描述
配置项	WhoSupplySysClock	设置板卡的时钟模式（主卡/从卡）
	PcmClockMode	设置 SHD、DTP 系列板卡上数字中继线的时钟模式
函数	SsmSetPcmClockMode	设置 SHD、DTP 系列板卡上数字中继线的时钟模式，与配置项 PcmClockMode 的功能完全相同

1.10 语音处理

1.10.1 语音播放（CTI 系列）

语音播放包括 6 种模式，如下图所示：



对于预加载语音片段、单个文件、文件序列、缓冲区序列这 4 种播放模式，放音操作启动后，驱动程序会先将源数据读取到内部的放音缓冲区(Playback Buffer)。放音缓冲区大小通过配置项 [PlayBufSize](#) 指定，缺省值为 32K 字节。

1.10.1.1 设置放音数据音量及去向

1.10.1.1.1 CTI 系列

有关放音数据的音量及其去向的详细内容，请参见本章中相应内容：

- ◆ SHD 系列板卡：请参见本章中 [SHD 系列板卡的工作原理框图](#) 部分。
- ◆ SHT 系列板卡：请参见本章中 [SHT 系列板卡的工作原理框图](#) 部分。
- ◆ SHN 系列板卡：请参见本章中 [SHN 系列板卡的工作原理框图](#) 部分。

1.10.1.1.2 REC 系列

有关放音数据的音量及其去向的详细内容，请参见本章中相应内容：

- ◆ ATP 系列板卡：请参见本章中 [ATP 系列板卡的工作原理框图](#) 部分。
- ◆ DST 系列板卡：请参见本章中 [DST 系列板卡的工作原理框图](#) 部分。
- ◆ DTP 系列板卡：请参见本章中 [DTP 系列板卡的工作原理框图](#) 部分。

1.10.1.2 设置播放任务的终止条件

放音任务启动后，除了播放任务正常终止以及被应用程序调用停止播放的函数所终止外，如果通道上检测到了符合特定条件的事件，驱动程序也会自动终止放音任务，从而简化应用程序的编写。这些条件包括：

- DTMF 检测器在来话信号中检测到 DTMF 字符

相关函数及配置项包括：

函数名称	功能简述	相关配置项
SsmSetDtmfStopPlay	设置放音任务是否因 DTMF 检测器 检测到 DTMF 字符而终止	DtmfStopPlayCharSet
SsmSetDTMFStopPlayCharSet	设置终止放音任务的 DTMF 字符集	
SsmGetDtmfStopPlayFlag	获取放音任务是否因 DTMF 检测器检测到 DTMF 字符而终止的标志	
SsmGetDTMFStopPlayCharSet	获取终止放音任务的 DTMF 字符集	

注意：当 DTMF 检测器在来话信号中检测到 DTMF 信号时，如果配置项 [DefaultPausePlayOnRxDtmf](#) 设置为 1，

为了保证 DTMF 检测器的准确性，驱动程序会暂时停止语音的播放，待 DTMF 信号消失后，再自动恢复语音的播放。

- Barge in 检测器在线路上检测到了 Barge in（语音活动）

函数名称	功能简述	相关配置项
SsmSetBargeinStopPlay	设置放音任务是否因 Barge in 检测器 检测到语音活动而终止	DefaultBargeInStopPlay
SsmGetBargeinStopPlayFlag	获取放音任务是否因 Barge in 检测器 检测到语音活动而终止的标志	

- 呼叫状态机检测到对端挂机

函数名称	功能简述	相关配置项
SsmSetHangupStopPlayFlag	设置放音任务是否因驱动程序状态机检测到对端挂机而终止	HangupStopPlay

- 应用程序在通道上调用了会引发下总线操作的总线操作函数

SHD 系列中某些型号板卡的特有属性，详细内容请参见本章中“[SHD 系列板卡的原理框图](#)”部分。

1.10.1.3 播放预加载文件

对于 IVR 等应用，通常会需要应用程序播放一些常用的语音片段，例如提示语、合成日期、时间或数字等。为了提高应用系统的运行效率，降低应用程序访问硬盘的次数，SynCTI 驱动程序提供了预加载文件模式，其工作原理为：事先将“0”、“1”、“2”……、“9”、“十”、“百”、“千”、“万”等常用语音片段录制到一个或多个文件中，并且为每条语音片段定义一个编号和别名，然后在调用驱动程序平台的初始化函数 [SsmStartCti](#) 时，由驱动程序自动将这些语音片段预先加载到内存，就可方便地播放诸如“125”等数字的连续语音。通过预加载文件模式，驱动程序可使语音片段实现“无缝连接”，效果大大优于使用一批单独的语音文件进行顺序播放。

驱动程序最多支持 65536 个预加载语音片段。

预加载语音片段可以通过编写配置文件 ShIndex.ini 实现。有关 ShIndex.ini 文件的详细信息请参见第 3.2 节的“[预加载语音配置文件 ShIndex.ini \(CTI 系列\)](#)”。

应用程序也可以通过调用函数来动态加载语音片段。相关函数见下表所示：

- 有关加载和卸载语音片段的函数

函数名称	描述
SsmSetMaxIdxSeg	设置预加载语音片段的总数
SsmGetTotalIndexSeg	获取预加载语音片段的总数
SsmLoadIndexData	将指定的语音片段加载到内存

SsmFreeIndexData	卸载已经加载到内存的语音片段
----------------------------------	----------------

- 有关播放预加载语音片段的专用函数

函数名称	描述
SsmPlayIndexList	顺序播放指定的一个或多个预加载语音片段
SsmPlayIndexString	顺序播放指定的一个或多个预加载语音片段
SsmStopPlayIndex	停止播放预加载语音片段

1.10.1.4 播放单个文件

播放单个文件，是指对单个语音文件进行播放。支持的文件格式包括标准的 WAV 文件、MP3 文件和无格式文件（没有文件头的文件，编码格式需要另外指定）。

播放单个文件的相关函数及事件如下表所示：

函数或事件名称	描述
SsmPlayFile	播放单个文件，启动文件播放任务
SsmStopPlayFile	终止文件播放任务
SsmPausePlay	暂停（挂起）文件播放任务
SsmRestartPlay	恢复因为调用函数 SsmPausePlay 而暂停的文件播放任务
SsmFastFwdPlay	跳过 1 秒钟的语音数据后继续播放
SsmFastBwdPlay	回退 1 秒钟的语音数据后继续播放
SsmSetPlayTime	跳转到文件中的指定时间位置后继续播放
SsmSetPlayPrct	跳转到文件中的指定百分比位置后继续播放
SsmGetDataBytesToPlay	查询尚未播放的语音数据字节数
SsmGetDataBytesPlayed	查询已经完成播放的语音数据长度
SsmGetPlayingFileInfo	获取正在播放的放音文件的相关信息
SsmGetPlayedTimeEx	获得从文件首部的第一个语音数据字节开始，到文件中的当前放音位置的播放持续时间
E_PROC_PlayFile	文件播放任务的进展事件

1.10.1.5 播放文件序列

播放文件序列，是指应用程序连续提交多个具有相同语音编码格式的语音文件，由驱动程序负责按照提交的先后顺序，实现文件的连续自动播放，文件之间可以实现“无缝连接”。

函数 [SsmAddToFileList](#) 用来向驱动程序提交需要播放的文件。可以提交的文件数量的最大值由配置项 [MaxPlayFileList](#) 决定。

当应用程序调用 [SsmPlayFileList](#) 函数后，驱动程序内部就启动了一个文件序列播放任务，并开始播放第 1 个文件。每当完成一个文件的播放后，驱动程序会向应用程序抛出 [E_PROC_PlayFileList](#) 事件，然后开始播放下一个文件。当全部文件播放完毕后，驱动程序会自动关闭全部文件，终止文件序列播放任务，并向应用程序抛出 [E_PROC_PlayEnd](#) 事件。

播放文件序列的相关函数及输出事件如下表所示：

函数或事件名称	描述
SsmAddToFileList	向驱动程序提交一个需要播放的语音文件
SsmPlayFileList	启动文件序列播放任务
SsmClearFileList	关闭并清除已提交给驱动程序的全部语音文件
SsmStopPlayFileList	终止文件序列播放任务
E_PROC_PlayFileList	完成一个语音文件的播放后，驱动程序抛出的事件

1.10.1.6 播放单个内存

播放单个内存，是指由应用程序自行分配一块缓冲区，在将语音数据写入到缓冲区后，向驱动程序提交一个播放任务。驱动程序完成该缓冲区中的语音数据的播放后，向应用程序抛出事件，播放任务结束。

播放单个内存的相关函数及输出事件如下表所示：

函数或事件名称	描述
SsmPlayMem	启动单个内存播放任务
SsmSetStopPlayOffset	重新设置缓冲区中停止播放的位置
SsmStopPlayMem	终止单个内存播放任务
SsmGetPlayOffset	获取单个内存播放任务时，驱动程序内部的放音指针
E_PROC_PlayMem	输出单个内存播放任务的进展情况

1.10.1.7 播放内存序列

内存序列播放模式是指由应用程序以内存缓冲区的方式预先提交一组语音块，然后由驱动程序按照设定顺序对

其进行连续播放的操作，语音块之间为无缝连接。语音块由应用程序自行管理。

播放内存序列的相关函数包括：

函数名称	描述
SsmAddToPlayMemList	向驱动程序提交一个内存缓冲区
SsmClearPlayMemList	清除已经提交给驱动程序的全部缓冲区
SsmPlayMemList	启动内存序列播放任务
SsmStopPlayMemList	终止内存序列播放任务

1.10.1.8 播放乒乓内存

乒乓内存播放模式采用双缓冲区机制。乒乓内存播放模式的工作机制为：在启动放音前，应用程序首先准备好一个（或两个）放音缓冲区并填充好放音数据，然后调用一次（或连续两次）[SsmPlayMemBlock](#) 函数将缓冲区提交给驱动程序。驱动程序收到播放命令后，启动乒乓内存播放任务。当一个缓冲区内的语音数据播放完毕后，驱动程序自动播放另一个缓冲区，同时调用应用程序设置的回调函数，将完成播放的缓冲区信息传递给应用程序。应用程序在回调函数中将语音数据填充到下一个缓冲区，然后 [SsmPlayMemBlock](#) 函数再次将缓冲区提交给驱动程序。

与单个内存播放模式相比，乒乓内存播放模式具有下列特点：

- 可以将缓冲区最小设置为 192 字节。
- 支持回调的工作方式。

由于 Windows 和 Linux 都不是真正的实时操作系统，将语音数据从应用程序的内存缓冲区传输到板卡的语音解码器，会产生一定的时间延迟。另外，由于语音数据需要先从用户态（应用程序）传输到内核态（设备驱动程序），再从内核态输出到板卡，所以这个时间延迟还不是恒定值，会随着应用系统的繁忙程度而变化。要减小传输延迟时间，比较有效的办法就是减小缓冲区的大小，同时提高应用程序更新语音数据的代码的优先级。驱动程序的线程优先级是非常高的，回调函数中的代码也就同时获得了与驱动程序相同的优先级。由此可见，乒乓内存播放模式可以将语音数据从应用程序的内存缓冲区转移到板卡的语音解码器所产生的时间延迟做到最小，适用于对放音操作要求实时性很高的应用场合。

播放乒乓内存的相关函数包括：

函数名称	描述
SsmPlayMemBlock	向驱动程序提交一个缓冲区，并启动乒乓内存播放任务
SsmStopPlayMemBlock	终止乒乓内存播放任务

1.10.1.9 终止播放任务的通用函数

[SsmStopPlay](#) 函数可以终止任何类型的播放任务。

1.10.1.10 获取播放任务的相关信息

下列函数可以用来获取播放任务的相关信息：

函数名称	描述
SsmQueryPlayFormat	查询通道是否支持指定的放音格式
SsmGetPlayType	查询通道上正在执行的放音任务类型
SsmCheckPlay	获取放音任务的完成情况
SsmGetPlayedPercentage	获取已经完成播放的数据长度占总长度的百分比
SsmGetPlayedTime	获取本次放音任务持续的时间

1.10.2 录音操作

1.10.2.1 设置录音信号源

录音操作的信号源与板卡的型号有关，详细内容请参见相应板卡系列的原理框图部分。

1.10.2.2 设置录音任务的终止条件

录音任务启动后，除了录音任务正常终止以及被应用程序调用停止录音的函数所终止外，如果通道上检测到了符合特定条件的事件，驱动程序也会自动终止录音任务，从而简化应用程序的编写。这些条件包括：

- (1) [DTMF 检测器](#)在来话信号中检测到 DTMF 字符

相关函数及配置项包括：

函数名称	功能简述	相关配置项
SsmSetDTMFStopRecCharSet	设置录音任务是否因 DTMF 检测器 检测到 DTMF 字符而终止	DtmfStopRecCharSet

- (2) 通道的呼叫状态机检测到对端挂机

相关函数及配置项包括：

函数名称	功能简述	相关配置项
SsmSetHangupStopRecFlag	设置录音任务是否因驱动程序状态机检测到对端挂机而终止	HangupStopRec

1.10.2.3 设置预录音功能

预录音是指由驱动程序按照预设参数启动一个预录音任务，将录音数据缓存到驱动程序内部的录音缓冲区中。当应用程序调用文件录音函数时，可以将缓冲区中的一段录音数据先保存到文件中，再开始正常的文件录音任务。预录音功能可以防止因应用程序响应迟缓等原因而导致前面语音数据的丢失。

预录音缓冲区的大小可以通过配置项 [RecordBufSize](#) 设置。

与预录音操作相关的函数或配置项包括：

函数名称	功能简述	相关配置项
SsmSetPrerecord	设置预录音功能。	PrerecordEnable PrerecordMode PrerecordInsertTime PrerecordCodec
SsmGetPrerecordState	获取预录音功能的设置	

1.10.2.4 设置录音文件截尾功能

录音文件截尾功能是指文件录音任务终止时，驱动程序自动将文件尾部的一段语音数据抛弃。此项功能适用于一些特殊的应用场合，例如，文件录音任务因为驱动程序检测到 DTMF 信号终止，但应用程序不希望该 DTMF 字符出现在录音文件中。

录音文件截尾功能只支持文件录音函数，较多与”检测到 DTMF 时自动终止文件录音任务”的功能配合使用。

与此项功能相关的函数或配置项包括：

函数名称	功能简述	相关配置项
SsmSetTruncateTail	设置录音文件截尾功能的工作参数	TruncateTailOnRecordToFile
SsmGetTruncateTailTime	获取预设的录音文件截尾功能的工作参数	

1.10.2.5 设置录音 AGC

自动增益控制（Automatic Gain Control,AGC）模块。如果开启 AGC 功能，驱动程序会根据输入信号的幅度对信号自动进行调节，对小信号进行提升，对大信号进行衰减。

函数 [SsmSetRecAGC](#) 用于开启或关闭 AGC 功能。AGC 模块的工作参数可以通过下列配置项进行设置：

- ◊ [AGCMAXGAIN](#)
- ◊ [AGCMINGAIN](#)
- ◊ [AGCMAXLEVEL](#)
- ◊ [AGCMINLEVEL](#)
- ◊ [AGCDOWNRATIO](#)
- ◊ [AGCUPRATIO](#)
- ◊ [AGCKEPTIME](#)

对于 DST-24B/PCI, DST-24B/PCI+, DST-24B/PCIe, DST-24B/PCIe+卡，AGC 模块的工作参数可以通过下列配置项对上下行两个方向分别进行设置：

- ◆ [AGCMAXGAINUPLINK](#)
- ◆ [AGCMINGAINUPLINK](#)
- ◆ [AGCMAXLEVELUPLINK](#)
- ◆ [AGCMINLEVELUPLINK](#)
- ◆ [AGCDOWNRATIOUPLINK](#)
- ◆ [AGCUPRATIOUPLINK](#)
- ◆ [AGCMAXGAINDOWNLINK](#)
- ◆ [AGCMINGAINDOWNLINK](#)
- ◆ [AGCMAXLEVELDOWNLINK](#)
- ◆ [AGCMINLEVELDOWNLINK](#)
- ◆ [AGCDOWNRATIODOWNLINK](#)
- ◆ [AGCUPRATIODOWNLINK](#)

注意：

- 三汇语音卡的 AGC 的工作原理是根据原始输入信号的电平，自动调节 AGC 电路中的放大器增益系数，从而将输出信号的电平尽可能控制在预定的范围内。针对不同的应用环境和不同的用户偏好，可以通过设定 AGC 参数实现不同的 AGC 效果。三汇语音卡 AGC 的参数有 3 类：第一类，[AGCMAXGAIN](#) ([AGCMAXGAINUPLINK](#)、[AGCMAXGAINDOWNLINK](#) 和 [AGCMINGAIN](#) ([AGCMINGAINUPLINK](#)、[AGCMINGAINDOWNLINK](#)) 用于控制 DSP 增益系数的变化范围；第二类，[AGCMAXLEVEL](#) ([AGCMAXLEVELUPLINK](#)、[AGCMAXLEVELDOWNLINK](#)) 和 [AGCMINLEVEL](#) ([AGCMINLEVELUPLINK](#)、[AGCMINLEVELDOWNLINK](#)) 用于设定期望的输出电平；第 3 类，[AGCDOWNRATIO](#) ([AGCDOWNRATIOUPLINK](#)、[AGCDOWNRATIODOWNLINK](#))、[AGCUPRATIO](#) ([AGCUPRATIOUPLINK](#)、[AGCUPRATIODOWNLINK](#)) 和 [AGCKEETIME](#) 用于控制 DSP 增益系数的变化速度；
- AGC 的缺省配置已经能够适应绝大多数的应用场景。如果确实需要调整 AGC 的工作参数，请在本公司的技术人员指导下进行；
- 当通道迁移到 S_CALL_STANDBY 状态时，驱动程序会自动关闭 AGC 功能，应用程序仍然可以在通道上进行录音操作，但由于此状态下 AGC 功能已关闭，因此，录音数据不再具有 AGC 功能；若应用程序希望在这种情况下恢复录音数据的 AGC 功能，需要对配置项 [OpenRecEnAndPlayEnOnIdle](#) 进行设置。

1.10.2.6 获取录音操作的信息

相关函数包括：

函数名称	功能简述
<u>SsmQueryRecFormat</u>	查询通道是否支持指定的录音格式
<u>SsmGetRecType</u>	获取正在执行的录音任务类型
<u>SsmGetRecTime</u>	获取录音任务持续的时间长度
<u>SsmCheckRecord</u>	查询录音任务的完成情况

1.10.2.7 录音函数概述

驱动程序在内部会为每一个通道分配一块录音缓冲区，缓冲区的大小可以通过配置项 [RecordBufSize](#) 进行设置，缺省值为 32768 字节。

录音函数包括文件录音、内存录音和乒乓缓冲区录音 3 类。

1.10.2.7.1 文件录音

相关函数包括：

函数名称	功能简述
SsmRecToFile	
SsmRecToFileA	启动文件录音任务
SsmRecToFileEx	
SsmStopRecToFile	终止文件录音任务
SsmPauseRecToFile	暂停文件录音任务
SsmRestartRecToFile	恢复文件录音任务
SsmChkRecToFile	查询文件录音任务的完成情况
SsmGetDataBytesToRecord	获取尚未完成的录音数据字节数

1.10.2.7.2 内存录音

内存录音是指应用程序调用 [SsmRecToMem](#) 向驱动程序提交一个缓冲区，驱动程序将录音数据写入该缓冲区，并按照设定的条件抛出进展消息。

相关函数包括：

函数名称	功能简述
SsmRecToMem	启动内存录音任务
SsmStopRecToMem	终止内存录音任务
SsmGetRecOffset	获取内存录音任务执行过程中，驱动程序内部的录音指针

1.10.2.7.3 乒乓缓冲区录音

乒乓缓冲区录音采用双缓冲区机制，它的工作原理为：在启动录音前，应用程序首先准备好一个（或两个）录音缓冲区，然后调用一次（或连续两次）[SsmRecordMemBlock](#) 函数将缓冲区提交给驱动程序。应用程序首次调用[SsmRecordMemBlock](#) 函数后，驱动程序会启动乒乓缓冲区录音任务，开始将录音数据写入到缓冲区。当一个缓冲区内写满后，驱动程序自动切换到另一个缓冲区，同时调用应用程序设置的回调函数，将完成录制的缓冲区信息传递给应用程序。在回调函数中，应用程序可以调用[SsmRecordMemBlock](#) 函数提交新的缓冲区，并对录音数据进行后续处理。

与内存录音相比，乒乓缓冲区录音具有下列特点：

- 可以将缓冲区最小设置为 192 字节。
- 支持回调的工作方式。

由于 Windows 和 Linux 都不是真正的实时操作系统，将录音数据从板卡传输到应用程序的内存缓冲区，会产生一定的时间延迟。另外，由于语音数据需要从内核态（设备驱动程序）传输到用户态（应用程序），涉及到从内核态到用户态的切换，因此这个时间延迟不是恒定值，会随着应用系统的繁忙程度而变化。要减小传输延迟时间，比较有效的方法就是减小缓冲区的大小，同时提高应用程序处理语音数据的代码的优先级。驱动程序的线程优先级非常高，回调函数中的代码具有与驱动程序相同的优先级。由此可见，乒乓缓冲区录音模式可以将语音数据的传输延迟减小到最低限度，适用于要求实时性很高的应用场合。

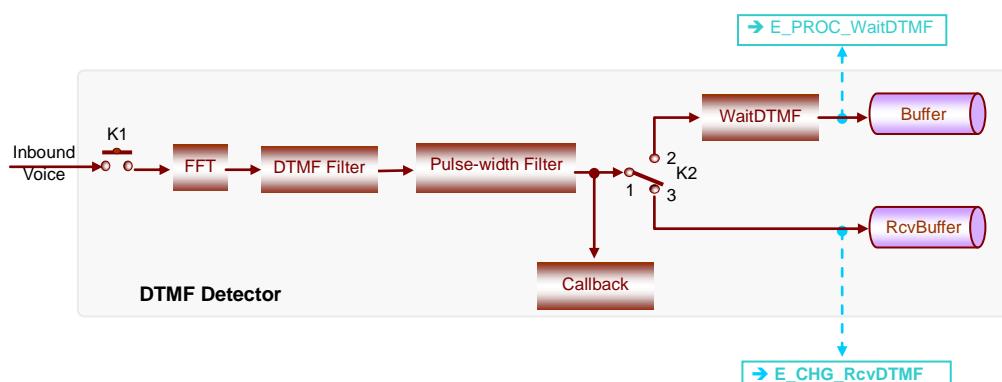
乒乓缓冲区录音的相关函数包括：

函数名称	功能简述
SsmRecordMemBlock	向驱动程序提交一个缓冲区，并启动乒乓内存录音任务
SsmStopRecordMemBlock	终止乒乓内存录音任务

1.10.3 DTMF 检测器(DTMF Detector)

三汇所有系列的板卡都为每个通道配置了独立的 DTMF 检测器。

DTMF 检测器的内部工作原理框图如下图所示：



开关 K1 由配置项 [AlwaysEnableRxDtmf](#)、驱动程序和应用程序共同控制。当 [AlwaysEnableRxDtmf](#) 设置为 1

时, K1 总是处于闭合状态; 如果设置为 0, 驱动程序会根据通道的状态迁移来决定 K1 状态, 当通道进入通话状态时闭合 K1, 离开通话状态并完成拆线后断开 K1。应用程序可以调用 [SsmEnableRxDtmf](#) 来控制 K1 的状态。

对于坐席通道, 当通道处于“空闲”状态时, 在驱动程序的自动控制下, DTMF 检测器处于关闭状态, 但可以通过配置项 [RcvDtmfOnIdle](#) 开启。

FFT 模块对输入语音信号进行 FFT 变换, 将语音信号从时域变换到频域。

DTMF 滤波器 (DTMF Filter) 检查输入信号的频率特征中是否包含 DTMF 字符的信号, 并以电平方式输出判断结果。

脉宽滤波器 (Pulse-width Filter) 用于对 DTMF 信号的脉冲宽度进行过滤, 以消除语音信号中夹杂的 DTMF 信号对检测结果的影响。检测到的 DTMF 字符在送给 Callback 模块的同时, 经由 K2 开关输出。

开关 K2 由驱动程序自动控制, 用来控制 DTMF 的输出路径。K2 通常处于 1-3 连接状态, 检测到的 DTMF 字符都送到 RcvBuffer 缓冲区中进行保存, 同时向应用程序抛出 [E_CHG_RcvDTMF](#) 事件。

RcvBuffer 缓冲区存放收到的 DTMF 字符, 它的大小可通过配置项 [RxDtmfBufSize](#) 进行指定。访问 RcvBuffer 缓冲区的函数包括:

函数名称	功能简述
SsmClearRxDtmfBuf	清空驱动程序内部的 DTMF 按键号码接收缓冲区
SsmGetDtmfStr SsmGetDtmfStrA	获取保存在 DTMF 检测器缓冲区的 DTMF 字符
SsmGetRxDtmfLen	获取 DTMF 检测器缓冲区中保存的 DTMF 字符个数
SsmGet1stDtmf SsmGet1stDtmfClr	获取 DTMF 检测器缓冲区中最先收到的 DTMF 字符
SsmGetLastDtmf	获取 DTMF 检测器缓冲区中最后收到的 DTMF 字符

当应用程序调用 [SsmSetWaitDtmf](#) 函数启动一个 WaitDtmf 任务后, K2 自动切换到 1-2 连接状态, 检测到的 DTMF 字符送到了 WaitDtmf 模块进行后续处理。注意: 一旦启动了 WaitDtmf 任务以后, 检测到的 DTMF 字符不再进入 RcvBuffer 中, 但是, RcvBuffer 中已有的 DTMF 字符会被复制到 Buffer 中。

1.10.3.1 DTMF 滤波器 (DTMF Filter)

DTMF 滤波器检查输入信号的频率特征中是否包含 DTMF 字符, 并防止将语音信号误判为 DTMF 信号。

DTMF 字符与频率的对应关系如下表所示:

	1209Hz	1336Hz	1477Hz	1633Hz
697Hz	1	2	3	A
770Hz	4	5	6	B

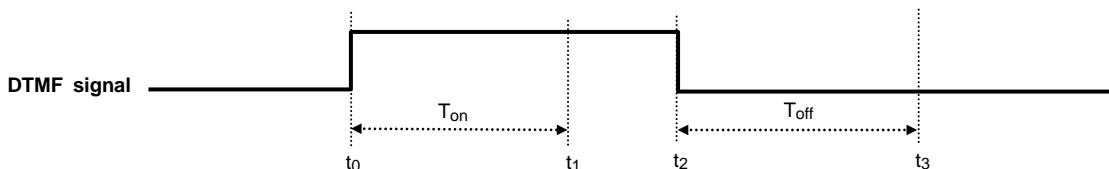
852Hz	7	8	9	C
941Hz	*	0	#	D

下列配置项或函数用于强化 DTMF 滤波器的判决条件：

配置项或函数	描述
配置项： DualAndAllFreqEnScale 函数： SsmSetFlag （携带 F_DualAndAllFreqEnScale 参数）	设置 DTMF 信号的带内能量占全频能量的百分比判定门限。DTMF 滤波器实时计算输入信号中 DTMF 的带内能量占全频能量的百分比，然后与某个设定的判定门限进行比较。如果大于设定门限，则判定线路上的信号为 DTMF 信号，并将判定结果以电平方式输出（高电平表示检测到信号音，低电平表示未检测到信号音）。
DtmfEnergy	设置 DTMF 信号的带内能量的绝对值。只有当 DTMF 信号的带内能量大于本配置项的设定值时，驱动程序才认为是真正的 DTMF 信号。
配置项： HighAndLowFreqEnScale 函数： SsmSetFlag （携带 F_HighAndLowFreqEnScale 参数）	设置 DTMF 信号中，高频分量部分能量与低频分量部分能量之间的比例。只有当 DTMF 信号的高频分量部分能量与低频分量部分能量之间的比例大于本配置项的设定值时，驱动程序才认为是真正的 DTMF 信号。

1.10.3.2 DTMF 脉宽滤波器

DTMF 脉宽滤波器的工作原理如下图所示：



上图中， T_{on} 和 T_{off} 分别表示 DTMF 信号的高电平最小保持时间和低电平最小保持时间。当实际的输入 DTMF 信号电平从低电平转换到高电平后，只有当高电平保持时间和低电平保持时间分别大于 T_{on} 和 T_{off} 后，驱动程序才会判定线路上出现了真正的 DTMF 字符，并将此 DTMF 字符写入缓冲区，同时向应用程序抛出 [E_CHG_RcvDTMF](#) 事件。

T_{on} 和 T_{off} 由配置项 [ReceiveDtmfSensitive](#) 或函数 [SsmSetFlag](#)（携带 F_RCVDTMFSENS 参数）进行设置。

配置项 [OmitABCD](#) 用来决定是否丢弃收到的 a、b、c 或 d 字符。如果设置为保留，则收到的 a、b、c 或 d 字符将与其它字符一样进入后续处理模块，否则被丢弃。

配置项 [DtrmOnDtmfHighLevel](#) 用来决定驱动程序输出 DTMF 字符的时机。如果设置为高电平输出，当 DTMF 信号的高电平持续时间大于 T_{on} 时（即上图中的 t_1 时刻）驱动程序就立即将检测到的 DTMF 字符输出；如果设置为低电平输出，则当高电平转换到低电平，并且低电平持续时间大于 T_{off} 时（即上图中的 t_3 时刻），驱动程序才将检测到的 DTMF 字符输出。

1.10.3.3 WaitDtmf 任务

WaitDtmf 任务用于向驱动程序提交一个接收特定 DTMF 字符串的任务。应用程序调用 [SsmSetWaitDtmf](#) 函数后，就启动了一个 WaitDtmf 任务。WaitDtmf 任务启动后，DTMF 检测器输出的 DTMF 字符不再进入 RcvBuffer，而是直接进入了 WaitDtmf 模块的缓冲区，并根据预设条件判断是否应该终止本次任务。如果满足停止接收的条件，驱动程序向应用程序抛出 [E PROC WaitDTMF](#) 事件，本次 WaitDtmf 任务终止。

有关 WaitDtmf 的函数包括：

函数/事件名称	功能简述
SsmSetWaitDtmf SsmSetWaitDtmfEx SsmSetWaitDtmfExA	启动 WaitDtmf 任务
SsmCancelWaitDtmf	取消 WaitDtmf 任务
SsmChkWaitDtmf	查询 WaitDtmf 任务的进展情况
E PROC WaitDTMF	通知 WaitDtmf 任务的进展情况

1.10.3.4 Callback 方式编程

Callback 模块处理应用程序的回调函数。当驱动程序检测到 DTMF 字符后，如果应用程序已通过调用函数 [SsmSetRxDtmfHandler](#) 设置了一个回调函数，会调用此回调函数。

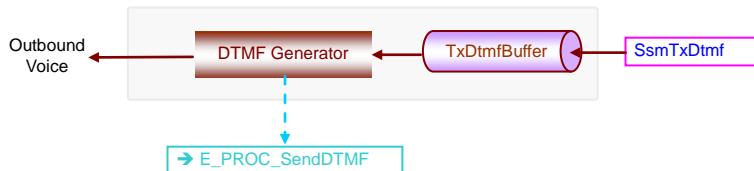
1.10.3.5 检测脉冲方式的按键信号

对于模拟中继线录音通道，DTMF 检测器还支持脉冲方式的按键信息，但需要对配置项 [EnablePulseKeyDetect](#) 进行设置。

1.10.4 DTMF 发生器(DTMF Generator) (CTI 系列)

DTMF 发生器可以在线路上产生标准 DTMF 字符，包括：“0”、“1”、“2”、“3”、“4”、“5”、“6”、“7”、“8”、“9”、“*”、“#”、“A”、“B”、“C”、“D”。

DTMF 发生器的原理框图如下图所示：

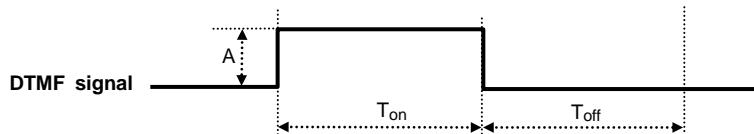


三汇板卡为每个通道都配置了独立的 DTMF 发生器 (DTMF Generator) 和发送缓冲区 (TxDtmfBuffer)。

TxDtmfBuffer 的大小可以在 [TxDtmfBufSize](#) 配置项中指定，缺省值为 50 个字符。

DTMF 发生器可以在线路上产生出频率、幅度完全符国家关于电话机 DTMF 拨号盘发号标准的 DTMF 信号。

DTMF 发生器产生的典型 DTMF 信号如下图所示：



一个 DTMF 字符的波形特征用 T_{on} 和 T_{off} 这两个参数来表示, T_{on} 表示高电平持续时间, T_{off} 表示低电平持续时间, 可以通过配置项 [TxDtmfTimePara](#) 或函数 [SsmSetTxDtmfPara](#) 进行设置。函数 [SsmGetTxDtmfPara](#) 用来获取 T_{on} 和 T_{off} 这两个参数的设置。

DTMF 信号的幅度 A 可通过函数 [SsmSetFlag](#) (携带 F_TXDTMFAMP 参数) 设置, 默认值: SHT、ATP 系列为 0X8976; SHD、DTP 系列为 0X3f3f, 其中高 8 位设置低频率的幅度, 低 8 位设置高频率的幅度。幅度 A 的值与 DB 值的转换关系为: $Y=20\cdot\lg(X/16)$, 其中 Y 为 DB 值, X 为设定值。以默认值 0X8976 为例, 通过上述转换公式可得, 低频区的 DB 值为: $18.65=20\cdot\lg(137/16)$ 。高频 DB 值为 $17.35=20\cdot\lg(118/16)$ 。

应用程序调用 [SsmTxDtmf](#) 后, 驱动程序会将待发送的 DTMF 字符存放在 TxDtmfBuffer 中, 同时, 启动一个发送 DTMF 的任务。每当 DTMF 发生器完成一个字符的发送, 会自动从 TxDtmfBuffer 中取出下一个字符进行发送。如果 TxDtmfBuffer 为空, 表明任务终止, 驱动程序就向应用程序抛出 [E PROC SendDTMF](#) 事件。

函数 [SsmStopTxDtmf](#) 可以用来随时终止发送任务, [SsmChkTxDtmf](#) 函数可以用来查询任务的进展情况。

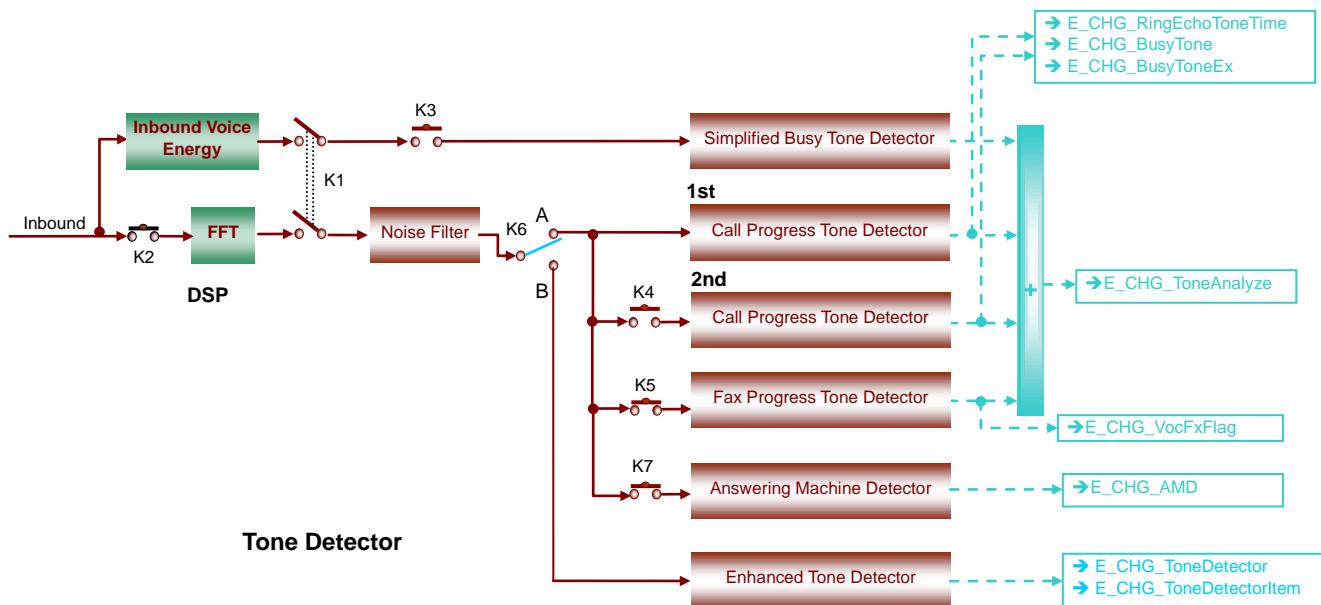
1.10.5 信号音检测器(Tone Detector)

每个三汇语音通道均内置了一个独立的信号音检测器。信号音检测器的主要作用:

- 模拟线路的呼叫进程音检测。
- 传真机信号音的检测。
- 其它特殊应用, 比如在自动呼叫测试系统中用于数字中继通道上的信号音检测等。

Tip: 如果您的应用系统不包括上述类型的应用, 可以不必阅读本小节。如果您的系统中包含模拟中继通道或模拟中继线录音通道, 请仔细阅读本小节, 本小节描述的内容对您应用系统的兼容性很重要。

信号音检测器的内部功能框图及相关事件输出如下图所示:



开关 K1 控制信号音检测器的工作状态。K1 可以通过函数 [SsmStartToneAnalyze/ SsmCloseToneAnalyze](#) 来设置。注意：K1 开关同时控制了远端摘机检测器。驱动程序会根据通道的状态迁移情况，自动控制 K1，系统初始化后，K1 处于关闭状态，详细内容请参见通道的状态迁移部分。

K2 由 DSP 程序自动控制，如果开启通道的 FSK 接收器，K2 断开，否则闭合。

K6 用于切换是否使用增强信号音检测器，可由配置项 [ToneDetectorMode](#) 来设置。K6 默认接通 A，采用第一进程音检测器、第二进程音检测器和传真进程音检测器进行检测；当 K6 接通 B 时，采用增强信号音检测器进行检测。

整个信号音检测器由下列模块组成：

- **FFT 模块**

FFT 模块把输入语音信号从时域变换到频域，变换结果送到后续模块进行信号音分析和计算。FFT 模块同时计算线路上语音信号的全频能量。FFT 的计算都在 DSP 芯片里完成，不占用主机资源。

- **噪声滤波器(Noise Filter)**

用于过滤线路上的噪声。

- **呼叫进程音检测器（Call Progress Tone Detector）**

用于检测线路上是否存在呼叫进程音。信号音的频率可以是单音频，也可以是双音频。每个信号音检测器包含了 2 个相互独立的呼叫进程音检测器。开关 K4 用来控制是否启用第 2 呼叫进程音检测器，可以通过配置项 [Enable2ndToneAnalyzer](#) 或函数 [SsmStart2ndToneAnalyzer](#) 来设置，缺省设置为断开（关闭）。

- **传真进程音检测器（Fax Progress Tone Detector）**

用于检测线路上是否存在单音频的传真进程音。每个信号音检测器包含了 2 个相互独立的传真进程音检测器，但当开关 K4 连接到第 2 呼叫进程音检测器时，传真进程音检测器不可用。

- **简易忙音检测器(Simplified Busy Tone Detector)**

对于模拟中继线通道，当应用程序启动了 FSK 接收器后，由于板卡上的 DSP 程序会自动断开 K2 开关，信号音检测器的输入被断开，无法正常工作。在某些使用 SHT 系列板卡的特殊应用场合，应用程序需要在接收 FSK 数据时仍然能够检测忙音，以及时判断对端的挂机动作。简易忙音检测器就是为了满足这种需要而提供的功能。当 K2 被 DSP 自动断开时，应用程序通过闭合 K1 和 K3，就可以使用简易忙音检测器。简易忙音检测器的输入只有 Barge-in 检测器输出的来话信号能量，因此无法判定忙音信号的频率特征，只能通过检测忙音的波形特征来判断线路上是否出现了忙音，因此称作简易忙音检测器。

- **增强信号音检测器（Enhanced Tone Detector）**

用于替代呼叫进程音检测器和传真进程音检测器的信号音检测功能。该信号音检测器除了可以检测到呼叫进程音检测器和传真进程音检测器检测的信号音外，还能够检测特殊信号音（比如 SIT 信号音）。

注意：目前只有部分板卡支持增强信号音检测器，详细见下表：

板卡型号	是否支持增强信号音检测器
SHT-8A/PCI	支持

SHT-8B/PCI	支持
SHT-8B/PCI/FAX	支持
SHT-8C/PCI/FAX	支持
SHT-8C/PCI/EC	支持
SHT-16A-CT/PCI	支持
SHT-16B-CT/PCI	支持
SHT-16B-CT/PCI/FAX	支持
SHT-16B-CT/cPCI	支持
SHT-16B-CT/cPCI/MP3	支持
SHT-16B-CT/cPCI/FAX	支持
SHT-16C-CT/PCI/FAX	支持
SHT-16C-CT/PCI/EC	支持
SHT-16D-CT/PCIe	支持
SHT-4A/USB	支持
SHT-2A/USB	支持
SHT-2B/USB	支持
SHT-4B/USB	支持
SHF-2D/PCI	支持
SHF-4D/PCI	支持
SHF-4D/PCIe	支持

- 机器应答检测器（Answering Machine Detector）

对端摘机后，机器应答检测器用于区分应答方是机器还是人。开关 K7 用来控制是否启用自动应答检测器，可以通过配置项 [EnableAMD](#) 来设置，缺省设置为闭合（开启）。

同时下列配置项也会影响机器应答检测器的行为，详细如下：

[AMDNoSoundAfterDialTime](#): 用于判断拨号后线路上无声的时间是否超时。当无声的时间超过 AMDNoSoundAfterDialTime 的设定值时，会抛出 [E_CHG_AMD](#) 事件（[E_CHG_AMD](#) 事件的参数值为 5）。

[AMDNoSoundTime](#): 用于判断检测到信号音或者彩铃后无声的时间是否超时。当无声的时间超过 AMDNoSoundTime 的设定值时，会抛出 [E_CHG_AMD](#) 事件（[E_CHG_AMD](#) 事件的参数值为 4）。

[AMDTIMEOUT](#): 用于判断整个 AMD 检测过程是否超时。当整个 AMD 检测过程的时间超过 AMDTIMEOUT 的设定值时，会抛出 [E_CHG_AMD](#) 事件（[E_CHG_AMD](#) 事件的参数值为 3）。

[AMDToneCount](#): 用于判断检测到信号音的时间是否超时。当整个检测到信号音的时间超过 AMDToneCount 的设定值时，会抛出 [E_CHG_AMD](#) 事件（[E_CHG_AMD](#) 事件的参数值为 1）。

[AMDTOn](#): 用于设置声音进入高电平的最低持续时间。

AMDOff: 用于设置声音进入低电平的最低持续时间。

AMDTimeA: 用于设置真人摘机的问候语之前的最低静音持续时间。

AMDTimeB: 用于设置真人摘机的问候语的最低持续时间。

AMDTimeC: 用于设置真人摘机的问候语的最大持续时间。

AMDTimeD: 用于设置真人摘机的问候语之后的最低静音持续时间。

AMDSilentEnergy: 用于设置声音能量是否为静音的阀值。

1.10.5.1 FFT 模块

板载 DSP 芯片中的 FFT 模块对输入语音信号进行 FFT 变换，将语音信号从时域变换到频域，计算结果作为后续模块的输入。FFT 每隔 16 毫秒对线路上的语音信号进行一次 FFT 计算。

FFT 完成后，驱动程序会向应用程序抛出下列事件：

- E_OverallEnergy: 将 FFT 计算得到的全频能量传递给应用程序。多数情况下应用程序都不必响应此事件。
- E_CHG_PeakFq: 将计算出的频谱中处于峰值位置的信号频率值及其带内能量传递给应用程序。函数 SsmSetPeakFqDetectBW 用来设置峰值频率的带宽。多数情况下应用程序都不必响应此事件。注意：只有当峰值频率值发生变化时才会抛出本事件。

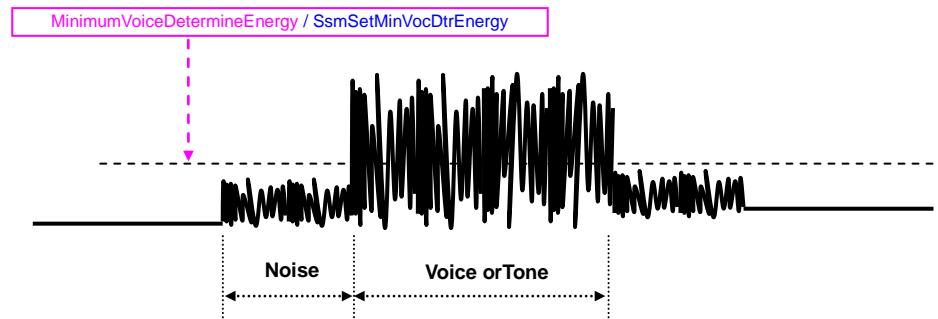
注意：

- 全频能量的输出所采用的量纲不是分贝（DB），这是为了降低计算量。如果应用程序需要使用 DB 作为量纲，则全频能量输出结果与 DB 的转换关系为（假设 X 为通过 API 函数获得的能量值）：
SHT 系列板卡: $DB=10*\lg(X/72,000,000)$
SHD 系列板卡: $DB=10*\lg(X/33,000,000)$
SHN 系列板卡: $DB=10*\lg(X/33,000,000)$
ATP 系列板卡: $DB=10*\lg(X/72,000,000)$
DTP 系列板卡: $DB=10*\lg(X/33,000,000)$
DST 系列板卡: $DB=10*\lg(X/33,000,000)$
24DA 板卡: $DB=10*\lg(X/22,000,000,000)$
- 如果启动了 FSK 接收器，FFT 模停止工作。

1.10.5.2 噪声滤波器(Noise Filter)

在大多数情况下，噪声滤波器使用缺省配置即可，无须配置。

噪声滤波器模块的工作原理如下图所示：



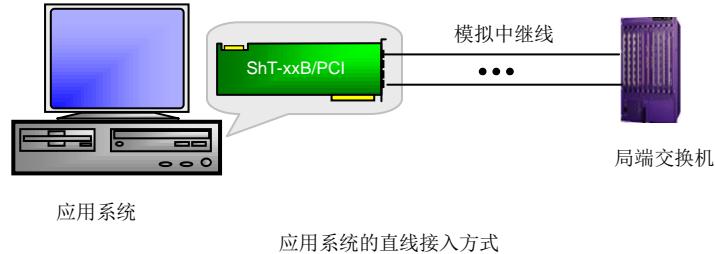
当全频能量小于配置项 MinimumVoiceDetermineEnergy （或 SsmSetMinVocDtrEnergy 函数）指定的噪声门

限，驱动程序会判定为噪声；否则判定为真实的语音信号。

1.10.5.3 呼叫进程音检测器(Call Progress Tone Detector) (SHT 系列 only)

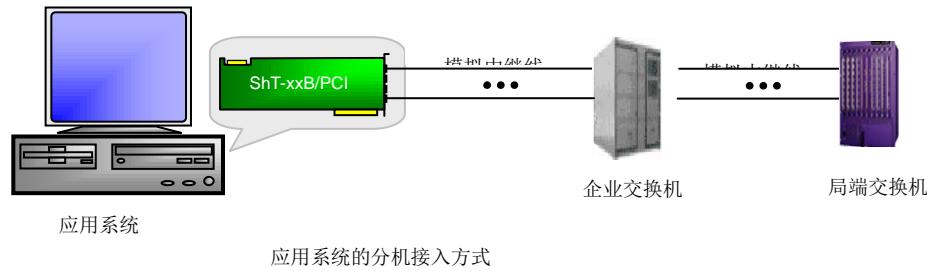
对于模拟中继线通道，由于进程音检测器的分析结果直接影响到呼叫进程，因此，准确地使用和设置呼叫进程音显得非常重要。从应用系统的接入方式来看，通常有两种：直线接入方式和分机接入方式。

直线接入方式的应用系统结构如下图所示：



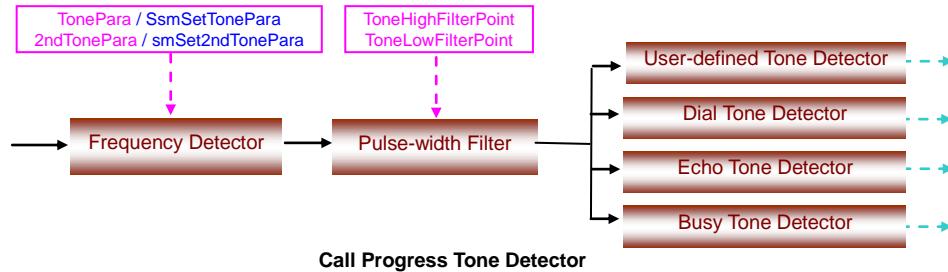
此类应用系统只需要检测一种类型的呼叫进程音。

分机方式接入的应用系统结构如下图所示：



由于企业交换机、局端交换机的信号音可能不相同，因此，需要驱动程序能够同时检测 2 种不同的呼叫进程音。

呼叫进程音检测器由第 1 呼叫进程音检测器和第 2 呼叫进程音检测器组成，其工作原理框图如下图所示：



呼叫进程音检测器由频率检测器 (Frequency Detector)、脉宽滤波器 (Pulse-width Filter)、拨号音检测器 (Dial Tone Detector)、回铃音检测器 (Echo Tone Detector)、忙音检测器 (Busy Tone Detector) 和自定义信号音检测器 (User-defined Tone Detector) 组成。

自定义信号音检测器只用于某些特殊的场合。

拨号音检测器、回铃音检测器、忙音检测器和自定义信号音检测器可以同时工作。

- 频率检测器

频率检测器检查输入信号的频率特征，判断是否包含指定频率的信号音。频率检测器实时计算输入信号中指定的信号音的带内能量占全频能量的百分比，然后与某个设定的判定门限进行比较。如果大于设定门限，则判定线路上的信号为信号音，并将判定结果以电平方式输出（高电平表示检测到信号音，低电平表示未检测到信号音）。

配置项 [TonePara](#) 或函数 [SsmSetTonePara](#) 用于设置第 1 呼叫进程音检测器的频率检测器的 2 个频率参数，包括中心频率和带宽，以及带内能量百分比判定门限。

配置项 [2ndTonePara](#) 或函数 [SsmSet2ndTonePara](#) 用于设置第 2 呼叫进程音检测器的频率检测器的频率参数，包括中心频率和带宽，以及带内能量百分比判定门限。

- **脉宽滤波器**

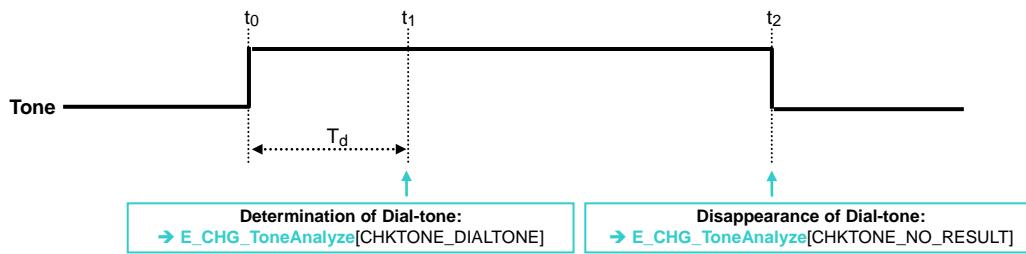
脉宽滤波器用于对信号音的脉冲宽度进行过滤，以消除语音信号中夹杂的信号音对检测结果的影响。当信号音电平从低电平向高电平跳变时，只有当高电平保持时间高于配置项 [ToneHighFilterPoint](#) 指定的时间，驱动程序才确认信号音的开始；当信号音电平从高电平向低电平跳变，只有当低电平保持时间高于配置项 [ToneLowFilterPoint](#) 指定的时间时，驱动程序才确认信号音的消失。

注意：

- (1) 脉宽滤波器的配置参数同时对 1 呼叫进程音检测器、第 2 呼叫进程音检测器以及传真进程音检测器起作用；
- (2) 对于模拟中继线类型的通道，第 1 呼叫进程音检测器中的拨号音检测器、回铃音检测器、忙音检测器的检测结果会自动作为状态机的输入事件；而对于第 2 呼叫进程音检测器，拨号音检测器、回铃音检测器、忙音检测器的检测结果是否作为状态机的输入事件，取决于配置项 [Check2ndToneOnAutoDial](#) 的设置。

1.10.5.3.1 拨号音检测器 (Dial Tone Detector)

拨号音的检测由拨号音检测器(Dial Tone Detector)完成。典型的拨号音波形如下图所示：

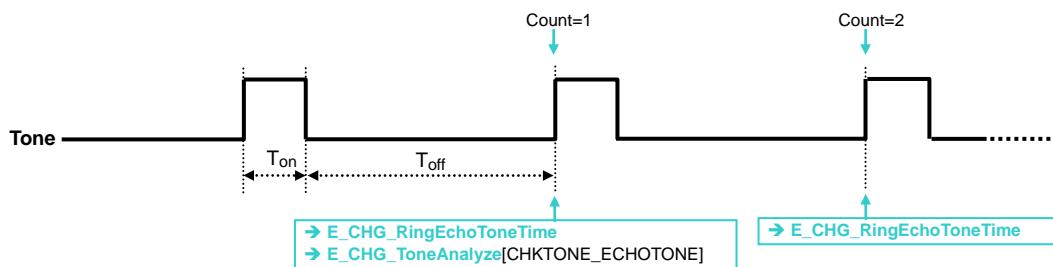


驱动程序对信号音的高电平进行计时，当计时到达拨号音判别门限 T_d 后，信号音检测器就会抛出 [E_CHG_ToneAnalyze](#) 事件（带 `CHKTONE_DIALTONE` 参数）。当拨号音消失后，拨号音检测器会抛出 [E_CHG_ToneAnalyze](#) 事件（带 `CHKTONE_NO_RESULT` 参数）。

对于第 1 呼叫进程音检测器的拨号音检测器， T_d 由配置项 [IsDialToneDetermineTime](#) 或函数 [SsmSetIsDialToneDtrTime](#) 进行指定；对于第 2 呼叫进程音检测器的拨号音检测器(如果开启)， T_d 由配置项 [2ndIsDialToneDetermineTime](#) 或函数 [SsmSet2ndIsDialToneDtrTime](#) 进行指定。

1.10.5.3.2 回铃音检测器

回铃音的检测由回铃音检测器(Echo Tone Detector)完成。典型的回铃音波形如下图所示：



回铃音的主要参数包括：高电平持续时间 T_{on} 和低电平持续时间 T_{off} 。当检测到一次与设定值相吻合的高电平时间和低电平时间，呼叫进程音分析器就判决为回铃音，回铃音计数器将加 1。 T_{on} 和 T_{off} 允许的误差均为 $\pm 20\%$ 。

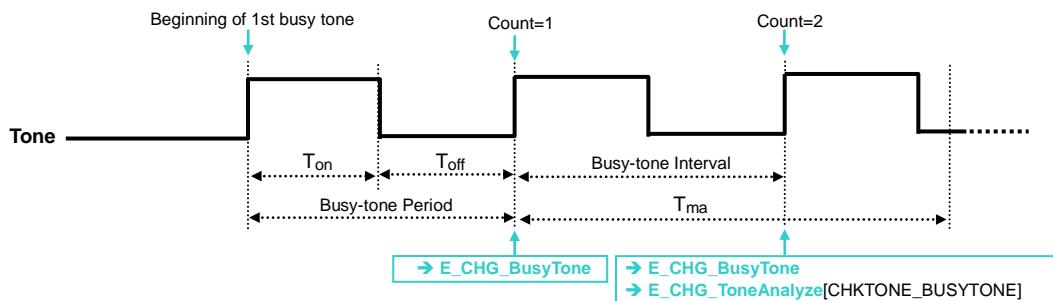
对于第 1 呼叫进程音检测器的回铃音检测器，配置项 [RingEchoTonePara](#) 或函数 [SsmSetRingEchoTonePara](#) 用于设置 T_{on} 和 T_{off} ；对于第 2 呼叫进程音检测器的回铃音检测器（如果启用），配置项 [2ndRingEchoTonePara](#) 或函数 [SsmSet2ndRingEchoTonePara](#) 用于设置第 2 呼叫进程音检测器的 T_{on} 和 T_{off} 。

每当第 1 呼叫进程音检测器或第 2 呼叫进程音检测器中的任何一个回铃音检测器的回铃音计数器值发生变化时，驱动程序都会抛出 [E_CHG_RingEchoToneTime](#) 事件；而只有当回铃音计数器值变化为 1 时，驱动程序才抛出 [E_CHG_ToneAnalyze](#) 事件(携带 CHKTONE_ECHOTONE 参数)。函数 [SsmGetRingEchoToneTime](#) 也可用于获取回铃音的保持时间。

1.10.5.3.3 忙音检测器

1.10.5.3.1 常规功能描述

忙音的检测由忙音检测器完成。典型的忙音波形如下图所示：



一个忙音周期分为高电平 (T_{on}) 部分和低电平部分 (T_{off})，由于忙音信号的占空比为 50%，因此 T_{on} 和 T_{off} 的保持时间相同，驱动程序也只使用忙音周期来表达忙音的特征参数，如上图所示。忙音周期允许的误差为 $\pm 20\%$ 。

当输入信号的高、低电平持续时间与设置的忙音参数相匹配时，忙音检测器就会判定为忙音，相应的忙音计数器（上图中的 Count）就加 1，同时抛出 [E_CHG_BusyTone](#) 事件。当忙音计数器的值等于某个指定的值时，忙音检测器会抛出 [E_CHG_ToneAnalyze](#) 事件（带 CHKTONE_BUSYTONE 参数）。忙音检测器的结果也可以通过下列函数获取：

函数名称	功能简述
SsmGetBusyToneCount	获取第 1 呼叫进程音检测器输出的忙音周期的个数

SsmGet2ndBusyToneCount	获取第 2 呼叫进程音检测器输出的忙音周期的个数
SsmGetBusyToneLen	获取第 1 呼叫进程音检测器输出的忙音保持时间
SsmGet2ndBusyToneLen	获取第 2 呼叫进程音检测器输出的忙音保持时间

每个忙音检测器可以同时对 4 组具有不同忙音周期的忙音信号进行检测。

第 1 呼叫进程音检测器的忙音检测器的工作参数设置:

- 配置项 [BusyTonePeriod](#) (或 [SsmSetBusyTonePeriod](#)、[SsmSetBusyTonePeriodEx](#) 函数) 用来设置忙音周期。
- 配置项 [IsBusyToneDetermineCount](#) (或 [SsmSetIsBusyToneDtrCnt](#) 函数) 可以用来设置驱动程序在检测到第几个忙音时抛出 [E_CHG_ToneAnalyze](#) 事件。

第 2 呼叫进程音检测器的忙音检测器的工作参数设置:

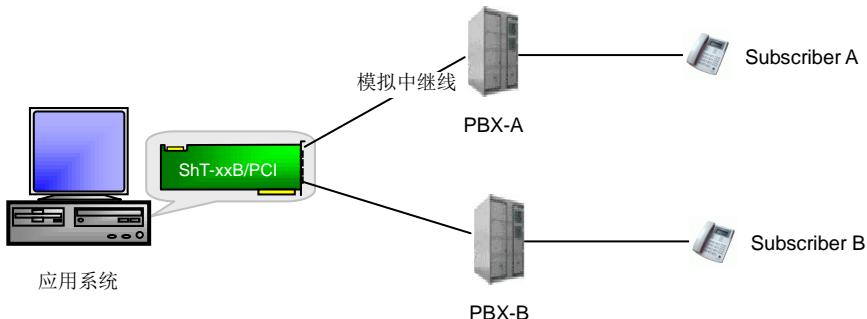
- 配置项 [2ndBusyTonePeriod](#) (或 [SsmSet2ndBusyTonePeriod](#)、[SsmSetBusyTonePeriodEx](#) 函数) 可以用来设置忙音周期。
- 配置项 [2ndIsBusyToneDetermineCount](#) (或 [SsmSet2ndIsBusyToneDtrCnt](#) 函数) 可以用来设置驱动程序在检测到第几个忙音时抛出 [E_CHG_ToneAnalyze](#) 事件。

在实际的应用中，由于语音信号中偶尔会出现类似于忙音特征的信号，从而导致驱动程序可能会对忙音进行误判。忙音检测器通过设置一个保护时间门限（记为 T_{max} ）来解决这种问题。当两个忙音周期之间的间隔大于 T_{max} 时，忙音计数器自动清零，从而避免了忙音的误检测。

配置项 [MaxBsTnOffTime](#) 用来设置忙音检测的保护时间 T_{max} 。第 1、第 2 呼叫进程音检测器的忙音检测器均使用此配置项。

1.10.5.3.2 背靠背忙音检测功能

对于模拟中继通道，在某些特殊的应用场合下，常规的忙音检测方法可能会失效。例如，对于下图中所示的基于 SHT 系列板卡的应用系统：



用户 A 通过 PBX-A 接入到应用系统的 SHT 系列板卡的中继通道，用户 B 通过 PBX-B 接入到应用系统的 SHT 系列板卡的中继通道，应用系统通过交换总线实现了用户 A 和用户 B 之间的双向通话。当其中一方挂机后（比如用户 A），PBX-A 会向板卡发送忙音；用户 B 听到忙音后也挂机，从而导致 PBX-B 也向板卡发送忙音。这时候如果板卡的电路交换尚未清除，由于回波的影响，任何一个通道上的忙音波形就会发生畸变，不再具有常规忙音的波形特征，忙音检测器就无法正确地检测出忙音，从而导致这 2 个通道被永远占用。

为了解决这个问题，忙音检测器增加了背靠背忙音检测的功能，当发生这类情况时仍然能够正确地检测出忙音。背靠背忙音检测功能无需配置，但其结果输出不是放在信号音检测器的结果输出中，而是单独 [E_CHG_BusyToneEx](#) 事件提供给应用程序。函数 [SsmGetBusyToneEx](#) 也可以用来获得背靠背忙音检测的结果。

注意：只有第 1 呼叫进程音检测器才具有背靠背忙音检测的功能。

1.10.5.3.4 自定义信号音检测器 (User-defined Tone Detector)

自定义信号音检测器只用于某些特殊的场合，比如，在拨号音检测器、回铃音检测器和忙音检测器工作时，还需要同时对一些用户自定义的波形进行检测。自定义信号音检测器可以检测连续信号音和周期性信号音。

第 1 呼叫进程音检测器的自定义信号音检测器的工作参数设置：

- ◆ 配置项 [AppointedToneAnalyzerSwitch](#) 用于设置工作状态。
- ◆ 配置项 [IsAppointedToneDetermineTime](#) 用来设置连续信号音的最小持续时间。判别原理与拨号音检测器相同。当信号音实际持续时间大于此配置项指定门限时，驱动程序输出 [E_CHG_ToneAnalyze](#) 事件（带 CHKTONE_APPPOINTEDTONE 参数）。
- ◆ 配置项 [AppointedTonePara](#)、[IsAppointedToneDetermineCount](#) 用来设置周期性信号的高电平持续时间和低电平持续时间、最小计数周期。判别原理与回铃音检测器相同。当信号音实际周期计数大于 IsAppointedToneDetermineCount 指定的最小计数周期时，驱动程序输出 [E_CHG_ToneAnalyze](#) 事件（带 CHKTONE_APPPOINTEDTONE 参数）。

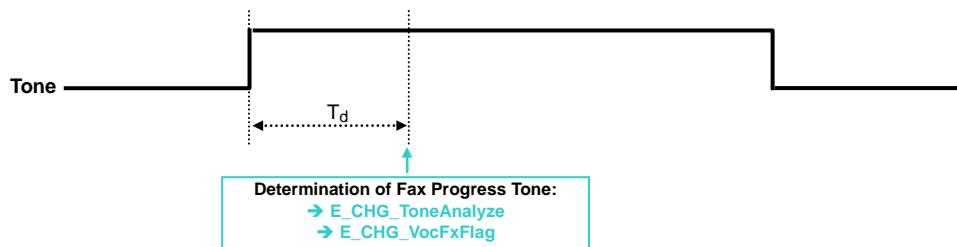
第 2 呼叫进程音检测器的自定义信号音检测器的工作参数设置：

- ◆ 配置项 [2ndAppointedToneAnalyzerSwitch](#) 用于设置工作状态。
- ◆ 配置项 [2ndIsAppointedToneDetermineTime](#) 用来设置连续信号音的最小持续时间。判别原理与拨号音检测器相同。当信号音实际持续时间大于此配置项指定门限时，驱动程序输出 [E_CHG_ToneAnalyze](#) 事件（带 CHKTONE_APPPOINTEDTONE 参数）。
- ◆ 配置项 [2ndAppointedTonePara](#)、[2ndIsAppointedToneDetermineCount](#) 用来设置周期性信号的高电平持续时间和低电平持续时间、最小计数周期。判别原理与回铃音检测器相同。当信号音实际周期计数大于 2ndIsAppointedToneDetermineCount 指定的最小计数周期时，驱动程序输出 [E_CHG_ToneAnalyze](#) 事件（带 CHKTONE_APPPOINTEDTONE 参数）。

1.10.5.4 传真进程音检测器(CTI 系列)

传真进程音检测器共有 2 个，称为第 1 传真进程音检测器和第 2 传真进程音检测器，用于同时检测 2 个具有不同频率特征的传真进程音。传真进程音为单音频、连续性的信号音。

传真进程音的波形特征如下图所示：



传真进程音检测器在检测到符合频率特征的信号音后开始计时，当计时达到判别门限 T_d 后，驱动程序就会顺序抛出 [E_CHG_VocFxFlag](#)、[E_CHG_ToneAnalyze](#) 事件（带 CHKTONE_VOICEF1 或 CHKTONE_VOICEF2 参数）。

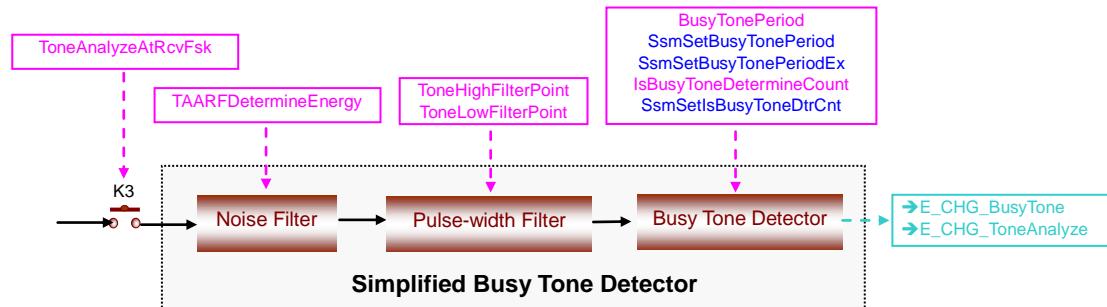
应用程序也可以通过函数 [SsmGetVocFxFlag](#) 获取检测结果。

传真进程音检测器的工作参数包括中心频率、带宽、带内能量占全频能量的百分比门限、最小保持时间 T_d ，第 1 传真进程音检测器的工作参数可以通过配置项 [VoiceFreqF1Para](#) 或函数 [SsmSetVoiceFxPara](#) 进行设置，第 2 传真进程音检测器的工作参数可以通过配置项 [VoiceFreqF2Para](#) 或函数 [SsmSetVoiceFxPara](#) 进行设置。

注意：对于模拟中继线类型的通道，在 [AutoDial](#) 任务的执行过程中，如果在线路上检测到了传真机的应答信号音，通常表明被叫（传真机）已经摘机，因此传真进程音检测器的检测结果会自动作为状态机的输入事件。

1.10.5.5 简易忙音检测器(Simplified Busy Tone Detector)

简易忙音检测器的原理框图、相关配置项或函数、输出事件如下图所示：



简易忙音检测器的工作状态由 K3 开关控制，可以通过配置项 [ToneAnalyzeAtRcvFsk](#) 设置。

噪声滤波器的工作原理与前面的“噪声滤波器”节完全一样，只是噪声判别门限是通过配置项 [TAARFDetermineEnergy](#) 来设置。在大多数情况下，噪声滤波器使用缺省配置即可，无须配置。

脉宽滤波器的工作原理与前面章节中“呼叫进程音检测器”中的“脉宽滤波器”部分完全一样，所用的配置项也完全相同。

与呼叫进程音检测器所不同的是，简易忙音检测器不检测输入信号的频率特征，即假定输入信号必定是信号音，因此不包含类似于呼叫进程音检测器中的频率检测器模块。

忙音的波形特征检测算法与呼叫进程音检测器中的常规忙音检测算法相同，并且同样可以同时对 4 组具有不同周期的忙音信号进行检测，波形特征的工作参数也使用第 1 呼叫进程音检测器中的忙音检测器所使用的波形参数，即

- ◆ 配置项 [BusyTonePeriod](#) 或函数 [SsmSetBusyTonePeriod](#)、[SsmSetBusyTonePeriodEx](#) 用来设置忙音周期。
- ◆ 配置项 [IsBusyToneDetermineCount](#) 或函数 [SsmSetIsBusyToneDtrCnt](#) 用来设置驱动程序在检测到第几个忙音时抛出 [E_CHG_ToneAnalyze](#) 事件。

简易忙音检测器也具有忙音抗误检测功能，也是通过配置项 [MaxBsTnOffTime](#) 用来设置忙音检测的保护时间 T_{max} 。

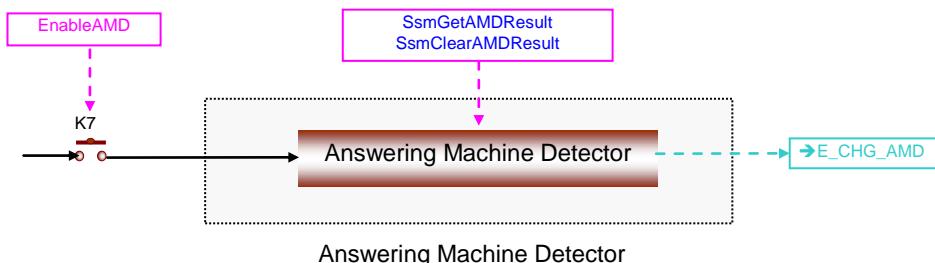
简易忙音检测器的输出与标准忙音检测器的输出方式完全一样。每当检测出一个忙音周期，忙音计数器（上图中的 Count）就加 1，同时抛出 [E_CHG_BusyTone](#) 事件。当忙音计数器的值等于某个指定的值时，忙音检测器会抛出 [E_CHG_ToneAnalyze](#) 事件（带 CHKTONE_BUSYTONE 参数）。

注意：

- 如果是模拟中继通道，[E_CHG_ToneAnalyze](#) 事件会同时被自动传递给该通道的呼叫状态机进行处理；
- 只有 SHT 系列的模拟中继通道才有简易忙音检测器。

1.10.5.6 机器应答检测器 (Answering Machine Detector)

机器应答检测器的原理框图、相关配置或函数、输出事件如下图所示：



机器应答检测器的工作由 K7 开关控制，可以通过配置项 [EnableAMD](#) 控制。

机器应答检测器是通过对线路语音的分析来判断应答方为机器应答还是人工应答，该检测器可以广泛用于外拨系统，提高座席接线员的工作效率。目前检测对方为机器应答的成功率 $\geq 90\%$ 。

机器应答检测器的结果可通过 [E_CHG_AMD](#) 事件携带参数输出或者使用函数 [SsmGetAMDResult](#) 通过其返回值获取。使用函数 [SsmClearAMDResult](#) 可以用来清除上一次的检测结果。

注意：

- 机器应答检测器需要信号音检测器开启的情况下才能正常工作。
- 如果是模拟中继通道，机器应答检测器会在拨号完毕后自动开启；如果是数字中继通道，机器应答检测器会在通道进入通话状态时自动开启。

1.10.5.7 增强信号音检测器(Enhanced Tone Detector)

增强信号音检测器原理是将信号音分为三类来检测，即连续性信号音、周期性信号音和 SIT 信号音（3 个连续单音频）。利用信号音的不同特征设置参数来检测对应的信号音。

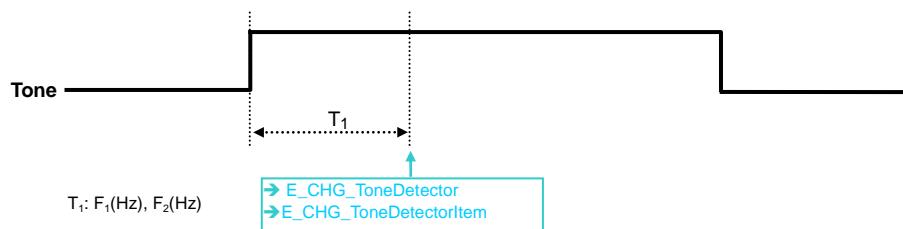
使用到的各参数说明如下：

- ◆ 第一中心频率 F_1
 - 对于连续性信号音，当检测到单音频时， F_1 就表示该单音频的频率；当检测到双音频时， F_1 就表示信号音的第一中心频率。
 - 对于周期性信号音，当检测到单音频时， F_1 就表示该单音频的频率；当检测到双音频时，该值表示信号音的第一中心频率。
 - 对于 SIT 信号音， F_1 表示第一段的频率。
- ◆ 第二中心频率 F_2
 - 对于连续性信号音，当检测到单音频时， F_2 等于 0；当检测到双音频时，该值表示信号音的第二中心频率。
 - 对于周期性信号音，当检测到单音频时， F_2 等于 0；当检测到双音频时，该值表示信号音的第二中心频率。
 - 对于 SIT 信号音， F_2 表示第二段的频率。
- ◆ 第三段的频率 F_3
 - 只对于 SIT 信号音有效， F_3 表示第三段的频率。
- ◆ 持续时间 T_1, T_2, T_3
 - 对于连续性信号音， T_1 表示高电平需要持续的时间； T_2, T_3 无效。
 - 对于周期性信号音， T_1 表示一个周期内高电平持续时间； T_2 表示一个周期内低电平持续时间； T_3 无效。
 - 对于 SIT 信号音， T_1, T_2, T_3 分别表示三段连续单音频的持续时间。

- ◆ C_{on}, C_{off} , 高低电平滤波点数（每个点时长为 16ms）
 - 用来设定波形上升沿和下降沿判断所需要的时间。当滤波点数超过 C_{on} 我们判断为进入高电平；当低电平滤波点数大于 C_{off} ，我们认为进入低电平。
- ◆ $F_{err}(\text{Hz})$ 频率误差门限
 - 用来指定参数 F_1, F_2, F_3 的允许误差范围，在误差范围内，仍然判断为 F_1, F_2, F_3 。

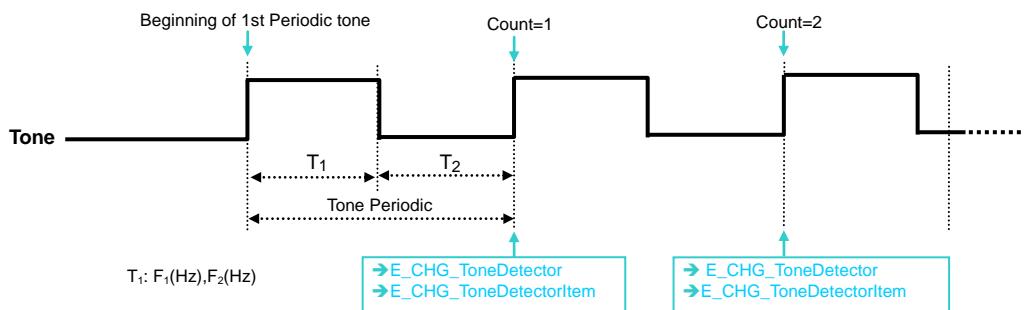
三类信号音示意图如下：

- 连续性信号音



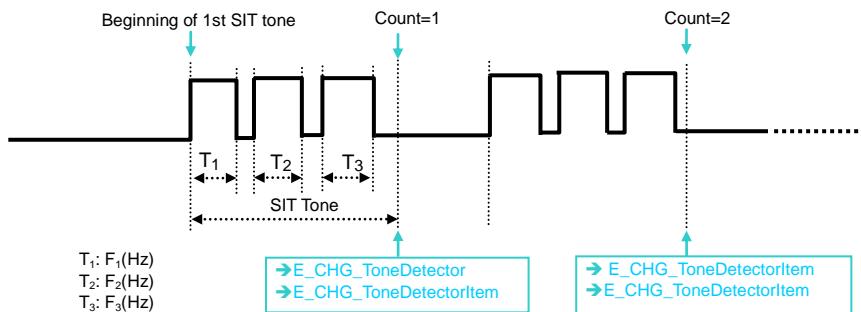
拨号音、传真信号音 F_1/F_2 属于连续性信号音，应作为连续性信号音配置。在检测到持续高电平时间 T_1 后抛出事件 [E_CHG_ToneDetector](#) 和 [E_CHG_ToneDetectorItem](#)。

- 周期性信号音



回铃音、催挂音、忙音属于周期性信号音，应作为周期性信号音配置。在检测到一个完整的周期之后会抛出事件 [E_CHG_ToneDetector](#) 和 [E_CHG_ToneDetectorItem](#)。

- SIT 信号音



SIT 信号音是连续三个单音频组成，三个单音频有各自自己的频率。此类应作为 SIT 信号音配置。在检测到三个连续单音频之后抛出事件 [E_CHG_ToneDetector](#) 和 [E_CHG_ToneDetectorItem](#)。

相关配置项：

- ◆ [ToneDetectorItem\[n\]](#) 详细描述了将要检测的信号音的各种参数，如频率，低电平持续时间，高电平持续时间等。
- ◆ [MaxToneDetectorItem](#) 表示增强信号音检测器将要检测的信号音种类的个数。
- ◆ [VoiceOffDetermineTime](#) 用于设置无声检测的时间。在普通模式下该参数直接取于回铃音周期，增强模式下需要另外设置。一般该参数取回铃音周期的 1.2 倍。

事件 [E_CHG_ToneDetector](#) 的 dwParam 参数包含了信号音的类型以及该信号音类型对应 [ToneDetectorItem\[N\]](#) 中的 N。[E_CHG_ToneDetectorItem](#) 事件记录了信号音的周期，dwParam 的低两字节指示了信号音持续的周期数。

为了和之前的应用程序保持兼容，增强型信号音检测器除了提供 [E_CHG_ToneDetector](#) 和 [E_CHG_ToneDetectorItem](#) 事件接口外，仍然提供以下兼容接口：

- ◆ API 接口：[SsmStartToneAnalyze](#)，[SsmCloseToneAnalyze](#)，[SsmGetToneAnalyzeResult](#)，[SsmClearToneAnalyzeResult](#)
- ◆ 事件接口：[E_CHG_ToneAnalyze](#)

1.10.6 信号音发生器(Tone Generator) (CTI 系列)

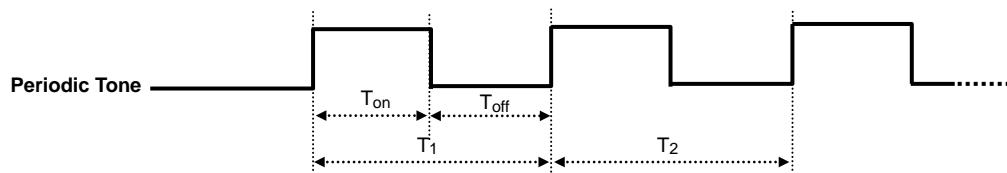
信号音是具有某种恒定频率的、符合一定节拍的呼叫进程音。呼叫进程音采用不同的节拍来表示不同的线路状态，如对端挂机、线路忙等。三汇板卡的每个通道都具有独立的信号音发生器 (Tone Generator)。

信号音发生器的工作参数包括中心频率（可以是单音频或双音频）、信号幅度和节拍。根据节拍的变化，信号音可以分为持续性信号音和周期性信号音。

持续性信号音是指信号音发生器一旦启动后，线路上的输出就保持高电平，直到信号音发生器被关闭。拨号音就属于持续性信号音，其波形特征如下图所示：



周期性信号音是指信号音由数个信号周期组成，每个周期由一段持续的高电平和一段持续的低电平组成。忙音和回铃音属于周期性信号音。周期性信号音的典型波形如下图所示：



上图中, T_1 、 T_2 分别为信号音的第 1 个、第 2 个周期; T_{on} 和 T_{off} 分别为一个信号音周期内的高电平持续时间和低电平持续时间。 T_{on} 与 T_{off} 的比值称为占空比。忙音的占空比通常为 1: 1, 回铃音的占空比通常为 1: 4。

1.10.6.1 设置信号音发生器的工作参数

与频率和幅度相关的配置项或函数如下表所示:

函数名称	简要描述	相关配置项
SsmQueryOpSendTone	查询通道是否支持信号音发生器的操作	
SsmSetTxTonePara	设置每个通道上信号音发生器的频率和幅度	DefaultSendToneFrequency DefaultSendToneVolume
SsmGetTxTonePara	获取每个通道上信号音发生器的频率和幅度	

1.10.6.2 产生信号音

与信号音发生器相关的函数如下表所示:

函数名称	简要描述
SsmSendTone	启动信号音发生器
SsmSendToneEx	启动信号音发生器 (可在参数中设置 T_{on} 与 T_{off})
SsmStopSendTone	关闭信号音发生器
SsmChkSendTone	获取信号音发生器正在发送的信号音类型

注意: 一旦启动了信号音发生器后, 不能执行下列操作 (120A 系列板卡除外):

- 语音播放;
- DTMF 发生器;
- 发送 FSK 数据。

1.10.7 回波抵消器 (CTI 系列)

回波抵消功能可以改善通话的质量, 大幅提升 DTMF 检测器、Barge in 检测器的准确性。

回波抵消器的工作状态可以通过函数 [SsmSetEchoCanceller](#) 或配置项 [EnableEchoCanceller](#) 进行设置, 缺省值为开启。注意: 由于回波抵消功能开启时会改变来话信号, 因此, 如果需要在通道上进行 FSK 数据接收操作, 请关闭回波抵消器, 待 FSK 数据接收操作完成后再重新开启, 以保证数据通信的可靠性。

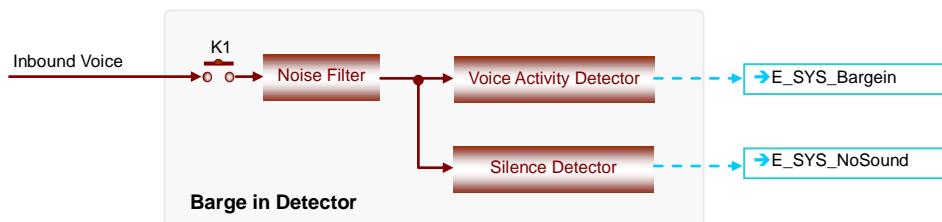
对于 SHD 系列的 30 通道、60 通道和 120 通道规格的板卡, 通过修改配置项 [LoadFskBin](#) 的设置, 可以增强回波抵消器的性能和效果, 但由于 DSP 的资源耗尽, 因此会丧失录音功能和 FSK 收/发功能。

Synway 板卡提供的回波抵消器的工作模式和参数的缺省值已经能够满足绝大多数应用场景，只有在某些特殊的场合，才需要调整回波抵消器的工作模式和工作参数，请在 Synway 公司技术人员的指导下进行。回波抵消器的工作参数可以通过函数 [SsmSetEchoCancelDelaySize](#) 或配置项 [EchoCancelDelaySize](#) 进行设置。

1.10.8 Barge in 检测器（Barge in Detector）

Barge in 检测器用于检测线路上的语音活动，通常应用于声控录音、自动语音识别（ASR）等场合。如果通道具有回波抵消能力，由于本端去话信号（例如，放音操作时产生的语音）的回波经回波抵消器后不会进入 Barge in 检测器，Barge in 检测器的检测结果就会非常准确。通道是否具有 Bargein 检测器，请查询相关的硬件说明资料。

Barge in 检测器由噪声滤波器（Noise Filter）、语音活动检测器（Voice Activity Detector）和静默检测器（Silence Detector）组成，如下图所示：



开关 K1 控制 Barge in 检测器的工作状态，由驱动程序自动控制。驱动程序的控制机制为：

通道类型	驱动程序的动作	触发条件
需要参与会议调度的会议通道	闭合 K1	通道加入会议室
	断开 K1	退出会议室
TUP/ISUP 通道 No.1 信令通道	闭合 K1	通道状态迁移到 S_CALL_TALKING
	断开 K1	通道状态迁移到 S_CALL_WAIT_REMOTE_PICKUP (去话呼叫) 或 S_CALL_RINGING (来话呼叫)
ISDN 通道	闭合 K1	通道状态迁移到 S_CALL_WAIT_REMOTE_PICKUP (去话呼叫) 或 S_CALL_RINGING (来话呼叫)
	断开 K1	通道状态迁移到 S_CALL_TALKING
模拟中继线通道	闭合 K1	通道状态迁移到 S_CALL_STANDBY
	断开 K1	通道状态迁移到 S_CALL_STANDBY
坐席通道	闭合 K1	通道状态迁移到 S_CALL_PICKUPED
	断开 K1	通道状态迁移到 S_CALL_STANDBY
模拟中继线录音通道	闭合 K1	检测到摘机
	断开 K1	检测到挂机

对于 SHD 系列板卡，如果希望通道处于任何状态时，K1 开关都保持闭合状态，可以通过配置项

[AlwaysDetectBargeIn](#) 实现。

对于 SHN、DTP、SHV 系列板卡，开关 K1 的控制仅由配置项 [AlwaysDetectBargeIn](#) 实现，驱动程序不会进行自动控制。

噪声滤波器用于区分线路噪声和真实的语音信号。配置项 [VoiceEnergyMinValue](#) 用来设置噪声滤波器的噪声门限，当来话信号或去话信号的强度低于设定值时，会被判定为线路噪声。函数 [SsmSetBargeInSens](#) 或配置项 [BargeInSensitive](#) 用来设置噪声滤波器的灵敏度。当来话中包含了 DTMF 信号时，是否将 DTMF 信号视为语音信号，取决于配置项 [DefaultDtmfIsSound](#) 的设定值。

语音活动检测器用于检测线路上的语音活动。当来话信号中检测到了语音活动，并且保持时间超过函数 [SsmGetIsBargeInDtrmTime](#) 或配置项 [BargeInDtrmTime](#) 的设定值时，输出 [E_SYS_BargeIn](#) 事件。函数 [SsmDetectBargeIn](#) 也可以用来获取检测结果。

静默检测器用于判断线路上是否保持静默。当来话信号保持静默的时间超过函数 [SsmSetNoSoundDtrmTime](#) 或配置项 [IsNoSoundDtrmTime](#) 的设定值时，输出 [E_SYS_NoSound](#) 事件。函数 [SsmDetectNoSound](#) 可以用来获取静默的检测结果，函数 [SsmGetNoSoundTime](#) 用来获取静默的保持时间。

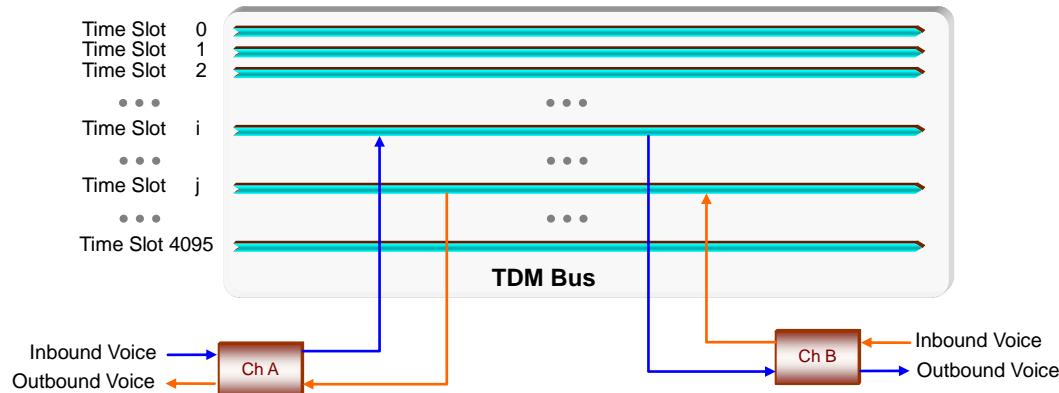
1.10.9 TDM 交换总线

Synway 板卡支持 2 种模式的语音交换能力：

- 1) 内部交换能力：板卡内部的通道之间的语音交换能力。所有的 Synway 板卡都具有内部交换能力
- 1) 外部交换能力：分布在不同板卡上的通道之间的交换能力。除 4 通道 (-USB 型号) 和 8 通道规格的 SHT、ATP 系列板卡外，所有 Synway 板卡都具有外部交换能力。

cPCI 总线规格的板卡采用 H.110 总线，其余板卡采用 H.100 总线。

使用 TDM 交换总线进行语音交换的原理如下图所示：



TDM 交换总线的原理

上图中，建立通道 A 与通道 B 的双向连接的操作过程为：Ch A 的来话上行到 TDM 总线的时隙 i (Time Slot i)，将时隙 i 下行到 Ch B，实现了将 Ch A 的来话送给 Ch B 听的功能；同样，Ch B 的来话上行到 TDM 总线的时隙 j (Time Slot j)，将时隙 j 下行到 Ch A，就实现了将 Ch B 的来话送给 Ch A 听的功能。

系统初始化时，如果配置项 [InVoiceToBus](#) 设置为 1 (缺省值)，驱动程序会将通道的来话信号连接到 TDM 总线的某个时隙上。

常用的 TDM 总线交换的函数包括：

交换类别	函数名称	功能简述
------	------	------

建立双向连接	SsmTalkWith	建立 2 个通道之间的双向连接
	SsmTalkWithEx	建立 2 个通道之间的双向连接, 可以设置参与交换的 2 个通道的音量
拆除双向连接	SsmStopTalkWith	拆除 ch1 到 ch2 的双向连接
建立单向连接	SsmListenTo	建立说者通道到听者通道的单向连接
	SsmListenToEx	建立说者通道到听者通道的单向连接, 可设置说者通道的音量
	SsmLinkFrom	建立说者通道到听者的单向连接, 允许听者通道同时听多个通道
	SsmLinkFromEx	建立说者通道到听者通道的单向连接, 允许听者通道同时听多个通道, 并且可设置说者通道的音量
	SsmLinkFromAllCh	批量建立一个说者通道到多个听者通道的单向连接, 允许听者通道同时听多个通道, 并且可设置说者通道的音量
拆除单向连接	SsmStopListenTo	拆除用函数 SsmListenTo 或 SsmListenToEx 建立的单向连接
	SsmStopLinkFrom	拆除用函数 SsmLinkFrom 或 SsmLinkFromEx 建立的单向连接
	SsmUnLinkFromAllCh	拆除用函数 SsmLinkFromAllCh 建立的单向连接
拆除全部连接	SsmClearChBusLink	拆除通道上建立的所有总线连接, 使通道恢复到驱动程序初始化完成时的总线连接状态

高级的 TDM 总线交换函数用于建立或拆除通道与总线时隙之间的单向连接, 包括下列函数:

交换类别	函数名称	功能简述
通道上行到总线时隙	SsmLinkToBus	改变上总线混音器的输出信号上行到总线的时隙
	SsmUnLinkToBus	恢复将上总线混音器的输出信号连接到缺省的总线时隙
总线时隙下行到通道	SsmLinkFromBus	建立总线时隙到听者通道的单向连接
	SsmLinkFromBusEx	建立总线时隙到听者通道的单向连接, 并可设置音量
	SsmUnLinkFromBus	拆除总线时隙到听者通道的单向连接

1.10.10 分布式电话会议(CTI 系列)

1.10.10.1 基本概念

➤ 分布式电话会议技术

CTI 系列板卡上的每个语音通道都具有独立的混音器，不需要借助额外的电话会议板卡，就可以实现电话会议功能，称为分布式电话会议。分布式电话会议技术是 Synway 公司的专利技术。注意，当开启总线软交换功能时，驱动将不支持会议功能。对于没有 CT-BUS 总线接口的卡，由于没有 CT-BUS 总线芯片，因此也不支持跨卡电话会议功能。

➤ 会议室

会议室由参与同一个会议的通道组成。每个会议室具有一个唯一的编号，可由应用程序指定，也可由驱动程序执行分配。一个会议室中通道的总数最少可以为 3 个，最多可以达到整个应用系统的通道总数，可以通过配置项 [ConfDefaultMaxGroupMember](#) 进行设置，或者在调用函数 [SsmCreateConfGroup](#) 创建会议室时在参数中指定。

会议室的个数可以通过配置项 [ConfMaxGroup](#) 进行设置。

➤ 会议室编号

每个会议室都有一个唯一的编号，称为会议室编号。会议室编号从 0 开始，最大编号为配置项 [ConfMaxGroup](#) 的设定值减 1。

➤ 会议通道

加入某个会议的通道称为会议通道。

➤ 会议通道的编号

会议通道在会议室中的逻辑编号，从 0 开始编号。注意：会议通道的编号与通道编号不一定相同。

➤ 会议调度程序

由于会议混音器的输入信号源的数量是有限的，因此，对于成员数量较多的会议室，当同时发言的通道数量超过会议混音器允许的最大值时，必然有一部分通道的声音无法进入会议混音器。SynCTI 驱动程序包含了一个会议调度程序，可以对会议室中的发言情况进行自动调度。只有满足调度条件的通道，会议调度程序才会允许其声音信号进入会议混音器。

会议调度算法以会议通道的调度优先级作为调度依据。在分配会议混音器资源时，首先满足优先级高的通道，最后满足优先级低的通道，直到全部会议混音器资源耗尽为止。

假设可供调度的会议混音器资源为 N，主持人通道数量为 m，主席通道数量为 n，动态发言通道中正在发言的通道数量为 k，会议调度算法的调度规则为：

- (1) 首先为 m 个主持人分配会议混音器。分配完毕后，如果还有剩余资源，进入下一步。
- (2) 如果 $n \leq N - m$ ，所有主席通道均可以获得会议混音器；如果 $n > N - m$ ，调度程序将全部 n 个主席通道的语音信号强度（即嗓门大小）按照从高到低的方式进行排序，前面 $N - m$ 个将获得会议混音器。如果处理完主席通道后还有多余的混音器资源，进入下一个优先级的分配。
- (3) 如果 $k \leq N - m - n$ ，所有正在发言的 k 个动态发言通道均获得会议混音器；如果 $k > N - m - n$ ，调度程序将全部 k 个动态发言通道的语音信号强度按照从高到低的方式进行排序，排在前面的 $N - m - n$ 个会议通道将获得会议混音器。在动态发言通道占用一个混音器后，如果没有其它优先级更高的通道与其抢占混音器，该通道会

一直保持混音器的占用状态。

➤ 会议通道的发言模式

会议通道加入到会议室时，需要给通道分配发言权限。SynCTI 驱动程序支持 6 种发言模式，如下表所示：

发言模式	描述
主持人	可以发言，也可以听取其它通道的发言。此模式具有最高的调度优先级。具有主持人发言模式的会议通道简称为主持人通道。建议应用程序为每个会议室只设置 1 个主持人通道
主席	可以发言，也可以听取其它通道的发言。此模式的调度优先级低于主持人模式，但高于动态发言模式。具有主席发言模式的会议通道简称为主席通道
动态发言	可以发言，也可以听取其它通道的发言。此模式的调度优先级低于主席模式。具有动态发言模式的会议通道简称为动态发言通道
听众	不能发言，只能听取其它通道的发言。听众通道不参与会议调度。具有听众发言模式的会议通道简称为听众通道
背景音乐	可以发言，但能否听取其它通道的发言取决于配置项 PlayVoicesListen 的设置，一般用于向会议室播放背景音乐。 注意：此模式的通道是否参与会议调度、以何种优先级进行调度，取决于配置项 BackgroundVoicePriority 的设置。 具有背景音乐发言模式的会议通道简称为背景音乐通道
动态只发言	可以发言，但不能听取其它通道的发言。具有动态发言模式但不能听取其他通道发言的会议通道简称为动态只发言通道

➤ DTMF 锯位 (DTMF Clamping)

当参加会议的人在话机上进行按键操作时，线路上的 DTMF 信号音会进入会议混音器，被其它会议通道听见。Synway 驱动程序支持 DTMF 锯位功能，可以有效过滤按键产生的 DTMF 信号。

DTMF 锯位功能的实现原理是：当板卡上的 DTMF 检测器在会议通道上检测到 DTMF 信号时，会立即断开该通道的来话信号；当 DTMF 信号消失后，自动恢复来话信号。

配置项 [ClearInVoiceOnRxDtmf](#) 或函数 [SsmSetFlag](#)（携带 F_ClearInVoiceOnRcvDtmf 参数）用于设置是否开启 DTMF 锯位功能。

➤ 禁止发言功能(Coming Voice Blocking)

会议通道加入会议室后，在正常情况下，只有会议通道满足调度规则，它的来话信号才可以进入会议混音器，被其它会议通道听见。如果需要临时禁止该通道的发言，可通过配置项 [InVoiceToBus](#) 或函数 [SsmSetFlag](#)（携带 F_InVoiceToBus 参数）实现。

1.10.10.2 管理会议室

下表是有关会议室的函数和配置项的一览表：

函数名称	类别	功能描述
SsmCreateConfGroup	函数	创建一个会议室
SsmFreeConfGroup	函数	注销一个会议室

ConfDefaultMaxGroupMember	配置项	设置会议室的工作参数
ConfDefaultMaxGroupSpeaker		
ConfDefaultMaxGroupSpeaking		
SsmGetConfCfgInfo	函数	获取会议室的相关信息
SsmGetTotalConfGroup		
SsmGetConfGrpInfo		
SsmGetConfGrpCfgInfo		
SsmGetConfGrpId		
SsmValidateGrpId		
SsmGetConfGrpMmbrId	函数	获取会议室内成员的相关信息
SsmGetConfGrpMmbrInfo		
SsmGetConfChInfo		

1.10.10.3 管理会议通道

有关会议通道的相关函数和配置项如下表所示：

函数名称	类别	功能描述
SsmJoinConfGroup	函数	将通道加入到会议室
SsmExitConfGroup	函数	将通道退出会议室
SsmSetListenVlmInConf	函数	设置通道听到的会议声音的音量

1.10.10.4 给会议室播放背景音乐

如果需要给会议室播放背景音乐、会议通知等信息，实现方式为：

- 选择一个执行语音播放任务的会议通道。选择通道的方法包括：
 - ✧ 为会议室单独设置一个背景音乐通道。这种方式非常灵活，可以按照实际需要设置通道的发言优先级，但缺点是需要占用一个额外的通道。
 - ✧ 在会议室中选择一个主持人通道。如果会议室中有主持人发言模式的会议通道，这是最合理的解决方案，但在编写程序时应尽量让该通道最后退出会议室。
 - ✧ 如果会议室中没有主持人会议通道，可选择一个主席通道或动态发言通道。建议优先选择主席通道。这种方式的缺点是，如果执行播放任务的会议通道因某种原因（如连接意外断开等）中途退出会议室时，应用程序需要将播放任务转移到其它会议通道上。
- 调用函数 [SsmSetPlayDest](#) 闭合 SHT/SHD/SHN 系列板卡原理框图中的 k1-1 开关，更多信息请参见本章中“[SHT 系列板卡的工作原理框图](#)”或“[SHD 系列板卡的工作原理框图](#)”或“[SHN 系列板卡的工作原理框图](#)”部分内容。

- 调用语音播放的函数，更多信息请参见本章中“[语音播放](#)”部分内容。

1.10.10.5 录制会议室的声音

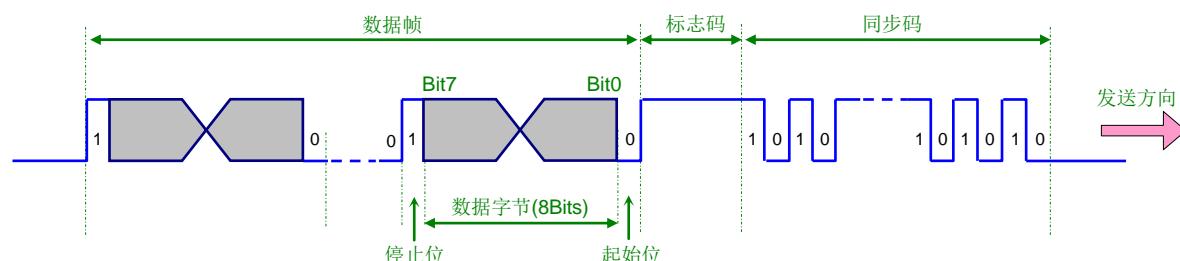
如果需要录制会议室播的声音，实现方式为：

- 选择一个执行录音任务的会议通道。任何一个会议通道都可以用来执行录音任务，但应尽可能选择中途不会退出会议室的通道，以避免在通道之间转移录音任务。建议的通道选择方法包括：
 - ✧ 优先选择用于播放背景音乐的会议通道。
 - ✧ 选择一个主持人通道。
 - ✧ 选择一个主席通道。
 - ✧ 选择一个动态发言通道。
 - ✧ 选择一个听众通道或动态发言通道。
- 如果执行录音任务的会议通道同时也执行播放背景音乐的任务，并且是 SHD/SHN 系列板卡上的通道，则应调用函数 [SsmSetRecBack](#) 闭合“[SHD 系列板卡的工作原理框图](#)”或“[SHN 系列板卡的工作原理框图](#)”中的 k6-1 开关和 k6-2 开关。
- 调用语音录制的函数，更多信息请参见本章中“[录音操作](#)”部分内容。

1.11 FSK 数据收发器（CTI 系列）

BFSK (Frequency Shift Keying)，是采用不同的载波频率表示二进制数据的 0 和 1 的数字通信技术，通常用于在电话线上发送主叫号码、短信等信息。

BFSK 的数据流以帧为单位，帧格式如下图所示：



一段完整的 BFSK 数据流由同步码、标志码和数据三部分组成。同步码是 0/1 翻转的波形，标志码为高电平 1。每个数据字节的前面会插入起始位 0，后面会插入停止位 1。每个数据字节的比特 0 最先发送，比特 7 最后发送。

注意：部分协议将同步码和标志码一起称为“同步引导串”来标志同步是否建立。其中，标志码被称为“同步结束字符”。

Synway 的 CTI 系列语音卡为每个语音信道均配备了 FSK 数据收发器，采用半双工工作模式，波特率为 1200bps，比特 1 对应的频率是 1200Hz，比特 0 对应的频率是 2200Hz。

1.11.1 FSK 数据发送器

FSK 数据发送器的相关函数、配置项和事件如下表所示：

类别	名称	描述
配置项	FreqBit0 FreqBit1 Baudrate MdlAmp	设置 FSK 数据发送器工作参数
函数	SsmSetFskPara	设置 FSK 数据发送器的工作参数
	SsmTransFskData	将数据转换为 BFSK 格式的二进制数据流
	SsmStartSendFsk	启动 FSK 数据发送器
	SsmCheckSendFsk	查询 FSK 数据发送器的进展情况
	SsmStopSendFsk	停止 FSK 数据发送器
事件	E_PROC_SendFSK	当驱动程序完成全部数据的发送后，会向应用程序抛出此事件

1.11.2 FSK 数据接收器

Synway 板卡的 FSK 数据接收器的波特率 1200bps, 比特 0 的载波频率为 2200Hz, 比特 1 的载波频率为 1200Hz, 均不可通过函数或配置项进行修改。

有关 FSK 接收器的函数、事件和配置项如下表所示：

名称	描述
配置项: ToneAnalyzeAtRcvFsk	设置 FSK 数据接收器工作时是否分析信号音
配置项: FskMarkSignal	设置 FSK 数据接收器判定接收数据的条件
配置项: FskFrameMode	设置 FSK 数据接收器默认接收的帧格式
配置项: FskEchoCancelDelay 函数: SsmSetFlag (携带 F_EchoCancelInFsk 参数)	当 FSK 接收器处于工作状态时，是否关闭回波抵消器
函数: SsmStartRcvFSK 函数: SsmStartRcvFSK_II 函数: SsmStopRcvFSK	启动，停止 FSK 数据接收器
函数: SsmGetRcvFSK 函数: SsmCheckRcvFSK	获取 FSK 数据及 FSK 数据接收器状态查询
函数: SsmClearRcvFSKBuf	清空驱动程序 FSK 数据接收器缓冲区
事件: E_PROC_RcvFSK	启动 FSK 数据接收器后，驱动检测到接收 FSK 任务结束后，会向应用程序抛出此事件

1.12 传真 (CTI 系列)

1.12.1 传真资源通道的数量

对于 SHT 系列、SHD 系列，下列型号的板卡支持传真资源通道，包括：

- ◆ SHT-8B/PCI/FAX
- ◆ SHT-8C/PCI/FAX
- ◆ SHT-16B-CT/PCI/FAX
- ◆ SHT-16B-CT/cPCI/FAX
- ◆ SHT-16C-CT/PCI/FAX
- ◆ SHT-16D-CT/PCI
- ◆ SHD-30B-CT/PCI/SS7/FAX
- ◆ SHD-60B-CT/PCI/SS7/FAX
- ◆ SHD-30B-CT/cPCI/SS7/FAX
- ◆ SHD-60B-CT/cPCI/SS7/FAX
- ◆ SHD-30C-CT/PCI/FAX
- ◆ SHD-60C-CT/PCI/FAX
- ◆ SHD-30E-CT/PCI/FAX(SSW)
- ◆ SHD-60E-CT/PCI/FAX(SSW)

- ◆ SHD-120E-CT/PCI/FAX(SSW)
- ◆ SHD-240E-CT/PCI/FAX(SSW)
- ◆ SHD-30E-CT/PCIe/FAX
- ◆ SHD-60E-CT/PCIe/FAX
- ◆ SHD-120E-CT/PCIe/FAX
- ◆ SHD-240E-CT/PCIe/FAX

对于 SHF 系列板卡，下列型号的板卡支持传真资源通道：

- ◆ SHF-2D/PCI
- ◆ SHF-4D/PCI
- ◆ SHF-4D/PCIe

各板卡传真资源通道如下表所示：

Board Model	传真资源通道数量											
	2		4		16		24		32		64	
	N _{Fax}	N _{Voc}										
SHT-8B/PCI/FAX	-	-	4	8	-	-	-	-	-	-	-	-
SHT-8C/PCI/FAX	-	-	4	8	-	-	-	-	-	-	-	-
SHT-16B-CT/PCI/FAX	-	-	4	16	-	-	-	-	-	-	-	-
SHT-16B-CT/cPCI/FAX	-	-	4	16	-	-	-	-	-	-	-	-
SHT-16C-CT/PCI/FAX	-	-	4	16	-	-	-	-	-	-	-	-
SHT-16D-CT/PCIe	-	-	4	16	-	-	-	-	-	-	-	-
SHD-30B-CT/PCI/SS7/FAX	-	-	-	-	-	-	24	30	-	-	-	-
SHD-60B-CT/PCI/SS7/FAX	-	-	-	-	16	60	-	-	-	-	-	-
SHD-30B-CT/cPCI/SS7/FAX	-	-	-	-	-	-	24	30	-	-	-	-
SHD-60B-CT/cPCI/SS7/FAX	-	-	-	-	16	60	-	-	-	-	-	-
SHD-30C-CT/PCI/FAX	-	-	-	-	-	-	24	30	-	-	-	-
SHD-60C-CT/PCI/FAX	-	-	-	-	16	60	-	-	-	-	-	-
SHD-30E-CT/PCI/FAX(SSW)	-	-	-	-	-	-	-	-	32	30	-	-
SHD-60E-CT/PCI/FAX(SSW)	-	-	-	-	-	-	-	-	-	-	64	60
SHD-120E-CT/PCI/FAX(SSW)	-	-	-	-	-	-	-	-	-	-	64	120
SHD-240E-CT/PCI/FAX(SSW)	-	-	-	-	-	-	-	-	-	-	64	240
SHD-30E-CT/PCIe/FAX	-	-	-	-	-	-	-	-	32	30	-	-
SHD-60E-CT/PCIe/FAX	-	-	-	-	-	-	-	-	-	-	64	60
SHD-120E-CT/PCIe/FAX	-	-	-	-	-	-	-	-	-	-	64	120
SHD-240E-CT/PCIe/FAX	-	-	-	-	-	-	-	-	-	-	64	240
SHF-2D/PCI	2	2	-	-	-	-	-	-	-	-	-	-
SHF-4D/PCI	-	-	4	4	-	-	-	-	-	-	-	-
SHF-4D/PCIe	-	-	4	4	-	-	-	-	-	-	-	-

注：上表中，N_{Fax}: 可用的传真资源通道总数；N_{Voc}: 可用的语音通道总数；-: 表示不支持。

1.12.2 传真操作支持的速率

Synway 板卡支持传真操作的型号，以及支持的传真速率如下表所示：

板卡型号	4800bps		9600bps		12000bps		14400bps		33600bps	
	发送	接收	发送	接收	发送	接收	发送	接收	发送	接收
SHT-8B/PCI/FAX	√	√	√		√		√			
SHT-8C/PCI/FAX	√	√	√	√	√	√	√			
SHT-16B-CT/PCI/FAX	√	√	√		√		√			
SHT-16B-CT/cPCI/FAX	√	√	√		√		√			
SHT-16C-CT/PCI/FAX	√	√	√	√	√	√	√			
SHT-16D-CT/PCIe	√	√	√	√	√	√	√	√		
SHD-30B-CT/PCI/SS7/FAX	√	√	√		√		√			
SHD-60B-CT/PCI/SS7/FAX	√	√	√		√		√			
SHD-30B-CT/cPCI/SS7/FAX	√	√	√		√		√			
SHD-60B-CT/cPCI/SS7/FAX	√	√	√		√		√			
SHD-30C-CT/PCI/FAX	√	√	√	√	√	√	√	√		
SHD-60C-CT/PCI/FAX	√	√	√	√	√	√	√	√		
SHD-30E-CT/PCI/FAX(SSW)	√	√	√	√	√	√	√	√		
SHD-60E-CT/PCI/FAX(SSW)	√	√	√	√	√	√	√	√		
SHD-120E-CT/PCI/FAX(SSW)	√	√	√	√	√	√	√	√		
SHD-240E-CT/PCI/FAX(SSW)	√	√	√	√	√	√	√	√		
SHD-30E-CT/PCIe/FAX	√	√	√	√	√	√	√	√		
SHD-60E-CT/PCIe/FAX	√	√	√	√	√	√	√	√		
SHD-120E-CT/PCIe/FAX	√	√	√	√	√	√	√	√		
SHD-240E-CT/PCIe/FAX	√	√	√	√	√	√	√	√		
SHF-2D/PCI	√	√	√	√	√	√	√	√	√	√
SHF-4D/PCI	√	√	√	√	√	√	√	√	√	√
SHF-4D/PCIe	√	√	√	√	√	√	√	√	√	√

1.12.3 传真操作支持的文件格式

Synway 板卡的传真支持 T.30 通信协议和 T.4 传真图像协议，发送的传真数据最高支持的格式可通过配置项进行设置，目前支持的有 MH、MR、MMR 编码格式。发送的传真文件支持标准 Tiff 文件。

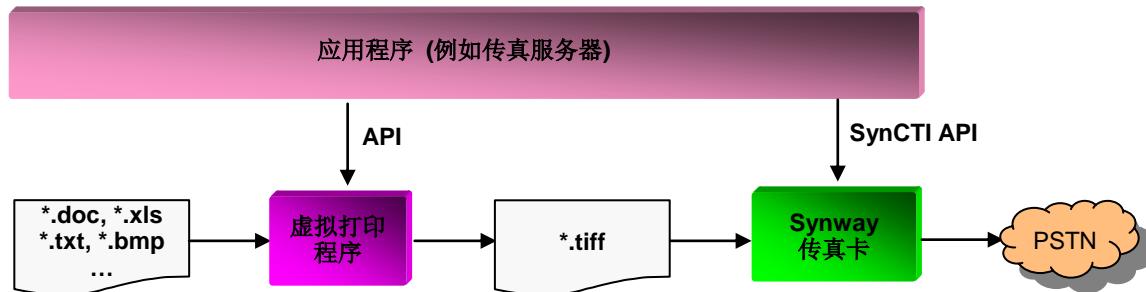
Tiff 文件是用于传真数据保存的主流文件，通用性强，其查看、归档、修改等都比较方便。Synway 传真支持的 Tiff 文件的属性包括：

- ✓ 黑白色
- ✓ MH、MR、MMR 编码格式
- ✓ 204*196、204 * 98、200 * 200、200 * 100 的分辨率
- ✓ 页面数据是每行 1728 个像素
- ✓ 其它属性，如下表所示：

Tiff 压缩属性	T4 能力属性	备注
2	--	CCITT HM—精简型 MH 编码
3	0	CCTIT G3 1D—MH 编码 无 EOL 标记行结束
3	4	CCTIT G3 1D—MH 编码 有 EOL 标记行结束
3	1	CCITT G3 2D—MR 编码 无 EOL 标记行结束
3	5	CCITT G3 2D—MR 编码 有 EOL 标记行结束
4	--	CCITT G4—MMR 编码

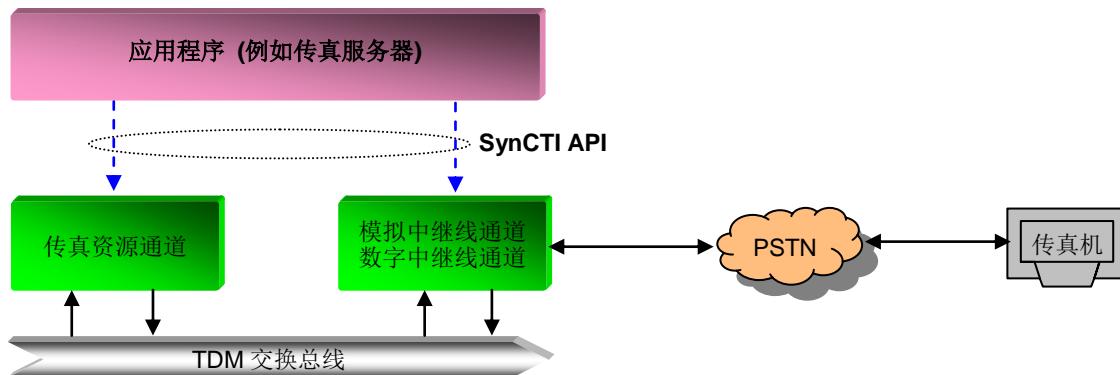
注：TIFF 压缩属性为 3 的情况下，T4 能力属性只能区分 MH 编码还是 MR 编码，至于是否有 EOL 标记行结束有很大的不准确性，驱动里增加了自适应功能来进一步确认文件的真正属性。

应用程序发送任意格式文件的工作原理如下图所示：



上图中，虚拟打印程序用于将任何格式的文件转换 tiff 格式。

一次完整的传真发送或接收过程需要使用模拟或数字中继线通道和传真资源通道，如下图所示：



传真的发送（或接收）过程为：

- (1) 在模拟或数字中继线通道上发起并建立呼叫
- (2) 建立传真资源通道与模拟或数字中继线通道的双向连接
- (3) 发送（或接收）传真文件
- (4) 拆除呼叫

1.12.4 设置传真操作的工作参数

设置传真操作的工作参数的相关函数如下表所示：

函数名称	功能概述
SsmFaxSetMaxSpeed	传真收/发的初始速率，亦为接收或发送传真时使用的最大速率。此速率对全部通道都起作用
SsmFaxSetID	设置本方的识别代码。通常为本方的电话号码，也可以设置为字母。这个代码将在接收或发送传真时被传送给对方传真机

1.12.5 获取传真操作的信息

传真过程中的相关参数可以通过调用以下函数获得：

函数名称	功能概述及获取的信息
SsmFaxGetSpeed	获取传真实际收/发速率。可能值为 24、48、72、96、120、144、168、192、216、240、264、288、312、336，分别表示 2400、4800、7200、9600、12000、14400、16800、19200、21600、24000、26400、28800、31200、33600bps
SsmFaxGetCodeMode	获取传真实际收/发编解码格式。可能值为 0、1、2，分别表示 MH、MR、MMR
SsmFaxGetID	获取对方的识别码。一般要在传真握手完成后才能获得
SsmFaxGetPages	获取传真收发过程中已经完成的页数。如果收/发传真结束后，调用本函数返回零，则表示该次传真收/发过程失败
SsmFaxGetAllBytes	获取当前传真发送页的总字节数

<u>SsmFaxGetSendBytes</u>	获取传真发送过程中当前页已经发送的字节数。当调用 SsmFaxStartSend 函数后，可以用本函数查看当前页已发送的字节数
<u>SsmFaxGetRcvBytes</u>	获取传真接收过程中当前页已接收的字节数。当调用 SsmFaxStartReceive 函数后，可以用本函数查看当前页收到的字节数
<u>SsmFaxGetChStateMsg</u>	获取指定通道上正在进行的传真收/发过程所处的状态信息
<u>SsmFaxCheckEnd</u>	检查指定通道上的传真接收或发送任务的执行情况

1.12.6 发送传真

有关传真发送的函数如下表所示：

函数名称	功能概述
<u>SsmFaxStartSend</u>	启动发送传真文件的任务
<u>SsmFaxStartSendEx</u>	启动发送传真文件的任务，可指定起始页和结束页
<u>SsmFaxSendMultiFile</u>	启动发送多个传真文件的任务
<u>SsmFaxSendMultiFileEx</u>	启动发送多个传真文件的任务，可指定起始页和结束页
<u>SsmFaxAppendSend</u>	追加单文件的传真发送，传真资源通道必须处于传真发送状态
<u>SsmFaxCheckEnd</u>	获取传真资源通道的工作状态
<u>SsmFaxSetMaxSpeed</u>	设置传真收发的速率
<u>SsmFaxSetID</u>	设置本方的传真识别码，对方传真的屏幕会显示该识别码
<u>SsmFaxGetID</u>	获取对方的传真识别码
<u>SsmFaxGetAllBytes</u>	获取当前传真发送页的总字节数
<u>SsmFaxGetSendBytes</u>	获取传真发送过程中当前页已经发送出去的字节数
<u>SsmFaxGetSpeed</u>	获取当前传真收发的速度
<u>SsmFaxGetPages</u>	获取传真收发过程中已经完成的页数
<u>SsmFaxStop</u>	强制停止当前的传真过程

<u>SsmFaxGetCodeMode</u>	获取传真过程中的编解码类型
------------------------------------------	---------------

1.12.7 接收传真

有关接收传真的函数如下表所示：

函数名称	功能概述
<u>SsmFaxStartReceive</u>	开始传真接收，接收到的传真数据将保存在设定的传真文件中
<u>SsmFaxCheckEnd</u>	获取传真资源通道的工作状态
<u>SsmFaxSetMaxSpeed</u>	设置传真收发的速率
<u>SsmFaxSetID</u>	设置本方的传真识别码，对方传真的屏幕会显示该识别码
<u>SsmFaxGetID</u>	获取对方的传真识别码
<u>SsmFaxGetRcvBytes</u>	获取传真接收过程中当前页已接收的字节数
<u>SsmFaxGetSpeed</u>	获取当前传真收发的速度
<u>SsmFaxGetPages</u>	获取传真收发过程中已经完成的页数
<u>SsmFaxStop</u>	强制停止当前的传真过程
<u>SsmFaxGetCodeMode</u>	获取传真过程中的编解码类型

1.13 板载音频功率放大器

ATP 系列、DST 系列、SHT 系列板卡上的 0 通道上都安装了模拟音频放大电路和扬声器输出接口，可以直接连接外置扬声器，将声音送到扬声器进行播放。有关板载音频功率放大器的详细内容请参见相应板卡的原理框图部分。

1.14 编写基于 Synway 板卡的应用程序

1.14.1 设置驱动程序的事件输出模式

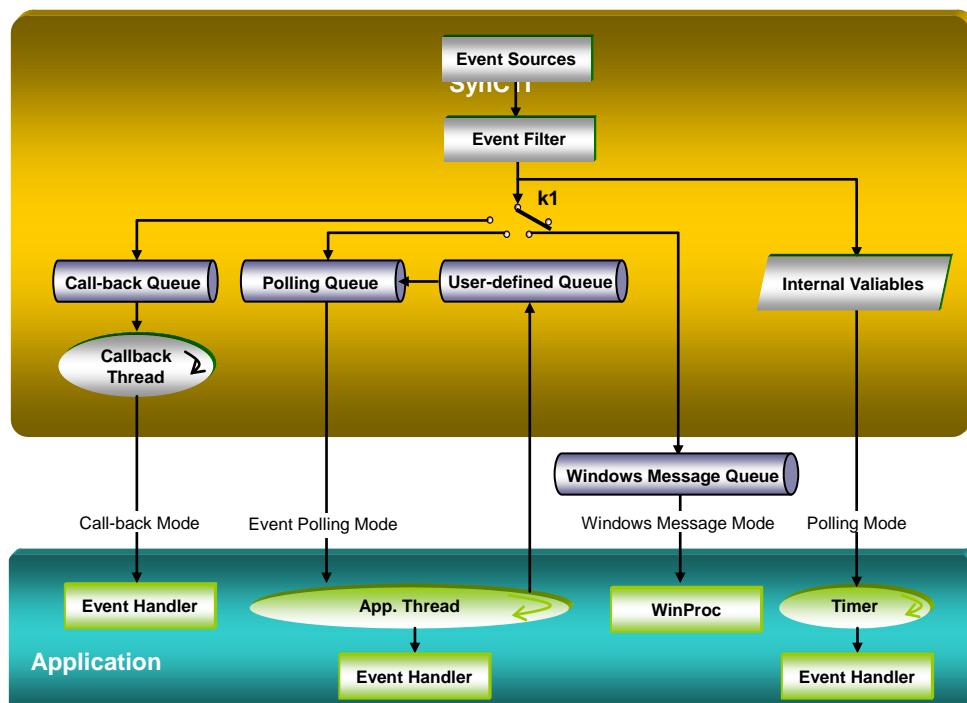
对 Synway 板卡进行编程，一般包括三部分：

- 初始化代码，包括初始化板卡和驱动程序，只需要在应用程序启动时调用一次。
- 业务处理代码，向驱动程序发布命令，对驱动程序通过事件返回的信息进行及时处理，以实现具体的业务。
- 卸载代码，包括释放板卡和驱动程序占用的资源，只需要在应用程序退出时调用一次。

SynCTI 驱动程序支持下列编程模式：

- ◆ 轮询模式：由应用程序不停地调用驱动程序提供的相关查询函数，以获取任务的进展情况。这种编程方式由于消耗计算机的资源较多，效率很低，只能适用于容量较小的应用系统，目前已经被逐渐淘汰。因此，本书不对此编程方式进行更多的说明。
- ◆ 事件等待：应用程序调用驱动程序提供的事件等待函数，当驱动程序没有事件时，应用程序的调用者线程被阻塞；当驱动程序抛出某个事件时，应用程序的调用者线程被重新激活，恢复对事件的处理。
- ◆ 事件回调：由应用程序向驱动程序注册一个回调函数，当驱动程序有事件发生时，由驱动程序调用回调函数，对事件进行处理。
- ◆ Windows 消息模式（只适用于 Windows 操作系统，且输出事件的数据结构只能是 MESSAGE_INFO）：驱动程序将事件发送到 Windows 的消息队列中，通过 Windows 统一的消息队列处理机制，来对事件进行处理。由于 Windows 消息模式可以携带的参数有限，因此实际使用这种编程模式的应用程序并不多，本书也不对此编程模式进行更多的说明。

这四种编程模式的消息过程如下图所示：



开关 k1 由函数 [SsmSetEvent](#) 进行控制。SynCTI 驱动程序启动后，会自动工作在轮询模式。如果应用程序要使用事件等待方式或事件回调方式，需要调用函数 [SsmSetEvent](#) 来改变驱动程序的工作模式，详细内容请参见后面的编程示例部分内容。

1.14.1.1 事件等待方式的编程示例

```
#include <windows.h>          // 包含需要的 Windows 头文件
#include "shpa3api.h"           // 包含 SynCTI 驱动程序需要的头文件
......

void main()
{
    // 初始化 SynCTI 驱动程序和板卡
    if ( ! SsmStartCti (.....) )
    {
        .....                         // 失败
        MessageBox( "Initialize Voice card failure" );
    }
}
```

```
        return;
    }

    //获取板卡、通道的相关信息
    int nMaxCh = SsmGetMaxCh();           //获取总的通道数量
    for (int ch=0; ch < nMaxCh; ch++)
    {
        int nChType = SsmGetChType (ch);      //获取通道类型
        .....
    }

    //设置驱动程序抛出事件的模式
    EVENT_SET_INFO EventMode;
    EventMode.DwWorkMode = EVENT_POLLING;     //使用事件等待方式
    SsmSetEvent(0xffff,-1,true, &EventMode);

    //事务处理
    MESSAGE_INFO Event;
    While(1)
    {
        memset(&MessageInfo, 0, sizeof(PMESSAGE_INFO));
        If ( 应用程序退出 ) break;           //应用程序退出代码

        // 等待并响应驱动程序抛出的事件
        If ( SsmWaitForEvent (50, &Event) == 0 )   //等待事件发生
        {
            switch(Event.EventCode)
            {
                case E_CHG_ChState:          //通道的状态发生变化
                    .....                         //业务处理代码
                    break;
                    .....
                case E_XXXXXX:
                    .....                         //业务处理代码
                    break;
                default:
                    break;
            }
        }
        SsmCloseCti();                      //退出时，关闭驱动程序
    }
}
```

如果需要使用扩展的输出事件的数据结构，可以按照下列方式编写上述代码中的“事务处理”部分：

```
//事务处理

UCHAR pucBuffer[300];
SSM_EVENT SsmEvent;

while(1)

{
    memset(&SsmEvent, 0, sizeof(SSM_EVENT));
    memset(pucBuffer, 0, sizeof(UCHAR)* 300);
    SsmEvent.pvBuffer = (PVOID)pucBuffer;
    SsmEvent.dwBufferLength = 300;
    if(SsmWaitForEventA(50, &SsmEvent) == 0)
    {
        switch(SsmEvent.wEventCode)
        {
            case E_CHG_ChState:
```

```
.....          //do something
break;
case E_RCV_DSTDChannel:
switch(SsmEvent.dwParam)
{
case DST_AUDIO_CHG:
.....          //do something
break;
case DST_MSG_CHG:
.....          //do something
break;
default:
break;
}
default:
break;
}
}
```

1.14.1.2 事件回调方式的编程示例

```
#include <windows.h>          // 包含需要的 Windows 头文件
#include "shpa3api.h"           // 包含 SynCTI 驱动程序需要的头文件
BOOL bExit = FALSE;            // 控制应用程序退出的变量
.....
//应用程序提供的回调函数，用于处理驱动程序抛出的事件
int CALLBACK MyCallback (WORD wEvent, int nReference, DWORD dwParam, DWORD dwUser)
{
    switch(wEvent)
    {
        case E_CHG_ChState:
.....          //do something
break;

        case E_XXXXXX:
.....          //do something
bExit = TRUE;                //退出应用程序
break;

.....
default:
break;
}

return 1;
```

```
}

//应用程序的主线程
void main()
{
    .....
    if ( ! SsmStartCti (.....) )           //初始化 SynCTI 驱动程序和板卡
    {
        .....
        MessageBox( "Initialize Voice card failure" );
        return;
    }

    int nMaxCh = SsmGetMaxCh();           //获取总的通道数量
    for (int ch=0; ch < nMaxCh; ch++)
    {
        int nChType = SsmGetChType (ch);   //获取通道类型
        .....
    }

    //设置驱动程序抛出事件的模式
    EVENT_SET_INFO EventMode;
    EventMode.DwWorkMode = EVENT_CALLBACK;      //事件回调模式
    EventMode.lpHandlerParam = MyCallback;       //注册回调函数
    SsmSetEvent(0xffff,-1,true, &EventMode);

    While(!bExit);                          //等待程序结束

    SsmCloseCti();                         //退出时，关闭驱动程序
}
```

如果需要使用扩展的输出事件的数据结构，可以按照下列方式编写应用程序提供的回调函数：

```
int CALLBACK MyCallback (PSSM_EVENT pEvent)
{
    switch(pEvent->wEventCode)
    {
        case E_CHG_ChState:
            .....
            //do something
            break;

        case E_XXXXXX:
            .....
            bExit = TRUE;          //退出应用程序
            break;

        case E_RCV_DSTDChannel:
            switch(pEvent->dwParam)
            {
                case DST_AUDIO_CHG:
                    .....
                    //do something
                    break;

                case DST_MSG_CHG:
                    .....
                    break;
            }
    }
}
```

```
.....          //do something

break;

default:
    break;

}

default:
    break;

}
return 1;
}
```

注意： 使用回调模式编程，需要在回调函数声明前增加 **CALLBACK** 关键字，否则可能会因为资源释放，出现程序崩溃的问题。

1.14.2 输出事件的数据结构

SynCTI 驱动程序在输出事件时，可以使用 2 种数据结构：**MESSAGE_INFO** 和 **SSM_EVENT**。**MESSAGE_INFO** 中包含了常用的参数，适用于 CTI 系列板卡和 REC 系列板卡；**SSM_EVENT** 是对 **MESSAGE_INFO** 的扩展，能提供更多的事件信息，通常用于 DST 系列板卡，但不适用于 Windows 消息编程模式，且要求 SynCTI 驱动程序的版本为 4.7.3.0 或更高。

例如，有多块卡合用的情况，其中一块是 DST 系列板卡，那么在输出事件时，必须使用 **SSM_EVENT** 这个数据结构，编程模式也只能选择事件等待模式或事件回调模式。

注意： **MESSAGE_INFO** 适用于函数 [SsmWaitForEvent](#) 和 [SsmGetEvent](#)，**SSM_EVENT** 适用于 [SsmWaitForEventA](#) 和 [SsmGetEventA](#)。

1.14.2.1 MESSAGE_INFO

MESSAGE_INFO 结构的声明为：

```
typedef struct _MESSAGE_INFO
{
    WORD    wEvent;           //事件编码
    int     nReference;       //参考值
    DWORD   dwParam;         //输出参数
} MESSAGE_INFO, *PMESSAGE_INFO;
```

➤ **wEvent**

wEvent 为事件编码。SynCTI 驱动程序抛出的事件大致分为下列几种：

- ❖ **E_CHG_xxxx:** 驱动程序内部的某个状态或计数器发生变化
- ❖ **E_PROC_xxxx:** 应用程序提交的某个任务有进展
- ❖ **E_SYS_xxxx:** 驱动程序检测到某个事件发生
- ❖ **E_RCV_xxxx:** 驱动程序从对端交换机收到某个消息或事件

其中，**xxxx** 表示事件的标志符。

SynCTI 驱动程序将内部产生的事件分为 2 类：常用事件和非常用事件。

全部事件的编码及其特性如下表所示：

事件 编码	shpa3api.h 中的宏	常用 事件	描述
0x0000	E_PROC_Recognize	√	语音识别结束
0x0001	E_CHG_ISDNStatus	—	ISDN: ISDN 协议的 LAPD 层状态发生变化
0x0002	E_RCV_Ss7Msu	—	SS7: 从 SS7 服务器收到新的消息 (MSU)
0x0003	E_CHG_Mtp3State	—	SS7: SS7 服务器 Mtp3 层的状态发生变化, 通常用于指示到某个 DPC 的路由是否可用
0x0004	保留	—	
0x0005	E_CHG_FaxPages	√	传真通道: 驱动程序完成一页传真的接收或发送
0x0006	E_PROC_FaxEnd	√	传真通道: 驱动程序完成传真的发送或接收任务
0x0007	E_CHG_PcmLinkStatus	√	数字中继线的同步状态发生变化
0x0008	E_CHG_LineVoltage	—	模拟电话线路上的电压发生变化
0x0009	E_RCV_CAS	—	SS1 通道: 对端交换机的 ABCD 信令码发生变化
0x000A	E_RCV_R2	—	SS1 通道: 收到对端交换机的 R2 信号
0x000B	E_PROC_WaitDTMF	√	WaitDTMF 任务完成, 任务通过函数 SsmSetWaitDtmf 、 SsmSetWaitDtmfEx 或 SsmSetWaitDtmfExA 提交
0x000C	E_CHG_RcvDTMF	√	DTMF 检测器: 收到一个 DTMF 字符
0x000D	E_PROC_SendDTMF	√	DTMF 发生器: 发送 DTMF 的任务完成, 发送 DTMF 任务由函数 SsmTxDtmf 启动
0x000E	E_PROC_SendFlash	√	发送闪断信号的任务完成
0x000F	E_PROC_PlayEnd	√	放音操作: 放音任务结束。放音任务可以由下列函数之一启动: ◊ SsmPlayFile ◊ SsmPlayIndexString ◊ SsmPlayIndexList ◊ SsmPlayFileList ◊ SsmPlayMem ◊ SsmPlayMemList ◊ SsmPlayMemBlock
0x0010	E_PROC_PlayFile	—	放音操作: 文件放音进程指示
0x0011	E_PROC_PlayFileList	—	放音操作: 驱动程序完成文件序列中一个文件的播放
0x0012	E_PROC_PlayMem	—	放音操作: 单个内存播放任务的进展指示
0x0013	E_PROC_RecordEnd	√	录音操作: 录音任务终止
0x0014	E_PROC_RecordFile	—	录音操作: 文件录音任务的进展指示
0x0015	E_PROC_RecordMem	—	录音操作: 内存录音任务的进展指示
0x0016	E_PROC_SendFSK	√	FSK 数据发送器完成全部数据的发送
0x0017	E_PROC_RcvFSK	√	RcvFSK 任务结束
0x0018	E_CHG_ChState	√	状态机: 通道状态发生变化
0x0019	E_PROC_AutoDial	√	状态机: AutoDial 任务有进展
0x001A	E_CHG_RemoteChBlock	—	TUP/ISUP 通道: 闭塞对端通道的操作完成
0x001B	E_CHG_RemotePCMBlock	—	TUP/ISUP 通道: 闭塞对端 PCM 的操作完成
0x001C	E_SYS_ActualPickup	—	模拟中继线通道: 摘机命令执行完毕
0x001D	E_CHG_RingFlag	—	模拟中继线通道/模拟电话线录音通道: 铃流信号的电平发生变化
0x001E	E_CHG_RingCount	√	模拟中继线通道: 铃流信号检测器中信号周期的计数器发生变化
0x001F	E_CHG_CIDExBuf	√	FSK 和 DTMF 的主叫号码信息扩展缓冲区: 主叫号码扩展缓冲区的长度发生变化
0x0020	E_CHG_RxPhoNumBuf	√	DTMF 检测器: 收到新的被叫号码
0x0021	E_CHG_PolarRvrsCount	—	模拟中继线通道: 在线路上检测到一次极性反转
0x0022	E_SYS_RemotePickup	—	模拟中继线通道: 增强的远端摘机检测器检测到被叫用户摘机
0x0023	E_CHG_FlashCount	√	坐席通道或者录音通道: 在电话机上检测到一次闪断操作
0x0024	E_CHG_HookState	√	坐席通道: 话机上检测到摘机或挂机动作
0x0025	E_CHG_ToneAnalyze	√	信号音检测器: 分析结果发生变化
0x0026	E_OverallEnergy	—	信号音检测器: 线路上的全频能量发生变化
0x0027	E_CHG_OvrlEnrgLevel	√	信号音检测器: 全频能量标识输出事件, 由 OvrlEnrgEventOut 配置项决定是否输出
0x0028	E_CHG_BusyTone	√	信号音检测器: 呼叫进程音检测器检测到的忙音周期的个数发生变化
0x0029	E_CHG_BusyToneEx	—	信号音检测器: 采用背靠背忙音检测算法检测到的忙音信号

0x002A	E_CHG_VocFxFlag	√	信号音检测器：单音频信号音的电平发生变化，通常用于传真信号音的检测
0x002B	E_CHG_ToneValue	—	信号音检测器：信号音的电平发生变化
0x002C	E_CHG_RingEchoToneTime	—	信号音检测器：回铃音计数器的值发生变化
0x002D	E_CHG_PeakFrq	—	信号音检测器：峰值频率发生变化
0x002E	E_SYS_BargeIn	√	Barge In 检测器：检测结果发生变化
0x002F	E_SYS_NoSound	√	信号音检测器：线路上保持静默
0x0030	E_SYS_TIMEOUT	√	全局定时器：由函数 SsmStartTimer 启动的定时器发生溢出
0x0031	E_CHG_SpyState	—	DTP 系列：监控电路的状态发生变化
0x0032	保留		
0x003c	E_CHG_CICRxPhoNumBuf	—	SS7 虚电路：收到新的被叫号码
0x003d	E_CHG_CICState	—	SS7 虚电路：电路状态发生变化
0x003e	E_PROC_CICAUTO_DIAL	—	SS7 虚电路：ShgAutoDial 任务有进展
0x003f	E_RCV_Ss7IsupUtuinf	—	SS7：收到 USR 消息事件
0x0040	E_CHG_Mtp2Status	√	SS7 信令链路：信令链路的状态发生变化
0x0041	E_RCV_DSTDChannel	√	DST 系列：D 通道事件
0x0042	E_RCV_Ss7SpyMsu	—	SS7：从 SS7 服务器收到新的监控消息（MSU）
0x0043	E_CHG_ToneDetector	√	信号音检测器：新检测模式结果输出事件
0x0044	E_CHG_ToneDetectorItem	—	信号音检测器：新检测模式信号音周期计数事件
0x0046	E_PROC_FaxDcnTag	√	传真通道：传真接收成功结束时，判断对方传真是否强制停止过
0x0047	E_CHG_AMD	√	信号音检测器，分析是否有真人摘机
0x0048	E_RCV_Ss7IsupCpg	—	SS7：收到 CPG 消息事件
0x0049	E_CHG_CbChStatus	—	大容量坐席：监控板卡通道与大容量坐席模块连接状态发生的变化
0x004a	E_RCV_SS7Mtp2Msu	√	SS7：收到一条 MTP2 MSU。（需要配置 AppHandleMtp2Msu=1 ）
0x0050	E_REFER_Status	—	SsmplInitiateTransfer 呼转后的状态
0x0051	E_CHG_SpyHangupInfo	—	DTP 系列：监控电路收到挂机事件
0x0052	E_CHG_CallBackRingCount	√	模拟中继线通道：回呼铃流信号检测器中信号周期的计数器发生变化
0x0053	E_CHG_RcvMF	—	ATP 系列：收到新的 MF 字符
0x0054	E_CHG_Pcm32LineState	√	PCM1280E：PCM32 通道的同步状态值或信号状态值变化
0x0055	E_RCV_SPY_CAS	—	DTP 系列：监控电路收到 CAS 信令
0x0056	E_CHG_RingEchoToneCnt	√	模拟中继通道：回铃音计数发生变化
0x0057	E_Ss7_L2ToL3_IND	√	SS7：指示 MTP2 变化以及收到 MTP2 MSU。（需要配置 AppHandleMtp2Msu=2 ）
0x0060	E_RCV_IPR_DChannel	√	IPR 系列：D 通道事件
0x0061	E_RCV_IPR_DONGLE_ADDED	√	IPR 系列：检测到 USB KEY
0x0062	E_RCV_IPR_DONGLE_REMOVED	√	IPR 系列：检测到 USB KEY 被移除
0x0063	E_RCV_IPR_NIC_LINKED	√	IPR 系列：检测到指定网卡（保留，暂时未使用）
0x0064	E_RCV_IPR_NIC_UNLINKED	√	IPR 系列：没有检测到指定网卡
0x0065	E_RCV_IPR_AUTH_OVERFLOW	√	IPR 系列：授权溢出
0x0066	E_RCV_IPR_MEDIA_SESSION_STARTED	√	IPR 系列：Session 开启
0x0067	E_RCV_IPR_MEDIA_SESSION_STOPED	√	IPR 系列：Session 关闭，当抛出事件时，驱动会自动停止转发 RTP。因此此时再调用函数 SsmIPRStopSendSession 会返回失败
0x0068	E_RCV_IPR_AUX_MEDIA_SESSION_STARTED	√	IPR 系列：辅助 Session 开启，此事件在分机呼叫分机时出现
0x0069	E_RCV_IPR_MEDIA_SESSION_FORWARDING	√	IPR 系列：Session 转发中
0x006a	E_RCV_IPR_MEDIA_SESSION_FORWARD_STOPED	√	IPR 系列：Session 转发停止
0x006b	E_RCV_IPR_STATION_ADDED	√	IPR 系列：检测到终端加入
0x006c	E_RCV_IPR_STATION_REMOVED	√	IPR 系列：检测到终端退出

	MOVED		
0x006d	E_IPR_LINK_REC_SLAVER_CONNECTED	√	IPR 系列: 检测到录音 Slaver 连接上来
0x006e	E_IPR_LINK_REC_SLAVER_DISCONNECTED	√	IPR 系列: 检测到录音 Slaver 连接断开
0x006f	E_IPR_SLAVER_INIT_CB	√	IPR 系列: 对录音 Slaver 初始化情况的反馈
0x0070	E_IPR_ACTIVE_SESSION_CB	√	IPR 系列: 对要求 IPRR 通道开启 Session 接收 (比如调用 SsmIPRAActiveSession) 的反馈
0x0071	E_IPR_DEACTIVE_SESSION_CB	√	IPR 系列: 对要求 IPRR 通道关闭 Session 接收 (比如调用 SsmIPRDeActiveSession) 的反馈
0x0072	E_IPR_START_REC_CB	√	IPR 系列: 对要求 IPRR 通道进行录音工作 (比如调用 SsmRecToFile 、 SsmRecToMem) 的反馈
0x0073	E_IPR_STOP_REC_CB	√	IPR 系列: 对要求 IPRR 通道停止录音工作 (比如调用 SsmStopRecToFile 、 SsmStopRecToMem) 的反馈
0x0074	E_IPR_PAUSE_REC_CB	√	IPR 系列: 对要求 IPRR 通道暂停录音工作 (比如调用 SsmPauseRecToFile) 的反馈
0x0075	E_IPR_RESTART_REC_CB	√	IPR 系列: 对要求 IPRR 通道重新开始录音工作 (比如调用 SsmRestartRecToFile) 的反馈
0x0076	E_IPR_START_SLAVER_CB	√	IPR 系列: 对要求打开 Slaver (比如调用 SsmIPRStartRecSlaver) 的反馈
0x0077	E_IPR_CLOSE_SLAVER_CB	√	IPR 系列: 对要求关闭 Slaver (比如调用 SsmIPRCloseRecSlaver) 的反馈
0x0078	E_IPR_RCV_DTMF	√	IPR 系列: 检测到带内 DTMF 或 RFC2833 形式的 DTMF
0x0079	E_IPR_ACTIVE_AND_REC_CB	√	对要求 IPRR 通道开启 Session 接收并开始录音 (比如调用 SsmIPRAActiveAndRecToFile) 的反馈
0x007a	E_IPR_DEACTIVE_AND_STOPREC_CB	√	对要求 IPRR 通道关闭 Session 接收并停止录音 (比如调用 SsmIPRDeActiveAndStopRecToFile) 的反馈
0x007b	E_RCV_IPA_DONGLE_ADDED	√	IPA 系列: 检测到 USB KEY
0x007c	E_RCV_IPA_DONGLE_REMOVED	√	IPA 系列: 检测到 USB KEY 被移除
0x007d	E_RCV_IPA_APPLICATION_PENDING	√	IPR 系列: 检测到 USB KEY 故障或者被拔除, 一段时间之后, 应用程序进入 pending 状态
0x007e	E_RCV_IPR_AUX_MEDIA_SESSION_STOPED	√	IPR 系列: 辅助 Session 关闭, 此事件在分机呼叫分机时出现
0x007f	E_BOARD_ICMP_CHANGE	√	SHN B 型卡: ICMP 结果改变
0x0080	E_RCV_IsdnSpyMsu	—	ISDN: 从 ISDN 服务器收到新的监控消息 (MSU)
0x0081	E_RCV_DecodeSs7Msu	—	SS7: 从 SS7 服务器收到新的解码消息
0x0082	E_CHG_RCV_SELCALL	√	ATP、SHT 系列: 接收到 Selcall Tone 的一个数据
0x0083	E_RCV_IsdnL2SpyMsu	—	DTP 卡收到 ISDN 二层原始消息, 由 SsmGetIsdnL2SpyMsu 函数获取消息内容
0x0084	E_CHG_AMD_TIME	√	AMD 状态检测事件
0x0085	E_REG_OPTIONS_RESPONSE	—	IP 卡系列: Option 消息响应事件
0x0088	E_REG_REQUEST	√	注册鉴权模式下应用程序接收到的注册请求
0x0089	E_REG_REGSTATUS	√	注册用户状态发生变化
0x008a	E_RCV_REFER	√	呼叫转移
0x008b	E_REG_NOTIFY_RESPONSE	—	IP 卡系列: Notify 消息的响应事件
0x008c	E_SIPSTACK_REGSTATUS	√	协议栈注册状态发生变化
0x008d	E_HOLDING_CALL	√	IP 通道 Hold 事件
0x008e	E_SIP_RESPONSE	√	SIP 消息响应事件
0x008f	E_RCV_IPR_IGMP_MESSAGE	√	IPR 系列: 用于 ROIP 声控录音时提供 IGMP 信息通知, 以便应用程序能对 IGMP 及时做出响应, 该事件通过配置项 IGMPEventEnable 控制输出
0x0090	E_SYS_Lost Interrupt	√	板卡出现中断丢失

➤ **nReference 和 dwParam**

参数 nReference 和 dwParam 的物理含义与参数 wEvent 的值有关，如下表所示：

事件类型	nReference	dwParam
E_PROC_Recognize	通道的逻辑编号	输出识别结果，取值范围为： 1: 取得识别结果(Accept) 2: 未听懂(Reject) 3: 未听清(Failed) 4: 未听到(Silence)
E_CHG_ISDNStatus	数字中继线的逻辑编号	不使用。函数 SsmISDNGetStatus 可用来获取具体的输出参数
E_RCV_Ss7Msu	不使用	不使用。函数 SsmGetSs7Msu 用于取出消息
E_CHG_Mtp3State	目的信令点(DPC)的编号	输出本端到指定 DPC 的路由上的 SS7 信令服务是否可用的信息，具体为： 1: 可用 0: 不可用
E_CHG_FaxPages	通道的逻辑编号	输出驱动程序完成发送或接收的页面总数
E_PROC_FaxEnd	通道的逻辑编号	输出传真发送或接收任务的进展情况，具体为： 0: 传真任务尚未完成，包括握手过程以及传真接收或发送过程 1: 传真任务已经完成，通道回到空闲状态 2: 传真接收或发送过程中发生错误，或者该任务被应用程序调用函数 SsmFaxStop 终止，通道回到空闲状态 3: 已经完成全部传真数据的发送或接收，开始与对端协商断开连接
E_PROC_FaxDcnTag	通道的逻辑编号	0: 传真过程完整 1: 传真在接收过程中因收到 DCN (结束信息) 而结束，可能是传真机强制停止引起的
E_CHG_PcmLinkStatus	数字中继线的逻辑编号	输出数字中继线的状态值。详细内容请参见函数 SsmGetPcmLinkStatus 中 pwPcmLinkStatus 参数的说明
E_CHG_LineVoltage	通道的逻辑编号	输出模拟电话线上的电压值，单位为伏特
E_RCV_CAS	通道的逻辑编号	输出收到的 ABCD 信令码的最新值，低 4 位比特有效：Bit3 Bit2 Bit1 Bit0: 对应于 ABCD 码 其余比特：全部为 0
E_RCV_R2	通道的逻辑编号	输出最新接收到的 R2 信号，取值范围为： 0: R2 信号消失 1~15: 前向 R2 信号的值 1~6: 后向 R2 信号的值
E_PROC_WaitDTMF	通道的逻辑编号	输出 WaitDTMF 任务的终止原因，具体取值为： 1: 任务超时 2: 接收到指定的结束字符 3: 接收到指定长度的 DTMF 字符串 函数 SsmChkWaitDtmf 可用于获取更多的信息
E_CHG_RcvDTMF	通道的逻辑编号	高 16 比特为收到的 DTMF 字符的总数，低 16 比特为最新收到的字符，相当于连续调用 SsmGetRxDtmfLen 和 SsmGetLastDtmf 函数
E_PROC_SendDTMF	通道的逻辑编号	=0: 完成缓冲区全部 DTMF 字符的发送 =1: 发送任务被应用程序终止
E_PROC_SendFlash	通道的逻辑编号	不使用
E_PROC_PlayEnd	通道的逻辑编号	输出放音任务终止的原因，具体为： 1: 全部语音数据播放完毕 2: 因收到 DTMF 字符而终止 3: 因检测到 Barge in 而终止 4: 因检测到对端用户的挂机动作而终止 5: 被应用程序终止 6: 文件播放任务被暂停 7: 因操作总线而终止

		<p>8: 文件播放任务因网络故障而终止 更多信息请参见函数 SsmCheckPlay 的说明</p>
E_PROC_PlayFile	通道的逻辑编号	<p>输出下列参数之一： ◆ 完成播放的百分比 ◆ 完成播放的时间长度 ◆ 完成播放的字节总数 ◆ 尚未完成播放的字节总数 具体输出哪一个，由 SsmSetEvent 函数决定，缺省值为输出放音时间</p>
E_PROC_PlayFileList	通道的逻辑编号	<p>正在播放的语音文件在文件序列中的索引值（从 0 开始编号）</p>
E_PROC_PlayMem	通道的逻辑编号	<p>输出下列参数之一： -1: 驱动程序的放音指针越过缓冲区的中间位置 -2: 驱动程序的放音指针越过缓冲区尾部后，返回到缓冲区头部 其它值：驱动程序的放音指针在缓冲区中的偏移量，单位为字节</p>
E_PROC_RecordEnd	通道的逻辑编号	<p>录音任务的终止原因，具体为： 1: 被应用程序终止 2: 因检测到 DTMF 字符而终止 3: 因检测到对端用户的挂机动作而终止 4: 因录制的数据到达指定长度或时间而终止 5: 文件录音被暂停 6: 将录音数据写入到文件失败 7: RTP 超时 8: 会话中修改 RTP 负载格式，并且新的负载格式不被支持</p>
E_PROC_RecordFile	通道的逻辑编号	<p>输出下列参数之一： ◆ 录音时间 ◆ 完成录音的字节总数 具体输出哪一个，由 SsmSetEvent 函数决定，缺省值为输出录音时间</p>
E_PROC_RecordMem	通道的逻辑编号	<p>输出下列参数之一： -1: 驱动程序内部的录音指针越过内存缓冲区的 1/2 -2: 驱动程序内部的录音指针越过内存缓冲区的尾部 其它值：录音指针偏移量</p>
E_PROC_SendFSK	通道的逻辑编号	<p>=0: 驱动程序完成全部 FSK 数据的发送 =1: 发送任务被函数 SsmStopSendFsk 终止</p>
E_PROC_RcvFSK	通道的逻辑编号	<p>FSK 数据接收器的停止原因： 1: 超时，接收失败 2: 收到指定的结束特征字节而结束 3: 收到指定长度 FSK 数据而结束 4: 收到指定格式的数据而结束 5: 应用程序调用函数 SsmStopRcvFSK 中断 应用程序得到本事件后，可以调用函数 SsmGetRcvFSK 取得收到的 FSK 数据</p>
E_CHG_ChState	通道的逻辑编号	<p>高 16 位比特表示通道的上一个状态值； 低 16 位比特表示通道的新状态值。 有关通道状态的取值及其描述请参见函数 SsmGetChState 的说明</p>
E_PROC_AutoDial	通道的逻辑编号	<p>AutoDial 任务进展值，参数的具体含义请参见函数 SsmChkAutoDial 的说明</p>
E_CHG_RemoteChBlock	通道的逻辑编号	<p>闭塞/解闭塞对端电路操作的进展情况，具体值为： 0: 已经成功解除由本端导致的闭塞 1: 已经成功闭塞对端 2: 正在等待对端交换机的闭塞证实信号 3: 正在等待对端交换机的解除闭塞证实信号</p>
E_CHG_RemotePCMBlock	数字中继线的逻辑编号	<p>闭塞/解闭塞对端数字中继线的进展情况，参数返回低 16 位为： 0: 已经成功解除由本端导致的闭塞 1: 已经成功闭塞对端</p>

		2: 正在等待对端交换机的闭塞证实信号 3: 正在等待对端交换机的解除闭塞证实信号 高 16 位为闭塞的模式。
E_SYS_ActualPickup	通道的逻辑编号	0 (保留, 未使用)
E_CHG_RingFlag	通道的逻辑编号	铃流信号的电平发生变化: Bit31: 铃流电平标志, 1: 有铃流; 0: 无铃流 Bit30~Bit0: 上一个铃流电平的保持时间, 单位: 毫秒
E_CHG_RingCount	通道的逻辑编号	铃流信号周期的个数
E_CHG_CIDExBuf	通道的逻辑编号	主叫号码扩展缓冲区中的字符总数。调用 SsmGetCallerIdEx 函数可以获取字符内容
E_CHG_RxPhoNumBuf	通道的逻辑编号	被叫号码缓冲区中的字符总数。调用 SsmGetPhoNumStr 函数可取得被叫号码的具体内容
E_CHG_PolarRvrsCount	通道的逻辑编号	驱动程序内部保存极性反转的计数器的值
E_SYS_RemotePickup	通道的逻辑编号	0 (保留, 未使用)
E_CHG_FlashCount	通道的逻辑编号	检测到的闪断的次数
E_CHG_HookState	通道的逻辑编号	用户对话机的动作: 0: 挂机 1: 摘机
E_CHG_ToneAnalyze	通道的逻辑编号	信号音分析器结果: 信号音分析器忙音结果: 高 16 比特: 如果使用普通信号音检测器, 高 16 比特表示呼叫进程音分析器的编号。0: 第一组; 1: 第二组; 2: 当低 16 比特返回 7、8 时; 3: 当低 16 比特返回 4、5、6 时。如果使用增强型信号音检测器, 高 16 比特一直返回 0。 低 16 比特: 检测到的信号音, 具体取值为: 1: 检测到拨号音 2: 检测到忙音 3: 检测到回铃音 4: 检测到回铃音后, 线路保持静默 5: 检测到无声 6: 检测到说话声, 用于自动拨号时检测被叫用户是否应答(摘机) 7: 检测到 F1 频率的信号音, F1 频率参数由函数 SsmSetVoiceFxPara 设置。用于自动拨号时检测被叫用户的应答类型 8: 检测到 F2 频率的信号音, F2 频率参数由函数 SsmSetVoiceFxPara 设置。用于自动拨号时检测被叫用户的应答类型 9: 检测到用户指定的信号音类型
E_OverallEnergy	通道的逻辑编号	信号音分析器全频能量值
E_CHG_AMD	通道的逻辑编号	真人摘机事件分析结果: 0: 检测到真人摘机 1: 检测到信号音 2: 检测到彩铃或提示音 3: 超时 4: 检测到信号音或提示音后线路无声 5: 检测到拨号后线路无声 6: 检测到忙音
E_CHG_OvrlEnrgLevel	通道的逻辑编号	信号音分析器全频能量标识值, 低 16 比特: Bit15 (信号音标志位): 0/1 无/有信号音 Bit14 (函数操作状态位): 0 保留 Bit13-0 (持续时间): 有/无信号音的持续时间
E_CHG_BusyTone	通道的逻辑编号	信号音检测器的忙音检测结果: 高 16 比特: 指示信号音分析器, 0 第一组, 1 第二组 低 16 比特: 忙音周期的个数
E_CHG_BusyToneEx	通道的逻辑编号	信号音检测器采用背靠背忙音检测算法检测到的忙音信号:

		0: 忙音信号消失 1: 检测到忙音信号
E_CHG_VocFxFlag	通道的逻辑编号	传真信号音的类型, 具体为: 1: 检测到 F1 2: 检测到 F2
E_CHG_ToneValue	通道的逻辑编号	信号音分析器: 信号音电平变化值。 低 16 比特有效: Bit15(信号音标志位) : 0/1 有/无信号音 Bit14(函数操作状态位): 0/1 操作成功/操作失败 Bit13-0(持续时间): 有/无信号音的持续时间
E_CHG_RingEchoToneTime	通道的逻辑编号	信号音分析器: 回铃音时间, 以一个回铃音周期变化, 单位秒
E_CHG_PeakFrq	通道的逻辑编号	信号音分析器: 峰值频率
E_SYS_BargeIn	通道的逻辑编号	BargeIn 检测器的检测结果发生变化: 0: Barge In 消失 1: 检测到 BargeIn
E_SYS_NoSound	通道的逻辑编号	不使用
E_SYS_TIMEOUT	定时器的编号	不使用
E_CHG_SpyState	SpyCic 的逻辑编号	高 16 位比特表示 SpyCic 的上一个状态值 低 16 位比特表示 SpyCic 的新状态值 有关 SpyCic 状态的取值及其描述请参见函数 SpyGetState 的说明
E_CHG_CICRxPhoNumBuf	虚电路编号	被叫号码长度。 应用程序得到该事件后, 要调用 ShgGetPhoNumStr 函数取得被叫号码
E_CHG_CICState	虚电路编号	电路状态值。详细内容参见 ShgGetChState 的说明
E_PROC_CICAUTOdial	虚电路编号	ShgAutoDial 任务进展值, 具体为: 1: DIAL_ECHOTONE = 2, 发送完被叫号码后检测到了回铃音 3: DIAL_BUSYTONE= 4, 被叫用户忙, 自动拨号结束 6: DIAL_VOICE= 7, 被叫用户摘机, 自动拨号结束 9: DIAL_NOANSWER= 10, 无人接听, 自动拨号失败 10: DIAL_FAILURE= 11, 自动拨号失败 11: DIAL_INVALID_PHONUM = 12, 空号, 自动拨号结束
E_RCV_DSTDChannel	通道的逻辑编号	子事件码, 如 DST_AUDIO_CHG、DST_MSG_CHG 等以“DST_”开头的子事件码, 具体事件码的含义请参见《SynCTI 程序员手册 -D 信道事件用户手册》
E_RCV_Ss7IsupUtuinf	应用程序通道号	用户至用户信息, 其所指向的空间中, 第一个字节是数据长度 (unsigned int), 随后是数据 (unsigned char[])
E_CHG_Mtp2Status	SS7 信令链路号	SS7 信令链路状态值, 具体为: 1: 业务中断 2: 初始定位 3: 已定位/准备 4: 已定位/未准备 5: 业务开通 6: 处理机故障
E_RCV_Ss7SpyMsu	不使用	不使用。函数 SsmGetSs7SpyMsu 用于取出消息
E_CHG_ToneDetector	通道的逻辑编号	低两位字节, 检测到的信号音类型 高两位字节, Item 编号, 为 ToneDetectorItem 项目编号。 更多信息请参考配置项 ToneDetectorItem
E_CHG_ToneDetectorItem	通道的逻辑编号	低两位字节, 周期计数 高两位字节, Item 编号, 为 ToneDetectorItem 项目编号。 更多信息请参考配置项 ToneDetectorItem
E_RCV_Ss7IsupCpg	通道的逻辑编号	CPG 消息的长度。
E_CHG_CbChStatus	通道的逻辑编号	0: 线路连接 1: 线路断开
E_RCV_SS7Mtp2Msu	SS7 链路编号	当前 MSU 消息的长度
E_CHG_RingEchoToneCnt	通道的逻辑编号	回铃音周期数
E_Ss7_L2ToL3_IND	SS7 链路编号	enum {

		<pre> MTP2_MSU_RX_IND =1, SS7_MTP2_IN_SVC, SS7_MTP2_OUT_SVC, SS7_MTP2_Rem_PR_OUT, SS7_MTP2_Rem_PR_OK, SS7_MTP2_RXD_BSNT, SS7_MTP2_RTVD_MSG, SS7_MTP2_RTVL_COMPL, SS7_MTP2_RTVL_NOT_POS, }; 当 dwParam 为 MTP2_MSU_RX_IND 时, 表示收到 MTP2 MSU 消息, 请使用 SSM_EVENT 事件结构获取消息内容。 </pre>
E_RCV_IPR_DChannel	不使用	子事件码, 如 DST_AUDIO_CHG、DST_MSG_CHG 等以“DST_”开头的子事件码, 具体事件码的含义请参见《SynCTI 程序员手册-D 信道事件用户手册》
E_RCV_IPR_DONGLEADED	不使用	不使用
E_RCV_IPR_DONGLE_REMOVED	不使用	不使用
E_RCV_IPR_NIC_LINKED	不使用	不使用
E_RCV_IPR_NIC_UNLINKED	不使用	不使用
E_RCV_IPR_AUTH_OVERFLOW	溢出类型: 1: Session 数量溢出 2: Station 数量溢出	高 16 位比特表示 Session 数量 低 16 位比特表示 Station 数量
E_REFER_Status	通道的逻辑编号	呼转后的状态值, 具体为: 0: 表示 Refer 空闲 1: 表示 Refer 在 Trying 状态, 即 the subscription is pending 2: 180RingBack 状态 3: 200ok, 3XX 状态 4: 4XX, 5XX, 6XX 失败状态 40: 表示通道为非 Use 状态
E_CHG_SpyHangupInfo	SpyCic 的逻辑编号	高 16 位比特表示发送拆线消息 SpyCic 所在的监控 PCM 号; 低 16 位比特表示 SpyCic 的挂机信息, 具体为: 0: 被叫方先挂机 1: 主叫方先挂机 有关 SpyCic 挂机信息及其描述请参见函数 SpyGetHangupInfo 的说明
E_CHG_CallBackRingCount	通道的逻辑编号	回呼铃流信号周期的个数
E_CHG_RcvMF	通道的逻辑编号	高 16 比特为收到的 MF 字符的总数, 低 16 比特为最新收到的字符
E_CHG_Pcm32LineState	通道的逻辑编号	使用低 16 位, 其中 bit0-bit7 为该通道同步状态值; bit8-bit15 为该通道信号状态值, 具体请参考函数: SsmGetPcm32LineState
E_RCV_SPY_CAS	通道的逻辑编号	收到的 ABCD 信令码, 低 4 位有效
E_RCV_IPR_MEDIA_SESSION_STARTED	通道的逻辑编号	SessionId, 具体的 Session 信息可以从 SSM_EVENT 的 pvBuffer 中获取, 可将 pvBuffer 转换为 pIPR_SessionInfo
E_RCV_IPR_MEDIA_SESSION_STOPED	通道的逻辑编号	SessionId, 具体的 Session 信息可以从 SSM_EVENT 的 pvBuffer 中获取, 可将 pvBuffer 转换为 pIPR_SessionInfo
E_RCV_IPR_AUX_MEDIA_SESSION_STARTED	通道的逻辑编号	SessionId, 具体的 Session 信息可以从 SSM_EVENT 的 pvBuffer 中获取, 可将 pvBuffer 转换为 pIPR_SessionInfo
E_RCV_IPR_MEDIA_SESSION_FORWARDING	通道的逻辑编号	SessionId
E_RCV_IPR_MEDIA_SESSION_FORWARD_STOPED	通道的逻辑编号	SessionId
E_RCV_IPR_STATIONADED	不使用	不使用

E_RCV_IPR_STATION_REMOVED	不使用	不使用
E_IPR_LINK_REC_SLAVER_CONNECTED	不使用	高 16 位为 SlaverId
E_IPR_LINK_REC_SLAVER_DISCONNECTED	不使用	高 16 位为 SlaverId
E_IPR_SLAVER_INIT_CB	不使用	高 16 位为 SlaverId, 低 16 位为 Slaver 返回值, 返回值具体表示信息: 0 成功, 1 未知, 2 超时, 3 报文数据异常, 4 该 SessionId 已处于活动中, 5 忙资源列表中找不到该 SessionId, 6 创建新的 Session 失败, 7 文件创建失败, 8 找不到活动的 SessionId, 9 该 SessionId 已处于录音中, 10 该 SessionId 已处于录音停止中, 11 该 SessionId 未处于录音状态, 12 该 SessionId 未处于录音到文件状态, 13 该 SessionId 未处于录音暂停状态, 14 要求录音的数据长度不够, 15 初始化 WAV 文件头出错, 16 该 Slaver 之前已经被分配过资源并开启, 17 申请资源过程失败, 18 该 Slaver 本来就没有被开启, 19 不支持的文件编码格式, 20 不支持的 RTP, 21 没有足够的空闲资源。
E_IPR_ACTIVE_SESSION_CB	通道的逻辑编号	表示的内容同 E_IPR_SLAVER_INIT_CB 的 dwParam
E_IPR_DEACTIVE_SESSION_CB	通道的逻辑编号	表示的内容同 E_IPR_SLAVER_INIT_CB 的 dwParam
E_IPR_START_REC_CB	通道的逻辑编号	表示的内容同 E_IPR_SLAVER_INIT_CB 的 dwParam
E_IPR_STOP_REC_CB	通道的逻辑编号	表示的内容同 E_IPR_SLAVER_INIT_CB 的 dwParam
E_IPR_PAUSE_REC_CB	通道的逻辑编号	表示的内容同 E_IPR_SLAVER_INIT_CB 的 dwParam
E_IPR_RESTART_REC_CB	通道的逻辑编号	表示的内容同 E_IPR_SLAVER_INIT_CB 的 dwParam
E_IPR_START_SLAVER_CB	通道的逻辑编号	表示的内容同 E_IPR_SLAVER_INIT_CB 的 dwParam
E_IPR_CLOSE_SLAVER_CB	通道的逻辑编号	表示的内容同 E_IPR_SLAVER_INIT_CB 的 dwParam
E_IPR_RCV_DTMF	通道的逻辑编号	高两个字节表示对应的 SlaverId, 第三个字节表示 DTMF 的方向, 0 表示 primary, 1 表示 secondary, 第四个字节表示 DTMF 信息
E_IPR_ACTIVE_AND_REC_CB	通道的逻辑编号	表示的内容同 E_IPR_SLAVER_INIT_CB 的 dwParam
E_IPR_DEACTIVE_AND_STOPREC_CB	通道的逻辑编号	表示的内容同 E_IPR_SLAVER_INIT_CB 的 dwParam
E_RCV_IPA_DONGLE_ADDRESS	不使用	不使用
E_RCV_IPA_DONGLE_REMOVED	不使用	不使用
E_RCV_IPA_APPLICATION_PENDING	不使用	不使用
E_RCV_IPR_AUX_MEDIA_SESSION_STOPED	通道的逻辑编号	SessionId, 具体的 Session 信息可以从 SSM_EVENT 的 pvBuffer 中获取, 可将 pvBuffer 转换为 pIPR_SessionInfo
E_BOARD_ICMP_CHANGE	板卡编号	ICMP 结果,: 0: ICMP 异常 1: ICMP 正常 2: A3 无法 ping 通板卡网口 3: A3 能够 ping 通板卡网口 4: shn537.bin 程序异常 5: shn537.bin 程序正常 说明: 本事件只有在 ICMP 事件状态改变或者 shn537.bin 有变化时才会抛出。0、1 是 shn537.bin 程序主动向外 ping, 需要配合 SsmCheckBoardICMP 使用。
E_RCV_IsdnSpyMsu	数字中继线的逻辑编号	当前 MSU 消息的长度
E_RCV_DecodeSs7Msu	不使用	不使用。函数 SsmGetDecodeSs7Msu 用于取出解码消息
E_CHG_RCV_SELCALL	通道的逻辑编号	接收到的 SELCALL TONE 数据
E_RCV_IsdnL2SpyMsu	数字中继线的逻辑	ISDN 二层中新收到的消息字节大小

	编号	
E_CHG_AMD_TIME	通道的逻辑编号	<p>Bit15~Bit0: 表示高低电平的持续计数。时间计算公式: 计数*16 (毫秒);</p> <p>Bit23~Bit16: 表示 AMD 当前状态:</p> <pre>enum { T1_WaitOff=0,//等待低电平状态,此状态需要一直保持低电平, 直到低电平持续时间大于 AMDTOff 才会进入 T1_CountOff 状态。 其间,如果检测到高电平就会重新开始计算低电平持续时间。 T1_CountOff=1,//检测低电平持续时间,须累加计算并加上 T1_WaitOff 中的低电平持续时间,直到出现高电平且持续时间累 加大于 AMDTOn 时,判断低电平持续时间和: 如果大于 AMDTimeA,进入 T2_CountOn 状态;如果小于 AMDTimeA,则 重新进入 T1_WaitOff 状态。 T2_CountOn=2,//检测高电平持续时间,累加计算并加上 T1_CountOff 中的高电平持续时间。如果高电平持续时间和超过 AMDTimeC 则抛出彩铃事件,状态重新进入 T1_WaitOff;如果检 测到低电平且持续时间大于 AMDTOff (若中间出现高电平则会重 新计算低电平持续时间),判断高电平持续时间和: 如果小于 AMDTimeB,进入 T1_CountOff 状态;如果大于 AMDTimeC,则 抛出彩铃事件,状态重新进入 T1_CountOff;如果在两者之间则 进入 T3_CountOff 状态,同时高电平持续时间清零。 T3_CountOff=3,//检测低电平持续时间,累加计算并加上 T2_CountOn 中的低电平持续时间,如果低电平持续时间和超过 AMDTimeD 则抛出真人摘机事件,AMD 检测结束;如果检测到 高电平且持续时间大于 AMDTOn (若中间出现低电平则会重新计 算高电平持续时间),判断低电平持续时间和: 如果大于 AMDTimeD 则抛出真人摘机事件,AMD 检测结束;否则抛出彩 铃事件,重新进入 T1_WaitOff 状态。 }; Bit31~Bit24: 表示当前的高低电平状态: 0: 表示抛出的是低电平持续时间。 1: 表示抛出的是高电平持续时间。</pre>
E_REG_OPTIONS_RESPONSE	不使用	Option 消息的响应值,从 SSM_EVENT 的 pvBuffer 中可以获取响应消息的服务器地址
E_REG_REQUEST	不使用	不使用
E_REG_REGSTATUS	不使用	注册索引号
E_RCV_REFER	通道的逻辑编号	不使用
E_REG_NOTIFY_RESPONSE	不使用	Notify 消息的响应值,从 SSM_EVENT 的 pvBuffer 中可以获取 Notify 消息的 To 字段内容。
E_SIPSTACK_REGSTATUS	注册 ID	<p>Bit15~Bit0: 表示注册状态:</p> <p>2: 注册成功;</p> <p>3: 注册失败;</p> <p>-1 (0xFFFF): 取消注册成功。</p> <p>Bit31~Bit16: 表示注册消息响应码,如果发送注册消息后,对端无响应为-1 (0xFFFF)</p>
E_HOLDING_CALL	通道的逻辑编号	1, 呼叫保持成功, 0, 取消呼叫保持成功
E_SIP_RESPONSE	通道的逻辑编号	SIP 响应消息码
E_SYS_Lost_Interrupt	不使用	板卡的 ID 编号 (对应配置工具中的 ID)

1.14.2.2 SSM_EVENT

SSM_EVENT 结构的声明为:

```
typedef struct _SSM_EVENT
{
}
```

```
WORD    wEventCode;      //事件编码
int     nReference;      //参考值
DWORD   dwParam;        //输出参数
DWORD   dwUser;         //用户参数
DWORD   dwSubReason;    //子原因值
DWORD   dwXtraInfo;    //扩展信息值
PVOID   pvBuffer;       //事件信息缓冲指针
DWORD   dwBufferLength; //事件信息缓冲大小
DWORD   dwDataLength;   //事件信息大小
DWORD   dwEventFlag;    //事件标记
DWORD   dwReserved1;    //保留参数 1
LONGLONG  lReserved2;    //保留参数 2
}SSM_EVENT, *PSSM_EVENT;
```

结构中的参数说明如下：

➤ **wEventCode、nReference、dwParam**

这 3 个参数的含义与 MESSAGE_INFO 结构中的 wEvent、nReference 和 dwParam 完全相同。

➤ **dwUser**

参数 **dwUser** 的含义与驱动程序输出事件的模式有关，如下表所示：

事件输出模式	dwUser 的含义
事件等待	当事件为用户自定义事件时，dwUser 的值等于调用 SsmPutUserEventA 时的参数 pEvent 的参数 dwUser 的值，否则没有意义
事件回调	透明传递应用程序在调用函数 SsmSetEvent 时参数 pEventSet 中的 dwUser 参数值

➤ **dwSubReason**

参数 dwSubReason 为子原因值，只有当参数 wEventCode 等于 [E_RCV_DSTDChannel](#) 或 [E_RCV_IPR_DChannel](#) 时才有意义。dwSubReason 的具体含义与驱动程序抛出事件时携带的 dwParam 参数有关，当输出 E_IPR_ACTIVE_AND_REC_CB 事件时，dwSubReason 参数存放 SESSIONID 的值，更多信息请参见《SynCTI 程序员手册-D 信道事件用户手册》中的相关内容。

➤ **dwXtraInfo**

参数 dwXtraInfo 为扩展信息，只有当参数 wEventCode 等于 [E_RCV_DSTDChannel](#)、[E_RCV_IPR_DChannel](#)、[E_RCV_IPR_STATION_ADDED](#)、[E_RCV_IPR_STATION_REMOVED](#) 时才有意义。dwXtraInfo 的具体含义与驱动程序抛出事件时携带的 dwParam 参数有关，其中在 E_RCV_IPR_STATION_ADDED、E_RCV_IPR_STATION_REMOVED 事件时高 16 位表示传输协议，低 16 位表示 StationId。[E_RCV_DSTDChannel](#)、[E_RCV_IPR_DChannel](#) 相关的更多信息请参见《SynCTI 程序员手册-D 信道事件用户手册》中的相关内容。

➤ **pvBuffer、dwBufferLength、dwDataLength**

pvBuffer、dwBufferLength、dwDataLength 只在参数 wEventCode 为 [E_RCV_DSTDChannel](#)、[E_RCV_IPR_DChannel](#)、[E_RCV_IPR_MEDIA_SESSION_STARTED](#)、[E_RCV_IPR_MEDIA_SESSION_STOPED](#)、[E_RCV_IPR_AUX_MEDIA_SESSION_STARTED](#) 时有意义。具体含义与驱动程序输出事件的模式有关，如下表所示：

事件输出模式	含义
--------	----

事件等待	<p>pvBuffer 为指向事件信息缓冲区的指针。应用程序在调用函数 SsmWaitForEventA 或 SsmGetEventA 时，应该为 pvBuffer 分配不小于 512 个字节的内存空间，而且应该设置 dwBufferLength 的值为实际分配的内存空间的大小（单位为字节）。函数 SsmWaitForEventA 或 SsmGetEventA 返回事件后，pvBuffer 中内容的含义与驱动程序抛出事件时携带的 dwParam 参数有关，更多信息请参见《SynCTI 程序员手册-D 信道事件用户手册》中的相关内容。dwDataLength 为 pvBuffer 中数据的实际长度（单位为字节）</p>
事件回调	<p>pvBuffer 为指向事件信息缓冲区的指针。驱动程序内部产生 E_RCV_DSTDChannel、E_RCV_IPR_DChannel、E_RCV_IPR_MEDIA_SESSION_STARTED、E_RCV_IPR_MEDIA_SESSION_STOPED、E_RCV_IPR_AUX_MEDIA_SESSION_STARTED 事件后，在调用应用程序预先注册的回调函数前，会自动为 pvBuffer 分配内存空间，因此，参数 dwBufferLength 没有意义。pvBuffer 中内容的含义与驱动程序抛出事件时携带的 dwParam 参数有关，其中当事件为 E_RCV_IPR_MEDIA_SESSION_STARTED、E_RCV_IPR_MEDIA_SESSION_STOPED、E_RCV_IPR_AUX_MEDIA_SESSION_STARTED 时，pvBuffer 里保存的是结构体 IPR_SessionInfo 的所有信息，可以将 pvBuffer 强制转化为 pIPR_SessionInfo 来方便信息获取，E_RCV_DSTDChannel、E_RCV_IPR_Channel 相关的更多信息请参见《SynCTI 程序员手册-D 信道事件用户手册》中的相关内容。dwDataLength 为 pvBuffer 中数据的实际长度（单位为字节）</p>

➤ dwEventFlag

dwEventFlag 为事件标记。dwEventFlag 的 Bit0 和 Bit2 有效，其余比特保留未用。Bit0 和 Bit2 的含义如下表所示：

比特	含义
Bit0	=0：本事件为驱动程序创建的事件 =1：本事件为应用程序创建的事件，即函数 SsmPutUserEventA 向驱动程序提交的自定义事件
Bit2	=0：事件信息没有被截断 =1：事件信息被截断 Bit2 只在事件输出模式为事件等待时有意义。只要符合以下两个条件之一 Bit2 将被驱动程序设置为 1： ◆ 应用程序调用函数 SsmPutUserEventA 时参数 pEventSet 中的 dwDataLength 参数值大于 256 ◆ 事件在输出之前临时保存在驱动程序中，应用程序调用函数 SsmWaitForEventA 时参数 pEventSet 中的 dwBufferLength 参数值小于驱动程序中该事件的参数 dwDataLength

➤ dwReserved1、IIReserved2

保留参数。

1.14.2.3 IPR_SessionInfo

IPR_SessionInfo 的结构体为

```
//session 信息
typedef struct IPR_SessionInfo
{

```

```
int nCallRef; //呼叫索引，  
int nStationId; //该 Session 对应的 StationId  
int nStationId2;//该 Session 对应的另一端 StationId  
DWORD dwSessionId;  
IPR_Addr PrimaryAddr; //会话一方的 IP 地址和端口  
int nPrimaryCodec; //会话一方 Session 使用的编解码 ID  
IPR_Addr SecondaryAddr; //会话另一方的 IP 地址和端口  
int nSecondaryCodec; //会话另一方 Session 使用的编解码 ID  
char szForwardingIp[32]; //Session 转发的 IP 地址，没转发时为空  
int nForwardingPPort; //会话一方 RTP 转发的端口，没转发时为-1  
int nForwardingSPort; //会话另一方 RTP 转发的端口，没转发时为-1  
  
}*pIPR_SessionInfo;
```

1.14.2.4 Reg_Info

Reg_Info 的结构体为

```
typedef struct tag_Reg_Info  
{  
    int    nRegIndex;//注册用户索引号  
    int    nRegState;//注册状态: 0 表示注册失败; 1 表示注册成功  
    int    nT0State;//用于定时器 T0 状态控制  
    char   szTelNo[MAX_SIPIP_ADDR_LENGTH];//注册用户名  
    char   szClientAddress[MAX_SIPIP_ADDR_LENGTH];//注册客户端 IP 地址  
    char   szClientPort[MAX_SIPIP_ADDR_LENGTH];//注册终端端口号  
    char   szUserID[MAX_SIPIP_ADDR_LENGTH];//认证用户名  
    int    nExpires; //注册有效期  
    int    nRegCountForAuthT0;//用于定时器 T0 计数  
    int    nRegCountForAuthT1;//用于定时器 T1 计数  
    char   szNonce[MAX_SIPIP_ADDR_LENGTH];  
    char   szDispalyName[MAX_SIPIP_ADDR_LENGTH];  
    int    nAuth;//鉴权参数, 取值为 0 和 1  
    char szPassword[MAX_SIPIP_ADDR_LENGTH];//用户密码  
};Reg_Info;
```

1.14.2.5 RegResp

RegRes 的 p 结构体为

```
typedef struct tag_RegResp
{
    int nAuth;//鉴权参数，取值为 0 和 1
    char szPassword[MAX_SIPIP_ADDR_LENGTH];//用户密码
    char szUsername[MAX_SIPIP_ADDR_LENGTH];//认证用户名（配用）和用户名保持一致
}RegResp;
```

1.14.3 事件的过滤

应用程序可以决定一个 SynCTI 驱动程序抛出或不抛出一个事件。这项功能是通过函数 [SsmSetEvent](#) 来实现的。

当应用程序以 wEvent=0xffff 作为参数调用函数 [SsmSetEvent](#) 时，驱动程序抛不抛出事件，取决于配置项 [DefaultEventOutput](#) 的设置，详细信息请参见该配置项的说明。

1.14.4 使用自定义事件

应用程序在运行过程中，也可能产生自己的事件，称为自定义事件。为了方便应用程序的编程，驱动程序提供了自定义事件的透明传输功能，即应用程序可以将自定义事件缓存到驱动程序内部的事件队列中，再通过驱动程序提供的事件获取函数进行处理。

函数 [SsmPutUserEvent](#) 和 [SsmPutUserEventA](#) 可以用来向驱动程序提交自定义事件。

1.14.5 常见编程平台下的编程要点

本小节介绍 SynCTI for Windows 驱动程序分别在 MS VC/C++、VB、C++ BUILDER、Delphi、PB6.5 以及其他编程平台下的编程要点，和 SynCTI for Linux 驱动程序的 C 语言编程要点。

1.14.5.1 MS VC/C++

SynCTI 驱动程序提供了在 MS VC++ 编程平台下需要用到下列文件：

- ◆ shpa3api.h： C/C++头文件。
- ◆ shp_a3.lib： C/C++输入库文件。

1.14.5.2 VB

在 Visual Basic 6.0 中，对动态链接库 SHP_A3.DLL 的调用是通过 Shpa3api.bas（由驱动程序提供）来声明的，请参见 Shpa3api.bas。

在本质上，采用 VB 编写的应用程序也是通过调用 shp_a3.dll 来实现对 Synway 板卡硬件的控制，因此，上一节中有关在 VC 环境下编程的说明，同样适用于 VB，请仔细阅读上一节。在开发应用程序时，需要经常参考 C 语言的示范程序。

在 VB 6.0 中，需要首先对 DLL 中的函数进行声明，然后才能调用相关函数。如何声明 DLL 中的函数，请参考 VB 中的 HELP 文件“Declare statement”。

当 shp_a3.dll 中有新的函数未能包含在 VB 声明文件 Shpa3api.bas 中时，可以参考该函数在 C 语言中的声明，来对需要在 VB 下调用的函数进行声明。例如，在 Shpa3api.h 中声明的一个函数：

```
int WINAPI SsmPlayFile(int ch, LPSTR pszFileName, int nFormat, DWORD dwStartPos, DWORD dwLen);
```

在 Shpa3api.bas 中声明为：

```
Declare Function SsmPlayFile Lib "shp_a3.dll" (ByVal ch As Long, ByVal pszFileName As String,
                                                ByVal nFormat as Long, ByVal dwStartPos as Long,
                                                ByVal dwLen As Long) As Long
```

由于 VC 采用 32 位方式进行编译，类型 BOOL 为 32 位、WORD 为 16 位、DWORD 为 32 位，因此，在 VB 中，相应地使用 Long (32 位)、 Integer (16 位)、 Long (32 位)。另外，需要注意的是，所有参数和返回值的字节长度要相等。

同在 9x/NT 下用 C 语言编程一样，VB 对板卡的操作也包括三部分：

- 初始化代码，包括初始化驱动程序、获取通道总数、通道类型等，在 Form_Load 中运行。
- 事务处理代码，在 TIMER 中运行，或放在 Form_Activate 函数里，具体用法请参见相关演示程序。
- 卸载代码，包括卸载驱动程序、关闭板卡等，在 Form_Unload 中运行。

注意：

- Form_Activate 是一个死循环，如果将处理部分放在这里，要调用 DoEvents，以便让其它程序能正常工作。
- 由于 VB 是解释性语言，在调用 DLL 时，如果某函数在 C 语言中的参数为字符串，并且在函数调用完毕后，需要通过该参数返回数据，如 SsmGetLastErrMsg、SsmGetDtmfDStr 等函数，相关参数必须预先分配空间。例如：

```
Dim ErrStr As String * 200
SsmGetLastErrMsg( ErrStr)
```
- 在 Windows NT 环境下，如果将初始化代码和主处理程序代码分别放在 Form_Load、Timer 中，必须保证在成功调用 SsmStartCti 后再执行 Timer 中的代码。具体方法：使用一个全局变量，在 Form_Load 的尾部将其置为 TRUE；在 Timer 中根据该变量的值来决定是否开始工作。也可以在 Form_Load 的开始部分调用 Timer.Enable=False，到最后调用 Timer.Enable = True；在 Form_Unload 时，先调用 Timer.Enable=False，再调用 SsmCloseCti。
- VB 6.0 本身不支持多线程，因而如果使用事件回调方式进行编程的话，部分 VB 6.0 自带的函数不能很好的工作在回调函数中（已知的有 Right, Left 等函数），并且 VB 6.0 的 IDE 不能在回调函数内设置断点，给调试带来极大的麻烦。建议 VB 6.0 避免使用事件回调方式编程，推荐使用事件等待方式编程。

1.14.5.3 C++ BUILDER

驱动程序提供的 Shpa3api.h 可以直接使用。

shp_a3.lib 是在 VC 环境编译的，不能直接用于 C++ BUILDER 工程文件的连接。C++ BUILDER 提供了一个从 DLL 中提取输入库 LIB 的工具软件 IMPLIB.EXE，可用该软件生成一个 LIB 用于开发。IMPLIB.EXE 的用法，请参考其随机的帮助文件。IMPLIB.EXE 的使用方法：

```
cd \windows\system
implib shp_a3.lib shp_a3.dll
```

将生成的 shp_a3.lib 拷贝到应用程序的开发目录下即可。

注意事项：

- 如果硬盘上装有两套 Borland 公司的 C 语言开发环境，要保证使用 C++ BUILDER 所附带的 IMPLIB.EXE。否则会发生错误！

1.14.5.4 Delphi

在 Delphi 编译环境下，对 shp_a3.dll 的调用是通过 Shpa3api.pas 来声明的，详细情况请参见 Shpa3api.pas。

与 VB6.0 相同，在 Delphi 中也需要首先对 DLL 中的函数进行声明，然后才能对该函数进行调用。关于如何声明 DLL 中的函数，请参考 Delphi 提供的 HELP 文件“external declarations”。如果 shp_a3.dll 中有某个函数没有包含在文件 Shpa3api.pas 中，可以参考该函数在 C 语言中的声明，自己来声明 Delphi 下的函数调用。例如，在 Shpa3api.h 中有下面的函数：

```
int WINAPI SsmPlayFile(int ch, LPSTR pszFileName, int nFormat, DWORD dwStartPos, DWORD dwLen);
```

在 Delphi 中可以声明为：

```
function SsmPlayFile (ch: Integer; pszFileName: PCHAR; nFormat:Integer; dwStartPos:DWORD;dwLen:DWORD ) : Integer; stdcall; external 'Shp_a3.dll'.
```

注意：

- 声明的最后必须加 stdcall; external 'Shp_a3.dll';
- C 语言中的 char* 应声明为 pchar。

1.14.5.5 PB6.5

在 Power Builder 编译环境下，对 shp_a3.dll 的调用是通过在“Decalre”菜单中的“Global External Function”来声明的。如果 shp_a3.dll 中有某个函数没有包含在声明中时，可以参考该函数在 C 语言中的声明，自行声明在 PB6.5 下的函数调用。例如，在 Shpa3api.h 中的函数声明为：

```
int WINAPI SsmPlayFile(int ch, LPSTR pszFileName, int nFormat, DWORD dwStartPos, DWORD dwLen);
```

在 Power Builder 环境下：

```
Function Long SsmPlayFile (Long ch, String pszFileName, Long nFormat, UnsignedLong dwStartPos, UnsignedLong dwLen) Library "shp_a3.dll"
```

注意： PB6.5 编译环境下的参数类型与 VB 相拟，请参阅“VB 编程平台编程要点”小节。

1.14.5.6 其它

在 Windows 95/NT 的 32 位编程环境中，许多编程语言都可以用各种方式来调用 DLL 动态库，因此，同样可以实现对动态库 SHP_A3.DLL 的调用。由于编程语言种类繁多，本手册无法一一描述。请仔细阅读你所用的编程语言的说明。

1.14.5.7 Linux 操作系统

C/C++ 应用程序需要用到下列文件（由驱动程序提供）：

- shpa3api.h：C/C++ 头文件
- libshpa3.so：驱动程序调用接口库

在 Linux 下用 C / C++ 语言调用 SynCTI 驱动程序时，一般包括三部分：

- 初始化代码，包括初始化卡、得到通道数和通道类型等等，在应用程序启动时调用一次。
- 事务处理代码，向驱动程序发布命令，对驱动程序抛出的事件进行处理等。具体用法请参见示例程序。
- 结束部分，包括释放语音卡、释放驱动程序，在应用程序退出时调用。

1.14.5.8 字符串的使用

不少高级语言的字符串使用方式和 C 语言的有很大差异，因此在调用驱动中有字符串参数的函数时，请注意遵循 C 语言的规范。

对于输出型的字符串参数，如 SsmGetDtmfStr 中的第二个参数，应用程序应该事先分配储存空间。许多高级语言能自动管理字符串的分配空间，但是对于跨语言平台调用的时候，因其不能预知对方的需求，仍然需要手工分配空间。

VB6.0 可以如下使用：

```
Dim DtmfStr As String*200
SsmGetDtmfStr(Ch, DtmfStr)
```

VB.Net 可以如下使用：

```
Dim DtmfStr As New String(Chr(0), 200)
SsmGetDtmfStr(Ch, DtmfStr)
```

C#可以如下使用：

```
String DtmfStr = new String((Char)0, 200);
SsmGetDtmfStr(Ch, DtmfStr);
```

对于输入型的字符串参数，如 SsmStartCti 中的两个参数，因为我方 DLL 不会修改其内容，所以无需显式的分配空间，但是需要注意的是必须在字符串尾部添加'\0'字符，以符合 C 语言的习惯。

VB.net 可以如下使用：

```
Dim ShConfig as String
Dim ShIndex As String
ShConfig = My.Application.Info.DirectoryPath & "\ShConfig.ini" & Chr(0)
ShIndex = My.Application.Info.DirectoryPath & "\ShIndex.ini" & Chr(0)
SsmStartCti(ShConfig, ShIndex)
```

1.14.6 配套演示程序（CTI 系列）

为便于开发人员迅速理解一些函数或功能的使用方法，Synway 提供了大量的演示程序，如下表所示：

名称	描述	编程平台					
		Windows					Linux
		VB	VC	C#	PB	DELPHI	gcc
Call	演示如何处理电话呼入	√	√	—	√	√	√
Dial	演示如何实现电话呼出	√	√	—	√	√	√
Conference Demo	演示电话会议的编程方法，适用于坐席通道、模拟外线通道、数字中继通道。更多内容请参见 Demo 文件夹下的 DemoConferenceUserManual.pdf (英文) 或 DemoConferenceUserManual_cn.pdf (中文)	—	√	—	—	—	—
Fax	演示如何用 Synway 板卡实现一个简易的传真服务器。详见 Demo 文件夹下的 DemoFAXUserManual.pdf (英文) 或 DemoFAXUserManual_cn.pdf (中文)	√	√	√	—	√	√
Pbx	用 Synway 板卡模拟一台小型交换机。详见 DemoPbxUserManual.pdf (英文) 或 DemoPbxUserManual_cn.pdf (中文)	—	√	√	—	—	—
	数字中继卡调试平台。用于调试基于数字中继卡的应用程序，具有简单的交换机功能	—	√	—	—	—	—
FSK	演示如何利用 Synway 板卡的 FSK 数据传输功能实现接收	—	√	—	—	—	—

	和发送数据					
Ss7Server	实现 SS7 信令服务器功能的演示程序	—	√	—	—	—
Test	演示 API 接口函数的使用方法				—	√
多线程VC	演示如何使用多线程方式对 Synway 板卡进行编程	—	√	—	—	—
事件驱动 demo	演示如何使用 SynCTI 驱动程序抛出的事件进行编程	—	√	—	—	—

1.14.7 配套演示程序（REC 系列）

为便于开发人员迅速理解一些函数或功能的使用方法，Synway 提供了大量的演示程序，如下表所示：

演示程序名称	描述	编程平台			
		Windows		Linux	
		VB	VC	C#	gcc
Recorder-ATP	演示如何使用 ATP 系列录音卡实现对模拟电话线路进行侦听和录音。详细说明请查阅 DemoATPUserManual.pdf (英文) 或 DemoATPUserManual_cn.pdf (中文)	—	√	√	√
Recorder-DST	演示如何使用 DST 系列板卡对数字电话用户线进行高阻并线录音、监听等功能。详见 DemoDSTUserManual.pdf (英文) 或 DemoDSTUserManual_cn.pdf (中文)	√	√	—	—
Recorder-DTP	演示如何使用 DTP 系列板卡实现对数字中继线 (E1) 进行监控录音的功能	√	√	—	—
RecPlay UseMemBlock	演示如何使用乒乓内存录音方式	—	√	—	—

1.14.8 使用驱动程序提供的定时器

驱动程序向应用程序提供了 2 种方式的定时器：全局定时器和通道定时器。

- ◆ 全局定时器

驱动程序内部设置了 128 个系统定时器，使用 Syn 板卡的硬件中断信号作为计时脉冲，精度优于操作系统提供的定时器。函数 [SsmStartTimer](#) 用于向驱动程序申请一个定时器，并设置其工作参数。每当定时器发生溢出时，驱动程序会向应用程序抛出 [E_SYS_TIMEOUT](#) 事件。函数 [SsmStopTimer](#) 用于终止一个定时器。

- ◆ 通道定时器

通道定时器适用于轮询方式的编程模式。通道定时器启动后，驱动程序只是将当前时刻保存到变量中，并不会在驱动程序内部判断定时器是否溢出。只有当应用程序调用 [ElapseTime](#) 函数时，驱动程序才会返回定时器的保持时间。驱动程序为每个通道分配了 10 个定时器，通过函数 [StartTimer](#) 启动。

1.14.9 使用驱动程序提供的调试功能

SynCTI 驱动程序带有调试功能，通常用于开发阶段、测试阶段及试运行的应用系统，具有 API 调试功能和事件消息调试功能。当应用程序调用 API 函数发生错误，或者事件消息出错时，或驱动程序内部发生某种差错时，如果相应的调试信息输出开关为开启状态，驱动程序会输出错误信息，以帮助开发人员迅速确定出错位置。

需要调试应用程序时，可使用下述配置项：

配置项	描述
ApiLogEnable	设置是否输出应用程序调用 API 函数的调试信息
ApiLogSetEventRange	在设置 ApiLogEnable 的条件下，设置事件输出写 LOG 的事件范围
ApiLogSetChRange	在设置 ApiLogEnable 的条件下，设置事件输出写 LOG 的通道范围

如果应用程序采用轮询方式，需要在定时器中不停地调用一些查询函数，从而导致大量的函数调用信息输出。下列配置项可以禁止一些常用的查询类型的函数输出调用信息：

配置项	描述
<u>Mask_SsmGetNoSoundTime</u>	控制是否输出 <u>SsmGetNoSoundTime</u> 函数的调用信息
<u>Mask_SsmGetChStateKeepTime</u>	控制是否输出 <u>SsmGetChStateKeepTime</u> 函数的调用信息
<u>Mask_SsmGetPlayedTime</u>	控制是否输出 <u>SsmGetPlayedTime</u> 函数的调用信息
<u>Mask_SsmGetPlayedPercentage</u>	控制是否输出 <u>SsmGetPlayedPercentage</u> 函数的调用信息
<u>Mask_SsmGetPlayOffset</u>	控制是否输出 <u>SsmGetPlayOffset</u> 函数的调用信息
<u>Mask_SsmGetRecTime</u>	控制是否输出 <u>SsmGetRecTime</u> 函数的调用信息
<u>Mask_SsmGetRecOffset</u>	控制是否输出 <u>SsmGetRecOffset</u> 函数的调用信息
<u>Mask_SsmGetChState</u>	控制是否输出 <u>SsmGetChState</u> 函数的调用信息

驱动程序输出的 API 调试信息需要借助于第三方的工具软件 Debugview.exe 来显示和观察。该工具软件支持 Windows Me、Windows 2000、Whistler Beta 1、Beta 2、以及 RC 1。

DebugView.exe 是一款绿色软件，不需要安装，直接将下载的 ZIP 文件解压缩后即可直接运行。Debugview.exe 可以从以下站点免费下载：<http://technet.microsoft.com/en-us/sysinternals/bb896647.aspx#top>。

1.15 SHT 系列板卡（CTI 系列）

1.15.1 SHT 系列板卡简介

SHT 系列板卡支持的功能如下表所示：

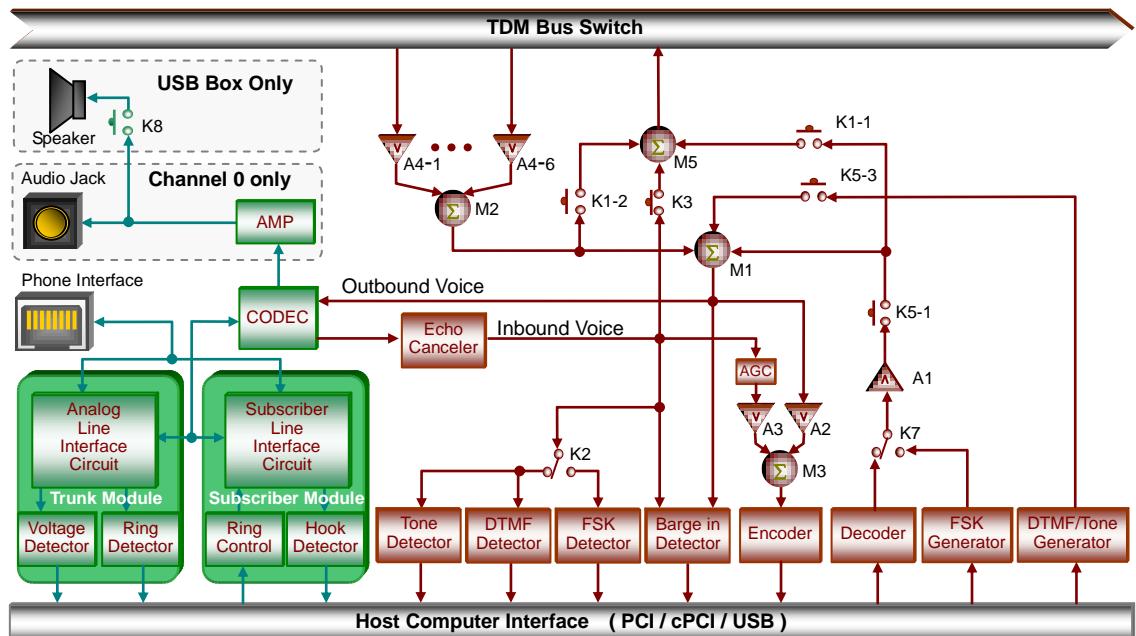
Board Model	Form Factor	Max. Ports	Voice Processing Capabilities	Conferencing Capabilities	Voltage Detector	Audio Socket	FSK	TDM Bus	Max Fax Resource
SHT-2A/USB	USB	2	√	√	√	√	√	N/A	N/A
SHT-4A/USB	USB	4	√	√	√	√	√	N/A	N/A
SHT-2B/USB	USB	2	√	√	√	√	√	N/A	N/A
SHT-4B/USB	USB	4	√	√	√	√	√	N/A	N/A
SHT-8B/PCI	PCI	8	√	√	√	√	√	N/A	N/A
SHT-8B/PCI/FAX	PCI	8	√	√	√	√	√	N/A	4
SHT-8C/PCI/FAX	PCI	8	√	√	√	√	√	N/A	4
SHT-8C/PCI/EC	PCI	8	√	√	√	√	√	N/A	N/A
SHT-16C-CT/PCI/EC	PCI	16	√	√	√	√	√	H'100	N/A
SHT-16D-CT/PCIe	PCIe	16	√	√	√	√	√	H.100	N/A*
SHT-16A-CT/PCI	PCI	16	√	√	√	√	√	H.100	N/A
SHT-16B-CT/PCI	PCI	16	√	√	√	√	√	H.100	N/A
SHT-16B-CT/PCI/FAX	PCI	16	√	√	√	√	√	H.100	4
SHT-16C-CT/PCI/FAX	PCI	16	√	√	√	√	√	H.100	4
SHT-16B-CT/PCI/MP3	PCI	16	√	√	√	√	√	H.100	N/A
SHT-120A-CT/PCI	PCI	120	√	√	N/A	N/A	N/A	N/A	N/A
SHT-16B-CT/cPCI	cPCI	16	√	√	√	√	√	H.110	N/A
SHT-16B-CT/cPCI/MP3	cPCI	16	√	√	√	√	√	H.110	N/A
SHT-16B-CT/cPCI/FAX	cPCI	16	√	√	√	√	√	H.110	8
SHT-120A-CT/cPCI	cPCI	120	√	√	N/A	N/A	N/A	N/A	N/A

注意：型号为 **SHT-16B-CT/PCI/FAX** 和 **SHT-16B-CT/PCI/MP3** 的板卡具有不同的硬件结构，但由于历史的原因，这 2 个型号的板卡上固件中写入的板卡 ID 号相同，因此，需要对配置项 DSP3WORKMODE 进行设置，以便使驱动程序能够正确区分这 2 种型号的板卡。

如果是 **SHT-16D-CT/PCIe** 型号的板卡，插上一个 **F021** 模块，可支持 4 路 V.17 传真算法。

1.15.2 SHT 系列板卡的工作原理框图

SHT 系列板卡的工作原理图如下图所示：



原理框图中的模块及符号说明：

名称	描述
M1	去话混音器
K7	语音播放/FSK数据发送器的切换开关，由驱动程序自动控制。当应用程序调用播放函数时，K7指向语音编码器模块 Decoder；当应用程序调用 SsmStartSendFSK 函数启动发送 FSK 数据时，K7 执行 FSK Generator。有关语音播放的函数请参见 语音播放 部分内容
A1	放音的音量调节器，可以通过配置项 DefaultPlayVolume 设置，也可通过函数 SsmSetPlayVolume 或 SsmSetPlayGain 进行调节，缺省的音量设置为 0
K5-1	控制放音操作的开关，由驱动程序自动控制通常处于断开状态，只有当应用程序调用播放函数时，驱动程序才会自动闭合 K5-1 开关；当放音任务结束时，驱动程序自动断开 K5-1。有关语音播放的函数请参见 语音播放 部分内容
K5-3	信号音发生器 、 DTMF发生器 的信号输出控制开关，由驱动程序自动控制，通常处于断开状态，只有当应用程序调用 SsmTxDtmf 函数启动 DTMF 发生器，或者调用 SsmSendTone / SsmSendToneEx 函数启动信号音发生器后，驱动程序才会自动闭合 K5-3 开关；发生器的任务终止后，K5-3 自动断开
M2	下总线信号混音器
A4-1 ... A4-6	下总线信号进入 M2 前的音量调节器，可以通过函数 SsmSetListenVlmInConf 进行设置，缺省值为 0DB。另外，也可以通过配置项 DefaultSpeakVolume 或下列函数对某个将要下总线过来的通道音量进行设置： ◆ SsmTalkWithEx ◆ SsmListenToEx / SsmLinkFromEx / SsmLinkFromAllCh
M3	录音混音器。M3 的输出进入编码器（Encoder）模块，然后形成录音数据
A3	来话信号进入录音混音器 M3 前的音量调节器，可以通过函数 SsmSetRecVolume 进行设置，也可以通过配置项 DefaultRecordVolume 设置，缺省值为 0
A2	去话信号进入录音混音器 M3 前的音量调节器，可以通过函数 SsmSetRecMixer 进行设置，也可以通过配置项 DefaultRecordMixerVolume 设置，缺省值 -7，关闭 A2
M5	上总线混音器。用于将放音数据、下总线信号混音器 M2 的输出、来话信号进行混音，混音结果上行到 TDM 交换总线。板卡初始化时，会将 M5 的输出上行到 TDM 总线的某个时隙（记为 ts）上，函数 SsmLinkToBus 可以用来改变 ts
K3	来话信号是否进入上总线混音器 M5 的控制开关，可通过函数 SsmSetFlag （携带 F_InVoiceToBus 参数）或配置项 InVoiceToBus 进行设置，缺省值为闭合
K1-2	M2 的输出信号是否重新回到 TDM 交换总线的控制开关，可以通过函数 SsmSetFlag （携带 F_MixerResToBus 参数）进行设置，缺省值为断开
K1-1	控制放音数据是否进入混音器 M5，进而上行到 TDM 交换总线上的开关，通常为断开状态，可以通过函数 SsmSetPlayDest 进行控制
K2	FSK 检测器与其它检测器之间的切换开关，通常指向其它的检测器，只有在开启 FSK 检测器期间才会指向 FSK 检测器

➤ 放音操作

应用程序调用播放函数后，驱动程序自动将 K7 指向语音解码器 Decoder，K5-1 闭合。解码后的语音数据经过音量调节器 A1 进行音量调节后，可以有 2 个去向：

- ◆ 进入 M1 混音器，与其它信号源混音后形成最终的去话信号；如果是 0 通道，去话信号会同时送到板载扬声器端口。
- ◆ 在 K1-1 控制下进入混音器 M5，与其它信号源混音后，形成上总线的信号源。

➤ 录音操作的信号源选择范例

灵活应用上述录音控制开关和音量调节器可以实现很多特殊的应用。下面范例中，ch1 表示通道 1，ch2 表示通道 2。

- ◆ 范例 1：只录制 ch1 的来话信号，使用缺省音量。实现代码为：

```
SsmSetRecVolume(ch1, 0);           // 开启 A3, 设置增益为 0DB
SsmSetRecMixer(ch1, FALSE, 0);    // 关闭 A2
SsmRecToFile(ch1, .....);        // 启动文件录音任务.....
```

- ◆ 范例 2：ch1 与 ch2 建立了双向通话后，在 ch1 上同时录制 ch1 的来话信号（音量提升 6DB）+ 总线混音器 M2 输出信号（音量衰减 3DB）。实现代码为：

```
SsmTalkWith(ch1, ch2);          // ch1 与 ch2 通过 TDM 交换总线建立双向连接
SsmSetRecVolume(ch1, 2);         // 开启 A3, 设置增益为 2*3DB=6DB
SsmSetRecMixer(ch1, TRUE, -1);   // 开启 A2, 设置增益为 -1*3DB=-3DB
SsmRecToFile(ch1, .....);       // 启动文件录音任务.....
```

- ◆ 范例 3：ch1 加入了一个电话会议室（假设会议室编号为 n），在 ch1 上播放背景音乐，同时通过 ch1 对电话会议进行录音。所有信号源均为正常音量，即增益为 0。实现代码为：

```
SsmJoinConfGroup(n, ch1, .....); // ch1 加入会议室
SsmSetPlayDest(int ch, 1);      // 闭合 K1-1, 将放音数据送到总线上
SsmPlayFile(ch1, .....);        // ch1 启动放音任务, 播放背景音乐
SsmSetRecVolume(ch1, 0);         // 开启 A3, 设置增益为 0DB
SsmSetRecMixer(ch1, TRUE, 0);    // 开启 A2, 设置增益为 0DB
SsmRecToFile(ch1, .....);       // 启动文件录音任务.....
```

➤ 板载扬声器输出端口

SHT 系列板卡的 0 通道具有模拟音频放大电路和板载扬声器输出端口，可直接连接到外置的扬声器或耳机，用来对语音信号进行实时监听。其主要用途为：

- ◆ 将某个通道的来话信号经 TDM 交换总线直接送到扬声器进行播放。实现此功能的函数为 [SsmListenTo](#) 或 [SsmListenToEx](#)。
- ◆ 调用语音播放的函数将应用程序录制的语音文件直接送到板载扬声器进行播放。有关语音播放函数的详细内容请参见本章中[语音播放](#)部分。

原理框图中的 AMP 模块是模拟音频放大电路的功率放大器，它的增益可以通过函数 [SsmSetPowerAmpVlm](#) 设置。

USB 总线的语音盒（型号中带“/USB”标记）的 0 通道除了在板卡上安装了扬声器插座外，还有一个板载扬声器，可以将声音字节播放出来。开关 K8 用于选择将 0 通道的输出信号是送到扬声器插座，还是板载扬声器。K8 可以通过函数 [SsmSetLine0OutTo](#) 或配置项 [USBLine0Output](#) 进行设置，缺省为输出到扬声器插座。

注意：将语音信号送到 0 通道上板载扬声器的同时，也会作为去话信号送到线路上。

1.15.3 模拟中继线通道

1.15.3.1 在模拟线路上产生闪断信号

闪断信号是指电话机进入通话后，通过快速拍打叉簧而产生的短时挂机信号，通常用于与交换机配合实现转分机等操作。在话机上拍下叉簧的时间应该足够短，以免交换机将闪断信号误判为挂机信号。

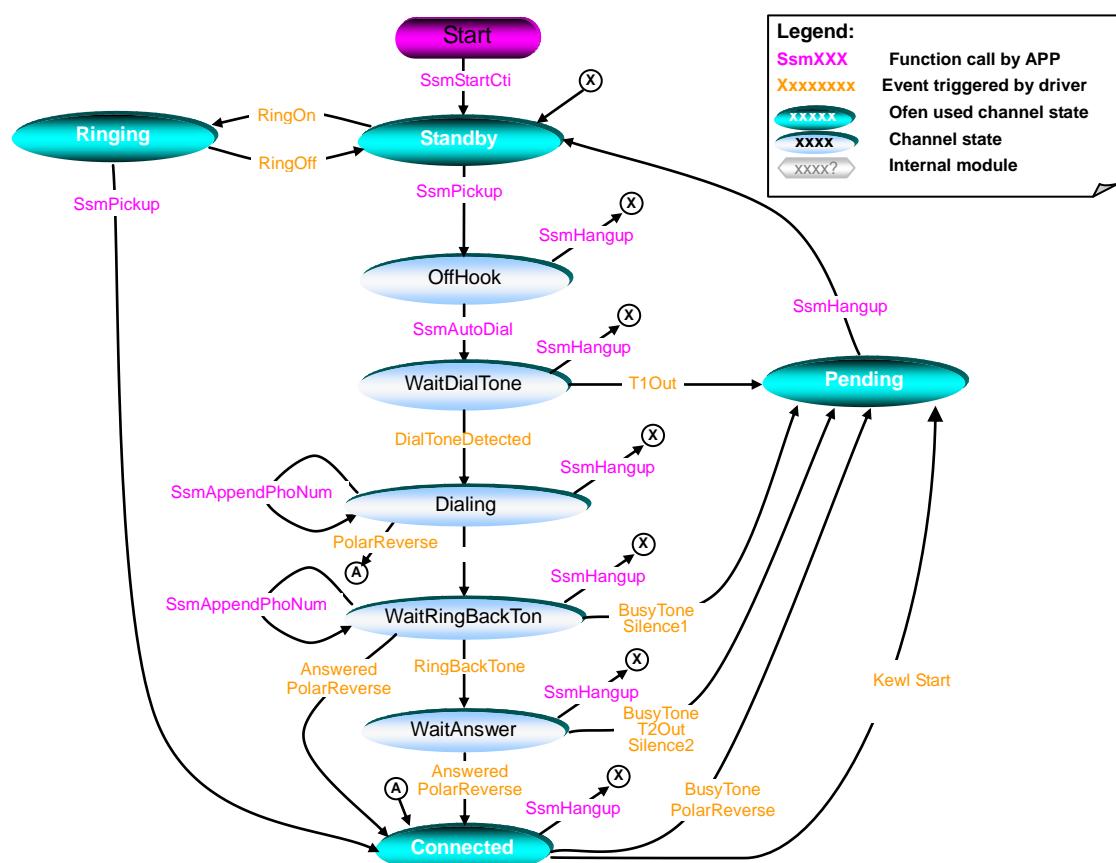
在 SynCTI 驱动程序中，可以有 2 种方法在模拟中继线上产生一次闪断信号：

- 调用 [SsmTxFlash](#) 函数。可直接指定闪断信号的保持时间。
- 使用 ‘!’ 字符作为参数调用函数 [SsmTxDtmf](#)。此时，产生的闪断信号的保持时间可以通过配置项 [DefaultTxFlashTime](#) 进行设置，缺省值为 500ms。

1.15.3.2 检测极性反转信号

有关检测极性反转信号的详细内容请参见本章中“[模拟电话线的电压变化](#)”部分内容。

1.15.3.3 模拟中继线通道的状态转移



状态转移图中，各状态标识的说明如下表所示：

状态标识	值	宏定义	描述
Standby	0	S_CALL_STANDBY	<p>“空闲”状态 在通道迁移到“空闲”状态前，驱动程序会：</p> <ul style="list-style-type: none"> ➤ 清空 DTMF 检测器 的所有缓冲区 ➤ 清空 CallerID 缓冲区 ➤ 关闭 DTMF 发生器

			<ul style="list-style-type: none"> ➤ 关闭信号音发生器 ➤ 清零并启动铃流信号检测器 ➤ 清零并关闭信号音检测器 ➤ 开启（如果 Caller ID 检测器工作在 DTMF 模式）或关闭（如果 Caller ID 检测器工作在 FSK 模式）DTMF 检测器 ➤ 关闭极性反转检测器（如果硬件支持极性反转信号检测功能） ➤ 将 AutoDial 任务的进展值设置为 DIAL_STANDBY ➤ 终止 WaitDtmf 任务（如果启动）
OffHook	1	S_CALL_PICKUPED	<p>“摘机”状态。 在通道迁移到“摘机”状态前，驱动程序会：</p> <ul style="list-style-type: none"> ➤ 清零并启动信号音检测器 ➤ 清零并关闭铃流信号检测器 ➤ 如果 Caller ID 检测器工作在 FSK 模式，启动 DTMF 检测器
Ringing	2	S_CALL_RINGING	“振铃”状态
Connected	3	S_CALL_TALKING	<p>“连接”状态，呼叫双方可以进行通话。 在通道迁移到“连接”状态前，驱动程序会：</p> <ul style="list-style-type: none"> ➤ 清零并启动信号音检测器 ➤ 如果 Caller ID 检测器工作在 FSK 模式，启动 DTMF 检测器 ➤ 清零铃流信号检测器的计数器
WaitDialtone	4	S_CALL_ANALOG_WAITDIALTONE	<p>“等待拨号音”状态。 在通道进入“等待拨号音”状态前，驱动程序会：</p> <ul style="list-style-type: none"> ➤ 清零信号音检测器 ➤ 将 AutoDial 任务进展值设置为 DIAL_DIALING
Dialing	5	S_CALL_ANALOG_TXPHONUM	<p>“拨号”状态。 在通道进入“拨号”状态前，驱动程序会：</p> <ul style="list-style-type: none"> ➤ 清零并关闭信号音检测器 ➤ 关闭 DTMF 检测器
WaitRingBackTone	6	S_CALL_ANALOG_WAITDIALRESULT	<p>“等待回铃音”状态。 在通道进入“等待回铃音”状态前，驱动程序会：</p> <ul style="list-style-type: none"> ➤ 清零并启动信号音检测器 ➤ 启动增强的远端摘机检测器，详细信息请参见本章中“增强的远端摘机检测器”部分内容 ➤ 启动 DTMF 检测器 ➤ 启动远端摘机检测器
Pending	7	S_CALL_PENDING	“未决”状态
WaitAnswer	9	S_CALL_WAIT_REMOTE_PICKUP	<p>“等待被叫用户摘机”状态。此时，被叫方应该听到振铃信号音。如果本端是坐席通道通过数字中继通道进行电话呼出，则应向坐席通道发送回铃音信号。 在通道进入“等待被叫用户摘机”状态前，驱动程序会：</p> <ul style="list-style-type: none"> ➤ 将 AutoDial 任务进展值设置为 DIAL_ECHOTONE，并向应用程序抛出E_PROC_AutoDial事件

上表中，状态的值与宏定义均在文件 ShpA3Api.h 中。

驱动程序内部事件的描述如下表所示：

事件名称	描述
RingOn	驱动程序在模拟中继线上检测到了振铃信号时触发此内部事件。详细信息请参见本章中“ 模拟电话线上的铃流信号 ”部分内容

RingOff	驱动程序在模拟中继线上检测到了振铃信号消失时触发此内部事件。详细信息请参见本章中“ 模拟电话线上的铃流信号 ”部分内容
T1Out	T1 定时器溢出。T1 定时器由配置项 MaxWaitDialToneTime 进行设置，缺省值为 3 秒。 T1 溢出后，驱动程序的操作为： ➤ 设置 AutoDial 任务的进展值为 DIAL_NO_DIALTONE，并向应用程序抛出 E_PROC_AutoDial 事件 ➤ 将 PendingReason 设置为 ANALOGOUT_NO_DIALTONE，然后迁移到 Pending 状态
T2Out	T2 定时器溢出。T2 定时器由配置项 MaxWaitAutoDialAnswerTime 进行设置，缺省值为 25 秒。 T2 溢出后，驱动程序的操作为： ➤ 设置 AutoDial 任务的进展值为 DIAL_NOANSWER，并向应用程序抛出 E_PROC_AutoDial 事件 ➤ 将 PendingReason 设置为 ANALOGOUT_NOANSWER，然后迁移到 Pending 状态
Silence1	远端摘机检测器输出 E_CHG_ToneAnalyze 事件（携带 CHKTONE_NOVOICE 参数）时触发此内部事件。有关远端摘机检测器的详细信息请参见本章中“ 远端摘机检测器 ”部分内容。 驱动程序会： ➤ 将 AutoDial 任务进展值设置为 DIAL_NOVOICE，并向应用程序抛出 E_PROC_AutoDial 事件 ➤ 将 PendingReason 设置为 ANALOGOUT_NOVOICE，然后迁移到 Pending 状态
Silence2	远端摘机检测器输出 E_CHG_ToneAnalyze 事件（携带 CHKTONE_ECHO_NOVOICE 参数）时触发此内部事件。有关远端摘机检测器的详细信息请参见本章中“ 远端摘机检测器 ”部分内容。 驱动程序会： ➤ 将 AutoDial 任务进展值设置为 DIAL_ECHO_NOVOICE，并向应用程序抛出 E_PROC_AutoDial 事件 ➤ 将 PendingReason 设置为 ANALOGOUT_ECHO_NOVOICE，然后迁移到 Pending 状态
DialToneDetected	信号音检测器在线路上检测到拨号音时触发此内部事件。详细信息请参见本章中“ 信号音检测器 ”部分内容
EndofDTMF	驱动程序将 DTMF 发送缓冲区中的全部字符发送完（拨号结束）时触发此内部事件
RingBackTone	信号音检测器在线路上检测到回铃音时触发此内部事件。详细信息请参见本章中“ 信号音检测器 ”部分内容
Answered	信号音检测器在线路上检测到下列结果时触发此内部事件： ➤ CHKTONE_VOICEF1：设置 AutoDial 任务进展值为 DIAL_VOICEF1，并向应用程序抛出 E_PROC_AutoDial 事件 ➤ DIAL_VOICEF2：设置 AutoDial 任务进展值为 DIAL_VOICEF2，并向应用程序抛出 E_PROC_AutoDial 事件 ➤ DIAL_VOICE：设置 AutoDial 任务进展值为 DIAL_VOICE，并向应用程序抛出 E_PROC_AutoDial 事件
PolarReverse	如果电压检测器在线路上检测到了极性反转信号，并且配置项 DisablePolarReverse 设置为 0 时触发此内部事件； 如果通道处于 Dialing、WaitRingBackTone 或 WaitAnswer 状态时，驱动程序会将 AutoDial 任务进展值设置为 DIAL_VOICE，并向应用程序抛出 E_PROC_AutoDial 事件
BusyTone	信号音检测器在线路上检测到忙音时触发此内部事件。有关设置忙音检测参数详细信息请参见本章中“ 信号音检测器 ”部分内容。 检测到忙音时，如果通道处于 Talking 状态，驱动程序会： ➤ 将 PendingReason 设置为 ANALOGOUT_TALKING_REMOTE_HANGUPED，然后迁移到 Pending 状态 如果是 WaitRingBackTone 或 WaitAnswer 状态，驱动程序会： ➤ 将 AutoDial 任务进展值设置为 DIAL_BUSYTONE，并向应用程序抛出 E_PROC_AutoDial 事件 ➤ 将 PendingReason 设置为 ANALOGOUT_BUSYTONE，然后迁移到 Pending 状态
Kewl Start	该事件用于检测远端挂机。当通道处于 Connected 状态时，如果电压检测器检测到线路电压处于 KSVoltageThreshold 大小（默认值为 0）并且保持时间达到 KSKeepTime （默认值为 256ms），则触发该事件，判定检测到远端挂机，通道迁移到 Pending 状态。Kewl Start 功能

由配置项 EnableKewlStart 开启。

状态机的输出事件类型如下表所示：

事件名称	描述
E_CHG_ChState	当通道的状态发生迁移时，驱动程序向应用程序抛出本事件
E_PROC_AutoDial	应用程序调用 SsmAutoDial 函数即启动了 AutoDial 任务。当 AutoDial 任务有进展时，驱动程序向应用程序抛出本事件。有关驱动程序抛出本事件的详细说明请参见前面章节

注：事件的声明在文件 `ShpA3Api.h` 中。

1.15.3.4 远端摘机检测器(Remote Pickup Detector) (SHT 系列 only)

远端摘机检测器只适用于模拟中继线通道，用于去话呼叫时精确检测被叫用户是否摘机。

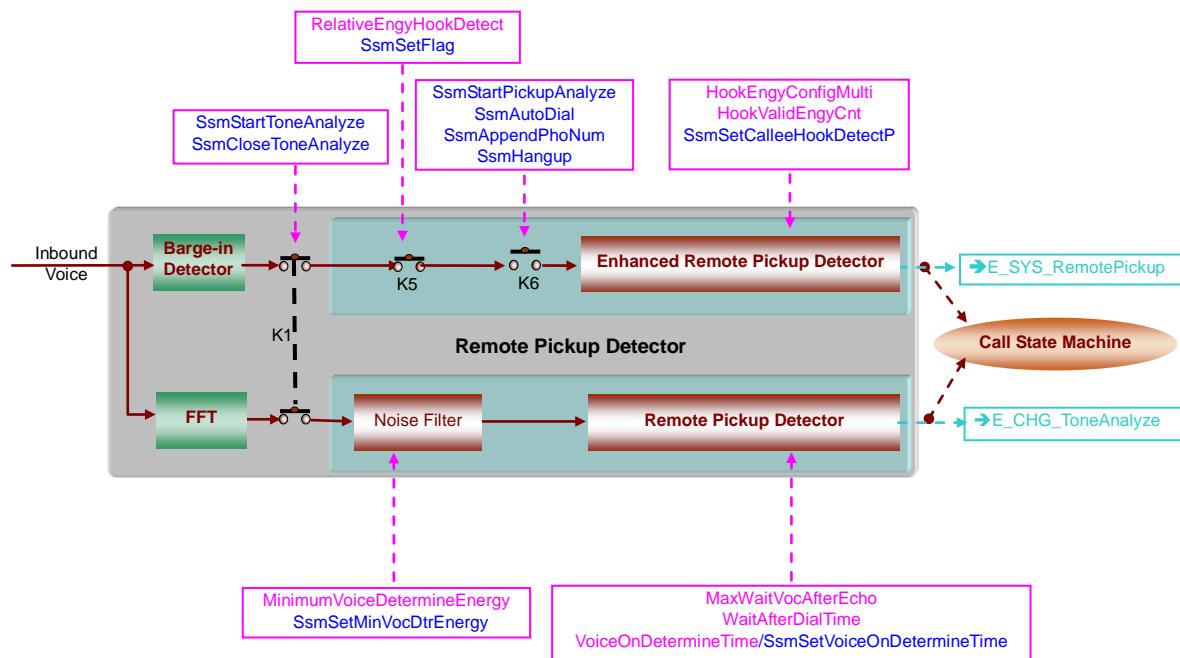
对于模拟中继线路，在进行去话呼叫时，当完成了被叫电话号码的发送后，如何判断被叫方是否摘机有时是一个比较困难的事情。目前，判断被叫摘机有 2 种方式：基于极性反转检测和基于语音活动检测的判断。

有关检测极性反转信号的详细内容请参见本章中“[模拟电话线的电压变化](#)”部分内容。

对于基于语音活动检测的远端摘机判断模式，拨号结束后大致会出现下列 2 种情形：

- 被叫方听到振铃信号后（不论时间长短），摘机，并且立即说话（比如，说“hello”）。在这种情况下，主叫通道对线路上的语音能量进行检测，当线路上的语音能量强度和持续时间大于预设门限时就认为被叫摘机。需要注意的是，当被叫设置了彩铃时，由于交换机会用播放的彩铃信号（语音信号）作为回铃音回送给主叫方，因此这种检测方法会失效。
- 被叫方听到振铃信号后摘机，但保持静默（不说话）。在这种情况下，主叫方的线上不会出现回铃音，也不会有任何语音信号，主叫通道通常无法判断被叫是否摘机，需要由应用程序自行进行判断。

三汇驱动程序为每个模拟中继通道都提供了独立的基于语音活动检测的远端摘机检测器，它的内部原理框图、相关的配置项和函数、输出的事件如下图所示：



远端摘机检测器包括常规的远端摘机检测器（包括上图中的 Noise Filter、Frequency Detector、Remote Pickup Detector）和增强的远端摘机检测器（包括上图中的 K5 开关、K6 开关和 Enhanced Remote Pickup Detector）组成，由 k1 开关进行控制。开关 K1 可以通过函数 [SsmStartToneAnalyze](#) / [SsmCloseToneAnalyze](#) 来设置。注意：开关 K1 同时控制了信号音检测器和能量检测器。

常规的远端摘机检测器使用板载 DSP 的 FFT 模块输出数据作为输入，适用于普通的应用场合。它抛出的是 [E_CHG_ToneAnalyze](#) 事件。

增强的远端摘机检测器使用一种特殊的算法来判断模拟电话线上的摘机动作，对远端摘机判断的准确度要高于常规的远端摘机检测器。它抛出的是 [E_SYS_RemotePickup](#) 事件。

1.15.3.4.1 普通的远端摘机检测器

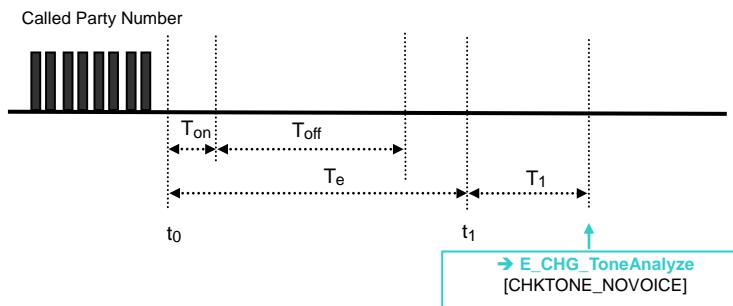
普通的远端摘机检测器检测线路上的语音活动。

噪声滤波器过滤掉线路上的噪声。它的工作原理和工作参数设置与“信号音检测器”节的噪声滤波器完全一样，详细内容请参见“信号音检测器”节的相关部分。

当噪声滤波器判定线路上存在语音信号后，只有当该语音信号不是信号音分析器中第 1 呼叫进程音检测器所指定的信号音，远端摘机检测器才认为线路上真正出现了语音。

当满足下列 3 种情况之一时远端摘机检测器就输出 [E_CHG_ToneAnalyze](#) 事件：

- (1) 被叫方一听到振铃信号就摘机，然后保持静默。在这种情况下，主叫方的线路上不会出现回铃音(也可能出现短暂的回铃音信号，但因不是一个完整的周期而被过滤掉，无法检测出来)，也不会有任何语音信号，如下图所示：



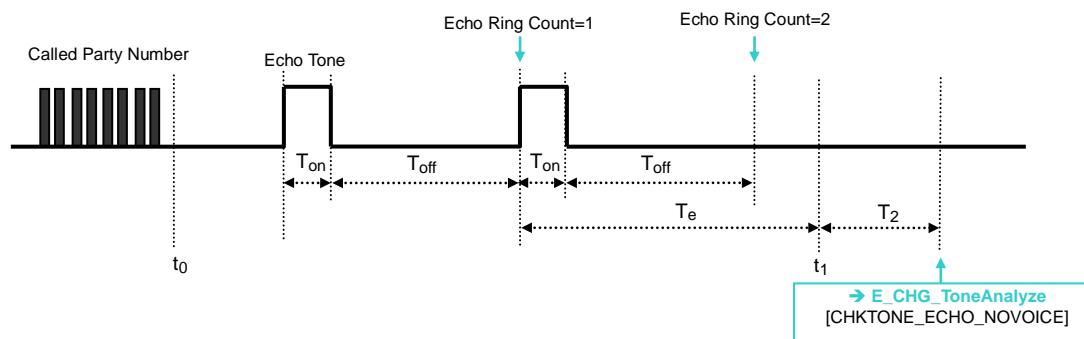
上图中，

T_{on} 、 T_{off} : 分别表示回铃音波形参数设置的高电平持续时间和低电平持续时间。

T_e : 回铃音周期的最大允许误差， $T_e = (T_{on} + T_{off}) * (1 + 20\%)$ 。

T_1 ：拨号完成后的静默持续时间判定门限。从 t_1 处开始，如果线路上保持静默的持续时间大于 T_1 ，驱动程序就输出 [E_CHG_ToneAnalyze](#) (携带 CHKTONE_NOVOICE 参数)。 T_1 由配置项 [WaitAfterDialTime](#) 指定。

- (2) 被叫方听完至少一声振铃信号后才摘机，但随后保持静默。在这种情况下，主叫通道上会检测到至少 1 次回铃音，但同样检测不到任何语音活动信号，如下图所示：



上图中，

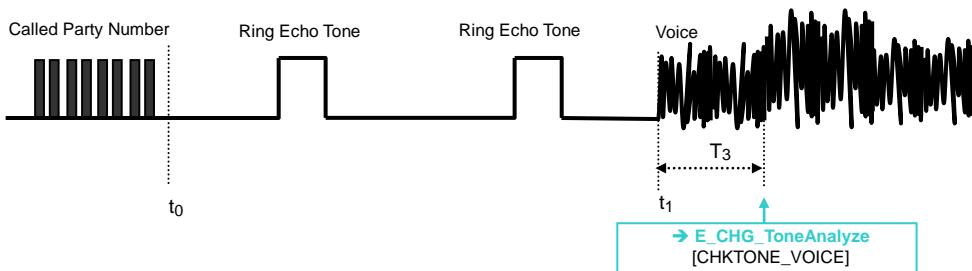
t_0 : 表示拨号完毕的时刻。

T_{on} 、 T_{off} : 分别表示回铃音波形参数设置的高电平持续时间和低电平持续时间。

T_e : 回铃音周期的最大允许误差， $T_e = (T_{on}+T_{off})*(1+20\%)$ 。

T_2 ：当回铃音消失后的静默持续时间判定门限。从 t_1 处（即判断回铃音信号消失的时间点）开始，如果线路上保持静默的持续时间大于 T_2 ，驱动程序就输出 [E_CHG_ToneAnalyze](#)（携带 [CHKTONE_ECHO_NOVOICE](#) 参数）。 T_2 由配置项 [MaxWaitVocAfterEcho](#) 指定。

(3) 被叫方听到振铃信号后（不论时间长短），摘机，并且立即说话（比如，说“hello”）。



上图中， T_3 为语音信号判定门限。从出现语音信号的 t_1 时刻开始，当语音信号持续时间大于 T_3 所指定的判定门限时，驱动程序就输出 [E_CHG_ToneAnalyze](#)（携带 [CHKTONE_VOICE](#) 参数）。 T_3 由配置项 [VoiceOnDetermineTime](#) 或函数 [SsmSetVoiceOnDetermineTime](#) 指定。函数 [SsmGetVoiceOnDetermineTime](#) 可以用来获取驱动程序保存的 T_3 的值。

1.15.3.4.2 增强的远端摘机检测器

K5 开关由配置项 [RelativeEngyHookDetect](#) 或函数 [SsmSetFlag](#)（携带 [F_RELATIVEENGYHOOKDETECT](#) 参数）进行设置，函数 [SsmGetFlag](#)（携带 [F_RELATIVEENGYHOOKDETECT](#) 参数）可以用来获取 K5 开关的状态。系统初始化成功后，K5 开关处于闭合状态。

K6 开关可以通过应用程序控制，也可由驱动程序自动控制。系统初始化成功后，K6 开关处于断开状态。

当满足下列条件时，K6 开关将被自动闭合以启动远端摘机检测器：

- 应用程序调用 [SsmStartPickupAnalyze](#) 函数。
- 应用程序调用 [SsmAutoDial](#) 函数发起去话呼叫后，在 [S_CALL_ANALOG_TXPHONUM](#) 状态，驱动程序

会将被叫号码逐个发送给对端交换机。当完成全部被叫号码的发送后，通道会迁移到 S_CALL_ANALOG_WAITDIALRESULT 状态。在状态迁移之前，驱动程序会自动闭合 K6。

- 如果通道处于 S_CALL_ANALOG_TXPHONUM 或 S_CALL_ANALOG_WAITDIALRESULT 状态时，应用程序调用了 [SsmAppendPhoNum](#) 函数。

当满足下列条件时，驱动程序自动断开 K6 开关（关闭远端摘机检测器）：

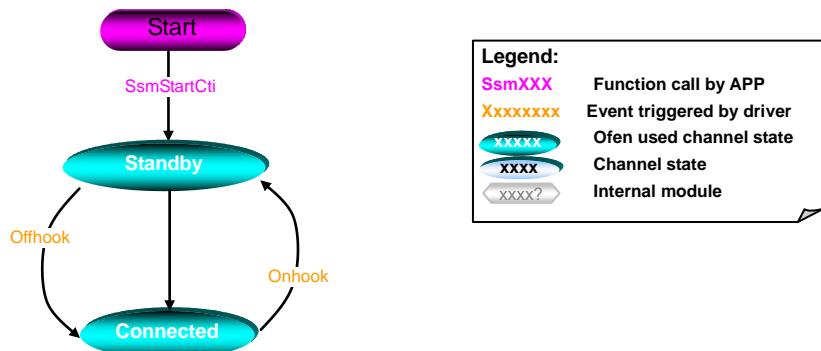
- 当远端摘机检测器完成一次检测并输出 [E_SYS_RemotePickup](#) 事件后。
- 应用程序调用 [SsmHangup](#) 函数。
- 模拟中继线通道的状态机收到信号音检测器抛出的“检测到忙音”事件后。

配置项 [HookEngyConfigMulti](#)、[HookValidEngyCnt](#) 或函数 [SsmSetCalleeHookDetectP](#) 用于设置远端摘机检测器的工作参数。

当远端摘机检测器检测到被叫摘机后，会抛出 [E_SYS_RemotePickup](#) 事件，并将此事件送到呼叫状态机进行后续处理。[SsmGetPickup](#) 函数可以用来取得检测结果。

1.15.4 坐席通道

1.15.4.1 状态迁移图



状态转移图中，各状态标识的说明如下表所示：

状态标识	值	宏定义	状态描述
Standby	0	S_CALL_STANDBY	空闲状态。 在通道迁移到“空闲”状态时，驱动程序会： <ul style="list-style-type: none"> ➤ 清空 DTMF 检测器 的所有缓冲区 ➤ 关闭 DTMF 发生器，并终止尚未完成的 WaitDtmf 任务 ➤ 关闭 信号音发生器
OffHook	1	S_CALL_PICKUPED	“摘机”状态。 在通道迁移到“摘机”状态时，驱动程序会： <ul style="list-style-type: none"> ➤ 如果配置项 AutoSendDialTone 设置为 1，开始向话机发送拨号音 ➤ 清空 DTMF 检测器 的所有缓冲区，并将其启动

上表中，状态的值与宏定义均在文件 ShpA3Api.h 中。

注意：当通道的状态发生迁移时，驱动程序都会向应用程序抛出 [E_CHG_ChState](#) 事件。

驱动程序自动触发的内部事件如下表示：

事件	事件描述
Offhook	当驱动程序检测到话机的摘机动作时触发此内部事件
Onhook	当驱动程序检测到话机的挂机动作时触发此内部事件

1.15.4.2 检测话机的摘机/挂机动作

当话机上发生摘机或挂机动作时，线路上可能会产生一些抖动信号。为了防止误检测，需要对其进行滤波处理，以消除抖动信号对摘机、挂机检测的准确性。配置项 [UserOnHookFilterTime](#) 用于设置摘机信号的最小保持时间，只有当驱动程序检测到摘机信号的保持时间大于该配置项的设定值时，驱动程序才会判定为摘机。

每当驱动程序判定摘机动作或挂机动作时，都会向应用程序抛出 [E_CHG_HookState](#) 事件。函数 [SsmGetHookState](#) 也可以用来获取坐席通道的摘/挂机状态。

1.15.4.3 检测话机的闪断信号

由于坐席通道用于模拟交换机上用户模块电路的特性，因此 Synway 板卡的坐席通道具有检测电话机上产生的闪断信号的能力。有关闪断信号的详细介绍请参见本章中“[模拟电话线上的闪断信号](#)”部分内容。

函数 [SsmSetLocalFlashTime](#) 或配置项 [MaxLocalFlashTime](#) 用于设置判定闪断信号的最大保持时间 N_{max} ，[LocalHookFilterTime](#) 用于设置判定闪断信号的最小保持时间 N_{min} 。当用户按下话机的叉簧，在保持一定时间（假定为 n ）后再次放开叉簧，如果 $N_{min} < n < N_{max}$ ，驱动程序判定用户发送了一个闪断信号，否则认为是用户的挂机动作。

驱动程序内部为每个坐席通道配备了闪断信号计数器，每当检测到一次闪断信号时，该计数器就加 1，同时向应用程序抛出 [E_CHG_FlashCount](#) 事件。函数 [SsmGetFlashCount](#) 也可以用来获取闪断信号的计数器值。每当坐席通道的状态迁移到空闲状态时，驱动程序都会将计数器自动清零。函数 [SsmClearFlashCount](#) 也可用于将计数器清零。

1.15.4.4 向电话机发送振铃信号

Synway 板卡的坐席通道都具有铃流发生器，用于向电话机发送振铃信号。常规振铃信号的波形特征为 1 秒高电平、4 秒低电平的周期性信号，但使用函数 [SsmSetRingPeriod](#) 可以调整振铃信号的高低电平持续时间，而使用配置项 [UserChGenerateRingMode](#) 可以将振铃信号的波形设置为持续性的信号。

[SsmStartRing](#)、[SsmStartRingWithCIDStr](#) 均可以用来向话机发送振铃信号。[SsmStartRing](#) 可以向模拟电话机和磁石话机发送振铃信号，[SsmStartRingWithCIDStr](#) 还可以向话机发送主叫号码信息，但只用于坐席通道。

驱动程序为每个坐席通道或磁石话机通道设置了一个发送振铃信号的计数器，每当完成一次振铃信号的发送（高电平），会将计数器加 1。函数 [SsmCheckSendRing](#) 可以用来获取该计数器的值。

在任何时刻，应用程序都可以调用函数 [SsmStopRing](#) 来停止向话机发送振铃信号。

1.15.4.5 在电话线上产生极性反转信号

有关极性反转信号的详细内容请参见本章中“[模拟电话线的电压变化](#)”部分。

Synway 板卡的坐席通道是否具有在模拟电话线上产生极性反转信号的功能，取决于 B 型 1.0 坐席模块是否带有极性反转的功能（说明：有些 B 型 1.0 坐席模块是不带有极性反转功能的）。

通道要使用极性反转信号的功能，需要通过配置项 [UserSendPolar](#) 进行设置。

函数 [SsmGetPolarState](#) 用于获取电话线上的当前极性，函数 [SsmSetPolarState](#) 用于设置电话线上的电压极性。

1.15.4.6 检测到话机摘机后自动送拨号音

当与坐席通道相连的电话机摘机后，是否由驱动程序自动向电话机发送拨号音，由配置项 [AutoSendDialTone](#) 决定。在发送拨号音期间，如果检测到 DTMF 字符，是否由驱动程序自动停止拨号音的发送，由配置项 [StopSendDialToneOnDtmf](#) 决定。

1.15.5 联合模块

对于 SHT 系列板卡, Synway 公司还提供一种特殊的业务模块的组合, 即将一个模拟中继线通道和一个坐席通道的接入电路安装到一个业务模块上, 称为联合模块。联合模块最大的特点, 就是具有“停电直通”功能, 当应用系统因故停电时, 联合模块上的模拟中继线通道与坐席通道是直通的, 因此, 坐席通道仍然能够经由模拟中继线通道与电话网进行通讯。

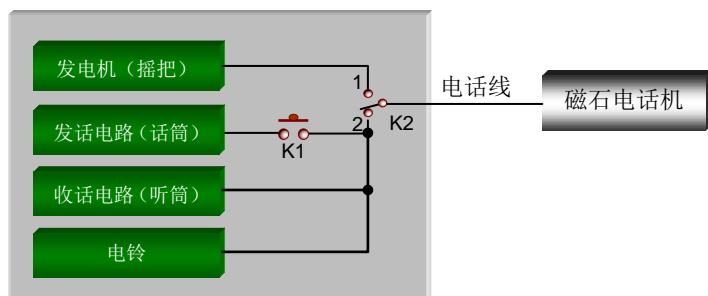
函数 [SsmSetUnimoduleState](#) 或配置项 [UnimoduleState](#) 用于设置联合模块上模拟中继线通道和坐席通道之间的连接关系。

1.15.6 磁石电话机通道

1.15.6.1 概述

在 SHT 系列板卡上安装了磁石模块的通道称为磁石电话机通道, 简称磁石通道, 用于连接磁石电话机。

磁石模块的内部结构与一个磁石电话机相似。磁石电话机的工作原理如下图所示:



磁石电话机内部由发话电话、收话电路、电铃、发电机和 K1、K2 两个开关组成(图中各部件均以单线方式示意连接, 实际为双线构成回路)。当摇动发电机手柄, K2 自动跳至 1 点, 将铃流送至对方话机的电铃。按下握柄 K1 (相当于普通话机的叉簧松开), 发话电路接通, 可以发话 (设置 K1 的主要作用是不发话时断开干电池)。

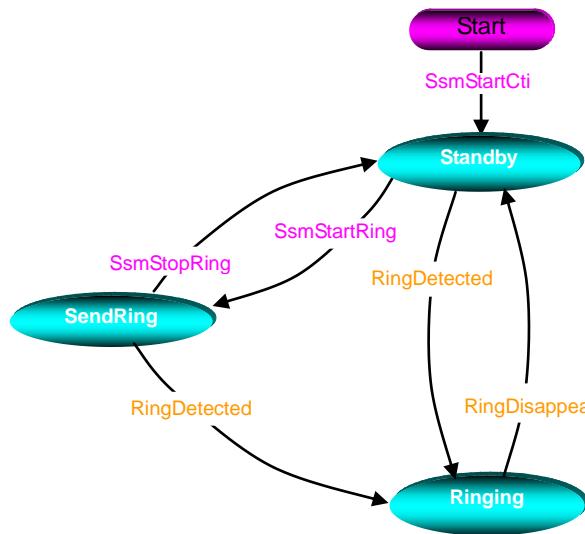
磁石模块的特点:

- 磁石模块只有空闲、发送振铃和振铃三个状态, 不能执行摘机、挂机操作。发送铃流信号的节奏与坐席模块相同, 即响 1 秒、停 4 秒, 在 4 秒钟的低电平期间仍可以正常检测振铃。
- 内置铃流信号发生器, 可以在线路上产生铃流信号, 其工作原理与操作方法与 SHT 系列 B 型板卡及以后版本的坐席模块的铃流发生器相同。如果在线路上已经存在着铃流信号, 则不能在磁石模块上发送铃流信号; 如果磁石模块在发送铃流信号时在线路上也检测到了铃流信号, 则磁石模块会自动停止本端发送的铃流信号。
- 磁石模块省略了 K1, 即当处于空闲和振铃状态时, 收发话电路总是接通的。

注意:

由于板卡上的业务模块识别电路不能识别磁石模块, 而是会将其识别为模拟中继线业务模块, 因此, 要使用磁石模块, 必须对配置项 [IsMagnetModule](#) 进行设置, 以便驱动程序能够正确区分这 2 种业务模块。

1.15.6.2 状态迁移图



在状态转移图中，各状态标识的说明如下表所示：

状态标识	值	宏定义	状态描述
Standby	0	S_CALL_STANDBY	空闲状态。驱动程序初始化成功后，进入本状态
SendRing	100	S_CALL_SENDRING	发送振铃状态。通道正在向磁石电话发送振铃。同时检测是否有铃流。铃流的检测可以通过配置项 RingDetectFilterPara 进行设置
Ringing	2	S_CALL_RINGING	振铃状态。磁石电话正在振铃。驱动检测振铃是否消失

1.15.7 EM 模块

E&M 是一个用户电话交换机和 PBX 上的通用中继信令技术。EM 接口是一种模拟接口，其信息与接续信令是分开的。EM 模块跟三汇其它模块一样，一个模块对应两个通道，第一个通道为 EM 控制通道(EM Control)，第二个通道为 EM 语音通道(EM Voice)。

对 EM 控制通道可操作的 API 有 SsmPickup 和 SsmHangup，获取状态的函数有 SsmGetChState 和 SsmGetHookState。SsmPickup 和 SsmHangup 向对端发送 1 和 0 信号，SsmGetHookState 主要检测对端发送过来的 0 或 1 信令。SsmGetChState 函数取本地摘挂机状态，这个状态由本地调用 SsmPickup 和 SsmHangup 改变。

对 EM 语音通道的操作与其它模块语音通道相同。

1.15.8 使用未安装业务模块的通道

SHT 系列板卡采用模块化结构，由底板和具有不同功能的业务模块组成。如果在板卡上没有安装任何业务模块，底板上的 DSP 芯片仍然能够正常工作，因此，仍然能够提供一定的语音处理功能。但是，由于没有电话线的物理接口电路，因此，它的语音信号只能来自 TDM 交换总线。在某些特殊的应用场合，当其它具有 TDM 交换总线的板卡需要录音功能时，可把语音数据通过 TDM 总线交换到未安装业务模块的通道，将 Barge in 检测器的输出作为录音操作的控制条件，即可进行录音操作。配置项 [NoModuleChBusRec](#) 或函数 [SsmSetNoModuleChBusRec](#) 用于提供这种功能。未安装业务模块通道具有的语音处理能力包括：录音、TDM 总线交换、DTMF 接收及 Bargeln 检测功能。

注意：当 SHT 系列板卡的底板上没有安装任何业务模块时，通道的类型编号为 20，表示无模块。

1.16 SHD 系列板卡 (CTI 系列)

1.16.1 SHD 系列板卡简介

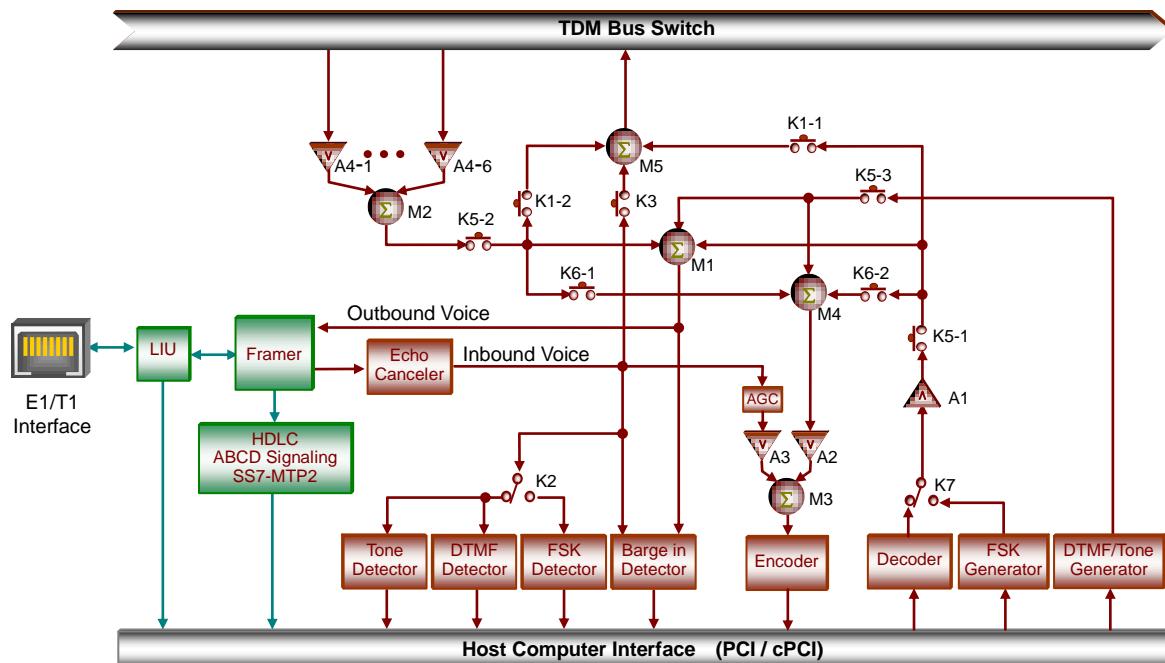
SHD 系列板卡支持的功能如下表所示：

Board Model	Form Factor	Trunk Type	Max. Ports	Voice Processing Capabilities	Conferencing Capabilities	FSK	Max Fax Resource	SS1	ISDN PRI	SS7	SS7 Signaling Links
SHD-30A-CT/PCI/SS1	PCI	1 E1	30	√	√	√	N/A	√	N/A	N/A	N/A
SHD-30A-CT/PCI/ISDN	PCI	1 E1	30	√	√	√	N/A	√	√	N/A	N/A
SHD-30A-CT/PCI/SS7	PCI	1 E1	30	√	√	√	N/A	√	√	√	1
SHD-60A-CT/PCI/SS1	PCI	2 E1	60	√	√	√	N/A	√	N/A	N/A	N/A
SHD-60A-CT/PCI/ISDN	PCI	2 E1	60	√	√	√	N/A	√	√	N/A	N/A
SHD-60A-CT/PCI/SS7	PCI	2 E1	60	√	√	√	N/A	√	√	√	2
SHD-120A-CT/PCI/SS1	PCI	4 E1	120	√	√	√	N/A	√	N/A	N/A	N/A
SHD-120A-CT/PCI/ISDN	PCI	4 E1	120	√	√	√	N/A	√	√	N/A	N/A
SHD-120A-CT/PCI/SS7	PCI	4 E1	120	√	√	√	N/A	√	√	√	4
SHD-30B-CT/PCI/SS7/FAX	PCI	1 E1	30	√	√	√	24	√	√	√	1
SHD-60B-CT/PCI/SS7/FAX	PCI	2 E1	60	√	√	√	16	√	√	√	2
SHD-30C-CT/PCI	PCI	1 E1/T1	30/23	√	√	√	N/A	√(E1Only)	√	√(E1Only)	1
SHD-30C-CT/PCI/FAX	PCI	1 E1/T1	30/23	√	√	√	24	√(E1Only)	√	√(E1Only)	1
SHD-60C-CT/PCI	PCI	2 E1/T1	60/46	√	√	√	N/A	√(E1Only)	√	√(E1Only)	2
SHD-60C-CT/PCI/FAX	PCI	2 E1/T1	60/46	√	√	√	16	√(E1Only)	√	√(E1Only)	2
SHD-120D-CT/PCI	PCI	4 E1/T1/J1	120/92	√	√	√	N/A	√	√	√	4
SHD-120D-CT/PCI/EC	PCI	4 E1/T1/J1	120/92	√	√	√	N/A	√	√	√	4
SHD-120D-CT/PCI/CAS	PCI	8 E1/T1	120/92	√	√	√	N/A	√	N/A	N/A	N/A
SHD-240D-CT/PCI	PCI	8 E1/T1/J1	240/184	√	√	√	N/A	√	√	√	8
SHD-240D-CT/PCI/EC	PCI	8 E1/T1/J1	240/184	√	√	√	N/A	√	√	√	8
SHD-240D-CT/PCI/CAS	PCI	8 E1/T1	240/184	√	√	√	N/A	√	N/A	N/A	N/A
SHD-30E-CT/PCI(SSW)	PCI(SSW)	1 E1/T1/J1	30/23	√	√	√	N/A	√	√	√	1
SHD-30E-CT/PCI/FAX(SSW)	PCI(SSW)	1 E1/T1/J1	30/23	√	√	√	32	√	√	√	1
SHD-30E-CT/PCI/EC(SSW)	PCI(SSW)	1 E1/T1/J1	30/23	√	√	√	N/A	√	√	√	1
SHD-60E-CT/PCI(SSW)	PCI(SSW)	2 E1/T1/J1	60/46	√	√	√	N/A	√	√	√	2
SHD-60E-CT/PCI/FAX(SSW)	PCI(SSW)	2 E1/T1/J1	60/46	√	√	√	64	√	√	√	2
SHD-60E-CT/PCI/EC(SSW)	PCI(SSW)	2 E1/T1/J1	60/46	√	√	√	N/A	√	√	√	2
SHD-120E-CT/PCI(SSW)	PCI(SSW)	4 E1/T1/J1	120/92	√	√	√	N/A	√	√	√	4
SHD-120E-CT/PCI/FAX(SSW)	PCI(SSW)	4 E1/T1/J1	120/92	√	√	√	64	√	√	√	4
SHD-120E-CT/PCI/EC(SSW)	PCI(SSW)	4 E1/T1/J1	120/82	√	√	√	N/A	√	√	√	4
SHD-240E-CT/PCI(SSW)	PCI(SSW)	8 E1/T1/J1	240/184	√	√	√	N/A	√	√	√	8
SHD-240E-CT/PCI/FAX(SSW)	PCI(SSW)	8 E1/T1/J1	240/184	√	√	√	64	√	√	√	8
SHD-240E-CT/PCI/EC(S)	PCI(SS)	8 E1/T1/J1	240/184	√	√	√	N/A	√	√	√	8

SW)	W)										
SHD-30E-CT/PCIe	PCIe	1 E1/T1/J1	30/23	√	√	√	N/A	√	√	√	1
SHD-30E-CT/PCIe/FAX	PCIe	1 E1/T1/J1	30/23	√	√	√	32	√	√	√	1
SHD-30E-CT/PCIe/EC	PCIe	1 E1/T1/J1	30/23	√	√	√	N/A	√	√	√	1
SHD-60E-CT/PCIe	PCIe	2 E1/T1/J1	60/46	√	√	√	N/A	√	√	√	2
SHD-60E-CT/PCIe/FAX	PCIe	2 E1/T1/J1	60/46	√	√	√	64	√	√	√	2
SHD-60E-CT/PCIe/EC	PCIe	2 E1/T1/J1	60/46	√	√	√	N/A	√	√	√	2
SHD-120E-CT/PCIe	PCIe	4 E1/T1/J1	120/92	√	√	√	N/A	√	√	√	4
SHD-120E-CT/PCIe/FAX	PCIe	4 E1/T1/J1	120/92	√	√	√	64	√	√	√	4
SHD-120E-CT/PCIe/EC	PCIe	4 E1/T1/J1	120/82	√	√	√	N/A	√	√	√	4
SHD-240E-CT/PCIe	PCIe	8 E1/T1/J1	240/184	√	√	√	N/A	√	√	√	8
SHD-240E-CT/PCIe/FAX	PCIe	8 E1/T1/J1	240/184	√	√	√	64	√	√	√	8
SHD-240E-CT/PCIe/EC	PCIe	8 E1/T1/J1	240/184	√	√	√	N/A	√	√	√	8
SHD-240E-CT/PCIe/VAR	PCIe	8 E1/T1/J1	240/184	√	√	√	N/A	√	√	√	8
SHD-30A-CT/cPCI/SS7	cPCI	1 E1	30	√	√	√	N/A	√	√	√	1
SHD-60A-CT/cPCI/SS7	cPCI	2 E1	60	√	√	√	N/A	√	√	√	2
SHD-120A-CT/cPCI/SS7	cPCI	4 E1	120	√	√	√	N/A	√	√	√	4
SHD-240A-CT/cPCI	cPCI	8 E1	240	√	√	N/A	N/A	N/A	√	√	8
SHD-480A-CT/cPCI	cPCI	16 E1	480	√	√	N/A	N/A	N/A	√	√	16
SHD-30B-CT/cPCI/SS7/FAX	cPCI	1 E1	30	√	√	√	24	√	√	√	1
SHD-60B-CT/cPCI/SS7/FAX	cPCI	2 E1	60	√	√	√	16	√	√	√	2
SHD-240S-CT/cPCI	cPCI	8 E1	240	√	N/A	N/A	N/A	N/A	√	√	8
SHD-480S-CT/cPCI	cPCI	16 E1	480	√	N/A	N/A	N/A	N/A	√	√	16

注：SHD 系列板卡均具有 TDM 交换总线功能，因此未在上图中列出。

1.16.2 SHD 系列板卡的工作原理框图



SHD 系列板卡的工作原理图如下图所示：

原理框图中有关录音操作的符号说明：

名称	描述
M1	去话混音器
K5-3 ^[1]	信号音发生器 、 DTMF 发生器 的信号输出控制开关，由驱动程序自动控制，通常处于断开状态，只有当

	应用程序调用 SsmTxDtmf 函数启动 DTMF 发生器, 或者调用 SsmSendTone / SsmSendToneEx 函数启动信号音发生器后, 驱动程序才会自动闭合 K5-3 开关; 上述任务结束后, K5-3 自动断开
M2	下总线信号混音器
A4-1 ... A4-6	下总线信号进入 M2 前的音量调节器, 可以通过函数 SsmSetListenVlmInConf 进行设置, 缺省值为 0DB。另外, 也可以通过配置项 DefaultSpeakVolume 或下列函数对将要下总线过来的通道音量进行设置: ◆ SsmTalkWithEx ◆ SsmListenToEx / SsmLinkFromEx / SsmLinkFromAllCh
K5-2 ^[1]	下总线信号混音器的输出信号控制开关, 由驱动程序自动控制。当应用程序调用的总线连接函数中包含下总线的操作时, 驱动程序会自动闭合 K5-2 开关; 当 M2 的输入全部被断开时, 驱动程序自动断开 K5-2。有关交换总线的函数请参见本章中 交换总线 部分内容
M3	录音混音器。M3 的输出进入编码器 (Encoder) 模块, 然后形成录音数据
A3	来话信号进入录音混音器 M3 前的音量调节器, 可以通过函数 SsmSetRecVolume 进行设置, 也可以通过配置项 DefaultRecordVolume 设置, 缺省值为 0
A2	去话信号进入录音混音器 M3 前的音量调节器, 可以通过函数 SsmSetRecMixer 进行设置, 也可以通过配置项 DefaultRecordMixerVolume 设置, 缺省值 -7, 关闭 A2
M4	录音用途的去话混音器。用于将下总线信号混音器的输出、放音数据、信号音发生器/DTMF 发生器的输出进行混音, 混音结果进入录音混音器 M3
K6-1 K6-2	开关 K6-1 控制总线混音器 M2 的输出是否作为录音混音器 M3 的一个输入信号源; 开关 K6-2 控制放音数据是否作为录音混音器 M3 的一个输入信号源。函数 SsmSetRecBack 用来设置这 2 个开关
M5	上总线混音器。用于将放音数据、下总线信号混音器 M2 的输出、来话信号进行混音, 混音结果上相到 TDM 交换总线。 板卡初始化时, 会将 M5 的输出上行到 TDM 总线的某个时隙 (记为 ts) 上, 函数 SsmLinkToBus 可以用来改变 ts
K3	来话信号是否进入上总线混音器 M5 的控制开关, 可通过函数 SsmSetFlag (携带 F_InVoiceToBus 参数) 或配置项 InVoiceToBus 进行设置, 缺省值为闭合
K1-1 ^[1]	控制放音数据是否进入混音器 M5, 进而上行到 TDM 交换总线上的开关, 通常为断开状态, 可以通过函数 SsmSetPlayDest 进行控制。K1-1 只能用于电话会议中给会议室播放背景音的用途
K1-2	M2 的输出信号是否重新回到 TDM 交换总线的控制开关, 可以通过函数 SsmSetFlag (携带 F_MixerResToBus 参数) 进行设置, 缺省值为断开
	与放音操作相关的模块
K7	语音播放/FSK 数据发送器的切换开关, 由驱动程序自动控制。当应用程序调用播放函数时, K7 指向语音编码器模块 Decoder; 当应用程序调用 SsmStartSendFSK 函数启动发送 FSK 数据时, K7 执行 FSK Generator。有关语音播放的函数请参见 语音播放 部分内容
A1	放音的音量调节器, 可以通过配置项 DefaultPlayVolume 设置, 也可通过函数 SsmSetPlayVolume 或 SsmSetPlayGain 进行调节, 缺省的音量设置为 0
K5-1 ^[1]	控制放音操作的开关, 由驱动程序自动控制。当应用程序调用播放函数时, 驱动程序自动开启 K5-1 开关; 当放音任务结束时, 驱动程序自动关闭 K5-1。有关语音播放的函数请参见 语音播放 部分内容
K2	FSK 检测器与其它检测器之间的切换开关, 通常指向其它的检测器, 只有在开启 FSK 检测器期间才会指向 FSK 检测器

注[1]:

- 对于 SHD 系列的某些型号的板卡, 由于板载 DSP 芯片的资源限制, 当 K1-1 断开时, K5-1、K5-2、K5-3 中只能有一个闭合。因此, 当通道正在进行放音操作 (此时 K5-1 已经闭合) 或启动了信号音发生器 (此时 K5-3 已经闭合) 时, 如果应用程序调用了会引发 K5-2 闭合的总线操作函数, 如果配置项 [LinkFromStopPlayAndTone](#) 设置为 1, 放音操作将会被立即终止, K5-2 立即闭合; 如果配置项 [LinkFromStopPlayAndTone](#) 设置为 0, K5-2 在放音操作正常终止后才会闭合。引发 K5-2 闭合的总线操作函数包括:

- [SsmTalkWith](#) / [SsmTalkWithEx](#)
- [SsmListenTo](#) / [SsmListenToEx](#) / [SsmLinkFrom](#) / [SsmLinkFromEx](#) / [SsmLinkFromAllCh](#)
- [SsmLinkFromBus](#) / [SsmLinkFromBusEx](#)

涉及的型号包括:

- SHD-30A-CT/PCI/SS1
- SHD-30A-CT/PCI/ISDN

- SHD-30A-CT/PCI/SS7
- SHD-30B-CT/PCI/SS7/FAX
- SHD-60A-CT/PCI/SS1
- SHD-60A-CT/PCI/ISDN
- SHD-60A-CT/PCI/SS7
- SHD-60B-CT/PCI/SS7/FAX
- SHD-120A-CT/PCI/SS1
- SHD-120A-CT/PCI/ISDN
- SHD-120A-CT/PCI/SS7
- SHD-30A-CT/cPCI/SS7
- SHD-30B-CT/cPCI/SS7/FAX
- SHD-60A-CT/cPCI/SS7
- SHD-60B-CT/cPCI/SS7/FAX
- SHD-120A-CT/cPCI/SS7
- SHD-30C-CT/PCI
- SHD-60C-CT/PCI
- SHD-30C-CT/PCI/FAX
- SHD-60C-CT/PCI/FAX

- 如果 K1-1 闭合，则 K5-1、K5-2 可以同时闭合，但 K5-3 必须保持断开。

➤ 放音操作

应用程序调用播放函数后，K7 指向语音解码器 Decoder，K5-1 闭合。解码后的语音数据经过音量调节器 A1 进行音量调节后，可以有 3 个去向：

- ✧ 进入 M1 混音器，与其它信号源混音后形成去话信号。
- ✧ 在 K6-2 控制下，进入 M4 混音器，与其它信号源混音后形成录音操作的信号源。
- ✧ 在 K1-1 控制下，进入混音器 M5，与其它信号源混音后形成上总线的信号源，只能用于电话会议中给会议室播放背景音。

➤ 录音操作的信号源选择范例

灵活应用上述录音控制开关和音量调节器可以实现很多特殊的应用。下面范例中，ch1 表示通道 1，ch2 表示通道 2。

- ✧ 范例 1：只录制 ch1 的来话信号，使用缺省音量。实现代码为：

```
SsmSetRecVolume(ch1, 0);           // 开启 A3，设置增益为 0DB
SsmSetRecMixer(ch1, FALSE, 0);     // 关闭 A2
SsmRecToFile(ch1, .....);         // 启动文件录音任务.....
```

- ✧ 范例 2：ch1 与 ch2 建立了双向通话后，在 ch1 上同时录制 ch1 的来话信号（音量提升 6DB）+ 总线混音器 M2 输出信号（音量衰减 3DB）。实现代码为：

```
SsmTalkWith(ch1, ch2);           // 建立双向连接: ch1↔ch2, K5-2 闭合
SsmSetRecBack(ch1,2);             // K6-1 闭合, K6-2 断开
SsmSetRecVolume(ch1, 2);          // 开启 A3，设置增益为 2*3DB=6DB
SsmSetRecMixer(ch1, TRUE, -1);    // 开启 A2，设置增益为 -1*3DB=-3DB
SsmRecToFile(ch1, .....);         // 启动文件录音任务.....
```

- ✧ 范例 3：ch1 以主人身份加入了一个电话会议室（假设会议室编号为 n），在 ch1 上播放背景音乐，同时通过 ch1 对电话会议进行录音（包括背景音乐）。所有信号源均为正常音量，即增益为 0。实现代码为：

```
SsmJoinConfGroup(n, ch1, .....); // ch1 加入会议室: K5-2 闭合, K3 闭合
SsmSetPlayDest(int ch, 1);       // 闭合 K1-1
SsmSetRecBack(ch1,1);            // K6-1 闭合, K6-2 闭合
SsmPlayFile(ch1, .....);         // ch1 启动放音任务: K7 指向 Decoder, K5-1 闭合
```

```
SsmSetRecVolume(ch1, 0);           // 开启 A3, 设置增益为 0dB
SsmSetRecMixer(ch1, TRUE, 0);      // 开启 A2, 设置增益为 0dB
SsmRecToFile(ch1, .....);         // 启动文件录音任务.....
```

1.16.3 设置来话呼叫的收号规则

来话呼叫过程由通道的状态机自动实现，应用程序只需要预先设置好收号规则即可。收号规则是指当出现来话呼叫时，本端接收对端交换机发送的被叫方电话号码、主叫方电话号码等信息的规则。

来话呼叫的收号规则有 2 种工作模式，由配置项 [DefaultRcvPhoNumLen](#) 进行设置：

- ◆ 定长方式：指本端的电话号码只有一个，并且具有固定的长度。例如，“110”。当驱动程序收齐了设定长度的被叫号码时，就认为收号过程结束。如果收号规则设置为定长方式，配置项 [DefaultRcvCallerID](#) 和 [RcvPhoNumCfgLen](#) 起作用。
- ◆ 入局字冠方式：可以为本端的应用程序设置多个被叫号码，每个号码都以一个字冠开头。如果收号规则设置为入局字冠方式，配置项 [MaxPhoNumRule](#) 和 [Rule](#) 用于设置具体的收号规则。

TUP 通道、ISUP 通道、ISDN 通道和 SS1 通道均可以独立设置收号规则：

- ◆ TUP 通道的收号规则在[TUP]节中进行设置。
- ◆ ISUP 通道的收号规则在[ISUP]节中进行设置。
- ◆ ISDN 通道的收号规则在[ISDN]节中进行设置。
- ◆ SS1 通道的收号规则在[SS1]节中进行设置。

1.16.4 SS7 协议的编程

1.16.4.1 OPC 和 DPC

信令点(Signaling Point, SP)是信令网中传送各种信令消息的源点和目的点。对于国际信令点编码，CCITT 建议采用 14bit 编码，编码方式见下图：

3bit	8bit	3bit
大区或洲 (ZONE)	国家或地区 (AREA)	信令点 (POINT)

中国的信令点编码采用 24bit 方式，各比特的含义如下图所示：

8bit	8bit	8bit
主信令区编码	分信令区编码	信令点编码

Synway 的 SHD 系列板卡同时支持国际信令点编码方式和中国信令点编码方式。

信令消息的起源点称为 OPC(Originating Point Code)，信令消息的目的点称为 DPC(Destination Point Code)。OPC、DPC 均在 Ss7Server.ini 文件中设置。

1.16.4.2 时隙分配

SHD 系列板卡的 PCM 时隙分配如下表所示：

Board Model	Sync. Time Slot	Voice Time Slot	Signaling Time Slot		
			SS1	ISDN PRI	SS7
SHD-30A-CT/PCI/SS1	0	1-15,17-31	16	N/A	N/A
SHD-30A-CT/PCI/ISDN	0	1-15,17-31	16	16	N/A
SHD-30A-CT/PCI/SS7	0	1-15,17-31	16	16	16 ^[1]
		2-31	-	-	1 ^[1]
SHD-60A-CT/PCI/SS1	0	1-15,17-31	16	N/A	N/A
SHD-60A-CT/PCI/ISDN	0	1-15,17-31	16	16	N/A
SHD-60A-CT/PCI/SS7	0	1-15,17-31	16	16	16 ^[1]
		2-31	-	-	1 ^[1]
SHD-120A-CT/PCI/SS1	0	1-15,17-31	16	N/A	N/A
SHD-120A-CT/PCI/ISDN	0	1-15,17-31	16	16	N/A
SHD-120A-CT/PCI/SS7	0	1-15,17-31	16	16	16 ^[1]
SHD-30B-CT/PCI/SS7/FAX	0	1-15,17-31	16	16	16 ^[1]
		2-31	-	-	1 ^[1]
SHD-60B-CT/PCI/SS7/FAX	0	1-15,17-31	16	16	16 ^[1]
		2-31	-	-	1 ^[1]
SHD-30C-CT/PCI	0(E1) N/A(T1)	1-15,17-31(E1) 0-22(T1 ISDN)	16(E1)	16(E1) 23(T1)	16 ^[1] (E1)
SHD-30C-CT/PCI/FAX	0(E1) N/A(T1)	1-15,17-31(E1) 0-22(T1 ISDN)	16(E1)	16(E1) 23(T1)	16 ^[1] (E1)
SHD-60C-CT/PCI	0(E1) N/A(T1)	1-15,17-31(E1) 0-22(T1 ISDN)	16(E1)	16(E1) 23(T1)	16 ^[1] (E1)
SHD-60C-CT/PCI/ FAX	0(E1) N/A(T1)	1-15,17-31(E1) 0-22(T1 ISDN)	16(E1)	16(E1) 23(T1)	16 ^[1] (E1)
SHD-120D-CT/PCI	0	1-15,17-31	16	16	16 ^[1]
SHD-120D-CT/PCI/EC	0	1-15,17-31	16	16	16 ^[1]
SHD-120D-CT/PCI/CAS	0	1-15,17-31	16	-	-
SHD-240D-CT/PCI	0	1-15,17-31	16	16	16 ^[1]
SHD-240D-CT/PCI/EC	0	1-15,17-31	16	16	16 ^[1]
SHD-240D-CT/PCI/CAS	0	1-15,17-31	16	-	-
SHD-30E-CT/PCI(SSW)	0	1-15,17-31	16	16	16 ^[1]
SHD-30E-CT/PCI/FAX(SSW)	0	1-15,17-31	16	16	16 ^[1]
SHD-30E-CT/PCI/EC(SSW)	0	1-15,17-31	16	16	16 ^[1]
SHD-60E-CT/PCI(SSW)	0	1-15,17-31	16	16	16 ^[1]
SHD-60E-CT/PCI/FAX(SSW)	0	1-15,17-31	16	16	16 ^[1]
SHD-60E-CT/PCI/EC(SSW)	0	1-15,17-31	16	16	16 ^[1]
SHD-120E-CT/PCI(SSW)	0	1-15,17-31	16	16	16 ^[1]
SHD-120E-CT/PCI/FAX(SSW)	0	1-15,17-31	16	16	16 ^[1]
SHD-120E-CT/PCI/EC(SSW)	0	1-15,17-31	16	16	16 ^[1]
SHD-240E-CT/PCI(SSW)	0	1-15,17-31	16	16	16 ^[1]
SHD-240E-CT/PCI/FAX(SSW)	0	1-15,17-31	16	16	16 ^[1]
SHD-240E-CT/PCI/EC(SSW)	0	1-15,17-31	16	16	16 ^[1]
SHD-30E-CT/PCIe	0	1-15,17-31	16	16	16 ^[1]
SHD-30E-CT/PCIe/FAX	0	1-15,17-31	16	16	16 ^[1]
SHD-30E-CT/PCIe/EC	0	1-15,17-31	16	16	16 ^[1]
SHD-60E-CT/PCIe	0	1-15,17-31	16	16	16 ^[1]
SHD-60E-CT/PCIe/FAX	0	1-15,17-31	16	16	16 ^[1]
SHD-60E-CT/PCIe/EC	0	1-15,17-31	16	16	16 ^[1]
SHD-120E-CT/PCIe	0	1-15,17-31	16	16	16 ^[1]
SHD-120E-CT/PCIe/FAX	0	1-15,17-31	16	16	16 ^[1]

SHD-120E-CT/PCIe/EC	0	1-15,17-31	16	16	16 ^[1]
SHD-240E-CT/PCIe	0	1-15,17-31	16	16	16 ^[1]
SHD-240E-CT/PCIe/FAX	0	1-15,17-31	16	16	16 ^[1]
SHD-240E-CT/PCIe/EC	0	1-15,17-31	16	16	16 ^[1]
SHD-240E-CT/PCIe/VAR	0	1-15,17-31	16	16	16 ^[1]
SHD-30A-CT/cPCI/SS7	0	1-15,17-31	16	16	16 ^[1]
		2-31	-	-	1 ^[1]
SHD-60A-CT/cPCI/SS7	0	1-15,17-31	16	16	16 ^[1]
		2-31	-	-	1 ^[1]
SHD-120A-CT/cPCI/SS7	0	1-15,17-31	16	16	16 ^[1]
		2-31	-	-	1 ^[1]
SHD-240A-CT/cPCI	0	1-15,17-31	N/A	16	16 ^[1]
SHD-480A-CT/cPCI	0	1-15,17-31	N/A	16	16 ^[1]
SHD-30B-CT/cPCI/SS7/FAX	0	1-15,17-31	16	16	16 ^[1]
		2-31	-	-	1 ^[1]
SHD-60B-CT/cPCI/SS7/FAX	0	1-15,17-31	16	16	16 ^[1]
		2-31	-	-	1 ^[1]
SHD-240S-CT/cPCI	0	1-15,17-31	N/A	16	16 ^[1]
SHD-480S-CT/cPCI	0	1-15,17-31	N/A	16	16 ^[1]

Note [1]: 数字中继线上是否包含 SS7 信令链路，以及信令链路具体由哪个时隙承载，由配置项 [UseTS16AsCircuit](#) 和 [Ss7SignalingTS](#) 决定。

1.16.4.3 设置地址信号中备用值的对应字符

在 TUP 和 ISUP 协议中，被叫号码、主叫号码等信息在消息中不是按照标准的 ASCII 字符进行传输，而是使用编码值。而驱动程序返回给应用程序的则是标准的 ASCII 字符形式的主叫号码、被叫号码，因此，需要设置一个翻译规则，将消息中的编码值翻译成统一的 ASCII 字符。配置项 [AddressSignal](#) 用于实现将地址信号中的备用值映射到特定的字符，如下表所示：

地址信号的值		对应字符	属性
16 进制	10 进制		
0	0	'0'	不可更改
1	1	'1'	不可更改
2	2	'2'	不可更改
3	3	'3'	不可更改
4	4	'4'	不可更改
5	5	'5'	不可更改
6	6	'6'	不可更改
7	7	'7'	不可更改
8	8	'8'	不可更改

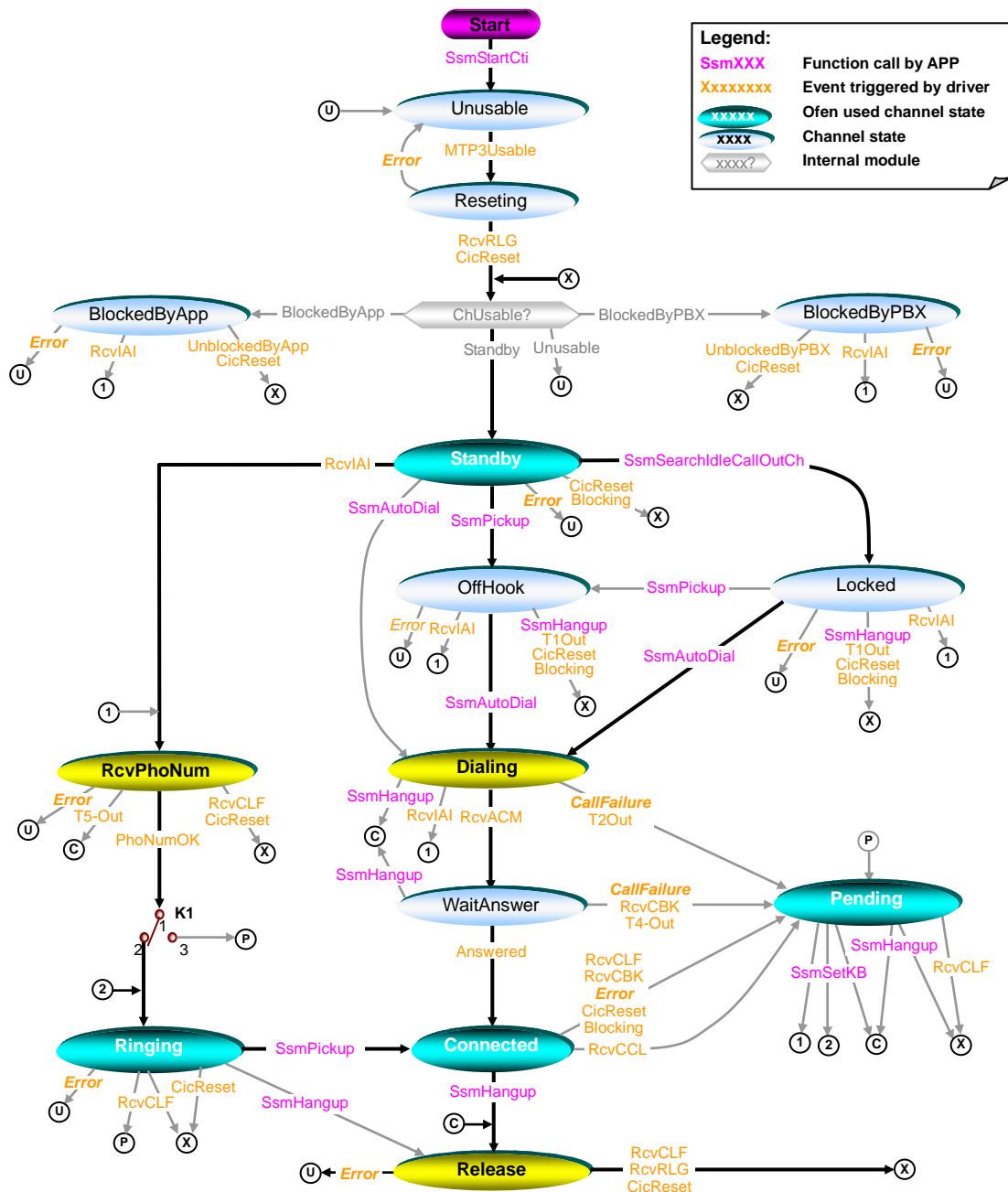
9	9	'9'	不可更改
a	10	备用	可通过配置项 AddressSignal 进行设置
b	11	备用	可通过配置项 AddressSignal 进行设置
c	12	备用	可通过配置项 AddressSignal 进行设置
d	13	备用	可通过配置项 AddressSignal 进行设置
e	14	备用	可通过配置项 AddressSignal 进行设置
f	15	ST 信号	不可更改

1.16.4.4 SS7 信令应用系统的配置实例

有关 SS7 信令应用系统的配置实例，请参见第 4 章中“[SS7 信令服务器的应用及配置实例](#)”部分内容。

1.16.4.5 TUP 协议的编程与应用

1.16.4.5.1 TUP 通道的状态转移



驱动程序内部事件的描述如下表所示：

事件名称	描述
Mtp3Usable	驱动程序收到 SS7 Server 的信令可用消息后，会向所有通道的状态机发送本事件，使通道从 Unusable 迁移到 Resetting 状态，开始进行电路复原
CicReset	电路复原成功，当驱动程序收到电路复原消息（GRA、GRS 或 RSC 消息）时触发本事件。 驱动程序的操作：如果本通道已经将对端闭塞，或者已经向对端发送了闭塞消息并等待对端

	响应应答消息，则驱动程序会向对端发送 BLO 消息，以保持对对端的闭塞，同时向应用程序抛出 E_CHG_RemoteChBlock 事件																
Blocking	本通道被本端闭塞或被对端闭塞。触发条件为： <ul style="list-style-type: none">• 对端闭塞：驱动程序收到对端交换机的群闭塞消息（MGB、SGB、HGB）或电路闭塞消息（BLO）• 本端闭塞：应用程序调用 SsmBlockLocalCh 或 SsmBlockLocalPCM 函数																
UnblockedByPBX	解除对端闭塞：驱动程序收到对端交换机的 MGU、HGU、SGU、UBL 消息时触发本事件																
UnblockedByApp	解除本端闭塞：应用程序调用解除本端闭塞函数 SsmUnblockLocalPCM 或 SsmUnblockLocalCh 后触发本事件																
RcvIAI	驱动程序收到对端交换机的来话呼叫建立请求，即 IAI 或 IAM 消息。驱动程序是否接受此来话呼叫请求，取决于通道的状态： <table border="1"><thead><tr><th>状态</th><th>描述</th></tr></thead><tbody><tr><td>BlockedByPBX</td><td>在本端闭塞状态，虽然通道不能进行去话呼叫，但依然可以接受来话呼叫。</td></tr><tr><td>BlockedByApp</td><td>在去话呼叫过程中收到对端交换机的 IAI/IAM 消息，表明发生了“同抢”。根据 TUP 协议规定，如果通道的 CIC 编号为本端主控的电路，则此 IAI/IAM 消息将被丢弃，本端继续去话呼叫；否则，本端放弃去话呼叫，转而处理来话呼叫。</td></tr><tr><td>Dialing</td><td>通道正在准备进行去话呼叫，但尚未送出 IAI 或 IAM 消息。此时收到对端的来话呼叫请求，应予以响应。</td></tr><tr><td>OffHook</td><td>通道正在准备进行去话呼叫，但尚未送出 IAI 或 IAM 消息。此时收到对端的来话呼叫请求，应予以响应。</td></tr><tr><td>Locked</td><td>通道正在准备进行去话呼叫，但尚未送出 IAI 或 IAM 消息。此时收到对端的来话呼叫请求，应予以响应。</td></tr><tr><td>Standby</td><td>接受来话呼叫。</td></tr><tr><td>其它</td><td>拒绝。</td></tr></tbody></table> <p>如果驱动程序接受来话呼叫，在将 IAM 消息中的被叫号码信息保存到缓冲区后，向应用程序抛出 E_CHG_RxPhoNumBuf 事件，然后将通道迁移到 RcvPhoNum 状态</p>	状态	描述	BlockedByPBX	在本端闭塞状态，虽然通道不能进行去话呼叫，但依然可以接受来话呼叫。	BlockedByApp	在去话呼叫过程中收到对端交换机的 IAI/IAM 消息，表明发生了“同抢”。根据 TUP 协议规定，如果通道的 CIC 编号为本端主控的电路，则此 IAI/IAM 消息将被丢弃，本端继续去话呼叫；否则，本端放弃去话呼叫，转而处理来话呼叫。	Dialing	通道正在准备进行去话呼叫，但尚未送出 IAI 或 IAM 消息。此时收到对端的来话呼叫请求，应予以响应。	OffHook	通道正在准备进行去话呼叫，但尚未送出 IAI 或 IAM 消息。此时收到对端的来话呼叫请求，应予以响应。	Locked	通道正在准备进行去话呼叫，但尚未送出 IAI 或 IAM 消息。此时收到对端的来话呼叫请求，应予以响应。	Standby	接受来话呼叫。	其它	拒绝。
状态	描述																
BlockedByPBX	在本端闭塞状态，虽然通道不能进行去话呼叫，但依然可以接受来话呼叫。																
BlockedByApp	在去话呼叫过程中收到对端交换机的 IAI/IAM 消息，表明发生了“同抢”。根据 TUP 协议规定，如果通道的 CIC 编号为本端主控的电路，则此 IAI/IAM 消息将被丢弃，本端继续去话呼叫；否则，本端放弃去话呼叫，转而处理来话呼叫。																
Dialing	通道正在准备进行去话呼叫，但尚未送出 IAI 或 IAM 消息。此时收到对端的来话呼叫请求，应予以响应。																
OffHook	通道正在准备进行去话呼叫，但尚未送出 IAI 或 IAM 消息。此时收到对端的来话呼叫请求，应予以响应。																
Locked	通道正在准备进行去话呼叫，但尚未送出 IAI 或 IAM 消息。此时收到对端的来话呼叫请求，应予以响应。																
Standby	接受来话呼叫。																
其它	拒绝。																
RcvRLG	驱动程序收到对端交换机的 RLG 消息（释放监护信号，Release-guard signal）																
RcvCLF	驱动程序收到对端交换机的 CLF 消息。当本事件发生时，如果通道处于 <ul style="list-style-type: none">• RcvPhoNum、Ringing 状态： 表明在来话呼叫的建立过程中，对端交换机（主叫方）取消了本次呼叫。 如果通道处于 Ringing 状态，如果配置项 RingToPending 设置为 1，驱动程序会设置 PendingReason 为 PEND_RemoteHangupOnRinging，并将通道迁移到 Pending 状态；否则，则驱动程序开始拆除本次来话呼叫• Release 状态中的 S_TUP_WaitCLF 子状态： 表明通道在拆线时收到了预期中的 CLF 消息，拆线过程完成• Pending 状态： 不管进入 Pending 状态的原因如何，在 Pending 状态收到 CLF 消息都将导致驱动程序自动发送 RLG 消息，并开始拆除连接• Connected 状态： 对于来话呼叫，在 Connected 状态收到 CLF 消息表明主叫用户先挂机。驱动程序会将 PendingReason 设置为 PEND_RemoteHangupOnTalking 后，然后将通道迁移到 Pending 状态，但不会发送 RLG 消息																
RcvACM	驱动程序收到对端交换机的 ACM 消息（被叫地址全）。在去话呼叫接续过程中，当应用程序调用 SsmAutoDial 函数启动一次去话呼叫任务后，驱动程序会向对端交换机发送 IAI、IAM 消息。如果被叫方交换机认为已经完成被叫号码等必要信息的接收，会向本端发送 ACM 消息。 驱动程序只在 Dialing 、 AppSetCallerID 状态响应本事件。当收到 ACM 消息后，驱动首先保存 ACM 消息，接着将 AutoDial 任务的进展设置为 DIAL_ECHOTONE，然后迁移到 WaitAnswer 状态。函数 SsmGetKB 可以用来获取收到的 ACM 的信息																
Answered	当驱动程序收到对端交换机的应答信号时触发本事件。去话呼叫时，当对端交换机处于向被叫用户振铃的状态时，如果被叫用户摘机，对端交换机会向本端发送应答信号。应答信号是指下列消息之一： <ul style="list-style-type: none">◆ ANU 消息（应答信号、无指示）◆ ANC（应答信号、计费）◆ ANN（应答信号、免费） 函数 SsmGetTupFlag 可以获取应答信号使用的具体消息。 本事件会导致通道从 WaitAnswer 状态迁移到 Connected 状态。在迁移到新的状态前，驱																

	<p>动程序还会自动进行下列操作：</p> <ul style="list-style-type: none"> ◆ 如果配置项 AlwaysEnableRxDtmf 设置为 0， DTMF 检测器 将被自动启动 ◆ 如果配置项 AlwaysDetectBargeIn 设置为 0， Barge-in 检测器 将被自动启动 ◆ 信号音检测器 将被清零并自动启动 ◆ 将 AutoDial 任务的进展设置为 DIAL_VOICE，任务结束，并向应用程序抛出 E_PROC_AutoDial 事件
RcvCBK	<p>驱动程序收到对端交换机的 CBK 消息。当本事件发生时，如果通道处于</p> <ul style="list-style-type: none"> • Connected 状态： 表明在通话过程中检测到远端用户先挂机（去话方向）。驱动程序将 PendingReason 设置为 PEND_CalleeHangupOnTalking，然后将通道状态迁移到 Pending • WaitAnswer 状态： 表明被叫用户拒绝了本次呼叫，AutoDial 任务结束。驱动程序执行下列操作： <ul style="list-style-type: none"> ◆ 将 AutoDial 任务的进展值设置为 DIAL_FAILURE，将 AutoDial 任务的失败原因设置为 ATDL_RcvCBK，并向应用程序抛出 E_PROC_AutoDial 事件 ◆ 将 PendingReason 设置为 PEND_CalleeHangupOnWaitRemotePickUp，然后将通道状态迁移到 Pending
RcvCCL	<p>驱动程序收到对端交换机的 CCL 消息，表明在 Connected 状态，远端用户先挂机（来话方向）。</p> <p>收到 RcvCCL 事件后，驱动程序会将 PendingReason 设置为 PEND_RemoteHangupOnTalking 后，然后将通道状态迁移到 Pending。如果应用程序在 Connected 状态调用函数 SsmSetCalleeHoldFlag 开启了“锁定主叫用户”功能，驱动程序会自动向对端交换机发送 OPR 消息，但状态保持在 Connected 状态</p>
T1Out	T1 定时器溢出。T1 设置为 60 秒，在通道迁移到 Locked 或 OffHook 状态时自动启动
T2Out	<p>T2 定时器溢出。T2 设置为 30 秒，在通道迁移到 Dialing 状态时自动启动。T2 溢出后驱动程序的操作为：</p> <ul style="list-style-type: none"> ◆ 设置 AutoDial 任务的进展值为 DIAL_FAILURE，失败原因设置为 ATDL_WaitDialAnsTimeout，并向应用程序抛出 E_PROC_AutoDial 事件 ◆ 将 PendingReason 设置为 PEND_AutoDialFailed，然后迁移到 Pending 状态
T4Out	<p>T4 定时器溢出。T4 在配置项 MaxWaitAutoDialAnswerTime 中设置，在通道迁移到 WaitAnswer 状态时自动启动。T4 溢出后驱动程序的操作为：</p> <ul style="list-style-type: none"> ◆ 设置 AutoDial 任务的进展值为 DIAL_NOANSWER，失败原因设置为 ATDL_WaitRemotePickupTimeout，并向应用程序抛出 E_PROC_AutoDial 事件 ◆ 将 PendingReason 设置为 PEND_AutoDialFailed，然后迁移到 Pending 状态
T5Out	<p>T5 定时器溢出。T5 设置为 20 秒，在通道迁移到 RcvPhoNum 状态的 S_TUP_WaitPrefix 或 S_TUP_WaitSAM 子状态时自动启动。T5 溢出后，驱动程序会自动向对端交换机发送 ADI 消息（表示号码不足），然后迁移到 Release 状态</p>
PhoNumOK	<p>当发生下列事件时，驱动程序会触发本事件：</p> <ul style="list-style-type: none"> ◆ 驱动程序按照预设的收号规则，完成了全部被叫号码等信息的接收 ◆ 通道处于 RcvPhoNum 状态的 _TUP_WaitGSM 子状态时，设定值为 5 秒的定时器溢出，驱动程序不再等待对端交换机的 GSM 消息 ◆ 通道处于 RcvPhoNum 状态的 _TUP_WaitGSM 子状态时，收到对端交换机的 GSM 消息 <p>本事件发生后，如果应用程序通过函数 SsmEnableAutoSendKB 或配置项 AutoSendACM 进行开启了“自动应答来话呼叫”功能，驱动程序会自动向对端交换机发送 ACM 消息，然后迁移到 Ringing 状态。配置项 DefaultACM 可用来定制驱动程序自动发送的 ACM 消息中指示被叫用户状态的消息表示语字段，配置项 RcvPhoNumCfgLen 可以用来设置驱动程序保存被叫号码的方式，函数 SsmSetKB 可以用来设置 ACM 消息的后向呼叫表示语中的计费表示语。如果没有开启“自动应答来话呼叫”功能，驱动程序会将 PendingReason 设置为 PEND_WaitBckStpMsg，然后通道迁移到 Suspend 状态，将控制权交给应用程序，由应用程序决定是否接受本次呼叫</p>
Error	<p>当驱动程序检测到下列情况之一时触发本事件：</p> <ul style="list-style-type: none"> ➢ 本通道所在数字中继线的 0 时隙上的同步信号丢失（例如线路断开、时钟配置错误等） ➢ TUP 协议栈与 SS7 Server 的之间 TCP/IP 连接断开 ➢ 收到 SS7 Server 的信令不可用消息 <p>当本事件发生时，驱动程序的操作与通道的状态有关，具体为：</p>

	<ul style="list-style-type: none"> ➤ Dialing、AppSetCallerID 或 WaitAnswer 状态：驱动程序设置 AutoDial 任务的进展值为 DIAL_FAILURE，失败原因设置为 ATDL_PcmSyncLos 或 ATDL_Mtp3Unusable，并向应用程序抛出 E PROC AutoDial 事件 ➤ Dialing、AppSetCallerID、WaitAnswer、Connected 或 Pending 状态：将 PendingReason 设置为 PEND_PcmSyncLos 或 PEND_SsxUnusable，然后迁移到 Pending 状态
CallFailure	<p>在去话呼叫过程中，如果呼叫失败，驱动程序会触发本事件。当发生下列事件时，去话呼叫失败：</p> <ul style="list-style-type: none"> ➤ 收到对端交换机的呼叫失败消息。呼叫失败消息包括： <ul style="list-style-type: none"> ● CFL 消息（呼叫失败）。驱动程序的操作为： <ul style="list-style-type: none"> ✧ 设置 AutoDial 任务的结果为 DIAL_FAILURE，失败原因设置为 ATDL_RcvCFL，并向应用程序抛出 E PROC AutoDial 事件 ✧ 驱动程序在设置了 PendingReason 后迁移到 Pending 状态。如果通道处于 WaitAnswer 状态，PendingReason 被设置为 PEND_CalleeHangupOnWaitRemotePickUp；如果通道处于 Dialing 状态，PendingReason 被设置为 PEND_AutoDialFailed ● SSB、SLB、STB 消息（被叫用户忙）。驱动程序的操作为： <ul style="list-style-type: none"> ✧ 驱动程序设置 AutoDial 的结果为 DIAL_BUSYTONE，失败原因分别设置为 ATDL_RcvSSB、ATDL_RcvSLB 或 ATDL_RcvSTB，并向应用程序抛出 E PROC AutoDial 事件 ✧ 将 PendingReason 设置为 PEND_AutoDialFailed 后迁移到 Pending 状态 ● UNN 消息（被叫用户号码为空号）。驱动程序的操作为： <ul style="list-style-type: none"> ✧ 设置 AutoDial 的结果为 DIAL_INVALID_PHONUM，失败原因设置为 ATDL_RcvUNN，并向应用程序抛出 E PROC AutoDial 事件 ✧ 将 PendingReason 设置为 PEND_AutoDialFailed 后迁移到 Pending 状态 ● SEC、CGC、NNC、LOS、SST、ACB、DPN、EUM、ADI 消息（其它的呼叫错误）。驱动程序的操作为： <ul style="list-style-type: none"> ✧ 设置 AutoDial 的结果为 DIAL_FAILURE，失败原因分别设置为 ATDL_RcvSEC、ATDL_RcvCGC、ATDL_RcvNNC、ATDL_RcvLOS、ATDL_RcvSST、ATDL_RcvACB、ATDL_RcvDPN、ATDL_RcvEUM、ATDL_RcvADI，并向应用程序抛出 E PROC AutoDial 事件 ✧ 将 PendingReason 设置为 PEND_AutoDialFailed 后迁移到 Pending 状态 ➤ 发生了 CicReset 事件。 ➤ 发生了 Blocking 事件。 ➤ 发生了 Error 事件。

状态转移图中，各状态标识的说明如下表所示：

状态标识	描述
Unusable	通道不可用
Reseting	电路复位
BlockedByPBX	对端闭塞状态。对端交换机进行维护时，会向本端发送闭塞消息，以阻止本端对它发起电话呼叫。当通道处于对端闭塞状态时，驱动程序不支持应用程序发出的去话呼叫指令，但仍然可以接受和处理来话呼叫
BlockedByApp	本端闭塞状态。此时，应用程序不能进行电话呼出操作。本端闭塞通常用于系统维护功能。应用程序可以通过调用函数 SsmBlockLocalCh 或 SsmBlockLocalPCM 将通道闭塞，以通知驱动程序阻止电话呼出。 注意： 在本端闭塞状态下，通道仍然可以接受和处理来话呼叫
Standby	空闲状态。驱动程序初始化成功后，如果通道正常复位，应该进入本状态。应用程序可以在本状态调用 SsmAutoDial 函数发起去话呼叫业务。 当通道因为呼叫结束而迁移到空闲状态时，如果配置项 AutoClearCallerIdBufOnHangup 设置为 1，驱动程序会自动清空主叫号码缓冲区
OffHook	摘机状态。应用程序可以在本状态调用 SsmAutoDial 函数发起去话呼叫业务。通道进入 OffHook 状态时，驱动程序启动定时器 T1。如果应用程序在 T1 溢出之前没有调用 SsmAutoDial 函数，驱动程序就会将此通道进行回收
Locked	呼出锁定状态。应用程序可以在本状态调用 SsmAutoDial 函数发起去话呼叫业务。通道进入本

	状态时会启动定时器 T1。如果应用程序在 T1 溢出之前没有调用 SsmPickup 或 SsmAutoDial 函数，驱动程序就会将此通道进行回收
Pending	未决状态
RcvPhoNum	<p>来话接续过程中的内部状态，由 S_TUP_WaitSAM、S_TUP_WaitPrefix 和 S_TUP_WaitGSM 子状态组成。</p> <p>驱动程序根据预设的收号规则对收到的被叫号码进行判断，如果符合收号规则，则通道会迁移到后续状态。有关收号规则的详细信息请参见本章中“设置来话呼叫的收号规则”部分内容。</p> <p>如果对端交换机发送的 IAM/IAI 消息中包含的被叫号码信息不足，通道会停留在 S_TUP_WaitSAM 子状态，每当驱动程序收到对端交换机的 SAO 或 SAM 消息，驱动程序都会将其保存到内部缓冲区中，并向应用程序抛出 E_CHG_RxPhoNumBuf 事件。</p> <p>当驱动程序按照收号规则收齐了全部被叫号码后，如果需要接收主叫方的信息（如主叫方号码、主叫用户类别等），但对端交换机尚未提供，驱动程序会向对端交换机发送 GRQ 消息，通道会迁移到 S_TUP_WaitGSM 子状态。配置项 ReqTypeIndicators 用于定制发送给对端交换机的 GRQ 消息中的请求类型指示语字段。</p> <p>当通道收到的被叫号码不符合收号规则时，驱动自动发送 REL（带空号原因值）消息，然后通道迁移到 Release 状态。</p> <p>在 RcvPhoNum 状态，应用程序可以：</p> <ul style="list-style-type: none"> ◆ 调用 SsmHangup/SsmHangupEx 函数拒绝本次呼叫。驱动程序会发送 REL 消息，配置项 DefaultHangupRELInd 或 函数 SsmSetIsupFlag（携带 ISUP_REL_DENY_SetToOther 参数）可用来设 REL 消息的拒绝原因值。
Ringing	本端振铃状态。应用程序可以用 SsmPickup 函数接受呼叫，也可以调用 SsmHangup 函数拒绝呼叫
Dialing	<p>来话呼叫过程中的状态，由 S_TUP_WaitDialAnswer 和 S_TUP_WaitSetCallerID 子状态组成。</p> <p>应用程序调用 SsmAutoDial 或 SsmAutoDialEx 函数会引发驱动程序向对端交换机发送 IAM 消息，通道会迁移到 S_TUP_WaitDialAnswer 子状态。在 S_TUP_WaitDialAnswer 子状态，如果对端交换机希望接收主叫信息，但 IAM/IAI 消息中没有提供，会向本端发送 GRQ 消息。驱动程序收到 GRQ 消息后，</p> <ul style="list-style-type: none"> ➢ 如果本通道的去话方向的主叫号码缓冲区为空，并且收到的 GRQ 消息表明对端交换机向本端请求主叫号码，则 <ul style="list-style-type: none"> ◆ 如果配置项 AutoSendGSM 设置为 1，驱动程序会自动发送 GSM 消息 ◆ 如果配置项 AutoSendGSM 设置为 0，通道会迁移到 S_TUP_WaitSetCallerID 子状态，等待应用程序来设置主叫号码等信息，同时启动一个设定值为 2 秒的定时器。如果定时器溢出时应用程序仍然没有调用函数 SsmSetTxCallerId，驱动程序会重新迁移到 S_TUP_WaitDialAnswer 子状态 ➢ 否则，驱动程序主叫号码缓冲区中的字符串作为主叫号码参数，自动发送 GSM 消息
WaitAnswer	等待被叫用户摘机状态，只适用于去话呼叫。此时，被叫方应该听到振铃音。如果本端是坐席通道通过数字中继通道进行电话呼出，则应向坐席通道发送回铃音信号
Connected	连接状态，呼叫双方可以进行通话。当通道迁移到 Connected 时，
	<ul style="list-style-type: none"> • 如果配置项 AlwaysEnableRxDtmf 设置为 0，DTMF 检测器 将被启动 • 如果配置项 AlwaysDetectBargin 设置为 0，Barge-in 检测器 将被启动 • 信号音检测器 将被清零并启动
Release	拆线状态，由 S_TUP_WaitCLF 和 S_TUP_WaitRLG 子状态组成

上表中，状态的值与宏定义均在文件 ShpA3Api.h 中。

状态转移图中，各状态标识在头文件 ShpA3Api.h 中的声明如下表所示：

状态标识	ShpA3Api.h 中的值	ShpA3Api.h 中的宏定义
Standby	0	S_CALL_STANDBY
OffHook	1	S_CALL_PICKUPED
Ringing	2	S_CALL_RINGING

Connected	3	S_CALL_TALKING
Pending	7	S_CALL_PENDING
WaitAnswer	9	S_CALL_WAIT_REMOTE_PICKUP
BlockedByPBX	10	S_CALL_RemoteBlock
Unusable	11	S_CALL_UNAVAILABLE
Locked	12	S_CALL_LOCKED
BlockedByApp	20	S_CALL_LocalBlock
Reseting	70	S_TUP_WaitPcmReset
RcvPhoNum	71	S_TUP_WaitSAM
	72	S_TUP_WaitGSM
	74	S_TUP_WaitPrefix
Dialing	75	S_TUP_WaitDialAnswer
	77	S_TUP_WaitSetCallerID
Release	73	S_TUP_WaitCLF
	76	S_TUP_WaitRLG

上表中，状态的值与宏定义均在文件 ShpA3Api.h 中。

状态转移图中，ChUsable 为驱动程序内部的收号规则判断模块，它的输出描述如下表所示：

模块输出	判决条件
Unusable	如果 Error 事件保持有效，通道不可用
BlockedByApp	Blocking 事件保持有效，并且通道被应用程序闭塞
BlockedByPBX	Blocking 事件保持有效，并且通道被对端交换机闭塞
Standby	Error 事件保持无效

注意：在此模块，驱动程序还要做下列清理工作：

- 信号音检测器将被清零并自动关闭；
- 如果应用程序启动了 WaitDtmf 任务，该任务将被自动终止；
- 如果应用程序启动了 DTMF 发送任务，该任务将被自动终止；
- DTMF 检测器的缓冲区将被自动清空，如果配置项 [AlwaysEnableRxDtmf](#) 设置为 0，DTMF 检测器将被自动关闭；
- 如果配置项 [AlwaysDetectBargeIn](#) 设置为 0，Barge-in 检测器将被自动关闭；
- 如果配置项 [AutoClearCallerIdBufOnHangup](#) 设置为 1，存放主叫号码的缓冲区将被自动清空。

1.16.4.5.2 通道的闭塞与解除

1.16.4.5.2.1 通道闭塞的概念

(1) 通道闭塞

通道被闭塞后，相关通道的呼出（去话呼叫）功能被禁止，但来话接续正常进行。

(2) 本地闭塞

相关通道因本方应用程序的指令（函数调用）而阻断本方指定通道的去话业务。

(3) 对端闭塞

对端交换机通过向本方发送特定的电路/电路群闭塞消息以阻断本方的去话业务（即阻断对端交换机的来话业务）。对端交换机发送的闭塞消息由驱动程序自动处理。

(4) 闭塞对端

应用程序通过向对端交换机发送特定的电路/电路群闭塞消息以阻止对端交换机的去话业务（即阻断本端的来话业务）。

1.16.4.5.2.2 通道闭塞的作用

通道闭塞只适用于数字中继通道。当应用程序因维护等原因需要停机时，需要实施下列 4 个步骤以保证正常的退出顺序：

- (1) 闭塞本端所有通道，以阻止本端新的去话业务。
- (2) 闭塞对端所有通道，以阻止对端新的去话业务（即本端新的来话业务）。
- (3) 等待当前呼叫接续结束。
- (4) 程序退出运行。

当通道被闭塞时，驱动程序不再处理新的去话业务（即自动呼出接续相关函数都将返回失败）。

1.16.4.5.2.3 通道闭塞的途径

一个通道可能因下述原因进入闭塞状态：

- (1) 收到对端交换机闭塞消息。对于 TUP 协议，通道可能因为收到对端交换机发送的电路闭塞信号 BLO、电路群闭塞信号 SGB、HGB 或 MGB 而闭塞。
- (2) 应用程序调用闭塞本地通道的函数 `SsmBlockLocalCh()` 和 / 或闭塞本地 PCM 的函数 `SsmBlockLocalPCM()`。

1.16.4.5.2.4 实现通道闭塞的函数列表

实现通道闭塞的函数及其类别如下表：

闭塞类别	函数名称	输出事件
闭塞本端	SsmBlockLocalCh	
	SsmUnblockLocalCh	
	SsmQueryLocalChBlockState	
	SsmBlockLocalPCM	
	SsmUnblockLocalPCM	
	SsmQueryLocalPCMBLOCKState	
闭塞对端	SsmQueryOpBlockRemoteCh	
	SsmBlockRemoteCh	收到对端的证实消息（BLA 或 UBA）后抛出 E_CHG_RemoteChBlock 事件
	SsmUnblockRemoteCh	
	SsmGetRemoteChBlockStatus	收到对端的群证实消息（MBA/HBA/SBA 或 MUA/HUA/SUA）后抛出 E_CHG_RemoteChBlock 事件
	SsmBlockRemotePCM	
	SsmUnblockRemotePCM	

		SsmGetRemotePCMBlockStatus	
--	--	--------------------------------------------	--

注：闭塞对端时，本端驱动程序向对端交换机发送的闭塞电路群消息中的范围字段可以通过配置项 [SendGRMRange](#) 进行设置。

1.16.4.5.3 电路的复原

当发生下列事件时，TUP 协议的状态机会自动向对端交换机发送电路群复原消息，将所有的 TUP 通道复原：

- ◆ SS7 信令服务器检测到 MTP3 层提供的服务重新可用时。
- ◆ 数字中继线的同步信号恢复时。

电路群复原消息中的范围字段可以通过配置项 [SendGRMRange](#) 进行设置。

1.16.4.5.4 “锁定主叫用户”功能

来话呼叫时，当呼叫双方进入通话后，如果主叫方先挂机，对端交换机会向本端发送 CCL 消息。驱动程序收到 CCL 消息后，如果应用程序在此之前调用 [SsmSetCalleeHoldFlag](#) 函数设置了“锁定主叫用户”功能，驱动将向对端交换机回应 OPR 消息。

1.16.4.5.5 TUP 协议的配置

使用 TUP 协议时，相关的配置项如下表所示：

配置节	配置项	描述
[BoardId=x]	PcmNumber	设置板卡上数字中继线的总数，以及每条数字中继线的信令类型、时钟的工作模式、线路的传输类型
	PcmSSx	
	PcmClockMode	
	PcmLinkType	
	UseTS16AsCircuit Ss7SignalingTS	设置数字中继线内是否包含 SS7 信令链路，以及具体提供信令链路的时隙编号
[PcmInfo]	TotalPcm	设置本机数字中继线的总数，以及建立数字中继线的逻辑编号与物理编号之间的映射关系
	Pcm	
[SS7]	AutoHandleTup	设置是否使用驱动程序提供的呼叫状态机

	Ss7ServerIP	设置 SS7 信令服务器和本机的 IP 地址
	SecondServerIP	
	LocalIP	
[TUP]	MaxWaitAutoDialAnswerTime	去话呼叫：设置工作参数
	CalloutCallerId	
	SetSTSsignal	
	AutoSendGSM	
	ConnectReqMsg	
	AddressSignal	
	CalloutIAM_MsgPntr	
	CalloutIAM_CAT	
	CallingIndicatorBit	
	ReqTypeIndicators	
	OriginalCalleeAddrInd	去话呼叫：定制向对端交换机发送的 IAM 消息
	CallerAddrInd	
	DefaultACM	
	AutoSendACM	
	ReqTypeIndicators	
	RingToPending	
	RcvPhoNumCfgLen	
	DefaultRcvPhoNumLen	来话呼叫：设置收号规则
	DefaultRcvCallerID	
	MaxPhoNumRule	
	Rule	
		来话呼叫：设置拒绝来话呼叫时使用的消息类型

	HangupRingSendCBK	
--	-----------------------------------	--

注：**XXXX** 字样为必须配置的项目，**XXXX** 为可选的配置项目

1.16.4.6 ISUP 协议的编程与应用

1.16.4.6.1 ISUP 协议的配置

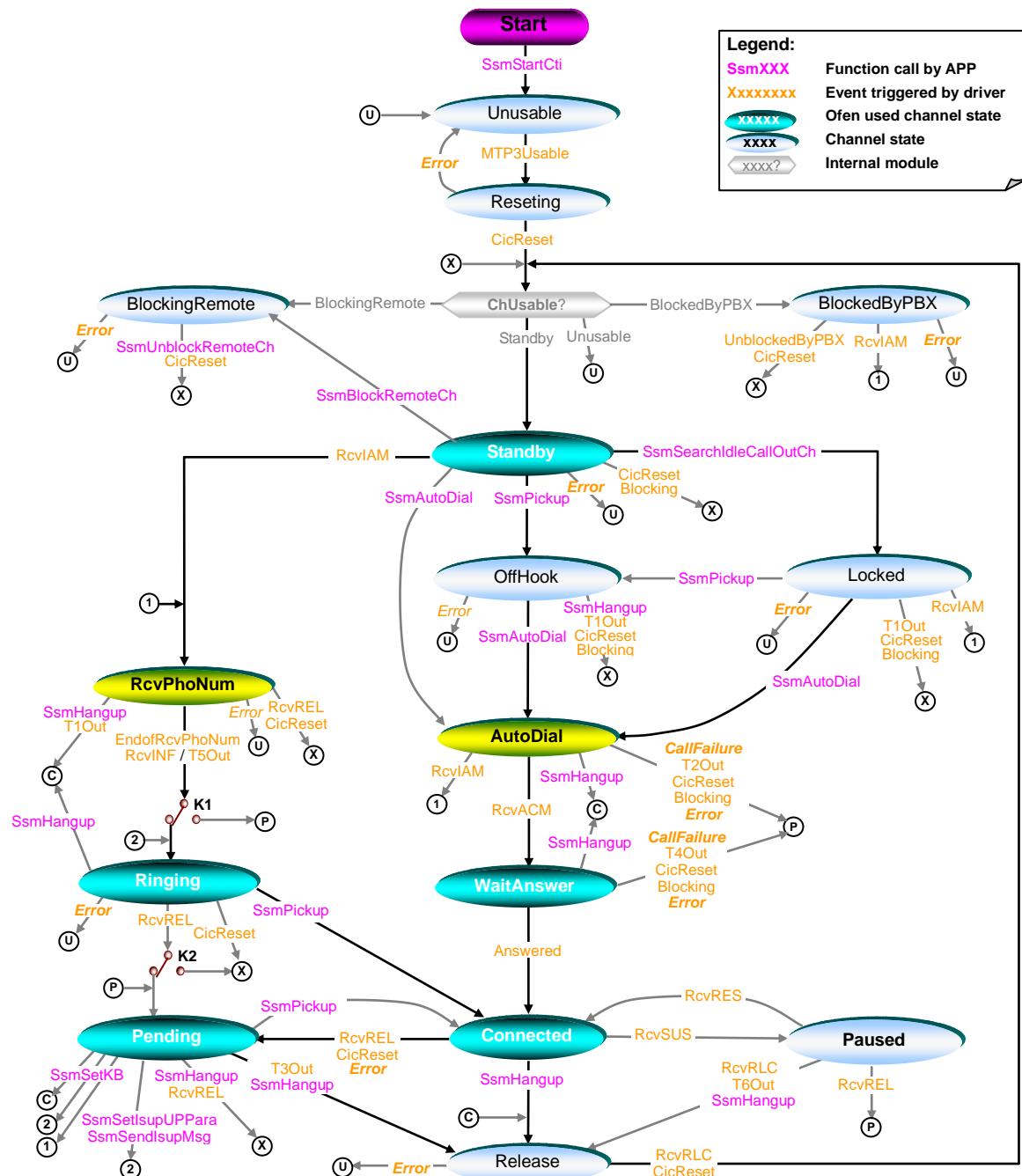
使用 ISUP 协议时，相关的配置项如下表所示：

配置节	配置项	描述
[BoardId=x]	PcmNumber	设置板卡上数字中继线的总数，以及每条数字中继线的信令类型、时钟的工作模式、线路的传输类型
	PcmSSx	
	PcmClockMode	
	PcmLinkType	
	UseTS16AsCircuit Ss7SignalingTS	设置数字中继线内是否包含 SS7 信令链路，以及具体提供信令链路的时隙编号
[PcmInfo]	TotalPcm	设置本机数字中继线的总数，以及建立数字中继线的逻辑编号与物理编号之间的映射关系
	Pcm	
[SS7]	TotalsupPcm	设置本机中使用 ISUP 协议的数字中继线
	IsupPcm	
	AutoHandleIsup	
	Ss7ServerIP	
	SecondServerIP	
	LocalIP	设置 SS7 信令服务器和本机的 IP 地址
[ISUP]	MaxWaitAutoDialAnswerTime	
	CalloutCallerId	去话呼叫：设置工作参数
	SetSTSsignal	
	DefaultNatureOfConnectionInd	去话呼叫：定制向对端交换机发送的 IAM 消息

DefaultIAM_ForceForwardCallInd	
DefaultIAM_CAT	
DefaultIAM_TransmissionMediumRequirement	
DefaultIAM_CalleeParam	
DefaultIAM_CallerParam	
DefaultIAM_OriginalCalleeParam	
DefaultIAM_RedirectingNumber	
bSubscriberSI, SubscriberSI	
bOptionalFCI, OptionalFCI	
Usr2UsrInfo	
AutoSendINF	去话呼叫：设置应答 INF 消息的模式
RingToPending	
AutoSendACM	
DefaultCalledPickupMsg	
DefaultBackwardCallInd	来话呼叫：设置工作参数
DefaultHangupRELInd	
DefaultCauseInd	
SaveRGNTo1stPhoNumStr	
DefaultRcvPhoNumLen	
DefaultRcvCallerID	
MaxPhoNumRule	来话呼叫：设置收号规则
Rule	
AddressSignal	设置 ISUP 协议状态机的工作参数

注：**XXXX** 字样为必须配置的项目，**XXXX** 为可选的配置项目

1.16.4.6.2 ISUP 通道的状态转移



上图中，驱动程序产生的内部事件如下表所述：

事件名称	描述
Mtp3Usable	驱动程序收到 SS7 Server 的信令可用消息后，会向所有通道的状态机发送本事件，使通道从 Unusable 状态迁移到 Resetting 状态，开始电路复原
CicReset	当驱动程序收到电路复原消息（GRA、GRS 或 RSC）时触发本事件。 本事件发生后，如果通道处于 AutoDial 或 WaitAnswer 状态，去话呼叫失败，驱动程序会向应用程序抛出 E_PROC_AutoDial 事件，函数 SsmGetAutoDialFailureReason 可以用来获取

	AutoDial 失败的具体原因														
Blocking	本通道被本端闭塞或被对端闭塞： 对端闭塞：驱动程序收到对端交换机的群闭塞消息（CGB）或电路闭塞消息（BLO）														
UnblockedByPBX	解除对本端电路的闭塞：驱动程序收到对端交换机的 CGU、UBL 消息时触发本事件														
RcvIAM	<p>驱动程序收到对端交换机的来话呼叫建立请求消息（IAM）。 驱动程序会自动保存最新收到的 IAM 原始消息，函数 SsmGetIsupUPPara（携带 C_ISUP_IAM 参数）可以用来获取此原始消息，SsmGetRedirectionInfoReason、SsmGetKA 可以分别获取 IAM 消息里的转呼原因值字段和“主叫方类别”参数。 驱动程序是否接受此来话呼叫请求，取决于通道的状态：</p> <table border="1"> <thead> <tr> <th>通道状态</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>BlockedByPBX</td> <td>在本端闭塞状态，虽然通道不能进行去话呼叫，但依然可以接受来话呼叫，因此，驱动程序会对 IAM 消息予以响应</td> </tr> <tr> <td>AutoDial</td> <td>如果在去话呼叫接续过程中收到对端交换机的呼叫建立请求消息 IAM，表明发生“同抢”。根据 ISUP 协议规定，如果通道的 CIC 编号为本端主控的电路，则 IAM 消息将被丢弃，本端继续去话呼叫；否则，本端放弃去话呼叫，转而处理来话呼叫，本次去话呼叫失败，驱动程序会向应用程序抛出 E_PROC_AutoDial 事件，函数 SsmGetAutoDialFailureReason 可以用来获取 AutoDial 失败的具体原因</td> </tr> <tr> <td>OffHook</td> <td>通道正在准备进行去话呼叫，但尚未送出 IAM 消息。此时收到对端的来话呼叫请求，驱动程序会对 IAM 消息予以响应</td> </tr> <tr> <td>Locked</td> <td>接受来话呼叫</td> </tr> <tr> <td>Standby</td> <td>对此 IAM 消息予以拒绝</td> </tr> <tr> <td>其它</td> <td>对此 IAM 消息予以拒绝</td> </tr> </tbody> </table> <p>如果驱动程序接受来话呼叫，在将 IAM 消息中的被叫号码信息保存到缓冲区后，向应用程序抛出 E_CHG_RxPhoNumBuf 事件，然后将通道迁移到 RcvPhoNum 状态</p>	通道状态	描述	BlockedByPBX	在本端闭塞状态，虽然通道不能进行去话呼叫，但依然可以接受来话呼叫，因此，驱动程序会对 IAM 消息予以响应	AutoDial	如果在去话呼叫接续过程中收到对端交换机的呼叫建立请求消息 IAM，表明发生“同抢”。根据 ISUP 协议规定，如果通道的 CIC 编号为本端主控的电路，则 IAM 消息将被丢弃，本端继续去话呼叫；否则，本端放弃去话呼叫，转而处理来话呼叫，本次去话呼叫失败，驱动程序会向应用程序抛出 E_PROC_AutoDial 事件，函数 SsmGetAutoDialFailureReason 可以用来获取 AutoDial 失败的具体原因	OffHook	通道正在准备进行去话呼叫，但尚未送出 IAM 消息。此时收到对端的来话呼叫请求，驱动程序会对 IAM 消息予以响应	Locked	接受来话呼叫	Standby	对此 IAM 消息予以拒绝	其它	对此 IAM 消息予以拒绝
通道状态	描述														
BlockedByPBX	在本端闭塞状态，虽然通道不能进行去话呼叫，但依然可以接受来话呼叫，因此，驱动程序会对 IAM 消息予以响应														
AutoDial	如果在去话呼叫接续过程中收到对端交换机的呼叫建立请求消息 IAM，表明发生“同抢”。根据 ISUP 协议规定，如果通道的 CIC 编号为本端主控的电路，则 IAM 消息将被丢弃，本端继续去话呼叫；否则，本端放弃去话呼叫，转而处理来话呼叫，本次去话呼叫失败，驱动程序会向应用程序抛出 E_PROC_AutoDial 事件，函数 SsmGetAutoDialFailureReason 可以用来获取 AutoDial 失败的具体原因														
OffHook	通道正在准备进行去话呼叫，但尚未送出 IAM 消息。此时收到对端的来话呼叫请求，驱动程序会对 IAM 消息予以响应														
Locked	接受来话呼叫														
Standby	对此 IAM 消息予以拒绝														
其它	对此 IAM 消息予以拒绝														
EnfOfRcvPhoNum	驱动程序根据预设的收号规则完成全部被叫号码的接收。有关收号规则的详细信息请参见本章中“ 设置来话呼叫的收号规则 ”部分内容														
RcvINF	驱动程序收到对端交换机的 INF 消息。INF 消息包含了主叫方的相关信息														
RcvREL	<p>驱动程序收到对端交换机的拆线消息（REL）。函数 SsmGetReleaseReason 可以获取具体的释放原因。</p> <p>当本事件发生时，如果通道处于：</p> <ul style="list-style-type: none"> RcvPhoNum 状态： 表明在来话呼叫的建立过程中，对端交换机（主叫方）取消了本次呼叫 Ringing 状态： 表明在来话呼叫的建立过程中，当本端处于振铃状态时，对端交换机取消了本次呼叫。应用程序可以通过对 K2 开关的控制来决定是否干预通道的状态迁移。K2 由配置项 RingToPending 决定。如果配置项 RingToPending 设置为 1，驱动程序会设置 PendingReason 为 PEND_RemoteHangupOnRinging，然后将通道迁移到 Pending 状态，以等待应用程序的进一步处理；否则，驱动程序开始拆除本次来话呼叫 Connected 状态： 表明主叫用户已经挂机。驱动程序会将 PendingReason 设置为 PEND_RemoteHangupOnConnected，然后将通道状态会迁移到 Pending 														
RcvRLC	驱动程序收到对端交换机的 RLC 消息，拆线过程结束														
RcvACM	<p>驱动程序收到对端交换机的 ACM 消息（被叫地址全）。</p> <p>在去话呼叫接续过程中，当应用程序调用 SsmAutoDial 函数启动一次去话呼叫任务后，驱动程序会向对端交换机发送 IAM 消息。当对端交换机完成被叫号码等必要信息的接收后，会向本端发送 ACM 消息，指示被叫用户的空闲状态。驱动程序收到 ACM 消息后，会将原始的 ACM 消息保存到内部缓冲区中，函数 SsmGetIsupUPPara（携带 C_ISUP_ACN 参数）可以用来获取此消息，然后将 AutoDial 任务的进展设置为 DIAL_ECHOTONE，并将通道迁移到 WaitAnswer 状态。函数 SsmGetKB 可以用来获取此 ACM 消息中后向呼叫表示语字段（共 2 个字节）中的低 8 位比特。</p> <p>对端交换机在发送了 ACM 消息后，可能会继续发送 CPG 消息（呼叫进展）。函数 SsmGetCpg 可以取出原始的 CPG 消息。</p>														
Answered	在去话呼叫过程中，对端交换机完成被叫号码的接收后，如果被叫用户处于空闲状态，会向被叫用户发出振铃信号。如果被叫用户摘机，对端交换机会向本端发送应答消息（ANM）。														

	驱动程序收到 ANM 消息后, 将 AutoDial 任务的进展设置为 DIAL_VOICE, AutoDial 任务终止, 并向应用程序抛出 E_PROC_AutoDial 事件
RcvSUS	收到对端交换机的 SUS (暂停) 消息
RcvRES	收到对端交换机的 REC (恢复) 消息
CallFailure	在去话呼叫过程中, 当收到对端交换机的 REL 拆线等消息时, 去话呼叫失败, 驱动程序会向应用程序抛出 E_PROC_AutoDial 事件, 函数 SsmGetAutoDialFailureReason 可以用来获取 AutoDial 失败的具体原因
T1Out	T1 定时器溢出。T1 设置为 60 秒, 在通道迁移到 Locked 或 OffHook 状态时自动启动
T2Out	T2 定时器溢出。T2 设置为 25 秒, 在通道迁移到 AutoDial 状态中的 S_ISUP_WaitDialAnswer 子状态时自动启动。如果 T2 溢出, 本次去话呼叫失败, 驱动程序会向应用程序抛出 E_PROC_AutoDial 事件, 函数 SsmGetAutoDialFailureReason 可以用来获取 AutoDial 失败的具体原因
T3Out	T3 定时器溢出。通道进入 Pending 状态时启动, 设定值为 60 秒
T4Out	T4 在配置项 MaxWaitAutoDialAnswerTime 中设置, 在通道迁移到 WaitAnswer 状态时自动启动。如果 T4 溢出, 驱动程序会向应用程序抛出 E_PROC_AutoDial 事件, 函数 SsmGetAutoDialFailureReason 可以用来获取 AutoDial 失败的具体原因
T5Out	T5 定时器溢出。T5 设置为 10 秒, 在通道迁移到 RcvPhoNum 状态的 S_ISUP_WaitINF 子状态时自动启动。T5 溢出后, 驱动程序不再等待对端交换机的 INF 消息
T6Out	T6 定时器溢出。T6 设置为 180 秒, 在通道迁移到 Suspend 状态时自动启动
Error	当驱动程序检测到下列情况之一时触发本事件: <ul style="list-style-type: none"> ◆ 本通道所在数字中继线的 0 时隙上的同步信号丢失 (例如线路断开、时钟配置错误等) ◆ ISUP 协议栈与 SS7 Server 之间的 TCP/IP 连接断开 ◆ 收到 SS7 Server 的信令不可用消息 当本事件发生时, 如果通道处于 AutoDial 或 WaitAnswer 状态, 去话呼叫失败, 驱动程序会向应用程序抛出 E_PROC_AutoDial 事件, 函数 SsmGetAutoDialFailureReason 可以用来获取 AutoDial 失败的具体原因

各状态标识的说明如下表所示:

状态标识	描述
Unusable	“不可用”状态
Reseting	“电路复位”状态
BlockingRemote	“闭塞对端交换机”状态。在 BlockingRemote 状态, 对端交换机不能发起去话呼叫, 但本端可以发起去话呼叫
BlockedByPBX	“被对端交换机闭塞”状态。对端交换机进行维护时, 会向本端发送闭塞消息, 以阻止本端对它发起电话呼叫。当通道处于对端闭塞状态时, 驱动程序不支持应用程序发出的去话呼叫指令, 但仍然可以接受和处理来话呼叫
Standby	“空闲”状态。驱动程序初始化成功后, 如果通道正常复位, 应该进入本状态。 应用程序可以在本状态调用 SsmAutoDial 函数发起去话呼叫业务。当通道因为呼叫结束而迁移到空闲状态时, 如果配置项 AutoClearCallerIdBufOnHangup 设置为 1, 驱动程序会自动清空主叫号码缓冲区
Locked	“呼出锁定”状态。应用程序可以在本状态直接调用 SsmAutoDial 函数发起去话呼叫业务。 通道迁移到本状态时, 驱动程序会启动定时器 T1。如果应用程序在 T1 溢出之前没有调用 SsmPickup 或 SsmAutoDial 函数, 驱动程序就会将此通道进行回收。T1 定时器的设定值为 60 秒
OffHook	“摘机”状态。应用程序可以在本状态调用 SsmAutoDial 函数发起去话呼叫业务。 通道进入 OffHook 状态时, 驱动程序会启动定时器 T1。如果应用程序在 T1 溢出之前没有调用 SsmAutoDial 函数, 驱动程序就会将此通道进行回收
AutoDial	去话接续过程中的内部状态, 由 S_ISUP_WaitDialAnswer 状态和 S_ISUP_WaitSetCallerID 状态组成。 应用程序调用 SsmAutoDial 或 SsmAutoDialEx 函数后会引发驱动程序向对端交换机发送 IAM

	<p>消息，并会迁移到 S_ISUP_WaitDialAnswer 子状态。在 S_ISUP_WaitDialAnswer 子状态，如果对端交换机希望接收主叫信息，但 IAM 消息中没有提供，会向本端发送 INR 消息。驱动程序收到 INR 消息后，</p> <ul style="list-style-type: none"> ◆ 如果驱动程序内部的去话方向的主叫号码缓冲区不为空，驱动程序会将该主叫号码信息装配到 INF 消息中，并自动将 INF 消息回送给对端交换机，同时保持状态不变 ◆ 如果驱动程序内部的去话方向的主叫号码缓冲区为空，并且配置项 AutoSendINF 设置为 1，驱动程序以配置项 CalloutCallerId 或函数 SsmSetTxCallerId 设置的主叫信息作为参数，自动应答 INF 消息，并保持状态不变 ◆ 如果驱动程序内部的去话方向的主叫号码缓冲区为空，并且配置项 AutoSendINF 设置为 0，通道会迁移到 S_ISUP_WaitSetCallerID 子状态，同时启动 T3 定时器。在 S_ISUP_WaitSetCallerID 子状态，如果应用程序调用了 SsmSetTxCallerId 函数，驱动程序会将设置的主叫号码信息装配到 INF 消息中，并将 INF 消息回送给对端交换机。如果直到 T3 定时器溢出时，应用程序仍然没有调用 SsmSetTxCallerId 函数，驱动程序会以配置项 CalloutCallerId 的设定值作为参数，自动向对端发送 INF 消息，然后将通道迁移到 S_ISUP_WaitDialAnswer 子状态
WaitAnswer	<p>“等待被叫用户摘机”状态，只适用于去话呼叫。此时，被叫方应该听到振铃音。如果本端是坐席通道通过数字中继通道进行电话呼出，则应向坐席通道发送回铃音信号</p>
Pending	<p>“挂起”状态。函数 SsmGetPendingReason 可用于获取具体的挂起原因。</p> <p>➤ 如果 PendingReason 为 PEND_WaitBckStpMsg，表明来话呼叫的接续过程已经完成，正等待应用程序接受或拒绝来话呼叫。应用程序可以：</p> <ul style="list-style-type: none"> ◆ 调用 SsmSetKB 函数接受或拒绝来话呼叫 <ul style="list-style-type: none"> ◆ 如果 KB=1/6/7：驱动程序会向对端交换机发送 ACM 消息，接受来话呼叫。如果应用程序在调用 SsmSetKB 函数之前已经调用 SsmSetIsupUPPara 函数(携带 C_ISUP_ACM 参数)向驱动程序提交了一个自行构建的 ACM 消息，该 ACM 消息将被发送给对端交换机；否则，驱动程序会使用自动构建一个 ACM 消息。发送完 ACM 消息后，通道会迁移到 S_CALL_RINGING 状态 ◆ 如果 KB=2/3/4/5/9，驱动程序会向对端交换机发送 REL 消息，拒绝来话呼叫，REL 消息中携带的拒绝原因值分别为被叫忙/地址不全/呼叫拒绝/无应答/用户缺席，通道开始拆除连接，迁移到 Release 状态； ◆ 如果 KB=8，驱动程序会向对端交换机发送 INR 消息，通道的状态会重新回到 RcvPhoNum 状态 ◆ 顺序调用 SsmSetIsupUPPara 函数(携带 C_ISUP_ACM 参数)和 SsmSendIsupMsg 函数接受来话呼叫，然后通道会迁移到 S_CALL_RINGING 状态 ◆ 调用 SsmHangup 或 SsmHangupEx 函数，拒绝本次来话呼叫，驱动程序开始拆除连接，通道会迁移到 Release 状态。更多信息请参见相关函数的说明 ◆ 调用 SsmPickup 函数接受来话呼叫。驱动程序会向对端交换机发送 CON 消息，然后通道会迁移到 Connected 状态 <p>注意：如果驱动程序在本状态收到对端交换机的拆线消息（REL），驱动程序会自动发送 RLC 消息，然后将通道迁移到 Standby 状态。</p> <p>➤ 无论是因为什么原因导致通道进入 Pending 状态，应用程序都可以调用 SsmSetIsupFlag (携带 ISUP_PhoNumREL 参数)，命令驱动程序向对端交换机发送包含号码改发信息的拆线消息（REL），并使通道迁移到 Release 状态</p> <p>➤ 如果 PendingReason 为：</p> <ul style="list-style-type: none"> ◆ PEND_RemoteHangupOnTalking (来话方向，进入通话后检测到主叫用户先挂机) ◆ PEND_CalleeHangupOnTalking (去话方向，进入通话后检测到被叫用户先挂机) ◆ PEND_RemoteHangupOnRing (来话方向，本端振铃时，主叫方取消呼叫) <p>应用程序可以调用 SsmHangup 或 SsmHangupEx 函数挂机</p>
RcvPhoNum	<p>来话接续过程中的内部状态，由 S_ISUP_Suspend 和 S_ISUP_WaitINF 子状态组成。驱动程序根据预设的收号规则对收到的被叫号码进行判断，如果符合收号规则，则通道会迁移到后续状态。有关收号规则的详细信息请参见本章中“设置来话呼叫的收号规则”部分内容。</p> <p>➤ 在来话呼叫过程中，K1 开关控制“自动应答来话呼叫”功能。当通道完成了全部被叫号码以及其它相关信息的接收后，</p> <ul style="list-style-type: none"> ◆ 如果开启了“自动应答来话呼叫”功能，驱动程序会自动向对端交换机发送 ACM 消息，然后迁移到 Ringing 状态。配置项 DefaultBackwardCallInd 可用来设置驱动程序自动发送的 ACM 消息中的后向呼叫表示语，函数 SsmSetKB 可以用来设置 ACM 消息的后向呼叫表示语中的计费表示语 ◆ 如果没有开启“自动应答来话呼叫”功能，驱动程序会将 PendingReason 设置为

	<p>PEND_WaitBckStpMsg, 然后通道迁移到 Suspend 状态, 将控制权交给应用程序, 由应用程序决定是否接受本次呼叫 K1 开关可通过函数 SsmEnableAutoSendKB 或配置项 AutoSendACM 进行设置, 缺省值为开启。</p> <ul style="list-style-type: none"> ➤ 每当驱动程序收到对端交换机的 SAM 消息, 驱动程序都会将其保存到内部缓冲区中, 并向应用程序抛出 E_CHG_RxPhoNumBuf 事件 ➤ 当通道收到被叫号码不符合收号规则时, 驱动自动发送 REL (带空号原因值) 消息, 然后通道迁移到 Release 状态 ➤ 应用程序也可调用 SsmHangup/SsmHangupEx 函数拒绝本次呼叫, 默认发送带拒绝原因值的 REL 消息, 配置项 DefaultHangupRELInd 设置挂机发送 REL 消息的原因值, 或调用函数 SsmSetIsupFlag 参数 nType=ISUP_REL_DENY_SetToOther(=100) 改变默认原因值
Ringing	<p>“振铃”状态。 应用程序可以:</p> <ul style="list-style-type: none"> ✧ 调用 SsmPickup 或 SsmPickupANX 函数接受呼叫。配置项 DefaultCalledPickupMsg 用于选择驱动程序向对端交换机发送的消息类型 (ANM 或 CON), 配置项 DefaultBackwardCallInd 可以设置 ACM 和 CON 消息中的后向呼叫表示语 ✧ 调用 SsmHangup 或 SsmHangupEx 函数拒绝本次呼叫 <p>如果通道处于 Ringing 状态时, 驱动程序收到对端交换机的 REL 消息 (即状态图中的 RcvRLC 事件), 表明对端交换机取消了本次呼叫。此时, K2 开关可以用来控制通道的状态迁移。K2 由配置项 RingToPending 决定。如果配置项 RingToPending 设置为 1, 驱动程序会设置 PendingReason 为 PEND_RemoteHangupOnRinging, 然后将通道迁移到 Pending 状态, 以等待应用程序的进一步处理; 否则, 驱动程序立即开始自动拆除本次来话呼叫</p>
Connected	<p>“连接”状态, 呼叫双方可以进行通话。 当通道迁移到 Connected 状态时, 驱动程序会:</p> <ul style="list-style-type: none"> ✧ 如果配置项 AlwaysEnableRxDtmf 设置为 0, 自动启动 DTMF 检测器 ✧ 如果配置项 AlwaysDetectBargeIn 设置为 0, 自动启动 Barge-in 检测器 ✧ 清零并启动 信号音检测器 <p>应用程序可以:</p> <ul style="list-style-type: none"> ✧ 调用任何语音处理的相关函数 ✧ 随时调用调用 SsmHangup 或 SsmHangupEx 函数终止本次呼叫 <p>在 Connected 状态, 如果驱动程序收到对端交换机的拆线消息 (状态图中的 RcvREL 事件), 表明通话时对端用户已经挂机</p>
Paused	“暂停”状态
Release	“拆线”状态。在 Release 状态, 驱动程序会等待对方发送 RLC 消息 (释放完成消息)

上表中, 状态的值与宏定义均在文件 **ShpA3Api.h** 中声明, 具体如下标所示:

状态标识	ShpA3Api.h 中声明	
	值	宏
Standby	0	S_CALL_STANDBY
OffHook	1	S_CALL_PICKUPED
Ringing	2	S_CALL_RINGING
Connected	3	S_CALL_CONNECTED
Pending	7	S_CALL_PENDING
WaitAnswer	9	S_CALL_WAIT_REMOTE_PICKUP
Unusable	11	S_CALL_UNAVAILABLE
Locked	12	S_CALL_LOCKED
Release	121	S_ISUP_WaitRLC
Reseting	122	S_ISUP_WaitReset
BlockingRemote	123	S_ISUP_LocallyBlocked
BlockedByPBX	124	S_ISUP_RemotelyBlocked
Paused	129	S_ISUP_Suspend

RcvPhoNum	120	S_ISUP_WaitSAM
	126	S_ISUP_WaitINF
AutoDial	125	S_ISUP_WaitDialAnswer
	127	S_ISUP_WaitSetCallerID

驱动程序内部的逻辑判断模块 **ChUsable** 用于判断通道是否可用。它的判断逻辑与输出的关系为：

模块输出	判决条件
Unusable	如果 Error 事件保持有效，通道不可用
BlockedByPBX	通道被对端交换机闭塞
BlockingRemote	通道上保持对对端交换机的闭塞
Standby	Error 事件保持无效

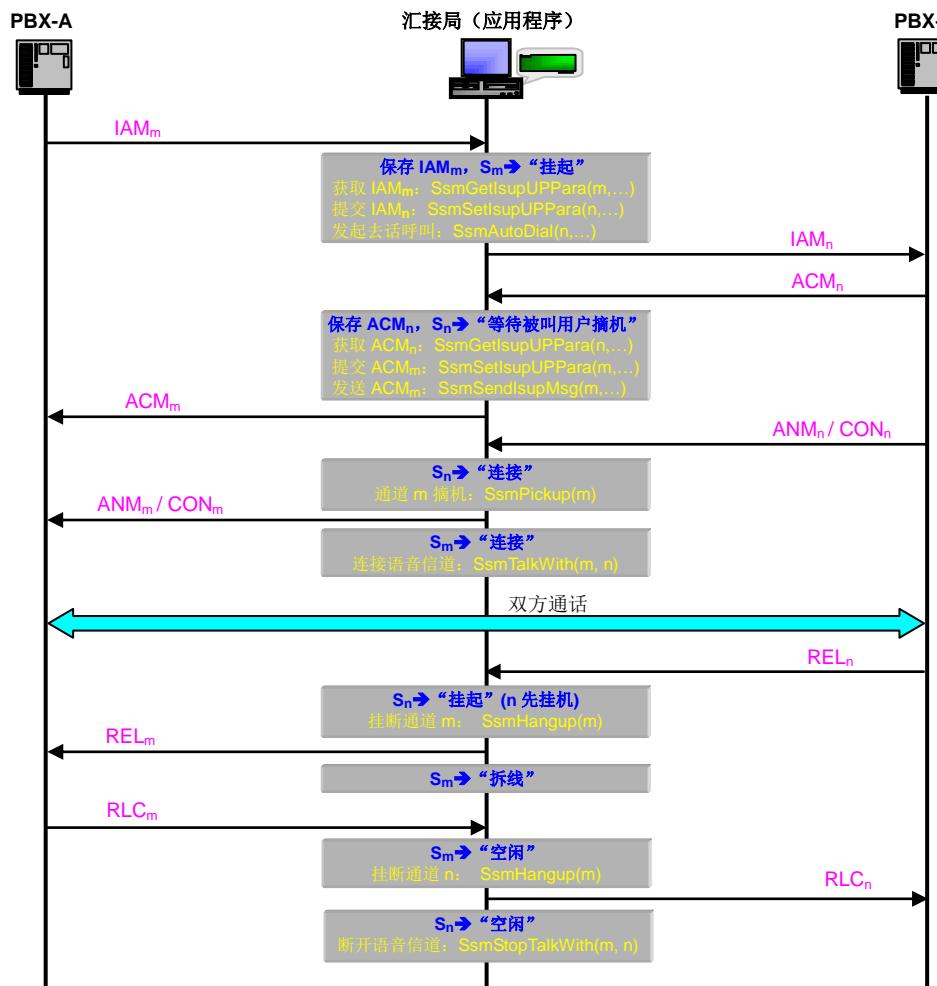
注意：在 **ChUsable** 模块中，驱动程序还要自动执行下列操作：

- 信号音检测器将被清零并自动关闭；
- 如果应用程序启动了 **WaitDtmf** 任务，该任务将被自动终止；
- 如果应用程序启动了 **DTMF** 发送任务，该任务将被自动终止；
- **DTMF** 检测器的缓冲区将被自动清空，如果配置项 [**AlwaysEnableRxDtmf**](#) 设置为 0，**DTMF** 检测器将被自动关闭；
- 如果配置项 [**AlwaysDetectBargeIn**](#) 设置为 0，**Barge-in** 检测器将被自动关闭；
- 如果配置项 [**AutoClearCallerIdBufOnHangup**](#) 设置为 1，存放主叫号码的缓冲区将被自动清空。

1.16.4.6.3 ISUP 通道工作于汇接局方式

通常情况下，使用 ISUP 协议的 SHD 系列板卡都工作在端局模式下。如果希望 ISUP 通道工作在汇接局模式下，可以按照下列方式实现：

假设应用程序中，通道 m 在 PBX-A 上的电路编号也为 m，并且通道 m 的“自动应答来话呼叫”功能为关闭状态；通道 n 在 PBX-B 上的电路编号为 n。ISUP 通道工作于汇接局方式的转接过程如下图所示。



在上图中，

蓝色字体部分表示驱动程序的内部操作；

S_m、S_n 分别表示通道 m 和通道 n 的状态；

XXX_m、XXX_n 分别表示应用程序与 PBX-A 和 PBX-B 之间的 ISUP 消息；

黄色字体部分为应用程序的动作。

1.16.4.6.4 不使用 SynCTI 提供的 ISUP 协议状态机

如果应用程序希望自行实现 ISUP 协议的状态机，可以通过修改配置项 [AutoHandleISUP](#) 实现，更多信息请参见“[SS7 的高级编程](#)”部分内容。

1.16.4.7 SS7 的高级编程

1.16.4.7.1 使用 MTP3 层提供的服务

应用程序也可以直接使用 SynCTI 驱动程序提供的 MTP3 接口功能，来自行实现 SS7 第 4 层的任何协议。如果下列配置项设置为 0：

◆ [AutoHandleTup](#) (对于 TUP 协议)

- ◆ [AutoHandleIsup](#) (对于 ISUP 协议)
- ◆ [AutoHandleSccp](#) (对于 SCCP 协议)

每当驱动程序从 SS7 服务器收到一条消息，都不会被送到驱动程序内部的状态机，而是存入一个缓冲区进行缓存，同时抛出 [E_RCV_Ss7Msu](#) 事件。应用程序收到 [E_RCV_Ss7Msu](#) 事件后，可以调用函数 [SsmGetSs7Msu](#) 取出消息，然后进行后续处理。函数 [SsmSendSs7Msu](#) 可以用来发送一条消息。

每当 SS7 服务器到某个 DPC 的路由状态发生变化时（可用，或不可用），会向 SynCTI 驱动程序发送相关消息，从而引发驱动程序向应用程序抛出 [E_CHG_Mtp3State](#) 事件。对于单个 DPC 的应用场合，函数 [SsmGetMtp3State](#) 可以获取到达该 DPC 的信令路由是否可用；如果应用系统连接到了多个 DPC，函数 [SsmGetMtp3StateEx](#) 可以获取到达指定 DPC 的信令路由是否可用。函数 [SsmGetMtp2Status](#) 可以用来获取指定的 64Kbps 的信令链路的工作状态。

1.16.4.7.2 使用 SS7 服务器的客户端软件编程接口

对于不具有七号信令功能的设备，可以使用基于 SHD 系列板卡的 SS7 信令服务器，通过 TCP/IP 与该设备连接，使其具有 SS7 信令处理能力。此类应用系统需要使用 SS7 信令服务器提供的客户端软件编程接口。SS7 信令服务器提供的客户端软件编程接口以动态连接库的形式提供，其内部封装了 TCP/IP 通讯模块和 SS7 信令处理功能，向应用程序提供透明的 SS7 信令功能，更多信息请参见《SS7 信令服务器的客户端软件编程接口》。

1.16.4.7.3 使用 SS7 服务器的虚电路编程接口

虚电路是指 SS7 电路交换中不占用语音资源的电路，一般应用于预付费系统。

SynCTI 驱动程序通过将配置项 `Pcm[k]=i,j` 中的参数 `i` 指定为 -1，来设定没有与某个物理数字中继线绑定的虚拟数字中继线，通过 SS7 信令服务器中的 `ss7server.ini` 配置文件中 `[ISUPRouter]` 节来绑定虚拟数字中继线及目的信令点码。有关虚电路的 API 函数的详细信息请参见文档《SynCTI 程序员手册-SS7 信令虚电路的编程.doc》。

1.16.4.7.4 使用 SCCP 协议的编程接口

SynCTI 驱动程序提供了 SCCP 协议的高级编程接口，更多信息请参见文档中的说明。如果应用程序需要自行实现 SCCP，需要将配置项 [AutoHandleSccp](#) 设置为 0，并使用 SynCTI 驱动程序提供的 MTP3 层服务即可。更多信息请参见本章中“[使用 MTP3 层提供的服务](#)”部分内容。

1.16.5 ISDN 通道的编程

1.16.5.1 基本概念

1.16.5.1.1 网络侧模式与用户侧模式

在 ISDN 协议中，由于通过数字中继线连接两端的协议并不完全对等，因此，在 ISDN 协议中有“用户侧”和“网络侧”的区别。用户侧指用户终端，即指与 ISDN 交换机相连的一端；网络侧是指与 ISDN 用户终端相连的另外一端。在本文档中，将 SynCTI 驱动程序中工作于用户侧模式的数字中继线称为用户侧数字中继线，将工作于网络侧模式的数字中继线称为网络侧数字中继线。

1.16.5.1.2 TEI 值

TEI：终端设备标识符，用于标识连接在服务接入点内点到点数据链路连接中的终端。TEI 子字段允许规定 128 个 TEI 值，因此，允许的取值范围为 0~127，其中，0~63 表示固定的 TEI 值（由设备选择），表明 L2 接收/发送 I 帧或 UI 帧所使用的 TEI 不变；64~126 表示动态的 TEI 值，表明 L2 接收/发送 I 帧或 UI 帧所使用的 TEI 可以变化（由网络选择）；127 只适用于广播链路。

配置项 [UserTEIValue](#)、[NetTEIValue](#) 分别用于配置用户侧、网络侧的 TEI 值。

1.16.5.1.3 通路识别表示方法

在 ISDN 信令的呼叫过程中，用来指示每次呼叫所使用的通道信息的内容，称为通路识别。通路识别主要用于识别信令程序所控制的接口范围内的通道。通路识别有下列 2 种表示方法：

- ◆ 时隙图方式：通路识别信息用 32 Bits (4 个字节) 来表达，分别对应 32 个时隙，bit=1 表示该时隙被使用，bit=0 表示未被使用。实际应用中，通常只有 1 个比特被置 1，其余比特置 0。
- ◆ 号码方式：通路识别信息用 8 Bits (1 个字节) 来表达，最高位 Bit7 固定为 1，其余 7 个比特（即 Bit6~Bit0）的值即表示相应的时隙编号。

配置项 [NetChIdentify](#) 用于设置网络侧的某个链路呼出时通路识别表示方式；配置项 [UserChIdentify](#) 用于设置用户侧的某个链路呼出时通路识别表示方式。

1.16.5.1.4 端局方式与汇接局方式

电话网分为本地电话网和长途电话网。本地电话网是在同一编号区内的网络，由端局、汇接局和传输链路组成；长途电话网是在不同的编号区之间通话的网络，由长途交换局和传输链路组成。目前，电话交换局是电话网中的核心，采用数字程控交换设备，每一路电话编码为 64Kbit/s 的数字信号，占据一次群中的某一时隙，在信令的控制下进行时隙交换，从而和各个不同的用户相连。根据服务区域的大小，电话交换局可以分为一级中心、二级中心、三级中心、四级中心和五级中心，即 C1、C2、C3、C4 和 C5。其中 C1、C2、C3、C4 为长途汇接局，C5 为端局。随着电话网的数字化进程的实现，C1、C2 合并为一级，即 DC1，C3、C4 合并为一级，即 DC2，中国的电话网从五级网演变为三级网，一级交换中心之间形成网状连接。

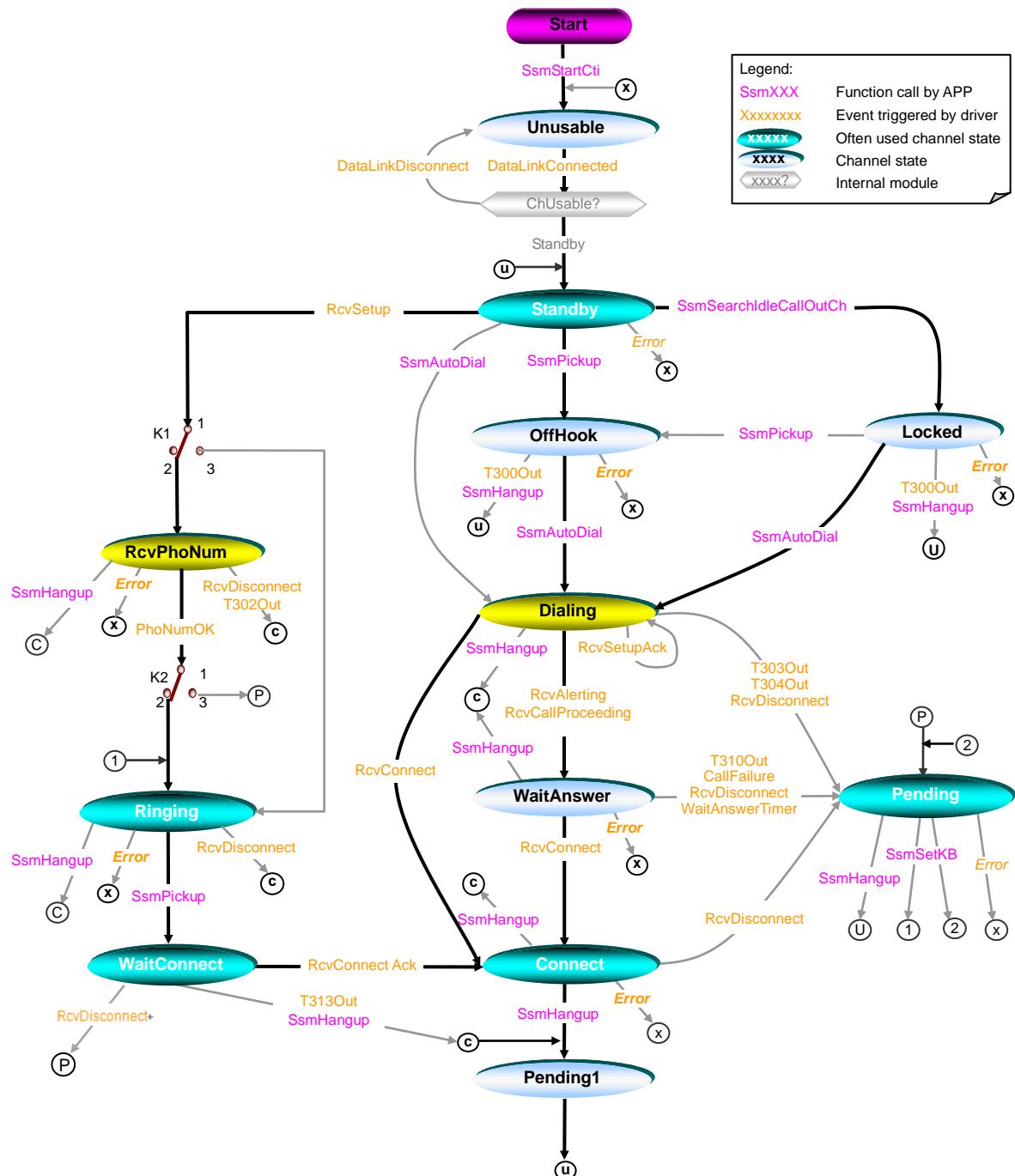
SynCTI 驱动程序提供的 ISDN 协议状态机可以既可以工作在端局方式，也可以工作在汇接局方式，可通过配置项 [UserIsReceivePhoNum](#)（用户侧）或 [NetIsReceivePhoNum](#)（网络侧）进行设置。在处理来话呼叫时，如果驱动程序工作于端局方式，会根据预设的收号规则接收被叫号码等信息；如果工作于汇接局模式，则预设收号规则将不起作用，任何号码都可以呼入到本端。有关收号规则的更多信息请参见本章中“[设置来话呼叫的收号规则](#)”部分内容。

1.16.5.1.5 ISDN 号码与 ISDN 地址

ITU-T 把 ISDN 号码和 ISDN 地址分开。ISDN 号码是与 ISDN 网络及 ISDN 编号方案有关的号码，它包含了足够的信息，使网络能够为呼叫选择正确的路由。一个 ISDN 地址包括 ISDN 号码及必备的和/或可选的附加寻址信息，

这个附加信息并不是用来供 ISDN 选择呼叫路由，而是供用户将呼叫分配到合适的终端。

1.16.5.2 ISDN 通道的状态转移图



状态转移图中，驱动程序自动触发的内部事件如下表示：

事件	事件描述
DataLinkConnected	当数据链路层与对端交换机的连接建立后，会触发本事件。驱动程序会向应用程序抛出 E_CHG_ISDNStatus 事件
DataLinkDisconnected	当数据链路层与对端交换机的连接被断开。驱动程序会向应用程序抛出 E_CHG_ISDNStatus 事件
RcvSetup	收到对端交换机的 SETUP (呼叫建立请求) 消息。

	<p>驱动程序会根据通道的当前状态，判断是否接受本次来话呼叫：</p> <ul style="list-style-type: none"> ◆ Standby 状态：接受此来话呼叫。驱动程序会将 SETUP 消息中的被叫号码等信息保存到内部缓冲区，并向应用程序抛出 E_CHG_RxPhoNumBuf 事件。如果驱动程序工作在汇接局方式，则来话呼叫的收号规则将被忽略，任何号码都可以被驱动程序接受，驱动程序会将通道直接迁移到 Ringing 状态。K1 开关是用来是否以汇接局的方式进行接续。K1 的状态可以通过配置项 UserIsReceivePhoNum（用户侧）或 NetIsReceivePhoNum（网络侧）来设置，更多信息请参见本章中“端局方式与汇接局方式”部分内容 ◆ 其它状态：丢弃此 SETUP 消息
PhoNumOK	驱动程序根据预设的收号规则，完成全部号码（包括主叫号码和被叫号码）的接收
RcvDisconnect	<p>收到对端交换机的 DISCONNECT 消息。 当本事件发生时，如果通道处于：</p> <ul style="list-style-type: none"> ◆ RcvPhoNum、WaitAnswer、Ringing 状态：表明在来话呼叫的建立过程中，对端交换机（主叫方）取消了本次呼叫 ◆ Connect 状态：对于来话呼叫，在 Connect 状态收到 DISCONNECT 消息，表明对端交换机先挂机。驱动程序会向对端交换机发送 RELEASE 消息，在将 PendingReason 设置为 PEND_ISDN_CALLOVER 后，将通道迁移到 Pending 状态 ◆ Dialing 状态：表明去话呼叫失败，具体的失败原因可以通过函数过 SsmGetAutoDialFailureReason 获得
RcvConnectAck	收到对端交换机的 CONNECT ACK 消息。本端向对端交换机发送 CONNECT 消息后，如果收到 CONNECT ACK 消息，表示连接建立
RcvConnect	收到对端交换机的 CONNECT 消息。如果通道处于 WaitAnswer 状态，驱动程序会自动向对端交换机应答 CONNECT ACK 消息。
RcvAlerting	收到对端交换机的 ALERTING 消息。此时，被叫方已经进入振铃状态
RcvCallProceeding	收到对端交换机的 CALL PROCEEDING 消息。此时，被叫方已经进入振铃状态
RcvSetupAck	收到对端交换机的 SETUP ACK 消息。本事件不会影响通道的状态迁移
RcvRelease	收到对端交换机的 RELEASE 消息。驱动程序会自动向对端交换机应答 RELEASE COMPLETE 消息，并开始拆除连接
RcvReleaseComplete	收到对端交换机的 RELEASE COMPLETE 消息。本端向对端交换机发送 RELEASE 消息后，如果收到 RELEASE COMPLETE 消息，表示连接拆除完毕
T300Out	T300 定时器溢出，T300 设置为 60s，搜索到空闲通道和摘机后操作的时间限制，以保证通道的最大利用率。用户侧通过配置项 UserDialTime 、网络侧通过 NetDialTime 来设置溢出时间
T302Out	来话呼叫时，当驱动程序收到对端交换机的 SETUP 消息后，会根据预设的收号规则对 SETUP 消息中的被叫方号码进行检查。如果号码没有收齐，驱动程序会将通道的状态迁移到 RcvPhoNum 状态的 S_ISDN_IN_RCVING_NO 子状态，并启动定时器 T302。如果 T302 溢出，本次来话呼叫失败，驱动程序会自动向对端交换机发送 RELEASE 消息，并将通道状态迁移到 Standby 。 T302 可以通过配置项 UserT302 （用户侧）或 NetT302 （网络侧）进行设置
T313Out	来话呼叫时，当通道进入 Ringing 状态后，如果应用程序调用 SsmPickup 函数摘机，驱动程序会向对端交换机自动发送 CONNECT 消息，并迁移到 WaitConnect 状态，同时启动定时器 T313。如果 T313 溢出，本次来话呼叫失败，驱动程序会向对端交换机发送 DISCONNECT 消息。 T313 可以通过配置项 UserT313 进行设置，但仅适用于用户侧
T303Out	去话呼叫时，当本端送出 SETUP 消息后，驱动程序会启动定时器 T303。如果 T303 溢出时，驱动程序还没有从对端交换机收到 ALERTING、CALL PROCEEDING、SETUP ACK 或 CONNECT 消息，驱动程序会重新发送一次 SETUP 消息，并重新启动 T303。如果 T303 再次溢出，本次去话呼叫失败，驱动程序会向对端交换机发送 RELE COMP 消息，并将通道迁移到 Pending 状态。 T303 可以通过配置项 UserT303 （用户侧）或 NetT303 （网络侧）进行设置
T304Out	去话呼叫时，当本端送出 SETUP 消息后，如果对端交换机不能从 SETUP 消息中获得全部需要的被叫号码等信息，会向本端发送 SETUP ACK 消息。本端收到此 SETUP ACK 消息后，会等待应用程序追加被叫号码，同时会启动定时器 T304。此时，应用程序可以调用函数 SsmAppendPhoNum 追加被叫号码。驱动程序收到新的号码后，会向对端交换机发送 INFO 消息，并重新启动 T304。如果 T304 溢出时，驱动程序还没有从对端交换机收到 CALL PROCEEDING、ALERTING、CONNECT 或 DISCONNECT 消息，本次去话

	呼叫失败，驱动程序会向对端交换机发送 DISCONNECT 消息，并将通道迁移到 Pending 状态。 T304 可以通过配置项 UserT304 （用户侧）或 NetT304 （网络侧）进行设置
T310Out	去话呼叫时，当本端送出 SETUP 消息、并且从对端交换机收到 CALL PROCEEDING 消息后，驱动程序会启动定时器 T310。如果 T310 溢出时，驱动程序还没有从对端交换机收到 ALERTING 或 CONNECT 消息，本次去话呼叫失败，驱动程序会向对端交换机发送 DISCONNECT 消息，并将通道迁移到 Pending 状态。 T310 可以通过配置项 UserWaitAfterCallProceeding （用户侧）或 NetWaitAfterCallProceeding （网络侧）进行设置
Error	当驱动程序检测到下列情况之一时触发本事件： <ul style="list-style-type: none"> ◆ 本通道所在数字中继线的 0 时隙上的同步信号丢失（例如线路断开、时钟配置错误等） ◆ 从对端交换机收到错误消息 ◆ 数据链路层无法建立对端交换机的连接 驱动程序的动作： <ul style="list-style-type: none"> ◆ 在 Dialing、WaitAnswer 状态：去话呼叫失败。驱动程序会设置 AutoDial 任务的进展值为 DIAL_FAILURE，失败原因设置为 ATDL_PcmSyncLos，并向应用程序抛出 E_PROC_AutoDial 事件 ◆ 其它状态：通道将迁移到不可用状态
WaitAnswerTimer	去话呼叫时，当本端送出 SETUP 消息、并且从对端交换机收到 ALERTING 消息后，驱动程序会启动定时器 WaitAnswerTimer。如果 WaitAnswerTimer 溢出时，驱动程序还没有从对端交换机收到 CONNECT 消息，驱动程序会将通道迁移到 Pending 状态，但此时呼叫并没有被释放，用户需要自行调用 SsmHangup 释放呼叫。WaitAnswerTimer 可以通过配置项 MaxWaitAutoDialAnswerTime 进行设置。

各状态标识的说明如下表所示：

状态标识	描述
Unusable	通道“不可用”状态。 <ul style="list-style-type: none"> ◆ 如果 DatalinkDisconnect、Error 事件之一保持有效，通道不可用，输出 Unusable ◆ 其它情况，通道空闲，输出 Standby
Standby	“空闲”状态。 驱动程序初始化成功后，如果 ISDN 的数据链路层顺利建立到对端交换机的连接，应该进入本状态。应用程序可以在本状态调用 SsmAutoDial 函数发起去话呼叫业务。当通道因为呼叫结束而迁移到本状态时，如果配置项 AutoClearCallerIdBufOnHangup 设置为 1，驱动程序会自动清空主叫号码缓冲区
OffHook	“摘机”状态。 通道进入 OffHook 状态时，驱动程序会启动定时器 T300。如果应用程序在 T300 溢出之前没有调用 SsmAutoDial 函数，驱动程序会将此通道进行回收，将通道重新迁移到空闲状态
Locked	“呼出锁定”状态。 应用程序调用 SsmSearchIdleCallOutCh 函数时，驱动程序会将函数返回的通道临时锁定，同时启动定时器 T300。如果应用程序在 T300 溢出之前没有调用 SsmPickup 或 SsmAutoDial 函数，驱动程序就会将此通道进行回收
Dialing	去话呼叫过程中的中间状态，由 S_ISDN_OUT_WATI_NET_RESPONSE 和 S_ISDN_OUT_PLS_APPEND_NO 两个子状态组成。应用程序调用 SsmAutoDial 函数后，驱动程序会： <ul style="list-style-type: none"> ◆ 向对端交换机发送 SETUP 消息 ◆ 同时启动 T303 定时器 ◆ 将通道迁移到 S_ISDN_OUT_WATI_NET_RESPONSE 子状态，以等待对端交换机的响应信号 ◆ 向应用程序抛出 E_PROC_AutoDial 事件。 在 S_ISDN_OUT_WATI_NET_RESPONSE 子状态，如果收到对端交换机的 SETUP ACK 消息，表明对方需要本端提供进一步的号码信息，驱动程序会将通道迁移到 S_ISDN_OUT_PLS_APPEND_NO 子状态，同时启动定时器 T304。此时，应用程序可以调用 SsmAppendPhoNum 追加被叫号码
WaitAnswer	“等待被叫用户摘机”状态。通道迁移到本状态时，驱动程序会启动定时器 T310。函数 SsmGetFlag （携带 F_ISDNNet_WaitRemotePickup 参数）可以用来获取通道迁移到

	本状态的具体原因
WaitConnect	等待对端交换机应答 CONNECT ACK 消息。通道迁移到本状态时，驱动程序会启动定时器 T313
Connect	<p>“连接”状态，呼叫双方可以进行通话。</p> <p>当通道迁移到 Connected 时：</p> <ul style="list-style-type: none"> ◆ 如果配置项 AlwaysEnableRxDtmf 设置为 0，DTMF 检测器将被启动 ◆ 如果配置项 AlwaysDetectBargeIn 设置为 0，Barge-in 检测器将被启动 ◆ 信号音检测器将被清零并启动 <p>驱动内部响应事件：</p> <ul style="list-style-type: none"> ◆ RcvDisconnect 事件：表明通话时对端先挂机，驱动程序在将挂起原因设置为 ISDN_CALLOVER 后迁移到“挂起”状态 ◆ Error 事件：表明通话过程中遇到 Error 事件之一，通道将迁移到“不可用”状态
Ringing	“振铃”状态。通道完成了全部被叫号码以及其它相关信息的接收
Pending	“挂起”状态，等待应用程序的进一步指令
Pending¹	“挂起”状态，应用程序可以忽略此状态，不需要进一步指令，驱动程序会自动跳转到 Standby 状态。
RcvPhoNum	<p>来话呼叫时的内部状态，由 S_ISDN_IN_CHK_CALL_IN 和 S_ISDN_IN_RCVING_NO 子状态组成。</p> <p>收到对端交换机的 SETUP 消息后，如果 SETUP 消息中包含了本端收号所需要的全部信息，驱动程序会触发内部事件 PhoNumOK；否则，驱动程序会向对端交换机发送 SETUP ACK 消息，并将通道迁移到 S_ISDN_IN_CHK_CALL_IN 子状态，等待对端交换机发送进一步的号码信息。在 S_ISDN_IN_CHK_CALL_IN 子状态，如果收到对端交换机的 INFO 消息，驱动程序会将通道迁移到 S_ISDN_IN_RCVING_NO 子状态。</p> <p>如果收到的被叫号码不符合收号规则，驱动程序会自动向对端交换机发送 RELEASE 消息，然后将通道迁移到 S_CALL_STANDBY 状态。</p> <p>配置项组合 DefaultRcvPhoNumLen、DefaultRcvCallerID、RcvPhoNumCfgLen 或 DefaultRcvPhoNumLen、MaxPhoNumRule、Rule 可以用来设置收号规则。有关收号规则的详细信息请参见本章中“设置来话呼叫的收号规则”部分内容。</p> <p>通道完成收号过程后，迁移的下一个状态取决于开关 K2。K2 开关用来决定是由驱动程序自动应答 ALERTING 消息，还是由应用程序接管。K2 的状态可以通过配置项 UserSideAutoSendAck（用户侧）或 NetSideAutoSendAck（网络侧）或函数 SsmEnableAutoSendKB 进行设置。如果设置为由驱动程序自动应答，驱动程序在向对端交换机发送 ALERTING 后，将通道迁移到 Ringing 状态；否则，驱动程序会启动一个定时器，定时器的时长由配置项 UserSideDefaultAckTimer（用户侧）或 NetSideDefaultAckTimer（网络侧）进行设置，在将 PendingReason 设置为 PEND_WAIT_BCK_STP_MSG 后，驱动程序会将通道迁移到 Pending 状态，将控制权交给应用程序，由应用程序决定是否接受本次呼叫。如果定时器溢出，驱动程序会根据配置项 UserSideDefaultAck（用户侧）或 NetSideDefaultAck（网络侧）的设置，自动向对端交换机发送应答消息</p>

各状态标识在头文件 ShpA3Api.h 中的声明如下表所示：

状态标识	ShpA3Api.h 中的值	ShpA3Api.h 中的宏定义
Unusable	11	S_CALL_UNAVAILABLE
Standby	0	S_CALL_STANDBY
OffHook	1	S_CALL_PICKUPED
Dialing	81	S_ISDN_OUT_WAIT_NET_RESPONSE
	82	S_ISDN_OUT_PLS_APPEND_NO
WaitAnswer	9	S_CALL_WAIT_REMOTE_PICKUP
Connect	3	S_CALL_TALKING
WaitConnect	85	S_ISDN_IN_WAIT_TALK
Ringing	2	S_CALL_RINGING
Pending	7	S_CALL_PENDING
Locked	12	S_CALL_LOCKED
RcvPhoNum	83	S_ISDN_IN_CHK_CALL_IN
	84	S_ISDN_IN_RCVING_NO

驱动程序内部逻辑判断模块的描述如下表所示：

模块输出	判决条件
Unusable	如果 Error 、 DataLinkDisconnect 、 Circuit Reset 之一事件保持有效，通道不可用
Standby	Error 事件保持无效、 DataLinkConnect 事件保持有效、电路重起成功，通道状态空闲

注意：在此模块中，驱动程序还要做下列清理工作：

- 清零并关闭信号音检测器；
- 如果应用程序启动了 [WaitDtmf](#) 任务，该任务将被自动终止；
- 如果应用程序启动了 [DTMF](#) 发送任务，该任务将被自动终止；
- DTMF 检测器的缓冲区将被自动清空，如果配置项 [AlwaysEnableRxDtmf](#) 设置为 0，DTMF 检测器将被自动关闭；
- 如果配置项 [AlwaysDetectBargeIn](#) 设置为 0，Barge-in 检测器将被自动关闭；
- 如果配置项 [AutoClearCallerIdBufOnHangup](#) 设置为 1，存放主叫号码的缓冲区将被自动清空；

状态机的输出事件类型如下表所示：

事件名称	描述
E_CHG_ChState	当通道的状态发生迁移时，驱动程序向应用程序抛出本事件
E_PROC_AutoDial	应用程序调用 SsmAutoDial 函数即启动了 AutoDial 任务。当 AutoDial 任务有进展时，驱动程序向应用程序抛出本事件
E_CHG_RxPhoNumBuf	在来话呼叫过程中，每当驱动程序收到对端交换机的被叫号码信息驱动程序向应用程序抛出本事件
E_CHG_ISDNStatus	驱动程序收到 ISDN 协议数据链路层可用消息触发本事件，并向应用程序抛出本事件

注：事件的声明在文件 `ShpA3Api.h` 中。

1.16.5.3 ISDN 协议的配置

使用 ISDN 协议时，相关的配置项如下表所示：

配置节	配置项	
[BoardId=x]	PcmNumber	设置板卡上数字中继线的总数，以及每条数字中继线的信令类型、时钟的工作模式、线路的传输类型
	PcmSSx	
	PcmClockMode	
	PcmLinkType	
[PcmInfo]	TotalPcm	设置本机数字中继线的总数，以及建立数字中继线的逻辑编号与物理编号之间的映射关系
	Pcm	
[ISDN]	AutoHandleIsdn	设置是否使用驱动程序提供的呼叫状态机
	UseISDNMode	

	<u>TotalUserLinker</u>	设置本机中 ISDN 用户侧协议的数字中继线的总数，以及中继线的逻辑编号与物理编号之间的映射关系
	<u>UserPcmLink</u>	
	<u>UserTEIValue</u>	
	<u>UserRestartTime</u>	
	<u>UserEstablishTime</u>	ISDN 用户侧协议基本参数设置
	<u>UserT302</u>	
	<u>UserCrcMode</u>	
	<u>UserStatusReason</u>	
	<u>UserDialTime</u>	
	<u>UserWaitAfterCallProceeding</u>	ISDN 用户侧去话呼叫：设置工作参数
	<u>UserTxCallingPartyNum</u>	
	<u>NetTxCallingPartyNum</u>	
	<u>UserChIdentify</u>	
	<u>UserCalledNoSet</u>	
	<u>UserCallingNoSet</u>	
	<u>UserNumIsFull</u>	ISDN 用户侧去话呼叫：定制向对端交换机发送的 SETUP 消息
	<u>UserChPreference</u>	
	<u>UserHighLayerCompatible</u>	
	<u>UserLowLayerCompatible</u>	
	<u>CalloutCallerId</u>	
	<u>UserSideAutoSendAck</u>	
	<u>UserSideDefaultAckTimer</u>	ISDN 用户侧来话呼叫：设置工作参数
	<u>UserIsReceivePhoNum</u>	
	<u>UserVoiceFormat</u>	ISDN 用户侧硬件编码设置
	<u>TotalNetLinker</u>	设置本机中 ISDN 网络侧协议的数字中继线的总数，以及中继

	<u>NetPcmLink</u>	线的逻辑编号与物理编号之间的映射关系
	<u>NetTEIValue</u>	
	<u>NetRestartTime</u>	
	<u>NetEstablishTime</u>	ISDN 网络侧协议基本参数设置
	<u>NetT302</u>	
	<u>NetCrcMode</u>	
	<u>NetStatusReason</u>	
	<u>NetDialTime</u>	
	<u>NetWaitAfterCallProceeding</u>	ISDN 网络侧去话呼叫：设置工作参数
	<u>NetChIdentify</u>	
	<u>NetCalledNoSet</u>	
	<u>NetCallingNoSet</u>	
	<u>NetNumIsFull</u>	ISDN 网络侧去话呼叫：定制向对端交换机发送的 SETUP 消息
	<u>NetChPreference</u>	
	<u>NetHighLayerCompatible</u>	
	<u>NetLowLayerCompatible</u>	
	<u>CalloutCallerId</u>	
	<u>NetSideAutoSendAck</u>	
	<u>NetSideDefaultAckTimer</u>	
	<u>NetIsReceivePhoNum</u>	ISDN 网络侧来话呼叫：设置工作参数
	<u>DefaultRcvCallerID</u>	
	<u>NetVoiceFormat</u>	ISDN 网络侧硬件编码设置
	<u>DefaultRcvPhoNumLen</u>	
	<u>MaxPhoNumRule</u>	ISDN 用户侧、网络侧来话呼叫：收号规则设制

	<u>Rule</u>	
	<u>UserStatusReason</u> <u>NetStatusReason</u>	ISDN 用户侧、网络侧的高级选项
[DebugView]	<u>IsdnLogEnable</u>	ISDN 用户侧、网络侧日志输出设置
	<u>DecodelsdnMsg</u>	
	<u>IsdnDebugLog</u>	
	<u>LogCreatePeriod</u>	
	<u>LogMaxKeep</u>	
	<u>LogMaxPeriod</u>	
	<u>LogFilepath</u>	

注：XXXX 字样为必须配置的项目，XXXX 为可选的配置项目

1.16.5.4 ISDN 高级编程接口

SynCTI 驱动程序还提供了高级编程接口，输出 HDLC 层的原始消息，由应用程序自己处理 ISDN 消息。有关 ISDN 高级编程接口的配置项和函数如下表所示：

类型	名称	描述
配置项	<u>AutoHandleIsdn</u>	设置是否使用 ISDN 的高级编程接口
函数	<u>SsmSendIsdnMSU</u>	发送一条消息
	<u>SsmCheckIsdnMSU</u>	查询是否有消息到达
	<u>SsmGetIsdnMSU</u>	取出一条 ISDN 消息

1.16.6 SS1 通道编程指南

1.16.6.1 选择信令协议

SynCTI 支持 No.1 信令和 Line side 协议，可通过配置项 ProtocolType 进行设置。如果使用 No.1 信令，配置项 mfcr2_Protocol 用于选择国别。

1.16.6.2 设置接续过程的基本工作参数

配置项 TxCas_CD 用于设置本端向对端发送 ABCD 信令码时 C/D 码元的值。配置项 RxCASFilterTime 用于设置对端交换机送出的 ABCD 信令的最小保持时间，以消除线路上的信号抖动，适用于信号质量比较差的 E1/T1 线路。

R2 信号的最小保持时间可以通过配置项 RxR2FilterTime 或函数 SsmSetFlag（携带 F_RXR2FILTERNODE 参数）来进行设置。

当应用程序退出时，配置项 [IsBlockSS1In](#) 用于设置是否向对端交换机发送闭塞命令，以通知对端交换机不要再向本端发起呼叫。

1.16.6.3 设置状态机中通道的工作方式

配置项 [EnableAutoCall](#) 用于设置是否使用 SynCTI 驱动程序提供的呼叫状态机，函数 [SsmSetAutoCallDirection](#) 或配置项 [AutoCallInTimeSlot](#) 用于设置通道的呼叫方向，如可以将通道设置成只处理来话呼叫或只处理去话呼叫。

1.16.6.4 中国 1 号信令的状态机描述

1.16.6.4.1 来话呼叫时启用号码拦截功能

“被叫号码拦截”功能，常用于一些关键的部门，是为了减少主叫方的误拨号对应用系统造成的影响。例如，对于采用 Synway 板卡构建的 110 报警系统，如果主叫方拨打的是“110”，这是一个合法的被叫号码，应用程序应该处理；如果主叫方拨打的是“1101”，报警系统就不应该处理本次来话呼叫。

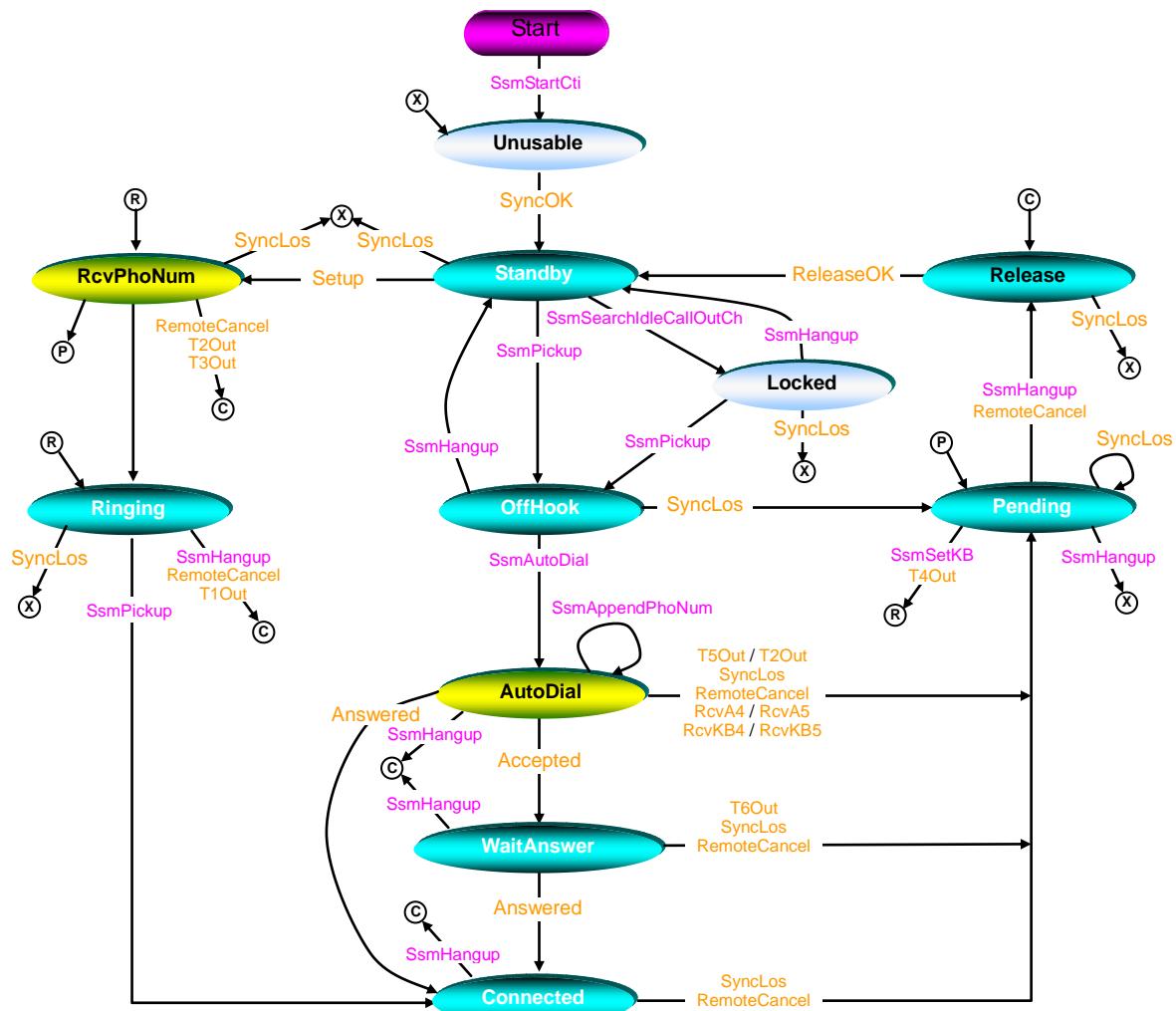
“被叫号码拦截”功能及工作模式由配置项 [PhoNumHoldup](#) 或函数 [SsmSetFlag](#)（携带 [F_RCVPHONUMHOLDUP](#) 参数）控制，其工作参数可以通过配置项 [A1ToA3pWaitTime](#) 和 [A3pTime](#) 进行设置。例如，一个 110 接警系统可以如下设置拦截 110x 形式的被叫号码：

```
PhoNumHoldup=1
A1ToA3pWaitTime=1000
A3pTime=150
```

1.16.6.4.2 与 Dialogic 公司 SS1 通道对接

使用 Synway 板卡的 SS1 通道 Dialogic 公司的 SS1 通道进行对接呼叫时，如果 Synway 板卡作为呼入端，需要对配置项 [ToRingingDelayTime](#) 进行设置。

1.16.6.4.3 中国 1 号信令的状态转移



上图中，内部事件的描述如下表所示：

事件名称	描述
RemoteCancel	对端交换机取消了呼叫
SyncLos	数字中继线 0 时隙上的同步信号丢失
SyncOK	检测到数字中继线 0 时隙上的同步信号
Setup	收到对端交换机的来话呼叫请求
T1Out	T1 定时器溢出。 1 在通道迁移到“振铃”状态时启动。T1 的设定值为 75 秒
T2Out	T2 定时器溢出。 SS1 通道在呼叫过程中，需要等待对端交换机的响应信号。T2 设置最大等待时间，如果 T2 溢出，表明本次互控过程失败。T2 可以通过配置项 MaxWaitMfcTime 进行设置
T3Out	T3 定时器溢出。 SS1 通道在来话呼叫过程中，当本端完成被叫号码的接收后，会向对端交换机发送 A3 信号，同时启动 T3，等待 KD 信号。如果 T3 溢出时对端交换机还没有送出 KD 信号，来话呼叫接续失败。T3 可以通过配置项 MaxWaitKDTIME 进行设置
T4Out	T4 定时器溢出。

	SS1 通道在来话呼叫过程中，当本端完成被叫号码的接收后，如果配置项 AutoSendKB 设置为 0，表明应用程序希望执行决定是否应答本次呼叫，通道会进入“挂起”状态，并启动定时器 T4。如果 T4 溢出时，应用程序还没有调用 SsmSetKB 函数对呼叫进行应答，驱动程序会拒绝本次来话呼叫。T4 可以通过配置项 MaxWaitSetKBTime 进行设置
T5Out	T5 定时器溢出。 应用程序调用 SsmAutoDial 函数发起去话呼叫后，驱动程序会向对端交换机发送“占用”请求信号，并等待对端交换机的证实信号。如果 T5 溢出时，仍然没有收到对端交换机的证实信号，本次去话呼叫失败。T5 可以通过配置项 MaxWaitOccupyAckTime 进行设置
T6Out	T6 定时器溢出。 应用程序调用 SsmAutoDial 函数发起去话呼叫后，如果被叫用户空闲，通道会启动 T6。如果 T6 溢出时，被叫用户仍然没有摘机，本次去话呼叫失败。T6 可以通过配置项 MaxWaitAutoDialAnswerTime 进行设置
RcvA4	去话接续过程中收到对端交换机的 A4 信号（机键拥塞，接续尚未到达被叫之前遇到空号），去话呼叫失败
RcvA5	去话接续过程中收到对端交换机的 A5 信号（空号），去话呼叫失败
RcvKB4	去话接续过程中，收到对端交换机的 KB 信号为 4（用户忙或机键拥塞），去话呼叫失败
RcvKB5	去话接续过程中，收到对端交换机的 KB 信号为 5（被叫用户为空号），去话呼叫失败
Accepted	去话接续过程中，对端交换机完成被叫号码的接收后，向本端发送了 KB 信号值为 1 或 6，表明被叫用户空闲
Answered	去话接续过程中，被叫用户接受了呼叫
ReleaseOK	完成拆除与对端交换机的连接

各状态的说明如下表所示：

状态标识	描述
Standby	空闲状态。驱动程序初始化成功后，如果通道正常复位，应该进入本状态。应用程序可以在本状态调用 SsmAutoDial 函数发起去话呼叫业务。 当通道因为呼叫结束而迁移到空闲状态时，如果配置项 AutoClearCallerIdBufOnHangup 设置为 1，驱动程序会自动清空主叫号码缓冲区
OffHook	“摘机”状态。 如果在“摘机”状态发生 SyncLos 事件，驱动程序会： ◆ 将 AutoDial 任务的进展值设置为 DIAL_FAILURE，失败原因设置为 ATDL_PcmSyncLos，并向应用程序抛出 E_PROC_AutoDial 事件 ◆ 将挂起原因设置为 PEND_PcmSyncLos
Ringing	本端振铃状态。应用程序可以： ◆ 调用函数 SsmGetCallerId 获取主叫方的信息 ◆ 调用 SsmPickup 函数接受呼叫，也可以调用 SsmHangup 函数拒绝呼叫
Connected	连接状态，呼叫双方可以进行通话。当通道迁移到 Talking 时，驱动程序会： ◆ 启动 DTMF 检测器 ◆ 启动 Barge-in 检测器 ◆ 清零并启动信号音检测器 驱动程序对事件的响应为： ➤ RemoteCancel 事件： 表明通话时对端先挂机，驱动程序在将挂起原因设置为 PEND_RemoteHangupOnTalking 后迁移到“挂起”状态 ➤ SyncLos 事件： ◆ 将 AutoDial 任务的进展值设置为 DIAL_FAILURE，失败原因设置为 ATDL_PcmSyncLos，并向应用程序抛出 E_PROC_AutoDial 事件 ◆ 将挂起原因设置为 PEND_PcmSyncLos
Pending	等待应用程序进一步的指示。 驱动程序对事件的响应为： ➤ T4Out 事件：

	<p>驱动程序自动将 KB 信号设置为 4 (机键拥塞), 然后恢复来话接续过程</p> <ul style="list-style-type: none"> ➤ 应用程序调用 SsmSetKB 函数: 如果挂起原因为 PEND_WaitBckStpMsg, 通道将恢复来话接续过程 ➤ SyncLos 事件: ◆ 将 AutoDial 任务的进展值设置为 DIAL_FAILURE, 失败原因设置为 ATDL_PcmSyncLos, 并向应用程序抛出 E_PROC_AutoDial 事件 ◆ 将挂起原因设置为 PEND_PcmSyncLos ➤ 应用程序调用 SsmHangup 函数: 如果挂起原因为 PEND_PcmSyncLos, 通道状态将迁移到“不可用”; 其它的挂起原因则进入“拆线”状态
WaitAnswer	<p>等待被叫用户摘机。此时, 被叫方应该听到振铃音。如果本端是坐席通道通过数字中继通道进行电话呼出, 则应向坐席通道发送回铃音信号。</p> <p>驱动程序对事件的响应为:</p> <ul style="list-style-type: none"> ➤ T6Out 事件: ◆ 将 AutoDial 任务的进展值设置为 DIAL_NOANSWER, 失败原因设置为 ATDL_WaitRemotePickupTimeout, 并向应用程序抛出 E_PROC_AutoDial 事件 ◆ 将挂起原因设置为 SS1OUT_NOANSWER ➤ SyncLos 事件: ◆ 将 AutoDial 任务的进展值设置为 DIAL_FAILURE, 失败原因设置为 ATDL_PcmSyncLos, 并向应用程序抛出 E_PROC_AutoDial 事件 ◆ 将挂起原因设置为 PEND_PcmSyncLos ➤ RemoteCancel 事件: ◆ 将 AutoDial 任务的进展值设置为 DIAL_FAILURE, 失败原因设置为 ATDL_SS1RcvCAS_HANGUP, 并向应用程序抛出 E_PROC_AutoDial 事件 ◆ 将挂起原因设置为 SS1OUT_DIALING_BWD_HANGUP
Unusable	“通道不可用”
Locked	“呼出锁定”状态
Release	“拆除连接”状态
RcvPhoNum	<p>来话接续过程中的内部状态, 由多个状态组成。驱动程序根据预设的收号规则对收到的被叫号码进行判断, 如果符合收号规则, 则通道会迁移到后续状态。有关收号规则的详细信息请参见本章中“设置来话呼叫的收号规则”部分内容。</p> <ul style="list-style-type: none"> ➤ 在来话呼叫过程中, 当通道完成了全部被叫号码以及其它相关信息的接收后, 需要向对端交换机发送 KB 信号, ◆ 如果开启了“自动应答来话呼叫”功能, 驱动程序会将配置项 MfcKB 的设定值作为 KB 信号发送给对端交换机, 然后迁移到 Ringing 状态; ◆ 如果没有开启“自动应答来话呼叫”功能, 驱动程序会将 PendingReason 设置为 PEND_WaitBckStpMsg, 然后通道迁移到 Pending 状态, 将控制权交给应用程序, 由应用程序决定是否接受本次呼叫。 “自动接受来话”开关可通过函数 SsmEnableAutoSendKB 或配置项 AutoSendKB 进行设置, 缺省值为开启。 ➤ 每当驱动程序收到主叫方的一个号码, 驱动程序都会抛出 E_CHG_RxPhoNumBuf 事件
AutoDial	<p>去话接续过程中的内部状态, 由多个状态组成。</p> <p>去话呼叫过程中, 应用程序可以设置的参数包括:</p> <ul style="list-style-type: none"> ➤ 本端需要向对端交换机提供的 KA 信号 (主叫类别), 可以通过函数 SsmSetKA 或配置项 MfcKA 进行设置 ➤ 本端需要向对端交换机提供的 KD 信号 (发端业务类别), 可以通过函数 SsmSetKD 或配置项 MfcKD 进行设置 ➤ 如果对端交换机需要本端提供主叫方电话号码, 驱动程序会在收到 A6 信号、发送完 KA 信号后会自动将保存在缓冲区中的本端电话号码发送给对端交换机。本端的电话号码可通过函数 SsmSetTxCallerId 或配置项 TxCallerId 进行设置 <p>如果去话呼叫失败, 驱动程序会依据内部事件的类型做下列处理:</p> <ul style="list-style-type: none"> ➤ SyncLos 事件:

	<ul style="list-style-type: none">◆ 将 AutoDial 任务的进展值设置为 DIAL_FAILURE，失败原因设置为 ATDL_PcmSyncLos，并向应用程序抛出 E_PROC_AutoDial 事件◆ 将挂起原因设置为 PEND_PcmSyncLos➤ T5Out 事件：<ul style="list-style-type: none">◆ 将 AutoDial 任务的进展值设置为 DIAL_NO_DIALTONE，失败原因设置为 ATDL_SS1WaitOccupyAckTimeout，并向应用程序抛出 E_PROC_AutoDial 事件◆ 将挂起原因设置为 SS1OUT_NOBWDACK➤ RemoteCancel 事件：<ul style="list-style-type: none">◆ 将 AutoDial 任务的进展值设置为 DIAL_FAILURE，失败原因设置为 ATDL_SS1RcvCAS_HANGUP，并向应用程序抛出 E_PROC_AutoDial 事件◆ 将挂起原因设置为 SS1OUT_DIALING_BWD_HANGUP➤ RcvA4 事件：<ul style="list-style-type: none">◆ 将 AutoDial 任务的进展值设置为 DIAL_BUSYTONE，失败原因设置为 ATDL_SS1RcvA4，并向应用程序抛出 E_PROC_AutoDial 事件◆ 将挂起原因设置为 SS1OUT_BWD_A4➤ RcvKB4 事件：<ul style="list-style-type: none">◆ 将 AutoDial 任务的进展值设置为 DIAL_BUSYTONE，失败原因设置为 ATDL_SS1RcvKB4，并向应用程序抛出 E_PROC_AutoDial 事件◆ 将挂起原因设置为 SS1OUT_BWD_KB4➤ RcvA5 事件：<ul style="list-style-type: none">◆ 将 AutoDial 任务的进展值设置为 DIAL_INVALID_PHONUM，失败原因设置为 ATDL_SS1RcvA5，并向应用程序抛出 E_PROC_AutoDial 事件◆ 将挂起原因设置为 SS1OUT_BWD_A5➤ RcvKB5 事件：<ul style="list-style-type: none">◆ 将 AutoDial 任务的进展值设置为 DIAL_INVALID_PHONUM，失败原因设置为 ATDL_SS1RcvKB5，并向应用程序抛出 E_PROC_AutoDial 事件◆ 将挂起原因设置为 SS1OUT_BWD_KB5➤ T2Out 事件：<ul style="list-style-type: none">将 AutoDial 任务的进展值为 DIAL_FAILURE，失败原因设置为下列值之一：<ul style="list-style-type: none">◆ ATDL_SS1WaitAxTimeout◆ ATDL_SS1WaitAxStopTimeout◆ ATDL_SS1WaitAxTimeoutOnTxCallerId◆ ATDL_SS1WaitAxStopTimeoutOnTxCallerId◆ ATDL_SS1WaitKBTimeout◆ ATDL_SS1WaitKBStopTimeout对应的挂起原因分别设置为<ul style="list-style-type: none">◆ SS1OUT_TIMEOUT_BWD_A◆ SS1OUT_TIMEOUT_BWD_A_STO◆ SS1OUT_TIMEOUT_CALLERID_BWD_A1◆ SS1OUT_TIMEOUT_CALLERID_BWD_A1_STOP◆ SS1OUT_TIMEOUT_BWD_KB◆ SS1OUT_TIMEOUT_BWD_KB_STOP;并向应用程序抛出 E_PROC_AutoDial 事件
--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

状态标识、状态值之间的对应关系如下表所示：

状态标识	值	ShpA3Api.h 对应的宏定义
Standby	0	S_CALL_STANDBY
OffHook	1	S_CALL_PICKUPED
Ringing	2	S_CALL_RINGING
Connected	3	S_CALL_TALKING
Pending	7	S_CALL_PENDING
WaitAnswer	9	S_CALL_WAIT_REMOTE_PICKUP
Unusable	11	S_CALL_UNAVAILABLE

Locked	12	S_CALL_LOCKED
Release	36	S_CALL_SS1WaitIdleCAS
RcvPhoNum	—	S_CALL_Ss1InWaitPhoNum S_CALL_Ss1InWaitFwdStop S_CALL_Ss1InWaitCallerID S_CALL_Ss1InWaitKD S_CALL_Ss1InWaitKDStop S_CALL_SS1WaitIdleCAS S_CALL_SS1PhoNumHoldup S_CALL_Ss1InWaitStopSendA3p
AutoDial	—	S_CALL_Ss1OutWaitBwdAck S_CALL_Ss1OutTxPhoNum S_CALL_Ss1OutWaitAppendPhoNum S_CALL_Ss1OutTxCallerID S_CALL_Ss1OutWaitKB S_CALL_Ss1OutDetectA3p

状态机的输出事件类型如下表所示：

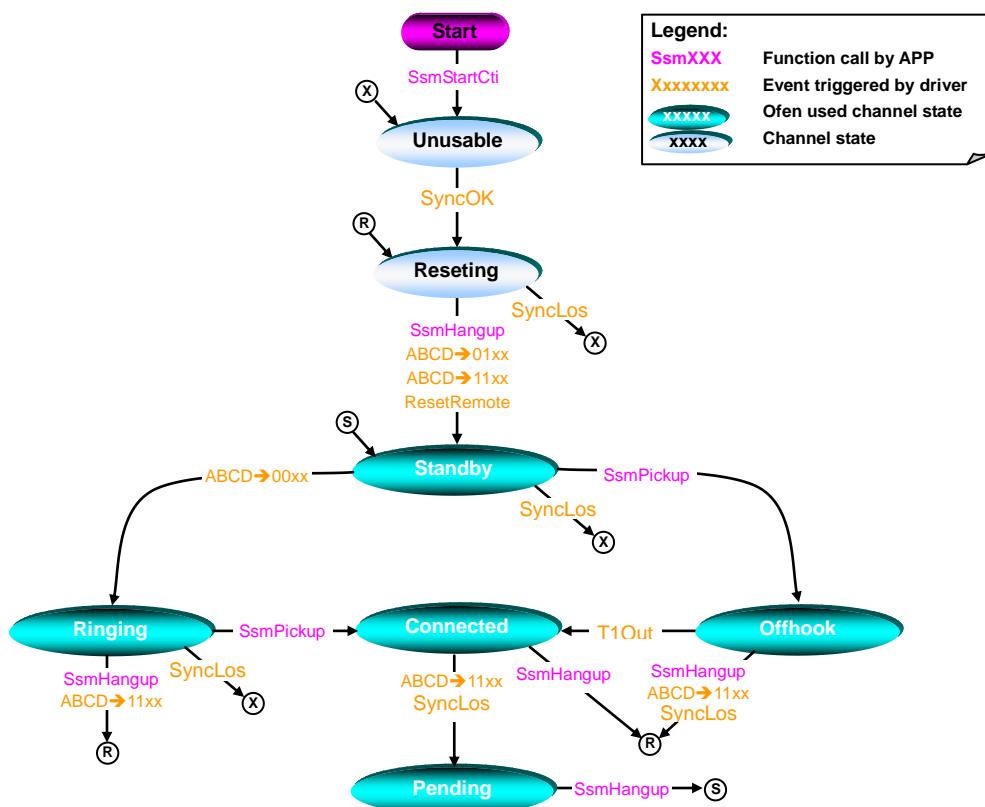
事件名称	描述
E_CHG_ChState	当通道的状态发生迁移时，驱动程序向应用程序抛出本事件
E_CHG_RxPhoNumBuf	在来话呼叫过程中，每当驱动程序收到对端交换机的被叫号码信息，驱动程序向应用程序抛出本事件
E_RCV_R2	当应用程序调用 SsmGetR2 函数获得 R2 消息时，如果驱动程序收到对端的新的 R2 信号值，驱动程序向应用程序抛出本事件
E_RCV_CAS	当应用程序调用 SsmGetCAS 函数取得指定通道上发送 ABCD 信令的当前值时，如果驱动程序收到了当前的 CAS 值，驱动程序向应用程序抛出本事件
E_CHG_CIDExBuf	驱动程序收到 CID 扩展接收缓冲区有变化后，驱动程序向应用程序抛出本事件。应用程序得到该事件后，可调用 SsmGetCallerIdEx 函数取得 CID 扩展接收缓冲区内容

1.16.6.4.4 输出呼叫过程中的调试信息

配置项 [MfcR2ToRxCallerIdBuf](#) 用于控制是否将呼叫过程中对端交换机发送的 R2 信号保存到主叫号码扩展缓冲区中，以便于对呼叫过程进行观察。函数 [SsmGetCallerIdEx](#) 用于取出该缓冲区中的字符串。

配置项 [Ss1LogEnable](#) 用于控制是否将接收和发送的 ABCD 位码和 R2 输出到日志文件 Ss1Output.log 中。

1.16.6.5 Line side 协议的状态机描述



上图中，内部事件的描述如下表所示：

事件名称	描述
SyncLos	数字中继线 0 时隙上的同步信号丢失
SyncOK	检测到数字中继线 0 时隙上的同步信号
ABCD→00xx	对端交换机的 ABCD 位码变化为 00xx
ABCD→01xx	对端交换机的 ABCD 位码变化为 01xx
ABCD→10xx	对端交换机的 ABCD 位码变化为 10xx
ResetRemote	驱动程序完成对对端交换机的电路复位操作，即向对端交换机发送空闲指示的 ABCD 位码。具体发送的空闲指示的 ABCD 位码的值由配置项 Ss1SendIdleState 决定
T1Out	T1 定时器溢出。T1 定时器在通道迁移到“摘机”状态时启动。T1 可以通过配置项 LSWaitPickup 进行设置

各状态的说明如下表所示：

状态标识	描述
Standby	空闲状态。驱动程序初始化成功后，如果通道正常复位，应该进入本状态。如果应用程序在本状态调用 SsmPickup ，驱动程序会向对端交换机发送 ABCD=11xx
OffHook	“摘机”状态。在“摘机”状态，应用程序可以向对端交换机发送被叫号码等信息，如何向对端交换机发送被叫号码等信息，取决于交换机的类型，详细信息请向交换机厂商查询。通道进入“摘机”状态时，会启动定时器 T1
Ringing	本端“振铃”状态。在“振铃”状态，应用程序可以调用 SsmPickup 函数接受呼叫，也可以调用 SsmHangup 函数拒绝呼叫
Connected	“连接”状态，呼叫双方可以进行通话。通道迁移到“连接”状态时，驱动程序会自动开启 DTMF 检测器

Pending	“挂起”状态。需要注意的是，驱动程序不会设置挂起原因，因此，应用程序调用 SsmGetPendingReason 函数没有任何意义
Unusable	“通道不可用”状态
Reseting	“电路复位”状态

状态标识、状态值之间的对应关系如下表所示：

状态标识	值	ShpA3Api.h 对应的宏定义
Standby	0	S_CALL_STANDBY
OffHook	1	S_CALL_PICKUPED
Ringing	2	S_CALL_RINGING
Connected	3	S_CALL_TALKING
Pending	7	S_CALL_PENDING
Unusable	11	S_CALL_UNAVAILABLE
Reseting	36	S_CALL_SS1WaitIdleCAS

状态机的输出事件类型如下表所示：

事件名称	描述
E_CHG_ChState	当通道的状态发生迁移时，驱动程序向应用程序抛出本事件

1.16.6.6 SS1 通道的高级编程接口

SynCTI 驱动程序实现的呼叫状态机已经能够满足大多数的应用场合。如果应用程序希望自行实现呼叫过程，可以使用低级 API 函数。配置项 [EnableAutoCall](#) 用于关闭驱动程序实现的状态机，也可以在系统启动后通过调用函数 [SsmSetAutoCallDirection](#) 实现。驱动程序关闭 SS1 状态机后，通过收/发 CAS、R2 自行实现 SS1 协议栈及 MFC 互控过程。

有关 1 号信令接续控制的事件包括：

- [E_RCV_CAS](#)
- [E_RCV_R2](#)

有关中国 1 号信令接续控制的高级编程接口函数包括：

- [SsmSendCAS](#)
- [SsmGetSendingCAS](#)
- [SsmGetCAS](#)
- [SsmSetRxR2Mode](#)
- [SsmGetR2](#)
- [SsmSendR2](#)
- [SsmSendR2Ex](#)
- [SsmStopSendR2](#)
- [SsmGetSendingR2](#)
- [SsmEnableRxDtmf](#)

注意：

- 应用程序使用底层接续控制函数完成接续过程后，如果希望接收 DTMF 按键字符，则必须调用函数 [SsmEnableRxDtmf](#) 以开启 DTMF 接收器。同样，通话结束后也应该调用该函数以关闭 DTMF 接收器；
- 如果希望应用程序能够无缝地从 SS1 信令移植到 SS7 信令，则不应该使用本小节的函数来自行控制呼叫接续过程，而应该使用系统提供的自动呼叫接续函数。

1.16.6.7 No.1 信令的配置

使用 SS1 协议时，相关的配置项如下表所示：

配置节	配置项	
[BoardId=x]	<u>PcmNumber</u>	设置板卡上数字中继线的总数，以及每条数字中继线的信令类型、时钟的工作模式、线路的传输类型
	<u>PcmSSx</u>	
	<u>PcmClockMode</u>	
	<u>PcmLinkType</u>	
	<u>EnableAutoCall</u>	
	<u>AutoCallInTimeSlot</u>	
[PcmInfo]	<u>TotalPcm</u>	设置本机数字中继线的总数，以及建立数字中继线的逻辑编号与物理编号之间的映射关系
	<u>Pcm</u>	
[SS1Config]	<u>TxCas_CD</u>	SS1 信令 CAS、R2 接续参数设置
	<u>RxCASFilterTime</u>	
	<u>MaxWaitMfcTime</u>	
	<u>tonesgroupA</u>	
	<u>tonesgroupB</u>	
	<u>tonesendofinfo</u>	
	<u>tonesanswer</u>	
	<u>tonesrepeatrequest</u>	
	<u>IsBlockSS1In</u>	
	<u>PcmSyncMask</u>	
	<u>ProtocolType</u>	
	<u>LWaitPickup</u>	
	<u>Ss1SendIdleState</u>	
	<u>mfc2_Protocol</u>	MFCR2 协议国家选择以及相应参数设置
	<u>MaxWaitAutoDialAnswerTime</u>	SS1 去话呼叫：设置工作参数
	<u>MaxWaitOccupyAckTime</u>	

	MaxWaitKBTime	
	RepeatPhoNumOn1stR2bwdsA5	
	MfcKD	
	MfcKA	
	TxCallerId	
	MaxWaitSetKBTime	
	MaxWaitKDTime	
	ToRingingDelayTime	
	A1ToA3pWaitTime	
	MfcKB	SS1 来话呼叫：设置工作参数
	AutoSendKB	
	PhoNumHoldup	
	A3pTime	
	DefaultRcvPhoNumLen	
	DefaultRcvCallerID	
	MaxPhoNumRule	
	Rule	
	MfcR2ToRxCallerIdBuf	SS1 信令 CAS、R2 日志输出设置
	IsBlockSS1In	
[DebugView]	Ss1LogEnable	SS1 信令 CAS、R2 日志输出设置
	Ss1LogCreateMode	
	LogCreatePeriod	
	LogMaxKeep	
	LogMaxPeriod	
	LogFilepath	

注：**XXXX** 字样为必须配置的项目，**XXXX** 为可选的配置项目

1.16.7 连接到远端坐席接入模块

SHD 系列板卡可以直接连接到远端坐席接入模块（Channel bank），以构建大容量的坐席系统。在连接到远端坐席接入模块后，SHD 系列板卡的通道类型将变成坐席通道，具有 SHT 系列板卡中坐席通道的全部属性。

配置项 [UsageMode](#) 用于决定是将 SHD 系列板卡作为呼叫用板卡，还是连接到远端坐席接入模块。

配置项 [CBProtocolType](#) 用于设置连接到远端坐席接入模块时使用的信令方式。

配置项 [CBChannelType](#) 用于设置连接到远端坐席接入模块时的通道类型。

SHD 系列板卡连接到远端坐席接入模块后，通道的外部特性与对应的 SHT 系列板卡通道的外部特性完全相同，因此，通道的工作参数的配置项和配置方法也完全一样，详细信息请参见 SHT 板卡的相关内容。

1.17 SHV 系列板卡（CTI 系列）

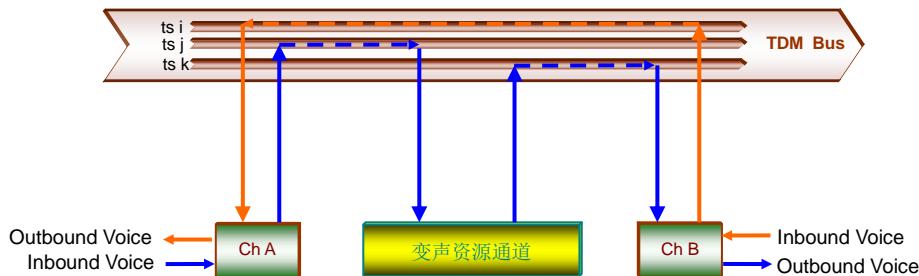
1.17.1 SHV 系列板卡简介

SHV 系列板卡支持的功能如下表所示：

Board Model	Form Factor	Max. Ports	TDM Bus
SHV-120A-CT/PCI	PCI	120	H.100
SHV-240A-CT/PCI	PCI	240	H.100
SHV-240A-CT/cPCI	cPCI	240	H.110

1.17.2 SHV 系列板卡的典型应用原理框图

SHV 系列板卡的典型应用如下图所示：



上图中，chA 与 ChB 之间通过 TDM 交换总线建立双向连接，ChB 的来话信号经时隙 i 连接到 ChA，形成 ChA 的去话信号；ChA 的来话信号经时隙 j 进入变声资源通道，ChA 的来话信号在变声资源通道内进行变声处理后，通过时隙 k 连接到 ChB，形成 ChB 的去话信号。这样，ChA 听到 ChB 的声音为原始声音，而 ChB 听到 ChA 的声音为变声后的声音。

1.17.3 有关变声功能的函数一览表

SHV 系列板卡是资源卡，必须与 SHD 系列或 SHT 系列配合使用。有关变声功能的函数如下表所示：

函数名称	描述
SsmGetMaxVCh	获取应用系统内变声资源通道的总数
SsmGetMaxFreeVCh	获取空闲的变声资源通道的总数
SsmBindVCh	将语音通道绑定到变声资源通道
SsmUnbindVCh	解除语音通道与变声资源通道的绑定
SsmSetVoiceEffect	设置变声的效果
SsmGetVoiceEffect	获取变声的效果
SsmSetVoiceEffectEx	设置 240E VAR 卡的变声参数。
SsmGetVoiceEffectEx	获取 240E VAR 卡的变声参数。

1.18 SHN 系列板卡 (CTI 系列)

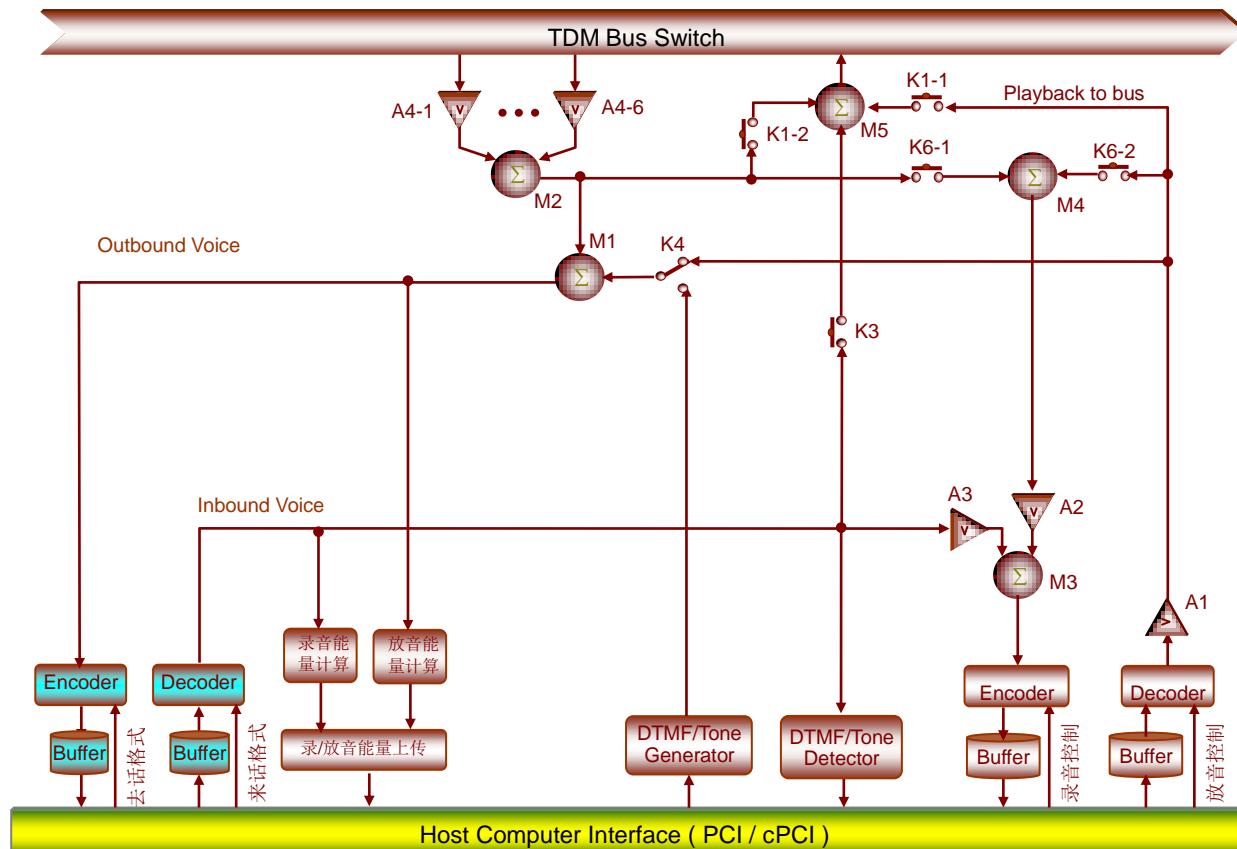
1.18.1 SHN 系列板卡简介

SHN 系列板卡支持的功能如下表所示：

Board Model	Form Factor	Max. Ports	Voice Processing Capabilities	Conferencing Capabilities	Voltage Detector	TDM Bus	Max Fax Resource	SIP	H.323
SHN-32A-CT/PCI	PCI	32	√	√	√	√	N/A	√	N/A
SHN-8B-CT/PCI+	PCI	8	√	√	√	√	N/A	√	N/A
SHN-16B-CT/PCI+	PCI	16	√	√	√	√	N/A	√	N/A
SHN-32B-CT/PCI+	PCI	32	√	√	√	√	N/A	√	N/A
SHN-60B-CT/PCI+	PCI	60	√	√	√	√	N/A	√	N/A
SHN-120B-CT/PCI+	PCI	120	√	√	√	√	N/A	√	N/A
SHN-60B-CT/PCle+	PCle	60	√	√	√	√	N/A	√	N/A
SHN-120B-CT/PCle+	PCle	120	√	√	√	√	N/A	√	N/A
SHN-120B-CT/PCle/VAR	PCle	120	√	√	√	√	N/A	√	N/A
SHN-480C-CT/PCle	PCle	480	√	√	√	√	N/A	√	N/A

1.18.2 SHN-32A-CT/PCI 板卡的工作原理框图

SHN-32A-CT/PCI 板卡的工作原理图如下图所示：



原理框图中有关录音操作的符号说明：

名称	描述
M1	去话混音器
M2	下总线信号混音器
M3	录音混音器。M3 的输出进入编码器（Encoder）模块，然后形成录音数据
M4	录音用途的去话混音器。用于将下总线信号混音器的输出、放音数据、信号音发生器/DTMF 发生器的输出进行混音，混音结果进入录音混音器 M3
M5	上总线混音器。用于将放音数据、下总线信号混音器 M2 的输出、来话信号进行混音，混音结果上行到 TDM 交换总线。 板卡初始化时，会将 M5 的输出上行到 TDM 总线的某个时隙（记为 ts）上，函数 SsmLinkToBus 可以用来改变 ts
A1	放音的音量调节器，可以通过配置项 DefaultPlayVolume 设置，也可通过函数 SsmSetPlayVolume 或 SsmSetPlayGain 进行调节，缺省的音量设置为 0
A2	去话信号进入录音混音器 M3 前的音量调节器，可以通过函数 SsmSetRecMixer 进行设置，也可以通过配置项 DefaultRecordMixerVolume 设置，缺省值-7，关闭 A2
A3	来话信号进入录音混音器 M3 前的音量调节器，可以通过函数 SsmSetRecVolume 进行设置，也可以通过配置项 DefaultRecordVolume 设置，缺省值为 0
A4-1 ... A4-6	下总线信号进入 M2 前的音量调节器，可以通过函数 SsmSetListenVlmInConf 进行设置，缺省值为 0DB
K1-1	控制放音数据是否进入混音器 M5 并进而上行到 TDM 交换总线上的开关，通常为断开状态，可以通过函数 SsmSetPlayDest 进行控制。K1-1 仅用于电话会议中给会议室播放背景音
K1-2	M2 的输出信号是否重新回到 TDM 交换总线的控制开关，可以通过函数 SsmSetFlag （携带 F_MixerResToBus 参数）进行设置，缺省值为断开
K3	来话信号是否进入上总线混音器 M5 的控制开关，可通过函数 SsmSetFlag （携带 F_InVoiceToBus 参数）或配置项 InVoiceToBus 进行设置，缺省值为闭合
K4	发送 DTMF/信号音与去话方向放音不能同时进行
K6-1 K6-2	开关 K6-1 控制总线混音器 M2 的输出是否作为录音混音器 M3 的一个输入信号源；开关 K6-2 控制放音数据是否作为录音混音器 M3 的一个输入信号源。函数 SsmSetRecBack 用来设置这 2 个开关

➤ 放音操作

应用程序调用播放函数时，解码后的语音数据经过音量调节器 A1 进行音量调节后，可以有 3 个去向：

- ◆ 进入 M1 混音器，与其它信号源混音后形成去话信号。
- ◆ 在 K6-2 控制下，进入 M4 混音器，与其它信号源混音后形成录音操作的信号源。
- ◆ 在 K1-1 控制下，进入混音器 M5，与其它信号源混音后形成上总线的信号源，仅用于电话会议中给会议室播放背景音。

➤ 录音操作的信号源选择范例

灵活应用上述录音控制开关和音量调节器可以实现很多特殊的应用。下面范例中，ch1 表示通道 1，ch2 表示通道 2。

- ◆ 范例 1：只录制 ch1 的来话信号，使用缺省音量。实现代码为：

```
SsmSetRecVolume(ch1, 0);           // 开启 A3，设置增益为 0DB
SsmSetRecMixer(ch1, FALSE, 0);     // 关闭 A2
SsmRecToFile(ch1, .....);         // 启动文件录音任务.....
```

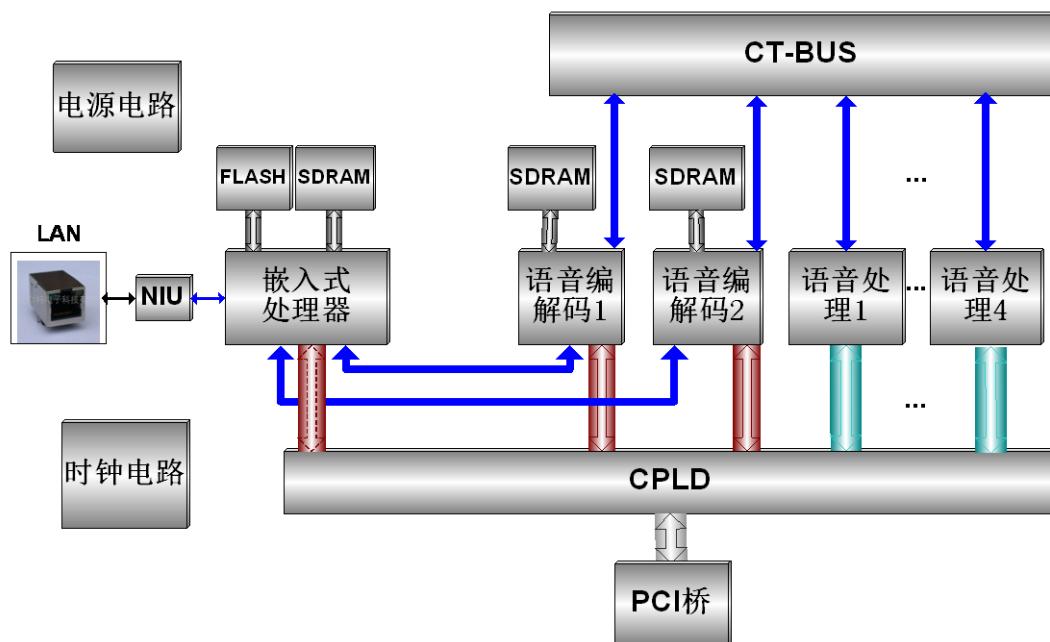
- ◆ 范例 2：ch1 与 ch2 建立双向通话后，在 ch1 上同时录制 ch1 的来话信号（音量提升 6DB）和总线混音器 M2 的输出信号（音量衰减 3DB）。实现代码为：

```
SsmTalkWith(ch1, ch2);           // 建立双向连接: ch1↔ch2
SsmSetRecBack(ch1,2);            // K6-1 闭合，K6-2 断开
SsmSetRecVolume(ch1, 2);          // 开启 A3，设置增益为 2*3DB=6DB
SsmSetRecMixer(ch1, TRUE, -1);   // 开启 A2，设置增益为-1*3DB=-3DB
SsmRecToFile(ch1, .....);        // 启动文件录音任务.....
```

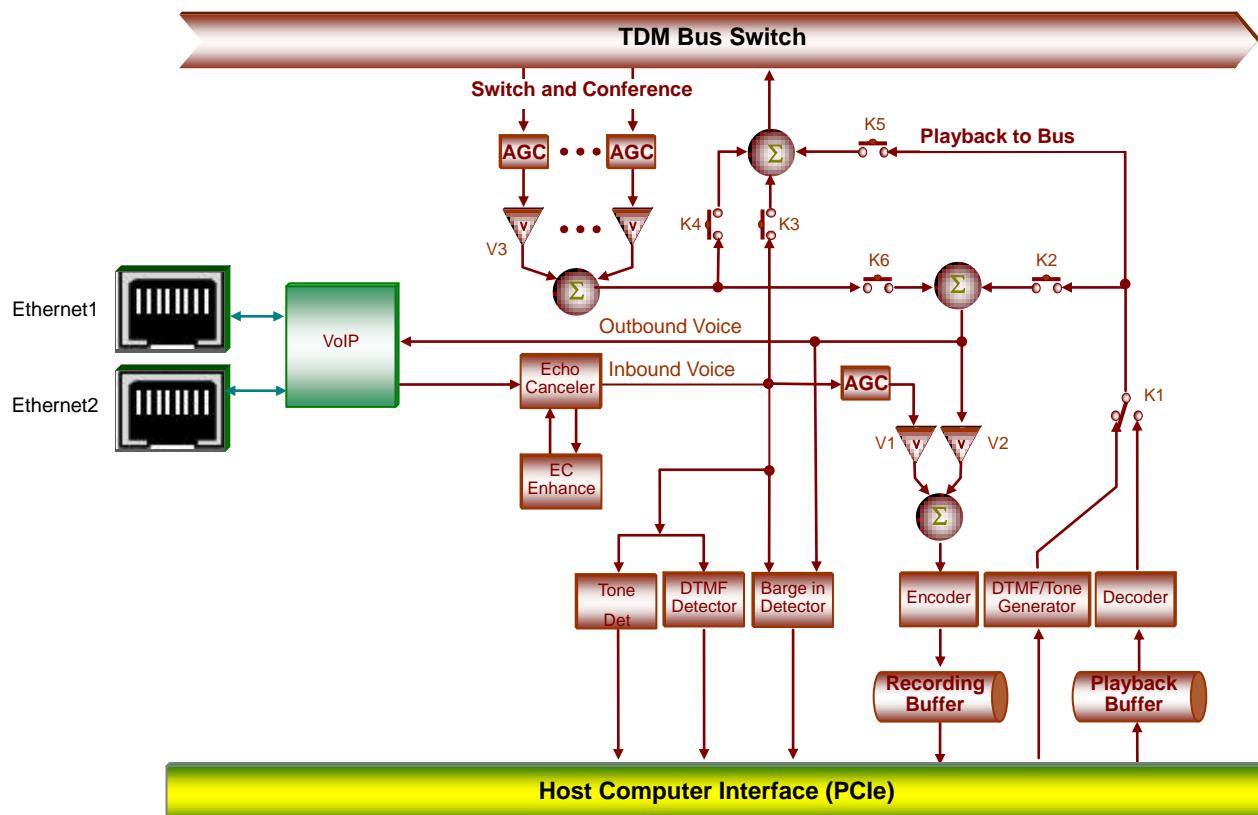
◆ 范例 3: ch1 以主持人身份加入了一个电话会议室 (假设会议室编号为 n), 在 ch1 上播放背景音乐, 同时通过 ch1 对电话会议进行录音 (包括背景音乐)。所有信号源均为正常音量, 即增益为 0。实现代码为:

```
SsmJoinConfGroup(n, ch1, .....); // ch1 加入会议室: K3 闭合
SsmSetPlayDest(int ch, 1); // 闭合 K1-1
SsmSetRecBack(ch1,1); // K6-1 闭合, K6-2 闭合
SsmPlayFile(ch1, .....); // ch1 启动放音任务: K4 连接左右两端
SsmSetRecVolume(ch1, 0); // 开启 A3, 设置增益为 0DB
SsmSetRecMixer(ch1, TRUE, 0); // 开启 A2, 设置增益为 0DB
SsmRecToFile(ch1, .....); // 启动文件录音任务.....
```

1.18.3 SHN B 系列板卡工作原理框图



1.18.4 SHN C 系列板卡工作原理框图



1.18.5 SIP 通道的编程

1.18.5.1 SIP 编程基本概念

➤ SIP 消息

有 2 种类型的 SIP 消息：

- Requests (请求消息): 从 Client 发送给 Server 的请求消息
- Responses (响应代码): 从 Server 发送给 Client 的响应消息

下表是 SHN 系列板卡 SIP 通道支持的请求消息列表：

Method	Description
INVITE	发起呼叫、修改呼叫参数(re-INVITE)
ACK	对 INVITE 最终响应的确认
BYE	结束一个会话
CANCEL	取消正在发起的呼叫
OPTIONS	查询远端的能力
REGISTER	注册
INFO	发送信息，在 SHN 系列板卡中也用于发送带外 DTMF 字符
MESSAGE	传输即时数据
SUBSCRIBE	订阅用户代理的状态
NOTIFY	通知订阅者当前状态

➤ 消息分段

SIP 消息由以下几个部分组成：

- 一个起始行

- 一个或多个头字段
- 一个标识头字段结束的空行
- 一个可选消息体

起始行、每个消息头行和空行必须以回车换行序列（CRLF）终止。注意：即使没有消息体，也必须有空行。

➤ SDP

SDP 是一个用来描述会话的协议，SIP 消息携带 SDP 的方式类似于 EMAIL 中携带文挡或是 HTTP 中携带 WEB 页面。该协议定义在 RFC2327 中，它规定了对描述会话的必要信息如何进行编码。

SDP 包通常包括：会话信息（Session information）和媒体信息（Media information）。

会话信息包括：

- 会话的主人和会话标识
- 会话激活的时间

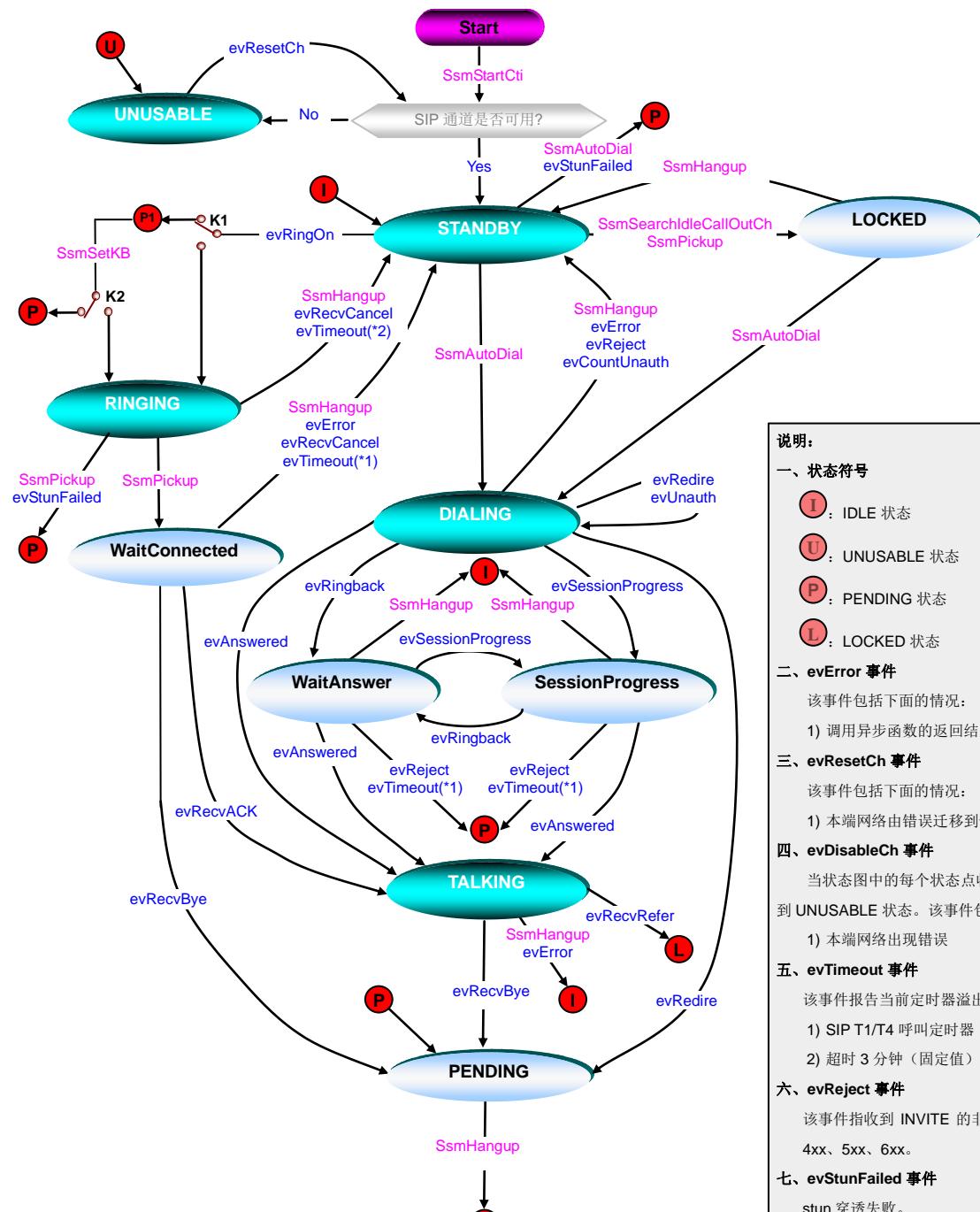
考虑到会话的资源受到会话的各方限制，因此还可以包括下面这些实用信息：

- 带宽信息
- 会话的联系信息，如：E-Mail 地址、电话号码

媒体信息包括：

- 媒体的类型，如：video、audio
- 传输的协议，如：RTP/UDP/IP、H.320
- 媒体的格式，如：H.261 video and MPEG video
- 多播的地址和传输的端口（IP multicast session）
- 联系信息中的远程地址和传输端口（IP unicast session）

1.18.5.2 SIP 通道的状态转移图



说明:

一、状态符号

- (I) : IDLE 状态
- (U) : UNUSABLE 状态
- (P) : PENDING 状态
- (L) : LOCKED 状态

二、evError 事件

该事件包括下面的情况:

- 1) 调用异步函数的返回结果显示失败

三、evResetCh 事件

该事件包括下面的情况:

- 1) 本端网络由错误迁移到恢复成功

四、evDisableCh 事件

当状态图中的每个状态点收到本事件时，都迁移到 UNUSABLE 状态。该事件包括下面的情况:

- 1) 本端网络出现错误

五、evTimeout 事件

该事件报告当前定时器溢出，定时器包括:

- 1) SIP T1/T4 呼叫定时器
- 2) 超时 3 分钟（固定值）

六、evReject 事件

该事件指收到 INVITE 的非 2xx 最终响应，包括 4xx、5xx、6xx。

七、evStunFailed 事件

stun 穿透失败。

各状态标识的说明如下表所示:

状态标识	描述
Unusable	通道“不可用”状态。 ◆ 在状态图中的任何状态点上收到 evDisableCh 事件，通道迁移到不可用状态，输出 [Unusable]。
Standby	“空闲”状态，板卡准备就绪后，进入空闲状态。 ◆ 当调用函数 SsmAutoDial 发起一个呼叫时，函数调用成功时，迁移到[DIALING]状态；

	<p>由于 stun 穿透失败导致函数调用失败时，迁移到[Pending]状态</p> <ul style="list-style-type: none"> ◆ 当调用函数 SsmSearchIdleCallOutCh 查找可用通道时，迁移到[Locked]状态 ◆ 当调用函数 SsmPickup 时，锁定一个空闲通道，并迁移到[Locked]状态 ◆ 收到 evRingOn (接收到呼入) 并有空闲通道时，应用程序通过 K1 开关的控制来决定通道的状态迁移。K1 由配置项 UseSipKB 决定。如果配置项 UseSipKB 设置为 1，通道迁移到 Pending1 (等待 APP 设置后向建立消息) 状态；如果配置项 UseSipKB 设置为 0，通道迁移到[Ringing]状态。对于处在 Pending1 (等待 APP 设置后向建立消息) 状态的通道，当调用函数 SsmSetKB 函数接受或拒绝来话呼叫时，应用程序通过 K2 开关的控制来决定通道的状态迁移。K2 由 SsmSetKB 函数的 btSigKB 参数决定。如果 btSigKB=1，通道迁移到[Ringing]状态；如果 btSigKB=2 通道迁移到 [Pending] 状态 ◆ 收到 evRingOn (接收到呼入) 但是没有空闲通道时，A3 自动发送 403 响应码并设置原因 ◆ 收到 evDisableCh (网络错误、无可用通道)，迁移到[Unusable]状态
DIALING	<p>“等待呼叫结果”状态，在收到远端的响应前将一直保持在这个状态。</p> <ul style="list-style-type: none"> ◆ 在该状态下，如果收到 evRingback 事件(收到远端发送 180 响应)，状态迁移到 [WaitAnswer] ◆ 在该状态下，如果收到 evSessionProgress 事件(收到远端发送 183 响应)，状态迁移到[SessionProgres] ◆ 如果收到 evRedire 事件(收到远端发送 3xx 响应)，SynSIP 依靠该通道进行再次呼叫，并自动发送 ACK 消息到远端 ◆ 如果收到 evRedire 事件(收到远端发送 3xx 响应)，但 3xx 消息不完整或者 SynSIP 解析 3xx 消息失败，状态迁移到[Pending] ◆ 如果收到 evUauth 事件(收到远端发送 401/407 响应)，SynSIP 自动发送带认证信息的呼叫请求到远端 ◆ 如果收到 evAnswered 事件(收到远端发送 2xx 响应)，状态迁移到[Talking]，同时 SynSIP 会自动发送 ACK 消息到远端 ◆ 如果收到 evReject(对端拒绝呼叫)、evError(拨号为异步函数)、evTimeout(定时器溢出)、evCountUnauth(认证未能通过)，状态迁移到[Standby]，同时设置原因码。此时 SynSIP 会依据当前事件(evErrAnsw 和 evCountUnauth)的需要决定是否发送 ACK 消息到远端 ◆ 如果用户在该状态下调用函数 SsmHangup 挂机，状态迁移到[Standby]，同时 SynSIP 会自动发送 CANCEL 消息到远端 ◆ 通道进入[Dialing]状态时，SynSIP 会启动定时器。如果应用程序在定时器溢出之前没有发生状态迁移，驱动程序会对此通道进行回收，将通道重新迁移到空闲状态 ◆ 收到 evDisableCh (网络错误、无可用通道)，迁移到[Unusable]状态
Locked	<p>“呼出锁定”状态。</p> <ul style="list-style-type: none"> ◆ 应用程序调用 SsmSearchIdleCallOutCh 函数时，驱动程序会将函数返回的通道临时锁定 ◆ 收到 evDisableCh (网络错误、无可用通道)，迁移到[Unusable]状态
WaitAnswer	<p>“回铃”状态。</p> <ul style="list-style-type: none"> ◆ 在该状态下，如果收到 evSessionProgress 事件(收到远端发送 183 响应)，状态迁移到[SessionProgres] ◆ 在该状态下，如果收到 evAnswered 事件(收到远端发送 2xx 响应)，状态迁移到 [Talking] ◆ 如果收到 evReject(对端拒绝呼叫)、evTimeout(定时器溢出)，状态迁移到[Pending]，同时设置原因码。此时 SynSIP 会依据当前错误原因决定是否发送 ACK 消息到远端 ◆ 用户如果在该状态下调用函数 SsmHangup 挂机，状态迁移到[Standby]，同时 SIP 栈会自动发送 CANCEL 消息到远端 ◆ 收到 evDisableCh (网络错误、无可用通道)，迁移到[Unusable]状态
SessionProgress	<p>“会话进行”状态。</p> <ul style="list-style-type: none"> ◆ 在该状态下，如果收到 evRingback 事件(收到远端发送 180 响应)，状态迁移到 [WaitAnswer] ◆ 在该状态下，如果收到 evAnswered 事件(收到远端发送 2xx 响应)，状态迁移到 [Talking] ◆ 如果收到 evReject(对端拒绝呼叫)、evTimeout(定时器溢出)，状态迁移到[Pending]，同时设置原因码。此时 SynSIP 会依据当前错误原因决定是否发送 ACK 消息到远端

	<ul style="list-style-type: none"> ◆ 用户如果在该状态下调用函数 SsmHangup 挂机, 状态迁移到[Standby], 同时 SIP 栈会自动发送 CANCEL 消息到远端 ◆ 收到 evDisableCh (网络错误、无可用通道), 迁移到[Unusable]状态
Ringing	<p>被叫“振铃”状态。当接收到呼入请求并且有空闲通道时, SynSIP 自动回应 180 响应, 锁住当前通道, 并迁移到本状态。</p> <ul style="list-style-type: none"> ◆ 在该状态下, 用户决定接受当前呼叫请求, 调用函数 SsmPickup, 函数调用成功时, 状态迁移到[WaitConnected], 此时 SynSIP 自动产生 200 OK 响应发送到对端; 由于 stun 穿透失败导致函数调用失败时, 状态迁移到[Pending]状态 ◆ 用户也可以拒绝呼叫, 此时调用函数 SsmHangup 拒绝呼叫, 状态迁移到[Standby]。此时 SynSIP 自动产生 403 Forbidden 响应发送到对端 ◆ 如果收到 evRecvCancel 事件(主叫取消呼叫), 状态迁移到[Standby]。此时 SynSIP 自动产生 200 OK 响应 CANCEL 消息 ◆ 如果收到 evTimeout (定时器溢出), 状态迁移到[Standby]。SynSIP 自动产生发送 400 错误响应 ◆ 收到 evDisableCh (网络错误、无可用通道), 迁移到[Unusable]状态
WaitConnected	<p>“等待连接”状态。</p> <ul style="list-style-type: none"> ◆ 在该状态下, 用户如果调用函数 SsmHangup 挂机, 状态迁移到[Standby]。此时 SynSIP 自动产生 BYE 消息发送到主叫 ◆ 如果收到 evRecvCancel 事件(主叫取消呼叫), 状态迁移到[Standby], SynSIP 产生 200 OK 响应相关 CANCEL 消息 ◆ 如果收到 evRecvAck 事件(主叫已连接), 状态迁移到[Talking] ◆ 如果收到 evRecvBye 事件, 状态迁移到[Pending]。SynSIP 产生 200 OK 响应 BYE 消息 ◆ 如果收到 evError、evTimeout(定时器溢出), 状态迁移到[Standby] ◆ 收到 evDisableCh (网络错误、无可用通道), 迁移到[Unusable]状态
Talking	<p>通话状态。此状态下, 音频通道都已连接。</p> <ul style="list-style-type: none"> ◆ 在此状态下, 主被叫都可通过函数 SsmHangup 挂机, 此时 SynSIP 自动产生 BYE 消息通知对方, 同时状态迁移到[Standby] ◆ 如果收到 evRecvBye 事件(对方挂机), 状态迁移到[Pending] ◆ 收到 evDisableCh (网络错误、无可用通道), 状态迁移到[Unusable] ◆ 如果收到 evRecvRefer 事件 (收到 refer 消息), 状态迁移到[Locked]
Pending	<p>“挂起”状态。</p> <ul style="list-style-type: none"> ◆ 在呼叫状态机迁移的过程中, 呼叫接收到 SIP 协议规定的被终止状态, 此时呼叫已无法建立, 用户必须在此释放呼叫过程中分配的所有资源及复位电路。必须调用函数 SsmHangup, 线路才会复位到[Standby]状态 ◆ 收到 evDisableCh (网络错误、无可用通道), 迁移到[Unusable]状态

各状态标识在头文件 ShpA3Api.h 中的声明如下表所示:

状态标识	ShpA3Api.h 中的值	ShpA3Api.h 中的宏定义
Unusable	134	S_CALL_VOIP_CHANNEL_UNUSABLE
Standby	0	S_CALL_STANDBY
Dialing	132	S_CALL_VOIP_DIALING
Locked	12	S_CALL_LOCKED
WaitAnswer	9	S_CALL_WAIT_REMOTE_PICKUP
Ringing	2	S_CALL_RINGING
WaitConnect	133	S_CALL_VOIP_WAIT_CONNECTED
Talking	3	S_CALL_TALKING
Pending	7	S_CALL_PENDING

驱动程序内部逻辑判断模块的描述如下表所示:

模块输出	判决条件
Unusable	若事件 evDisableCh 、 DataLinkDisconnect 、 Circuit Reset 之一保持有效, 则通道不可用
Standby	若 evDisableCh 事件保持无效, DataLinkConnect 事件保持有效, 且电路重起成功, 则通道状态为空闲

注意: 在此模块中, 驱动程序还要做下列清理工作:

- 清零并关闭信号音检测器；
- 如果应用程序启动了 WaitDtmf 任务，该任务将被自动终止；
- 如果应用程序启动了 DTMF 发送任务，该任务将被自动终止；
- DTMF 检测器的缓冲区将被自动清空。如果配置项 [AlwaysEnableRxDtmf](#) 设置为 0，DTMF 检测器将被自动关闭；
- 如果配置项 [AlwaysDetectBargeIn](#) 设置为 0，Barge-in 检测器将被自动关闭；
- 如果配置项 [AutoClearCallerIdBufOnHangup](#) 设置为 1，存放主叫号码的缓冲区将被自动清空；
- 所有 SynSIP 中的定时器复位。

状态机的输出事件类型如下表所示：

事件名称	描述
E_CHG_ChState	当通道的状态发生迁移时，驱动程序向应用程序抛出本事件
E_PROC_AutoDial	应用程序调用 SsmAutoDial 函数即启动了 AutoDial 任务。当 AutoDial 任务有进展时，驱动程序向应用程序抛出本事件

注：事件的声明在文件 ShpA3Api.h 中。

1.18.5.3 SIP 通道的配置

使用 SHN-32A-CT/PCI 卡的 SIP 协议时，相关的配置项如下表所示：

配置节	配置项	描述
[BoardId=x]	ProtocolType	设置板卡上使用的 IP 通信协议类型或者是否启用 IP 资源卡，以及接收 DTMF 的方式
	RecvDtmfType	
	DisplayName	设置 SIP 会话时的工作参数，依次是：
	SendDtmfType	◆ 显示名称
	AudioCodecList	◆ 发送 DTMF 的方式
	UserName	◆ 使用的语音编解码
	Register	设置与注册相关的参数，依次是：
	Domain	◆ 当启动注册时，即注册用户名；否则设置为 SIP Uri 中@前的用户名
	RegPassword	◆ 是否启动注册功能
	RegRealm	◆ 注册服务器的地址
[SIP]	RegExpires	◆ 注册时使用的密码（如果需要）
	LocalSipIp	◆ SIP 服务器别名
	LocalSipPort	◆ 注册有效时间，默认为 3600，最小值为 300
	RTPRange	设置 SIP 协议的侦听地址和端口
	LogFile	RTP 端口范围
	LogLevel	设置板卡高级配置项，依次是：
	EventThreadNum	◆ 日志保存的文件
	HeartInterval	◆ 日志等级
	SipTransportProtocol	◆ 捕获 SIP 协议事件的线程池大小
	RetransmitLost200OK	◆ 端口保持心跳间隔（用于 NAT 穿透）
		设置 SIP 协议使用 TCP 还是 UDP
		设置板卡发送 200OK 消息以后是否需要等待收到 ACK 消息后才建立通话

使用 SHN B 系列板卡和 C 系列板卡的 SIP 协议时，相关的配置项如下表所示：

配置节	配置项	
[BoardId=x]	<i>ProtocolType</i>	设置板卡上使用的 IP 通信协议类型，以及接收 DTMF 的方式
	<i>RecvDtmfType</i>	
	<i>BoardIP</i>	设置 SHN B 和 SHN C 系列板卡的网卡信息，依次是：
	<i>Submask</i>	<ul style="list-style-type: none"> ◆ B 型卡和 C 型卡的网卡 IP 地址(B 型卡和 C 型卡语音通道) ◆ 子网掩码 ◆ 网关地址 ◆ DNS 地址
	<i>Gateway</i>	
	<i>DNS</i>	
	<i>RTPRange</i>	设置 SIP 会话时的工作参数，依次是：
	<i>DisplayName</i>	<ul style="list-style-type: none"> ◆ RTP 端口范围 ◆ 显示名称 ◆ 发送 DTMF 的方式 ◆ 使用的语音编解码
	<i>SendDtmfType</i>	
	<i>AudioCodecList</i>	
[SIP]	<i>UserName</i>	设置与注册相关的参数，依次是：
	<i>Register</i>	<ul style="list-style-type: none"> ◆ 当启动注册时，即注册用户名；否则设置为 SIP Uri 中@前的用户名 ◆ 是否启动注册功能 ◆ 注册服务器的地址 ◆ 注册时使用的密码（如果需要） ◆ SIP 服务器别名 ◆ 注册有效时间，默认为 3600，最小值为 300
	<i>Domain</i>	
	<i>RegPassword</i>	
	<i>RegRealm</i>	
	<i>RegExpires</i>	
	<i>LocalSipIp</i>	设置 SIP 协议的侦听地址和端口
	<i>LocalSipPort</i>	
	<i>RTPRange</i>	RTP 端口范围
	<i>LogFile</i>	设置板卡高级配置项，依次是：
	<i>LogLevel</i>	<ul style="list-style-type: none"> ◆ 日志保存的文件 ◆ 日志等级 ◆ 捕获 SIP 协议事件的线程池大小 ◆ 端口保持心跳间隔（用于 NAT 穿透）
	<i>EventThreadNum</i>	
	<i>HeartInterval</i>	

	SipTransportProtocol	设置 SIP 协议使用 TCP 还是 UDP
	RetransmitLost200OK	设置板卡发送 200OK 消息以后是否需要等待收到 ACK 消息后才建立通话

注: **XXXX** 字样为必须配置的项目, **XXXX** 为可选的配置项目

1.18.6 IP 资源卡专用参数和结构说明

1.18.6.1 IP 资源卡 RTP 收发模式定义

```
#define IPM_SENDRECV 0x0000 //接收发送
#define IPM_SENDONLY 0x0001 //只发送
#define IPM_RECVONLY 0x0002 //只接收
```

1.18.6.2 IP 资源卡专用结构说明

```
struct MediaParam
```

```
{  
    int mode;  
    char localIP[50];  
    int localPort;  
    char remoteIP[50];  
    int remotePort;  
    int sendCodecType;  
    int dtmfpayload;  
};
```

mode: 表示 RTP 收发模式, 共有三个值:

IPM_SENDRECV(接收发送)

IPM_SENDONLY(只发送)

IPM_RECVONLY(只接收)

localIP: 表示 RTP 倾听的本地地址。

localPort: 表示 RTP 倾听的本地端口。

remoteIP: 表示 RTP 发送的远程地址。

remotePort: 表示 RTP 发送的远程端口。

SendCodecType: 表示 RTP 发送的负载编码格式。0: 711U, 8: 711A, 18: G729, 3: GSM, 4: G723_1, 9: G722, 96: AMR, 98: ILBC。

dtmfpayload: 表示 RFC2833 发送 DTMF 的值

1.18.7 IP C 型卡专用配置项说明

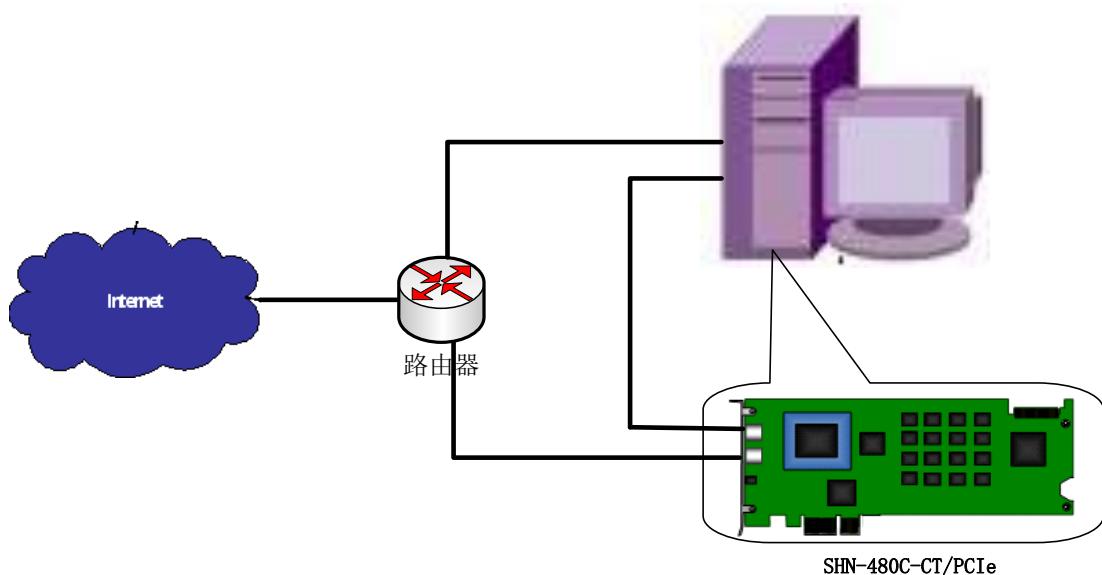
配置节	配置项	
[BoardId=x]	<i>OctMac0</i>	板卡 Ethernet 1 的 MAC 地址
	<i>OctMac1</i>	板卡 Ethernet 2 的 MAC 地址
[DHCP]	<i>DhcpServer</i>	设置 DHCP 服务器的 IP 地址
	<i>ProcessorCtrlMac</i>	设置本地主机控制消息的 MAC 地址
	<i>FilterMacRange</i>	设置可以被 DHCP 服务器分配 IP 地址的 MAC 地址
	<i>BootFileName</i>	设置 DHCP 服务器加载固件的名称

	<i>LogLevel</i>	配置 DHCP 和 TFTP 的日志，设置为 None 即关闭日志
	<i>DHCPRange</i>	配置 DHCP IP 地址范围

1.18.8 IP C 型卡的环境与使用

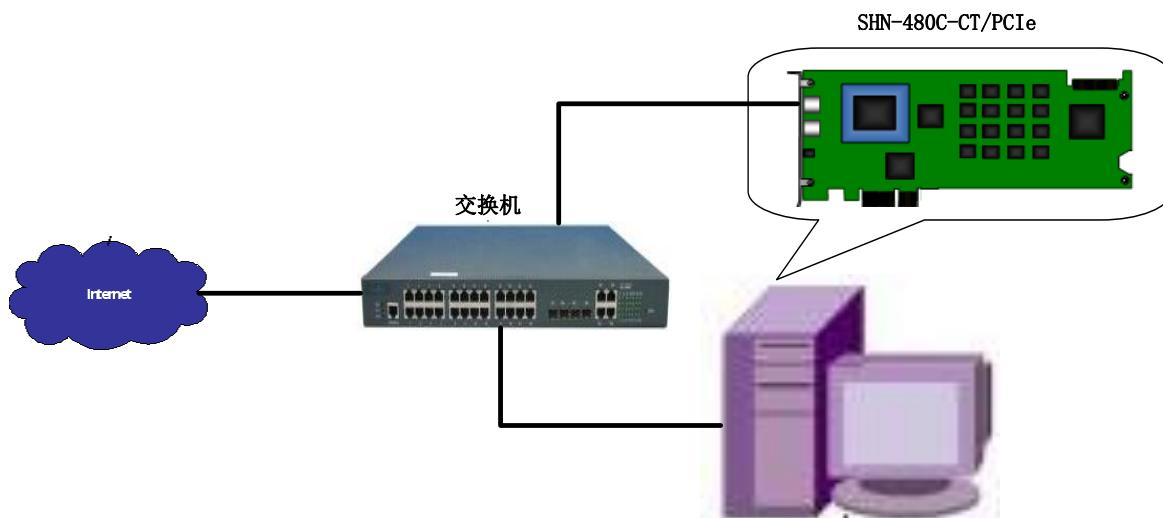
SHN-480C-CT/PCIe 卡拥有 2 个千兆网口，提供多种连接方式。

第一种接入方式（推荐使用）：



注意：此连接方式需要使用板卡的 2 个网口，其中 Ethernet1 口连接到计算机，Ethernet2 口连接到路由器。同时，要求主机至少有两个以上网口。由于板卡启动时需要通过 Ethernet1 口来加载芯片，如果 Ethernet1 口没有与计算机连接，则板卡不能正常启动。所以这种连接方式必须保证 Ethernet1 口与计算机网口直连。与 Ethernet1 连接的计算机网口必须静态配置一个 IPv4 地址，而且配置的 IP 地址最好不要与计算机其他的网口设置在同一个网段，否则会可能出现固件加载失败。

第二种接入方式：



注意：此连接方式只需使用网口 Ethernet1。由于板卡启动时需要通过 Ethernet1 口来加载芯片，如果 Ethernet1

口没有接入网线，则板卡不能正常启动。所以这种连接方式必须保证 Ethernet1 口与计算机网口连接到交换机。此处建议使用交换机，而不使用路由器。因为使用路由器可能出现因分配的 IP 有误而导致板卡不能正常启动的结果。

1.18.9 IP C 型卡的工作原理

一、板载芯片通过 Ethernet1 与计算机通信进行固件加载

因为板载芯片正常工作需要加载固件，固件采用即时加载方式，即板卡启用时才加载固件，通过外网口中的 Ethernet1（不可更改）进行固件加载。加载的固件在驱动安装成功后即存放在计算机上，需要板载芯片通过网络进行加载。板卡一旦复位或者断电，固件就必须重新加载。所以在板卡启用加载固件时必须保持外部网线正确连接到计算机上。

板卡在启动时会复位板载芯片 OCT，此时，OCT 会自动启动 TFTP server 和 DHCP client 两个线程。TFTP server 默认以 IP 192.168.0.201 开始工作，DHCP client 每隔 1 秒向所连接的网络发送 DHCP Discover。同时驱动会启动 OpenDHCPserver 和 OpenTFTPServerMT 两个进程，通过计算机网口与 OCT 的 Ethernet1 口进行交互来加载固件。

方法 A 通过驱动程序使用 OpenDHCPserver 和 OpenTFTPServerMT 两个进程来实现加载（网络中不存在其他的 DHCP server）

步骤 1：当 OpenDHCPserver 收到来自 DHCP client 的 DHCP Discover 请求后会回复一个 DHCP Offer 消息，此消息用来告知 OCT 分配到的 IP 地址（比如 192.168.10.250）、Server IP 地址和需要加载的固件名（Boot file name: oct2200.img）。配置项 **DHCPRange** 用来指定 OpenDHCPserver 可以分配的 IP 地址范围（比如 **DHCPRange=192.168.10.250-192.168.10.254**），地址范围内的 IP 地址必须保证是没有正在被使用的，否则会因 IP 冲突导致固件加载失败。配置项 **DhcpServer** 用来指定 OpenDHCPserver 的 Server IP 地址（**DhcpServer=192.168.10.10**）。配置项 **BootFileName**（**BootFileName= oct2200.img**）用来指定 OCT 可以得到的固件的固件名。

步骤 2：DHCP client 收到 Offer 消息之后，OCT 会停止 DHCP client 并修改 Ethernet1 的 IP 地址，同时检查 Offer 消息里的 Server IP 和 Boot file name 是否存在，此时分两种情况：

- 1) 如果都存在则关闭它的 TFTP server 同时通过 Ethernet1 向 Server IP 地址发送 TFTP 读请求，进入步骤 3。
- 2) 如果 Server IP 或者 Boot file name 不存在，比如 **BootFileName** 配置项被删除，那么 OCT 会把 TFTP server 的默认 IP 地址 192.168.0.201 修改为分配得到的 IP 地址（192.168.10.250），然后板卡启动 TFTP client 线程向 192.168.10.250 地址发送写请求，进入步骤 4。

步骤 3：OpenTFTPServerMT 监控到 TFTP 读请求消息之后开始向 OCT 发送 **BootFileName** 指定的固件，直到完成固件的加载。

步骤 4：TFTP server 收到 TFTP 写请求消息之后停止 DHCP client，开始收取板卡 TFTP client 发送的固件包（因为驱动程序知道要向 OCT 发送的固件所以此种方式无需指定 **BootFileName**），直到完成固件的加载。

方法 B 通过驱动程序启动 TFTP client 来实现加载（网络中存在其他的 DHCP server，比如路由器有 DHCP 功能）

这种方式必须保证携带有 DHCP 的网络设备分配的 IP 地址与主机是同一个网段，否则会导致加载失败。加载方式与**方法 A**中**步骤 2**的**2)**相似。如果计算机的 IP 地址与 TFTP server 默认的 IP 192.168.0.201 是同一个网段，那么也会使用此种方式进行加载。

注意：

- 默认以方法 **A** 进行加载。若方法 **A** 失败，则以方法 **B** 加载；如果 **BootFileName** 配置项被删除或没有配置，则以方法 **B** 进行加载。
- 计算机如果有防火墙，必须保证防火墙不会禁用 67、68、69 这三个端口，因为 DHCP 分配 IP 地址和 TFTP 数据传输时需要使用这些端口，否则会造成加载失败。解决方法可以是直接关闭防火墙。

二、SIP 信令与语音传输

不管采用何种接入方式，SIP 信令只走主机网口，语音传输（RTP）会根据接入方式的不同选择不同的板载网口。如果是第一种接入方式，那么 RTP 走 Ethernet2，主机控制 OCT 消息走 Ethernet1；如果是第二种接入方式，那么 RTP 与控制消息都走 Ethernet1。

1.19 DST 系列板卡（REC 系列）

1.19.1 DST 系列板卡简介

数字电话交换机，如 Siemens HIPath、Alcatel 4200、Alcatel 4400 系列等，与数字电话机相连时，电话线上上传输的不是模拟音频信号，而是纯数字信号。数字电话交换机与数字电话机之间的物理连接线被称为数字用户线。数字用户线由一个的 D 信道和 2 个 B 信道组成，每个信道的速率为 64kbps。D 信道用于传输与呼叫相关的信令信息，2 个 B 信道可以同时用来传输语音数据和其它数据，如传真数据等。

DST 系列板卡是用来采集数字用户线上 B 信道的语音数据和 D 信道的信令数据的专用录音卡。DST 系列板卡的线路输入部分采用了高阻电路，信号采集电路使用了现场可编程的芯片，通过应用程序的简单设置，就可以使同片板卡支持不同型号的交换机和数字电话机。

DST 系列板卡包括 A 型数字电话录音卡和 B 型数字电话录音卡。

DST 系列板卡的语音处理能力包括：

- 采用高阻方式采集 B 信道上的语音数据，不对原有通话双方造成任何干扰，也不会被原有通话双方所察觉
- 以指定的语音编码格式对语音数据进行压缩和录制
- 录音数据的自动增益控制（AGC）
- 检测 DTMF 信号
- 检测语音活动
- 检测特定频率的单音频和双音频信号音
- 实时播放采集的语音信号
- 总线交换
- 回放采集的语音数据

DST 系列 B 型板卡除具备上述功能外，还新增了下列功能：

- 多种线路故障检测，包括断线、信号电平、信噪比和帧同步
- B1、B2 语音通道切换
- 增强型板卡支持硬件处理的 GSM、G.729A 和 MP3 格式的编码

D 信道上的信令信息可以用来获取电话呼叫的相关信息。DST 系列板卡的 D 信道信令处理能力包括检测数字电话机的下列操作或状态变化：

- 摘机(Off-hook)、挂机(On-hook)

- 拨号键(Dial keys)
- 功能键(Function keys)
- 语音通道的开/关 (Voice Channel open/close)
- 指示灯(LED)
- 屏幕(LCD)上的显示内容

DST 系列板卡的型号如下表所示:

Board Model	Form Factor	Max. Ports per Board	Module Type	Max. Modules per Board
SHR-16DA-CT/PCI	PCI / PCI-X	16	MOD_16DA	4
SHR-24DA-CT/PCI	PCI / PCI-X	24	MOD_24DA	3
DST-24B/PCI	PCI	24	MOD_24DB	3 (包括一个底板固定模块)
DST-24B/PCI(SSW)	PCI	24	MOD_24DB	3 (包括一个底板固定模块)
DST-24B/PCI+	PCI	24	MOD_24DB	3 (包括一个底板固定模块)
DST-24B/PCI+(SSW)	PCI	24	MOD_24DB	3 (包括一个底板固定模块)
DST-24B/PCIe	PCIe	24	MOD_24DB	3 (包括一个底板固定模块)
DST-24B/PCIe+	PCIe	24	MOD_24DB	3 (包括一个底板固定模块)

DST 系列板卡支持的模块型号如下表所示:

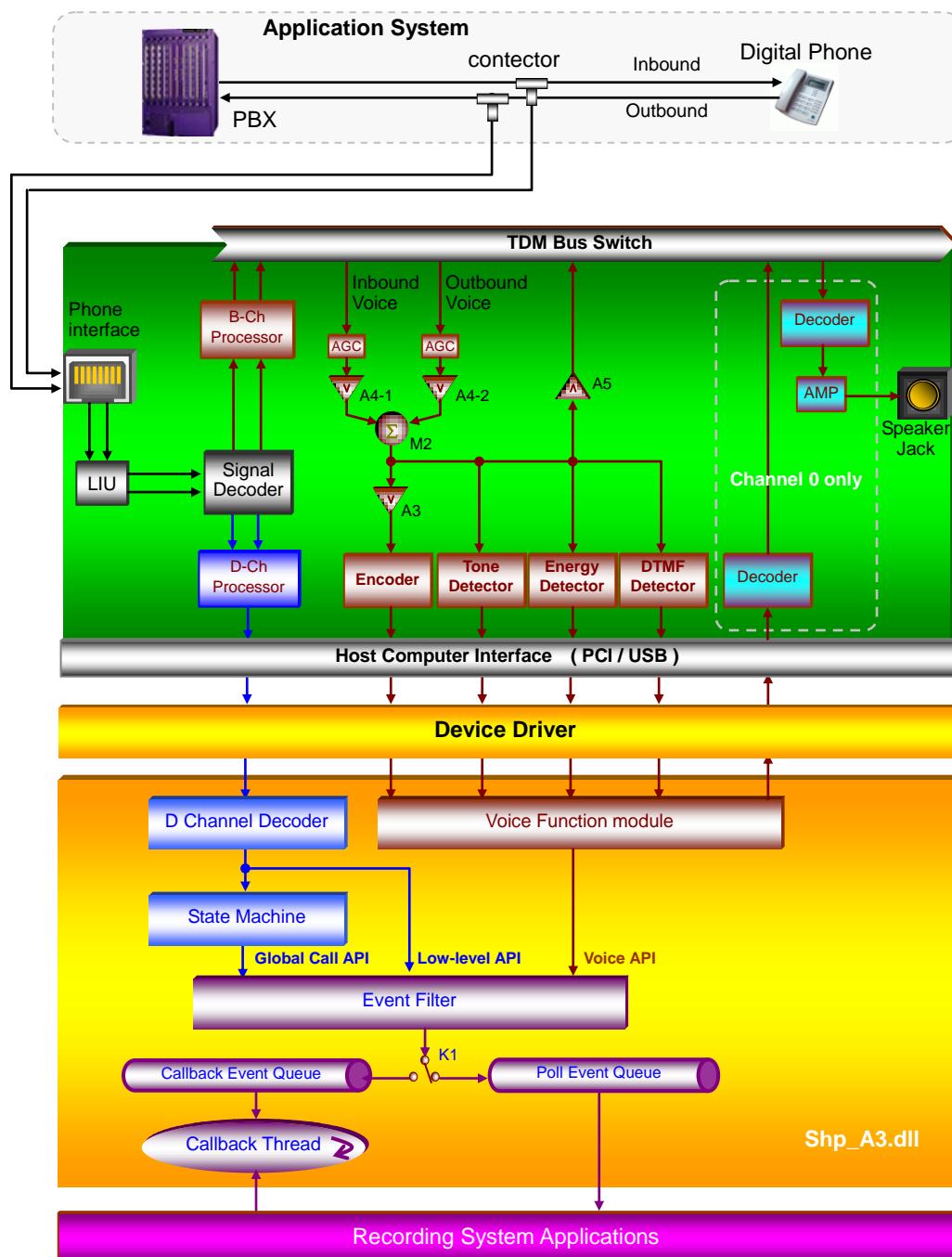
Module Type	2 线输入时的通道数	4 线输入时的通道数
MOD_16DA	4	2
MOD_24DA	8	4
MOD_24DB	8	4

1.19.2 DST 系列板卡支持的交换机型号

DST 系列板卡支持的交换机型号参见三汇公司提供的文档 DST Boards Supported PBX Models_cn。

1.19.3 DST 系列板卡的工作原理框图

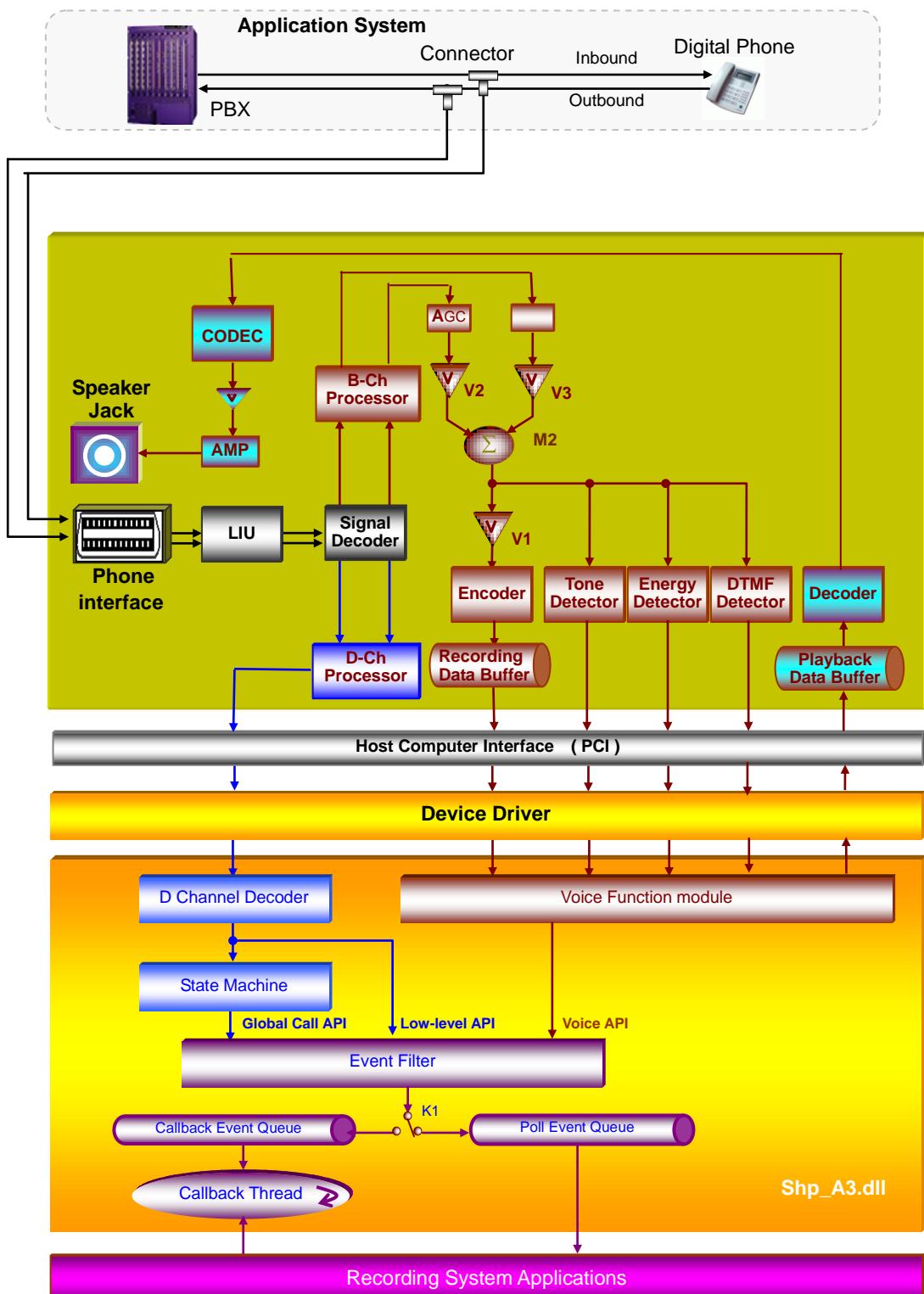
DST 系列 A 型板卡的工作原理图如下图所示:



DST 系列板卡 A 型卡原理框图中，各功能模块及符号的含义如下表所示：

名称	描述
M2	下总线信号混音器，用于将去话信号和来话信号进行混音
A4-1	来话信号（以数字话机侧为基准）的音量调节器，可以通过函数 DTRSetMixerVolume 进行调节，缺省值为 0
A4-2	去话信号（以数字话机侧为基准）的音量调节器，可以通过函数 DTRSetMixerVolume 进行调节，缺省值为 0
A3	录音音量调节器，可以通过配置项 DefaultRecordVolume 设置，也可通过函数 SsmSetRecVolume 进行调节
A5	用于将混音器的输出信号上行到总线，作为实时监听的信号源，一般情况下不需要操作 A5

DST 系列板卡 B 型卡的工作原理图：



DST 系列板卡 B 型卡原理框图中，各功能模块及符号的含义如下表所示：

名称	描述
M2	下总线信号混音器，用于将去话信号和来话信号进行混音
V-2	来话信号（以数字话机侧为基准）的音量调节器，可以通过函数 DTRSetMixerVolume 进行调节，缺省值为 0
V-3	去话信号（以数字话机侧为基准）的音量调节器，可以通过函数 DTRSetMixerVolume 进行调节，缺省值为 0
V1	录音音量调节器，可以通过配置项 DefaultRecordVolume 设置，也可通过函数 SsmSetRecVolume 进

行调节

➤ 录音操作的信号源选择范例

灵活应用上述录音控制开关和音量调节器可以实现很多特殊的应用。

- ◆ 范例 1：录制来话信号+去话信号，正常音量。实现代码为：

```
DTRSetMixerVolume(ch, 0, 0, 0); // 设置 A4-1 增益 0DB, A4-2 增益 0DB
SsmSetRecVolume(ch, 0); // 开启 A3, 设置增益 0DB
SsmRecToFile(ch, .....); // 启动文件录音任务.....
```

- ◆ 范例 2：只录制来话信号，使用缺省音量。实现代码为：

```
DTRSetMixerVolume(ch, 0, 3, -7); // 设置 A4-1 增益为 3*3DB=9DB, A4-2 增益为-∞
SsmSetRecVolume(ch, -3); // 开启 A3, 设置增益为-3*3DB=-9DB
SsmRecToFile(ch, .....); // 启动文件录音任务.....
```

➤ 放音操作的数据流向

只有 0 通道才支持放音操作。当应用程序对 0 通道调用播放函数时，驱动程序自动将放音内容上行到总线，然后再将其下行到板载音频功率放大器。

➤ 板载扬声器输出端口

每片 DST 系列板卡的 0 通道上都内置了具有模拟音频放大电路的板载扬声器输出端口，可直接将声音信号连接到外置的扬声器或耳机，其主要用途为：

- ◆ 将本卡上某个通道的来话信号（或其它板卡上某个通道的语音信号经 CT-BUS）直接送到扬声器进行播放。
函数 [SsmListenTo](#) 或 [SsmListenToEx](#) 用于实现此功能。
- ◆ 在 0 通道上调用语音播放的函数，将应用程序录制的语音数据直播放到扬声器。有关语音播放函数的详细内容请参见本章中“[语音播放](#)”部分。

原理框图中的 AMP 模块是模拟音频放大电路的功率放大器，它的增益可以通过设函数 [SsmSetPowerAmpVlm](#) 设置。

1.19.4 DST 系列板卡的配置

配置项 [PBXType](#) 和 [PhoneType](#) 分别用来设置交换机和数字电话机的型号；配置项 [DefaultVoiceFormat](#) 用来设置数字用户线上的语音编码格式。

配置项 [DEventUpdates](#) 用来过滤重复的事件。

B 型数字电话录音卡新增了一些配置项。配置项 [DstRecRawData](#) 用来设置板卡工作模式；配置项 [SetVoxChSelectEnable](#) 和 [VoxChSelect](#) 用于设置语音通道选择；[SetAnalogCtrlEnable](#)、[AnalogCtrl](#)、[SetFilterSwitchEnable](#)、[FilterSwitch](#)、[SetVoltageReferenceEnable](#) 和 [VoltageReference](#) 为高级配置项，一般情况下无需配置，如果需要对这些配置项进行配置时，请在我方技术人员指导下进行配置。

1.19.5 DST 系列板卡的语音操作

DST 系列板卡支持的语音操作包括：

➤ 录音操作

支持文件录音和内存录音，支持的语音编码格式包括：A-率、U-率、IMA ADPCM、G.729A 等。更多信息请参见本章中有关“[录音操作](#)”部分内容。

➤ 放音操作

DST 系列板卡的 0 通道都支持放音操作，包括播放单个文件和播放单个内存。放音操作支持的语音编码格式包括：A-率、U-率、IMA ADPCM、G.729A 等。更多信息请参见本章中有关“[语音播放](#)”部分内容。

➤ **实时监听**

利用 DST 系列板卡的板载扬声器输出端口可以实现对任意通道的输入语音信号进行实时监听。应用程序可以调用 [SsmListenTo](#)、[SsmListenToEx](#)、[SsmLinkFrom](#) 或 [SsmLinkFromEx](#) 启动实时监听，调用 [SsmStopListenTo](#) 或 [SsmStopLinkFrom](#) 停止实时监听。函数 [SsmSetPowerAmpVlm](#) 可以用来调节音量。注意：如果被监听的端口与连接扬声器的端口不在同一片板卡上，则需要使用 CT-BUS 连接这两片板卡。

➤ **调节输入语音信号的音量**

函数 [DTRSetMixerVolume](#) 可以用来设置交换机至话机或话机至交换机的语音信号的音量。

➤ **DTMF 检测器**

更多信息请参见本章中有关“[DTMF 检测器](#)”部分内容。

➤ **信号音检测器**

更多信息请参见本章中有关“[信号音检测器](#)”部分内容。

➤ **Barge in 检测器**

当无法解析出 D 信道上的信令信息时，可以利用 Barge in 检测器的检测结果（检测到语音活动/检测到静默）作为触发条件，来启动或停止录音等操作。更多信息请参见本章中有关“[Barge in 检测器](#)”部分内容。

1.19.6 DST 系列板卡的信令编程

D 信道上使用的信令协议没有一个统一的标准，通常由交换机或电话机制造商自行制定，不同的制造商，甚至同一个制造商生产的不同产品系列也会有很大差别。

驱动程序负责提取 D 信道上的原始信令信息，以 D 信道事件的方式将其提供给应用程序，应用程序根据驱动抛出的 D 信道事件来完成呼叫过程的分析。

1.19.6.1 使用 D 信道的原始信令消息

D 信道上的原始信令消息中，由交换机发送给话机的消息称为交换机消息，对应的事件称为交换机事件；由话机发送给交换机的消息称为话机消息，对应的事件称为话机事件。常见的交换机事件包括：

- ◊ 振铃事件
- ◊ 语音通道事件
- ◊ LCD 显示屏事件
- ◊ 指示灯事件

常见的话机事件包括：

- ◊ 摘机/挂机事件
- ◊ 按键事件

另外，当话机上发生了某些特定的操作时，话机不必向交换机报告，因此也不会向交换机发送任何消息。这样，即使驱动程序截获了 D 信道上的全部原始信令消息，也有可能无法得到话机上发生这类操作的任何信息。这类操作包括：

- ◊ 当用户操作话机上的音量调节按键时。
- ◊ 话机在 LCD 显示屏上显示某些不依赖于交换机的信息，如对某些按键的功能进行设置等。

- ◆ 话机改变了语音通道的状态，但不必向交换机报告。
- ◆ 话机上 LED 灯的状态发生改变，但不必向交换机报告。

有关 D 信道原始信令消息的详细说明，请参见《DST 事件参考手册》文档的说明。

应用程序应该根据特定型号交换机/话机的呼叫过程，利用部分或全部 D 信道原始信令消息作为状态迁移的条件，为其建立单独的呼叫状态机，然后根据通道的状态变化，来准确记录呼叫过程中的相关信息，以及相关的语音操作，如启动和停止录音操作等。

对于一些可能影响呼叫状态机的状态迁移、但无法在 D 信道原始信令消息中获得的事件，如语音信道中的忙音、回铃音等信号音，可以通过驱动程序提供的其它语音处理功能获取，更多信息请参见本章中“[语音处理](#)”部分内容。

1.20 ATP 系列板卡（REC 系列）

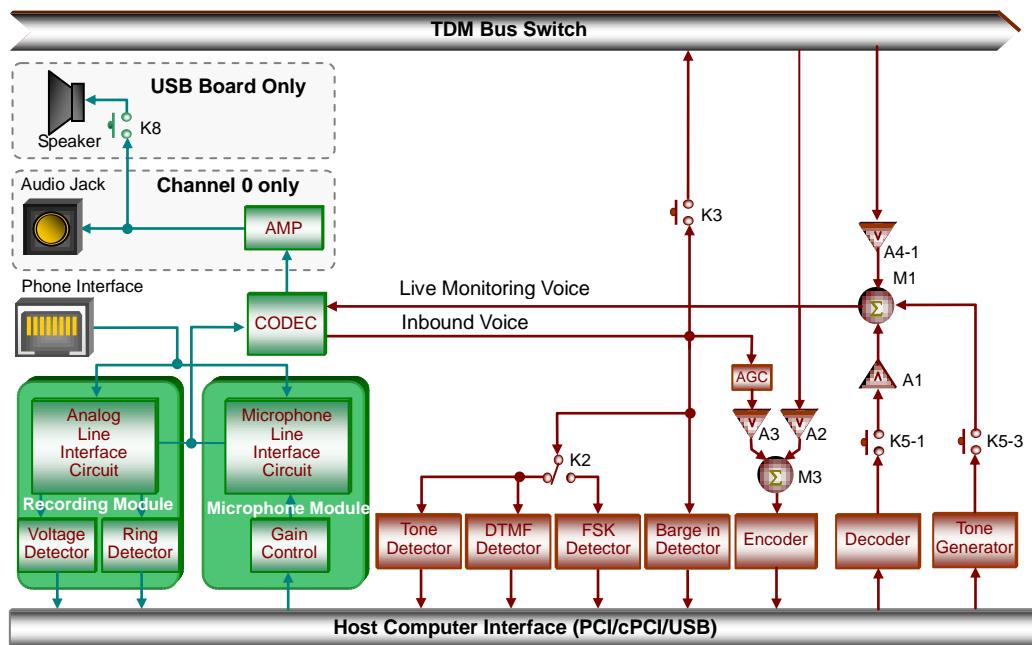
1.20.1 ATP 系列板卡简介

ATP 系列板卡支持的功能如下表所示：

Board Model	Form Factor	Max. Ports	Voltage Detector	CallerID Detector	DTMF Detector	Tone Detector	Energy Detector	TDM Bus	Audio Socket
SHT-2A/USB	USB	2	√	√	√	√	√	N/A	√
SHT-2B/USB	USB	2	√	√	√	√	√	N/A	√
SHT-4A/USB	USB	4	√	√	√	√	√	N/A	√
SHT-4B/USB	USB	4	√	√	√	√	√	N/A	√
SHT-8A/PCI	PCI	8	√	√	√	√	√	N/A	√
SHT-8B/PCI	PCI	8	√	√	√	√	√	N/A	√
SHT-16A-CT/PCI	PCI	16	√	√	√	√	√	H.100	√
SHT-16B-CT/PCI	PCI	16	√	√	√	√	√	H.100	√
SHT-16B-CT/PCI/MP3	PCI	16	√	√	√	√	√	H.100	√
ATP-24A/PCI	PCI	24	√	√	√	√	√	N/A	√
ATP-24A/PCI+	PCI	24	√	√	√	√	√	N/A	√
ATP-24A/PCle	PCle	24	√	√	√	√	√	N/A	√
ATP-24A/PCle+	PCle	24	√	√	√	√	√	N/A	√
SHT-16B-CT/cPCI	cPCI	16	√	√	√	√	√	H.110	√
SHT-16B-CT/cPCI/MP3	cPCI	16	√	√	√	√	√	H.110	√

1.20.2 ATP 系列板卡的工作原理框图

ATP 系列板卡的工作原理图如下图所示：



原理框图中有关录音操作的符号说明：

名称	描述
M1	实时监听混音器
A4-1	下总线信号的音量调节器
A1	0 通道的放音音量调节器，可通过函数 SsmSetPlayVolume/SsmSetPlayGain 或配置项 DefaultPlayVolume 进行设置，缺省值为 0
K5-1	控制放音操作的开关，由驱动程序自动控制，通常为断开状态，只有当应用程序在 0 通道上调用播放函数时，驱动程序自动开启 K5-1 开关；当放音任务结束时，驱动程序自动关闭 K5-1。有关语音播放的函数请参见 语音播放 部分内容
K5-3	信号音发生器输出控制开关，由驱动程序自动控制，通常处于断开状态，只有当应用程序调用 SsmSendTone/SsmSendToneEx 函数启动信号音发生器后，驱动程序才会自动闭合 K5-3 开关；发生器的任务终止后，K5-3 自动断开
M3	录音混音器。 M3 的输出进入编码器 (Encoder) 模块，然后形成录音数据
A3	来话信号进入录音混音器 M3 前的音量调节器，可以通过函数 SsmSetRecVolume 进行设置，也可以通过配置项 DefaultRecordVolume 设置，缺省值为 0
A2	去话信号进入录音混音器 M3 前的音量调节器，可以通过函数 SsmSetRecMixer 进行设置，也可以通过配置项 DefaultRecordMixerVolume 设置，缺省值 -7，关闭 A2。本音量调节器仅用于某些非常特殊的 应用场合
K3	来话信号是否进入上总线混音器的控制开关，通常为闭合状态，可以通过函数 SsmSetFlag (携带 <code>F_InVoiceToBus</code> 参数) 或配置项 InVoiceToBus 进行设置
K2	FSK 检测器与其它检测器之间的切换开关，通常指向其它的检测器，只有在开启 FSK 检测器期间才会指向 FSK 检测器

➤ 通道类型与业务模块

ATP 系列板卡(不包括 ATP-24A 系列板卡)可以安装 2 种类型的模块：模拟线路录音模块和麦克风录音模块，但驱动程序并不区分这 2 种模块，它们的通道类型均为模拟录音通道。配置项 [MicGain](#) 用于设置模拟录音通道输入信号的增益。对于模拟中继线录音模块，语音信号来自模拟电话线，应将输入信号的增益设置为正常增益；对于麦克风录音模块，应将输入信号的增益设置为 20DB。

➤ 放音操作

只有 0 通道才支持放音操作。应用程序对 0 通道调用播放函数时，语音数据会被送到板载音频功率放大器。

➤ 录音操作

ATP 系列板卡的录音操作非常简单，只需要控制 A3 的增益即可。

◆ 范例 1：录制 ch1 的来话信号，使用缺省音量。实现代码为：

```
SsmSetRecVolume(ch1, 0);           // 开启 A3, 设置增益为 0dB
SsmRecToFile(ch1, .....);         // 启动文件录音任务.....
```

➤ 板载扬声器输出端口

ATP 系列板卡的 0 通道具有模拟音频放大电路和板载扬声器输出端口，可直接连接到外置的扬声器或耳机，用来对语音信号进行实时监听。其主要用途为：

- ◆ 将某个通道的来话信号经 TDM 交换总线直接送到扬声器进行播放。实现此功能的函数为 [SsmListenTo](#) 或 [SsmListenToEx](#)。
- ◆ 调用语音播放的函数将应用程序录制的语音文件直接送到板载扬声器进行播放。有关语音播放函数的详细内容请参见本章中[语音播放](#)部分。

原理框图中的 AMP 模块是模拟音频放大电路的功率放大器，它的增益可以通过设函数 [SsmSetPowerAmpVlm](#) 设置。

USB 总线的录音盒（型号中带“/USB”标记）的 0 通道除了在板卡上安装了扬声器插座外，还有一个板载扬声器，可以将声音字节播放出来。开关 K8 用于选择将 0 通道的输出信号是送到扬声器插座，还是板载扬声器。K8 可以通过函数 [SsmSetLine0OutTo](#) 或配置项 [USBLine0Output](#) 进行设置，缺省为输出到扬声器插座。

注意：对于 8 通道规格的 ATP 系列板卡，由于板卡上没有 TDM 交换总线，驱动程序提供了软交换功能，仍然可以将一片板卡上的语音信号送到另外一片板卡（通常为连接了外置扬声器的板卡）上进行实时监听，但需要对配置项 [BusPlayListen](#) 进行设置。

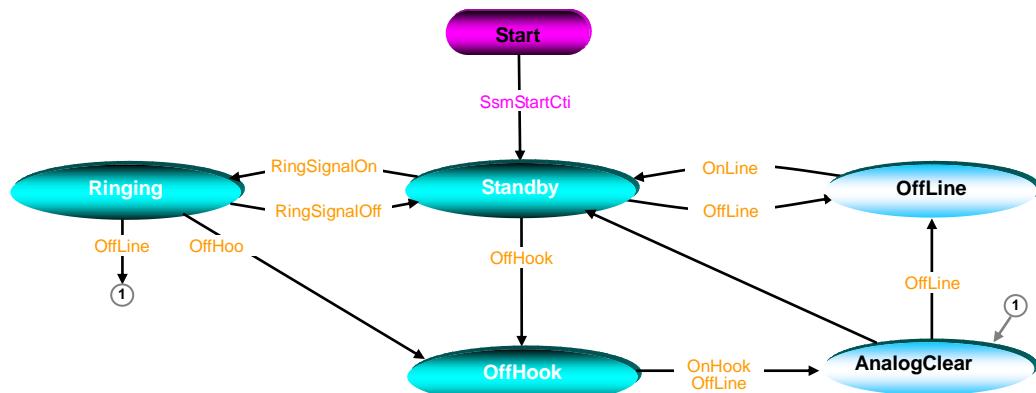
➤ 未安装业务模块的通道的特殊应用

当 ATP 系列板卡的通道上没有安装任何业务模块时，可以将此通道作为一个具有录音功能的资源通道使用，此时，通道的 Barge in 检测器、录音编码器 Encoder、下总线操作仍然能够正常工作，从而实现调用录音函数对来自 TDM 交换总线的语音数据进行录制的功能。Barge in 检测器用于提供声控控制。这项特殊应用可以通过配置项 [NoModuleChBusRec](#) 或函数 [SsmSetNoModuleChBusRec](#) 进行设置。

注意：

录音前，应该先调用函数 [SsmSetRecVolume](#) 或设置配置项 [DefaultRecordVolume](#) 将音量调节器 A3 关闭。

1.20.3 模拟中继线录音通道的状态迁移图



状态转移图中，各状态标识的说明如下表所示：

状态标识	值	宏定义	状态描述
Standby	0	S_CALL_STANDBY	空闲状态。驱动程序初始化成功后，进入本状态
OffHook	1	S_CALL_PICKUPED	摘机状态。当通道迁移到本状态时，驱动程序会： ◆ 清零并启动信号音检测器 ◆ 启动 DTMF 检测器 ◆ 如果开启了预录音功能，启动预录音操作
Ringing	2	S_CALL_RINGING	振铃状态，表示有电话正在打入。此时，应用程序可以调用函数 SsmGetCallerId 或 SsmGetCallerIdEx 获取主叫号码的信息，更多信息请参见本章中“ 模拟电话线上的主叫号码 ”部分内容
OffLine	8	S_CALL_OFFLINE	线路断开状态
AnalogClear	10	S_CALL_ANALOG_CLEAR	内部的拆线状态，只持续 8 毫秒，主要用于驱动程序执行一些清理工作，包括： ◆ 清零并关闭信号音检测器 ◆ 如果配置项 AutoClearCallerIdBufOnHangup 设置为 1，清空驱动程序内部保存 Caller ID 的缓冲区 ◆ 如果 Caller ID 的接收制式为 FSK，启动 FSK 制式的 Caller ID 检测器，关闭 DTMF 检测器；否则，启动 DTMF 检测器 ◆ 如果预录音操作仍在进行，立即关闭 ◆ 清空驱动程序内部保存 DTMF 字符的缓冲区 ◆ 启动铃流信号检测器

驱动程序自动触发的内部事件如下表示：

事件	事件描述
RingSignalOn	驱动程序检测到振铃信号。更多信息请参见本章中“ 模拟电话线上的铃流信号 ”部分内容
RingSignalOff	振铃信号消失
OffHook	被监测的电话机上发生了摘机操作。更多信息请参见本章中“ 模拟电话线的电压变化 ”部分内容
OnHook	被监测的电话机上发生了挂机操作
OffLine	被监测的电话机与交换机之间的连接断开，或者高阻复接线到板卡端口的连接断开。更多信息请参见本章中“ 模拟电话线的电压变化 ”部分内容
OnLine	被断开的连接已经恢复

1.20.4 检测摘机/挂机状态

有关录音通道如何检测摘机动作和挂机动作的详细内容请参见“[模拟电话线的电压变化](#)”部分。

1.21 DTP 系列板卡（REC 系列）

1.21.1 DTP 系列板卡简介

DTP 系列板卡支持的功能如下表所示：

Board Model	Form Factor	Trunk Type	Max. Ports	Hi-Z Line Interface Circuit	Voice Processing Capabilities	TDM Bus	DSS1	ISDN PRI	SS7	SS7 Signaling Links
SHD-30A-CT/PCI/FJ	PCI	E1	30	√	√	H.100	√	√	√	1
SHD-60A-CT/PCI/FJ	PCI	E1	60	√	√	H.100	√	√	√	2
SHD-30B-CT/PCI/FJ	PCI	E1/T1	30/23	√	√	H.100	√	√	√	1
SHD-60B-CT/PCI/FJ	PCI	E1/T1	60/46	√	√	H.100	√	√	√	2
DTP-30C/PCI	PCI	E1/T1	30/23	√	√	H.100	√	√	√	1

DTP-30C/PCI+	PCI	E1/T1	30/23	√	√	H.100	√	√	√	1
DTP-60C/PCI	PCI	E1/T1	60/46	√	√	H.100	√	√	√	2
DTP-60C/PCI+	PCI	E1/T1	60/46	√	√	H.100	√	√	√	2
DTP-120C/PCI	PCI	E1/T1	120/92	√	√	H.100	√	√	√	4
DTP-120C/PCI+	PCI	E1/T1	120/92	√	√	H.100	√	√	√	4
DTP-30C/PCle	PCle	E1/T1	30/23	√	√	H.100	√	√	√	1
DTP-30C/PCle+	PCle	E1/T1	30/23	√	√	H.100	√	√	√	1
DTP-60C/PCle	PCle	E1/T1	60/46	√	√	H.100	√	√	√	2
DTP-60C/PCle+	PCle	E1/T1	60/46	√	√	H.100	√	√	√	2
DTP-120C/PCle	PCle	E1/T1	120/92	√	√	H.100	√	√	√	4
DTP-120C/PCle+	PCle	E1/T1	120/92	√	√	H.100	√	√	√	4

上表中的语音处理能力 (Voice Processing Capabilities) 包括:

- ✓ [DTMF 检测器](#)
- ✓ [信号音检测器](#)
- ✓ [Barge in 检测器](#)
- ✓ [录音操作](#)
- ✓ [语音播放](#)
- ✓ [TDM 总线交换](#)

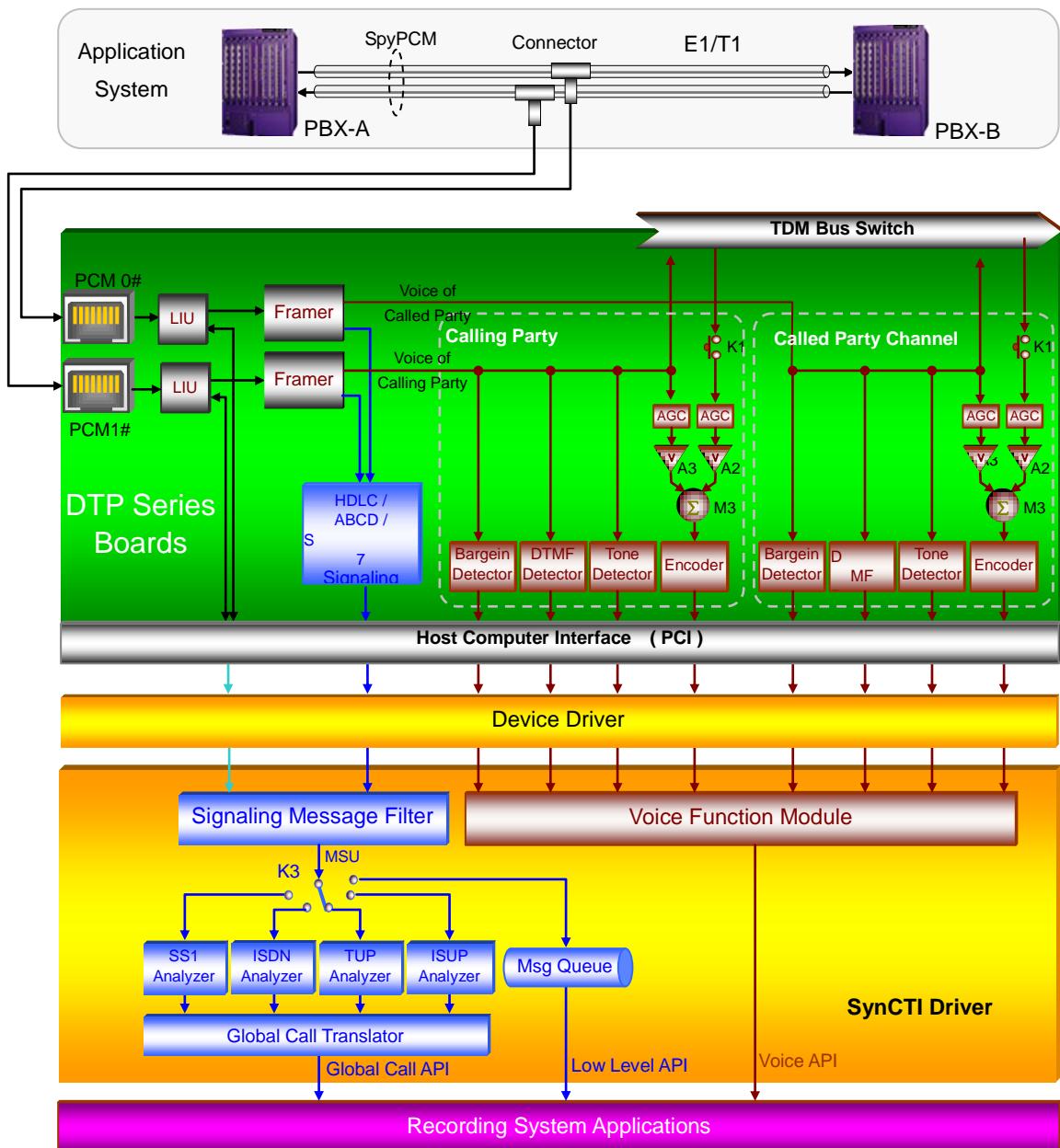
DTP 系列板卡上的线路接口单元 (LIU) 内置了高阻输入电路, 因此可以直接连接被监控的 E1/T1 线路上去, 不需要额外的高阻复接设备。

对一个 E1/T1 链路进行监控时, 需要同时采集和处理其上行和下行信号, 因此需要使用 2 个 E1/T1 接口单元, 并且只需要处理输入信号。

获取的监控信息包括 2 部分: 信令消息和语音信道数据。

1.21.2 DTP 系列板卡的工作原理框图

DTP 系列板卡的工作原理图如下图所示:



原理框图中与板卡相关的符号说明：

名称	描述
M3	录音混音器。M3 的输出进入编码器 (Encoder) 模块，然后形成录音数据
A3	来话信号进入录音混音器 M3 前的音量调节器，可以通过函数 SsmSetRecVolume 进行设置，也可以通过配置项 DefaultRecordVolume 设置，缺省值为 0
A2	去话信号进入录音混音器 M3 前的音量调节器，可以通过函数 SsmSetRecMixer 进行设置，也可以通过配置项 DefaultRecordMixerVolume 设置，缺省值 -7，关闭 A2
K3	选择信令或协议的开关，可以通过配置项 PcmSSx 进行选择。注意：驱动程序目前还不提供将原始信令消息直接通过 Msg Queue 传递给应用程序的功能

驱动平台已经通过呼叫状态机的方式实现了对 SS1、ISDN、SS7-TUP、SS7-ISUP 信令的分析功能，应用程序可以通过统一的 API 函数（即上图中的 Global Call API）实现对此状态机的控制和访问。本节主要描述与 Global Call API 有关的内容。

对于 SS7 信令的 ISUP 协议和 TUP 协议，驱动程序实现的呼叫状态机在处理 ISUP 或 TUP 消息的同时，如果

配置项 [bOpenSpySS7Mstu](#) 设置为 1，还可以将收到的原始消息送到一个内部缓冲区中。应用程序可以通过函数 [SsmGetSs7SpyMstu](#) 从该缓冲区中获取原始的 ISUP 或 TUP 消息。

语音部分的操作也非常简单。DTP 系列板卡同时为语音信道的来话信号和去话信号配置了独立的语音通道，每个语音信道都具有独立的 DTMF 检测器、信号音检测器、能量检测器、语音编码器、混音录音器等功能。

1.21.3 术语介绍

- 物理 PCM

指板卡上的具有数字中继线处理能力的硬件电路。与 SHD 系列不同的是，DTP 系列的 PCM 处理电路只处理输入信号，却没有输出信号的能力，因此，DTP 系列的卡内 PCM 工作在单工方式。

有关物理 PCM 的更详细信息请参见本章中“[数字中继线](#)”部分。

- 被监控的数字中继线 (the PCM line to be monitored, 本书简称 SpyPCM)

SpyPCM 是指被高阻复接的 E1/T1 线路，如上图所示。

由于 E1/T1 线路采用全双工的工作方式（1 收 1 发），因此，为了同时采集和处理 SpyPCM 上的上行和下行信号，在板卡上需要 2 套独立的 PCM 处理电路（即物理 PCM）来分别处理 SpyPCM 的上行和下行信号。这 2 条物理数字中继线及相关的处理电路在逻辑上组合在一起，就能够完全实现对 SpyPCM 上的语音信号和信令消息的采集和处理。

应用程序在调用相关函数时，需要提供 SpyPCM 的编号，即 SpyPCM 的逻辑编号。SpyPCM 的逻辑编号从 0 开始编号，它与板卡上的 2 条物理 PCM 之间的绑定关系由配置项 [TotalSpyPcm](#) 和 [SpyPcm](#) 确定。

当与 SpyPcm 绑定的 2 条物理 PCM 中的任何 1 条的状态发生变化时，函数 [SpyGetLinkStatus](#) 可以用来获取 SpyPcm 的状态。

- 监控电路 (Circuit in a SpyPCM, 本书简称 SpyCic)

SpyCic 是指 SpyPCM 中的某个具体的时隙（电路）。对于 SS7 信令，SpyCic 就是 TUP 或 ISUP 消息中的 CIC 字段；对于 ISDN PRI 信令和 SS1 信令，SpyCic 就是 PCM 中的时隙编号。

SpyCic 有 2 个编号：物理编号和逻辑编号。SpyCic 的物理编号，是指 SpyCic 在一个 PCM 中的编号，与该 PCM 中的时隙编号有一一对应关系，如下表所示：

SpyCic 的物理编号	0~14	15~29
对应的 PCM 中的时隙编号	1~15	17~31

SpyCic 的逻辑编号，是指将整个应用系统中全部 PCM 的 SpyCic 从 0 开始进行统一编号。驱动程序提供的关于 SpyCic 的函数均以 SpyCic 的逻辑编号作为输入参数。SpyCic 的逻辑编号与其物理编号之间的映射关系由配置文件中 [AppSpyCICTable] 节的配置项 [TotalAppSpyCIC](#) 和 [AppSpyCIC](#) 确定。函数 [SpyGetMaxCic](#) 可以用来获取应用系统中的 SpyCic 的总数。

由于 1 条 SpyPCM 与 2 条物理 PCM 绑定，因此，SpyCic 的逻辑编号与通道编号存在一一对应关系，从逻辑上看，1 个 SpyCic 由驱动程序中的 2 个通道组成。

- 被叫方通道 (Called Party Channel)、主叫方通道 (Calling Party Channel)

对于每一次电话呼叫，与 SpyCic 绑定的 2 个通道既可能作为主叫方，也可能作为被叫方。在 SynCTI 驱动程序实现的信令分析状态机中，处理主叫方送往被叫方的语音信号的通道称为主叫方通道，处理被叫方送往主叫方的语

音信号的通道称为被叫方通道。需要注意的是，主叫方通道或被叫方通道并非是固定的，跟呼叫的方向紧密相关。例如，一个通道在某次呼叫过程中是主叫方，因此是主叫方通道；下次呼叫过程中可能是被叫方，因此就会变成被叫方通道。

函数 [SpyGetCallInCh](#) 和 [SpyGetCallOutCh](#) 分别用来获取 SpyCic 所绑定的 2 个通道在本次呼叫中的主叫方通道编号和被叫方通道编号。函数 [SpyChToCic](#) 可以用来根据通道编号查询其对应的 SpyCic 的逻辑编号。

SynCTI 驱动程序实现的信令分析状态机均基于 SpyCic，但是应用程序可以对绑定的任何一个通道调用相关的语音处理函数，来获取上行或下行信号中的 DTMF 等信息。

1.21.4 DTP 系列板卡的语音功能

1.21.4.1 获取 DTMF

当基于某个 SpyCic 的呼叫建立后，调用函数 [SpyGetCallInCh](#)、[SpyGetCallOutCh](#) 可以分别获得本次呼叫的主叫方通道号和被叫方通道号，用这 2 个通道号作为参数调用有关 DTMF 检测器的相关函数，就可以获得呼叫双方的 DTMF 信号。有关 DTMF 检测器的信息请参见本章中“[DTMF 检测器](#)”部分。

1.21.4.2 语音录制

语音录制功能支持文件录音方式和内存录音方式，相关函数或事件包括：

函数或事件名称	简要描述
SpyRecToFile	基于电路编号的函数，启动文件录音任务
SpyStopRecToFile	基于电路编号的函数，停止文件录音任务
E_PROC_RecordFile	文件录音任务完成后驱动程序抛出的事件
SpyRecToMem	基于电路编号的函数，启动内存录音任务
SpyStopRecToMem	基于电路编号的函数，停止内存录音任务
E_PROC_RecordMem	内存录音任务执行期间驱动程序抛出的事件

相关配置项包括：

配置项名称	简要描述
DspCoder	设置板卡上资源 DSP 加载的编码器，使板卡可以选择更多的录音格式。（目前录音格式包括 GSM 和 G.729A）

DTP 系列板卡实现录音操作的方式有多种。假设某个 SpyCic 由主叫方通道（记为 CallingPartyCh）和被叫方通道（记为 CalledPartyCh）组成，呼叫双方进入通话状态后，需要实现对此 SpyCic 上的主叫方声音（使用缺省音量）和被叫方声音（使用缺省音量）进行混合录音，基于 SpyCic 的实现代码为：

```
SpyRecToFile(SpyCic, 2, "Voc.wav", 7, 0, 0xffffffff, 0xffffffff, 0);
```

如果需要对录音过程进行更精确的控制，可以使用基于通道的函数来实现。例如，同样实现上述功能，但希望

将主叫方声音提升 3DB，将被叫方声音衰减 3DB，则实现代码为：

```

int CalledPartyCh = SpyGetCallInCh(SpyCic);           // 获取主叫方通道编号
int CallingPartyCh= SpyGetCallOutCh (SpyCic);        // 获取被叫方通道编号
SsmListenTo(CallingPartyCh, CalledPartyCh);          // 闭合 CalledPartyCh 上的 K1 开关
SsmSetRecVolume(CalledPartyCh, 1);                   // 开启 A3，设置增益为 1*3DB=3DB
SsmSetRecMixer(CalledPartyCh, TRUE, -1);            // 开启 A2，设置增益为-1*3DB=-3DB
SsmRecToFile(CalledPartyCh, "Voc.wav",...);          // 启动文件录音任务.....

```

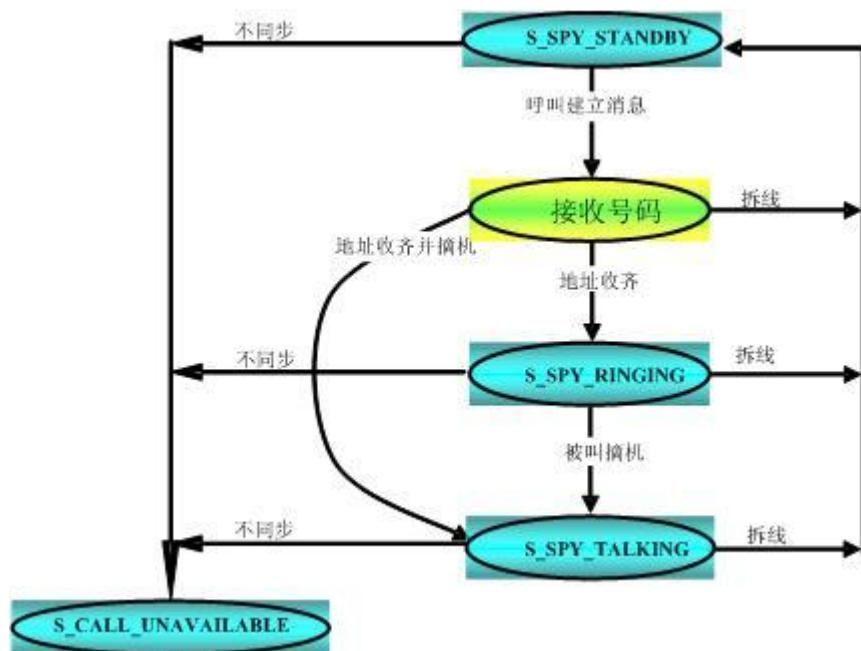
使用 G.729A 或 GSM 录音的客户需要注意的是，由于板卡使用额外的资源 DSP 实现录音，所以功能有所限制。具体表现在：

1. 同一块板卡上只能使用 G.729A 或 GSM 之一录音，无法在不同通道使用不同格式（其他例如 A 率，μ 率等不受此限制，可以任意使用）。
2. 使用 G.729A 或 GSM 录音时，只能对主叫通道或被叫通道单独录音或混音录音，不能同时录音。
3. 由于可用的总线交换方式被锁死，采用 G.729A 或 GSM 格式录音，当用上述的精准控制音量的方法时，函数 SsmListenTo 的参数必须是同一 CIC 的主叫通道和被叫通道，而像 SsmListenToEx 这类带音量的总线函数，其音量参数将被忽略。

1.21.5 驱动程序实现的状态机

1.21.5.1 数字中继录音通道的通用状态机模型

下图是 DTP 系列板卡使用 SynCTI 驱动程序提供的呼叫状态机的通用状态转移图：



每当 SpyCic 上的状态发生迁移时，驱动程序都会向应用程序抛出 [E_CHG_SpyState](#) 事件。函数 [SpyGetState](#) 可以用来获取 SpyCic 的当前状态。

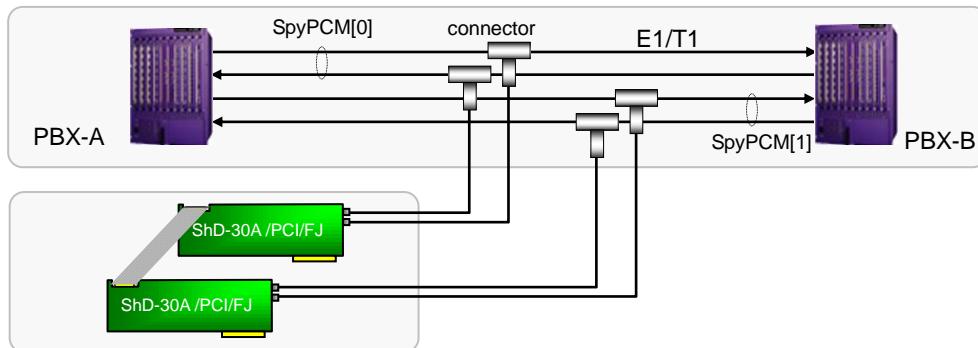
1.21.5.2 获取呼叫信息

有关获取呼叫状态机信息的相关函数或事件包括：

函数或事件名称	简要描述
E_CHG_SpyState	SS1 信令、ISDN 信令、SS7 信令 TUP 协议或 SS7 信令 ISUP 协议的状态机发生状态迁移时，驱动程序抛出的事件
E_CHG_SpyHangupInfo	ISDN 信令下，监控到线路上有拆线消息时，驱动程序抛出的事件
SpyGetState	获取 SpyCic 的状态
SpyGetCallerId	获取本次呼叫的主叫方号码
SpyGetCalleeId	获取本次呼叫的被叫方号码
SpyGetCallInCh	获取本次呼叫的主叫方通道编号
SpyGetCallOutCh	获取本次呼叫的被叫方通道编号
SpyGetHangupInfo	获取本次呼叫的挂机信息。

1.21.6 DTP 系列板卡的应用及配置实例

1.21.6.1 使用 No.1 信令/ISDN PRI 协议的系统



上图中，SpyPCM[0]使用 No.1 信令，SpyPCM[1]使用 ISDN 协议，使用 2 片 SHD-30A-CT/PCI/FJ 板卡，则应用程序的配置文件 ShConfig.ini 中的相关配置项可以按照下列方式进行设置：

[BoardId=0]

```
.....
PcmNumber=2
PcmSSx[0]=1          //No.1 信令
PcmSSx[1]=1          //No.1 信令
PcmClockMode[0]=2     //从时钟
PcmClockMode[1]=2     //从时钟
```

[BoardId=1]

```
.....
PcmNumber=2
```

```

PcmSSx[0]=0           //ISDN 协议
PcmSSx[1]=0           //ISDN 协议
PcmClockMode[0]=0     //主时钟，线路同步方式
PcmClockMode[1]=2     //从时钟
    
```

[PcmInfo]

```

TotalPcm=4
Pcm[0]=0,0
Pcm[1]=0,1
Pcm[2]=1,0
Pcm[3]=1,1
    
```

[SpyPcm]

```

TotalSpyPcm=2
SpyPcm[0]=Pcm[0],Pcm[1]
SpyPcm[1]=Pcm[2],Pcm[3]
    
```

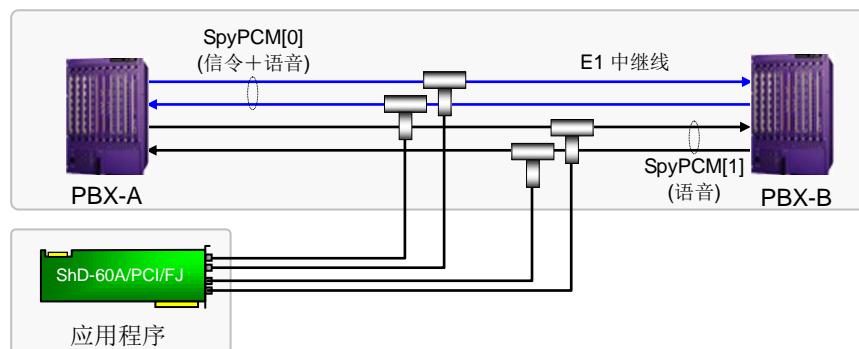
[AppSpyCICTable]

```

TotalAppSpyCIC=60      //总共 60 个 SpyCIC(语音信道)
AppSpyCIC[0]=SpyPcm[0],0..29 //SpyCIC[0]对应于通道 0 和通道 1，依此类推
AppSpyCIC[30]=SpyPcm[1],0..29
    
```

1.21.6.2 使用 SS7 信令的系统

关于监控 SS7 链路的系统，除配置上述 SS1 或 ISDN 所必需的配置项外，还需增加[SS7Spy]一节，具体有以下典型应用：



上图中，PBX-A 与 PBX-B 之间有 2 条数字中继线，SpyPCM[0]的 16 时隙为信令链路，使用 SS7 信令的 TUP 协议，SpyPCM[1]的 16 时隙不使用，其余时隙均作为语音信道使用。以 TUP 协议为例，假设 SpyPCM[0]在 TUP 消息的 CIC 中的数字中继线编号为 4，SpyPCM[1]的编号为 5，则可以按照下述方式设置应用程序的配置文件：

[BoardId=0]

.....

```

PcmNumber=4
PcmSSx[0]=7           //SS7 信令
PcmSSx[1]=7           //SS7 信令
PcmSSx[2]=7           //SS7 信令
PcmSSx[3]=7           //SS7 信令
    
```

```

PcmClockMode[0]=0           //主时钟, 线路同步方式
PcmClockMode[1]=2           //从时钟
PcmClockMode[2]=2           //从时钟
PcmClockMode[3]=2           //从时钟
    
```

[PcmInfo]

```

TotalPcm=4
Pcm[0]=0,0
Pcm[1]=0,1
Pcm[2]=0,2
Pcm[3]=0,3
    
```

[SpyPcm]

```

TotalSpyPcm=2
SpyPcm[0]=Pcm[0],Pcm[1]
SpyPcm[1]=Pcm[2],Pcm[3]
    
```

[AppSpyCICTable]

```

TotalAppSpyCIC=60           //总共 60 个 SpyCIC(语音信道)
AppSpyCIC[0]=SpyPcm[0],0..29 //SpyCIC[0]对应于通道 0 和通道 1, 依此类推
AppSpyCIC[30]=SpyPcm[1],0..29
    
```

[SS7Spy]

```

TotalSpyLinkSet=1           //总共只有 1 个信令链路组
SpyLinkSet[0]=SpyPcm[0]      //设置链路组中信令链路的物理位置
SpyLinkSet[0].SpyCICPcm[4]=SpyPcm[0]
SpyLinkSet[0].SpyCICPcm[5]=SpyPcm[1]
SpySpCodeLen=24             //设置信令点码编码标准, 中国编码为 24bit, 国际编码为 14bit
    
```

1.22 SHF 系列板卡 (CTI 系列)

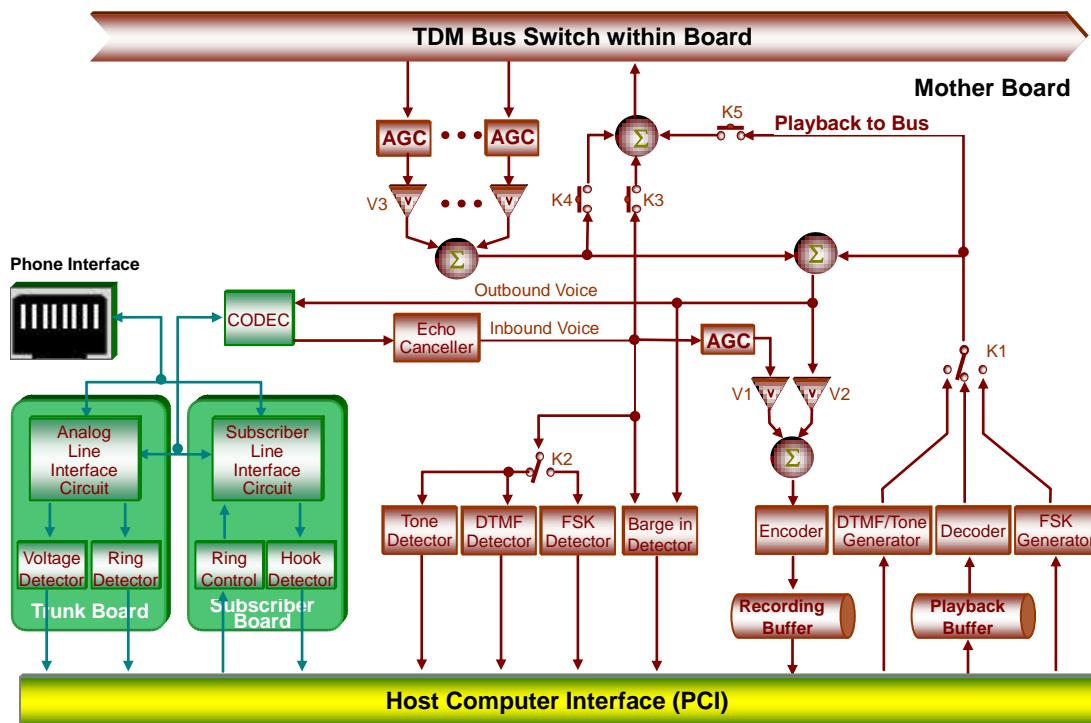
1.22.1 SHF 系列板卡简介

SHF 系列板卡支持的功能如下表所示:

Board Model	Form Factor	Max. Ports	Voice Processing Capabilities	Conferencing Capabilities	Voltage Detector	Audio Socket	FSK	TDM Bus	Max Fax Resource
SHF-2D/PCI	PCI	2	√	√	√	√	√	N/A	2
SHF-4D/PCI	PCI	4	√	√	√	√	√	N/A	4
SHF-4D/PCIe	PCIe	4	√	√	√	√	√	N/A	4

1.22.2 SHF 系列板卡的工作原理框图

SHF 系列板卡的工作原理图如下图所示:



原理框图中的模块及符号说明：

名称	描述
M1	去话混音器
M2	下总线信号混音器
M3	录音混音器。M3 的输出进入编码器 (Encoder) 模块，然后形成录音数据
M5	上总线混音器。
K1	语音播放/FSK 数据发送器的切换开关，由驱动程序自动控制。当应用程序调用播放函数时，K1 指向语音编码器模块 Decoder；当应用程序调用 SsmStartSendFSK 函数启动发送 FSK 数据时，K1 执行 FSK Generator。有关语音播放的函数请参见 语音播放 部分内容
K2	FSK 检测器与其它检测器之间的切换开关，通常指向其它的检测器，只有在开启 FSK 检测器期间才会指向 FSK 检测器
K3	来话信号是否进入上总线混音器 M5 的控制开关，可通过函数 SsmSetFlag (携带 F_InVoiceToBus 参数) 或配置项 InVoiceToBus 进行设置，缺省值为闭合
K4	M2 的输出信号是否重新回到 TDM 交换总线的控制开关，可以通过函数 SsmSetFlag (携带 F_MixerResToBus 参数) 进行设置，缺省值为断开
K5	控制放音数据是否进入混音器 M5，进而上行到 TDM 交换总线上的开关，通常为断开状态，可以通过函数 SsmSetPlayDest 进行控制。K5 只能用于电话会议中给会议室播放背景音
V1	来话信号进入录音混音器 M3 前的音量调节器，可以通过函数 SsmSetRecVolume 进行设置，也可以通过配置项 DefaultRecordVolume 设置，缺省值为 0
V2	去话信号进入录音混音器 M3 前的音量调节器，可以通过函数 SsmSetRecMixer 进行设置，也可以通过配置项 DefaultRecordMixerVolume 设置，缺省值 -7，关闭 V2
V3	下总线信号进入 M2 前的音量调节器，可以通过函数 SsmSetListenVlmInConf 进行设置，缺省值为 0dB。

● 放音操作

应用程序调用播放函数后，驱动程序自动将 K1 指向语音解码器 Decoder。解码后的语音数据可以有 2 个去向：

- 进入 M1 混音器，与其它信号源混音后形成最终的去话信号。
- 在 K5 控制下进入混音器 M5，与其它信号源混音后，形成上总线的信号源。只能用于电话会议中给会议室播放背景音。

● 录音操作的信号源选择范例

灵活应用上述录音控制开关和音量调节器可以实现很多特殊的应用。下面范例中，ch1 表示通道 1，ch2 表示通道 2。

- 范例 1：只录制 ch1 的来话信号，使用缺省音量。实现代码为：

```
SsmSetRecVolume(ch1, 0);           // 开启 V1, 设置增益为 0DB
SsmSetRecMixer(ch1, FALSE, 0);     // 关闭 V2
SsmRecToFile(ch1, .....);         // 启动文件录音任务.....
```

- 范例 2：ch1 与 ch2 建立了双向通话后，在 ch1 上同时录制 ch1 的来话信号（音量提升 6DB）+总线混音器 M2 输出信号（音量衰减 3DB）。实现代码为：

```
SsmTalkWith(ch1, ch2);           // ch1 与 ch2 通过 TDM 交换总线建立双向连接
SsmSetRecVolume(ch1, 2);          // 开启 V1, 设置增益为 2*3DB=6DB
SsmSetRecMixer(ch1, TRUE, -1);    // 开启 V2, 设置增益为-1*3DB=-3DB
SsmRecToFile(ch1, .....);        // 启动文件录音任务.....
```

- 范例 3：ch1 加入了一个电话会议室（假设会议室编号为 n），在 ch1 上播放背景音乐，同时通过 ch1 对电话会议进行录音。所有信号源均为正常音量，即增益为 0。实现代码为：

```
SsmJoinConfGroup(n, ch1, .....); // ch1 加入会议室
SsmSetPlayDest(int ch, 1);       // 闭合 K5, 将放音数据送到总线上
SsmPlayFile(ch1, .....);         // ch1 启动放音任务, 播放背景音乐
SsmSetRecVolume(ch1, 0);          // 开启 V1, 设置增益为 0DB
SsmSetRecMixer(ch1, TRUE, 0);     // 开启 V2, 设置增益为 0DB
SsmRecToFile(ch1, .....);        // 启动文件录音任务.....
```

1.23 IPR 系列产品（REC 系列）

1.23.1 IPR 系列产品简介

IPR 系列产品是用来采集 VoIP 链路的语音数据和信令数据的专用录音软件。IPR 系列产品采用端口镜像的形式采集 IP 网络上对应链路的数据，通过应用程序的简单设置，就可以使同片板卡支持不同型号的交换机和 IP 终端。

IPR 系列产品包括 SynIPAnalyzer 和 SynIPRecorder。

SynIPAnalyzer 的处理能力包括：

- ◆ 采用端口镜像的形式采集网络上的语音数据，不对原有通话双方造成任何干扰，也不会被原有通话双方所察觉
- ◆ 检测信令形式的 DTMF 信号
- ◆ 支持多种交换机/话机，多个端口进行同时监控
- ◆ 将采集到的 RTP 进行转发
- ◆ 支持终端、呼叫管理，给他们分配独立的 ID

SynIPRecorder 的处理能力包括：

- ◆ 以指定的语音编码格式对语音数据进行压缩和录制
- ◆ 可以通过函数设定录音音量
- ◆ 检测 RFC2833 形式的 DTMF 信号
- ◆ 模块分为主从模式，方便分布式部署和扩容
- ◆ 支持多种编码格式的 RTP 数据的接收

- ◆ 支持只录来话、去话或来去话混音等多种录音方式

D 信道上的信令信息可以用来获取电话呼叫的相关信息。D 信道信令处理能力包括检测 IP 电话机的下列操作或状态变化：

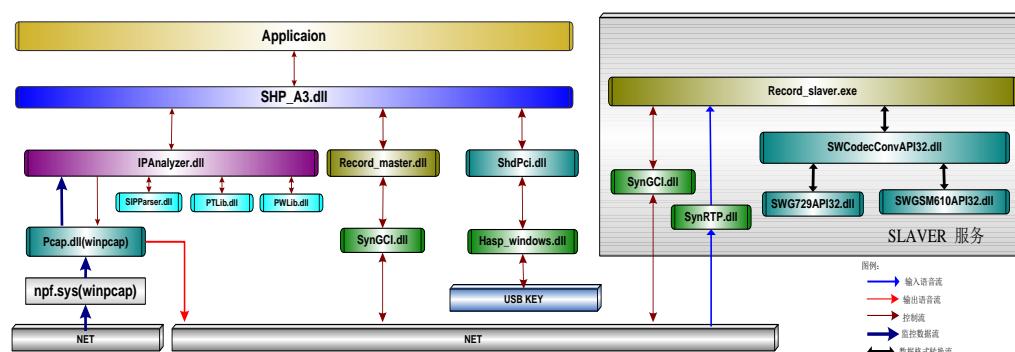
- ◆ 摘机(Off-hook)、挂机(On-hook)
- ◆ 拨号键(Dial keys)
- ◆ 功能键(Function keys)
- ◆ 语音通道的开/关 (Voice Channel open/close)
- ◆ 指示灯(LED)
- ◆ 屏幕(LCD)上的显示内容

IPR 系列产品的使用注意事项如下表所示：

类型	安装	使用	授权升级文件
USB-KEY	-	-	独立的 exe 文件
NTP-480A/PCIe	安装完成后需要重启	1、对于有 UAC(用户帐户控制)的操作系统，需要在管理员权限下运行 2、网络中其它设备的 IP 地址不允许设置为 192.168.1.7 或 192.168.1.6，否则可能引起冲突	dat 文件，使用 ShCtiConfig.exe 配置工具进行升级

1.23.2 IPR 系列产品的系统模型

IPR 系列产品的系统模型如下图所示：



上图中各模块的功能如下表所示：

模块名称	功能描述
Application	用户的应用程序
SHP_A3.DLL	为应用程序提供 API，实现板卡的各项功能
IPAnalyzer.dll	实现信令解析的核心模块，SynIPAnalyzer 的灵魂
SIPParser.dll	SIP 消息的解析模块
PTLib.dll、PWLib.dll	H.323 消息的解析模块
pcap.dll、npf.sys	数据包抓包模块
record_master.dll	SynIPRecorder 的 Master 模块，负责信令的传递、接收和解析
SynGCI.dll	通讯模块，负责底层数据的通讯
ShdPci.dll	负责跟硬件层交互的 DLL
hasp_windows.dll	与 USB KEY 交互的 DLL
USB KEY	负责授权管理

record_slaver.exe	SynIPRecorder 的 slaver 模块，负责 RTP 的接收、解码、编码、混音、录音等工作
SynRTP.dll	RTP 接收模块，主要负责 Jitter 功能
SwCodecConvAPI32.dll	语音数据的编解码主模块
SWG729API32.dll	负责 G.729 编解码的模块
SWGSM610API32.dll	负责 GSM 编解码的模块

1.23.3 SynIPAnalyzer 的相关概念

- 端口镜像 (port Mirroring)

把交换机一个或多个端口 (VLAN) 的数据镜像到一个或多个端口的方法。

- Station:

这里是指 VoIP 线路上的一个终端，它可以是软件形式的 VoIP 电话，也可以是硬件的 VoIP 话机。

- Call:

表示呼叫的整个过程，它从终端间的拨号或发送呼叫消息开始到终端挂机或发送类似挂机消息结束。一个 Station 可能产生多个 Call，比如一个手机可以同时保留多个通话。

- Session:

一次会话的建立，更具体的是一次语音流的建立，一个 Call 中可能会产生多个 Session，但这些 Session 是线性存在的，不可能同时存在。比如手机的一个通话，被叫方接通后即代表一个 Session 产生，当一方按呼叫保留键时，则这个 Session 停止，而当取消保留重新进入通话时，则又产生一个新的 Session。

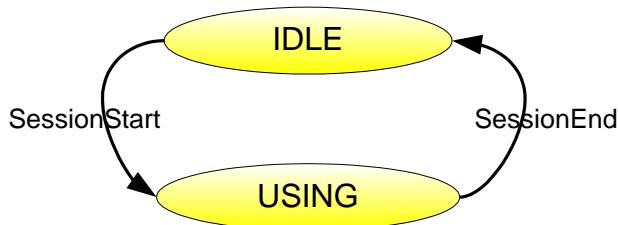
- 通道:

SynIPAnalyzer 上的通道是与 Session 对应的，当一个新的 Session 建立时，它会自动在驱动上搜索空闲通道并与之绑定，且通道进入 USING 状态，当 Session 终止时，则自动取消与通道绑定，通道也恢复为 IDLE 状态，具体可以查看 SynIPAnalyzer 的通道状态迁移图。

- 授权策略:

SynIPAnalyzer 是以通道数与同时监控的 Station 数统一授权的形式，即假如用户购买了 30 路的授权，则可用通道是 30 路，它可以同时绑定 30 个 Session，当 30 路都被占用后，新检测到的 Session 不会再发 E_RCV_IPR_MEDIA_SESSION_STARTED 事件上来，而是发 E_RCV_IPR_AUTH_OVERFLOW 表示授权已经溢出。而当前同时活动的 Station 数量无法从通道数上体现出来，可以通过 SsmIPRGetStationCount 来获取，同样当同时活动的 Station 总数超过 30 时，则也会发 E_RCV_IPR_AUTH_OVERFLOW 事件来提醒。

1.23.4 SynIPAnalyzer 的通道状态迁移图



各状态标识的说明如下表所示：

状态标识	描述
IDLE	通道“空闲”状态。

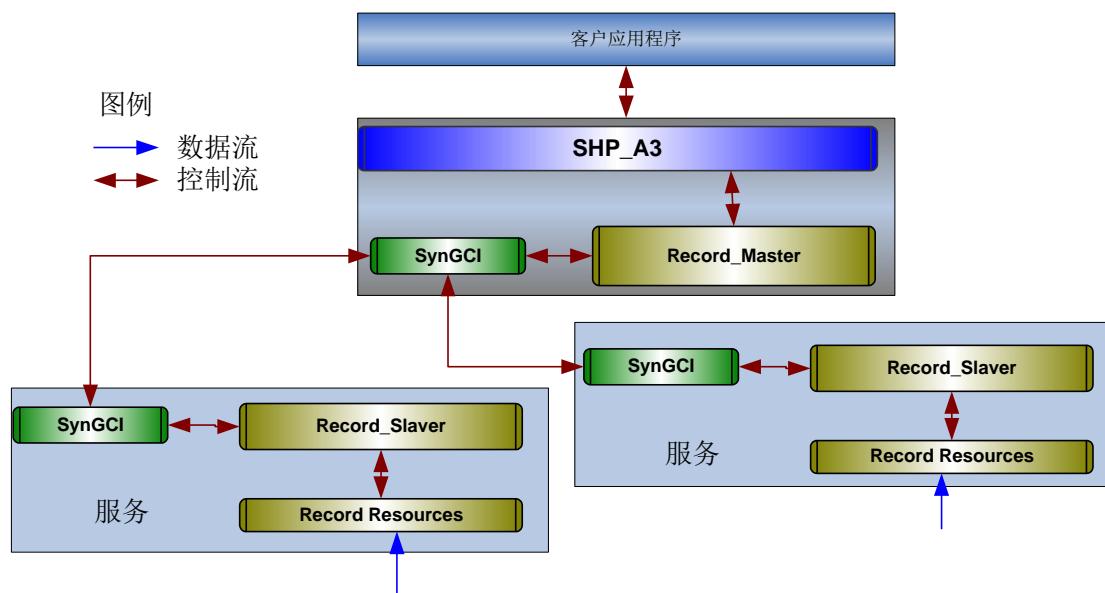
	➤ 当没有 Session 与之绑定时，通道处于空闲状态
USING	“占用”状态。 ➤ 当有 Session 与之绑定时，通道处于占用状态

1.23.5 SynIPRecorder 的相关概念

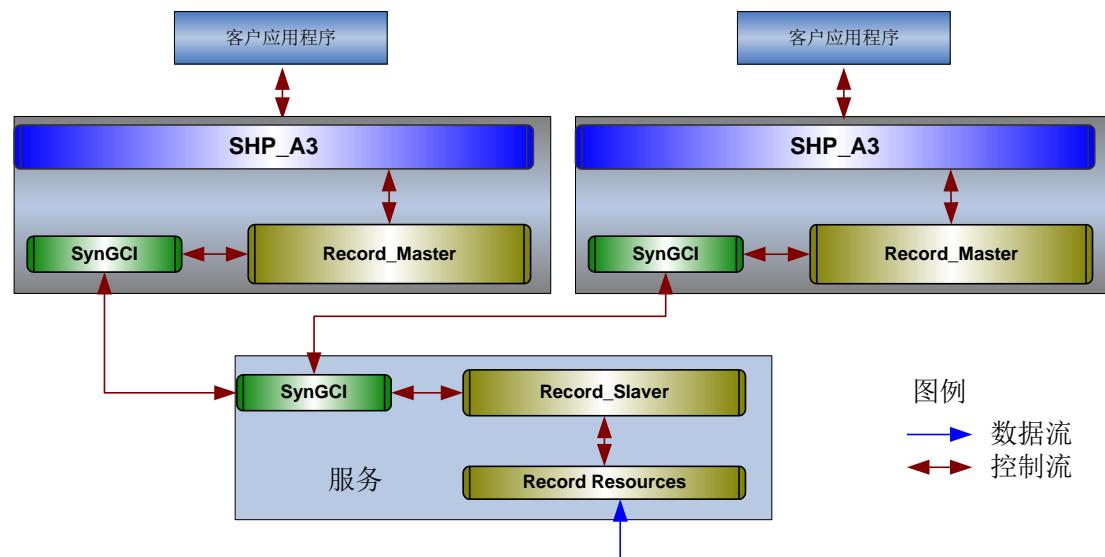
- 分布式录音：

SynIPRecorder 的 Master 端和 Slaver 端可以分布在不同机器上，统一由 TCP 协议建立通讯。SynIPRecorder 可以采用两种分布式结构，其一为一个 Master 与多个 Slaver 相连，由 Master 对 Slaver 进行统一管理，方便录音通道总数的扩展；其二为一个 Slaver 与多个 Master 相连，便于录音文件的管理。

一个 Master 与多个 Slaver 相连的分布方式见下图：



一个 Slaver 与多个 Master 相连的分布方式见下图：



- Record Master:

Master 负责所有 Slaver 的调度和通道授权管理，它通过 SHP_A3.DLL 向应用程序提供 API 使应用程序可以对

Slaver 进行控制。

- Record Slaver:

每个 Slaver 都是以服务的形式运行在一台机器上，在初始状态它只有与指定 Master 通讯的功能，需要在 Master 端调用 SsmIPRStartRecSlaver，分配一定的录音资源数和线程数后，它才处于真正的工作中，此时 Master 可以调用 SsmIPRAactiveSession 和 SsmRecToFile 函数，启动 Slaver 的一个资源进行对应 RTP 包的接收和录音。

- Record Resources:

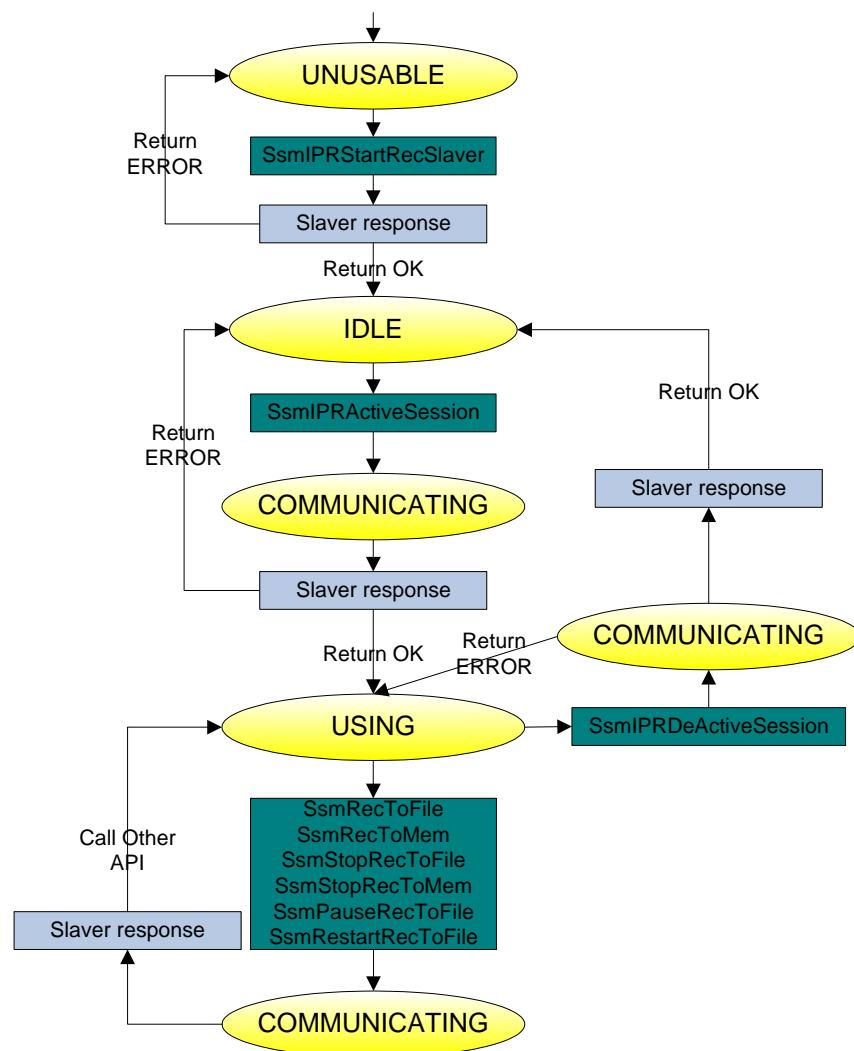
一个 Slaver 能同时进行多少个 Session 的语音由 Record Resources 的数量决定，这是一开始由 Master 指定的，每个 Resources 可以认为是一个独立的录音单元。

- Session:

这里的 Session 概念与 SynIPAnalyzer 的一致，其实就是从 SynIPAnalyzer 上转发过来的 Session。

授权策略：SynIPRecorder 也以通道数进行授权，统一在 Master 端进行管理。假如授权数是 30，则一个 Master 可以同时进行 30 路的录音，但与 Master 相连的 Slaver 可以不受限制，Master 可以决定这 30 路的授权在不同的 Slaver 上怎么分配。比如 Master 连接了 2 个 Slaver（Slaver A 和 Slaver B），它可以决定 Slaver A 分配 10 个 Record Resources，而 Slaver B 分配 20 个 Record Resources。

1.23.6 SynIPRecorder 的通道状态迁移图



状态标识	描述
UNUSABLE	通道“不可用”状态。 ➢ 当没有连接 Slaver 或没在 Slaver 端创建录音资源时通道为不可用
IDLE	“空闲”状态。 ➢ 表示这个通道可以被用来录音
COMMUNICATING	“通讯中”状态。 ➢ 当 Master 向 Slaver 端发送请求并等待 slaver 的回复时，通道会处在通讯中状态
USING	“使用中”状态。 ➢ 表示这个通道正在被一个 Session 占据，不能再处理其他 Session

1.23.7 IPR 系列产品的语音操作

IPR 系列产品支持的语音操作包括：

- 录音操作

支持的语音录音编码格式包括：A-率、μ-率、IMA ADPCM、G729、GSM、MP3 等。

- RTP 数据接收操作

支持 RTP 解码格式包括：A-率、μ-率、G.729、G.722、G723、GSM 等。

- 调节语音信号的录音音量

函数 [SsmIPRSetRecVolume](#) 可以用来设置交换机至话机或话机至交换机的语音信号的音量。

1.23.8 IPR 系列产品的信令编程

D 信道上使用的信令协议没有一个统一的标准，通常由交换机或电话机制造商自行制定，不同的制造商，甚至同一个制造商生产的不同产品系列也会有很大差别。

驱动程序负责提取 D 信道上的原始信令信息，以 D 信道事件的方式将其提供给应用程序，应用程序根据驱动抛出的 D 信道事件来完成呼叫过程的分析。

1.23.8.1 使用 D 信道的原始信令消息

D 信道上的原始信令消息中，由交换机发送给话机的消息称为交换机消息，对应的事件称为交换机事件；由话机发送给交换机的消息称为话机消息，对应的事件称为话机事件。常见的交换机事件包括：

- ◆ 振铃事件
- ◆ 语音通道事件
- ◆ LCD 显示屏事件
- ◆ 指示灯事件

常见的话机事件包括：

- ◆ 摘机/挂机事件
- ◆ 按键事件

另外，当话机上发生了某些特定的操作时，话机不必向交换机报告，因此也不会向交换机发送任何消息。这样，即使驱动程序截获了 D 信道上的全部原始信令消息，也有可能无法得到话机上发生这类操作的任何信息。这类操作包括：

- 当用户操作话机上的音量调节按键时。
- 话机在 LCD 显示屏上显示某些不依赖于交换机的信息，如对某些按键的功能进行设置等。
- 话机改变了语音通道的状态，但不必向交换机报告。
- 话机上 LED 灯的状态发生改变，但不必向交换机报告。

有关 D 信道原始信令消息的详细说明，请参见《D 信道事件参考手册》文档的说明。

应用程序应该根据特定型号交换机/话机的呼叫过程，利用部分或全部 D 信道原始信令消息作为状态迁移的条件，为其建立单独的呼叫状态机，然后根据通道的状态变化，来准确记录呼叫过程中的相关信息，以及相关的语音操作，如启动和停止录音操作等。

对于一些可能影响呼叫状态机的状态迁移、但无法在 D 信道原始信令消息中获得的事件，如语音信道中的忙音、回铃音等信号音，可以通过驱动程序提供的其它语音处理功能获取。

1.23.9 IPR 系列产品支持的协议类型及相关参数

1.23.9.1 IPR 通用的宏定义和结构体

#define IP_TCP 6 // TCP 协议

```
#define IP_UDP 17 // UDP 协议
#define IPR_MAX_NON_STATION_LIST 25 //非 Station 列表的上限
#define IPR_MAX_ADDITIONAL_PTL_PORT_LIST 20 //一个协议同时监控的端口上限
#define IPR_MAX_H323_PTL_PORT_LIST IPR_MAX_ADDITIONAL_PTL_PORT_LIST //H323 同时被监控的端口上限
#define IPR_MAX_SIP_PTL_PORT_LIST IPR_MAX_ADDITIONAL_PTL_PORT_LIST //SIP 同时被监控的端口上限
typedef struct
{
    UCHAR Protocol; //传输类型(i.e., MT_TCP or MT_UDP)
    USHORT Port; // 端口号
}IPR_MONITOR_ITEM;
```

1.23.9.2 支持的协议

协议类型	协议 ID	宏定义
SIP	0	PTL_SIP
CISCO SCCP	1	PTL_CISCO_SKINNY
AVAYA H323	2	PTL_AVAYA_H323
SHORTEL MGCP	3	PTL_SHORTEL_MGCP
H323	4	PTL_H323
PANASONIC MGCP	5	PTL_PANASONIC_MGCP
TOSHIBA MEGACO	6	PTL_TOSHIBA_MEGACO
SIEMENS H323	7	PTL_SIEMENS_H323
ALCATEL	8	PTL_ALCATEL
MITEL	9	PTL_MITEL
LG Nortel	10	PTL_LG_NORTEL
Samsung	11	PTL_SAMSUNG
Tadicom MGCP	12	PTL_TADICOM_MGCP
Zenitel	13	PTL_ZENITEL
Nortel UNISTIM	14	PTL_NORTEL_UNISTIM

1.23.9.3 协议配置结构体

```
typedef struct
{
    IPR_MONITOR_ITEM Transport; // SIP 主要的监控端口和传输类型
    ULONG ProxyIPAddress; // SIP Proxy/ALG/PBX IP Address
    USHORT NonStationListCount; //非 Station IP 的数量
    ULONG NonStationList [IPR_MAX_NON_STATION_LIST]; //非 Station IP 的地址
    USHORT TransportAdditionalCount; //SIP 额外的监控端口数量
    IPR_MONITOR_ITEMTransport_Additional[IPR_MAX_SIP_PTL_PORT_LIST]; // SIP 额外的监控端口
```

```
DWORD dwSpecial; //用于辅助处理，目前支持设置为 1，用于支持 sip/aasp 协议
BOOL bMixCSProtocol; //当呼叫过程中混合使用 UDP 和 TCP 协议时使用
} IPR_SIP_CFGS;
```

1.23.9.3.2 CISCO SCCP 协议的配置结构体

```
typedef struct
{
    IPR_MONITOR_ITEM SCCP; // SCCP 的监控端口和传输类型
} IPR_SCCP_CFGS;
```

1.23.9.3.3 AVAYA H323 协议的配置结构体

```
typedef struct
{
    IPR_MONITOR_ITEM H225CS; //AVAYA H323 的监控端口和传输类型
    IPR_MONITOR_ITEM H225RAS; //AVAYA H323 注册管理端口与传输类型
    USHORT NonStationListCount; //非 Station 的 IP 数量
    ULONG NonStationList [IPR_MAX_NON_STATION_LIST]; //非 Station 的 IP 地址
    USHORT H225CSAdditionalCount; //AVAYA H323 额外的监控端口数量
    IPR_MONITOR_ITEM H225CS_Additional[IPR_MAX_H323_PTL_PORT_LIST];
    // AVAYA H323 额外监控端口
    USHORT H225RASAdditionalCount; //AVAYA H323 额外注册管理端口数量
    IPR_MONITOR_ITEM H225RAS_Additional[IPR_MAX_H323_PTL_PORT_LIST];
    //AVAYA H323 额外注册管理端口
} IPR_H323_CFGS;
```

1.23.9.3.4 SHORTEL MGCP 协议的配置结构体

```
typedef struct
{
    IPR_MONITOR_ITEM CallAgent;
    //Shortel MGCP 话机（Agent）端口与传输类型
    IPR_MONITOR_ITEM Gateway;
    //Shortel MGCP 交换机（Gateway）端口与传输类型
} IPR_SHORTEL_MGCP_CFGS;
```

1.23.9.3.5 H323 协议的配置结构体

```
typedef struct
{
```

```
IPR_MONITOR_ITEM H225CS; //H323 的监控端口和传输类型
IPR_MONITOR_ITEM H225RAS; //H323 注册管理端口与传输类型
USHORT NonStationListCount; //非 Station 的 IP 数量
ULONG NonStationList [IPR_MAX_NON_STATION_LIST]; //非 Station 的 IP 地址
USHORT H225CSAdditionalCount; //H323 额外的监控端口数量
IPR_MONITOR_ITEM H225CS_Additional[IPR_MAX_H323_PTL_PORT_LIST];
//H323 额外监控端口
USHORT H225RASAdditionalCount; //H323 额外注册管理端口数量
IPR_MONITOR_ITEM H225RAS_Additional[IPR_MAX_H323_PTL_PORT_LIST];
//H323 额外注册管理端口
}IPR_H323_CFGS;
```

1.23.9.3.6 PANASONIC MGCP 协议的配置结构体

```
typedef struct
{
    IPR_MONITOR_ITEM CallAgent;
    //Panasonic MGCP 话机 (Agent) 端口与传输类型
    IPR_MONITOR_ITEM Gateway;
    //Panasonic MGCP 交换机 (Gateway) 端口与传输类型
}IPR_PANASONIC_MGCP_CFGS;
```

1.23.9.3.7 TOSHIBA MEGACO 协议的配置结构体

```
typedef struct
{
    IPR_MONITOR_ITEM Megaco_H248; // Toshiba MEGACO 端口与传输类型
}IPR_TOSHIBA_MEGACO_CFGS;
```

1.23.9.3.8 Siemens H323 协议的配置结构体

```
typedef struct
{
    IPR_MONITOR_ITEM Proprietary; // Siemens H323 专用参数
    IPR_MONITOR_ITEM H225CS; // Siemens H323 H225CS 参数
}IPR_SIEMENS_H323_CFGS;
```

1.23.9.3.9 Alcatel 协议的配置结构体

```
typedef struct
{
```

```
    IPR_MONITOR_ITEM Alcate;      // Alcate 参数
}IPR_ALCATEL_CFGS;
```

1.23.9.3.10 Mitel 协议的配置结构体

```
typedef struct
{
    IPR_MONITOR_ITEM Mitel;      // Mitel 参数
}IPR_MITEL_CFGS;
```

1.23.9.3.11 LG Nortel 协议的配置结构体

```
typedef struct
{
    IPR_MONITOR_ITEM LTPS;        // LG Nortel parameters of "Line Terminal Proxy Server"
    IPR_MONITOR_ITEM Phone;       // LG Nortel parameters of IP Phone
    USHORT NonStationListCount;
    ULONG NonStationList[IPR_MAX_NON_STATION_LIST];
}IPR_LG_NORTEL_CFGS;
```

1.23.9.3.12 Samsung 协议的配置结构体

```
typedef struct
{
    IPR_MONITOR_ITEM Samsung;     // Samsung Parameters
}IPR_SAMSUNG_CFGS;
```

1.23.9.3.13 Tadicom MGCP 协议的配置结构体

```
typedef struct
{
    IPR_MONITOR_ITEM CallAgent; // Tadicom MGCP parameters of Call Agent side
    IPR_MONITOR_ITEM Gateway; // Tadicom MGCP parameters of Agent side
}IPR_TADICOM_MGCP_CFGS;
```

1.23.9.3.14 Zenitel 协议的配置结构体

```
typedef struct
{
    IPR_MONITOR_ITEM Zenitel; // Zenitel Parameters
```

```
}IPR_ZENITEL_CFGS;
```

1.23.9.3.15 Nortel UNIStim 协议的配置结构体

```
typedef struct
{
    IPR_MONITOR_ITEM NortelUNIStim; // Nortel UNIStim 参数
}IPR_NORTEL_UNISTIM_CFGS;
```

1.23.9.4 使用 SynIPAnalyzer 与 SynIPRecorder 搭建的 VoIP 录音系统编程举例

```
//一、初始化板卡及注册回调函数
//Step 1、启动板卡及注册回调函数
EVENT_SET_INFO EventMode;
EventMode.dwWorkMode = EVENT_CALLBACKA; //使用事件回调的形式
EventMode.lpHandlerParam = EventCallback; //回调函数
EventMode.dwUser = (DWORD)this;
int nIsSsmStartCtiOK = SsmStartCtiEx("ShConfig.ini", "ShIndex.ini", true, &EventMode);
if(nIsSsmStartCtiOK != 0)
{
    SsmGetLastErrMsg(szErrMsg);
    AfxMessageBox(szErrMsg, MB_OK, 0);
    return FALSE;
}
//Step 2、获取板卡总数
if(SsmGetMaxUsableBoard() != SsmGetMaxCfgBoard())
{
    SsmGetLastErrMsg(szErrMsg);
    AfxMessageBox(szErrMsg, MB_OK, 0);
}
//Step 3、获取通道总数
MaxLine=SsmGetMaxCh();
nIPRChNum = 0;
nUsedIPRChNum = 0;
nIPAChNum = 0;
bUpdateSlaverDisplay = FALSE;
//Step 4、获取每个通道的通道类型
for(int i = 0; i < MaxLine; i++)
{
    if(SsmGetChType(i) == IPRR_CH)
    {
        nIPRChNum++;
    }
    else if(SsmGetChType(i) == IPRA_CH)
```

```
{  
    nIPACHNum++;  
}  
}  
  
//二、回调函数实现  
int EventCallback(PSSM_EVENT pEvent)  
{  
    BOOL bFind = FALSE;  
    int i, nResult, nPtlType, nStationId;  
    char szEvtName[100], szIPP[50], szIPS[50], szTemp[100];  
    pIPR_SessionInfo pSessionInfo;  
    PIPR_CISCO_SCCP_CALL_INFO pSCCPIInfo;  
    PIPR_CALL_INFO pCallInfo;  
    SYSTEMTIME time;  
  
    nResult = 0;  
    pIPRRecorderDlg->bErrorOccurred = FALSE;  
    switch(pEvent->wEventCode)  
    {  
        case E_CHG_ChState:  
            //SynIPAnalyzer 或 SynIPRecorder 通道状态改变  
            UpdateChannelState();  
            break;  
        case E_RCV_IPR_DChannel:  
            //SynIPAnalyzer 获取到 IPR 的 D 信道事件  
            //根据 D 信道事件更新相关信息  
            if(pEvent->dwParam == DST_CISCO_SCCP_CALL_INFO)  
            {  
                //CISCO 的 CALL INFO 事件，更新 CALL 信息  
                nPtlType = pEvent->dwXtralInfo >> 16;  
                nStationId = pEvent->dwXtralInfo & 0xffff;  
                pSCCPIInfo = (PIPR_CISCO_SCCP_CALL_INFO)pEvent->pvBuffer;  
                //更新 CALL 信息  
                //...  
            }  
            else if(pEvent->dwParam == DST_CALL_SUSPENDED || pEvent->dwParam ==  
DST_CALL_RELEASED)  
            {  
                //呼叫释放，更新呼叫信息  
                nPtlType = pEvent->dwXtralInfo >> 16;  
                nStationId = pEvent->dwXtralInfo & 0xffff;  
                pCallInfo = (PIPR_CALL_INFO)pEvent->pvBuffer;  
            }  
    }  
}
```

```
break;

case E_RCV_IPR_STATION_ADDED:
    //SynIPAnalyzer 检测到新 Station 加入
    break;

case E_RCV_IPR_STATION_REMOVED:
    //SynIPAnalyzer 检测到 Station 退出
    break;

case E_RCV_IPR_AUTH_OVERFLOW:
    //SynIPAnalyzer 或 SynIPRecorder 授权溢出
    break;

case E_RCV_IPR_MEDIA_SESSION_STARTED:
    //SynIPAnalyzer 检测到新 Session 产生
    //Step 1、先检查是否有 Slaver 连上，否则无法录音
    if(nSlaverCount > 0)
    {
        //从事件中获取 Session 相应信息
        pSessionInfo = (pIPR_SessionInfo)pEvent->pvBuffer;

        wsprintf(szIPP, "%d.%d.%d.%d", pSessionInfo->PrimaryAddr.S_un_b.s_b1,
pSessionInfo->PrimaryAddr.S_un_b.s_b2,
            pSessionInfo->PrimaryAddr.S_un_b.s_b3, pSessionInfo->PrimaryAddr.S_un_b.s_b4);
        wsprintf(szIPS, "%d.%d.%d.%d", pSessionInfo->SecondaryAddr.S_un_b.s_b1,
pSessionInfo->SecondaryAddr.S_un_b.s_b2,
            pSessionInfo->SecondaryAddr.S_un_b.s_b3, pSessionInfo->SecondaryAddr.S_un_b.s_b4);

        //Step 2、找到 SynIPRecorder 的空闲通道
        for(i=0; i<MaxLine; i++)
        {
            if(hannelInfo[i].nChType == IPRR_CH)
            {
                ChannelInfo[i].nWorkState = SsmGetChState(i);
                if(ChannelInfo[i].nWorkState == S_CALL_STANDBY)
                {
                    bFind = TRUE;
                    break;
                }
            }
        }
        if(!bFind)
            break;

        //Step 3、调用 SynIPRecorder 的 API SsmIPRAciveSession
        //将 Session 信息传给 SynIPRecorder，并指定接收该 Session 的 Slaver,
        //以及获取 Slaver 端接收该 Session 的端口
```

```
if(SsmIPRActiveSession(i, IPR_SlaverAddr[0].nRecSlaverID,
                      pSessionInfo->dwSessionId,
                      szIPP, pSessionInfo->PrimaryAddr.usPort,
                      &pSessionInfo->nFowardingPPort,
                      pSessionInfo->nPrimaryCodec,
                      szIPS, pSessionInfo->SecondaryAddr.usPort,
                      &pSessionInfo->nFowardingSPort,
                      pSessionInfo->nSecondaryCodec) != 0)
{
    SsmGetLastErrMsg(szErrMsg);
    nResult = -1;
}

//Step 4、将 RTP 转发端口保存到对应的 IPR 通道，调用 SsmIPRSendSession 时使用
ChannelInfo[pEvent->nReference].nFowardingPPort = pSessionInfo->nFowardingPPort;
ChannelInfo[pEvent->nReference].nFowardingSPort = pSessionInfo->nFowardingSPort;
}
break;

case E_RCV_IPR_AUX_MEDIA_SESSION_STARTED:
//该事件不用管理，它表示重复检测到了 E_RCV_IPR_MEDIA_SESSION_STARTED 事件;
break;

case E_RCV_IPR_MEDIA_SESSION_STOPED:
//SynIPAnalyzer 检测到新 Session 停止
//Step 1、从事件中获取 Session 信息
pSessionInfo = (pIPR_SessionInfo)pEvent->pvBuffer;

//Step 2、找到与之绑定的 SynIPAnalyzer 通道
for(i=0; i<MaxLine; i++)
{
    if(ChannelInfo[i].dwSessionId == pSessionInfo->dwSessionId
       && ChannelInfo[i].nChType == IPRA_CH)
    {
        bFind = TRUE;
        break;
    }
}
if(!bFind)
    break;

//Step 3、调用 SynIPAnalyzer 的 API SsmIPRStopSendSession
//停止 RTP 转发
SsmIPRStopSendSession(i);

//Step 4、找到接收该 RTP 的 SynIPRecorder 通道
```

```
for(i=0; i<MaxLine; i++)
{
    if(ChannelInfo[i].dwSessionId == pSessionInfo->dwSessionId
        && ChannelInfo[i].nChType == IPRR_CH)
    {
        ChannelInfo[i].dwSessionId = 0;
        ChannelInfo[i].nPtlType = -1;
        ChannelInfo[i].nStationId = -1;
        bFind = TRUE;
        break;
    }
}
if(!bFind)
    break;

//Step 5、调用 SynIPRecorder 的 API SsmStopRecToFile
//告诉 Slaver 停止录音
if(SsmStopRecToFile(i) != 0)
{
    SsmGetLastErrMsg(pIPRecorderDlg->szErrMsg);
    pIPRecorderDlg->bErrorOccurred = TRUE;
    nResult = -1;
}
break;

case E_IPR_SLAVER_INIT_CB:
    //Slaver 刚与 Master 连接时，对 Master 的初始化消息的回复
    break;
case E_IPR_START_SLAVER_CB:
    //Slaver 对 Master 调用 SsmIPRStartRecSlaver 函数的回复
    UpdateChannelState();
    break;
case E_IPR_CLOSE_SLAVER_CB:
    //Slaver 对 Master 调用 SsmIPRCloseRecSlaver 函数的回复
    break;

case E_IPR_ACTIVE_SESSION_CB:
    //Slaver 对 Master 调用 SsmIPRActiveSession 函数的回复
    //Step 1、先检查 SsmIPRActiveSession 的调用在 Slaver 端是否成功
    if(pEvent->dwParam != 0)
    {
        //如果不成功,
        //do something
        //...
        break ;
    }
```

```
}

//Step 2、如果前面检查调用成功，则调用 SynIPRecorder 的 API SsmRecToFile
//通知 Slaver 开始录音
if(SsmRecToFile(pEvent->nReference, szRecFilePath, 6, 0, -1, -1, 0) != 0)
{
    SsmGetLastErrMsg(szErrMsg);
    nResult = -1;
}

//Step 3、调用 SynIPAnalyzer 的 API SsmIPRSendSession
bFind = FALSE;
for(i=0; i<MaxLine; i++)
{
    if(ChannelInfo[i].dwSessionId == ChannelInfo[pEvent->nReference].dwSessionId
        && ChannelInfo[i].nChType == IPRA_CH)
    {
        if(ChannelInfo[i].dwSessionId)
        {
            bFind = TRUE;
            break;
        }
    }
}
if(!bFind)
    break;
//将 Session 往指定的 IP 地址和端口发送
wsprintf(szIPP_Rec, "%d.%d.%d.%d", IPR_SlaverAddr[0].ipAddr.S_un_b.s_b1,
          IPR_SlaverAddr[0].ipAddr.S_un_b.s_b2,
          IPR_SlaverAddr[0].ipAddr.S_un_b.s_b3,
          IPR_SlaverAddr[0].ipAddr.S_un_b.s_b4);
wsprintf(szIPS_Rec, "%d.%d.%d.%d", IPR_SlaverAddr[0].ipAddr.S_un_b.s_b1,
          IPR_SlaverAddr[0].ipAddr.S_un_b.s_b2,
          IPR_SlaverAddr[0].ipAddr.S_un_b.s_b3,
          IPR_SlaverAddr[0].ipAddr.S_un_b.s_b4);
if(SsmIPRSendSession(i, szIPP, ChannelInfo[i].nForwardingPPort,
                     szIPS, ChannelInfo[i].nForwardingSPort) != 0)
{
    SsmGetLastErrMsg(szErrMsg);
    nResult = -1;
}
break;

case E_IPR_DEACTIVE_SESSION_CB:
    //Slaver 对 Master 调用 SsmIPRDeActiveSession 函数的回复
```

```
break;  
case E_IPR_START_REC_CB:  
    //Slaver 对 Master 调用 SsmRecToFile 或 SsmRecToMem 函数的回复  
    break;  
case E_IPR_STOP_REC_CB:  
    //Slaver 对 Master 调用 SsmStopRecToFile 或 SsmStopRecToMem 函数的回复  
    If(SsmIPRDeActiveSession(pEvent->nReference)  
    {  
        SsmGetLastErrMsg(szErrMsg);  
        nResult = -1;  
    }  
    break;  
case E_IPR_LINK_REC_SLAVER_CONNECTED:  
    //Master 检测到有 Slaver 连接上来  
    break;  
case E_IPR_LINK_REC_SLAVER_DISCONNECTED:  
    //Master 检测到有 Slaver 断开连接  
    break;  
default:  
    break;  
}  
  
//在图形界面上显示事件信息  
EventDisplay(szEvtName, pEvent);  
  
return nResult;  
}
```

1.24 HMP 系列产品（CTI 系列）

1.24.1 HMP 系列产品简介

HMP 系列产品是一款集 SIP、SIP Server、放音、录音、转码、传真、会议等功能的专用软件。用于构建媒体网关、媒体服务器、IVR、呼叫中心、会议等系统。HMP 系列产品采用分布式架构，最大支持 2400 路媒体资源和信令资源。HMP 是纯软件产品，便于获取、开发、部署、维护、扩展。HMP 由两大部分组成：HMP Client 和 HMP Server。

HMP Server 的处理能力包括：

- ◆ 录音、放音
- ◆ 会议
- ◆ 语音编解码
- ◆ 检测和发送信令形式、RFC2833 的 DTMF 信号
- ◆ 检测带内的 DTMF 信号

- ❖ SIP

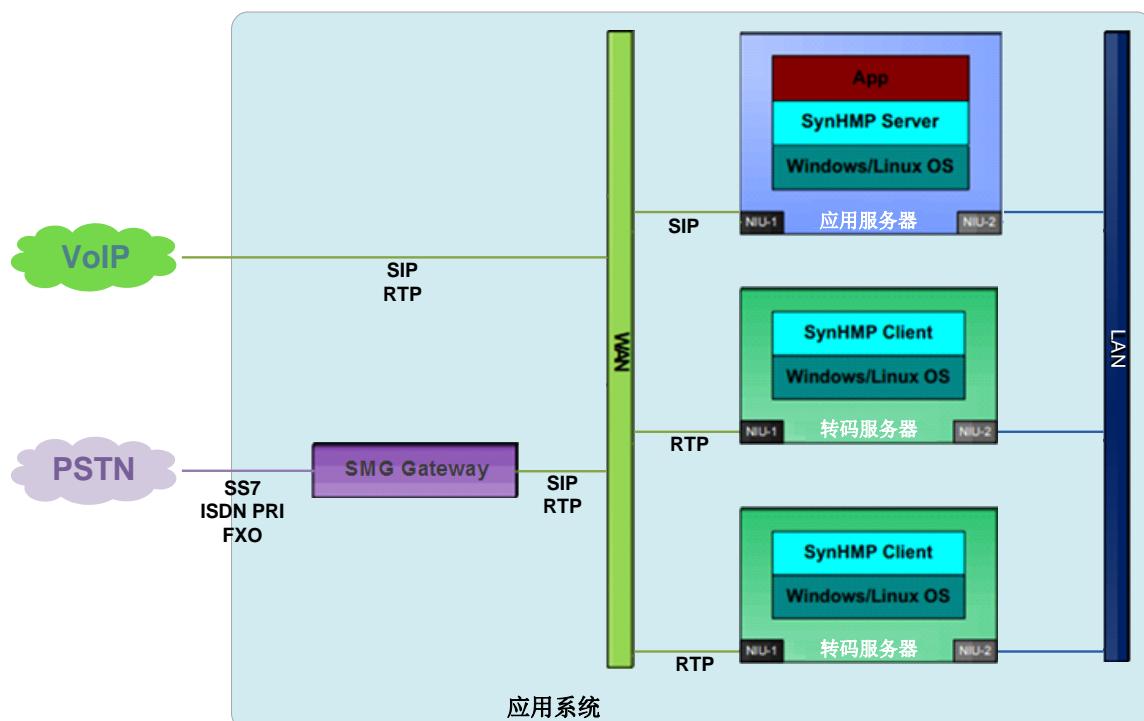
HMP Client 的处理能力包括:

- ❖ 模块分布式部署和扩容
- ❖ 支持多种编码格式的 RTP 数据的接收和发送

HMP 系列产品使用 USB KEY 进行授权如下表所示:

类型	安装	使用	授权升级文件
USB-KEY	-	-	独立的 exe 文件

1.24.2 HMP 系列产品的系统架构



➤ 应用服务器

- 双网口, 建议千兆
- 操作系统, 建议 Linux Centos 或 Windows Server
- SynHMP Server, 最多支持 2400 路, 建议 8 核 3.0G 及以上 CPU
- APP, 建议采用事件编程

➤ 转码服务器

- 双网口, 建议千兆
- 操作系统, 建议 Linux Centos
- SynHMP Client, 支持的通道数量根据编解码格式而定, 通道数量少时可在应用服务器上运行, 通道数量多时可在多台转码服务器上运行

1.24.3 HMP 的配置与使用

HMP 由两大部分组成, HMP Client 和 HMP Server。

HMP Client (由 HMPCodec 程序和 HMPLink 组成):

用于接收对端 RTP 数据，并对其编解码之后转发数据给 HMP Server。同时用于接收来自 HMP Server 的 RTP 数据并转码成需要的 RTP 格式后发送给对端。HMP Client 可以分布在不同的主机和不同的操作系统上。其服务程序为：HMPCodec，使用的配置文件为 HMPCodec.ini

HMP Server (由 HMPVirtualDevice、HMPRouter、HMPLink、ShpPci、SynSip、SHP_A3 和应用程序组成):

运行在一个主机上，控制一个或者多个 HMP Client。HMP Server 用于接收来自 HMP Client 的转发 RTP 数据，转化成语音数据并录制语音。转化放音数据为 RTP 数据并转发到 HMP Client 服务器。

主要功能如下：基本功能与 B 型 IP 卡类似，有 SIP、会议、卡内总线交换、录音、放音、DT 收发、注册等。接口与原有 IP 卡一致。

HMP 的检测和配置：检测机制与 IPR 类似，需要使用 usb key 进行加密，但 usb key 烧写的内容是与 IPR 区分开的。

1.24.4 HMP Client 专用配置项

配置节	配置项	
[HMPCodec]	<i>NICNum</i>	HMP Client 使用的网口数
	<i>LocalIP[n]</i>	HMP Client 所在主机 IP 地址
	<i>RemoteIP[n]</i>	HMP Server 所在主机 IP 地址
	<i>LocalPort[n]</i>	HMP Client 所在主机端口号
	<i>RemotePort[n]</i>	HMP Server 所在主机的端口号
	<i>MaxRtpThread</i>	HMP Client RTP 编解码线程数
	<i>LogType</i>	HMP Client 日志输出类型
	<i>LogLocation</i>	HMP Client 日志输出路径

1.24.5 HMP Server 专用配置项

配置节	配置项	
[HMPRouter]	<i>TotalMediaForward</i>	HMP Client 的数量
	<i>MaxMediaThread</i>	HMP Server 媒体处理线程数
	<i>LocalIP[n]</i>	HMP Server 所在主机 IP 地址
	<i>TotalCh[n]</i>	HMP Client 可以进行编解码的通道数量
	<i>LocalPort[n]</i>	HMP Server 的端口号
	<i>RemoteIP[n]</i>	HMP Client 所在主机 IP 地址
	<i>RemotePort[n]</i>	HMP Client 的端口号

1.24.6 HMP Server 与 HMP Client 配置使用范例

- 单机单网卡单客户端（单机单网卡最大支持 1200 路通道，通道数超过此数量可以使用单机多网卡或多机多网卡）：

HMP Server (主机 A 的 IP 地址：192.168.0.101)

配置文件：ShConfig.ini

[HMPRouter]

TotalMediaForward=1

MaxMediaThread=4

TotalCh[0]=600

LocalIP[0]=127.0.0.1
LocalPort[0]=5051
RemoteIP[0]=127.0.0.1
RemotePort[0]=5050
RtpIP[0]=192.168.0.101

HMP Client

[HMPCodec]
NICNum=1
LocalIP[0]=127.0.0.1
RemoteIP[0]=127.0.0.1
LocalPort[0]=5050
RemotePort[0]=5051
MaxRtpThread=4
LogType=2

2. 单机多网卡单客户端（客户端与服务端分布在同一台主机上，多网卡模式下每个网卡最大支持 800 路）：

HMP Server （主机 A 的 IP 地址：192.168.0.101、192.168.0.102）

配置文件：ShConfig.ini
[HMPRouter]
TotalMediaForward=2
MaxMediaThread=4
TotalCh[0]=600
LocalIP[0]=192.168.0.101
LocalPort[0]=5051
RemoteIP[0]=192.168.0.101
RemotePort[0]=5050
RtpIP[0]=192.168.0.101
TotalCh[1]=600
LocalIP[1]=192.168.0.102
LocalPort[1]=5053
RemoteIP[1]=192.168.0.102
RemotePort[1]=5052
RtpIP[1]=192.168.0.102

HMP Client

配置文件：HMPCodec.ini
[HMPCodec]
NICNum=2
LocalIP[0]=192.168.0.101
RemoteIP[0]=192.168.0.101
LocalPort[0]=5050
RemotePort[0]=5051
LocalIP[1]=192.168.0.102
RemoteIP[1]=192.168.0.102
LocalPort[1]=5052

RemotePort[1]=5053
MaxRtpThread=4
LogType=2

3. 多机器多网卡多客户端（客户端与服务端分布在不同的主机上，多网卡模式下每个网卡最大支持 800 路）：

HMP Server A（主机 A 的 IP 地址：192.168.0.101、192.168.0.102）

配置文件：ShConfig.ini

```
[HMPRouter]
TotalMediaForward=2
MaxMediaThread=4
TotalCh[0]=600
LocalIP[0]=192.168.0.101
LocalPort[0]=5051
RemoteIP[0]=192.168.0.201
RemotePort[0]=5050
RtpIP[0]= 192.168.0.201
TotalCh[1]=600
LocalIP[1]=192.168.0.102
LocalPort[1]=5053
RemoteIP[1]=192.168.0.202
RemotePort[1]=5052
RtpIP[1]=192.168.0.202
```

HMP Client（主机 B 的 IP 地址：192.168.0.201）

配置文件：HMPCodec.ini

```
[HMPCodec]
NICNum=1
LocalIP[0]=192.168.0.201
RemoteIP[0]=192.168.0.101
LocalPort[0]=5050
RemotePort[0]=5051
MaxRtpThread=4
LogType=2
```

HMP Client（主机 C 的 IP 地址：192.168.0.202）

配置文件：HMPCodec.ini

```
[HMPCodec]
NICNum=1
LocalIP[0]=192.168.0.202
RemoteIP[0]=192.168.0.102
LocalPort[0]=5052
RemotePort[0]=5053
MaxRtpThread=4
LogType=2
```

2 SynCTI 开发包函数说明

2.1 系统函数

2.1.1 驱动平台初始化函数

2.1.1.1 SsmStartCti

SsmStartCti 是整个 SynCTI 驱动程序开发包的入口函数，用来初始化 SynCTI 驱动平台和板卡。

函数原型：

```
int SsmStartCti(LPSTR lpSsmCfgFileName, LPSTR lpIndexCfgFileName)
```

参数说明：

lpSsmCfgFileName	SynCTI 驱动程序的配置文件名 (*.INI)。若为空字符串，在 Windows 操作系统下，驱动程序会自动使用系统目录下的“ShConfig.ini”配置文件；在 Linux 操作系统下，驱动程序会自动使用“/etc/shcti/”目录下的“ShConfig.ini”配置文件
lpIndexCfgFileName	预加载语音数据的配置文件 (*.INI)。若为空字符串，在“Windows”操作系统下，驱动程序会自动使用系统目录下的“ShIndex.ini”配置文件；在 Linux 操作系统下，驱动程序会自动使用“/etc/shcti/”下的“ShIndex.ini”配置文件

返回值：

-2	初始化失败，因为驱动程序已经装载
-1	初始化失败，失败原因可以通过 SsmGetLastErrMsg 函数获得
0	初始化成功

功能描述：

本函数为 SynCTI 驱动程序的入口。本函数将执行下述操作：

- 初始化板卡；
- 如果 lpIndexCfgFileName 不为空，根据配置文件中的信息将预加载语音数据从文件中加载到内存；
- 根据配置文件中语音卡的类型、数量、端口地址、中断号等基本信息，注册和初始化各语音卡，并分配系统资源；
- 为录、放音操作分配系统资源；
- 设置中断服务程序。

注意事项：

- 只有在本函数或函数 [SsmStartCtiEx](#) 调用成功后，才能调用本驱动程序中的其它任何函数 ([SsmGetLastErrMsg](#) 和 [SsmGetLastErrCode](#) 除外)；
- 应用程序在退出之前，必须调用函数 [SsmCloseCti](#) 关闭本驱动程序以释放驱动程序占用的资源；
- 有关配置文件的详细信息请参见第 4 章；
- 当配置了多块板卡时，如果只有部分板卡初始化成功，驱动程序也会返回 0。应用程序可以通过调用函数 [SsmGetMaxCfgBoard](#) 和 [SsmGetMaxUsableBoard](#) 来判别是否全部板卡初始化成功。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmStartCtiEx](#), [SsmCloseCti](#), [SsmGetMaxCfgBoard](#), [SsmGetMaxUsableBoard](#)

2.1.1.2 SsmStartCtiEx

SsmStartCtiEx 是整个 SynCTI 驱动程序开发包的入口函数，用来初始化 SynCTI 驱动平台和板卡，与 [SsmStartCti](#) 不同的是它可以设置驱动程序的事件输出模式。

函数原型:

```
int SsmStartCtiEx(LPSTR lpSsmCfgFileName, LPSTR lpIndexCfgFileName, BOOL bEnable,
PEVENT_SET_INFO pEventSet)
```

参数说明:

lpSsmCfgFileName	SynCTI 驱动程序的配置文件名 (*.INI)。若为空字符串，在 Windows 操作系统下，驱动程序会自动使用系统目录下的“ShConfig.ini”配置文件；在 Linux 操作系统下，驱动程序会自动使用“/etc/shcti/”目录下的“ShConfig.ini”配置文件																												
lpIndexCfgFileName	预加载语音数据的配置文件 (*.INI)。若为空字符串，在“Windows 操作系统下，驱动程序会自动使用系统目录下的“ShIndex.ini”配置文件；在 Linux 操作系统下，驱动程序会自动使用“/etc/shcti/”下的“ShIndex.ini”配置文件”																												
bEnable	事件输出的控制标志。bEnable=FALSE，表示默认不输出事件；bEnable=TRUE 表示默认输出事件由 DefaultEventOutput 指定。																												
pEventSet	<p>指向具有 EVENT_SET_INFO 结构的对象的指针，用于设置事件的输出模式。参数不能为 NULL，否则该函数与 SsmStartCti 功能一样，事件模式设置无效。EVENT_SET_INFO 结构的声明为：</p> <pre>typedef struct _EVENT_SET_INFO { DWORD dwWorkMode; LPVOID lpHandlerParam; DWORD dwOutCondition; DWORD dwOutParamVal; DWORD dwUser; } EVENT_SET_INFO,*PEVENT_SET_INFO;</pre> <p>各参数的含义如下表所示：</p> <table border="1"> <thead> <tr> <th colspan="3">参数名称: dwWorkMode</th> </tr> <tr> <th colspan="3">设置驱动程序抛出事件的模式。取值范围：</th> </tr> <tr> <th>取值</th> <th>event.h 中的宏</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>NO_EVENT</td> <td>驱动程序不输出任何事件。</td> </tr> <tr> <td>1</td> <td>EVENT_POLLING</td> <td>驱动程序工作于事件等待模式，参数 dwOutCondition 和 dwOutParamVal 有效，lpHandlerParam 和 dwUser 无效。</td> </tr> <tr> <td>2</td> <td>EVENT_CALLBACK</td> <td>驱动程序工作于事件回调模式，此时参数 lpHandlerParam 和 dwUser 有效。</td> </tr> <tr> <td>3</td> <td>EVENT_MESSAGE</td> <td>驱动程序工作于 Windows 消息模式，所有的消息都将被发送到 windows 消息队列中，此模式在 Linux 系统下设置无效。</td> </tr> <tr> <td>4</td> <td>EVENT_POLLINGA</td> <td>驱动程序工作于事件等待模式，参数 dwOutCondition 和 dwOutParamVal 有效，lpHandlerParam 和 dwUser 无效。</td> </tr> <tr> <td>5</td> <td>EVENT_CALLBACKA</td> <td>驱动程序工作于事件回调模式，此时参数 lpHandlerParam 和 dwUser 有效。</td> </tr> </tbody> </table> <p>参数名称: lpHandlerParam</p> <p>仅当 dwWorkMode=EVENT_CALLBACK 或 EVENT_CALLBACKA 时有效，dwWorkMode 为其它值时，本参数将被忽略。</p> <p>> dwWorkMode=EVENT_CALLBACK</p> <p>lpHandlerParam 为回调函数指针</p> <p>◆ lpHandlerParam 的格式为：</p> <pre>int CALLBACK CallBackFun(WORD wEvent,int nReference,DWORD dwParam,DWORD dwUser);</pre> <p>其中，参数 wEvent、nReference 和 dwParam 的含义与 MESSAGE_INFO 结构中声明的变量的含义完全相同，详细信息请参见函数 SsmWaitForEvent 或</p>		参数名称: dwWorkMode			设置驱动程序抛出事件的模式。取值范围：			取值	event.h 中的宏	描述	0	NO_EVENT	驱动程序不输出任何事件。	1	EVENT_POLLING	驱动程序工作于事件等待模式，参数 dwOutCondition 和 dwOutParamVal 有效，lpHandlerParam 和 dwUser 无效。	2	EVENT_CALLBACK	驱动程序工作于事件回调模式，此时参数 lpHandlerParam 和 dwUser 有效。	3	EVENT_MESSAGE	驱动程序工作于 Windows 消息模式，所有的消息都将被发送到 windows 消息队列中，此模式在 Linux 系统下设置无效。	4	EVENT_POLLINGA	驱动程序工作于事件等待模式，参数 dwOutCondition 和 dwOutParamVal 有效，lpHandlerParam 和 dwUser 无效。	5	EVENT_CALLBACKA	驱动程序工作于事件回调模式，此时参数 lpHandlerParam 和 dwUser 有效。
参数名称: dwWorkMode																													
设置驱动程序抛出事件的模式。取值范围：																													
取值	event.h 中的宏	描述																											
0	NO_EVENT	驱动程序不输出任何事件。																											
1	EVENT_POLLING	驱动程序工作于事件等待模式，参数 dwOutCondition 和 dwOutParamVal 有效，lpHandlerParam 和 dwUser 无效。																											
2	EVENT_CALLBACK	驱动程序工作于事件回调模式，此时参数 lpHandlerParam 和 dwUser 有效。																											
3	EVENT_MESSAGE	驱动程序工作于 Windows 消息模式，所有的消息都将被发送到 windows 消息队列中，此模式在 Linux 系统下设置无效。																											
4	EVENT_POLLINGA	驱动程序工作于事件等待模式，参数 dwOutCondition 和 dwOutParamVal 有效，lpHandlerParam 和 dwUser 无效。																											
5	EVENT_CALLBACKA	驱动程序工作于事件回调模式，此时参数 lpHandlerParam 和 dwUser 有效。																											

	<p>SsmGetEvent 的说明。参数 dwUser 即为调用本函数时应用程序传递给驱动程序的 dwUser 参数。</p> <p>➢ dwWorkMode=EVENT_CALLBACKA <i>lpHandlerParam</i> 为回调函数指针 ◆ <i>lpHandlerParam</i> 的格式为： int CALLBACK CallBackFunA(PSSM_EVENT pEvent); 其中， pEvent 是指向 SSM_EVENT 结构对象的指针，用于返回事件信息，详细说明请参见函数 SsmWaitForEventA 或 SsmGetEventA。 注意：当回调函数返回 -1 时，会立即禁止所有事件的抛出；如果需要重新抛出事件，请使用函数 SsmSetEvent 再次进行设置。</p> <p>➢ dwWorkMode=EVENT_MESSAGE <i>pHandlerParam</i> 为应用程序接收驱动事件的窗口句柄。</p>
	<p>参数名称： dwUser 用户自定义参数，仅当 dwWorkMode=EVENT_CALLBACK 或者 EVENT_CALLBACKA 时有效， dwWorkMode 为其它值时，本参数将被忽略。驱动程序会保存此参数，当驱动程序抛出事件时，会将此参数透明地传递给 <i>lpHandlerParam</i> 指定的回调函数中的同名参数。</p> <p>参数名称： dwOutCondition、dwOutParamVal dwOutCondition 设置事件的输出条件， dwOutParamVal 设置事件输出时携带的参数类别。当事件发生时，驱动程序会将 dwOutParamVal 设定的数据类别放在结构 MESSAGE_INFO 或 SSM_EVENT 的 dwParam 参数中。 dwOutCondition、 dwOutParamVal 在此函数中无需设置，即使设置也会被还原成默认值。</p>

返回值：

-2	初始化失败，因为驱动程序已经装载
-1	初始化失败，失败原因可以通过 SsmGetLastErrMsg 函数获得
0	初始化成功

功能描述：

本函数为 SynCTI 驱动程序的入口。本函数将执行下述操作：

- 初始化板卡；
- 设置驱动程序的事件输出模式；
- 如果 *lpIndexCfgFileName* 不为空，根据配置文件中的信息将预加载语音数据从文件中加载到内存；
- 根据配置文件中语音卡的类型、数量、端口地址、中断号等基本信息，注册和初始化各语音卡，并分配系统资源；
- 为录、放音操作分配系统资源；
- 设置中断服务程序。

注意事项：

- 只有在本函数或函数 [SsmStartCti](#) 调用成功后，才能调用本驱动程序中的其它任何函数([SsmGetLastErrMsg](#) 和 [SsmGetLastErrCode](#) 除外)；
- 本函数的最后一个参数 *pEventSet* 与 [SsmSetEvent](#) 函数的最后一个参数使用的是同一个结构体，但是用法略有差别，请留意区分。
- 应用程序在退出之前，必须调用函数 [SsmCloseCti](#) 关闭本驱动程序以释放驱动程序占用的资源；
- 有关配置文件的详细信息请参见第 4 章；
- 当配置了多块板卡时，如果只有部分板卡初始化成功，驱动程序也会返回 0。应用程序可以通过调用函数 [SsmGetMaxCfgBoard](#) 和 [SsmGetMaxUsableBoard](#) 来判别是否全部板卡初始化成功。

相关信息：

驱动程序版本要求	SynCTI Ver. 5.3.2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmStartCti](#), [SsmCloseCti](#), [SsmGetMaxCfgBoard](#), [SsmGetMaxUsableBoard](#), [SsmSetEvent](#)

2.1.1.3 SsmCloseCti

关闭 SynCTI 驱动程序。

函数原型:

```
int SsmCloseCti (void)
```

参数说明:

无

返回值:

1	关闭 SynCTI 驱动程序成功
---	------------------

功能描述:

关闭 SynCTI 驱动程序。

注意事项:

- 应用程序在退出系统之前，必须调用此函数。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmStartCti](#)

2.1.2 设置 API 和事件输出方式的函数

2.1.2.1 SsmSetLogEnable

设置是否输出指定类型的日志以及输出日志的方式。

函数原型:

```
int WINAPI SsmSetLogEnable(int nLogType,int nLogEnable,int nLogCreateMode)
```

参数说明:

nLogType	日志类型, 取值范围 0~8: =0: ShCtiApiLog =1: ShCtiSs1Log =2: ShCtiLsdnLog =3: ShCtiSpySs7Log =4: ShCtiSpySs1Log =5: ShCtiSpyLsdnLog =6: ShCtiSystemInfoLog =7: ShCtiDstLog =8: ShCtiFaxLog
	设置是否输出日志。 当 nLogType=0 时, nLogEnable 可取如下值: =0: 不输出日志文件 =1: 输出 API 函数调用信息到 DebugView =2: 输出 API 函数调用信息到 LOG 文件 =3: 输出事件信息到 LOG 文件 =4: 同时输出 API 函数调用和事件信息到 LOG 文件
	当 nLogType=1, 2, 3, 4, 5, 6, 7, 8 时, nLogEnable 可取如下值: =0: 不输出

	=1: 输出到 LOG 文件 =2: 输出到 DebugView
nLogCreateMode	日志的创建方式, 只有 nLogEnable 设置为输出日志文件时本参数才有效。 =0: 所有通道信息写在一个日志文件中 =1: 每个通道单独生成一个日志文件。当 nLogType=0, 1, 4, 7, 8 时, 设置此值才有效

返回值:

-1	操作失败
0	操作成功

功能描述:

设置是否输出指定类型的日志和输出日志的方式。不管输出哪一种日志, 驱动都会创建 ShCtiLog 文件夹, 并将生成的日志文件保存在这个文件夹下。

参数 nLogType 中描述的日志类型, 具体含义如下:

日志类型	含义
ShCtiApiLog	记录 API 函数调用和驱动抛出的事件的相关信息
ShCtiSs1Log	记录接收和发送的 ABCD 位码和 R2
ShCtiLsdnLog	记录 ISDN 协议数据链路层 1 帧消息
ShCtiSpySs7Log	记录驱动程序收到的 1 帧消息
ShCtiSpySs1Log	记录驱动程序收到的与一号信令相关的 CAS 和 R2
ShCtiSpyLsdnLog	记录驱动程序收到的 1 帧消息
ShCtiSystemInfoLog	记录系统信息, 如链路同步信息、本端阻抗、E1/T1、CRC-4 和其它板卡运行信息等
ShCtiDstLog	记录数字电话监控信令消息
ShCtiFaxLog	记录传真交互命令消息

下面的配置项可以实现相同功能:

[ApiLogEnable](#)
[ApiLogCreateMode](#)
[Ss1LogEnable](#)
[Ss1LogCreateMode](#)
[SpySs1LogEnable](#)
[SpySs1CreateMode](#)
[LsdnLogEnable](#)
[SpySs7LogEnable](#)
[SpyLsdnLogEnable](#)
[SystemInfoLogEnable](#)
[DSTLogEnable](#)
[DSTLogCreateMode](#)
[FaxLogEnable](#)
[FaxLogCreateMode](#)

注意事项: 无

相关信息:

驱动程序版本要求	SynCTI Ver.5.1.1.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetLogAttribute](#), [SsmGetLogAttribute](#), [SsmSetApiLogRange](#)

2.1.2.2 SsmSetLogAttribute

设置生成日志的周期、保存的最新的日志数目、自动停止输出日志的周期数以及日志输出路径。

函数原型:

```
Int WINAPI SsmSetLogAttribute(int nLogCreatePeriod,int nLogMaxKeep,int nLogMaxPeriod,LPSTR pLogFilePath)
```

参数说明:

nLogCreatePeriod	生成日志的周期，单位小时，取值范围: $nLogCreatePeriod > 0$
nLogMaxKeep	当生成的日志文件数目大于本参数设置的值时，最先生成的多出的日志将被自动删除。取值范围: $nLogMaxKeep > 0$ 。如果 $nLogMaxKeep = -1$ ，则表不删除，日志数目无限制。
nLogMaxPeriod	累计的日志创建周期达到本参数设置的次数后，将自动停止输出日志。取值范围: $nLogMaxPeriod > 0$ 。如果 $nLogMaxPeriod = -1$ ，表可不断创建日志，日志生成的数目无限制。
pLogFilePath	日志存放路径。注意：必须是已有路径，否则函数调用失败。如：“D:\”。如果设置为 NULL ，则表示是当前工作路径；如果只是写一个名字，则在当前工作路径下创建此文件夹。

返回值:

-1	操作失败
0	操作成功

功能描述:

设置生成日志的周期、保存的最新的日志数目、自动停止输出日志的周期数以及日志输出路径。

本函数设定的值对如下日志有效：

日志类型	含义
ShCtiApiLog	记录 API 函数调用和驱动抛出的事件的相关信息
ShCtiSs1Log	记录接收和发送的 ABCD 位码和 R2
ShCtiLsdnLog	记录 ISDN 协议数据链路层 1 帧消息
ShCtiSpySs7Log	记录驱动程序收到的 1 帧消息
ShCtiSpySs1Log	记录驱动程序收到的与一号信令相关的 CAS 和 R2
ShCtiSpyLsdnLog	记录驱动程序收到的 1 帧消息
ShCtiSystemInfoLog	记录系统信息，如链路同步信息、本端阻抗、E1/T1、CRC-4 和其它板卡运行信息等
ShCtiDstLog	记录数字电话监控信令消息

下面的配置项可以实现相同功能：

[LogCreatePeriod](#)
[LogMaxKeep](#)
[LogMaxPeriod](#)
[LogFilepath](#)

注意事项: 无

相关信息:

驱动程序版本要求	SynCTI Ver.5.1.1.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetLogEnable](#), [SsmGetLogAttribute](#), [SsmSetApiLogRange](#)

2.1.2.3 SsmSetApiLogRange

设置事件和 API 输出写日志的通道范围，以及写入日志的事件编号范围。

函数原型:

```
int WINAPI SsmSetApiLogRange(int nChStart, int nChEnd, int nEventStart, int nEventEnd)
```

参数说明:

nChStart	表示事件和 API 函数输出写日志的起始通道号。当 nChStart=-1 时，表示输出通道范围为 <nChEnd
nChEnd	表示事件和 API 函数输出写日志的结束通道号，当 nChEnd=-1，表示输出通道范围为 >nChStart
nEventStart	表示写入日志的起始事件号，当 nEventStart =-1 时，表示输出 <nEventEnd 的事件
nEventEnd	表示写入日志的结束事件号，当 nEventEnd =-1 时，表示输出 >nEventStart 的事件

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

在 [ApiLogEnable](#) 设置为输出 Log 的条件下，设置事件和 API 输出写日志的通道范围，以及写入日志的事件编号范围。

下面的配置项可以实现相同功能：

[ApiLogSetEventRange](#)

[ApiLogSetChRange](#)

注意事项: 无**相关信息:**

驱动程序版本要求	SynCTI Ver.5.1.1.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetLogEnable](#), [SsmGetLogAttribute](#), [SsmSetLogAttribute](#)

2.1.2.4 SsmGetLogAttribute

获取生成日志的周期、保存的最新的日志数目、自动停止输出日志的周期数以及日志输出路径。

函数原型:

```
int WINAPI SsmGetLogAttribute(PINT pLogCreatePeriod,PINT pLogMaxKeep,PINT pLogMaxPeriod,LPSTR pLogFilePath)
```

参数说明:

pLogCreatePeriod	生成日志的周期，单位小时，取值范围: pLogCreatePeriod>0
pLogMaxKeep	当生成的日志文件数目大于本参数设置的值时，最先生成的多出的日志将被自动删除。取值范围: pLogMaxKeep >0。如果 pLogMaxKeep =-1，则表不删除，日志数目无限制。
pLogMaxPeriod	累计的日志创建周期达到本参数设置的次数后，将自动停止输出日志。取值范围: pLogMaxPeriod >0。如果 pLogMaxPeriod = -1，表可不断创建日志，日志生成的数目无限制。
pLogFilePath	日志存放路径。,注意: 必须是已有路径，否则函数调用失败。如：“D:\”。如果设置为 NULL，则表示是当前工作路径；如果只是写一个名字，则在当前工作路径下创建此文件夹。

返回值:

-1	操作失败
0	操作成功

功能描述:

获取生成日志的周期、保存的最新的日志数目、自动停止输出日志的周期数以及日志输出路径。

相关信息:

驱动程序版本要求	SynCTI Ver.5.1.1.0 或更高
----------	------------------------

Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetLogEnable](#), [SsmSetLogAttribute](#), [SsmSetApiLogRange](#)

2.1.3 有关定时器的函数

2.1.3.1 SsmStartTimer

向驱动程序申请一个系统定时器。

函数原型:

```
int SsmStartTimer(WORD wDelay, WORD wMode)
```

参数说明:

wDelay	定时器的初始值，单位为毫秒，取值范围: 0x0008~0xffff 设置定时器的工作模式。
wMode	=0(TIMER_ONE): 单次模式，定时器一旦溢出，就自动停止； =1(TIMER_PERIODIC): 周期性模式，每次发生溢出后，由驱动程序自动重新装载初始值 wDelay，直到应用程序调用 SsmStopTimer 函数

返回值:

-1	操作失败
≥0	申请成功，返回定时器的 ID

功能描述:

向驱动程序申请一个系统定时器，并设置该定时器的工作参数。每当定时器发生溢出时，驱动程序会向应用程序抛出 [E_SYS_TIMEOUT](#) 事件，更多内容请参见第 1 章中“[使用驱动程序提供的定时器](#)”部分内容。

注意事项:

- 本函数必须在函数 [SsmSetEvent](#) 之后调用，否则驱动不会输出 TIMEOUT 事件。

相关信息:

驱动程序版本要求	SynCTI Ver.4.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmStopTimer](#), [StartTimer](#)

相关事件: [E_SYS_TIMEOUT](#)

2.1.3.2 SsmStopTimer

停止一个定时器。

函数原型:

```
int SsmStopTimer(int nTimer)
```

参数说明:

nTimer	定时器的 ID，即调用函数 SsmStartTimer 时的返回值
--------	---------------------------------------------------

返回值:

-1	失败
0	停止成功

功能描述:

停止一个定时器，该定时器由函数 [SsmStartTimer](#) 申请。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver. 4.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmStartTimer](#)

2.1.3.3 StartTimer

启动通道定时器。

函数原型:

```
int StartTimer(int ch, WORD wClockID)
```

参数说明:

ch	通道号
wClockID	定时器 ID, 取值范围: 0~9

返回值:

-1	失败
0	成功

功能描述:

启动通道定时器。

通道定时器启动后, 驱动程序只是将当前时刻保存到变量中, 并不会在驱动程序内部判断定时器是否溢出。只有当应用程序调用 [ElapseTime](#) 函数时, 驱动程序才会返回定时器的保持时间。

注意事项:

- 本函数启动的定时器不输出任何事件。

相关信息:

驱动程序版本要求	SynCTI Ver. 4.7.1.5 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [ElapseTime](#)

2.1.3.4 ElapseTime

获取通道定时器自启动以来的保持时间。

函数原型:

```
DWORD ElapseTime(int ch, WORD ClockType)
```

参数说明:

ch	通道号
ClockType	定时器 ID, 取值范围: 0~9

返回值:

保持时间, 单位为 8 毫秒的倍数。

功能描述:

获取通道定时器自启动以来的保持时间。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 4.7.1.5 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [StartTimer](#)

2.1.4 获取系统错误信息的函数

2.1.4.1 SsmGetLastErrCode

获取应用程序调用 API 函数失败时的错误代码。

函数原型:

```
int SsmGetLastErrCode()
```

参数说明:

无

返回值:

返回值如下表所示:

返回值	shpa3api.h 中的宏	描述
0	C_ERROR_INIT_FAILURE	初始化失败
1	C_ERROR_SSMAPI_UNOPENED	驱动程序的 API 接口尚未向应用程序开放
2	C_ERROR_INVALID_APPCH	非法的通道编号
3	C_ERROR_UNSUPPORTED_OP	不支持的操作
4	C_ERROR_INDEX_UNOPENED	内存放音功能尚未向应用程序开放
5	C_ERROR_INVALID_BUSCH	非法的交换总线通道的逻辑编号
6	C_ERROR_OP_UNOPENED	指定操作没有向应用程序开放
7	C_ERROR_INVALID_FORMAT	非法语音编码格式
8	C_ERROR_INVALID_PARAMETER	非法调用参数
9	C_ERROR_FILEOP_FAILURE	文件操作失败
10	C_ERROR_MEMORY_FAILURE	内存操作失败
11	C_ERROR_RESOURCE_USEUP	相关资源耗尽
12	C_ERROR_SYSTEM	系统错误
13	C_ERROR_IdleChNotFound	没有空闲通道
14	C_ERROR_OP_FAILURE	操作失败
15	C_ERROR_INVALID_APPSPYCIC	非法的监控电路编号
16	C_ERROR_FAX_NOFILE	传真文件错误
17	C_ERROR_VCH_INVALID_SCALE	参数越界

功能描述:

获取应用程序调用 API 函数失败时的错误代码。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver.3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetLastErrMsg](#)

2.1.4.2 SsmGetLastErrMsg

请参见 [SsmGetLastErrMsgA](#)

2.1.4.3 SsmGetLastErrMsgA

获取最新一次调用 API 函数失败时的错误信息。

函数原型:

```
void SsmGetLastErrMsg(LPSTR szErrMsgBuf)  
char* SsmGetLastErrMsgA(void)
```

参数说明:

szErrMsgBuf	存放错误信息字符串的首地址指针，由应用程序分配存储空间，不能少于 300 字节
-------------	-----------------------------------------

返回值:

void	无
char*	驱动程序内部保存错误信息字符串的首地址指针

功能描述:

获取最新一次调用 API 函数失败时的错误信息。

注意事项:

- 本函数应该在函数调用失败时立即调用。

相关信息:

驱动程序版本要求	SynCTI Ver.2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetLastErrCode](#)

2.1.4.4 fBmp_GetErrMsg

获取最新一次调用 Bmp 函数失败的错误信息。

函数原型:

```
void fBmp_GetErrMsg(char *buf)
```

参数说明:

buf	存放错误信息字符串的缓冲区指针
-----	-----------------

返回值: 无。

功能描述:

当调用 Bmp 函数返回失败时，驱动程序将会给出详细的错误信息和错误号，调用该函数可获取该信息。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	BmpApi.h
Library	BmpUtil.lib
DLL	BmpUtil.dll

相关函数:

2.1.5 获取板卡信息的函数

2.1.5.1 获取安装板卡的信息

2.1.5.1.1 SsmGetMaxCfgBoard

获取配置文件中设置的 Synway 板卡的总数。

函数原型:

```
int SsmGetMaxCfgBoard()
```

参数说明: 无

返回值:

-1	调用失败
≥ 0	返回板卡总数

功能描述:

获取配置文件中设置的 Synway 板卡的总数。

注意事项:

无。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetMaxUsableBoard](#)

2.1.5.1.2 SsmGetMaxUsableBoard

获取驱动程序初始化成功的板卡总数。

函数原型:

```
int SsmGetMaxUsableBoard()
```

参数说明: 无

返回值:

-1	调用失败
≥ 0	最大的可用的卡数

功能描述:

获取驱动程序初始化成功的板卡总数。

注意事项: 无。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetMaxCfgBoard](#)

2.1.5.1.3 GetTotalPciBoard

取得系统中已安装的板卡的总数。

函数原型:

```
int GetTotalPciBoard(int *pTotalBoards)
```

参数说明:

pTotalBoards	存放总卡数的整型指针，空间由应用程序分配
--------------	----------------------

返回值:

-1	调用失败
0	调用成功

功能描述:

本函数取得系统中已安装的板卡的个数。如果调用成功则将板卡个数放到 pTotalBoards 中。

注意事项:

调用本函数无需经过 Shp_a3.dll 来控制，可以直接取得操作系统中已经安装的板卡个数。在启动 Shp_a3.dll 之前调用，这样就可以由上层程序修改 ShConfig.ini 配置文件，确保在调用 [SsmStartCti](#) 函数时肯定能启动成功。

如果用户加载 Shinitpci.dll 之后又卸载或安装了某块板卡，需要先调用 [ReloadPciBoardInfo](#)，然后再调用本函数以获得最新信息。

相关信息:

驱动程序版本要求	和 SynCTI 驱动版本无关
Header	Shinitpci.h
Library	Shinitpci.lib
DLL	Shinitpci.dll

相关函数: [GetPciBoardSerialNo](#), [GetPciBoardModel](#), [ReloadPciBoardInfo](#)

2.1.5.1.4 ReloadPciBoardInfo

重新读取板卡信息。

函数原型:

```
int ReloadPciBoardInfo()
```

参数说明: 无

返回值:

-1	调用失败
0	调用成功

功能描述:

重新读取板卡信息。一般在卸载或新安装了某板卡后才需要调用。

[GetPciBoardSerialNo](#), [GetPciBoardModel](#), [GetTotalPciBoard](#) 的信息在 Shinitpci.dll 加载时已经被初始化了。

如果用户加载 Shinitpci.dll 之后又卸载或安装了某块板卡，可以在调用本函数后再调用 Shinitpci.dll 中的其它函数。

注意事项:

调用本函数无需经过 Shp_a3.dll 来控制。

相关信息:

驱动程序版本要求	和 SynCTI 驱动版本无关
Header	Shinitpci.h
Library	Shinitpci.lib
DLL	Shinitpci.dll

相关函数: [GetPciBoardSerialNo](#), [GetPciBoardModel](#), [GetTotalPciBoard](#)

2.1.5.2 获取板卡授权号

2.1.5.2.1 SsmGetAccreditId

请参见 [SsmGetAccreditIdEx](#)

2.1.5.2.2 SsmGetAccreditIdEx

获取保存在板卡固件中的客户授权号。SsmGetAccreditId 的返回值中包含了板卡型号和客户授权号信息，SsmGetAccreditIdEx 只返回客户授权号。

函数原型:

```
int SsmGetAccreditId(int nBld)  
int SsmGetAccreditIdEx(int nBld)
```

参数说明:

nBld	配置文件中指定的板卡 ID 编号
------	------------------

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得	
0	板卡上没有客户授权号的信息	
>0	SsmGetAccreditId	返回值的低 18 比特有效，记为 Bit ₁₇ ...Bit ₀ : Bit ₁₇ ~Bit ₁₀ : 板卡型号 Bit ₉ ~Bit ₀ : 客户授权号
	SsmGetAccreditIdEx	返回客户授权号

功能描述:

获取保存在板卡固件中的客户授权号。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver.3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetBoardModel](#)

2.1.5.3 获取板卡型号

2.1.5.3.1 SsmGetBoardModel

获取板卡的型号。

函数原型:

```
int SsmGetBoardModel(int nBld)
```

参数说明:

nBld	配置文件中指定的板卡 ID 编号
------	------------------

返回值:

返回值为-1 表示调用失败，失败原因可以通过函数 [SsmGetLastErrMsg](#) 获取；返回值为其它值时，其对应的板卡型号如下表所示：

返回值	对应的板卡型号
0x07	SHD-60A-CT/PCI/SS1
0x09	SHT-16A-CT/PCI
0xa	SHT-8A/PCI
0xb	SHD-60A-CT/PCI/SS7
0xc	SHD-60A-CT/PCI/ISDN
0xd	SHD-120A-CT/PCI/SS1
0xe	SHD-120A-CT/PCI/SS7
0xf	SHD-120A-CT/PCI/ISDN
0x14	SHT-16B-CT/cPCI
0x15	SHD-60A-CT/PCI/FJ
0x16	SHD-30A-CT/cPCI
0x17	SHD-60A-CT/cPCI
0x18	SHD-120A-CT/cPCI
0x1d	SHD-30A-CT/PCI/SS1
0x1e	SHD-30A-CT/PCI/SS7
0x1f	SHD-30A-CT/PCI/ISDN
0x20	SHD-30A-CT/PCI/FJ
0x24	SHT-16B-CT/PCI
0x25	SHT-16B-CT/PCI/FAX 或 SHT-16B-CT/PCI/MP3 [1]
0x26	SHT-8B/PCI
0x27	SHT-8B/PCI/FAX
0x28	SHD-30B-CT/PCI/FAX
0x29	SHD-60B-CT/PCI/FAX
0x2c	SHD-240A-CT/cPCI
0x2d	SHD-480A-CT/cPCI
0x2f	SHD-240S-CT/cPCI
0x30	SHD-480S-CT/cPCI
0x31	SHT-16B-CT/cPCI/MP3
0x32	SHT-16B-CT/cPCI/FAX 或 SHT-16B-CT/cPCI/MP3 [1]
0x36	SHD-60B-CT/cPCI/FAX
0x3b	SHR-16DA-CT/PCI
0x43	DST-1600
0x46	SHT-2A/USB
0x47	SHT-4A/USB
0x4b	SHD-30C-CT/PCI
0x4c	SHD-30C-CT/PCI/FAX
0x4d	SHD-60C-CT/PCI
0x4e	SHD-60C-CT/PCI/FAX
0x55	SHV-120A-CT/PCI
0x56	SHV-240A-CT/PCI
0x57	SHD-240D-CT/PCI
0x58	SHD-240D-CT/PCI/EC
0x59	SHD-120D-CT/PCI
0x5a	SHD-120D-CT/PCI/EC
0x5b	SHT-8C/PCI/FAX

0x5c	SHT-16C-CT/PCI/FAX
0x5d	SHN-32A-CT/PCI
0x5f	SHT-2B/USB
0x60	SHT-4B/USB
0x65	SHV-240A-CT/cPCI
0x66	SHD-30B-CT/PCI/FJ
0x67	SHD-60B-CT/PCI/FJ
0x68	ATP-24A/PCI
0x69	DST-24B/PCI
0x69	DST-24B/PCI(SSW)
0x6a	SHT-16C-CT/PCI/EC
0x6b	SHT-8C/PCI/EC
0x6c	ATP-24A/PCI+
0x6d	SHN-120B-CT/PCI+
0x6e	DST-24B/PCI+
0x6e	DST-24B/PCI+(SSW)
0x6f	DTP-30C/PCIe
0x70	DTP-30C/PCIe+
0x71	DTP-60C/PCIe
0x72	DTP-60C/PCIe+
0x73	DTP-120C/PCIe
0x74	DTP-120C/PCIe+
0x75	SHD-30E-CT/PCIe
0x76	SHD-30E-CT/PCIe/EC
0x77	SHD-30E-CT/PCIe/FAX
0x78	SHD-60E-CT/PCIe
0x79	SHD-60E-CT/PCIe/EC
0x7a	SHD-60E-CT/PCIe/FAX
0x7b	SHD-120E-CT/PCIe
0x7c	SHD-120E-CT/PCIe/EC
0x7d	SHD-120E-CT/PCIe/FAX
0x7e	SHD-240E-CT/PCIe
0x7f	SHD-240E-CT/PCIe/EC
0x80	SHD-240E-CT/PCIe/FAX
0x83	SHN-60B-CT/PCI+
0x84	ATP-24A/PCIe
0x85	ATP-24A/PCIe+
0x86	SHD-120D-CT/PCI/CAS
0x87	SHD-240D-CT/PCI/CAS
0x88	SHN-32B-CT/PCI+
0x89	SHN-16B-CT/PCI+
0x8a	SHN-8B-CT/PCI+
0x8b	DST-24B/PCIe
0x8c	DST-24B/PCIe+
0x8e	SHD-120E-CT/PCI(SSW)
0x8f	SHD-120E-CT/PCI/EC(SSW)
0x90	SHD-120E-CT/PCI/FAX(SSW)
0x91	SHD-240E-CT/PCI(SSW)
0x92	SHD-240E-CT/PCI/EC(SSW)
0x93	SHD-240E-CT/PCI/FAX(SSW)
0x94	SHD-30E-CT/PCI(SSW)
0x95	SHD-30E-CT/PCI/EC(SSW)
0x96	SHD-30E-CT/PCI/FAX(SSW)
0x97	SHD-60E-CT/PCI(SSW)
0x98	SHD-60E-CT/PCI/EC(SSW)

0x99	SHD-60E-CT/PCI/FAX(SSW)
0x9b	SHF-2D/PCI
0x9c	SHF-4D/PCI
0xB3	SHF-4D/PCIe
0x9d	DTP-30C/PCI
0x9e	DTP-30C/PCI+
0x9f	DTP-60C/PCI
0xa0	DTP-60C/PCI+
0xa1	DTP-120C/PCI
0xa2	DTP-120C/PCI+
0xa4	SHT-16D-CT/PCIe
0xa6	PCM1280E
0xa7	SHD-240E-CT/PCIe/VAR
0xa9	SHN-60B-CT/PCIe+
0xaa	SHN-120B-CT/PCIe+
0xad	SHN-120B-CT/PCIe/VAR
0xae	SHN-480C-CT/PCIe
0xfd	SynIPRecorder
0xfe	SynIPAnalyzer

注[1]: 型号为 SHT-16B-CT/PCI/FAX 和 SHT-16B-CT/PCI/MP3 的板卡具有不同的硬件结构，但由于历史的原因，这 2 个型号的板卡上固件中写入的板卡 ID 号相同，使用配置项 [DSP3WORKMODE](#) 进行区分。

功能描述:

获取板卡的型号。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver.3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetAccreditIdEx](#), [SsmGetAccreditId](#)

2.1.5.3.2 GetPciBoardModel

获取板卡的型号。

函数原型:

```
int GetPciBoardModel(int nBId, LPSTR lpBoardModel)
```

参数说明:

nBId	实际安装板卡的 ID 编号
lpBoardModel	指定序号的板卡型号，用字符串表示。如下表：

板卡类型 & 相对应字符

返回值	含义
“SHD-60A-CT/PCI/SS1”	表示 PCI 60 路 1 号信令卡
“SHD-60A-CT/PCI/SS7”	表示 PCI 60 路 7 号信令卡
“SHD-60A-CT/PCI/ISDN”	表示 PCI 60 路 ISDN 信令卡
“SHD-120A-CT/PCI/SS1”	表示 PCI 120 路 1 号信令卡
“SHD-120A-CT/PCI/SS7”	表示 PCI 120 路 7 号信令卡
“SHD-120A-CT/PCI/ISDN”	表示 PCI 120 路 ISDN 信令卡
“SHD-60B-CT/PCI/FAX”	表示 PCI 60 路 B 型卡，该卡可配置成带传真通道的信令卡

“SHD-30B-CT/PCI/FAX”	表示 PCI 30 路 B 型卡, 该卡可配置成带传真通道的信令卡
“SHD-30A-CT/PCI/SS1”	表示 PCI 30 路 1 号 信令卡
“SHD-30A-CT/PCI/SS7”	表示 PCI 30 路 7 号信令卡
“SHD-30A-CT/PCI/ISDN”	表示 PCI 30 路 ISDN 信令卡
“SHD-30A-CT/PCI/FJ”	表示 PCI 30 路 A 型高阻复接卡
“SHD-60A-CT/PCI/FJ”	表示 PCI 60 路 A 型高阻复接卡
“SHD-30B-CT/PCI/FJ”	表示 PCI 30 路 B 型高阻复接卡
“SHD-60B-CT/PCI/FJ”	表示 PCI 60 路 B 型高阻复接卡
“SHT-16B-CT/PCI/FAX” 或 SHT-16B-CT/PCI/MP3 ^[1]	表示 PCI 16 路 B 型 MP3/FAX 录音卡
“SHT-8B/PCI”	表示 PCI 8 路 B 型模块卡
“SHT-8B/PCI/FAX”	表示 PCI 8 路 B 型带软传真模块卡
“SHT-8C/PCI/FAX”	表示 PCI 8 路 C 型带软传真模块卡
“SHT-8C/PCI/EC”	表示 PCI 8 路 C 型卡
“SHT-16C-CT/PCI/EC”	表示 PCI 16 路 C 型卡
“SHT-16B-CT/PCI”	表示 PCI 16 路 B 型卡
“SHT-16C-CT/PCI/FAX”	表示 PCI 16 路 C 型带软传真模块卡
“SHT-16A-CT/PCI/FAX”	表示 PCI 16 路 A 型带软传真模块卡
“DST-1600”	表示 PCI 16 路 B 型卡
“SHT-16B-CT/cPCI”	表示 cPCI 16 路 B 型模块卡
“SHD-30A-CT/cPCI”	表示 cPCI 30 路卡 (支持 SS1、ISDN、SS7)
“SHD-60A-CT/cPCI”	表示 cPCI 60 路卡 (支持 SS1、ISDN、SS7)
“SHT-16B-CT/cPCI/MP3” 或 SHT-16B-CT/cPCI/FAX”	表示 cPCI 16 路 B 型 MP3/FAX 录音卡
“SHD-480A-CT/cPCI”	表示 cPCI 带会议功能的 16E1 卡
“SHD-480S-CT/cPCI”	表示 cPCI 不带会议功能的 16E1 卡
“SHD-60B-CT/cPCI/FAX”	表示 cPCI 60 路 B 型卡, 该卡可配置成带传真通道的信令卡
“SHR-16DA-CT/PCI”	表示 PCI 16 路数字电话录音卡
“SHR-24DA-CT/PCI”	表示 PCI 24 路数字电话录音卡
“SHT-120A-CT/PCI”	表示 PCI 120 路大容量模拟卡
“SHT-2B/USB”	表示 2 路 B 型 USB 模块卡
“SHT-4B/USB”	表示 4 路 B 型 USB 模块卡
“SHT-2A/USB”	表示 2 路 USB 模块卡
“SHT-4A/USB”	表示 4 路 USB 模块卡
“SHT-120A-CT/cPCI”	表示 cPCI 120 路大容量模拟卡
“SHD-240D-CT/PCI”	表示 PCI 8 中继数字卡
“SHD-240D-CT/PCI/EC”	表示 PCI 带增强回波消除 8 中继数字卡
“SHD-120D-CT/PCI”	表示 PCI 4 中继数字卡
“SHD-120D-CT/PCI/EC”	表示 PCI 带增强回波消除 4 中继数字卡
“SHD-120D-CT/PCI/CAS”	表示 PCI 支持 1 号信令的 4 中继数字卡
“SHD-240D-CT/PCI/CAS”	表示 PCI 支持 1 号信令的 8 中继数字卡
“SHD-30C-CT/PCI”	表示 PCI 30 路 C 型卡
“SHD-30C-CT/PCI/FAX”	表示 PCI 30 路带传真资源的 C 型卡
“SHD-60C-CT/PCI”	表示 PCI 60 路 C 型卡
“SHD-60C-CT/PCI/FAX”	表示 PCI 60 路带传真资源的 C 型卡
“SHF-2D/PCI”	表示 PCI 2 路 D 型模拟传真卡
“SHF-4D/PCI”	表示 PCI 4 路 D 型模拟传真卡
“SHF-4D/PCIe”	表示 PCI e4 路 D 型模拟传真卡
“DTP-30C/PCI”	表示 PCI 30 路高阻复接卡
“DTP-30C/PCI+”	表示 PCI 30 路高阻复接卡
“DTP-60C/PCI”	表示 PCI 60 路高阻复接卡
“DTP-60C/PCI+”	表示 PCI 60 路高阻复接卡
“DTP-120C/PCI”	表示 PCI 120 路高阻复接卡
“DTP-120C/PCI+”	表示 PCI 120 路高阻复接卡
“DTP-30C/PCIe”	表示 PCIe 30 路高阻复接卡

“DTP-30C/PCIe+”	表示 PCIe 30 路高阻复接卡
“DTP-60C/PCIe”	表示 PCIe 60 路高阻复接卡
“DTP-60C/PCIe+”	表示 PCIe 60 路高阻复接卡
“DTP-120C/PCIe”	表示 PCIe 120 路高阻复接卡
“DTP-120C/PCIe+”	表示 PCIe 120 路高阻复接卡
“SHN-32A-CT/PCI”	表示 32 路 IP 卡
“SHN-8B-CT/PCI+”	表示 8 路 B 型 IP 卡
“SHN-16B-CT/PCI+”	表示 16 路 B 型 IP 卡
“SHN-32B-CT/PCI+”	表示 32 路 B 型 IP 卡
“SHN-60B-CT/PCI+”	表示 60 路 B 型 IP 卡
“SHN-120B-CT/PCI+”	表示 120 路 B 型 IP 卡
“SHN-60B-CT/PCIe+”	表示 PCIe 60 路 B 型 IP 卡
“SHN-120B-CT/PCIe+”	表示 PCIe 120 路 B 型 IP 卡
“SHN-120B-CT/PCIe/VAR”	表示 PCIe 带 60 路变声资源的 120 路 B 型 IP 卡
“SHN-480C-CT/PCIe”	表示 PCIe 480 路 C 型 IP 卡
“ATP-24A/PCI”	表示 ATP-24A/PCI 模拟电话录音卡
“ATP-24A/PCI+”	表示 ATP-24A/PCI+模拟电话录音卡
“ATP-24A/PCIe”	表示 ATP-24A/PCIe 模拟电话录音卡
“ATP-24A/PCIe+”	表示 ATP-24A/PCIe+模拟电话录音卡
“DST-24B/PCI”	表示 PCI 24 路 B 型数字电话录音卡
“DST-24B/PCI(SSW)”	表示 PCI 24 路 B 型数字电话录音卡
“DST-24B/PCI+”	表示 PCI 增强型 24 路 B 型数字电话录音卡
“DST-24B/PCI+(SSW)”	表示 PCI 增强型 24 路 B 型数字电话录音卡
“DST-24B/PCIe”	表示 PCIe 24 路 B 型数字电话录音卡
“DST-24B/PCIe+”	表示 PCIe 增强型 24 路 B 型数字电话录音卡
“SHD-30E-CT/PCI(SSW)”	表示 PCI 1 中继数字卡
“SHD-30E-CT/PCI/FAX(SSW)”	表示 PCI 带传真资源 1 中继数字卡
“SHD-30E-CT/PCI/EC(SSW)”	表示 PCI 带增强回波消除 1 中继数字卡
“SHD-60E-CT/PCI(SSW)”	表示 PCI 2 中继数字卡
“SHD-60E-CT/PCI/FAX(SSW)”	表示 PCI 带传真资源 2 中继数字卡
“SHD-60E-CT/PCI/EC(SSW)”	表示 PCI 带增强回波消除 2 中继数字卡
“SHD-120E-CT/PCI(SSW)”	表示 PCI 4 中继数字卡
“SHD-120E-CT/PCI/FAX(SSW)”	表示 PCI 带传真资源 4 中继数字卡
“SHD-120E-CT/PCI/EC(SSW)”	表示 PCI 带增强回波消除 4 中继数字卡
“SHD-240E-CT/PCI(SSW)”	表示 PCI 8 中继数字卡
“SHD-240E-CT/PCI/FAX(SSW)”	表示 PCI 带传真资源 8 中继数字卡
“SHD-240E-CT/PCI/EC(SSW)”	表示 PCI 带增强回波消除 8 中继数字卡
“SHD-30E-CT/PCIe”	表示 PCIe 1 中继数字卡
“SHD-30E-CT/PCIe/FAX”	表示 PCIe 带传真资源 1 中继数字卡
“SHD-30E-CT/PCIe/EC”	表示 PCIe 带增强回波消除 1 中继数字卡
“SHD-60E-CT/PCIe”	表示 PCIe 2 中继数字卡
“SHD-60E-CT/PCIe/FAX”	表示 PCIe 带传真资源 2 中继数字卡
“SHD-60E-CT/PCIe/EC”	表示 PCIe 带增强回波消除 2 中继数字卡
“SHD-120E-CT/PCIe”	表示 PCIe 4 中继数字卡
“SHD-120E-CT/PCIe/FAX”	表示 PCIe 带传真资源 4 中继数字卡
“SHD-120E-CT/PCIe/EC”	表示 PCIe 带增强回波消除 4 中继数字卡
“SHD-240E-CT/PCIe”	表示 PCIe 8 中继数字卡
“SHD-240E-CT/PCIe/FAX”	表示 PCIe 带传真资源 8 中继数字卡
“SHD-240E-CT/PCIe/EC”	表示 PCIe 带增强回波消除 8 中继数字卡
“SHD-240E-CT/PCIe/VAR”	表示 PCIe 带变声资源 8 中继数字卡
“SHT-16D-CT/PCIe”	表示 PCIe 16 路 D 型模块卡
“PCM1280E”	表示 PCIe 128 路 PCM32 录音卡

返回值：

0	调用成功
---	------

-1	调用失败
----	------

功能描述:

本函数取得指定编号 ID 板卡的板卡型号。如果调用成功则将板卡型号放到 lpBoardModel 中。

注意事项:

本函数调用无需经过 Shp_a3.dll 来控制，可以直接取得操作系统中已经安装的板卡型号。

如果用户加载 Shinitpci.dll 之后又卸载或安装了某块板卡，需要先调用 [ReloadPciBoardInfo](#)，然后再调用本函数以获得最新信息。

相关信息:

驱动程序版本要求	和 SynCTI 驱动版本无关
Header	Shinitpci.h
Library	Shinitpci.lib
DLL	Shinitpci.dll

相关函数: [GetTotalPciBoard](#), [GetPciBoardSerialNo](#), [ReloadPciBoardInfo](#)

2.1.5.3.3 SsmGetBoardName

获取板卡的型号。

函数原型:

```
int SsmGetBoardName(int nBId, LPSTR lpBoardModel)
```

参数说明:

nBId	配置文件中指定的板卡 ID 编号
lpBoardModel	指定序号的板卡型号，用字符串表示。如下表：

板卡类型&相对应字符

返回值	含义
“SHD-60A-CT/PCI/SS1”	表示 PCI 60 路 1 号信令卡
“SHD-60A-CT/PCI/SS7”	表示 PCI 60 路 7 号信令卡
“SHD-60A-CT/PCI/ISDN”	表示 PCI 60 路 ISDN 信令卡
“SHD-120A-CT/PCI/SS1”	表示 PCI 120 路 1 号信令卡
“SHD-120A-CT/PCI/SS7”	表示 PCI 120 路 7 号信令卡
“SHD-120A-CT/PCI/ISDN”	表示 PCI 120 路 ISDN 信令卡
“SHD-60B-CT/PCI/FAX”	表示 PCI 60 路 B 型卡，该卡可配置成带传真通道的信令卡
“SHD-30B-CT/PCI/FAX”	表示 PCI 30 路 B 型卡，该卡可配置成带传真通道的信令卡
“SHD-30A-CT/PCI/SS1”	表示 PCI 30 路 1 号信令卡
“SHD-30A-CT/PCI/SS7”	表示 PCI 30 路 7 号信令卡
“SHD-30A-CT/PCI/ISDN”	表示 PCI 30 路 ISDN 信令卡
“SHD-30A-CT/PCI/FJ”	表示 PCI 30 路 A 型高阻复接卡
“SHD-60A-CT/PCI/FJ”	表示 PCI 60 路 A 型高阻复接卡
“SHD-30B-CT/PCI/FJ”	表示 PCI 30 路 B 型高阻复接卡
“SHD-60B-CT/PCI/FJ”	表示 PCI 60 路 B 型高阻复接卡
“SHT-16B-CT/PCI/FAX” 或 “SHT-16B-CT/PCI/MP3 ^[1] ”	表示 PCI 16 路 B 型 MP3/FAX 录音卡
“SHT-8B/PCI”	表示 PCI 8 路 B 型模块卡
“SHT-8B/PCI/FAX”	表示 PCI 8 路 B 型带软传真模块卡
“SHT-8C/PCI/FAX”	表示 PCI 8 路 C 型带软传真模块卡
“SHT-8C/PCI/EC”	表示 PCI 8 路 C 型卡
“SHT-16C-CT/PCI/EC”	表示 PCI 16 路 C 型卡
“SHT-16B-CT/PCI”	表示 PCI 16 路 B 型卡

“SHT-16C-CT/PCI/FAX”	表示 PCI 16 路 C 型带软传真模块卡
“SHT-16A-CT/PCI/FAX”	表示 PCI 16 路 A 型带软传真模块卡
“DST-1600”	表示 PCI 16 路 B 型卡
“SHT-16B-CT/cPCI”	表示 cPCI 16 路 B 型模块卡
“SHD-30A-CT/cPCI”	表示 cPCI 30 路卡 (支持 SS1、ISDN、SS7)
“SHD-60A-CT/cPCI”	表示 cPCI 60 路卡 (支持 SS1、ISDN、SS7)
“SHT-16B-CT/cPCI/MP3” 或 “SHT-16B-CT/cPCI/FAX”	表示 cPCI 16 路 B 型 MP3/FAX 录音卡
“SHD-480A-CT/cPCI”	表示 cPCI 带会议功能的 16E1 卡
“SHD-480S-CT/cPCI”	表示 cPCI 不带会议功能的 16E1 卡
“SHD-60B-CT/cPCI/FAX”	表示 cPCI 60 路 B 型卡，该卡可配置成带传真通道的信令卡
“SHR-16DA-CT/PCI”	表示 PCI 16 路数字电话录音卡
“SHR-24DA-CT/PCI”	表示 PCI 24 路数字电话录音卡
“SHT-120A-CT/PCI”	表示 PCI 120 路大容量模拟卡
“SHT-2B/USB”	表示 2 路 B 型 USB 模块卡
“SHT-4B/USB”	表示 4 路 B 型 USB 模块卡
“SHT-2A/USB”	表示 2 路 A 型 USB 模块卡
“SHT-4A/USB”	表示 4 路 A 型 USB 模块卡
“SHT-120A-CT/cPCI”	表示 cPCI 120 路大容量模拟卡
“SHD-240D-CT/PCI”	表示 PCI 8 中继数字卡
“SHD-240D-CT/PCI/EC”	表示 PCI 带增强回波消除 8 中继数字卡
“SHD-120D-CT/PCI”	表示 PCI 4 中继数字卡
“SHD-120D-CT/PCI/EC”	表示 PCI 带增强回波消除 4 中继数字卡
“SHD-120D-CT/PCI/CAS”	表示 PCI 支持 1 号信令的 4 中继数字卡
“SHD-240D-CT/PCI/CAS”	表示 PCI 支持 1 号信令的 8 中继数字卡
“SHD-30C-CT/PCI”	表示 PCI 30 路 C 型卡
“SHD-30C-CT/PCI/FAX”	表示 PCI 30 路带传真资源的 C 型卡
“SHD-60C-CT/PCI”	表示 PCI 60 路 C 型卡
“SHD-60C-CT/PCI/FAX”	表示 PCI 60 路带传真资源的 C 型卡
“SHF-2D/PCI”	表示 PCI 2 路 D 型模拟传真卡
“SHF-4D/PCI”	表示 PCI 4 路 D 型模拟传真卡
“SHF-4D/PCle”	表示 PCle 4 路 D 型模拟传真卡
“DTP-30C/PCI”	表示 PCI 30 路高阻复接卡
“DTP-30C/PCI+”	表示 PCI 30 路高阻复接卡
“DTP-60C/PCI”	表示 PCI 60 路高阻复接卡
“DTP-60C/PCI+”	表示 PCI 60 路高阻复接卡
“DTP-120C/PCI”	表示 PCI 120 路高阻复接卡
“DTP-120C/PCI+”	表示 PCI 120 路高阻复接卡
“DTP-30C/PCle”	表示 PCle 30 路高阻复接卡
“DTP-30C/PCle+”	表示 PCle 30 路高阻复接卡
“DTP-60C/PCle”	表示 PCle 60 路高阻复接卡
“DTP-60C/PCle+”	表示 PCle 60 路高阻复接卡
“DTP-120C/PCle”	表示 PCle 120 路高阻复接卡
“DTP-120C/PCle+”	表示 PCle 120 路高阻复接卡
“SHN-32A-CT/PCI”	表示 32 路 IP 卡
“SHN-8B-CT/PCI+”	表示 8 路 B 型 IP 卡
“SHN-16B-CT/PCI+”	表示 16 路 B 型 IP 卡
“SHN-32B-CT/PCI+”	表示 32 路 B 型 IP 卡
“SHN-60B-CT/PCI+”	表示 60 路 B 型 IP 卡
“SHN-120B-CT/PCI+”	表示 120 路 B 型 IP 卡
“SHN-60B-CT/PCle+”	表示 PCle 60 路 B 型 IP 卡
“SHN-120B-CT/PCle+”	表示 PCle 120 路 B 型 IP 卡
“SHN-120B-CT/PCle/VAR”	表示 PCle 带 60 路变声资源的 120 路 B 型 IP 卡
“SHN-480C-CT/PCle”	表示 PCle 480 路 C 型 IP 卡
“ATP-24A/PCI”	表示 ATP-24A/PCI 模拟电话录音卡

“ATP-24A/PCI+”	表示 ATP-24A/PCI+模拟电话录音卡
“ATP-24A/PCIe”	表示 ATP-24A/PCIe 模拟电话录音卡
“ATP-24A/PCIe+”	表示 ATP-24A/PCIe+模拟电话录音卡
“DST-24B/PCI”	表示 PCI 24 路 B 型数字电话录音卡
“DST-24B/PCI+”	表示 PCI 增强型 24 路 B 型数字电话录音卡
“DST-24B/PCIe”	表示 PCIe24 路 B 型数字电话录音卡
“DST-24B/PCIe+”	表示 PCIe 增强型 24 路 B 型数字电话录音卡
“SHD-30E-CT/PCI(SSW)”	表示 PCI 1 中继数字卡
“SHD-30E-CT/PCI/FAX(SSW)”	表示 PCI 带传真资源 1 中继数字卡
“SHD-30E-CT/PCI/EC(SSW)”	表示 PCI 带增强回波消除 1 中继数字卡
“SHD-60E-CT/PCI(SSW)”	表示 PCI 2 中继数字卡
“SHD-60E-CT/PCI/FAX(SSW)”	表示 PCI 带传真资源 2 中继数字卡
“SHD-60E-CT/PCI/EC(SSW)”	表示 PCI 带增强回波消除 2 中继数字卡
“SHD-120E-CT/PCI(SSW)”	表示 PCI 4 中继数字卡
“SHD-120E-CT/PCI/FAX(SSW)”	表示 PCI 带传真资源 4 中继数字卡
“SHD-120E-CT/PCI/EC(SSW)”	表示 PCI 带增强回波消除 4 中继数字卡
“SHD-240E-CT/PCI(SSW)”	表示 PCI 8 中继数字卡
“SHD-240E-CT/PCI/FAX(SSW)”	表示 PCI 带传真资源 8 中继数字卡
“SHD-240E-CT/PCI/EC(SSW)”	表示 PCI 带增强回波消除 8 中继数字卡
“SHD-30E-CT/PCIe”	表示 PCIe 1 中继数字卡
“SHD-30E-CT/PCIe/FAX”	表示 PCIe 带传真资源 1 中继数字卡
“SHD-30E-CT/PCIe/EC”	表示 PCIe 带增强回波消除 1 中继数字卡
“SHD-60E-CT/PCIe”	表示 PCIe 2 中继数字卡
“SHD-60E-CT/PCIe/FAX”	表示 PCIe 带传真资源 2 中继数字卡
“SHD-60E-CT/PCIe/EC”	表示 PCIe 带增强回波消除 2 中继数字卡
“SHD-120E-CT/PCIe”	表示 PCIe 4 中继数字卡
“SHD-120E-CT/PCIe/FAX”	表示 PCIe 带传真资源 4 中继数字卡
“SHD-120E-CT/PCIe/EC”	表示 PCIe 带增强回波消除 4 中继数字卡
“SHD-240E-CT/PCIe”	表示 PCIe 8 中继数字卡
“SHD-240E-CT/PCIe/FAX”	表示 PCIe 带传真资源 8 中继数字卡
“SHD-240E-CT/PCIe/EC”	表示 PCIe 带增强回波消除 8 中继数字卡
“SHD-240E-CT/PCIe/VAR”	表示 PCIe 带变声资源 8 中继数字卡
“SHT-16D-CT/PCIe”	表示 PCIe 16 路 D 型模块卡
“PCM1280E”	表示 PCIe 128 路 PCM32 录音卡
“SynIPAnalyzer”	表示 SynIPAnalyzer
“SynIPRecorder”	表示 SynIPRecorder

返回值:

0	调用成功
-1	调用失败

功能描述:

本函数取得指定编号 ID 板卡的板卡型号。如果调用成功则将板卡型号放到 `IpBoardModel` 中。

注意事项:

本函数与 [GetPciBoardModel](#) 实现同样的功能，但是调用需经过 `Shp_a3.dll` 来控制。

相关信息:

驱动程序版本要求	SynCTI Ver.5.2.0.3 或更高
Header	<code>shpa3api.h</code>
Library	<code>shp_a3.lib</code>
DLL	<code>shp_a3.dll</code>

相关函数: [SsmGetAccreditIdEx](#), [SsmGetAccreditId](#), [SsmGetBoardModel](#)

2.1.5.4 获取板卡序列号

2.1.5.4.1 SsmGetPciSerialNo

获取板卡的序列号。

函数原型:

DWORD SsmGetPciSerialNo(int nBld)

参数说明:

nBld	板卡的 ID 号, 对应于配置文件中[BoardId=x]节中的 x 的值
------	---------------------------------------

返回值:

0	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
>0	板卡的序列号

功能描述:

获取板卡的序列号。Synway 板卡在出厂时, 会将板卡的序列号贴在板卡的背面上, 更多信息请参见板卡的硬件说明手册。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver.3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetAccreditIdEx](#), [SsmGetAccreditId](#)

2.1.5.4.2 GetPciBoardSerialNo

获取板卡的序列号。

函数原型:

DWORD GetPciBoardSerialNo (int nBld, DWORD pSerialNo)

参数说明:

nBld	实际安装板卡的 ID 编号
pSerialNo	存放卡序列号的指针

返回值:

-1	调用失败
0	调用成功号

功能描述:

本函数取得指定板卡的序列号。如果调用成功, 则将板卡序列号放到 pSerialNo 中。

注意事项:

本函数调用无需经过 Shp_a3.dll 来控制, 可以直接取得操作系统中已经安装的板卡的序列号。

如果用户加载 Shinitpci.dll 之后又卸载或安装了某块板卡, 需要先调用 [ReloadPciBoardInfo](#), 然后再调用本函数以获得最新信息。

相关信息:

驱动程序版本要求	和 SynCTI 驱动版本无关
Header	Shinitpci.h
Library	Shinitpci.lib
DLL	Shinitpci.dll

相关函数: [GetPciBoardModel](#), [GetTotalPciBoard](#), [ReloadPciBoardInfo](#)

2.1.5.5 获取文件的版本信息

2.1.5.5.1 SsmGetDIIVersion

获取文件“shp_a3.dll”的版本信息。

函数原型:

```
int SsmGetDIIVersion(PSSM_VERSION pDLLVersion)
```

参数说明:

pDLLVersion	文件版本信息结构指针
-------------	------------

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

获取应用程序调用文件“shp_a3.dll”的版本信息。

返回的文件版本信息的结构体具体如下:

```
typedef struct _VERSION
{
    UCHAR ucMajor;//主版本号
    UCHAR ucMinor;//次版本号
    USHORT usInternal;//内部版本号
    USHORT usBuild;//编译版本号
    UCHAR ucRelease;//编译模式
    UCHAR ucFeature;//编译环境
}SSM_VERSION, *PSSM_VERSION;
```

例如：安装驱动 5030 版本后，调用本函数获取的结果为：

```
typedef struct _VERSION
{
    UCHAR ucMajor;//=5
    UCHAR ucMinor;//=0
    USHORT usInternal;//=3
    USHORT usBuild; //=0
    UCHAR ucRelease;//=空格（保留未使用）
    UCHAR ucFeature; //=空格（保留未使用）
}SSM_VERSION, *PSSM_VERSION;
```

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver.3.1 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.1.6 获取通道属性的函数

2.1.6.1 SsmGetMaxCh

获取应用系统中的通道总数。

函数原型:

```
int SsmGetMaxCh(void)
```

参数说明: 无

返回值:

-1	调用失败
≥ 0	通道总数

功能描述:

获取应用系统中的通道总数。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver.3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetChType](#)

2.1.6.2 SsmGetChType

获取通道的类型。

函数原型:

```
int SsmGetChType(int ch)
```

参数说明:

ch	通道的逻辑编号
----	---------

返回值:

-1	调用失败
0	模拟中继线通道
2	坐席通道
3	模拟中继线录音通道
4	SS1 通道
6	TUP 通道
7	ISDN 通道（用户侧）
8	ISDN 通道（网络侧）
9	传真资源通道

10	磁石通道
11	ISUP 通道（中国 SS7 信令 ISUP）
12	数字电话线录音通道
13	Channel Bank 的 EM 通道
14	变声通道
16	SIP 通道
17	IP 资源卡通道
19	DASS2 通道
20	SHT 系列板卡未安装业务模块的通道
21	EM 控制通道
22	EM 语音通道
25	IPR 通道，即 SynIPR 录音通道
26	IPA 通道，即 SynIPR 数据包解析通道
27	DPNSS 通道

功能描述:

获取通道的类型。

注意事项:

- 配合 [SsmGetAutoCallDirection](#) 一起使用可以获取更具体的通道类型的简要说明。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetMaxCh](#)

2.1.6.3 SsmGetChHdInfo

获取通道所在板卡的编号，以及通道在板卡内部的物理编号。

函数原型:

```
int SsmGetChHdInfo(int ch, int * pnBld, int * pnBCh)
```

参数说明:

ch	通道编号
pnBld	返回 ch 所在的板卡编号，由应用程序分配存储空间
pnBCh	返回 ch 在板卡上的物理编号，由应用程序分配存储空间

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

获取通道所在板卡的编号，以及通道在板卡内部的物理编号。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver. 4.7.1.5 或更高
Header	shpa3api.h

Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetAppChId](#)

2.1.6.4 SsmGetAppChId

根据板卡的编号、通道的物理编号查询其逻辑编号。

函数原型:

```
int SsmGetAppChId(int * ch, int nBId, int nBCh)
```

参数说明:

ch	返回通道的逻辑编号, 由应用程序分配存储空间
nBId	通道所在板卡的编号
nBCh	通道的物理编号

返回值:

-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

根据板卡的编号、通道的物理编号查询其逻辑编号。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 4.7.1.5 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetChHdInfo](#)

2.1.7 设置通道属性的函数

2.1.7.1 SsmSetFlag

设置通道的控制属性。

函数原型:

```
int SsmSetFlag(int ch, int nType, long lPara)
```

参数说明:

ch	通道号	nType 选择需要设置的参数类别, lValue 为参数的值。nType、lValue 的取值范围及物理含义如下表所示:		
nType lPara	nType	描述	lPara	
	取值 shpa3api.h 中的宏			
	1 F_RCVDTMFSENS	设置 DTMF 信号的高电平、低电平的最小保持时间。 更多信息请参见配置项 ReceiveDtmfSensitive 的说明。	参见配置项 ReceiveDtmfSensitive 的说明	
	2 F_TXDTMFAMP	设置 DTMF 发生器产生的 DTMF 信号的幅度, 初始默认值: SHT、ATP 系列为 0X8976; SHD、DTP 系列为 0X3f3f。	高 8 位和低 8 位分别设置组成 DTMF 信号的两个频率的幅度。	
	5 F_RXR2FILTERTIME	设置 MFC R2 信号的最小保持时间。更多信息请参见配置项 RxR2FilterTime 的说明。	取值范围: 16~96 (必须为 16 的整数倍), 单位: 毫秒。	
	9 F_ClearInVoiceOnRcvDtmf	设置 DTMF 钳位功能。更多信息请参见配置项 ClearInVoiceOnRxDtmf 的说明。	=0: 关闭	

			=1: 开启 =0: 否 =1: 是
10	F_MixerResToBus	设置“下总线信号混音器”的输出信号是否作为“上总线混音器”的输入信号源。本功能只支持 SHT 系列（8/16 通道规格）和 SHD 系列板卡，更多信息请参见相关板卡的工作原理框图中的 K1-2 开关。	
11	F_HighAndLowFreqEnScale	设置 DTMF 信号中，高频分量部分能量与低频分量部分能量之间的比例。更多信息请参见配置项 HighAndLowFreqEnScale 的说明。	参见配置项 HighAndLowFreqEnScale 的说明。
12	F_DualAndAllFreqEnScale	设置 DTMF 信号的带内能量占全频能量的百分比判定门限。更多信息请参见配置项 DualAndAllFreqEnScale 的说明。	参见配置项 DualAndAllFreqEnScale 的说明。
13	F_EchoCancelInFsk	当 FSK 接收器正在工作时，是否关闭回波抵消器。更多信息请参见配置项 FskEchoCancelDelay 的说明。	=0: 关闭; =1: 开启。
17	F_ClearInVoiceOnRcv450Hz	设置信号音的总线切音	=0: 信号音上总线（默认） =1: 信号音不上总线
18	FSKMinGate	设置接收 FSK 的能量门限值	=0: 不设置 n(n>0): 所设置的门限
19	F_RECTOFILEA_CALLBACKTIMEA	设置函数 SsmRecToFileA 或 SsmRecToFileB 的回调时间。更多信息请参见该函数的说明。 注意：如果录音格式为 MP3（采用软编码），本参数设置的功能无效	取值范围： a 率, μ 率: 1365>n≥16 ADPCM: 2730>n≥64 硬件 MP3: 10922>n≥16 G729A: 10922>n≥20 VOX: 2730>n≥64 硬件 GSM: 6553>n≥40 软件 GSM: 1365>n≥16 单位: 毫秒 其他格式会在播放到一半缓冲区的时候进行一次回调，在播放完缓冲区的时候进行一次回调。
20	F_CALLERIDTYPE	设置对端交换机在模拟电话线上发送主叫号码的模式。更多信息请参见配置项 CallerIdStyle 的说明。	=0: DTMF 制式; =1: FSK 制式。
21	F_InVoiceToBusA	设置是否将通道的来话信号送到 TDM 交换总线上，也就是设置板卡原理框图中 K3 开关的状态。更多信息请参见配置项 InVoiceToBus 的说明。	=0: 关闭; =1: 开启。
22	F_EchoCancelInFskA	当 FSK 接收器正在工作时，是否关闭回波抵消器。更多信息请参见配置项 FskEchoCancelDelay 的说明。	=0: 关闭; =1: 开启。
23	F_ChToRingingOnRingCntA	设置铃流信号检测器的工作参数。详细信息请参见第 1 章中“ 模拟电话线上的铃流信号 ”部分内容，或配置项 ChToRingingOnRingCnt 的说明。	参见配置项 ChToRingingOnRingCnt 的说明。
24	F_SetAdjustCtlA	设置回波系数是否进行学习	=0: 关闭; =1: 开启。
25	F_RCVPHONUMHOLDUPA	设置“被叫号码拦截”功能的控制开关。更多信息请参见配置项 PhoNumHoldup 的说明。	取值范围: 0/1/2。
26	F_RELATIVEENGYHOOKDETECTA	设置模拟中继线通道上“增强的远端摘机检测器”的工作状态。更多信息请参见第 1 章中“ 增强的远端摘机检测器 ”部分内容，或配置项 RelativeEngyHookDetect 的说明。	=0: 关闭; =1: 开启。
27	F_CUTDTMFWHILERECORDING	设置监控卡通道录音时，是否对 DTMF 声音进行过滤。	=0: 关闭; =1: 开启。
28	F_AlwaysDetectBargein	动态设置 Bargein 功能的开关。	=0: 关闭; =1: 开启。

29	F_NoiseFilteringMinGate	动态设置底噪过滤门限。	=0: 关闭 0<n≤1000: 门限值。
30	F_SipSetSendDTMFType	动态设置 SIP 通道的 DTMF 发送方式。	=0: RFC2833; =1: 信令; =2: 带内。
31	F_FastPlayTime	设置通道放音的快进快退步长	参见配置项 FastPlayTime 的说明。

返回值:

-1	调用失败, 功能不支持
0	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
1	调用成功

功能描述:

设置通道的控制属性。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 4.0.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetFlag](#)

2.1.7.2 SsmGetFlag

获取通道上的功能开关的状态。

函数原型:

```
int SsmGetFlag(int ch, int nType, long* plValue)
```

参数说明:

ch	通道号	nType 选择需要设置的参数类别, 具体取值请参见函数的说明。plValue 返回参数的设定值。nType、 plValue 的物理含义如下表所示:
nType	取值	shpa3api.h 中的宏
1	F_RCVDTMFSENS	
2	F_TXDTMFAMP	
5	F_RXR2FILTERTIME	
9	F_ClearInVoiceOnRcvDtmf	详细信息请参见函数 SsmSetFlag 的说明。
10	F_MixerResToBus	
11	F_HighAndLowFreqEnScale	
12	F_DualAndAllFreqEnScale	
15	F_ISDNNet_WaitRemotePickup	plValue 返回驱动程序将通道迁移到 S_CALL_WAIT_REMOTE_PICKUP 状态的具体原因, 即收到对端交换机的消息类型: ◆ plValue=4: CALL PROCEEDING 消息; ◆ plValue=5: ALERTING 消息; ◆ plValue=6: CONNECT 消息。

		注意：本功能要求 SynCTI 的版本为 4.7.1.5 或更高。
17	F_ClearInVoiceOnRcv450Hz	
18	F_FSKMinGate	
19	F_RECTOFILEA_CALLBACKTIMEA	
20	F_CALLERIDTYPE	
21	F_InVoiceToBusA	
22	F_EchoCancelInFskA	
23	F_ChToRingingOnRingCntA	详细信息请参见函数 SsmSetFlag 的说明。
24	F_SetAdjustCtlA	
25	F_RCVPHONUMHOLDUPA	
26	F_RELATIVEENGGYHOOKDETECTA	
29	F_NoiseFilteringMinGate	
30	F_SipSetSendDTMFType	
31	F_FastPlayTime	

返回值：

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述：

获取通道上的功能开关的状态。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 4.0.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmSetFlag](#)

2.2 有关事件方式编程的函数

2.2.1 获取事件的函数

2.2.1.1 SsmWaitForEvent

请参见 [SsmWaitForEventA](#)。

2.2.1.2 SsmWaitForEventA

获取 Synway 驱动程序输出的事件。SsmWaitForEvent、SsmWaitForEventA 是异步函数，如果有事件则立即返回；如果没有事件，调用者线程将被阻塞，直到有事件发生或者超时才返回。

函数原型：

```
int SsmWaitForEvent( DWORD dwTimeOut, PMESSAGE_INFO pEvent)  
int SsmWaitForEventA( DWORD dwTimeOut, PSSM_EVENT pEventEx)
```

参数说明：

dwTimeOut	设置 SsmWaitForEvent 函数的工作模式，取值范围： 0: SsmWaitForEvent 工作在同步模式下，即不管是否有事件，函数都立即返回。此时，SsmWaitForEvent 与 SsmGetEvent 的功能完全一样； 0xffffffff: 如果没有事件，函数一直挂起，直到有事件时才返回；
-----------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------

	其它值： 设置最大等待时间，单位为毫秒。在等待事件期间，如果有事件发生，函数立即返回；如果没有事件发生，等待时间超过本参数的设定值后，函数才返回。
pEvent pEventEx	指向 MESSAGE_INFO 结构对象(pEvent)或 SSM_EVENT 结构对象(pEventEx)的指针，用于返回事件信息，仅当函数的返回值为 0 时才有效。更多信息请参见第 1 章中“MESSAGE_INFO”或“SSM_EVENT”部分内容。

返回值：

	有以下任意一种情况发生都将返回-2：
-2	◆ API 接口未打开 ◆ 设置为非事件轮询模式 ◆ 内存消息队列未分配
-1	超时
0	有事件发生，事件信息在 pEvent 或 pEventEx 中返回

功能描述：

获取 Synway 驱动程序输出的事件。如果驱动程序内部的事件缓冲区不为空，SsmWaitForEvent 或 SsmWaitForEventA 会立即返回；如果没有事件，本函数会等待事件的发生，直到等待时间超过参数 dwTimeOut 的设定值时，才会返回。

注意事项：

- 只有当驱动程序工作于事件等待模式时才能调用本函数，否则函数将返回失败；
- 如果使用 MESSAGE_INFO，只能调用函数 SsmWaitForEvent；如果使用 SSM_EVENT，只能调用函数 SsmWaitForEventA；
- 除了输出事件的数据结构不同以外，函数 SsmWaitForEvent 与 SsmWaitForEventA 的功能均完全相同。SsmWaitForEventA 通常用于 DST 系列板卡的 D 信道事件编程；
- 事件队列中最多能够保存的事件总数 MaxEventNum=TotalAppCh*MaxEventPerChannel，TotalAppCh 用来设置系统中的通道总数，MaxEventPerChannel 用来设置一个通道的事件队列的深度；
- 如果事件数大于 MaxEventNum，新的事件不会被保存到事件队列中。

相关信息：

驱动程序版本要求	SsmWaitForEvent 要求 SynCTI Ver. 4.0 或更高； SsmWaitForEventA 要求 SynCTI Ver. 4.7.3.0 或更高。
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmSetEvent](#)

2.2.1.3 SsmGetEvent

请参见 [SsmGetEventA](#)。

2.2.1.4 SsmGetEventA

获取 Synway 驱动程序输出的事件。SsmGetEvent、SsmGetEventA 是同步函数，不论有没有事件都会立即返回。

函数原型：

```
int SsmGetEvent(PMESSAGE_INFO pEvent)
```

```
int SsmGetEventA(PSSM_EVENT pEventEx)
```

参数说明:

pEvent	指向 MESSAGE_INFO 结构对象(pEvent)或 SSM_EVENT 结构对象(pEventEx)的指针, 用于返回事件信息, 仅当函数的返回值为 0 时才有效。更多信息请参见第 1 章中“MESSAGE_INFO”或“SSM_EVENT”部分内容。
pEventEx	

返回值:

-1	有以下任意一种情况发生都将返回-1: ◆ API 接口未打开 ◆ 设置为非事件轮询模式 ◆ 没有事件输出
0	有事件发生, 事件信息在 pEvent 或 pEventEx 中返回

功能描述:

获取 Synway 驱动程序输出的事件。

注意事项:

- 只有当驱动程序工作于事件等待模式时才能调用本函数, 否则函数将返回失败;
- 如果使用 MESSAGE_INFO, 只能调用函数 SsmGetEvent; 如果使用 SSM_EVENT, 只能调用函数 SsmGetEventA;
- 除了输出事件的数据结构不同以外, 函数 SsmGetEvent 与 SsmGetEventA 的功能均完全相同。SsmGetEventA 通常用于 DST 系列板卡的 D 信道事件编程。

相关信息:

驱动程序版本要求	SsmGetEvent 要求 SynCTI Ver. 4.0 或更高; SsmGetEventA 要求 SynCTI Ver. 4.7.3.0 或更高。
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetEvent](#)

2.2.2 过滤输出事件

2.2.2.1 SsmSetEvent

设置驱动程序的事件输出模式, 以及允许或禁止驱动程序输出某个特定事件。

函数原型:

```
int SsmSetEvent(WORD wEvent, int nReference, BOOL bEnable, PEVENT_SET_INFO pEventSet)
```

参数说明:

wEvent	选择事件, 取值范围: 0~0xffff: 事件编码, 取值范围请参见第 1 章中“MESSAGE_INFO”部分内容。此时, SsmSetEvent 用于通知驱动程序抛出或不抛出特定事件。 0xffff: 设置整个驱动程序的工作模式, 即事件等待模式和事件回调模式, 此时参数 nReference 必须为-1。驱动程序内部的事件队列的深度可以通过配置项 MaxEventPerChannel 进行设置。更多信息请参见第 1 章中“ 设置驱动程序的事件输出模式 ”部分内容。		
nReference	事件的参考值, wEvent=0xffff 时, 本参数必须为-1, 用于设置整个驱动程序的工作模式。当参数 wEvent≠0xffff 时, nReference 的物理含义取决于 wEvent 的类型, 如下表所示: <table border="1"><tr><td>wEvent</td><td>nReference 的含义及取值范围</td></tr></table>	wEvent	nReference 的含义及取值范围
wEvent	nReference 的含义及取值范围		

	E_CHG_ISDNStatus E_CHG_PcmLinkStatus E_CHG_RemotePCMBlock	数字中继线的逻辑编号: nReference=-1: 对系统中的全部数字中继线进行设置; 0≤nReference≤M-1: 对本参数指定的数字中继线进行设置 注: M 为系统中的数字中继线总数																								
	E_RCV_Ss7Msu E_RCV_Ss7SpyMsu	0-保留 0-保留																								
	E_CHG_Mtp3State	DPC 的编号: nReference=-1: 对系统中的全部 DPC 设置; 0≤nReference≤M-1: 对本参数指定的 DPC 进行设置																								
	E_SYS_TIMEOUT	定时器的 ID: nReference=-1: 对系统中的全部定时器设置; 0≤nReference≤M-1: 对本参数指定的定时器进行设置																								
	E_CHG_SpyState	SpyCic 的逻辑编号: nReference=-1: 对系统中的全部被监控电路设置; 0≤nReference≤M-1: 对本参数指定的被监控电路进行设置																								
	E_CHG_SpyHangupInfo	SpyCic 的逻辑编号: nReference=-1: 对系统中的全部被监控电路设置; 0≤nReference≤M-1: 对本参数指定的被监控电路进行设置																								
	E_CHG_CICRxPhoNumBuf E_CHG_CICState E_PROC_CICAuotDial	虚电路编号: nReference=-1: 对系统中的虚电路设置; 0≤nReference≤M-1: 对本参数指定的虚电路进行设置																								
bEnable	其它事件	通道的逻辑编号: nReference=-1: 对系统中的全部 N 个通道进行设置 0≤nReference≤N-1: 对本参数指定的通道进行设置																								
		事件输出的控制标志。当 wEvent=0xffff 且 nReference=-1 时, bEnable=False, 表示默认输出事件为空; bEnable=True 表示默认输出事件由 DefaultEventOutput 指定。当参数 wEvent≠0xffff 时, bEnable 取值范围: =True: 抛出 wEvent =False: 不抛出 wEvent																								
pEventSet		指向具有 EVENT_SET_INFO 结构的对象的指针, 用于设置事件的输出条件。如果参数 wEvent=0xffff, 本参数不能为 NULL。EVENT_SET_INFO 结构的声明为: <pre>typedef struct _EVENT_SET_INFO { DWORD dwWorkMode; LPVOID lpHandlerParam; DWORD dwOutCondition; DWORD dwOutParamVal; DWORD dwUser; } EVENT_SET_INFO,*PEVENT_SET_INFO;</pre>																								
		各参数的含义如下表所示:																								
		<table border="1"> <thead> <tr> <th colspan="3">参数名称: dwWorkMode</th> </tr> </thead> <tbody> <tr> <td colspan="3">设置驱动程序抛出事件的模式, 只有当参数 wEvent=0xffff 时才有效, wEvent 为其它值时本参数将被忽略。取值范围:</td> </tr> <tr> <th>取值</th> <th>event.h 中的宏</th> <th>描述</th> </tr> <tr> <td>0</td> <td>NO_EVENT</td> <td>驱动程序不输出任何事件。</td> </tr> <tr> <td>1</td> <td>EVENT_POLLING</td> <td>驱动程序工作于事件等待模式, 参数 dwOutCondition 和 dwOutParamVal 有效, lpHandlerParam 和 dwUser 无效。</td> </tr> <tr> <td>2</td> <td>EVENT_CALLBACK</td> <td>驱动程序工作于事件回调模式, 此时参数 lpHandlerParam 和 dwUser 有效。</td> </tr> <tr> <td>3</td> <td>EVENT_MESSAGE</td> <td>驱动程序工作于 Windows 消息模式, 所有的消息都将被发送到 windows 消息队列中。</td> </tr> <tr> <td>4</td> <td>EVENT_POLLINGA</td> <td>驱动程序工作于事件等待模式, 参数 dwOutCondition</td> </tr> </tbody> </table>	参数名称: dwWorkMode			设置驱动程序抛出事件的模式, 只有当参数 wEvent=0xffff 时才有效, wEvent 为其它值时本参数将被忽略。取值范围:			取值	event.h 中的宏	描述	0	NO_EVENT	驱动程序不输出任何事件。	1	EVENT_POLLING	驱动程序工作于事件等待模式, 参数 dwOutCondition 和 dwOutParamVal 有效, lpHandlerParam 和 dwUser 无效。	2	EVENT_CALLBACK	驱动程序工作于事件回调模式, 此时参数 lpHandlerParam 和 dwUser 有效。	3	EVENT_MESSAGE	驱动程序工作于 Windows 消息模式, 所有的消息都将被发送到 windows 消息队列中。	4	EVENT_POLLINGA	驱动程序工作于事件等待模式, 参数 dwOutCondition
参数名称: dwWorkMode																										
设置驱动程序抛出事件的模式, 只有当参数 wEvent=0xffff 时才有效, wEvent 为其它值时本参数将被忽略。取值范围:																										
取值	event.h 中的宏	描述																								
0	NO_EVENT	驱动程序不输出任何事件。																								
1	EVENT_POLLING	驱动程序工作于事件等待模式, 参数 dwOutCondition 和 dwOutParamVal 有效, lpHandlerParam 和 dwUser 无效。																								
2	EVENT_CALLBACK	驱动程序工作于事件回调模式, 此时参数 lpHandlerParam 和 dwUser 有效。																								
3	EVENT_MESSAGE	驱动程序工作于 Windows 消息模式, 所有的消息都将被发送到 windows 消息队列中。																								
4	EVENT_POLLINGA	驱动程序工作于事件等待模式, 参数 dwOutCondition																								

		和 dwOutParamVal 有效， IpHandlerParam 和 dwUser 无效。
5	EVENT_CALLBACKA	驱动程序工作于事件回调模式，此时参数 IpHandlerParam 和 dwUser 有效。
参数名称: <i>IpHandlerParam</i>		
仅当 dwWorkMode=EVENT_CALLBACK ， EVENT_CALLBACKA 或 EVENT_MESSAGE 时有效， dwWorkMode 为其它值时，本参数将被忽略。		
➢ dwWorkMode=EVENT_CALLBACK <i>IpHandlerParam</i> 为回调函数指针 <i>IpHandlerParam</i> 的格式为： <pre>int CALLBACK CallBackFun(WORD wEvent,int nReference,DWORD dwParam,DWORD dwUser);</pre> 其中，参数 wEvent 、 nReference 和 dwParam 的含义与 MESSAGE_INFO 结构中声明的变量的含义完全相同，详细信息请参见函数 SsmWaitForEvent 或 SsmGetEvent 的说明。参数 dwUser 即为调用本函数时应用程序传递给驱动程序的 dwUser 参数。		
➢ dwWorkMode=EVENT_CALLBACKA <i>IpHandlerParam</i> 为回调函数指针。 <i>IpHandlerParam</i> 的格式为： <pre>int CALLBACK CallBackFunA(PSSM_EVENT pEvent);</pre> 其中， pEvent 是指向 SSM_EVENT 结构对象的指针，用于返回事件信息，详细说明请参见函数 SsmWaitForEventA 或 SsmGetEventA 。 注意：当回调函数返回-1 时，会立即禁止所有事件的抛出；如果需要重新抛出事件，请使用函数 SsmSetEvent 再次进行设置。		
➢ dwWorkMode=EVENT_MESSAGE <i>IpHandlerParam</i> 为应用程序接收驱动事件的窗口句柄。		
参数名称: <i>dwUser</i>		
用户自定义参数，仅当 dwWorkMode=EVENT_CALLBACK 或者 EVENT_CALLBACKA 时有效， dwWorkMode 为其它值时，本参数将被忽略。驱动程序会保存此参数，当驱动程序抛出事件时，会将此参数透明地传递给 IpHandlerParam 指定的回调函数中的同名参数。		
参数名称: <i>dwOutCondition</i> 、 <i>dwOutParamVal</i>		
dwOutCondition 设置事件的输出条件， dwOutParamVal 设置事件输出时携带的参数类别。当事件发生时，驱动程序会将 dwOutParamVal 设定的数据类别放在结构 MESSAGE_INFO 或 SSM_EVENT 的 dwParam 参数中。 dwOutCondition 、 dwOutParamVal 不是必选参数，若参数 dwOutCondition 使用缺省值时，可设置参数 pEventSet=NULL ；若参数 wEvent 的取值如下且参数 pEventSet≠NULL 时，在每次调用 SsmSetEvent 之前，参数 dwOutCondition 必须设置，否则会有不可预知的后果：		
◊ wEvent = E_CHG_LineVoltage: 当模拟电话线的电压变化幅度超过 dwOutCondition 的设定值时（默认可设为 5），驱动程序才会抛出 E_CHG_LineVoltage 事件。 dwOutParamVal 保留未用。当 wEvent=0xffff 且 nReference=-1 时，会将 dwOutCondition 自动设置为 5。		
◊ wEvent = E_PROC_PlayFile: dwOutCondition 设置驱动程序抛出 E_PROC_PlayFile 的时间间隔（默认可为 1000），单位为毫秒。 dwOutParamVal 设置输出参数的类型，可选值为： =0: 输出完成播放的百分比； =1: 输出完成播放的时间； =2: 输出完成播放的字节总数； =3: 输出尚未完成播放字节总数。 当 wEvent=0xffff 且 nReference=-1 时，会将 dwOutCondition 自动设置为 1000 毫秒， dwOutParamVal 设置为 1。		
◊ wEvent = E_PROC_PlayMem: ◆ dwOutParamVal=END_HALF_BUFFER: 当驱动程序内部的放音指针越过缓冲区的中间位置或尾部时，抛出 E_PROC_PlayMem 事件。如果放音指针位于缓冲区的前半部分，输出-1；位于后半部分，输出-2； ◆ dwOutParamVal=END_BUFFER: 当驱动程序的放音指针越过缓冲区的尾部时，抛出 E_PROC_PlayMem 事件，		

输出-2。

◆ dwOutParamVal=MEM_OFFSET:

驱动程序累计完成了一定时长的语音数据的播放后，抛出 E_PROC_PlayMem，输出驱动程序内部的放音指针（即在缓冲区中的偏移量）。时间长度由 dwOutCondition 指定，单位为毫秒。

◆ dwOutParamVal=MEM_BYTES:

驱动程序累计完成了一定字节数量的语音数据的播放后，抛出 E_PROC_PlayMem，输出驱动程序内部的放音指针（即在缓冲区中的偏移量）。字节数量由 dwOutCondition 指定（默认可设为 64）。

当 wEvent=0xffff 且 nReference=-1 时，会将 dwOutParamVal 自动设置为 MEM_OFFSET，dwOutCondition 自动设置为 64 毫秒。

◆ wEvent = E_PROC_RecordFile:

dwOutCondition 设置驱动程序抛出 E_PROC_RecordFile 事件的时间间隔（默认可设为 1000），单位为毫秒。dwOutParamVal 设置输出参数的类型，可选值为：

=RECORD_TIME: 输出已经录制完成的时间，单位为毫秒；

=RECORD_BYTES: 输出已经录制完成的字节总数，单位为字节。

当 wEvent=0xffff 且 nReference=-1 时，会将 dwOutCondition 自动设置为 1000 毫秒，dwOutParamVal 自动设置为 RECORD_TIME。

◆ wEvent = E_PROC_RecordMem:

◆ dwOutParamVal=END_HALF_BUFFER:

驱动程序内部的录音指针越过录音缓冲区的中间位置或尾部时，驱动程序抛出 E_PROC_RecordMem 事件。如果录音指针位于缓冲区的前半部分，输出-1；位于后半部分，输出-2。

◆ dwOutParamVal=END_BUFFER:

当驱动程序内部的录音指针越过缓冲区的尾部时，抛出 E_PROC_RecordMem 事件，输出-2。

◆ dwOutParamVal=MEM_OFFSET:

驱动程序累计完成了一定时长的语音数据的录音后，抛出 E_PROC_RecordMem，输出驱动程序内部的录音指针（即在缓冲区中的偏移量）。时间长度由 dwOutCondition 指定，单位为毫秒。

◆ dwOutParamVal=MEM_BYTES:

驱动程序累计完成了一定字节数量的语音数据的录音后，抛出 E_PROC_RecordMem，输出驱动程序内部的录音指针（即在缓冲区中的偏移量）。字节数量由 dwOutCondition 指定（默认可设为 64）。

当 wEvent=0xffff 且 nReference=-1 时，会将 dwOutParamVal 自动设置为 MEM_OFFSET，dwOutCondition 自动设置为 64 毫秒。

◆ wEvent = E_OverallEnergy:

当全频能量的变化幅度超过 dwOutCondition 的设定值时（默认可设为 1000），驱动程序才会抛出 E_OverallEnergy 事件。dwOutParamVal 保留未用。当 wEvent=0xffff 且 nReference=-1 时，会将 dwOutCondition 自动设置为 1000。

◆ wEvent = E_CHG_PeakFrq:

当峰值频率的变化幅度超过 dwOutCondition 的设定值时（默认可设为 100），驱动程序才会抛出 E_CHG_PeakFrq 事件。dwOutParamVal 保留未用。当 wEvent=0xffff 且 nReference=-1 时，会将 dwOutCondition 设置为 100。

◆ wEvent = 其它事件:

dwOutParamVal 和 dwOutParamVal 均保留未用，可设置为 0。

返回值:

-1	失败
0	成功

功能描述:

当 wEvent=0xffff 时，设置驱动程序的事件输出模式。有关 Synway 驱动程序的事件输出模式的信息请参见第 1 章中“[设置驱动程序的事件输出模式](#)”部分内容。

当 wEvent≠0xffff 时，允许或禁止驱动程序输出某个特定事件，若有输出参数，将设置其相关的输出参数。需首先设置事件的输出模式，再调整某个待定事件的输出规则。

注意事项:

- SynCTI 驱动程序启动后，会自动工作在轮询模式。如果应用程序使用事件等待方式或事件回调方式进行编程，必须在初始化驱动程序后，调用本函数来改变驱动程序的事件输出模式；
- 如果 wEvent=0xffff，应用程序只能调用一次本函数，并且参数 pEventSet 不能为 NULL，否则函数会返回失败。
- 如果 wEvent≠0xffff，pEventSet 可以为 NULL。如果 pEventSet 不为 NULL，其成员变量 dwWorkMode 必须和之前设置的事件输出模式一致。驱动目前尚不支持多个事件使用不同的方式输出。

相关信息：

驱动程序版本要求	SynCTI Ver.4.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数： [SsmWaitForEvent](#), [SsmWaitForEventA](#), [SsmGetEvent](#), [SsmGetEventA](#), [SsmPutUserEvent](#), [SsmPutUserEventA](#)

代码范例：请参见第 1 章中“[事件等待方式的编程示例](#)”或“[事件回调方式的编程示例](#)”部分内容。

2.2.3 获取编程模式

2.2.3.1 SsmGetEventMode

获取编程模式。

函数原型：

```
int SsmGetEventMode(WORD wEvent, int nReference, PWORD pwEnable, PEVENT_SET_INFO pEventSet)
```

参数说明：

wEvent	事件编码。 取值范围请参见第 1 章中“ MESSAGE_INFO ”部分内容，本参数不能取值为 0xffff
nReference	事件的参考值： nReference 的物理含义取决于 wEvent 的类型，具体含义同 SsmSetEvent 中参数 nReference，但不能取值为 -1
pwEnable	存放事件输出的控制标志，取值范围： =TRUE：抛出 wEvent =FALSE：不抛出 wEvent
pEventSet	指向具有 EVENT_SET_INFO 结构的对象的指针，用于存放 SsmSetEvent 函数设置的事件输出条件。EVENT_SET_INFO 结构的具体声明见 SsmSetEvent

返回值：

-1	失败
0	成功

功能描述：

获取应用程序所设置的驱动程序的事件输出模式。

注意事项：

- 当使用本函数获取应用程序的事件编程模式时，wEvent 参数只需传入某个事件，而不支持传入 0xffff。

相关信息：

驱动程序版本要求	SynCTI Ver.4.8.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数： [SsmSetEvent](#)

2.2.4 通过驱动程序输出应用程序的自定义事件

2.2.4.1 SsmPutUserEvent

请参见 [SsmPutUserEventA](#)

2.2.4.2 SsmPutUserEventA

将应用程序的自定义事件添加到驱动程序内部的事件队列中。

函数原型:

```
int SsmPutUserEvent( WORD wEvent, int nReference, DWORD dwParam)  
int SsmPutUserEventA(PSSM_EVENT pEvent)
```

参数说明:

wEvent	用户自定义的事件编码。注意：自定义的事件编码不要与驱动程序定义的事件编码重复，建议从 0xffffe 开始取值，逐步递减
nReference	事件输出参考值，会通过驱动程序内部的事件队列被透明地回传给应用程序
dwParam	事件输出参数，会通过驱动程序内部的事件队列被透明地回传给应用程序
pEvent	指向 SSM_EVENT 结构对象的指针，用于输入事件信息到驱动程序。pEvent 指向的结构对象会通过驱动程序内部的事件队列被透明地回传给应用程序

返回值:

-1	添加失败
0	添加成功

功能描述:

将应用程序的自定义事件添加到驱动程序内部的事件队列中。除了使用的事件数据结构不同以外，函数 SsmPutUserEvent 与 SsmPutUserEventA 的功能均完全相同。SsmPutUserEventA 通常用于 DST 系列板卡的 D 信道事件编程。

驱动程序会将应用程序提交的事件先缓存到一个临时缓冲区中，在硬件中断信号到达的时候再将其写入到驱动程序的事件输出队列。配置项 [MaxUserEventSize](#) 用于设置临时缓冲区的深度。

注意事项:

- 本函数只能在“事件等待”的编程模式下调用。有关事件等待模式的信息请参见第 1 章中“[设置驱动程序的事件输出模式](#)”部分内容；
- 如果使用 MESSAGE_INFO，只能使用函数 SsmPutUserEvent；如果使用 SSM_EVENT，只能使用函数 SsmPutUserEventA；
- 如果配置项 [MaxUserEventSize](#) 的设定值为 0，本函数会立即返回失败。

相关信息:

驱动程序版本要求	SsmPutUserEvent 要求 SynCTI Ver. 4.0 或更高； SsmPutUserEventA 要求 SynCTI Ver. 4.7.3.0 或更高；
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmWaitForEvent](#), [SsmWaitForEventA](#), [SsmGetEvent](#), [SsmGetEventA](#)

2.3 有关呼叫状态机的函数

2.3.1 有关去话呼叫的函数（CTI 系列）

2.3.1.1 搜索空闲通道

2.3.1.1.1 SsmSearchIdleCallOutCh

在具有去话呼叫能力的通道中搜索一个空闲的通道。

函数原型：

```
int SsmSearchIdleCallOutCh(WORD wSearchMode, DWORD dwPrecedence)
```

参数说明：

指定搜索的模式，可以是下列比特的操作：	
wSearchMode	比特
	Bit0
	是否在模拟中继线通道中搜索。=1：是；=0：否
	Bit1
	是否在 SS1 通道中搜索。=1：是；=0：否
	Bit2
	是否在 TUP 通道中搜索。=1：是；=0：否
	Bit3
	是否在 ISDN 通道（用户侧）中搜索。=1：是；=0：否
	Bit4
	是否在 ISDN 通道（网络侧）中搜索。=1：是；=0：否
	Bit5
	是否在 ISUP 通道中搜索。=1：是；=0：否
	Bit6
	是否在指定信令链路组对应的 TUP 通道中搜索。=1：是；=0：否
dwPrecedence	Bit7
	是否在指定信令链路组对应的 ISUP 通道中搜索。=1：是；=0：否
	Bit8
	是否在 SIP 类型的 IP 通道中搜索。=1：是；=0：否
	Bit9
	保留
	Bit10
	是否在指定 PCM 对应的 SS7 通道中搜索。=1：是；=0：否
	Bit11
	是否在指定 PCM 对应的 ISDN 通道中进行搜索。=1：是；=0：否
	Bit12
	是否指定 ISDN 或 SIP 通道搜索范围。=1：是；=0：否
	Bit13
	是否指定模拟中继通道搜索范围。=1：是；=0：否
	Bit14
	是否在模拟坐席通道中搜索。=1：是；=0：否
	Bit15
	是否指定模拟坐席通道搜索范围。=1：是；=0：否
wSearchMode 中 Bit6 或 Bit7 (TUP 或 ISUP 通道) 置为 1：选择信令链路组； wSearchMode 中 Bit10 或 Bit11 (SS7 或 ISDN 通道) 置为 1：选择 PCM 逻辑编号； wSearchMode 中 Bit12 置为 1：本参数高 16 位表示搜索的结束通道，低 16 位表示搜索的起始通道； wSearchMode 中 Bit13 置为 1：本参数高 16 位表示搜索的结束通道，低 16 位表示搜索的起始通道； wSearchMode 中 Bit15 置为 1：本参数高 16 位表示搜索的结束通道，低 16 位表示搜索的起始通道； 其它类型通道：应设置为 0	

返回值：

-1	调用失败。失败原因可以通过调用函数 SsmGetLastErrMsg 获得
≥0	搜索到的通道编号

功能描述：

在具有去话呼叫能力的通道中搜索一个空闲的通道。驱动程序内部维护了空闲通道的队列。

注意事项：

- 当函数调用后，对于模拟中继通道，通道状态仍旧保持为“空闲”状态；对于 ISDN、SS7、SS1、SIP 通道，通道迁移到“呼出锁定”状态；

- SS1, ISDN 协议中搜索通道, 按每一个 E1 的时隙顺序 0, 1, 2, 3...31 搜索。(除 0 和 16ts)
- SS7 协议中搜索通道, 按该 E1 所配置 OPC 和 DPC 的大小;
- 点码大的一方控偶时隙, 搜索通道函数根据偶时隙从 0, 2, 4, ……, 30, 31, 29, 27, ……, 1 依次选择。(除 0, 1 和 16 ts);
- 点码小的一方控奇时隙, 搜索通道函数根据奇时隙从 1, 3, 5, ……, 31, 30, 28, 26, ……, 0 依次选择。(除 0, 1 和 16 ts);
- 对于 ISDN 协议, 当设置为指定 PCM 搜索空闲通道(wSearchMode 比特 11 位为 1)时, 如果 dwPrecedence 为-1, 那么全 pcm 搜索空闲通道。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.3.1 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmAutoDial](#), [SsmChkAutoDial](#)

2.3.1.2 发起去话呼叫

2.3.1.2.1 SsmAutoDial

请参见 [SsmAutoDialEx](#)

2.3.1.2.2 SsmAutoDialEx

向驱动程序提交 AutoDial 任务, 发起一次去话呼叫。SsmAutoDialEx 可以设置 TUP 协议中的主叫用户线标识字段中的地址表示语(可用配置项 [CallerAddrInd](#) 配置), 通常用于 TUP 通道。

函数原型:

```
int SsmAutoDial(int ch, LPSTR szPhoNum)  
int SsmAutoDialEx(int ch, LPSTR szPhoNum, WORD wParam)
```

参数说明:

ch	通道编号 存放被叫方电话号码的缓冲区指针, 最大长度为 50 ◆ 模拟中继线通道: 如果 szPhoNum 中的字符既不是标准的 DTMF 字符, 也不是指定的闪断字符, 则该字符将被视为一个延时字符。有关闪断字符和延时字符的信息请参见函数 SsmTxDtmf 的说明 ◆ SS1 通道、TUP/ISUP 通道、ISDN 通道: 有效的电话号码必须是'0'~'9'之间的数字字符, 任何非数字字符将被驱动程序忽略 ◆ SIP 类型的 IP 通道, 格式为: ["display-name" <][sip:/sips:][user@]host[:port][>] ■ display-name: 本端别名, 缺省为空 ■ sip:/sips/: 缺省为 sip:, sips:则是 sip 消息传输加密模式 ■ user: 用户显示名, 缺省为空 ■ host: 可以是主机名、IPv4 格式的地址或 SIP 注册服务器域名 ■ port: 缺省为 5060												
wParam	本参数只适用于 TUP 通道 高 4 比特应设置为 0, 低 4 比特位(记为 DCBA)有效: <table border="1"><thead><tr><th>比特</th><th>含义</th><th>取值</th><th>描述</th></tr></thead><tbody><tr><td>BA</td><td>主叫号码性质表示语</td><td>00</td><td>市内用户号码</td></tr><tr><td></td><td></td><td>01</td><td>国内备用</td></tr></tbody></table>	比特	含义	取值	描述	BA	主叫号码性质表示语	00	市内用户号码			01	国内备用
比特	含义	取值	描述										
BA	主叫号码性质表示语	00	市内用户号码										
		01	国内备用										

		10	国内有效号码
		11	国际号码
C	提供主叫用户号码表示语	0	未限制提供主叫号码
		1	限制提供主叫号码
D	主叫号码不全表示语	0	未表示不全
		1	主叫号码不全

返回值:

-1	调用失败。失败原因可以通过调用函数 SsmGetLastErrMsg 获得
0	调用成功

功能描述:

向驱动程序提交 AutoDial 任务，发起一次去话呼叫。

AutoDial 任务启动后，通道的状态机开始自动进行去话呼叫接续，相关内容请参见通道的状态转移部分内容。

每当本次去话呼叫有任何进展时，驱动程序会向应用程序抛出 [E PROC AutoDial](#) 事件。应用程序也可以在调用 [SsmChkAutoDial](#) 函数对呼叫的进展情况进行查询。由于呼叫进展必然体现到通道的状态迁移上，因此，也可以通过 [E CHG ChState](#) 事件或 [SsmGetChState](#) 函数获得通道的状态迁移信息，来获取 AutoDial 任务的进展情况。

➤ ISUP 通道

驱动程序使用 IAM 消息发起去话呼叫。装配一条 IAM 消息的方式有 2 种：

(1) 由驱动程序自动装配。IAM 消息中的部分字段可以通过下列方式进行定制：

设置字段	方法
连接性质表示语 ^[1]	修改配置项 DefaultNatureOfConnectionInd 的设置
前向呼叫表示语 ^[1]	修改配置项 DefaultIAM_FowardCallInd 的设置
主叫用户类别 ^[1]	修改配置项 DefaultIAM_CAT 的设置，或调用 SsmSetISUPCAT 函数
传输媒介要求表示语 ^[1]	修改配置项 DefaultIAM_TransmissionMediumRequirement 的设置
被叫用户号码 ^[2]	字段中的第 1、第 2 个 8 位位组可以通过配置项 DefaultIAM_CalleeParam 函数 SsmSetIsupFlag （携带 ISUP_PhoNumParam 参数）设置；后续的 8 位位组（即号码部分）通过本函数的参数 szPhoNum 设置
主叫用户号码 ^[3]	字段中的第 1、第 2 个 8 位位组可以通过配置项 DefaultIAM_CallerParam 或函数 SsmSetIsupFlag （携带 ISUP_CallerParam 参数）设置；后续的 8 位位组（即号码部分）可以通过配置项 CalloutCallerId 或函数 SsmSetTxCallerId 设置。配置项 SetSTSsignal 用于设置主叫用户号码中是否需要包含 ST 信号
原被叫号码 ^[3]	字段中的第 1、第 2 个 8 位位组可以通过配置项 DefaultIAM_OriginalCalleeParam 设置；后续的 8 位位组（即号码部分）可以通过函数 SsmSetTxOriginalCallerID 设置
用户业务信息 ^[3]	配置项 bSubscriberSI 设置是否包含此字段，配置项 SubscriberSI 设置此字段的值
任选前向呼叫表示语 ^[3]	配置项 bOptionalFCI 设置是否包含此字段，配置项 OptionalFCI 设置此字段的值
用户到用户信息 ^[3]	修改配置项 Usr2UsrInfo 的设置，或者调用函数 SsmSetIsupParameter
其余可选参数 ^[3]	调用函数 SsmSetIsupParameter

注：表中的 ISUP_CallerParam 和 ISUP_PhoNumParam 在头文件 Shpa3api.h 中声明，^[1]表示长度固定的必备字段，^[2]表示长度可变的必备字段，^[3]表示可选字段。

(2) 由应用程序自行构建 IAM 消息。在调用本函数前，应用程序可以自行构建一个 IAM 消息，然后调用函数 [SsmSetIsupUPPara](#) 将 IAM 消息提交给驱动程序。驱动程序在执行本函数时，会自动覆盖消息中的 SIO、DPC、OPC、SLC、CIC 字段，以及被叫用户号码字段。

✧ TUP 通道

TUP 通道可以使用 IAI 消息或 IAM 消息作为初始地址消息发起去话呼叫。有关 IAM/IAI 消息的函数和配置项如下表所示：

消息类型	相关函数或配置项	描述
IAM	配置项： ConnectReqMsg	选择使用 IAM 还是 IAI 消息
IAI	配置项： CalloutIAM_CAT	设置主叫用户类别

IAI	配置项: CalloutAM_MsgPntr	设置消息表示语字段
	函数: SsmSetTxOriginalCallerID	设置原主叫地址
	配置项: OriginalCalleeAddrInd	设置原被叫地址表示语
	配置项: CalloutCallerId	设置主叫用户号码
	函数: SsmSetTxCallerId	
	配置项: CallingIndicatorBit	设置第 1 表示语八位位组中的附加主叫用户信息表示语和主叫用户线标识表示语
	配置项: SetSTSsignal	设置主叫用户号码参数的结尾是否需要包含 ST 信号
	配置项: CallerAddrInd	设置主叫用户线标识字段中的地址表示语

如果应用程序在调用本函数前，调用了函数 [SsmSetTxOriginalCallerID](#) 来设置原被叫地址，则驱动程序只使用 IAI 消息发起去话呼叫。IAI 消息中原被叫地址字段中的地址表示语可以通过配置项 [OriginalCalleeAddrInd](#) 进行设置。

◆ ISDN 通道

对于 ISDN 协议，本函数会引发驱动程序使用 SETUP 消息发起去话呼叫。有关 SETUP 消息的函数和配置项如下表所示：

消息类型	相关函数或配置项	描述
SETUP	配置项: UserCalledNoSet , NetCalledNoSet	设置被叫用户号码类型及编号方案
	配置项: UserCallingNoSet , NetCallingNoSet	设置主叫用户号码类型及编号方案
	配置项: UserNumIsFull , NetNumIsFull	设置是否在 SETUP 消息中包含“被叫号码全”参数
	配置项: UserChIdentify , NetChIdentify	设置通路识别表示方法
	配置项: UserChPreference , NetChPreference	设置是否需要优选通路类型
	配置项: CalloutCallerId	设置主叫用户号码（应在调用本函数前调用）
	函数: SsmSetTxCallerId	
	配置项: UserHighLayerCompatible , NetHighLayerCompatible	设置 SETUP 消息中是否包含“高层兼容性”字段
	配置项: UserLowLayerCompatible , NetLowLayerCompatible	设置 SETUP 消息中是否包含“低层兼容性”字段
	函数: SsmISDNSetTxSubAddr	设置主叫用户子地址（应在调用本函数前调用）
	函数: SsmISDNSetDialSubAddr	设置被叫方子地址（应在调用本函数前调用）
	函数: SsmSetTxOriginalCallerID	设置原主叫号码
	函数: SsmISDNSetCallerIdPresent	设置主叫号码是否允许显示字段

◆ SS1 通道

对于 SS1 通道，驱动程序在执行 AutoDial 任务时，应用程序可以通过函数或配置项对去话呼叫过程进行局部控制，如下表所示：

相关函数或配置项	描述
函数: SsmSetKA	设置“主叫用户类别”（即 KA 信号）
配置项: MfcKA	
函数: SsmSetKD	设置“发端业务性质”（即 KD 信号）
配置项: MfcKD	
配置项: CalloutCallerId	设置主叫用户（本端）号码
函数: SsmSetTxCallerId	

◆ IP 通道

对于 IP 通道，本函数会引发驱动程序使用协议指定的消息发起去话呼叫。具体来说，SIP 类型使用 INVITE 消息。有关这些消息的函数如下表所示：

消息类型	相关函数	描述
INVITE(SIP)	函数: SsmSetTxCallerId	设置主叫用户号码（应在调用本函数前调用）

函数执行结果可使用 [SsmChkAutoDial](#) 得到。若遇特殊情况，IP 通道拨号动作失败且调用 [SsmChkAutoDial](#) 总

是返回 11，用户可调用 [SsmGetAutoDialFailureReason](#) 获得具体失败原因。

注意事项：

- 本函数只支持 TUP 通道、ISUP 通道、SS1 通道、ISDN 通道、模拟中继线通道、IP 通道；
- TUP 通道：最大号码长度为 30；
- ISUP 通道：最大号码长度为 49；
- SS1 通道：最大号码长度为 50；
- ISDN 通道：最大号码长度为 30；
- IP 通道：最大号码长度为 128。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmAppendPhoNum](#), [SsmChkAutoDial](#), [SsmGetChState](#), [SsmGetPendingReason](#)

2.3.1.2.3 SsmAppendPhoNum

向 AutoDial 任务追加被叫方的电话号码。

函数原型：

```
int SsmAppendPhoNum(int ch, LPSTR szPhoNum)
```

参数说明：

ch	通道编号
szPhoNum	指向存放电话号码字符串的缓冲区指针。如果本通道是模拟中继线通道，则如果 szPhoNum 中的字符既不是标准的 DTMF 字符，也不是指定的闪断字符，该字符将被视为一个延时字符；如果是 SS1 的出中继通道，则 szPhoNum 中的任何非数字字符将被驱动程序忽略。拨号字符串的最大长度不得超过 50 个字节

返回值：

-1	调用失败。失败原因可以通过调用函数 SsmGetLastErrMsg 获得
0	调用成功

功能描述：

向 AutoDial 任务追加被叫方的电话号码。详细内容请参见相关通道的状态迁移图部分内容。

注意事项：

- 本函数只支持 TUP 通道、ISUP 通道、SS1 通道、ISDN 通道、模拟中继线通道；
- TUP 通道：追加的被叫方电话号码长度最大为 16；
- ISUP 通道：追加的被叫方电话号码长度最大为 50；
- SS1 通道：追加的被叫方电话号码长度最大为 50；
- ISDN 通道：追加的被叫方电话号码长度最大为 20。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmAutoDial](#), [SsmAutoDialEx](#)

2.3.1.2.4 SsmChkAutoDial

查询 AutoDial 任务的执行进展情况。

函数原型:

```
int SsmChkAutoDial(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1 表示调用失败，其它返回值的含义如下表所示：

返回值	shpa3api.h 中声明的宏	描述
0	DIAL_STANDBY	通道空闲，没有执行 AutoDial 任务
1	DIAL_DIALING	正在发送被叫号码
2	DIAL_ECHOTONE	回铃。 ◆ 模拟中继线通道：拨号完成后，线路上检测到了回铃音 ◆ SS1 通道：表明去话呼叫时，驱动程序收到对端交换机的后向 KB=1 或 KB=6 信号，表明被叫空闲 ◆ ISDN 通道：表明驱动程序收到对端交换机的提醒消息（ALERTING） ◆ TUP/ISUP 通道：表明驱动程序收到对端交换机的地址齐消息(ACM)
3	DIAL_NO_DIALTONE	没有在线路上检测到拨号音，AutoDial 任务失败。只适用于模拟中继线通道
4	DIAL_BUSYTONE	被叫用户忙，AutoDial 任务失败。 对于模拟中继线通道，表明线路上检测到了忙音信号
5	DIAL_ECHO_NOVOICE	拨号完成后，线路上先是出现了回铃音，然后保持静默。AutoDial 任务终止。只适用于模拟中继线通道
6	DIAL_NOVOICE	拨号完成后，线路上没有检测到回铃音，一直保持静默。AutoDial 任务终止。只适用于模拟中继线通道
7	DIAL_VOICE	被叫用户摘机，AutoDial 任务完成
8	DIAL_VOICEF1	被叫用户摘机（检测到 F1 频率的应答信号），AutoDial 任务完成。只适用于模拟中继线通道
9	DIAL_VOICEF2	被叫用户摘机（检测到 F2 频率的应答信号），AutoDial 任务完成。只适用于模拟中继线通道
10	DIAL_NOANSWER	被叫用户在指定时间内没有摘机，AutoDial 失败。不适用于 IP 卡 SIP 通道
11	DIAL_FAILURE	AutoDial 任务失败。失败原因可以通过函数 SsmGetAutoDialFailureReason 获得
12	DIAL_INVALID_PHONUM	被叫用户号码为空号，AutoDial 任务失败。不适用于 ISDN 通道和 IP 卡 SIP 通道
13	DIAL_SESSION_PROCESSING	IP 卡 SIP 通道收到 18X 消息(180 除外)
14	DIAL_ISDN_PROGRESS	ISDN 通道收到对端交换机的 PROGRESS 消息。详细的消息内容可以通过函数 SsmISDNGetProgressMsg 获得

功能描述:

查询 AutoDial 任务的执行进展情况。

注意事项:

- 坐席通道、磁石通道和录音通道不支持本操作；
- 如果 IP 通道和 ISDN 通道拨号动作失败，且调用本函数总是获得 11，用户可调用 [SsmGetAutoDialFailureReason](#) 获得具体失败原因。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmAutoDial](#), [SsmAutoDialEx](#)

2.3.1.2.5 SsmGetAutoDialFailureReason

获取去话呼叫失败的具体原因。

函数原型:

```
int SsmGetAutoDialFailureReason(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

返回值	宏	描述
-1	-1	操作失败
0	ATDL_NULL	没有呼出任务
1	ATDL_Cancel	AutoDial 任务被应用程序取消
2	ATDL_WaitDialAnsTimeout	等待被叫应答超时（针对 TUP 协议）
3	ATDL_WaitRemotePickupTimeout	等待被叫摘机超时
10	ATDL_Mtp3Unusable	SS7 信令：信令不可用
11	ATDL_RcvSSB	SS7 信令：收到对端交换机的 SSB 消息
12	ATDL_RcvSLB	SS7 信令：收到对端交换机的 SLB 消息
13	ATDL_RcvSTB	SS7 信令：收到对端交换机的 STB 消息
14	ATDL_RcvUNN	SS7 信令：收到对端交换机的 UNN 消息
15	ATDL_RcvSEC	SS7 信令：收到对端交换机的 SEC 消息
16	ATDL_RcvCGC	SS7 信令：收到对端交换机的 CGC 消息
17	ATDL_RcvNNC	SS7 信令：收到对端交换机的 NNC 消息
18	ATDL_RcvCFL	SS7 信令：收到对端交换机的 CFL 消息
19	ATDL_RcvLOS	SS7 信令：收到对端交换机的 LOS 消息
20	ATDL_RcvSST	SS7 信令：收到对端交换机的 SST 消息
21	ATDL_RcvACB	SS7 信令：收到对端交换机的 ACB 消息
22	ATDL_RcvDPN	SS7 信令：收到对端交换机的 DPN 消息
23	ATDL_RcvEUM	SS7 信令：收到对端交换机的 EUM 消息
24	ATDL_RcvADI	SS7 信令：收到对端交换机的 ADI 消息
25	ATDL_RcvBLO	SS7 信令：收到对端交换机的 BLO 消息
26	ATDL_DoubleOccupy	SS7 信令：检出同抢
27	ATDL_CircuitReset	SS7 信令：收到对端交换机的电路/群复原信号
28	ATDL_BlockedByRemote	SS7 信令：电路被对端交换机闭塞
40	ATDL_SS1WaitOccupyAckTimeout	No.1 信令：等待占用应答信号超时
41	ATDL_SS1RcvCAS_HANGUP	No.1 信令：收到后向拆线信号
42	ATDL_SS1RcvA4	No.1 信令：收到 A4 信号（机键拥塞）
43	ATDL_SS1RcvA5	No.1 信令：收到 A5 信号（空号）
44	ATDL_SS1RcvUndefinedAx	No.1 信令：收到未定义后向 A 信号

45	ATDL_SS1RcvUndefinedAxOnTxCallerId	No.1 信令：送主叫收到未定义 A
46	ATDL_SS1WaitAxTimeout	No.1 信令：等候接收后向 A 组信号超时
47	ATDL_SS1WaitAxStopTimeout	No.1 信令：等候后向 A 组信号停发超时
48	ATDL_SS1WaitAxTimeoutOnTxCallerId	No.1 信令：送主叫时等候 A 信号超时
49	ATDL_SS1WaitAxStopTimeoutOnTxCallerId	No.1 信令：送主叫 A 信号停发超时
50	ATDL_SS1RcvKB2	No.1 信令：收到 KB2 信号（用户“市忙”）
51	ATDL_SS1RcvKB3	No.1 信令：收到 KB3 信号（用户“长忙”）
52	ATDL_SS1RcvKB4	No.1 信令：收到 KB4 信号（机键拥塞）
53	ATDL_SS1RcvKB5	No.1 信令：收到 KB5 信号（空号）
54	ATDL_SS1RcvUndefinedKB	No.1 信令：收到未定义的 KB 信号
55	ATDL_SS1WaitKBTimeout	No.1 信令：接收后向 KB 信号超时
56	ATDL_SS1WaitKBStopTimeout	No.1 信令：等候对方方停发 KB 信号超时
60	ATDL_ISDNINETISBUS	ISDN：网络忙（不再使用）
61	ATDL_ISDNEMPTYNO	ISDN：空号
65	ATDL_IllegalMessage	SS7 信令：收到对端交换机的非法消息
66	ATDL_RcvREL	ISUP：收到对端交换机的释放消息（REL）
67	ATDL_RcvCBK	TUP：收到对端交换机的 CBK 消息
68	ATDL_IPInvalidPhonum	IP：所拨号码无效
69	ATDL_IPRemoteBusy	IP：对端忙
70	ATDL_IPBeenRefused	IP：被拒绝
71	ATDL_IPDnsFail	IP：DNS 无效
72	ATDL_IPCodecUnSupport	IP：不支持的 CODEC 类型
73	ATDL_IPOutOfResources	IP：没有可用的资源
74	ATDL_IPLocalNetworkErr	IP：本端网络出现错误
75	ATDL_IPRemoteNetworkErr	IP：远端网络出现错误
4xx、5xx、 6xx		IP：远端回复的 SIP 信令值
100+n		其他原因，n 为 ISDN 协议抛出的原因号码（具体原因可查询 附录 1 ISDN 通道释放原因值表 ）
其它		保留

功能描述：

获取去话呼叫失败的具体原因。

注意事项：

- SynCTI 驱动对宏 ATDL_IPRemoteBusy、ATDL_IPDnsFail、ATDL_IPCodecUnSupport、ATDL_IPLocalNetworkErr、及 ATDL_IPRemoteNetworkErr 暂时不支持的。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmAutoDial](#), [SsmAutoDialEx](#)

2.3.1.2.6 SsmSetTxCallerId

设置去话呼叫时的本端电话号码，即主叫方电话号码。

函数原型：

```
int SsmSetTxCallerId(int ch, LPSTR pszTxCallerId)
```

参数说明：

ch	通道编号
----	------

pszTxCallerId

- 对于 SIP 类型的 IP 通道, pszTxCallerId 设置的是完整 SIP URI 的 displayname 部份, 最大长度不能超过 511。一个完整的 SIP URI 格式为: "displayname" <sip:user@host:port>;
- 对于其他类型的通道, 该参数存放主叫方电话号码的 ASCII 码形式字符串, TUP 通道长度最大为 15 个字符, ISDN 通道长度最大为 20 个字符, 其他通道的最大长度不能超过 50 个字符。号码字符必须为数字字符'0'~'9'中的一个, 其它类型的字符将被忽略;
- 调用函数时, 如将参数 pszTxCallerId 设置为空字符串 ("") 时, 可以清空保存在内部缓冲区中的数据。

返回值:

-1	调用失败
≥0	实际发送的主叫号码的个数

功能描述:

设置去话呼叫时本端的电话号码, 即主叫方电话号码。

在去话呼叫过程中, 如果对端交换机要求本端发送主叫方电话号码, 驱动程序会自动将 pszTxCallerId 中的字符发送给对端交换机。

注意事项:

- 本函数必须在函数 [SsmAutoDial](#) 或 [SsmAutoDialEx](#) 之前调用。参数的值一旦设置后, 会被驱动程序保存在内部缓冲区中, 并会在后续的 IAM 消息中继续使用。再次调用本函数并将 pszTxCallerId 置为空字符串, 表示取消设置, 发送的主叫号码为空;
- SS1 通道的配置项 [TxCallerId](#) 可以实现相同功能, 缺省值为空字符串;
- 本函数只支持 SS1 通道、TUP 通道、ISUP 通道、ISDN 通道、IP 通道;
- 对于 TUP 协议和 ISUP 协议, 某些交换机要求主叫方发送的主叫号码字符串应包含 ST 信号 (脉冲终了信号)。配置项 [SetSTSsignal](#) 可以用来设置此项功能;
- 函数 [SsmSetTxCallerId](#) 调用失败, 会导致函数所设置的主叫号码发送不成功, [SsmAutoDial](#) 会按照配置文件 shconfig.ini 中配置项 [CalloutCallerId](#) 的定义来发送主叫号码, 如果在配置项 [CalloutCallerId](#) 设置有号码的情况下, 发送该配置项设置的号码给被叫作为主叫号码。在配置项中没有设置主叫号码时, 发送的主叫号码为空。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetPhoNumLen](#), [SsmAutoDial](#), [SsmAutoDialEx](#)

2.3.1.2.7 SsmGetTxCallerId

获取预设的本端的主叫方电话号码。

函数原型:

```
int SsmGetTxCallerId(int ch, LPSTR pszTxCallerId)
```

参数说明:

ch	通道编号
pszTxCallerId	<ul style="list-style-type: none">对于 SIP 类型的 IP 通道, 返回通过 SsmSetTxCallerId 设定的字符串其他类型的通道, 返回主叫号码字符串

返回值:

-1	调用失败
≥0	pszTxCallerId 中字符的个数

功能描述:

获取预设的本端的主叫方电话号码。

注意事项:

- 本函数只支持 SS1 通道、TUP 通道、ISUP 通道、ISDN 通道、IP 通道。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetTxCallerId](#)

2.3.1.2.8 SsmSetTxOriginalCallerID

设置呼叫建立消息中的改发号码或原被叫号码。

函数原型:

```
int SsmSetTxOriginalCallerID(int ch, BYTE* pszTxCallerId)
```

参数说明:

ch	通道编号
pszTxCallerId	存放原被叫号码（对于 ISUP 协议或 TUP 协议）或改发号码（对于 ISDN 协议）的字符串指针，字符总数不能超过 16 个

返回值:

-1	调用失败
>=0	pszTxCallerId 指示的号码实际长度

功能描述:

去话呼叫时，设置向对端交换机发送的呼叫建立消息中的改发号码（对于 ISDN 协议）或原被叫号码（对于 ISUP 协议或 TUP 协议）。

注意事项:

- 本函数必须在函数 [SsmAutoDial](#) 或 [SsmAutoDialEx](#) 之前调用，当驱动程序完成去话呼叫建立消息的装配后，会将此缓冲区清空；
- 对于 TUP 通道，如果应用程序在调用 [SsmAutoDial](#) 或 [SsmAutoDialEx](#) 前调用了本函数，则驱动程序只使用 IAI 消息发起去话呼叫。本函数设置 IAI 消息中原被叫地址字段中原被叫地址，地址表示语可以通过配置项 [OriginalCalleeAddrInd](#) 进行设置；
- 本函数只支持 TUP 通道、ISUP 通道、ISDN 通道。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmAutoDial](#), [SsmAutoDialEx](#)

2.3.1.2.9 SsmSetTxRedirectingNum

设置呼叫建立消息中改发的号码。

函数原型:

```
int SsmSetTxRedirectingNum(int ch, BYTE* pszTxredirectingNum)
```

参数说明:

ch	通道编号
pszTxredirectingNum	存放改发的号码的字符串指针, 字符总数不能超过 16 个

返回值:

-1	调用失败
≥0	pszTxredirectingNum 指示的号码实际长度

功能描述:

去话呼叫时, 设置向对端交换机发送的改发的号码。

注意事项:

- 本函数必须在函数 [SsmAutoDial](#) 或 [SsmAutoDialEx](#) 之前调用;
- 参数的值一旦设置后, 会被驱动程序保存在内部缓冲区中, 并会在后续的 IAM 消息中继续使用;
- 本函数设置的参数的有效期直至重启程序, 重新加载配置文件为止;
- 取消方法: 将函数的第二个参数设置为空字符串即可。SsmSetTxRedirectingNum(CurCh, (LPBYTE)""), 只能取消由本函数设置的号码;
- 对于 ISUP 通道, 如果应用程序在调用 [SsmAutoDial](#) 或 [SsmAutoDialEx](#) 前调用了本函数, 则驱动程序只使用 IAM 消息发起去话呼叫。本函数设置 IAM 消息中改发的号码, 地址表示语等可以通过配置项 [DefaultIAM_RedirectingNumber](#) 进行设置;
- 本函数目前只支持 ISUP 通道。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.1.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmAutoDial](#), [SsmAutoDialEx](#)

2.3.1.2.10 SsmSetWaitAutoDialAnswerTime

设置等待被叫用户摘机的最大等待时间。

函数原型:

```
BOOL SsmSetWaitAutoDialAnswerTime(WORD wSeconds)
```

参数说明:

wSeconds	等待时间, 单位秒
----------	-----------

返回值:

FALSE	调用失败
TRUE	调用成功

功能描述:

去话呼叫时，设置等待被叫用户摘机的最大等待时间。

注意事项:

- 配置项 [MaxWaitAutoDialAnswerTime](#) 可以实现相同功能；
- 本函数只支持 SS1 通道、TUP 通道、ISUP 通道、模拟外线通道、ISDN 通道。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetWaitAutoDialAnswerTime](#)

2.3.1.2.11 SsmSetWaitAutoDialAnswerTimeEx

设置一个通道等待被叫用户摘机的最大等待时间。

函数原型:

Int WINAPI SsmSetWaitAutoDialAnswerTimeEx(int ch, int nSeconds)

参数说明:

ch	通道编号
nSeconds	等待时间，单位秒

返回值:

-1	调用失败
0	调用成功

功能描述:

去话呼叫时，设置等待被叫用户摘机的最大等待时间。

注意事项:

- 参数 nSeconds 必须大于等于 0，小于 0 则提示错误。如果等于 0，则为配置项 [MaxWaitAutoDialAnswerTime](#) 或者函数 [SsmSetWaitAutoDialAnswerTime](#) 起作用；大于 0，则函数 [SsmSetWaitAutoDialAnswerTimeEx](#) 有效。
- 本函数支持 SS1、TUP、ISUP、ISDN、IP、模拟中继通道设定，必须每次在 [SsmAutoDial](#) 之前调用才有效。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.1.1 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetWaitAutoDialAnswerTime](#)

2.3.1.2.12 SsmGetWaitAutoDialAnswerTime

获取预设的等待被叫用户摘机的最大等待时间。

函数原型:

```
BOOL SsmGetWaitAutoDialAnswerTime(WORD* pwSeconds)
```

参数说明:

pwSeconds	返回等待时间，单位秒
-----------	------------

返回值:

FALSE	调用失败
TRUE	调用成功

功能描述:

去话呼叫时，获取预设的等待被叫用户摘机的最大等待时间。

注意事项:

- 本函数只支持 SS1 通道、TUP 通道、ISUP 通道、ISDN 通道。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetWaitAutoDialAnswerTime](#)

2.3.1.2.13 SsmGetKB

获取被叫用户状态。

函数原型:

```
int SsmGetKB(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败 被叫用户状态，具体含义与通道类型及呼叫方向有关： ➤ 去话呼叫： ◆ SS1 通道：返回对端发送本端的 KB 信号值，具体取值及含义请参见函数 SsmSetKB 的说明 ◆ TUP 通道：返回从对端交换机收到的 ACM 消息中消息表示语字段，共 8 个比特，具体含义请参见 TUP 协议的相关文档 ◆ ISUP 通道：返回从对端交换机收到的 ACM 消息中后向呼叫表示语字段(共 2 个字节)中的低 8 位比特，具体含义请参见 ISUP 协议的相关文档 ◆ ISDN 通道：不支持 ➤ 来话呼叫：返回应用程序调用函数 SsmSetKB 函数时的 KB 信号参数值，其取值及含义请参见函数 SsmSetKB 的说明
≥0	

功能描述:

获取被叫用户状态。

注意事项:

- 对于去话呼叫，建议在通道迁移到 S_CALL_WAIT_REMOTE_PICKUP 状态时调用本函数。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetKB](#)

2.3.1.3 SS1 通道的专用函数

2.3.1.3.1 SsmSetKA

设置去话呼叫时的“主叫用户类别”信号。

函数原型:

```
int SsmSetKA(int ch, BYTE btSigKA)
```

参数说明:

ch	通道编号	KA 信号值, 取值范围: 1≤btSigKA≤10, 具体如下表所示:												
btSigKA		KA 编码	KA 信号内容 (包括 KOA)											
			步进制市话局		纵横制、程控市话局 (也包括准电子)		KA		KA		KOA			
		1	普	定期	普	定期	普	定期	普	定期	普	定期		
		2	用户表, 立即		用户表, 立即		用户表, 立即		用户表, 立即		用户表, 立即			
		3	通	打印机, 立即	通	打印机, 立即	通	打印机, 立即	通	打印机, 立即	通	打印机, 立即		
		4	备用		备用		备用		备用		备用			
		5	普通免费		普通免费		普通免费		普通免费		普通免费			
		6	备用		备用		备用		备用		备用			
		7	备用		备用		备用		备用		备用			
		8	备用		优先, 定期		优先, 定期		优先, 定期		优先, 定期			
		9	(郊话自动有权、长途自动无权)		备用		备用		备用		备用			
		10	(长郊自动无权)		优先, 免费		优先, 免费		优先, 免费		优先, 免费			
		11			备用									
		12												
		13			计划用于测试									
		14			备用									
		15			备用									

返回值:

-1	调用失败
0	调用成功

功能描述:

去话呼叫时，设置 R2 信令中的 KA 信号值（主叫用户类别）。更多信息请参见第 1 章中“[中国 1 号信令的状态转移](#)”部分内容。

注意事项:

- 本函数只支持 SHD 系列板卡的 SS1 通道；
- 本函数应该在 [SsmAutoDial](#) 或 [SsmAutoDialEx](#) 函数之前调用；

- 配置项 [MfcKA](#) 可以实现相同功能。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetKA](#)

2.3.1.3.2 SsmSetKD

去话呼叫时，设置 KD 信号值。

函数原型:

```
int SsmSetKD(int ch, BYTE btSigKD)
```

参数说明:

ch	通道编号														
KD 信号的编码值，取值范围如下表所示：															
btSigKD	<table border="1"><thead><tr><th>KD 值</th><th>描述</th></tr></thead><tbody><tr><td>1</td><td>长途话务员半自动呼叫</td></tr><tr><td>2</td><td>长途自动呼叫（电话通信或用户传真、用户数据通信）</td></tr><tr><td>3</td><td>市内电话（用户呼叫半自动话务员及国际半自动话务员）</td></tr><tr><td>4</td><td>市内用户传真或用户数据通信及优先用户</td></tr><tr><td>5</td><td>半自动核对主叫号码口</td></tr><tr><td>6</td><td>测试呼叫</td></tr></tbody></table>	KD 值	描述	1	长途话务员半自动呼叫	2	长途自动呼叫（电话通信或用户传真、用户数据通信）	3	市内电话（用户呼叫半自动话务员及国际半自动话务员）	4	市内用户传真或用户数据通信及优先用户	5	半自动核对主叫号码口	6	测试呼叫
KD 值	描述														
1	长途话务员半自动呼叫														
2	长途自动呼叫（电话通信或用户传真、用户数据通信）														
3	市内电话（用户呼叫半自动话务员及国际半自动话务员）														
4	市内用户传真或用户数据通信及优先用户														
5	半自动核对主叫号码口														
6	测试呼叫														

返回值:

-1	调用失败
0	调用成功

功能描述:

去话呼叫时，设置 KD 信号值（发端业务性质）。更多信息请参见第 1 章中“[中国 1 号信令的状态转移](#)”部分内容。

注意事项:

- 本函数只支持 SHD 系列板卡的 SS1 通道；
- 本函数应该在 [SsmAutoDial](#) 或 [SsmAutoDialEx](#) 函数之前调用；
- 配置项 [MfcKD](#) 可以实现相同功能。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetKD](#)

2.3.2 有关来话呼叫的函数

2.3.2.1 获取主叫用户信息

2.3.2.1.1 SsmChkOpCallerId

取得指定通道是否支持接收主叫号码信息的操作。

函数原型:

```
int SsmChkOpCallerId (int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
0	指定通道不支持接收主叫信息操作
1	指定通道支持接收主叫信息操作

功能描述:

取得指定通道是否支持接收主叫号码信息的操作。

注意事项:

- 坐席通道、磁石通道不支持本操作。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetCallerId](#)

2.3.2.1.2 SsmGetCallerId

请参见 [SsmGetCallerIdEx](#)

2.3.2.1.3 SsmGetCallerIdA

请参见 [SsmGetCallerIdEx](#)

2.3.2.1.4 SsmGetCallerIdEx

获取来话呼叫的主叫方号码信息。`SsmGetCallerId`、`SsmGetCallerIdA` 用来获取保存在基本缓冲区中的主叫号码信息；`SsmGetCallerIdEx` 用来获取保存在扩展缓冲区中的主叫号码信息，只适用于模拟中继线通道和模拟中继线录音通道。

函数原型:

```
int SsmGetCallerId(int ch, LPSTR szCallerId)
int SsmGetCallerIdEx(int ch, LPSTR szCallerIdEx)
char* SsmGetCallerIdA(int ch)
```

参数说明:

ch	通道编号
szCallerId	存放主叫号码基本信息的字符串指针, 由应用程序分配存储空间, 大小不小于 255 字节
szCallerIdEx	存放主叫号码扩展信息的字符串指针, 由应用程序分配存储空间, 大小不小于 255 字节 注意: 对于模拟中继线通道和模拟中继线录音通道, szCallerIdEx 中存放的是 FSK 制式的主叫号码的原始数据, 详细内容请参见第 1 章中“ 模拟电话线上的主叫号码 ”部分

返回值:

函数名称	返回值	描述
SsmGetCallerId	-1	调用失败
SsmGetCallerIdEx	≥0	主叫号码信息的长度
SsmGetCallerIdA	NULL	调用失败
	其他	驱动程序内部保存主叫号码基本信息的字符串指针

功能描述:

来话呼叫时, 获取保存在内部缓冲区中的主叫方号码信息。

对于 SS1 通道, 如果配置项 [MfcR2ToRxCallerIdBuf](#) 设置为 1, SsmGetCallerIdEx 可以取出存放在驱动程序内部的主叫号码扩展信息缓冲区中的对端交换机送出的 R2 信号, 更多信息请参见第 1 章中“[中国 1 号信令的状态机描述](#)”部分内容。

➤ IP 通道:

- 1) 对 SIP 类型的 IP 通道调用本函数, 获得的主叫号码是 SIP URI (SIP 消息中的 From 字段), 其格式为:
[sip:sips:]user[@host[:port]]。

注意事项:

- 坐席通道、磁石通道不支持本操作。

相关信息:

驱动程序版本要求	SynCTI Ver.4.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmClearCallerId](#), [SsmClearCallerIdEx](#)

2.3.2.1.5 SsmGetCallerName

获取模拟电话线上 Fsk 主叫信息中的姓名字符串。

函数原型:

```
int WINAPI SsmGetCallerName(int ch, LPSTR pszCallerName)
```

参数说明:

ch	通道编号
pszCallerName	存放姓名字符串的缓冲区指针, 由应用程序分配存储空间, 不能低于 50 个字符

返回值:

-1	调用失败
≥0	返回 pszCallerName 中字符的个数

功能描述:

获取模拟电话线上 Fsk 主叫信息中的姓名字符串。更多信息请参见第 1 章中“[模拟电话线上的主叫号码](#)”部分内容。

注意事项:

- 本函数只支持 SHT 系列板卡的模拟中继线通道和 ATP 系列板卡的模拟电话线录音通道。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetCallerId](#), [SsmGetCallerIdA](#), [SsmGetCallerIdEx](#)

2.3.2.1.6 SsmClearCallerId

请参见 [SsmClearCallerIdEx](#)

2.3.2.1.7 SsmClearCallerIdEx

清空驱动程序内部保存主叫号码信息的缓冲区。

函数原型:

```
int SsmClearCallerId (int ch)  
int SsmClearCallerIdEx(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
0	调用成功

功能描述:

清空驱动程序内部保存主叫号码信息的缓冲区。[SsmClearCallerId](#) 清空基本缓冲区, [SsmClearCallerIdEx](#) 清空扩展缓冲区。

注意事项:

- 本函数支持 SHD、ATP、DTP、DST 系列除 ISDN 和 IP 通道之外的全部通道, 以及 SHT 系列的模拟中继线通道。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib

DLL	shp_a3.dll
-----	------------

相关函数: [SsmGetCallerId](#), [SsmGetCallerIdA](#), [SsmGetCallerIdEx](#)

2.3.2.1.8 SsmGetKA

来话呼叫时，获取主叫用户类别。

函数原型：

```
int SsmGetKA(int ch)
```

参数说明：

ch	通道编号
----	------

返回值：

-1	调用失败 返回“主叫用户类别”，具体含义与通道类型有关： ◆ SS1 通道：如果本次呼叫为来话呼叫，此返回值即为 KA 信号值，具体含义请参见函数 SsmSetKA 的说明；如果为去话呼叫，此返回值表示在本次呼叫过程中，本端发送给对端的“主叫用户类别” ◆ ISUP 通道：在来话呼叫时，此返回值表示对端交换机的 IAM 消息中的“主叫方类别”参数值，其含义如下表所示：																
≥0	<table border="1"><thead><tr><th>返回值</th><th>注释</th></tr></thead><tbody><tr><td>9</td><td>国内话务员（国内用）</td></tr><tr><td>10</td><td>普通主叫用户，用于长途局和长途局，以及长途局和市话局之间</td></tr><tr><td>11</td><td>优先主叫用户</td></tr><tr><td>12</td><td>数据呼叫（音频带数据）</td></tr><tr><td>13</td><td>测试呼叫</td></tr><tr><td>14</td><td>付费电话</td></tr><tr><td>15</td><td>普通主叫用户，用于市话局之间</td></tr></tbody></table> ◆ TUP 通道：在来话呼叫时，此返回值表示对端交换机的 IAI/IAM 消息中的“主叫方类别”参数值	返回值	注释	9	国内话务员（国内用）	10	普通主叫用户，用于长途局和长途局，以及长途局和市话局之间	11	优先主叫用户	12	数据呼叫（音频带数据）	13	测试呼叫	14	付费电话	15	普通主叫用户，用于市话局之间
返回值	注释																
9	国内话务员（国内用）																
10	普通主叫用户，用于长途局和长途局，以及长途局和市话局之间																
11	优先主叫用户																
12	数据呼叫（音频带数据）																
13	测试呼叫																
14	付费电话																
15	普通主叫用户，用于市话局之间																

功能描述：

来话呼叫时，获取主叫用户类别。

注意事项：

- 本函数只适用于 SS1 通道、TUP 通道和 ISUP 通道；
- 建议应用程序在通道迁移到 S_CALL_RINGING 状态时调用本函数。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.1.5 或更高
Header	shp_a3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetKA](#), [SsmGetCallerId](#), [SsmGetChState](#)

2.3.2.1.9 SsmGetKD

来话呼叫时，获取发端业务性质。

函数原型：

```
int SsmGetKD(int ch)
```

参数说明：

ch	通道编号
----	------

返回值：

-1	调用失败
≥0	返回主叫方发送的 KD 信号值，具体取值及含义请参见函数 SsmSetKD 的说明

功能描述：

来话呼叫时，获取主叫方的 KD 信号值（即发端业务性质）。

注意事项：

- 本函数只适用于 SHD 系列板卡的 SS1 通道和 DTP 系列板卡的 SpyCic 中被叫方通道；
- 建议在通道迁移到 S_CALL_RINGING 状态时调用本函数。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数： [SsmGetChState](#)

2.3.2.2 获取被叫用户信息

2.3.2.2.1 SsmGetPhoNumStr

请参见 [SsmGetPhoNumLen](#)

2.3.2.2.2 SsmGetPhoNumStrA

请参见 [SsmGetPhoNumLen](#)

2.3.2.2.3 SsmGetPhoNumLen

来话呼叫时，获取被叫方号码信息。[SsmGetPhoNumLen](#) 获取号码的长度，[SsmGetPhoNumStr](#) 和 [SsmGetPhoNumStrA](#) 获取具体的号码值。

函数原型：

```
int SsmGetPhoNumLen(int ch)  
int SsmGetPhoNumStr(int ch, LPSTR pszPhoNum)
```

char* SsmGetPhoNumStrA(int ch)

参数说明:

ch	通道编号
pszPhoNum	存放 ASCII 形式的被叫号码的缓冲区指针, 由应用程序分配空间, 不小于 50 个字符

返回值:

函数名称	返回值	描述
SsmGetPhoNumLen	-1	调用失败
SsmGetPhoNumStr	≥0	被叫号码的长度
SsmGetPhoNumStrA	NULL	调用失败
	其他	返回驱动程序内部保存被叫号码的缓冲区指针

功能描述:

来话呼叫时, 对端交换机会向本端发送呼叫建立消息, 驱动程序收到此消息后, 会将其中的被叫号码分离出来, 保存到内部缓冲区中。本函数用于该缓冲区中的被叫号码信息。

➤ IP 通道

- 1) 对 SIP 类型的 IP 通道调用本函数, 获得的被叫号码是 SIP URI (SIP 消息中的 To 字段), 其格式为: ["displayname"] [sip:/sips:]user[@host[:port]]。

注意事项:

- [SsmGetPhoNumLen](#)、[SsmGetPhoNumStr](#) 和 [SsmGetPhoNumStrA](#) 函数只适用于 SS1 通道、TUP 通道、ISUP 通道、ISDN 通道、IP 通道上的来话呼叫;
- [SsmGetPhoNumStrA](#) 函数只适用于 SS1 通道、TUP 通道、ISUP 通道、ISDN 通道, 不支持 IP 通道上的来话呼叫;
- 驱动程序内部的被叫号码缓冲区长度为 50 个字符, 如果收到的被叫号码字符个数超过 50, 多余的号码将被丢弃。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关事件: [E_CHG_RxPhoNumBuf](#)

2.3.2.3 获取第 1 被叫号码信息

2.3.2.3.1 SsmGet1stPhoNumLen

请参见 [SsmGet1stPhoNumStrA](#)

2.3.2.3.2 SsmGet1stPhoNumStr

请参见 [SsmGet1stPhoNumStrA](#)

2.3.2.3.3 SsmGet1stPhoNumStrA

获取呼叫建立消息中的改发号码 (ISDN 通道)、原被叫号码 (TUP 通道) 或原被叫号码/改发号码 (ISUP 通道) 信息。SsmGet1stPhoNumLen 用于获取信息的长度，SsmGet1stPhoNumStr 和 SsmGet1stPhoNumStrA 用于获取信息本身。

函数原型：

```
int SsmGet1stPhoNumStr(int ch, LPSTR pszPhoNum)
char *SsmGet1stPhoNumStrA(int ch)
int SsmGet1stPhoNumLen(int ch)
```

参数说明：

ch	通道编号
pszPhoNum	存放号码字符串的缓冲区指针，由应用程序分配存储空间，不少于 50 个字符

返回值：

函数名称	返回值	描述
SsmGet1stPhoNumStr	-1	调用失败
SsmGet1stPhoNumLen	≥ 0	号码字符串的长度
SsmGet1stPhoNumStrA	NULL	调用失败
	其他	返回驱动程序内部保存号码字符串的缓冲区指针

功能描述：

来话呼叫时，获取呼叫建立消息中的被叫号码信息。

➤ **ISDN 通道**

获取 SETUP 消息中的改发号码 (Redirecting Number)。

➤ **TUP 通道**

获取 IAI 或 IAM 消息中的原被叫号码。

➤ **ISUP 通道**

获取 IAM 消息中的原被叫号码信息或改发号码信息。对于 ISUP 通道，本函数既可以返回 IAM 消息中的改发号码信息，也可以返回原被叫号码信息，具体是哪一个，取决于配置项 [SaveRGNTo1stPhoNumStr](#) 的设置：

- ◆ SaveRGNTo1stPhoNumStr 设置为 0：如果 IAM 消息中有原被叫号码字段，则返回原被叫号码信息；如果没有，则返回空；
- ◆ SaveRGNTo1stPhoNumStr 设置为 1：如果 IAM 消息中只有改发号码字段，则返回改发号码信息；如果 IAM 消息中只有原被叫号码字段，则返回原被叫号码信息；如果既有改发号码字段，又有原被叫号码字段，则返回排在消息后面的字段中的信息，比如，这 2 个字段在消息中的排列顺序为“改发号码字段|原被叫号码字段”，则返回原被叫号码信息；如果排列顺序为“原被叫号码字段|改发号码字段”，则返回改发号码信息。

如果需要单独获取 ISUP 通道的 IAM 消息中的改发号码信息或原被叫号码信息，请使用函数 [SsmGetIsupParameter](#)。

注意事项：

- 驱动程序内部的缓冲区长度为 50 个字符，当收到的号码数目超过该长度限制时，多余的号码将被丢弃。

相关信息：

驱动程序版本要求	SynCTI Ver.3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数： [SsmGetChState](#)

2.3.2.4 设置来话接续过程参数

2.3.2.4.1 SsmEnableAutoSendKB

设置“自动应答来话呼叫”功能。

函数原型:

```
int SsmEnableAutoSendKB(int ch, BOOL bEnable)
```

参数说明:

ch	通道编号
bEnable	=TRUE: 开启 =FALSE: 关闭

返回值:

-1	调用失败
0	调用成功

功能描述:

设置“自动应答来话呼叫”功能。

注意事项:

- 本函数只适用于 SHD 系列板卡;
- 对于 SS1 入中继通道, 本函数设置的参数也可以通过配置项 [AutoSendKB](#) 进行设置, 更多内容请参见第 1 章中 “[中国 1 号信令的状态转移](#)” 部分内容;
- 对于 ISUP / TUP 通道, 本函数设置的参数也可以通过配置项 [AutoSendACM](#) 进行设置, 更多内容请参见第 1 章中 “[ISUP 通道的状态转移](#)” 或 “[TUP 通道的状态转移](#)” 部分内容;
- 对于 ISDN 通道, 本函数设置的参数也可以通过配置项 [UserSideAutoSendAck](#) (用户侧) 或 [NetSideAutoSendAck](#) (网络侧) 进行设置, 更多信息请参见第 1 章中 “[ISDN 通道的状态转移图](#)” 部分内容。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetKB](#), [SsmGetAutoSendKBFlag](#)

2.3.2.4.2 SsmGetAutoSendKBFlag

获取“自动应答来话呼叫”功能的标志。

函数原型:

```
int SsmGetAutoSendKBFlag(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
0	关闭
1	开启

功能描述:

获取“自动应答来话呼叫”功能的标志。

注意事项:

- 本函数只适用于 SHD 系列板卡。

相关信息:

驱动程序版本要求	SynCTI Ver. 4.7.1.5 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmEnableAutoSendKB](#)

2.3.2.4.3 SsmSetKB

来话呼叫时，向对端交换机发送后向应答消息，设置被叫用户的状态。

函数原型:

```
int SsmSetKB(int ch, BYTE btSigKB)
```

参数说明:

ch	通道编号			
被叫状态指示值，具体设置方法与通道类型有关：				
➤ SS1 通道: btSigKB 表示向对端交换机发送的后向 KB 信号，其含义为：				
btSigKB	收到 KD=1、2 或 6	KB 信号内容		
1	被叫用户空闲	收到 KD=3 或 4 被叫用户空闲 互不控制复原		
2	被叫用户“市忙”	备用		
3	被叫用户“长忙”			
4	机键拥塞	被叫用户忙或机键拥塞		
5	被叫用户为空号	被叫用户为空号		
6	备用	被叫用户为空闲 主叫控制复原		
➤ TUP 通道: 本参数表示驱动程序发送的 TUP 消息，具体含义为：				
btSigKB	含义	驱动程序发送的消息		
1	被叫用户空闲计费	ACM		
2	被叫用户“市忙”	SLB		
3	被叫用户“长忙”	STB		
4	机键拥塞	SEC		
5	被叫用户为空号	UNN		
6	备用			
7	被叫用户空闲免费	ACM		
8	请求主叫号码信息	GRQ (状态会迁移到“等待 GSM”)		
9	交换机拥塞信号	SEC		
10	电路群拥塞信号	CGC		
11	地址不全信号	ADI		
12	呼叫故障信号	CFL		
13	线路不工作信号	LOS		
14	发送专用信息信号	SST		

	15	接入拒绝信号	ACB
	16	不提供数字通路信号	DPN
	17	忙	SSB
➤ ISUP 通道: 本参数表示驱动程序发送的 ISUP 消息, 具体含义为:			
	btSigKB	含义	驱动程序的操作
	1/6	被叫用户空闲, 计费	发送 ACM 消息, 接受来话呼叫
	2	被叫用户忙	发送 REL 消息, 拒绝来话呼叫
	3	地址不全	发送 REL 消息, 拒绝来话呼叫
	4	呼叫拒绝	发送 REL 消息, 拒绝来话呼叫
	5	无应答	发送 REL 消息, 拒绝来话呼叫
	7	被叫用户空闲, 免费	发送 ACM 消息, 接受来话呼叫
	8	请求主叫号码信息	发送 INR 消息, 状态迁移到“等待 INF”
	9	用户缺席	发送 REL 消息, 拒绝来话呼叫
➤ ISDN 通道: 本参数表示驱动程序发送的 ISDN 消息, 具体含义为:			
	btSigKB	含义	驱动程序动作
	1	被叫用户空闲	发送 ALERT 消息
	2	被叫用户忙	发送 DISCONNECT 消息, 原因=被叫用户忙
	4	呼叫拒绝	发送 DISCONNECT 消息, 原因=呼叫拒绝
	5	无应答	发送 DISCONNECT 消息, 原因=被叫用户无应答
	其他	备用	
➤ SIP 通道: 本参数表示驱动程序发送的 SIP 消息, 具体含义为:			
	btSigKB	含义	驱动程序动作
	1	被叫用户空闲	发送 180 消息
	2	被叫用户忙	发送 486 消息, 原因=被叫用户忙
	其他	备用	

返回值:

-1	调用失败 (注意: ISUP_CH 类型通道返回 -1 或 0 代表调用失败)
0	调用成功 (注意: ISUP_CH 类型通道返回 1 代表调用成功)

功能描述:

来话呼叫时, 当驱动程序按照预设的收号规则完成被叫号码等信息的接收后, 如果应用程序通过函数 [SsmEnableAutoSendKB](#) 或配置项 [AutoSendACM](#) 关闭了“自动应答来话呼叫”功能, 通道的状态会迁移到 S_CALL_PENDING, 将控制权交还给应用程序, 由应用程序向对端交换机发送后向应答消息, 设置被叫用户的状态。应用程序可以通过调用本函数来接受或拒绝本次来话呼叫。

更多信息请按照通道类型分别参见第 1 章中的相关内容:

- ◆ SS1 通道: “[中国 1 号信令的状态转移](#)”
- ◆ ISUP 通道: “[ISUP 通道的状态转移](#)”
- ◆ TUP 通道: “[TUP 通道的状态转移](#)”
- ◆ ISDN 通道: “[ISDN 通道的状态转移图](#)”
- ◆ SIP 通道: “[SIP 通道的状态转移图](#)”

注意事项:

- 对于 ISUP 通道, 如果参数 btSigKB 设置为 1、6 或 7, 驱动程序会向对端交换机应答 ACM 消息。这个 ACM 消息的构建方式为: 如果应用程序在调用本函数前调用了 [SsmSetIsupUPPara](#) 函数 (携带 C_ISUP_ACM 参数) 向驱动程序提交了一个自行构建的 ACM 消息, 使用该 ACM 消息; 否则, 使用驱动程序自动构建的 ACM 消息。更多信息请参见第 1 章中“[ISUP 通道的状态转移](#)”部分内容。
- 对于 ISDN 通道, [SsmSetCharge](#) 可以设置相应通道的通话是计费的还是免费的。

相关信息:

驱动程序版本要求	SynCTI Ver. 4.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmEnableAutoSendKB](#)

2.3.3 通道状态迁移事件

2.3.3.1 SsmGetChState

获取通道的当前状态。

函数原型:

```
int SsmGetChState(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

返回值-1 表示调用失败，其它返回值的含义如下表所示：

值	shpa3api.h 中的宏	状态描述
0	S_CALL_STANDBY S_FAX_Wait	空闲
1	S_CALL_PICKUPED	摘机
2	S_CALL_RINGING	振铃
3	S_CALL_TALKING	通话
4	S_CALL_ANALOG_WAITDIALTONE	模拟中继线通道：去话呼叫，等待拨号音
5	S_CALL_ANALOG_TXPHONUM	模拟中继线通道：去话呼叫，拨号
6	S_CALL_ANALOG_WAITDIALRESULT	模拟中继线通道：去话呼叫，等待拨号结果
7	S_CALL_PENDING	挂起状态。函数 SsmGetPendingReason 可以用来取得挂起原因
8	S_CALL_OFFLINE	线路断开状态
9	S_CALL_WAIT_REMOTE_PICKUP	去话呼叫，等候被叫用户摘机
10	S_CALL_ANALOG_CLEAR	模拟中继线通道：内部状态
11	S_CALL_UNAVAILABLE	通道不可用
12	S_CALL_LOCKED	呼出锁定
19	S_CALL_RemoteBlock	对端闭塞
20	S_CALL_LocalBlock	本地闭塞
30	S_CALL_Ss1InWaitPhoNum	SS1 通道：接收被叫号码
31	S_CALL_Ss1InWaitFwdStop	SS1 通道：等待对端交换机停发前向信号
32	S_CALL_Ss1InWaitCallerID	SS1 通道：接收 Caller ID
33	S_CALL_Ss1InWaitKD	SS1 通道：接收 KD 信号
34	S_CALL_Ss1InWaitKDStop	SS1 通道：等待对端交换机停发 KD 信号
35	S_CALL_SS1_SAYIDLE	SS1 通道：向对端交换机发送示闲信号
36	S_CALL_SS1WaitIdleCAS	SS1 通道：等待对端交换机的示闲信号
37	S_CALL_SS1PhoNumHoldup	SS1 通道：号码拦截状态
38	S_CALL_Ss1InWaitStopSendA3p	SS1 通道：等待对端交换机停发脉冲方式的 A3 信号
40	S_CALL_Ss1OutWaitBwdAck	SS1 通道：等待对端交换机应答占用证实信号
41	S_CALL_Ss1OutTxPhoNum	SS1 通道：发送被叫号码
42	S_CALL_Ss1OutWaitAppendPhoNum	SS1 通道：等待应用程序追加电话号码
43	S_CALL_Ss1OutTxCallerID	SS1 通道：发送主叫号码
44	S_CALL_Ss1OutWaitKB	SS1 通道：等待对端交换机的 KB 信号
45	S_CALL_Ss1OutDetectA3p	SS1 通道：等待对端交换机的 A3 脉冲信号
50	S_FAX_ROUND	FAX 通道：状态转移过程中
51	S_FAX_PhaseA	FAX 通道：传真呼叫建立 (PhaseA)
52	S_FAX_PhaseB	FAX 通道：传真报文前处理 (PhaseB)
53	S_FAX_SendDCS	FAX 通道：传真发送中向接收方发送 DCS 信号
54	S_FAX_Train	FAX 通道：传真报文传输前传输训练
55	S_FAX_PhaseC	FAX 通道：传真报文传输中 (PhaseC)
56	S_FAX_PhaseD	FAX 通道：传真报文后处理(PhaseD)

57	S_FAX_NextPage	FAX 通道: 传真报文传输下一页
58	S_FAX_AllSent	FAX 通道: 传真发送中报文传输结束
59	S_FAX_PhaseE	FAX 通道: 传真呼叫释放(PhaseE)
60	S_FAX_Reset	FAX 通道: 复位 MODEM
61	S_FAX_Init	FAX 通道: 初始化 MODEM
62	S_FAX_RcvDCS	FAX 通道: 传真接收, 接收发方的 DCS 信号
63	S_FAX_SendFTT	FAX 通道: 传真接收, 发送训练失败信号 FTT
64	S_FAX_SendCFR	FAX 通道: 传真接收, 发送可接受的证实信号 CFR
65	S_FAX_SendPPS	FAX 通道: 传真发送, ECM 模式下, 传真进行后续协商
66	S_FAX_RcvPPR	FAX 通道: 传真发送 PPS 后接收 PPR 信号
67	S_FAX_RepeatECMPage	FAX 通道: 传真发送, ECM 模式下, 传真进行数据重发
68	S_FAX_CTC_CTR	FAX 通道: ECM 模式下, 4 次重发后, 进行肯定协商
69	S_FAX_SendPPR	FAX 通道: ECM 模式下, 要求发送方重新发送数据
70	S_TUP_WaitPcmReset	TUP 通道: 电路复原
71	S_TUP_WaitSAM	TUP 通道: 等待对端交换机的后续地址消息
72	S_TUP_WaitGSM	TUP 通道: 等待对端交换机的 GSM 消息
73	S_TUP_WaitCLF	TUP 通道: 等待对端交换机的拆线消息
74	S_TUP_WaitPrefix	TUP 通道: 接收入局字冠
75	S_TUP_WaitDialAnswer	TUP 通道: 等待对端交换机的消息回应
76	S_TUP_WaitRLG	TUP 通道: 等待对端交换机的 RLG 消息
77	S_TUP_WaitSetCallerID	TUP 通道: 等待应用程序设置主叫号码
81	S_ISDN_OUT_WAIT_NET_RESPONSE	ISDN 通道: 等待网络响应
82	S_ISDN_OUT_PLS_APPEND_NO	ISDN 通道: 等待应用程序追加号码
83	S_ISDN_IN_CHK_CALL_IN	ISDN 通道: 检测到呼入
84	S_ISDN_IN_RECVING_NO	ISDN 通道: 正在接收号码
85	S_ISDN_IN_WAIT_TALK	ISDN 通道: 准备进入通话
86	S_ISDN_OUT_WAIT_ALERT	ISDN 通道: 等待对方发提醒信号
87	S_ISDN_CALL_BEGIN	ISDN 通道: 发起呼叫(去话), 或检测到呼入(来话)
88	S_ISDN_WAIT_HUANGUP	ISDN 通道: 等待挂机
89	S_ISDN_IN_CALL_PROCEEDING	ISDN 通道: 呼叫进程
100	S_CALL_SENDRING	磁石通道: 发送振铃
105	S_SPY_RCVPHONUM	监控通道: 接收号码
110	S_SPY_SS1RESET	SS1 监控: 复原
111	S_SPY_SS1WAITBWDACK	SS1 监控: 等待后向证实
112	S_SPY_SS1WAITKB	SS1 监控: 等待 KB
120	S_ISUP_WaitSAM	ISUP 通道: 等待对端交换机的 SAM
121	S_ISUP_WaitRLC	ISUP 通道: 等待对端交换机的释放监护信号 RLC
122	S_ISUP_WaitReset	ISUP 通道: 电路复原
123	S_ISUP_LocallyBlocked	ISUP 通道: 本地闭塞
124	S_ISUP_RemotelyBlocked	ISUP 通道: 远端闭塞
125	S_ISUP_WaitDialAnswer	ISUP 通道: 等待对端交换机的消息回应
126	S_ISUP_WaitINF	ISUP 通道: 等待对端交换机的 INF 消息
127	S_ISUP_WaitSetCallerID	ISUP 通道: 等待应用程序设置主叫号码
128	S_DTRC_ACTIVE	DTR 通道: 被监控话路处于非空闲状态
129	S_ISUP_Suspend	ISUP 通道: 暂停
130	S_CALL_EM_TXPHONUM	E/M 通道: 拨号或变声资源被占用
131	S_CALL_EM_WaitidleCAS	E/M 通道: 等待对端的示闲信号
132	S_CALL_VOIP_DIALING	IP 通道: VoIP 主叫拨号状态
133	S_CALL_VOIP_WAIT_CONNECTED	IP 通道: VoIP 被叫摘机等待进入通话状态
134	S_CALL_VOIP_CHANNEL_UNUSABLE	IP 通道: VoIP 通道目前不可用
135	S_CALL_DISCONNECT	USB 连接断开
136	S_CALL_SS1WaitFlashEnd	SS1 通道: 等待闪断发送结束
137	S_CALL_FlashEnd	SS1 通道: 闪断结束
139	S_CALL_SIGNAL_ERROR	DTR 通道: 帧同步正常但信号不完整

140	S_CALL_FRAME_ERROR	DTR 通道: 帧同步不正常但信号完整
150	S_CALL_VOIP_SESSION_PROCEEDING	IP 通道: 会话进行, 该值为收到 183 时的状态
151	S_CALL_VOIP_REG_ING	IP 通道: SIP 通道注册中
152	S_CALL_VOIP_REG_FAILED	IP 通道: SIP 通道注册失败
153	S_CALL_VOIP_CALL_ON_HOLD	IP 通道: SIP 通道呼叫保持状态
160	S_IP_MEDIA_LOCK	MEDIA 媒体通道: 媒体通道锁定状态
161	S_IP_MEDIA_OPEN	MEDIA 媒体通道: 媒体通道打开状态
162	S_SPY_RBSWAITACK	RBS 监控通道: 等待应答
163	S_SPY_RBSSENDACK	RBS 监控通道: 发送应答
170	S_IPR_USING	IPR 系列产品通道: 通道被占用状态
171	S_IPR_COMMUNICATING	IPR 系列产品通道: 通道处于通讯中
172	S_ISUP_WaitCOT	ISUP 通道: 等待 COT 消息
300	S_FAX_EOR_ERR	FAX 通道: ECM 模式下, 传真多次重发后, 进行否定处理
301	S_FAX_RNR_RR	FAX 通道: ECM 模式下, 接收方忙
302	S_FAX_RTN	FAX 通道: 传真接收, 接收报文否定及重新训练
303	S_FAX_NextPage_EOM	FAX 通道: 下页发送需要从 PhaseB 开始, 重新训练
400	S_FAX_V34_PhaseV8	FAX 通道: V.34 模式下, 处在 V.8 训练阶段
其它		保留

功能描述:

获取通道的当前状态。

注意事项:

- 为提高应用程序的运行效率, 建议使用 [E_CHG_ChState](#) 事件来获取通道状态的变化信息。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetChStateKeepTime](#), [SsmGetPendingReason](#)

2.3.3.2 SsmGetChStateKeepTime

获取通道在当前状态的保持时间。

函数原型:

```
long SsmGetChStateKeepTime(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
≥0	本通道在当前状态的停留时间, 单位为毫秒

功能描述:

获取通道在当前状态的保持时间。

灵活使用本函数可以得到通道的许多信息, 例如:

- 当呼入通道处于 S_CALL_RINGING 状态时, 本函数可以取得振铃持续的时间;
- 当本通道处于 S_CALL_TALKING 状态时, 本函数可以取得通话持续的时间;
- 当呼出通道处于 S_CALL_WAIT_REMOTE_PICKUP 状态时, 本函数可以取得等待被叫用户摘机时的等待时间。

注意事项:

- 该函数不支持 IP 卡。

相关信息:

驱动程序版本要求	SynCTI Ver.3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

 相关函数: [SsmGetChState](#)

2.3.4 获取 PendingReason

2.3.4.1 SsmGetPendingReason

获取通道进入 S_CALL_PENDING 状态的具体原因。

函数原型:

```
int SsmGetPendingReason(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1 表示调用失败，其它值的含义如下表所示:

返回值	宏	适用通道类型	描述
0	ANALOGOUT_NO_DIALTONE	模拟中继线通道	去话呼叫失败: 未检测到拨号音
1	ANALOGOUT_BUSYTONE	模拟中继线通道	去话呼叫失败: 检测到忙音
2	ANALOGOUT_ECHO_NOVOICE	模拟中继线通道	去话呼叫: 检测到回铃音后, 线路上保持静默, 驱动程序无法判别被叫是否摘机
3	ANALOGOUT_NOANSWER	模拟中继线通道	去话呼叫: 检测到回铃信后, 被叫用户未在配置项 MaxWaitAutoDialAnswerTime 指定的时间内应答
4	ANALOGOUT_TALKING_REMOTE_HANGUPED	模拟中继线通道	通道在“连接”状态时检测到对端用户挂机
5	ANALOGOUT_NOVOICE	模拟中继线通道	去话呼叫: 拨号过程已经完成, 未在线路上检测到回铃音和任何其它的语音信号, 驱动程序无法判断被叫用户是否摘机
10	PEND_WaitBckStpMsg	ISUP / TUP ISDN SS1	来话呼叫: 驱动程序按照预设的收号规则完成被叫号码等信息的接收后, “自动应答来话呼叫”功能未开启, 需要应用程序自行决定是否接受此呼叫
11	SS1IN_BWD_KB5	SS1	来话呼叫: 收到对端交换机的 KD 信号后, 应用程序设置 KB=5 (空号), 等待主叫方的拆线信号
12	PEND_RemoteHangupOnTalking	ISUP / TUP SS1	来话呼叫: 双方通话时, 对端挂机
13	PEND_AutoDialFailed	ISUP / TUP / ISDN	去话呼叫失败。具体的自动拨号失败原因, 可调用 SsmGetAutoDialFailureReason() 获得
14	PEND_SsxUnusable	ISUP / TUP	SS7 信令不可用
15	PEND_CircuitReset	ISUP / TUP	发生电路复原事件
16	PEND_PcmSyncLos	ISUP / TUP SS1	数字中继线的基本帧 (0 时隙) 同步信号丢失

20	SS1OUT_TALKING_REMOTE_HANGUPED PEND_CalleeHangupOnTalking	SS1 ISUP / TUP	去话呼叫：双方通话时，对端挂机
21	SS1OUT_NOANSWER	SS1	去话呼叫失败：被叫用户未在配置项 MaxWaitAutoDialAnswerTime 指定的时间内应答
22	SS1OUT_NOBWDACK	SS1	去话呼叫失败：等待对端交换机的“占用证实”信号超时。
23	SS1OUT_DIALING_BWD_HANGUP	SS1	去话呼叫失败：对端交换机取消呼叫
24	SS1OUT_BWD_A5	SS1	去话呼叫失败：收到对端交换机的 A5 信号（空号）
25	SS1OUT_BWD_KB5	SS1	去话呼叫失败：收到对端交换机的 KB=5（空号）
26	SS1OUT_BWD_KB2	SS1	去话呼叫失败：收到对端交换机的 KB=2（被叫用户“市忙”）
27	SS1OUT_BWD_KB3	SS1	去话呼叫失败：收到对端交换机的 KB=3（被叫用户“长忙”）
28	SS1OUT_BWD_A4	SS1	去话呼叫失败：收到对端交换机的 A4 信号（机键拥塞）
29	SS1OUT_BWD_KB4	SS1	去话呼叫失败：收到对端交换机的 KB=4 信号（机键拥塞）
30	SS1OUT_TIMEOUT_BWD_A	SS1	去话呼叫失败：等待对端交换机的后向 A 组信号超时
31	SS1OUT_TIMEOUT_BWD_A_STOP	SS1	去话呼叫失败：等待对端交换机停发后向 A 组信号超时
32	SS1OUT_TIMEOUT_BWD_KB	SS1	去话呼叫失败：等待对端交换机的 KB 信号超时
33	SS1OUT_TIMEOUT_BWD_KB_STOP	SS1	去话呼叫失败：等待对端交换机停发 KB 信号超时
34	SS1OUT_TIMEOUT_CALLERID_BWD_A1	SS1	去话呼叫失败：向对端交换机发送主叫号码时，等待对端交换机后向 A 组信号超时
35	SS1OUT_TIMEOUT_CALLERID_BWD_A1_STOP	SS1	去话呼叫失败：向对端交换机发送主叫号码时，等待对端交换机停发后向 A 组信号超时，自动拨号失败
36	SS1OUT_UNDEFINED_CALLERID_BWD_A	SS1	去话呼叫失败：向对端交换机发送主叫号码时，收到未定义的后向 A 组信号
37	SS1OUT_UNDEFINED_BWD_A		去话呼叫失败：收到未定义的后向 A 组信号
38	SS1OUT_UNDEFINED_BWD_KB		去话呼叫失败：收到未定义的 KB 信号
41	ISDN_CALLOVER	ISDN	呼叫结束，对方先挂机
42	ISDN_WAIT_RELEASE	ISDN	收到对方的“拆线”消息，等待释放本端链路
43	ISDN_HANGING	ISDN	本端先挂机，正在拆线
44	ISDN_RELEASEING	ISDN	正在释放呼叫
45	ISDN_UNALLOCATED_NUMBER	ISDN	未分配的号码
46	ISDN_NETWORK_BUSY	ISDN	网络忙
47	ISDN_CIRCUIT_NOT_AVAILABLE	ISDN	指定的电路不可用
48	PEND_CalleeHangupOnWaitRemotePickUp	TUP	去话呼叫失败：等待被叫用户摘机时，收到对端交换机的拆线消息
49	ISUP_HardCircuitBlock	ISUP	收到对端交换机的硬件闭塞消息
50	ISUP_RemoteSuspend	ISUP	T6 定时器溢出。有关 T6 定时器的更多信息请参见第 1 章中 “ ISUP 通道的状态转移 ” 部分内容
51	PEND_RcvHGBOrSGB	TUP	收到对端交换机闭塞消息（SGB/HGB）
52	ISDN_NO_ANSWER	ISDN	无应答
53	ISDN_CALL_REJ	ISDN	呼叫拒绝

54	PEND_RemoteHangupOnRinging	ISUP / TUP	来话呼叫：通道处于“振铃”状态时，对端交换机取消了呼叫
55	ISDN_NO_ROUTE	ISDN	无法路由，原因可能为“移动电话不在服务区”
56	ISDN_NO_ROUTE_TO_DEST	ISDN	无法路由，原因可能为“移动电话关机”
57	EM_USER_BUSY	E/M	用户忙
58	EM_CH_ERROR	E/M	通道故障
59	EM_LOCAL_HANGUP	E/M	本地先挂机
60	EM_LOCAL_NOANSWER	E/M	本地无应答
61	EM_REMOTE_HANGUP	E/M	对端先挂机
62	EM_REMOTE_NOANSWER	E/M	对端无应答
63	PEND_RemoteHangupOnSuspend	ISUP	“暂停”状态时，对端用户挂机
64	PEND_CalleeHangupOnSuspend	ISUP	“暂停”状态时，被叫用户挂机
65	ISDN_NORMAL_UNSPEC	ISDN	呼叫正常结束
66	ISDN_USER_ABSENCE	ISDN	用户缺席
67	ISDN_INVALID_NUMBER_FORMAT	ISDN	无效的号码格式
68	ISDN_NO_CIRCUIT_AVAILABLE	ISDN	无可用的电路/通路
69	IP_REMOTE_CRASHED	SIP	对方异常死机
70	IP_REMOTE_CLOSED	SIP	对方 BYE 或者对方 CANCEL
71	IP_DIAL_TIMEOUT	SIP	IP 卡呼叫超时
72	IP_REMOTE_REJECTED	SIP	IP 卡被对方拒绝
73	IP_REFER_SECCED	SIP	IP 卡呼叫转移成功后挂起
74	IP_REFER_REFUSED	SIP	IP 卡呼叫转移失败后挂起
75	IP_STUN_FAILED	SIP	IP 卡呼叫过程中 stun 穿透失败
76	IP_NOTRCV_ACK	SIP	IP 卡摘机后未收到 ACK 挂起
77	IP_REDIREDT_FAIL	SIP	IP 卡呼叫重定向失败
其它	保留		

功能描述：

获取通道迁移到 S_CALL_PENDING 状态的具体原因。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.3.5 通道摘机函数（CTI 系列）

2.3.5.1 SsmPickup

请参见 [SsmPickupANX](#)

2.3.5.2 SsmPickupANX

在通道上执行摘机操作。SsmPickupANX 除了具备 SsmPickup 的全部功能外，还可以在应答消息中指示被叫用户的计费情况，但只适用于 TUP 通道和 ISUP 通道。

函数原型：

```
int SsmPickup(int ch)  
int SsmPickupANX(int ch, int nANX)
```

参数说明：

ch	通道编号
	=0: 无指示
nANX	=1: 计费
	=2: 免费

返回值：

-1	操作失败
0	操作成功

功能描述：

在通道上执行摘机操作。

◆ **ISUP 通道：**

对于 ISUP 通道，应用程序调用本函数时，如果通道处于“振铃”状态，驱动程序会根据配置项 [DefaultCalledPickupMsg](#) 的设置向对端交换机发送 CON（地址全并摘机）或 ANM（应答）消息，消息中的后向呼叫表示语字段可以通过配置项 [DefaultBackwardCallInd](#) 进行设置；如果通道处于“空闲”或“锁定”状态，驱动程序不会向对端交换机发送任何消息。

◆ **ISDN 通道：**

对于 ISDN 通道，应用程序调用本函数时，如果通道处于“振铃”状态，驱动程序会向对端交换机发送 CONNECT 消息，同时启动定时器 T313，然后将通道迁移到 S_ISDN_IN_WAIT_TALK 状态；如果通道处于“空闲”或“锁定”状态，驱动程序不会向对端交换机发送任何消息。

◆ **模拟中继线通道：**

对于模拟中继线通道的来话呼叫，应用程序调用本函数时，如果通道处于 S_CALL_RINGING 状态，并且铃流信号的电平为高电平，为防止线路上的铃流电压对板卡造成损害，驱动程序不会马上向硬件电路发布摘机命令。只有当铃流信号转变为低电平后，驱动程序才会真正执行摘机命令，并向应用程序抛出 [E_SYS_ActualPickup](#) 事件。函数 [SsmCheckActualPickup](#) 也可以用来查询摘机命令是否执行完毕。

◆ **IP 通道：**

对处于“空闲”状态的 IP 通道调用本函数，通道将处于“呼出锁定”状态。在“呼出锁定”状态下，用户可调用 [SsmHangup](#) 使通道回到空闲，也可调用拨号函数呼出。对处于“振铃”状态的 IP 通道调用本函数，其意义相当于接收来电。其他状态下调用本函数，返回错误。

注意事项：

- SsmPickup 适用于 SS1 通道、TUP 通道、ISUP 通道、ISDN 通道、模拟中继线通道、IP 通道和 EM 控制通道，SsmPickupANX 只适用于 TUP 通道和 ISUP 通道；
- 对于模拟中继线通道，应用程序调用本函数后，驱动程序会自动清零铃流信号检测器，清零并启动信号音检测器。

相关信息：

驱动程序版本要求	SsmPickup 要求 SynCTI Ver. 3.0 或更高 SsmPickupANX 要求 SynCTI Ver. 4.7.1.5 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmHangup](#), [SsmHangupEx](#)

2.3.6 通道挂机函数（CTI 系列）

2.3.6.1 SsmHangup

请参见 [SsmHangupEx](#)

2.3.6.2 SsmHangupEx

向通道发布挂机命令。SsmHangupEx 还可以向驱动程序传递挂机原因等信息，但只适用于 ISUP 通道。

函数原型：

```
int SsmHangup(int ch)  
int SsmHangupEx(int ch, UCHAR ucCauseVal)
```

参数说明：

ch	通道编号
ucCauseVal	挂机的释放原因，具体值请参考 ISUP 相关协议说明

返回值：

-1	挂机操作失败，失败原因可以通过 SsmGetLastErrMsg 函数获得
0	挂机操作成功

功能描述：

向通道发布挂机命令。驱动程序执行的操作取决于通道的类型。

➤ ISUP 通道

本函数会引发驱动程序向对端交换机发送 REL (释放) 消息，可以用来拒绝来话呼叫，也可以用来取消去话呼叫。如果应用程序调用的是 SsmHangupEx，REL 消息中的释放原因由参数 ucCauseVal 决定；如果调用的是 SsmHangup，释放原因会因通道的状态而异：

通道状态	REL 消息中的释放原因
S_CALL_PICKUPED	不发送 REL 消息
S_ISUP_WaitDialAnswer	
S_CALL_WAIT_REMOTE_PICKUP	C_ISUP_REL_NORMAL_REL
S_CALL_TALKING	
S_ISUP_WaitSetCallerID	
S_ISUP_Suspend	C_ISUP_REL_DENY
S_ISUP_WaitSAM	
S_ISUP_WaitINF	
S_CALL_RINGING	
S_CALL_PENDING (挂起原因为 PEND_WaitBckStpMsg)	由配置项 DefaultHangupRELInd 或函数 SsmSetISupFlag (携带 ISUP_REL_DENY_SetToOther 参数) 的设定值决定

注：上表中使用的宏在头文件 shpa3api.h 中声明。

➤ TUP 通道

如果通道处于 **S_CALL_RINGING** 状态，调用本函数表明应用程序拒绝来话呼叫，驱动程序会向对端交换机发送呼叫拒绝消息。呼叫拒绝消息可以是 CBK 或 CFL 消息，具体使用哪一个，由配置项 [HangupRingSendCBK](#) 决定。

➤ ISDN 通道

对于 ISDN 通道，调用本函数后：

- ◆ 如果通道处于 **S_CALL_PENDING** 状态，驱动程序会向对端交换机发送“RELEASE”消息；
- ◆ 如果通道处于 **S_ISDN_OUT_PLS_APPEND_NO** 状态，本次呼叫将被挂断，驱动程序会向对端交换机发送“DISCONNECT”消息，消息中的挂机原因字段会被设置为“号码未分配”；
- ◆ 如通道处于 **S_CALL_TALKING** 状态，表明本端先挂机，驱动程序会向对端交换机发送“DISCONNECT”消息，消息中的挂机原因会被设置为“呼叫正常结束”；
- ◆ 如果通道处于 **S_CALL_RINGING** 状态，表明应用程序拒绝本次来话呼叫，驱动程序会向对端交换机发送“DISCONNECT”消息，消息中的挂机原因将被设置为“呼叫拒绝”。

➤ IP 通道

对于 IP 通道，调用本函数后：

- ◆ 对于“呼出锁定”状态的 IP 通道调用本函数，通道将迁移到“空闲”状态；
- ◆ 对于非空闲状态的 IP 通道调用本函数，IP 状态机将根据当前的状态，触发不同的协议命令，并使 IP 会话结束。

注意事项：

- **SsmHangupEx** 函数适用于 ISUP 通道、TUP 通道、ISDN 通道、ANALOG 通道、SS1 通道。IP 通道和 EM 控制通道只能使用 [SsmHangup](#) 函数；
- 对于 ISDN 通道，应用程序可以在调用本函数前先调用函数 [SsmISDNSetHangupRzn](#) 设置挂机原因。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高 SsmHangupEx : SynCTI Ver. 4.7.1.5 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmPickup](#), [SsmPickupANX](#)

2.3.7 设置通道的呼叫方向

2.3.7.1 SsmSetAutoCallDirection

设置通道的呼叫方向。

函数原型：

```
int SsmSetAutoCallDirection(int ch, BOOL bEnAutoCall, int nDirection)
```

参数说明：

ch	通道编号	是否由驱动程序自动进行呼叫处理。
bEnAutoCall	=TRUE: 是 =FALSE: 否。此时, nDirection 将被忽略	呼叫方向, 仅当 bEnAutoCall 设置为 TRUE 时有效。取值范围:
nDirection	=0: 只能进行来话呼叫 =1: 只能进行去话呼叫	

=2: 既可以进行来话呼叫，也可以进行去话呼叫（此值对 SS1 无效）

返回值:

-1	调用失败。失败原因可以通过调用函数 SsmGetLastErrMsg 获得
≥0	调用成功

功能描述:

设置通道的呼叫方向。

注意事项:

- 本函数只支持 SHD 系列板卡的 SS1 通道和 ISDN 通道，以及 SHN 系列卡的 SIP 通道；
- 对于 SHD 系列板卡的 SS1 通道和 ISDN 通道，只有在通道处于 S_CALL_STANDBY 状态时才能调用本函数；对于 SHN 系列板卡的 SIP 通道，本函数的调用不受通道状态限制。
- 本函数对 SS7 中继通道不起作用。如果需要对 SS7 通道设置呼叫方向，可以通过阻塞函数来设置，具体参考函数 [SsmBlockLocalCh](#)。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetChState](#), [SsmGetAutoCallDirection](#)

2.3.7.2 SsmGetAutoCallDirection

获取通道的呼叫方向。

函数原型:

```
int SsmGetAutoCallDirection(int ch, int* pnDirection)
```

参数说明:

ch	通道编号
	返回呼叫方向，仅当 bEnAutoCall 设置为 TRUE 时有效。取值范围：
pnDirection	=0: 只能进行来话呼叫 =1: 只能进行去话呼叫 =2: 既可以进行来话呼叫，也可以进行去话呼叫 =3: 运行在监控模式下，并且信令模式为 ISDN 或 TUP

返回值:

-1	调用失败
0	本通道不是由驱动程序自动进行呼叫处理
1	本通道由驱动程序自动进行呼叫处理

功能描述:

获取通道的呼叫方向。

注意事项:
相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetAutoCallDirection](#)

2.3.8 获取释放原因

2.3.8.1 SsmGetReleaseReason

获取对端交换机发送的释放消息中的释放原因值。

函数原型:

```
WORD SsmGetReleaseReason(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

0	原因未指明 释放消息中的原因值，具体的消息类型及含义与通道类型有关： ➤ ISUP 通道：返回 REL 消息中的释放原因，具体取值及含义请参见 附录 1 ISDN 通道释放原因值表 ，常见的有： 0x01: 空号 0x10: 正常拆线 0x11: 用户忙 0x12: 无应答 0x15: 拒绝 0x1c: 地址不全 0x1f: 正常 0x2a: 交换设备拥塞 0x14: 用户缺席 0x16: 号码改变
>0	➤ ISDN 通道：返回 ISDN 消息 ISDN 通道释放原因值表字段，具体取值及含义请参见 附录 1 ISDN 通道释放原因值表

功能描述:

获取对端交换机发送的释放消息中的释放原因值。

注意事项:

- 在开始处理来话呼叫前（即当驱动程序从对端交换机收到一条呼叫建立消息时），驱动程序都会将释放原因值设置为 0；
- 本函数只适用于 ISDN 通道（包括网络侧和用户侧）和 ISUP 通道。

相关信息:

驱动程序版本要求	SynCTI Ver. 4.7.1.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.3.9 TUP 通道专用函数

2.3.9.1 SsmSetCalleeHoldFlag

设置 TUP 通道的“锁定主叫用户”功能。

函数原型:

```
int SsmSetCalleeHoldFlag(int ch, BOOL bFlag)
```

参数说明:

ch	通道编号
bFlag	=TRUE: 开启 =FALSE: 关闭

返回值:

-1	操作失败。失败原因可以通过调用函数 SsmGetLastErrMsg 获得
0	操作成功

功能描述:

设置 TUP 通道的“锁定主叫用户”功能。更多信息请参见第 1 章中“[锁定主叫用户功能](#)”部分内容。

注意事项:

- 本函数只适用于 SHD 系列板卡的 TUP 通道；
- 只用当通道处于 S_CALL_TALKING 状态时，才可以调用本函数；
- 每次来话呼叫时，驱动程序在将通道迁移到 S_CALL_TALKING 状态前，会自动将“锁定主叫用户”功能设置为关闭状态。

相关信息:

驱动程序版本要求	SynCTI Ver. 4.7 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.3.9.2 SsmGetTupFlag

获取 TUP 消息中的特定参数或字段值。

函数原型:

```
int SsmGetTupFlag(int ch,int nType,WORD *pdwParaValue)
```

参数说明:

ch	通道编号
nType	选择参数或字段类型。取值范围: =1(Tup_ANX): 去话呼叫时，当通道处于 S_CALL_WAIT_REMOTE_PICKUP 状态时，如果被叫用户接受此呼叫，对端交换机会向本端发送应答信号(Answer Signal)，驱动程序会将通道迁移到 S_CALL_TALKING 状态。此时，参数 pdwParaValue 可以获取具体的应答消息类型
pdwParaValue	返回参数或字段的值。具体含义与 nType 有关: nType=Tup_ANX: 0x06: ANU 消息 (无指示, No Indication) 0x16: ANC 消息 (计费, Charge) 0x26: ANN 消息 (免费, No Charge)

返回值:

1	成功获取消息
-1	调用失败，失败原因可以从 SsmGetLastErrMsg 函数中获取

功能描述:

获取 TUP 消息中的特定参数或字段值。

注意事项:

- 本函数只适用于 TUP 通道。

相关信息:

驱动程序版本要求	SynCTI Ver. 4.7.1.9 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: 无

2.3.9.3 SsmSetUpParameter

设置呼出通道的 Tup 消息可选参数。

函数原型:

```
int SsmSetUpParameter (int nBCh, UCHAR ucMsgTypeCode, UCHAR ucParamTypeCode, WORD wLength,  
P UCHAR pucContent);
```

参数说明:

nBCh	通道编号
ucMsgTypeCode	消息类型。取值范围: 0x21: IAI 消息 0x11: IAM 消息 其它: 保留
ucParamTypeCode	参数类型。具体取值与 ucMsgTypeCode 有关: ucMsgTypeCode = IAI 消息: 0x00: 设置 IAI 中原被叫地址性质表示语 0x01: 设置 IAI 消息呼出用户类别字段 0x02: 设置 IAI 消息中的主叫用户线标识字段的地址表示语 ucMsgTypeCode = IAM 消息: 0x01: 设置 IAM 消息呼出用户类别字段 其它: 保留
wLength	参数内容的字节长度
pucContent	参数内容的缓冲区指针

返回值:

-1	参数设置失败
0	设置成功

功能描述:

设置呼出通道的 Tup 消息可选参数。目前只对 IAI 中原被叫地址性质表示语、主叫用户线标识字段的地址表示语、IAI/IAM 消息中呼出用户类别字段有效。相关配置项: [CalloutIAM_CAT](#)。

相关信息:

驱动程序版本要求	SynCTI Ver. 4.7.3.1 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: 无

2.3.9.4 SsmGetUpParameter

获取收到 IAM 消息中的指定字段。

函数原型:

```
int SsmGetUpParameter (int nBCh, UCHAR ucMsgTypeCode, UCHAR ucParamTypeCode, PUCHAR pucContent,  
WORD wNumberOfBytesToWrite, LPWORD lpNumberOfBytesWritten);
```

参数说明:

nBCh	通道编号
ucMsgTypeCode	消息类型。取值范围: 0x11: IAM 消息 其它: 保留
ucParamTypeCode	参数类型。具体取值与 ucMsgTypeCode 有关: 0xE3: 获取 IAM 消息的 CIC 字段 其它: 保留
pucContent	存放参数内容的缓冲区指针
wNumberOfBytesToWrite	缓冲区的大小
lpNumberOfBytesWritten	实际返回的数据长度

返回值:

-1	获取失败
0	获取成功

功能描述:

获取收到 IAM 消息中指定的字段。目前只支持获取 IAM 消息的 CIC 字段。

相关信息:

驱动程序版本要求	SynCTI Ver.5.3.1.2 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: 无

2.3.10 ISUP 通道专用函数

2.3.10.1 SsmSetISUPCAT

设置 ISUP 协议中 IAM 消息中的主叫用户类别。

函数原型:

```
int SsmSetISUPCAT(int nch, UCHAR ucCallerCAT)
```

参数说明:

nch	通道编号
ucCallerCAT	十六进制的主叫用户类别

返回值: 0

功能描述:

设置 ISUP 协议中 IAM 消息中的主叫用户类别。本函数设置的参数也可以通过配置项 [DefaultIAM_CAT](#) 进行设置。

注意事项:

- 本函数应该在调用 [SsmAutoDial](#) 或 [SsmAutoDialEx](#) 前调用;
- 本函数设置的参数的有效期直至重启程序重新加载配置文件为止。

相关信息:

驱动程序版本要求	SynCTI Ver. 4.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:**2.3.10.2 SsmSetIsupParameter**

设置本端发送给对端交换机的 isup 消息中的可选字段。

函数原型:

```
int SsmSetIsupParameter(int ch, UCHAR ucMsuType, UCHAR ucParamType, WORD wLength, PUCHAR pucContent)
```

参数说明:

ch	通道编号
ucMsuType	消息类型。取值范围: 0x01: IAM 消息 0X09: ANM 消息 其它: 保留
ucParamType	参数类型。具体取值与 ucMsuType 有关: ucMsuType= IAM 消息: 可以设置可选字段 ucMsuType= ANM 消息: 可以设置除了下列可选字段以外的其它可选字段: ◆ 后向呼叫指示 每个可选字段的参数类型请参见 附录 2 ISUP 可选参数列表和部分参数介绍 。 ucMsuType=其它: 保留
wLength	参数的长度, 单位为字节。如果 ucMsuType=IAM 消息, 将 wLength 设置为 0 表示 IAM 消息中不再包含 ucParamType 指定的可选字段
pucContent	参数的内容。如果 ucMsuType=IAM, 将 pucContent 设置为 NULL 表示取消 IAM 消息中由 ucParamType 指定的可选字段, 只能取消本函数设置的字段

返回值:

-1	调用失败
0	调用成功

功能描述:

设置本端发送给对端交换机的 isup 消息中的可选字段。

注意事项:

- 如果 ucMsuType 设置为 IAM 消息, 本函数应该在 [SsmAutoDial](#) 之前调用。参数的值一旦设置后, 会被驱动程序保存在内部缓冲区中, 并会在后续的 IAM 消息中继续使用。对于 IAM 消息中的用户到用户信息字段, 也可以通过配置项 [Usr2UsrInfo](#) 进行设置;
- 本函数设置的参数的有效期直至重启程序重新加载配置文件为止;
- 取消设置时, 需要将 wLength 设置为 0, 同时将 pucContent 设置为 NULL, 否则调用函数会返回失败;
- 通道必须已经设置携带指定的可选字段, 才能成功取消设置, 否则调用函数会返回失败。

相关信息:

驱动程序版本要求	SynCTI Ver. 4.7.2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetIsupParameter](#)

代码范例:

在 IAM 消息中增加改发信息字段。改发信息字段的参数类型为 0x13，共包含 2 个字节，改发信息参数字段中的第 1 个 8 位位组（记为 HGFEDCBA，A 为最低 bit 位）的含义为：

比特	含义	取值	描述
CBA	改发表示语	000	无改发
		001	呼叫重新选路
		010	呼叫重新选路，所有的改发信息提供限制
		011	呼叫变更
		100	呼叫变更，所有的改发信息提供限制
		110	呼叫变更，改发号码提供限制
		其它	保留
D	备用		
HGFE	原改发原因	0000	未知 / 不可用
		0001	用户忙
		0010	无应答
		0011	无条件
		其它	保留

第 2 个 8 位位组（记为 PONMLKJI，I 为最低 bit 位）的含义为：

比特	含义	取值	描述
KJI	改发计数器	1~5	呼叫经过改发的次数用 1 和 5 之间的二进制数表示
		0	应设置为 0
PONM	改发原因	0000	未知 / 不可用
		0001	用户忙
		0010	无应答
		0011	无条件
		0110	移动用户不可及
		其它	保留

如果设置改发信息为用户无应答前转，则相关的代码如下：

```

UCHAR IsupParamRI[2];
IsupParamRI[0] = 0x03;      //设置改发信息内容
IsupParamRI[1] = 0x21;
SsmSetIsupParameter(ch, 0x01, 0x13, 2, IsupParamRI);
SsmAutoDial(ch,...);

```

2.3.10.3 SsmGetIsupParameter

获取对端交换机发送给本端的 isup 消息中的可选字段。

函数原型：

```

int SsmGetIsupParameter(int nBCh, UCHAR ucMsuType, UCHAR ucParamType, PUCHAR pucContent, WORD wBufSize, LPWORD lpNumberOfBytesWritten)

```

参数说明：

nBCh	通道编号
ucMsuType	消息类型。取值范围： 0x01: IAM 消息 0x06: ACM 消息 其它：保留
ucParamType	<ul style="list-style-type: none"> ● IAM 消息内容字段： 0xE3: 获取 IAM 消息的 CIC 字段 0xE7: 获取 IAM 消息的 TMR 字段 0xE8: 获取 IAM 消息的主叫子地址 0xE9: 获取 IAM 消息的被叫子地址 ● 可选字段的类型。具体取值与 ucMsuType 有关，更多信息请参见附录 2 ISUP 可选参数列表和部分参数介绍
pucContent	保存可选字段内容的缓冲区指针，由应用程序分配存储空间，缓冲区长度的建议值为 272。 注意： 返回的信息内容中不包括可选字段的类型编码（1 字节）和长度（1 字节）
wBufSize	pucContent 的长度，单位为字节
lpNumberOfBytesWritten	返回驱动程序写入到 pucContent 中的数据的实际长度，单位为字节

返回值：

-1	调用失败
0	调用成功

功能描述:

获取对端交换机发送给本端的 isup 消息中的可选字段。

注意事项:
相关信息:

驱动程序版本要求	SynCTI Ver. 4.7.2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetIsupParameter](#)

2.3.10.4 SsmSetIsupFlag

ISUP 中用户部分参数设置。

函数原型:

```
int SsmSetIsupFlag(int ch, int nType, DWORD dwValue, PVOID pV)
```

参数说明:

ch	通道编号 选择参数的类别。可取的值为: ISUP_PhoNumParam=2: ISUP_CallerParam=1: ISUP_REL_DENY_SetToOther=100: nType ISUP_PhoNumREL=3: ISUP_IAM_TMR=4: DefaultIAM_OriginalCalleeParam=101: ISUP_PhoNumRELEx=5: dwValue 设置参数的值, 与 nType 的取值有关:	设置 IAM 消息中被叫用户号码字段的被叫参数, 此时参数 dwValue 为设置值, 参数 pV 无效 设置主叫用户号码字段的主叫参数, 此时参数 dwValue 为设置值, 参数 pV 无效 当应用程序调用 SsmHangup 函数而引发驱动程序向对端交换机发送 REL 消息时, 本参数 dwValue 可以设置 REL 消息中携带的释放原因值, 参数 pV 无效 向对端交换机发送包含号码改发信息的拆线消息 (REL), 此时参数 pV 有效 设置 IAM 消息里传输媒介参数, 参数 dwValue 为设置值, 参数 pV 无效 原被叫参数 与 ISUP_PhoNumREL 类似, 向对端交换机发送包含号码改发信息的拆线消息 (REL), 并可以自定义释放原因值, 此时参数 pV 有效 请参见配置项 DefaultIAM_CalleeParam 的说明 请参见配置项 DefaultIAM_CallerParam 的说明 请参见配置项 DefaultHangupRELInd 的说明 请参见配置项 DefaultIAM_TransmissionMediumRequirement 的说明 原被叫参数的值:
----	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

		备 用: 0x1000 用户号码: 0x1001 未 知: 0x1002 国内号码: 0x1003 国际号码: 0x1004
	ISUP_PhoNumREL	保留未用
pV 的含义与 nType 的取值有关:		
➤ nType=ISUP_PhoNumREL:		
pV 是指向 ISUP_RIREL 结构对象的指针，用于定制 REL 消息中的参数。ISUP_RIREL 结构在 Shpa3api.h 中声明如下:		
<pre>typedef struct tag_ISUP_RIREL{ WORD wRIMsg; WORD wRIPhoNumPara; WORD wPhoNumLen; UCHAR ucRIPhoNum[20]; }ISUP_RIREL,*PISUP_RIREL;</pre>		
上述结构中的参数含义为:		
<p>wRIMsg: “改发信息”字段的比特 PONMLKJIHGFEDCBA;</p> <p>wRIPhoNumPara: “改发号码”字段的比特 Bit15...Bit0;</p> <p>wPhoNumLen: ucRIPhoNum 中字符的个数;</p> <p>ucRIPhoNum: 改发的电话号码字符串。</p>		
➤ nType=ISUP_PhoNumRELEX:		
pV 是指向 ISUP_RIRELEX 结构对象的指针，用于定制 REL 消息中的参数。ISUP_RIRELEX 结构在 Shpa3api.h 中声明如下:		
<pre>typedef struct tag_ISUP_RIRELEX { UCHAR ucCauseInd; WORD wRIMsg; WORD wRIPhoNumPara; WORD wPhoNumLen; UCHAR ucRIPhoNum[20]; }ISUP_RIRELEX,*PISUP_RIRELEX;</pre>		
上述结构中的参数含义为:		
<p>ucCauseInd: 释放原因值;</p> <p>wRIMsg: “改发信息”字段的比特 PONMLKJIHGFEDCBA;</p> <p>wRIPhoNumPara: “改发号码”字段的比特 Bit15...Bit0;</p> <p>wPhoNumLen: ucRIPhoNum 中字符的个数;</p> <p>ucRIPhoNum: 改发的电话号码字符串。</p>		
➤ nType=其它: 保留未用		

注：表中用到的常量均在头文件 Shpa3api.h 中声明。

返回值：

-1	失败
1	成功设置

功能描述：

设置 ISUP 中用户部分参数。

注意事项:

- ISUP_IAM_TMR=4 设置项只有在 SynCTI Ver. 5.1.0.0 或更高版本上才支持, ISUP_PhoNumRELEx=5 设置项只有在 SynCTI Ver. 5.3.2.0 或更高版本上才支持;
- 当 nType 为 ISUP_PhoNumREL 时, 对 ch 通道调用 [SsmSetIsupFlag](#) 设置用户部分参数仅对调用后的本次呼叫起作用。nType 为其它类型时设置的参数有效期直到重启程序重新加载配置文件为止。

相关信息:

驱动程序版本要求	SynCTI Ver. 4.7.1.5 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetIsupFlag](#)

2.3.10.5 SsmGetIsupFlag

获取 ISUP 中用户部分参数设置。

函数原型:

```
int SsmGetIsupFlag(int ch, int nType, DWORD *pd)
```

参数说明:

ch	通道号
nType	选择参数类型, 详细内容请参见函数 SsmSetIsupFlag 的说明
*pd	参数的值, 详细内容请参见函数 SsmSetIsupFlag 的说明

返回值:

-1	失败
1	调用成功

功能描述:

获得 ISUP 中用户部分参数的设置值。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 4.7.1.5 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetIsupFlag](#)

2.3.10.6 SsmGetIsupUPPara

获取保存在驱动程序内部的 ISUP 原始消息。

函数原型:

```
int SsmGetIsupUPPara(int nBCh, WORD wMsuType, LPWORD pwMsuSize, PUCHAR pucRawMsu)
```

参数说明:

nBCh	通道编号
------	------

消息类型			
wMsuType	取值	shpa3api.h 中的宏	pucContent 参数对应的消息
	0x01	C_ISUP_IAM	初始地址消息 (IAM)
	0x06	C_ISUP_ACM	地址全消息 (ACM)
pwMsuSize	返回消息的实际长度, 单位为字节		
pucRawMsu	存放 ISUP 消息的缓冲区指针。缓冲区由应用程序分配, 长度必须大于或等于 281 字节		

返回值:

1	调用成功
-1	错误的通道编号
-2	调用失败, 参数 wMsuType 越界

功能描述:

获取保存在驱动程序内部的 ISUP 原始消息。更多信息请参见第 1 章中“[ISUP 通道的状态转移](#)”部分内容。

注意事项:

- 本函数只支持 ISUP 通道;
- 保存在驱动程序内部的 ISUP 消息在 ISUP 挂断后会被清空。

相关信息:

驱动程序版本要求	SynCTI Ver. 4.7.1.5 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetIsupUPPara](#)

2.3.10.7 SsmSetIsupUPPara

向驱动程序提交一个自行构建的 ISUP 消息。

函数原型:

```
int SsmSetIsupUPPara(int nBCh, WORD wMsuType, LPWORD pwMsuSize, PUCHAR pucRawMsu)
```

参数说明:

nBCh	通道编号		
wMsuType	消息类型		
wMsuType	取值	shpa3api.h 中的宏	pucContent 参数对应的消息
	0x01	C_ISUP_IAM	初始地址消息 (IAM)
	0x06	C_ISUP_ACM	地址全消息 (ACM)
pwMsuSize	ISUP 消息的长度, 包括 SIO 字段, 单位为字节, 最大不能超过 281 个字节		
pucRawMsu	消息数据的首地址指针		

返回值:

1	成功
-1	错误的通道编号
-2	调用失败, 参数 wMsuType 越界
-3	校验失败, 消息内容不符合 ISUP 协议规则

功能描述:

向驱动程序提交一个自行构建的 ISUP 消息。

注意事项:

- 本函数只支持 ISUP 通道;
- 如果 wMsuType 为 C_ISUP_IAM, 本函数必须在 [SsmAutoDial](#) 或 [SsmAutoDialEx](#) 前调用;
- 如果 wMsuType 为 C_ISUP_ACM, 应用程序提交的 ACM 消息只有在通道处于“挂起”状态, 并且挂起

原因为 PEND_WaitBckStpMsg 时才可以被发送给对端交换机。更多信息请参见第 1 章中 “[ISUP 通道的状态转移](#)” 部分内容；

- 驱动程序在发送应用程序提交的 ISUP 消息时，会自动覆盖消息中的 SIO、DPC、OPC、SLC、CIC 字段。如果 wMsuType 为 C_ISUP_IAM，驱动程序还会用 [SsmAutoDial](#) 或 [SsmAutoDialEx](#) 中的被叫号码覆盖 pucRawMsu 中的相应字段；
- 本函数设置的参数的有效周期仅对本次有效。

使用范例：

```
WORD acmmsglen=18;  
UCHAR acmmsg[20]={0,0,0,0,0,0,0,0,0,6,22,20,1,41,1,3,0};  
其中 acmmsg[13]必须为 1。  
SsmSetIsupUPPara(channel, C_ISUP_ACN, &acmmsglen, acmmsg);
```

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.1.5 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmAutoDial](#), [SsmAutoDialEx](#), [SsmGetIsupUPPara](#), [SsmSendIsupMsg](#)

2.3.10.8 SsmSendIsupMsg

向对端交换机发送一条 ISUP 消息。

函数原型：

```
int SsmSendIsupMsg(int nBCh, WORD wMsuType)
```

参数说明：

nBCh	通道编号						
wMsuType	消息类型						
	<table border="1"><tr><td>取值</td><td>shpa3api.h 中的宏</td><td>消息类型</td></tr><tr><td>0x06</td><td>C_ISUP_ACN</td><td>地址全消息 (ACM)</td></tr></table>	取值	shpa3api.h 中的宏	消息类型	0x06	C_ISUP_ACN	地址全消息 (ACM)
取值	shpa3api.h 中的宏	消息类型					
0x06	C_ISUP_ACN	地址全消息 (ACM)					

返回值：

1	成功
-1	错误的通道编号
-2	调用失败，参数 wMsuType 越界
-3	发送消息失败

功能描述：

向对端交换机发送一条 ISUP 消息。ISUP 消息必须预先通过调用 [SsmSetIsupUPPara](#) 函数提交给驱动程序。

注意事项：

- 本函数只支持 ISUP 通道；
- 只有在通道处于 S_CALL_PENDING 状态，并且挂起原因为 PEND_WaitBckStpMsg 时，才允许调用本函数。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.7.1.5 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmSetIsupUPPara](#)

2.3.10.9 SsmGetRedirectionInfReason

来话呼叫时，对于 ISUP 通道，从收到的 IAM 消息里取出转呼的原因值；对于 ISDN 通道，从收到的 Setup 消息里取出转呼的原因值。

函数原型：

```
int SsmGetRedirectionInfReason(int ch)
```

参数说明：

ch	通道编号
----	------

返回值：

-1	失败
	转呼的原因值： 0: 不知/不可获得 1: 用户忙 2: 无应答 3: 无条件 4: 通知时偏转 5: 偏转立即响应 6: 移动用户不可达到
≥0	

功能描述：

来话呼叫时，对于 ISUP 通道，从收到的 IAM 消息里取出转呼的原因值；对于 ISDN 通道，从收到的 Setup 消息里取出信息单元 Facility (0x1c) 或者信息单元 Redirecting Number (0x74) 的转呼原因值，对于获取哪种信息单元的转呼原因值可以由配置项 [GetRedirectionReason](#) 决定。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 5.3.0.3 或更高, 对于获取信息单元 Facility(0x1c)的转呼原因值要求 SynCTI Ver. 5.3.2.3 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数： [SsmGetIsupUPPara](#)

2.3.10.10 SsmGetRedirectionInfNum

获取 ISDN 通道 redirection 号码。

函数原型：

```
int WINAPI SsmGetRedirectionInfNum(int ch, LPSTR szRedirectNum)
```

参数说明：

ch	通道编号
szRedirectNum	存放 redirection 号码的缓冲区指针，由应用程序分配存储空间，不能少于 32 个字节

返回值：

≥0	获取成功，返回 redirection 号码的长度（字节数）
-1	调用失败

功能描述：

获取 ISDN 通道 redirection 号码。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.0.3 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: 无

2.3.10.11 SsmIsHaveCpg

请参见 [SsmGetCpg](#)

2.3.10.12 SsmGetCpg

SsmIsHaveCpg 查询 ACM 消息后是否跟随了 CPG 消息, 如果有, SsmGetCpg 用于取出保存在驱动程序内部的原始 CPG 消息。

函数原型:

```
int SsmIsHaveCpg(int ch)  
int SsmGetCpg(int ch, char* pszContent, int* piLength)
```

参数说明:

ch	通道编号
pszContent	返回 CPG 消息的内容。有关 CPG 消息的详细信息请参见 ISUP 协议的相关文档
piLength	返回 CPG 消息的长度

返回值:

-1	操作失败。失败原因可以通过调用函数 SsmGetLastErrMsg 获得
0	SsmIsHaveCpg: 没有 CPG 消息 SsmGetCpg: 调用成功
1	SsmIsHaveCpg: 有 CPG 消息。只有当 SsmIsHaveCpg 返回 1 时, 调用 SsmGetCpg 才有意义

功能描述:

去话呼叫时, 对端交换机在发送了 ACM 消息后, 可能会继续发送 CPG 消息(呼叫进展)。驱动程序收到此 CPG 消息后, 会将其保存在内部缓冲区中。本函数用于查询 ACM 消息后是否跟随了 CPG 消息。

注意事项:

- 只有当通道处于 S_CALL_WAIT_REMOTE_PICKUP 状态时, 才可以调用本函数;
- 每当发生下列事件时, 驱动程序会自动清除保存在内部缓冲区中的 CPG 消息:
 - 通道迁移到 S_CALL_STANDBY 状态;
 - 收到对端交换机的 ACM 消息。
- 本函数只支持 ISUP 通道。

相关信息:

驱动程序版本要求	SynCTI Ver. 4.7.1.0 或更高
Header	shpa3api.h
Library	shp_a3.lib

DLL	shp_a3.dll
-----	------------

相关事件: [E_CHG_ChState](#), [E_RCV_Ss7IsupCpq](#)

2.3.10.13 SsmlsupGetUsr

检查并获取 USR 消息。

函数原型:

```
BOOL SsmlsupGetUsr(int *ch, PUCHAR pucData, PUCHAR ucLen)
```

参数说明:

ch	收到消息的通道号的指针
pucData	用于存放收到的 USR 消息中“用户至用户信息”的数据的缓冲区，需要 130 字节
ucLen	pucData 的长度

返回值:

0	取到新的数据
其他	未取到新的数据

功能描述:

检查并获取 USR 消息。执行本函数后，新的消息将从系统中清除。

注意事项: 无。

相关信息:

驱动程序版本要求	SynCTI Ver. 4.7.3.1 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关事件: [E_RCV_Ss7IsupUtuinf](#)

2.3.10.14 SsmlsupSendUsr

发送 USR 消息。

函数原型:

```
BOOL SsmlsupSendUsr(int ch, PUCHAR pucData, UCHAR ucLen)
```

参数说明:

ch	通道编号
pucData	作为 USR 消息中的“用户至用户信息”发送的数据
ucLen	pucData 的长度，协议中规定有效值为 2~130

返回值:

0	调用成功
其他	调用失败

功能描述:

发送 USR 消息。

注意事项: 无。

相关信息:

驱动程序版本要求	SynCTI Ver. 4.7.3.1 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关事件:

2.3.11 ISDN 通道专用函数

本节提供的函数专用于 ISDN 通道的呼叫。

2.3.11.1 设置 Setup 消息的参数

2.3.11.1.1 SsmISDNSetDialSubAddr

请参见 [SsmISDNSetDialSubAddrEx](#)

2.3.11.1.2 SsmISDNSetDialSubAddrEx

去话呼叫时，设置 setup 消息中的被叫方子地址。

函数原型:

```
int SsmISDNSetDialSubAddr(int ch, LPSTR lpSubAddress)  
int WINAPI SsmISDNSetDialSubAddrEx(int ch, LPBYTE lpSubAddressEx, UCHAR ucSubAddressLen)
```

参数说明:

ch	通道编号
lpSubAddress	被叫的子地址，由字符‘0’~‘9’组成，最大长度为 20 个字节，由应用程序分配存储空间
lpSubAddressEx	被叫的子地址，包含子地址类型字节和子地址信息内容的缓冲区
ucSubAddressLen	子地址类型字节和子地址信息内容缓冲区的总长度

返回值:

0	调用成功
-1	调用失败，函数 SsmGetLastErrMsg 可以获取失败原因

功能描述:

去话呼叫时，设置 setup 消息中的被叫方子地址。

注意事项:

- 本函数必须在 [SsmAutoDial](#) 函数之前调用；
- 本函数设置的参数的有效周期直到重新加载配置文件为止。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmAutoDial](#)

2.3.11.1.3 SsmISDNSetTxSubAddr

请参见 [SsmISDNSetTxSubAddrEx](#)

2.3.11.1.4 SsmISDNSetTxSubAddrEx

去话呼叫时，设置 setup 消息中的主叫子地址。

函数原型：

```
int SsmISDNSetTxSubAddr(int ch, LPSTR lpSubAddress)
```

```
int WINAPI SsmISDNSetTxSubAddrEx(int ch, LPBYTE lpSubAddressEx, UCHAR ucSubAddressLen)
```

参数说明：

ch	通道编号
lpSubAddress	主叫用户子地址，由字符'0'~'9'组成，最大长度为 20 个字节，由应用程序分配存储空间
lpSubAddressEx	主叫用户子地址，包含子地址类型字节和子地址信息内容的缓冲区
ucSubAddressLen	子地址类型字节和子地址信息内容缓冲区的总长度

返回值：

0	设置成功
-1	调用失败，函数 SsmGetLastErrMsg 可以获取失败原因

功能描述：

去话呼叫时，设置 setup 消息中的主叫用户子地址信息。

注意事项：

- 本函数必须在 [SsmAutoDial](#) 函数之前调用；
- 本函数设置的参数的有效周期仅对本次有效。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数： [SsmAutoDial](#), [SsmISDNGetTxCallerSubAddr](#)

2.3.11.1.5 SsmISDNSetCallerIdPresent

设置信令消息中主叫号码是否允许显示字段。

函数原型：

```
int SsmISDNSetCallerIdPresent(int ch, UCHAR ucPresentation)
```

参数说明：

ch	通道编号
ucPresentation	=0: 不允许显示 =1: 允许显示，用户提供 =2: 允许显示，网络提供

返回值：

0	调用成功
-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得

功能描述:

设置信令消息中的主叫号码是否允许显示字段。

注意事项:

- 本函数设置的参数的有效周期直到重启程序重新加载配置文件来初始化链路时为止。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.3.11.1.6 SsmSetIsdnParameter

向 SETUP 消息添加任意字段或者动态发送自定义非 SETUP 消息。

函数原型:

```
int WINAPI SsmSetIsdnParameter(int nBCh, UCHAR ucMsgTypeCode, UCHAR ucParamTypeCode, PUCHAR pucContent, WORD wNumberOfBytesToWrite)
```

参数说明:

nBCh	通道编号
ucMsgTypeCode	消息类型, 支持 SETUP 消息 (0x05) 和其他自定义消息
ucParamTypeCode	字段类型编码
pucContent	字段内容缓冲区 (字段类型编码和内容长度除外)
wNumberOfBytesToWrite	字段内容的实际长度

返回值:

0	调用成功
-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得

功能描述:

向 SETUP 消息里添加任意字段或者动态发送自定义非 SETUP 消息。

注意事项:

- 对于 SETUP 消息, 本函数设置的参数的有效周期直到 AutoDial 操作开始。
- 对于其他自定义消息, 本函数向消息中添加自定义字段并一次性发送。
- 可以多次调用本函数添加多个自定义字段。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.2.2 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:**代码范例:**

```
//发送一条自定义非 SETUP 消息: 08 02 80 01 7b 70 06 81 54 32 30 30 30
```

```
char msu [100]="";
msu [0]=0x81 ;
msu [1]=0x54 ;
msu [2]=0x32 ;
msu [3]=0x30 ;
msu [4]=0x30 ;
msu [5]=0x30;
SsmSetIsdnParameter( 0, 0x7b, 0x70, msu, 6);
```

2.3.11.1.7 SsmSetIsdnParameterA

向 SETUP 消息添加任意字段或者动态发送自定义非 SETUP 消息。

函数原型:

```
int      WINAPI      SsmSetIsdnParameterA(int      nBCh,      UCHAR      ucMsgTypeCode,      PUCHAR
pucParamTypeCode,PUCHAR pucContent, PUCHAR pucNumberOfBytesToWrite,WORD wNumberOfUnit)
```

参数说明:

nBCh	通道编号
ucMsgTypeCode	消息类型, 支持 SETUP 消息 (0x05) 和其他自定义消息
pucParamTypeCode	每个信息单元类型编码
pucContent	每个信息单元内容缓冲区 (字段类型编码和内容长度除外)
pucNumberOfBytesTo Write	每个信息单元内容的实际长度
wNumberOfUnit	需要添加的信息单元数目

返回值:

0	调用成功
-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得

功能描述:

向 SETUP 消息里添加任意字段或者动态发送自定义非 SETUP 消息。

注意事项:

- 对于 SETUP 消息, 本函数设置的参数的有效周期直到 AutoDial 操作开始。
- 对于其他自定义消息, 本函数可向消息中添加多个自定义字段并一次性发送。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.3.1 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

代码范例:

```
//一次性发送包含多个信息单元的非 SETUP 消息: 08 02 80 01 7d 08 03 82 e4 6c 14 01 01
解析:
08 //协议鉴别语: isdn 消息
02 80 01 //02 呼叫参考长度, 80 01 呼叫参考内容
7d //消息类型: 状态
08 03 82 e4 6c //08 第一个信息单元: 原因, 03 长度, 82 e4 6c 内容
14 01 01 //14 第二个信息单元: 呼叫状态, 01 长度, 01 内容
```

```
char msu [100]="";
msu [0]=0x82;//第一个信息单元内容
```

```
msu [1]=0xe4 ;
msu [2]=0x6c ;
msu [3]=0x01 ;//第二个信息单元内容

char ParamType[50]="";//信息单元类型编码
ParamType[0]=0x08;// 第一个信息单元
ParamType[1]=0x14;// 第二个信息单元

char NumberOfBytesToWrite [50]="";//信息单元实际长度
NumberOfBytesToWrite [0]=0x03;// 第一个信息单元长度
NumberOfBytesToWrite [1]=0x01;// 第二个信息单元长度

SsmSetIsdnParameterA (0, 0x7d, ParamType, msu, NumberOfBytesToWrite, 2);
```

2.3.11.2 获取来话呼叫信息

2.3.11.2.1 获取子地址信息

2.3.11.2.1.1 SsmISDNGetSubAddr

获取被叫用户子地址信息。

函数原型:

```
int SsmISDNGetSubAddr(int ch, LPSTR IpSubAddress)
```

参数说明:

ch	通道编号
IpSubAddress	存放被叫用户子地址信息的缓冲区，由应用程序分配存储空间，长度不能少于 21 字节

返回值:

≥0	IpSubAddress 中字符的实际长度，单位为字节
-1	调用失败，函数 SsmGetLastErrMsg 可以获取失败原因

功能描述:

来话呼叫时，来自对端交换机的消息中包含了被叫用户子地址信息。本函数获取从消息中提取出来的 ASCII 字符形式的被叫用户子地址信息。

注意事项: 无

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmISDNGetCallerSubAddr](#)

2.3.11.2.1.2 SsmISDNGetCallerSubAddr

获取主叫用户子地址信息。

函数原型:

```
int SsmISDNGetCallerSubAddr(int ch, LPSTR IpSubAddress)
```

参数说明:

ch	通道编号
lpSubAddress	存放主叫用户子地址信息的缓冲区，由应用程序分配存储空间，长度不能少于 21 字节

返回值:

≥0	主叫子地址长度
-1	调用失败，函数 SsmGetLastErrMsg 可以获取失败原因

功能描述:

来话呼叫时，来自对端交换机的消息中包含了主叫用户子地址信息。本函数获取从消息中提取出来的 ASCII 字符形式的主叫用户子地址信息。

注意事项: 无**相关信息:**

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmISDNGetSubAddr](#)

2.3.11.2.2 获得用户自定义主叫号码

2.3.11.2.2.1 Ssm GetUserCallerId

获取用户自定义主叫号码。

函数原型:

```
int Ssm GetUserCallerId(int ch, LPSTR szCallerId)
```

参数说明:

ch	通道编号
szCallerId	放置呼入时用户自定义主叫号码的缓冲区。最大长度 21 字节

返回值:

-1	调用失败，函数 SsmGetLastErrMsg 可以获取失败原因
≥0	自定义主叫号码长度

功能描述:

获取用户自定义主叫号码。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver. 4.7.3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.3.11.2.3 获取 SETUP 消息指定字段

2.3.11.2.3.1 SsmGetIsdnParameter

获取 SETUP 消息指定信令单元内容。

函数原型:

```
int WINAPI SsmGetIsdnParameter(int nBCh, UCHAR ucMsgTypeCode, UCHAR ucParamTypeCode, UCHAR ucParamIndex, PUCHAR pucContent, WORD wNumberOfBytesToWrite, LPWORD lpNumberOfBytesWritten)
```

参数说明:

nBCh	通道编号
ucMsgTypeCode	目前仅支持 SETUP 消息 (0x05)
ucParamTypeCode	指定信令单元编码
ucParamIndex	指定信令单元的序号，一般为 0。如果有多个同类型信令单元，以此为序号指定获取其中的一个信令单元
pucContent	应用程序申请的缓冲区，用于存放信令单元的内容（信令单元编码和信令单元内容长度字段除外）
wNumberOfBytesToWrite	应用程序申请的缓冲大小，长度不小于 300 字节
lpNumberOfBytesWritten	实际获取到的信令单元内容长度

返回值:

-1	调用失败，函数 SsmGetLastErrMsg 可以获取失败原因
0	获取成功

功能描述:

获取 SETUP 消息指定信令单元内容。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.1.4 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.3.11.3 设置挂机原因

2.3.11.3.1 SsmISDNSetHangupRzn

设置挂机原因。

函数原型:

```
int SsmISDNSetHangupRzn(int ch, int nReason)
```

参数说明:

ch	通道编号
	挂机原因, 取值范围: 0: 正常挂机 1: 用户忙 2: 空号 3: 拒绝本次呼叫
nReason	

返回值:

0	调用成功
-1	调用失败, 函数 SsmGetLastErrMsg 可以获取失败原因

功能描述:

设定挂机原因。

注意事项:

- 本函数必须在 [SsmHangup](#) 函数之前调用;
- 本函数设置的参数的有效周期直到重新加载配置文件为止。

相关信息:

驱动程序版本要求	SynCTI Ver. 4.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetPendingReason](#)

2.3.11.4 其它函数

2.3.11.4.1 SsmISDNGetDisplayMsg

获取 ISDN 通道的显示信息。

函数原型:

```
int SsmISDNGetDisplayMsg(int ch, LPSTR lpDispMsg)
```

参数说明:

ch	通道编号
lpDispMsg	存放显示信息的缓冲区, 最大长度为 100 个字节

返回值:

≥0	显示信息长度
-1	调用失败, 函数 SsmGetLastErrMsg 可以获取失败原因

功能描述:

获取 ISDN 通道的显示信息。显示信息为 ASCII 形式的字符串, 通常由网络侧发送给用户侧, 可能包含问候语或其他的附加信息。

注意事项: 无

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib

DLL

shp_a3.dll

相关函数: 无

2.3.11.4.2 SsmISDNGetTxCallerSubAddr

获取本端设置的主叫方子地址信息。

函数原型:

```
int SsmISDNGetTxCallerSubAddr(int ch, LPSTR lpSubAddress)
```

参数说明:

ch	通道编号
lpSubAddress	放置呼出时主叫子地址信息的缓冲区。最大长度 21 字节

返回值:

≥0	主叫子地址长度
-1	调用失败, 函数 SsmGetLastErrMsg 可以获取失败原因

功能描述:

获取本端设置的主叫方子地址信息。

注意事项:

- 本函数只适用于去话呼叫。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmISDNGetSubAddr](#)

2.3.11.4.3 SsmSetNumType

设置指定通道的主 / 被叫号码类型。

函数原型:

```
int SsmSetNumType(int ch, int nNumClass, int nNumType)
```

参数说明:

ch	通道编号
nNumClass	主叫 / 被叫号码: 1: 主叫号码 2: 被叫号码
nNumType	号码类型: 0x00: 未知 0x10: 国际号码 0x20: 国内号码 0x30: 网络特定号码 0x40: 指定号码或用户号码 其他值: 保留

返回值:

-1	操作失败
----	------

0	操作成功
---	------

功能描述:

设置指定通道主 / 被叫号码类型。

注意事项:

- 本函数只适用于 ISDN 通道。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.0.2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetNumType](#), [NetCallingNoSet](#)

2.3.11.4.4 SsmGetNumType

获取指定通道的主 / 被叫号码类型。

函数原型:

```
int SsmGetNumType(int ch, int nNumClass, int* pNumType)
```

参数说明:

ch	通道编号
	主叫 / 被叫号码:
nNumClass	1: 主叫号码 2: 被叫号码
	号码类型: 0x00: 未知 0x10: 国际号码 0x20: 国内号码 0x30: 网络特定号码 0x40: 指定号码或用户号码 其他值: 保留
pNumType	

返回值:

-1	操作失败
0	操作成功

功能描述:

获取指定通道主 / 被叫号码类型。

注意事项:

- 本函数只适用于 ISDN 通道。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.0.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetNumType](#), [NetCallingNoSet](#)

2.3.11.4.5 SsmSetCharge

设置相应通道的通话是计费的还是免费的。

函数原型:

```
int SsmSetCharge(int ch,int ChargeFlag)
```

参数说明:

ch	通道编号
ChargeFlag	0: 免费 1: 计费 其他值保留

返回值:

-1	调用失败
0	调用成功

功能描述:

设置相应通道的通话是计费的还是免费的。

注意事项:

- 本函数只支持 ISDN 通道，并且只针对 PBX 的某些特定机型；
- 本函数可以在呼出时、回铃或者通话状态下调用；
- 本函数可以在呼入时或通话状态下调用。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.0.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.4 有关信号音发生器的函数（CTI 系列）

2.4.1 设置信号音发生器的工作参数

2.4.1.1 SsmSetTxTonePara

本函数设置信号音发生器产生的信号音的频率和音量。

函数原型:

```
int SsmSetTxTonePara(int ch, int nFreq1, int nVolume1, int nFreq2, int nVolume2)
```

参数说明:

ch	通道编号
nFreq1	设置音频的第 1 个频率，单位为 Hz，取值范围：300~3400，缺省值为 450
nVolume1	设置音频的第 1 个音量，取值范围：-7~+6。大于 0 表示音量提升，小于 0 表示音量衰减，-7 表示关闭第 1 音频。缺省值为 0。取值乘以 3 即为分贝值
nFreq2	设置音频的第 2 个频率，单位为 Hz，取值范围：300~3400，缺省值为 0，表示不使用本频率，只发送单音频信号音

nVolume2	设置音频的第 2 个音量，取值范围：-7~+6。大于 0 表示音量提升，小于 0 表示音量衰减，-7 表示关闭第 2 音频。缺省值为-7（不使用第 2 音频）。取值乘以 3 即为分贝值
----------	----------------------------------------------------------------------------------------------

返回值：

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述：

本函数设置信号音发生器产生的信号音的频率和音量。

注意事项：

- 配置项 [DefaultSendToneFrequency](#) 和 [DefaultSendToneVolume](#) 可以实现相同功能，缺省的信号音发生器为 450Hz 单音频信号音。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmSendTone](#)

2.4.1.2 SsmQueryOpSendTone

查询通道是否具有信号音发生器。

函数原型：

```
int SsmQueryOpSendTone (int ch)
```

参数说明：

ch	通道编号
----	------

返回值：

-1	调用失败
0	没有
1	有

功能描述：

查询通道是否具有信号音发生器。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.4.1.3 SsmGetTxTonePara

获取信号音发生器中的频率参数和音量。

函数原型：

```
int SsmGetTxTonePara(int ch, int* nFreq1, int* nVolume1, int* nFreq2, int* nVolume2)
```

参数说明:

ch	通道编号
nFreq1	返回信号音参数中第 1 音频的频率, 单位: Hz
nVolume1	返回信号音参数中第 1 音频的音量, 取值范围: -7~+6, 大于 0 表示音量提升, 小于 0 表示音量衰减, -7 表示不使用该频率。取值乘以 3 即为分贝值
nFreq2	返回信号音参数中第 2 音频的频率, 单位为 Hz
nVolume2	返回信号音参数中第 2 音频的音量, 取值范围: -7~+6, 大于 0 表示音量提升, 小于 0 表示音量衰减, -7 表示不使用该频率。取值乘以 3 即为分贝值

返回值:

-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

取得本通道上送信号音时的频率和音量。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver. 2.0.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetTxTonePara](#)

2.4.2 发送信号音

2.4.2.1 SsmSendTone

请参见 [SsmSendToneEx](#)

2.4.2.2 SsmSendToneEx

启动信号音发生器, 在线路上产生信号音。[SsmSendTone](#) 用于发送驱动程序预定义类型的信号音, [SsmSendToneEx](#) 可以指定信号音的高电平保持时间和低电平保持时间。

函数原型:

```
int SsmSendTone(int ch, int nToneType)  
int SsmSendToneEx(int ch, DWORD dwOnTime, DWORD dwOffTime)
```

参数说明:

ch	通道编号
nToneType	信号音类型, 取值包括: 0: 拨号音 1: 忙音 2: 回铃音 3: 催挂音
dwOnTime	信号音高电平持续时间, 单位为毫秒, 必须为 8 的整数倍

dwOffTime	信号音低电平持续时间，单位为毫秒，必须为 8 的整数倍
-----------	-----------------------------

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

在指定通道上产生单音频或双音频信号音。

注意事项:

- 信号音的频率特征和音量可以通过函数 [SsmSetTxTonePara](#) 设置或配置项 [DefaultSendToneFrequency](#)、[DefaultSendToneVolume](#) 进行设置；
- 信号音发生器一旦启动后，线路上将持续出现指定的信号音，直到应用程序调用 [SsmStopSendTone](#)。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmStopSendTone](#), [SsmSetTxTonePara](#), [SsmChkSendTone](#)

2.4.2.3 SsmStopSendTone

关闭信号音发生器。

函数原型:

```
int SsmStopSendTone(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

关闭信号音发生器。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSendTone](#), [SsmSendToneEx](#)

2.4.2.4 SsmChkSendTone

查询信号音发生器是否启动，以及正在发送的信号音类型。

函数原型:

```
int SsmChkSendTone(int ch, int* pnToneType)
```

参数说明:

ch	通道编号
	返回通道上正在发送的信号音类型: 0: 拨号音 1: 忙音 2: 回铃音 3: 催挂音 4: 自定义信号音 (检测到函数 SsmSendToneEx 产生的信号音) 5: 无信号音
pnToneType	

返回值:

-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
0	通道不在发送信号音
1	通道正在发送信号音

功能描述:

查询信号音发生器是否启动, 以及正在发送的信号音类型。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver. 2.0.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSendTone](#), [SsmSendToneEx](#)

2.5 有关信号音检测器的函数

2.5.1 有关信号音检测器的常用函数

2.5.1.1 设置工作状态

2.5.1.1.1 SsmStartToneAnalyze

启动信号音检测器。

函数原型:

```
int SsmStartToneAnalyze(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

启动信号音检测器。详细内容请参见第1章的[信号音检测器](#)部分。

注意事项:

- 本函数在默认方式下只适用于模拟外线通道、磁石通道和录音通道；
- 对于数字卡/IP卡，如要手动开启或关闭信号音分析，需先在 ShConfig.ini 文件的[BoardId=x]节中做如下配置：DefaultToneCheckState=1，然后才可调用函数[SsmStartToneAnalyze/SsmCloseToneAnalyze](#)开启或关闭信号音检测器。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmCloseToneAnalyze](#), [SsmStart2ndToneAnalyzer](#)

2.5.1.1.2 SsmCloseToneAnalyze

关闭信号音分析器。

函数原型:

```
int SsmCloseToneAnalyze (int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

关闭信号音检测器。详细内容请参见第1章的[信号音检测器](#)部分。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmStartToneAnalyze](#)

2.5.1.1.3 SsmQueryOpToneAnalyze

查询通道是否具有信号音检测器。

函数原型:

```
int SsmQueryOpToneAnalyze(int ch)
```

参数说明:

ch	通道编号
返回值:	
-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
0	无
1	有

功能描述:

查询通道是否具有信号音检测器。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.5.1.2 获得检测结果

2.5.1.2.1 SsmGetToneAnalyzeResult

获取信号音检测器的检测结果。

函数原型:

```
int SsmGetToneAnalyzeResult (int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
0	信号音分析正在进行
1	检测到拨号音
2	检测到忙音
3	检测到回铃音
4	检测到回铃音后, 线路保持静默
5	检测到无声
6	检测到语音应答信号。对于模拟电话线, 通常表示被叫用户是否应答(摘机)
7	检测到具有 F1 频率特征的单音频信号音。F1 的参数由函数 SsmSetVoiceFxPara 设置, 通常用于检测传真机的信号音
8	检测到具有 F2 频率特征的单音频信号音。F2 频率参数由函数 SsmSetVoiceFxPara 设置, 通常用于检测传真机的信号音
9	检测到用户自定义的信号音类型
n>9	检测到用户自定义的特殊信号音(如 SIT 信号音), 详情请参阅“ 增强信号音检测器 ”一节

功能描述:

获取信号音检测器的检测结果。

注意事项:

- 本函数在默认方式下只适用于模拟外线通道、磁石通道和录音通道；
- 对于数字卡，如要进行信号音分析，需在 ShConfig.ini 文件中的[BoardId=x]节下对如下配置项进行配置：
DefaultToneCheckState=1。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmClearToneAnalyzeResult](#)

2.5.1.3 清除检测结果

2.5.1.3.1 SsmClearToneAnalyzeResult

清除信号音检测器的检测结果。

函数原型:

```
int SsmClearToneAnalyzeResult (int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
0	调用成功

功能描述:

清除信号音检测器的检测结果。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetToneAnalyzeResult](#)

2.5.2 有关 FFT 的函数

2.5.2.1 设置工作参数

2.5.2.1.1 SsmSetPeakFrqDetectBW

设置检测来话信号中的峰值频率时使用的带宽。

函数原型:

```
int SsmSetPeakFrqDetectBW(int ch, WORD nPeakBW)
```

参数说明:

ch	通道编号
nPeakBW	带宽, 单位为 Hz, 取值范围; 40~80, 缺省值为 80Hz

返回值:

-1	调用失败
0	调用成功

功能描述:

设置检测来话信号中的峰值频率时使用的带宽。更多信息请参见第 1 章中“[信号音检测器](#)”部分内容。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmQueryOpPeakFrqDetect](#), [SsmGetPeakFrqDetectBW](#)

2.5.2.2 获取工作参数

2.5.2.2.1 SsmQueryOpPeakFrqDetect

查询通道是否支持检测峰值频率的操作。

函数原型:

```
int SsmQueryOpPeakFrqDetect (int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
0	不支持
1	支持

功能描述:

查询通道是否支持检测峰值频率的操作。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.5.2.2.2 SsmGetPeakFrqDetectBW

获取检测峰值频率使用的带宽参数。

函数原型:

```
int SsmGetPeakFrqDetectBW(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
≥ 0	带宽的设置值

功能描述:

获取检测峰值频率使用的带宽参数。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmQueryOpPeakFrqDetect](#), [SsmSetPeakFrqDetectBW](#)

2.5.2.3 获取检测结果

2.5.2.3.1 SsmGetPeakFrq

获取来话信号中的峰值频率。

函数原型:

```
int SsmGetPeakFrq (int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
≥ 0	来话信号中的峰值频率

功能描述:

获取来话信号中的峰值频率。更多信息请参见第 1 章中“[信号音检测器](#)”部分内容。

注意事项:

- [E_CHG_PeakFrq](#) 事件可以实现相同的功能。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
----------	---------------------

Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmQueryOpPeakFrqDetect](#), [SsmSetPeakFrqDetectBW](#)

2.5.2.3.2 SsmGetPeakFrqEnergy

获取来话信号中的峰值频率部分具有的能量值。

函数原型:

```
long SsmGetPeakFrqEnergy(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
≥0	来话信号中的峰值频率部分具有的能量值

功能描述:

获取来话信号中的峰值频率部分具有的能量值。更多信息请参见第 1 章中“[信号音检测器](#)”部分内容。

注意事项:

- [E_CHG_PeakFrq](#) 事件可以实现相同的功能。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetPeakFrqDetectBW](#), [SsmGetPeakFrq](#)

2.5.2.3.3 SsmGetOverallEnergy

请参见 [SsmGetOverallEnergyAllCh](#)

2.5.2.3.4 SsmGetOverallEnergyAllCh

获取来话信号经过 FFT 变换后的全频能量值。SsmGetOverallEnergy 对单个通道进行操作，SsmGetOverallEnergyAllCh 可以同时获取多个通道的全频能量值。

函数原型:

```
int SsmGetOverallEnergy(int ch)
```

```
int SsmGetOverallEnergyAllCh (int nBeginCh, int nChNum, PDWORD pdwEnergyTable)
```

参数说明:

Ch	通道编号
nBeginCh	起始通道编号

nChNum	通道的数量
PdwEnergyTable	存放从 nBeginCh 开始的 nChNum 个通道的能量值，由应用程序分配空间

返回值:

函数名称	返回值	描述
SsmGetOverallEnergy	-1	调用失败
	≥0	全频能量值
SsmGetOverallEnergyAllCh	-1	调用失败
	0	调用成功

功能描述:

获取来话信号经过 FFT 变换后的全频能量值。

注意事项:

- 有关全频能量的单位与 DB 的换算方法请参见第 1 章中“[信号音检测器](#)”节中的“FFT 模块”部分。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetPeakFrqEnergy](#)

2.5.3 设置噪声滤波器的参数

2.5.3.1 SsmSetMinVocDtrEnergy

设置线路上的噪音判别门限。

函数原型:

```
int SsmSetMinVocDtrEnergy(int ch, DWORD dwMinVocDtrEnergy)
```

参数说明:

ch	通道编号
dwMinVocDtrEnergy	噪音判定门限，取值范围: ≥700，缺省值为 100000。本参数的单位与 DB 的换算方法请参见第 1 章中“ 信号音检测器 ”节中的“FFT 模块”部分

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

设置线路上的噪音判别门限。如果线路上来话信号的能量值小于 dwMinVocDtrEnergy 的设定值，驱动程序将其视为噪声。

注意事项:

- 配置项 [MinimumVoiceDetermineEnergy](#) 可以实现相同功能。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h

Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetMinVocDtrEnergy](#)

2.5.3.2 SsmGetMinVocDtrEnergy

获取驱动程序判定线路上出现的信号是声音还是噪声的能量门限值。

函数原型:

```
int SsmGetMinVocDtrEnergy(int ch, PDWORD pdwMinVocDtrEnergy)
```

参数说明:

ch	通道编号
pdwMinVocDtrEnergy	返回判定声音还是噪声的能量门限值

返回值:

-1	调用失败
0	调用成功

功能描述:

获取驱动程序判定线路上出现的信号是声音还是噪声的能量门限值。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetMinVocDtrEnergy](#)

2.5.4 有关呼叫进程音检测器的函数

2.5.4.1 设置第 2 呼叫进程音检测器的工作状态

2.5.4.1.1 SsmStart2ndToneAnalyzer

开启第 2 呼叫进程音检测器。

函数原型:

```
int SsmStart2ndToneAnalyzer(int ch, BOOL bEnable)
```

参数说明:

ch	通道编号
bEnable	第 2 组信号音分析器使能开关。=TRUE: 开启; =FALSE: 关闭

返回值:

-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

开启第 2 呼叫进程音检测器。

注意事项:

- 本函数设置的功能也可以通过配置项 [Enable2ndToneAnalyzer](#) 进行设置；
- 因为第 2 呼叫进程音检测器占用函数 [SsmSetVoiceFxPara](#) 使用的 FFT 分析器，因此当开启第 2 呼叫进程音检测器后，驱动程序不再检测 [SsmSetVoiceFxPara](#) 函数指定的特定信号音；
- 必须在开启第 1 呼叫进程音检测器的前提下，开启第 2 呼叫进程音检测器才有效。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGet2ndToneAnalyzerState](#)

2.5.4.1.2 SsmGet2ndToneAnalyzerState

查询第 2 呼叫进程音检测器的工作状态。

函数原型:

```
int SsmGet2ndToneAnalyzerState(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
0	关闭
1	开启

功能描述:

查询第 2 呼叫进程音检测器的工作状态。

注意事项:

- 本函数在默认方式下只适用于模拟外线通道、磁石通道和录音通道。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmStart2ndToneAnalyzer](#)

2.5.4.2 获取第 2 呼叫进程音检测器的结果

2.5.4.2.1 SsmGet2ndToneAnalyzeResult

获取第 2 呼叫进程音检测器的结果。

函数原型:

```
int SsmGet2ndToneAnalyzeResult(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
0	信号音分析正在进行
1	检测到拨号音
2	检测到忙音
3	检测到回铃音
4	检测到回铃音后，线路保持静默
5	检测到无声
6	检测到语音应答信号。对于模拟电话线，通常表示被叫用户是否应答（摘机）
7	检测到具有 F_1 频率特征的单音频信号音。 F_1 的参数由函数 SsmSetVoiceFxPara 设置，通常用于检测传真机的信号音
8	检测到具有 F_2 频率特征的单音频信号音。 F_2 的参数由函数 SsmSetVoiceFxPara 设置，通常用于检测传真机的信号音
9	检测到用户自定义的信号音类型
>9	检测到用户自定义的特殊信号音（如 SIT 信号音），更多信息请参见第 1 章中“ 增强信号音检测器 ”部分内容

功能描述:

获取第 2 呼叫进程音检测器的结果。

注意事项:

- 如打开第二组信号音分析器，发现两组信号音频率值一样，则驱动默认只有一组信号音分析器起作用。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmClear2ndToneAnalyzeResult](#)

2.5.4.2.2 SsmClear2ndToneAnalyzeResult

清除第 2 呼叫进程音检测器的结果。

函数原型:

```
int SsmClear2ndToneAnalyzeResult(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
0	调用成功

功能描述:

清除第 2 呼叫进程音检测器的结果。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGet2ndToneAnalyzeResult](#)

2.5.4.3 设置频率检测器的参数

2.5.4.3.1 SsmSetTonePara

请参见 [SsmSet2ndTonePara](#)

2.5.4.3.2 SsmSet2ndTonePara

设置呼叫进程音检测器中频率检测器的工作参数。SsmSetTonePara、SsmSet2ndTonePara 分别用于设置第 1、第 2 呼叫进程音检测器的频率参数。

函数原型:

```
int SsmSetTonePara(int ch, WORD wF1, WORD wBw1, WORD wF2, WORD wBw2, DWORD dwMinEnrgRatio)
int SsmSet2ndTonePara(int ch, WORD wF1, WORD wBw1, WORD wF2, WORD wBw2, DWORD dwMinEnrgRatio)
```

参数说明:

ch	通道编号
wF1	第 1 个中心频率, 单位为 Hz, 取值范围: 300~3400
wBw1	第 1 个中心频率的带宽, 单位为 Hz, 取值范围: 40~120
wF2	第 2 个中心频率, 单位为 Hz, 取值范围: 0 或者 300~3400
wBw2	第 2 个中心频率的带宽, 单位为 Hz, 取值范围: 0 或者 40~120
dwMinEnrgRatio	判定门限, 单位: %, 取值范围: 15~80

返回值:

-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

设置呼叫进程音检测器中频率检测器的工作参数。详细内容请参见第 1 章中[呼叫进程音检测器](#)部分。

注意事项:

- 配置项 [TonePara](#) 可以实现与 SsmSetTonePara 相同的功能, 缺省值为 450Hz 单音频信号音;
- 配置项 [2ndTonePara](#) 可以实现与 SsmSet2ndTonePara 相同的功能。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
----------	---------------------

Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetTonePara](#), [SsmGet2ndTonePara](#)

2.5.4.3.3 SsmGetTonePara

请参见 [SsmGet2ndTonePara](#)

2.5.4.3.4 SsmGet2ndTonePara

获取呼叫进程音检测器中频率检测器的工作参数。**SsmGetTonePara**、**SsmGet2ndTonePara** 分别用于获取第 1、第 2 呼叫进程音检测器的频率参数。

函数原型:

```
int SsmGetTonePara(int ch, PWORD pwF1, PWORD pwBw1, PWORD pwF2, PWORD pwBw2, PDWORD pdwMinEnrgRatio)  
int SsmGet2ndTonePara(int ch, PWORD pwF1, PWORD pwBw1, PWORD pwF2, PWORD pwBw2, PDWORD pdwMinEnrgRatio)
```

参数说明:

ch	通道编号
pwF1	返回第 1 音频的中心频率, 单位为 Hz
pwBw1	返回第 1 音频的带宽, 单位为 Hz
pwF2	返回第 2 音频的中心频率, 单位为 Hz
pwBw2	返回第 2 音频的带宽, 单位为 Hz
pdwMinEnrgRatio	返回设置的判决是否信号音的带内能量占全频能量的百分比门限

返回值:

-1	调用失败
0	调用成功

功能描述:

获取呼叫进程音检测器中频率检测器的工作参数。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetTonePara](#), [SsmSet2ndTonePara](#)

2.5.4.4 设置拨号音检测器的参数

2.5.4.4.1 SsmSetIsDialToneDtrTime

请参见 [SsmSet2ndIsDialToneDtrTime](#)

2.5.4.4.2 SsmSet2ndIsDialToneDtrTime

设置拨号音检测器的最小保持时间。SsmSetIsDialToneDtrTime、SsmSet2ndIsDialToneDtrTime 分别用于设置第 1、第 2 呼叫进程音检测器。

函数原型:

```
int SsmSetIsDialToneDtrTime(int ch, WORD wlsDialToneDtrTime)  
int SsmSet2ndIsDialToneDtrTime(int ch, WORD wlsDialToneDtrTime)
```

参数说明:

ch	通道编号
wlsDialToneDtrTime	拨号音的最小保持时间，单位为 ms，取值范围：≥1300ms

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

设置[拨号音检测器](#)的最小保持时间。

注意事项:

- 配置项 [IsDialToneDetermineTime](#) 可以实现与 SsmSetIsDialToneDtrTime 相同的功能；
- 配置项 [2ndIsDialToneDetermineTime](#) 可以实现与 SsmSet2ndIsDialToneDtrTime 相同的功能。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetIsDialToneDtrTime](#), [SsmGet2ndIsDialToneDtrTime](#)

2.5.4.4.3 SsmGetIsDialToneDtrTime

请参见 [SsmGet2ndIsDialToneDtrTime](#)

2.5.4.4.4 SsmGet2ndIsDialToneDtrTime

获取[拨号音检测器](#)的最小保持时间。SsmGetIsDialToneDtrTime、SsmGet2ndIsDialToneDtrTime 分别用于设置第 1、第 2 呼叫进程音检测器。

函数原型:

```
int SsmGetIsDialToneDtrTime(int ch, PWORD pwIsDialToneDtrTime)
int SsmGet2ndIsDialToneDtrTime(int ch, PWORD pwIsDialToneDtrTime)
```

参数说明:

ch	通道编号
pwIsDialToneDtrTime	返回最小保持时间, 单位为毫秒

返回值:

-1	调用失败
0	调用成功

功能描述:

获取拨号音检测器的最小保持时间。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetIsDialToneDtrTime](#), [SsmSet2ndIsDialToneDtrTime](#)

2.5.4.5 设置回铃音检测器的参数

2.5.4.5.1 SsmSetRingEchoTonePara

请参见 [SsmSet2ndRingEchoTonePara](#)

2.5.4.5.2 SsmSet2ndRingEchoTonePara

设置回铃音检测器的高电平保持时间和低电平保持时间。[SsmSetRingEchoTonePara](#) 适用于第 1 呼叫进程音检测器, [SsmSet2ndRingEchoTonePara](#) 适用于第 2 呼叫进程音检测器。

函数原型:

```
int SsmSetRingEchoTonePara(int ch, WORD wTon, WORD wToff)
int SsmSet2ndRingEchoTonePara(int ch, WORD wTon, WORD wToff)
```

参数说明:

ch	通道编号
wTon	高电平保持时间, 单位为毫秒, 取值范围: 500~2500
wToff	低电平保持时间, 单位为毫秒, 取值范围: 1500 ~ 6000

返回值:

-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

设置回铃音检测器的高电平保持时间和低电平保持时间。

注意事项:

- SsmSetRingEchoTonePara 设置的参数可以通过配置项 [RingEchoTonePara](#) 进行设置，缺省值：高电平保持时间为 1000 毫秒，低电平保持时间为 4000 毫秒；SsmSet2ndRingEchoTonePara 设置的参数可以通过配置项 [2ndRingEchoTonePara](#) 进行设置，缺省值：高电平保持时间为 1000 毫秒，低电平保持时间为 4000 毫秒。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmGetRingEchoTonePara](#), [SsmGet2ndRingEchoTonePara](#)

2.5.4.5.3 SsmGetRingEchoToneTime

获取回铃音的保持时间。

函数原型：

```
int SsmGetRingEchoToneTime(int ch)
```

参数说明：

ch	通道编号
----	------

返回值：

-1	调用失败
≥0	保持时间，单位：毫秒

功能描述：

获取回铃音的保持时间。

注意事项：

- 本函数只适用于模拟中继线通道

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.5.4.5.4 SsmGetRingEchoTonePara

请参见 [SsmGet2ndRingEchoTonePara](#)

2.5.4.5.5 SsmGet2ndRingEchoTonePara

获取回铃音检测器的高电平保持时间和低电平保持时间。SsmGetRingEchoTonePara、SsmGet2ndRingEchoTonePara 分别用于第 1、第 2 呼叫进程音检测器。

函数原型:

```
int SsmGetRingEchoTonePara(int ch, PWORD pwTon, PWORD pwToff)
int SsmGet2ndRingEchoTonePara(int ch, PWORD pwTon, PWORD pwToff)
```

参数说明:

ch	通道编号
pwTon	返回高电平保持时间, 单位为毫秒
pwToff	返回低电平保持时间, 单位为毫秒

返回值:

-1	调用失败
0	调用成功

功能描述:

获取回铃音检测器的高电平保持时间和低电平保持时间。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetRingEchoTonePara](#), [SsmSet2ndRingEchoTonePara](#)

2.5.4.6 设置忙音检测器的参数

2.5.4.6.1 SsmSetBusyTonePeriodEx

请参见 [SsmSet2ndBusyTonePeriod](#)

2.5.4.6.2 SsmSetBusyTonePeriod

请参见 [SsmSet2ndBusyTonePeriod](#)

2.5.4.6.3 SsmSet2ndBusyTonePeriod

设置呼叫进程音检测器中忙音检测器的忙音周期。SsmSetBusyTonePeriod 适用于第 1 呼叫进程音检测器，SsmSet2ndBusyTonePeriod 适用于第 2 呼叫进程音检测器，SsmSetBusyTonePeriodEx 函数可以同时适用于第 1、第 2 呼叫进程音检测器。

函数原型:

```
int SsmSetBusyTonePeriodEx(int ch, int nCptdNo, WORD wMax, PWORD pwPeriod)
int SsmSetBusyTonePeriod(int ch, WORD wPeriod)
int SsmSet2ndBusyTonePeriod(int ch, WORD wPeriod)
```

参数说明:

ch	通道编号
nCptdNo	=1: 设置第 1 组呼叫进程音检测器

	=2: 设置第 2 组呼叫进程音检测器
wMax	忙音周期的个数。取值范围: 1~4
pwPeriod	存放忙音周期参数的指针, 由应用程序提供存储空间
wPeriod	忙音周期, 单位为毫秒, 取值范围: 200~2000。本参数也可在配置文件中指定, 缺省值为 700ms (350ms/ON, 350ms/OFF)

返回值:

-1	调用失败
0	调用成功

功能描述:

设置呼叫进程音检测器中忙音检测器的忙音周期。SsmSetBusyTonePeriodEx是新版本的函数, 功能更加强大, 可以设置多达四个不同的忙音周期。更多信息请参见第 1 章中“忙音检测器”部分内容。

注意事项:

- 忙音周期也可以分别通过配置项 BusyTonePeriod 或 2ndBusyTonePeriod 进行设置;
- 建议使用 SsmSetBusyTonePeriodEx 函数。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetBusyTonePeriodEx](#), [SsmGetBusyTonePeriod](#), [SsmGet2ndBusyTonePeriod](#)

代码范例:

系统需要在第 1 呼叫进程音检测器上同时检测周期分别为 700 毫秒、800 毫秒、900 毫秒、1000 毫秒的 4 种忙音, 在第 2 呼叫进程音检测器上只检测周期为 700 毫秒的忙音, 实现代码为:

```
WORD BusyTonePeriod[4] = {700,800,900,1000};  
SsmSetBusyTonePeriodEx(ch, 1, 4, BusyTonePeriod);  
SsmSet2ndBusyTonePeriod(ch, 700);
```

2.5.4.6.4 SsmSetIsBusyToneDtrCnt

请参见 [SsmSet2ndIsBusyToneDtrCnt](#)

2.5.4.6.5 SsmSet2ndIsBusyToneDtrCnt

设置忙音周期的最少持续个数。SsmSetIsBusyToneDtrCnt 适用于第 1 呼叫进程音检测器, SsmSet2ndIsBusyToneDtrCnt 适用于第 2 呼叫进程音检测器。

函数原型:

```
int SsmSetIsBusyToneDtrCnt(int ch, WORD wlsBusyToneDtrCnt)  
int SsmSet2ndIsBusyToneDtrCnt(int ch, WORD wlsBusyToneDtrCnt)
```

参数说明:

ch	通道编号
wlsBusyToneDtrCnt	忙音周期的最小持续个数, 取值范围: 1~50

返回值:

-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
----	-----------------------------------------------------

0	调用成功
---	------

功能描述:

设置忙音周期的最少持续个数。更多信息请参见第 1 章中[忙音检测器](#)部分。

注意事项:

- 配置项 [IsBusyToneDetermineCount](#)、[2ndIsBusyToneDetermineCount](#) 也可实现同样功能。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetIsBusyToneDtrCnt](#), [SsmGet2ndIsBusyToneDtrCnt](#)

2.5.4.6.6 SsmGetBusyTonePeriodEx

请参见 [SsmGet2ndBusyTonePeriod](#)

2.5.4.6.7 SsmGetBusyTonePeriod

请参见 [SsmGet2ndBusyTonePeriod](#)

2.5.4.6.8 SsmGet2ndBusyTonePeriod

获取忙音判定周期的设定值。[SsmGetBusyTonePeriod](#) 适用于第 1 呼叫进程音检测器，[SsmGet2ndBusyTonePeriod](#) 适用于第 2 呼叫进程音检测器，[SsmGetBusyTonePeriodEx](#) 函数可以同时适用于第 1、第 2 呼叫进程音检测器。

函数原型:

```
int SsmGetBusyTonePeriodEx(int ch, int nCptdNo, PWORD pwPeriod)
int SsmGetBusyTonePeriod(int ch, PWORD pwBusyTonePeriod)
int SsmGet2ndBusyTonePeriod(int ch, PWORD pwBusyTonePeriod)
```

参数说明:

ch	通道编号
nCptdNo	=1: 设置第 1 组呼叫进程音检测器 =2: 设置第 2 组呼叫进程音检测器
pwPeriod	返回忙音周期设定值的指针，单位为毫秒，由应用程序提供存储空间，大小为 4 WORD
pwBusyTonePeriod	返回忙音周期设定值，单位为毫秒

返回值:

-1	调用失败
0	调用成功
>0	设置的忙音周期个数

功能描述:

获取忙音判定周期的设定值。更多信息请参见第 1 章中[“忙音检测器”](#)部分内容。

注意事项:

- 建议使用 SsmGetBusyTonePeriodEx 函数。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetBusyTonePeriodEx](#), [SsmSetBusyTonePeriod](#), [SsmSet2ndBusyTonePeriod](#)

2.5.4.6.9 SsmGetIsBusyToneDtrCnt

请参见 [SsmGet2ndIsBusyToneDtrCnt](#)

2.5.4.6.10 SsmGet2ndIsBusyToneDtrCnt

获取忙音周期最少持续个数的设定值。SsmGetIsBusyToneDtrCnt 适用于第 1 呼叫进程音检测器，SsmGet2ndIsBusyToneDtrCnt 适用于第 2 呼叫进程音检测器。

函数原型:

```
int SsmGetIsBusyToneDtrCnt(int ch, PWORD pwIsBusyToneDtrCnt)  
int SsmGet2ndIsBusyToneDtrCnt(int ch, PWORD pwIsBusyToneDtrCnt)
```

参数说明:

ch	通道编号
pwIsBusyToneDtrCnt	返回设置的忙音周期的最少持续个数

返回值:

-1	调用失败
0	调用成功

功能描述:

获取忙音周期最少持续个数的设定值。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetIsBusyToneDtrCnt](#), [SsmSet2ndIsBusyToneDtrCnt](#)

2.5.4.7 获取检测结果

2.5.4.7.1 SsmGetBusyToneLen

请参见 [SsmGet2ndBusyToneLen](#)

2.5.4.7.2 SsmGet2ndBusyToneLen

获取忙音的保持时间。SsmGetBusyToneLen 适用于第 1 呼叫进程音检测器，SsmGet2ndBusyToneLen 适用于第 2 呼叫进程音检测器。

函数原型：

```
int SsmGetBusyToneLen(int ch)  
int SsmGet2ndBusyToneLen(int ch)
```

参数说明：

ch	通道编号
----	------

返回值：

-1	调用失败
≥ 0	忙音持续的时间，单位为秒

功能描述：

获取忙音的保持时间。更多信息请参见第 1 章中“[忙音检测器](#)”部分内容。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmGetBusyToneCount](#)

2.5.4.7.3 SsmGetBusyToneCount

请参见 [SsmGet2ndBusyToneCount](#)

2.5.4.7.4 SsmGet2ndBusyToneCount

获取忙音周期的个数。SsmGetBusyToneCount 适用于第 1 呼叫进程音检测器，SsmGet2ndBusyToneCount 适用于第 2 呼叫进程音检测器。

函数原型：

```
int SsmGetBusyToneCount(int ch)  
int SsmGet2ndBusyToneCount(int ch)
```

参数说明：

ch	通道编号
----	------

返回值：

-1	调用失败
≥ 0	忙音周期持续的个数

功能描述:

获取忙音周期的个数。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetBusyToneLen](#)

2.5.4.8 获取采用背靠背忙音信号检测算法的检测结果

2.5.4.8.1 SsmGetBusyToneEx

获取忙音检测器采用背靠背忙音信号检测算法时的检测结果。

函数原型:

```
int SsmGetBusyToneEx(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
0	没有检测到忙音信号
1	检测到忙音信号

功能描述:

获取忙音检测器采用背靠背忙音信号检测算法时的检测结果。更多信息请参见第 1 章中“[忙音检测器](#)”部分内容。

注意事项:

- 使用 [E_CHG_BusyToneEx](#) 事件可以实现同样的功能。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmClearToneAnalyzeResult](#)

2.5.4.9 获取 Selcall Tone 信号检测算法的检测结果

2.5.4.9.1 SsmGetSelcallToneStr

获取保存在 Selcall Tone 检测器缓冲区的数据字符。

函数原型:

```
int SsmGetSelcallToneStr(int ch, LPSTR pszBuf)
```

参数说明:

ch	通道编号
pszBuf	存放接收 Selcall Tone 字符串的指针, 由应用程序分配存储空间, 必须大于 256 字节

返回值:

函数名称	返回值	描述
SsmGetDtmfStr	-1	调用失败
	≥ 0	Selcall Tone 检测器缓冲区中的字符个数
SsmGetDtmfStrA	NULL	调用失败

功能描述:

获取保存在 Selcall Tone 缓冲区的字符, 字符为标准的 ASCII 码。

注意事项:

- 本函数调用时, 驱动程序不会清空 Selcall Tone 检测器缓冲区, 需要调用 SsmClearSelcallToneBuf 清空驱动的接收缓冲区。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmClearSelcallToneBuf](#), [SsmGetSelcallToneLen](#)

相关事件: [E_CHG_RCV_SELCALL](#)

2.5.4.9.2 SsmGetSelcallToneLen

获取 Selcall Tone 检测器缓冲区中保存的字符个数。

函数原型:

```
int SsmGetSelcallToneLen(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败。失败原因可以通过调用函数 SsmGetLastErrMsg 获得
≥ 0	Selcall Tone 检测器缓冲区中保存的字符个数

功能描述:

获取 Selcall Tone 检测器缓冲区中保存的字符个数。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver. 5.3.3.0 或更高
Header	shpa3api.h

Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmClearSelcallToneBuf](#), [SsmGetSelcallToneStr](#)

相关事件: [E_CHG_RCV_SELCALL](#)

2.5.4.9.3 SsmClearSelcallToneBuf

清空驱动程序内部的 Selcall Tone 接收缓冲区。

函数原型:

```
Int SsmClearSelcallToneBuf(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败。失败原因可以通过调用函数 SsmGetLastErrMsg 获得
0	调用成功

功能描述:

清空驱动程序内部的 Selcall Tone 接收缓冲区。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetSelcallToneLen](#), [SsmGetSelcallToneStr](#)

2.5 有关传真进程音检测器的函数

2.5.5.1 设置工作参数

2.5.5.1.1 SsmSetVoiceFxPara

设置传真进程音检测器的工作参数。

函数原型:

```
int SsmSetVoiceFxPara(int ch, WORD wSelFx, WORD wFx, WORD wFxBW, DWORD dwIsVocFxRatio, WORD wIsVocFxDtrTime)
```

参数说明:

ch	通道编号
wSelFx	选择传真进程音检测器, 取值范围: 1: 设置第 1 传真进程音检测器的参数 2: 设置第 2 传真进程音检测器的参数
wFx	中心频率, 单位: Hz, 取值范围: 300~3400
wFxBW	带宽, 单位: Hz, 取值范围: 40~120
dwIsVocFxRatio	带内能量占全频能量的百分比的最小门限值, 单位: %, 取值范围: 15 ~ 80

wlsVocFxDrTime	最小保持时间, 单位: 毫秒, 取值范围: 16~1024, 且为 16 的整数倍
----------------	-------------------------------------------

返回值:

-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

设置传真进程音检测器的工作参数。更多信息请参见第 1 章中“[传真进程音检测器](#)”部分内容。

注意事项:

- 配置项 [VoiceFreqF1Para](#)、[VoiceFreqF2Para](#) 可以实现相同功能。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetVocFxFlag](#), [SsmGetVoiceFxPara](#)

2.5.5.1.2 SsmGetVoiceFxPara

获取传真进程音检测器的工作参数。

函数原型:

```
int SsmGetVoiceFxPara(int ch, WORD wSelFx, PWORD pwFx, PWORD pwFxBW, PDWORD pdwlsVocFxRatio,  
PWORD pwlsVocFxDrTime)
```

参数说明:

ch	通道编号
wSelFx	选择传真进程音检测器, 取值范围: 1: 设置第 1 传真进程音检测器的参数 2: 设置第 2 传真进程音检测器的参数
pwFx	返回中心频率, 单位: Hz
pwFxBW	返回带宽, 单位: Hz
pdwlsVocFxRatio	返回带内能量占全频能量的百分比的最小门限值, 单位: %
wlsVocFxDrTime	返回最小保持时间, 单位: 毫秒

返回值:

-1	调用失败
0	调用成功

功能描述:

获取传真进程音检测器的工作参数。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetVoiceFxPara](#)

2.5.5.2 获取检测结果

2.5.5.2.1 SsmGetVocFxFlag

获取传真进程音检测器的检测结果。

函数原型:

```
int SsmGetVocFxFlag(int ch, int nSelFx, BOOL bClear)
```

参数说明:

ch	通道编号
nSelFx	选择检测的声音频率。 =1: 选择 F1; =2: 选择 F2
bClear	指示驱动程序在读取指定标志后是否清除检测结果。 =TRUE: 清除检测结果; =FALSE: 不清除检测结果

返回值:

-1	调用失败
0	没有发现指定频率的单音频声音
1	发现指定频率的单音频声音

功能描述:

获取传真进程音检测器的检测结果。更多信息请参见第 1 章中“[传真进程音检测器](#)”部分内容。

注意事项:

- 建议使用事件 [E_CHG_VocFxFlag](#) 来实现同样功能。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetVoiceFxPara](#)

2.5.6 有关机器应答检测器的函数

2.5.6.1 获取与处理机器应答检测结果

2.5.6.1.1 SsmGetAMDResult

获取机器应答检测器的检测结果。

函数原型:

```
int SsmGetAMDResult (int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败或无效状态
0	检测到真人摘机, 检测结束
1	检测到信号音, 检测继续

2	检测到彩铃或提示音，检测继续
3	超时，检测结束
4	检测到信号音或提示音后线路无声，检测结束
5	检测到拨号后线路无声，检测结束
6	检测到忙音，检测结束
7	检测到对方为传真机，检测结束
8	检测到提示音，检测结束

功能描述:

获取机器应答检测器（AMD）的检测结果。

注意事项:

- 建议使用事件 [E_CHG_AMD](#) 来实现同样功能。
- 只有当配置项 [EnableAMDBeep](#) 设置为 1 时，返回值 8 才有效，此时，返回值 2 则表示检测到彩铃，检测结束。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.3.0 或更高
Header	shpa3api.h
Library	Shp_a3.lib
DLL	Shp_a3.dll

2.5.6.1.2 SsmClearAMDResult

清除机器应答检测器的检测结果。

函数原型:

```
int SsmClearAMDResult(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
0	清除成功

功能描述:

清除机器应答检测器（AMD）的检测结果。

注意事项:

- 建议在进行机器应答检测前调用该函数。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.0.3.0 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.5.6.1.3 SsmControlAMD

手动控制 AMD 检测器的开启和关闭。

函数原型:

```
int WINAPI SsmControlAMD(int ch,BOOL bStart)
```

参数说明:

ch	通道编号 是否开启 AMD 检测器的控制开关。
bStart	TRUE: 开启 FALSE: 关闭

返回值:

-1	调用失败
0	调用成功

功能描述:

手动控制 AMD 检测器的开启和关闭。

注意事项:

- 使用该函数时，通道的状态也同时会影响检测器的开启和关闭。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.0.3.0 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.5.6.1.4 SsmSetAMDPara

设置 AMD 检测器真人摘机的判断参数。

函数原型:

```
int SsmSetAMDPara(int ch, DWORD AMDTimeA, DWORD AMDTimeB, DWORD AMDTimeC, DWORD AMDTimeD)
```

参数说明:

ch	通道编号
AMDTimeA	真人摘机问候语之前的最低静音持续时间。单位毫秒 (ms)
AMDTimeB	真人摘机问候语的最低持续时间。单位毫秒 (ms)
AMDTimeC	真人摘机问候语的最大持续时间。单位毫秒 (ms)
AMDTimeD	真人摘机问候语之后的最低静音持续时间。单位毫秒 (ms)

返回值:

-1	调用失败，失败原因可以通过调用函数 SsmGetLastErrMsg 获得
0	调用成功

功能描述:

设置 AMD 检测器真人摘机的判断参数，如真人摘机问候语之前的最低静音持续时间、问候语的最低持续时间和最大持续时间以及问候语之后的最低静音持续时间。

注意事项:

- 配置项 [AMDTimeA](#)、[AMDTimeB](#)、[AMDTimeC](#)、[AMDTimeD](#) 可以实现相同功能。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.3.1 或更高
Header	Shpa3api.h
Library	shp_a3.lib

DLL

shp_a3.dll

2.6 有关 DTMF 检测器的函数

驱动程序内部有接收 DTMF 字符串缓冲区，缓冲区大小可以在配置文件中指定，缺省值为 200。应用程序应当在驱动程序的缓冲区溢出前及时读走 DTMF 字符，否则，如果驱动程序再次收到一个新的字符，将会导致最早收到的 DTMF 字符丢失。

对于坐席通道和录音通道，驱动程序检测到摘机或挂机时将自动清空接收 DTMF 缓冲区；对于其它类型通道，驱动程序在收到挂机命令时自动清空接收缓冲区。

本节所有函数不支持磁石通道。

2.6.1 启动/关闭 DTMF 检测器

2.6.1.1 SsmEnableRxDtmf

启动或关闭 DTMF 检测器。

函数原型：

```
int SsmEnableRxDtmf(int ch, BOOL bRun)
```

参数说明：

ch	通道编号
	是否开启 DTMF 检测器的控制开关。
bRun	TRUE：开启 FALSE：关闭

返回值：

-1	调用失败。失败原因可以通过调用函数 SsmGetLastErrMsg 获得
0	调用成功

功能描述：

启动或关闭 DTMF 检测器。

注意事项：

- 驱动程序会根据通道的状态迁移自动启动或关闭 DTMF 检测器。当用户需要自行控制 DTMF 检测器的开关时才需要调用本函数。但在 SS1 信令模式下，只有当应用程序自行控制呼叫过程时才支持调用本函数。
- 本函数支持 SHD 系列、DST 系列及 SHN 系列板卡。
- 配置项 [AlwaysEnableRxDtmf](#) 也可以实现类似功能。
- 本函数的有效期为应用程序调用它时到通道状态改变前。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmSetRxDtmfHandler](#), [SsmGetDtmfStr](#)

2.6.2 设置回调函数

2.6.2.1 SsmSetRxDtmfHandler

设置 [DTMF 检测器](#)输出 DTMF 信号时的回调函数。

函数原型:

```
int SsmSetRxDtmfHandler(int ch, RXDTMFHANDLER pfnOnRcvDtmf, PVOID pV)
```

参数说明:

ch	通道编号 回调函数的指针。回调函数类型为 RXDTMFHANDLER，在 shpa3api.h 中声明为: <code>typedef void (*RXDTMFHANDLER) (int ch, char cDtmf, int nDTStatus, PVOID pV);</code> 上式中，
pfnOnRcvDtmf	<code>ch:</code> 通道编号; <code>cDtmf:</code> DTMF 字符 (ASCII 码形式); <code>nDTStatus:</code> DTMF 信号类型, 0 表示 DTMF 信号消失, 1 表示 DTMF 信号出; <code>pV:</code> PVOID 类型指针, 通常用于应用程序向回调函数传递其数据结构。这个指针将由驱动程序透明地传递给回调函数。
pV	PVOID 类型指针。应用程序可以通过本参数向回调函数传递需要的对象或数据结构

返回值:

-1	调用失败。失败原因可以通过调用函数 SsmGetLastErrMsg 获得
0	调用成功

功能描述:

设置 [DTMF 检测器](#)输出 DTMF 信号时的回调函数。

注意事项:

- DTMF 检测器每检测一个完整的 DTMF 字符, 会调用 2 次本函数设置的回调函数, 一次是 DTMF 信号出现的时候, 另一次是 DTMF 信号消失的时候。更多信息请参见第 1 章中的 [DTMF 检测器](#) 部分;
- 收到的 DTMF 字符仍然会进入 DTMF 检测器的缓冲区。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.6.3 获取 DTMF 字符

2.6.3.1 SsmClearRxDtmfBuf

清空驱动程序内部的 DTMF 按键号码接收缓冲区。

函数原型:

```
int SsmClearRxDtmfBuf(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败。失败原因可以通过调用函数 SsmGetLastErrMsg 获得
0	调用成功

功能描述:

清空驱动程序内部的 DTMF 接收缓冲区。

注意事项:

- 当通道迁移到空闲状态时，驱动程序会自动清空 DTMF 检测器的缓冲区。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetDtmfStr](#)

2.6.3.2 SsmGetDtmfStr

请参见 [SsmGetDtmfStrA](#)

2.6.3.3 SsmGetDtmfStrA

获取保存在 DTMF 检测器缓冲区的 DTMF 字符。

函数原型:

```
int SsmGetDtmfStr(int ch, LPSTR pszDtmf)  
char* SsmGetDtmfStrA(int ch)
```

参数说明:

ch	通道编号
pszDtmf	存放接收 DTMF 字符串的指针，由应用程序分配存储空间

返回值:

函数名称	返回值	描述
SsmGetDtmfStr	-1	调用失败
	≥0	DTMF 检测器中的字符个数
SsmGetDtmfStrA	NULL	调用失败
	其它	返回 DTMF 检测器缓冲区的指针

功能描述:

获取保存在 [DTMF 检测器](#) 缓冲区的 DTMF 字符，字符为标准的 ASCII 码。SsmGetDtmfStr 和 SsmGetDtmfStrA 功能完全一样，只是返回 DTMF 检测器缓冲区的方式不一样。

注意事项:

- 本函数调用时，驱动程序不会清空 DTMF 检测器缓冲区。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetRxDtmfLen](#), [SsmGet1stDtmf](#), [SsmGet1stDtmfClr](#), [SsmGetLastDtmf](#)

2.6.3.4 SsmGetRxDtmfLen

获取 DTMF 检测器缓冲区中保存的 DTMF 字符个数。

函数原型:

```
int SsmGetRxDtmfLen(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败。失败原因可以通过调用函数 SsmGetLastErrMsg 获得
≥0	DTMF 检测器缓冲区中保存的 DTMF 字符个数

功能描述:

获取 DTMF 检测器缓冲区中保存的 DTMF 字符个数。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetDtmfStr](#), [SsmGetDtmfStrA](#)

2.6.3.5 SsmGet1stDtmf

请参见 [SsmGet1stDtmfClr](#)

2.6.3.6 SsmGet1stDtmfClr

获取 DTMF 检测器缓冲区中最先收到的 DTMF 字符。

函数原型:

```
int SsmGet1stDtmf(int ch, char* pcDtmf)  
int SsmGet1stDtmfClr(int ch, char* pcDtmf)
```

参数说明:

ch	通道编号
pcDtmf	存放 DTMF 字符的缓冲区首地址

返回值:

-1	调用失败。失败原因可以通过调用函数 SsmGetLastErrMsg 获得
0	DTMF 检测器缓冲区为空
1	DTMF 检测器缓冲区中有字符

功能描述:

获取 DTMF 检测器缓冲区中最先收到的 DTMF 字符。[SsmGet1stDtmf](#) 在获取 DTMF 字符后，该字符仍然保留在缓冲区中；[SsmGet1stDtmfClr](#) 则会将取到的字符从缓冲区中清除。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetDtmfStr](#), [SsmGetDtmfStrA](#)

2.6.3.7 SsmGetLastDtmf

获取 DTMF 检测器缓冲区中最后收到的 DTMF 字符。

函数原型:

```
int SsmGetLastDtmf(int ch, char* pcDtmf)
```

参数说明:

ch	通道编号
pcDtmf	存放 DTMF 字符的缓冲区首地址

返回值:

-1	调用失败。失败原因可以通过调用函数 SsmGetLastErrMsg 获得
0	DTMF 检测器缓冲区为空
1	DTMF 检测器缓冲区中有字符

功能描述:

获取 DTMF 检测器缓冲区中最后收到的 DTMF 字符。

注意事项: 无

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetRxDtmfLen](#), [SsmGet1stDtmfClr](#), [SsmGetDtmfStr](#), [SsmGetDtmfStrA](#)

2.6.4 使用 WaitDtmf 任务

2.6.4.1 SsmSetWaitDtmf

请参见 [SsmSetWaitDtmfExB](#)

2.6.4.2 SsmSetWaitDtmfEx

请参见 [SsmSetWaitDtmfExB](#)

2.6.4.3 SsmSetWaitDtmfExA

请参见 [SsmSetWaitDtmfExB](#)

2.6.4.4 SsmSetWaitDtmfExB

启动 WaitDtmf 任务。SsmSetWaitDtmf 的最大等待时间的精度为秒级，而 SsmSetWaitDtmfEx 和 SsmSetWaitDtmfExA 的最大等待时间的精度为毫秒级；SsmSetWaitDtmfEx 只支持 1 个终止字符，而 SsmSetWaitDtmfExA 支持多个终止字符。SsmSetWaitDtmfExB 与 SsmSetWaitDtmfExA 功能完全相同，但 SsmSetWaitDtmfExB 的参数 wTimeOut2 的取值范围更大。

函数原型:

```
int SsmSetWaitDtmf(int ch, WORD wTimeOut1, WORD wMaxLen, char cEndChar, BOOL bWithEndChar)
int SsmSetWaitDtmfEx(int ch, WORD wTimeOut2, WORD wMaxLen, char cEndChar, BOOL bWithEndChar)
int SsmSetWaitDtmfExA(int ch, WORD wTimeOut2, WORD wMaxLen, char* szEndChar, BOOL bWithEndChar)
int SsmSetWaitDtmfExB(int ch, DWORD wTimeOut2, WORD wMaxLen, char* szEndChar, BOOL bWithEndChar)
```

参数说明:

ch	通道编号
wTimeOut1	最大等待时间, 单位: 秒, 取值范围: 1~524
wTimeOut2	最大等待时间, 单位: 毫秒, 取值范围: 0~65535 (<code>SsmSetWaitDtmfExA</code>), 0~ $2^{32}-1$ (<code>SsmSetWaitDtmfExB</code>)
wMaxLen	等待接收 DTMF 的最大长度, 取值范围: 1~350
cEndChar	结束字符。如果驱动程序收到此字符, <code>WaitDtmf</code> 任务结束
szEndChar	结束字符集, 最多可设值 16 个。如果驱动程序收到此字符集中的任何一个字符, <code>WaitDtmf</code> 任务结束
bWithEndChar	<code>WaitDtmf</code> 任务结束后, 驱动程序返回给应用程序的字符串中是否包含收到的结束字符由本参数决定: =TRUE: 包括 =FALSE: 不包括

返回值:

-1	调用失败。失败原因可以通过调用函数 SsmGetLastErrMsg 获得
0	设置成功

功能描述:

向驱动程序提交一个接收特定 DTMF 字符串的任务, 即 `WaitDtmf` 任务。任务成功启动后, 驱动程序对 DTMF 接收缓冲区已有数据进行检查。同时, 在任务启动之后接收到的 DTMF 字符将不再进入 DTMF 检测器的接收缓冲区。直到下列条件满足才注销该任务:

- ◆ 收到由参数 `cEndChar` 或 `szEndChar` 中指定的结束字符;
- ◆ 已经接收的 DTMF 数量等于参数 `wMaxLen` 设定的最大接收长度;
- ◆ 等待时间超出参数 `wTimeOut1` 或 `wTimeOut2` 所指定的时间。

当 `WaitDtmf` 任务结束后, 驱动程序会向应用程序抛出 [E PROC WaitDTMF](#) 事件。任务的执行结果可以通过函数 [SsmChkWaitDtmf](#) 得到。

函数 [SsmCancelWaitDtmf](#) 可以用来取消 `WaitDtmf` 任务。

注意事项:

- 驱动程序在指定通道上每当收到一个 DTMF 字符, 会自动清零定时器, 重新开始计时;
- 调用本函数时, 不会清除 DTMF 缓冲区中以前收到的字符, 即缓冲区中残留的字符会影响本函数执行结果;
- 如果通道上启动了 `WaitDtmf` 任务, 在任务启动之后收到的 DTMF 字符将不再进入 DTMF 检测器的接收缓冲区。并且, 在 `WaitDtmf` 任务执行期间, 将不会产生 [E CHG RcvDTMF](#) 事件;
- 结束字符或结束字符集区分大小写。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	<code>shpa3api.h</code>
Library	<code>shp_a3.lib</code>
DLL	<code>shp_a3.dll</code>

相关函数: [SsmChkWaitDtmf](#), [SsmCancelWaitDtmf](#)

2.6.4.5 SsmCancelWaitDtmf

取消 `WaitDtmf` 任务。

函数原型:

```
int SsmCancelWaitDtmf(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	出现错误
0	调用成功

功能描述:

取消 WaitDtmf 任务。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetWaitDtmf](#), [SsmSetWaitDtmfEx](#), [SsmSetWaitDtmfExA](#), [SsmChkWaitDtmf](#)

2.6.4.6 SsmChkWaitDtmf

查询 WaitDtmf 任务的进展情况。

函数原型:

```
int SsmChkWaitDtmf(int ch, LPSTR pszDtmf)
```

参数说明:

ch	通道编号
pszDtmf	存放 DTMF 的缓冲区指针, 由应用程序分配存储空间

返回值:

-1	出现错误
0	正在接收
1	接收超时, 任务结束; 或者通道上没有 WaitDtmf 任务
2	任务由于接收到指定的结束字符而结束
3	任务由于接收到指定长度的 DTMF 字符串而结束

功能描述:

查询 WaitDtmf 任务的进展情况。

注意事项:

- [E_PROC_WaitDTMF](#) 事件提供相同的功能

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关事件: [E_PROC_WaitDTMF](#)

2.7 有关 DTMF 发生器的函数（CTI 系列）

2.7.1 设置 DTMF 发生器的参数

2.7.1.1 SsmSetTxDtmfPara

设置 DTMF 发生器产生的 DTMF 字符的高电平保持时间和低电平保持时间。

函数原型:

```
int SsmSetTxDtmfPara(int ch, WORD wOnTime, WORD wOffTime)
```

参数说明:

ch	通道编号
wOnTime	高电平保持时间，单位为毫秒，取值范围: ≥40，必须为 8 的整数倍
wOffTime	低电平保持时间，单位为毫秒，取值范围: ≥40，必须为 8 的整数倍

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

设置 [DTMF 发生器](#) 产生的 DTMF 字符的高电平保持时间和低电平保持时间。

注意事项:

- 配置项 [TxDtmfTimePara](#) 可以实现相同功能。
- 对 IP 通道而言，仅在采用带内方式发送时调用本函数才有效。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetTxDtmfPara](#), [SsmTxDtmf](#)

2.7.1.2 SsmQueryTxDtmf

查询通道是否具有 [DTMF 发生器](#)。

函数原型:

```
int SsmQueryTxDtmf(int ch, LPSTR pszReserved)
```

参数说明:

ch	通道编号
PszReserved	保留的字符串指针，可不使用

返回值:

-1	调用失败
0	指定通道不支持发送标准 DTMF 字符的操作
1	指定通道支持发送标准 DTMF 字符的操作

功能描述:

查询通道是否具有 [DTMF 发生器](#)。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:**2.7.1.3 SsmGetTxDtmfPara**

获取 DTMF 发生器产生的 DTMF 字符的高电平保持时间和低电平保持时间。

函数原型:

```
int SsmGetTxDtmfPara(int ch, PWORD pwOnTime, PWORD pwOffTime)
```

参数说明:

ch	通道编号
pwOnTime	返回高电平保持时间, 单位为毫秒
pwOffTime	返回低电平保持时间, 单位为毫秒

返回值:

-1	调用失败
=0	调用成功

功能描述:

获取 DTMF 发生器产生的 DTMF 字符的高电平保持时间和低电平保持时间。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetTxDtmfPara](#)**2.7.2 启动 DTMF 任务****2.7.2.1 SsmTxDtmf**

在线路的去话方向上发送 DTMF 字符串。

函数原型:

```
int SsmTxDtmf(int ch, LPSTR pszDtmf)
```

参数说明:

ch	通道编号
pszDtmf	<ul style="list-style-type: none">● 字符串的指针, 必须以'0'字符结束, 字符串的个数不能超过配置项 TxDtmfBufSize 的设定值。字符的取值范围:<ul style="list-style-type: none">a) '0~'9','*','#','A','B','C','D': 标准的 DTMF 信号b) 'T': 在线路上产生一次 flash, 只适用于模拟中继线通道c) 其它字符: 在线路上保持一段时间的静默。保持时间由配置项 DefaultTxDelayTime 设置, 缺省值为 2000 毫秒● 对于 IP 通道, 本参数的意义随 DTMF 发送方式不同而有以下变化:<ul style="list-style-type: none">a) 采用带内、RFC2833 事件发送方式时, 仅支持'0~'9','*','#','A','B','C','D'这 15 个标准的 DTMF 字符

b) 采用带外发送方式时，可使用 ASCII 字符集中的任意字符

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

在线路的去话方向上发送 DTMF 字符串。

驱动程序将 `pszDtmf` 中的字符拷贝到 DTMF 发生器内部的发送缓冲区后，立即启动 DTMF 发送任务。当发送缓冲区中的全部字符发送完毕后，驱动程序会向应用程序抛出 [E PROC_SendDTMF](#) 事件。函数 [SsmChkTxDtmf](#) 也可以用来查询发送任务的进展情况，函数 [SsmStopTxDtmf](#) 可以用来终止发送任务。

注意事项:

- DTMF 发生器的发送缓冲区大小可以通过配置项 [TxDtmfBufSize](#) 设置，缺省值为 50 个字符；
- 本函数只支持 SHT 系列、SHD 系列、SHN 系列板卡。
- 对 IP 通道而言，本函数需注意以下几点：
 - a) 采用带外方式发送时，本函数是同步函数。采用其他方式发送时是异步函数。
 - b) 只有在采用带内方式发送时，调用函数 [SsmStopTxDtmf](#) 结束发送过程才有效。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmChkTxDtmf](#), [SsmStopTxDtmf](#), [SsmSetTxDtmfPara](#)

2.7.2.2 SsmStopTxDtmf

取消 DTMF 发生器的发送任务。

函数原型:

```
int SsmStopTxDtmf(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

取消 DTMF 发生器的发送任务。

注意事项:

- 对 IP 通道而言，仅在采用带内方式发送时调用本函数才有效。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmTxDtmf](#)

2.7.2.3 SsmChkTxDtmf

查询 DTMF 发生器发送任务的进展情况。

函数原型:

```
int SsmChkTxDtmf(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	出现错误
0	发送 DTMF 字符串的任务完成
1	发送 DTMF 字符串的正在进行

功能描述:

查询 DTMF 发生器发送任务的进展情况。

注意事项:

- [E_PROC_SendDTMF](#) 事件可以实现相同功能;
- 对 IP 通道而言, 仅在采用带内方式发送时调用本函数才有效。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmTxDtmf](#), [SsmStopTxDtmf](#)

2.8 有关 Barge-in 检测器的函数

2.8.1 设置 Barge-in 检测器的参数

2.8.1.1 SsmSetBargeInSens

设置 Barge In 检测器的灵敏度。

函数原型:

```
int SsmSetBargeInSens(int ch, int nSens)
```

参数说明:

ch	通道编号
nSens	敏感度, 取值范围: 0~31, 值越大越灵敏

返回值:

-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

设置 Barge In 检测器的灵敏度, 更多信息请参见第 1 章中 [Barge in 检测器](#) 部分。

注意事项:

- 配置项 [BargeInSensitive](#) 可以实现相同功能。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetBargeInSens](#), [SsmSetIsBargeInDtrmTime](#), [SsmSetNoSoundDtrmTime](#)

2.8.1.2 SsmSetIsBargeInDtrmTime

设置 BargeIn 的最小保持时间。

函数原型:

```
int SsmSetIsBargeInDtrmTime(int ch, WORD wMinKeepTime)
```

参数说明:

ch	通道编号
wMinKeepTime	最小保持时间, 单位: 毫秒, 取值范围: ≥ 16 , 必须为 16 毫秒的整数倍

返回值:

-1	不支持或参数设置出错
0	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
1	调用成功

功能描述:

设置 BargeIn 的最小保持时间。

当来话中检测到语音信号, 并且保持时间大于本函数的设定值, BargeIn 检测器会判决输出“Barge In”的检测结果, 并向应用程序抛出 [E_SYS_BargeIn](#) 事件。更多信息请参见第 1 章中“[Barge in 检测器](#)”部分内容。

注意事项:

- 配置项 [BargeInDtrmTime](#) 可以实现相同功能, 缺省值为 32 毫秒。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetIsBargeInDtrmTime](#), [SsmSetBargeInSens](#), [SsmSetNoSoundDtrmTime](#)

2.8.1.3 SsmSetVoiceEnergyMinValue

设置噪音判断门限值。

函数原型:

```
int SsmSetVoiceEnergyMinValue(int ch, DWORD nVoiceEnergyMinValue)
```

参数说明:

ch	通道编号
nVoiceEnergyMinValue	噪音判断门限值, 范围 $0 < nVoiceEnergyMinValue \leq 0x7fffffff$

返回值:

-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

设置 Barge in 检测器的噪声判定门限。

注意事项:

配置项 [VoiceEnergyMinValue](#) 可以实现相同的功能，缺省值为 100000。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.1.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetVoiceEnergyMinValue](#)

2.8.1.4 SsmSetNoSoundDtrmTime

设置静默的最小保持时间。

函数原型:

```
int SsmSetNoSoundDtrmTime(int ch, DWORD wMinKeepTime)
```

参数说明:

ch	通道编号
wMinKeepTime	最小保持时间，单位：毫秒，取值范围： ≥ 16 ，必须为 16 毫秒的整数倍

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

设置静默的最小保持时间。

当来话中的语音信号消失（即保持静默），并且保持时间大于本函数的设定值，BargeIn 检测器会判决输出“无声”的检测结果，并向应用程序抛出 [E_SYS_NoSound](#) 事件。更多信息请参见第 1 章中“[Barge in 检测器](#)”部分内容。

注意事项:

- 配置项 [IsNoSoundDtrmTime](#) 可以实现相同功能，缺省值为 5000 毫秒。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetIsBargeInDtrmTime](#), [SsmSetBargeInSens](#), [SsmGetNoSoundDtrmTime](#)

2.8.1.5 SsmGetBargeInSens

获取 Barge In 检测器的灵敏度的设定值。

函数原型:

```
int SsmGetBargeInSens(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
≥0	Barge In 检测器的灵敏度

功能描述:

获取 Barge In 检测器的灵敏度的设定值。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetBargeInSens](#)

2.8.1.6 SsmGetIsBargeInDtrmTime

获取 Bargeln 的最小保持时间的设定值。

函数原型:

```
int SsmGetIsBargeInDtrmTime(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
≥0	最小保持时间，单位为毫秒

功能描述:

获取 Bargeln 的最小保持时间的设定值。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetIsBargeInDtrmTime](#)

2.8.1.7 SsmGetVoiceEnergyMinValue

获取噪音判断门限值。

函数原型:

```
DWORD SsmGetVoiceEnergyMinValue(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
≥0	调用成功，返回噪音门限判定值

功能描述:

获取 Barge in 检测器的噪声判定门限值。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver. 5.1.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetVoiceEnergyMinValue](#)

2.8.1.8 SsmGetNoSoundDtrmTime

获取静默的最小保持时间的设定值。

函数原型:

```
int SsmGetNoSoundDtrmTime(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
≥0	最小保持时间，单位为毫秒

功能描述:

获取静默的最小保持时间的设定值。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetNoSoundDtrmTime](#)

2.8.2 获取 Barge-in 检测器的检测结果

2.8.2.1 SsmDetectBargeIn

获取 Bargein 检测器的检测结果。

函数原型:

```
int SsmDetectBargeIn(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
0	没有检测到 Bargein
1	检测到 Bargein

功能描述:

获取 Bargein 检测器的检测结果。更多信息请参见第 1 章中 “[Barge in 检测器](#)” 部分内容。

注意事项:

- 推荐使用 [E_SYS_BargeIn](#) 事件来获取检测结果。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmDetectNoSound](#)

2.8.2.2 SsmDetectNoSound

获取静默检测器的检测结果。

函数原型:

```
int SsmDetectNoSound(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
0	没有检测到“无声”
1	检测到“无声”

功能描述:

获取静默检测器的检测结果。

注意事项:

- 推荐使用 [E_SYS_NoSound](#) 事件来获取检测结果。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmDetectBargeIn](#)

2.8.2.3 SsmGetNoSoundTime

获取无声的保持时间。

函数原型:

```
int SsmGetNoSoundTime(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
----	------

≥0	无声的保持时间，单位为毫秒
----	---------------

功能描述:

获取无声的保持时间。更多信息请参见第 1 章中“[Barge in 检测器](#)”部分内容。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmDetectNoSound](#)

2.9 有关放音操作的函数

2.9.1 设置放音数据的去向

2.9.1.1 SsmSetPlayDest

设置是否将放音数据送到 TDM 交换总线上。

函数原型:

```
int SsmSetPlayDest(int ch, int nSelDest)
```

参数说明:

ch	通道编号
nSelDest	放音数据是否送上 TDM 交换总线的选择开关。对于 SHD 系列板卡和 SHT 系列板卡，本参数控制 SHD 和 SHT 系列板卡工作原理框图中开关 K1-1 的状态。取值范围: 0: 不送上总线（开关断开） 1: 送上总线（开关闭合）

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

设置是否将放音数据送到 TDM 交换总线上。更多信息请参见第 1 章中“[SHD 系列板卡的工作原理框图](#)”或“[SHT 系列板卡的工作原理框图](#)”部分内容。

注意事项:

- 本函数只支持 SHT 系列、SHD 系列、SHN 系列板卡。

相关信息:

驱动程序版本要求	SynCTI Ver. 4.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.9.2 设置放音的音量

2.9.2.1 SsmSetPlayVolume

请参见 [SsmSetPlayGain](#)

2.9.2.2 SsmSetPlayGain

设置 SHD/SHT/SHN 系列板卡的放音音量。SsmSetPlayGain 提供比 SsmSetPlayVolume 更为精细的控制。

函数原型:

```
int SsmSetPlayVolume(int ch, int nGain)
int SsmSetPlayGain(int ch, WORD wGainLevel)
```

参数说明:

ch	通道编号
nGain	音量的增益，取值范围：-7~+6。大于 0 表示提升，小于 0 表示衰减，-7 表示完全关闭声音输出，缺省值为 0。音量值乘以 3 即为分贝值
wGainLevel	16 进制方式表达的音量的增益，低 8 比特有效，高 8 比特保留，应设置为 0。取值范围： wGainLevel=0: 完全关闭声音输出 wGainLevel=1: 缩小 16 倍 wGainLevel=16: 增益保持不变（0DB） 32<=wGainLevel<=255: 除以 16 后得到的数即为增益放大倍数 wGainLevel=255: 放大 16 倍

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

2 个函数均用来设置 SHD/SHT/ATP/SHN 系列板卡中音量调节器 A1 的增益。有关音量调节器 A1 的详细内容请参见 [SHD 系列板卡的工作原理框图](#)或 [SHT 系列板卡的工作原理框图](#)部分。

注意事项:

- 配置项 [DefaultPlayVolume](#) 也可以实现 SsmSetPlayVolume 的功能。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.9.3 设置播放任务的终止条件

2.9.3.1 SsmSetDtmfStopPlay

设置放音任务是否因 [DTMF 检测器](#)检测到 DTMF 字符而终止。

函数原型:

```
int SsmSetDtmfStopPlay(int ch, BOOL bTerminatePlaybackOnDTMF)
```

参数说明:

ch	通道编号
bTerminatePlaybackO	使能标志，取值范围：

nDTMF	TRUE: 开启 FALSE: 关闭
-------	-----------------------

返回值:

-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

设置放音任务是否因 [DTMF 检测器](#) 检测到 DTMF 字符而终止。

如果参数 bTerminatePlaybackOnDTMF 设置为 TRUE, 当通道上正在执行放音任务时, 如果 [DTMF 检测器](#) 在来话信号中检测到了 DTMF 字符, 并且该 DTMF 字符包含在用 [SsmSetDTMFStopPlayCharSet](#) 函数或配置项 [DtmfStopPlayCharSet](#) 预设的字符集中, 驱动程序就会立即终止放音任务。

注意事项:

- 配置项 [DefaultDtmfStopPlay](#) 也可以实现相同功能, 缺省值为关闭此项功能;
- 本函数设置的标志对预播放加载文件、单个文件、文件序列、单个内存、乒乓内存和内存序列均起作用。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetDTMFStopPlayCharSet](#), [SsmGetDtmfStopPlayFlag](#)

2.9.3.2 SsmSetDTMFStopPlayCharSet

设置终止放音任务的 DTMF 字符集。

函数原型:

```
int SsmSetDTMFStopPlayCharSet(int ch, LPSTR lpstrDtmf CharSet)
```

参数说明:

ch	通道编号
lpstrDtmf CharSet	终止放音任务的 DTMF 字符集指针。可以使用的 DTMF 字符包括: '0'、'1'、'2'、'3'、'4'、'5'、'6'、'7'、'8'、'9'、'*'、'#'、'a'、'b'、'c'、'd'

返回值:

-1	调用失败, 失败原因为参数错误, 或通道不支持本操作, 具体失败原因可以通过调用函数 SsmGetLastErrMsg 获得
0	调用失败, 失败原因为命令无法传递到板卡, 具体原因可通过调用函数 SsmGetLastErrMsg 获得
1	调用成功

功能描述:

设置终止放音任务的 DTMF 字符集。

注意事项:

- 配置项 [DtmfStopPlayCharSet](#) 也可以实现相同功能, 缺省的 DTMF 字符集为全集。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib

DLL	shp_a3.dll
-----	------------

相关函数: [SsmSetDtmfStopPlay](#), [SsmGetDTMFStopPlayCharSet](#)

2.9.3.3 SsmGetDtmfStopPlayFlag

获取放音任务是否因 DTMF 检测器检测到 DTMF 字符而终止的标志。

函数原型:

```
int SsmGetDtmfStopPlayFlag(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
0	标志为关闭状态
1	标志为开启状态

功能描述:

获取放音任务是否因 DTMF 检测器检测到 DTMF 字符而终止的标志。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetDtmfStopPlay](#)

2.9.3.4 SsmGetDTMFStopPlayCharSet

获取终止放音任务的 DTMF 字符集。

函数原型:

```
int SsmGetDTMFStopPlayCharSet(int ch, LPSTR lpstrDtmf CharSet)
```

参数说明:

ch	通道编号
lpstrDtmf CharSet	返回预设的 DMTF 字符集

返回值:

-1	调用失败。失败原因可以通过调用函数 SsmGetLastErrMsg 获得
≥0	字符集长度

功能描述:

获取终止放音任务的 DTMF 字符集。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib

DLL	shp_a3.dll
-----	------------

相关函数: [SsmSetDTMFStopPlayCharSet](#)

2.9.3.5 SsmSetBargeinStopPlay

设置放音任务是否因 [Barge in 检测器](#)检测到语音活动而终止。

函数原型:

```
int SsmSetBargeinStopPlay(int ch, BOOL bTerminatePlaybackOnBargein)
```

参数说明:

ch	通道编号
bTerminatePlaybackOnBargein	使能标志, 取值范围: TRUE: 开启 FALSE: 关闭

返回值:

-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

设置放音任务是否因 [Barge in 检测器](#)检测到语音活动而终止。

如果参数 bTerminatePlaybackOnBargein 设置为 TRUE, 当通道上正在执行放音任务时, 如果 [Barge in 检测器](#)在来话信号中检测到了语音活动, 驱动程序就会立即终止放音任务。

注意事项:

- 配置项 [DefaultBargeInStopPlay](#) 也可以实现相同功能, 缺省值为关闭;
- 本函数设置的标志对预播放加载文件、单个文件、文件序列、单个内存、乒乓内存和内存序列均起作用。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetBargeinStopPlayFlag](#)

2.9.3.6 SsmGetBargeinStopPlayFlag

获取放音任务是否因 [Barge in 检测器](#)检测到语音活动而终止的标志。

函数原型:

```
int SsmGetBargeinStopPlayFlag(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
0	标志为关闭状态
1	标志为开启状态

功能描述:

获取放音任务是否因 [Barge in 检测器](#)检测到语音活动而终止的标志。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetBargeinStopPlay](#)

2.9.3.7 SsmSetHangupStopPlayFlag

设置放音任务是否因驱动程序状态机检测到对端挂机而终止。

函数原型:

```
int SsmSetHangupStopPlayFlag(int ch, BOOL bHangupStopPlayFlag )
```

参数说明:

ch	通道号
bHangupStopPlayFlag	驱动程序检测到对端挂机时, 是否终止放音任务的标志, 取值范围: TRUE: 开启此项功能 FALSE: 关闭此项功能
-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

返回值:

-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

设置放音任务是否因驱动程序状态机在通道 Ch 上检测到对端挂机而终止。

注意事项:

- 配置项 [HangupStopPlay](#) 也可以实现相同功能, 缺省值为关闭。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.9.4 播放单个文件的专用函数

2.9.4.1 SsmPlayFile

播放单个文件, 启动文件播放任务。

函数原型:

```
int SsmPlayFile(int ch, LPSTR pszFileName, int nFormat, DWORD dwStartPos, DWORD dwLen)
```

```
int SsmPlayFileW(int ch, LPCWSTR lpFileName, int nFormat, DWORD dwStartPos, DWORD dwLen)
```

参数说明:

ch	通道编号
pszFileName lpFileName	语音文件名, pszFileName 为 UTF8 编码, lpFileName 为 UNICODE 编码。支持标准的 wav 文件和无格式文件。如果是标准的 wav 文件, 语音编码格式从文件头中获取, nFormat 参数将被忽略。如果是无格式文件, 语音编码格式由 nFormat 参数指定

语音数据的编码格式，取值范围及其含义如下表所示：

nFormat	取值	编码格式	备注
	-2	PCM16	
	1	PCM8	
	6	A-Law	
	7	μ-Law	
	17	IMA ADPCM	要求板卡具有硬件解码能力
	23	VOX	
	49	GSM	
	85	MP3	
	131	G.729A	

注意：nFormat 为 -1 表示默认的编码格式，默认的编码格式与 ShConfig.ini 中的 DefaultRecordFormat 这一项有关。具体到每个型号的板卡，是否支持、以何种方式支持上表中的编码格式，请参见第 1 章中“[板卡支持的 CODEC](#)”部分内容。

dwStartPos

文件中从实际的语音数据部分开始计算的语音片段的起始位置。对于标准的 wav 文件，起始位置为 0 是指文件头后的第 1 个语音数据字节；对于无格式文件，是从文件的头部开始计算

dwLen

播放语音数据的长度（字节数）。如果本参数大于 dwStartPos 位置到文件尾的长度，则使用后者的长度作为实际的放音长度

返回值：

-1

调用失败，失败原因可以从函数 [SsmGetLastErrMsg](#) 取得

0

调用成功

功能描述：

播放单个文件。

调用本函数后，驱动程序内部会启动一个文件播放任务，当发生下列事件时，文件播放任务终止：

- ◆ 当文件中的全部数据播放完毕；
- ◆ 应用程序调用 [SsmStopPlayFile](#) 或 [SsmStopPlay](#)；
- ◆ 驱动程序检测到 DTMF、Barge in 或对端挂机。详细内容请参见第 1 章中[设置播放任务的终止条件](#)部分。
- ◆ SHD 系列的某些型号板卡在 [LinkFromStopPlayAndTone](#) 配置项设置为 1 时，通道上发生了下总线的操作。
详细内容请参见第 1 章中“[SHD 系列板卡的原理框图](#)”部分。

文件播放任务终止后，驱动程序会指定关闭文件，并向应用程序抛出 [E PROC PlayEnd](#) 事件。函数 [SsmCheckPlay](#) 也可以用来查询文件播放任务的完成情况。

在文件播放任务的执行期间，驱动程序会每隔一定时间向应用程序抛出 [E PROC PlayFile](#) 事件。驱动程序是否抛出此事件及抛出事件的时间间隔可以通过函数 [SsmSetEvent](#)（携带 E_PROC_PlayFile 参数）设置，缺省值为每隔 1000ms 抛出事件，输出播放的时间长度。

在文件播放任务进行期间，应用程序可以调用下列函数来暂停（挂起）或调整放音位置：

- ◆ [SsmPausePlay](#)
- ◆ [SsmFastFwdPlay](#)
- ◆ [SsmFastBwdPlay](#)
- ◆ [SsmSetPlayTime](#)
- ◆ [SsmSetPlayPrct](#)

也可以调用下列函数或获取相关信息：

- ◆ [SsmGetDataBytesToPlay](#)
- ◆ [SsmGetPlayedTime](#)
- ◆ [SsmGetPlayedPercentage](#)

注意事项：

- 文件是以“共享读取”方式打开，因此本函数支持“边录边放”的功能；
- 如果在 0 通道上调用本函数，对于 ATP/DST 系列板卡，播放的声音直接送到板载扬声器端口，但不会送

到线路上；对于 SHT 系列板卡，播放的声音除了输出到线路上外，还会同时送到板载扬声器端口和线路上；

- 由于历史原因，G729A 曾经使用过的语音格式参数值 65411 在低版本依然有效，版本向下兼容。如果使用 G729A 格式录音，建议采用 131。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmStopPlayFile](#), [SsmStopPlay](#), [SsmCheckPlay](#), [SsmPausePlay](#), [SsmRestartPlay](#), [SsmFastFwdPlay](#), [SsmFastBwdPlay](#), [SsmSetPlayTime](#), [SsmSetPlayPrct](#), [SsmGetDataBytesToPlay](#), [SsmGetPlayedTime](#), [SsmGetPlayedPercentage](#)

2.9.4.2 SsmStopPlayFile

终止文件播放任务。

函数原型：

```
int SsmStopPlayFile(int ch)
```

参数说明：

ch	通道编号
----	------

返回值：

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述：

终止调用 [SsmPlayFile](#) 函数启动的文件播放任务。

注意事项：

- 本函数可以用 [SsmStopPlay](#) 函数来完全替代。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmPlayFile](#), [SsmPausePlay](#), [SsmRestartPlay](#), [SsmFastFwdPlay](#), [SsmFastBwdPlay](#), [SsmSetPlayTime](#), [SsmSetPlayPrct](#), [SsmStopPlay](#)

2.9.4.3 SsmPausePlay

暂停（挂起）文件播放任务。

函数原型：

```
int SsmPausePlay(int ch)
```

参数说明：

ch	通道编号
----	------

返回值：

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

暂停（挂起）文件播放任务。文件播放任务暂停后，应用程序只能执行下列操作：

- 调用函数 [SsmStopPlayFile](#) 或 [SsmStopPlay](#) 终止文件播放任务；
- 调用函数 [SsmRestartPlay](#) 恢复文件播放任务。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmRestartPlay](#), [SsmStopPlayFile](#), [SsmStopPlay](#)

2.9.4.4 SsmRestartPlay

恢复因为调用函数 [SsmPausePlay](#) 而暂停的文件播放任务。

函数原型:

```
int SsmRestartPlay(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

恢复因为调用函数 [SsmPausePlay](#) 而暂停的文件播放任务。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmPausePlay](#), [SsmStopPlayFile](#), [SsmStopPlay](#)

2.9.4.5 SsmFastFwdPlay

跳过 1 秒钟的语音数据后继续播放。

函数原型:

```
int SsmFastFwdPlay(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

从当前放音位置开始，跳过 1 秒钟（可通过配置项 [FastPlayTime](#) 进行设置）的数据，然后重新启动单个文件放音任务。

注意事项:

- 如果文件中剩余的语音数据不足 1 秒，驱动程序视为全部语音数据播放完毕，将终止本次播放任务。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmFastBwdPlay](#), [SsmStopPlayFile](#), [SsmStopPlay](#)

2.9.4.6 SsmFastBwdPlay

回退 1 秒钟的语音数据后继续播放。

函数原型:

```
int SsmFastBwdPlay(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

从当前放音位置开始，回退 1 秒钟（可通过配置项 [FastPlayTime](#) 进行设置）的数据，然后继续文件播放任务。

注意事项:

- 如果已经回退到了文件首部，则从文件首部开始播放。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmFastFwdPlay](#), [SsmPausePlay](#), [SsmStopPlayFile](#), [SsmStopPlay](#)

2.9.4.7 SsmSetPlayTime

跳转到文件中的指定时间位置后继续播放。

函数原型:

```
int SsmSetPlayTime(int ch, DWORD dwTime)
```

参数说明:

ch	通道编号
dwTime	以时间作为基准的新的播放位置，单位为毫秒

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

跳转到文件中的指定时间位置后继续播放。函数 [SsmGetPlayingFileInfo](#) 可以用来获取文件的持续时间等信息。

注意事项:

- 如果参数 dwTime 所对应的文件偏移量超过语音数据的长度，驱动程序视为全部语音数据播放完毕，将终止本次播放任务。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetPlayingFileInfo](#), [SsmSetPlayPrct](#), [SsmStopPlayFile](#), [SsmStopPlay](#)

2.9.4.8 SsmSetPlayPrct

跳转到文件中的指定百分比位置后继续播放。

函数原型:

```
int SsmSetPlayPrct(int ch, DWORD dwPercentage)
```

参数说明:

ch	通道编号
dwPercentage	以占整个文件的百分比作为基准的新的放音起始位置，取值范围：0~100

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

跳转到文件中的指定百分比位置后继续播放，新的位置以百分比方式提供。函数 [SsmGetPlayingFileInfo](#) 可以用来获取文件的时间信息。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetPlayTime](#), [SsmStopPlayFile](#), [SsmStopPlay](#)

2.9.4.9 SsmGetDataBytesToPlay

查询尚未播放的语音数据字节数。

函数原型:

```
int SsmGetDataBytesToPlay(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
≥0	尚未完成播放的语音数据长度, 单位为字节

功能描述:

查询尚未播放的语音数据字节数。

注意事项:

- 本函数只支持由函数 [SsmPlayFile](#) 启动的放音任务。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetDataBytesPlayed](#), [SsmGetPlayedTime](#), [SsmGetPlayedPercentage](#)

2.9.4.10 SsmGetDataBytesPlayed

查询已经完成播放的语音数据长度。

函数原型:

```
int SsmGetDataBytesPlayed(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
≥0	已经完成播放的语音数据长度, 单位为字节

功能描述:

查询已经完成播放的语音数据长度。

注意事项:

- 本函数只支持由函数 [SsmPlayFile](#) 启动的放音任务。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetDataBytesToPlay](#), [SsmGetPlayedTime](#), [SsmGetPlayedPercentage](#)

2.9.4.11 SsmGetPlayedTimeEx

获得从文件首部的第一个语音数据字节开始, 到文件中的当前放音位置的播放持续时间。

函数原型:

```
int SsmGetPlayedTimeEx(int ch)
```

参数说明:

ch	通道编号
返回值:	
-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
≥0	返回从文件首部的第 1 个语音数据字节开始, 到文件中的当前放音位置的播放持续时间, 单位为毫秒

功能描述:

获得从文件首部的第 1 个语音数据字节开始, 到文件中的当前放音位置的播放持续时间。

例如, 调用 `SsmPlayFile(ch, "Voice.voc", 7, 16000)` 启动文件播放任务后, 当驱动程序从文件偏移量 16000 处开始播放, 并且播放完 10 秒的语音数据后, 调用本函数, 得到的返回值为 $10+2=12$ 秒 (16000 字节 μ 率格式语音数据的时长为 2 秒)。

注意事项:

- 本函数只适用于 [SsmPlayFile](#) 启动的放音任务。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetPlayedTime](#)

2.9.4.12 SsmGetPlayingFileInfo

获取正在播放的放音文件的相关信息。

函数原型:

```
int SsmGetPlayingFileInfo(int ch, int *pnFormat, long *pnTime)
```

参数说明:

ch	通道编号
pnFormat	返回正在播放的文件的语音编码格式。有关 SynCTI 驱动平台中语音编码/解码格式的编码值请参见第 1 章中 SynCTI 支持的 CODEC 部分
pnTime	文件中语音数据的总时长, 单位毫秒

返回值:

-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

应用程序调用 [SsmPlayFile](#) 函数后, 本函数用于获取正在播放的放音文件的相关信息。

注意事项:

- 本函数只适用于 [SsmPlayFile](#) 启动的放音任务。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmPlayFile](#), [SsmGetPlayedTime](#), [SsmGetPlayedTimeEx](#), [SsmGetPlayedPercentage](#)

2.9.4.13 SsmPlayFileW

请参见 [SsmPlayFile](#)

2.9.5 播放文件序列的函数

2.9.5.1 SsmClearFileList

关闭并清除已提交给驱动程序的全部语音文件。

函数原型:

```
int SsmClearFileList(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

关闭并清除已提交给驱动程序的全部语音文件。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmAddToFileList](#)

2.9.5.2 SsmAddToFileList

向驱动程序提交一个需要播放的语音文件。

函数原型:

```
int SsmAddToFileList(int ch, LPSTR pszFileName, int nFormat, DWORD dwStartPos, DWORD dwLen)
```

参数说明:

ch	通道编号																								
pszFileName	语音文件名，可以是标准的 wav 文件或无格式文件。如果是标准的 wav 文件，语音编码格式从文件头中获取，nFormat 参数将被忽略。如果是无格式文件，语音编码格式由 nFormat 参数指定																								
nFormat	语音数据的编码格式，取值范围及其含义如下表所示：																								
	<table border="1"><thead><tr><th>取值</th><th>编码格式</th><th>备注</th></tr></thead><tbody><tr><td>-2</td><td>PCM16</td><td></td></tr><tr><td>1</td><td>PCM8</td><td></td></tr><tr><td>6</td><td>A-Law</td><td></td></tr><tr><td>7</td><td>μ-Law</td><td></td></tr><tr><td>17</td><td>IMA ADPCM</td><td>要求板卡具有硬件解码能力</td></tr><tr><td>23</td><td>VOX</td><td></td></tr><tr><td>85</td><td>MP3</td><td></td></tr></tbody></table>	取值	编码格式	备注	-2	PCM16		1	PCM8		6	A-Law		7	μ-Law		17	IMA ADPCM	要求板卡具有硬件解码能力	23	VOX		85	MP3	
取值	编码格式	备注																							
-2	PCM16																								
1	PCM8																								
6	A-Law																								
7	μ-Law																								
17	IMA ADPCM	要求板卡具有硬件解码能力																							
23	VOX																								
85	MP3																								
dwStartPos	具体到每个型号的板卡，是否支持、以何种方式支持上表中的编码格式，请参见第 1 章中“ 板卡支持的 CODEC ”部分内容 文件中从实际的语音数据部分开始计算的起始偏移量。例如，本参数为 0，对于标准的 wav 文件，表示从文件头后的第 1 个语音数据字节开始播放；对于无格式文件，是从文件的头部开始播放																								

dwLen	播放语音数据的长度（字节数）。如果本参数大于文件中从 dwStartPos 开始计算的实际字节数，驱动程序视为播放到文件结尾
-------	----------------------------------------------------------------

返回值：

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述：

向驱动程序提交一个需要播放的语音文件。完成文件的提交后，即可以调用函数 [SsmPlayFileList](#) 启动文件序列播放任务。

注意事项：

- 本函数调用成功后，提交的文件被以“共享读”方式打开，并且一直保持打开状态，直到调用函数 [SsmClearFileList](#) 才会被关闭；
- 本函数可以连续调用，以提交多个文件，但最多不能超过配置项 [MaxPlayFileList](#) 的设定值，并且，所有文件都应该具有相同的语音编码格式。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数： [SsmPlayFileList](#), [SsmStopPlay](#), [SsmClearFileList](#)

2.9.5.3 SsmPlayFileList

启动文件序列播放任务。

函数原型：

```
int SsmPlayFileList(int ch)
```

参数说明：

ch	通道编号
----	------

返回值：

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述：

启动文件序列播放任务。应用程序调用本函数后，驱动程序内部就启动了一个文件序列播放任务，并开始播放第 1 个文件。每当完成一个文件的播放后，驱动程序都会向应用程序抛出 [E PROC_PlayFileList](#) 事件，然后开始播放下一个文件。

当发生下列事件时，驱动程序会终止文件播放任务：

- 当文件中的全部数据播放完毕；
- 应用程序调用 [SsmStopPlayFileList](#) 或 [SsmStopPlay](#)；
- 驱动程序检测到 DTMF、Barge in 或对端挂机。详细内容请参见第 1 章中 [设置播放任务的终止条件](#) 部分。
- SHD 系列的某些型号板卡在 [LinkFromStopPlayAndTone](#) 配置项设置为 1 时，通道上发生了下总线的操作。
详细内容请参见第 1 章中“[SHD 系列板卡的原理框图](#)”部分。

文件播放任务终止后，驱动程序会并向应用程序抛出 [E PROC_PlayEnd](#) 事件。函数 [SsmCheckPlay](#) 也可以用来查询文件播放任务的完成情况。

注意事项：

- 调用本函数前，如果文件序列是空的，可以通过调用 [SsmAddToFileList](#) 函数向文件序列中添加文件；
- 文件播放任务终止后，驱动程序不会关闭提交的文件，可通过调用 [SsmClearFileList](#) 函数来清空文件列表；
- 如果在 0 通道上调用本函数，对于 ATP/DST 系列板卡，播放的声音直接送到板载扬声器端口，但不会送到线路上；对于 SHT 系列板卡，播放的声音除了输出到线路上外，还会同时送到板载扬声器端口和线路上。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmStopPlayFileList](#), [SsmStopPlay](#), [SsmCheckPlay](#), [SsmAddToFileList](#), [SsmClearFileList](#)

2.9.5.4 SsmStopPlayFileList

终止文件序列播放任务。

函数原型:

```
int SsmStopPlayFileList(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

终止文件序列播放任务，该任务必须是由函数 [SsmPlayFileList](#) 启动。

注意事项:

- 本函数可以用 [SsmStopPlay](#) 函数来完全替代。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmPlayFileList](#), [SsmClearFileList](#), [SsmStopPlay](#)

2.9.6 播放单个内存的函数

2.9.6.1 SsmPlayMem

启动单个内存播放任务。

函数原型:

```
int SsmPlayMem(int ch, int nFormat, LPBYTE pBuf, DWORD dwBufSize, DWORD dwStartOffset, DWORD dwStopOffset)
```

参数说明:

ch	通道编号
----	------

	语音数据的编码格式，取值范围及其含义如下表所示：		
nFormat	取值	编码格式	备注
	1	PCM8	
	6	A-Law	
	7	μ-Law	
	17	IMA ADPCM	要求板卡具有硬件解码能力
	-2	PCM16	
	85	MP3	
	具体到每个型号的板卡，是否支持、以何种方式支持上表中的编码格式，请参见第 1 章中“ 板卡支持的 CODEC ”部分内容		
pBuf	存放语音数据的缓冲区地址		
dwBufSize	存放语音数据的缓冲区大小，单位为字节，不能低于 768 字节		
dwStartOffset	起始放音位置（偏移量），单位为字节，从缓冲区首地址开始计算		
dwStopOffset	结束放音位置（偏移量），单位为字节，从缓冲区首地址开始计算。如果本参数小于缓冲区长度，当驱动程序内部的放音指针到达 dwStopOffset 指定的位置后，本次放音任务结束；否则，表示对该缓冲区进行循环播放，直到应用程序调用 SsmStopPlayMem ，本次放音任务才会终止		

返回值：

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述：

启动单个内存播放任务。

调用本函数后，驱动程序内部会启动一个单个内存播放任务，同时维护一个初始值等于 dwStartOffset 的放音指针（偏移量），用于从提交的缓冲区中读取数据。当发生下列事件时，单个内存播放任务终止：

- ◆ 如果 dwStopOffset 小于提交的缓冲区大小，当驱动程序内部的放音指针到达参数 dwStopOffset 指定的停止放音位置，表明缓冲区中的指定数据播放完毕；
- ◆ 应用程序调用 [SsmStopPlayMem](#) 或 [SsmStopPlay](#)；
- ◆ 驱动程序检测到 DTMF、Barge in 或对端挂机。详细内容请参见第 1 章中[设置播放任务的终止条件](#)部分。
- ◆ SHD 系列的某些型号板卡在 [LinkFromStopPlayAndTone](#) 配置项设置为 1 时，通道上发生了下总线的操作。详细内容请参见第 1 章中“[SHD 系列板卡的原理框图](#)”部分。

单个内存播放任务终止后，驱动程序会向应用程序抛出 [E PROC_PlayEnd](#) 事件。函数 [SsmCheckPlay](#) 也可以用来查询单个内存播放任务的完成情况。

在单个内存播放任务的执行期间，每当满足设定的条件时，驱动程序都会向应用程序抛出 [E PROC_PlayMem](#) 事件。[E PROC_PlayMem](#) 事件的输出条件可以通过函数 [SsmSetEvent](#) 进行设置，可以设定为下列 3 种方式之一：

- ◆ 驱动程序完成了指定时间长度的语音数据的播放；
- ◆ 驱动程序内部的放音指针越过缓冲区的 1/2 处；
- ◆ 驱动程序内部的放音指针到达缓冲末尾。

有关 [E PROC_PlayMem](#) 事件的输出条件设置请参见第 1 章中“[MESSAGE_INFO](#)”部分内容。

函数 [SsmSetStopPlayOffset](#) 可以用来重新设置放音停止位置，函数 [SsmStopPlayMem](#) 或 [SsmStopPlay](#) 可以用来终止单个内存播放任务，函数 [SsmGetPlayOffset](#) 可以用来获取驱动程序内部的放音指针。

注意事项：

- 如果在 0 通道上调用本函数，对于 ATP/DST 系列板卡，播放的声音直接送到板载扬声器端口，但不会送到线路上；对于 SHT 系列板卡，播放的声音除了输出到线路上外，还会同时送到板载扬声器端口和线路上。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib

DLL	shp_a3.dll
-----	------------

相关函数: [SsmStopPlayMem](#), [SsmStopPlay](#), [SsmSetStopPlayOffset](#), [SsmGetPlayOffset](#), [SsmSetEvent](#)

2.9.6.2 SsmStopPlayMem

终止由函数 [SsmPlayMem](#) 启动的单个内存播放任务。

函数原型:

```
int WINAPI SsmStopPlayMem(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

终止由函数 [SsmPlayMem](#) 启动的单个内存播放任务。

注意事项:

- 本函数调用成功后, 驱动程序仍然会向应用程序抛出 [E PROC PlayEnd](#) 事件;
- 函数 [SsmStopPlay](#) 可以实现相同功能。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmPlayMem](#), [SsmSetStopPlayOffset](#), [SsmStopPlay](#)

2.9.6.3 SsmSetStopPlayOffset

重新设置缓冲区中停止播放的位置。

函数原型:

```
int SsmSetStopPlayOffset(int ch, DWORD dwStopPlayOffset)
```

参数说明:

ch	通道编号
dwStopPlayOffset	内存放音的停止偏移值

返回值:

-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

重新设置缓冲区中停止播放的位置。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetPlayOffset](#)

2.9.6.4 SsmGetPlayOffset

获取单个内存播放任务时，驱动程序内部的放音指针。

函数原型：

```
int SsmGetPlayOffset(int ch, DWORD* pdwPlayOffset)
```

参数说明：

ch	通道编号
pdwPlayOffset	返回驱动程序内部的放音指针，一个以缓冲区首地址为基准的偏移量，单位为字节

返回值：

-1	调用失败
0	调用成功

功能描述：

获取单个内存播放任务时，驱动程序内部的放音指针。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmSetStopPlayOffset](#)

2.9.7 播放指定内存序列的函数

2.9.7.1 SsmAddToPlayMemList

向驱动程序提交一个内存缓冲区。

函数原型：

```
int SsmAddToPlayMemList(LPBYTE pBuf, DWORD dwDataLen, int nFormat)
```

参数说明：

pBuf	指向语音数据缓冲区的指针。内存缓冲区由应用程序分配空间
dwDataLen	语音数据缓冲区的大小，单位为字节
nFormat	语音编码格式。取值范围： 6: A-率 7: μ-率 17: IMA ADPCM

返回值：

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述：

向驱动程序提交一个内存缓冲区，缓冲区中是需要播放的语音数据。驱动程序为每次提交的缓冲区分配一个编号，第 1 次提交的缓冲区编号为 0，依此类推。

本函数可以连续多次调用，但最多不能超过配置项 [MaxPlayMemList](#) 设定的值。驱动程序在播放时，会按照内存缓冲区的提交顺序依次播放。

注意事项：

- 本函数提交的内存缓冲区并非针对某个特定的通道，所有通道都可以使用已经提交的缓冲区；

- 提交的所有缓冲区中都必须使用相同的语音编码格式；
- 驱动程序在执行本函数时，只是记录缓冲区的信息，不会将缓冲区中的数据拷贝到驱动程序内部进行缓存，因此，应用程序必须保证没有任何通道在执行内存序列播放任务时，才能释放已经提交的缓冲区；
- 如果数据是 ADPCM 格式，则需注意帧结构，帧数据混乱会导致杂音。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmClearPlayMemList](#), [SsmPlayMemList](#)

2.9.7.2 SsmClearPlayMemList

清除已经提交给驱动程序的全部缓冲区。

函数原型:

```
int SsmClearPlayMemList(void)
```

参数说明: 无

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

清除已经提交给驱动程序的全部缓冲区。

注意事项:

- 在调用本函数以前，必须保证没有任何通道在执行内存序列播放任务。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmAddToPlayMemList](#)

2.9.7.3 SsmPlayMemList

启动内存序列播放任务。

函数原型:

```
int SsmPlayMemList(int ch, PWORD pMemList, WORD wMemListLen)
```

参数说明:

ch	通道编号
pMemList	数组指针，里面存放需要播放的缓冲区编号。缓冲区编号必须预先通过函数 SsmAddToPlayMemList 提交给驱动程序
wMemListLen	数组长度

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

启动内存序列播放任务。

调用本函数后，驱动程序内部会启动一个内存序列播放任务，按照 pMemList 中指定的顺序进行播放。当发生下列事件时，内存序列播放任务终止：

- ◆ pMemList 中指定的缓冲区中的语音数据全部播放完毕；
- ◆ 应用程序调用 [SsmStopPlayMemList](#) 或 [SsmStopPlay](#)；
- ◆ 驱动程序检测到 DTMF、Barge in 或对端挂机。详细内容请参见第 1 章中[设置播放任务的终止条件](#)部分。
- ◆ SHD 系列的某些型号板卡在 [LinkFromStopPlayAndTone](#) 配置项设置为 1 时，通道上发生了下总线的操作。详细内容请参见第 1 章中“[SHD 系列板卡的原理框图](#)”部分。

内存序列播放任务终止后，驱动程序会向应用程序抛出 [E PROC PlayEnd](#) 事件。函数 [SsmCheckPlay](#) 也可以用来查询内存序列播放任务的完成情况。

注意事项：

- 如果通道不支持对调用 [SsmAddToPlayMemList](#) 函数提交缓冲区时指定的语音编码格式进行硬件解码，本函数会失败。通道是否支持对某种语音编码格式进行硬件解码，取决于它所在板卡的型号，更多信息请参见第 1 章中相关内容；
- 如果在 0 通道上调用本函数，对于 ATP/DST 系列板卡，播放的声音直接送到板载扬声器端口，但不会送到线路上；对于 SHT 系列板卡，播放的声音除了输出到线路上外，还会同时送到板载扬声器端口和线路上。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmStopPlayMemList](#), [SsmStopPlay](#), [SsmCheckPlay](#), [SsmAddToPlayMemList](#)

2.9.7.4 SsmStopPlayMemList

终止由函数 [SsmPlayMemList](#) 启动内存序列播放任务。

函数原型：

```
int SsmStopPlayMemList(int ch)
```

参数说明：

ch	通道编号
----	------

返回值：

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述：

终止由函数 [SsmPlayMemList](#) 启动内存序列播放任务。

注意事项：

- 本函数调用成功后，驱动程序仍然会向应用程序抛出 [E PROC PlayEnd](#) 事件；
- 本函数可以用 [SsmStopPlay](#) 函数来完全替代。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmPlayMemList](#), [SsmStopPlay](#)

2.9.8 播放预加载语音的函数 (CTI 系列)

2.9.8.1 预加载语音信息的初始化函数

2.9.8.1.1 SsmSetMaxIdxSeg

设置预加载语音片段的总数。

函数原型:

```
int SsmSetMaxIdxSeg(WORD wMaxIdxSeg)
```

参数说明:

wMaxIdxSeg	最大索引片段数
------------	---------

返回值:

-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

设置预加载语音片段的总数。配置项 [MaxIndexSeg](#) 也可以实现相同功能。

注意事项:

- 调用本函数时, 如果此前已经存在预加载的语音片段, 它们都将被卸载。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetTotalIndexSeg](#)

2.9.8.1.2 SsmGetTotalIndexSeg

获取预加载语音片段的总数。

函数原型:

```
int SsmGetTotalIndexSeg ()
```

参数说明: 无

返回值:

-1	调用失败
≥ 0	总的索引片段数

功能描述:

获取预加载语音片段的总数。

注意事项: 无。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
----------	---------------------

Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetMaxIdxSeg](#)

2.9.8.1.3 SsmLoadIndexData

将指定的语音片段加载到内存。

函数原型:

```
int SsmLoadIndexData(int nSegNo, LPSTR pAlias, int nCodec, LPSTR pVocFile, long lStartPos, long lLen)
```

参数说明:

nSegNo	为语音片段分配的编号, 取值范围: 0~N, N 为设置的预加载语音片段总数, 可以通过函数 SsmGetTotalIndexSeq 获得																					
pAlias	为语音片段分配的别名, 长度不能超过 20 个字符。不能以数字‘0’~‘9’作为别名字符串的首字符																					
nCodec	语音数据的编码格式, 取值范围及其含义如下表所示: <table border="1"> <thead> <tr> <th>取值</th> <th>编码格式</th> <th>备注</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>PCM_8</td> <td></td> </tr> <tr> <td>-2</td> <td>PCM_16</td> <td></td> </tr> <tr> <td>6</td> <td>A-Law</td> <td></td> </tr> <tr> <td>7</td> <td>μ-Law</td> <td></td> </tr> <tr> <td>17</td> <td>IMA ADPCM</td> <td>要求板卡具有硬件解码能力</td> </tr> <tr> <td>23</td> <td>VOX</td> <td>要求板卡具有硬件解码能力</td> </tr> </tbody> </table> 具体到每个型号的板卡, 是否支持、以何种方式支持上表中的编码格式, 请参见第 1 章中“ 板卡支持的 CODEC ”部分内容	取值	编码格式	备注	1	PCM_8		-2	PCM_16		6	A-Law		7	μ-Law		17	IMA ADPCM	要求板卡具有硬件解码能力	23	VOX	要求板卡具有硬件解码能力
取值	编码格式	备注																				
1	PCM_8																					
-2	PCM_16																					
6	A-Law																					
7	μ-Law																					
17	IMA ADPCM	要求板卡具有硬件解码能力																				
23	VOX	要求板卡具有硬件解码能力																				
pVocFile	包含语音片段的文件名, 可以是标准的 wav 格式文件, 也可以是没有文件头的无格式文件。如果是标准的 wav 文件, 参数 nCodec 指定的格式将被忽略; 否则被视为无格式文件, 编码格式由参数 nCodec 指定																					
lStartPos	文件中从实际的语音数据部分开始计算的语音片段的起始位置。对于标准的 wav 文件, 起始位置为 0 是指文件头后的第 1 个语音数据字节; 对于无格式文件, 是从文件的头部开始计算																					
lLen	语音片段的长度, 单位为字节。注意: <ul style="list-style-type: none"> 如果本参数大于 lStartPos 到文件尾的数据长度, 则实际加载的语音片段长度为 lStartPos 到文件尾 如果 lLen 设置为-1, 表示从 lStartPos 装载到文件尾 语音片段的字节长度与时间长度之间的换算值取决于编码格式, 相关内容请参见第 1 章中的“ SynCTI 支持的 CODEC ”部分																					

返回值:

-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

本函数将语音片段数据从指定文件加载到内存。语音片段的数据编码格式按照下列规则确定:

1. 如果 pVocFile 是标准 WAV 文件, 则参数 nCodec 将被忽略;
2. 如果 pVocFile 不是标准 WAV 文件, 则该文件将被视为无格式文件, 其编码格式由参数 nCodec 指定。

注意事项:

- 对每个预加载语音片段都需要调用一次本函数;
- 配置文件 ShIndex.ini 中的配置节 [SegNo](#) 可以实现相同功能;

- 应用程序应保证所有的预加载语音片段都具有相同的语音编码格式。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmFreeIndexData](#)

2.9.8.1.4 SsmFreeIndexData

卸载已经加载到内存的语音片段。

函数原型:

```
int SsmFreeIndexData(int nSegNo)
```

参数说明:

nSegNo	为语音片段分配的编号
--------	------------

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

卸载已经加载到内存的语音片段。卸载后，nSegNo 可以用来加载新的语音片段数据。

注意事项:

- 驱动程序在执行卸载 [SsmCloseCti](#) 函数时，会自动释放全部预加载语音片段所占用的内存等资源。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmLoadIndexData](#)

2.9.8.1.5 SsmLoadChIndexData

将语音片段数据从指定文件加载到相对应通道的内存中。

函数原型:

```
int WINAPI SsmLoadChIndexData(int ch, int nSegNo, LPSTR pAlias, int nCodec, LPSTR pVocFile, long lStartPos, long lLen)
```

参数说明:

ch	通道号
nSegNo	分配给内存索引语音片段的段号
pAlias	分配给内存索引语音片段的别名，别名长度不能超过 20 个字符
nCodec	语音片段数据的编码格式，可选值：6 (A 率)、7 (μ 率)、17 (IMA ADPCM)

pVocFile	包含语音数据的文件名
lStartPos	语音片段数据在语音文件的起始位置（不包括文件头）
lLen	装载语音数据的长度（字节数）。如果本参数指定的长度大于 lStartPos 到文件尾的数据长度，则实际的装载长度为后者的长度；如果 lLen=-1，则表示从 lStartPos 装载到文件尾

返回值：

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述：

本函数将语音片段数据从指定文件加载到相对应通道的内存中。语音片段的数据编码格式按照下列规则确定：

- 3、如果 pVocFile 是标准 WAV 文件，则参数 nCodec 将被忽略；
- 4、如果 pVocFile 不是标准 WAV 文件，则该文件将被视为无格式文件，其编码格式由参数 nCodec 指定。

注意事项：

- 驱动程序在执行卸载 [SsmCloseCti](#) 函数时，会自动释放全部预加载语音片段所占用的内存等资源。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.8.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmFreeChIndexData](#)

2.9.8.1.6 SsmFreeChIndexData

卸载通道上已经装载到内存的语音片段数据。

函数原型：

```
int WINAPI SsmFreeChIndexData(int ch, int nSegNo)
```

参数说明：

ch	通道号
nSegNo	索引片段号

返回值：

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述：

卸载通道上已经装载到内存的语音片段数据。卸载后，该段号可以用来装载新的语音片段数据。

注意事项：

- 驱动程序在执行卸载 [SsmCloseCti](#) 函数时，会自动释放全部预加载语音片段所占用的内存等资源。

相关信息：

驱动程序版本要求	SynCTI Ver. 4.8.0.0 或更高
Header	shpa3api.h

Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmLoadChIndexData](#)

2.9.8.2 播放预加载语音的函数

2.9.8.2.1 SsmPlayIndexString

请参见 [SsmPlayIndexList](#)

2.9.8.2.2 SsmPlayIndexList

顺序播放指定的一个或多个预加载语音片段。

函数原型:

```
int SsmPlayIndexList(int ch, WORD wIdxListLen, PWORD pwIdxList)  
int SsmPlayIndexString(int ch, LPSTR pszIdxStr)
```

参数说明:

ch	通道编号
wIdxListLen	需要播放的预加载语音片段的数量, 取值范围: 1~N-1, N 由配置项 MaxPlayIndexList 设定
pwIdxList	存放预加载语音片段编号的数组, 必须具有相同的语音编码格式
pszIdxStr	包含预加载语音片段信息的字符串, 存放需要播放的语音片段的别名或编号的序列。每一个预加载语音片段都可以使用编号或者别名来表征, 片段之间用‘,’隔开。每个编号或别名对应于内存中的一个语音数据片段。编号(或别名)与相应语音数据片段的对应关系由系统初始化函数 SsmStartCti 根据内存语音索引对照表文件 ShIndex.ini 确定, 或者由函数 SsmLoadIndexData 和 SsmFreeIndexData 分别装载或卸载。pszIdxStr 中指定的预加载语音片段的总数不能超过配置项 MaxPlayIndexList 的设定值。注意: pszIdxStr 中指定的所有语音片段必须具备相同的编码格式

返回值:

-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

按照 pwIdxList 或 pszIdxStr 指定的内容播放预加载语音片段。

调用本函数后, 驱动程序内部会启动一个预加载语音播放任务, 当发生下列事件时, 预加载语音播放任务终止:

- ◆ 参数 pwIdxList 或 pszIdxStr 指定的全部语音数据播放完毕;
- ◆ 应用程序调用 [SsmStopPlayIndex](#) 或 [SsmStopPlay](#);
- ◆ 驱动程序检测到 DTMF、Barge in 或对端挂机。详细内容请参见第 1 章中 [设置播放任务的终止条件](#) 部分。
- ◆ SHD 系列的某些型号板卡在 [LinkFromStopPlayAndTone](#) 配置项设置为 1 时, 通道上发生了下总线的操作。
详细内容请参见第 1 章中 “[SHD 系列板卡的原理框图](#)” 部分。

预加载语音播放任务终止后, 驱动程序会向应用程序抛出 [E_PROC_PlayEnd](#) 事件。函数 [SsmCheckPlay](#) 也可以用来查询预加载播放任务的完成情况。

应用程序可以随时调用函数 [SsmStopPlayIndex](#) 或 [SsmStopPlay](#) 来终止播放任务。

注意事项:

- 本函数只支持对 pwIdxList 或 pszIdxStr 中语音数据编码格式具备硬件解码功能的通道, 详细内容请参见函

数 [SsmLoadIndexData](#) 的说明。

- 如果在 0 通道上调用本函数，对于 ATP/DST 系列板卡，播放的声音直接送到板载扬声器端口，但不会送到线路上；对于 SHT 系列板卡，播放的声音除了输出到线路上外，还会同时送到板载扬声器端口和线路上。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmStopPlayIndex](#), [SsmStopPlay](#), [SsmCheckPlay](#), [SsmLoadIndexData](#)

2.9.8.2.3 SsmStopPlayIndex

停止播放预加载语音片段。

函数原型：

```
int SsmStopPlayIndex(int ch)
```

参数说明：

ch	通道编号
----	------

返回值：

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述：

停止播放预加载语音片段。播放预加载语音片段的任务终止后，驱动程序会将终止播放的原因设置为 `CHKPLAY_APPLICATION_END`，并向应用程序抛出 [E PROC PlayEnd](#) 事件。

注意事项：

- 播放任务必须由函数 [SsmPlayIndexString](#) 或 [SsmPlayIndexList](#) 启动；
- 本函数可以用 [SsmStopPlay](#) 函数来完全替代。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmPlayIndexString](#), [SsmPlayIndexList](#), [SsmStopPlay](#)

2.9.9 播放乒乓缓存的函数（CTI 系列）

2.9.9.1 SsmPlayMemBlock

向驱动程序提交一个缓冲区，并启动乒乓内存播放任务。

函数原型：

```
int SsmPlayMemBlock(int ch, int nFormat, LPBYTE pBuf, DWORD dwBufSize, PLAYMEMBLOCKHANDLER pfnCallback, PVOID pV)
```

参数说明：

ch	通道编号 语音编码格式。取值范围： -2: PCM16 1: PCM8 6: A-率 7: μ-率 17: IMA ADPCM（要求板卡具有硬件解码能力）
nFormat	
pBuf	指向语音数据缓冲区的指针
dwBufSize	语音数据缓冲区的大小，单位为字节。dwBufSize 的最小值按照下面规则确定： ● 如果配置项 RecordAndPlayUseAsIP 设置为 0 (缺省值)，dwBufSize 应不低于 768 字节 ● 如果 RecordAndPlayUseAsIP 设置为 1，对于 A-率、μ-率、PCM8 和 PCM16，dwBufSize 应不低于 192 字节 (推荐值: Windows 操作系统下应不低于 192 字节, Linux 下应不低于 384 字节); 对于 IMA ADPCM 格式，缓冲区大小应不低于 768 字节
dwBufSize	回调函数指针，当驱动程序完成一个缓冲区的播放后或该缓冲区的放音任务被终止会自动调用本参数设置的回调函数。NULL 表示不设置回调函数。回调函数的原型在 shpa3api.h 中声明为： <pre>typedef BOOL (*PLAYMEMBLOCKHANDLER) (int ch, int nEndReason, PUCHAR pucBuf, DWORD dwStopOffset, PVOID pV);</pre> 其中，ch: 通道编号 nEndReason: 结束原因。取值范围： 1: 缓冲区播放完毕 2: 放音任务因检测到 DTMF 按键字符而终止 3: 放音任务因检测到 bargein 而终止 4: 放音任务因检测到对端挂机而终止 5: 放音任务被应用程序终止 pucBuf: 播放完毕的缓冲区指针 dwStopOffset: 播放完毕时，驱动程序内部的放音指针在缓冲区中偏移量，单位为字节 pV: 应用程序在调用本函数时传递的指针
pV	PVOID 类型指针，通常用于应用程序向回调函数传递一些必要的数据，如数据结构等。这个指针将由驱动程序透明地传递给回调函数

返回值：

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述：

向驱动程序提交一个缓冲区，并启动乒乓内存播放任务。有关乒乓内存播放模式的详细内容请参见第 1 章中[播放乒乓内存](#)部分。

首次调用本函数时，驱动程序内部会启动一个乒乓内存播放任务，并开始按照提交缓冲区的顺序进行播放。每当驱动程序完成一个缓冲区的播放后，会自动切换到另一个缓冲区并开始播放，同时调用应用程序设置的回调函数，将完成播放的缓冲区信息传递给应用程序。应用程序可以在回调函数中更新缓冲区中的语音数据，然后本函数再次将缓冲区提交给驱动程序。

当发生下列事件时，乒乓内存播放任务终止：

- ◆ 驱动程序完成全部已提交的缓冲区的播放；
- ◆ 应用程序调用 [SsmStopPlayMemBlock](#) 或 [SsmStopPlay](#)；
- ◆ 驱动程序检测到 DTMF、Barge in 或对端挂机。详细内容请参见第 1 章中[设置播放任务的终止条件](#)部分。
- ◆ SHD 系列的某些型号板卡在 [LinkFromStopPlayAndTone](#) 配置项设置为 1 时，通道上发生了下总线的操作。详细内容请参见第 1 章中“[SHD 系列板卡的原理框图](#)”部分。

函数 [SsmCheckPlay](#) 也可以用来查询乒乓内存播放任务的完成情况。

注意事项：

- 如果通道不支持对 nFormat 参数指定的编码格式进行硬件解码，本函数会失败。通道是否支持对某种语音

编码格式进行硬件解码，取决于它所在板卡的型号，详细内容请参见第 1 章中相关内容；

- 在回调函数中，应用程序完成语音数据的更新和提交的代码执行时间应尽可能短，不能大于驱动程序的中断时间（8 毫秒），否则可能引起驱动程序出错；
- 应用程序可以随时调用函数 [SsmStopPlayMemBlock](#) 或 [SsmStopPlay](#) 以终止双缓冲内存放音操作，但不能在回调函数中调用上述函数；
- 乒乓内存播放任务终止后，驱动程序会向应用程序抛出 [E PROC PlayEnd](#) 事件；
- 如果在 0 通道上调用本函数，对于 ATP/DST 系列板卡，播放的声音直接送到板载扬声器端口，但不会送到线路上；对于 SHT 系列板卡，播放的声音除了输出到线路上外，还会同时送到板载扬声器端口和线路上；
- 不能同时向驱动提交两个以上的缓冲区；
- 不能使用定时器来调用 [SsmPlayMemBlock](#) 函数。

代码范例：

```
.....//初始化并填充缓冲区 szPlayBuf1, szPlayBuf2。  
BOOL PlayStart(int nCh)  
{  
    if(SsmPlayMemBlock(nCh, 6, szPlayBuf1, dwBufSize, PlayCallBack, NULL) != 0)//向驱动程序提交一个缓冲区  
        szPlayBuf1，并启动乒乓内存播放任务。  
    {  
        SsmGetLastErrMsg(szErrMsg);  
        cout<<szErrMsg<<endl;  
        return FALSE;  
    }  
    if(SsmPlayMemBlock(nCh, 6, szPlayBuf2, dwBufSize, PlayCallBack, NULL) != 0)//向驱动程序提交一个缓冲区  
        szPlayBuf2。  
    {  
        SsmGetLastErrMsg(szErrMsg);  
        cout<<szErrMsg<<endl;  
        return FALSE;  
    }  
    return TRUE;  
}  
BOOL PlayCallBack(int ch, int nEndReason, PUCHAR pucBuf, DWORD dwStopOffset, PVOID pV)  
{  
    int nResult = 0;  
    if(pucBuf == szPlayBuf1)  
    {  
        .....//重新填充缓冲区 szPlayBuf1  
        nResult = SsmPlayMemBlock(ch, 6, szPlayBuf1, dwBufSize, PlayCallBack, pV); //向驱动程序提交一个缓冲区  
        szPlayBuf1。  
    }  
    else if(pucBuf == szPlayBuf2)  
    {  
        .....//重新填充缓冲区 szPlayBuf2  
        nResult = SsmPlayMemBlock(ch, 6, szPlayBuf2, dwBufSize, PlayCallBack, pV); //向驱动程序提交一个缓冲区  
        szPlayBuf2。  
    }  
}
```

```
    }  
    return nResult;  
}
```

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmStopPlayMemBlock](#), [SsmStopPlay](#), [SsmPlayMem](#), [SsmPlayMemList](#)

2.9.9.2 SsmStopPlayMemBlock

终止由 [SsmPlayMemBlock](#) 启动的乒乓内存播放任务。

函数原型:

```
int SsmStopPlayMemBlock(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

终止由函数 [SsmPlayMemBlock](#) 启动的乒乓内存播放任务。

注意事项:

- 本函数不能在调用函数 [SsmPlayMemBlock](#) 时设置的回调函数中调用；
- 本函数可以用 [SsmStopPlay](#) 函数来完全替代。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmPlayMemBlock](#), [SsmStopPlay](#)

2.9.10 终止播放任务的通用函数

2.9.10.1 SsmStopPlay

终止以任何模式启动的播放任务。

函数原型:

```
int SsmStopPlay(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
----	----------------------------------------------------

0	调用成功
---	------

功能描述:

终止以任何模式启动的播放任务。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmStopPlayFile](#), [SsmStopPlayFileList](#), [SsmStopPlayMem](#), [SsmStopPlayMemList](#), [SsmStopPlayIndex](#), [SsmStopPlayMemBlock](#)

2.9.11 获取播放任务的相关信息

2.9.11.1 SsmQueryPlayFormat

查询通道是否支持指定的放音格式。

函数原型:

```
int SsmQueryPlayFormat (int ch, int nFormat)
```

参数说明:

ch	通道编号
nFormat	放音格式, 取值范围及详细内容请参见第 1 章的 SynCTI 支持的 CODEC 部分

返回值:

0	不支持
1	支持, 硬件解码方式
2	支持, 软件解码方式

功能描述:

查询通道是否支持指定的放音格式。

注意事项:

- 本函数适用于任何类型的通道。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.9.11.2 SsmGetPlayType

查询通道上正在执行的放音任务类型。

函数原型:

```
int SsmGetPlayType(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
0	没有放音任务
1	正在播放单个文件
2	正在播放单个文件，但被应用程序暂停
3	正在播放文件序列
4	正在播放预加载语音
5	正在播放单个内存
6	正在播放内存序列
7	正在发送 FSK 数据（注：发送 FSK 数据会占用放音资源）
8	正在使用放音操作来实现不具备卡间交换总线的通道的语音交换
9	正在播放乒乓内存

功能描述:

查询通道上正在执行的放音任务类型。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.9.11.3 SsmCheckPlay

获取放音任务的完成情况。

函数原型:

```
int SsmCheckPlay(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
0	放音正在进行
1	放音任务终止：全部数据播放完毕或该通道没有执行放音任务
2	放音任务因检测到 DTMF 字符而终止。详细内容请参见第 1 章中 设置播放任务的终止条件部分
3	放音任务因检测到 bargein 而终止。详细内容请参见第 1 章中 设置播放任务的终止条件部分
4	放音任务因检测到对端挂机而终止。详细内容请参见第 1 章中 设置播放任务的终止条件部分
5	放音任务被应用程序终止
6	放音任务被应用程序调用 SsmPausePlay 函数暂停。放音任务必须为调用 SsmPlayFile 函数启动的文件播放任务
7	放音任务因通道上执行了下总线操作而终止。详细内容请参见第 1 章中 设置播放任务的终止条件部分
8	放音任务因网络中断而停止。当应用程序调用 SsmPlayFile 函数播放网络上其它计算机上的语音文件时，如果驱动程序因各种原因，比如网络连接断开等，无法读取文件时，才会返回此值

功能描述:

获取放音任务的完成情况。

注意事项:

- 放音任务可以由 [SsmPlayFile](#)、[SsmPlayFileList](#)、[SsmPlayMemList](#)、[SsmPlayIndexList](#)、[SsmPlayIndexString](#)、[SsmPlayMem](#)、[SsmPlayMemBlock](#) 中任何一个函数启动。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmPlayFile](#)、[SsmPlayFileList](#)、[SsmPlayMemList](#)、[SsmPlayIndexList](#)、[SsmPlayIndexString](#)、[SsmPlayMem](#)、[SsmPlayMemBlock](#)

相关事件：[E PROC PlayEnd](#), [E PROC PlayFile](#), [E PROC PlayFileList](#), [E PROC PlayMem](#)

2.9.11.4 SsmGetPlayedPercentage

获取已经完成播放的数据长度占总长度的百分比。

函数原型:

```
int SsmGetPlayedPercentage(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
≥0	已播放完成的字节数占总字节数的百分比

功能描述:

获取已经完成播放的数据长度占总长度的百分比。

注意事项:

- 本函数支持由函数 [SsmPlayIndexList](#)、[SsmPlayIndexString](#)、[SsmPlayFile](#)、[SsmPlayFileList](#)、[SsmPlayMemList](#) 启动的放音任务。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmGetPlayedTime](#)

2.9.11.5 SsmGetPlayedTime

获取本次放音任务持续的时间。

函数原型:

```
int SsmGetPlayedTime(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
≥0	放音任务持续的时间，单位为毫秒

功能描述:

获取本次放音任务持续的时间。

注意事项:

- 本函数支持由函数 [SsmPlayFile](#)、[SsmPlayFileList](#)、[SsmPlayMemList](#)、[SsmPlayIndexList](#)、[SsmPlayIndexString](#)、[SsmPlayMem](#) 启动的放音任务。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetPlayedPercentage](#), [SsmGetDataBytesToPlay](#), [SsmGetPlayedTimeEx](#), [SsmGetPlayingFileInfo](#)

2.10 有关录音操作的函数

2.10.1 设置录音操作的参数

2.10.1.1 设置信号源及其音量

2.10.1.1.1 SsmSetRecVolume

请参见 [SpySetRecVolume](#)

2.10.1.1.2 SpySetRecVolume

设置进入录音混音器 M3 的来话信号的音量调节器 A3 的增益。

函数原型:

```
int SsmSetRecVolume(int ch, int nVolume)
int SpySetRecVolume(int nCic, WORD wDirection, int nVolume)
```

参数说明:

ch	通道编号
nCic	SpyCic 的逻辑编号。更多信息请参见第 1 章中“ DTP 系列板卡 ”部分内容
wDirection	音量调节器位置, 取值范围: 0: 只调节主叫方音量, 用于只录制主叫方声音时 1: 只调节被叫方音量, 用于只录制被叫方声音时 2: 同时调节主叫方音量和被叫方音量, 用于录制主叫方声音或者录制被叫方声音时
nVolume	录音音量, 取值范围: -7~+6, -7 表示关闭, 其余取值表示音量增益(乘以 3 即为分贝), 大于 0 表示音量提升, 小于 0 表示音量衰减

返回值:

-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

设置进入录音混音器 M3 的来话信号的音量调节器 A3 的增益。有关 M3 和 A3 的详细内容请参见第 1 章中相应板卡的原理框图部分。

注意事项:

- 通过配置项 [DefaultRecordVolume](#) 可以实现相同功能，缺省值为 0；
- 本函数设置的参数不支持 [SynIPRecorder](#)，对除此以外的所有类型的录音函数都起作用。若要设置 [SynIPRecorder](#) 的音量，请使用 [SsmIPRSetRecVolume](#)。

相关信息:

驱动程序版本要求	SsmSetRecVolume 要求的 SynCTI 版本为 Ver. 2.0 或更高; SpySetRecVolume 要求的 SynCTI 版本为 Ver. 5.3.1.3 或更高。
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetRecMixer](#), [SsmSetRecBack](#), [SsmIPRSetRecVolume](#)

2.10.1.1.3 SsmSetRecMixer

请参见 [SpySetRecMixer](#)

2.10.1.1.4 SpySetRecMixer

设置进入录音混音器 M3 的其它信号源的音量调节器 A2 的增益。

函数原型:

```
int SsmSetRecMixer(int ch, BOOL bEnRecMixer, int nMixerVolume)  
int SpySetRecMixer(int nCic, BOOL bEnRecMixer, int nMixerVolume)
```

参数说明:

ch	通道编号
nCic	SpyCic 的逻辑编号。更多信息请参见第 1 章中“ DTP 系列板卡 ”部分内容
bEnRecMixer	=FALSE: 关闭音量调节器 A2，录音操作的数据源仅为本通道来话信号 =TRUE: 开启音量调节器 A2，录音操作的数据源除了本通道来话信号外，还包括其它来源
nMixerVolume	其它录音信号源的音量增益，仅当 bEnRecMixer 开关为开启时有效。取值范围: -7~+6，其中： -7: 关闭 A2，相当于将 bEnRecMixer 参数设置为 FALSE -6~6: 音量增益（乘以 3 即为分贝），大于 0 表示提升，小于 0 表示衰减

返回值:

-1	调用失败
0	调用成功

功能描述:

设置进入录音混音器 M3 的其它信号源的音量调节器 A2 的增益。有关 M3 和 A2 的详细内容请参见第 1 章中相应板卡的原理框图部分。

注意事项:

- 配置项 [DefaultRecordMixerVolume](#) 可以实现相同功能，缺省为关闭 A2；
- 本函数设置的参数所有类型的录音函数都起作用；

- 使用 `SpySetRecMixer` 函数时，当当前 CIC 不在录音时，只设置主叫方通道的混音音量，不打开混音开关，`bEnRecMixer` 无效；当当前 CIC 正在混音录音时，可设置主叫方通道的混音开关及音量，`bEnRecMixer` 有效；当当前 CIC 处于其他模式录音时，返回失败；
- 本函数不支持 DST 系列板卡；
- 如果两块不同卡上的通道正在通过软交换执行有关交换总线的操作，那么必须关闭音量调节器 A2；
- `SsmSetRecMixer` 函数不支持 ATP-24A 系列板卡。

相关信息:

驱动程序版本要求	<code>SsmSetRecMixer</code> 要求的 SynCTI 版本为 Ver. 2.0 或更高； <code>SpySetRecMixer</code> 要求的 SynCTI 版本为 Ver. 5.3.1.3 或更高。
Header	<code>shpa3api.h</code>
Library	<code>shp_a3.lib</code>
DLL	<code>shp_a3.dll</code>

相关函数: [SsmSetRecVolume](#), [SsmSetRecBack](#), [SsmGetRecMixerState](#), [SsmQueryOpRecMixer](#)

2.10.1.1.5 SsmQueryOpRecMixer

查询通道是否支持设置音量调节器 A2 的增益的操作。

函数原型:

```
int SsmQueryOpRecMixer(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败，失败原因可以从函数 <code>SsmGetLastErrMsg</code> 取得
0	不支持
1	支持

功能描述:

查询通道是否支持设置音量调节器 A2 的增益的操作。

注意事项:

- 本函数不支持 DST 系列板卡。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	<code>shpa3api.h</code>
Library	<code>shp_a3.lib</code>
DLL	<code>shp_a3.dll</code>

相关函数: [SsmSetRecMixer](#)

2.10.1.1.6 SsmGetRecMixerState

获取音量调节器 A2 的增益参数设置。

函数原型:

```
int SsmGetRecMixerState(int ch, int* pnEnRecMixer, int* pnMixerVolume)
```

参数说明:

ch	通道编号
pnEnRecMixer	存放录音混音器开关状态的指针
pnMixerVolume	存放录音混音器中本通道来话信号外的另一路声音信号的音量。得到的音量值乘以 3 即为分贝

返回值:

-1	调用失败
0	调用成功

功能描述:

获取音量调节器 A2 的增益参数设置。

取得指定通道上的录音混音器的状态。如果录音混音器开关状态为开启，参数 pnEnRecMixer 将被置为 1，否则为 0。参数 pnMixerVolume 用于返回加入本通道来话信号混音的另一路声音的音量。

注意事项:

- 本函数设置的录音混音器开关对文件录音和内存录音函数都有效。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetRecMixer](#), [SsmQueryOpRecMixer](#)

2.10.1.1.7 DTRSetMixerVolume

设置 DST 系列板卡上来话/去话音量调节器 A4-1、A4-2 的增益。

函数原型:

```
int DTRSetMixerVolume(int ch, int nGroup, int nInboundGain, int nOutboundGain)
```

参数说明:

ch	通道编号
nGroup	选择 B 信道。数字电话机为 2B+D 模式，由于目前驱动程序不支持对 2 个 B 信道同时进行处理，本参数必须设置为 0
nInboundGain	来话信号音量调节器（即第 1 章中 DST 系列板卡的工作原理框图 中的音量调节器 A4-1）的增益，取值范围-7~+6，负值为衰减，正值为提升，-7 表示关闭。本参数乘以 3 即为分贝值
nOutboundGain	去话信号音量调节器（即第 1 章中 DST 系列板卡的工作原理框图 中的音量调节器 A4-2）的增益，取值范围-7~+6，负值为衰减，正值为提升，-7 表示关闭。本参数乘以 3 即为分贝值

返回值:

-1	调用失败
0	调用成功

功能描述:

设置 DST 系列板卡上来话/去话音量调节器 A4-1、A4-2 的增益。有关 A4-1、A4-2 的详细内容请参见第 1 章中 [DST 系列板卡的工作原理框图](#) 部分。

注意事项:

- 本函数只支持 DST 系列板卡。

相关信息:

驱动程序版本要求	SynCTI Ver. 4.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetRecVolume](#), [DTRGetMixerVolume](#)

2.10.1.1.8 DTRGetMixerVolume

获取 DST 系列板卡上来话/去话音量调节器 A4-1、A4-2 的增益参数。

函数原型:

```
int DTRGetMixerVolume(int ch, int nGroup, int* pnInboundGain, int* pnOutboundGain)
```

参数说明:

ch	通道编号
nGroup	选择 B 信道, 必须设置为 0
pnInboundGain	返回来话信号音量调节器 (即第 1 章中 DST 系列板卡的工作原理框图 中的音量调节器 A4-1) 的增益参数, 取值范围-6~+6, 负值为衰减, 正值为提升。本参数乘以 3 即为分贝值
pnOutboundGain	返回去话信号音量调节器 (即第 1 章中 DST 系列板卡的工作原理框图 中的音量调节器 A4-2) 的增益参数, 取值范围-6~+6, 负值为衰减, 正值为提升。本参数乘以 3 即为分贝值

返回值:

-1	调用失败
0	调用成功

功能描述:

获取 DST 系列板卡上来话/去话音量调节器 A4-1、A4-2 的增益参数。有关 A4-1、A4-2 的详细内容请参见第 1 章中 [DST 系列板卡的工作原理框图](#) 部分。

注意事项:

- 本函数只支持 DST 系列板卡。

相关信息:

驱动程序版本要求	SynCTI Ver. 4.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [DTRSetMixerVolume](#)

2.10.1.1.9 SsmSetRecBack

设置是否将总线混音器的输出、放音数据作为录音混音器的输入信号源。

函数原型:

```
int SsmSetRecBack(int ch, int nSelector)
```

参数说明:

ch	通道编号
	设置开关 k6-2 和 k6-1 的状态。取值为: 0: k6-2 断开, k6-1 断开 1: k6-2 闭合, k6-1 闭合 2: k6-2 断开, k6-1 闭合 3: k6-2 闭合, k6-1 断开
nSelector	nSelector 的默认值为 1, 即开关 k6-2 闭合, k6-1 闭合。有关这 2 个开关的更多信息请参见第 1 章中 “SHD 系列板卡的工作原理框图” 部分

返回值:

-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

设置是否将总线混音器的输出、放音数据作为录音混音器的输入信号源。

注意事项:

- 本函数只支持 SHD、SHN 系列板卡。

相关信息:

驱动程序版本要求	SynCTI Ver. 4.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetRecVolume](#), [SsmSetRecMixer](#)

2.10.1.1.10 SsmSetNoModuleChBusRec

将没有安装任何业务模块的 ATP 系列板卡上的通道作为一个具有录音功能的资源通道使用。

函数原型:

```
int SsmSetNoModuleChBusRec(int ch, int nUsedAsResourceRecCh)
```

参数说明:

ch	通道编号
nUsedAsResourceRe	1: 启用此功能 0: 不使用此功能
cCh	

返回值:

-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

将没有安装任何业务模块的 ATP 系列板卡上的通道作为一个具有录音功能的资源通道使用。详细内容请参见第 1 章中 [ATP 系列板卡的工作原理框图](#) 的“未安装业务模块的通道的特殊应用”部分。

注意事项:

- 启动录音前, 应该先调用函数 [SsmSetRecVolume](#) 或设置配置项 [DefaultRecordVolume](#), 以关闭来话信号的音量调节器;
- 录音的信号源只能来自 TDM 交换总线;
- 配置项 [NoModuleChBusRec](#) 可以实现相同功能;
- 本函数只支持 ATP 系列板卡。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetRecVolume](#)

2.10.1.2 设置录音任务的终止条件

2.10.1.2.1 SsmSetDTMFStopRec CharSet

设置录音任务是否因 [DTMF 检测器](#) 检测到 DTMF 字符而终止。

函数原型:

```
int SsmSetDTMFStopRec CharSet(int ch, LPSTR lpstrDtmf CharSet)
```

参数说明:

ch	通道号
lpstrDtmf CharSet	ASCII 形式的终止录音任务的 DTMF 字符集指针。可以使用的 DTMF 字符包括: '0'、'1'、'2'、'3'、'4'、'5'、'6'、'7'、'8'、'9'、'*'、'#'、'a'、'b'、'c'、'd'，也可以设置为空，表示不使用此项功能

返回值:

-1	调用失败。失败原因可以通过调用函数 SsmGetLastErrMsg 获得
0	调用成功

功能描述:

设置录音任务是否因 [DTMF 检测器](#) 检测到 DTMF 字符而终止。

如果参数 lpstrDtmf CharSet 不为空，当通道上正在执行录音任务时，如果 [DTMF 检测器](#) 在来话信号中检测到了 DTMF 字符，并且该 DTMF 字符包含在用 lpstrDtmf CharSet 参数指定的 DTMF 字符集中，驱动程序就会立即终止录音任务。

注意事项:

- 配置项 [DtmfStopRec CharSet](#) 可以实现相同功能；
- 由文件录音函数、内存录音函数、乒乓内存录音函数启动的录音任务均支持此项功能。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetHangupStopRecFlag](#), [SsmGetDTMFStopRec CharSet](#)

2.10.1.2.2 SsmGetDTMFStopRec CharSet

查询录音任务是否因 [DTMF 检测器](#) 检测到 DTMF 字符而终止。

函数原型:

```
int SsmGetDTMFStopRec CharSet(int ch, LPSTR lpstrDtmf CharSet)
```

参数说明:

ch	通道编号
lpstrDtmf CharSet	返回 ASCII 形式的 DTMF 字符集

返回值:

-1	调用失败。失败原因可以通过调用函数 SsmGetLastErrMsg 获得
0	此项功能被关闭
>0	预设的 DTMF 字符集中字符的个数

功能描述:

查询录音任务是否因 [DTMF 检测器](#) 检测到 DTMF 字符而终止。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetDTMFStopRec CharSet](#)

2.10.1.2.3 SsmSetHangupStopRecFlag

设置录音任务是否因驱动程序状态机检测到对端挂机而终止。

函数原型:

```
int SsmSetHangupStopRecFlag(int ch, BOOL bHangupStopRecFlag )
```

参数说明:

ch	通道编号
bHangupStopRecFlag	驱动程序检测到对端挂机时，是否终止录音任务的标志。取值范围： TRUE: 开启此项功能 FALSE: 关闭此项功能
-1	调用失败。失败原因可以通过调用函数 SsmGetLastErrMsg 获得
0	调用成功

返回值:

-1	调用失败。失败原因可以通过调用函数 SsmGetLastErrMsg 获得
0	调用成功

功能描述:

设置录音任务是否因驱动程序状态机检测到对端挂机而终止。

注意事项:

- 配置项 [HangupStopRec](#) 可以实现相同功能。驱动程序启动时，此功能为关闭状态。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetDTMFStopRec CharSet](#)

2.10.1.3 设置预录音功能 (REC 系列)

2.10.1.3.1 SsmSetPrerecord

设置预录音功能。

函数原型:

```
int SsmSetPrerecord(int ch, BOOL bEnable, int nMode, WORD wInsertTime, int nFormat)
```

参数说明:

ch	通道编号
bEnable	TRUE: 开启预录音功能 FALSE: 关闭预录音功能 注: 本参数对应的配置项为 PrerecordEnable
nMode	设置预录音功能的工作模式: 0: 当 Barge in 检测器检测到 Barge in 时开始将录音数据写入到内部缓冲区 1: 当模拟录音通道在线路上检测到摘机动作时开始将录音数据写入到内部缓冲区 注: 本参数对应的配置项为 PrerecordMode
nInsertTime	预录音功能启动后, 当应用程序启动文件录音任务时, 本参数设置将预录音的数据写入到文件中的时间长度, 单位为毫秒 注: 本参数对应的配置项为 PrerecordInsertTime
nFormat	设置录音的编码格式, 取值范围: 1: PCM8 6: A-率 7: μ-率 17: IMA ADPCM (注: 要求板卡具有硬件编码器) 49: GSM 6.10 85: MP3 注: 本参数对应的配置项为 PrerecordCodec

返回值:

-1	调用失败
0	调用成功

功能描述:

设置预录音功能。有关预录音功能的详细内容请参见第 1 章中[设置预录音功能](#)部分。

注意事项:

- 本函数适用于 SHT、SHF 系列板卡的外线通道、ATP 系列板卡和 SHD 系列板卡的 TUP 通道、ISUP 通道、ISDN 通道;
- 预录音功能只支持文件录音函数。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetPrerecordState](#)

2.10.1.3.2 SsmGetPrerecordState

获取预录音功能的设置。

函数原型:

```
int SsmGetPrerecordState (int ch , int* pnMode, PWORD pwInsertTime, int* pnFormat)
```

参数说明:

ch	通道编号
pnMode	返回预录音的工作模式
pnInsertTime	返回预录音数据写入到文件中的时间长度, 单位为毫秒
pnFormat	返回预录音操作的录音格式

返回值:

-1	调用失败
0	预录音功能为关闭状态
1	预录音功能为开启状态。只有返回值为 1 时参数表中返回的值才有意义

功能描述:

获取预录音功能的设置。

注意事项:

- 相关参数的含义请参见 [SsmSetPrerecord](#) 函数;
- 本函数适用于 SHT、SHF 系列板卡的外线通道和 ATP 系列板卡;
- 预录音功能只支持文件录音函数。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.1 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetPrerecord](#)

2.10.1.4 设置录音文件截尾功能

2.10.1.4.1 SsmSetTruncateTail

设置录音文件截尾功能的工作参数。

函数原型:

```
int SsmSetTruncateTail(int ch, DWORD dwTime)
```

参数说明:

ch	通道编号
dwTime	设置需要抛弃的录音数据长度, 单位为毫秒。0 表示关闭此项功能。 对于 SynCTI Ver. 3.4.x, 在进行截尾操作时, 截尾时间长度为终止文件录音任务的 DTMF 字符的持续时间; 对于 Ver. 3.5 及以后版本, 如果 dwTime 大于终止文件录音任务的 DTMF 字符的持续时间, 被截断的数据长度为 dwTime; 否则为该 DTMF 的实际持续时间。 本参数也可以通过配置项 TruncateTailOnRecordToFile 进行设置

返回值:

-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

设置录音文件截尾功能的工作参数。有关录音文件截尾功能的详细内容请参见第 1 章中[设置录音文件截尾功能](#)部分。

文件录音任务启动后，如果开启了录音文件截尾功能和 DTMF 字符终止文件录音任务功能，当 DTMF 检测器检测到了 DTMF 字符时，如果该 DTMF 字符落在函数 [SsmSetDTMFStopRecCharSet](#) 或配置项 [DtmfStopRecCharSet](#) 设定的字符集中，驱动程序会立即终止文件录音任务，并将文件尾部的一段语音数据删除，被删除的语音数据长度由本函数设置。

注意事项：

- 如果文件录音操作为正常终止，即完成了指定长度数据的录制，驱动程序不会进行截尾操作；
- 录音文件截尾功能只支持文件录音函数。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmRecToFile](#), [SsmRecToFileA](#), [SsmRecToFileB](#), [SsmRecToFileEx](#) , [SsmGetTruncateTailTime](#)

2.10.1.4.2 SsmGetTruncateTailTime

获取录音文件截尾功能的工作参数。

函数原型：

```
long SsmGetTruncateTailTime(int ch)
```

参数说明：

ch	通道编号
----	------

返回值：

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	此项功能未开启
>0	返回设置的截尾时间参数，单位为毫秒

功能描述：

获取录音文件截尾功能的工作参数。

注意事项：**相关信息：**

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmSetTruncateTail](#)

2.10.1.5 设置录音 AGC

2.10.1.5.1 SsmSetRecAGC

开启或关闭 AGC 功能。

函数原型：

```
int SsmSetRecAGC(int ch, int nAGCSwitch)
```

参数说明:

ch	通道号 录音自动增益控制开关 (AGC)。 对于 DST 系列 B 型数字电话录音卡, Bit0、Bit1、Bit2 有效: Bit0=0, 下行 AGC 关闭; Bit0=1, 下行 AGC 打开。 Bit1=0, 上行 AGC 关闭; Bit1=1, 上行 AGC 打开。 Bit2=0, 直流分离关闭; Bit2=1, 直流分离打开。 对于其他型号板卡, Bit0 有效: Bit0=0, AGC 关闭; Bit0=1, AGC 打开。
nAGCSwitch	

返回值:

-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

开启或关闭自动增益控制 (Automatic Gain Control,AGC) 功能。如果开启 AGC 功能, 驱动程序会根据输入信号的幅度对信号自动进行调节, 对小信号进行提升, 对大信号进行衰减。

注意事项:

- 本函数设置的录音 AGC 开关对任何类型的录音操作均有效;
- 配置项 [AutoRecAgcSwitch](#) 可以实现相同功能, 缺省值为 0;
- SHT 板卡如果需要一直开启 AGC, 除了使用本函数外, 还需要修改配置项 [OpenRecEnAndPlayEnOnIdle](#) 的值为 1。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetRecAGCSwitch](#)

2.10.1.5.2 SsmGetRecAGCSwitch

获取录音 AGC 的工作状态。

函数原型:

```
int SsmGetRecAGCSwitch(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
	对于 DST 系列 B 型数字电话录音卡, Bit0、Bit1、Bit2 有效: Bit0=0, 下行 AGC 关闭; Bit0=1, 下行 AGC 打开。 Bit1=0, 上行 AGC 关闭; Bit1=1, 上行 AGC 打开。 Bit2=0, 直流分离关闭; Bit2=1, 直流分离打开。
>0	对于其他型号板卡, Bit0 有效: Bit0=0, AGC 关闭; Bit0=1, AGC 打开。

功能描述:

获取指定通道上 AGC 模块的工作状态。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetRecAGC](#)

2.10.1.6 设置录音立体声开关

2.10.1.6.1 SsmSetRecStereo

开启或关闭立体声录音开关。

函数原型:

```
int WINAPI SsmSetRecStereo(int ch, BOOL bRecStereo)
```

参数说明:

ch	通道号
	立体声录音开关。
bRecStereo	TRUE: 开启; FALSE: 关闭

返回值:

-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

开启或关闭立体声录音开关, 如果开启了此开关, 录音时产生的是立体声声音文件, 如果关闭了开关, 录音时产生的是单声道声音文件。

注意事项:

- 本函数仅适用于 DST 系列 B 型卡;
- 本函数仅支持 A 率和μ率录音格式。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.2.2 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.10.2 获取录音操作的信息

2.10.2.1 SsmQueryRecFormat

查询通道是否支持指定的录音格式。

函数原型:

int SsmQueryRecFormat (int ch, int nFormat)

参数说明:

ch	通道编号
nFormat	录音格式

返回值:

0	不支持该格式
1	支持该格式, 硬编码
2	支持该格式, 软编码

功能描述:

查询通道是否支持指定的录音格式。

注意事项:

- 本函数支持所有的板卡和通道类型。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.10.2.2 SsmGetRecType

获取正在执行的录音任务类型。

函数原型:

int SsmGetRecType(int ch)

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
0	通道上没有录音任务
1	正在执行文件录音任务
2	文件录音任务被暂停
3	正在执行内存录音任务
4	正在执行乒乓缓冲区录音任务

功能描述:

获取正在执行的录音任务类型。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.10.2.3 SsmGetRecTime

获取录音任务持续的时间长度。

函数原型:

```
int SsmGetRecTime(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
≥ 0	录音任务持续的时间，单位为毫秒

功能描述:

获取录音任务持续的时间。

注意事项:

- 本函数只支持文件录音任务和内存录音任务。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmRecToFile](#), [SsmRecToFileA](#), [SsmRecToFileB](#), [SsmRecToFileEx](#), [SsmRecToMem](#)

2.10.2.4 SsmCheckRecord

查询录音任务的完成情况。

函数原型:

```
int SsmCheckRecord (int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
0	录音任务进行中
1	录音任务被应用程序终止
2	录音任务因检测到 DTMF 按键字符而终止
3	录音操作因检测到对端挂机而终止
4	录音任务自然终止
5	文件录音任务被暂停
6	文件录音任务因写入文件失败而终止
7	RTP 超时

功能描述:

查询录音任务的完成情况。

注意事项:

- 本函数可适用于任何方式启动的录音任务，包括 [SsmRecToFile](#), [SsmRecToFileA](#), [SsmRecToFileB](#),

[SsmRecToFileEx](#) , [SsmRecToMem](#), [SsmRecordMemBlock](#)。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数 : [SsmRecToFile](#) , [SsmRecToFileA](#) , [SsmRecToFileB](#) , [SsmRecToFileEx](#) , [SsmRecToMem](#) , [SsmRecordMemBlock](#)

2.10.3 录制到文件的函数

2.10.3.1 SsmRecToFile

请参见 [SpyRecToFile](#)

2.10.3.2 SsmRecToFileA

请参见 [SpyRecToFile](#)

2.10.3.3 SsmRecToFileB

请参见 [SpyRecToFile](#)

2.10.3.4 SsmRecToFileEx

请参见 [SpyRecToFile](#)

2.10.3.5 SpyRecToFileA

请参见 [SpyRecToFile](#)

2.10.3.6 SpyRecToFileB

请参见 [SpyRecToFile](#)

2.10.3.7 SpyRecToFile

启动指定通道上的文件录音任务。

SsmRecToFileA 除了具有 SsmRecToFile 的全部功能外, 还提供了通过回调函数来获得缓存在驱动程序内存中的录音数据流的能力。

SsmRecToFileEx 除了具有 SsmRecToFile 的全部功能外, 还提供利用 Barge in 检测器输出的 Bargin 信号来触发录音数据写入文件的功能。

SpyRecToFile、SpyRecToFileA、SpyRecToFileB 启动基于电路编号的文件录音任务, 只适用于 DTP 系列板卡。

SsmRecToFileW 、 SsmRecToFileAW 、 SsmRecToFileBW 、 SsmRecToFileExW 和 SsmRecToFile 、 SsmRecToFileA 、 SsmRecToFileB 、 SsmRecToFileEx 函数相比, 录音文件名为 UNICODE 编码格式。

函数原型:

```
int SsmRecToFile(int ch, LPSTR pszFileName, int nFormat, DWORD dwStartPos, DWORD dwBytes, DWORD dwTime, int nMask)
```

```
int SsmRecToFileW(int ch, LPCWSTR lpFileName, int nFormat, DWORD dwStartPos, DWORD dwBytes, DWORD dwTime, int nMask)
```

```
int SsmRecToFileA(int ch, LPSTR pszFileName, int nFormat, DWORD dwStartPos, DWORD dwBytes, DWORD dwTime, int nMask, LPRECTOMEM pfnCallbackA)
int SsmRecToFileAW(int ch, LPCWSTR lpFileName, int nFormat, DWORD dwStartPos, DWORD dwBytes, DWORD dwTime, int nMask, LPRECTOMEM pfnCallbackA)
int SsmRecToFileB(int ch, LPSTR pszFileName, int nFormat, DWORD dwStartPos, DWORD dwBytes, DWORD dwTime, int nMask, LPRECTOMEM pfnCallbackB, PVOID pVoid)
int SsmRecToFileBW(int ch, LPCWSTR lpFileName, int nFormat, DWORD dwStartPos, DWORD dwBytes, DWORD dwTime, int nMask, LPRECTOMEM pfnCallbackB, PVOID pVoid)
int SsmRecToFileEx(int ch, LPSTR pszFileName, int nFormat, DWORD dwStartPos, DWORD dwBytes, DWORD dwTime, int nMask, BOOL bSaveToFileOnBargin, DWORD dwRollbackTime)
int SsmRecToFileExW(int ch, LPCWSTR lpFileName, int nFormat, DWORD dwStartPos, DWORD dwBytes, DWORD dwTime, int nMask, BOOL bSaveToFileOnBargin, DWORD dwRollbackTime)
int SpyRecToFileA( int nCic, WORD wDirection, LPSTR pszFileName, int nFormat, DWORD dwStartPos, DWORD dwBytes, DWORD dwTime, int nMask, LPRECTOMEM pfnCallbackA)
int SpyRecToFileB( int nCic, WORD wDirection, LPSTR pszFileName, int nFormat, DWORD dwStartPos, DWORD dwBytes, DWORD dwTime, int nMask, LPRECTOMEM pfnCallbackB, PVOID pVoid )
int SpyRecToFile( int nCic, WORD wDirection, LPSTR pszFileName, int nFormat, DWORD dwStartPos, DWORD dwBytes, DWORD dwTime, int nMask )
```

参数说明:

ch	通道编号	保存语音数据的文件名, pszFileName 为 UTF8 编码; lpFileName 为 UNICODE 编码格式。 如果参数 pszFileName (lpFileName) 声明的文件不存在, 驱动程序按照下列规则自动创建一个新文件: <ul style="list-style-type: none">◆ 如果 pszFileName (lpFileName) 以 ".wav" 作为文件扩展名, 驱动程序认为应用程序希望录制标准 WAV 文件, 因此会创建一个标准的 WAV 文件◆ 如果不是以 ".wav" 作为文件扩展名, 驱动程序判断为无格式文件, 文件中的数据全部是语音数据, 没有文件头																																																												
pszFileName lpFileName	录音数据的编码格式, 取值范围如下表所示:	<table border="1"><thead><tr><th>取值</th><th>CODEC</th><th>SsmRecToFile</th><th>SsmRecToFileA SsmRecToFileB</th><th>SsmRecToFileEx</th><th>SpyRecToFile</th></tr></thead><tbody><tr><td>-2</td><td>PCM16</td><td>☆</td><td>☆</td><td>—</td><td>☆</td></tr><tr><td>1</td><td>PCM8</td><td>☆</td><td>☆</td><td>—</td><td>☆</td></tr><tr><td>6</td><td>A-Law</td><td>√</td><td>√</td><td>√</td><td>√</td></tr><tr><td>7</td><td>μ-Law</td><td>√</td><td>√</td><td>√</td><td>√</td></tr><tr><td>17</td><td>IMA ADPCM</td><td>√</td><td>√</td><td>√</td><td>√</td></tr><tr><td>23</td><td>VOX</td><td>√</td><td>√</td><td>—</td><td>√</td></tr><tr><td>49</td><td>GSM</td><td>◇</td><td>◇</td><td>—</td><td>◇</td></tr><tr><td>85</td><td>MP3</td><td>◇</td><td>◇</td><td>—</td><td>◇</td></tr><tr><td>131</td><td>G.729A</td><td>√</td><td>√</td><td>—</td><td>√</td></tr></tbody></table>	取值	CODEC	SsmRecToFile	SsmRecToFileA SsmRecToFileB	SsmRecToFileEx	SpyRecToFile	-2	PCM16	☆	☆	—	☆	1	PCM8	☆	☆	—	☆	6	A-Law	√	√	√	√	7	μ-Law	√	√	√	√	17	IMA ADPCM	√	√	√	√	23	VOX	√	√	—	√	49	GSM	◇	◇	—	◇	85	MP3	◇	◇	—	◇	131	G.729A	√	√	—	√
取值	CODEC	SsmRecToFile	SsmRecToFileA SsmRecToFileB	SsmRecToFileEx	SpyRecToFile																																																									
-2	PCM16	☆	☆	—	☆																																																									
1	PCM8	☆	☆	—	☆																																																									
6	A-Law	√	√	√	√																																																									
7	μ-Law	√	√	√	√																																																									
17	IMA ADPCM	√	√	√	√																																																									
23	VOX	√	√	—	√																																																									
49	GSM	◇	◇	—	◇																																																									
85	MP3	◇	◇	—	◇																																																									
131	G.729A	√	√	—	√																																																									
nFormat	图例: ☆ 使用软件编码器, 由驱动程序通过软件编码实现 √ 使用硬件编码器实现, 要求硬件具有编码器。更多信息请参见第 1 章“ SynCTI 支持的 CODEC ” ★ 软件编码, 通过外挂 ACM 程序实现 — 不支持 ◇ 如果硬件支持, 使用硬件编码器; 否则使用软件编码器, 通过外挂 ACM 程序实现																																																													
	实际的录音编码格式按照下列规则确定: 如果参数 pszFileName (lpFileName) 声明的文件已经存在, 并且是标准的 wav 文件, 则编码格式为 wav 文件头中指定的编码格式, 本参数将被忽略; 如果已经存在的文件不是标准的 wav 文件, 或者文件为新创建文件, 编码格式由																																																													

	nFormat 决定。 注意: nFormat 为-1 表示默认的编码格式, 默认的编码格式与 ShConfig.ini 中的 DefaultRecordFormat 这一项有关。如果开启了预录音功能, nFormat 必须与预录音功能设置的语音编码格式相同, 否则本函数会返回失败。
dwStartPos	第 1 包录音数据写入到文件时的起始偏移量。如果目标文件具有 wav 文件头, dwStartPos 从语音数据区的第一个直接开始计算, 不包括文件头长度。 ◆ 如果文件已经存在: 如果 dwStartPos 小于或等于文件中的语音数据长度, 从本参数指定的位置开始写入数据; 如果 dwStartPos 大于文件中的语音数据长度, 从文件尾部开始写入录音数据 ◆ 如果是新创建文件: 本参数将被忽略。如果是标准 wav 文件, 紧随文件头开始写入; 如果不是, 从文件首部开始写入
dwBytes	设置需要录制的语音数据的长度, 单位为字节。 当录制完成的总字节数大于本参数设置的字节数时, 录音任务结束。将本参数设置为一个很大的值(例如 0xffffffff)就可以达到持续录音的目的。 本参数是否起作用由 nMask 参数控制。 注意: 如果使用具有帧结构的语音编码格式(如 IMA ADPCM、GSM、MP3、G.729A 等), 并且 dwBytes 不是帧长度的整数倍时, 驱动程序会自动调整 dwBytes 为帧长度的整数倍, 以保证帧结构的完整性
dwTime	设置需要录制的语音数据的总时长, 单位为毫秒。 当录制任务持续的总时长大于本参数设置的时长时, 录音任务结束。将本参数设置为一个很大的值(例如 0xffffffff)就可以达到持续录音的目的。 本参数是否起作用由 nMask 参数控制。 注意: 如果使用具有帧结构的语音编码格式(如 IMA ADPCM、GSM、MP3、G.729A 等), 并且 dwTime 参数对应的字节数不是帧长度的整数倍时, 驱动程序会自动调整 dwTime 为帧长度的整数倍, 以保证帧结构的完整性
nMask	设置停止录音的方式。取值范围: 0: 以 dwBytes 参数作为停止录音的条件 1: 以 dwTime 参数作为停止录音的条件
pfnCallbackA pfnCallbackB	设置回调函数指针。如果 pfnCallbackA 或 pfnCallbackB 为 NULL, SsmRecToFileA 或 SsmRecToFileB 的功能与 SsmRecToFile 完全相同, SpyRecToFileA 或 SpyRecToFileB 的功能与 SpyRecToFile 完全相同; 如果不为 NULL, 当驱动程序的内部缓冲区积累了 16384 字节的录音数据并将其一次性写入文件后, 会调用本参数设置的回调函数, 将缓存的语音数据流同步传递给应用程序。函数 SsmSetFlag (携带 F_RECTOFILEA_CALLBACKTIME 参数) 可以用来设置回调的间隔。 回调函数 pfnCallbackA 的原型: <pre>void CALLBACK RecFileCallback(int ch, LPBYTE lpData, DWORD dwDataLen);</pre> 回调函数 pfnCallbackB 的原型: <pre>void CALLBACK RecFileCallback (int ch, LPBYTE lpData, DWORD dwDataLen, PVOID pVoid);</pre> 上式中, ch: 语音通道号 lpData: 驱动程序内部的录音数据缓冲区地址 dwDataLen: 录音数据长度(字节数) pVoid: 调用 SsmRecToFileB 或 SpyRecToFileB 时传入的指针。
pVoid	PVOID 类型的指针, 用于将应用程序的对象指针透明地传输给回调函数 pfnCallbackB 是否启用 Barge in 检测器 的输出来触发将录音数据写入文件的功能。取值范围: TRUE: 启用 FALSE: 不启用
bSaveToFileOnBargin	如果本参数为 FALSE, dwRollbackTime 参数将被忽略, SsmRecToFileEx 函数与 SsmRecToFile 功能完全一致
dwRollbackTime	本参数用于决定将 Barge in 信号产生以前多少时间的语音数据保存到文件, 单位为毫秒, 最大值为 2000ms。仅当 bSaveToFileOnBargin 参数设置为 TRUE 时有效。 由于 Barge in 检测器 检测到 Barge in 需要一定的时间, 为了保证录制的语音数据的完整性, 需要合理设置本参数。建议本参数应略大于函数 SsmSetIsBargeInDtrmTime 或配置项 BargeInDtrmTime 所设置的参数

nCic	SpyCic 的逻辑编号。更多信息请参见第 1 章中“ DTP 系列板卡 ”部分内容 设置录音信号源，取值范围： 0：只录制主叫方的声音。驱动程序自动在与 nCic 绑定的主叫方通道上调用 SsmRecToFile 函数，相关的输出消息在主叫方通道上返回 1：只录制被叫方的声音。驱动程序自动在与 nCic 绑定的被叫方通道上调用 SsmRecToFile 函数，相关的输出消息在被叫方通道上返回 2：混合录制主叫方与被叫方的声音。驱动程序自动在与 nCic 绑定的被叫方通道上调用 SsmRecToFile 函数，相关的输出消息在被叫方通道上返回
------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

返回值：

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述：

启动指定通道上的文件录音任务。

[SsmRecToFileA](#) 和 [SsmRecToFileB](#) 除了具有 [SsmRecToFile](#) 的全部功能外，可以将保存在驱动程序内部缓冲区的录音数据流以回调函数的方式开放给应用程序。如果参数 pfnCallbackA（或 pfnCallbackB）为 NULL，[SsmRecToFileA](#)（或 [SsmRecToFileB](#)）的功能与 [SsmRecToFile](#) 函数完全相同。

[SpyRecToFileA](#) 和 [SpyRecToFileB](#) 除了具有 [SpyRecToFile](#) 的全部功能外，可以将保存在驱动程序内部缓冲区的录音数据流以回调函数的方式开放给应用程序。如果参数 pfnCallbackA（或 pfnCallbackB）为 NULL，[SpyRecToFileA](#)（或 [SpyRecToFileB](#)）的功能与 [SpyRecToFile](#) 函数完全相同。

[SsmRecToFileEx](#) 函数启动录音任务后，驱动程序只是将录音数据缓存到内部的录音缓冲区中，并不将录音数据写入到文件。只有当 [Barge in](#) 检测器检测到 [Barge in](#) 后，驱动程序才开始将内部缓冲中的录音数据写入到文件。如果在本次录音任务终止时仍然没有检测到 [Bargin](#)，则不会将任何数据写入到录音文件。

应用程序调用文件录音的函数后，驱动程序内部会启动一个文件录音任务，当发生下列事件时，文件录音任务终止：

- ◆ 文件录音任务自然终止，即驱动程序录制的语音数据长度到达参数 dwBytes 或 dwTime 参数指定的长度；
- ◆ 驱动程序将语音数据写入文件时失败；
- ◆ 应用程序调用 [SsmStopRecToFile](#)；
- ◆ 驱动程序检测到 DTMF 或对端挂机，详细内容请参见第 1 章中[设置录音任务的终止条件](#)部分。

文件录音任务终止后，驱动程序会自动关闭文件，并向应用程序抛出 [E_PROC_RecordEnd](#) 事件。如果开启了录音文件截尾功能，驱动程序会自动将文件尾部的一段数据抛弃。有关录音文件截尾功能的详细内容请参见第 1 章中[设置录音文件截尾功能](#)部分。

在文件录音任务的执行期间，驱动程序会每隔 1000ms 向应用程序抛出 [E_PROC_RecordFile](#) 事件。驱动程序是否抛出此事件及抛出事件的条件可以通过 [SsmSetEvent](#) 函数（携带 E_PROC_RecordFile 参数）设置，缺省值为每隔 1000 毫秒抛出此事件。函数 [SsmChkRecToFile](#) 或 [SsmCheckRecord](#) 也可以用来查询文件录音任务的完成情况。

在文件录音任务执行过程中，应用程序可以：

- ◆ 调用 [SsmPauseRecToFile](#) 函数暂停文件录音任务；
- ◆ 调用 [SsmStopRecToFile](#) 函数终止文件录音任务；
- ◆ 调用 [SsmChkRecToFile](#) 或 [SsmCheckRecord](#) 函数查询文件录音任务的完成情况；
- ◆ 调用 [SsmGetRecTime](#) 函数获取文件录音任务持续的时间；
- ◆ 调用 [SsmGetDataBytesToRecord](#) 函数获取尚未完成的录音数据字节数。

注意事项：

- 有关选择录音信号源、设置音量的详细内容请参见第 1 章中“[设置录音信号源](#)”部分内容；

- 如果开启了预录音功能，驱动程序会根据预录音的参数设置将保存在内部缓冲区的预录音数据先写入到文件，然后才开始执行正常的文件录音任务，将后续录音数据写入文件。有关预录音功能的详细内容请参见第1章中“[设置预录音功能](#)”部分内容；
- 如果nFormat为GSM或MP3，必须将配置项[GsmCodecEnable](#)设置为非0，并保证Windows系统中已经安装了相应的音频编码解码器(ACM)驱动程序；
- SpyRecToFile启动基于电路编号的文件录音任务，只适用于DTP系列板卡。SpyRecToFile函数是基于电路编号的函数，它向应用程序抛出的消息是基于呼叫过程中与nCic绑定的主叫方或被叫方通道。函数[SpyGetCallInCh](#)、[SpyGetCallOutCh](#)分别用于获取本次呼叫中的主叫方和被叫方的通道编号。函数[SpyStopRecToFile](#)用于停止SpyRecToFile函数启动的文件录音任务；
- 由于历史原因，G729A曾经使用过的语音格式参数值65411在低版本依然有效，版本向下兼容。如果使用G729A格式录音，建议采用131；
- SsmRecToFile同时也支持SynIPRecorder，该函数返回成功只是表示它已经往Slaver发送开始录音到文件的命令，至于该命令在Slaver端是否成功处理需要通过事件[E_IPR_START_REC_CB](#)上携带的dwParam的值来确定。

相关信息：

驱动程序版本要求	SsmRecToFileB要求的SynCTI版本为Ver.4.7.2.0或更高； SpyRecToFileA、SpyRecToFileB要求的SynCTI版本为Ver.5.3.1.3或更高； 其余函数要求SynCTI Ver.2.0或更高。
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmStopRecToFile](#), [SsmChkRecToFile](#), [SsmGetRecTime](#), [SsmPauseRecToFile](#), [SsmRestartRecToFile](#), [SsmGetDataBytesToRecord](#), [SsmSetDTMFStopRecCharSet](#), [SsmSetHangupStopRecFlag](#), [SsmSetRecVolume](#), [SsmSetRecMixer](#), [SsmSetRecBack](#), [SsmSetRecAGC](#)

2.10.3.8 SsmStopRecToFile

请参见[SpyStopRecToFile](#)

2.10.3.9 SpyStopRecToFile

终止用文件录音任务。

函数原型：

```
int SsmStopRecToFile(int ch)  
int SpyStopRecToFile(int nCic)
```

参数说明：

ch	通道编号
nCic	被监控电路编号

返回值：

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述：

SsmStopRecToFile终止由函数[SsmRecToFile](#)、[SsmRecToFileA](#)、[SsmRecToFileB](#)或[SsmRecToFileEx](#)启动的文件录音任务，SpyStopRecToFile用于终止调用[SpyRecToFile](#)、[SpyRecToFileA](#)、[SpyRecToFileB](#)函数启动的文件录音任务。

注意事项:

- 本函数调用后，驱动程序也会抛出 [E PROC RecordEnd](#) 事件；
- SsmStopRecToFile 同时也支持 SynIPRecorder，此时它返回成功只是表示它已经往 Slaver 发送停止录音到文件的命令，至于该命令在 Slaver 端是否成功处理需要通过事件 [E IPR_STOP_REC_CB](#) 上携带的 dwParam 的值来确定。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmRecToFile](#), [SsmRecToFileA](#), [SsmRecToFileB](#), [SsmRecToFileEx](#), [SsmSetDTMFStopRecCharSet](#), [SsmSetHangupStopRecFlag](#), [SsmSetRecVolume](#), [SsmSetRecMixer](#), [SsmSetRecBack](#)

2.10.3.10 SsmPauseRecToFile

暂停文件录音任务。

函数原型:

```
int SsmPauseRecToFile(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

暂停由函数 [SsmRecToFile](#)、[SsmRecToFileA](#)、[SsmRecToFileB](#) 或 [SsmRecToFileEx](#) 启动的文件录音任务。文件录音任务被挂起后，只能通过函数 [SsmRestartRecToFile](#) 恢复，或者通过函数 [SsmStopRecToFile](#) 终止。

注意事项:

- 本函数只支持 [SsmRecToFile](#), [SsmRecToFileA](#), [SsmRecToFileB](#), [SsmRecToFileEx](#) 启动的任务；
- 该函数同时也支持 SynIPRecorder，此时它返回成功只是表示它已经往 Slaver 发送暂停录音到文件的命令，至于该命令在 Slaver 端是否成功处理需要通过事件 [E IPR_PAUSE_REC_CB](#) 上携带的 dwParam 的值来确定。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmRestartRecToFile](#), [SsmStopRecToFile](#)

2.10.3.11 SsmRestartRecToFile

恢复文件录音任务。

函数原型:

```
int SsmRestartRecToFile(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

恢复因 [SsmPauseRecToFile](#) 函数而暂停的文件录音任务。

注意事项:

- 该函数同时也支持 SynIPRecorder，此时它返回成功只是表示它已经往 Slaver 发送重新开始录音到文件的命令，至于该命令在 Slaver 端是否成功处理需要通过事件 [E_IPR_RESTART_REC_CB](#) 上携带的 dwParam 的值来确定。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmPauseRecToFile](#)

2.10.3.12 SsmChkRecToFile

查询文件录音任务的完成情况。

函数原型:

```
int SsmChkRecToFile(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	文件录音任务正常结束
1	文件录音任务正在进行
2	文件录音任务被暂停

功能描述:

查询文件录音任务的完成情况。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmRecToFile](#), [SsmRecToFileA](#), [SsmRecToFileB](#), [SsmRecToFileEx](#)

2.10.3.13 SsmGetDataBytesToRecord

获取尚未完成的文件录音数据字节数或者获取已经完成的双缓冲录音数据字节数。

函数原型:

```
int SsmGetDataBytesToRecord(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
≥0	文件录音任务中尚未完成录制的语音数据字节数，或双缓冲录音任务中已经完成录制的语音数据字节数

功能描述:

获取文件录音任务启动后，尚未完成的录音数据字节数；或者获取双缓冲录音任务启动后，已经完成的录音数据字节数。

注意事项:

- 本函数对录音格式为 GSM 的双缓冲录音任务无效。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetRecTime](#)

2.10.3.14 SsmRecStereoToFile

启动立体声文件录音。

函数原型:

```
int WINAPI SsmRecStereoToFile(LPSTR pszFileName, DWORD ch1, DWORD ch2, int nFormat)
```

参数说明:

pszFileName	指定立体声录音文件名，以 wav 为结尾
ch1	通道编号 1
ch2	通道编号 2
nFormat	立体声录音文件格式，仅支持 A-Law, μ-Law。6 表示使用 A-Law, 7 表示使用 μ-Law

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	立体声文件录音任务正常启动

功能描述:

启动立体声文件录音。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.2.1 或更高
----------	-------------------------

Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmStopRecStereoToFile](#)、[SsmChkStereoToFile](#)

2.10.3.15 SsmStopRecStereoToFile

停止立体声文件录音。

函数原型:

```
int WINAPI SsmStopRecStereoToFile(DWORD ch1, DWORD ch2)
```

参数说明:

ch1	通道编号 1
ch2	通道编号 2

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	立体声文件录音任务正常停止

功能描述:

停止立体声文件录音。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.2.1 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmRecStereoToFile](#)、[SsmChkStereoToFile](#)

2.10.3.16 SsmChkStereoToFile

查询立体声文件录音进展。

函数原型:

```
int WINAPI SsmChkStereoToFile(DWORD ch1, DWORD ch2)
```

参数说明:

ch1	通道编号 1
ch2	通道编号 2

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	立体声文件录音任务正常停止
1	立体声文件录音任务正在进行

功能描述:

查询立体声文件录音进展。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.2.1 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmRecStereoToFile](#)、[SsmStopRecStereoToFile](#)

2.10.3.17 SsmPauseRecToFileEx

暂停文件录音任务，适用于 IPR 通道。

函数原型:

```
int SsmPauseRecToFileEx(int ch, BOOL bInsertSilence)
```

参数说明:

ch	通道编号
bInsertSilence	暂停录音阶段是否填充静音数据

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

暂停由函数 [SsmRecToFile](#)、[SsmRecToFileA](#) 或 [SsmRecToFileB](#) 启动的文件录音任务。文件录音任务被挂起后，只能通过函数 [SsmRestartRecToFile](#) 恢复，或者通过函数 [SsmStopRecToFile](#) 终止。

该函数为 IPR 通道专用，当 VOIP 呼叫进入诸如呼叫保持状态时，一般会停止 RTP 流的交互，为了录制完整的呼叫过程，此时通常需要调用 [SsmPauseRecToFileEx](#) 函数以避免因 RTP 超时而导致的录音中断；抑或是需要避免录音过程中出现的 DTMF 语音进入录音文件，同样需要调用 [SsmPauseRecToFileEx](#) 函数。以上两种情况下均可以通过参数 [bInsertSilence](#) 来决定暂停录音期间是否填充静音数据，灵活满足个性化需求。

注意事项:

- 函数只支持 [SsmRecToFile](#), [SsmRecToFileA](#), [SsmRecToFileB](#) 启动的任务；
- 该函数只支持 [SynIPRecorder](#)，它返回成功只是表示它已经往 [Slaver](#) 发送暂停录音到文件的命令，至于该命令在 [Slaver](#) 端是否成功处理需要通过事件 [E_IPR_PAUSE_REC_CB](#) 上携带的 [dwParam](#) 的值来确定。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.4.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmRestartRecToFile](#), [SsmStopRecToFile](#)

2.10.3.18 SsmRecToFileW

请参见 [SpyRecToFile](#)

2.10.3.19 SsmRecToFileAW

请参见 [SpyRecToFile](#)

2.10.3.20 SsmRecToFileBW

请参见 [SpyRecToFile](#)

2.10.3.21 SsmRecToFileExW

请参见 [SpyRecToFile](#)

2.10.4 录制到指定内存的函数

2.10.4.1 SsmRecToMem

请参见 [SpyRecToMem](#)

2.10.4.2 SpyRecToMem

启动内存录音任务。

函数原型:

```
int SsmRecToMem(int ch, int nFormat, LPBYTE pBuf, DWORD dwBufSize, DWORD dwStartOffset)
int SpyRecToMem(int nCic, WORD wDirection, int nFormat, LPBYTE pBuf, DWORD dwBufSize, DWORD dwStartOffset)
```

参数说明:

ch	通道编号	录音数据的编码格式。取值范围:	备注
nFormat	1	PCM8	
	-2	PCM16	
	6	A-Law	
	7	μ-Law	
	17	IMA ADPCM	要求板卡具有硬件编码能力或是 SynIPRecorder
	23	VOX	要求板卡具有硬件编码能力
	49	GSM	
	85	MP3	要求板卡具有硬件编码能力
	131	G.729A	要求板卡具有硬件编码能力
	通道是否支持硬件编码器的详细内容请参见第 1 章中 “ SynCTI 支持的 CODEC ” 部分		
pBuf	录音缓冲区的首地址指针, 由应用程序分配空间		
dwBufSize	录音缓冲区的大小, 单位为字节		
dwStartOffset	起始录音位置在录音缓冲区中的偏移量。取值范围: 0~dwBufSize-1		
nCic	SpyCic 的逻辑编号。更多信息请参见第 1 章中 “ DTP 系列板卡 ” 部分内容		
wDirection	设置录音信号源, 取值范围:		
	0: 只录制主叫方的声音。驱动程序自动在与 nCic 绑定的主叫方通道上调用 SsmRecToMem 函数, 因此, 相关的输出消息也在主叫方通道上返回		
	1: 只录制被叫方的声音。驱动程序自动在与 nCic 绑定的被叫方通道上调用 SsmRecToMem 函数, 因此, 相关的输出消息也在被叫方通道上返回		
	2: 混合录制主叫方与被叫方的声音。驱动程序自动在与 nCic 绑定的被叫方通道上调用 SsmRecToMem 函数, 相关的输出消息在主叫方通道上返回		

返回值:

-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

启动内存录音任务。SpyRecToMem 启动基于电路编号的内存录音任务, 只适用于 DTP 系列板卡。

调用本函数后, 驱动程序内部会启动一个内存录音任务, 从参数 dwStartOffset 指定的偏移量开始将录音数据写

入到 pBuf，同时维护一个初始值等于 dwStartOffset 的录音指针（偏移量），用于向提交的缓冲区中写入数据。当缓冲区写满后，自动从缓冲区头部开始循环写入。

当发生下列事件时，内存录音任务终止：

- ◆ 应用程序调用 [SsmStopRecToMem](#)；
- ◆ 驱动程序检测到 DTMF 或对端挂机。详细内容请参见第 1 章中[设置录音任务的终止条件](#)部分。

函数 [SsmCheckRecord](#) 可以用来查询内存录音任务的完成情况。

在内存录音任务的执行期间，每当满足设定的条件时，驱动程序都会向应用程序抛出 [E PROC_RecordMem](#) 事件。[E PROC_RecordMem](#) 事件的输出条件可以通过函数 [SsmSetEvent](#)（携带 E_PROC_RecordMem 参数）进行设置，可以选择下列 3 种方式之一：

- ◆ 当驱动程序完成了指定时间长度的语音数据的录制；
- ◆ 驱动程序内部的录音指针越过缓冲区的 dwBufSize /2 位置；
- ◆ 驱动程序内部的录音指针到达缓冲末尾。

函数 [SsmStopRecToMem](#) 可以用来终止内存录音任务，函数 [SsmGetRecOffset](#) 可以用来获取驱动程序内部的录音指针。

注意事项：

- 有关选择录音信号源、设置音量的详细内容请参见第 1 章中[设置录音信号源](#)部分；
- SpyRecToMem 启动基于电路编号的内存录音任务，只适用于 DTP 系列板卡。SpyRecToMem 函数是基于电路编号的函数，它向应用程序抛出的消息是基于呼叫过程中与 nCic 绑定的主叫方或被叫方通道。函数 [SpyGetCallInCh](#)、[SpyGetCallOutCh](#) 分别用于获取本次呼叫中的主叫方和被叫方的通道编号。函数 [SpyStopRecToMem](#) 用于停止 SpyRecToMem 函数启动的内存录音任务；
- 由于历史原因，G729A 曾经使用过的语音格式参数值 65411 在低版本依然有效，版本向下兼容。如果使用 G729A 格式录音，建议采用 131；
- 该函数同时也支持 SynIPRecorder，此时它返回成功只是表示它已经往 Slaver 发送开始录音到内存的命令，至于该命令在 Slaver 端是否成功处理需要通过事件 [E IPR_START_REC_CB](#) 上携带的 dwParam 的值来确定。

相关信息：

驱动程序版本要求	SsmRecToMem 要求的 SynCTI 版本为 Ver. 2.0 或更高； SpyRecToMem 要求的 SynCTI 版本为 Ver. 5.3.1.3 或更高。
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmStopRecToMem](#), [SsmGetRecOffset](#), [SsmSetDTMFStopRecCharSet](#), [SsmSetHangupStopRecFlag](#),
[SsmSetRecVolume](#), [SsmSetRecMixer](#), [SsmSetRecBack](#)

2.10.4.3 SsmStopRecToMem

请参见 [SpyStopRecToMem](#)

2.10.4.4 SpyStopRecToMem

终止内存录音任务。

函数原型：

```
int SsmStopRecToMem(int ch)  
int SpyStopRecToMem(int nCic)
```

参数说明:

ch	通道编号
nCic	SpyCic 的逻辑编号。更多信息请参见第 1 章中“ DTP 系列板卡 ”部分内容

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

终止内存录音任务。SpyStopRecToMem 用于终止调用 [SpyRecToMem](#) 函数启动的内存录音任务。

注意事项:

- 录音操作停止后，驱动程序会抛出 [E PROC RecordEnd](#) 事件；
- 该函数同时也支持 SynIPRecorder，此时它返回成功只是表示它已经往 Slaver 发送停止录音到内存的命令，至于该命令在 Slaver 端是否成功处理需要通过事件 [E IPR STOP REC CB](#) 上携带的 dwParam 的值来确定。

相关信息:

驱动程序版本要求	SsmStopRecToMem 要求的 SynCTI 版本为 Ver. 2.0 或更高； SpyStopRecToMem 要求的 SynCTI 版本为 Ver. 5.3.1.3 或更高。
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmRecToMem](#)

2.10.4.5 SsmGetRecOffset

获取内存录音任务执行过程中，驱动程序内部的录音指针。

函数原型:

```
int SsmGetRecOffset(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
≥0	驱动程序内部的录音指针，其值为相对于 SsmRecToMem 函数提交的缓冲区首地址的偏移量，单位为字节

功能描述:

获取内存录音任务执行过程中，驱动程序内部的录音指针。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmRecToMem](#)

2.10.5 录制到指定乒乓缓存的函数 (CTI 系列)

2.10.5.1 SsmRecordMemBlock

向驱动程序提交一个缓冲区，并启动乒乓内存录音任务。

函数原型：

```
int SsmRecordMemBlock(int ch, int nFormat, LPBYTE pBuf, DWORD dwBufSize,  
RECORDMEMBLOCKHANDLER pfnCallback, PVOID pV)
```

参数说明：

ch	通道编号																														
nFormat	录音数据的编码格式。取值范围： <table border="1"><thead><tr><th>取值</th><th>编码格式</th><th>备注</th></tr></thead><tbody><tr><td>1</td><td>PCM8</td><td></td></tr><tr><td>-2</td><td>PCM16</td><td></td></tr><tr><td>6</td><td>A-Law</td><td></td></tr><tr><td>7</td><td>μ-Law</td><td></td></tr><tr><td>17</td><td>IMA ADPCM</td><td>要求板卡具有硬件编码能力</td></tr><tr><td>23</td><td>VOX</td><td>要求板卡具有硬件编码能力</td></tr><tr><td>49</td><td>GSM</td><td></td></tr><tr><td>85</td><td>MP3</td><td>要求板卡具有硬件编码能力</td></tr><tr><td>131</td><td>G.729A</td><td>要求板卡具有硬件编码能力</td></tr></tbody></table>	取值	编码格式	备注	1	PCM8		-2	PCM16		6	A-Law		7	μ-Law		17	IMA ADPCM	要求板卡具有硬件编码能力	23	VOX	要求板卡具有硬件编码能力	49	GSM		85	MP3	要求板卡具有硬件编码能力	131	G.729A	要求板卡具有硬件编码能力
取值	编码格式	备注																													
1	PCM8																														
-2	PCM16																														
6	A-Law																														
7	μ-Law																														
17	IMA ADPCM	要求板卡具有硬件编码能力																													
23	VOX	要求板卡具有硬件编码能力																													
49	GSM																														
85	MP3	要求板卡具有硬件编码能力																													
131	G.729A	要求板卡具有硬件编码能力																													
pBuf	录音缓冲区指针																														
dwBufSize	录音缓冲区大小，单位为字节。dwBufSize 的最小值按照下面规则确定： 如果配置项 RecordAndPlayUseAsIP 设置为 0（缺省值），对于 A-率、μ-率、PCM8、PCM16、G729A、mp3、GC8、VOX、IMA ADPCM，dwBufSize 应不低于 768 字节。对于硬件 GSM 编码，dwBufSize 应不低于 260 字节；对于软件 GSM 编码，dwBufSize 应不低于 2080 字节。 RecordAndPlayUseAsIP 设置为 1，对于 A-率、μ-率、PCM8 和 PCM16，dwBufSize 应不低于 192 字节（推荐值：Windows 操作系统下应不低于 192 字节，Linux 下应不低于 384 字节）对于 IMA ADPCM、VOX、G729A、mp3、GC8 格式，缓冲区大小应不低于 768 字节。对于硬件 GSM 编码，dwBufSize 应不低于 260 字节；对于软件 GSM 编码，dwBufSize 应不低于 2080 字节。																														
pfnCallback	回调函数指针，NULL 表示不设置回调函数。如果本参数不为 NULL，当驱动程序完成一个缓冲区的录制后或该缓冲区的录音任务被终止时会自动调用本参数设置的回调函数。回调函数的原型在 shpa3api.h 中声明为： <pre>typedef BOOL (WINAPI * RECORDMEMBLOCKHANDLER) (int ch, int nEndReason, PUCHAR pucBuf, DWORD dwStopOffset, PVOID pVoid);</pre> 其中， ch: 通道编号 nEndReason: 结束原因。取值范围： <ol style="list-style-type: none">1: 录音任务被应用程序终止2: 录音任务因检测到 DTMF 按键字符而结束3: 录音任务因检测到对端挂机而结束4: 缓冲区录音完毕 pucBuf: 录制完毕的缓冲区指针 dwStopOffset: 缓冲区录制完毕时，驱动程序内部的录音指针在缓冲区中偏移量，单位为字节。如果																														

	返回值为 0xffffffff，则为无效值。
pVoid	pVoid 类型指针，通常用于应用程序向回调函数传递其数据结构。这个指针将由驱动程序透明地传递给回调函数

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

向驱动程序提交一个缓冲区，并启动乒乓缓冲区录音任务。有关乒乓缓冲区录音的详细内容请参见第 1 章中[乒乓缓冲区录音](#)部分。

当发生下列事件时，乒乓缓冲区录音任务终止：

- ◆ 驱动程序完成全部已提交的缓冲区的录制；
- ◆ 应用程序调用 [SsmStopRecordMemBlock](#)；
- ◆ 驱动程序检测到 DTMF 或对端挂机。详细内容请参见第 1 章中[设置录音任务的终止条件](#)部分。

函数 [SsmCheckRecord](#) 可以用来查询内存序列播放任务的完成情况。

注意事项:

- 有关选择录音信号源、设置音量的详细内容请参见第 1 章中[设置录音信号源](#)部分；
- 应用程序可以随时调用函数 [SsmStopRecordMemBlock](#) 终止乒乓缓冲区录音任务，该函数不能在回调函数中调用；
- 在回调函数中，应用程序的代码执行时间应尽量短，不能大于驱动程序的硬件中断时间（8 毫秒），否则可能引起驱动程序出错；
- 乒乓缓冲区录音任务终止后，驱动程序会向应用程序抛出 [E PROC RecordEnd](#) 事件；
- 不能同时向驱动提交两个以上的缓冲区；
- 不能使用定时器来调用 [SsmRecordMemBlock](#) 函数；
- 由于历史原因，G729A 曾经使用过的语音编码参数值 65411 在低版本依然有效，版本向下兼容。如果使用 G729A 格式录音，建议采用 131。

代码范例:

```
.....//初始化缓冲区 szRecBuf1, szRecBuf2。  
BOOL RecStart(int nCh)  
{  
    if(SsmRecordMemBlock(nCh, 6, szRecBuf1, dwBufSize, RecordMemBlockHandler, NULL) != 0)//向驱动程序  
    提交一个缓冲区 szRecBuf1，并启动乒乓内存录音任务。  
    {  
        SsmGetLastErrMsg(szErrMsg);  
        cout<<szErrMsg<<endl;  
        return FALSE;  
    }  
    if(SsmRecordMemBlock(nCh, 6, szRecBuf2, dwBufSize, RecordMemBlockHandler, NULL) != 0)//向驱动程序  
    提交一个缓冲区 szRecBuf2。  
    {  
        SsmGetLastErrMsg(szErrMsg);  
        cout<<szErrMsg<<endl;
```

```
return FALSE;
}
return TRUE;
}
BOOL RecordMemBlockHandler(int ch, int nEndReason, PUCHAR pucBuf, DWORD dwStopOffset, PVOID pV)
{
    int nResult = 0;
    if(pucBuf == szRecBuf1)
    {
        .....//保存 szRecBuf1 内数据
        nResult = SsmRecordMemBlock(ch, 6, szRecBuf1, dwBufSize, RecordMemBlockHandler, pV); //向驱动程序
        提交一个缓冲区 szRecBuf1。
    }
    else if(pucBuf == szRecBuf2)
    {
        .....//保存 szRecBuf2 内数据
        nResult = SsmRecordMemBlock(ch, 6, szRecBuf2, dwBufSize, RecordMemBlockHandler, pV); //向驱动程序提
        交一个缓冲区 szRecBuf2。
    }
    return nResult;
}
```

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmStopRecordMemBlock](#), [SsmRecToMem](#)

2.10.5.2 SsmStopRecordMemBlock

终止乒乓缓冲区录音任务。

函数原型:

```
int SsmStopRecordMemBlock (int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

终止乒乓缓冲区录音任务。

注意事项:

- 本函数不能在 [SsmRecordMemBlock](#) 设置的回调函数中调用；

- 录音操作停止后，驱动程序会抛出 [E_PROC_RecordEnd](#) 事件。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmRecordMemBlock](#)

2.11 有关交换总线的函数

2.11.1 双向连接函数（CTI 系列）

2.11.1.1 建立双向连接

2.11.1.1.1 SsmTalkWith

请参见 [SsmTalkWithEx](#)

2.11.1.1.2 SsmTalkWithEx

建立 2 个通道之间的双向连接。SsmTalkWithEx 可以设置参与交换的 2 个通道的音量。

函数原型：

```
int SsmTalkWith(int ch1,int ch2)  
int SsmTalkWithEx(int ch1, int nVlm1, int ch2, int nVlm2)
```

参数说明：

ch1	通道编号 1
ch2	通道编号 2
nVlm1	ch1 的音量，取值范围：-7~+6，负值表示衰减，正值表示提升，缺省值为 0。取值乘以 3 即为分贝值
nVlm2	ch2 的音量，取值范围：-7~+6，负值表示衰减，正值表示提升，缺省值为 0。取值乘以 3 即为分贝值

返回值：

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述：

建立指定通道 ch1 到 ch2 的双向连接，即 ch1 可以听到 ch2 的声音，ch2 也可以听到 ch1 的声音。如果 ch1 或 ch2 上的来话信号还没有上行到交换总线，驱动程序会自动为其申请一个交换总线时隙，并自动将来话信号上行到该时隙。

注意事项：

- 本函数只适用于 SHT/SHD/SHN 系列板卡，参数 nVlm1 和 nVlm2 用来设置第 1 章 “[SHT 系列板卡的工作原理框图](#)” 或 “[SHD 系列板卡的工作原理框图](#)” 或 “[SHN 系列板卡的工作原理框图](#)” 中的音量调节器

A4-1...A4-6。A4-1...A4-6 的增益也可以通过配置项 [DefaultSpeakVolume](#) 进行设置，缺省值为 0DB；

- 如果板卡的型号中不包含“-CT”字样，则 ch1 和 ch2 必须在同一片板卡上，否则函数会失败；
- 拆除连接时，应使用 [SsmStopTalkWith](#) 函数，并且必须保证建立/断开连接的操作是成对出现、一一对应的。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmStopTalkWith](#), [SsmListenTo](#), [SsmListenToEx](#), [SsmLinkFrom](#), [SsmLinkFromEx](#)

2.11.1.2 拆除双向连接

2.11.1.2.1 SsmStopTalkWith

拆除 2 个通道之间的双向连接。

函数原型：

```
int SsmStopTalkWith(int ch1,int ch2)
```

参数说明：

ch1	听者/说者通道编号
ch2	说者/听者通道编号

返回值：

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述：

拆除 ch1 到 ch2 的双向连接。

注意事项：

- 本函数只适用于 SHT/SHD/SHN 系列板卡。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmTalkWith](#), [SsmTalkWithEx](#)

2.11.2 单向连接函数

2.11.2.1 建立单向连接

2.11.2.1.1 SsmListenTo

请参见 [SsmLinkFromAllCh](#)

2.11.2.1.2 SsmListenToEx

请参见 [SsmLinkFromAllCh](#)

2.11.2.1.3 SsmLinkFrom

请参见 [SsmLinkFromAllCh](#)

2.11.2.1.4 SsmLinkFromEx

请参见 [SsmLinkFromAllCh](#)

2.11.2.1.5 SsmLinkFromAllCh

建立说者通道到听者通道的单向连接。使用 `SsmListenTo`、`SsmListenToEx` 建立两个通道之间的单向连接，并且一个通道只能听一个通道；使用 `SsmLinkFrom`、`SsmLinkFromEx` 建立的两个通道之间的单向连接，一个通道可以同时听多个通道。`SsmListenToEx`、`SsmLinkFromEx` 还可以在设置单向连接的音量。`SsmLinkFromAllCh` 可以一次建立单个说者通道到多个听者通道单向连接。

函数原型：

```
int SsmListenTo(int nSourceCh, int nListeningCh)
int SsmListenToEx(int nSourceCh, int nVolume, int nListeningCh)
int SsmLinkFrom(int nSourceCh, int nListeningCh)
int SsmLinkFromEx(int nSourceCh, int nVolume, int nListeningCh)
int SsmLinkFromAllCh(int nSourceCh, int nVolume, int* pnListenerTable,int nListenerNum)
```

参数说明：

nSourceCh	说者通道编号
nListeningCh	听者通道编号 注意：如果 nListeningCh 为 ATP 或 DST 系列板卡的通道 0，说者的声音将被送到板载扬声器端口
nVolume	说者的音量，取值范围：-7~+6，正值表示提升，负值表示衰减，缺省值为 0。取值乘以 3 即为分贝值。 <code>nVolume</code> 用来设置来话信号经过总线后下到另外一个通道的下总线混音器 M2 时的音量，详细内容请参见第 1 章中板卡的工作原理框图部分。 A4-1...A4-6 的增益也可以通过配置项 DefaultSpeakVolume 进行设置，缺省值为 0dB
pnListenerTable	听者通道表指针，存储空间由应用程序分配。听者通道必须连续存放
nListenerNum	<code>pnListenerTable</code> 中听者通道的数目

返回值：

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
----	----------------------------------------------------

0

调用成功

功能描述:

建立说者通道到听者通道的单向连接，即用 nListeningCh 听 nSourceCh 的声音，但 nSourceCh 无法听到 nListeningCh 的声音。

SsmListenToEx 除了具备 SsmListenTo 的全部功能外，还能设置连接时的音量。使用 SsmListenTo、SsmListenToEx 建立的单向连接，一个通道在同一时刻只能听一个通道。使用 SsmListenTo、SsmListenToEx 建立的连接，应使用 [SsmStopListenTo](#) 函数来拆除连接，并且必须保证建立/断开连接的操作是成对出现、一一对应的。

SsmLinkFromEx 除了具备 SsmLinkFrom 的全部功能外，还能设置连接时的音量。使用 SsmLinkFrom、SsmLinkFromEx 建立的单向连接应使用 [SsmStopLinkFrom](#) 函数来拆除连接，并且必须保证建立/断开连接的操作是成对出现、一一对应的。

SsmLinkFromAllCh 可以一次建立单个说者通道到多个听者通道单向连接，相当于对多个听者通道连续调用 SsmLinkFromEx。使用 SsmLinkFromAllCh 建立的单向连接应使用 [SsmUnLinkFromAllCh](#) 函数来拆除连接，并且必须保证建立/断开连接的操作是成对出现、一一对应的。

SsmLinkFrom、SsmLinkFromEx、SsmLinkFromAllCh 允许一个通道同时听多个说者通道的声音，这些说者通道的声音通过 TDM 总线交换到听者通道的会议混音器中，经过混音后的声音作为去话送给听者。

注意事项:

- 对于 SHT/SHD/SHN 系列板卡，由于下总线混音器 M2 具有 6 个输入信号源 A₄₋₁~A₄₋₆，因此，每个通道最多可以同时听 6 个通道的声音，即，一个通道最多可以调用 6 次 SsmLinkFrom、SsmLinkFromEx 函数。详细内容请参见第 1 章中“[SHT 系列板卡的工作原理框图](#)”部分或“[SHD 系列板卡的工作原理框图](#)”部分或“[SHN 系列板卡的工作原理框图](#)”部分。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SmStopListenTo](#), [SsmStopLinkFrom](#), [SsmUnLinkFromAllCh](#), [SsmTalkWith](#), [SsmTalkWithEx](#)

代码范例:

本示例代码实现下列功能：

- 通道 0 听通道 5 的声音；
- 通道 1 听通道 6 的声音（音量提升 6 个单位）；
- 通道 2 想要同时听通道 9、通道 10、通道 11 的声音；
- 通道 3 想要同时听通道 9（正常音量）、通道 10（音量衰减 1 个单位）、通道 11（音量提升 2 个单位）；
- 通道 4 的声音要同时送给通道 12、13、14 听（缺省音量）

代码实现：

```
SsmListenTo ( 5, 0);           // 为听者通道 0 建立连接
SsmListenToEx ( 6, 6, 1);      // 为听者通道 1 建立连接
SsmLinkFrom ( 9, 2);          // 为听者通道 2 建立连接
SsmLinkFrom ( 10, 2);
SsmLinkFrom ( 11, 2);
```

```
SsmLinkFromEx ( 9, 0, 3);           // 为听者通道 3 建立连接
SsmLinkFromEx (10, -1, 3);
SsmLinkFromEx (11, 2, 3);

int ListenerChList[3]={12, 13, 14};    // 建立说者通道 4 到听者通道 12、13、14 的单向连接
SsmLinkFromAllCh (4, 0, ListenerChList, 3);

....                                     // 应用程序的其它操作

SmStopListenTo ( 5, 0);                // 为听者通道 0 拆除连接
SmStopListenTo ( 6, 1);                // 为听者通道 0 拆除连接
SsmStopLinkFrom ( 9, 2);                // 为听者通道 1 拆除连接
SsmStopLinkFrom (10, 2);
SsmStopLinkFrom (11, 2);

SsmStopLinkFrom ( 9, 3);                // 为听者通道 2 拆除连接
SsmStopLinkFrom (10, 3);
SsmStopLinkFrom (11, 3);

SsmLinkFromAllCh (4, ListenerChList, 3); // 拆除通道 4 到听者通道 12、13、14 的单向连接
```

2.11.2.2 拆除单向连接

2.11.2.2.1 SsmStopListenTo

请参见 [SsmUnLinkFromAllCh](#)

2.11.2.2.2 SsmStopLinkFrom

请参见 [SsmUnLinkFromAllCh](#)

2.11.2.2.3 SsmUnLinkFromAllCh

拆除说者通道到听者通道的单向连接。SsmStopListenTo 用于拆除 SsmListenTo 或 SsmListenToEx 建立的单向连接，SsmStopLinkFrom 用于拆除 SsmLinkFrom 或 SsmLinkFromEx 建立的单向连接，SsmUnLinkFromAllCh 用于拆除 SsmLinkFromAllCh 建立的单向连接。

函数原型:

```
int SsmStopListenTo(int nSourceCh,int nListeningCh)
int SsmStopLinkFrom(int nSourceCh,int nListeningCh)
int SsmUnLinkFromAllCh (int nSourceCh, int* pnListenerTable,int nListenerNum)
```

参数说明:

nSourceCh	说者通道编号
nListeningCh	听者通道编号
pnListenerTable	听者通道表指针，存储空间由应用程序分配。听者通道必须连续存放
nListenerNum	pnListenerTable 中听者通道的数目

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

拆除说者通道到听者通道的单向连接。

注意事项:

- 建立单向连接的函数和拆除单向连接的函数必须配对使用。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmListenTo](#), [SsmListenToEx](#), [SsmLinkFrom](#), [SsmLinkFromEx](#), [SsmLinkFromAllCh](#)

2.11.3 获取总线信息函数

2.11.3.1 SsmGetChBusInfo

获取与 ch 通道相关的总线信息。

函数原型:

```
int WINAPI SsmGetChBusInfo(int ch, PBUS_OP* p)
```

参数说明:

ch	通道编号
p	Bus 结构指针

返回值:

-1	没有监听其他通道
-2	通道不支持该函数
>=0	监听的通道号

功能描述:

获取与 ch 通道相关的总线信息。

Bus 结构说明

{	BOOL	bEnHwOpBus;	总线开关
	BOOL	bEnHwOpSetLinkFromVlm;	总线连接音量设置开关
	int	nST;	总线默认连接的码流
	int	nTs;	总线默认连接的时隙
	int	nToBusCh;	默认连接所占用的总时隙
	int	nPlayST;	录音模块放音上总线默认连接的码流
	int	nPlayTs;	录音模块放音上总线默认连接的时隙
	int	nPlayToBusCh;	录音模块放音上总线所占用的总时隙
	int	nSpeakerVlm;	音量
	int	nTotListener;	总线连接聆听本通道的总听众数
	int	* pnListenerCh;	总线连接聆听本通道的通道号
	int	nFromSpeaker;	说者通道的通道号
	int	nDefaultSpeakerVlm;	总线连接音量
	int	nTotChS;	内部使用
	int	*pnChS;	内部使用

int	nBindCh;	内部使用
int	nToBusChForVox;}	内部使用

该 Bus 结构是驱动内部定义结构，有些对应用层是没有意义的，用户可不必关心。

注意事项：

- 此函数只支持两个通道之间的总线判定，比如 [SsmListenTo](#)、[SsmLinkFrom](#)、[SsmTalkWith](#) 等有效，对会议室成员的总线判断无效。

相关信息：

驱动程序版本要求	SynCTI Ver.3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.11.4 高级总线操作函数（CTI 系列）

2.11.4.1 强制拆除通道的所有总线连接

2.11.4.1.1 SsmClearChBusLink

拆除通道上建立的所有总线连接，使通道恢复到驱动程序初始化完成时的总线连接状态。

函数原型：

```
int SsmClearChBusLink(int nCh)
```

参数说明：

nCh	监听通道编号
-----	--------

返回值：

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述：

拆除在通道 nCh 上调用任何函数所建立的总线连接，包括：

- > [SsmListenTo](#)、[SsmListenToEx](#)、[SsmLinkFrom](#)、[SsmLinkFromEx](#)、[SsmLinkFromAllCh](#)
- > [SsmTalkWith](#)、[SsmTalkWithEx](#)

使通道恢复到驱动程序初始化完成时的总线连接状态。

注意事项：

- 如果通道已经加入了一个电话会议室中，其总线连接也将被强制拆除。

相关信息：

驱动程序版本要求	SynCTI Ver.4.5.6.2 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.11.4.2 通道到总线时隙的单向连接

2.11.4.2.1 SsmLinkToBus

改变上总线混音器的输出信号上行到总线的时隙。

函数原型:

```
int SsmLinkToBus(int ch, int ts)
```

参数说明:

ch	通道编号
ts	可用的 TDM 总线公共时隙编号, 取值范围: N~4095, N 为系统中的通道总数的 2 倍。 有关可用的公共时隙编号请参见第 1 章中 “ TDM 交换总线 ” 部分

返回值:

-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

板卡初始化时已经将通道 ch 的上总线混音器的输出信号连接到了某个公共时隙 (假设为 ts0), 本函数将该输出信号重新连接到时隙 ts。拆除连接应使用 [SsmUnLinkToBus](#) 函数。

注意事项:

- 时隙 ts 必须是未被使用的公共时隙;
- 本函数必须与 [SsmUnLinkToBus](#) 成对调用, 否则会引起信号混乱;
- 本函数是底层函数, 只适用于将 Synway 板卡与第 3 方板卡进行互联的场合;
- 配置项 [EnableCommonTimeSlot](#) 可以开放 TDM 总线上的所有公共时隙。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmUnLinkToBus](#), [SsmLinkFromBus](#), [SsmLinkFromBusEx](#)

2.11.4.2.2 SsmUnLinkToBus

恢复将上总线混音器的输出信号连接到缺省的总线时隙。

函数原型:

```
int SsmUnLinkToBus(int ch, int ts)
```

参数说明:

ch	通道编号
ts	时隙编号

返回值:

-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

拆除通道 ch 到总线时隙 ts 的单向连接，并恢复将上总线混音器的输出信号连接到缺省的总线时隙。

注意事项:

- 本函数必须与 [SsmLinkToBus](#) 成对调用，否则会引起信号混乱；
- 本函数是底层函数，只适用于将 Synway 板卡与第 3 方板卡进行互联的场合。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmLinkToBus](#)

2.11.4.3 总线时隙到通道的单向连接

2.11.4.3.1 SsmLinkFromBus

请参见 [SsmLinkFromBusEx](#)

2.11.4.3.2 SsmLinkFromBusEx

建立总线时隙到听者通道的单向连接。SsmLinkFromBusEx 除了具备 SsmLinkFromBus 的全部功能外，还可以设置音量。

函数原型:

```
int SsmLinkFromBus(int ts, int ch)  
int SsmLinkFromBusEx(int ts, int ch, int nVolume)
```

参数说明:

ch	通道编号
ts	时隙编号
nVolume	ts 时隙上的数据进入通道 ch 时的音量，取值范围：-6~+6，正值表示提升，负值表示衰减，缺省值为 0。取值乘以 3 即为分贝值

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

建立总线时隙 ts 到听者通道 ch 的单向连接。

注意事项:

- SsmLinkFromBus 或 SsmLinkFromBusEx 必须与 [SsmUnLinkFromBus](#) 成对调用，否则会引起信号混乱。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmLinkToBus](#), [SsmUnLinkFromBus](#)

2.11.4.3.3 SsmUnLinkFromBus

拆除总线时隙到听者通道的单向连接。

函数原型:

```
int SsmUnLinkFromBus(int ts, int ch)
```

参数说明:

ch	通道编号
ts	时隙编号

返回值:

-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

拆除总线时隙 ts 到听者通道 ch 的单向连接。

注意事项:

- [SsmLinkFromBus](#) 或 [SsmLinkFromBusEx](#) 必须与 SsmUnLinkFromBus 成对调用, 否则会引起信号混乱。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmLinkFromBus](#), [SsmLinkFromBusEx](#)

2.11.4.3.4 SsmListenToPlay

实现通道对录音通道的放音监听。

函数原型:

```
int WINAPI SsmListenToPlay(int ch1, int vlm, int ch2)
```

参数说明:

ch1	监听通道编号
ch2	放音通道编号, 要求该通道类型为录音通道
vlm	放音的音量

返回值:

-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

实现通道 ch1 对录音通道 ch2 的放音监听。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmUnListenToPlay](#)

2.11.4.3.5 SsmUnListenToPlay

结束通道对录音通话的放音监听。

函数原型:

```
int WINAPI SsmUnListenToPlay(int ch1, int ch2)
```

参数说明:

ch1	监听通道编号
ch2	放音通道编号, 要求该通道类型为录音通道

返回值:

-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

结束通道 ch1 对录音通道 ch2 的放音监听。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmListenToPlay](#)

2.12 有关板载音频功率放大器的函数

2.12.1 SsmQueryOpPowerAmp

查询通道是否具有板载音频放大电路。

函数原型:

```
int SsmQueryOpPowerAmp(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
0	本通道没有板载音频放大电路
1	本通道具有板载音频放大电路

功能描述:

查询通道是否具有板载音频放大电路。有关板载音频放大电路的内容请参见对应板卡的原理框图部分。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetPowerAmpVlm](#), [SetVolume](#)

2.12.2 SsmSetPowerAmpVlm

请参见 [SetVolume](#)

2.12.3 SetVolume

设置板载功率放大器的增益。SetVolume 是较早版本的函数，推荐使用 SsmSetPowerAmpVlm。

函数原型:

```
void SsmSetPowerAmpVlm(int ch, int nGain)  
void SetVolume(DWORD dwBoardId, DWORD nGain)
```

参数说明:

ch	通道编号 注意: ch 对应的板卡内部编号必须为 0, 否则函数会失败
dwBoardId	板卡 ID 编号 功率放大器的增益, 取值范围: 0~7, 其中: 7: 0DB 6: -6DB 5: -12DB 4: -18DB 3: -24DB 2: -30DB 1: -36DB 0: 关闭 缺省值为 3 (-24DB)
nGain	

返回值: 无

功能描述:

设置板载功率放大器的增益。有关板载音频放大电路的内容请参见对应板卡的原理框图部分。

注意事项:

- 只有 ATP 系列、SHT 系列或 DST 系列板卡的 0 通道上才具有板载功率放大器。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmQueryOpPowerAmp](#)

2.13 有关板载扬声器的函数

2.13.1 SsmSetLine0OutTo

关闭或开启 USB 录音盒/语音盒中的内置板载扬声器。

函数原型:

```
int WINAPI SsmSetLine0OutTo(BOOL bEnable)
```

参数说明:

bEnable	=FALSE: 将声音送到板载扬声器进行播放 =TRUE: 将声音送到板载扬声器插座, 由外置扬声器进行播放
---------	-----------------------------------------------------------

返回值:

-1	调用失败
0	操作成功

功能描述:

开启或关闭 USB 录音盒/语音盒中的内置板载扬声器。更多信息请参见第 1 章中 “[ATP 系列板卡的工作原理图](#)” 或 “[SHT 系列板卡的工作原理图](#)” 部分内容。

注意事项:

- 本函数只适用于 ATP 系列的 USB 录音盒、SHT 系列的 USB 语音盒;
- 配置项 [USBLine0Output](#) 可以实现相同功能。

相关信息:

驱动程序版本要求	SynCTI Ver. 4.7.1.7 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetPowerAmpVlm](#)

2.14 有关回波抵消器（Echo Canceller）的函数（CTI 系列）

2.14.1 设置回波抵消器的工作状态

2.14.1.1 SsmQueryOpEchoCanceller

查询通道是否支持回波抵消。

函数原型:

```
int SsmQueryOpEchoCanceller(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
0	不支持
1	支持

功能描述:

查询通道是否支持回波抵消。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.14.1.2 SsmSetEchoCanceller

设置回波抵消器的工作状态。

函数原型:

```
int SsmSetEchoCanceller(int ch, BOOL bEnable)
```

参数说明:

ch	通道编号
bEnable	TRUE: 开启 FALSE: 关闭

返回值:

-1	调用失败
0	调用成功

功能描述:

设置回波抵消器的工作状态。

注意事项:

- 配置项 [EnableEchoCanceller](#) 可以实现相同功能，缺省值为开启；

- 详细内容请参见第1章中回波抵消器部分；
- 本函数只支持 SHD、SHT 系列板卡。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmGetEchoCancellerState](#)

2.14.1.3 SsmGetEchoCancellerState

获取回波抵消器的工作状态。

函数原型：

```
int SsmGetEchoCancellerState(int ch)
```

参数说明：

ch	通道编号
----	------

返回值：

-1	调用失败
0	关闭
1	开启

功能描述：

获取回波抵消器的工作状态。

注意事项：

- 本函数只支持 SHD、SHT 系列板卡。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmSetEchoCanceller](#)

2.14.2 设置回波抵消器的工作参数

2.14.2.1 SsmSetEchoCancelDelaySize

设置回波抵消器的滤波延迟系数。

函数原型：

```
int SsmSetEchoCancelDelaySize(int ch,WORD wSize)
```

参数说明：

ch	通道编号
wSize	滤波延迟系数，取值范围：0~31

返回值：

-1	调用失败
----	------

0	成功
---	----

功能描述:

设置回波抵消器的滤波延迟系数。

注意事项:

- 本函数只支持 SHD、SHT 系列板卡；
- 配置项 [EchoCancelDelaySize](#) 可以实现相同功能。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetEchoCancelDelaySize](#)

2.14.2.2 SsmGetEchoCancelDelaySize

获取回波抵消器的滤波延迟系数。

函数原型:

```
int SsmGetEchoCancelDelaySize(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
≥ 0	滤波延迟系数

功能描述:

获取指定通道上的滤波延迟系数。

注意事项:

- 本函数只支持 SHD、SHT 系列板卡。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetEchoCancelDelaySize](#)

2.15 有关电话会议的函数 (CTI 系列)

2.15.1 会议室初始化函数

2.15.1.1 SsmCreateConfGroup

创建一个会议室。

函数原型:

int SsmCreateConfGroup(int nMaxMember, int nMaxSpeaker, int nMaxSpeaking, int nMaxSilenceTime)

参数说明:

nMaxMember	会议室中的最大通道数, 取值范围: -1: 使用配置项 ConfDefaultMaxGroupMember 指定的参数 3~N: N 为应用系统中的通道总数
nMaxSpeaker	设置具有发言权的会议通道的最大个数。主持人、主席、动态发言模式均具有发言权, 背景音乐模式是否具有发言权取决于配置项 BackgroundVoicePriority 的设置。取值范围: -1: 使用配置项 ConfDefaultMaxGroupSpeaker 指定的参数 ≥1: 会议通道总数
nMaxSpeaking	允许同时发言的最大成员数, 取值范围: -1: 使用配置项 ConfDefaultMaxGroupSpeaking 指定的参数 1~6: 设定值
nMaxSilenceTime	静默的最小保持时间。取值范围: -1: 使用配置项 ConfDefaultMaxSilenceTime 指定的参数 ≥1: 设定值, 单位为秒 更多信息请参见第 1 章中“ 分布式电话会议 ”部分内容 注: 目前该参数已经无效, 直接设置为-1 即可。

返回值:

-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
≥0	新创建的电话会议室的 ID 号

功能描述:

创建一个会议室。

注意事项: 无

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmFreeConfGroup](#), [SsmJoinConfGroup](#), [SsmGetConfGrpInfo](#), [SsmGetConfGrpCfgInfo](#)

2.15.1.2 SsmFreeConfGroup

注销一个会议室。

函数原型:

int void SsmFreeConfGroup(int nGrpId)

参数说明:

nGrpId	会议室编号
--------	-------

返回值:

-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

注销一个会议室, 释放相关的资源。

注意事项:

- 如果会议室中还有尚未退出会议的通道, 驱动程序会自动调用 [SsmExitConfGroup](#) 函数将其退出会议室。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
----------	---------------------

Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmCreateConfGroup](#)

2.15.2 会议室成员操作函数

2.15.2.1 SsmJoinConfGroup

将通道加入到会议室。

函数原型:

```
int SsmJoinConfGroup(int nGrpId, int ch, WORD wJoinMode, int nMixerVolume, BOOL bCreateAlways, BOOL bExitGrpAlways)
```

参数说明:

nGrpId	会议室编号
ch	通道编号 通道的发言模式, 取值范围: 0: 主持人模式; 1: 动态发言模式; 2: 听众模式; 3: 背景音乐模式, 由配置项 PlayVoiceIsListen 决定能否听取其它通道的发言: 当 PlayVoiceIsListen=0 时, 只有发言权, 无法听取其他通道的声音, 用于对会议放音; 当 PlayVoiceIsListen=1 时, 既有发言权, 也能听取其他通道的声音 (驱动默认方式)。 4: 主席模式 5: 动态只发言模式 有关发言模式及会议调度算法的更多信息请参见第 1 章中 “ 分布式电话会议 ” 部分内容
wJoinMode	本通道来话信号进入其他成员会议下总线混音器 M2 前的音量, 取值范围: -6~+6, 大于 0 表示音量提升, 小于 0 表示音量衰减。取值乘以 3 即为分贝值。该音量为会议组中的其他人听到该成员说话时的音量
nMixerVolume	调用本函数时, 如果参数 nGrpId 指定的会议室尚未创建, 本参数指示是否由驱动程序以 nGrpId 作为会议室编号自动创建一个新的会议室。 TRUE: 是 FALSE: 否, 函数将返回失败
bCreateAlways	如果通道已经加入到其它的会议室, 本参数指示是否由驱动程序自动将通道退出该会议室。 TRUE: 是 FALSE: 否, 函数将返回失败
bExitGrpAlways	如果通道已经加入到其它的会议室, 本参数指示是否由驱动程序自动将通道退出该会议室。 TRUE: 是 FALSE: 否, 函数将返回失败

返回值:

-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

将通道加入到会议室。

注意事项: 无

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmExitConfGroup](#), [SsmSetListenVlmInConf](#)

2.15.2.2 SsmExitConfGroup

将通道退出会议室。

函数原型:

```
int SsmExitConfGroup(int ch, BOOL bFreeGrpAlways)
```

参数说明:

ch	通道编号 如果会议室中只剩下本通道，本参数指示是否由驱动程序自动注销会议室。
bFreeGrpAlways	TRUE: 注销 FALSE: 不注销

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

将通道退出会议室。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmJoinConfGroup](#)

2.15.2.3 SsmSetListenVlmInConf

设置下总线信号进入 M2（下总线信号混音器）前的音量调节器。

函数原型:

```
int SsmSetListenVlmInConf(int ch, int nVlm)
```

参数说明:

ch	通道号
nVlm	下总线音量，取值范围为: -7~6

返回值:

0	函数调用成功
-1	函数调用失败

功能描述:

设置下总线信号进入 M2（下总线信号混音器）前的音量调节器，即设置 SHT/SHD/SHN 系列板卡原理框图中的 A4-1…A4-6。本函数调整当前通道听会议室其他成员说话时音量的大小，其他成员听会议成员说话的音量不变。

注意事项:

- 本函数只有在通道加入会议室前调用才有效。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmJoinConfGroup](#)

2.15.2.4 SsmSetContactInConf

设置电话会议中一通道不听另一通道的功能

函数原型:

```
int SsmSetContactInConf(int nGrpID, int chFrom, int chTo, BOOL bFlag)
```

参数说明:

nGrpID	会议编号
chFrom	发言通道号
chTo	聆听通道号
bFlag	TRUE: 表示电话会议中 chTo 通道屏蔽 chFrom 通道的发言 FALSE: 表示电话会议中 chTo 通道需要听 chFrom 通道的发言

返回值:

-1	函数不被支持
0	函数调用失败
1	函数调用成功

功能描述:

实现电话会议中一通道听不到另一通道的功能

注意事项:

- 只能实现一个通道的发言屏蔽，比如通道 A 屏蔽通道 B 后，不能继续屏蔽通道 C，如需要屏蔽 C，则需要关闭通道 A 对通道 B 的屏蔽。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: 无

2.15.3 获取会议室信息的函数

2.15.3.1 SsmGetConfCfgInfo

获取分布式电话会议的配置参数。

函数原型:

```
int SsmGetConfCfgInfo(PWORD pwMaxMember, PWORD pwMaxSpeaker, PWORD pwMaxSpeaking, PWORD pwMaxSilenceTime)
```

参数说明:

pwMaxMember	返回配置项 ConfDefaultMaxGroupMember 的设定值
pwMaxSpeaker	返回配置项 ConfDefaultMaxGroupSpeaker 的设定值
pwMaxSpeaking	返回配置项 ConfDefaultMaxGroupSpeaking 的设定值
pwMaxSilenceTime	返回配置项 ConfDefaultMaxSilenceTime 的设定值

返回值:

-1	调用失败
≥0	返回配置项 ConfMaxGroup 的设定值

功能描述:

获取分布式电话会议的配置参数。

注意事项：无

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmGetConfGrpInfo](#)

2.15.3.2 SsmGetTotalConfGroup

获取会议室的实际数量。

函数原型：

```
int SsmGetTotalConfGroup()
```

参数说明：无

返回值：

-1	调用失败
≥ 0	会议室的实际数量

功能描述：

获取会议室的实际数量。

注意事项：无

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmGetConfGrpId](#)

2.15.3.3 SsmGetConfGrpCfgInfo

获取会议室的工作参数的设定值。

函数原型：

```
int SsmGetConfGrpCfgInfo(int nGrpId, PWORD pwMaxMember, PWORD pwMaxSpeaker, PWORD pwMaxSpeaking, PWORD pwMaxSilenceTime)
```

参数说明：

nGrpId	会议室编号
pwMaxMember	返回会议通道的总数
pwMaxSpeaker	返回具有发言权的会议通道的总数
pwMaxSpeaking	返回允许同时发言的会议通道的总数
pwMaxSilenceTime	返回会议通道的最小静默保持时间，单位为秒

返回值：

-1	调用失败
0	调用成功

功能描述：

获取会议室工作参数的设定值，即应用程序调用[SsmCreateConfGroup](#)函数创建会议室时设定的参数。

注意事项：无

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetConfCfgInfo](#)

2.15.3.4 SsmGetConfGrpInfo

获取会议室的统计信息。

函数原型:

```
int SsmGetConfGrpInfo(int nGrpId, PWORD pwTotalMember, PWORD pwTotalSpeaker, PWORD pwTotalSpeaking)
```

参数说明:

nGrpId	会议室编号
pwTotalMember	返回会议通道的实际数量
pwTotalSpeaker	返回会议室中具有发言权通道的实际数量
pwTotalSpeaking	正在使用会议混音器的会议通道的实际数量

返回值:

-1	调用失败
0	调用成功

功能描述:

获取会议室的统计信息。

注意事项: 无

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetConfCfgInfo](#)

2.15.3.5 SsmGetConfGrpId

获取全部会议室的编号表。

函数原型:

```
int SsmGetConfGrpId(int* pnGrpId)
```

参数说明:

pnGrpId	返回会议室编号的数组指针
---------	--------------

返回值:

-1	调用失败
0	调用成功

功能描述:

获取全部会议室的编号表。

注意事项: 无

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmValidateGrpId](#)

2.15.3.6 SsmValidateGrpId

查询会议室是否已经被创建。

函数原型:

```
int SsmValidateGrpId(int nGrpId)
```

参数说明:

nGrpId	会议室编号
--------	-------

返回值:

1	会议室已经创建
0	会议室尚未创建

功能描述:

查询会议室是否已经被创建。

注意事项: 无

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetConfGrpId](#)

2.15.4 获取会议室成员信息的函数

2.15.4.1 SsmGetConfGrpMmbrId

获取已经加入会议室的会议通道编号，该处“会议通道编号”仅为当前会议室内部通道编号，不是整个系统的通道逻辑编号。

函数原型:

```
int SsmGetConfGrpMmbrId(int nGrpId, int* pnMmbrId)
```

参数说明:

nGrpId	会议室编号
pnMmbrId	返回会议室内成员 ID 的数组指针

返回值:

-1	调用失败
0	调用成功

功能描述:

获取已经加入会议室的会议通道编号，该处“会议通道编号”仅为当前会议室内部通道编号，不是整个系统的通道逻辑编号。

注意事项: 无

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetConfGrpMmbrInfo](#)

2.15.4.2 SsmGetConfGrpMmbrInfo

获取会议室中的会议通道的详细信息。

函数原型:

```
int SsmGetConfGrpMmbrInfo(int nGrpId, int nMmbrId, int* pnAppCh, PWORD pwJoinMode, PWORD pwIsSpeaking, PDWORD pdwSilenceTime)
```

参数说明:

nGrpId	会议室编号
nMmbrId	会议通道的编号
pnAppCh	返回会议通道编号对应的通道逻辑编号
pwJoinMode	返回会议通道发言模式
pwIsSpeaking	返回会议通道是否正在发言
pdwSilenceTime	返回会议通道保持静默的时间，单位为秒

返回值:

-1	调用失败
0	调用成功

功能描述:

根据会议室编号和会议通道编号，获取会议室中的会议通道的详细信息。

注意事项: 无**相关信息:**

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetConfChInfo](#)

2.15.4.3 SsmGetConfChInfo

查询通道是否加入了某个会议室。

函数原型:

```
int SsmGetConfChInfo(int ch, int* pnGrpId, int* pnMmbrId, PWORD pwJoinMode, PWORD pwIsSpeaking, PDWORD pdwSilenceTime)
```

参数说明:

ch	通道编号
pnGrpId	返回会议室的编号
pnMmbrId	返回 ch 在会议室中的成员编号
pwJoinMode	返回 ch 加入会议室的发言模式
pwIsSpeaking	返回 ch 是否正在发言的标志，pwIsSpeaking=1 为发言，pwIsSpeaking=0 为不发言
pdwSilenceTime	返回 ch 保持静默的时间，单位为秒

返回值:

-1	调用失败
0	调用成功

功能描述:

查询通道是否加入了某个会议室。

注意事项: 无**相关信息:**

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetConfGrpMmbrInfo](#)

2.16 有关 FSK 收发器的函数（CTI 系列）

2.16.1 FSK 数据发送器

2.16.1.1 SsmSetFskPara

设置 FSK 数据发送器的工作参数。

函数原型:

```
int SsmSetFskPara(int nFreq0, int nFreq1, int nBaudrate, int nMdIAmp)
```

参数说明:

nFreq0	比特 0 的载波频率, 单位: Hz, 取值范围: 300~3400。本参数也可以通过配置项 FreqBit0 进行设置, 缺省值未 2200Hz
nFreq1	比特 1 的载波频率, 单位: Hz, 取值范围: 300~3400。本参数也可以通过配置项 FreqBit1 进行设置, 缺省值为 1200Hz
nBaudrate	波特率, 单位: bps, 取值范围: 300~3400。本参数也可以通过配置项 Baudrate 进行设置, 缺省值为 1200bps
nMdIAmp	调制幅度, 取值范围: 0~255。本参数也可以通过配置项 MdIAmp 进行设置, 缺省值为 128

返回值:

-1	调用失败
0	调用成功

功能描述:

设置 FSK 数据发送器的工作参数。

注意事项:

- 如果接收端也使用 Synway 板卡, 通常无需设置, 使用相关配置项的缺省值即可。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetFskPara](#)

2.16.1.2 SsmGetFskPara

获取 FSK 数据发送器的工作参数。

函数原型:

```
int SsmGetFskPara(int *nFreqBit0, int *nFreqBit1, int *nBaudrate, int *nMdlAmp)
```

参数说明:

nFreqBit0	返回比特 0 的载波频率, 单位为 Hz
nFreqBit1	返回比特 1 的载波频率, 单位为 Hz
nBaudrate	返回二进制码流的波特率, 单位为 bps
nMdlAmp	返回调制信号的幅度

返回值:

-1	调用失败
0	调用成功

功能描述:

获取 FSK 数据发送器的工作参数。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetFskPara](#)

2.16.1.3 SsmTransFskData

将数据转换为 BFSK 格式的二进制数据流。

函数原型:

```
int SsmTransFskData(unsigned char* pSrc, int nSrcLen,int nSyncLen,int nSyncOffLen,unsigned char* pStream)
```

参数说明:

pSrc	存放数据的缓冲区指针
nSrcLen	pSrc 中数据的长度, 单位为字节
nSyncLen	同步码的比特个数, 通常为 8 的整数倍, 可以根据实际需要确定, 建议设置为 96
nSyncOffLen	标志码的比特个数, 通常为 8 的整数倍, 可以根据实际需要确定, 建议设置为 32
pStream	存放转换后的 BFSK 数据流的缓冲区指针, 由应用程序分配空间, 大小可以按照下面公式计算: $(nSyncLen + nSyncOffLen + nSrcLen \times 10) / 8 + 1$

返回值:

-1	调用失败
≥ 0	pStream 中实际的比特个数

功能描述:

将数据转换为 BFSK 格式的二进制数据流。有关 BFSK 数据流格式的详细内容请参见第 1 章中 “[FSK 数据收发器](#)” 部分。

注意事项:

- 部分协议将同步码和标志码一起称为“同步引导串”来标志同步是否建立。其中，标志码被称为“同步结束字符”。在本函数中，用参数 nSyncLen 代表同步码，参数 nSyncOffLen 代表标志码。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmStartSendFSK](#)

2.16.1.4 SsmStartSendFSK

启动 FSK 数据发送器。

函数原型:

```
int SsmStartSendFSK(int ch, LPSTR pStream, DWORD dwMaxBit)
```

参数说明:

ch	通道编号
pStream	存放 BFSK 数据流的数据缓冲区指针。更多信息请参见第 1 章中“ FSK 数据收发器 ”部分内容
dwMaxBit	pStream 中有效的比特总数，可以是任意长度

返回值:

-1	调用失败
0	调用成功

功能描述:

启动 FSK 数据发送器。

FSK 数据发送器的工作参数可以通过函数 [SsmSetFskPara](#) 或配置项 [FreqBit0](#)、[FreqBit1](#)、[Baudrate](#)、[MdlAmp](#) 进行设置。

FSK 数据发送器启动后，应用程序可以随时调用函数 [SsmStopSendFsk](#) 终止。当驱动程序完成 pStream 中全部比特的发送后，会向应用程序抛出 [E PROC_SendFSK](#) 事件。应用程序也可以调用函数 [SsmCheckSendFsk](#) 来查询 FSK 数据发送器的进展情况。

注意事项:

- 在 FSK 数据发送器完成全部数据的发送前，不能启动语音播放操作，不能使用信号音发生器和 DTMF 发生器；
- 在启动 FSK 数据发送器前，为保证数据的完整性，如果开启了回波消除功能，请将其关闭；
- 原始数据必须经函数 [SsmTransFskData](#) 转换为 BFSK 数据流后，才能调用本函数；
- 本函数不支持 ATP 系列、DTP 系列和 SHN 系列板卡。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmCheckSendFsk](#), [SsmStopSendFsk](#), [SsmTransFskData](#)

代码范例:

```
char Data[]="12345";           //发送字符串"12345"  
unsigned char Stream[256];      //存放转换后的 BFSK 比特流  
  
int nMaxBit = SsmTransFskData( (unsigned char*)Data, 5,96, 32, Stream);
```

SsmStartSendFSK(ch, (PCHAR)Stream, nMaxBit);

2.16.1.5 SsmCheckSendFsk

查询 FSK 数据发送器的进展情况。

函数原型:

int SsmCheckSendFsk(int ch)

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
0	完成
1	未完成

功能描述:

查询 FSK 数据发送器的进展情况。

注意事项:

- 如果应用程序使用基于事件的编程方式，建议使用 [E_PROC_SendFSK](#) 事件。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmStartSendFSK](#)

2.16.1.6 SsmStopSendFsk

停止 FSK 数据发送器。

函数原型:

int SsmStopSendFsk(int ch)

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
0	调用成功

功能描述:

停止 FSK 数据发送器。

注意事项:

- 驱动程序执行完本函数后，会向应用程序抛出 [E_PROC_SendFSK](#) 事件。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h

Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmStartSendFSK](#)

2.16.2 FSK 数据接收器

2.16.2.1 SsmStartRcvFSK

启动 FSK 数据接收器。

函数原型:

```
int WINAPI SsmStartRcvFSK(int ch, WORD wTimeOut, WORD wMaxLen, UCHAR ucEndCode, WORD wEndCodeCount)
```

参数说明:

ch	通道编号
wTimeOut	定时器的时长，单位为毫秒。
wMaxLen	FSK 数据的最大长度，单位为字节，取值范围：1~1024
ucEndCode	用于指示 FSK 数据包结束的特征字符
wEndCodeCount	ucEndCode 连续出现的最少个数，取值范围：0~10

返回值:

-1	调用失败，失败原因可以通过调用函数 SsmGetLastErrMsg 获得
0	调用成功

功能描述:

启动 FSK 数据接收器，开始执行 FSK 数据接收任务。

FSK 数据接收器启动后，驱动程序会启动一个定时器，并按照下列逻辑决定是否终止本次接收任务：

- ◆ 如果在 FSK 数据中发现由参数 ucEndCode 设定的特征结束字符，并且该字符连续出现的次数大于或等于 wEndCodeCount，本次接收任务终止，驱动程序会向应用程序抛出 [E PROC_RcvFSK](#) 事件（dwParam 被设置为 2）；
- ◆ 如果 FSK 数据的长度（字节总数）大于或等于 wMaxLen，本次 FSK 数据接收任务终止，驱动程序向应用程序抛出 [E PROC_RcvFSK](#) 事件（dwParam 为 3）。
- ◆ 如果线路上的 BFSK 的信号消失，并且消失时间超过 100 毫秒，本次接收任务结束，驱动程序向应用程序抛出 [E PROC_RcvFSK](#) 事件（dwParam 为 1）。
- ◆ 如果定时器溢出，本次接收任务结束，驱动向应用程序抛出 [E PROC_RcvFSK](#) 事件（dwParam 为 1）。

注意事项:

- Synway 板卡的 FSK 数据接收器的波特率 1200bps，比特 0 的载波频率为 2200Hz，比特 1 的载波频率为 1200Hz，均不可通过函数或配置项进行修改；
- 本函数不支持 ATP 系列、DTP 系列和 SHN 系列板卡。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetRcvFSK](#), [SsmStopRcvFSK](#), [SsmStartRcvFSK_II](#)

代码范例:

```
.....
SsmStartRcvFSK (0, 10000, 260, "#", 1);           //启动 FSK 数据接收器

MESSAGE_INFO Event;
If ( SsmWaitForEvent (64, &Event) == 0 )           //等待事件发生
{
    switch(Event.EventCode)
    {
        case E_PROC_RcvFSK:
            if(Event.dwParam == 2)           //FSK 数据接收器终止
                //收齐带结束符="#"的 FSK 数据
            {
                unsigned char DataBuf[260];
                SsmGetRcvFSK(Event.nReference, DataBuf); //取出数据
                .....                           //处理代码
            }
            .....
        break;
        .....
    }
}
.....
```

2.16.2.2 SsmStartRcvFSK_II

启动 FSK 接收任务。

函数原型:

```
int WINAPI SsmStartRcvFSK_II(int ch, WORD wTimeOut, WORD wMaxLen, PUCHAR pucMarkCodeBuf, UCHAR ucMarkCodeCount)
```

参数说明:

ch	通道编号
wTimeOut	定时器的时长，单位为毫秒
wMaxLen	FSK 数据的最大长度，单位为字节，取值范围：1~260
pucMarkCodeBuf	存放标志码的缓冲区，最多可以设置 10 个不同的标志码
ucMarkCodeCount	pucMarkCodeBuf 中标志码的个数，取值范围：0~10。如果本参数设置为 0，参数 pucMarkCodeBuf 将被忽略

返回值:

-1	调用失败，失败原因可以通过调用函数 SsmGetLastErrMsg 获得
0	启动 FSK 接收器成功

功能描述:

启动 FSK 数据接收器，开始执行 FSK 数据接收任务。

本函数支持 2 种接收方式：长度方式和帧方式。帧方式是指需要传输的 FSK 数据具有固定的帧结构，驱动程序需要根据帧的标志（即标志码）对帧进行定位和分离；长度方式是指驱动程序只检查接收数据的长度，当收到的 FSK 数据长度大于 wMaxLen 后，接收任务就结束。如果 ucMarkCodeCount 大于 0，驱动程序优先使用帧方式进行接收，否则，按照长度方式进行接收，此时，参数 wMaxLen 有效。

1 帧数据由 4 个字段组成：标志码字段、长度字段、数据体字段和校验字节字段，如下图所示：



标志码字段： 表示帧的开始，占用 1 个字节，用于帧的定位。标志码的具体取值可以由应用程序自行决定；

长度字段： 表示数据体部分的长度，单位为字节，占用 1 个字节；

数据体： 需要传输的数据，最长不能超过 255 个字节；

校验字节：占用 1 个字节。注意：校验字节由应用程序执行产生，驱动程序并不检查数据的完整性。

FSK 数据接收器的执行逻辑为：

- ◆ 启动一个定时器，等待 FSK 数据到达；
 - ◆ 标志码方式：FSK 数据到达后，驱动程序在接收数据的前 10 个字节中寻找由参数 pucMarkCodeBuf 设定的标志码。如果找到，表明驱动程序顺利完成帧的定位，当驱动程序在完成 1 帧数据的接收后，本次接收任务终止，驱动程序向应用程序抛出 [E PROC_RcvFSK](#) 事件（dwParam 被设置为 4）；如果无法找到标志码，转而按照长度方式进行接收；
 - ◆ 长度方式：当 FSK 数据的长度超过参数 wMaxLen 的设定值后，本次接收任务终止，驱动程序向应用程序抛出 [E PROC_RcvFSK](#) 事件（dwParam 被设置为 3）；
 - ◆ 如果线路上的 BFSK 的信号消失，并且消失时间超过 100 毫秒，本次接收任务终止，驱动程序向应用程序抛出 [E PROC_RcvFSK](#) 事件（如果已经收到的 fsk 数据长度小于 wMaxLen，dwParam 为 1，否则为 3）；
 - ◆ 如果定时器溢出，本次接收任务终止，驱动程序向应用程序抛出 [E PROC_RcvFSK](#) 事件（dwParam 为 1）。

注意事项:

- Synway 板卡的 FSK 数据接收器的波特率 1200bps，比特 0 的载波频率为 2200Hz，比特 1 的载波频率为 1200Hz，均不可通过函数或配置项进行修改。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetRcvFSK](#), [SsmStopRcvFSK](#), [SsmStartRcvFSK](#)

代码范例：

```

    .....
    SsmStartRcvFSK_II(0, 10000, 1, "#", 1); //启动 FSK 数据接收器

    MESSAGE_INFO Event;
    If (SsmWaitForEvent(64, &Event) == 0) //等待事件发生
    {
        switch(Event.EventCode)
        {
            case E_PROC_RcvFSK:
                if(Event.dwParam == 4) //FSK 数据接收器终止
                    //收到 1 帧 FSK 数据
                {
                    unsigned char DataBuf[260];
                    SsmGetRcvFSK(Event.nReference, DataBuf); //取出数据
                    ..... //处理代码
                }
                else
                {
                    ..... //接收失败的处理代码
                }
            break;
            .....
        }
    }
}

```

2.16.2.3 SsmStartRcvFSK III

启动 FSK 接收任务。

函数原型.

```
int WINAPI SsmStartRcvESK (int ch,WORD wTimeOut,WORD wMaxLen,PUSHORT pucMarkCodeBuf, UCHAR ucMarkCodeBufSize);
```

ucMarkCodeCount)

参数说明:

ch	通道编号
wTimeOut	定时器的时长，单位为毫秒
wMaxLen	FSK 数据的最大长度，单位为字节，取值范围：1~1024
pucMarkCodeBuf	存放标志码的缓冲区，最多可以设置 10 个不同的标志码
ucMarkCodeCount	pucMarkCodeBuf 中标志码的个数，取值范围：0~10。如果本参数设置为 0，参数 pucMarkCodeBuf 将被忽略；如果本参数设置为 5，表示 pucMarkCodeBuf 的前 5 个标志码起作用。

返回值:

-1	调用失败，失败原因可以通过调用函数 SsmGetLastErrMsg 获得
0	启动 FSK 接收器成功

功能描述:

启动 FSK 数据接收器，开始执行 FSK 数据接收任务。

本函数支持 2 种接收方式：长度方式和帧方式。帧方式是指需要传输的 FSK 数据具有固定的帧结构，驱动程序需要根据帧的标志（即标志码）对帧进行定位和分离；长度方式是指驱动程序只检查接收数据的长度，当收到的 FSK 数据长度大于 wMaxLen 后，接收任务就结束。如果 ucMarkCodeCount 大于 0，驱动程序优先使用帧方式进行接收，否则，按照长度方式进行接收，此时，参数 wMaxLen 有效。

1 帧数据由 4 个字段组成：标志码字段、长度字段、数据体字段和校验字节字段，如下图所示：

标志码	数据长度	数据体	校验字节
-----	------	-----	------

标志码字段： 表示帧的开始，占用 1 个字节，用于帧的定位。标志码的具体取值可以由应用程序自行决定；

长度字段： 表示数据体部分的长度，单位为字节，占用 2 个字节；

数据体： 需要传输的数据，最长不能超过 1020 个字节；

校验字节： 占用 1 个字节。注意：校验字节由应用程序执行产生，驱动程序并不检查数据的完整性。

FSK 数据接收器的执行逻辑为：

- ◆ 启动一个定时器，等待 FSK 数据到达；
- ◆ 标志码方式：FSK 数据到达后，驱动程序在接收数据的前 10 个字节中寻找由参数 pucMarkCodeBuf 设定的标志码。如果找到，表明驱动程序顺利完成帧的定位，当驱动程序在完成 1 帧数据的接收后，本次接收任务终止，驱动程序向应用程序抛出 [E PROC RcvFSK](#) 事件（dwParam 被设置为 4）；如果无法找到标志码，转而按照长度方式进行接收；
- ◆ 长度方式：当 FSK 数据的长度超过参数 wMaxLen 的设定值后，本次接收任务终止，驱动程序向应用程序抛出 [E PROC RcvFSK](#) 事件（dwParam 被设置为 3）；
- ◆ 如果线路上的 BFSK 的信号消失，并且消失时间超过 100 毫秒，本次接收任务终止，驱动程序向应用程序抛出 [E PROC RcvFSK](#) 事件（如果已经收到的 FSK 数据长度小于 wMaxLen，dwParam 为 1，否则为 3）；
- ◆ 如果定时器溢出，本次接收任务终止，驱动程序向应用程序抛出 [E PROC RcvFSK](#) 事件（dwParam 为 1）。

注意事项：

- Synway 板卡的 FSK 数据接收器的波特率 1200bps，比特 0 的载波频率为 2200Hz，比特 1 的载波频率为 1200Hz，均不可通过函数或配置项进行修改。

- 帧格式中的数据长度占用 2 个字节，前一字节表示高位数据长度，后一字节表示低位数据，例如：

sz[5] = {0xFF 0x0 0x3 0x1 0x2 0x3} 为正确的 FSK 数据，其中：

sz[0] = 0xFF 为开始标志；

sz[1] = 0x0, sz[2] = 0x3 表示长度 3 字节；

后面 0x1 0x2 0x3 为数据体。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.0.1.0 后更高版本
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetRcvFSK](#), [SsmStopRcvFSK](#), [SsmStartRcvFSK](#), [SsmStartRcvFSK_II](#)

代码范例：

```

    .....
    SsmStartRcvFSK_III(0, 10000, 1, "#", 1); //启动 FSK 数据接收器

    MESSAGE_INFO Event;
    If (SsmWaitForEvent(64, &Event) == 0) //等待事件发生
    {
        switch(Event.EventCode)
        {
            case E_PROC_RcvFSK:
                if(Event.dwParam == 4) //FSK 数据接收器终止
                    //收到 1 帧 FSK 数据
                {
                    unsigned char DataBuf[260];
                    SsmGetRcvFSK(Event.nReference, DataBuf); //取出数据
                    ..... //处理代码
                }
                else
                {
                    ..... //接收失败的处理代码
                }
            break;
            .....
        }
    }
}

```

2.16.2.4 SsmCheckRcvFSK

查询 FSK 数据接收任务的进展情况。

函数原型：

```
int SsmCheckRcvFSK(int ch)
```

参数说明:

ch	通道编号
----	------

返回值：

-1	调用失败
0	FSK 数据接收器尚未完成本次接收任务
1	因超时而终止
2	因收到结束符而终止
3	因收到指定长度数据而终止
4	因收到指定数据格式数据而终止
5	因应用程序调用 SsmStopRcvFSK 函数而终止

功能描述:

查询 FSK 数据接收任务的进展情况，该任务可以是由函数 `SsmStartRcvFSK` 或 `SsmStartRcvFSK_II` 启动。

注意事项:

- 如果应用程序使用基于事件的编程方式，建议使用 E PROC RcvFSK 事件。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetRcvFSK](#)

2.16.2.5 SsmStopRcvFSK

取消 FSK 数据接收任务。

函数原型:

```
int SsmStopRcvFSK(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
0	调用成功

功能描述:

取消 FSK 数据接收任务。本函数可以取消由函数 [SsmStartRcvFSK](#) 或 [SsmStartRcvFSK II](#) 启动的 FSK 数据接收任务。

注意事项:

- 驱动程序执行完本函数后，会向应用程序抛出 [E PROC RcvFSK](#) 事件。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetRcvFSK](#), [SsmCheckRcvFSK](#)

2.16.2.6 SsmClearRcvFSKBuf

清空驱动程序内部的 FSK 数据接收缓冲区。

函数原型:

```
int SsmClearRcvFSKBuf(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	出现错误，失败原因可以通过调用函数获得
0	成功

功能描述:

清空驱动程序内部的 FSK 数据接收缓冲区。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetRcvFSK](#)

2.16.2.7 SsmGetRcvFSK

取出保存在驱动程序内部缓冲区中的 FSK 数据。

函数原型:

```
int SsmGetRcvFSK(int ch, PUCHAR pucBuf)
```

参数说明:

ch	通道编号
pucBuf	缓冲区指针, 由应用程序分配存储空间, 不能少于 260 字节

返回值:

-1	调用失败
≥0	收到的 FSK 数据的字节长度 (不包括结束符)

功能描述:

取出保存在驱动程序内部缓冲区中的 FSK 数据。

注意事项:

- 每当应用程序调用 [SsmStartRcvFSK](#) 或 [SsmStartRcvFSK_II](#) 启动接收任务时, 驱动程序都会将保存 FSK 数据的内部缓冲区自动清空。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmCheckRcvFSK](#), [SsmClearRcvFSKBuf](#)

2.17 有关电压检测器的函数

2.17.1 忽略电压检测结果

2.17.1.1 SsmSetIgnoreLineVoltage

设置是否忽略线路上的电压检测结果。

函数原型:

```
int SsmSetIgnoreLineVoltage(int ch, BOOL bIgnore)
```

参数说明:

ch	通道编号
bIgnore	=FALSE: 不忽略

	=TRUE: 忽略
--	-----------

返回值:

-1	操作失败
0	操作成功

功能描述:

设置是否忽略线路上的电压检测结果。如果忽略线路上的电压检测结果，录音通道会一直保持“摘机”状态。
详细内容请参见第1章中“[模拟电话线的电压变化](#)”部分。

注意事项:

- 本函数只适用于ATP系列板卡的录音通道（包括模拟线路录音模块和麦克风模块）；
- 本函数设置的参数也可以通过配置项[IgnoreLineVoltage](#)进行设置，缺省值为不忽略。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetIgnoreLineVoltage](#)

2.17.1.2 SsmGetIgnoreLineVoltage

查询通道是否忽略线路上的电压检测结果。

函数原型:

```
int SsmGetIgnoreLineVoltage(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	操作失败
0	不忽略
1	忽略

功能描述:

查询通道是否忽略线路上的电压检测结果。

注意事项:

- 本函数只适用于ATP系列板卡的录音通道（包括模拟线路录音模块和麦克风模块）。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetIgnoreLineVoltage](#)

2.17.2 设置检测摘机/挂机动作的门限电压

2.17.2.1 SsmSetDtrmLineVoltage

设置模拟中继线录音通道摘机/挂机的判定电压。

函数原型:

```
int SsmSetDtrmLineVoltage(int ch, WORD wDtrmVoltage)
```

参数说明:

ch	通道编号
wDtrmVoltage	判定电压, 单位为伏特, 取值范围: 5~48

返回值:

-1	调用失败
0	调用成功

功能描述:

设置模拟中继线录音通道摘机/挂机的判定电压。详细内容请参见第1章中“[模拟电话线的电压变化](#)”部分。

注意事项:

- 本函数只适用于ATP系列板卡的录音通道(包括模拟线路录音模块和麦克风模块);
- 本函数设置的参数也可以通过配置项[IsHangupDtrmVoltage](#)进行设置,缺省值为26V。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetDtrmLineVoltage](#)

2.17.2.2 SsmGetDtrmLineVoltage

获取模拟中继线录音通道摘机/挂机的判定电压。

函数原型:

```
int SsmGetDtrmLineVoltage(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
≥0	返回通道上的电话摘/挂机的判别电压, 单位为伏特(V)

功能描述:

获取模拟中继线录音通道摘机/挂机的判定电压。

注意事项:

- 本函数只适用于ATP系列板卡的录音通道(包括模拟线路录音模块和麦克风模块)。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
----------	---------------------

Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetDtrmLineVoltage](#)

2.17.3 获取电压检测器的输出结果

2.17.3.1 SsmGetLineVoltage

获取线路上的电压值。

函数原型:

```
int SsmGetLineVoltage(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
≥0	返回线路上的电压值，单位为伏特（V）

功能描述:

获取线路上的电压值。使用事件 [E_CHG_LineVoltage](#) 也可以用于获取线路的电压值。详细内容请参见第 1 章中“[模拟电话线的电压变化](#)”部分。

注意事项:

- 本函数只适用于 ATP 系列板卡的录音通道和 SHT 系列板卡的模拟中继线通道；
- 模拟中继线上的电压值可能会在 1V-2V 的范围内波动，这是正常现象。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.17.4 有关极性反转检测器的函数

2.17.4.1 SsmQueryOpPolarRvrs

查询指定通道是否支持极性反转侦测操作。

函数原型:

```
int WINAPI SsmQueryOpPolarRvrs(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
0	指定通道不支持极性反转侦测操作
1	指定通道支持极性反转侦测操作

功能描述:

查询指定通道是否支持极性反转侦测操作。详细内容请参见第1章中“[模拟电话线的电压变化](#)”部分。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.17.4.2 SsmGetPolarRvrsCount

获取通道自摘机以后累计检测到的极性反转次数。

函数原型:

```
int SsmGetPolarRvrsCount(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
≥0	极性反转的次数

功能描述:

获取通道自摘机以后累计检测到的极性反转次数。详细内容请参见第1章中“[模拟电话线的电压变化](#)”部分。

注意事项:

- 本函数只适用于具有极性反转侦测的通道；
- 使用[E_CHG_PolarRvrsCount](#)事件也可以实现同样的功能。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关事件: [E_CHG_PolarRvrsCount](#)

2.17.5 有关铃流信号检测器的函数

2.17.5.1 SsmGetRingCount

获取检测到的铃流信号次数。

函数原型:

```
int SsmGetRingCount(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
≥0	振铃计数

功能描述:

获取检测到的铃流信号次数。详细信息请参见第 1 章中的“[模拟电话线上的铃流信号](#)”部分内容。

注意事项:

- 事件 [E_CHG_RingCount](#) 也可实现相同功能。如果希望应用程序具有良好的可移植性，推荐使用事件 [E_CHG_RingCount](#)；
- 本函数只支持 SHT 系列的模拟中继线通道和 ATP 系列的模拟电话线录音通道。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmClearRingCount](#), [SsmGetChState](#)

2.17.5.2 SsmGetCallBackRingCount

获取检测到的回呼铃流信号次数。

函数原型:

```
int SsmGetCallBackRingCount(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
≥0	振铃计数

功能描述:

获取检测到的回呼铃流信号次数。

注意事项:

- 事件 [E_CHG_CallBackRingCount](#) 也可实现相同功能。如果希望应用程序具有良好的可移植性，推荐使用事件 [E_CHG_CallBackRingCount](#)；
- 本函数只支持 SHT 系列的模拟中继线通道和 ATP 系列的模拟电话线录音通道。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.0.4 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmClearRingCount](#), [SsmGetChState](#)

2.17.5.3 SsmClearRingCount

清除保存在驱动程序内部的铃流信号的计数值。

函数原型:

int SsmClearRingCount (int ch)

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
0	调用成功

功能描述:

清除保存在驱动程序内部的铃流信号的计数值。详细信息请参见第 1 章中的“[模拟电话线上的铃流信号](#)”部分内容。

注意事项:

- 本函数只支持 SHT 系列的模拟中继线通道和 ATP 系列的模拟电话线录音通道。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetRingCount](#)

2.17.5.4 SsmGetRingFlag

获取振铃状态标志。

函数原型:

int SsmGetRingFlag (int ch)

参数说明:

ch	通道编号
----	------

返回值:

0	非振铃状态
1	振铃状态

功能描述:

获取振铃状态标志。详细信息请参见第 1 章中的“[模拟电话线上的铃流信号](#)”部分内容。

注意事项:

- 本函数只支持 SHT 系列的模拟中继线通道和 ATP 系列的模拟电话线录音通道。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.18 有关数字中继线信息的函数

2.18.1 SsmGetMaxPcm

获取应用系统中的数字中继线总数。

函数原型:

```
int SsmGetMaxPcm()
```

参数说明:

返回值:

-1	调用失败
≥0	系统中的数字中继线的总数, 由[PcmInfo]配置节中的配置项 TotalPcm 决定

功能描述:

获取应用系统中的数字中继线总数。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetPcmInfo](#)

2.18.2 SsmGetPcmInfo

获取数字中继线接口单元 (LIU) 的工作参数。

函数原型:

```
int SsmGetPcmInfo(int nPcmNo, int* pnSSxMode, int* pnBoardId, int* pnBoardPcmNo, int* pnUsePcmTS16,
PDWORD pdwRcvrMode, PDWORD pdwEnableAutoCall, PDWORD pdwAutoCallDirection)
```

参数说明:

nPcmNo	数字中继线的逻辑编号, 取值范围: 0~N-1, N 为[PcmInfo]配置节中配置项 TotalPcm 的设定值, 可通过函数 SsmGetMaxPcm 获得
pnBoardId	返回数字中继线所在板卡的 ID 号
pnBoardPcmNo	返回数字中继线在板卡上的内部编号
pnSSxMode	返回 PCM 上使用的信令方式: 0: ISDN 协议 (用户侧) 1: 中国 1 号信令 (简称 SS1) 2: ISDN 协议 (网络侧) 7: No. 7 号信令 (简称 SS7)
pnUsePcmTS16	返回 PCM 的 16 时隙是否作为话路使用。 1: 是 0: 不是
PdwRcvrMode	返回数字中继线接口单元 (LIU) 的工作模式, 包含 32 比特, 各比特定义如下: Bit1, Bit0: 时钟设置, 取值范围: 00: 线路同步方式、主时钟 01: 自振荡方式、主时钟 10: 从时钟 11: 保留 Bit2: 线路的接入方式

	Bit3: LIU 是否对 16 时隙进行消抖处理 0: 120 欧姆双绞线 1: 75 欧姆同轴电缆 Bit4: 设置语音信道的隔位反转功能 (ADI) 0: 关闭 (缺省值) 1: 开启 Bit6,Bit5: 设置 LIU 的接收均衡器增益 00: 由 LIU 自动控制(缺省值) 01: 0 DB 10: 12 DB 11: 6 DB Bit7: 保留 Bit15~Bit8: 设置 LIU 的消除抖动的控制字 Bit31~Bit16: 保留
pdwEnableAutoCall	返回 32 个时隙的自动呼叫控制字 (32 比特), 最高位比特 (bit31) 对应时隙 0, 最低位比特 (bit0) 对应于时隙 31。比特为 1 表示允许自动呼叫, 为 0 表示禁止自动呼叫
pdwAutoCallDirection	返回 32 个时隙的自动呼叫方向, 一个比特对应一个时隙, 最高位比特 (bit31) 对应时隙 0, 最低位比特 (bit0) 对应时隙 31。比特值为 0 表示只允许呼入, 为 1 表示只允许呼出

返回值:

-1	调用失败
0	调用成功

功能描述:

获取数字中继线接口单元 (LIU) 的工作参数。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetPcmLinkStatus](#)

2.18.3 SsmGetPcmLinkStatus

获取数字中继线的同步状态。

函数原型:

```
int SsmGetPcmLinkStatus(int nPcmNo, PWORD pwPcmLinkStatus)
```

参数说明:

nPcmNo	数字中继线的逻辑编号, 取值范围: 0~N-1, N 为[PcmInfo]配置节中配置项 TotalPcm 的设定值, 可通过函数 SsmGetMaxPcm 获得
pwPcmLinkStatus	返回数字中继线的同步状态, 共 16 个比特, bit0 为最低有效位。比特值为 0 表示工作正常, 为 1 时: 对于板卡型号 (SHD-30A-CT/PCI/SS1、SHD-30A-CT/PCI/ISDN、SHD-30A-CT/PCI/SS7、SHD-60A-CT/PCI/SS1、SHD-60A-CT/PCI/ISDN、SHD-60A-CT/PCI/SS7、SHD-120A-CT/PCI/SS1、SHD-120A-CT/PCI/ISDN、SHD-120A-CT/PCI/SS7、SHD-30B-CT/PCI/SS7/FAX、SHD-60B-CT/PCI/SS7/FAX、SHD-30A-CT/cPCI/SS7、SHD-60A-CT/cPCI/SS7、SHD-120A-CT/cPCI/SS7、SHD-30B-CT/cPCI/SS7/FAX、SHD-60B-CT/cPCI/SS7/FAX、SHD-30A-CT/PCI/FJ、SHD-60A-CT/PCI/FJ): bit0=1: 基本帧同步丢失

	<p>bit1=1: 基本帧同步丢失时间大于 100ms bit2=1: 复帧同步丢失 (SS1 only) bit3=1: CRC 复帧同步丢失 bit4=1: 远端告警指示 bit5=1: 信号告警指示 bit6=1: 16 时隙全 1 告警信息 (SS1 only) bit7=1: 信号丢失 bit8=1: 辅助告警 (SS1 only) bit9=1: 复帧失步告警 (SS1 only) 其余比特: 保留, 全部为 0</p> <p>对于板卡型号 (SHD-60B-CT/PCI/FJ、DTP-30C/PCI、DTP-30C/PCI+、DTP-60C/PCI、 DTP-60C/PCI+、DTP-120C/PCI、DTP-120C/PCI+、DTP-30C/PCIe、DTP-30C/PCIe+、 DTP-60C/PCIe、DTP-60C/PCIe+、DTP-120C/PCIe、DTP-120C/PCIe+、 SHD-120D-CT/PCI、SHD-120D-CT/PCI/EC、SHD-120D-CT/PCI/CAS、 SHD-240D-CT/PCI、SHD-240D-CT/PCI/EC、SHD-240D-CT/PCI/CAS、 SHD-30E-CT/PCI、SHD-30E-CT/PCI/FAX、SHD-30E-CT/PCI/EC、SHD-60E-CT/PCI、 SHD-60E-CT/PCI/FAX、SHD-60E-CT/PCI/EC、SHD-120E-CT/PCI、 SHD-120E-CT/PCI/FAX、SHD-120E-CT/PCI/EC、SHD-240E-CT/PCI、 SHD-240E-CT/PCI/FAX、SHD-240E-CT/PCI/EC、SHD-30E-CT/PCIe、 SHD-30E-CT/PCIe/FAX、SHD-30E-CT/PCIe/EC、SHD-60E-CT/PCIe、 SHD-60E-CT/PCIe/FAX、SHD-60E-CT/PCIe/EC、SHD-120E-CT/PCIe、 SHD-120E-CT/PCIe/FAX、SHD-120E-CT/PCIe/EC、SHD-240E-CT/PCIe、 SHD-240E-CT/PCIe/FAX、SHD-240E-CT/PCIe/EC、SHD-240E-CT/PCIe/VAR):</p> <p>bit0=1: 基本帧同步丢失 bit1=1: 基本帧同步丢失时间大于 100ms bit2=1: CAS 重新同步(SS1、E1) bit3=1: CRC 重新同步(E1 only) bit4=1: 远端告警指示 bit5=1: 信号状态告警指示 bit6=1: 16 时隙全 1 告警信息(E1 only) bit7=1: 信号丢失 bit9=1: 远端 MF 告警(SS1、E1) bit10=1: 断路 bit11=1: 短路 其余比特: 保留, 全部为 0</p> <p>对于板卡型号 (SHD-480A-CT/cPCI、SHD-480S-CT/cPCI、SHD-240S-CT/cPCI、 SHD-240A-CT/cPCI):</p> <p>bit1=1: FAS 同步生效 bit2=1: CAS MF 同步生效 bit3=1: CRC4 MF 同步生效 bit8=1: 同步丢失 bit9=1: 载体丢失 bit10=1: 远端告警 其余比特: 保留, 全部为 0</p> <p>对于板卡型号 (SHD-30C-CT/PCI、SHD-60C-CT/PCI、SHD-30C-CT/PCI/FAX、 SHD-60C-CT/PCI/FAX、SHD-30B-CT/PCI/FJ):</p> <p>bit1=1: 不同步时间大于 120ms 其余比特: 保留, 全部为 0</p>
--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

返回值:

-1	调用失败
0	调用成功

功能描述:

获取数字中继线的同步状态。

注意事项:

- 对于中国 No.1 信令, 有的交换机支持 CRC 复帧校验, 有的不支持。因此, 当出现 CRC 复帧同步丢失时

不一定表明线路同步出现故障；

- 对于 SS7 信令和 ISDN 信令，16 时隙通常用于传输共路信令，没有复帧和 CRC 复帧的概念，因此，只要基本帧同步信息正常（bit0 和 bit1 均为 0），就可以正常使用。另外，由于 SS7 信令和 ISDN 信令均为数字信令，具有一定的容错能力，允许基本帧同步出现小范围的抖动。所以，如果将本函数返回的线路同步状态用于告警等目的，通常以 bit0 从 0 变化为 1 作为告警依据。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmGetPcmInfo](#)

2.18.4 SsmPcmTsToCh

根据 PCM 的逻辑编号和时隙编号查询对应的通道编号。

函数原型：

```
int SsmPcmTsToCh(int nLocalPcmNo, int nTs)
```

参数说明：

nLocalPcmNo	数字中继线的逻辑编号。有关数字中继线的详细内容请参见第 1 章中“ 基本概念 ”部分
nTs	时隙编号，取值范围：1~31

返回值：

-1	操作失败
≥0	对应通道的逻辑编号

功能描述：

根据 PCM 的逻辑编号和时隙编号查询对应的通道编号。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmChToPcmTs](#)

2.18.5 SsmChToPcmTs

由通道编号查询对应的 PCM 逻辑编号和时隙编号。

函数原型：

```
int SsmChToPcmTs(int ch, int* pnLocalPcmNo, int* pnTs)
```

参数说明：

ch	通道编号
pnLocalPcmNo	返回通道 ch 所在的 PCM 逻辑编号
pnTs	通道 ch 所在的时隙编号

返回值：

-1	操作失败
≥0	返回 pnLocalPcmNo 参数中的值

功能描述：

由通道编号查询对应的 PCM 逻辑编号和时隙编号。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmPcmTsToCh](#)

2.18.6 SsmSetPcmClockMode

设置数字中继线的时钟模式。

函数原型:

```
int SsmSetPcmClockMode(int nPcmNo, int nClockMode)
```

参数说明:

nPcmNo	数字中继线的逻辑编号, 取值范围: 0~N-1 (N 为配置项 TotalPcm 的设定值, 可以通过函数 SsmGetMaxPcm 获取)
nClockMode	时钟模式, 取值范围: 0: 主时钟、线路同步方式 (板卡必须为主卡) 1: 主时钟、自振荡方式 (板卡必须为主卡) 2: 从时钟

返回值:

-1	调用失败
0	调用成功

功能描述:

设置数字中继线的时钟模式。更多信息请参见第 1 章中 “[配置应用系统的时钟](#)” 部分内容。

注意事项:

- 本函数只适用于 SHD、DTP 系列板卡;
- 配置项 [PcmClockMode](#) 可以实现相同功能。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetPcmInfo](#)

2.18.7 SsmGetInboundLinkSet

来话呼叫时, 获取来自对端交换机的 TUP/ISUP 消息所使用的链路组信息。

函数原型:

```
int SsmGetInboundLinkSet(int ch, LPWORD pwLinkSetNo, LPSTR pszOpc, LPSTR pszDpc)
```

参数说明:

ch	通道号
pwLinkSetNo	链路组编号
pszOpc	OPC
pszDpc	DPC

返回值:

1	调用成功
-1	调用失败

功能描述:

来话呼叫时，获取来自对端交换机的 TUP/ISUP 消息所使用的链路组信息，包括链路组的编号、DPC、OPC。

注意事项:

- 本函数只适用于 TUP/ISUP 通道。

相关信息:

驱动程序版本要求	SynCTI Ver. 4.7.1.2 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.18.8 SsmGetCbChStatus

获取大容量坐席模块的连线状态。

函数原型:

```
int WINAPI SsmGetCbChStatus(int ch, PWORD pwCBLinkStatus)
```

参数说明:

ch	通道号
pwCBLinkStatus	返回通道连接状态

返回值:

0	调用成功
-1	调用失败

功能描述:

获取大容量坐席模块的连线状态。参数 pwCBLinkStatus 等于 0 表示线路是连接的，pwCBLinkStatus 等于 1 表示线路是断开的。

注意事项:

- 本函数只适用于大容量连线检测。
- 使用事件 [E_CHG_CbChStatus](#) 可实现同样功能。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.0.3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: 无**相关事件:** [E_CHG_CbChStatus](#)

2.18.9 SsmSetPcmPowerDown

设置数字中继线的工作模式。

函数原型:

```
int WINAPI SsmSetPcmPowerDown(int nPcmNo, int nPowerMode)
```

参数说明:

nPcmNo	数字中继线的逻辑编号，取值范围：0~N-1（N 为配置项 TotalPcm 的设定值，可以通过函数 SsmGetMaxPcm 获取）
nPowerMode	工作模式，取值范围： 0：表示正常工作模式

1: 表示休息模式（类似于该 PCM 未接线）

返回值:

-1	调用失败
0	调用成功

功能描述:

对指定的 PCM 进行工作状态设置，可以将其设置为工作或者休息状态。

注意事项:

- 本函数只适用于 SHD 系列 D 型、E 型板卡；

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.0.1 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.19 SHT 系列板卡专用函数 (CTI 系列)

2.19.1 设置联合模块工作模式

2.19.1.1 SsmSetUnimoduleState

设置是否将联合模块上的模拟中继线通道直接连接到坐席通道。

函数原型:

```
int SsmSetUnimoduleState(int ch,int nState)
```

参数说明:

ch	联合模块中模拟中继线通道的编号
nState	=0: 模拟中继线通道直接连接到坐席通道 =1: 模拟中继线通道与坐席通道没有任何联系，相当于两个完全独立控制的通道

返回值:

-1	失败
0	成功

功能描述:

设置是否将联合模块上的模拟中继线通道直接连接到坐席通道。

注意事项:

- 配置项 [UnimoduleState](#) 可以实现相同功能。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.19.2 模拟中继通道专用函数

2.19.2.1 在模拟线路上产生闪断信号

2.19.2.1.1 SsmTxFlash

在模拟中继线路上产生一次闪断（相当于话机的拍叉簧动作）。

函数原型：

```
int SsmTxFlash(int ch, WORD time)
```

参数说明：

ch	通道编号
time	设置闪断的保持时间，单位为毫秒，取值范围：32~1000，通常设置为 500ms

返回值：

-1	调用失败。失败原因可以通过调用函数 SsmGetLastErrMsg 获得
0	调用成功

功能描述：

在模拟中继线路上产生一次闪断，即模拟话机的拍叉簧动作。

当驱动程序完成一次产生闪断的操作后，会向应用程序抛出 [E_PROC_SendFlash](#) 事件。操作执行的进展情况也可以用函数 [SsmChkTxFlash](#) 取得。

注意事项：

- 本函数支持 SHT 系列板卡的模拟中继线通道和 SS1 信令里的 LineSide 协议。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmChkTxFlash](#)

2.19.2.1.2 SsmSetTxFlashCharTime

设置用函数 [SsmTxDtmf](#) 在模拟中继线上产生的闪断的保持时间。

函数原型：

```
int SsmSetTxFlashCharTime (int ch,WORD time)
```

参数说明：

ch	通道编号
Time	闪断信号的保持时间，单位为毫秒，取值范围：32~1000，默认值为 500

返回值：

0	操作成功
-1	操作失败。失败原因可以通过调用函数 SsmGetLastErrMsg 获得

功能描述：

使用'!'字符作为参数调用函数 [SsmTxDtmf](#) 也可以在模拟中继线上产生一次闪断，本函数用于设置该字符产生的闪断信号的保持时间。

注意事项:

- 配置项 [DefaultTxFlashTime](#) 可以实现相同功能，缺省值为 500ms。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetTxFlashCharTime](#), [SsmTxDtmf](#)

2.19.2.1.3 SsmGetTxFlashCharTime

获取用函数 [SsmTxDtmf](#) 在模拟中继线上产生的闪断的保持时间。

函数原型:

```
int SsmGetTxFlashCharTime(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

>0	闪断信号的保持时间，单位为毫秒
-1	操作失败。失败原因可以通过调用函数 SsmGetLastErrMsg 获得

功能描述:

获得调用函数 [SsmTxDtmf](#) 产生的闪断信号的保持时间。

注意事项: 无

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetTxFlashCharTime](#)

2.19.2.1.4 SsmChkTxFlash

查询在模拟中继线上产生闪断信号的操作是否完成。

函数原型:

```
int SsmChkTxFlash(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	出现错误
0	产生闪断的操作已经完成
1	产生闪断的操作正在进行

功能描述:

查询在模拟中继线上产生闪断信号的操作是否完成。

注意事项:

- 只有模拟中继线通道才支持产生闪断信号的操作。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmTxFlash](#)

2.19.2.1.5 SsmTxFlashEx

在模拟中继线路上产生一次闪断（相当于话机的拍叉簧动作），同时设置闪断后通道的当前状态。

函数原型:

```
int WINAPI SsmTxFlashEx(int ch, WORD time, int nChState, BOOL bIgnoreDITn)
```

参数说明:

ch	通道编号
time	设置闪断的保持时间，单位为毫秒，取值范围：32~1000，通常设置为 500ms
nChState	设置闪断后通道的状态，取值范围：S_CALL_PICKUPED(1)、S_CALL_TALKING(3)
bIgnoreDITn	设置闪断后是否忽略拨号音，取值范围：FALSE（不忽略）、TRUE（忽略）

返回值:

-1	调用失败。失败原因可以通过调用函数 SsmGetLastErrMsg 获得
0	调用成功

功能描述:

在指定通道上产生一次闪断（拍叉簧）动作，闪断持续的时间由参数 **time** 指定。同时设置闪断后通道的当前状态。

注意事项:

- 本函数只支持 SHT 系列板卡的模拟中继线通道。

相关信息:

驱动程序版本要求	SynCTI Ver. 4.8.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmTxFlash](#)、[SsmChkTxFlash](#)

2.19.2.2 通道状态的强制迁移

2.19.2.2.1 SsmSetChState

强制模拟中继线通道的状态迁移到通话状态。

函数原型:

```
int SsmSetChState (int ch, int nState)
```

参数说明:

ch	通道编号
nState	通道的新状态, 必须设置为 3

返回值:

-1	调用失败
0	调用成功

功能描述:

当模拟中继线通道由于某种意外发生挂起时, 本函数可以强制模拟中继线通道的状态迁移到通话状态。

注意事项:

- 本函数只适用于 SHT 系列板卡的模拟中继线通道。只有在某些特殊的应用场合, 例如应用程序发现通道非正常挂起后, 希望能将通道恢复为通话状态时, 才会用到本函数。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.19.2.3 去话接续时检测被叫摘机的函数

2.19.2.3.1 SsmStartPickupAnalyze

启动“增强的远端摘机检测器”。

函数原型:

```
int SsmStartPickupAnalyze(int ch)
```

参数说明:

ch	通道号
----	-----

返回值:

-1	失败
0	成功

功能描述:

启动“增强的远端摘机检测器”。更多信息请参见第 1 章中“[增强的远端摘机检测器](#)”部分内容。

注意事项:

- 本函数只适用于 SHT 系列板卡的模拟中继线通道。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.5.2.4 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetPickup](#)

2.19.2.3.2 SsmSetCalleeHookDetectP

设置模拟中继线通道上的“增强的远端摘机检测器”的工作参数。

函数原型:

```
int SsmSetCalleeHookDetectP(int ch, WORD wMulti, WORD wValidTime)
```

参数说明:

ch	通道编号
wMulti	灵敏度, 更多信息请参见配置项 HookEngyConfigMulti 的说明
wValidTime	最小保持时间, 更多信息请参见配置项 HookValidEngyCnt 的说明

返回值:

-1	失败
0	成功

功能描述:

设置模拟中继线通道上的“增强的远端摘机检测器”的工作参数。更多信息请参见第1章中“[增强的远端摘机检测器](#)”部分内容。

注意事项:

- 本函数只适用于SHT系列板卡的模拟中继线通道。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.5.2.4 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmStartPickupAnalyze](#)

2.19.2.3.3 SsmGetPickup

查询模拟中继线通道上“增强的远端摘机检测器”的检测结果。

函数原型:

```
int SsmGetPickup (int ch)
```

参数说明:

ch	通道号
----	-----

返回值:

-1	失败
0	被叫用户没有摘机
1	被叫用户已经摘机

功能描述:

查询模拟中继线通道上“增强的远端摘机检测器”的检测结果。更多信息请参见第1章中“[增强的远端摘机检测器](#)”部分内容。

注意事项:

- 使用 [E_SYS_RemotePickup](#) 事件可以实现相同功能（推荐使用）；
- 本函数只适用于 SHT 系列板卡的模拟中继线通道。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.5.2.4 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmStartPickupAnalyze](#), [SsmSetCalleeHookDetectP](#)

2.19.2.4 检测通道摘机执行情况的函数

2.19.2.4.1 SsmCheckActualPickup

查询摘机操作是否执行完毕。

函数原型：

```
int SsmCheckActualPickup(int ch)
```

参数说明：

ch	通道编号
----	------

返回值：

-1	操作失败
0	指定通道处于实际挂机状态
1	指定通道处于实际摘机状态

功能描述：

查询摘机操作是否执行完毕。更多信息请参见函数 [SsmPickup](#) 的说明。

注意事项：

- 本函数只适用于 SHT 系列板卡的模拟中继线通道。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmPickup](#)

2.19.2.5 检测通道是否为模拟中继转成录音通道的函数

2.19.2.5.1 SsmGetIsAnalogToRec

查询该通道是否为模拟中继转录音通道。

函数原型：

```
int WINAPI SsmGetIsAnalogToRec(int ch);
```

参数说明：

ch	通道编号
----	------

返回值:

-1	操作失败
0	指定通道非模拟转录音通道或者非模拟通道
1	指定通道为模拟通道且已转换成录音通道使用.

功能描述:

查询该通道是否为模拟中继转录音通道。更多信息请参见配置项 [SetAnalogChToRecCh](#) 的说明。

注意事项:

- 本函数只适用于 SHT 系列板卡的模拟中继线通道。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.1.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关配置项: [SetAnalogChToRecCh](#)

2.19.3 坐席通道专用函数

2.19.3.1 向电话机发送振铃信号

2.19.3.1.1 SsmStartRing

请参见 [SsmStartRingWithCIDStr](#)

2.19.3.1.2 SsmStartRingWithCIDStr

向电话机发送振铃信号。`SsmStartRing` 只是产生振铃信号，`SsmStartRingWithCIDStr` 还可以向话机发送主叫号码信息。

函数原型:

```
int SsmStartRing(int ch)  
int SsmStartRingWithCIDStr(int ch, LPSTR pCIDBuf, DWORD dwCIDLen, DWORD dwDelayTime)
```

参数说明:

ch	通道编号
pCIDBuf	存放主叫号码信息的缓冲区首地址。例如：“057188861158”
dwCIDLen	pCIDBuf 中主叫号码字符的个数，取值范围：1~20
dwDelayTime	从第 1 个振铃信号的下降沿开始，到线路上出现第 1 个 FSK 数据的时间间隔，单位为毫秒，取值范围：500~1500 注：FSK 方式的主叫号码在第 1 个与第 2 个振铃信号之间传送

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

向电话机发送振铃信号。

函数 [SsmStopRing](#) 可以用来随时终止向电话机发送振铃信号。

注意事项:

- 本函数只适用于 SHT 系列板卡的坐席通道, SsmStartRing 函数可以适用于坐席通道和磁石话机通道, SsmStartRingWithCIDStr 只适用于坐席通道;
- 如果使用 SsmStartRingWithCIDStr 函数, 在函数执行期间, 不能调用任何语音播放的函数。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmStopRing](#)

2.19.3.1.3fPcm_ConvertFskCID

将主叫号码信息用 FSK 方式调制成适合于在模拟电话线上传输的音频信号。

函数原型:

```
DWORD fPcm_ConvertFskCID(LPSTR pFskStream, int nFskLen, LPSTR pszCIDNumber, LPSTR pszTime,  
LPSTR pszName, int nMode)
```

参数说明:

pFskStream	保存调制后的 FSK 比特流的缓冲区首地址, 输出参数, 由应用程序分配存储空间, 空间应不少于 500 字节
nFskLen	pFskStream 缓冲区的大小, 单位为字节
pszCIDNumber	存放主叫号码的缓冲区首地址, 字符个数应在 4~14 之间
pszTime	存放时间信息的缓冲区首地址, 长度为 8 字节, 格式为 “MMDDhhmm”, MM 为月, DD 为日, hh 为时, mm 为分。例如 “12041224” 表示 12 月 4 日 12 点 24 分 如果 pszTime 为 NULL, 驱动程序会使用系统时间来填充
pszName	存放附加信息的缓冲区首地址, 长度为 4~14 字节, 可以为空。 注: 大多数电话机不能显示附加信息
nMode	FSK 主叫号码的帧格式, 0 表示复帧, 1 表示单帧

返回值:

0	调制失败
>0	返回调制后总比特数

功能描述:

将主叫号码信息用 FSK 方式调制成适合于在模拟电话线上传输的音频信号。

注意事项: 无

相关信息:

驱动程序版本要求	ShCti4.5.2.5 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

2.19.3.1.4SsmStopRing

停止向坐席通道或磁石话机通道发送振铃信号。

函数原型:

```
int SsmStopRing(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

停止向坐席通道或磁石话机通道发送振铃信号。发送振铃信号的操作可以是由函数 [SsmStartRing](#)、[SsmStartRingWithCIDStr](#) 发起。

注意事项:

- 本函数只适用于 SHT 系列板卡的坐席通道和磁石话机通道；
- 本函数调用后，保存在驱动程序内部的振铃信号计数器将被清零。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmStartRing](#), [SsmStartRingWithCIDStr](#)

2.19.3.1.5 SsmCheckSendRing

获取坐席通道和磁石话机通道的铃流发生器的进展情况。

函数原型:

```
int SsmCheckSendRing(int ch,int * pnCnt)
```

参数说明:

ch	通道编号
pnCnt	返回保存在驱动程序内部的振铃信号计数器值

返回值:

-1	调用失败
0	通道上没有发送振铃信号
1	通道上正在发送振铃信号

功能描述:

获取坐席通道和磁石话机通道的铃流发生器的进展情况。

注意事项:

- 本函数只适用于 SHT 系列板卡的坐席通道和磁石话机通道。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmStartRing](#), [SsmStartRingWithCIDStr](#)

2.19.3.1.6 SsmSetRingPeriod

设置坐席通道铃流发生器的高低电平持续时间。

函数原型:

```
int WINAPI SsmSetRingPeriod(int ch,WORD wHigh,WORD wLow)
```

参数说明:

ch	通道编号
wHigh	铃流信号高电平持续时间, 单位 ms, 最小 16ms
wLow	铃流信号低电平持续时间, 单位 ms, 最小 16ms

返回值:

-1	调用失败
0	调用成功

功能描述:

设置坐席通道铃流发生器的高低电平持续时间。

注意事项:

- 本函数适用于 SHT 系列板卡的坐席通道和大容量坐席卡的坐席通道。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.1.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmStartRing](#), [SsmStartRingWithCIDStr](#)

2.19.3.2 检测到话机摘机后自动送拨号音

2.19.3.2.1 SsmSetASDT

设置当检测到与坐席通道相连的话机摘机后, 是否由驱动程序自动发送拨号音的功能。

函数原型:

```
int SsmSetASDT(int ch, BOOL bEnAutoSendDialTone)
```

参数说明:

ch	通道编号
bEnAutoSendDialTone	=TRUE: 开启 =FALSE: 关闭

返回值:

-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

设置当检测到与坐席通道相连的话机摘机后, 是否由驱动程序自动发送拨号音的功能。

注意事项:

- 本函数只适用于 SHT 系列板卡的坐席通道。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetASDT](#)

2.19.3.2 SsmGetASDT

获取当检测到与坐席通道相连的话机摘机后，是否由驱动程序自动发送拨号音的标志。

函数原型:

```
int SsmGetASDT(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
0	状态关
1	状态开

功能描述:

获取当检测到与坐席通道相连的话机摘机后，是否由驱动程序自动发送拨号音的标志。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetASDT](#)

2.19.3.3 检测话机的闪断信号

2.19.3.3.1 SsmSetLocalFlashTime

设置判断闪断信号的最大保持时间。

函数原型:

```
int SsmSetLocalFlashTime(int nFlashTime)
```

参数说明:

nFlashTime	闪断信号的最大保持时间，单位为毫秒，取值范围：32~2000
------------	--------------------------------

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

设置判断闪断信号的最大保持时间。详细信息请参见第1章中“[检测话机的闪断信号](#)”部分内容。

注意事项:

- 本函数只适用于坐席通道;
- 配置项 [MaxLocalFlashTime](#) 也可实现相同功能, 缺省值为 700 毫秒。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.19.3.3.2 SsmGetFlashCount

获取坐席通道或者录音通道上保存在驱动程序内部的闪断信号计数器的值。

函数原型:

```
int SsmGetFlashCount(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
≥0	闪断信号计数器的值

功能描述:

获取坐席通道或者录音通道上保存在驱动程序内部的闪断信号计数器的值。

注意事项:

- 本函数只适用于坐席通道和录音通道;
- 坐席通道在进入摘机状态或挂机状态时驱动程序自动将闪断计数器清零。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmClearFlashCount](#)

2.19.3.3.3 SsmClearFlashCount

清零指定坐席或录音通道上的闪断计数器。

函数原型:

```
int SsmClearFlashCount (int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
0	调用成功

功能描述:

清零指定坐席通道或者录音通道上的闪断计数器。

注意事项:

- 本函数只适用于坐席通道和录音通道；
- 坐席通道在进入摘机状态或挂机状态时驱动程序自动将闪断计数器清零。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetFlashCount](#)

2.19.3.4 检测话机的摘机/挂机动作

2.19.3.4.1 SsmGetHookState

获取坐席和 EM 控制通道的摘机/挂机状态。

函数原型:

```
int SsmGetHookState(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
0	挂机状态
1	摘机状态

功能描述:

获取坐席和 EM 控制通道的摘机/挂机状态。使用 [E_CHG_HookState](#) 事件也可以实现同样功能。

注意事项:

- 本函数只适用于 SHT 系列板卡的坐席通道和 EM 控制通道；
- 当坐席通道迁移到“摘机”状态或“挂机”状态时，驱动程序会将内部的闪断计数器自动清零。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关事件: [E_CHG_HookState](#)

2.19.3.5 在电话线上产生极性反转信号

2.19.3.5.1 SsmSetPolarState

设置与坐席通道相连的模拟电话线的馈电电压极性。

函数原型:

```
int SsmSetPolarState(int ch, int nPolar)
```

参数说明:

ch	通道编号
nPolar	要设置的通道极性。取值范围: 0 或 1

返回值:

-1	调用失败
0	调用成功

功能描述:

设置与坐席通道相连的模拟电话线的馈电电压极性。通道的当前电压极性可以通过函数 [SsmGetPolarState](#) 获得。
详细信息请参见第 1 章中 “[在电话线上产生极性反转信号](#)” 部分内容。

注意事项:

- 只有在配置项 [UserSendPolar](#) 的设定值为 1 时，才能调用本函数；
- 本函数只适用于 SHT 系列板卡的坐席通道。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetPolarState](#)

代码范例:

```
int v = SsmGetPolarState(ch);
SsmSetPolarState(ch, ~v);
```

2.19.3.5.2 SsmGetPolarState

获取与坐席通道相连的模拟电话线的馈电电压极性。

函数原型:

```
int SsmGetPolarState(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
0,1	返回当前的极性状态

功能描述:

获取与坐席通道相连的模拟电话线的馈电电压极性。

注意事项:

- 本函数只适用于 SHT 系列板卡的坐席通道。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetPolarState](#)

2.19.4 有关远端摘机检测器的函数（SHT 系列 only）

2.19.4.1 有关普通远端摘机检测器的函数

2.19.4.1.1 设置工作参数

2.19.4.1.1.1 SsmSetVoiceOnDetermineTime

设置语音活动信号的最小保持时间。

函数原型:

```
int SsmSetVoiceOnDetermineTime(int ch, WORD wlsVocDtrTime)
```

参数说明:

ch	通道编号
wlsVocDtrTime	最小保持时间，单位为毫秒，取值范围：16ms~1024ms

返回值:

-1	调用失败，失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

设置语音活动信号的最小保持时间。只有当线路上连续出现语音活动信号，并且保持的时间大于参数 wlsVocDtrTime 的设定值时，驱动程序才会确认线路上出现了真正的语音活动信号。更多信息请参见第 1 章中“[普通的远端摘机检测器](#)”部分内容。

注意事项:

- 配置项 [VoiceOnDetermineTime](#) 可以实现相同功能；
- 本函数只适用于 SHT 系列板卡的模拟中继线通道。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetVoiceOnDetermineTime](#)

2.19.4.1.2 获取工作参数

2.19.4.1.2.1 SsmGetVoiceOnDetermineTime

获取语音活动信号的最小保持时间。

函数原型:

```
int SsmGetVoiceOnDetermineTime(int ch, PWORD pwlsVocDtrTime)
```

参数说明:

ch	通道编号
pwlsVocDtrTime	返回最小保持时间, 单位为毫秒

返回值:

-1	调用失败, 失败原因可以从函数 SsmGetLastErrMsg 取得
0	调用成功

功能描述:

取得驱动程序判定线路上出现话音信号的最小判决时间。

注意事项:

- 本函数只适用于 SHT 系列板卡的模拟中继线通道。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetVoiceOnDetermineTime](#)

2.20 SHD 系列板卡的高级编程 API (CTI 系列)

2.20.1 通道的闭塞控制 (CTI 系列)

2.20.1.1 闭塞本端的函数

2.20.1.1.1 闭塞单个通道

2.20.1.1.1.1 SsmBlockLocalCh

闭塞本端通道, 以禁止其进行去话呼叫, 但仍然可以处理来话呼叫 (TUP 通道和 ISUP 通道); 或来话及去话呼叫均禁止 (ISDN 通道)。

函数原型:

```
int SsmBlockLocalCh(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
0	调用成功

功能描述:

闭塞本端通道。通道被闭塞后，对于 TUP 和 ISUP 通道，不能进行去话呼叫，但仍然可以处理来话呼叫；对于 ISDN 通道，来话及去话呼叫均不支持。有关通道闭塞的详细内容请参见第 1 章中“[通道的闭塞与解除](#)”部分内容。

本函数适用于 TUP、ISUP 通道及 ISDN 通道，会对通道的状态机产生影响。有关 TUP、ISUP 及 ISDN 通道的状态机的详细说明请参见第 1 章中“[TUP 通道的状态转移](#)”和“[ISDN 通道的状态转移图](#)”部分内容（本函数触发的事件在状态机迁移图中的名称为 Blocking）。

注意事项:

- 本函数只适用于 TUP、ISUP 通道及 ISDN 通道；
- 本函数可以在任意时刻调用，但当 TUP 或 ISUP 通道上正在执行 AutoDial 任务时，本函数会导致任务立即失败；而当 ISDN 通道正在执行 AutoDial 任务时，本函数不会导致任务立即失败。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmUnblockLocalCh](#), [SsmBlockLocalPCM](#), [SsmQueryLocalChBlockState](#),
[SsmBlockRemoteCh](#), [SsmBlockRemotePCM](#)

2.20.1.1.1.2 SsmUnblockLocalCh

解除对本端通道闭塞。

函数原型:

```
int SsmUnblockLocalCh(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
0	调用成功

功能描述:

本函数解除对本端通道闭塞，包括因应用程序调用函数 [SsmBlockLocalCh](#) 或 [SsmBlockLocalPCM](#) 导致的某个通道的本地闭塞。

注意事项:

- 本函数只适用于 TUP、ISUP 通道及 ISDN 通道。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmBlockLocalCh](#), [SsmUnblockLocalPCM](#), [SsmQueryLocalChBlockState](#)

2.20.1.1.1.3 SsmQueryLocalChBlockState

获取通道的本端闭塞状态。

函数原型:

```
int SsmQueryLocalChBlockState(int ch, PDWORD pdwBlockState)
```

参数说明:

ch	通道编号 返回通道的本端闭塞状态。比特为 1 表示闭塞，0 表示未闭塞。各比特的含义为：														
pdwBlockState	<table border="1"><tr><td>Bit 0</td><td>因应用程序调用 SsmBlockLocalCh 函数而闭塞</td></tr><tr><td>Bit 1</td><td>因收到对端电路闭塞消息 BLO 而闭塞</td></tr><tr><td>Bit 2</td><td>因收到对端电路群闭塞消息 SGB 而闭塞</td></tr><tr><td>Bit 3</td><td>因收到对端电路群闭塞消息 HGB 而闭塞</td></tr><tr><td>Bit 4</td><td>因收到对端电路群闭塞消息 MGB 而闭塞</td></tr><tr><td>Bit 5</td><td>因应用程序调用 SsmBlockLocalPCM 函数而闭塞</td></tr><tr><td>Bit31~Bit6</td><td>保留</td></tr></table>	Bit 0	因应用程序调用 SsmBlockLocalCh 函数而闭塞	Bit 1	因收到对端电路闭塞消息 BLO 而闭塞	Bit 2	因收到对端电路群闭塞消息 SGB 而闭塞	Bit 3	因收到对端电路群闭塞消息 HGB 而闭塞	Bit 4	因收到对端电路群闭塞消息 MGB 而闭塞	Bit 5	因应用程序调用 SsmBlockLocalPCM 函数而闭塞	Bit31~Bit6	保留
Bit 0	因应用程序调用 SsmBlockLocalCh 函数而闭塞														
Bit 1	因收到对端电路闭塞消息 BLO 而闭塞														
Bit 2	因收到对端电路群闭塞消息 SGB 而闭塞														
Bit 3	因收到对端电路群闭塞消息 HGB 而闭塞														
Bit 4	因收到对端电路群闭塞消息 MGB 而闭塞														
Bit 5	因应用程序调用 SsmBlockLocalPCM 函数而闭塞														
Bit31~Bit6	保留														

返回值:

-1	调用失败
0	调用成功

功能描述:

本函数获取通道是否被本端闭塞，以及闭塞的具体原因。

注意事项:

- 本函数只适用于 TUP 通道。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmBlockLocalCh](#), [SsmBlockLocalPCM](#)

2.20.1.1.2 闭塞整条数字中继线

2.20.1.1.2.1 SsmBlockLocalPCM

闭塞本端电路群（整条数字中继线）。

函数原型:

```
int SsmBlockLocalPCM(int nLocalPcmNo)
```

参数说明:

nLocalPcmNo	PCM 的逻辑编号
-------------	-----------

返回值:

-1	调用失败
0	调用成功

功能描述:

本函数闭塞本端整条数字中继线，使其包含的本端全部通道都不能进行去话呼叫，但仍然能够处理来话呼叫，

本函数的执行效果相当于对该 PCM 中的全部通道调用 [SsmBlockLocalCh](#)。有关本函数的更多信息请参见函数 [SsmBlockLocalCh](#) 的说明。

注意事项:

- 本函数只适用于 TUP、ISUP 通道。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmUnblockLocalPCM](#), [SsmQueryLocalPCMBlockState](#), [SsmBlockLocalCh](#)

2.20.1.1.2.2 SsmUnblockLocalPCM

解除对本端整条数字中继线的闭塞。

函数原型:

```
int SsmUnblockLocalPCM(int nLocalPcmNo)
```

参数说明:

nLocalPcmNo	PCM 的逻辑编号
-------------	-----------

返回值:

-1	调用失败
0	调用成功

功能描述:

本函数解除数字中继线上由于应用程序调用函数 [SsmBlockLocalPCM](#) 而引起的闭塞。本函数的执行效果相当于对该数字中继线所包含的所有通道调用 [SsmUnblockLocalCh](#)。因此，有关本函数的情况可参见函数 [SsmUnblockLocalCh](#) 的说明。

注意事项:

- 本函数只适用于 TUP、ISUP 通道。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmBlockLocalPCM](#), [SsmQueryLocalPCMBlockState](#), [SsmUnblockLocalCh](#)

2.20.1.1.2.3 SsmQueryLocalPCMBlockState

获取数字中继线的本地闭塞状态。

函数原型:

```
int SsmQueryLocalPCMBlockState(int nLocalPcmNo, PDWORD pdwBlockState)
```

参数说明:

nLocalPcmNo	PCM 的逻辑编号
pdwBlockState	返回通道的本端闭塞状态。比特为 1 表示闭塞，0 表示未闭塞。各比特的含义为：

	Bit1~Bit0	保留
	Bit 2	因收到对端交换机的电路群闭塞消息 SGB 而闭塞
	Bit 3	因收到对端交换机的电路群闭塞消息 HGB 而闭塞
	Bit 4	因收到对端交换机的电路群闭塞消息 MGB 而闭塞
	Bit 5	因应用程序调用 SsmBlockLocalPCM 函数而闭塞
	Bit31~Bit6	保留

返回值:

-1	调用失败
0	调用成功

功能描述:

本函数获取数字中继线的本地闭塞状态。

注意事项:

- 本函数只适用于 TUP 通道。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmBlockLocalCh](#), [SsmBlockLocalPCM](#)

2.20.1.2 闭塞对端的函数

2.20.1.2.1 闭塞单个通道

2.20.1.2.1.1 SsmQueryOpBlockRemoteCh

查询通道是否支持闭塞对端的操作。

函数原型:

```
int SsmQueryOpBlockRemoteCh(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
0	指定通道不支持闭塞对端的操作
1	指定通道支持闭塞对端的操作

功能描述:

查询通道是否支持闭塞对端通道的操作。

注意事项:

- 只有 TUP 通道和 ISUP 通道才支持闭塞对端通道的操作。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib

DLL	shp_a3.dll
-----	------------

相关函数:

2.20.1.2.1.2 SsmBlockRemoteCh

闭塞对端通道。

函数原型:

```
int SsmBlockRemoteCh(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
0	调用成功

功能描述:

闭塞对端通道，以阻止对端交换机在此通道上向本端发起呼叫，但本端仍可以向对端交换机发起去话呼叫。对端通道是指对端交换机中具有与本端通道相同的 CIC 编号的电路。当对端通道进入闭塞状态后，其去话呼叫被禁止，本端通道上就可以保证没有来话呼叫。闭塞对端通道的功能通常用于系统维护等目的。

闭塞对端通道的消息处理过程为:

- (1) 驱动程序向对端交换机发送电路闭塞消息 BLO;
- (2) 对端交换机收到 BLO 消息后，应答闭塞证实信号 BLA;
- (3) 收到对端交换机的 BLA 消息后，驱动程序向应用程序抛出 [E_CHG_RemoteChBlock](#) 事件，闭塞对端通道操作完成。

函数 [SsmGetRemoteChBlockStatus](#) 也可以用来检查闭塞对端通道的操作是否完成。

注意事项:

- 只有 TUP 通道和 ISUP 通道才支持闭塞对端的操作，调用本函数后 TUP 通道为空闲；ISUP 通道为本地闭塞，但仍可外呼，也可以使用 [SsmSearchIdleCallOutCh](#) 函数搜索到该通道。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmUnblockRemoteCh](#), [SsmGetRemoteChBlockStatus](#)

2.20.1.2.1.3 SsmUnblockRemoteCh

解除对对端通道的闭塞。

函数原型:

```
int SsmUnblockRemoteCh(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
----	------

0	调用成功
---	------

功能描述:

解除对对端通道的闭塞，使对端交换机恢复在此通道上向本端发起呼叫的能力。解除闭塞对端通道的消息处理过程为：

- (1) 驱动程序向对端交换机发送电路解除闭塞消息 UBL;
- (2) 对端交换机收到 UBL 消息后，应答解除闭塞证实信号 UBA;
- (3) 收到对端交换机的 UBA 消息后，驱动程序向应用程序抛出 [E_CHG_RemoteChBlock](#) 事件，解除闭塞对端通道操作完成。

函数 [SsmGetRemoteChBlockStatus](#) 也可以用来检查解除闭塞对端通道的操作是否完成。

注意事项:

- 只有 TUP 通道和 ISUP 通道才支持闭塞对端的操作。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmBlockRemoteCh](#), [SsmGetRemoteChBlockStatus](#), [SsmBlockRemotePCM](#)

2.20.1.2.1.4 SsmGetRemoteChBlockStatus

查询对端交换机上的通道是否被本端闭塞。

函数原型:

```
int SsmGetRemoteChBlockStatus(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
0	没有被本端闭塞，或已经成功解除由本端导致的闭塞
1	已经成功闭塞对端
2	已经发出电路闭塞消息，正在等待对端交换机的闭塞证实信号
3	已经发出电路解除闭塞消息，正在等待对端交换机的解除闭塞证实信号

功能描述:

查询对端交换机上的通道是否被本端闭塞，返回结果可以用来检查函数 [SsmBlockRemoteCh](#) 或 [SsmUnblockRemoteCh](#) 的执行进展情况。

注意事项:

- 只有 TUP 通道和 ISUP 通道才支持闭塞对端的操作。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmBlockRemoteCh](#), [SsmUnblockRemoteCh](#)

2.20.1.2.2 闭塞整条数字中继线

2.20.1.2.2.1 SsmBlockRemotePCM

闭塞对端电路群。

函数原型:

```
int SsmBlockRemotePCM(int nLocalPcmNo, DWORD dwBlockMode)
```

参数说明:

nLocalPcmNo	PCM 逻辑编号 选择向对端交换机发送的电路群闭塞消息。取值范围为: TUP 协议: 1: 发送面向维护的群闭塞消息 MGB 2: 发送面向硬件故障的群闭塞消息 HGB 3: 发送软件产生的群闭塞消息 SGB ISUP 协议: 0: 发送面向维护的群闭塞消息 CGB 1: 发送面向硬件故障的群闭塞消息 CGB
dwBlockMode	

返回值:

-1	调用失败
0	调用成功

功能描述:

用电路群闭塞消息将对端交换机的整条数字中继线闭塞，使对端交换机中包含在此 PCM 上的全部通道都不能进行去话呼叫，只能处理来话呼叫。本函数的执行效果相当于对数字中继线上的全部通道同时调用 [SsmBlockRemoteCh](#)。

闭塞对端电路群的消息处理过程为：

- (1) 驱动程序根据 dwBlockMode 参数的设置，向对端交换机发送指定的电路群闭塞消息；
- (2) 对端交换机收到电路群闭塞消息后，应答电路群闭塞证实消息（TUP 协议的电路群闭塞消息 MGB、HGB、SGB 所对应的电路群闭塞证实消息分别为 MBA、HBA、SBA，ISUP 协议的电路群闭塞消息 CGB 所对应的电路群闭塞证实消息为 CGBA）；
- (3) 收到对端交换机的电路群闭塞证实消息后，驱动程序向应用程序抛出 [E_CHG_RemotePCMBLOCK](#) 事件，闭塞对端电路群的操作完成。

函数 [SsmGetRemotePCMBLOCKStatus](#) 也可以用来检查对端电路群的解除闭塞操作是否完成。

注意事项:

- 本函数只支持 TUP 协议和 ISUP 协议。调用本函数后 TUP 通道显示为空闲；ISUP 通道显示为本地闭塞，但仍可外呼，也可以使用 [SsmSearchIdleCallOutCh](#) 函数搜索到该通道；
- 按照 TUP 协议和 ISUP 协议的规定，电路群闭塞消息需要发送 2 次才有效，所以驱动程序会向对端交换机连续发送 2 次电路群闭塞消息，因此应用程序只需调用本函数一次。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmUnblockRemotePCM](#), [SsmGetRemotePCMBlockStatus](#)

相关事件: [E_CHG_RemotePCMBlock](#)

2.20.1.2.2.2 SsmUnblockRemotePCM

解除闭塞对端电路群。

函数原型:

```
int SsmUnblockRemotePCM(int nLocalPcmNo, DWORD dwUnblockMode)
```

参数说明:

nLocalPcmNo	PCM 逻辑编号
	选择向对端交换机发送的电路群解除闭塞消息。取值范围为: TUP 协议: 1: 发送 MGU, 以解除因 MGB 消息而引起的电路群闭塞 2: 发送 HGU, 以解除因 HGB 消息而引起的电路群闭塞 3: 发送 SGU, 以解除因 SGB 消息而引起的电路群闭塞 ISUP 协议: 0: 发送面向维护的群解除闭塞消息 CGU 1: 发送面向硬件故障的群解除闭塞消息 CGU
dwBlockMode	

返回值:

-1	调用失败
0	调用成功

功能描述:

解除对对端交换机的整条数字中继线闭塞, 使对端交换机中包含在此数字中继线上的全部通道恢复去话呼叫的能力。本函数的执行效果相当于对数字中继线上的全部通道同时调用 [SsmUnblockRemoteCh](#)。

闭塞对端电路群的消息处理过程为:

- (1) 驱动程序根据 dwBlockMode 参数的设置, 向对端交换机发送指定的电路群解除闭塞消息;
- (2) 对端交换机收到电路群解除闭塞消息后, 应答电路群解除闭塞证实消息 (TUP 协议的电路群解除闭塞消息 MGU、HGU、SGU 所对应的电路群解除闭塞证实消息分别为 MUA、HUA、SUA, ISUP 协议的电路群解除闭塞消息 CGU 所对应的电路群解除闭塞证实消息为 CGUA);
- (3) 收到对端交换机的电路群解除闭塞证实消息后, 驱动程序向应用程序抛出 [E_CHG_RemotePCMBlock](#) 事件, 解除闭塞对端电路群的操作完成。

函数 [SsmGetRemotePCMBlockStatus](#) 也可以用来检查闭塞解除对端电路群的操作是否完成。

注意事项:

- 本函数只支持 TUP 协议和 ISUP 协议;
- 按照 TUP 协议和 ISUP 协议的规定, 电路群解除闭塞消息需要发送 2 次才有效, 所以驱动程序会向对端交换机连续发送 2 次电路群解除闭塞消息, 因此, 应用程序只需调用本函数一次。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmBlockRemotePCM](#), [SsmGetRemotePCMBlockStatus](#)

相关事件: [E_CHG_RemotePCMBlock](#)

2.20.1.2.2.3 SsmGetRemotePCMBlockStatus

获取电路群的闭塞状态。

函数原型:

```
int SsmGetRemotePCMBlockStatus(int nLocalPcmNo, DWORD dwBlockMode)
```

参数说明:

nLocalPcmNo	PCM 逻辑编号
	选择希望查询的电路群闭塞类型。取值范围为: TUP 协议: 1: 面向维护的电路群闭塞状态 2: 面向硬件故障的电路群闭塞状态 3: 软件产生的电路群闭塞状态 ISUP 协议: 0: 面向维护的电路群闭塞状态 1: 面向硬件故障的电路群闭塞状态
dwBlockMode	

返回值:

-1	调用失败
0	没有闭塞对端电路群, 或者已经成功解除闭塞对端电路群
1	已经成功闭塞对端电路群
2	已经发出电路群闭塞消息, 正在等待对端交换机的证实信号
3	已经发出电路群解除闭塞消息, 正在等待对端交换机的证实信号

功能描述:

获取电路群的闭塞状态。返回结果可以用来检查函数 [SsmBlockRemotePCM](#) 或 [SsmUnblockRemotePCM](#) 的执行情况。

注意事项:

- 本函数只支持 TUP 协议和 ISUP 协议。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmBlockRemotePCM](#), [SsmUnblockRemotePCM](#)

2.20.2 基于 SS7-MTP3 的高级编程 API

2.20.2.1 SsmSendSs7Msu

向对端信令点发送一条消息 (MSU)。

函数原型:

```
int SsmSendSs7Msu(WORD wMsuLength, PUCHAR pucMsuBuf)
```

参数说明:

wMsuLength	发送 MSU 的长度, 单位为字节, 最大不能超过 273
pucMsuBuf	存放 MSU 数据的首地址指针。pucMsuBuf 中的第 1 个字节 (即 pucMsuBuf[0]) 必须为

SIO 字段
返回值:

0	发送成功
	发送失败。失败原因可能为:
-1	<ul style="list-style-type: none"> ◆ MTP3 业务中断 ◆ 系统不支持 SS7 信令 ◆ 系统的 API 接口函数尚未开放

功能描述:

向对端信令点发送一条消息 (MSU)。

注意事项: 无

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetSs7Msu](#)

2.20.2.2 SsmGetSs7Msu

查询驱动程序的 MSU 接收缓冲区中是否有 MSU 消息。

函数原型:

```
int SsmGetSs7Msu(PUCHAR* ppucMsuBuf)
```

参数说明:

ppucMsuBuf	返回保存在驱动程序内部缓冲区的 MSU 消息的首地址指针。MSU 消息的第一个字节为 SIO 字段。(注意: ppucMsgBuf 指针应该在调用前声明, 并且不能指定其长度)
------------	------------------------------------------------------------------------------------------

返回值:

0	驱动程序内部缓冲区的 MSU 消息接收缓冲区为空
>0	当前 MSU 的长度, 单位为字节
	操作失败。失败原因可能是:
-1	<ul style="list-style-type: none"> ◆ 系统不支持 SS7 信令 ◆ 系统的 API 接口函数尚未开放

功能描述:

查询驱动程序中的 MSU 接收缓冲区中是否有 MSU 消息, 如果有, 取出最早收到的消息。

每当驱动程序收到一条 MSU, 会将其保存到驱动程序内部的 MSU 接收缓冲区中, 然后向应用程序抛出 [E_RCV_Ss7Msu](#) 消息。

注意事项:

- SS7 MSU 的输出方式可由配置项 [GetMsuOnAutoHandle](#) 设定;
- 驱动程序内部的 MSU 接收缓冲区填满后, 后续到达的 MSU 将被丢弃。因此, 应用程序应保证尽快将消息取出并处理。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib

DLL	shp_a3.dll
-----	------------

相关函数: [SsmSendSs7Msu](#)

相关事件: [E_RCV_Ss7Msu](#)

2.20.2.3 SsmGetMtp3State

请参见 [SsmGetMtp3StateEx](#)

2.20.2.4 SsmGetMtp3StateEx

获取源信令点到目的地信令点之间的业务是否开通。

函数原型:

```
int SsmGetMtp3State()  
int SsmGetMtp3StateEx(int nDpcNo)
```

参数说明:

nDpcNo	目的地信令点的编号。目的地的信令点编号与信令点编码之间的对应关系在 SS7 信服务 器的配置程序中指定
--------	--------------------------------------------------------

返回值:

1	业务开通
0	业务中断
-1	调用失败

功能描述:

获取源信令点到目的地信令点之间的业务是否开通。只有当源信令点到目的地信令点处于业务开通状态时，应用程序才能调用函数 [SsmSendSs7Msu](#) 向 DPC 发送消息。

SsmGetMtp3StateEx 支持多个目的地信令点的应用场合，而 SsmGetMtp3State 是一个老版本的函数，只适用于系统中与单个目的地信令点直联的应用场合，相当于将 nDpcNo 设置为 0 时对 SsmGetMtp3StateEx 进行调用。

注意事项:

- 由于 SS7 信令具有自恢复功能，因此当应用程序使用本函数作为 SS7 信令系统监控的告警源时，应该在检测到业务中断并持续一定时间后，才发出告警信号；
- 当 SS7 Server 检测到到达目的地信令点的链路状态发生变化时，在保存该信令点的业务状态后，会向驱动程序抛出 [E_CHG_Mtp3State](#) 事件。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetMtp2Status](#)

相关事件: [E_CHG_Mtp3State](#)

2.20.2.5 SsmGetMtp2Status

获取 SS7 信令链路组中的 64kbps 信令链路的工作状态。

函数原型:

```
int SsmGetMtp2Status(int nLinkNum)
```

参数说明:

nLinkNum	信令链路编号，在SS7信服务器的配置程序中指定
----------	-------------------------

返回值：

0	DSP的程序已经上载成功，但未能启动运行
1	业务中断
2	初始定位中
3	已定位/准备好
4	已定位/未准备好
5	业务开通
6	处理机故障
-1	调用失败

功能描述：

本函数取得SS7信令下信令链路的状态，只有在状态等于5（业务开通）时，该信令链路才可用。

注意事项：

- 本函数返回MTP2层的信息，通常用于显示等监视和维护目的。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmGetMtp3State](#), [SsmGetMtp3StateEx](#)

2.20.2.6 SsmSendSs7MsuEx

根据输入的通道号(ch)对消息头打包，消息内容由应用程序自行配置。

函数原型：

```
int WINAPI SsmSendSs7MsuEx(int ch, int nNewStep, WORD wMsuLength, PUCHAR pucMsuBuf)
```

参数说明：

ch	通道号
nNewStep	预留
wMsuLength	消息内容长度，内容长度从消息类型开始计算，不包括消息头
pucMsuBuf	消息长度

返回值：

-1	失败，参数错误
0	失败，发送失败
1	成功

功能描述：

该函数根据输入的通道号(ch)对消息头打包，消息内容由应用程序自行配置。

注意事项：无**相关信息：**

驱动程序版本要求	SynCTI Ver. 5.1.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：无

2.20.2.7 SsmGetMaxSs7link

获取系统中已配置的 SS7 link 总数。

函数原型:

```
int WINAPI SsmGetMaxSs7link( )
```

参数说明:

返回值:

-1	失败
≥ 0	系统中已配置的 SS7 link 总数

功能描述:

获取系统中已配置的 SS7 link 总数。

注意事项: 无

相关信息:

驱动程序版本要求	SynCTI Ver. 5.1.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.20.2.8 SsmGetSs7Mtp2Msu

在指定的 SS7 link 链路上获取 MTP2 层 MSU 消息。

函数原型:

```
int WINAPI SsmGetSs7Mtp2Msu(int ss7link, PUCHAR pucPara, PUCHAR* ppucMsuBuf)
```

参数说明:

ss7link	SS7 link 编号
pucPara	预留
ppucMsuBuf	返回保存在驱动程序内部缓冲区的 MSU 消息的首地址指针。MSU 消息的第一个字节为 SIO 字段。(注意: ppucMsgBuf 指针应该在调用前声明。)

返回值:

	操作失败。失败原因可能是:
-1	1) 系统不支持 SS7 信令 2) 系统的 API 接口函数尚未开放
0	驱动程序内部缓冲区的 MSU 消息接收缓冲区为空
>0	当前 MSU 的长度, 单位为字节

功能描述:

查询驱动程序中的 MTP2 MSU 接收缓冲区中是否有 MSU 消息, 如果有, 取出最早收到的消息。

注意事项:

- SS7 MTP2 MSU 处理方式可由配置项 [AppHandleMtp2Msu](#) 设定;
- 驱动程序内部的 MSU 接收缓冲区填满后, 后续到达的 MSU 将被丢弃。因此, 应用程序应保证尽快将消息取出并处理。
- 可以用事件 [E_RCV_SS7Mtp2Msu](#) 获取消息到达通知。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.1.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSendSs7Mtp2Msu](#)

2.20.2.9 SsmSendSs7Mtp2Msu

在指定的 SS7 link 链路上发送 MTP2 层 MSU 消息。

函数原型:

```
int WINAPI SsmSendSs7Mtp2Msu(int ss7link, WORD wMsuLength, PUCHAR pucMsuBuf)
```

参数说明:

ss7link	SS7 link 编号
wMsuLength	发送 MSU 的长度, 单位为字节, 最大不能超过 273
pucMsuBuf	MSU 消息的首地址指针, MSU 消息的第一个字节为 SIO 字段

返回值:

-1	发送失败
0	发送成功

功能描述:

在指定的 SS7 link 链路上发送 MTP2 层 MSU 消息。

注意事项:

- SS7 MTP2 MSU 处理方式可由配置项 [AppHandleMtp2Msu](#) 设定。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.1.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetSs7Mtp2Msu](#)

2.20.2.10 SsmSs7Mtp2CmdCtrl

在指定的 SS7 link 链路上发送 MTP2 控制命令。

函数原型:

```
int WINAPI SsmSs7Mtp2CmdCtrl(int ss7link, int l3_cmd, unsigned char *param, int len)
```

参数说明:

ss7link	SS7 link 编号
	MTP3 层命令 取值范围: enum { SS7_MTP2_START = 1, SS7_MTP2_STOP, SS7_MTP2_EMGCY,
l3_cmd	

	SS7_MTP2_EMGCY_CLRD, SS7_MTP2_RTV_BSNT, SS7_MTP2_RTVL_REQ };
param	命令参数
len	命令参数的长度

返回值:

-1	发送失败
0	发送成功

功能描述:

在指定的 SS7 link 链路上发送 MTP2 控制命令。

注意事项:

- SS7 MTP2 命令发送功能由配置项 [AppHandleMtp2Msu](#) 设定。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.2.1 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetSs7Mtp2Msu](#)

2.20.2.11 SsmGetDecodeSs7Msu

从驱动程序的 MSU 接收缓冲区中获取主被叫号码、DPC、OPC 以及原被叫号码信息。

函数原型:

```
int SsmGetDecodeSs7Msu(Ss7Msu *pDecodeSs7Msu)
```

参数说明:

pDecodeSs7Msu	指向 Ss7Msu 结构的指针
---------------	-----------------

返回值:

0	驱动程序内部缓冲区的 IAM 或 IAI 接收缓冲区为空或者消息已取完
>0	当前 IAM 或 IAI 消息的长度, 单位为字节
-1	操作失败。失败原因可能是: ◆ 系统不支持 SS7 信令 ◆ 系统的 API 接口函数尚未开放

功能描述:

完成一次 ISUP 或 TUP 通话后, 能获取含主被叫号码, DPC、OPC 等参数的消息, 并取出最早收到的消息。

每当驱动程序收到一条 MSU, 会将其保存到驱动程序内部的 MSU 接收缓冲区中, 然后向应用程序抛出 E_RCV_DecodeSs7Msu 消息。

返回的 Ss7Msu 结构体具体如下:

```
struct Ss7Msu
```

```
{
```

```
    UCHAR ucCallerNumber[50];//主叫号码
```

```
    UCHAR ucCalledNumber[50];//被叫号码
```

```
    DWORD dwDPC[3];           //DPC
```

```
DWORD dwOPC[3];           //OPC
UCHAR ucOriginalCalledNum[50];//原始被叫号码
WORD wCicPcm;             //CIC 字段的 PCM 参数值
WORD wCicTs;               //CIC 字段的时隙号
};

注意事项:
```

- SS7 MSU 的输出方式可由配置项 GetMsuOnAutoHandle 设定；
- 驱动程序内部的 MSU 接收缓冲区填满后，后续到达的 MSU 将被丢弃。因此，应用程序应保证尽快将消息取出并处理。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关事件: [E_RCV_DecodeSs7Msu](#)

2.20.3 有关 ISDN 信令的高级函数

2.20.3.1 SsmGetIsdnMsu

从驱动程序的内部缓冲区中取出一条 ISDN 消息。

函数原型:

```
int SsmGetIsdnMsu(int nPcmId, PUCHAR pucMsuBuf)
```

参数说明:

nPcmId	数字中继线的逻辑编号
pucMsuBuf	返回 ISDN 消息。pucMsuBuf 由应用程序分配存储空间，大小不能低于 256 字节

返回值:

0	缓冲区中没有消息
>0	pucMsuBuf 中消息的实际长度，单位为字节
-1	调用失败

功能描述:

从驱动程序的内部缓冲区中取出一条 ISDN 消息。

注意事项:

- 当配置项 [AutoHandleIsdn](#) 设置为 1 时，由驱动程序处理消息；当配置项 [AutoHandleIsdn](#) 设置为 0 时，由应用程序处理消息。
- 在实际应用中，只有当配置项 [AutoHandleIsdn](#) 设置为 0 时，调用本函数才有意义。
- 当用 DTP 卡监控 ISDN 线路时，驱动程序的内部缓冲区收到一条 ISDN 消息，就会抛出 [E_RCV_IsdnSpyMsu](#) 事件。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmCheckIsdnMsu](#)

2.20.3.2 SsmSendIsdnMsu

发送一条 ISDN 消息。

函数原型:

```
int SsmSendIsdnMsu(int nPcmId, int nMsgLen, PUCHAR pucMsuBuf)
```

参数说明:

nPcmId	数字中继线的逻辑编号
nMsgLen	pucMsuBuf 中消息的实际长度, 单位为字节
pucMsuBuf	ISDN 消息缓冲区

返回值:

0	调用成功
-1	调用失败

功能描述:

在指定的数字中继线的 16 时隙上发送一条 ISDN 消息。

注意事项:

- 当配置项 [AutoHandleIsdn](#) 设置为 1 时, 由驱动程序处理消息; 当配置项 [AutoHandleIsdn](#) 设置为 0 时, 由应用程序处理消息。
- 在实际应用中, 只有当配置项 [AutoHandleIsdn](#) 设置为 0 时, 调用本函数才有意义。

相关信息:

驱动程序版本要求	SynCTI Ver. 4.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetIsdnMsu](#)

2.20.3.3 SsmCheckIsdnMsu

查询驱动程序的内部缓冲区中是否有 ISDN 消息。

函数原型:

```
int SsmCheckIsdnMsu(int nPcmId)
```

参数说明:

nPcmId	数字中继线的逻辑编号
--------	------------

返回值:

-1	错误
≥ 0	ISDN 消息的条数

功能描述:

查询驱动程序的内部缓冲区中是否有 ISDN 消息。

注意事项:

- 当配置项 [AutoHandleIsdn](#) 设置为 1 时, 由驱动程序处理消息; 当配置项 [AutoHandleIsdn](#) 设置为 0 时, 由应用程序处理消息。
- 在实际应用中, 只有当配置项 [AutoHandleIsdn](#) 设置为 0 时, 调用本函数才有意义。

相关信息:

驱动程序版本要求	SynCTI Ver. 4.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetIsdnMsu](#)

2.20.3.4 SsmISDNGetStatus

获取 ISDN 协议 HDLC 链路的状态。

函数原型:

```
int SsmISDNGetStatus(int nPcmNo, int *pL3Start, int *pL2DStatus, int *pL2D_L3Atom, int * pL3_L2DAtom, int *pRef_ind)
```

参数说明:

nPcmNo	本地数字中继线的逻辑编号
pL3Start	返回 ISDN 协议第 3 层 (L3) 是否启动的信息。返回值为 1 表示启动, 0 表示未启动
	返回 ISDN 协议第 2 层 (L2) 的工作状态。工作状态总共有 8 种, 分别如下:
	State 1 TEI 未指定
	State 2 分配等待 TEI
	State 3 建立等待 TEI
pL2Dstatus	State 4 TEI 已指定
	State 5 等待建立
	State 6 等待恢复
	State 7 建立多帧
	State 8 定时器恢复
	(详细参见 Q921)
pL2D_L3Atom	返回接收到的 ISDN 协议第 2 层 (L2) 到 ISDN 协议第 3 层 (L3) 的消息原语(参见 Q921)
pL3_L2DAtom	返回已发送的 ISDN 协议第 3 层 (L3) 到 ISDN 协议第 2 层 (L2) 的消息原语(参见 Q921)
pRef_ind	返回 ISDN 协议内部标示值 (固定值为 0xfe)

返回值:

0	调用成功
-1	调用失败

功能描述:

获取 ISDN 协议 HDLC 链路的状态。

注意事项:

- 本函数通常在 [E_CHG_ISDNStatus](#) 事件的响应代码中调用。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关事件: [E_CHG_ISDNStatus](#)

2.20.3.5 SsmGetUserInfo

获取 User-User 消息单元的内容。

函数原型:

```
int WINAPI SsmGetUserInfo( int ch, PUCHAR pUUID)
```

参数说明:

ch	通道编号
pUUID	存放 User-User 消息单元内容的缓冲区指针, 由应用程序分配存储空间, 不能少于 131 个字节

返回值:

≥0	获取成功, 返回 User-User 消息单元内容的长度 (字节数)
-1	调用失败

功能描述:

获取 User-User 消息单元内容。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver. 5.1.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetUserInfo](#)

2.20.3.6 SsmSetUserInfo

设置 SETUP 消息 User-User 消息单元的内容。

函数原型:

```
int WINAPI SsmSetUserInfo(int ch, PUCHAR pUUID, WORD wLen)
```

参数说明:

ch	通道编号
pUUID	存放 User-User 消息单元内容的缓冲区指针 (消息类型和长度字段除外)
wLen	User-User 消息单元内容的长度

返回值:

0	设置成功
-1	调用失败

功能描述:

设置 SETUP 消息 User-User 消息单元的内容。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver. 5.3.1.4 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetUserInfo](#)

2.20.3.7 SsmISDNGetProgressMsg

获取 PROGRESS 消息的内容。

函数原型:

```
int WINAPI SsmISDNGetProgressMsg(int ch, BYTE* pbMsg)
```

参数说明:

ch	通道编号
pbMsg	存放 PROGRESS 消息的缓冲区指针，由应用程序分配存储空间，不能少于 300 个字节

返回值：

≥0	获取成功，返回 PROGRESS 消息的长度（字节数）
-1	调用失败

功能描述：

获取 PROGRESS 消息内容。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 5.2.0.1 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：无

2.20.3.8 SsmGetIsdnMsuEx

从驱动程序的内部缓冲区中取出一条指定的 ISDN 消息。

函数原型：

```
int WINAPI SsmGetIsdnMsuEx(int nPcmId,int nMsgType, PUCHAR pucMsuBuf, WORD wMsgLen)
```

参数说明：

nPcmId	数字中继线的逻辑编号
nMsgType	指定消息类型，目前只支持 0x20
pucMsuBuf	返回 ISDN 消息。pucMsuBuf 由应用程序分配存储空间，大小不能低于 256 字节
wMsgLen	返回消息长度，wMsgLen 由应用程序分配存储空间，大小不能低于 1 字节

返回值：

nCh	返回消息的通道值
-1	调用失败或者没有消息

功能描述：

从驱动程序的内部缓冲区中取出一条指定的 ISDN 消息。

注意事项：

- 当配置项 AutoHandleIsdn 设置为 1 时，由驱动程序处理消息，但是 0x20 消息能使用本接口获取；
- 在实际应用中，只有当配置项 AutoHandleIsdn 设置为 1 时，调用本函数才有意义；当配置项 AutoHandleIsdn 设置为 0 时，调用 SsmGetIsdnMsu 来获取消息。

相关信息：

驱动程序版本要求	SynCTI Ver. 5.4.2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmCheckIsdnMsu](#), [SsmGetIsdnMsu](#)

2.20.3.9 SsmSendIsdnMsuEx

AutoHandleIsdn 为 1 时发送一条 ISDN 消息。

函数原型：

```
int SsmSendIsdnMsuEx(int ch,int nMsgType, int nMsgLen, PUCHAR pucMsuBuf)
```

参数说明:

ch	通道编号
nMsgType	指定消息类型，保留目前
nMsgLen	pucMsuBuf 中消息的实际长度，单位为字节
pucMsuBuf	ISDN 消息缓冲区，此消息缓冲区存放的应该是消息类型之后的内容

返回值:

0	调用成功
-1	调用失败

功能描述:

在指定的数字中继线的 16 时隙上发送一条 ISDN 消息。

注意事项:

- 当配置项 AutoHandleIsdn 设置为 1 时，由驱动程序处理消息，但调用本接口可以发送消息；
- 在实际应用中，只有当配置项 AutoHandleIsdn 设置为 1 时，调用本函数才有意义；当配置项 AutoHandleIsdn 设置为 0 时，调用 SsmSendIsdnMsu 来发送消息。
- 此函数发送 ISDN 消息，消息的前 4 个字节协议鉴别语和呼叫参考驱动会自己构建，不需要应用程序传递，应用程序消息缓冲区只需要存放消息类型之后的消息。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.4.2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetIsdnMsuEx](#)

2.20.4 有关中国 1 号信令的高级函数

2.20.4.1 控制 ABCD 信令比特

2.20.4.1.1 SsmGetCAS

获取通道上 ABCD 信令的接收端状态。

函数原型:

```
int SsmGetCAS(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
	返回值的高 4 位比特为 0，最低 4 个比特为本端收到的 CAS 信令比特：
≥0	bit3 bit2 bit1 bit0 A B C D

功能描述:

获取对端交换机发送的 ABCD 信令的当前值。

注意事项:

- ABCD 信令在 E1 的 16 时隙中传递;
- 本函数实现的功能也可以通过 [E_RCV_CAS](#) 事件获取，推荐使用 [E_RCV_CAS](#) 事件方式。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmSendCAS](#)

2.20.4.1.2 SsmSendCAS

设置通道上 ABCD 信令的发送端状态。

函数原型：

```
int SsmSendCAS(int ch, BYTE btCas)
```

参数说明：

ch	通道编号
btCas	发送 CAS 信令值，高 4 位比特为 0，低 4 位比特为 ABCD 信令比特的新状态： bit3 bit2 bit1 bit0 A B C D

返回值：

-1	调用失败
0	调用成功

功能描述：

设置本端 ABCD 信令的状态。

注意事项：

- ABCD 信令在 E1 的 16 时隙中传递；
- 如果通道配置为自动接续，调用本函数时将得到返回值-1。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmGetSendingCAS](#), [SsmGetCAS](#)

2.20.4.1.3 SsmGetSendingCAS

获取通道上 ABCD 信令的发送端状态。

函数原型：

```
int SsmGetSendingCAS(int ch)
```

参数说明：

ch	通道编号
----	------

返回值：

-1	调用失败
≥ 0	返回值的最低 4 个比特为发送端的 ABCD 信令比特: bit3 bit2 bit1 bit0 A B C D 其余比特保留

功能描述:

获取通道上 ABCD 信令的发送端状态。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetCAS](#), [SsmSendCAS](#)

2.20.4.1.4 SsmSetSendCASFlag

设置通道是否允许发送 CAS 信号。

函数原型:

```
int WINAPI SsmSetSendCASFlag(int ch, int nCASFlag)
```

参数说明:

ch	通道编号
	是否允许发送 CAS 信号
nCASFlag	0: 不允许 1: 允许

返回值:

-1	调用失败
=0	调用成功

功能描述:

设置通道是否允许发送 CAS 信号。

注意事项: 无**相关信息:**

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetSendCASFlag](#), [SsmSendCAS](#)

2.20.4.1.5 SsmGetSendCASFlag

获取通道是否允许发送 CAS 信号的标志。

函数原型:

```
int WINAPI SsmGetSendCASFlag(int ch, int* pCASFlag)
```

参数说明:

ch	通道编号
pCASFlag	得到是否允许发送 CAS 信号标志的指针

返回值:

-1	调用失败
=0	调用成功

功能描述:

获取通道是否允许发送 CAS 信号的标志。

注意事项: 无

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetCAS](#), [SsmSetSendCASFlag](#)

2.20.4.2 控制 R2 信号收发器

2.20.4.2.1 SsmSetRxR2Mode

设置 R2 信号接收器。

函数原型:

```
int SsmSetRxR2Mode(int ch, int nMode, BOOL bEnable)
```

参数说明:

ch	通道编号
nMode	R2 信号接收器的工作模式: =0: 接收后向 R2 信号 =1: 接收前向 R2 信号
bEnable	R2 信号接收器的工作状态: =TRUE: 开启 =FALSE: 关闭

返回值:

-1	调用失败
0	调用成功

功能描述:

设置 R2 信号接收器。

注意事项:

- 如果通道配置为自动接续，调用本函数时将得到返回值-1；
- R2 信号收发器和 DTMF 检测器不能同时工作，因此，使用 R2 信号收发器时，必须关闭 DTMF 检测器。MFC 过程结束后，需要自行调用 [SsmEnableRxDtmf](#) 启动 DTMF 检测器。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetR2](#)

2.20.4.2.2 SsmGetR2

获取 R2 信号接收器的检测结果。

函数原型:

```
int SsmGetR2(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
0	线路上没有检测到 R2 信号
1~15	线路上检测到了 R2 信号。对于前向 R2 信号，返回值范围为 1~15；对于后向 R2 信号，返回值范围为 1~6

功能描述:

获取 R2 信号接收器的检测结果。

注意事项:

- 如果通道配置为自动接续，调用本函数时将得到返回值-1。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetRxR2Mode](#), [SsmSendR2](#), [SsmSendR2Ex](#)

2.20.4.2.3 SsmSendR2

在通道上产生一个 R2 信号。

函数原型:

```
int SsmSendR2(int ch, int nMode, BYTE btR2)
```

参数说明:

ch	通道编号
nMode	R2 信号发生器的工作模式: =0: 产生后向 R2 信号

	=1: 产生前向 R2 信号
	R2 信号的编码值:
btR2	<ul style="list-style-type: none">● 对于前向 R2 信号, 取值范围为: 1~15● 对于后向 R2 信号, 取值范围为: 1~6

返回值:

-1	调用失败
0	调用成功

功能描述:

在通道上产生一个 R2 信号, 并一直持续到应用程序调用 [SsmStopSendR2](#) 函数后才会停止。

注意事项:

- 如果通道配置为自动接续, 调用本函数时将得到返回值-1。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmStopSendR2](#), [SsmGetSendingR2](#), [SsmGetR2](#), [SsmSendR2Ex](#)

2.20.4.2.4 SsmSendR2Ex

在通道上产生一个指定时间长度的 R2 信号。

函数原型:

```
int SsmSendR2Ex(int ch, int nMode, BYTE btR2, DWORD dwKeepTime)
```

参数说明:

ch	通道编号
nMode	R2 信号发生器的工作模式: =0: 产生后向 R2 信号 =1: 产生前向 R2 信号
btR2	R2 信号的编码值: <ul style="list-style-type: none">● 对于前向 R2 信号, 取值范围为: 1~15● 对于后向 R2 信号, 取值范围为: 1~6
dwKeepTime	R2 信号保持时间, 单位为毫秒

返回值:

-1	调用失败
0	调用成功

功能描述:

在通道上产生一个持续时间为 dwKeepTime 毫秒的 R2 信号, 应用程序也可以调用 [SsmStopSendR2](#) 函数停止 R2 信号。

注意事项:

- 如果通道配置为自动接续, 调用本函数时将得到返回值-1。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmStopSendR2](#), [SsmGetSendingR2](#), [SsmGetR2](#), [SsmSendR2](#)

2.20.4.2.5 SsmStopSendR2

停止在通道上产生一个 R2 信号。

函数原型:

```
int SsmStopSendR2(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
0	调用成功

功能描述:

停止在通道上产生一个 R2 信号。

注意事项:

- 如果通道配置为自动接续, 调用本函数时将得到返回值-1。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSendR2](#), [SsmSendR2Ex](#)

2.20.4.2.6 SsmGetSendingR2

获取通道上 R2 信号发生器的工作模式。

函数原型:

```
int SsmGetSendingR2(int ch, int* pnMode, BYTE* pbtR2)
```

参数说明:

ch	通道编号
	返回 R2 信号发生器的工作模式:
pnMode	=0: 产生后向 R2 信号 =1: 产生前向 R2 信号
pbtR2	对于前向 R2 信号, 取值范围为: 1~15 对于后向 R2 信号, 取值范围为: 1~6

返回值:

-1	调用失败
0	调用成功

功能描述:

获取通道上 R2 信号发生器的工作模式。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSendR2](#), [SsmSendR2Ex](#)

2.21 SHV 系列板卡的专用函数（CTI 系列）

2.21.1 SsmGetMaxVCh

获取应用系统内变声资源通道的总数。

函数原型:

```
int SsmGetMaxVCh()
```

参数说明: 无

返回值:

-1	调用失败
≥ 0	变声资源通道的总数

功能描述:

获取应用系统内变声资源通道的总数。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 4.7.2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetMaxFreeVCh](#)

2.21.2 SsmGetMaxFreeVCh

获取空闲的变声资源通道的总数。

函数原型:

```
int SsmGetMaxFreeVCh()
```

参数说明: 无

返回值:

-1	调用失败
≥ 0	空闲的变声资源通道的总数

功能描述:

获取空闲的变声资源通道的总数。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 4.7.2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetMaxVCh](#)

2.21.3 SsmBindVCh

将语音通道绑定到变声资源通道。

函数原型:

```
int SsmBindVCh(int iCh)
```

参数说明: 无

iCh	语音通道编号
-----	--------

返回值:

0	调用成功
-1	调用失败: iCh 越界
-2	调用失败: iCh 不支持总线交换功能
-3	调用失败: 没有空闲的变声资源通道
-4	调用失败: iCh 已经绑定到某个变声资源通道
-5	调用失败: 驱动程序尚未加载

功能描述:

将语音通道绑定到变声资源通道。

驱动程序执行本函数时,会自动分配一个空闲的变声资源通道,并将此变声资源通道绑定到语音通道 iCh 上。iCh 通道的来话信号会首先进入变声资源通道进行变声处理,然后再送到交换总线上。

注意事项:

- 驱动程序加载成功后,应用程序可以在任何时候调用本函数。

相关信息:

驱动程序版本要求	SynCTI Ver. 4.7.2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmUnBindVCh](#), [SsmSetVoiceEffect](#)

代码范例:

例 1: 通道 1 与通道 2 进行双向连接, 通道 1 的声音需要变声, 通道 2 的声音不需要变声, 实现代码如下:

```
..... //发起呼叫的代码
SsmBindVCh(1); //将通道 1 绑定到一个变声资源通道
SsmSetVoiceEffect(1,180); //设置变声效果
SsmTalkWith(1,2); //建立通道 1 与通道 2 的双向连接
..... //通话
SsmUnBindVCh(1); //通话结束, 解除通道 1 与通道 2 的双向连接
SsmStopTalkWith(1,2); //拆除双向连接
```

例 2: 通道 1、通道 2 和通道 3 加入一个会议室, 通道 1、通道 2 要求变声功能。实现代码如下:

```
int GroupID = SsmCreateConfGroup(...); //创建一个会议室
.....
SsmBindVCh(1);
```

```
SsmJoinConfGroup(GroupID,1,...);      //通道 1 加入会议室
SsmBindVCh(2);
SsmJoinConfGroup(GroupID,2,...);      //通道 2 加入会议室
SsmJoinConfGroup(GroupID,3,...);      //通道 3 加入会议室
....
```

2.21.4 SsmUnBindVCh

解除语音通道与变声资源通道的绑定。

函数原型:

```
int SsmUnBindVCh(int iCh)
```

参数说明:

iCh	语音通道编号
-----	--------

返回值:

0	调用成功
-1	调用失败: iCh 越界
-2	调用失败: iCh 不支持总线交换功能
-3	调用失败: iCh 没有绑定到某个变声资源通道
-4	调用失败: 驱动程序尚未加载

功能描述:

解除语音通道与变声资源通道的绑定。变声资源通道解除绑定后，会由驱动程序自动回收，进入空闲队列。

注意事项:

- 如果语音通道不再需要变声功能，应解除与变声资源通道的绑定。

相关信息:

驱动程序版本要求	SynCTI Ver. 4.7.2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmBindVCh](#)

2.21.5 SsmSetVoiceEffect

设置变声的效果。

函数原型:

```
int SsmSetVoiceEffect(int iCh, int iValue)
```

参数说明:

iCh	语音通道编号
iValue	变声的效果，取值范围: 70~220, 128 表示原始声音（即不变声），大于 128 为女声效果，小于 128 为男声效果

返回值:

0	调用成功
-1	调用失败: iCh 越界
-2	调用失败: iCh 没有绑定到某个变声资源通道
-3	调用失败: iValue 越界
-4	调用失败: 驱动程序尚未加载

功能描述:

设置变声的效果。

注意事项:

- 本函数必须在语音通道已经绑定到某个变声资源通道后才能调用；
- 变声的效果难以具体描述，请根据实际测试的效果对 iValue 进行设置。

相关信息:

驱动程序版本要求	SynCTI Ver. 4.7.2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmBindVCh](#), [SsmGetVoiceEffect](#)

2.21.6 SsmGetVoiceEffect

获取变声的效果。

函数原型:

```
int SsmGetVoiceEffect(int iCh)
```

参数说明:

iCh	语音通道编号
-----	--------

返回值:

>0	调用成功，返回变声的效果，具体取值请参见函数 SsmSetVoiceEffect 的说明
-1	调用失败：iCh 越界
-2	调用失败：iCh 没有绑定到某个变声资源通道
-3	调用失败：驱动程序尚未加载

功能描述:

获取变声的效果。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 4.7.2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetVoiceEffect](#)

2.21.7 SsmSetVoiceEffectEx

设置 240E VAR 卡的变声参数。

函数原型:

```
int WINAPI SsmSetVoiceEffectEx(int iCh, int VarType, int VarParamA, int VarParamB, int VarParamC)
```

参数说明:

iCh	语音通道编号
VarType	变声类型 (0: 不变声 1: 变音色 (包括年龄和性别) 2: 变沙哑)
VarParamA	变声参数 A (音色变化时: 取值 20-180, =100 为不变换; 沙哑化时: 取值 8-800, 取值越大变化越明显)
VarParamB	变声参数 B (音色变化时: 取值 20-180, =100 为不变换; 沙哑化时 (代表音量): 取值 1-300, 取值越大音量越大)
VarParamC	变声参数 C (音色变化时: 取值 20-180, =100 为不变换; 沙哑化时: 不起作用)

返回值:

0	调用成功
-1	调用失败: iCh 越界
-2	调用失败: iCh 没有绑定到某个变声资源通道
-3	调用失败: 变声参数越界
-4	调用失败: 驱动程序尚未加载

功能描述:

设置 240E VAR 卡变声的参数。

注意事项:

- 本函数必须在语音通道已经绑定到某个变声资源通道后才能调用；
- 变声的效果难以具体描述，请根据实际测试的效果对各变声参数进行设置，以下参数可供参考：
 - 年轻男变年轻女: VarParamA= 66; VarParamB = 100; VarParamC= 82;
 - 年轻男变老年女: VarParamA= 72; VarParamB = 100; VarParamC= 88;
 - 年轻男变老年男: VarParamA= 140; VarParamB = 100; VarParamC= 118;
 - 年轻男变小孩: VarParamA= 66; VarParamB = 100; VarParamC= 66;
 - 年轻女变年轻男: VarParamA= 140; VarParamB = 100; VarParamC= 118;
 - 年轻女变老年女: VarParamA= 120; VarParamB = 100; VarParamC= 108;
 - 年轻女变老年男: VarParamA= 180; VarParamB = 100; VarParamC= 136;
 - 年轻女变小孩: VarParamA= 88; VarParamB = 100; VarParamC= 88;
 - 年轻男变沙哑: VarParamA= 405; VarParamB = 30;
 - 年轻女变沙哑: VarParamA= 600; VarParamB = 10。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.2.3 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmBindVCh](#), [SsmGetVoiceEffectEx](#)

2.21.8 SsmGetVoiceEffectEx

获取 240E VAR 卡的变声参数。

函数原型:

```
int WINAPI SsmGetVoiceEffectEx(int iCh, int* pVarType, int* pVarParamA, int* pVarParamB, int* pVarParamC)
```

参数说明:

iCh	语音通道编号
pVarType	用于存放变声类型的 int 变量的指针
pVarParamA	用于存放变声参数 A 的 int 变量的指针
pVarParamB	用于存放变声参数 B 的 int 变量的指针
pVarParamC	用于存放变声参数 C 的 int 变量的指针

返回值:

>0	调用成功, 返回变声的效果, 具体取值请参见函数 SsmSetVoiceEffectEx 的说明
-1	调用失败: iCh 越界或某参数为空指针
-2	调用失败: iCh 没有绑定到某个变声资源通道
-3	调用失败: 驱动程序尚未加载

功能描述:

获取 240E VAR 变声的参数。

注意事项:
相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.2.3 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetVoiceEffectEx](#)

2.22 SHN 系列板卡的专用函数 (CTI 系列)

2.22.1 SIP 通道的专用编程接口

2.22.1.1 SsmlpGetSessionCodecType

取得指定通道当前通话中实际使用的 Codec 类型。

函数原型:

```
int SsmlpGetSessionCodecType(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
6	指定通道当前正在使用 A-Law 格式进行通话
7	指定通道当前正在使用 μ-Law 格式进行通话
49	指定通道当前正在使用 GSM 格式进行通话
131	指定通道当前正在使用 G.729A 格式进行通话
4	指定通道当前正在使用 G.723_1 格式进行通话
9	指定通道当前正在使用 G.722 格式进行通话
96	指定通道当前正在使用 AMR 格式进行通话
98	指定通道当前正在使用 ILBC 格式进行通话

功能描述:

取得指定通道当前通话中实际使用的 Codec 类型。

注意事项:

- 只有在通道处于通话状态时调用本函数才有意义。
- 在 5.1.0.0 以前版本该函数名为 SsmlpGetUsingCodecType。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.3.0 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.1.2 SsmlpSetForwardNum

设置指定通道的转寄号码。

函数原型:

```
int SsmlpSetForwardNum(int ch, LPSTR pszForwardNum)
```

参数说明:

ch	通道编号
pszForwardNum	转寄号码, pszForwardNum 为空字符串, 表示取消设置

返回值:

-1	调用失败
0	调用成功

功能描述:

设置指定通道的转寄号码。设置以后, 对该通道的任何呼入将无条件地转寄到设置的号码。

注意事项:

- 如果以前对某指定通道有设置转寄号码的, 再次调用本函数并将 pszForwardNum 置为空字符串, 表示取消设置。
- 使用 IP 卡转寄时, 禁止转寄号码和被叫号码相同。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.0.0.0 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.1.3 SsmplInitiateTransfer

设置指定通道的转移号码。

函数原型:

```
int SsmplInitiateTransfer(int ch, LPSTR pszTransferTo)
```

参数说明:

ch	通道编号
pszTransferTo	转移号码

返回值:

-1	调用失败
0	调用成功

功能描述:

设置指定通道的转移号码。对通话中的通道调用该函数, 将该次呼叫转移到设置的号码上。

注意事项:

- 只有在通道处于通话状态时调用本函数才有意义。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.0.0.0 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.1.4 SsmplGetMessageField

取得指定通道上用于建立呼叫的相关消息中的一系列字段值。

函数原型:

```
int SsmIpGetMessageField(int ch, int type, LPSTR szBuffer, int *pSize)
```

参数说明:

ch	通道编号
Type	需要取得的字段类型
SzBuffer	用于保存取得的字段内容的缓冲区
PSize	该指针指向的值为用户提供的缓冲区大小。当本函数返回后，如果成功，该指针指向的值为取得的有效数据大小；如果失败，该指针指向的值为实际需要的缓冲区大小

返回值:

-1	调用失败
0	调用成功

功能描述:

取得指定通道上用于建立呼叫的相关消息中的一系列字段值。

参数 type 的取值如下:

- TYPE_MESSAGE_CONTACT

取得 INVITE 或者 REINVITE 消息中的 Contact 字段内容。该参数仅在作为被叫方使用时有效。如果消息体中存在多个 Contact 字段，那么返回的格式如下：

Contact0	\0	Contact1	\0	...	Contactn	\0	\0
----------	----	----------	----	-----	----------	----	----

- TYPE_MESSAGE_SDP

取得远端 INVITE 或者 200OK 消息中的 SDP 字段内容。该参数作为被叫方使用和主叫方使用时都有效：当作为被叫方使用时，取得的是远端 INVITE 消息中的 SDP 字段内容；当作为主叫方使用时，取得的是远端 200OK 消息中的 SDP 字段内容。

注意事项:

- 取 Contact 字段内容：只有在通道处于振铃、通话或挂起状态时调用本函数才有意义。
- 取 SDP 字段内容：当板卡作为被叫方使用时，只有在通道处于振铃、通话或挂起状态时调用本函数才有意义；当板卡作为主叫方使用时，只有在通道处于通话或挂起状态时调用本函数才有意义。
- 取 SDP 字段功能要求 SynCTI 版本为 Ver. 5.3.1.4 或更高。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.0.0.0 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.1.5 SsmIPGetMsgFieldStr

取得指定通道上用于建立呼叫的相关消息中的一系列字段值。

函数原型:

```
int SsmIPGetMsgFieldStr(int ch, int nMsgType, LPSTR szParaType, LPSTR szBuf, int nBufSize)
```

参数说明:

ch	通道编号	
nMsgType	取值	描述
	0	获取 invite 消息中的字段

	1	获取 480 消息中的字段
	2	获取 183 消息中的字段
	3	获取 181 消息中的字段
szParaType	字段名称	
szBuf	用于保存取得的字段内容的缓冲区	
nBufSize	szBuf 缓冲区大小	

返回值:

-1	调用失败
>=0	调用成功, 表示 szBuf 缓冲区中获取的有效数据的长度

功能描述:

取得指定通道上用于建立呼叫的相关消息中的一系列字段值。

注意事项:

- 若需要获取 invite 消息中的字段, 需要在通道处于振铃或者通话状态下调用该函数;
- 若需要获取 480 消息中的字段, 则需要在通道处于挂起状态时调用该函数;
- 若需要获取 183、181 消息中的字段, 一般需要在通道处于 S_CALL_VOIP_SESSION_PROCEEDING 状态时调用该函数;
- 本函数目前只支持获取 invite、480、183 和 180 消息中的字段。

c 语言范例:

```
char szParaType[128] = "History-Info";
char szBuf[128];
int nBufSize = sizeof(szBuf);
if(SsmGetChState(ch) == S_CALL_VOIP_SESSION_PROCEEDING)
    SsmIPGetMsgFieldStr(ch,3,szParaType,szBuf,nBufSize); //3 表示获取的 181 消息里的字段内容
```

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.3.0 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.1.6 SsmSipMsgSetHeader

增加 IP 通道外呼的时候 INVITE 消息中携带的字段。

函数原型:

```
int WINAPI SsmSipMsgSetHeader(char* h_Name,char* h_Value)
```

参数说明:

h_Name	字段名称
h_Value	字段内容

返回值:

-1	调用失败
0	调用成功

功能描述:

增加 IP 通道外呼的时候 INVITE 消息中携带的字段。

注意事项:

- 本函数仅用于增加非关键性字段。
- 本函数设置的参数仅对本次呼叫有效。
- 配置项 [SipMsgHeaderName](#) 和 [SipMsgHeaderValue](#) 可以实现相同功能。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.2.1 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.1.7 SsmSipMsgSetHeaderA

增加指定的 IP 通道外呼的时候 INVITE 消息中携带的字段。

函数原型:

```
int WINAPI SsmSipMsgSetHeaderA(int nCh, char* h_Name, char* h_Value)
```

参数说明:

nCh	通道号
h_Name	字段名称
h_Value	字段内容

返回值:

-1	调用失败
0	调用成功

功能描述:

增加指定的 IP 通道外呼的时候 INVITE 消息中携带的字段。

注意事项:

- 本函数仅用于增加非关键性字段；
- 可以多次调用此函数，增加多个字段；
- 如果需要清除通道上的所有字段，可以调用此函数，并将 h_Name 和 h_Value 设置成 NULL。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.2.6 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.1.8 SsmSipStackRegister

SynSIP 协议栈向 Sip Server 发起注册。

函数原型:

```
int WINAPI SsmSipStackRegister(LPCSTR szRegSrvAddr, LPCSTR szOutBoundAddr, LPCSTR szDisplayName,
LPCSTR szUserName, LPCSTR szAuthUserName, LPCSTR szPasswd, LPCSTR szRealm, int nExpires)
```

参数说明:

szRegSrvAddr	Sip Sever 的 IP 地址
--------------	-------------------

szOutBoundAddr	Sip Server 的外绑地址
szDisplayName	显示用户名
szUserName	对应板卡注册的用户名
szAuthUserName	认证用户名 (Sip Server 的注册需要认证时才有用)
szPasswd	对应板卡注册的密码 (Sip Server 的注册需要认证时才有用)
szRealm	Sip 服务器别名
nExpires	注册的有效期

返回值:

-1	调用函数失败
>0	注册 ID

功能描述:

SynSIP 协议栈向 Sip Server 发起注册。

注意事项:

- 函数调用成功后返回的注册 ID 需要客户自行维护，后续查询注册信息、将通道和注册信息绑定操作都需要用到注册 ID 参数；
- 如果取消注册 (nExpires=0 的情况)，必须先解除绑定。
- 注册成功与否的结果无法从函数调用后的返回值中获取，需要调用 [SsmSipGetRegInfo](#) 函数来获取。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.3.0 或更高
Header	Shpa3api.h
Library	shp_a3.lib、SynSip.lib
DLL	shp_a3.dll、SynSip.dll

相关函数: [SsmSipStackUnRegister](#)、[SsmSipStackRemoveRegister](#)、[SsmSipGetRegInfo](#)

2.22.1.9 SsmSipBindChWithRegInfo

将通道和注册信息绑定。

函数原型:

```
int WINAPI SsmSipBindChWithRegInfo(int nChID, int nRegID)
```

参数说明:

nChID	通道号
nRegId	注册 ID

返回值:

-1	绑定失败
0	绑定成功

功能描述:

将通道和注册信息绑定。

注意事项:

- 调用此函数时，通道必须处于空闲状态，否则函数调用会失败；
- 如果通道已经绑定了注册信息，函数调用会失败。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.2.6 或更高
Header	Shpa3api.h
Library	shp_a3.lib、SynSip.lib
DLL	shp_a3.dll、SynSip.dll

相关函数: [SipUnBindChWithRegInfo](#)

2.22.1.10 SsmSipUnBindChWithOutRegInfo

取消通道和注册信息的绑定。

函数原型:

```
int SamSipUnBindChWithOutRegInfo(int nChID);
```

参数说明:

nChID	通道号
-------	-----

返回值:

-1	取消绑定失败
0	取消绑定成功

功能描述:

取消通道和注册信息的绑定。

注意事项:

- 调用此函数时，通道必须处于空闲、注册失败或注册中状态，否则函数调用会失败。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.4.2.0 或更高
Header	Shpa3api.h
Library	shp_a3.lib、SynSip.lib
DLL	shp_a3.dll、SynSip.dll

相关函数: [SsmSipBindChWithRegInfo](#)

2.22.1.11 SsmSipStackUnRegister

取消 SynSIP 协议栈到 Sip Server 的注册。

函数原型:

```
int WINAPI SsmSipStackUnRegister(int nRegID)
```

参数说明:

nRegID	注册 ID
--------	-------

返回值:

-1	函数调用失败
0	函数调用成功

功能描述:

取消 SynSIP 协议栈到 Sip Server 的注册。

注意事项:

- 此函数的功能与调用 `SsmSipStackRegister` 函数并设置 `nExpire=0` 功能相同。调用此函数时，需要解除通

道和注册信息的绑定，并且注册状态必须为注册成功。

相关信息：

驱动程序版本要求	SynCTI Ver. 5.3.2.6 或更高
Header	Shpa3api.h
Library	shp_a3.lib、SynSip.lib
DLL	shp_a3.dll、SynSip.dll

相关函数：[SsmSipStackRegister](#)

2.22.1.12 SsmSipStackRemoveRegister

删除 SynSIP 协议栈上的注册对象。

函数原型：

```
int WINAPI SsmSipStackRemoveRegister(int nRegID)
```

参数说明：

nRegID	注册 ID
--------	-------

返回值：

-1	函数调用失败
0	函数调用成功

功能描述：

删除 SynSIP 协议栈上的注册对象。

注意事项：

- 调用此函数时，注册信息必须没有与任何通道绑定，并且注册状态为非“注册成功”状态。

相关信息：

驱动程序版本要求	SynCTI Ver. 5.3.2.6 或更高
Header	Shpa3api.h
Library	shp_a3.lib、SynSip.lib
DLL	shp_a3.dll、SynSip.dll

相关函数：[SsmSipStackRegister](#)

2.22.1.13 SsmSipGetRegInfo

查询注册信息。

函数原型：

```
int WINAPI SsmSipGetRegInfo(int nRegID, RegInfo *pstRegInfo)
```

参数说明：

nRegID	注册 ID
pstRegInfo	注册信息结构体指针

返回值：

-1	函数调用失败或者没有找到对应的注册信息
0	函数调用成功并且获取到了对应的注册信息

功能描述：

查询注册信息。

返回的 RegInfo 结构体如下：

```
typedef struct tag_RegInfo
{
    int nRegID; //注册 ID
    enRegState enRegS; //注册状态 0 注册初始化，1 注册中，2 注册成功，3 注册失败
    int nRetryCount; //注册次数
    BOOL bIsUpdateReg; //是否注册成功
    int nExpires; //注册间隔
    int nOriginalRegID; //原始注册 ID

    int nTotalBindNum; //总的绑定通道数
    int sBindChList[MAX_SIPIP_BINDCHNUM]; //绑定列表
    char szServerAddress[MAX_SIPIP_ADDR_LENGTH]; //注册服务器地址
    char szOutBoundAddr[MAX_SIPIP_ADDR_LENGTH]; //服务器外绑地址
    char szAuthName[MAX_SIPIP_ADDR_LENGTH]; //认证用户名
    char szUserName[MAX_SIPIP_ADDR_LENGTH]; //注册用户名
    char szPassword[MAX_SIPIP_ADDR_LENGTH]; //密码
    char szRealm[MAX_SIPIP_ADDR_LENGTH]; //服务器别名
    char szLocalIdentity[MAX_SIPIP_ADDR_LENGTH]; //本地 SIP From URL
    char szDisplayName[MAX_SIPIP_ADDR_LENGTH]; //显示用户名
}RegInfo;
```

相关信息：

驱动程序版本要求	SynCTI Ver. 5.3.3.0 或更高
Header	Shpa3api.h
Library	shp_a3.lib、SynSip.lib
DLL	shp_a3.dll、SynSip.dll

相关函数：[SsmSipStackRegister](#)

2.22.1.14 SsmSipSetTxUserName

设置去话呼叫时本端的 UserName。

函数原型：

```
int SsmSipSetTxUserName(int ch, LPSTR pszUserName)
```

参数说明：

ch	通道号
pszUserName	pszUserName 设置的是完整 SIP URI 的 UserName 部份，最大长度不能超过 50。一个完整的 SIP URI 格式为：“displayname” <sip:user@host:port>

返回值：

-1	设置失败
0	User Name 的长度为 0，无效
>0	User Name 的长度

功能描述：

在非注册的情况下，设置去话时本端的 UserName。

当在配置文件中 sip 节下增加配置项 EnableSetUsername=1，也可对注册后的 UserName 进行修改。

注意事项：

- 一般情况下该函数只有在非注册的情况下才有效；
- 当增加配置项 EnableSetUsername=1 后，在注册的情况下也有效，此功能在 5313 以上版本支持；
- 该函数只要在呼叫前设置才有效。

相关信息：

驱动程序版本要求	SynCTI Ver. 5.1.0.0 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.1.15 SsmSetIpFlag

设置 IP 卡的一些参数。

函数原型：

```
int SsmSetIpFlag(int ch,int Type,LPSTR pszBuffer)
```

参数说明：

ch	通道号
Type	需要修改的类型
pszBuffer	修改以后的值

返回值：

-1	设置失败
0	pszBuffer 的长度为 0，无效
>0	pszBuffer 的长度

功能描述：

用来设定 IP 卡的一些参数。

注意事项：

- 当 Type = 1，用来设置去话时，Invite 里 from 字段中的 IP 地址，只有在非注册时才有效；
- Type 值等于其他值时，该函数无效。

相关信息：

驱动程序版本要求	SynCTI Ver. 5.1.0.0 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.1.16 SsmSipGetReferStatus

呼叫转移以后，获得第三方呼叫的状态。

函数原型：

```
int SsmSipGetReferStatus(int ch)
```

参数说明：

ch	通道号
返回值:	
-1	获取失败
0	没有进行呼叫转移操作
40	表示通道在不可用状态
1	前转到第三方的状态为 Trying, 即 the subscription is pending
2	180 RingBack 状态
3	200 OK、3XX 状态
4	4XX、5XX、6XX 失败状态

功能描述:

当板卡调用 [SsmplInitiateTransfer](#) 函数转移到第三方时, 取得第三方的呼叫状态。

注意事项:

- 只有在使用 [SsmplInitiateTransfer](#) 进行呼叫转移以后才有用。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.1.0.0 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.1.17 SsmSipSendRequest

请参见 [SsmSipSendRequestA](#)。

2.22.1.18 SsmSipSendRequestA

在呼叫状态下发送 Notify 消息或 Option 消息。

函数原型:

```
int      WINAPI      SsmSIPSendRequest(int      ch,           LPSTR      SipMessageType,      char
SipMessageNewHeaders[SipMessageHeadersNum][SipMessageHeadersLen], int SipMessageNewHeadersNum,
LPSTR SipMessageBody)

int WINAPI SsmSIPSendRequestA(int nCh, LPSTR pSipMessageType, char **pSipMessageNewHeaders, int
nSipMessageNewHeadersNum, LPSTR pSipMessageBody)
```

参数说明:

ch	通道号
nCh	
SipMessageType	消息类型, Notify 或 Option 消息
pSipMessageType	
SipMessageNewHeaders	消息头内容
pSipMessageNewHeaders	
SipMessageNewHeadersNum	消息头数目
nSipMessageNewHeadersNum	
SipMessageBody	消息体
pSipMessageBody	

返回值:

-1	调用失败
0	调用成功

功能描述:

在呼叫状态下发送 Notify 消息或 Option 消息。

注意事项:

- 发送 Notify 消息:

Notify 消息分成两部分，消息头和消息体，消息头在 Synsip 层的存储方式分成两种，一种是常规字段如：from、to、contact 字段等，这类字段一般不需要修改，在发送消息的时候自动添加，如：Content-Type 字段，默认情况下发送 Notify 会带上这个字段，目前本函数支持 Content-Type、to、request-uri 字段的修改；另一类字段如：P-Preferred-Identity、Event 等字段属于特殊字段，统一存放在一个头字段列表中，所以 Synsip 接口是统一的；

- 发送 Option 消息:

Option 消息只有消息头，头字段的插入和修改和 Notify 相同。

相关信息:

驱动程序版本要求	SsmSIPSendRequest 要求 SynCTI Ver. 5.3.2.2 或更高； SsmSIPSendRequestA 要求 SynCTI Ver. 5.3.2.6 或更高。
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.1.19 SsmSetHangupReason

设置 Sip 通道 Ringing 状态下的挂机原因。

函数原型:

```
int WINAPI SsmSetHangupReason(int ch, int nReason)
```

参数说明:

ch	通道号
nReason	4xx、5xx、6xx 类型的 Sip 消息

返回值:

-1	设置失败
0	设置成功

功能描述:

设置 Sip 通道 Ringing 状态下的挂机原因。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver. 5.3.2.2 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.1.20 SsmSipGetBoardRegStatus

查询板卡的注册状态。

函数原型:

```
int SsmSipGetBoardRegStatus(int nBId, LPSTR pszRegFailInfo)
```

参数说明:

nBId	需要查询注册状态的板卡 ID
pszRegFailInfo	注册出错时的返回消息

返回值:

-1	表示注册失败或该板卡不为 IP 卡
≥1	表示注册成功

功能描述:

获得该板卡最后一次发送注册请求时返回的状态，参数 `pszRegFailInfo` 返回注册出错原因，注册成功时无附加信息，若该参数为 `NULL`，则不返回注册出错原因。

注意事项:

- SIP 协议中不存在实时的注册状态查询信令，该函数只能获得最后一次发送注册请求时返回的状态，不具备实时性；
- 想要该函数获得的注册状态的有效性提高，可将注册有效期缩短，如：注册有效期为 300 (S)，则该功能函数获得的状态最坏情况是 5 分钟前的情况。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.1.0.0 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.1.21 SsmSipGetChRegStatus

查询通道的注册状态。

函数原型:

```
int SsmSipGetChRegStatus(int nChId, LPSTR pszRegFailInfo)
```

参数说明:

nChId	需要查询注册状态的通道 ID
pszRegFailInfo	注册出错时的返回消息

返回值:

-1	表示注册失败或该通道不为 IP 通道
≥1	表示注册成功

功能描述:

获得该通道最后一次发送注册请求时返回的状态，参数 `pszRegFailInfo` 返回注册出错原因，注册成功时无附加信息，若该参数为 `NULL`，则不返回注册出错原因。

注意事项:

- SIP 协议中不存在实时的注册状态查询信令，该函数只能获得最后一次发送注册请求时返回的状态，不具备实时性；

备实时性；

- 想要该函数获得的注册状态的有效性提高，可将注册有效期缩短，如：注册有效期为 300 (S)，则该功能函数获得的状态最坏情况是 5 分钟前的情况。

相关信息：

驱动程序版本要求	SynCTI Ver. 5.1.0.0 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.1.22 SsmSetRcvRegisterCallBack

设置收到注册消息后的回调函数。

函数原型：

```
int WINAPI SsmSetRcvRegisterCallBack(RCVREGISTER RcvRegisterCallBack)
```

参数说明：

RcvRegisterCallBack	设置收到注册消息以后的回调函数。 函数原型： int WINAPI RcvRegisterCallBack(char* SipBuf, int SipSize, char* SipBufResponse); 其中， SipBuf: 原始数据； SipSize: 数据长度； SipBufResponse: 收到注册消息以后的回复内容； 返回值：收到注册消息以后的回复内容长度。
---------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

返回值：

-1	调用失败
0	调用成功

功能描述：

设置收到注册消息后的回调函数。

参数 RcvRegisterCallBack 用来获取注册消息的原始数据，并返回注册消息响应结果，驱动会将 SipBufResponse 的内容发送给注册发起方。

注意事项：

- 如果回调函数为空，协议栈直接返回，并打印错误日志；

相关信息：

驱动程序版本要求	SynCTI Ver. 5.3.2.1 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.1.23 SsmlpGetBoardMacAddress

获取 IP 板卡的 MAC 地址。

函数原型：

```
int SsmlpGetBoardMacAddress(int nBld, PUCHAR pucMacAddrBuff)
```

参数说明：

nBld	需要获取 MAC 地址的 IP 板卡 ID
pucMacAddrBuff	用于保存获取的 MAC 地址的缓冲区

返回值:

-1	表示获取失败或该板卡不为 B 型 IP 卡
6	表示获取的 MAC 地址长度

功能描述:

获取 B 型 IP 板卡的 MAC 地址。

参数 pucMacAddrBuff 用于保存获取的 MAC 地址。获取成功时，函数返回 MAC 地址的长度，即 6 个 unsigned char 类型的字节；获取失败时，函数返回 -1。

注意事项:

- IP 卡必须为 B 型卡以上，A 型卡不能获取 MAC 地址；
- 调用该接口必须在板卡启动起来以后才有效。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.1.3 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.1.24 SsmSipRegister

SIP 通道向 Sip Server 发起注册。

函数原型:

```
int SsmSipRegister(int nRegMode, int nParam1, int nParam2, LPCSTR szDisplayName, LPCSTR szUserName,
LPCSTR szPasswd, LPCSTR szAuthUserName, LPCSTR szRegSrvAddr, LPCSTR szRealm, LPCSTR
szOutboundProxy, int nExpires)
```

参数说明:

nRegMode	注册模式。取值范围：nRegMode=0, nRegMode=3
nParam1	起始通道号
nParam2	结束通道号。
szDisplayName	显示名称
szUserName	注册用户名
szPasswd	注册密码
szAuthUserName	认证用户名（SIP Server 的注册需要认证时才有用）
szRegSrvAddr	注册服务器地址（代理服务器地址）
szRealm	注册服务器别名
szOutboundProxy	注册外绑地址（某些注册服务器不能直接访问，会指定外绑服务器地址）
nExpires	注册有效期。默认值为 3600s，到达有效期后会自动重新发起注册

返回值:

-1	调用失败
0	调用成功

功能描述:

SIP 通道向 Sip Server 发起注册。

注意事项:

- 函数调用失败表示注册参数或通道状态不符合要求, **SynSipStack** 不发送注册消息。函数调用成功与否不代表注册是否成功;
- 如果有通道处于非 **IDLE**、**UNUSABLE** 和注册失败状态, 调用本函数将返回-1;
- 如果对应通道已经注册, 则只能发送更新注册的消息(只能更新 **Expires**), 即:
 - (1) **Expires** 等于 0, 表示取消已经存在的注册, 所有参数(除 **Expires** 外)必须和注册成功时的信息保持一致。
 - (2) **Expires** 不等于 0, 表示再次发送注册消息, 所有参数(除 **Expires** 外)必须和注册成功时的信息保持一致。
 - (3) 取消注册后, **Identity** 更新为 **sip:Username@IPAdress**, 还可以使用 IP 地址进行呼叫。
- 当 **nRegMode=3** 时, 函数不发起注册, 只增加认证信息;
- 当 **nRegMode=4** 时, 函数不发起注册, 只删除认证信息。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.2.6 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.1.25 SsmSipSetConnectionInforOfSDP

设置 SIP 信令 **Invite** 消息和 200 消息中 SDP 消息体的 C 字段内容中的 IP 地址(端口号由驱动自动分配, 无需设置), 告知对方设备向该指定地址发送 RTP 数据, 一般为局域网网关的外网地址。

函数原型:

```
int WINAPI SsmSipSetConnectionInforOfSDP(char * pConnectionInfo)
```

参数说明:

pConnectionInfo	C 字段信息中的 IP 地址
-----------------	----------------

返回值:

-1	调用失败。
0	调用成功。

功能描述:

设置 SIP 信令 **Invite** 消息和 200 消息中 SDP 消息体的 C 字段内容中的 IP 地址(端口号由驱动自动分配, 无需设置), 告知对方设备向该指定地址发送 RTP 数据, 一般为局域网网关的外网地址。

注意事项:

- 配置项 [MapIP](#) 可以实现相同功能;
- 当 **pConnectionInfo=NULL** 的时候, 本函数不起作用, 即驱动程序自动填写 C 字段中的 IP 地址。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.2.3 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.1.26 SsmSipSetMsgFieldParameter

设置 SIP 信令消息中某个字段的内容。

函数原型:

```
int WINAPI SsmSipSetMsgFieldParameter(int nCh, int nMsgType, DWORD dwParam, LPCSTR h_Name,  
LPCSTR h_Value)
```

参数说明:

nCh	通道号
nMsgType	消息类型, 0 表示 REFER 消息, 1 表示 INVITE
dwPara	参数保留, 默认设置为 0
h_Name	字段名称
h_Value	字段内容

返回值:

-1	调用失败。
0	调用成功。

功能描述:

设置 SIP 信令消息中某个字段的内容。

注意事项:

- 本函数目前只支持设置 Refer 和 INVITE 消息中 Request 字段、From 字段和 To 字段的 url;
- 每个通道最多可指定 10 项由本函数发起的 SIP 信令消息修改。

相关信息:

驱动程序版本要求	Refer 消息需要 SynCTI Ver. 5.3.2.5 或更高, INVITE 消息需要 SynCTI Ver. 5.4.0.0 或更 高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.1.27 SsmSipChHold

SIP 通道向对端发起或者取消呼叫保持。

函数原型:

```
int WINAPI SsmSipChHold(int ch, BOOL bHold)
```

参数说明:

ch	通道号
bHold	TRUE: 发起呼叫保持 FALSE: 取消呼叫保持

返回值:

-1	调用失败
0	调用成功

功能描述:

SIP 通道向对端发起或者取消呼叫保持。

注意事项:

- 函数只能在通道处于通话状态时使用。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.2.7 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.1.28 SsmSipChTransfer

SIP 通道向对端发起通话中呼叫转移。

函数原型:

```
int WINAPI SsmSipChTransfer(int ch, int nReferToCh)
```

参数说明:

ch	通道号
nReferToCh	呼叫转移目标对应的本端通道号

返回值:

-1	调用失败
0	调用成功

功能描述:

SIP 通道向对端发起通话中呼叫转移。假设对端设备 A 与参数 ch 指定的通道在通话，参数 nReferToCh 指定的通道和设备 C 通话，使用本函数可以实现将呼叫由 A 转移到 C，最终实现 A, C 之间通话。

注意事项:

- 函数只能在 ch 通道处于呼叫保持状态和 nReferToCh 通道处于通话状态时使用；
- 在 nReferToCh 通道与设备 C 通话中，只有 nReferToCh 通道为主叫方时，呼转才能成功；
- 只有 ch 通道的通话处于呼叫保持状态时，呼转才能成功，因此调用该函数前需要先成功调用函数 [SsmSipChHold](#)。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.2.7 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.1.29 SsmSipSetConnectionInforOfSDPEx

设置 SIP 信令 `Invite` 消息和 `200` 消息中 SDP 消息体的 `C` 字段内容中的 IP 地址（端口号由驱动自动分配，无需设置），告知对方设备向该指定地址发送 RTP 数据，一般为局域网网关的外网地址。

函数原型:

```
int WINAPI SsmSipSetConnectionInforOfSDPEx(int nCh, char * pConnectionInfo)
```

参数说明:

nCh	通道号
pConnectionInfo	C 字段信息中的 IP 地址

返回值:

-1	调用失败。
0	调用成功。

功能描述:

设置 SIP 信令 Invite 消息和 200 消息中 SDP 消息体的 C 字段内容中的 IP 地址（端口号由驱动自动分配，无需设置），告知对方设备向该指定地址发送 RTP 数据，一般为局域网网关的外网地址。

注意事项:

- 当 pConnectionInfo=NULL 的时候，本函数不起作用，即驱动程序自动填写 C 字段中的 IP 地址。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.3.0 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.1.30 SsmSipSetContactSection

指定通道发送 invite、180 和 200 消息中 Contact 字段的 host 和 port。

函数原型:

```
int WINAPI SsmSipSetContactSection(int nCh, char *szContactSection)
```

参数说明:

ch	通道号
szContactSection	IP 地址和端口，格式为：ipAddr:port

返回值:

-1	调用失败
0	调用成功

功能描述:

指定通道发送 invite、180 和 200 消息中 Contact 字段的 host 和 port。

注意事项:

- 如果 szContactSection 为 NULL，那么消息中 Contact 字段的 host 和 port 为板卡信令 IP 地址和端口。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.3.0 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.1.31 SsmSipChEnableRtpStun

设置从 nChFrom 到 nChTo 的 SIP 通道是否开启 RTP 穿透。

函数原型:

```
int WINAPI SsmSipChEnableRtpStun(int nChFrom, int nChTo, BOOL bRtpStun)
```

参数说明:

nChFrom	起始通道号
nChTo	结束通道号
bRtpStun	是否开启 RTP 穿透

返回值:

-1	调用失败
0	调用成功

功能描述:

设置从 nChFrom 到 nChTo 的 SIP 通道是否开启 RTP 穿透。

注意事项: 无

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.3.0 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.1.32 SsmSipOutCallSendOptions

SynSIP 协议栈向 Sip Server 发送 Option 消息。

函数原型:

```
int WINAPI SsmSipOutCallSendOptions(char *pszSipServerIP)
```

参数说明:

pszSipServerIP	Sip Sever 地址, 包含端口
----------------	--------------------

返回值:

-1	调用失败
0	调用成功

功能描述:

SynSIP 协议栈向 Sip Server 发送 Option 消息。

注意事项: 无

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.3.2 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.1.33 SsmSipChEnableSrtp

SIP 通道 Srtp 加密。

函数原型:

```
int WINAPI SsmSipChEnableSrtp (intnChFrom, intnChTo, BOOL bSrtp)
```

参数说明:

nChFrom	起始通道号
nChTo	结束通道号
bSrtp	Srtp 加密, TRUE: 打开 FALSE: 关闭

返回值:

-1	调用失败
0	调用成功

功能描述:

打开/关闭 SIP 通道 Srtp 加密。

注意事项: 无

相关信息:

驱动程序版本要求	SynCTI Ver. 5.4.3.0 或更高
----------	-------------------------

Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.1.34 SsmSipSetMultiNetIP

设置通道呼叫时使用的 IP 地址。

函数原型:

```
int WINAPI SsmSipSetMultiNetIP (int nCh, char *szSipSetMultiNetIP)
```

参数说明:

nCh	应用通道号
szSipSetMultiNetIP	使用的 IP 地址

返回值:

-1	调用失败
0	调用成功

功能描述:

设置通道呼叫时使用的 IP 地址。

注意事项:

- 使用前必须启用双网卡功能，即配置 [SipMultilpOn=1](#)。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.4.0.0 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.1.35 SsmSipOutCallSendNotify

指定 SIP 协议栈向外发送的 Notify 消息内容。

函数原型:

```
int WINAPI SsmSipOutCallSendNotify (LPSTR pszFrom,LPSTR pszTo,LPSTR *pszHeader,int nHeaderNum,LPSTR pszBody,int nTransType);
```

参数说明:

pszFrom	指定 Notify 消息中 From 字段内容
pszTo	指定 Notify 消息中 To 字段内容和发送 Notify 消息的目的地址
pszHeader	消息头指针数组
nHeaderNum	消息头数目
pszBody	指定 Notify 消息消息体内容
nTransType	传输协议类型, 0: UDP, 1: TCP

返回值:

-1	调用失败
0	调用成功

功能描述:

指定 SIP 协议栈向外发送的 Notify 消息内容。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.4.1.0 或更高
Header	Shpa3api.h

Library	shp_a3.lib
DLL	shp_a3.dll

2.22.1.36 SsmSipSetTransportProtocol

指定 Sip 消息的传输协议类型。

函数原型:

```
int WINAPI SsmSipSetTransportProtocol(int nCh,int nTransType)
```

参数说明:

nCh	通道号
nTransType	Sip 消息传输协议类型, 1: TCP; 0: UDP

返回值:

-1	调用失败
0	调用成功

功能描述:

指定 Sip 消息的传输协议类型。

注意事项: 无

相关信息:

驱动程序版本要求	SynCTI Ver. 5.4.1.0 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.1.37 SsmSipChHoldA

SIP 通道向对端发起或者取消呼叫保持。

函数原型:

```
int WINAPI SsmSipChHoldA(int ch)
```

参数说明:

ch	通道号

返回值:

-1	调用失败
0	调用成功

功能描述:

SIP 通道向对端发起或者取消呼叫保持。

注意事项:

- 函数只能在通道处于通话状态时使用。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.4.2.0 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.1.38 SsmSipChTransferA

SIP 通道向对端发起通话中呼叫转移。

函数原型:

```
int WINAPI SsmSipChTransferA(int ch)
```

参数说明:

ch	通道号
----	-----

返回值:

-1	调用失败
0	调用成功

功能描述:

SIP 通道向对端发起通话中呼叫转移。

注意事项:

- ch 通道必须在通话状态;
- 只有 ch 通道的通话处于呼叫保持状态时, 呼转才能成功, 因此调用该函数前需要先成功调用函数 [SsmSipChHoldA](#)。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.4.2.0 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.2 IP 资源卡的专用函数接口和说明

2.22.2.1 SsmLockMediaCh

锁定媒体通道。如果媒体通道空闲, 该通道进入锁定状态, 并获取该媒体通道可用的资源。

函数原型:

```
int SsmLockMediaCh(int ch)
```

参数说明:

ch	媒体通道编号
----	--------

返回值:

-1	调用失败。失败原因可以通过调用函数 SsmGetLastErrMsg 获得
> 0	调用成功。

功能描述:

如果媒体通道空闲, 该通道进入锁定状态, 并获取该媒体通道可用的资源。只能获取本地 RTP 地址和端口

注意事项:

- 该通道必须要为空闲通道, 成功后通道状态变成 S_IP_MEDIA_LOCK, 值为 160。IP 资源卡专用。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.2.0.0 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.2.2 SsmGetMediaChParam

获取媒体参数。如果通道处于打开状态，则获取当前的媒体参数。如果通道处于锁定状态，那么获取锁定参数。

函数原型：

```
int SsmGetMediaChParam(int ch, struct MediaParam *mParam)
```

参数说明：

ch	媒体通道
mParam	指向 MediaParam 结构的指针。

返回值：

-1	调用失败。失败原因可以通过调用函数 SsmGetLastErrMsg 获得
0	调用成功。

功能描述：

获取指定媒体通道的媒体参数。如果媒体通道为锁定状态，则只能获取本地 IP 地址和端口，参数其他值为空或者为 0。

如果媒体通道为打开状态，则能获取本地 IP 地址、本地端口、远程地址、远程端口、编码格式与发送接收模式。

如果通道不处于打开或者锁定状态，则返回-1。

注意事项：

- IP 资源卡专用

相关信息：

驱动程序版本要求	SynCTI Ver. 5.2.0.0 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.2.3 SsmOpenMediaCh

打开媒体通道

函数原型：

```
int SsmOpenMediaCh(int ch, struct MediaParam *mParam)
```

参数说明：

ch	媒体通道
mParam	指向 MediaParam 结构的指针

返回值：

-1	调用失败。失败原因可以通过调用函数 SsmGetLastErrMsg 获得
0	调用成功。

功能描述：

在指定媒体资源通道上创建 RTP Session 进行语音收发。参数包括发送接收模式、本地 RTP 端口和地址、远程 RTP 端口与地址、发送负载类型等。同时通道状态置成 S_IP_MEDIA_OPEN。

注意事项：

- IP 资源卡专用

相关信息：

驱动程序版本要求	SynCTI Ver. 5.2.0.0 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.2.4 SsmCloseMediaCh

关闭媒体通道

函数原型:

```
int SsmCloseMediaCh(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败。失败原因可以通过调用函数 SsmGetLastErrMsg 获得
0	调用成功。

功能描述:

关闭媒体通道，将通道状态置成锁定状态。

注意事项:

- 通道只在打开状态下该函数才能调用成功。只是关闭语音通道，并没有将通道空闲出来。IP 资源卡专用。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.2.0.0 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.2.5 SsmUnlockMediaCh

释放媒体通道。如果通道处于锁定或者打开状态，那么释放该通道锁定的资源，并进入空闲状态。

函数原型:

```
int SsmUnlockMediaCh(int ch)
```

参数说明:

ch	媒体通道编号
----	--------

返回值:

-1	调用失败。失败原因可以通过调用函数 SsmGetLastErrMsg 获得
> 0	调用成功。

功能描述:

如果通道处于非空闲状态，那么释放该通道锁定的资源，并进入空闲状态。

如果通道处于空闲状态，则返回-1。

注意事项:

- IP 资源卡专用

相关信息:

驱动程序版本要求	SynCTI Ver. 5.2.0.0 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.2.6 SsmUpdateMediaCh

更新媒体通道。

函数原型:

```
int SsmUpdateMediaCh (int ch, struct MediaParam *mParam)
```

参数说明:

ch	媒体通道
mParam	指向 MediaParam 结构的指针

返回值:

-1	调用失败。失败原因可以通过调用函数 SsmGetLastErrMsg 获得
0	调用成功。

功能描述:

在指定媒体资源通道上更新 RTP Session。参数包括发送接收模式、本地 RTP 端口和地址、远程 RTP 端口与地址、发送负载类型等。

注意事项:

- 只有在通道处于打开状态时调用本函数才有意义。IP 资源卡专用。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.2.0.0 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.2.7 SsmCheckMediaChRTPTimeOut

设置媒体资源通道接收远端 RTP 数据的超时时间或检查媒体资源通道接收远端 RTP 数据是否超时。

函数原型:

```
int WINAPI SsmCheckMediaChRTPTimeOut(int ch, int nCheckTime)
```

参数说明:

ch	通道号
nCheckTime	超时时间，单位为秒

返回值:

-1	调用失败
0	调用成功（驱动程序启动后，第一次调用此函数时）或者接收远端 RTP 数据没有超时
1	接收远端 RTP 数据超时

功能描述:

驱动程序启动后，此函数第一次调用时，用于设置媒体资源通道接收远端 RTP 数据的超时时间；其后调用此函数用于检查媒体资源通道接收远端 RTP 数据是否超时。

注意事项:

- 此函数可以反复调用，但只有在第一次调用时用于设置媒体资源通道接收远端 RTP 数据的超时时间，其后调用此函数时，nCheckTime 参数都会被忽略。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.2.6 或更高
Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.3 NAT 穿透专用接口函数

2.22.3.1 SsmIPGetStunPublicIP

获取 NAT 外面公网 IP 和端口。

函数原型:

```
int SsmIPGetStunPublicIP(int nBid,WORD nLocalPort,int nStunServerPort,LPSTR pstrStunServer,StunAddress4 *stunServerAdd,int flag)
```

参数说明:

nBid	板卡 ID
nLocalPort	本地端口
nStunServerPort	Stun 服务器的端口
pstrStunServer	Stun 服务器的 IP 或者域名
stunServerAdd	函数返回的公网 IP 和端口
flag	探测步骤

返回值:

-1	调用失败。失败原因可以通过调用函数 SsmGetLastErrMsg 获得
≥0	探测出来的 NAT 类型。0: 类型未知, 1: 没有 NAT, 2: ConeNat, 3: RestrictedNat, 4: PortRestrictedNat, 5: 对称型 NAT, 6: 对称型 NAT 带防火墙, 7: 探测不过去(探测消息发送失败), 8: 探测失败(没有收到 stun 服务器的回复)

功能描述:

对 IP 卡或者 IP 资源卡进行 NAT 探测，使用 stun 协议。如果 flag=0，时，则会进行 NAT 类型的探测，如果 flag=1，则只进行 NAT 公网 IP 和端口的探测，前一种花费时间相对较长。

注意事项:

- SHN IP 卡需要 RTP 穿透时，需要将 RTP 穿透的配置项设成启用，再调用本函数一次即可，以后的呼叫不用再调用本函数，会自动进行 RTP stun 探测；
- 使用 stun 穿透功能，需要分别进行信令穿透和 RTP 穿透，信令穿透需要在 SIP 节下增加 EnableSIPStun、SIPStunServerIP 两个配置项，RTP 穿透需要在在 BoardID 节下增加 EnableRTPStun、StunServerIP 两个配置项；
- 当该函数的返回值为-1、0、1、5、6、7、8 时（即没有获取到 stun 类型），此时板卡主叫和被叫都会报错，不会成功。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.2.0.0 或更高
----------	-------------------------

Header	Shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.22.4 SHN B 系列卡的专用函数

2.22.4.1 SsmCheckBoardIcmp

打开或关闭 B 型 IP 卡内置网口的的 ICMP 功能。

函数原型:

```
int SsmCheckBoardIcmp(int BoardID, char *sDestAddr, BOOL bRunIcmp)
```

参数说明:

BoardID	板卡 ID
sDestAddr	目的地址
bRunIcmp	打开标志, 1: 打开, 0: 关闭

返回值:

-1	调用失败。
0	调用成功。

功能描述:

打开或关闭 B 型 IP 卡内置网口的的 ICMP 功能。

注意事项:

- 当 ICMP 结果（正常/异常）改变时，驱动程序会抛出 [E_BOARD_ICMP_CHANGE](#) 事件。ICMP 结果的默认值为异常。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.2.6 或更高
Header	Shpa3api.h
Library	shp_a3.lib、Synsip.lib、
DLL	shp_a3.dll、Synsip.dll、M537.dll
BIN	Shn537.bin

2.22.4.2 SsmSipStackAddWhiteOrBlackList

增加黑白名单。

函数原型:

```
int WINAPI SsmSipStackAddWhiteOrBlackList(char *pszSipMsgIp,intnWhiteOrBlack)
```

参数说明:

pszSipMsgIp	IP 地址
nWhiteOrBlack	0: 黑名单, 1: 白名单

返回值:

-1	调用失败。
0	调用成功。

功能描述:

增加黑白名单。

注意事项:

- 既没设置黑名单，也没设置白名单，全部处理；
- 只设置白名单，只处理白名单中的 IP 地址，其他均不处理；
- 只设置黑名单，只过滤黑名单中的 IP 地址，其他均需处理；
- 既设置黑名单，也设置白名单，按照只有黑名单处理；
- 驱动内部最多允许增加 256 条黑名单 IP 地址和 256 条白名单 IP 地址；
- 黑白名单地址不能重复，否则函数调用失败。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.4.3.0 或更高
Header	Shpa3api.h
Library	shp_a3.lib、Synsip.lib
DLL	shp_a3.dll、Synsip.dll

2.22.4.3 SsmSipStackRemoveWhiteOrBlackList

删除黑白名单中的 IP 地址。

函数原型:

```
int WINAPI SsmSipStackRemoveWhiteOrBlackList(char *pszSipMsgIp,intnWhiteOrBlack)
```

参数说明:

pszSipMsgIp	IP 地址
nWhiteOrBlack	0: 黑名单, 1: 白名单

返回值:

-1	调用失败。
0	调用成功。

功能描述:

删除黑白名单中的 IP 地址。

注意事项: 无

相关信息:

驱动程序版本要求	SynCTI Ver. 5.4.3.0 或更高
Header	Shpa3api.h
Library	shp_a3.lib、Synsip.lib
DLL	shp_a3.dll、Synsip.dll

2.22.5 IP 卡注册服务系统专用函数

2.22.5.1 SsmAutoDialAgent

向驱动程序提交 AutoDial 任务，发起一次去话呼叫。

函数原型:

```
int SsmAutoDialAgent(int ch, LPSTR szCallerPhoNum, LPSTR szCalleePhoNum)
```

参数说明:

ch	通道编号
----	------

szCallerPhoNum	存放主叫方电话号码的缓冲区指针
szCalleePhoNum	存放被叫方电话号码的缓冲区指针

返回值:

-1	调用失败。失败原因可以通过调用函数 SsmGetLastErrMsg 获得
0	调用成功

功能描述:

向驱动程序提交 AutoDial 任务，发起一次去话呼叫。

AutoDial 任务启动后，通道的状态机开始自动进行去话呼叫接续，相关内容请参见通道的状态转移部分内容。

IP 通道

对于 IP 通道，本函数会引发驱动程序使用协议指定的消息发起去话呼叫。具体来说，SIP 类型使用 INVITE 消息。

注意事项:

- 本函数只适用于 SIP 通道；
- 被叫号码必须是事先已经注册到 IP 卡的用户名。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.4.1.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmAutoDial](#)

2.22.5.2 SsmSipRegResponse

向驱动内部发送注册认证信息。

函数原型:

```
int SsmSipRegResponse(LPCSTR pszUserName, RegResp *pstRegResp)
```

参数说明:

pszUserName	注册用户名
pstRegResp	RegResp 结构体缓冲区的首地址指针，用于存储需要向驱动发送的认证信息 RegResp 结构体内容

返回值:

-1	表示调用失败
0	表示调用成功

功能描述:

当应用程序收到事件 [E_REG_REQUEST](#) 后调用此函数将需要认证的用户对应的认证信息传给驱动认证。

RegResp 结构体信息如下:

```
typedef struct tag_RegResp
{
    int nAuth;//鉴权参数，取值为 0 和 1
    char szPassword[MAX_SIPIP_ADDR_LENGTH];//用户密码
```

```
char szUsername[MAX_SIPIP_ADDR_LENGTH];
}RegResp;
```

相关信息：

驱动程序版本要求	SynCTI Ver. 5.4.1.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关事件：[E_REG_REQUEST](#)

2.22.5.3 SsmSipGetUserInfoByIndex

通过注册索引号获取成功注册的用户信息。

函数原型：

```
int SsmSipGetUserInfoByIndex(int nRegIndex,Reg_Info *pstReg_Info)
```

参数说明：

nRegIndex	注册索引号
pstReg_Info	Reg_Info 结构体缓冲区的首地址指针, 用于存储获取到的注册成功的用户 Reg_Info 结构体内容

返回值：

-1	表示调用失败
0	表示调用成功

功能描述：

应用程序可以根据注册状态改变事件 [E_REG_REGSTATUS](#), 调用此函数来获取对应注册索引号的用户信息。

Reg_Info 结构体信息如下：

```
typedef struct tag_Reg_Info
{
    int    nRegIndex;//注册用户索引号
    int    nRegState;//注册状态：0 表示注册失败；1 表示注册成功
    int    nT0State;//用于定时器 T0 状态控制
    char   szTelNo[MAX_SIPIP_ADDR_LENGTH];//注册用户名
    char   szClientAddress[MAX_SIPIP_ADDR_LENGTH];//注册客户端 IP 地址
    char   szClientPort[MAX_SIPIP_ADDR_LENGTH];//注册终端端口号
    char   szUserID[MAX_SIPIP_ADDR_LENGTH];//认证用户名
    int    nExpires;
    int    nRegCountForAuthT0;//用于定时器 T0 计数
    int    nRegCountForAuthT1;//用于定时器 T1 计数
    char   szNonce[MAX_SIPIP_ADDR_LENGTH];

    int    nAuth;//鉴权参数，取值为 0 和 1
    char szPassword[MAX_SIPIP_ADDR_LENGTH];//用户密码
}
```

}Reg_Info;

相关信息:

驱动程序版本要求	SynCTI Ver. 5.4.1.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关事件: [E_REG_REGSTATUS](#)

2.22.5.4 SsmGetCallNum

获取呼叫转移时来话呼叫的主被叫方号码信息。

函数原型:

```
int SsmGetCallNum(int ch,LPSTR szCallerNum,LPSTR szCalleeNum)
```

参数说明:

ch	通道编号
szCallerNum	存放呼叫转移原始主叫号码基本信息的字符串指针，由应用程序分配存储空间
szCalleeNum	存放呼叫转移后的被叫号码基本信息的字符串指针，由应用程序分配存储空间

返回值:

-1	表示调用失败
0	表示调用成功

功能描述:

应用程序可以根据呼叫转移事件 [E_RCV_REFER](#)，调用此函数来获取对应通道的主被叫信息。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.4.1.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关事件: [E_RCV_REFER](#)

2.22.5.5 SsmSipStackSetUserAgent

用于设置 SHN 系列板卡或者 synHMP 的用户代理名。

函数原型:

```
int WINAPI SsmSipStackSetUserAgent(const char * pszUserAgent)
```

参数说明:

pszUserAgent	用户代理名
--------------	-------

返回值:

-1	调用失败
0	调用成功

功能描述:

用于设置 SHN 系列板卡或者 synHMP 的用户代理名。

注意事项:

- 配置项 **UserAgent** 可以实现相同功能。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.4.3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.23 ATP 系列录音卡专用函数（REC 系列）

2.23.1 麦克风通道的输入增益控制

2.23.1.1 SsmQueryOpMicGain

查询通道是否支持设置输入信号增益的操作。

函数原型:

```
int SsmQueryOpMicGain(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
0	不支持
1	支持

功能描述:

查询通道是否支持设置输入信号增益的操作。

注意事项:

- 只有 ATP 系列板卡的模拟录音通道（包括模拟中继线通道和麦克风录音通道）才支持此项功能。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.1 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetMicGain](#)

2.23.1.2 SsmSetMicGain

设置模拟录音通道的输入信号的增益。

函数原型:

```
int SsmSetMicGain(int ch, int nMicGain)
```

参数说明:

ch	通道编号
----	------

nGain	增益值: 0: 正常增益 1: 提升 20DB
-------	-------------------------------

返回值:

-1	调用失败
0	设置成功

功能描述:

设置模拟录音通道的输入信号的增益。

注意事项:

- 配置项 [MicGain](#) 可以实现相同功能；
- 本函数只适用于 ATP 系列板卡（不包括 ATP-24A 系列板卡）的模拟录音通道。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.1 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetMicGain](#)

2.23.1.3 SsmGetMicGain

获取模拟录音通道上设置的输入信号增益值。

函数原型:

```
int SsmGetMicGain(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败
0	正常增益
1	高增益 (20DB)

功能描述:

获取模拟录音通道上设置的输入信号增益值。

注意事项:

- 本函数只适用于 ATP 系列板卡的模拟录音通道。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.1 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmSetMicGain](#)

2.23.2 有关 MF 检测器的函数

2.23.2.1 SsmEnableRxMF

启动或关闭 MF 检测器。

函数原型:

```
int SsmEnableRxMF (int ch, BOOL bRun)
```

参数说明:

ch	通道编号
	是否开启 MF 检测器的控制开关:
bRun	TRUE: 开启; FALSE: 关闭

返回值:

-1	调用失败。失败原因可以通过调用函数 SsmGetLastErrMsg 获得
0	调用成功

功能描述:

启动或关闭 MF 检测器。

注意事项:

- 驱动程序默认关闭 MF 检测器。当用户需要自行控制 MF 检测器开关时才需要调用本函数；
- 本函数只支持 ATP 板卡；
- 配置项 [AlwaysEnableRxMF](#) 可以实现类似功能；
- 本函数的有效期为永久。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.2.3 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.23.2.2 SsmClearRxMFBuf

清空驱动程序内部的 MF 按键号码接收缓冲区。

函数原型:

```
int SsmClearRxMFBuf(int ch)
```

参数说明:

ch	通道编号
----	------

返回值:

-1	调用失败。失败原因可以通过调用函数 SsmGetLastErrMsg 获得
0	调用成功

功能描述:

清空驱动程序内部的 MF 接收缓冲区。

注意事项:

- 当通道迁移到空闲状态时，驱动程序会自动清空 MF 检测器的缓冲区。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.1.1 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.23.2.3 SsmGetMFStr

获取保存在 MF 检测器缓冲区的 MF 字符，字符为标准的 ASCII 码

函数原型:

```
int SsmGetMFStr(int ch, LPSTR pszMF)
```

参数说明:

ch	通道编号
pszMF	存放接收 MF 字符串的指针，由应用程序分配存储空间

返回值:

-1	调用失败
≥0	MF 检测器中的字符个数

功能描述:

获取保存在 MF 检测器缓冲区的 MF 字符，字符为标准的 ASCII 码

注意事项:

- 本函数调用时，驱动程序不会清空 MF 检测器缓冲区。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.1.1 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.24 DTP 系列板卡的专用函数（REC 系列）

2.24.1 基于状态机编程模式的函数

2.24.1.1 获取基本信息的函数

2.24.1.1.1 SpyGetMaxCic

获取系统中被监控电路的总数。

函数原型:

```
int SpyGetMaxCic()
```

参数说明：无

返回值：

-1	调用失败
≥0	SpyCic 的总数

功能描述：

获取系统中被监控电路的总数，即 [AppSpyCICTable] 配置节中的 [TotalAppSpyCIC](#) 配置项的设定值。有关 SpyCic 的更多信息，请参见第 1 章中 “[DTP 系列板卡](#)” 部分内容。

注意事项：

- 本函数只适用于 DTP 系列板卡。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SpyChToCic](#)

2.24.1.1.2 SpyChToCic

根据通道的逻辑编号来查找对应的 SpyCic 的逻辑编号。

函数原型：

```
int SpyChToCic(int ch)
```

参数说明：

ch	通道的逻辑编号
----	---------

返回值：

-1	失败
≥0	返回对应的 SpyCic 的逻辑编号

功能描述：

根据通道的逻辑编号来查找对应的 SpyCic 的逻辑编号。有关 SpyCic 的逻辑编号的更多信息，请参见第 1 章中 “[DTP 系列板卡](#)” 部分内容。

注意事项：

- 本函数只适用于 DTP 系列板卡。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SpyGetMaxCic](#)

2.24.1.2 获取呼叫进展的函数

2.24.1.2.1 SpyGetState

获取驱动程序基于 SpyCic 的状态机的当前状态。

函数原型:

```
int SpyGetState(int nCic)
```

参数说明:

nCic	被监控电路编号
------	---------

返回值:

返回值为-1 表示调用失败，其它返回值的含义如下表所示：

值	shpa3api.h 中的宏	状态描述
0	S_SPY_STANDBY	空闲
2	S_SPY_RINGING	振铃
3	S_SPY_TALKING	连接（通话）。当 SpyCic 移动到进入 S_SPY_TALKING 状态后，应用程序可以： ◆ 调用 SpyRecToFile 启动录音 ◆ 调用 SpyGetCalleeId 取得被叫号码 ◆ 调用 SpyGetCallerId 取得主叫号码 ◆ 以主叫方通道或被叫方通道为参数，调用基于通道的函数获取其它相关的信息。
11	S_CALL_UNAVAILABLE	通道不可用
105	S_SPY_RCVPHONUM	接收被叫号码
110	S_SPY_SS1RESET	电路复原
111	S_SPY_SS1WAITBWDACK	SS1：等待后向证实
112	S_SPY_SS1WAITKB	SS1：等待 KB 信号
其它	保留	

功能描述:

获取驱动程序基于 SpyCic 的状态机的当前状态。

注意事项:

- 建议使用 [E_CHG_SpyState](#) 事件来替代本函数；
- 本函数只适用于 DTP 系列板卡。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SpyGetCalleeId](#), [SpyGetCallerId](#), [SpyGetCallOutCh](#), [SpyGetCallInCh](#)

2.24.1.3 获取呼叫方向的函数

2.24.1.3.1 SpyGetCallInCh

请参见 [SpyGetCallOutCh](#)

2.24.1.3.2 SpyGetCallOutCh

获取当前呼叫中的被叫方通道（SpyGetCallOutCh）或主叫方通道（SpyGetCallInCh）的逻辑编号。

函数原型：

```
int SpyGetCallInCh(int nCic)  
int SpyGetCallOutCh(int nCic)
```

参数说明：

nCic	SpyCic 的逻辑编号。有关 SpyCic 的更多信息参见第 1 章中“ DTP 系列板卡 ”部分内容
------	----------------------------------------------------------------------

返回值：

-1	调用失败
≥0	SpyGetCallInCh 函数： 返回主叫方通道的逻辑编号 SpyGetCallOutCh 函数： 返回被叫方通道的逻辑编号

功能描述：

获取当前呼叫中的被叫方通道（SpyGetCallOutCh）或主叫方通道（SpyGetCallInCh）的逻辑编号。有关被叫方通道的更多信息请参见第 1 章中“[DTP 系列板卡](#)”部分内容。

注意事项：

- 当 SpyCic 的状态迁移到 S_SPY_RINGING 或 S_SPY_TALKING 时，调用 SpyGetCallOutCh 或 SpyGetCallInCh 才能得到有效信息；
- 本函数只适用于 DTP 系列板卡。

相关信息：

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SpyGetCallerId](#), [SpyGetCalleeId](#)

2.24.1.4 取得主叫方号码和被叫方号码的函数

2.24.1.4.1 SpyGetCallerId

请参见 [SpyGetCalleeId](#)

2.24.1.4.2 SpyGetCalleeId

获取本次呼叫的主叫方号码（SpyGetCallerId）或被叫方号码（SpyGetCalleeId）。

函数原型：

```
int SpyGetCallerId(int nCic, char *pszCallingPartyNumber)  
int SpyGetCalleeId(int nCic, char * pszCalledPartyNumber)
```

参数说明：

nCic	SpyCic 的逻辑编号
pszCallingPartyNumber	存放主叫方号码的缓冲区指针，由应用程序分配存储空间，不能少于 50 个字符
pszCalledPartyNumber	存放被叫方号码的缓冲区指针，由应用程序分配存储空间，不能少于 50 个字符

返回值:

-1	调用失败
≥0	号码的长度(个数)

功能描述:

呼叫建立后, SpyGetCallerId 函数获取本次呼叫的主叫方号码, SpyGetCalleeId 获取本次呼叫的被叫方号码。

注意事项:

- 当 SpyCic 的状态迁移到 S_SPY_RINGING 或 S_SPY_TALKING 时, 调用 SpyGetCallerId 或 SpyGetCalleeId 才能得到完整的信息;
- 本函数只适用于 DTP 系列板卡。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SpyGetCallOutCh](#), [SpyGetCallInCh](#)

2.24.1.5 获取呼叫挂机信息的函数

2.24.1.5.1 SpyGetHangupInfo

获取驱动程序基于 SpyCic 的当前挂机信息。

函数原型:

DWORD WINAPI SpyGetHangupInfo(int nCic)

参数说明:

nCic	被监控电路编号
------	---------

返回值:

-1	调用失败或无挂机信息
其他	高 16 位比特表示发送拆线消息 SpyCic 所在的监控 PCM 号; 低 16 位比特表示 SpyCic 的挂机信息, 具体为: 0: 被叫方先挂机 1: 主叫方先挂机

功能描述:

获取驱动程序基于 SpyCic 的当前挂机信息。

注意事项:

- 建议使用 [E_CHG_SpyHangupInfo](#) 事件来替代本函数;
- 本函数只适用于 DTP 系列板卡的 ISDN 线路监控。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.1.1.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SpyGetCalleeId](#), [SpyGetCallerId](#), [SpyGetCallOutCh](#), [SpyGetCallInCh](#)

2.24.1.6 取得主叫方号码类型和被叫方号码类型的函数

2.24.1.6.1 SpyGetCallerType

请参见 [SpyGetCalleeType](#)

2.24.1.6.2 SpyGetCalleeType

获取本次呼叫的主叫方号码类型（SpyGetCallerType）或被叫方号码类型（SpyGetCalleeType）。

函数原型：

```
int WINAPI SpyGetCallerType(int nCic)  
int WINAPI SpyGetCalleeType(int nCic)
```

参数说明：

nCic	SpyCic 的逻辑编号
------	--------------

返回值：

-1	调用失败
≥0	主/被叫号码类型

功能描述：

呼叫建立后，[SpyGetCallerType](#) 函数获取本次呼叫的主叫方号码类型，[SpyGetCalleeType](#) 获取本次呼叫的被叫方号码类型。

注意事项：

- 当 SpyCic 的状态迁移到 S_SPY_RINGING 或 S_SPY_TALKING 时，调用 [SpyGetCallerType](#) 或 [SpyGetCalleeType](#) 才能得到完整的信息；
- 本函数只适用于 DTP 系列板卡的 ISDN 线路监控。

相关信息：

驱动程序版本要求	SynCTI Ver. 5.1.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数： [SpyGetCallOutCh](#), [SpyGetCallInCh](#)

2.24.1.7 获取数字中继线状态的函数

2.24.1.7.1 SpyGetLinkStatus

获取被监控的信令链路中继线的工作状态。

函数原型：

```
int SpyGetLinkStatus(int nSpyPcmNo, UCHAR ucFlag);
```

参数说明：

nSpyPcmNo	被监控 PCM 的逻辑编号，取值范围：由配置项 TotalSpyPcm 决定
ucFlag	0: 选择上行数字中继线 1: 选择下行数字中继线

返回值：

-1	失败
0	故障
1	正常

功能描述:

获取被监控的信令链路中继线的工作状态。

注意事项:

- 本函数只适用于 DTP 系列板卡。

相关信息:

驱动程序版本要求	SynCTI Ver. 3.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.24.1.8 获取 SS7 信令呼叫状态机同步输出的原始信令消息

2.24.1.8.1 SsmGetSs7SpyMsu

获取 SS7 信令呼叫状态机同步输出的原始信令消息。

函数原型:

```
int SsmGetSs7SpyMsu (PUCHAR* ppucMsuBuf)
```

参数说明:

ppucMsuBuf	保存原始的 ISUP/TUP 消息的缓冲区指针, 由应用程序分配存储空间, 不能少于 273 个字节
------------	----------------------------------------------------

返回值:

0	没有消息
>0	返回消息的长度, 单位为字节
-1	调用失败

功能描述:

获取 SS7 信令呼叫状态机同步输出的原始信令消息。取出消息的方式为“先进先出”, 即总是返回最先收到的消息。每当驱动程序收到一条 MSU, 会将其保存到驱动程序内部的 MSU 接收缓冲区中, 然后向应用程序抛出 [E_RCV_Ss7SpyMsu](#) 消息。

注意事项:

- 只有当配置项 [bOpenSpySS7Msu](#) 设置为 1 时, 驱动程序才会将收到的 ISUP/TUP 消息复制到内部缓冲区, 同时向应用程序抛出 [E_RCV_Ss7SpyMsu](#) 消息。

相关信息:

驱动程序版本要求	SynCTI Ver. 4.7.2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: 无

相关事件: [E_RCV_Ss7SpyMsu](#)

2.24.1.9 取得连接号码的函数

2.24.1.9.1 SpyGetConId

获取本次呼叫的连接号码。

函数原型:

```
int WINAPI SpyGetConId (int nCic, char *pcCid)
```

参数说明:

nCic	SpyCic 的逻辑编号
pcCid	存放连接号码的缓冲区指针, 由应用程序分配存储空间, 不能少于 50 个字符

返回值:

≥0	号码的长度 (个数)
-1	调用失败

功能描述:

呼叫建立后, 获取本次呼叫的连接号码。

注意事项:

- 当 SpyCic 的状态迁移到 S_SPY_TALKING 时, 只有调用 SpyGetConId 函数才能得到完整的信息。
- 本函数只适用于 DTP 系列板卡的 ISDN 线路监控。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.1.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: 无

2.24.1.10 获取 ISDN 第 2 层 (L2) 监控原始消息的函数

2.24.1.10.1 SsmGetIsdnL2SpyMsu

从驱动程序的内部缓冲区中取出一条 ISDN 第 2 层 (L2) 监控原始消息。

函数原型:

```
int WINAPI SsmGetIsdnL2SpyMsu(int nPcmId, int nBufLen,PUCHAR pucMsuBuf);
```

参数说明:

nPcmId	数字中继线的逻辑编号
nBufLen	应用程序提供的 pucMsuBuf 空间字节大小
pucMsuBuf	返回 ISDN L2 层原始消息。pucMsuBuf 由应用程序分配存储空间

返回值:

0	缓冲区中没有消息
>0	pucMsuBuf 中消息的实际长度, 单位为字节
-1	调用失败

功能描述:

从驱动程序的内部缓冲区中取出一条 ISDN 第 2 层 (L2) 监控原始消息。

注意事项:

- 在实际应用中，只有当配置项 [bOpenSpyIsdnL2](#) 设置为 1 时，调用本函数才有意义。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.3.1 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: 无

相关事件: [E_RCV_IsdnL2SpyMsu](#)

2.25 DST 系列录音卡的专用函数 (REC 系列)

2.25.1 SsmGetDstChSNRofUplink

取得 DST 通道上行信号信噪比。

函数原型:

```
int SsmGetDstChSNRofUplink(int ch)
```

参数说明:

ch	通道号
----	-----

返回值:

-1	调用失败
≥ 0	线路信噪比值

功能描述:

取得指定 DST 通道当前上行信号信噪比值，信噪比可用来监测线路信号的质量。

注意事项:

- 本函数只支持 DST 系列 B 型板卡。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.1.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetDstChSNRofDownlink](#)

2.25.2 SsmGetDstChSNRofDownlink

取得 DST 通道下行信号信噪比。

函数原型:

```
int SsmGetDstChSNRofDownlink(int ch)
```

参数说明:

ch	通道号
----	-----

返回值:

-1	调用失败
≥ 0	线路信噪比值

功能描述:

取得指定 DST 通道当前下行信号信噪比值，信噪比可用来监测线路信号的质量。

注意事项:

- 本函数只支持 DST 系列 B 型板卡。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.1.0.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmGetDstChSNRofUplink](#)

2.25.3 SsmGetDstChVoltageState

取得 DST 通道线路的信号电平状态。

函数原型:

```
int SsmGetDstChVoltageState (int ch)
```

参数说明:

ch	通道号
----	-----

返回值:

-1	调用失败
0	正常
3	不正常

功能描述:

取得指定 DST 通道线路的信号电平状态。

注意事项:

- 本函数只支持 DST 系列 B 型板卡。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.2.0.1 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.26 PCM1280E 录音卡的专用函数（REC 系列）

2.26.1 SsmGetPcm32LineState

获取线路同步状态和信号状态。

函数原型:

```
int WINAPI SsmGetPcm32LineState(int ch)
```

参数说明:

ch	通道号
----	-----

返回值:

-1	调用失败 线路同步状态和信号状态值，仅使用低 16 位。其中 bit0-bit7 为该通道同步状态值，bit8-bit15 为该通道信号状态值。 取值范围如下：
≥0	<ul style="list-style-type: none"> a) 同步状态值的取值范围： <ul style="list-style-type: none"> 1) 0 表示-帧同步正常 2) 1 表示-断线 3) 2 表示-复位状态 4) 3 表示-帧同步不正常 b) 信号状态值的取值范围： <ul style="list-style-type: none"> 1) 0 表示-数据正常 2) 1 表示-断线或没有数据 3) 2 表示-复位状态 4) 3 表示-数据异常

功能描述:

获取线路同步状态和信号状态。

注意事项:

- 本函数只支持 PCM1280E 板卡。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.1.2 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.27 IPR 系列录音产品的专用函数（REC 系列）

2.27.1 SynIPRRecorder 的专用函数

2.27.1.1 SsmIPRSetRecVolume

设置 primary 和 secondary 的录音音量。

函数原型:

```
int SsmIPRSetRecVolume(int nChId, int nPrimaryVlm, int nSecondaryVlm)
```

参数说明:

nChId	通道号
nPrimaryVlm	primary 的录音音量。取值范围：-7~+6, -7 表示关闭，其余取值表示音量增益（乘以 3 即为分贝），大于 0 表示音量提升，小于 0 表示音量衰减
nSecondaryVlm	secondary 的录音音量。取值范围：-7~+6, -7 表示关闭，其余取值表示音量增益（乘以 3 即为分贝），大于 0 表示音量提升，小于 0 表示音量衰减

返回值:

-1	调用失败
≥0	调用成功

功能描述:

设置 primary 和 secondary 的录音音量。

注意事项:

- 本函数只支持 SynIPRecorder。

相关信息:

驱动程序版本要求	SynCTI Ver.5.3.2.0
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.27.1.2 SsmIPRSetMixerType

设置录音时的混音类型。

函数原型:

```
int SsmIPRSetMixerType(int nChId, int nFlag)
```

参数说明:

nChId	通道号
nFlag	0 只录 Primary 的声音； 1 只录 Secondary 的声音； 2 两边的声音都录

返回值:

-1	调用失败
≥0	调用成功

功能描述:

设置录音时的混音类型。

注意事项:

- 本函数只支持 SynIPRecorder。

相关信息:

驱动程序版本要求	SynCTI Ver.5.3.2.0
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmIPRGetMixerType](#)

2.27.1.3 SsmIPRGetMixerType

获取通道的混音类型。

函数原型:

```
int SsmIPRGetMixerType(int nChId)
```

参数说明:

nChId	通道号
-------	-----

返回值:

-1	调用失败
0	只录 Primary 的声音
1	只录 Secondary 的声音
2	两边的声音都录

功能描述:

获取通道的混音类型。

注意事项:

- 本函数只支持 SynIPRecorder。

相关信息:

驱动程序版本要求	SynCTI Ver.5.3.2.0
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmIPRSetMixerType](#)

2.27.1.4 SsmIPRGetRecSlaverCount

获取录音 Slaver 总数。

函数原型:

```
int SsmIPRGetRecSlaverCount(int nBld)
```

参数说明:

nBld	板卡 ID
------	-------

返回值:

<0	调用失败
>=0	表示成功, 返回值为当前录音 Slaver 总数

功能描述:

获取录音 Slaver 总数。

注意事项:

- 本函数只支持 SynIPRecorder。

相关信息:

驱动程序版本要求	SynCTI Ver.5.3.2.0
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmIPRGetRecSlaverList](#), [SsmIPRGetRecSlaverInfo](#)

2.27.1.5 SsmIPRGetRecSlaverList

获取指定数量的录音 Slaver 信息。

函数原型:

```
int SsmIPRGetRecSlaverList(int nBld, int nRecSlaverNum, int* nReturnSlaverNum, PIPR_SLAVERADDR pIPR_SlaverAddr)
```

参数说明:

nBld	板卡 ID
nRecSlaverNum	需要获取信息的录音 Slaver 数量
nReturnSlaverNum	实际查询到的有效 Slaver 的数量
pIPR_SlaverAddr	pIPR_SlaverAddr 为 PIPR_SLAVERADDR 类型的结构体数组的首地址, 用于返回各个 Slaver 的地址信息

返回值:

<0	调用失败
>=0	调用成功

功能描述:

获取指定数量录音 Slaver 信息。

注意事项:

- 本函数只支持 SynIPRecorder。

相关信息:

驱动程序版本要求	SynCTI Ver.5.3.2.0
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmIPRGetRecSlaverCount](#), [SsmIPRGetRecSlaverInfo](#)

2.27.1.6 SsmIPRGetRecSlaverInfo

获取某一个录音 Slaver 的详细信息。

函数原型:

```
int SsmIPRGetRecSlaverInfo(int nBld, int nRecSlaverId, BOOL *bStarted, int *nTotalResources, int *nUsedResources, int *nThreadPairs, PCOMPUTER_INFO pcomputerinfo)
```

参数说明:

nBld	板卡 ID
nRecSlaverId	需要获取信息的录音 Slaver ID
bStarted	该 Slaver 是否开启，既是否已处于工作状态
nTotalResources	该录音 Slaver 总的录音资源数
nUsedResources	该录音 Slaver 已用的录音资源数
nThreadPairs	该录音 Slaver 的线程对数
pcomputerinfo	指向结构体 COMPUTER_INFO 的指针，里面保存的是 slaver 所在机器的硬件信息

返回值:

<0	调用失败
>=0	调用成功

功能描述:

获取某一个录音 Slaver 的详细信息。

注意事项:

- 本函数只支持 SynIPRecorder。

相关信息:

驱动程序版本要求	SynCTI Ver.5.3.2.0
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmIPRGetRecSlaverCount](#), [SsmIPRGetRecSlaverList](#)

2.27.1.7 SsmIPRStartRecSlaver

启动录音 Slaver，使其进入正常工作状态。

函数原型:

```
int SsmIPRStartRecSlaver(int nBld, int nRecSlaverId, int* nTotalResources, int *nThreadPairs)
```

参数说明:

nBld	板卡 ID
nRecSlaverId	需要获取信息的录音 Slaver ID
nTotalResources	该录音 Slaver 总的录音资源数
nThreadPairs	该录音 Slaver 的线程对数, 根据 CPU 的数量赋值, 最大为 64, 建议设置和 CPU 的数量一致

返回值:

<0	调用失败
>=0	调用成功

功能描述:

启动录音 Slaver, 使其进入正常工作状态。

注意事项:

- 本函数只支持 SynIPRecorder;
- 该函数返回成功只是表示它已经往 Slaver 发送 Start Record Slaver 的命令, 至于该命令在 Slaver 端是否成功处理需要通过事件 [E_IPR_START_SLAVER_CB](#) 上携带的 dwParam 的值来确定。

相关信息:

驱动程序版本要求	SynCTI Ver.5.3.2.0
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmIPRCloseRecSlaver](#)

2.27.1.8 SsmIPRCloseRecSlaver

关闭录音 Slaver, 使其进入怠工状态。

函数原型:

```
int SsmIPRCloseRecSlaver(int nBld, int nRecSlaverId)
```

参数说明:

nBld	板卡 ID
nRecSlaverId	需要关闭的录音 Slaver ID

返回值:

<0	调用失败
>=0	调用成功

功能描述:

关闭录音 Slaver, 使其进入怠工状态。

注意事项:

- 本函数只支持 SynIPRecorder;
- 该函数返回成功只是表示它已经往 Slaver 发送 Close Record Slaver 的命令, 至于该命令在 Slaver 端是否成功处理需要通过事件 [E_IPR_CLOSE_SLAVER_CB](#) 上携带的 dwParam 的值来确定。

相关信息:

驱动程序版本要求	SynCTI Ver.5.3.2.0
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmIPRStartRecSlaver](#)

2.27.1.9 SsmIPRActiveSession

指定录音 Slaver 开启相应的端口用于接收 RTP 数据。

函数原型:

```
int SsmIPRActiveSession(int nChl, int nRecSlaverId, DWORD dwSessionId, LPSTR szPriAddr, int nPriPort, int* pnPriRcvPort, int nPriCodec, LPSTR szSecAddr, int nSecPort, int *pnSecRcvPort, int nSecCodec)
```

参数说明:

nChl	通道 ID														
nRecSlaverId	录音 Slaver ID														
dwSessionId	与通道 nChl 绑定的 Session 的 SessionId														
szPriAddr	primary 端 RTP 的 IP 地址														
nPriPort	primary 端 RTP 的端口														
pnPriRcvPort	用来接收 primary 端 RTP 的端口														
nPriCodec	primary 端的编解码类型, 支持的负载值如下表所示: <table border="1"><thead><tr><th>编解码类型</th><th>负载值</th></tr></thead><tbody><tr><td>μ-Law</td><td>0</td></tr><tr><td>A-Law</td><td>8</td></tr><tr><td>G723</td><td>4</td></tr><tr><td>G729</td><td>18</td></tr><tr><td>GSM</td><td>3</td></tr><tr><td>G722</td><td>9</td></tr></tbody></table>	编解码类型	负载值	μ-Law	0	A-Law	8	G723	4	G729	18	GSM	3	G722	9
编解码类型	负载值														
μ-Law	0														
A-Law	8														
G723	4														
G729	18														
GSM	3														
G722	9														
szSecAddr	secondary 端 RTP 的 IP 地址														
nSecPort	secondary 端 RTP 的端口														
pnSecRcvPort	用来接收 secondary 端 RTP 的端口														
nSecCodec	secondary 端的编解码类型, 支持的负载值如下表所示: <table border="1"><thead><tr><th>编解码类型</th><th>负载值</th></tr></thead><tbody><tr><td>μ-Law</td><td>0</td></tr><tr><td>A-Law</td><td>8</td></tr><tr><td>G723</td><td>4</td></tr><tr><td>G729</td><td>18</td></tr><tr><td>GSM</td><td>3</td></tr><tr><td>G722</td><td>9</td></tr></tbody></table>	编解码类型	负载值	μ-Law	0	A-Law	8	G723	4	G729	18	GSM	3	G722	9
编解码类型	负载值														
μ-Law	0														
A-Law	8														
G723	4														
G729	18														
GSM	3														
G722	9														

返回值:

<0	调用失败
>=0	调用成功

功能描述:

指定录音 Slaver 开启相应的端口用于接收 RTP 数据。

注意事项:

- 本函数只支持 SynIPRecorder;
- 该函数返回成功只是表示它已经往 Slaver 发送 ActiveSession 的命令, 至于该命令在 Slaver 端是否成功处理需要通过事件 [E_IPR_ACTIVE_SESSION_CB](#) 上携带的 dwParam 的值来确定;
- 参数 dwSessionId 与 IPAnalyzer 通道产生的 SessionId 不需要相同, dwSessionId 是用于区分 IPRecorder 通道的录音 Session;
- 参数 szPriAddr 与 szSecAddr 未使用, 使用“127.0.0.1”即可;
- 参数 nPriPort 与 nSecPort 未使用, 填入有效端口范围内 (1~65535) 的任意端口值均可;
- 参数 nPriCodec 与 nSecCodec 需填入 IPRecorder 支持的负载格式, 但是由于 IPRecorder 支持自适应负载格式, 所以 nPriCodec 和 nSecCodec 不一定要与实际通话使用的负载格式相同。

相关信息:

驱动程序版本要求	SynCTI Ver.5.3.2.7
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmIPRDeActiveSession](#)

2.27.1.10 SsmIPRDeActiveSession

关闭录音 Slaver 对应的 Session 接收端口。

函数原型:

```
int SsmIPRDeActiveSession(int nCh)
```

参数说明:

nChId	通道 ID
-------	-------

返回值:

<0	调用失败
>=0	调用成功

功能描述:

关闭录音 Slaver 对应的 Session 接收端口。

注意事项:

- 本函数只支持 SynIPRecorder;
- 该函数返回成功只是表示它已经往 Slaver 发送 DeActiveSession 的命令, 至于该命令在 Slaver 端是否成功处理需要通过事件 [E_IPR_DEACTIVE_SESSION_CB](#) 上携带的 dwParam 的值来确定。

相关信息:

驱动程序版本要求	SynCTI Ver.5.3.2.0
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmIPRAActiveSession](#)

2.27.1.11 SsmIPRAActiveAndRecToFile

指定录音 Slaver 开启相应的端口用于接收 RTP 数据同时开始录音。

函数原型:

```
int SsmIPRAActiveAndRecToFile(int nCh, int nRecSlaverId, int dwSessionId, int nCodec, int* pnPriRcvPort, int* pnSecRcvPort, LPCSTR pszFileName, int nFormat, DWORD dwStartPos, DWORD dwBytes, DWORD dwTime, int nMask)
```

参数说明:

nChId	通道 ID														
nRecSlaverId	录音 Slaver ID														
dwSessionId	与通道 nChId 绑定的 Session 的 SessionId														
nCodec	Session 上去话的编解码格式 (RTP 负载值)。支持的负载值如下表所示: <table border="1"><thead><tr><th>编解码类型</th><th>负载值</th></tr></thead><tbody><tr><td>μ-Law</td><td>0</td></tr><tr><td>A-Law</td><td>8</td></tr><tr><td>G723</td><td>4</td></tr><tr><td>G729</td><td>18</td></tr><tr><td>GSM</td><td>3</td></tr><tr><td>G722</td><td>9</td></tr></tbody></table>	编解码类型	负载值	μ-Law	0	A-Law	8	G723	4	G729	18	GSM	3	G722	9
编解码类型	负载值														
μ-Law	0														
A-Law	8														
G723	4														
G729	18														
GSM	3														
G722	9														

pnPriRcvPort	用来接收 primary 端 RTP 的端口
pnSecRcvPort	用来接收 secondary 端 RTP 的端口
pszFileName	具体可以查看 SsmRecToFile 对应的参数
nFormat	具体可以查看 SsmRecToFile 对应的参数
dwStartPos	具体可以查看 SsmRecToFile 对应的参数
dwBytes	具体可以查看 SsmRecToFile 对应的参数
dwTime	具体可以查看 SsmRecToFile 对应的参数
nMask	具体可以查看 SsmRecToFile 对应的参数

返回值:

<0	调用失败
>=0	调用成功

功能描述:

指定录音 Slaver 开启相应的端口用于接收 RTP 数据同时开始录音。

注意事项:

- 本函数只支持 SynIPRecorder。
- 该函数返回成功只是表示它已经往 Slaver 发送 ActiveSessionAndRecToFile 的命令, 至于该命令在 Slaver 端是否成功处理需要通过事件 [E_IPR_ACTIVE_AND_REC_CB](#) 上携带的 dwParam 的值来确定。

相关信息:

驱动程序版本要求	SynCTI Ver.5.3.2.7
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmIPRDeActiveAndStopRecToFile](#)

2.27.1.12 SsmIPRDeActiveAndStopRecToFile

关闭录音 Slaver 对应的 Session 接收端口, 并停止录音。

函数原型:

```
int SsmIPRDeActiveAndStopRecToFile (int nCh)
```

参数说明:

nChId	通道 ID
-------	-------

返回值:

<0	调用失败
>=0	调用成功

功能描述:

关闭录音 Slaver 对应的 Session 接收端口, 并停止录音。

注意事项:

- 本函数只支持 SynIPRecorder。
- 该函数返回成功只是表示它已经往 Slaver 发送 DeActiveSessionAndStopRecToFile 的命令, 至于该命令在 Slaver 端是否成功处理需要通过事件 [E_IPR_DEACTIVE_AND_STOPREC_CB](#) 上携带的 dwParam 的值来确定。

相关信息:

驱动程序版本要求	SynCTI Ver.5.3.2.0
Header	shpa3api.h

Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmIPRActiveAndRecToFile](#)

2.27.1.13 SsmGetUSBKeySerial

获取 SynIPRecorder 板卡或 SynIPAnalyzer 板卡的序列号。

函数原型:

DWORD WINAPI SsmGetUSBKeySerial(int nBld)

参数说明:

nBld	板卡 Id
------	-------

返回值:

0xffffffff	板卡 Id 对应的板卡并非 SynIPRecorder 板卡或 SynIPAnalyzer 板卡
>0	板卡的序列号

功能描述:

获取 SynIPRecorder 板卡或 SynIPAnalyzer 板卡的序列号。

注意事项:

- 本函数支持 SynIPRecorder 和 SynIPAnalyzer。

相关信息:

驱动程序版本要求	SynCTI Ver5.3.2.0
Header	Shpa3api.h
Library	Shp_a3.lib
DLL	Shp_a3.dll

2.27.1.14 SsmIsBoardIPR

判断指定的板卡是否为 SynIPRecorder 或 SynIPAnalyzer。

函数原型:

BOOL WINAPI SsmIsBoardIPR(int nBld)

参数说明:

nBld	板卡 Id
------	-------

返回值:

FALSE	由 nBld 指定的板卡不是 SynIPRecorder 或 SynIPAnalyzer。
TRUE	由 nBld 指定的板卡是 SynIPRecorder 或 SynIPAnalyzer。

功能描述:

判断指定的板卡是否为 SynIPRecorder 或 SynIPAnalyzer。

注意事项:

- 本函数支持 SynIPRecorder 和 SynIPAnalyzer。

相关信息:

驱动程序版本要求	SynCTI Ver5.3.2.0
Header	Shpa3api.h
Library	Shp_a3.lib
DLL	Shp_a3.dll

2.27.1.15 SsmIPRConnectToSlaver

在 Slaver 被动连接模式下 Master 可以调用此函数连接 Slaver。

函数原型:

```
int WINAPI SsmIPRConnectToSlaver(char* szSvrAddr, int nSvrPort)
```

参数说明:

szSvrAddr	要连接的 Slaver 的 IP 地址
nSvrPort	要连接的 Slaver 的端口

返回值:

-1	连接失败或调用失败, 具体错误信息请参考 SsmGetLastErrMsg
0	调用成功

功能描述:

在 Slaver 被动连接模式下 Master 可以调用此函数连接 Slaver。

注意事项:

- 本函数只支持 SynIPRecorder。
- 被动连接模式设置请查看配置项 [ActiveConnect](#)。

相关信息:

驱动程序版本要求	SynCTI Ver.5.3.2.0
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

2.27.2 相关函数: SynIPAnalyzer 的专用函数

2.27.2.1 SsmIPRAddProtocol

增加对某个协议的监测。

函数原型:

```
int SsmIPRAddProtocol(int nBId, int nPtId, PIPR_MONITOR_CFGS pParams, DWORD dwLen)
```

参数说明:

nBId	板卡 ID
nPtId	协议类型 ID
pParams	被监控协议的结构体参数 PIPR_MONITOR_CFGS
dwLen	pParam 参数的数据长度

返回值:

<0	调用失败
≥ 0	调用成功

功能描述:

增加对某个协议的监测。

IPR_MONITOR_CFGS 结构说明:

```
typedef struct
{
    union
    {
```

```
IPR_SCCP_CFGS CISCO; // CISCO SCCP协议
IPR_SIP_CFGS SIP; // SIP协议
IPR_H323_CFGS Avaya_H323; // Avaya H323协议
IPR_SHORTEL_MGCP_CFGS Shortel_MGCP; // Shortel MGCP协议
IPR_H323_CFGS H323; // H323协议
IPR_PANASONIC_MGCP_CFGS Panasonic_MGCP; // Panasonic MGCP协议
IPR_TOSHIBA_MEGACO_CFGS Toshiba_MEAGACO; // Toshiba MEAGACO协议
IPR_SIEMENS_H323_CFGS Siemens_H323; // Siemens H323参数
IPR_ALCATEL_CFGS ALCATEL; // Alcatel参数
IPR_MITEL_CFGS MITEL; // Mitel参数
IPR_LG_NORTEL_CFGS LG_Nortel; // LG Nortel参数
IPR_MITEL_CFGS SAMSUNG; // Samsung参数
IPR_TADICOM_MGCP_CFGS Tadicom_MGCP; // Tadicom MGCP参数
IPR_ZENITEL_CFGS ZENITEL; // Zenitel参数
IPR_NORTEL_UNISTIM_CFGS Nortel_UNIStim; // Nortel UNIStim参数

};

}IPR_MONITOR_CFGS, *PIPR_MONITOR_CFGS;
```

具体参数细节请参见 shpa3api.h 文件。

注意事项:

- 本函数只支持 SynIPAnalyzer。

相关信息:

驱动程序版本要求	SynCTI Ver.5.3.3.0
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmIPRRmvProtocol](#), [SsmIPRGetProtocol](#)

2.27.2.2 SsmIPRRmvProtocol

删除对某个协议的检测。

函数原型:

```
int SsmIPRRmvProtocol(int nBId, int nPtId)
```

参数说明:

nBId	板卡 ID
nPtId	协议类型 ID

返回值:

<0	调用失败
>=0	调用成功

功能描述:

删除对某个协议的检测。

注意事项:

- 本函数只支持 SynIPAnalyzer。

相关信息:

驱动程序版本要求	SynCTI Ver.5.3.2.0
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmIPRAddProtocol](#), [SsmIPRGetProtocol](#)

2.27.2.3 SsmIPRGetProtocol

获取对某个协议的监测参数。

函数原型:

```
int SsmIPRGetProtocol(int nPtId, PIPR_MONITOR_CFGS pParams, DWORD* dwLen)
```

参数说明:

nPtId	协议类型 ID
pParams	被监控协议的结构体参数 PIPR_MONITOR_CFGS
dwLen	pParam 参数的数据长度

返回值:

<0	调用失败
>=0	调用成功

功能描述:

获取对某个协议的监测参数。

注意事项:

- 本函数只支持 SynIPAnalyzer。

相关信息:

驱动程序版本要求	SynCTI Ver.5.3.2.0
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmIPRAddProtocol](#), [SsmIPRmvProtocol](#)

2.27.2.4 SsmIPRGetStationCount

获取当前活动的 Station 总数。

函数原型:

```
int SsmIPRGetStationCount(int nBId)
```

参数说明:

nBId	板卡 ID
------	-------

返回值:

<0	调用失败
>=0	调用成功, 返回值为当前活动的 Station 总数

功能描述:

获取当前活动的 Station 总数。

注意事项:

- 本函数只支持 SynIPAnalyzer。

相关信息:

驱动程序版本要求	SynCTI Ver.5.3.2.0
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmIPRGetStationList](#), [SsmIPRGetStationInfo](#)

2.27.2.5 SsmIPRGetStationList

获取指定数量的 Station 信息。

函数原型:

```
int SsmIPRGetStationList(int nBId, int nStationNum, PSTATION_LIST pStationList)
```

参数说明:

nBId	板卡 ID
nStationNum	需要获取信息的 Staion 数量
pStationList	指向 PSTATION_LIST 指针, 保存 Station 的信息

返回值:

<0	调用失败
>=0	调用成功

功能描述:

获取指定数量的 Station 信息。

注意事项:

- 本函数只支持 SynIPAnalyzer。

相关信息:

驱动程序版本要求	SynCTI Ver.5.3.2.0
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmIPRGetStationCount](#), [SsmIPRGetStationInfo](#)

2.27.2.6 SsmIPRGetStationInfo

获取某一个 Station 的详细信息。

函数原型:

```
int SsmIPRGetStationInfo(int nBId, int nStationId, int* nPtId, int* nTransType, int* nPort, LPSTR szIP, LPSTR szMAC)
```

参数说明:

nBId	板卡 ID
nStationId	需要获取信息的 StaionId
nPtId	呼叫控制协议的 ID
nTransType	传输协议的类型
nPort	信令端口
szIP	信令 IP, 不需要可以为 NULL
szMAC	信令 IP 对应的 MAC 地址, 不需要可以为 NULL

返回值:

<0	调用失败
>=0	调用成功

功能描述:

获取某一个 Station 的详细信息。

注意事项:

- 本函数只支持 SynIPAnalyzer。

相关信息:

驱动程序版本要求	SynCTI Ver.5.3.2.0
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmIPRGetStationCount](#), [SsmIPRGetStationList](#), [SsmIPRGetStationInfoEx](#)

2.27.2.7 SsmIPRGetStationInfoEx

获取一个 Station 的全部信息。

函数原型:

```
int SsmIPRGetStationInfoEx(int nBId,int nStationId, PStationInfoEx pSInfoEx)
```

参数说明:

nBId	板卡 ID
nStationId	需要获取信息的 Station Id
pSInfoEx	Station 信息结构体

返回值:

<0	调用失败
>=0	调用成功

功能描述:

获取一个 Station 的全部信息。

注意事项:

- 本函数只支持 SynIPAnalyzer。
- pSInfoEx 的内存需要调用者来管理。

相关信息:

驱动程序版本要求	SynCTI Ver.5.3.2.0
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmIPRGetStationCount](#), [SsmIPRGetStationList](#), [SsmIPRGetStationInfo](#)

2.27.2.8 SsmIPRSendSession

将指定的 Session 往指定的录音服务器发送。

函数原型:

```
int SsmIPRSendSession(int nChId, LPSTR szPriSlaverAddr, int nPriSlaverPort, LPSTR szSecSlaverAddr, int nSecSlaverPort)
```

参数说明:

nChId	通道 ID
szPriSlaverAddr	接收 primary 的 RTP 的录音 Slaver 的 IP 地址

nPriSlaverPort	接收 primary 的 RTP 的录音 Slaver 的端口地址
szSecSlaverAddr	接收 secondary 的 RTP 的录音 Salver 的 IP 地址
nSecSlaverPort	接收 secondary 的 RTP 的录音 Slaver 端口地址

返回值:

<0	调用失败
>=0	调用成功

功能描述:

将指定的 Session 往指定的录音服务器发送。

注意事项:

- 本函数只支持 SynIPAnalyzer。

相关信息:

驱动程序版本要求	SynCTI Ver.5.3.2.0
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmIPRStopSendSession](#), [SsmIPRGetSessionInfo](#)

2.27.2.9 SsmIPRStopSendSession

停止转发该 Session。

函数原型:

```
int SsmIPRStopSendSession(int nChId)
```

参数说明:

nChId	通道 ID
-------	-------

返回值:

<0	调用失败
>=0	调用成功

功能描述:

停止转发该 Session。

注意事项:

- 本函数只支持 SynIPAnalyzer。
- 当驱动抛出 [E_RCV_IPR_MEDIA_SESSION_STOPED](#) 事件时，会自动停止转发 RTP。因此此时再调用本函数会返回失败。

相关信息:

驱动程序版本要求	SynCTI Ver.5.3.2.0
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmIPRSendSession](#), [SsmIPRGetSessionInfo](#)

2.27.2.10 SsmIPRGetSessionInfo

获取指定 Session 的相关信息。

函数原型:

```
int SsmIPRGetSessionInfo(int nChId, pIPR_SessionInfo pSInfo)
```

参数说明:

nChId	通道 ID
pSInfo	获取到的 Session 信息, 存储在 pIPR_SessionInfo 中

返回值:

<0	调用失败
>=0	调用成功

功能描述:

获取指定 Session 的相关信息。

注意事项:

- 本函数只支持 SynIPAnalyzer。

相关信息:

驱动程序版本要求	SynCTI Ver.5.3.2.0
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmIPRSendSession](#), [SsmIPRStopSendSession](#)

2.27.2.11 SsmIPRGetMonitorType

获得目前的监控类型, 全部监控还是按 Station 监控。

函数原型:

```
int WINAPI SsmIPRGetMonitorType(int nBld)
```

参数说明:

nBld	板卡 ID
------	-------

返回值:

-1	调用失败
0	全部监控
1	按 Station 监控

功能描述:

获得目前的监控类型, 全部监控还是按 Station 监控。

注意事项:

- 本函数只支持 SynIPAnalyzer。

相关信息:

驱动程序版本要求	SynCTI Ver.5.3.2.0
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmIPRSetMonitorType](#)

2.27.2.12 SsmIPRSetMonitorType

设置监控的类型, 混合监控模式还是目标映射模式。

函数原型:

int SsmIPRSetMonitorType(int nBld, int nMType)

参数说明:

nBld	板卡 ID
nMType	监控的类型, 0: 混合监控模式; 1: 目标映射模式

返回值:

<0	调用失败
=0	调用成功

功能描述:

设置监控的类型，混合监控模式还是目标映射模式。

注意事项:

- 本函数只支持 SynIPAnalyzer。
- 在使用中需要修改监控类型并调用本函数时，驱动中所有会话信息被删除，相应的转发也停止，并删除所有 Station 信息。如果原先的监控类型为目标映射模式时，驱动会同时将监控列表清空。
- 如果希望将“目标映射模式”设置为初始监控类型，建议直接通过配置项 [MonitorType](#) 进行设置。因为有可能在调用本函数之前便监控到会话和 Station 的存在，从而耗费多余的时间和资源来处理监控到的会话。

相关信息:

驱动程序版本要求	SynCTI Ver.5.3.2.0
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmIPRGetMonitorType](#)

2.27.2.13 SsmIPRAddStationToMap

将需要监控的 Station 加入到映射列表中。

函数原型:

int SsmIPRAddStationToMap(int nBld, int nStationId, LPSTR szAddr, int nPort)

参数说明:

nBld	板卡 ID
nStationId	用户定义的 StationId
szAddr	需要监控的 Station 的 IP 地址或 MAC 地址
nPort	需要监控的 Station 的端口

返回值:

-1	调用失败
=0	调用成功

功能描述:

将需要监控的 Station 加入到映射列表中。

注意事项:

- 本函数只支持 SynIPAnalyzer。当 nPort 小于零时，szAddr 表示的是被监控的 Station 的 MAC 地址；当 nPort 大于等于零时，szAddr 表示的是被监控的 Station 的 IP 地址。其中 nPort 为零表示在由 szAddr 指定的 IP 地址上，任意端口都可监控到；当 nPort 大于零时，nPort 表示需要监控的端口号。MAC 地址格式为 “aa:bb:cc:dd:ee:ff”。

相关信息:

驱动程序版本要求	SynCTI Ver.5.3.2.0
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmIPRAddStationToMap](#), [SsmIPRmvStationFromMapEx](#), [SsmIPRAddStationToMapEx](#)

2.27.2.14 SsmIPRAddStationToMapEx

将一个 Station 加入到监控列表中。

函数原型:

```
int SsmIPRAddStationToMapEx(int nBId,int nStationId,LPSTR szAddr,int nPort,LPSTR szName,LPVOID lpReserve)
```

参数说明:

nBId	板卡 ID
nStationId	需要加入监控列表的 Station Id
szAddr	需要监控的 Station 的 IP 地址或 MAC 地址
nPort	需要监控的 Station 的端口
szName	需要监控的用户名
lpReserve	保留

返回值:

-1	调用失败
=0	调用成功

功能描述:

将一个 Station 加入到监控列表中。

注意事项:

- szAddr 和 szName 不能同时为空。当 szAddr 不为空时, szName 将被忽略, 此时函数等同于 [SsmIPRAddStationToMap](#)。当 szName 不为空时, szName 表示需要被监控的用户名。

相关信息:

驱动程序版本要求	SynCTI Ver.5.3.2.0
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmIPRmvStationFromMap](#), [SsmIPRmvStationFromMapEx](#), [SsmIPRAddStationToMap](#)

2.27.2.15 SsmIPRmvStationFromMap

将其中一个 Station 从监控列表中移除。

函数原型:

```
int SsmIPRmvStationFromMap(int nBId, LPSTR szAddr, int nPort)
```

参数说明:

nBId	板卡 ID
szAddr	需要取消监控的 Station 的 IP 地址或 MAC 地址
nPort	需要取消监控的 Station 的端口

返回值:

-1	调用失败
=0	调用成功

功能描述:

将其中一个 Station 从监控列表中移除。

注意事项:

- 本函数只支持 SynIPAnalyzer。当 nPort 小于零时, szAddr 表示需要取消监控的 Station 的 MAC 地址; 当 nPort 大于等于零时, szAddr 表示需要取消监控的 Station 的 IP 地址。其中当对 Station 的 MAC 地址取消监控时, 此函数的 nPort 不需要与函数 SsmIPRAddStationToMap 中的 nPort 相同。MAC 地址格式为 “aa:bb:cc:dd:ee:ff”。
- 本函数只适用于延时删除: 若需要删除的监控 Station 仍处于会话中, 则函数返回失败。在会话结束后, 指定的 Station 将会被自动删除。若是需要立即删除, 请参考函数 [SsmIPRRmvStationFromMapEx](#)。

相关信息:

驱动程序版本要求	SynCTI Ver.5.3.2.0
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmIPRAddStationToMap](#)、[SsmIPRRmvStationFromMapEx](#)

2.27.2.16 SsmIPRRmvStationFromMapEx

将指定的 Station 从监控列表中移除。

函数原型:

```
int SsmIPRRmvStationFromMapEx(int nBld, int nStationId, bool bDelAtOnce)
```

参数说明:

nBld	板卡 ID
nStationId	需要删除的 Station 的 ID
bDelAtOnce	是否立即移除

返回值:

<0	调用失败
=0	在延时删除中表示调用成功, 指定的 Station 将在会话结束后被删除
=1	调用成功

功能描述:

将其中一个 Station 从监控列表中移除。

注意事项:

- 本函数只支持 SynIPAnalyzer。
- 本函数有两种模式, 延时删除和立即删除。延时删除表示如果监控的 Station 仍处于会话中, 则指定的 Station 将在会话结束后被删除。立即删除表示即使监控的 Station 仍处于会话中, 函数也会立即停止监控其会话, 并停止转发 RTP, 接着删除指定的 Station。
- 返回值为零仅出现在延时删除中。

相关信息:

驱动程序版本要求	SynCTI Ver.5.3.2.0
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmIPRAddStationToMap](#)、[SsmIPRRmvStationFromMap](#)

2.27.2.17 SsmIPRChkForward

检查对应通道是否在进行 RTP 转发。

函数原型:

```
int SsmIPRChkForward(int nChId)
```

参数说明:

nChId	通道号
-------	-----

返回值:

0	没进行转发
1	进行转发
<0	函数调用失败

功能描述:

检查对应通道是否在进行 RTP 转发。

注意事项:

- 本函数只支持 SynIPAnalyzer。

相关信息:

驱动程序版本要求	SynCTI Ver.5.3.2.0
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.27.2.18 SsmIPRGetCallInfo

获取指定呼叫的信息。

函数原型:

```
int WINAPI SsmIPRGetCallInfo(int nBld, int nCallRef, pIPR_SIP_CALL_INFOEX pCallInfo)
```

参数说明:

nBld	板卡 ID
nCallRef	要查询的呼叫的索引
pCallInfo	用户提供的缓冲区，用于存储获得的呼叫信息

返回值:

-1	调用失败
=0	调用成功

功能描述:

获取指定 call 的信息。

注意事项:

- 本函数只支持 SynIPAnalyzer。

相关信息:

驱动程序版本要求	SynCTI Ver.5.3.2.1
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.27.2.19 SsmIPRGetMessageField

获取指定呼叫中某个字段的信息。

函数原型:

```
int WINAPI SsmIPRGetMessageField(int nBld, int nCallRef, LPSTR pStrFieldName, LPSTR pStrContentBuf, int* pSize)
```

参数说明:

nBld	板卡 ID
nCallRef	要查询的呼叫的索引
pStrFieldName	要获取信息的字段名, 如 CALL-ID
pStrContentBuf	用来接收信息的缓冲区
pSize	当输入时表示 pStrContentBuf 的长度, 输出时表示字段信息真实的长度

返回值:

-1	调用失败
=0	调用成功

功能描述:

获取指定呼叫中某个字段的信息。该函数具有获取整个原始 SIP 消息的功能, 当参数 pStrFieldName="SIP RAW MSG"时, 即可输出原始 SIP 消息。

注意事项:

- 本函数只支持 SynIPAnalyzer。

相关信息:

驱动程序版本要求	SynCTI Ver.5.3.2.7
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.27.3 IPR 系列录音卡用到的结构体

2.27.3.1 IPR_Addr

存放 IP 地址和端口的结构体

```
typedef union tagIPR_Addr
{
    __int64    nLLaddr; //地址+端口的组合

    struct
    {
        union
        {
            struct { unsigned char s_b1,s_b2,s_b3,s_b4; } S_un_b;
            unsigned long  S_addr;
        }; //IP 地址
        unsigned long  usPort;   //端口
    };
}IPR_Addr,*pIPR_Addr;
```

2.27.3.2 IPR_SLAVERADDR

存放 Slaver 端的地址信息的结构体

```
typedef struct tagIPR_SLAVERADDR
{
    int nRecSlaverID;    //SlaverID
    IPR_Addr ipAddr;    //Slaver 端的 IP 地址和端口
}IPR_SLAVERADDR, *PIPR_SLAVERADDR;
```

2.27.3.3 IPR_SessionInfo

Session 信息见第一章中的内容 [IPR_SessionInfo](#)。

2.27.3.4 COMPUTER_INFO

存放计算机信息

```
#define MAX_NET_ADAPTERS      6      //网络适配器的存放上限
typedef struct tagCOMPUTER_INFO
{
    char szOSVersion[128];        //操作系统型号
    int nCPUNO;                  //CPU 数量
    char szCPUInfo[256];         //CPU 信息
    ULONG ulPhysicalMemory;      //物理内存,单位为 MB (1K=1024)
    ULONG ulHardDisk;            //硬盘,单位为 GB   (1K=1000)
    int nNetAdapterNO;           //网络适配器数量
    char szMAC[MAX_NET_ADAPTERS][30];//MAC 地址
    char szIPAddr[MAX_NET_ADAPTERS][30];//IP 地址
    char szMask[MAX_NET_ADAPTERS][30]; //子网掩码
    char szGateway[MAX_NET_ADAPTERS][30]; //网关
}COMPUTER_INFO, *PCOMPUTER_INFO;
```

2.27.3.5 IPR_MONITOR_CFGS

SynIPAnalyzer 上 [SsmIPRAAddProtocol](#) 需要用到的结构体，配置监控协议用

```
{
union
{
    IPR_SCCP_CFGS CISCO;          // CISCO SCCP 协议参数
    IPR_SIP_CFGS SIP;             // SIP 协议参数
};
}IPR_MONITOR_CFGS, *PIPR_MONITOR_CFGS;
```

2.27.3.6 STATION_LIST

保存终端信息的列表

```
typedef struct tagStationList
{
    int nStationId;               //Station Id
```

```
UCHAR ucCallCtrlId; //信令协议 ID
}STATION_LIST,*PSTATION_LIST;
```

2.27.3.7 IPR_CALL_INFO

随呼叫状态事件输出的呼叫信息

```
#define IPR_MAX_CALL_ID_SIZE 120 //CallID 长度上限, 单位: 字节
typedef struct tagIPR_CALL_INFO
{
    ULONG             CallRef ;           //呼叫索引, 递增
    ULONG             CallSource ;        //呼叫方向
    ULONG             Cause ;            //原因
    char szCallerId[IPR_MAX_CALL_ID_SIZE]; //主叫名称
    char szCalledId[IPR_MAX_CALL_ID_SIZE]; //被叫名称
    char szReferredBy[IPR_MAX_CALL_ID_SIZE]; //被谁转继过来的
    char szReferTo[IPR_MAX_CALL_ID_SIZE]; //转继到哪去
}IPR_CALL_INFO, *PIPR_CALL_INFO;
```

2.27.3.8 IPR_CALL_INFOEX

详细的呼叫信息。

```
typedef struct tagIPR_CALL_INFOEX
{
    int nCallRef;           //呼叫索引
    int nStationId;        //主叫端对应的 StationId
    int nStationId2;       //被叫端对应的 StationId(i.e SIP p2p)

    unsigned int ulPrimaryIP; //主叫端 IP
    unsigned int ulSecondaryIP; //被叫端 IP

    unsigned short usPrimaryPort; //主叫端端口 little endian
    unsigned short usSecondaryPort; //被叫端端口 little endian

    ULONG             CallSource;        //呼叫方向
    ULONG             Cause;            //原因
    char szCallerId[IPR_MAX_CALL_ID_SIZE]; //主叫名称
    char szCalledId[IPR_MAX_CALL_ID_SIZE]; //被叫名称
    char szReferredBy[IPR_MAX_CALL_ID_SIZE]; //被谁转继过来的
    char szReferTo[IPR_MAX_CALL_ID_SIZE]; //转继到哪去

    char szSrcMAC[6];          //主叫 mac 地址
    char szDstMAC[6];          //被叫 mac 地址
}IPR_SIP_CALL_INFOEX, *pIPR_SIP_CALL_INFOEX;
```

2.27.3.9 StationInfoEx

SynIPAnalyzer 中 SsmIPRGetStationInfoEx 需要使用的结构体

```
typedef struct tagStationInfoEx
{
    int nStationId; //Station Id
    char szCallName[IPR_MAX_CALL_ID_SIZE]; //分机号或用户名
    LPVOID lpReserve; //保留
    IPR_Addr CallCtrlAddr; //呼叫控制地址+端口
    USHORT usSecPort; //另一控制端口，主要是给 H323 的
    UCHAR ucCallCtrlPtl; //呼叫控制协议
    UCHAR ucTransPtl; //传输协议
    UCHAR ucMacAddr[6]; //MAC 地址
}StationInfoEx,*PStationInfoEx;
```

2.28 有关语音编码格式转换的函数

2.28.1 fPcm_Mp3ConvertALaw

请参见 [fPcm_Mp3ConvertULaw](#)

2.28.2 fPcm_Mp3ConvertULaw

fPcm_Mp3ConvertALaw 将 Synway 板卡录制的 MP3 格式文件转换为 A 率格式文件, fPcm_Mp3ConvertULaw 将 MP3 格式文件转换为μ率格式文件。

函数原型:

```
int fPcm_Mp3ConvertALaw (char*szSourceFile, char*szTargetFile)
int fPcm_Mp3ConvertULaw (char*szSourceFile, char*szTargetFile)
```

参数说明:

szSourceFile	使用 Synway 板卡录制的 MP3 格式的源文件名，可包括文件的全路径。文件格式可以是无格式文件（即没有文件头），也可以是标准的 wav 文件。驱动程序不是通过检查文件扩展名，而是通过分析 szSourceFile 是否具有 wav 文件头，来判断源文件是否标准。如果不是标准 wav 文件，就将它视为无格式文件
SzTargetFile	目标文件名，可包括文件的全路径。如果目标文件不存在，驱动程序会自动创建。如果目标文件已经存在，它都将被覆盖。如果目标文件的扩展名为 wav，转换后的文件格式为标准的 wav 文件，否则为无格式文件

返回值:

1	调用成功
-1	调用失败。失败原因可以通过函数 fPcm_GetLastErrMsg 获得

功能描述:

fPcm_Mp3ConvertALaw 将 MP3 格式文件转换为 A 率格式文件, fPcm_Mp3ConvertULaw 将 MP3 格式文件转换为μ率格式文件。源文件必须是采用 Synway 板卡录制的 MP3 无格式文件或标准 wav 文件。

注意事项:

- 本函数由 ShPcmHandle.dll 提供，因此，可以在 [SsmStartCti](#) 之前使用；
- 本函数执行时，需要用到 MP3 的 ACM 解码引擎。

相关信息:

驱动程序版本要求	SynCTI Ver. 4.0 或更高
Header	ShPcmApi.h

Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数:

2.28.3 fPcm_AdpcmToAlaw

请参见 [fPcm_AdpcmToMp3](#)

2.28.4 fPcm_AdpcmToGsm

请参见 [fPcm_AdpcmToMp3](#)

2.28.5 fPcm_AdpcmToMp3

将 Synway 板卡录制的 IMA ADPCM 格式文件转换为 A 率格式文件 (fPcm_AdpcmToAlaw) 或 GSM 6.10 格式文件 (fPcm_AdpcmToGsm)，或 MP3 格式文件 (fPcm_AdpcmToMp3)。

函数原型:

```
int fPcm_AdpcmToAlaw(char*szSoureFile, char* szTargetFile)  
int fPcm_AdpcmToGsm(char*szSoureFile,char*szTargetFile)  
int fPcm_AdpcmToMp3(char*szSoureFile, char* szTargetFile)
```

参数说明:

szSoureFile	用 Synway 板卡录制的 IMA ADPCM 格式的源文件名，可包括文件的全路径。文件格式可以是无格式文件（即没有文件头），也可以是标准的 wav 文件。驱动程序不是通过检查文件扩展名，而是通过分析 szSourceFile 是否具有 wav 文件头，来判断源文件是否标准。如果不是标准 wav 文件，就将它视为无格式文件
szTargetFile	目标文件名，可包括文件的全路径。如果目标文件不存在，驱动程序会自动创建。如果目标文件已经存在，它都将被覆盖。如果目标文件的扩展名为 wav，转换后的文件格式为标准的 wav 文件，否则为无格式文件

返回值:

1	调用成功
-1	调用失败。失败原因可以通过函数 fPcm_GetLastErrMsg 获得

功能描述:

fPcm_AdpcmToAlaw 将 Synway 板卡录制的 IMA ADPCM 格式文件转换为 A 率格式文件, fPcm_AdpcmToGsm 将 IMA ADPCM 格式文件转换为 GSM 6.10 格式文件, fPcm_AdpcmToMp3 将 IMA ADPCM 格式文件转换为 Mp3 格式文件。

注意事项:

- 本函数由 ShPcmHandle.dll 提供，因此，可以在 [SsmStartCti](#) 之前使用。
- 调用本函数同时需要用到 macmcvt.dll 文件。
- fPcm_AdpcmToGsm 函数执行时，需要用到 GSM 的 ACM 编码引擎；fPcm_AdpcmToMp3 函数执行时，需要用到 MP3 的 ACM 编码引擎。

相关信息:

驱动程序版本要求	fPcm_AdpcmToAlaw 要求 SynCTI Ver. 4.0 或更高; fPcm_AdpcmToGsm 要求 SynCTI Ver. 4.7.2.0 或更高; fPcm_AdpcmToMp3 要求 SynCTI Ver. 4.7.3.1 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数: [fPcm_MemAdpcmToALAW](#), [fPcm_MemAdpcmToULAW](#)

2.28.6 fPcm_AlawToUlaw

请参见 [fPcm_UlawToAlaw](#)

2.28.7 fPcm_UlawToAlaw

将 Synway 板卡录制的 A-Law 格式文件转换为 μ-Law 格式文件 (fPcm_AlawToUlaw) 或者将 μ-Law 格式文件转换为 A-Law 格式文件 (fPcm_UlawToAlaw)。

函数原型:

```
int WINAPI fPcm_AlawToUlaw(char*szSourceFile, char*szTargetFile)  
int WINAPI fPcm_UlawToAlaw(char*szSourceFile, char*szTargetFile)
```

参数说明:

szSourceFile	用 Synway 板卡录制的 A-Law 或者 μ-Law 格式的源文件名，可包括文件的全路径。文件格式可以是无格式文件（即没有文件头），也可以是标准的 wav 文件。驱动程序不是通过检查文件扩展名，而是通过分析 szSourceFile 是否具有 wav 文件头，来判断源文件是否标准。如果不是标准 wav 文件，就将它视为无格式文件
szTargetFile	目标文件名，可包括文件的全路径。如果目标文件不存在，驱动程序会自动创建。如果目标文件已经存在，它将被覆盖。如果目标文件的扩展名为 wav，转换后的文件格式为标准的 wav 文件，否则为无格式文件

返回值:

1	调用成功
-1	调用失败。失败原因可以通过函数 fPcm_GetLastErrMsg 获得

功能描述:

fPcm_AlawToUlaw 将 Synway 板卡录制的 A-Law 格式文件转换为 μ-Law 格式文件, fPcm_UlawToAlaw 将 μ-Law 格式文件转换为 A-Law 格式文件。

注意事项:

- 本函数由 ShPcmHandle.dll 提供，因此，可以在 [SsmStartCti](#) 之前使用。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.0.0.0 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数:

2.28.8 fPcm_MemAdpcmToALAW

请参见 [fPcm_MemAdpcmToULAW](#)

2.28.9 fPcm_MemAdpcmToULAW

将 Synway 板卡录制的、保存在缓冲区中的 IMA ADPCM 格式数据转换成 A-law 或 μ-law。fPcm_MemAdpcmToALAW 将 IMA ADPCM 格式转换为 A-law, fPcm_MemAdpcmToULAW 将 IMA ADPCM 格式转换为 μ-law。

函数原型:

```
DWORD fPcm_MemAdpcmToALAW(char * pSource, DWORD dwSourceSize, char * pTarget, DWORD dwTargetSize)
```

```
DWORD fPcm_MemAdpcmToULAW(char * pSource, DWORD dwSourceSize, char * pTarget, DWORD dwTargetSize)
```

参数说明:

pSource	输入缓冲区的首地址指针
dwSourceSize	输入缓存的大小, 单位为字节, 由于 IMA ADPCM 格式的帧长度为 256 字节, 因此本参数必须是 256 字节的整数倍
pTarget	存放转换后数据的缓冲区(输出缓冲区)首地址指针, 由应用程序分配存储空间
dwTargetSize	输出缓冲区的大小。由于每帧 IMA ADPCM 格式数据转换成 A-law 或 μ -law 后, 会产生 505 字节的数据, 因此, 输出缓冲区的大小应该不小于 $dwSourceSize / 256 * 505 + 1$

返回值:

0	转换失败。失败原因可以通过函数 fPcm_GetLastErrMsg 获得
>0	转换后的语音数据字节数

功能描述:

将 Synway 板卡录制的、保存在缓冲区中的 IMA ADPCM 格式数据转换成 A-law 或 μ -law 格式。

注意事项:

- 本函数由 ShPcmHandle.dll 提供, 因此, 可以在 [SsmStartCti](#) 之前使用。

相关信息:

驱动程序版本要求	SynCTI Ver. 4.0 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数: [fPcm_AdpcmToAlaw](#)

2.28.10 fPcm_ALawConvertPcm16

请参见 [fPcm_ALawConvertMp3](#)

2.28.11 fPcm_ALawConvertPcm16_16K

请参见 [fPcm_ALawConvertMp3](#)

2.28.12 fPcm_ALawConvertPcm8

请参见 [fPcm_ALawConvertMp3](#)

2.28.13 fPCM_AlawConvertGC8

请参见 [fPcm_ALawConvertMp3](#)

2.28.14 fPcm_ALawConvertMp3

将 A-law 格式文件转换为 16 Bit PCM 格式文件 (fPcm_ALawConvertPcm16)、16 Bit 16KHz PCM 格式文件 (fPcm_ALawConvertPcm16_16K)、8 Bit 无符号 PCM 格式文件 (fPcm_ALawConvertPcm8)、G.729A 格式文件 (fPCM_AlawConvertGC8) 或 MP3 格式文件 (fPcm_ALawConvertMp3)。

函数原型:

```
int fPcm_ALawConvertPcm16(char*fnALaw, char* szTargetFile)
int WINAPI fPcm_ALawConvertPcm16_16K(char* fnALaw, char*szTargetFile)
int fPcm_AlawConvertPcm8(char* fnALaw, char* szTargetFile)
int fPCM_AlawConvertGC8(char * fnALaw,char*szTargetFile)
int fPcm_ALawConvertMp3(char*szSourceFile, char*szTargetFile)
```

参数说明:

fnALaw	A-law 格式的源文件，可包括文件的全路径。文件格式可以是无格式文件（即没有文件头），也可以是标准的 wav 文件。驱动程序不是通过检查文件扩展名，而是通过分析源文件头部是否具有 wav 文件头，来判断源文件是否是标准 wav 文件。如果不是，就将它视为无格式文件
szTargetFile	目标文件名，可包括文件的全路径。如果目标文件不存在，驱动程序会自动创建。如果目标文件已经存在，它都将被覆盖。如果目标文件的扩展名为 wav，转换后的文件格式为标准的 wav 文件，否则为无格式文件

返回值：

1	fPcm_ALawConvertPcm16 、 fPcm_ALawConvertPcm16_16K 、 fPcm_AlawConvertPcm8 或 fPcm_ALawConvertMp3 函数调用成功
0	fPCM_AlawConvertGC8 函数调用成功
-1	fPcm_ALawConvertPcm16 、 fPcm_ALawConvertPcm16_16K 、 fPcm_AlawConvertPcm8、fPCM_AlawConvertGC8 或 fPcm_ALawConvertMp3 函数调用失败。失败原因可以通过函数 fPcm_GetLastErrMsg 获得

功能描述：

fPcm_ALawConvertPcm16 将 A-law 格式文件转换为 16 Bit PCM 格式文件； fPcm_ALawConvertPcm16_16K 将 A-law 格式文件转换为 16 Bit 16KHz PCM 格式文件； fPcm_ALawConvertPcm8 将 A-law 格式文件转换为 8 Bit 无符号 PCM 格式文件； fPCM_AlawConvertGC8 将 A-law 格式文件转换为 G.729A 格式文件； fPcm_ALawConvertMp3 将 A-law 格式文件转换为 MP3 格式文件，但需安装 MP3 录音引擎才有效。

注意事项：

- 本函数由 ShPcmHandle.dll 提供，因此，可以在 [SsmStartCti](#) 之前使用。
- fPcm_ALawConvertPcm16 、 fPcm_ALawConvertPcm16_16K 、 fPcm_AlawConvertPcm8 及 fPcm_ALawConvertMp3 函数的返回值只有 1 或 -1。
- fPCM_AlawConvertGC8 函数的返回值只有 0 或 -1。
- 调用本函数同时需要用到 macmcvt.dll 文件。
- fPCM_AlawConvertGC8 函数执行时，需要用到 G729A 的 ACM 编码引擎； fPcm_ALawConvertMp3 函数执行时，需要用到 MP3 的 ACM 编码引擎。

相关信息：

驱动程序版本要求	fPcm_ALawConvertPcm16、fPcm_ALawConvertPcm8 要求 SynCTI Ver. 4.0 或更高，fPCM_AlawConvertGC8 要求 SynCTI Ver. 4.7.1.8 或更高，fPcm_ALawConvertMp3 要求 SynCTI Ver. 4.8.0.0 或更高， fPcm_ALawConvertPcm16_16K 要求 SynCTI Ver. 5.2.0.1 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数： [fPcm_Pcm16ConvertALaw](#)、[fPcm_GC8Convert](#)、[fPcm_MemAlawToPcm8](#)、[fPcm_MemAlawToPcm16](#)

2.28.15 fPcm_ULawConvertMp3

将 μ-Law 格式文件转换为 MP3 格式文件。

函数原型：

```
Int WINAPI fPcm_ULawConvertMp3(char*szSourceFile, char*szTargetFile)
```

参数说明：

szSourceFile	用 Synway 板卡录制的 μ-Law 格式的源文件名，可包括文件的全路径。文件格式可以是无格式文件（即没有文件头），也可以是标准的 wav 文件。驱动程序不是通过检查文件扩展名，而是通过分析 szSourceFile 是否具有 wav 文件头，来判断源文件是否是标准 wav 文件。如果不是，就将它视为无格式文件
szTargetFile	目标文件名，可包括文件的全路径。如果目标文件不存在，驱动程序会自动创建。如果目标文件已经存在，它将被覆盖。如果目标文件的扩展名为 wav，转换后的文件格式为标准的 wav 文件，否则为无格式文件

返回值:

1	调用成功
-1	调用失败。失败原因可以通过函数 fPcm_GetLastErrMsg 获得

功能描述:

将 Synway 板卡录制的 μ-Law 格式文件转换为 MP3 格式文件。

注意事项:

- 本函数由 ShPcmHandle.dll 提供，因此，可以在 [SsmStartCti](#) 之前使用。
- 调用本函数同时需要用到 macmcvt.dll 文件。
- 本函数执行时，需要用到 MP3 的 ACM 编码引擎。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.0.0.0 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数: [fPcm_ALawConvertMp3](#)

2.28.16 fPcm_GetDtmfAndMskFromMp3

将 DTMF 信息和 MSK 结构体信息从 MP3 格式文件中解析出来。

函数原型:

```
int fPcm_GetDtmfAndMskFromMp3(LPCSTR szSourceFile, char* pszDtmf, int* pcchDtmf, pMSK_INFO pMsk, int* pcchMsk)
```

参数说明:

szSourceFile	用 Synway 板卡录制的 16K 码流的 MP3 格式的源文件名，可包括文件的全路径。文件格式可以是无格式文件（即没有文件头），也可以是标准的 wav 文件。驱动程序不是通过检查文件扩展名，而是通过分析 szSourceFile 是否具有 wav 文件头，来判断源文件是否标准。如果不是标准 wav 文件，就将它视为无格式文件。
pszDtmf	DTMF 缓冲区的首地址指针，存放从 szSourceFile 文件中解析出来的 DTMF 字符，可以为 NULL。
pcchDtmf	DTMF 缓冲区的大小，可以为 NULL，如果 pcchDtmf 为 NULL，则函数不解析 szSourceFile 文件中包含的 DTMF；如果 pcchDtmf 不为 NULL，并且 pszDtmf 也不为 NULL，则函数开始解析 szSourceFile 文件包含的所有 DTMF；如果 pcchDtmf 不为 NULL，并且 pszDtmf 为 NULL，则函数开始计算 szSourceFile 文件包含的所有 Dtmf 个数。
pMsk	MSK 结构体缓冲区的首地址指针，用于存储从 szSourceFile 文件中解析出来的 MSK 信息结构体，可以为 NULL。
pcchMsk	MSK 结构体缓冲区的大小，可以为 NULL，如果 pcchMsk 为 NULL，则函数不解析 szSourceFile 文件中包含的 MSK 结构体；如果 pcchMsk 不为 NULL，并且 pMsk 也不为 NULL，则函数开始解析 szSourceFile 文件包含的所有 MSK 结构体；如果 pcchMsk 不为 NULL，并且 pMsk 为 NULL，则函数开始计算 szSourceFile 文件包含的所有 MSK 结构体的个数。

返回值:

1	调用成功
-1	调用失败。失败原因可以通过函数 fPcm_GetLastErrMsg 获得

功能描述:

将 DTMF 信息和 MSK 结构体信息从 MP3 格式文件中解析出来。

MSK 结构体信息如下:

```
typedef struct tagMSK_INFO
{
    unsigned char op;
    unsigned char arg;
    unsigned short unitID;
    unsigned char extra0;
    unsigned char extra1;
    unsigned char extra2;
    unsigned char extra3;
    BOOL bIsPacketDouble;
}MSK_INFO, *pMSK_INFO;
```

如果 bIsPacketDouble=TRUE，说明结构体中前面 7 个变量都解析到了值；如果 bIsPacketDouble=False，说明只解析出了结构体中的 op、arg 和 unitID。

注意事项:

- 本函数由 ShPcmHandle.dll 提供，因此，可以在 [SsmStartCti](#) 之前使用。
- 本函数执行时，需要用到 MP3 的 ACM 解码引擎。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.2.1 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数:

2.28.17 fPcm_ALawToVox

将 A-Law 格式文件转换为 vox 格式文件。

函数原型:

```
int WINAPI fPcm_ALawToVox (char *szSourceFile, char *szTargetFile)
```

参数说明:

szSourceFile	用 Synway 板卡录制的 A-Law 格式的源文件名，可包括文件的全路径。文件格式可以是无格式文件（即没有文件头），也可以是标准的 wav 文件。驱动程序不是通过检查文件扩展名，而是通过分析 szSourceFile 是否具有 wav 文件头，来判断源文件是否是标准 wav 文件。如果不是，就将它视为无格式文件
szTargetFile	目标文件名，可包括文件的全路径。如果目标文件不存在，驱动程序会自动创建。如果目标文件已经存在，它将被覆盖。

返回值:

0	调用成功
-1	调用失败。失败原因可以通过函数 fPcm_GetLastErrMsg 获得

功能描述:

将 Synway 板卡录制的 A-Law 格式文件转换为 vox 格式文件。

注意事项:

- 本函数由 ShPcmHandle.dll 提供, 因此, 可以在 [SsmStartCti](#) 之前使用。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.1.1.0 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数:

2.28.18 fPcm_MemAlawToPcm8

请参见 [fPcm_MemAlawToPcm16](#)

2.28.19 fPcm_MemAlawToPcm16

将保存在缓冲区中的 A-law 格式数据转换成 8 Bit 无符号 PCM 格式 (fPcm_MemAlawToPcm8) 或 16 Bit PCM 格式 (fPcm_MemAlawToPcm16)。

函数原型:

```
DWORD fPcm_MemAlawToPcm8 (char*pSrcBuf, DWORD dwSrcSize,char *pDstBuf,DWORD dwDstSize8)
```

```
DWORD fPcm_MemAlawToPcm16(char* pSrcBuf,DWORD dwSrcSize,char * pDstBuf,DWORD dwDstSize16)
```

参数说明:

pSrcBuf	源数据的缓冲区指针, 存放 A 率格式的语音数据
dwSrcSize	pSrcBuf 的大小, 单位为字节
pDstBuf	存放转换后数据的缓冲区指针, 由应用程序分配存储空间
dwDstSize8	pDstBuf 的大小, 单位为字节。dwDstSize8 必须大于或等于 dwSrcSize, dwDstSize16
dwDstSize16	必须大于或等于 dwSrcSize 的 2 倍

返回值:

0	转换失败。失败原因可以通过函数 fPcm_GetLastErrMsg 获得
>0	转换的字节数

功能描述:

fPcm_MemAlawToPcm8 将 A-law 转换为 8 Bit 无符号 PCM 格式, fPcm_MemAlawToPcm16 将 A-law 转换为 16 Bit PCM 格式。

注意事项:

- 本函数由 ShPcmHandle.dll 提供, 因此, 可以在 [SsmStartCti](#) 之前使用。

相关信息:

驱动程序版本要求	SynCTI Ver. 4.7.2.0 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数: [fPcm_ALawConvertPcm16](#), [fPcm_ALawConvertPcm8](#), [fPcm_MemPcm8ToAlaw](#)

2.28.20 fPcm_MemAlawToPcm16_16K

将保存在缓冲区中的 A-law 格式数据转换成 16 Bit 16KHz PCM 格式。

函数原型:

```
Int WINAPI fPcm_MemAlawToPcm16_16K(char * szSource, int nSourceLen, char * szTarget, int nTargetLen)
```

参数说明:

szSource	源数据的缓冲区指针，存放 A 率格式的语音数据
nSourceLen	szSource 的大小，单位为字节
szTarget	存放转换后数据的缓冲区指针，由应用程序分配存储空间
nTargetLen	szTarget 的大小，单位为字节。nTargetLen 必须大于或等于 nSourceLen 的 4 倍

返回值:

0	转换失败。失败原因可以通过函数 fPcm_GetLastErrMsg 获得
>0	转换的字节数

功能描述:

[fPcm_MemAlawToPcm16_16K](#) 将 A-law 转换为 16 Bit 16KHz PCM 格式。

注意事项:

- 本函数由 ShPcmHandle.dll 提供，因此，可以在 [SsmStartCti](#) 之前使用。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.2.0.1 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数: [fPcm_ALawConvertPcm16](#), [fPcm_ALawConvertPcm8](#), [fPcm_MemPcm8ToAlaw](#)

2.28.21 fPcm_MemGSMSToPcm16

将保存在缓冲区中的 GSM 格式数据转换成 16 Bit PCM 格式。

函数原型:

```
DWORD WINAPI fPcm_MemGSMSToPcm16(char *pSource, DWORD dwSourceSize, char *pTarget, DWORD dwTargetSize)
```

参数说明:

pSource	输入缓存的指针，存放 GSM 格式的语音数据
dwSourceSize	输入缓存的大小，单位为字节。大于等于 2600，小于 3250。最好为 65 的整数倍
pTarget	目标缓存指针，存放 pcm16 格式的语音数据
dwTargetSize	目标缓存大小，必须大于等于输入缓存大小的 10 倍

返回值:

0	转换失败
>0	转换的字节数

功能描述:

采用内存方式实现从 GSM 到 Pcm16 的格式转化。

注意事项:

- 本函数由 ShPcmHandle.dll 提供，因此，可以在 [SsmStartCti](#) 之前使用。
- 调用本函数同时需要用到 macmcvt.dll 文件。
- 本函数执行时需要用到 GSM 的 ACM 解码引擎。

相关信息:

驱动程序版本要求	SynCTI Ver. 4.8.0.0 或更高
Header	ShPcmApi.h

Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数：[fPcm_ALawConvertPcm16](#)，[fPcm_ALawConvertPcm8](#)，[fPcm_MemPcm8ToAlaw](#)，[fPcm_MemGSMTOPcm8](#)

2.28.22 fPcm_MemGSMTOPcm8

将保存在缓冲区中的 GSM 格式数据转换成 8 Bit 无符号 PCM 格式。

函数原型：

```
DWORD WINAPI fPcm_MemGSMTOPcm8(char *pSource, DWORD dwSourceSize, char *pTarget, DWORD dwTargetSize)
```

参数说明：

pSource	输入缓存的指针，存放 GSM 格式的语音数据
dwSourceSize	输入缓存的大小，单位为字节。大于等于 2600，小于 3250。最好为 65 的整数倍
pTarget	目标缓存指针，存放 pcm8 格式的语音数据
dwTargetSize	目标缓存大小，必须大于等于输入缓存大小的 5 倍

返回值：

0	转换失败
>0	转换的字节数

功能描述：

采用内存方式实现从 GSM 到无符号 Pcm8 的格式转化。

注意事项：

- 本函数由 ShPcmHandle.dll 提供，因此，可以在 [SsmStartCti](#) 之前使用。
- 调用本函数同时需要用到 macmcvt.dll 文件。
- 本函数执行时，需要用到 GSM 的 ACM 解码引擎。

相关信息：

驱动程序版本要求	SynCTI Ver. 5.0.3.0 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数：[fPcm_ALawConvertPcm16](#)，[fPcm_ALawConvertPcm8](#)，[fPcm_MemPcm8ToAlaw](#)，[fPcm_MemGSMTOPcm16](#)

2.28.23 fPcm_MemGSMTOUlaw

将保存在缓冲区中的 GSM 格式数据转换成 μ-law 格式。

函数原型：

```
DWORD WINAPI fPcm_MemGSMTOUlaw(char *pSource, DWORD dwSourceSize, char *pTarget, DWORD dwTargetSize)
```

参数说明：

pSource	输入缓存的指针，存放 GSM 格式的语音数据
dwSourceSize	输入缓存的大小，单位为字节。大于等于 2600，小于 3250。最好为 65 的整数倍
pTarget	目标缓存指针，存放 μ-law 格式的语音数据
dwTargetSize	目标缓存大小，必须大于等于输入缓存大小的 10 倍

返回值:

0	转换失败
>0	转换的字节数

功能描述:

采用内存方式实现从 GSM 到 μ-law 的格式转化。

注意事项:

- 本函数由 ShPcmHandle.dll 提供，因此，可以在 [SsmStartCti](#) 之前使用。
- 调用本函数同时需要用到 macmcvt.dll 文件。
- 本函数执行时，需要用到 GSM 的 ACM 解码引擎。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.0.3.0 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数：[fPcm_AlawConvertPcm16](#)，[fPcm_AlawConvertPcm8](#)，[fPcm_MemPcm8ToAlaw](#)，[fPcm_MemGSMTоКm16](#)

2.28.24 fPcm_Pcm16ConvertALaw

将 16 Bit PCM 格式文件转换为 A-law 格式的文件。

函数原型:

```
int fPcm_Pcm16ConvertALaw (char*fnPCM16, char* szTargetFile)
```

参数说明:

fnPCM16	16 Bit PCM 格式的源文件，可包括文件的全路径。文件格式可以是无格式文件（即没有文件头），也可以是标准的 wav 文件。驱动程序不是通过检查文件扩展名，而是通过分析源文件头部是否具有 wav 文件头，来判断源文件是否是标准 wav 文件。如果不是，就将它视为无格式文件
szTargetFile	目标文件名，可包括文件的全路径。如果目标文件不存在，驱动程序会自动创建。如果目标文件已经存在，它都将被覆盖。如果目标文件的扩展名为 wav，转换后的文件格式为标准的 wav 文件，否则为无格式文件

返回值:

1	调用成功
-1	调用失败。失败原因可以通过函数 fPcm_GetLastErrMsg 获得

功能描述:

将 16 Bit PCM 格式文件转换为 A-law 格式的文件。

注意事项:

- 本函数由 ShPcmHandle.dll 提供，因此，可以在 [SsmStartCti](#) 之前使用。

相关信息:

驱动程序版本要求	SynCTI Ver. 4.7.1.7 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数:

2.28.25 fPcm_Pcm16_16KConvertALaw

将 16 Bit 16KHz PCM 格式文件转换为 A-law 格式的文件。

函数原型:

```
int fPcm_Pcm16_16KConvertALaw (char*fnPCM16, char* szTargetFile)
```

参数说明:

fnPCM16	16 Bit 16KHz PCM 格式的源文件，可包括文件的全路径。文件格式可以是无格式文件（即没有文件头），也可以是标准的 wav 文件。驱动程序不是通过检查文件扩展名，而是通过分析源文件头部是否具有 wav 文件头，来判断源文件是否是标准 wav 文件。如果不是，就将它视为无格式文件
szTargetFile	目标文件名，可包括文件的全路径。如果目标文件不存在，驱动程序会自动创建。如果目标文件已经存在，它都将被覆盖。如果目标文件的扩展名为 wav，转换后的文件格式为标准的 wav 文件，否则为无格式文件

返回值:

1	调用成功
-1	调用失败。失败原因可以通过函数 fPcm_GetLastErrMsg 获得

功能描述:

将 16 Bit 16KHz PCM 格式文件转换为 A-law 格式的文件。

注意事项:

- 本函数由 ShPcmHandle.dll 提供，因此，可以在 [SsmStartCti](#) 之前使用。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.2.0.1 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数:

2.28.26 fPcm_MemPcm8ToAlaw

将保存在缓冲区中的 8 Bit 无符号 PCM 格式数据转换成 A-law 格式。

函数原型:

```
int fPcm_MemPcm8ToAlaw(char * szSource, int nSourceLen, char * szTarget, int nTargetLen)
```

参数说明:

szSource	输入缓冲区的首地址指针，存放 PCM8 率格式的语音数据
nSourceLen	szSource 的大小，单位为字节
szTarget	存放转换后的数据的缓冲区指针，由应用程序分配存储空间
nTargetLen	szTarget 的大小，必须大于或等于 nSourceLen

返回值:

0	转换失败。失败原因可以通过函数 fPcm_GetLastErrMsg 获得
>0	转换的字节数

功能描述:

将保存在缓冲区中的 8 Bit 无符号 PCM 格式数据转换成 A-law 格式。

注意事项:

- 本函数由 ShPcmHandle.dll 提供，因此，可以在 [SsmStartCti](#) 之前使用。

相关信息:

驱动程序版本要求	SynCTI Ver. 4.7.2.0 或更高
Header	ShPcmApi.h

Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数: [fPcm_MemAlawToPcm8](#), [fPcm_ALawConvertPcm8](#)

2.28.27 fPcm_MemAlawToUlaw

请参见 [fPcm_MemUlawtoAlaw](#)

2.28.28 fPcm_MemUlawtoAlaw

将保存在缓冲区中的 A-Law 格式数据转换成 μ-Law 格式 (fPcm_MemAlawToUlaw)。

将保存在缓冲区中的 μ-Law 格式数据转换成 A-Law 格式 (fPcm_MemUlawtoAlaw)。

函数原型:

```
int WINAPI fPcm_MemAlawToUlaw(char * szSource, int nSourceLen, char * szTarget, int nTargetLen)
```

```
int WINAPI fPcm_MemUlawtoAlaw(char * szSource, int nSourceLen, char * szTarget, int nTargetLen)
```

参数说明:

szSource	输入缓冲区的首地址指针, 存放 A-Law 或者 μ-Law 格式的语音数据
nSourceLen	szSource 的大小, 单位为字节
szTarget	存放转换后的数据的缓冲区指针, 由应用程序分配存储空间
nTargetLen	szTarget 的大小, 必须大于或等于 nSourceLen

返回值:

0	转换失败。失败原因可以通过函数 fPcm_GetLastErrMsg 获得
>0	转换的字节数

功能描述:

将保存在缓冲区中的 A-Law 格式数据转换成 μ-Law 格式。

将保存在缓冲区中的 μ-Law 格式数据转换成 A-Law 格式。

注意事项:

- 本函数由 ShPcmHandle.dll 提供, 因此, 可以在 [SsmStartCti](#) 之前使用。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.0.0.0 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数:

2.28.29 fPcm_MemPcm16ToAlaw

将保存在缓冲区中的 16 Bit 无符号 PCM 格式数据转换成 A-law 格式。

函数原型:

```
DWORD fPcm_MemPcm16ToAlaw(char* pSource, DWORD dwSourceSize, char* pTarget, DWORD dwTargetSize)
```

参数说明:

pSource	输入缓冲区的首地址指针, 存放 PCM16 率格式的语音数据
dwSourceSize	pSource 的大小, 单位为字节

pTarget	存放转换后的数据的缓冲区指针，由应用程序分配存储空间
dwTargetSize	pTarget 的大小，必须大于或等于 dwSourceSize 的一半

返回值:

0	转换失败。失败原因可以通过函数 fPcm_GetLastErrMsg 获得
>0	转换的字节数

功能描述:

将保存在缓冲区中的 16 Bit 无符号 PCM 格式数据转换成 A-law 格式。

注意事项:

- 本函数由 ShPcmHandle.dll 提供，因此，可以在 [SsmStartCti](#) 之前使用。

相关信息:

驱动程序版本要求	SynCTI Ver. 4.7.2.0 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数: [fPcm_MemAlawToPcm16](#), [fPcm_ALawConvertPcm16](#)

2.28.30 fPcm_MemPcm16_16KToAlaw

将保存在缓冲区中的 16 Bit 无符号 16KHz PCM 格式数据转换成 A-law 格式。

函数原型:

```
DWORD WINAPI fPcm_MemPcm16_16KToAlaw(char * pSource, DWORD dwSourceSize, char * pTarget,  
DWORD dwTargetSize)
```

参数说明:

pSource	输入缓冲区的首地址指针，存放 PCM16 率格式的语音数据
dwSourceSize	pSource 的大小，单位为字节
pTarget	存放转换后的数据的缓冲区指针，由应用程序分配存储空间
dwTargetSize	pTarget 的大小，必须大于或等于 dwSourceSize 的 1/4

返回值:

0	转换失败。失败原因可以通过函数 fPcm_GetLastErrMsg 获得
>0	转换的字节数

功能描述:

将保存在缓冲区中的 16 Bit 无符号 16KHz PCM 格式数据转换成 A-law 格式。

注意事项:

- 本函数由 ShPcmHandle.dll 提供，因此，可以在 [SsmStartCti](#) 之前使用。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.2.0.1 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数: [fPcm_MemAlawToPcm16](#), [fPcm_ALawConvertPcm16](#)

2.28.31 fPcm_GC8Convert

请参见 [fPcm_G729AConvert](#)

2.28.32 fPcm_Vox6KTo8K

请参见 [fPcm_Vox8KTo6K](#)

2.28.33 fPcm_Vox8KTo6K

实现 VOX 格式的语音文件在 6K 与 8K 采样率之间进行转换。

函数原型:

```
int fPcm_Vox6kTo8k(char *sz6kHzFile,char *sz8kHzFile)
```

```
int fPcm_Vox8KTo6K (char *sz8kHzFile,char *sz6kHzFile)
```

参数说明:

sz6kHzFile	采样率为 6kHz、编码格式为 Vox 的语音文件，没有文件头。如果不包含全路径，驱动程序在当前目录下查找文件
sz8kHzFile	采样率为 8kHz、编码格式为 Vox 的语音文件，没有文件头。如果不包含全路径，驱动程序在当前目录下查找文件

返回值:

1	调用成功
-1	调用失败。失败原因可以通过函数 fPcm_GetLastErrMsg 获得

功能描述:

fPcm_Vox6kTo8k 将采样率为 6kHz 的 Vox 文件转换成采样率为 8kHz 的 Vox 文件，fPcm_Vox8KTo6K 将采样率为 8kHz 的 Vox 文件转换成采样率为 6kHz 的 Vox 文件，。

注意事项:

- 本函数由 ShPcmHandle.dll 提供，因此，可以在 [SsmStartCti](#) 之前使用。

相关信息:

驱动程序版本要求	SynCTI Ver. 4.7.1.7 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数:

2.28.34 fPcm_ULawConvertGSM

将 μ-Law 格式文件转换为 GSM 格式的文件。

函数原型:

```
int WINAPI fPcm_ULawConvertGSM(char*szSourceFile, char*szTargetFile)
```

参数说明:

szSourceFile	μ-Law 格式的源文件，可包括文件的全路径。文件格式为标准的 wav 文件。如果不是，就将它视为无格式文件。驱动程序不是通过检查文件扩展名，而是通过分析源文件头部是否具有 wav 文件头，来判断源文件是否是标准 wav 文件
szTargetFile	目标文件名，可包括文件的全路径。如果目标文件不存在，驱动程序会自动创建。如果目标文件已经存在，它将被覆盖。如果目标文件的扩展名为 wav，转换后的文件格式为标准的 wav 文件，否则为无格式文件

返回值:

1	调用成功
-1	调用失败。失败原因可以通过函数 fPcm_GetLastErrMsg 获得

功能描述:

[fPcm_ULawConvertGSM](#) 将 μ-Law 格式文件转换为 GSM 格式的文件。

注意事项:

- 本函数由 [ShPcmHandle.dll](#) 提供，因此，可以在 [SsmStartCti](#) 之前使用。
- 调用本函数同时需要用到 [macmcvt.dll](#) 文件。
- 本函数执行时，需要用到 GSM 的 ACM 编码引擎。

相关信息:

驱动程序版本要求	要求 SynCTI Ver. 4.8.0.0 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll , macmcvt.dll

相关函数: [fPcm_Pcm16ConvertALaw](#)、[fPcm_GC8Convert](#)、[fPcm_MemAlawToPcm8](#)、[fPcm_MemAlawToPcm16](#)

2.28.35 fPCM_Pcm16ConvertG729A

将 PCM16 格式文件转换为 G.729A 格式文件。

函数原型:

```
Int WINAPI fPCM_Pcm16ConvertG729A(char*szSoureFile, char* szTargetFile)
```

参数说明:

szSoureFile	用 Synway 板卡录制的 PCM16 格式的源文件名，可包括文件的全路径。文件格式可以是无格式文件（即没有文件头），也可以是标准的 wav 文件。驱动程序不是通过检查文件扩展名，而是通过分析 szSourceFile 是否具有 wav 文件头，来判断源文件是否是标准 wav 文件。如果不是，就将它视为无格式文件
szTargetFile	目标文件名，可包括文件的全路径。如果目标文件不存在，驱动程序会自动创建。如果目标文件已经存在，它将被覆盖。如果目标文件的扩展名为 wav，转换后的文件格式为标准的 wav 文件，否则为无格式文件

返回值:

0	调用成功
-1	调用失败。失败原因可以通过函数 fPcm_GetLastErrMsg 获得

功能描述:

将 Synway 板卡录制的 PCM16 格式文件转换为 G.729A 格式文件。

注意事项:

- 本函数由 [ShPcmHandle.dll](#) 提供，因此，可以在 [SsmStartCti](#) 之前使用。
- 调用本函数同时需要用到 [macmcvt.dll](#)、[g729a.dll](#) 文件。
- 本函数执行时，需要用到 G729A 的 ACM 编码引擎。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.0.0.0 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数: [fPcm_Pcm16ConvertALaw](#)

2.28.36 fPcm_Pcm8ConvertGSM

将 PCM8 格式文件转换为 GSM 格式的文件。

函数原型:

```
Int WINAPI fPcm_Pcm8ConvertGSM(char*szSoureFile, char* szTargetFile)
```

参数说明:

szSoureFile	PCM 8 格式的源文件，可包括文件的全路径。文件格式为标准的 wav 文件。如果不是，就将它视为无格式文件。驱动程序不是通过检查文件扩展名，而是通过分析源文件头部是否具有 wav 文件头，来判断源文件是否是标准 wav 文件
szTargetFile	目标文件名，可包括文件的全路径。如果目标文件不存在，驱动程序会自动创建。如果目标文件已经存在，它将被覆盖。如果目标文件的扩展名为 wav，转换后的文件格式为标准的 wav 文件，否则为无格式文件

返回值:

0	调用成功
-1	调用失败。失败原因可以通过函数 fPcm_GetLastErrMsg 获得

功能描述:

将 PCM8 格式文件转换为 GSM 格式的文件。

注意事项:

- 本函数由 ShPcmHandle.dll 提供，因此，可以在 [SsmStartCti](#) 之前使用。
- 调用本函数同时需要用到 macmcvt.dll 文件。
- 本函数执行时，需要用到 GSM 的 ACM 编码引擎。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.1.0.0 或更高
Header	ShPcm.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数:

2.28.37 fPcm_GSMToMp3

将 GSM 格式文件转换为 MP3 格式文件。

函数原型:

```
Int WINAPI fPcm_GSMToMp3(char*szSoureFile, char* szTargetFile)
```

参数说明:

szSoureFile	用 Synway 板卡录制的 GSM 格式的源文件名，可包括文件的全路径。文件格式可以是无格式文件（即没有文件头），也可以是标准的 wav 文件。驱动程序不是通过检查文件扩展名，而是通过分析 szSourceFile 是否具有 wav 文件头，来判断源文件是否是标准 wav 文件。如果不是，就将它视为无格式文件
szTargetFile	目标文件名，可包括文件的全路径。如果目标文件不存在，驱动程序会自动创建。如果目标文件已经存在，它将被覆盖。如果目标文件的扩展名为 wav，转换后的文件格式为标准的 wav 文件；如果目标文件的扩展名为 mp3，转换后的文件格式为 mp3 格式，否则为无格式文件

返回值:

1	调用成功
-1	调用失败。失败原因可以通过函数 fPcm_GetLastErrMsg 获得

功能描述:

将 Synway 板卡录制的 GSM 格式文件转换为 MP3 格式文件。

注意事项:

- 本函数由 ShPcmHandle.dll 提供，因此，可以在 [SsmStartCti](#) 之前使用。
- 调用本函数同时需要用到 macmcvt.dll 文件。
- 本函数执行时，需要用到 GSM 的 ACM 解码引擎和 MP3 的 ACM 编码引擎。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.0.0.0 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数:

2.28.38 fPcm_G729AConvert

将 G.729A 格式的文件转换成其它语音编码格式的文件。

函数原型:

```
Int WINAPI fPcm_G729AConvert(char*szSoureFile, char* szTargetFile, int nTargetFormat)
```

参数说明:

szSoureFile	G.729A 格式的源文件，可包括文件的全路径。文件格式可以是无格式文件（即没有文件头），也可以是标准的 wav 文件。驱动程序不是通过检查文件扩展名，而是通过分析 szSourceFile 是否具有 wav 文件头，来判断源文件是否标准。如果不是标准 wav 文件，就将它视为无格式文件
szTargetFile	目标文件名，可包括文件的全路径。如果目标文件不存在，驱动程序会自动创建。如果目标文件已经存在，它都将被覆盖。如果目标文件的扩展名为 wav，转换后的文件格式为标准的 wav 文件，否则为无格式文件
nTargetFormat	目标文件的编码格式： =10001: 16 Bit PCM =6: A-law

返回值:

1	调用成功
-1	调用失败。失败原因可以通过函数 fPcm_GetLastErrMsg 获得

功能描述:

将 G.729A 格式的文件转换成其它语音编码格式的文件。

注意事项:

- 本函数由 ShPcmHandle.dll 提供，因此，可以在 [SsmStartCti](#) 之前使用。
- 调用本函数同时需要用到 macmcvt.dll、g729a.dll 文件。
- 本函数执行时，需要用到 G729A 的 ACM 解码引擎。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.0.0.0 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数: [fPCM_Pcm16ConvertG729A](#)

2.28.39 fPCM_MemULawToG729A

将保存在缓冲区中的 μ-Law 格式数据转换成 G.729A 格式。

函数原型:

Int WINAPI fPCM_MemULawToG729A(char* szSource, int nSourceLen, char* szTarget, int nTargetLen)

参数说明:

szSource	源数据的缓冲区指针，存放 μ-Law 格式的语音数据
nSourceLen	记录 szSource 的大小，单位为字节
szTarget	存放转换后数据的缓冲区指针，由应用程序分配存储空间。根据 μ-Law 格式和 G.729A 格式的大小比例，szTarget 必须是大于 nSourceLen/8 字节
nTargetLen	记录 szTarget 返回的实际大小，单位为字节

返回值:

>0	转换的字节数
0	调用失败。失败原因可以通过函数 fPcm_GetLastErrMsg 获得

功能描述:

将保存在缓冲区中的 μ-Law 格式数据转换成 G.729A 格式。

注意事项:

- 本函数由 ShPcmHandle.dll 提供，因此，可以在 [SsmStartCti](#) 之前使用。
- 调用本函数同时需要用到 macmcvt.dll、g729a.dll 文件。
- 本函数执行时，需要用到 G729A 的 ACM 编码引擎。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.1.1.0 或更高
Header	ShPcm.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数:

2.28.40 fPcm_MemMp3ToPcm16

将保存在缓冲区中的 mp3 格式数据转换成 pcm16 格式。

函数原型:

Int WINAPI fPCM_MemMp3ToPcm16 (char* szSource, int nSourceLen, char* szTarget, int nTargetLen)

参数说明:

szSource	源数据的缓冲区指针，存放 mp3 格式的语音数据
nSourceLen	记录 szSource 的大小，单位为字节
szTarget	存放转换后数据的缓冲区指针，由应用程序分配存储空间。根据 MP3 格式和 PCM16 格式的大小比例，szTarget 必须是大于 nSourceLen*16 字节
nTargetLen	记录 szTarget 返回的实际大小，单位为字节

返回值:

>0	转换的字节数
0	调用失败。失败原因可以通过函数 fPcm_GetLastErrMsg 获得

功能描述:

将保存在缓冲区中的 mp3 格式数据转换成 pcm16 格式。

注意事项:

- 本函数由 ShPcmHandle.dll 提供，因此，可以在 [SsmStartCti](#) 之前使用。
- 调用本函数同时需要用到 macmcvt.dll 文件。
- 本函数执行时，需要用到 MP3 的 ACM 解码引擎。
- 本函数不支持将声音数据分段转换，否则转换后的数据可能不完整。这是因为调用本函数时必须确保内存缓冲区中的 MP3 数据为帧的整数倍。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.1.1.0 或更高
Header	ShPcm.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数:**2.28.41 fPcm_MemG729AToPcm8**

请参见 [fPcm_MemG729AToPcm16](#)

2.28.42 fPcm_MemG729AToPcm16

将保存在缓冲区中的 G729A 格式数据转换成 PCM16 或者 PCM8 格式。

函数原型:

```
DWORD WINAPI fPcm_MemG729AToPcm16(char *pSource, DWORD dwSourceSize, char *pTarget, DWORD dwTargetSize);
```

```
DWORD WINAPI fPcm_MemG729AToPcm8(char *pSource, DWORD dwSourceSize, char *pTarget, DWORD dwTargetSize);
```

参数说明:

pSource	源数据的缓冲区指针, 存放 G729A 格式的语音数据
dwSourceSize	记录 pSource 的大小, 单位为字节
pTarget	存放转换后数据的缓冲区指针, 由应用程序分配存储空间。根据 G729A 格式和 PCM16 或 PCM8 格式的大小比例, 如果是 PCM16 格式, pTarget 必须是 dwSourceSize 的 16 倍; 如果是 PCM8 格式, pTarget 必须是 dwSourceSize 的 8 倍;
dwTargetSize	记录 pTarget 返回的实际大小, 单位为字节

返回值:

>0	转换的字节数
0	调用失败。失败原因可以通过函数 fPcm_GetLastErrMsg 获得

注意事项:

- 本函数由 ShPcmHandle.dll 提供, 因此, 可以在 [SsmStartCti](#) 之前使用。
- 调用本函数同时需要用到 macmcvt.dll、g729a.dll 文件。
- fPcm_MemG729AToPcm16 函数、fPcm_MemG729AToPcm8 函数执行时, 需要用到 G729A 的 ACM 解码引擎。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.1.1 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数:**2.28.43 fPcm_GetLastErrMsg**

获取语音编码格式转换函数调用失败的原因。

函数原型:

```
int fPcm_GetLastErrMsg(char*szMsg)
```

参数说明:

szMsg:	用来存放错误信息的指针，空间由应用程序分配
--------	-----------------------

返回值:

0	调用成功
-1	调用失败

功能描述:

获取语音编码格式转换函数调用失败的原因。

注意事项:

- 本函数由 ShPcmHandle.dll 提供，因此，可以在 [SsmStartCti](#) 之前使用。

相关信息:

驱动程序版本要求	SynCTI Ver. 4.0 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数:

2.28.44 fPcm_Close

释放本工具软件所占用的资源。

函数原型:

```
void fPcm_Close()
```

参数说明: 无

返回值: 无

功能描述:

释放本工具软件所占用的资源。

注意事项:

- 使用完本工具中所有转换函数后请务必调用此函数以释放所占用的资源。

相关信息:

驱动程序版本要求	SynCTI Ver. 4.0 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数:

2.28.45 fPcm_NotchFilter_ULAW

fPcm_NotchFilter_ULAW 将 Synway 板卡录制的 U-Law 格式文件通过点阻滤波算法过滤掉声音文件中的特定频率。

函数原型:

```
int WINAPI fPcm_NotchFilter_ULAW(LPCSTR szSourceFile, LPCSTR szTargetFile, UINT nFrequency, int nField, UINT nGain)
```

参数说明:

szSourceFile	用 Synway 板卡录制的 μ-Law 格式的源文件名，可包括文件的全路径。文件格式可以是无格式文件（即没有文件头），也可以是标准的 wav 文件。驱动程序不是通过检查文件扩展名，而是通过分析 szSourceFile 是否具有 wav 文件头，来判断源文件是否标准。如果不是标准 wav 文件，就将它视为无格式文件
szTargetFile	目标文件名，可包括文件的全路径。如果目标文件不存在，驱动程序会自动创建。如果目标文件已经存在，它将被覆盖。如果目标文件的扩展名为 wav，转换后的文件格式为标准的 wav 文件，否则为无格式文件
nFrequency	进行点阻滤波的特定频率
nField	点阻滤波频率的阈值范围，即滤波的计算频率为 nFrequency±nField
nGain	能量增益。该参数只能用来减小声音能量 (nGain>=1)，取值越大，调节越明显

返回值:

1	调用成功
-1	调用失败。失败原因可以通过函数 fPcm_GetLastErrMsg 获得

功能描述:

fPcm_NotchFilter_ULAW 将 Synway 板卡录制的 U-Law 格式文件通过点阻滤波算法过滤掉声音文件中的特定频率。

注意事项:

- 本函数由 ShPcmHandle.dll 提供，因此，可以在 [SsmStartCti](#) 之前使用。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.2.7 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数:

2.28.46 fPcm_ALawToAdpcm

fPcm_ALawToAdpcm 将 Synway 板卡录制的 A-Law 格式文件转换为 IMA ADPCM 格式文件。

函数原型:

```
in WINAPI fPcm_ALawToAdpcm(LPCSTR szSourceFile, LPCSTR szTargetFile)
```

参数说明:

szSourceFile	用 Synway 板卡录制的 A-Law 格式的源文件名，可包括文件的全路径。文件格式可以是无格式文件（即没有文件头），也可以是标准的 wav 文件。驱动程序不是通过检查文件扩展名，而是通过分析 szSourceFile 是否具有 wav 文件头，来判断源文件是否标准。如果不是标准 wav 文件，就将它视为无格式文件
szTargetFile	目标文件名，可包括文件的全路径。如果目标文件不存在，驱动程序会自动创建。如果目标文件已经存在，它将被覆盖。如果目标文件的扩展名为 wav，转换后的文件格式为标准的 wav 文件，否则为无格式文件

返回值:

0	调用成功
-1	调用失败。失败原因可以通过函数 fPcm_GetLastErrMsg 获得

功能描述:

fPcm_ALawToAdpcm 将 Synway 板卡录制的 A-Law 格式文件转换为 IMA ADPCM 格式文件。

注意事项:

- 本函数由 ShPcmHandle.dll 提供，因此，可以在 [SsmStartCti](#) 之前使用。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.3.1 或更高
----------	-------------------------

Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

相关函数:

2.28.47 fPcm_InitEx

初始化驱动内部转换语音数据所需的内存等资源。

函数原型:

```
int fPcm_InitEx(int nMaxCh)
```

参数说明:

nMaxCh	初始化 macmcvt.dll 中资源的通道数
--------	-------------------------

返回值:

0	调用失败。失败原因可以通过函数 fPcm_GetLastErrMsg 获得
1	调用成功。

功能描述:

初始化驱动转换语音数据所需的资源。

注意事项:

- 在调用 ShPcmHandle.dll 中的转换函数时, ShPcmHandle.dll 内部会使用默认通道数 100 去初始化 macmcvt.dll 资源, 此时应用可不必调用本函数。如果默认的通道数 100 不够应用程序使用, 可在应用程序调用本 dll 中转换函数前调用本函数来设定初始化的通道数。
- 如果程序同时调用 ShPcmHandle.dll 和 SHP_A3.dll, SHP_A3.dll 在调用 [SsmStartCti](#) 时会使用程序总通道数初始化 macmcvt.dll 资源, 此时应用程序不必调用本函数。
- 程序不再调用 ShPcmHandle.dll 中函数, 必须要调用 [fPcm_Close](#) 函数释放相关资源。

相关信息:

驱动程序版本要求	SynCTI Ver.5.3.3.2 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

2.28.48 fPcm_MemPcm16ToGSMEEx

将保存在缓冲区中的 pcm16 格式数据转换成 GSM 格式。

函数原型:

```
DWORD fPcm_MemPcm16ToGSMEEx(int nCh,char *pSource, DWORD dwSourceSize, char *pTarget, DWORD dwTargetSize)
```

参数说明:

nCh	转换时用的资源通道号, 应保证同一个 pcm16 源使用同一个 nCh, 不同 pcm16 源使用不同的 nCh
pSource	源数据的缓冲区指针, 存放 pcm16 格式的语音数据
dwSourceSize	pSource 的大小, 单位为字节
pTarget	存放转换后数据的缓冲区指针, 由应用程序分配存储空间
dwTargetSize	pTarget 的大小, 单位为字节。

返回值:

0	调用失败。失败原因可以通过函数 fPcm_GetLastErrMsg 获得
>0	调用成功。

功能描述:

将保存在缓冲区中的 pcm16 格式数据转换成 GSM 格式。

注意事项:

- 本函数由 ShPcmHandle.dll 提供, 因此, 可以在 [SsmStartCti](#) 之前使用。

相关信息:

驱动程序版本要求	SynCTI Ver.5.3.3.2 或更高
Header	ShPcmApi.h
Library	ShPcmHandle.lib
DLL	ShPcmHandle.dll

2.29 有关传真操作的函数（CTI 系列）

2.29.1 设置传真操作的工作参数

2.29.1.1 SsmFaxSetChSpeed

设置传真通道接收或发送传真时使用的最大速率。

函数原型:

```
int SsmFaxSetChSpeed(int ch, int speed)
```

参数说明:

ch	传真通道号
speed	最大速率, 单位为 bps。可选值为 4800bps、9600bps、14400bps、33600bps, 缺省值为 9600bps

返回值:

-1	调用失败
0	调用成功

功能描述:

设置传真通道接收或发送传真时使用的最大速率。

注意事项:

- 本函数设置的参数只对设置的传真通道起作用;
- 当设置的速率上限超过该板卡支持的速率上限时, 用该板卡支持的速率上限作为传真收/发时实际采用的速率上限。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.2.1 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmFaxGetSpeed](#), [SsmFaxSetMaxSpeed](#)

2.29.1.2 SsmFaxSetMaxSpeed

设置接收或发送传真时使用的最大速率。

函数原型:

```
void SsmFaxSetMaxSpeed(int speed)
```

参数说明:

speed	最大速率, 单位为 bps。可选值为 4800bps、9600bps、14400bps、33600bps, 缺省值为 9600bps
-------	--------------------------------------------------------------------

返回值: 无**功能描述:**

设置接收或发送传真时使用的最大速率。

注意事项:

- 本函数设置的参数对所有传真通道都起作用;
- 当设置的速率上限超过该板卡支持的速率上限时, 用该板卡支持的速率上限作为传真收/发时实际采用的速率上限。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmFaxGetSpeed](#), [SsmFaxSetChSpeed](#)

2.29.1.3 SsmFaxSetID

设置本端的识别代码。

函数原型:`int SsmFaxSetID(int ch,char *szID)`**参数说明:**

ch	传真通道编号
szID	ASCII 形式的本端识别代码字符串指针。允许长度: 少于 20 个字符

返回值:

-1	调用失败
0	调用成功

功能描述:

设置本端的识别代码。识别代码通常是本端的电话号码, 也可以是其它形式的 ASCII 字符。本端识别代码在接收或发送传真时会被传送给对端传真机。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmFaxGetID](#)

2.29.2 发送传真

2.29.2.1 SsmFaxStartSend

请参见 [SsmFaxStartSendEx](#)

2.29.2.2 SsmFaxStartSendEx

启动发送传真文件的任务。SsmFaxStartSend 用于发送整个文件，SsmFaxStartSendEx 用于发送一个文件中的指定页码段。

函数原型：

```
int SsmFaxStartSend(int ch,char *filename)  
int SsmFaxStartSendEx(int ch, char *filename, int nStartPage, int nEndPage)
```

参数说明：

ch	传真通道编号
filename	tiff 文件名, 文件的扩展名可以为 “.tif” 或 “.tiff”。有关传真通道支持的文件格式请参见第 1 章中 “ 传真 ” 部分内容
nStartPage	指定传真文件的起始页。如果 filename 中包含 N 页传真, 那么取值范围: 1~N
nEndPage	指定传真文件的结束页。如果 filename 中包含 N 页传真, 则取值范围: 1~N: 指定结束页的编号 ≥N: 发送到第 N 页 (含) -1: 发送到文件的最后一页 (即第 N 页) 注意: nEndPage 必须大于或等于 nStartPage

返回值：

0	函数调用成功
-1	函数调用失败。失败原因可能是由于该通道不处于空闲状态, 或打开文件失败

功能描述：

启动发送传真文件的任务。

发送传真文件的任务启动后, 驱动程序开始与对端传真机交换握手信息, 握手成功后将传真文件发送到对端传真机。

每当驱动程序完成一页传真的发送, 会向应用程序抛出 [E_CHG_FaxPages](#) 事件; 当驱动程序完成发送传真的任务后, 会向应用程序抛出 [E_PROC_FaxEnd](#) 事件。在发送传真文件的过程中, 应用程序可以随时调用函数 [SsmFaxCheckEnd](#) 来检查发送过程是否顺利完成, 或者调用 [SsmFaxStop](#) 函数来终止该任务。

注意事项：

- 必须在模拟中继线通道或数字中继线通道的来话或去话呼叫建立后, 才能启动发送传真文件的任务;
- 传真通道只能工作在单工方式, 即不能同时执行传真接收和发送任务。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmFaxCheckEnd](#), [SsmFaxStop](#), [SsmFaxSendMultiFile](#), [SsmFaxSendMultiFileEx](#), [SsmFaxAppendSend](#)

2.29.2.3 SsmFaxSendMultiFile

请参见 [SsmFaxSendMultiFileEx](#)

2.29.2.4 SsmFaxSendMultiFileEx

启动发送多个传真文件的任务。SsmFaxSendMultiFileEx 可指定页码范围。

函数原型：

```
int SsmFaxSendMultiFile(int ch, char * szPath, char * szFile)
```

```
int SsmFaxSendMultiFileEx(int ch, FAX_FILE_SCT *pFileNameList, int nNum)
```

参数说明:

ch	传真通道编号
szPath	传真文件 szFile 的路径。例如，如果将传真文件存放在 C 盘的 FaxFile 目录下，对于 C/C++，szPath 可写为：“C:\FaxFile\”
szFile	传真文件的文件名列表，文件的扩展名可以为 “.tif” 或 “.tiff”，相邻的文件名之间用 “;” 隔开。例如，要发送的传真文件分别为 “A.tif” 和 “B.tif”，szFile 应该写为：“A.tif;B.tif” 注意：szFile 中的传真文件必须具有相同的格式
pFileNameList	传真文件表。结构 FAX_FILE_SCT 的声明为： <pre>typedef struct tagFAX_FILE_SCT{ char szFileName[256]; //包含全路径的文件名 int nStartPage; //起始页，取值范围：1~N int nEndPage; //=1~N：指定结束页；=-1：到最后一页 int nReserve1; //保留 int nReserve2; //保留 }FAX_FILE_SCT, *PFAX_FILE_SCT;</pre> 上式中，N 为传真文件中的页面总数。例如，nStartPage=1，nEndPage=1 表示只发第 1 页；nStart=1,nEndPage=2 表示发送第 1 页和第 2 页；如果 nStart=1，nEndPage=-1 表示发送整个文件 注意：nEndPage 必须大于或等于 nStartPage
nNum	传真文件的个数

返回值:

0	调用成功
-1	调用失败，函数 SsmGetLastErrMsg 可获取失败原因

功能描述:

启动发送多个传真文件的任务。

调用本函数后，驱动程序开始与对端传真机交换握手信息，然后按照从左到右的顺序依次将 szFile 中指定的传真文件发送到对端传真机，或从 pFileNameList 表头开始发送文件，一直到最后一个文件发送完毕。

每当驱动程序完成一页传真的发送或接收，会向应用程序抛出 [E_CHG_FaxPages](#) 事件；当驱动程序完成发送传真的任务后，会向应用程序抛出 [E_PROC_FaxEnd](#) 事件。在发送多个传真文件任务的执行过程中，应用程序可以随时调用 [SsmFaxCheckEnd](#) 函数检查任务的完成情况，也可以调用 [SsmFaxStop](#) 来终止该任务。

注意事项:

- 必须在模拟中继线通道或数字中继线通道的来话或去话呼叫建立后，才能启动发送传真文件的任务；
- 传真通道只能工作在单工方式，即不能同时执行传真接收和发送任务；
- 文件路径中不能出现冒号、顿号、句号等标点符号。多文件同时发送时，要求传真文件的格式完全相同；

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmFaxCheckEnd](#), [SsmFaxStop](#), [SsmFaxStartSend](#), [SsmFaxStartSendEx](#)

2.29.2.5 SsmFaxAppendSend

向驱动程序追加一个传真文件。

函数原型:

```
int SsmFaxAppendSend (int ch,char *filename)
```

参数说明:

ch	传真通道编号
filename	传真文件名，文件的扩展名可以为“.tif”或“.tiff”。有关传真通道支持的文件格式请参见第1章中“ 传真 ”部分内容

返回值:

0	函数调用成功，驱动程序已经把指定的传真文件加入传真发送池
-1	函数调用失败，失败原因可能是由于该通道处于空闲状态，或打开文件失败

功能描述:

向驱动程序追加一个传真文件，适用于调用函数 [SsmFaxSendMultiFile](#)、[SsmFaxStartSend](#) 或 [SsmFaxStartSendEx](#) 引发的传真发送任务。

调用本函数后，驱动程序将追加的传真文件加入传真发送池，等当前的传真文件发送完成后，紧接着发送池里的文件。

注意事项:

- 必须在启动传真发送任务后，才能调用本函数；
- 驱动程序在发送传真过程中，如果被用户终止或发送失败，追加的传真文件将被丢弃。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmFaxStartSend](#), [SsmFaxStartSendEx](#), [SsmFaxSendMultiFile](#), [SsmFaxCheckEnd](#), [SsmFaxStop](#)

2.29.3 接收传真

2.29.3.1 SsmFaxStartReceive

启动一次传真接收任务。

函数原型:

```
int SsmFaxStartReceive(int ch,char *pszFileName)
```

参数说明:

ch	传真资源通道编号
pszFileName	保存传真数据的 Tiff 格式的文件名，必须以“.tif”或“.tiff”作为扩展名。如果文件已存在，原有内容将会被覆盖

返回值:

0	调用成功
-1	调用失败。失败原因可能是由于该通道不处于空闲状态，或打开文件失败

功能描述:

启动一次传真接收任务。

调用本函数后，驱动程序将自动开始与对方传真机开始握手过程，握手成功后，开始接收传真内容，并保存到

参数 `pszFileName` 指定的文件中。

每当驱动程序完成一页传真的接收，会向应用程序抛出 [E_CHG_FaxPages](#) 事件；当驱动程序完成全部传真页的接收任务后，会向应用程序抛出 [E_PROC_FaxEnd](#) 事件。在接收传真的过程中，应用程序随时可以调用 [SsmFaxStop](#) 函数终止传真接收任务。

注意事项：

- 传真通道只能工作在单工方式，即不能同时执行传真接收和发送任务。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmFaxCheckEnd](#), [SsmFaxStop](#)

2.29.4 终止传真任务

2.29.4.1 SsmFaxStop

终止传真接收或发送任务。

函数原型：

```
int SsmFaxStop(int ch)
```

参数说明：

ch	传真通道编号
----	--------

返回值：

-1	调用失败
0	调用成功

功能描述：

终止传真接收或发送任务。

本函数可以终止由函数 [SsmFaxSendMultiFile](#), [SsmFaxSendMultiFileEx](#), [SsmFaxStartSend](#), [SsmFaxStartSendEx](#) 发起的传真发送任务，也可以终止由函数 [SsmFaxStartReceive](#) 发起的传真接收任务。

本函数执行完毕后，驱动程序会向应用程序抛出 [E_PROC_FaxEnd](#) 事件。本函数的执行情况也可以调用函数 [SsmFaxCheckEnd](#) 来获取。

注意事项：

- 本函数是一个异步函数，执行的进展情况可以通过函数 [SsmFaxCheckEnd](#) 获取。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmFaxStartSend](#), [SsmFaxStartSendEx](#), [SsmFaxSendMultiFile](#), [SsmFaxSendMultiFileEx](#), [SsmFaxAppendSend](#), [SsmFaxStartReceive](#), [SsmFaxCheckEnd](#)

2.29.5 获取传真操作的信息

2.29.5.1 SsmFaxGetSpeed

获取传真接收或发送时实际使用的速率。

函数原型:

```
int SsmFaxGetSpeed(int ch)
```

参数说明:

ch	传真通道号
----	-------

返回值:

-1	函数不支持
>0	获取传真实际收/发速率。可能值为 24、48、72、96、120、144、168、192、216、240、264、288、312、336，分别表示 2400、4800、7200、9600、12000、14400、16800、19200、21600、24000、26400、28800、31200、33600bps

功能描述:

获取传真接收或发送时实际使用的速率。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmFaxSetMaxSpeed](#), [SsmFaxSetChSpeed](#)

2.29.5.2 SsmFaxCheckEnd

查询传真接收任务或传真发送任务的进展情况。

函数原型:

```
int SsmFaxCheckEnd(int ch)
```

参数说明:

ch	传真资源通道的编号
----	-----------

返回值:

0	任务尚未完成
1	任务已经完成，传真资源通道已进入“空闲”状态
2	驱动程序在执行最近一次传真接收或发送任务时发生错误，或者任务被应用程序调用 SsmFaxStop 函数终止，通道已经回到空闲状态。再次调用本函数将得到返回值 1；对于 SHF-4D 系列板卡，若该通道不再开启传真任务，则调用本函数得到的返回值仍然为 2，若该通道再次开启传真任务，则得到的返回值为 0
3	已经完成传真数据的接收或发送，正在进行断开连接的协商过程
-1	调用失败

功能描述:

查询传真接收任务或传真发送任务的进展情况。

注意事项:

- 本函数既适用于接收传真任务，也适用于发送传真任务；
- 如果函数返回值为 2，传真任务未完成，但传真数据的接收或发送可能已经完成；
- 如果应用程序采用事件模式进行编程，建议使用 [E_PROC_FaxEnd](#) 事件来取代本函数。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmFaxStartSend](#)，[SsmFaxStartSendEx](#)，[SsmFaxSendMultiFile](#)，[SsmFaxStop](#)，[SsmFaxSendMultiFileEx](#)，[SsmFaxStartReceive](#)，[SsmFaxAppendSend](#)

2.29.5.3 SsmFaxGetDcnTag

传真卡接收传真成功结束时，判断对方传真是否强制停止过。

函数原型:

```
int WINAPI SsmFaxGetDcnTag(int ch)
```

参数说明:

ch	传真通道号
----	-------

返回值:

-1	通道不支持此函数
0	传真过程完整
1	表示如果传真因收到 DCN（传真结束信息）而结束，可能是传真机强制停止引起的

功能描述:

传真卡接收传真成功结束时，判断对方传真是否强制停止过。

注意事项:

- 本函数适用于传真接收过程；
- 可调用本函数判断对方传真是否强制停止过，使用时机是在 [SsmFaxCheckEnd](#) 返回 1 后；
- 如果应用程序采用事件模式进行编程，建议使用 [E_PROC_FaxDcnTag](#) 事件来取代本函数。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.0.2.0 及以上
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：[SsmFaxCheckEnd](#)

2.29.5.4 SsmFaxGetChStateMsg

获取传真资源通道的状态。

函数原型:

```
int SsmFaxGetChStateMsg(int ch ,char *buf)
```

参数说明:

ch	传真通道编号
buf	返回状态字符串。buf 的存储空间必须由应用程序分配，并保证大于 100 个字符

返回值:

-1	调用失败
0	调用成功

功能描述:

获取传真资源通道的状态。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.29.5.5 SsmFaxGetPages

获取已经完成的传真接收或发送的页数。

函数原型:

```
int SsmFaxGetPages(int ch)
```

参数说明:

ch	传真通道编号
----	--------

返回值:

-1	调用失败
>0	已完成的传真页数

功能描述:

获取已经完成的传真接收或发送的页数。

注意事项:

- 如果传真接收或发送任务结束后, 调用本函数得到的返回值为 0, 表明本次传真接收或发送任务完全失败;
- 如果应用程序采用事件模式进行编程, 建议使用 [E_CHG_FaxPages](#) 事件来取代本函数。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.29.5.6 SsmFaxGetFailReason

获取传真资源通道传真失败的原因。

函数原型:

```
int SsmFaxGetFailReason(int ch)
```

参数说明:

ch	传真通道编号
----	--------

返回值:

-1	调用失败或者没有传真的失败信息
0	发送 Dis 失败

1	没收到 dcs 信息，对方不是传真机或者传真机没按开始键
2	接收过程中收到 dis，对方也是接收传真，属于对方错误操作
3	接收过程中没有检测到训练包
4	传真过程中，训练一直通不过
5	发送 FTT 信息失败
6	发送 cfr 失败
7	未检测到载波
8	接收载波结束后，未检测到后续帧
9	发送 MCF 失败
10	发送 PPR 失败
11	发送 CTR 失败
12	发送 RTN 失败
13	发送 ERR 失败
14	接收到不可识别的 Frame
15	接收到的页面数据不好
16	接收不到数据
17	发送方发送的页面数据错误
20	发送方没有收到 Dis 信息，对方不是传真机或者传真机没按开始键，或者传真卡没有检测到 Dis 信息
21	没有收到接收方对 train 的回应
22	发送传真数据失败
23	发送传真数据后，发送 Frame 失败
24	发送一页数据后，没有检测到接收方的响应
25	Dsp 跑飞
26	写文件失败
27	对方拒绝重发
28	应用层停止传真
29	发送 RR 连续失败
30	RR T5 超时
31	接收传真开始阶段受到干扰
32	接收到错误的 DCS
33	接收到错误的 DIS
34	传真通道与语音通道链接错误或提前调用了 SsmStopTalkWith

功能描述:

获取传真资源通道传真失败的原因。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:**2.29.5.7 SsmFaxGetID**

获取对端传真机的识别代码。

函数原型:

```
int SsmFaxGetID(int ch,char *myid)
```

参数说明:

ch	传真通道编号
----	--------

myid	存放识别码的缓冲区指针，由应用程序分配存储空间，缓冲区长度不能少于 20 字节
------	-----------------------------------------

返回值:

-1	调用失败
0	调用成功

功能描述:

获取对端传真机的识别代码。

注意事项:

- 请在握手过程完成才能调用本函数，否则无法得到正确的识别代码。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数: [SsmFaxSetID](#)

2.29.5.8 SsmFaxGetMode

获取传真通道的工作属性——发送或接受、精细模式或普通模式、ECM 方式或数据流方式。

函数原型:

```
int SsmFaxGetMode(int ch, int * pnDir, int *pnResMode, int * pnTransMode)
```

参数说明:

ch	传真通道编号
pnDir	存放发送或接收指示值 =0: 表示传真通道进行接收工作 =1: 表示传真通道进行发送工作
pnResMode	存放传真数据模式的指示值 =0: 表示普通模式 =1: 表示精细模式
pnTransMode	存放传真通道传输模式的指示值 =0: 表示数据流模式 =1: 表示 ECM 模式

返回值:

-1	通道不支持此操作
0	传真通道空闲，本函数取不到所需参数
1	函数调用成功

功能描述:

获取传真通道的工作属性——发送或接受、精细模式或普通模式、ECM 方式或数据流方式。

注意事项:

- 只有在握手过程完成之后才能调用本函数，否则无法得到正确的识别代码。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.29.5.9 SsmFaxGetAllBytes

发送传真时，获取当前传真发送页所包含的总字节数。

函数原型：

```
int SsmFaxGetAllBytes(int ch)
```

参数说明：

ch	传真通道号
----	-------

返回值：

-1	调用失败
≥0	当前传真页的总字节数

功能描述：

发送传真时，获取当前传真发送页所包含的总字节数。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.29.5.10 SsmFaxGetSendBytes

发送传真时，获取当前发送页中已经完成发送的字节数。

函数原型：

```
int SsmFaxGetSendBytes(int ch)
```

参数说明：

ch	传真通道编号
----	--------

返回值：

-1	调用失败
≥0	当前页已发送的字节数

功能描述：

发送传真时，获取当前发送页中已经完成发送的字节数。

注意事项：

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数：

2.29.5.11 SsmFaxGetRcvBytes

接收传真时，获取当前接收页中已经完成接收的字节数。

函数原型：

```
int SsmFaxGetRcvBytes(int ch)
```

参数说明:

ch	传真通道编号
----	--------

返回值:

-1	调用失败
≥0	当前页已接收的字节数

功能描述:

接收传真时，获取当前接收页中已经完成接收的字节数。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.29.5.12 SsmFaxGetCodeMode

获取传真通道的文件收发的编解码格式——MH、MR、MMR。

函数原型:

```
int SsmFaxGetCodecMode(int ch, DWORD * dwReserver)
```

参数说明:

ch	传真通道编号
dwReserver	保留

返回值:

0	MH
1	MR
2	MMR

功能描述:

获取传真通道的文件收发的编解码格式。

注意事项:

- 只有在握手过程完成之后才能调用本函数，否则无法得到正确的识别代码。

相关信息:

驱动程序版本要求	SynCTI Ver. 5.3.1.5 或更高
Header	shpa3api.h
Library	shp_a3.lib
DLL	shp_a3.dll

相关函数:

2.29.6 .tif 文件的相关操作

2.29.6.1 fBmp_ValidateFaxFile

判断.tif 文件是否为符合三汇板卡的传真格式。

函数原型:

```
int fBmp_ValidateFaxFile(char *szFile)
```

参数说明:

szFile	存放传真文件的字符串
--------	------------

返回值:

0	调用成功, 传真发送支持此.tif 文件
-1	传真发送不支持此.tif 文件, 具体原因可从 fBmp_GetErrMsg 获得
-2	打开文件失败, 具体原因可从 fBmp_GetErrMsg 获得

功能描述:

判断.tif 文件是否为符合三汇板卡的传真格式。

注意事项:**相关信息:**

驱动程序版本要求	SynCTI Ver. 4.7.3.1 或更高
Header	BmpApi.h
Library	BmpUtil.lib
DLL	BmpUtil.dll

相关函数:

2.29.6.2 fBmp_SetHeaderFormat

设置传真.tif 文件页眉属性

函数原型:

```
int fBmp_SetHeaderFormat(int nRow, int nFromX, int nFromY, char* szFrom, int nSubX, int nSubY, char* szSubject,  
int nToX, int nToY, char* szTo, int nTimeX, int nTimeY, char* szTime)
```

参数说明:

nRow	表示页眉的行数, 1~3 之间范围, 默认为 1
nFromX	表示 From 字段的起始位置, 默认为 5
nFromY	表示 From 所在的行数, 1~3 之间, 1 表示第一行, 默认为 1
szFrom	表示 From 提示字段, 默认为 “From:”, 最大可为 20byte
nSubX	表示 Subject 字段的起始位置, 默认为 270
nSubY	表示 Subject 所在的行数, 1~3 之间, 1 表示第一行, 默认为 1
szSubject	表示 Subject 提示字段, 默认为 “Sub:”, 最大可为 20byte
nToX	表示 To 字段的起始位置, 默认为 440
nToY	表示 To 所在的行数, 1~3 之间, 1 表示第一行, 默认为 1
szTo	表示 To 提示字段, 默认为 “To:”, 最大可为 20byte
nTimeX	表示 Time 字段的起始位置, 默认为 660
nTimeY	表示 Time 所在的行数, 1~3 之间, 1 表示第一行, 默认为 1
szTime	表示 Time 提示字段, 默认为 "", 最大可为 20byte

返回值:

-1	调用失败, 失败原因可从 fBmp_GetErrMsg 获得
1	操作成功

功能描述:

设置 [fBmp_AddTxtToTif](#) 所生成页眉的属性。

注意事项:

- 如果要将 Subject 放在第二行, 那么将 nRow =2, nSubY=2 即可;
- 横向位置的范围是 0~864。可根据需要调整各段数据的位置, 注意不要越出边界。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	BmpApi.h

Library	BmpUtil.lib
DLL	BmpUtil.dll

相关函数: [fBmp_AddTxtToTif](#), [fBmp_AddTxtToTif_Big](#), [fBmp_UniteTif](#)

2.29.6.3 fBmp_AddTxtToTif

增加页眉到.tif 文件上，只支持国际码编码。

函数原型:

```
int fBmp_AddTxtToTif(char szTifName, char* szFaxFrom, char*szFaxTo, char *szFaxSubject, char * szDataTime ,  
char *szTargetFile, DWORD dwReserve)
```

参数说明:

szTifName	需要加页眉的.tif 文件
szFaxFrom	传真发送方的信息，最大长度为 20byte，1 个汉字占 2byte
szFaxTo	传真接收方的信息，最大长度为 20byte，1 个汉字占 2byte
szFaxSubject	传真主题，最大占 10byte，1 个汉字占 2byte
szDataTime	时间信息，最大占 20byte
szTargetFile	要生成带页眉的 tif 文件，如果与 szTifName 重名则覆盖原文件 如果 dwReserve 赋值为 1，则前面几个参数的长度不受限制； 如果 dwReserve bit2 为 1，则 dwReserve 的高 16bit 表示需要加页眉的页； 如果 dwReserve bit3 为 1，则页眉自动编码当前页的页数。
dwReserve	

返回值:

-1	调用失败，失败原因可从 fBmp_GetErrMsg 获得
1	操作成功

功能描述:

增加页眉到.tif 文件上，并支持在页眉自动编码当前页的页数，格式为 P.%d/%d。在 szTifName 文件顶部增加 16 行、每行 1728 像素的页眉数据，把结果输出到 szTargetFile 文件。

注意事项:

- 如果 szTifName 与 szTargetFile 相同，则覆盖源文件；
- 顶部数据排列从左到右为 szFaxFrom、szSubject、szFaxTo、szDataTime。

相关信息:

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	BmpApi.h
Library	BmpUtil.lib
DLL	BmpUtil.dll

相关函数: [fBmp_SetHeaderFormat](#), [fBmp_AddTxtToTif_Big](#), [fBmp_CutTifHeader](#)

2.29.6.4 fBmp_AddTxtToTif_Big

增加页眉到.tif 文件上，而且在国际码的基础上增加支持大五码繁体字体编码。

函数原型:

```
int fBmp_AddTxtToTif_Big(char*szTifName, char*szFaxFrom, char*szFaxTo, char*szFaxSubject, Char  
*szDataTime, char *szTargetFile, DWORD dwReserve )
```

参数说明:

szTifName	需要加页眉的 tif 文件
szFaxFrom	传真发送方的信息，最大长度为 20byte，1 个汉字占 2byte
szFaxTo	传真接收方的信息，最大长度为 20byte，1 个汉字占 2byte
szFaxSubject	传真主题，最大占 10byte，1 个汉字占 2byte

szDateTime	时间信息，最大占 20byte
szTargetFile	要生成带页眉的.tif 文件，如果与 szTifName 重名则覆盖原文件
dwReserve	如果 dwReserve bit2 为 1，则 dwReserve 的高 16bit 表示需要加页眉的页； 如果 dwReserve bit3 为 1，则页眉自动编码当前页的页数。

返回值：

-1	调用失败,失败原因可从 fBmp_GetErrMsg 获得
1	操作成功

功能描述：

增加页眉到.tif 文件上，而且支持大五码繁体字体。在 szTifName 文件顶部增加 16 行，每行 1728 像素的页眉数据，把结果输出到 szTargetFile 文件。

注意事项：

- 如果 szTifName 与 szTargetFile 相同，则覆盖源文件；
- 顶部数据排列从左到右为 szFaxFrom、szSubject、szFaxTo、szDateTime；
- 函数支持操作系统支持的字体。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	BmpApi.h
Library	BmpUtil.lib
DLL	BmpUtil.dll

相关函数：[fBmp_SetHeaderFormat](#), [fBmp_AddTxtToTif](#), [fBmp_CutTifHeader](#)

2.29.6.5 fBmp_UniteTif

实现两个.tif 文件的叠加。

函数原型：

```
int fBmp_UniteTif(char *szHeadTif, char *szSourceTif, char *szTargetTif, DWORD dwReserve)
```

参数说明：

szHeadTif	用来加头的相对较小的.tif 文件，只能是单页文件
szSourceTif	用来被加头的.tif 文件，可以是多页文件
szTargetTif	生成的.tif 文件
dwReserve	保留参数

返回值：

-1	调用失败，失败原因可从 fBmp_GetErrMsg 获得
1	操作成功

功能描述：

实现两个.tif 文件的叠加，将 szHeadTif 文件加到 szSourceTif 文件上，生成 szTargetTif 文件。

注意事项：

- 如果 szSourceTif 与 szTargetTif 相同，则覆盖源文件；
- 如果 dwReserve 的第 2 Bit 为 1，表示 dwReserve 的高 16 bit 为需要叠加的页面。如果不指定该文件，则 SzSourceTif 的所有页的头上都要加 szHeadTif。

相关信息：

驱动程序版本要求	SynCTI Ver. 2.0 或更高
Header	BmpApi.h
Library	BmpUtil.lib
DLL	BmpUtil.dll

相关函数：[fBmp_SetHeaderFormat](#)

2.29.6.6 fBmp_CutTifHeader

实现.tif 文件的页眉截取功能。

函数原型:

```
int fBmp_CutTifHeader(char *Source, char*szTarget, int nHeight, int nPageNo, DWORD dwReserve)
```

参数说明:

szSource	需要将页眉截掉的.tif 源文件
szTarget	生成的目标文件, 如果文件已经存在则覆盖原有文件
nHeight	需要截掉的高度
nPageNo	.tif 文件中需要截掉页眉的指定页
dwReserve	保留参数

返回值:

-1	调用失败, 失败原因可从 fBmp_GetErrMsg 获得
1	操作成功

功能描述:

实现.tif 文件的页眉截取功能, 从.tif 文件顶部截取设定的相应部分。

注意事项:

- 如果 szSourceTif 与 szTargetFile 相同, 则覆盖源文件 szSourceTif;
- nHeight 一般在 16~40 之间, 单行的页眉为 20 左右;
- nPageNO 为指定页, 如果设置为-1, 则对所有页都截掉页眉。

相关信息:

驱动程序版本要求	SynCTI Ver. 4.7.1.5 或更高
Header	BmpApi.h
Library	BmpUtil.lib
DLL	BmpUtil.dll

相关函数: [fBmp_AddTxtToTif_Big](#), [fBmp_AddTxtToTif](#)

2.29.6.7 fBmp_GetFileAllPage

输出传真文件的页数。

函数原型:

```
int fBmp_GetFileAllPage(char *filename)
```

参数说明:

filename	存放传真文件的字符串
----------	------------

返回值:

-1	操作失败, 失败原因可能是文件打开失败, 或文件格式不对
>=0	操作成功, 输出文件的页数

功能描述:

输出传真文件 filename 的页数。

注意事项:

相关信息:

驱动程序版本要求	SynCTI Ver. 4.4.0.6 或更高
Header	BmpApi.h
Library	BmpUtil.lib
DLL	BmpUtil.dll

相关函数:

2.29.6.8 fBmp_SetHeaderFormatA

设置传真.tif文件页眉属性，确定页数的位置和前缀等。

函数原型：

```
int WINAPI fBmp_SetHeaderFormatA(int nRow, int nFromX, int nFromY, LPCSTR szFrom, int nSubX, int nSubY,  
LPCSTR szSubject, int nToX, int nToY, LPCSTR szTo, int nTimeX, int nTimeY, LPCSTR szTime, int nPageX, int  
nPageY, LPCTSTR szPage)
```

参数说明：

nRow	表示页眉的行数，1~3之间范围，默认为1
nFromX	表示 From 字段的起始位置，默认为 5
nFromY	表示 From 所在的行数，1~3之间，1 表示第一行， 默认为 1
szFrom	表示 From 提示字段， 默认为 “From:”， 最大可为 20byte
nSubX	表示 Subject 字段的起始位置， 默认为 270
nSubY	表示 Subject 所在的行数，1~3之间，1 表示第一行， 默认为 1
szSubject	表示 Subject 提示字段， 默认为 “Sub:”， 最大可为 20byte
nToX	表示 To 字段的起始位置， 默认为 440
nToY	表示 To 所在的行数，1~3之间，1 表示第一行， 默认为 1
szTo	表示 To 提示字段， 默认为 “To:”， 最大可为 20byte
nTimeX	表示 Time 字段的起始位置， 默认为 660
nTimeY	表示 Time 所在的行数，1~3之间，1 表示第一行， 默认为 1
szTime	表示 Time 提示字段， 默认为 "", 最大可为 20byte
nPageX	表示页编码的水平位置起始， 默认 780
nPageY	表示页编码的垂直起始位置， 默认 1
szPage	表示页编码的前缀提示字段， 默认 NULL

返回值：

-1	调用失败，失败原因可从 fBmp_GetErrMsg 获得
1	操作成功

功能描述：

设置 [fBmp_AddTxtToTif](#) 所生成页眉的属性，用于设置页数的位置和前缀。

注意事项：

- 如果要将 Subject 放在第二行，那么设置 nRow =2, nSubY=2 即可；
- 横向位置的范围是 0~864。可根据需要调整各段数据的位置，注意不要越出边界。

相关信息：

驱动程序版本要求	SynCTI Ver. 5.3.2.7 或更高
Header	BmpApi.h
Library	BmpUtil.lib
DLL	BmpUtil.dll

相关函数：[fBmp_AddTxtToTif](#), [fBmp_AddTxtToTif_Big](#), [fBmp_UniteTif](#)

3 SynCTI 驱动程序的配置

3.1 系统配置文件 ShConfig.ini

使用 Synway 板卡时，必须对板卡及驱动程序进行正确和必要的配置后，应用系统才能正常工作。

SynCTI 驱动程序的配置信息采用文本方式保存在 INI 文件中。在正确完成 SynCTI 驱动程序的安装后，安装程序会在安装目录下生成一个文件名称为 ShConfig.ini 的配置文件范本。开发人员可以通过文本编辑软件直接编写或修改配置文件，也可以通过 Synway 公司提供的系统配置工具软件来修改配置。

配置文件中，配置信息以节(Section)为单位，每个节可以包含一个或多个配置项(Item)。节的名称出现在一对方括号中，后续的每一个文本行就是一个配置项。节的作用范围为从本节的后面一行开始，直到下一个节的上面一行止。每一个配置项都具有“x=y”的形式，等号左边是配置项的名称，右边是配置内容。

3.1.1 必须配置项

3.1.1.1 设置板卡总数

3.1.1.1.1 TotalBoards

配置项	TotalBoards
节	[SystemConfig]
书写格式	TotalBoards=N
取值范围	N 等于系统中安装的 Synway 板卡数量。
功能描述	设置系统中安装的 Synway 板卡总数。

3.1.1.1.2 WhoSupplySysClock

配置项	WhoSupplySysClock
节	[SystemConfig]
书写格式	WhoSupplySysClock=n

取值范围	<p>n=-1: 将全部 Synway 板卡均设置为从卡； n=N: 将全部 Synway 板卡均设置为主卡； 0≤n≤N-1: 将逻辑编号为 n 的板卡设置为主卡，其余板卡均为从卡。 注：N 为配置项 TotalBoards 的设定值。</p>
功能描述	<p>设置应用系统中的 Synway 板卡的主卡。更多信息请参见第 1 章中“配置应用系统的时钟”部分内容。</p>

3.1.1.2 设置通道编号

3.1.1.2.1 TotalAppCh

配置项	TotalAppCh
节	[AppChToBoardChTable]
书写格式	TotalAppCh=M
取值范围	M 为系统中全部板卡上的通道总数。
功能描述	设置系统中的通道总数。

3.1.1.2.2 AppCh

配置项	AppCh
节	[AppChToBoardChTable]
书写格式	<p>格式 1: AppCh[i_s]=k,j_s...j_e 格式 2: AppCh[i]=k,j 上式中， i: 应用程序使用的通道号，通常从 0 开始编号 i_s: 应用程序使用的起始通道号 k: 需要建立映射的板卡 BoardID j: 通道在板卡内部的编号 j_s: 通道在板卡内部的起始编号 j_e: 通道在板卡内部的结束编号</p>
取值范围	
功能描述	建立应用程序的通道编号与板卡通道编号之间的映射。格式 1 用于批量建立映射，格式 2 用于逐个建立。

3.1.1.3 设置板卡信息

3.1.1.3.1 BoardModel

配置项	BoardModel
节	[BoardId=x]
书写格式	BoardModel=s
取值范围	s 是表示板卡型号的字符串。可用的型号请参见第 1 章中 Synway 板卡的分类 部分。
功能描述	设置板卡型号。
使用范例	BoardModel=SHT-16B-CT/PCI

3.1.1.3.2 BoardSerialNumber

配置项	BoardSerialNumber
节	[BoardId=x]
书写格式	BoardSerialNumber=N
取值范围	N 是板卡的序列号。
功能描述	设置板卡的序列号，具体取值请参见相关的硬件说明书。
使用范例	BoardSerialNumber=11

3.1.1.4 SHD/DTP 系列板卡

3.1.1.4.1 设置板卡复位功能

3.1.1.4.1.1 ResetBoardOnClose

配置项	ResetBoardOnClose
节	[BoardId=x]
书写格式	ResetBoardOnClose =n

取值范围	n=0: 不复位; n=1: 复位（缺省值）。
功能描述	设置在关闭板卡时是否将板卡复位。

3.1.1.4.2 设置数字中继线的工作参数

3.1.1.4.2.1 PcmNumber

请参见 [CRC-4](#)

3.1.1.4.2.2 PcmSSx

请参见 [CRC-4](#)

3.1.1.4.2.3 PcmClockMode

请参见 [CRC-4](#)

3.1.1.4.2.4 PcmLinkType

请参见 [CRC-4](#)

3.1.1.4.2.5 IsdnAutoBuildLink

请参见 [CRC-4](#)

3.1.1.4.2.6 CRC-4

配置项	PcmNumber PcmSSx PcmClockMode PcmLinkType IsdnAutoBuildLink CRC-4
节	[BoardId=x]

书写格式	<p>PcmNumber=M PcmSSx[i]=n PcmClockMode[i]=m PcmLinkType[i]=k IsdnAutoBuildLink[i]=b CRC-4[i]=c</p> <p>//注：要求 SynCTI Ver. 4.7.1.5 及以上版本</p>
取值范围	<p>M: 板卡上数字中继线的总数，与板卡型号有关，详细信息请参见第 1 章中“SHD 系列板卡简介”部分内容。</p> <p>i: 数字中继线在板卡上的物理编号，更多信息请参见相关手册的说明。取值范围：$0 \leq i \leq M$。</p> <p>n: 选择信令方式。取值范围：</p> <ul style="list-style-type: none"> n=0: ISDN 协议（用户侧）； n=1: No.1 信令（SS1）； n=2: ISDN 协议（网络侧）； n=3: DASS2 协议； n=4: DPNSS 协议； n=7: SS7 信令（SS7）。 <p>m: 设置本数字中继线的时钟模式，具体为：</p> <ul style="list-style-type: none"> m=0: 主时钟、线路同步方式（板卡必须为主卡）； m=1: 主时钟、自振荡方式（板卡必须为主卡）； m=2: 从时钟。 <p>k: 选择线路的传输类型。取值范围：</p> <ul style="list-style-type: none"> k=0: 120 欧姆双绞线； k=1: 75 欧姆同轴电缆。 <p>b: 设置 ISDN 信令是否使用自动建链功能</p> <ul style="list-style-type: none"> b=0: 关闭（缺省值） b=1: 启用（如果启用自动建链功能，请将[ISDN]节的手动建链配置项 UserSendEstablish 和 NetSendEstablish 均置为 0。） <p>注：当 TEI 值在 64—126 范围时，自动建链只能由用户侧发起，不能由网络侧发起。此时，网络侧需配置该配置项为 b=0。</p> <p>c: CRC-4 校验功能的控制开关：</p> <ul style="list-style-type: none"> c=0: 关闭； c=1: 启用（缺省值）。

功能描述	<p>PcmNumber 设置板卡上数字中继线的总数。</p> <p>PcmSSx 设置数字中继线的信令类型。</p> <p>PcmClockMode 设置时钟的工作模式，更多信息请参见第 1 章中“配置应用系统的时钟”部分内容。</p> <p>PcmLinkType 设置线路的传输类型。</p> <p>CRC-4 设置板卡上数字中继线的 CRC-4 校验功能。</p>
注意事项	<ul style="list-style-type: none"> ● 如果 PcmNumber 的设定值大于 0，配置项 PcmSSx、PcmClockMode、PcmLinkType 应从 i=0 开始，重复 M 次，即需要对每一条物理数字中继线进行配置； ● 这些配置项只适用于 SHD/DTP 系列板卡。 ● 目前只有 DTP 系列板卡支持 DASS2 和 DPNSS 协议。
使用范例	<p>假设系统中安装了 2 片 SHD-60A-CT/PCI/SS7 板卡，0#板卡使用 SS7 信令，采用 120 欧姆的双绞线接入，它的 0#数字中继线采用线路同步方式为应用程序提供主时钟；1#板卡使用 ISDN 协议（用户侧），采用 75 欧姆同轴电缆接入，则配置项的设置为：</p> <pre>[BoardId=x] PcmNumber=4 PcmSSx[0]=7 //SS7 信令方式 PcmSSx[1]=7 PcmSSx[2]=0 //ISDN 用户侧 PcmSSx[3]=0 PcmClockMode[0]=0 //主时钟，线路同步方式 PcmClockMode[1]=2 //必须设置为从时钟 PcmClockMode[2]=2 PcmClockMode[3]=2 PcmLinkType[0]=0 //120 欧姆双绞线 PcmLinkType[1]=0 PcmLinkType[2]=1 //75 欧姆同轴电缆 PcmLinkType[3]=1</pre>

3.1.1.4.2.7 loopback

配置项	loopback
节	[BoardId=x]
书写格式	loopback[pcm]=c

取值范围	pcm: 数字中继线在板卡上的物理编号, 更多信息请参见相关手册的说明。取值范围: $0 \leq i \leq M$ 。 c: 数字中继线环回(loopback)功能的控制: =0x1 : 发送数据从成帧器环回到接收端(LOOP_FRAME); =0x2 : LIU 发送器环回到 LIU 接收器(LOOP_LOCAL); =0x4 : LIU 模拟环回(LOOP_ANALOG, 仅适用于 D、E 型卡); =0x10: LIU 接收器解码后环回到 LIU 发送器(LOOP_REMOTE); =0x20: 有效载荷数据环回到发送端(LOOP_PAYLOAD)。
功能描述	设置中继线环回, 用于诊断或调试。
注意事项	<ul style="list-style-type: none">● 要求 SynCTI Ver. 5.1.0.0 及以上版本;● 该配置项只适用于 SHD 系列板卡。
使用范例	[BoardId=x] Loopback[0]=0x20

3.1.1.4.2.8 SpySyncAndCCS

配置项	SpySyncAndCCS
节	[BoardId=x]
书写格式	SpySyncAndCCS=n
取值范围	n=0: 关闭同步和信令时隙数据采集模式; n=1: 开启同步和信令时隙数据采集模式, 板卡正常使用时请勿开启此开关。
功能描述	此功能仅在线路出现同步或信令异常时使用, 通过添加配置项 SpySyncAndCCS=1 开启同步和信令时隙数据采集模式后, 使用驱动自带测试程序 TEST.EXE, 在每个 PCM 的第 0 通道录制板卡同步时隙原始数据, 在每个 PCM 的第 1 通道录制板卡接收的信令时隙的原始数据, 在每个 PCM 的第 2 通道录制板卡发送的信令时隙的原始数据, 录制原始数据时选择相应通道, 选择 A 率, 录制 PCM 格式声音文件即可。

注意事项	● 对于 ISDN 线路，使用此功能前需要通过配置项 UserCrcMode 和 NetCrcMode 打开 CRC 校验功能；
	● 对于 DTP C 型卡，如果要录制 T1 线路的同步数据，需要修改配置项 DefaultVoiceFormat 为 6，并使用 A 率格式来录制声音文件；如果要录制 T1（或 E1）线路的信令数据，需要修改配置项 DefaultVoiceFormat 为 7，并使用 U 率格式来录制声音文件；
	● 对于 1 号信令，板卡通过专用接口芯片发送信令数据，因此无法录制发送信令的原始数据；
	● 使用此功能时，请在我方技术支持指导下进行，原始数据文件录制完毕后须提供给我方开发人员进行分析；
	● 要求 SynCTI Ver. 5.3.0.3 及以上版本；
	● 该配置项只适用于部分 SHD/DTP 系列板卡，详见下表。

板卡支持的原始数据类别列表：

板卡型号	同步时隙原始数据	接收的信令时隙原始数据	发送的信令时隙原始数据
SHD-30A-CT/PCI/SS1	√	√	✗
SHD-30A-CT/PCI/ISDN	√	√	✗
SHD-30A-CT/PCI/SS7	√	√	✗
SHD-30A-CT/PCI/FJ	√	√	-
SHD-60A-CT/PCI/SS1	√	√	✗
SHD-60A-CT/PCI/ISDN	√	√	✗
SHD-60A-CT/PCI/SS7	√	√	✗
SHD-60A-CT/PCI/FJ	√	√	-
SHD-120A-CT/PCI/SS1	√	√	✗
SHD-120A-CT/PCI/ISDN	√	√	✗
SHD-120A-CT/PCI/SS7	√	√	✗
SHD-30B-CT/PCI/FJ	√	√	-
SHD-60B-CT/PCI/FJ	√	√	-
SHD-30C-CT/PCI	√	√	✗
SHD-30C-CT/PCI/FAX	√	√	✗
SHD-60C-CT/PCI	√	√	✗
SHD-60C-CT/PCI/FAX	√	√	✗
SHD-120D-CT/PCI	√	√	√
SHD-120D-CT/PCI/EC	√	√	√
SHD-240D-CT/PCI	√	√	√
SHD-240D-CT/PCI/EC	√	√	√
SHD-30E-CT/PCI(SSW)	✗	√	✗
SHD-30E-CT/PCI/EC(SSW)	✗	√	✗
SHD-30E-CT/PCI/FAX(SSW)	✗	√	✗
SHD-60E-CT/PCI(SSW)	✗	√	✗
SHD-60E-CT/PCI/EC(SSW)	✗	√	✗
SHD-60E-CT/PCI/FAX(SSW)	✗	√	✗
SHD-120E-CT/PCI(SSW)	√	√	√
SHD-120E-CT/PCI/EC(SSW)	√	√	√
SHD-120E-CT/PCI/FAX(SSW)	√	√	√

SHD-240E-CT/PCI(SSW)	√	√	√
SHD-240E-CT/PCI/EC(SSW)	√	√	√
SHD-240E-CT/PCI/FAX(SSW)	√	√	√
SHD-30E-CT/PCIe	✗	√	✗
SHD-30E-CT/PCIe/EC	✗	√	✗
SHD-30E-CT/PCIe/FAX	✗	√	✗
SHD-60E-CT/PCIe	✗	√	✗
SHD-60E-CT/PCIe/EC	✗	√	✗
SHD-60E-CT/PCIe/FAX	✗	√	✗
SHD-120E-CT/PCIe	√	√	√
SHD-120E-CT/PCIe/EC	√	√	√
SHD-120E-CT/PCIe/FAX	√	√	√
SHD-240E-CT/PCIe	√	√	√
SHD-240E-CT/PCIe/EC	√	√	√
SHD-240E-CT/PCIe/FAX	√	√	√
SHD-240E-CT/PCIe/VAR	√	√	√
DTP-30C/PCI	√	√	-
DTP-30C/PCI+	√	√	-
DTP-60C/PCI	√	√	-
DTP-60C/PCI+	√	√	-
DTP-120C/PCI	√	√	-
DTP-120C/PCI+	√	√	-
DTP-30C/PCIe	√	√	-
DTP-30C/PCIe+	√	√	-
DTP-60C/PCIe	√	√	-
DTP-60C/PCIe+	√	√	-
DTP-120C/PCIe	√	√	-
DTP-120C/PCIe+	√	√	-

图例: √: 支持

✗: 不支持

-: 无效

3.1.1.4.2.9framing

请参见 [syncc](#)。

3.1.1.4.2.10 coding

请参见 [syncc](#)。

3.1.1.4.2.11 syncc

配置项	framing coding syncc
-----	-------------------------------------------------

节	[BoardId=x]
书写格式	framing[pcm]=c coding[pcm]=d syncc[pcm]=e
取值范围	<p>pcm: 数字中继线在板卡上的物理编号, 更多信息请参见相关手册的说明。取值范围: $0 \leq i \leq M$。</p> <p>c: T1 线路帧格式设置:</p> <ul style="list-style-type: none"> 2: ESF framing 3: D4 framing 8: SLC-96 framing <p>d: T1 线路编码格式设置:</p> <ul style="list-style-type: none"> 2: B8ZS coding 5: AMI coding <p>e: syncc 设置:</p> <p>In D4 Framing Mode:</p> <ul style="list-style-type: none"> 0 = search for Ft pattern, then search for Fs pattern 1 = cross couple Ft and Fs pattern <p>In ESF Framing Mode:</p> <ul style="list-style-type: none"> 0 = search for FPS pattern only 1 = search for FPS and verify with CRC-6
功能描述	设置 T1 线路帧格式与帧编码方式。
注意事项	<ul style="list-style-type: none"> ● 本配置项要求 SynCTI Ver. 5.3.1.3 及以上版本; ● 本配置项只适用于 T1 线路。

3.1.1.4.3 设置 PCM 的逻辑编号

3.1.1.4.3.1 TotalPcm

请参见 [Pcm](#)

3.1.1.4.3.2 Pcm

配置项	TotalPcm Pcm
节	[PcmInfo]

书写格式	TotalPcm=M Pcm[m]=k,j
取值范围	<p>M: 应用系统中数字中继线的总数，必须小于或等于系统中的物理 PCM 线路的总数。</p> <p>m: 应用程序中的数字中继线的逻辑编号，必须从 0 开始编号，取值范围: $0 \leq m < M$ (M 为配置项 TotalPcm 的设定值)。</p> <p>k: 板卡的逻辑编号。取值范围:</p> <ul style="list-style-type: none"> ◆ $0 \leq k < N$: N 为配置项 TotalBoards 的设定值。将应用程序的数字中继线的逻辑编号与实际的物理数字中继线进行绑定。 ◆ k=-1: 将应用程序的数字中继线的逻辑编号与虚电路进行绑定，只适用于 SS7 信令的 TUP 协议和 ISUP 协议，要求驱动程序的版本为 4.7.1.0 或更高。有关虚电路的更多信息请参见第 1 章中“使用 SS7 服务器的虚电路编程接口”部分内容。 <p>j: 如果 $k \neq -1$, j 表示数字中继线的物理编号，必须是物理上存在的 PCM; 如果 $k = -1$, j 所对应的数字中继线可以是虚拟的。</p>
功能描述	<p>TotalPcm 设置应用程序中数字中继线的总数。</p> <p>Pcm 建立数字中继线的逻辑编号与物理编号之间的映射关系。</p>
注意事项	本配置项只适用于 SHD、DTP 系列板卡。
使用范例	<p>假设系统中安装了 2 片 SHD-60A-CT/PCI/ISDN 板卡，则需要按照下列方法建立数字中继线的逻辑编号与物理编号之间的映射:</p> <p>[PcmInfo]</p> <pre> TotalPcm=4 Pcm[0]=0,0 //将 0#板卡上的 0#数字中继线设置为应用程序的 0#数字中继线 Pcm[1]=0,1 //将 0#板卡上的 1#数字中继线设置为应用程序的 1#数字中继线 Pcm[2]=1,0 //将 1#板卡上的 0#数字中继线设置为应用程序的 2#数字中继线 Pcm[3]=1,1 //将 1#板卡上的 1#数字中继线设置为应用程序的 3#数字中继线 </pre>

3.1.1.4.3.32PcmIn1Line

配置项	2PcmIn1Line
节	[BoardId=x]
书写格式	2PcmIn1Line=m
取值范围	<p>m=0 (缺省值)：推荐 UMCT 智能交换机使用后出线板 RSD081 (8 口标准 RJ48C)，此时接口顺序和 PCM 编号顺序是一致的。对于后出线板 RSD082 (4 口复用 RJ48C)，此时第一个接口对应的 PCM 编号是 0、4，第二个对应 1、5，第三个对应 2、6，第四个对应 3、7。</p> <p>m=1：推荐 UMCT 智能交换机使用后出线板 RSD082 (4 口复用 RJ48C)，此时第一个接口对应的 PCM 编号是 0、1，第二个对应的是 2、3，第三个对应 4、5，第四个对应 6、7。对于后出线板 RSD081 (8 口标准 RJ48C)，此时第一到第八个接口依次对应的 PCM 序号是：0、2、4、6、1、3、5、7。</p>
功能描述	交换机上数字卡和接口分离，且交换机本身不能区分 RSD081 和 RSD082 这两种后出线板。因此合理使用此配置项可使接口上 PCM 顺序更符合一般使用习惯。

注意事项	本配置项适用于 UMCT 智能交换机上的 240 路 E 型数字卡。
------	------------------------------------

3.1.1.4.4 设置 SpyPCM 编号 (DTP 系列)

3.1.1.4.4.1 TotalSpyPcm

请参见 [SpyPcm](#)

3.1.1.4.4.2 SpyPcm

配置项	TotalSpyPcm SpyPcm
节	[SpyPcm]
书写格式	TotalSpyPcm=M SpyPcm[m]=Pcm[j],Pcm[k]
取值范围	M: 应用系统中需要监控的数字中继线的总数。 m: SpyPcm 的编号, 从 0 开始编号, $0 \leq m \leq M$ 。 j,k: 数字中继线的逻辑编号, 由[PcmInfo]节中的配置项 Pcm 中设置, 更多信息请参见第 1 章中“ 数字中继线 ”部分内容。建议用板卡上的 2 条编号连续的数字中继线接入端口来监控一条 SpyPCM, 因此, 在大多数情况下, $k=j+1$ 。注意: j 和 k 必须位于同一片板卡上。
功能描述	TotalSpyPcm 设置应用系统中需要监控的数字中继线的总数。 SpyPcm 设置用板卡上的哪 2 条数字中继线接入电路来处理被监控的数字中继线, 即将第 m 条 SpyPCM 绑定到第 j 条和第 k 条数字中继线处理电路。 更多信息请参见第 1 章中“ DTP 系列板卡 ”部分内容。
注意事项	本配置项只适用于 DTP 系列板卡。
使用范例	假设系统中安装了 1 片 SHD-60A-CT/PCI/FJ 板卡, 用来采集 2 条使用 75 欧姆同轴电缆的 E1 中继线上的 60 个语音信道 (30x2) 和 2 个 ISDN PRI 信令信道, 则需要按照下列方法进行配置: [BoardId=x] PcmNumber=4 PcmSSx[0]=0 //ISDN 用户侧 PcmSSx[1]=0 PcmSSx[2]=0 PcmSSx[3]=0 PcmClockMode[0]=0 //主时钟, 线路同步方式

	<pre> PcmClockMode[1]=2 //必须设置为从时钟 PcmClockMode[2]=2 PcmClockMode[3]=2 PcmLinkType[0]=1 //75 欧姆同轴电缆 PcmLinkType[1]=1 PcmLinkType[2]=1 PcmLinkType[3]=1 [PcmInfo] TotalPcm=4 Pcm[0]=0,0 //将 0#板卡上的 0#数字中继线设置为应用程序的 0#数字中继线 Pcm[1]=0,1 //将 0#板卡上的 1#数字中继线设置为应用程序的 1#数字中继线 Pcm[2]=0,2 //将 0#板卡上的 2#数字中继线设置为应用程序的 2#数字中继线 Pcm[3]=0,3 //将 0#板卡上的 3#数字中继线设置为应用程序的 3#数字中继线 [SpyPcm] TotalSpyPcm=2 SpyPcm[0]=Pcm[0],Pcm[1] //用 0#、1#PCM 来采集 0#SpyPCM 上的信息 SpyPcm[1]= Pcm[2],Pcm[3] //用 2#、3#PCM 来采集 1#SpyPCM 上的信息 [AppSpyCICTable] TotalAppSpyCIC=60 AppSpyCIC[0]=SpyPcm[0],0..29 //设置 SpyCIC 逻辑编号为 0~29 的物理位置 AppSpyCIC[30]=SpyPcm[1],0..29 //设置 SpyCIC 逻辑编号为 30~59 的物理位置 </pre>
--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3.1.1.4.3 SpyT1TransE1Line

配置项	SpyT1TransE1Line
节	[SpyPcm]
书写格式	SpyT1TransE1Line[k]=b
取值范围	k: 监控 Pcm 序号; b=0: 被监控的线路为普通的 E1 线路; b=1: 被监控的线路为 T1 转 E1 设备转换过来的线路。
功能描述	通路编码是采用 (E1: 1-30 或 T1: 1-23) 连续编码还是采用 (E1: 1-15, 17-31) 不连续编码。
注意事项	<ul style="list-style-type: none"> ● 本配置项只适用于 ISDN 线路的监控; ● 本配置项只适用于 DTP 系列板卡。

3.1.1.4.4 SpyDefaultGetCallerIdType

配置项	SpyDefaultGetCallerIdType
节	[SpyPcm]
书写格式	SpyDefaultGetCallerIdType=b
取值范围	b=0: 获取第二个主叫号码; b=1: 获取用户侧提供的主叫号码; b=2: 获取网络侧提供的主叫号码; b=3: 获取第一个主叫号码。
功能描述	获取信令中指定的主叫号码。

注意事项	<ul style="list-style-type: none">● 本配置项只适用于 ISDN 线路的监控；● 本配置项只适用于 DTP 系列板卡。
------	-----------------------------------------------------------------------------------------------------

3.1.1.4.5 设置 SpyCIC 编号（DTP 系列）

3.1.1.4.5.1 TotalAppSpyCIC

请参见 [AppSpyCIC](#)

3.1.1.4.5.2 AppSpyCIC

配置项	TotalAppSpyCIC AppSpyCIC
节	[AppSpyCICTable]
书写格式	TotalAppSpyCIC=N 格式 1：AppSpyCIC[n]=SpyPcm[m],k 格式 2：AppSpyCIC[n]=SpyPcm[m],k _s ..k _e
取值范围	N: SpyCIC 的总数。 n: SpyCIC 的逻辑编号，从 0 开始编号，0≤n≤N。 m: SpyPCM 的逻辑编号，详细内容请参见配置项 TotalSpyPcm 或 SpyPcm 的说明。 k: SpyPcm 上的电路编号，0≤k≤29。对于 SS7 信令，k 的值就是 TUP 或 ISUP 消息中的 CIC 字段；对于 ISDN PRI 信令，k 的值与 PCM 中的实习关系为： k _s : SpyPcm 上的起始电路编号，0≤k _s ≤29。 k _e : SpyPcm 上的结束电路编号，0≤k _e ≤29。
功能描述	TotalAppSpyCIC 设置被监控电路的总数。 AppSpyCIC 建立应用程序中 SpyCIC 的逻辑编号与 SpyPCM 上某个具体电路之间的映射。 更多信息请参见第 1 章中“ DTP 系列板卡 ”部分内容。
注意事项	<ul style="list-style-type: none">● 格式 1 用于逐个设置，格式 2 用于批量设置相当于从 k_s 开始，用格式 1 连续设置 k_e-k_s+1 次；● 本配置项只适用于 DTP 系列板卡。
使用范例	请参见配置项 TotalSpyPcm 中的使用范例。

3.1.1.4.6 SS7 信令的必须配置项 (DTP 系列)

3.1.1.4.6.1 TotalSpyLinkSet

请参见 [SpylsupCICPcm](#)

3.1.1.4.6.2 SpyLinkSet

请参见 [SpylsupCICPcm](#)

3.1.1.4.6.3 TotalSpyLinkPcm

请参见 [SpylsupCICPcm](#)

3.1.1.4.6.4 SpyLinkPcm

请参见 [SpylsupCICPcm](#)

3.1.1.4.6.5 SpyCICPcm

请参见 [SpylsupCICPcm](#)

3.1.1.4.6.6 SpySpCodeLen

请参见 [SpylsupCICPcm](#)

3.1.1.4.6.7 SpylsupCICPcm

配置项	TotalSpyLinkSet SpyLinkSet TotalSpyLinkPcm SpyLinkPcm SpyCICPcm SpySpCodeLen SpylsupCICPcm
节	[SS7Spy]

书写格式	<p>配置 1: 适用于单链路组:</p> <pre>TotalSpyLinkPcm=M SpyLinkPcm[m]=SpyPcm[x] SpyCICPcm[k]=SpyPcm[y] SpyIsupCICPcm[k]=SpyPcm[y] SpySpCodeLen=j</pre> <p>配置 2: 适用于单链路组或多链路组:</p> <pre>TotalSpyLinkSet=N 格式 1: SpyLinkSet[n]=SpyPcm[x] 格式 2: SpyLinkSet[n]=SpyPcm[x]+SpyPcm[y] SpyLinkSet[n].SpyCICPcm[k]=SpyPcm[m] SpyLinkSet[n].SpyIsupCICPcm[k]=SpyPcm[m] SpySpCodeLen=j</pre> <p>注: 格式 1 适用于链路组中只有 1 条信令链路的情况, 格式 2 适用于链路组中有 2 条信令链路的情况。</p> <p>注: 配置 2 的优先级高于配置 1。</p>
取值范围	<p>N: 信令链路组的总数, $1 \leq N \leq 48$, 必须对从 $n=0$ 开始, 将 SpyLinkSet 连续配置 N 次。</p> <p>n: 链路组的逻辑编号, $0 \leq n < N$。</p> <p>x,y, m: SpyPcm 的逻辑编号, $0 \leq x < M$, $0 \leq y < M$, $0 \leq m < M$。</p> <p>k: 电信局为 SpyPcm 分配的数字中继线编号, 即 TUP 或 ISUP 消息中 CIC 字段中 PCM 参数的值, 具体值可以向电信局获取。</p> <p>j: 取值为 24 或 14。信令点码编码标准, 24 为中国的信令点编码方式, 14 为国际的信令点编码方式。</p> <p>注: M 为配置项 <u>TotalSpyPcm</u> 的设定值, TotalSpyLinkPcm 的取值与 TotalSpyPcm 的值相同。</p>
功能描述	<p>TotalSpyLinkSet 设置 SS7 信令链路组的总数。</p> <p>SpyLinkSet 设置指定第 n 个链路组中信令链路的物理位置。</p> <p>TotalSpyLinkPcm 设置应用系统中需要监控的数字中继线的总数。</p> <p>SpyLinkPcm 设置指定链路组中信令链路的物理位置。</p> <p>SpyCICPcm、SpyIsupCICPcm 建立第 n 个链路组中 TUP 或 ISUP 消息的 CIC 字段的数字中继线编号与 SpyPCM 的逻辑编号之间的映射关系, SpyCICPcm 只适用于 TUP 协议, SpyIsupCICPcm 只适用于 ISUP 协议。</p>
注意事项	<ul style="list-style-type: none"> ● SpyLinkSet[n].SpyCICPcm[k]只适用于 TUP 协议; ● SpyLinkSet[n].SpyIsupCICPcm[k]只适用于 ISUP 协议; ● 本配置项只适用于 DTP 系列板卡的 SS7 信令; ● 本配置项要求 SynCTI 驱动程序的版本为 4.5.8.0 或更高。

使用范例	请参见第1章中“ DTP系列板卡的应用及配置实例 ”部分内容。
------	-------------------------------------------------

3.1.1.4.7 SS7信令的必须配置项（SHD系列）

3.1.1.4.7.1 设置使用 ISUP 协议的数字中继线

3.1.1.4.7.1.1 TotalIsupPcm

请参见 [IsupPcm](#)

3.1.1.4.7.1.2 IsupPcm

配置项	TotalIsupPcm IsupPcm
节	[SS7]
书写格式	TotalIsupPcm=M IsupPcm[m]=LocalPcm[k]
取值范围	M: 使用 ISUP 协议的数字中继线的总数, $0 \leq M < N$, N 为配置项 TotalPcm 的设定值。如果 M 不为 0, 必须设置配置项 IsupPcm。 m: 使用 ISUP 协议的数字中继线的逻辑编号, 从 0 开始编号。取值范围: $0 \leq m < M$ 。 k: 本应用系统中的数字中继线的逻辑编号, 取值范围: $0 \leq k < N$ 。数字中继线的逻辑编号由配置项 TotalPcm 和 Pcm 设置。
功能描述	TotalIsupPcm 设置应用系统中使用 ISUP 协议的数字中继线的总数。 IsupPcm 设置本机中使用 ISUP 协议的数字中继线。
注意事项	<ul style="list-style-type: none">如果系统中没有使用 ISUP 协议, 则不需要配置本配置项;如果 TotalIsupPcm 设置为 0, 则不需要设置配置项 IsupPcm, 并且配置项 AutoHandleIsup 将被忽略;本配置项只适用于 SHD 系列板卡。

3.1.1.4.8 ISDN 信令必须配置项 (SHD 系列)

3.1.1.4.8.1 设置运行模式

3.1.1.4.8.1.1 UseISDNMode

配置项	UseISDNMode
节	[ISDN]
书写格式	UseISDNMode=m
取值范围	m=0: 用户侧方式 (缺省值); m=1: 网络侧; m=2: 用户侧+网络侧。注意: 在此模式下, 同一片板卡上的全部数字中继线必须具有相同的模式, 即或者都是用户侧, 或者都是网络侧, 但不同板卡可以具有不同的模式。
功能描述	设置使用 ISDN 协议的模式。更多信息请参见第 1 章中“ 网络侧模式与用户侧模式 ”部分内容。

3.1.1.4.8.2 设置使用 ISDN 信令(用户侧)的数字中继线

3.1.1.4.8.2.1 TotalUserLinker

请参见 [UserTEIValue](#)

3.1.1.4.8.2.2 UserPcmLink

请参见 [UserTEIValue](#)

3.1.1.4.8.2.3 UserSendEstablish

请参见 [UserTEIValue](#)

3.1.1.4.8.2.4 UserTEIValue

配置项	TotalUserLinker UserPcmLink UserSendEstablish UserTEIValue
节	[ISDN]
书写格式	TotalUserLinker=N UserPcmLink[n]=LocalPCM[k] UserSendEstablish[n]=b UserTEIValue[n]=m

取值范围	<p>N: 系统中用户侧数字中继线的总数。</p> <p>n: 用户侧数字中继线的逻辑编号，从 0 开始编号。</p> <p>k: 数字中继线在本端的逻辑编号，详细内容请参见配置项 TotalPcm 和 Pcm 的说明。</p> <p>b: 本端是否发送建链消息。</p> <p>0: 不发送</p> <p>1: 发送（缺省值）</p> <p>m: 用户侧 TEI 值。取值范围：</p> <p>$0 \leq m \leq 63$: 固定 TEI;</p> <p>$m > 63$: 动态 TEI。</p> <p>m 的缺省值为 0。</p> <p>注意：假设网络侧的 TEI 值（NetTEIValue）为 x。如果 $0 \leq m \leq 63$，则 x 必须等于 m，否则会导致本端与网络侧之间无法建立链路；如果 $m > 63$，则 x 可以不必等于 m，可以取 $0 \leq x \leq 126$ 中的任何值。有关 TEI 值的更多信息，请参见第 1 章中“TEI 值”部分内容。</p>
功能描述	<p>TotalUserLinker 设置本端使用 ISDN 信令、并且工作于用户侧的数字中继线的总数。</p> <p>UserPcmLink 建立每条用户侧数字中继线的编号与其逻辑编号之间的映射关系。</p> <p>UserTEIValue 设置用户侧数字中继线的 TEI 值。</p>
注意事项	<ul style="list-style-type: none"> ● 如果 N>0，UserPcmLink 和 UserTEIValue 都必须设置 N 次； ● 本配置项只适用于用户侧模式。

3.1.1.4.8.3 设置使用 ISDN 信令(网络侧)的数字中继线

3.1.1.4.8.3.1 TotalNetLinker

请参见 [NetTEIValue](#)

3.1.1.4.8.3.2 NetPcmLink

请参见 [NetTEIValue](#)

3.1.1.4.8.3.3 NetSendEstablish

请参见 [NetTEIValue](#)

3.1.1.4.8.3.4 NetTEIValue

配置项	TotalNetLinker NetPcmLink NetSendEstablish NetTEIValue
节	[ISDN]

书写格式	TotalNetLinker=N NetPcmLink[n]=LocalPCM[k] NetSendEstablish[n]=b NetTEIValue[n]=x
取值范围	N: 系统中网络侧数字中继线的总数。 n: 网络侧数字中继线的逻辑编号, 从 0 开始编号。 k: 数字中继线在本端的逻辑编号, 详细内容请参见配置项 TotalPcm 和 Pcm 的说明。 b: 本端是否发送建链消息。 0: 不发送 (缺省值) 1: 发送 x: 网络侧的 TEI 值。取值范围: 假设用户侧的 TEI 值 (UserTEIValue) 为 m。如果 $0 \leq m \leq 63$, 则 x 必须等于 m, 否则会导致用户侧与网络侧之间无法建立链路; 如果 $m > 63$, 则 x 不必等于 m, 可以取 $0 \leq x \leq 126$ 的任何值。 x 的缺省值为 0。
功能描述	TotalNetLinker 设置本端工作于网络侧的数字中继线的总数。 NetPcmLink 建立每条网络侧数字中继线与其逻辑编号之间的映射关系。 NetTEIValue 设置每条网络侧数字中继线的 TEI 值。
注意事项	<ul style="list-style-type: none"> 如果 N>0, NetPcmLink 和 NetTEIValue 都必须设置 N 次; 本配置项只适用于网络侧模式。

3.1.1.4.9 SS1 信令的必须配置项 (SHD 系列)

3.1.1.4.9.1 ProtocolType

配置项	ProtocolType
节	[SS1Config]
书写格式	ProtocolType=m
取值范围	<p>m=0: 使用 1 号信令 (缺省值);</p> <p>m=1: 使用 LineSide 协议;</p> <p>m=2: 使用 ASB 协议;</p> <p>m=3: 使用 LineSide (OPS-FX) 协议;</p> <p>m=4: 使用 VPS 协议;</p> <p>m=5: 使用 ASBEL7 协议。</p>
功能描述	<p>设置 SS1 通道的信令方式。</p> <p>以下对不同的 ProtocolType 做简要说明, 用户可根据对端交换机的要求进行配置。</p> <p>m = 1、3、4 时, 发送的 CAS CD 位的值是读的配置项 TxCas_CD 的值, 空闲状态下收到 CAS AB 位为 0 时迁移至振铃。m=1 和 m=3 的区别仅在于摘机时发送的 CAS AB 位一个为 2, 一个为 3; m = 4 时, 通道摘机时发送的 AB 位为 0x0f。</p> <p>m = 2 时, 发送的 CAS AB 位的值是读的配置项 TxCas_CD 的值, 空闲状态下收到 CAS ABCD 位为 2 时迁移至振铃, 通道摘机时发送的 CAS CD 位为 2。</p> <p>m=5 时, 表示使用 ASBEL7 协议, 需要同时配置 Ss1SendIdleState=2, TxCas_CD=0。</p>

3.1.1.5 SHN 系列板卡

3.1.1.5.1 SHN 系列卡通用配置

3.1.1.5.1.1 LocalSipIp

注：在 5.1.0.0 以前版本该配置项名为 LocalIp，且在 5.1.0.0 版本前 B 型 IP 卡在 SIP 节无该配置项。

请参见 [LocalSipPort](#)

3.1.1.5.1.2 LocalSipPort

注：在 5.1.0.0 以前版本该配置项名为 LocalPort，且在 5.1.0.0 版本前 B 型 IP 卡在 SIP 节无该配置项。

配置项	LocalSipIp LocalSipPort
节	[SIP]
书写格式	LocalSipIp=m LocalSipPort=n
取值范围	m：设置 SIP 信令承载的网络地址。m 为字符串，取值形式为 Ipv4 地址，长度小于 256 个字符。 特殊情况：m=0.0.0.0：随机选择本地主机 IP 地址之一作为 SIP 信令承载的网络地址。 n：设置 SIP 承载的网络侦听端口，默认为 5060，取值范围为：1024 ~ 65536
功能描述	设置 IP 卡承载的 SIP 信令网络地址和端口。

3.1.1.5.1.3 SipSetConnectionIp

配置项	SipSetConnectionIp
节	[BoardId=x]
书写格式	SipSetConnectionIp=z
取值范围	z=a.b.c.d, SIP 信令 INVITE 消息和 200OK 消息中的 SDP 中的 C 字段内容，默认值为 0.0.0.0，表示不设置 INVITE 消息和 200OK 消息中的 SDP 中的 C 字段的内容
功能描述	设置 SIP 信令 INVITE 消息和 200OK 消息中的 SDP 中的 C 字段内容。
注意事项	● 与 SsmSipSetConnectionInforOfSDP 函数功能相同。

3.1.1.5.2 SHN A 系列卡必须配置项

3.1.1.5.2.1 设置协议类型

3.1.1.5.2.1.1 ProtocolType

配置项	ProtocolType
节	[BoardId=x]
书写格式	ProtocolType=m
取值范围	m: 设置本 IP 卡使用的协议类型, 具体为: m=1: 使用 SIP 类型。 m=2: 将 IP 卡设置 IP 资源卡。
功能描述	ProtocolType 设置本 IP 卡使用的通讯协议或设置 IP 资源卡。

3.1.1.5.2.2 SIP 信令的必须配置项

3.1.1.5.2.2.1 RTPRange

配置项	RTPRange
节	[SIP]
书写格式	RTPRange= t _L , t _H
取值范围	t _L : RTP 端口范围下限, 取值范围: 1024≤t _L , 缺省值为 6000。 t _H : RTP 端口范围上限, 取值范围: t _H ≤65535, 缺省值为 10000。
功能描述	设置 A 系列卡 RTP 端口的范围。

3.1.1.5.2.2.2 AudioCodecList

请参见 [SendDtmfType](#)

3.1.1.5.2.2.3 SendDtmfType

配置项	AudioCodecList SendDtmfType
节	[BoardId=x]

书写格式	AudioCodecList= p ₁ ,p ₂ ,..., p _n SendDtmfType=j
取值范围	<p>p₁,p₂,..p_n: 通话使用的语音编解码列表，其中 n≤10，值在前表示优先级高。其值的意义如下：</p> <ul style="list-style-type: none"> p_x=6: G.711 A-Law 格式; p_x=7: G.711 μ-Law 格式; p_x=4: G.723_1 格式; p_x=49: GSM 格式; p_x=9: G722 格式; p_x=96: AMR 格式; p_x=98: ILBC 格式; p_x=100: SILK(16)格式; p_x=102: OPUS(8)格式; p_x=131: G.729A 格式。 <p>B 型 IP 卡支持：A-Law, μ-Law, G.729A, G.723_1 和 GSM; C 型 IP 卡支持所有格式。</p> <p>j: 设置发送 DTMF 的方式，取值范围：</p> <ul style="list-style-type: none"> j=0: 采用 RFC2833 事件方式发送 DTMF; j=1: 采用信令方式发送 DTMF; j=2: 采用带内方式发送 DTMF。
功能描述	设置 SHN 系列卡通话语音的编解码格式、发送 DTMF 方式。
注意事项	配置项 SendDtmfType 对所有 IP 卡有效，HMP 不支持带内方式发送 DTMF。

3.1.1.5.2.2.4 RecvDtmfType

配置项	RecvDtmfType
节	[BoardId=x]
书写格式	RecvDtmfType=j
取值范围	<p>j: 设置接收 DTMF 的方式，取值范围：</p> <ul style="list-style-type: none"> j=0: 带内、信令、RFC2833 方式都支持; j=1: 仅接受信令和 RFC2833 方式; j=2: 仅接受带内方式; j=3: 仅接受信令方式。
功能描述	设置 SHN A 系列卡接收 DTMF 的方式。

3.1.1.5.3 SHN B 系列卡必须配置项

3.1.1.5.3.1 SIP 信令的必须配置项

3.1.1.5.3.1.1 BoardIP

注：在 5.1.0.0 版本前该配置项名为 LocalIp，且在 5.1.0.0 版本前该 IP 地址作为信令和 RTP 的共用 IP。

请参见 [SendDtmfType](#)

3.1.1.5.3.1.2 SubMask

请参见 [SendDtmfType](#)

3.1.1.5.3.1.3 Gateway

请参见 [SendDtmfType](#)

3.1.1.5.3.1.4 DNS

请参见 [SendDtmfType](#)

3.1.1.5.3.1.5 AudioCodecList

请参见 [SendDtmfType](#)

3.1.1.5.3.1.6 RTPRange

请参见 [SendDtmfType](#)

3.1.1.5.3.1.7 SendSipMsg

请参见 [SendDtmfType](#)

3.1.1.5.3.1.8 SendDtmfType

配置项	BoardIP SubMask Gateway DNS AudioCodecList RTPRange SendSipMsg SendDtmfType
节	[BoardId=x]

书写格式	BoardIP=m Submask=v Gateway=p DNS=L AudioCodecList= q ₁ ,q ₂ ,..., q _n RTPRange= t _L , t _H SendSipMsg=n SendDtmfType=j
取值范围	<p>m: 设置 SHN B 系列板卡自带网卡的 IP 地址。</p> <p>v: 设置 SHN B 系列板卡的子网掩码。v 为字符串, 取值形式为 Ipv4 地址, 长度小于 256 个字符。 默认为: v=255.255.255.0。</p> <p>p: SHN B 系列板卡的网关。</p> <p>L: 设置 DNS 地址。L 为字符串, 取值形式为 Ipv4 地址, 长度小于 256 个字符。默认为: L=192.168.1.100。</p> <p>q₁,q₂,..., q_n: 通话使用的语音编解码列表, 缺省值为 6,7,131,49, 其中 n≤10, 值在前表示优先级高。 其值的意义如下:</p> <ul style="list-style-type: none"> q_x=6: G.711 A-Law 格式; q_x=7: G.711 μ-Law 格式; q_x=131: G.729A 格式; q_x=49: GSM 格式 (仅支持 SHN B 型卡)。 <p>t_L: RTP 端口范围下限, 取值范围: 1024≤t_L, 缺省值为 6000。</p> <p>t_H: RTP 端口范围上限, 取值范围: t_H≤65535, 缺省值为 10000。</p> <p>备注: 6000 以下和 10000 以上的 RTP 端口可能已经被驱动占用, 因此 t_L、t_H 建议使用缺省值。</p> <p>n: 设置 SIP 通道在振铃下给对方放音的开关, 当且仅当 n=183 时, 该配置项才有效, 此时驱动收到 invite 呼叫的时候将自动回复 183+SDP, 而不再发送 180 消息, n 为其他值时, 配置项无效。 该配置项只对 B 型卡和 C 型卡有效, A 型卡不起作用。</p> <p>j: 设置发送 DTMF 的方式, 取值范围:</p> <ul style="list-style-type: none"> j=0: 采用 RFC2833 事件方式发送 DTMF; j=1: 采用信令方式发送 DTMF; j=2: 采用带内方式发送 DTMF。
功能描述	设置 SHN B 系列卡 SIP 信令承载的网络地址、发送 DTMF 方式、SIP 信令的工作参数。
注意事项	配置项 SendDtmfType 对所有 IP 卡有效, HMP 不支持带内方式发送 DTMF。

3.1.1.5.3.1.9 RecvDtmfType

配置项	RecvDtmfType
节	[BoardId=x]

书写格式	RecvDtmfType=j
取值范围	j: 设置接收 DTMF 的方式, 取值范围: j=0: 带内、信令、RFC2833 方式都支持; j=1: 仅接受信令和 RFC2833 方式; j=2: 仅接受带内方式; j=3: 仅接受信令方式。
功能描述	设置 SHN B 系列卡接收 DTMF 的方式。

3.1.1.5.3.1.10 TelephoneEventsPt

配置项	TelephoneEventsPt
节	[BoardId=x]
书写格式	TelephoneEventsPt=m
取值范围	m: m 为数字, 表示本地可接收的 DTMF RFC2833 负载格式, 取值范围为: 0 ~ 127, 默认值为 101。
功能描述	设置本地可接收的 DTMF RFC2833 的负载格式。

3.1.1.5.3.1.11 Send180After183

配置项	Send180After183
节	[BoardId=x]
书写格式	Send180After183=y
取值范围	y: 是否在 183 消息之后发送 180 消息; 0: 否 (缺省), 1: 是。
功能描述	在 183 消息之后发送 180 消息。
注意事项	需要配置 SendSipMsg=183。

3.1.1.5.4 SHN C 系列卡必须配置项

3.1.1.5.4.1 SHN C 系列卡专有配置项

3.1.1.5.4.1.1 OctMac0

配置项	OctMac0
-----	---------

节	[BoardId=x]
书写格式	OctMac0 =00-0c-90-2d-05-02
功能描述	设置 C 型 IP 卡 Ethernet 1 的 MAC 地址。

3.1.1.5.4.1.2 DhcpServer

配置项	DhcpServer
节	[DHCP]
书写格式	DhcpServer=201.123.115.65
功能描述	设置本地主机控制消息的 IP 地址，用于加载固件芯片和对芯片收发控制命令等。

3.1.1.5.4.1.3 ProcessorCtrlMac

配置项	ProcessorCtrlMac
节	[DHCP]
书写格式	ProcessorCtrlMac =00-1F-D0-5E-AA-36
功能描述	设置本地主机控制消息的 MAC 地址，用于加载固件芯片和芯片进行通信，进行控制命令的收发。

3.1.1.5.4.1.4 FilterMacRange

配置项	FilterMacRange
节	[DHCP]
书写格式	FilterMacRange=00:0c:90:2d:04:e6- 00:0c:90:2d:04:45- 00:0c:90:2d:04:66
功能描述	设置可以被 DHCP 服务器分配 IP 地址的 MAC 地址，只有设置了此 MAC 地址的网络设备在请求分配 IP 地址的时候才能被 DHCP 服务器识别并分配 IP 地址。

3.1.1.5.4.1.5 BootFileName

配置项	BootFileName
节	[DHCP]
书写格式	BootFileName=oct2200.img (缺省值)

功能描述	设置固件加载的名称，驱动会以 <code>tftp server</code> 来加载固件。如果本地网络中已经存在 DHCP 服务器，那么此配置项可以删除或者设置为空，之后驱动会以 <code>tftp client</code> 方式进行加载固件，但是网络中的 DHCP 服务器分配的 IP 地址必须是同网段，否则会导致固件加载失败。
------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3.1.1.5.4.1.6 LogLevel

配置项	LogLevel
节	[DHCP]
书写格式	LogLevel=All (缺省值)
取值范围	打开日志设置为 All，关闭日志设置为 None。
功能描述	设置 DHCP 和 TFTP 日志的开关。

3.1.1.5.4.1.7 DHCPRange

配置项	DHCPRange
节	[DHCP]
书写格式	DHCPRange=201.123.115.240-201.123.115.244
功能描述	配置 DHCP IP 地址范围。

3.1.1.5.4.2 SIP 信令的必须配置项

3.1.1.5.4.2.1 BoardIP

注：在 5.1.0.0 版本前该配置项名为 LocalIp，且在 5.1.0.0 版本前该 IP 地址作为信令和 RTP 的共用 IP。

请参见 [SendDtmfType](#)

3.1.1.5.4.2.2 SubMask

请参见 [SendDtmfType](#)

3.1.1.5.4.2.3 Gateway

请参见 [SendDtmfType](#)

3.1.1.5.4.2.4 DNS

请参见 [SendDtmfType](#)

3.1.1.5.4.2.5 AudioCodecList

请参见 [SendDtmfType](#)

3.1.1.5.4.2.6 RTPRange

请参见 [SendDtmfType](#)

3.1.1.5.4.2.7 SendSipMsg

请参见 [SendDtmfType](#)

3.1.1.5.4.2.8 SendDtmfType

配置项	BoardIP SubMask Gateway DNS AudioCodecList RTPRange SendSipMsg SendDtmfType
节	[BoardId=x]
书写格式	BoardIP=m Submask=v Gateway=p DNS=L AudioCodecList= q ₁ ,q ₂ ,..., q _n RTPRange= t _L , t _H SendSipMsg=n SendDtmfType=j

取值范围	<p>m: 设置 SHN C 系列板卡自带网卡的 IP 地址。</p> <p>v: 设置 SHN C 系列板卡的子网掩码。v 为字符串, 取值形式为 Ipv4 地址, 长度小于 256 个字符。默认为: v=255.255.255.0。</p> <p>p: SHN C 系列板卡的网关。</p> <p>L: 设置 DNS 地址。L 为字符串, 取值形式为 Ipv4 地址, 长度小于 256 个字符。默认为: L=192.168.1.100。</p> <p>q₁, q₂, ..., q_n: 通话使用的语音编解码列表, 缺省值为 6, 7, 131, 4, 9, 96, 98, 其中 n≤10, 值在前表示优先级高。其值的意义如下:</p> <ul style="list-style-type: none"> q_x=6: G.711 A-Law 格式; q_x=7: G.711 μ-Law 格式; q_x=131: G.729A 格式; q_x=4: G723_1 格式 (仅支持 SHN C 型卡, 语音正常的情况下 G723_1 能同时支持 280 路通道通话); q_x= 9: G722 格式 (仅支持 SHN C 型卡); q_x= 96: AMR 格式 (仅支持 SHN C 型卡, 语音正常的情况下 AMR 能同时支持 310 路通道通话); q_x= 98: ILBC 格式 (仅支持 SHN C 型卡); <p>t_L: RTP 端口范围下限, 取值范围: 1024≤t_L, 缺省值为 6000。</p> <p>t_H: RTP 端口范围上限, 取值范围: t_H≤65535, 缺省值为 10000。</p> <p>备注: 6000 以下和 10000 以上的 RTP 端口可能已经被驱动占用, 因此 t_L、t_H 建议使用缺省值。</p> <p>n: 设置 SIP 通道在振铃下给对方放音的开关, 当且仅当 n=183 时, 该配置项才有效, 此时驱动收到 invite 呼叫的时候将自动回复 183+SDP, 而不再发送 180 消息, n 为其他值时, 配置项无效。该配置项只对 B 型卡和 C 型卡有效, A 型卡不起作用。</p> <p>j: 设置发送 DTMF 的方式, 取值范围:</p> <ul style="list-style-type: none"> j=0: 采用 RFC2833 事件方式发送 DTMF; j=1: 采用信令方式发送 DTMF; j=2: 采用带内方式发送 DTMF。
功能描述	设置 SHN C 系列卡 SIP 信令承载的网络地址、发送 DTMF 方式、SIP 信令的工作参数。
注意事项	配置项 SendDtmfType 对所有 IP 卡有效, HMP 不支持带内方式发送 DTMF。

3.1.1.5.4.2.9 RecvDtmfType

配置项	RecvDtmfType
节	[BoardId=x]
书写格式	RecvDtmfType=j

取值范围	j: 设置接收 DTMF 的方式, 取值范围: j=0: 带内、信令、RFC2833 方式都支持; j=1: 仅接受信令和 RFC2833 方式; j=2: 仅接受带内方式; j=3: 仅接受信令方式。
功能描述	设置 SHN C 系列卡接收 DTMF 的方式。

3.1.1.5.4.2.10 TelephoneEventsPt

配置项	TelephoneEventsPt
节	[BoardId=x]
书写格式	TelephoneEventsPt=m
取值范围	m: m 为数字, 表示本地可接收的 DTMF RFC2833 负载格式, 取值范围为: 0 ~ 127, 默认值为 101。
功能描述	设置本地可接收的 DTMF RFC2833 的负载格式。

3.1.1.6 DST 系列板卡 (REC 系列 only)

3.1.1.6.1 PBXType

配置项	PBXType
节	[BoardId=x]
书写格式	PBXType=s
取值范围	s: 表示交换机型号的字符串, 可选的型号请参见第 1 章中 “ DST 系列板卡支持的交换机型号 ” 部分内容。
功能描述	设置 DST 系列板卡所连接的交换机型号。
使用范例	PBXType=ALCATEL4300

3.1.1.6.2 PhoneType

配置项	PhoneType
节	[BoardId=x]

书写格式	PhoneType=n ₁ ,n ₂ ,n ₃ ,n ₄ ,n ₅ ,n ₆ ,n ₇ ,n ₈ ,n ₉ ,n ₁₀ ,n ₁₁ ,n ₁₂ ,n ₁₃ ,n ₁₄ ,n ₁₅ ,n ₁₆
取值范围	n ₁ ~n ₁₆ : 表示数字电话机型号的数字, 可选值请参见第1章中“ DST系列板卡支持的交换机型号 ”部分内容。
功能描述	设置 DST 板卡所连接的数字电话机型号的设置值。

3.1.1.6.3 DstRecRawData

配置项	DstRecRawData
节	[BoardId=x]
书写格式	DstRecRawData=n
取值范围	n=0, 默认值, 表示普通工作模式; n=1, 表示原始数据采集模式。
功能描述	通过该配置项可以将板卡设置为原始数据采集模式, 采集线路上的原始波形数据, 用于分析和解决问题。
注意事项	<ul style="list-style-type: none">本配置项只适用于 DST 系列 B 型数字电话录音卡。

3.1.1.6.4 SetAnalogCtrlEnable

配置项	SetAnalogCtrlEnable
节	[BoardId=x]
书写格式	SetAnalogCtrlEnable=n
取值范围	n=0, AnalogCtrl 配置项无效; n=1, AnalogCtrl 配置项有效。
功能描述	设置配置项 AnalogCtrl 是否有效。
注意事项	<ul style="list-style-type: none">本配置项只适用于 DST 系列 B 型数字电话录音卡。一般情况下无需添加该配置项, 必要时请在我方技术人员指导下设置该配置项。

3.1.1.6.5 AnalogCtrl

配置项	AnalogCtrl																																																																																																												
节	[BoardId=x]																																																																																																												
书写格式	AnalogCtrl =0xabcd,0xabcd,0xabcd																																																																																																												
取值范围	<p>十六进制数据，取值请参考模拟开关参考值表，必要时可根据实际情况进行适当调整；三个数据分别对应3个模块。模拟开关参考值表如下：</p> <table border="1"> <thead> <tr> <th>交换机型号</th> <th>模拟开关参考值</th> <th>交换机型号</th> <th>模拟开关参考值</th> </tr> </thead> <tbody> <tr><td>Aastra Nexspan</td><td>0x6</td><td>Alcatel4200/4400</td><td>0x2</td></tr> <tr><td>AscotelIntelligate</td><td>0x4</td><td>Aspect</td><td>0x6</td></tr> <tr><td>AvayaDefinity-2w</td><td>0x2</td><td>AvayaDefinity-4w</td><td>0x4</td></tr> <tr><td>AvayaIndex</td><td>0x4</td><td>AvayaMerlingMagix</td><td>0x2</td></tr> <tr><td>AvayaMLX</td><td>0x4</td><td>Belgacom</td><td>0x2</td></tr> <tr><td>BoschIntegral2w</td><td>0x2</td><td>BoschIntegral4w</td><td>0x4</td></tr> <tr><td>BoschIntegral552w</td><td>0x2</td><td>BRI ISDN</td><td>0x4</td></tr> <tr><td>BRI ISDN1</td><td>0x4</td><td>Comdial</td><td>0x4</td></tr> <tr><td>Eon_EQ</td><td>0x4</td><td>Ericsson ELU25/28</td><td>0x4</td></tr> <tr><td>Ericsson ELU5</td><td>0x4</td><td>Fujitsu F9600</td><td>0x0</td></tr> <tr><td>Harris2B</td><td>0x4</td><td>Harris_1B_optic</td><td>0x6</td></tr> <tr><td>Harris_1B_workstation</td><td>0x6</td><td>IAP</td><td>0x4</td></tr> <tr><td>Intecom2-wire</td><td>0x2</td><td>Intercom4-wire</td><td>0x2</td></tr> <tr><td>Intel-Tel</td><td>0x4</td><td>iwatsu256k</td><td>0x2</td></tr> <tr><td>Iwatsu512k</td><td>0x2</td><td>LG_StareXCS</td><td>0x4</td></tr> <tr><td>MirtelSx2000</td><td>0x4</td><td>NEC</td><td>0x4</td></tr> <tr><td>nitsuko</td><td>0x2</td><td>Nortel Matra</td><td>0x6</td></tr> <tr><td>Nortel Meridian1</td><td>0x4</td><td>Nortel Norstar</td><td>0x4</td></tr> <tr><td>panasonic_DBS</td><td>0x0</td><td>Panasonic KX</td><td>0x2</td></tr> <tr><td>Panasonic 7600se</td><td>0x0</td><td>PhilipsSOPHOis30002w</td><td>0x2</td></tr> <tr><td>PhilipsSOPHOis30004w</td><td>0x4</td><td>RockwellSpectrum</td><td>0x4</td></tr> <tr><td>SamsungDCS-828</td><td>0x4</td><td>SamsungINFOREX</td><td>0x4</td></tr> <tr><td>Siemens Hicom</td><td>0x2</td><td>Siemens ISDT2W</td><td>0x0</td></tr> <tr><td>Siemens Realifis DT</td><td>0x4</td><td>Siemens RolmLink</td><td>0x2</td></tr> <tr><td>Tadran Coral</td><td>0x2</td><td>Teltronics(Harris2020)</td><td>0x6</td></tr> <tr><td>ToshibaStrataDK/CTX</td><td>0x4</td><td></td><td></td></tr> </tbody> </table>	交换机型号	模拟开关参考值	交换机型号	模拟开关参考值	Aastra Nexspan	0x6	Alcatel4200/4400	0x2	AscotelIntelligate	0x4	Aspect	0x6	AvayaDefinity-2w	0x2	AvayaDefinity-4w	0x4	AvayaIndex	0x4	AvayaMerlingMagix	0x2	AvayaMLX	0x4	Belgacom	0x2	BoschIntegral2w	0x2	BoschIntegral4w	0x4	BoschIntegral552w	0x2	BRI ISDN	0x4	BRI ISDN1	0x4	Comdial	0x4	Eon_EQ	0x4	Ericsson ELU25/28	0x4	Ericsson ELU5	0x4	Fujitsu F9600	0x0	Harris2B	0x4	Harris_1B_optic	0x6	Harris_1B_workstation	0x6	IAP	0x4	Intecom2-wire	0x2	Intercom4-wire	0x2	Intel-Tel	0x4	iwatsu256k	0x2	Iwatsu512k	0x2	LG_StareXCS	0x4	MirtelSx2000	0x4	NEC	0x4	nitsuko	0x2	Nortel Matra	0x6	Nortel Meridian1	0x4	Nortel Norstar	0x4	panasonic_DBS	0x0	Panasonic KX	0x2	Panasonic 7600se	0x0	PhilipsSOPHOis30002w	0x2	PhilipsSOPHOis30004w	0x4	RockwellSpectrum	0x4	SamsungDCS-828	0x4	SamsungINFOREX	0x4	Siemens Hicom	0x2	Siemens ISDT2W	0x0	Siemens Realifis DT	0x4	Siemens RolmLink	0x2	Tadran Coral	0x2	Teltronics(Harris2020)	0x6	ToshibaStrataDK/CTX	0x4		
交换机型号	模拟开关参考值	交换机型号	模拟开关参考值																																																																																																										
Aastra Nexspan	0x6	Alcatel4200/4400	0x2																																																																																																										
AscotelIntelligate	0x4	Aspect	0x6																																																																																																										
AvayaDefinity-2w	0x2	AvayaDefinity-4w	0x4																																																																																																										
AvayaIndex	0x4	AvayaMerlingMagix	0x2																																																																																																										
AvayaMLX	0x4	Belgacom	0x2																																																																																																										
BoschIntegral2w	0x2	BoschIntegral4w	0x4																																																																																																										
BoschIntegral552w	0x2	BRI ISDN	0x4																																																																																																										
BRI ISDN1	0x4	Comdial	0x4																																																																																																										
Eon_EQ	0x4	Ericsson ELU25/28	0x4																																																																																																										
Ericsson ELU5	0x4	Fujitsu F9600	0x0																																																																																																										
Harris2B	0x4	Harris_1B_optic	0x6																																																																																																										
Harris_1B_workstation	0x6	IAP	0x4																																																																																																										
Intecom2-wire	0x2	Intercom4-wire	0x2																																																																																																										
Intel-Tel	0x4	iwatsu256k	0x2																																																																																																										
Iwatsu512k	0x2	LG_StareXCS	0x4																																																																																																										
MirtelSx2000	0x4	NEC	0x4																																																																																																										
nitsuko	0x2	Nortel Matra	0x6																																																																																																										
Nortel Meridian1	0x4	Nortel Norstar	0x4																																																																																																										
panasonic_DBS	0x0	Panasonic KX	0x2																																																																																																										
Panasonic 7600se	0x0	PhilipsSOPHOis30002w	0x2																																																																																																										
PhilipsSOPHOis30004w	0x4	RockwellSpectrum	0x4																																																																																																										
SamsungDCS-828	0x4	SamsungINFOREX	0x4																																																																																																										
Siemens Hicom	0x2	Siemens ISDT2W	0x0																																																																																																										
Siemens Realifis DT	0x4	Siemens RolmLink	0x2																																																																																																										
Tadran Coral	0x2	Teltronics(Harris2020)	0x6																																																																																																										
ToshibaStrataDK/CTX	0x4																																																																																																												
功能描述	设置模拟开关值。																																																																																																												

注意事项	<ul style="list-style-type: none">● 本配置项只适用于 DST 系列 B 型数字电话录音卡。● 该配置项只有在配置项 SetAnalogCtrlEnable=1 时有效。● 一般情况下无需设置该配置项，如在使用该配置项时有任何疑问，请与我方技术支持联系。
------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3.1.1.6.6 SetVoxChSelectEnable

配置项	SetVoxChSelectEnable
节	[BoardId=x]
书写格式	SetVoxChSelectEnable =n
取值范围	n=0, VoxChSelect 配置项无效; n=1, VoxChSelect 配置项有效。
功能描述	设置配置项 VoxChSelect 是否有效。
注意事项	<ul style="list-style-type: none">● 本配置项只适用于 DST 系列 B 型数字电话录音卡。● 一般情况下无需设置该配置项，如有需要，请在我方技术人员指导下设置该配置项。

3.1.1.6.7 VoxChSelect

配置项	VoxChSelect
节	[BoardId=x]
书写格式	VoxChSelect =0xabcd,0xabcd,0xabcd
取值范围	十六进制数据，bit0-15 分别对应通道 0-7 路的语音通道选择，每个通道占用 2bit，具体含义为： 00 – B0; 01 – B1; 10 – B2。
功能描述	采用 DST 系列 B 型卡录音时，DST 板卡通道上的两个语音通道会被设置成不同的比特值（B0、B1 或 B2）。本配置项通过比特值设置将被录音的语音通道。默认录制 B0 语音通道。
使用范例	VoxChSelect=0x4924,0x0,0x5555，表示通道 0-7 需要录音的语音通道依次为 B0, B1, B2, B0, B1, B2, B0, B1，通道 8-15 需要录音的语音通道全部为 B0，通道 16-23 需要录音的语音通道全部为 B1。

注意事项	<ul style="list-style-type: none">● 本配置项只适用于 DST 系列 B 型数字电话录音卡。● 该配置项在设置了配置项 SetVoxChSelectEnable =1 时有效。● 一般情况下无需设置该配置项，如有需要，请在我方技术人员指导下设置该配置项。
------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3.1.1.6.8 SetFilterSwitchEnable

配置项	SetFilterSwitchEnable
节	[BoardId=x]
书写格式	SetFilterSwitchEnable =n
取值范围	n=0, FilterSwitch 配置项无效; n=1, FilterSwitch 配置项有效。
功能描述	设置配置项 FilterSwitch 是否有效。
注意事项	<ul style="list-style-type: none">● 本配置项只适用于 DST 系列 B 型数字电话录音卡。● 一般情况下无需设置该配置项，如有需要，请在我方技术人员指导下设置该配置项。

3.1.1.6.9 FilterSwitch

配置项	FilterSwitch
节	[BoardId=x]
书写格式	FilterSwitch =0abcd,0abcd,0abcd
取值范围	十六进制数据，具体数值根据实际情况决定，三个数据分别对应 3 个模块。
功能描述	设置滤波开关。
注意事项	<ul style="list-style-type: none">● 本配置项只适用于 DST 系列 B 型数字电话录音卡。● 该配置项只有在配置项 SetFilterSwitchEnable =1 时有效。● 一般情况下无需设置该配置项，如有需要，请在我方技术人员指导下设置该配置项。

3.1.1.6.10 SetVoltageReferenceEnable

配置项	SetVoltageReferenceEnable
-----	---------------------------

节	[BoardId=x]
书写格式	SetVoltageReferenceEnable =n
取值范围	n=0, VoltageReference 配置项无效; n=1, VoltageReference 配置项有效。
功能描述	设置配置项 VoltageReference 是否有效。
注意事项	<ul style="list-style-type: none"> ● 本配置项只适用于 DST 系列 B 型数字电话录音卡。 ● 一般情况下无需设置该配置项，如有需要，请在我方技术人员指导下设置该配置项。

3.1.1.6.11 VoltageReference

配置项	VoltageReference
节	[BoardId=x]
书写格式	VoltageReference =0xabcd,0xabcd,0xabcd
取值范围	十六进制数据，具体数值根据实际情况决定；三个数据分别对应 3 个模块。
功能描述	设置初始最高电平和参考电压幅度。
注意事项	<ul style="list-style-type: none"> ● 本配置项只适用于 DST 系列 B 型数字电话录音卡。 ● 该配置项只有在配置项 SetVoltageReferenceEnable =1 时才有效。 ● 一般情况下无需设置该配置项，如有需要，请在我方技术人员指导下设置该配置项。

3.1.1.7 IPR 系列产品必须配置项 (REC 系列 only)

3.1.1.7.1 SynIPRecorder 系列产品 Master 端的必须配置项

3.1.1.7.1.1 RecMasterIP

配置项	RecMasterIP
节	[BoardId=x]
书写格式	RecMasterIP =m
取值范围	m: m 为字符串，取值形式为 IPv4 地址，默认值为 127.0.0.1。

功能描述	SynIPRecorder Master 的主机 IP 地址。
------	---------------------------------

3.1.1.7.1.2 RecMasterPort

配置项	RecMasterPort
节	[BoardId=x]
书写格式	RecMasterPort =m
取值范围	m: m 为数字, 默认为 9888。
功能描述	SynIPRecorder Master 的主机监听端口。

3.1.1.7.1.3 RTPTimeOut

配置项	RTPTimeOut
节	[BoardId=x]
书写格式	RTPTimeOut =m
取值范围	m: m 为数字, 当一个录音通道长时间没检测到 RTP 时, 自动认为该录音通道对应的 Session 已经断开, 自动停止录音。其中值为 0 时表示不进行 RTP 超时检测, 默认值为 15, 单位为秒。
功能描述	设置检测 RTP 超时的时间。

3.1.1.7.2 SynIPAnalyzer 系列产品的必须配置项

3.1.1.7.2.1 MonitorNIC

配置项	MonitorNIC
节	[BoardId=x]
书写格式	MonitorNIC =m
取值范围	m: m 为字符串, 被监控的网卡标识。
功能描述	SynIPAnalyzer 被监控的网卡。

3.1.1.7.2.2 ForwardIP

配置项	ForwardIP
-----	------------------

节	[BoardId=x]
书写格式	ForwardIP =m
取值范围	m: m 为字符串, 取值形式为 IPv4 地址, 默认值为 127.0.0.1。
功能描述	SynIPAnalyzer 用来转发 RTP 数据的 IP 地址。

3.1.1.7.2.3 ForwardPort

配置项	ForwardPort
节	[BoardId=x]
书写格式	ForwardPort =m
取值范围	m: m 为数字, 默认为 20000。
功能描述	SynIPAnalyzer 用来转发 RTP 数据的端口。

3.1.1.7.2.4 MonitorType

配置项	MonitorType
节	[BoardId=x]
书写格式	MonitorType =m
取值范围	m=0: 混合监控模式(默认值); m=1: 目标映射模式
功能描述	SynIPAnalyzer 的监控类型。
注意事项	<ul style="list-style-type: none"> ● 将监控类型设置为目标映射模式 (m=1) 时, 需要在程序中显式调用函数 SsmIPRAAddStationToMap 和 SsmIPRRmvStationFromMap(Ex) 来管理 Station, 如果不调用 SsmIPRAAddStationToMap 来增加需要监控的目标 Station, 则驱动不能监控到任何话机。 ● 如果希望将“目标映射模式”设置为初始监控类型, 建议直接通过本配置项进行设置。因为有可能在调用设置监控类型函数 SsmIPRSetMonitorType 之前便监控到会话和 Station 的存在, 从而耗费多余的时间和资源来处理监控到的会话。

3.1.1.7.2.5 RTPTimeOut

配置项	RTPTimeOut
节	[BoardId=x]
书写格式	RTPTimeOut =m

取值范围	m: m 为数字, 当一个 Session 长时间没检测到 RTP 时, 自动认为该 Session 已经断开, 其中值为 0 时表示不进行 RTP 超时检测, 默认值为 15, 单位为: 秒。
功能描述	设置检测 RTP 超时的时间。

3.1.1.7.2.6 RtpFwdCtrl

配置项	RtpFwdCtrl
节	[BoardId=x]
书写格式	RtpFwdCtrl =m
取值范围	m=0: RTP 地址与 E_RCV_IPR_MEDIA_SESSION_STARTED 所给的两个地址完全一致或任意一个一致即进行 RTP 转发 (默认值); m=1: RTP 地址与 E_RCV_IPR_MEDIA_SESSION_STARTED 所给的两个地址必须完全一致才进行 RTP 转发。
功能描述	设置 SynIPAnalyzer 的 RTP 转发规则。

3.1.1.7.2.7 ThreadPairs

配置项	ThreadPairs
节	[BoardId=x]
书写格式	ThreadPairs =m
取值范围	m 为非零整数, 用于设置 IPA 主工作线程数量, 默认值为 2。对于大容量 IP 录音环境建议设置此配置项, 取值为 CPU 的数量, 最大值 64。
功能描述	设置 SynIPAnalyzer 的主工作线程数量。

3.1.1.7.2.8 IGMPEventEnable

配置项	IGMPEventEnable
节	[BoardId=x]
书写格式	IGMPEventEnable =m
取值范围	m=0: 不检测 E_RCV_IPR_IGMP_MESSAGE 事件(默认值); m=1: 检测 E_RCV_IPR_IGMP_MESSAGE 事件。
功能描述	设置 SynIPAnalyzer 的 E_RCV_IPR_IGMP_MESSAGE 事件检测开关, 通常只有 ROIP 录音环境需要才打开此开关。

3.1.1.8 HMP 系列产品必须配置项 (CTI 系列 only)

3.1.1.8.1 HMP 系列产品 Server 端的必须配置项

3.1.1.8.1.1 TotalMediaForward

配置项	TotalMediaForward
节	[HMPRouter]
书写格式	TotalMediaForward =n
取值范围	n: n 为数字, 取值范围为 0~8, 默认值为 0。
功能描述	设置 HMP Client 的数量。

3.1.1.8.1.2 LocalIP[n]

配置项	LocalIP
节	[HMPRouter]
书写格式	LocalIP[n] =m
取值范围	n: HMP Client 编号, 取值为 0~ TotalMediaForward-1。 m: m 为字符串, 取值形式为 IPv4 地址, 默认值为 127.0.0.1。
功能描述	HMP Server 所在的主机 IP, 用于 HMP Server 与 HMP Client 之间的内部 RTP 收发, 以及控制 HMP Client。

3.1.1.8.1.3 TotalCh[n]

配置项	TotalCh
节	[HMPRouter]
书写格式	TotalCh[n]=m
取值范围	n: HMP Client 编号, 取值为 0~ TotalMediaForward-1。 m: m 为数字, 默认值为 0。
功能描述	设置 HMP Client 可以编解码的通道数量。

3.1.1.8.1.4 LocalPort[n]

配置项	LocalPort
节	[HMPRouter]
书写格式	LocalPort[n] =m
取值范围	n: HMP Client 编号, 取值为 0~ TotalMediaForward-1。 m: 端口号, 默认值为 5051。
功能描述	HMP Server 所在主机端口号, 用于控制 HMP Client。

3.1.1.8.1.5 RemoteIP[n]

配置项	RemoteIP
节	[HMPRouter]
书写格式	RemoteIP[n] =m
取值范围	n: HMP Client 编号, 取值为 0~ TotalMediaForward-1。 m: m 为字符串, 取值形式为 IPv4 地址, 默认值为 127.0.0.1。
功能描述	HMP Client 所在主机的 IP 地址。

3.1.1.8.1.6 RemotePort[n]

配置项	RemotePrt
节	[HMPRouter]
书写格式	RemotePort[n] =m
取值范围	n: HMP Client 编号, 取值为 0~ TotalMediaForward-1。 m: 端口号, 默认值为 5050。
功能描述	HMP Client 所在主机的端口号。

3.1.1.8.1.7 RtpIP[n]

配置项	RtpIP
节	[HMPRouter]

书写格式	RtpIP[n] =m
取值范围	n: HMP Client 编号, 取值 0~ TotalMediaForward-1。 m: m 为字符串, 取值形式为 IPv4 地址, 默认值为 192.168.0.1。
功能描述	配置 HMP Client 用于收发 RTP 的主机地址, 与外部通信, 相当于 B 型 IP 卡的 RTP 地址, 不能为 127.0.0.1, 否则对端 RTP 无法到达本端。可以与 RemoteIP 相同, 也可以不同, 主要看 HMP Client 所在主机的网口个数和配置。

3.1.1.8 MaxMediaThread

配置项	MaxMediaThread
节	[HMPRouter]
书写格式	MaxMediaThread =m
取值范围	m: 配置 HMP Server RTP 编解码和语音处理的线程数。取值范围: 0<m≤16, 默认值为 4.
功能描述	HMP Server 处理更多 RTP 资源和语音通道时, 需要使用更多线程来处理, 可以根据 CPU 核数和通道数来合理分配线程数。

3.1.2 常用配置项

3.1.2.1 设置 SHD/DTP 系列板卡的协议

3.1.2.1.1 CardType

配置项	CardType
节	[BoardId=x]
书写格式	CardType=n
取值范围	n=0: E1 接口 (n=0 为缺省值); n=1: T1 接口; n=2: J1 接口。
功能描述	通过该配置项和 PcmSSx 配置项, 使系统可以在 T1/J1 ISDN 环境下使用。同时, 您需要根据线路语音编码格式, 修改以下配置项: 1. [BoardId=x] 节语音 DSP 接收/发送语音编码配置项 DefaultVoiceFormat 。 2. 若为 ISDN 线路, 则修改 [ISDN] 节语音编码格式配置项 UserVoiceFormat 或 NetVoiceFormat 。

3.1.2.2 设置驱动程序抛出的事件

3.1.2.2.1 DefaultEventOutput

配置项	DefaultEventOutput
节	[SystemConfig]
书写格式	DefaultEventOutput=m
取值范围	m=0: 不输出任何事件; m=1: 只输出常用事件。有关常用事件的更多信息请参见第 1 章中“ 驱动程序抛出的事件 ”部分内容; m=2: 输出全部事件。 缺省值为 1。
功能描述	设置驱动程序的事件过滤器。更多信息请参见第 1 章中“ 事件的过滤 ”部分内容。

3.1.2.2.2 OvrlEnrgEventOut

配置项	OvrlEnrgEventOut
节	[SystemConfig]
书写格式	OvrlEnrgEventOut=n
取值范围	n=0: E_CHG_OvrlEnrgLevel 事件不输出; n=1: E_CHG_OvrlEnrgLevel 事件输出, 其他非 0 值将被当做 1 处理。 缺省值为 0。
功能描述	设置驱动程序的 E_CHG_OvrlEnrgLevel 事件过滤器。 E_CHG_OvrlEnrgLevel 事件输出的前提条件是 DefaultEventOutput 不为 0。

3.1.2.3 设置驱动程序内部事件队列的深度

3.1.2.3.1 MaxEventPerChannel

配置项	MaxEventPerChannel
节	[SystemConfig]
书写格式	MaxEventPerChannel=n

取值范围	n>0，单位为事件的个数，缺省值为100。
功能描述	设置一个通道的事件队列的深度。当应用程序以 wEvent=0xffff 作为参数调用 SsmSetEvent 函数来启用事件等待或事件回调模式时，驱动程序会使用本配置项的设定值为事件队列分配内存。

3.1.2.3.2 MaxUserEventSize

配置项	MaxUserEventSize
节	[SystemConfig]
书写格式	MaxUserEventSize=n
取值范围	n≥0，缺省值为0（表示禁止应用程序向驱动程序添加自定义事件）。
功能描述	在驱动程序的一次硬件中断时间间隔（8ms）内，应用程序可以向驱动程序内部的事件输出队列中添加的自定义事件的最大数量。

3.1.2.4 设置状态机的通用参数

3.1.2.4.1 MaxWaitAutoDialAnswerTime

配置项	MaxWaitAutoDialAnswerTime
节	[SystemConfig] / [SS1Config] / [ISUP] / [TUP] / [ISDN] / [SIP]
书写格式	MaxWaitAutoDialAnswerTime=t
取值范围	t：最大等待时间，单位为秒，缺省值因通道类型而异，具体为： ◆ 模拟中继线通道：25秒； ◆ SS1通道：90秒； ◆ ISUP通道：180秒； ◆ TUP通道：60秒； ◆ ISDN通道：60秒； ◆ SIP通道：60秒。
功能描述	在去话呼叫过程中，当通道状态迁移到“WaitAnswer”后，等待被叫用户摘机的最大等待时间。有关 AutoDial 任务的执行过程请参见第1章中下列部分内容： ◆ 模拟中继线通道：“ 模拟中继线通道的状态转移 ”； ◆ SS1通道：“ 中国1号信令的状态转移 ”中的 T6 定时器； ◆ ISUP通道：“ ISUP通道的状态迁移 ”中的 T4 定时器； ◆ TUP通道：“ TUP通道的状态迁移 ”中的 T4 定时器； ◆ ISDN通道：“ ISDN通道的状态迁移 ”中的 WaitAnswerTimer 定时器； ◆ SIP通道：“ SIP通道的状态转移图 ”中的 evTimeout(*2)定时器。

注意事项	模拟中继线通道的配置项使用[SystemConfig]配置节；SS1 通道的配置项使用[SS1Config]配置节；ISUP 通道的配置项使用[ISUP]配置节；TUP 通道的配置项使用[TUP]配置节；ISDN 通道的配置项使用[ISDN]配置节；SIP 通道的配置项使用[SIP]配置节。
------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3.1.2.4.2 RingToPending

配置项	RingToPending
节	[TUP] / [ISUP]
书写格式	RingToPending=m
取值范围	m=0：响应对端的拆线消息，开始拆除连接； m=1：不响应对端交换机的拆线消息，通道迁移到“挂起”状态，并通过设置挂起原因为 PEND_RemoteHangupOnRinging，由应用程序自行决定何时拆除连接。 缺省值为 0。
功能描述	来话呼叫过程中，当本端处于“振铃”状态时，如果对端交换机放弃呼叫，会向本端发送拆线消息。当驱动程序收到此拆线消息后，本配置项用于控制通道迁移到下一个状态的路径。有关本配置项的更多信息请参见第 1 章中通道的状态转移部分内容。
注意事项	<ul style="list-style-type: none">● 本配置项只适用于 TUP 通道和 ISUP 通道；● TUP 通道的配置项使用[TUP]配置节，ISUP 通道的配置项使用[ISUP]配置节。

3.1.2.4.3 AutoClearCallerIdBufOnHangup

配置项	AutoClearCallerIdBufOnHangup
节	[SystemConfig]
书写格式	AutoClearCallerIdBufOnHangup=b
取值范围	b=0：否； b=1：是（缺省值）。
功能描述	呼叫结束时，本配置项决定是否由驱动程序自动清除保存主叫方号码缓冲区和扩展缓冲区。如果是 ISUP 通道或 TUP 通道，保存被叫方号码和第 1 被叫号码的缓冲区也将被清空。
注意事项	本配置项要求驱动程序的版本为 2.1 或更高。

3.1.2.4.4 AutomaticAisGeneration

配置项	AutomaticAisGeneration
-----	-------------------------------

节	[BoardId=x]
书写格式	AutomaticAisGeneration[n]=b
取值范围	n: 逻辑 Pcm 编号, 从 0 开始编号; b=0: 否; (缺省值); b=1: 是。
功能描述	当输入信号中断时, 是否启用自动生成 AIS 消息。
注意事项	<ul style="list-style-type: none"> ● 本配置项仅用于 8E1/16E1 的链路; ● 对第 4m 个逻辑 PCM 设置本配置项时, 系统同时会自动设置第 (4*m+1) 个、第 (4*m+2) 个、第 (4*m+3) 个逻辑 PCM 的 AutomaticAisGeneration (其中 m 是使逻辑 Pcm 索引有意义的自然数); ● 本配置项要求驱动程序的版本为 Ver4.7.3.1 或更高。

3.1.2.4.5 AutomaticRaiGereration

配置项	AutomaticRaiGereration
节	[BoardId=x]
书写格式	AutomaticRaiGereration =b
取值范围	b=0: 否 (缺省值); b=1: 是。
功能描述	当输入信号丢失时, 是否启用自动生成 RAI 消息。
注意事项	<ul style="list-style-type: none"> ● 本配置项仅用于 8E1/16E1 的链路; ● 对第 4m 个逻辑 PCM 设置本配置项时, 系统同时会自动设置第 (4*m+1) 个、第 (4*m+2) 个、第 (4*m+3) 个逻辑 PCM 的 AutomaticAisGeneration (其中 m 是使逻辑 Pcm 索引有意义的自然数); ● 本配置项要求驱动程序的版本为 Ver4.7.3.1 或更高。

3.1.2.4.6 IgnoreBlockInGra

配置项	IgnoreBlockInGra
节	[TUP]

书写格式	IgnoreBlockInGra=b
取值范围	b=0: 否(缺省值); b=1: 是。
功能描述	是否忽略 GRA 中出现的状态指示域。
注意事项	本配置项仅在 TUP 接续中有意义。

3.1.2.4.7 AllowTimeoutInSpyISDN

配置项	AllowTimeoutInSpyISDN
节	[BoardId=x]
书写格式	AllowTimeoutInSpyISDN=b
取值范围	b=1: 通道长时间处于 S_SPY_RCVPHONUM 状态超时后复位到 S_SPY_STANDBY 状态。(默认值); b=0: 通道长时间处于 S_SPY_RCVPHONUM 状态超时后无复位操作。
功能描述	用于设置通道长时间处于 S_SPY_RCVPHONUM 状态超时后是否复位到 S_SPY_STANDBY 状态。
注意事项	本配置项只适用于 DTP 卡。禁用这个配置可能导致不完整的通话过程使系统在 S_SPY_RCVPHONUM 状态长时间挂起。

3.1.2.5 设置数字线路上的语音编码格式

3.1.2.5.1 DefaultVoiceFormat

配置项	DefaultVoiceFormat
节	[BoardId=x]
书写格式	DefaultVoiceFormat=m
取值范围	m=6: A-Law; m=7: μ-Law。
功能描述	设置数字线路(包括 DST 系列板卡所连接的数字用户线和 SHD/DTP 系列板卡所连接的数字中继线)上 B 信道语音数据的编码格式。 北美地区和日本通常使用 μ-Law, 欧洲和中国使用 A-Law。B 信道语音数据的编码格式可以通过查阅交换机或电话机的相关文档来确定, 也可以通过实验方法来确定。

注意事项	<ul style="list-style-type: none">● 本配置项只适用于 DST、SHD、DTP 系列以及 C 型传真板卡；● 对于 C 型传真卡，此配置项将设置传真工作的码流格式（A-Law 或 μ-Law）；● 如果本配置项的设定值与实际的 B 信道语音数据编码格式不一致，采集到的语音数据会出现杂音。
------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3.1.2.6 设置信号音检测器

3.1.2.6.1 设置数字卡的信号音检测开关

3.1.2.6.1.1 DefaultToneCheckState

配置项	DefaultToneCheckState
节	[BoardId=x]
书写格式	DefaultToneCheckState=n
取值范围	n=0: 关闭 (缺省值); n=1: 使能信号音检测。
功能描述	设置数字卡/IP 卡的信号音检测使能开关。当 n=1 时，跟模拟卡一样，信号音检测器的开启或关闭受通道状态迁移影响。

3.1.2.6.2 设置噪声滤波器的工作参数

3.1.2.6.2.1 MinimumVoiceDetermineEnergy

配置项	MinimumVoiceDetermineEnergy
节	[SystemConfig]
书写格式	MinimumVoiceDetermineEnergy=e
取值范围	e: e≥700, 缺省值为 100000。本参数的单位与 DB 的换算方法请参见第 1 章中“ 信号音检测器 ”节中的“FFT 模块”部分。
功能描述	设置线路上的噪音判别门限，详细信息请参见函数 SsmSetMinVocDtrEnergy 的说明。

3.1.2.6.3 设置脉宽滤波器的工作参数 T

3.1.2.6.3.1 ToneHighFilterPoint

请参见 [ToneLowFilterPoint](#)

3.1.2.6.3.2 ToneLowFilterPoint

配置项	ToneHighFilterPoint ToneLowFilterPoint
节	[SystemConfig]
书写格式	ToneHighFilterPoint= t_H ToneLowFilterPoint= t_L
取值范围	t_H : 高电平的最小保持时间, 单位为 16 毫秒, $t_H > 0$, 取值范围: 缺省值为 4。 t_L : 低电平的最小保持时间, 单位为 16 毫秒, $t_L > 0$, 取值范围: 缺省值为 10。
功能描述	ToneHighFilterPoint 设置信号音的最小保持时间。如果来话信号中出现了信号音的频率特征, 但其保持时间小于本参数的设定值, 驱动程序认为是干扰信号。 ToneLowFilterPoint 设置确认信号音消失的最小保持时间。在已经判定了来话信号中出现信号音后, 如果该频率特征消失的时间大于本参数的设定值, 驱动程序才确认消失。

3.1.2.6.4 设置第 1 呼叫进程音检测器

3.1.2.6.4.1 设置频率检测器的工作参数

3.1.2.6.4.1.1 TonePara

配置项	TonePara
节	[SystemConfig]
书写格式	TonePara =f1,b1,f2,b2,r
取值范围	f1: 第 1 个中心频率, 单位为 Hz, 取值范围: 300~3400, 缺省值为 450。 b1: 第 1 个中心频率的带宽, 单位为 Hz, 取值范围: 40~120, 缺省值为 80。 f2: 第 2 个中心频率, 单位为 Hz, 取值范围: 300~3400, 缺省值为 0。 b2: 第 2 个中心频率的带宽, 单位为 Hz, 取值范围: 40~120, 缺省值为 0。 r: 判定门限, 单位: %, 取值范围: 15~80, 缺省值为 50。 更多信息请参见函数 SsmSetTonePara 的说明。
功能描述	设置呼叫进程音检测器中频率检测器的工作参数, 详细内容请参见第 1 章中 呼叫进程音检测器 部分。

3.1.2.6.4.2 设置拨号音检测器的工作参数

3.1.2.6.4.2.1 IsDialToneDetermineTime

配置项	IsDialToneDetermineTime
节	[SystemConfig]
书写格式	IsDialToneDetermineTime=t
取值范围	t: 最小保持时间, 单位为毫秒, $t \geq 1300$, 缺省值为 1500。
功能描述	设置第 1 呼叫进程音检测器的拨号音检测参数。详细内容请参见函数 SsmSetIsDialToneDtrTime 的说明。

3.1.2.6.4.3 设置回铃音检测器的工作参数

3.1.2.6.4.3.1 RingEchoTonePara

配置项	RingEchoTonePara
节	[SystemConfig]
书写格式	RingEchoTonePara=t _H ,t _L
取值范围	t _H : 高电平保持时间, 取值范围: $300 \leq t_H \leq 2500$, 单位为毫秒, 缺省值为 1000。 t _L : 低电平保持时间, 取值范围: $800 \leq t_L \leq 6000$, 单位为毫秒, 缺省值为 4000。
功能描述	设置回铃音检测器的高电平保持时间和低电平保持时间。详细内容请参见函数 SsmSetRingEchoTonePara 的说明。

3.1.2.6.4.4 设置忙音检测器的工作参数

3.1.2.6.4.4.1 BusyTonePeriod

请参见 [IsBusyToneDetermineCount](#)

3.1.2.6.4.4.2 IsBusyToneDetermineCount

配置项	BusyTonePeriod IsBusyToneDetermineCount
节	[SystemConfig]

书写格式	格式 1: BusyTonePeriod=p ₁ 格式 2: BusyTonePeriod=p ₁ ,p ₂ 格式 3: BusyTonePeriod=p ₁ ,p ₂ ,p ₃ 格式 4: BusyTonePeriod=p ₁ ,p ₂ ,p ₃ ,p ₄ IsBusyToneDetermineCount=n
取值范围	p ₁ ,p ₂ ,p ₃ ,p ₄ : 忙音周期, 单位为毫秒, 取值范围: 200≤p _i ≤2000, 缺省值 700ms。 n: 忙音周期的最小持续个数, 1≤n≤10, 缺省值为 2。
功能描述	设置第 1 呼叫进程音检测器 中忙音检测器的工作参数。详细内容请参见第 1 章中 忙音检测器 部分。 BusyTonePeriod 设置忙音周期, 最多可以设置 4 个, 中间用“,”隔开。格式 1、格式 2、格式 3、格式 4 分别适用于设置 1~4 个周期的情况, 详细内容请参见函数 SsmSetBusyTonePeriodEx 的说明; IsBusyToneDetermineCount 设置忙音周期的最小持续个数。
使用范例	BusyTonePeriod=700 //检测 1 种忙音, 周期为 700 毫秒 BusyTonePeriod=700,1400 //检测 2 种忙音, 周期分别为 700 毫秒和 1400 毫秒

3.1.2.6.4.5 设置 Kewl Start 功能的工作参数

3.1.2.6.4.5.1 EnableKewlStart

配置项	EnableKewlStart
节	[SystemConfig]
书写格式	EnableKewlStart=m
取值范围	m=0: 关闭 Kewl Start 功能 (默认值) m=1: 开启 Kewl Start 功能
功能描述	设置 Kewl Start 检测远端挂机功能是否开启。

3.1.2.6.4.5.2 KSVoltageThreshold

配置项	KSVoltageThreshold
节	[SystemConfig]
书写格式	KSVoltageThreshold=n
取值范围	检测远端挂机的门限电压, 单位伏特, 取值范围: n≥0, 默认值为 0

功能描述	设置 Kewl Start 功能检测远端挂机的门限电压。更多信息请参见第 1 章中“ 模拟电话线的电压变化 ”部分内容。
------	------------------------------------------------------------------------------

3.1.2.6.4.5.3 KSKeepTime

配置项	KSKeepTime
节	[SystemConfig]
书写格式	KSKeepTime=m
取值范围	保持时间，单位毫秒，取值范围：16 ~ 65535，默认值为 256
功能描述	设置 Kewl Start 功能检测远端挂机的过滤时间。更多信息请参见第 1 章中“ 模拟电话线的电压变化 ”部分内容。

3.1.2.6.4.6 设置自定义信号音检测器的工作参数

3.1.2.6.4.6.1 AppointedToneAnalyzerSwitch

配置项	AppointedToneAnalyzerSwitch
节	[SystemConfig]
书写格式	AppointedToneAnalyzerSwitch=m
取值范围	m=0: 关闭； m=1: 开启，检测连续性信号音，此时配置项 IsAppointedToneDetermineTime 有效； m=2 : 开启，检测周期性信号音，此时配置项 AppointedTonePara 和 IsAppointedToneDetermineCount 有效
功能描述	设置第 1 呼叫进程音检测器的自定义信号音检测器的工作状态。详细内容请参见第 1 章中 自定义信号音检测器 部分。

3.1.2.6.4.6.2 IsAppointedToneDetermineTime

配置项	IsAppointedToneDetermineTime
节	[SystemConfig]
书写格式	IsAppointedToneDetermineTime=t
取值范围	t≥16: 最小保持时间，单位为毫秒，缺省值为 80。

注意事项	<ul style="list-style-type: none">只有当配置项 AppointedToneAnalyzerSwitch 设置为 1 时本配置项才有效；如果本配置项的设定值小于配置项 IsDialToneDetermineTime 的设置值，会导致无法正常检测拨号音；如果本配置项的设定值大于 IsDialToneDetermineTime 的配置值，会导致无法检测自定义的连续性信号音。
------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3.1.2.6.4.6.3 AppointedTonePara

请参见 [IsAppointedToneDetermineCount](#)

3.1.2.6.4.6.4 IsAppointedToneDetermineCount

配置项	AppointedTonePara IsAppointedToneDetermineCount
节	[SystemConfig]
书写格式	AppointedTonePara= t_H, t_L IsAppointedToneDetermineCount= n
取值范围	t_H : 高电平保持时间, $16 \leq t_H \leq 6000$, 单位为毫秒, 缺省值为 2000。 t_L : 低电平保持时间, $16 \leq t_L \leq 6000$, 单位为毫秒, 缺省值为 2000。 n : 自定义周期性信号音的最小持续个数, $1 \leq n \leq 10$, 缺省值为 2。
功能描述	设置第 1 呼叫进程音检测器中自定义信号音检测器的周期性信号音工作参数。 AppointedTonePara 设置高电平保持时间和低电平保持时间, IsAppointedToneDetermineCount 设置最小计数周期。
注意事项	只有当配置项 AppointedToneAnalyzerSwitch 的设定值为 2 时, 本配置项才有效。

3.1.2.6.5 设置第 2 呼叫进程音检测器

3.1.2.6.5.1 设置工作状态

3.1.2.6.5.1.1 Enable2ndToneAnalyzer

配置项	Enable2ndToneAnalyzer
节	[SystemConfig]
书写格式	Enable2ndToneAnalyzer= b
取值范围	$b=1$: 开启; $b=0$: 关闭。

功能描述	设置第 2 呼叫进程音检测器的工作状态。详细内容请参见函数 SsmStart2ndToneAnalyzer 的说明。
------	----------------------------------------------------------------------------

3.1.2.6.5.1.2 Check2ndToneOnAutoDial

配置项	Check2ndToneOnAutoDial
节	[SystemConfig]
书写格式	Check2ndToneOnAutoDial=b
取值范围	b=1: 是; b=0: 否（缺省值）。
功能描述	设置第 2 呼叫进程音检测器的检测结果是否影响模拟中继线通道的状态迁移。更多信息请参见第 1 章中“ 呼叫进程音检测器 ”部分内容。

3.1.2.6.5.2 设置频率检测器的工作参数

3.1.2.6.5.2.1 2ndTonePara

配置项	2ndTonePara
节	[SystemConfig]
书写格式	TonePara =f ₁ ,b ₁ ,f ₂ ,b ₂ ,r
取值范围	f ₁ : 第 1 个中心频率, 单位为 Hz, 取值范围: 300~3400, 缺省值为 450; b ₁ : 第 1 个中心频率的带宽, 单位为 Hz, 取值范围: 40~120, 缺省值为 80; f ₂ : 第 2 个中心频率, 单位为 Hz, 取值范围: 300~3400, 缺省值为 0; b ₂ : 第 2 个中心频率的带宽, 单位为 Hz, 取值范围: 40~120, 缺省值为 0; r: 判定门限, 单位: %, 取值范围: 15~80, 缺省值为 50; 更多信息请参见函数 SsmSet2ndTonePara 的说明。
功能描述	设置第 2 呼叫进程音检测器中频率检测器的工作参数, 详细内容请参见第 1 章中 呼叫进程音检测器 部分。

3.1.2.6.5.3 设置拨号音检测器的工作参数

3.1.2.6.5.3.1 2ndIsDialToneDetermineTime

配置项	2ndIsDialToneDetermineTime
节	[SystemConfig]

书写格式	<code>2ndIsDialToneDetermineTime=t</code>
取值范围	t : 拨号音的最小保持时间, 单位为毫秒, $n \geq 1300$, 缺省值为 1500。
功能描述	设置第 2 呼叫进程音检测器的拨号音检测参数。详细内容请参见函数 SsmSet2ndIsDialToneDtrTime 的说明。

3.1.2.6.5.4 设置回铃音检测器的工作参数

3.1.2.6.5.4.1 2ndRingEchoTonePara

配置项	2ndRingEchoTonePara
节	[SystemConfig]
书写格式	<code>2ndRingEchoTonePara=t_H,t_L</code>
取值范围	t_H : 高电平保持时间, $300 \leq t_H \leq 2500$, 单位为毫秒, 缺省值为 1000; t_L : 低电平保持时间, $800 \leq t_L \leq 6000$, 单位为毫秒, 缺省值为 4000。
功能描述	设置第 2 呼叫进程音检测器的回铃音检测器的高电平保持时间和低电平保持时间。详细内容请参见函数 SsmSet2ndRingEchoTonePara 的说明。

3.1.2.6.5.5 设置忙音检测器的工作参数

3.1.2.6.5.5.1 2ndBusyTonePeriod

请参见 [2ndIsBusyToneDetermineCount](#)

3.1.2.6.5.5.2 2ndIsBusyToneDetermineCount

配置项	2ndBusyTonePeriod 2ndIsBusyToneDetermineCount
节	[SystemConfig]
书写格式	格式 1: <code>2ndBusyTonePeriod=p₁</code> 格式 2: <code>2ndBusyTonePeriod=p₁,p₂</code> 格式 3: <code>2ndBusyTonePeriod=p₁,p₂,p₃</code> 格式 4: <code>2ndBusyTonePeriod=p₁,p₂,p₃,p₄</code> <code>2ndIsBusyToneDetermineCount=n</code>
取值范围	p_1, p_2, p_3, p_4 : 忙音周期, 单位为毫秒, 取值范围: $200 \leq p_i \leq 2000$, 缺省值 700ms。 n : 忙音周期的最小持续个数, $1 \leq n \leq 10$, 缺省值为 2。

功能描述	设置第 2 呼叫进程音检测器中忙音检测器的工作参数。 2ndBusyTonePeriod 设置忙音周期，最多可以设置 4 个，中间用 “,” 隔开。格式 1、格式 2、格式 3、格式 4 分别适用于设置 1~4 个周期的情况，详细内容请参见函数 SsmSetBusyTonePeriodEx 的说明。 2ndIsBusyToneDetermineCount 设置忙音周期的最小持续个数。
使用范例	2ndBusyTonePeriod=700 //检测 1 种忙音，周期为 700 毫秒 2ndBusyTonePeriod=700,1400 //检测 2 种忙音，周期分别为 700 毫秒和 1400 毫秒

3.1.2.6.5.5.3 MaxBsTnOffTime

配置项	MaxBsTnOffTime
节	[SystemConfig]
书写格式	MaxBsTnOffTime=t
取值范围	t≥400：单位为毫秒，缺省值为 2000 毫秒。
功能描述	设置忙音检测的保护时间。详细内容请参见第 1 章中 忙音检测器 部分。
注意事项	本配置项对第 2 呼叫进程音检测器也起作用。

3.1.2.6.5.6 设置自定义信号音检测器的工作参数

3.1.2.6.5.6.1 2ndAppointedToneAnalyzerSwitch

配置项	2ndAppointedToneAnalyzerSwitch
节	[SystemConfig]
书写格式	2ndAppointedToneAnalyzerSwitch=m
取值范围	m=0：关闭； m=1，开启，检测连续性信号音； m=2：开启，检测周期性信号音。
功能描述	设置第 2 呼叫进程音检测器的自定义信号音检测器的工作状态。详细内容请参见第 1 章中 自定义信号音检测器 部分。

3.1.2.6.5.6.2 2ndIsAppointedToneDetermineTime

配置项	2ndIsAppointedToneDetermineTime
节	[SystemConfig]

书写格式	2ndIsAppointedToneDetermineTime=t
取值范围	t≥16: 最小保持时间, 单位为毫秒, 缺省值为 80。
功能描述	设置自定义的连续性信号音的最小保持时间, 适用于第 2 呼叫进程音检测器的自定义信号音检测器。详细内容请参见第 1 章中 自定义信号音检测器 部分。
注意事项	<ul style="list-style-type: none">只有当配置项 2ndAppointedToneAnalyzerSwitch 设置 1 时本配置项才有效;如果本配置项的设定值小于配置项 2ndIsDialToneDetermineTime 的设置值, 会导致无法正常检测拨号音; 如果本配置项的设定值大于 2ndIsDialToneDetermineTime 的配置值, 会导致无法检测自定义的连续性信号音。

3.1.2.6.5.6.3 2ndAppointedTonePara

请参见 [2ndIsAppointedToneDetermineCount](#)

3.1.2.6.5.6.4 2ndIsAppointedToneDetermineCount

配置项	2ndAppointedTonePara 2ndIsAppointedToneDetermineCount
节	[SystemConfig]
书写格式	2ndAppointedTonePara=t _H ,t _L 2ndIsAppointedToneDetermineCount=n
取值范围	t _H : 高电平保持时间, 16≤t _H ≤6000, 单位为毫秒, 缺省值为 2000。 t _L : 低电平保持时间, 16≤t _L ≤6000, 单位为毫秒, 缺省值为 2000。 n: 自定义周期性信号音的最小持续个数, 1≤n≤10, 缺省值为 2。
功能描述	设置第 2 呼叫进程音检测器中 自定义信号音检测器 的周期性信号音的工作参数。 2ndAppointedTonePara 设置高电平保持时间和低电平保持时间, 2ndIsAppointedToneDetermineCount 设置最小计数周期。
注意事项	只有当配置项 2ndAppointedToneAnalyzerSwitch 的设定值为 2 时, 本配置项才有效。

3.1.2.6.6 设置传真进程音检测器

3.1.2.6.6.1 VoiceFreqF1Para

请参见 [VoiceFreqF2Para](#)

3.1.2.6.6.2 VoiceFreqF2Para

配置项	VoiceFreqF1Para VoiceFreqF2Para
节	[SystemConfig]
书写格式	VoiceFreqF1Para=f ₁ ,b ₁ ,r ₁ ,t ₁ VoiceFreqF2Para=f ₂ ,b ₂ ,r ₂ ,t ₂
取值范围	f ₁ , f ₂ : 中心频率, 单位为 Hz, 取值范围: 300~3400, f ₁ 的缺省值为 1100, f ₂ 的缺省值为 2100。 b ₁ , b ₂ : 带宽, 单位为 Hz, 取值范围: 40~120, 缺省值均为 80。 r ₁ , r ₂ : 带内能量占全频能量的百分比的最小门限值, 单位为%, 取值范围: 15 ~ 80, 缺省值均为 50。 t ₁ , t ₂ : 最小保持时间, 单位: 毫秒, 取值范围: ≥64, 缺省值均为 300。
功能描述	设置传真进程音检测器的工作参数。VoiceFreqF1Para 设置第 1 传真进程音检测器, VoiceFreqF2Para 设置第 2 传真进程音检测器。 详细内容请参见函数 SsmSetVoiceFxPara 的说明。

3.1.2.6.7 设置简易忙音检测器

3.1.2.6.7.1 设置工作状态

3.1.2.6.7.1.1 ToneAnalyzeAtRcvFsk

配置项	ToneAnalyzeAtRcvFsk
节	[BoardId=x]
书写格式	ToneAnalyzeAtRcvFsk=b
取值范围	b=0: 否 (缺省值); b=1: 是。
功能描述	当 FSK 数据接收器正在工作时, 是否开启简易忙音检测器。更多信息请参见第 1 章中 “ 简易忙音检测器 ” 部分内容。
注意事项	本配置项通常只用于 SHT 系列板卡的模拟中继线通道, 当 ToneDetectorMode=1 时生效。

3.1.2.6.7.2 设置噪声滤波器的工作参数

3.1.2.6.7.2.1 TAARFDetermineEnergy

配置项	TAARFDetermineEnergy
节	[BoardId=x]
书写格式	TAARFDetermineEnergy=e
取值范围	e≥700，缺省值为100000。 本参数的单位与DB的换算方法请参见第1章中“ 信号音检测器 ”节中的“FFT模块”部分。
功能描述	设置线路上的噪音判别门限，详细内容请参见第1章中 简易忙音检测器 部分。
注意事项	本配置项只适用于简易忙音检测器。

3.1.2.6.8 设置回铃音检测方式

3.1.2.6.8.1 设置回铃音检测方式

3.1.2.6.8.1.1 IsEchoQuickDetect

配置项	IsEchoQuickDetect
节	[BoardId=x]
书写格式	IsEchoQuickDetect=b
取值范围	b=0：必须检测到完整的回铃音，并且回铃音从低电平开始计算（缺省值）； b=1：根据EchoOnTolerance、EchoOffTolerance设置的误差检测到回铃音即可，并且回铃音可从低电平计算，也可从高电平开始计算。
功能描述	设置回铃音检测方式。

3.1.2.6.8.2 设置检测回铃音高电平时允许的误差

3.1.2.6.8.2.1 EchoOnTolerance

配置项	EchoOnTolerance
节	[BoardId=x]
书写格式	EchoOnTolerance=e

取值范围	0≤e≤50，缺省值为 40。
功能描述	设置检测回铃音高电平时允许的误差。若设置为 40，则允许 40% 的误差，即 1 秒高电平检测到 600 毫秒时就认为是回铃音高电平。
注意事项	与配置项 IsEchoQuickDetect、EchoOffTolerance 配合使用。

3.1.2.6.8.3 设置检测回铃音低电平时允许的误差

3.1.2.6.8.3.1 EchoOffTolerance

配置项	EchoOffTolerance
节	[BoardId=x]
书写格式	EchoOffTolerance =e
取值范围	0≤e≤50，缺省值为 50。
功能描述	设置检测回铃音低电平时允许的误差。若设置为 50，则允许 50% 的误差，即 4 秒低电平检测到 2 秒时就认为是回铃音低电平。
注意事项	与配置项 IsEchoQuickDetect、EchoOnTolerance 配合使用。

3.1.2.6.9 设置 CCIR 信号音检测器

3.1.2.6.9.1 CCIREnableDetector

配置项	CCIREnableDetector
节	[SystemConfig]
书写格式	CCIREnableDetector=b
取值范围	b=0，不开启 CCIR 信号音检测器（缺省值）； b=1，开启 CCIR 信号音检测器。
功能描述	是否开启 CCIR 信号音检测器。

注意事项	<p>此配置项目前仅支持以下型号板卡：</p> <p>SHT-8A/PCI SHT-8B/PCI SHT-8B/PCI/FAX SHT-8C/PCI/FAX SHT-8C/PCI/EC SHT-16A-CT/PCI SHT-16B-CT/PCI SHT-16B-CT/PCI/FAX SHT-16B-CT/cPCI SHT-16B-CT/cPCI/MP3 SHT-16B-CT/cPCI/FAX SHT-16C-CT/PCI/FAX SHT-16C-CT/PCI/EC</p> <p>SHT-2A/USB SHT-4A/USB SHT-2B/USB SHT-4B/USB</p>
------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3.1.2.6.10 设置 SelcallTone 信号音检测器

3.1.2.6.10.1 SelcallToneDetectMode

配置项	SelcallToneDetectMode
节	[SystemConfig]
书写格式	SelcallToneDetectMode=n
取值范围	n=0: 不开启 Selcall Tone 信号音检测器（缺省值）; n=1: 以主叫方式接收 Selcall Tone; n=2: 以数据方式接收 Selcall Tone。
功能描述	是否开启 Selcall Tone 信号音检测器。
注意事项	此配置项目前仅支持 SHT、ATP 系列板卡。

3.1.2.6.10.2 SelcallToneType

配置项	SelcallToneType
节	[SystemConfig]
书写格式	SelcallToneType=n

取值范围	n=0: 从配置项 SelcallTonePara 读出 Selcall tone 频率参数（缺省值）; n=1: CCIR; n=2: EIA; n=3: EEA; n=4: ZVEI-I; n=5: ZVEI-II; n=6: ZVEI-III; n=7: PZVEI; n=8: NATEL; n=9: DZVEI。
功能描述	Selcall Tone 类型。
注意事项	此配置项目前仅支持 SHT、ATP 系列板卡。

3.1.2.6.10.3 SelcallTonePara

配置项	SelcallTonePara
节	[SystemConfig]
书写格式	SelcallTonePara=p0,p1,p2,.....,p15
取值范围	无限制。
功能描述	Selcall Tone 频率参数。
注意事项	在 SelcallToneTyp=0 时有效。

3.1.2.6.11 设置背靠背忙音检测器工作参数

3.1.2.6.11.1 ExToneLevel

配置项	ExToneLevel
节	[SystemConfig]
书写格式	ExToneLevel=n
取值范围	取值范围： $50 \leq n \leq 1000$ ，单位 16 毫秒 (16ms)，缺省值 187。当 n 的取值小于 50 或大于 1000 时，驱动按默认值 187 处理。
功能描述	用于设置背靠背忙音检测器忙音检测周期。

注意事项	只有第 1 呼叫进程音检测器才具有背靠背忙音检测的功能。
------	------------------------------

3.1.2.7 设置机器应答检测器

3.1.2.7.1 EnableAMD

配置项	EnableAMD
节	[SystemConfig]
书写格式	EnableAMD=b
取值范围	b=0: 否; b=1: 是 (缺省值)。
功能描述	用于设置, 当信号音检测器正在工作时, 是否开启机器应答检测器。更多信息请参见第 1 章中“ 机器应答检测器 ”部分。

3.1.2.7.2 AMDNoSoundAfterDialTime

配置项	AMDNoSoundAfterDialTime
节	[SystemConfig]
书写格式	AMDNoSoundAfterDialTime=n
取值范围	单位毫秒 (ms), 缺省值 15000。
功能描述	用于判断拨号后线路上无声的时间是否超时。当无声的时间超过 AMDNoSoundAfterDialTime 的设定值时, 会抛出 E_CHG_AMD 事件 (E_CHG_AMD 事件的参数值为 5)。

3.1.2.7.3 AMDNoSoundTime

配置项	AMDNoSoundTime
节	[SystemConfig]
书写格式	AMDNoSoundTime=n
取值范围	单位毫秒 (ms), 缺省值 10000。
功能描述	用于判断检测到信号音或者彩铃后无声的时间是否超时。当无声的时间超过 AMDNoSoundTime 的设定值时, 会抛出 E_CHG_AMD 事件 (E_CHG_AMD 事件的参数值为 4)。

3.1.2.7.4 AMDTimeOut

配置项	AMDTIMEOUT
节	[SystemConfig]
书写格式	AMDTIMEOUT=n
取值范围	单位毫秒 (ms), 缺省值 70000。
功能描述	用于判断整个 AMD 检测过程是否超时。当整个 AMD 检测过程的时间超过 AMDTimeOut 的设定值时, 会抛出 E_CHG_AMD 事件 (E_CHG_AMD 事件的参数值为 3)。

3.1.2.7.5 AMDToneCount

配置项	AMDToneCount
节	[SystemConfig]

书写格式	AMDToneCount=n
取值范围	单位 16 毫秒 (16ms), 缺省值 10, 即 $10 \times 16\text{ms} = 160\text{ms}$ 。
功能描述	用于判断是否检测到信号音的最低时间门限。当整个检测到信号音的时间超过 AMDToneCount 的设定值时, 会抛出 E_CHG_AMD 事件 (E_CHG_AMD 事件的参数值为 1)。

3.1.2.7.6 AMDTOn

配置项	AMDTOn
节	[SystemConfig]
书写格式	AMDTOn=n
取值范围	单位毫秒 (ms), 缺省值 80。
功能描述	用于设置声音进入高电平的最低持续时间。

3.1.2.7.7 AMDTOff

配置项	AMDToff
节	[SystemConfig]
书写格式	AMDToff=n
取值范围	单位毫秒 (ms), 缺省值 400。
功能描述	用于设置声音进入低电平的最低持续时间。

3.1.2.7.8 AMDTimeA

配置项	AMDTimeA
节	[SystemConfig]
书写格式	AMDTimeA=n
取值范围	单位毫秒 (ms), 缺省值 600。
功能描述	用于设置真人摘机的问候语之前的最低静音持续时间。

3.1.2.7.9 AMDTimeB

配置项	AMDTimeB
节	[SystemConfig]
书写格式	AMDTimeB=n
取值范围	单位毫秒 (ms), 缺省值 80。
功能描述	用于设置真人摘机的问候语的最低持续时间。

3.1.2.7.10 AMDTimeC

配置项	AMDTimeC
节	[SystemConfig]
书写格式	AMDTimeC=n
取值范围	单位毫秒 (ms), 缺省值 1200。
功能描述	用于设置真人摘机的问候语的最大持续时间。

3.1.2.7.11 AMDTimeD

配置项	AMDTimeD
节	[SystemConfig]
书写格式	AMDTimeD=n
取值范围	单位毫秒 (ms), 缺省值 1000。
功能描述	用于设置真人摘机的问候语之后的最低静音持续时间。

3.1.2.7.12 AMDSilentEnergy

配置项	AMDSilentEnergy
节	[BoardId=x]
书写格式	AMDSilentEnergy=n
取值范围	能量数值, 缺省值 17000。
功能描述	用于设置声音能量是否为静音的阀值。

3.1.2.7.13 EnableAMDBeep

配置项	EnableAMDBeep
节	[SystemConfig]
书写格式	EnableAMDBeep=b
取值范围	b=0: 否 (缺省值), 使用原来的 AMD 检测方式; b=1: 是。
功能描述	用于设置当信号音检测器工作时, 是否开启彩铃和提示音区分检测功能。

3.1.2.7.14 AMDTOffEx

配置项	AMDOffEx
节	[SystemConfig]
书写格式	AMDOffEx=n
取值范围	单位毫秒 (ms), 缺省值等于 AMDOff 的配置值。
功能描述	用于设置真人摘机后问候语之中低电平的最大持续时间。

3.1.2.8 设置增强信号音检测器

3.1.2.8.1 ToneDetectorMode

配置项	ToneDetectorMode
节	[SystemConfig]
书写格式	ToneDetectorMode =b
取值范围	b=0, 使用原信号音检测器 (缺省值); b=1, 使用增强信号音检测器。

功能描述	选择采用哪一种信号音检测器。
注意事项	此配置项目前仅支持以下型号板卡: SHT-8A/PCI SHT-8B/PCI SHT-8B/PCI/FAX SHT-8C/PCI/FAX SHT-8C/PCI/EC SHT-16A-CT/PCI SHT-16B-CT/PCI SHT-16B-CT/PCI/FAX SHT-16B-CT/cPCI SHT-16B-CT/cPCI/MP3 SHT-16B-CT/cPCI/FAX SHT-16C-CT/PCI/FAX SHT-16C-CT/PCI/EC SHT-2A/USB SHT-4A/USB SHT-2B/USB SHT-4B/USB

3.1.2.8.2 VoiceOffDetermineTime

配置项	VoiceOffDetermineTime
节	[SystemConfig]
书写格式	VoiceOffDetermineTime =n
取值范围	n, 单位毫秒 (ms), 一般取回铃音周期的 1.2 倍, n=5000 ms 是缺省值。
功能描述	设置无声检测的时间。

3.1.2.8.3 MaxToneDetectorItem

配置项	MaxToneDetectorItem
节	[SystemConfig]
书写格式	MaxToneDetectorItem =N
取值范围	1≤N≤20,缺省值为 5。
功能描述	设置将要检测的信号音类型的总数。

3.1.2.8.4 ToneDetectorItem

配置项	ToneDetectorItem
-----	-------------------------

节	[SystemConfig]		
书写格式	ToneDetectorItem[n] = $M, F_1, F_2, F_3, T_1, T_2, T_3, C_{on}, C_{off}, F_{err}, R_{bw}, T_{err}, P_{end}, C_{cnt}, T_{clr}, P_{pop}, E_{event}, S_{state}, S_{stop}, W_e$		
	本配置项的个数与 MaxToneDetectorItem 的配置值 N 有关，其中 n 的取值范围为 0~N-1。增强检测器支持 3 种信号音，其各项取值范围如下：		
取值范围	参数	连续性信号音	周期性信号音
	M	取值范围：=0 信号音为连续音频信号，如拨号音、传真音 F_1 、传真音 F_2	取值范围：=1 信号音为周期信号音，如忙音、催挂音、回铃音
	$F_1(\text{Hz})$	取值范围：($\geq 300 \ \&\& \leq 3400$) 第 1 中心频率	取值范围：($\geq 300 \ \&\& \leq 3400$) 第 1 中心频率
	$F_2(\text{Hz})$	取值范围：($=0 \ (\geq 300 \ \&\& \leq 3400)$) 第 2 中心频率，单音频时=0	取值范围：($=0 \ (\geq 300 \ \&\& \leq 3400)$) 第 2 中心频率，单音频时=0
	$F_3(\text{Hz})$	无效	无效
	$T_1(\text{ms})$	取值范围： ≥ 16 高电平持续时间	取值范围： ≥ 16 高电平持续时间
	$T_2(\text{ms})$	无效	取值范围： ≥ 16 低电平持续时间
	$T_3(\text{ms})$	无效	取值范围： ≥ 16 第 3 段持续时间
	C_{on}	取值范围： $(>0 \ \&\& \leq 100)$ 高电平滤波点数(每个点时长为 16ms)	取值范围： $(>0 \ \&\& \leq 100)$ 高电平滤波点数(每个点时长为 16ms)
	C_{off}	取值范围： $(>0 \ \&\& \leq 100)$ 低电平滤波点数(每个点时长为 16ms)	取值范围： $(>0 \ \&\& \leq 100)$ 低电平滤波点数(每个点时长为 16ms)
	$F_{err}(\text{Hz})$	取值范围： $(>0 \ \&\& \leq 3400)$ 频率误差门限	取值范围： $(>0 \ \&\& \leq 3400)$ 频率误差门限
	$R_{bw}(\%)$	取值范围： $(\geq 15 \ \&\& \leq 80)$ 带内能量占总能量的最小百分比门限	取值范围： $(\geq 15 \ \&\& \leq 80)$ 带内能量占总能量的最小百分比门限
	$T_{err}(\%)$	取值范围： $(>0 \ \&\& <100)$ 高电平保持时间允许的最大误差门限(百分比)	取值范围： $(>0 \ \&\& <100)$ 高电平保持时间允许的最大误差门限(百分比)
	P_{end}	无效	取值范围：0 或 1 一个完整周期的判别方式。0 表示低电平用门限判断，判断条件：低电平时间 $\geq T_2$ 。1 表示低电平用误差判断，判断条件： $T_2 * (1 - T_{err}(\%)) \leq \text{低电平时间} \leq T_2 * (1 + T_{err}(\%))$ 。

	C _{cnt}	无效	取值范围: (>0 && <=10) 检测到第 C _{cnt} 个完整周期后抛出 E_CHG_ToneDetector 事件	无效
	T _{clr(ms)}	无效	取值范围: >= 16 持续 T _{clr(ms)} 的时间没有检测到一个完整周期时清除已有的周期计数	无效
	E _{event}	检测到信号音后向应用程序抛出的事件类型, 具体见注 1	检测到信号音后向应用程序抛出的事件类型, 具体见注 1	检测到信号音后向应用程序抛出的事件类型, 具体见注 1
	S _{state}	取值范围: 0 或 1 是否将事件同时抛给通道的状态机, 0 不抛出; 1 抛出	取值范围: 0 或 1 是否将事件同时抛给通道的状态机, 0 不抛出; 1 抛出	取值范围: 0 或 1 是否将事件同时抛给通道的状态机, 0 不抛出; 1 抛出
	S _{stop}	取值范围: 0 或 1 检测到信号音后是否停止该信号音检测。0 表示继续检测; 1 表示停止检测	取值范围: 0 或 1 检测到信号音后是否停止该信号音检测。0 表示继续检测; 1 表示停止检测	取值范围: 0 或 1 检测到信号音后是否停止该信号音检测。0 表示继续检测; 1 表示停止检测
	W _e	取值范围: 0 或 1 0 表示信号音检测器工作在 FFT 分析时; 1 表示工作在接收 FSK 时	取值范围: 0 或 1 0 表示信号音检测器工作在 FFT 分析时; 1 表示工作在接收 FSK 时	取值范围: 0 或 1 0 表示信号音检测器工作在 FFT 分析时; 1 表示工作在接收 FSK 时

注 1: 驱动检测到信号音后会抛出 [E_CHG_ToneDetector](#) 事件, 事件参数 dwParam 的低 2 字节表示信号音的具体类型, 新信号音检测器将配置项 ToneDetectorParamItem[n] 中的 E_{event} 赋值给 dwParam 的低 2 字节, 所以 E_{event} 的取值范围为 0~0xFFFF, 其中 0~9 已经被驱动定义, 具体可见下表:

配置值	驱动中对应的定义标识	说明	模拟通道呼叫状态机是否使用	应用程序能否使用
0	CHKTONE_NO_RESULT	检测正在进行	N	N
1	CHKTONE_DIALTONE	检测到拨号音	Y	Y
2	CHKTONE_BUSYTONE	检测到忙音	Y	Y
3	CHKTONE_ECHOTONE	检测到回铃音	Y	Y
4	CHKTONE_ECHO_NOVOICE	检测到回铃音出现无声	Y	N
5	CHKTONE_NOVOICE	检测到无声	Y	N
6	CHKTONE_VOICE	检测到说话声	Y	N
7	CHKTONE_VOICEF1	检测到 F ₁ 频率的信号音	Y	Y
8	CHKTONE_VOICEF2	检测到 F ₂ 频率的信号音	Y	Y
9	CHKTONE_APPOINTEDTONE	检测到用户指定的信号音类型	N	Y
0xA~0x001F	驱动保留		N	N
0x0020~0xFFFF	未被使用		N	Y

其中 CHKTONE_DIALTONE、CHKTONE_BUSYTONE、CHKTONE_ECHOTONE、CHKTONE_VOICEF1、CHKTONE_VOICEF2 若未被应用程序使用则模拟通道呼叫状态机将无法工作。

下表是默认配置时 ToneDetectorParamItem [n] 的配置值, 保证采用新模式时默认情况下配置和输出与原检测模式默认时一致。

	参数	DIALTONE	BUSYTONE	ECHOTONE	VOICEF1	VOICEF2	
M	0	1	1	0	0	0	
F ₁ (Hz)	450	450	450	1100	2100		
F ₂ (Hz)	0	0	0	0	0		
F ₃ (Hz)	0	0	0	0	0		
T ₁ (ms)	1500	350	1000	250	250		
T ₂ (ms)	0	350	4000	0	0		
T ₃ (ms)	0	0	0	0	0		
C _{on}	4	4	4	4	4		
C _{off}	10	10	10	10	10		
F _{err} (Hz)	30	30	30	30	30		
R _{bw} (%)	50	50	50	50	50		
T _{err} (%)	20	20	20	20	20		
P _{end}	0	1	0	0	0		
C _{nt}	0	2	1	0	0		
T _{clr} (ms)	0	3000	6000	0	0		
E _{event}	1	2	3	7	8		
S _{state}	1	1	1	1	1		
S _{stop}	0	0	0	0	0		
W _e	0	0	0	0	0		
功能描述	设置各信号音检测参数，检测各种指定的信号音。						

3.1.2.9 设置信号音发生器(CTI 系列)

3.1.2.9.1 DefaultSendToneFrequency

请参见 [DefaultSendToneVolume](#)

3.1.2.9.2 DefaultSendToneVolume

配置项	DefaultSendToneFrequency DefaultSendToneVolume
节	[BoardId=x]
书写格式	DefaultSendToneFrequency=f ₁ ,f ₂ DefaultSendToneVolume=v ₁ ,v ₂
取值范围	f ₁ : 音频的第 1 个频率，单位为 Hz，缺省值为 450。 f ₂ : 音频的第 2 个频率，单位为 Hz。f ₂ 为 0 表示单音频信号音，大于 0 表示双音频信号音。缺省值为 0。 v ₁ : 音频的第 1 个音量，取值范围: -7≤v ₁ ≤+6, -7 表示关闭该频率，大于 0 表示音量提升，小于 0 表示音量衰减，设置值乘以 3 即为分贝值，缺省值为 0。 v ₂ : 音频的第 2 个音量，取值范围及含义同 v ₁ ，缺省值为-7。

功能描述	设置信号音发生器的工作参数。DefaultSendToneFrequency 设置音频信号的频率，DefaultSendToneVolume 设置音量。
使用范例	DefaultSendToneFrequency=450, 600 //使用双音频信号音 DefaultSendToneVolume=0,0

3.1.2.10 设置电话会议参数(CTI 系列)

3.1.2.10.1 ClearInVoiceOnRxDtmf

配置项	ClearInVoiceOnRxDtmf
节	[BoardId=x]
书写格式	格式 1: ClearInVoiceOnRxDtmf=n,n,n,n 格式 2: ClearInVoiceOnRxDtmf=n,n,n,n,n,n,n,n 格式 3: ClearInVoiceOnRxDtmf=n,n,n,n,n,n,n,n,n,n,n,n,n,n 格式 4: ClearInVoiceOnRxDtmf=n
取值范围	n=0: 关闭 (默认值); n=1: 开启。
功能描述	设置是否启用 DTMF 钳位功能, 即是否开启 DTMF 上总线切音功能。更多信息请参见第 1 章中“ 分布式电话会议 ”部分内容。
注意事项	<ul style="list-style-type: none">● 格式 1 适用于采用 USB 总线方式的 SHT 系列语音盒、ATP 系列录音盒;● 格式 2 适用于 8 通道规格的 SHT 系列板卡;● 格式 3 适用于 16 通道规格的 SHT 系列板卡;● 格式 4 适用于 SHD 系列板卡、SHN 系列板卡;● 函数 SsmSetFlag (携带 F_ClearInVoiceOnRcvDtmf 参数) 也可实现相同功能。

3.1.2.10.2 ClearInVoiceOnRx450Hz

配置项	ClearInVoiceOnRx450Hz
节	[BoardId=x]
书写格式	格式 1: ClearInVoiceOnRx450Hz =n,n,n,n,n,n,n 格式 2: ClearInVoiceOnRx450Hz =n,n,n,n,n,n,n,n,n,n,n,n,n

取值范围	n=0: 关闭 (缺省值); n=1: 启用。
功能描述	设置是否启用信号音总线切音功能。
注意事项	<ul style="list-style-type: none">● 格式 1 适用于 8 通道规格的 SHT 系列板卡;● 格式 2 适用于 16 通道规格的 SHT 系列板卡。

3.1.2.10.3ConfMaxGroup

配置项	ConfMaxGroup
节	[SystemConfig]
书写格式	ConfMaxGroup=n
取值范围	n: n≥0 (缺省值为 64)。
功能描述	设置可以创建的会议室的最大个数。

3.1.2.10.4ConfDefaultMaxGroupMember

配置项	ConfDefaultMaxGroupMember
节	[SystemConfig]
书写格式	ConfDefaultMaxGroupMember=n
取值范围	n: n≥0 (缺省值为 64)。
功能描述	设置可以加入会议室的会议通道的最大个数。

3.1.2.10.5ConfDefaultMaxGroupSpeaker

配置项	ConfDefaultMaxGroupSpeaker
节	[SystemConfig]
书写格式	ConfDefaultMaxGroupSpeaker=n

取值范围	n: 会议通道的个数，缺省值为 24。
功能描述	设置会议室中具有发言权的会议通道的最大个数。主持人、主席、动态发言模式均具有发言权模式，背景音乐模式是否具有发言权取决于配置项 BackgroundVoicePriority 的设置。

3.1.2.10.6 ConfDefaultMaxGroupSpeaking

配置项	ConfDefaultMaxGroupSpeaking
节	[SystemConfig]
书写格式	ConfDefaultMaxGroupSpeaking=n
取值范围	n≤6，缺省值为 6。
功能描述	设置会议室中允许同时发言的最大通道数。

3.1.2.10.7 ConfJoinedbyEnergy

配置项	ConfJoinedbyEnergy
节	[SystemConfig]
书写格式	ConfJoinedbyEnergy=n
取值范围	n=0: 无抢占； n=2: 支持抢占式发言(缺省值)。
功能描述	设置电话会议中动态发言时，是否支持抢占式发言。

3.1.2.10.8 ConfMaxListener

配置项	ConfMaxListener
节	[SystemConfig]
书写格式	ConfMaxListener =n
取值范围	n: 会议通道数，缺省值为 480。
功能描述	设置会议室中允许加入的最大听众数。

3.1.2.10.9 ConfDefaultMaxSilenceTime

配置项	ConfDefaultMaxSilenceTime
节	[SystemConfig]
书写格式	ConfDefaultMaxSilenceTime=t
取值范围	t: 保持时间, 单位为秒, 缺省值为 5 秒。
功能描述	设置判断通道是否保持静默的最小保持时间。更多信息请参见第 1 章中“ 分布式电话会议 ”部分内容。
注意事项	该配置项已无效。

3.1.2.10.10 BackgroundVoicePriority

配置项	BackgroundVoicePriority
节	[SystemConfig]
书写格式	BackgroundVoicePriority=n
取值范围	n=1: 高于主持人, 具有专用的会议混音器; n=2: 低于主持人, 但高于主席(缺省值); n=3: 低于主席, 但高于动态发言; n=4: 低于动态发言。
功能描述	设置播放背景音乐的通道的发言模式的调度优先级。更多信息请参见第 1 章中“ 分布式电话会议 ”部分内容。

3.1.2.10.11 PlayVoicelsListen

配置项	PlayVoicelsListen
节	[SystemConfig]
书写格式	PlayVoicelsListen =n
取值范围	n=0: 此通道只说不听; n=1: 此通道可说可听。

功能描述	设置电话会议中用于播放背景音的通道是否可听。
------	------------------------

3.1.2.10.12 QuitDynKeepingInConf

配置项	QuitDynKeepingInConf
节	[SystemConfig]
书写格式	QuitDynKeepingInConf =n
取值范围	<p>n=0: (默认) 当本次调度同时发言的人数 n 少于最大同时发言人数 m 时, 上一次调度中曾经发言, 但本次调度没有发言的人, 根据上次调度中音量排序结果, 声音最大的前 m-n 个人依然占用本地总线, 依然拥有发言权。这样其不会引起 Barge In 的小声音如背景噪音, 就会被其他会议成员听到。</p> <p>n=1: 动态发言的会议成员停止讲话后是立刻取消其讲话权利, 防止背景噪音被其他成员听到。</p>
功能描述	动态发言的会议成员停止讲话后是否立刻取消其讲话权利, 防止背景噪音被其他成员听到。背景噪音是指不会引起 Barge In 的小声音。
注意事项	<ul style="list-style-type: none">● 本配置在 5.3.1.3 (含) 之后版本有效;● 本配置对所有具备会议能力的通道都有效。

3.1.2.11 设置放音操作

3.1.2.11.1 设置播放任务的终止条件

3.1.2.11.1.1 DefaultDtmfStopPlay

请参见 [DtmfStopPlayCharSet](#)

3.1.2.11.1.2 DtmfStopPlayCharSet

配置项	DefaultDtmfStopPlay DtmfStopPlayCharSet
节	[BoardId=x]

书写格式	<p>格式 1: DefaultDtmfStopPlay=b,b,b,b DtmfStopPlayCharSet={s,s,s,s}</p> <p>格式 2: DefaultDtmfStopPlay=b,b,b,b,b,b,b DtmfStopPlayCharSet={s,s,s,s,s,s,s}</p> <p>格式 3: DefaultDtmfStopPlay=b,b,b,b,b,b,b,b,b,b,b DtmfStopPlayCharSet={s,s,s,s,s,s,s,s,s,s,s,s,s}</p> <p>格式 4: DefaultDtmfStopPlay=b DtmfStopPlayCharSet=s</p>
取值范围	<p>b=0: 否 (缺省值), 此时 DtmfStopPlayCharSet 无效;</p> <p>b=1: 是, 此时 DtmfStopPlayCharSet 有效。</p> <p>s={0123456789*#abcd }, 可以任意个数的 DTMF 字符集, 缺省值为全集。</p>
功能描述	DefaultDtmfStopPlay 设置放音任务是否因 DTMF 检测器 检测到 DTMF 字符而终止, 如果设置为是, DtmfStopPlayCharSet 设置 DTMF 字符集。
注意事项	<ul style="list-style-type: none"> ● 格式 1 适用于 SHT 系列的 USB 语音盒; ● 格式 2、格式 3 分别适用于 8 通道、16 通道规格的 SHT 系列; ● 格式 4 适用于 SHD 系列、SHN 系列板卡。
使用范例	<pre>DefaultDtmfStopPlay=1 DtmfStopPlayCharSet={#} // SHD 系列: 检测到#就停止放音任务, 其余字符不起作用</pre>

3.1.2.11.1.3 HangupStopPlay

配置项	HangupStopPlay
节	[BoardId=x]
书写格式	<p>格式 1: HangupStopPlay=b,b,b,b 格式 2: HangupStopPlay=b,b,b,b,b,b,b 格式 3: HangupStopPlay=b,b,b,b,b,b,b,b,b,b,b,b 格式 4: HangupStopPlay=b</p>
取值范围	<p>b=0: 否 (缺省值);</p> <p>b=1: 是。</p>
功能描述	设置放音任务是否因驱动程序状态机检测到对端挂机而终止。

注意事项	<ul style="list-style-type: none">● 格式 1 适用于 SHT 系列的 USB 语音盒;● 格式 2、格式 3 分别适用于 8 通道、16 通道规格的 SHT 系列;● 格式 4 适用于 SHD 系列、SHN 系列板卡。
------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------

3.1.2.11.1.4 DefaultBargeInStopPlay

配置项	DefaultBargeInStopPlay
节	[BoardId=x]
书写格式	格式 1: DefaultBargeInStopPlay=b,b,b,b 格式 2: DefaultBargeInStopPlay=b,b,b,b,b,b,b,b 格式 3: DefaultBargeInStopPlay=b,b,b,b,b,b,b,b,b,b,b,b 格式 4: DefaultBargeInStopPlay=b
取值范围	b=0: 否 (缺省值); b=1: 是。
功能描述	设置放音任务是否因 Barge in 检测器 检测到语音活动而终止。
注意事项	<ul style="list-style-type: none">● 格式 1 适用于 SHT 系列的 USB 语音盒;● 格式 2、格式 3 分别适用于 8 通道、16 通道规格的 SHT 系列;● 格式 4 适用于 SHD 系列板卡、SHN 系列板卡。

3.1.2.11.2 设置驱动程序内部的放音缓冲区

3.1.2.11.2.1 PlayBufSize

配置项	PlayBufSize
节	[SystemConfig]
书写格式	PlayBufSize=k
取值范围	k: 放音缓冲区的长度, 单位为字节, 必须为 512 的整数倍, 缺省值为 32768 字节。
功能描述	设置驱动程序内部为每一个通道分配的放音缓冲区的大小。

3.1.2.11.3 设置文件序列播放任务的参数

3.1.2.11.3.1 MaxPlayFileList

配置项	MaxPlayFileList
节	[SystemConfig]
书写格式	MaxPlayFileList=n
取值范围	0≤n≤65535，缺省值为 256。
功能描述	设置允许应用程序调用 SsmAddToFileList 向驱动程序提交的语音文件数量的上限。

3.1.2.11.4 设置预加载文件播放任务的参数

3.1.2.11.4.1 MaxPlayIndexList

配置项	MaxPlayIndexList
节	[SystemConfig]
书写格式	MaxPlayIndexList=n
取值范围	0≤n≤512，缺省值为 256。
功能描述	设置调用一次 SsmPlayIndexList 或 SsmPlayIndexString 函数所允许的语音片段个数的上限。

3.1.2.11.5 设置内存序列播放任务的参数

3.1.2.11.5.1 MaxPlayMemList

配置项	MaxPlayMemList
节	[SystemConfig]
书写格式	MaxPlayMemList=n
取值范围	0≤n≤65535，缺省值为 256。
功能描述	设置允许应用程序调用 SsmAddToPlayMemList 向驱动程序提交的缓冲区数量的上限。

3.1.2.11.6 设置文件播放任务的参数

3.1.2.11.6.1 FastPlayTime

配置项	FastPlayTime
节	[SystemConfig]
书写格式	FastPlayTime=t
取值范围	t 表示一次快进或快退操作对应的步长，单位为毫秒，缺省值为 1000。
功能描述	设置 SsmFastFwdPlay 和 SsmFastBwdPlay 函数的步长。

3.1.2.11.7 设置播放操作的音量

3.1.2.11.7.1 DefaultPlayVolume

配置项	DefaultPlayVolume
节	[BoardId=x]
书写格式	格式 1: DefaultPlayVolume=v,v,v,v 格式 2: DefaultPlayVolume=v,v,v,v,v,v,v,v 格式 3: DefaultPlayVolume=v,v,v,v,v,v,v,v,v,v,v,v,v 格式 4: DefaultPlayVolume=v
取值范围	-7≤v≤+6，缺省值为 0。
功能描述	设置 SHD/SHT 系列板卡中音量调节器 A1 的增益。
注意事项	<ul style="list-style-type: none">● 格式 1 适用于 SHT 系列的 USB 语音盒和 ATP 系列的 USB 录音盒；● 格式 2 适用于 8 通道规格的 SHT 系列、ATP 系列板卡；● 格式 3 适用于 16 通道规格的 SHT 系列、ATP 系列板卡、DST 系列板卡；● 格式 4 适用于 SHD 系列板卡、DTP 系列卡、SHN 系列板卡。

3.1.2.11.8 设置播放操作的编码格式

3.1.2.11.8.1 DefaultPlayFormat

配置项	DefaultPlayFormat
节	[BoardId=x]

书写格式	格式 1: DefaultPlayFormat =v,v,v,v 格式 2: DefaultPlayFormat =v,v,v,v,v,v,v 格式 3: DefaultPlayFormat =v,v,v,v,v,v,v,v,v,v,v,v 格式 4: DefaultPlayFormat =v
取值范围	请参见 SsmRecToFile 的 nFormat 参数的 9 种格式，缺省值 v=6，具体不同的板卡支持的格式参见 1.3.2 板卡支持的 CODEC 。
功能描述	设置通道的缺省放音数据编码格式。
注意事项	<ul style="list-style-type: none"> ● 格式 1 适用于 SHT 系列的 USB 语音盒和 ATP 系列的 USB 录音盒； ● 格式 2 适用于 8 通道规格的 SHT 系列、ATP 系列板卡； ● 格式 3 适用于 16 通道规格的 SHT 系列、ATP 系列板卡、DST 系列板卡； ● 格式 4 适用于 SHD 系列板卡、DTP 系列卡、SHN 系列板卡。

3.1.2.11.9 设置播放操作的其它参数

3.1.2.11.9.1 DefaultPausePlayOnRxDtmf

配置项	DefaultPausePlayOnRxDtmf
节	[BoardId=x]
书写格式	格式 1: DefaultPausePlayOnRxDtmf=b,b,b,b 格式 2: DefaultPausePlayObRxDtmf=b,b,b,b,b,b,b 格式 3: DefaultPausePlayObRxDtmf=b,b,b,b,b,b,b,b,b,b,b 格式 4: DefaultPausePlayObRxDtmf=b 格式 5: DefaultPausePlayObRxDtmf=[0,29]:b
取值范围	b=0: 否; b=1: 是（缺省值）。
功能描述	当 DTMF 检测器在来话信号中检测到 DTMF 信号时，如果通道上正在执行语音播放的任务，为了保证 DTMF 检测器的准确性，驱动程序需要暂时停止语音的播放，待 DTMF 信号消失后，再自动恢复语音的播放。本配置项设置是否需要启动此项功能。 一般而言，录音通道应该设置为 0，其余类型通道设置为 1。
注意事项	<ul style="list-style-type: none"> ● 格式 1 适用于 SHT 系列、ATP 系列的 USB 语音盒/录音盒； ● 格式 2 适用于 8 通道规格的 SHT 系列、ATP 系列板卡； ● 格式 3 适用于 16 通道规格的 SHT 系列、ATP 系列板卡； ● 格式 4 适用于 SHD 系列板卡、SHN 系列板卡； ● 格式 5 适用于 30 通道规格的远端坐席接入模块； ● 本配置项不支持 16 路 C 型 SHT 系列板卡。

3.1.2.11.9.2 PlayFilterFlag

配置项	PlayFilterFlag
节	[BoardId=x]
书写格式	格式 1: PlayFilterFlag =b,b,b,b,b,b,b 格式 2: PlayFilterFlag =b,b,b,b,b,b,b,b,b,b,b,b,b
取值范围	b=0: 否 (缺省值); b=1: 是。
功能描述	当以较大的音量放音时, 如果对方此时发送 DTMF 过来, 可能会让 DTMF 检测器无法获取到正确的 DTMF 号码, 这时就需要开启该配置项, 以避免音量过大对 DTMF 接收产生影响。本配置项用于设置是否需要开启此项功能。
注意事项	<ul style="list-style-type: none">● 本配置项只适用于 SHT-8C/PCI/EC、SHT-16C/PCI/EC 和 SHT-16C/PCI/FAX 卡;● 格式 1 适用于 8 通道规格的 SHT 系列板卡;● 格式 2 适用于 16 通道规格的 SHT 系列板卡。

3.1.2.12 设置录音操作

3.1.2.12.1 设置录音任务的终止条件

3.1.2.12.1.1 DtmfStopRecCharSet

配置项	DtmfStopRecCharSet
节	[BoardId=x]
书写格式	格式 1: DtmfStopRecCharSet={s,s,s,s} 格式 2: DtmfStopRecCharSet={s,s,s,s,s,s,s,s} 格式 3: DtmfStopRecCharSet={s,s,s,s,s,s,s,s,s,s,s,s,s,s} 格式 4: DtmfStopRecCharSet=s
取值范围	s 为 DTMF 字符集, 书写形式为{***}, 其中*可以是“0123456789*#abcd”中的任何一个, 个数可根据实际需要设置。s 为空集表示关闭此项功能, 非空表示开启此项功能。缺省值为空集。
功能描述	设置录音任务是否因 DTMF 检测器 检测到 DTMF 字符而终止。

注意事项	<ul style="list-style-type: none">● 格式 1 适用于 SHT 系列的 USB 语音盒和 ATP 系列的 USB 录音盒;● 格式 2 适用于 8 通道规格的 SHT 系列、ATP 系列板卡;● 格式 3 适用于 16 通道规格的 SHT 系列、ATP 系列、DST 系列板卡;● 格式 4 适用于 SHD 系列、DTP 系列、SHN 系列板卡。
使用范例	DtmfStopRecCharSet={0123456789*#abcd } //SHD 系列: 检测到任意字符就停止录音任务

3.1.2.12.1.2 HangupStopRec

配置项	HangupStopRec
节	[BoardId=x]
书写格式	格式 1: HangupStopRec=b,b,b,b 格式 2: HangupStopRec=b,b,b,b,b,b,b 格式 3: HangupStopRec=b,b,b,b,b,b,b,b,b,b,b,b 格式 4: HangupStopRec=b
取值范围	b=0: 否 (缺省值); b=1: 是。
功能描述	设置录音任务是否因驱动程序状态机检测到对端挂机而终止。
注意事项	<ul style="list-style-type: none">● 格式 1 适用于 SHT 系列的 USB 语音盒;● 格式 2、格式 3 分别适用于 8 通道、16 通道规格的 SHT 系列;● 格式 4 适用于 SHD 和 SHN 系列板卡。

3.1.2.12.2 设置录音信号源

3.1.2.12.2.1 DefaultRecordVolume

请参见 [DefaultRecordMixerVolume](#)

3.1.2.12.2.2 DefaultRecordMixerVolume

配置项	DefaultRecordVolume DefaultRecordMixerVolume
节	[BoardId=x]

书写格式	<p>格式 1: DefaultRecordVolume=v₃,v₃,v₃,v₃ DefaultRecordMixerVolume=v₂,v₂,v₂,v₂</p> <p>格式 2: DefaultRecordVolume=v₃,v₃,v₃,v₃,v₃,v₃,v₃,v₃ DefaultRecordMixerVolume=v₂,v₂,v₂,v₂,v₂,v₂,v₂,v₂</p> <p>格式 3: DefaultRecordVolume=v₃,v₃,v₃,v₃,v₃,v₃,v₃,v₃,v₃,v₃,v₃,v₃,v₃ DefaultRecordMixerVolume=v₂,v₂,v₂,v₂,v₂,v₂,v₂,v₂,v₂,v₂,v₂,v₂,v₂</p> <p>格式 4: DefaultRecordVolume=v₃ DefaultRecordMixerVolume=v₂</p>
取值范围	v ₃ ,v ₂ : 音量调节器的音量, -7≤v ₃ ≤+6, -7≤v ₂ ≤+6, -7 表示关闭, 其余取值表示音量增益(乘以 3 即为分贝), 大于 0 表示音量提升, 小于 0 表示音量衰减。v ₃ 的缺省值为 0, v ₂ 的缺省值为 -7。
功能描述	设置进入录音混音器 M3 的信号增益。DefaultRecordVolume 设置音量调节器 A3 的增益, DefaultRecordMixerVolume 设置音量调节器 A2 的增益。更多信息参见第 1 章中相应板卡的原理框图部分。
注意事项	<ul style="list-style-type: none"> ● 格式 1 适用于 SHT 系列的 USB 语音盒、ATP 系列的 USB 录音盒; ● 格式 2 适用于 8 通道规格的 SHT 系列、ATP 系列板卡; ● 格式 3 适用于 16 通道规格的 SHT 系列、ATP 系列, DefaultRecordVolume 还适用于 DST 系列板卡; ● 格式 4 适用于 SHD 系列、DTP 系列板卡、SHN 系列板卡。
使用范例	DefaultRecordVolume=1,1,1,1,1,1,1,1,1,1,1,1,1,1,1 //提升 3DB

3.1.2.12.3 设置文件录音的截尾功能

3.1.2.12.3.1 TruncateTailOnRecordToFile

配置项	TruncateTailOnRecordToFile
节	[SystemConfig]
书写格式	TruncateTailOnRecordToFile=t
取值范围	t=0: 关闭此项功能(缺省值); t>0: 从文件尾部删除的语音数据长度, 单位为毫秒。
功能描述	设置文件录音时的截尾时间长度。有关文件录音截尾功能的详细内容请参见函数 SsmSetTruncateTail 的说明。
注意事项	要求 SynCTI Ver. 2.1 或更高。

3.1.2.12.4 设置录音操作的编码格式

3.1.2.12.4.1 DefaultRecordFormat

配置项	DefaultRecordFormat
节	[BoardId=x]
书写格式	格式 1: DefaultRecordFormat =v,v,v,v 格式 2: DefaultRecordFormat =v,v,v,v,v,v,v,v 格式 3: DefaultRecordFormat =v,v,v,v,V,V,V,V,V,V,V,V,V,V,V,V 格式 4: DefaultRecordFormat =v
取值范围	请参见 SsmRecToFile 的 nFormat 参数的 9 种格式，缺省值 v=6 具体不同的板卡支持的格式参见 1.3.2 板卡支持的 CODEC 。
功能描述	设置通道的缺省录音数据编码格式。
注意事项	<ul style="list-style-type: none">● 格式 1 适用于 SHT 系列的 USB 语音盒和 ATP 系列的 USB 录音盒；● 格式 2 适用于 8 通道规格的 SHT 系列、ATP 系列板卡；● 格式 3 适用于 16 通道规格的 SHT 系列、ATP 系列板卡、DST 系列板卡；● 格式 4 适用于 SHD 系列板卡、DTP 系列卡、SHN 系列板卡。

3.1.2.12.5 设置驱动程序内部的录音缓冲区

3.1.2.12.5.1 RecordBufSize

配置项	RecordBufSize
节	[SystemConfig]
书写格式	RecordBufSize=k
取值范围	k: 缓冲区的长度，单位为字节，必须为 512 的整数倍，最小值为 4096，缺省值为 32768 字节。当启用 GsmCodecEnable 配置项时，最小值为 16384。
功能描述	设置驱动程序内部为每一个通道分配的录音缓冲区的大小。

3.1.2.12.6 设置录音文件为 wav 格式

3.1.2.12.6.1 RecAsWavFormat

配置项	RecAsWavFormat

节	[SystemConfig]
书写格式	RecAsWavFormat =K
取值范围	K=0: 维持原状, 即文件扩展名为 wav 或 mp3 的按照 wav 格式进行录音; K=1: 录音文件按照 wav 格式进行录音。
功能描述	设置录音文件为 wav 格式。

3.1.2.12.6.2 Mp3IsOnlyRead

配置项	Mp3IsOnlyRead
节	[SystemConfig]
书写格式	Mp3IsOnlyRead =K
取值范围	K=0: 录制的 MP3 文件为普通模式, 可读可写; K=1: 录制的 MP3 文件为只读模式, 可读不可写。
功能描述	设置 MP3 录音文件的属性。

3.1.2.12.7 设置 AGC 的工作参数

3.1.2.12.7.1 AGCMAXGAIN

请参见 [AGCMINGAINDOWNLINK](#)

3.1.2.12.7.2 AGCMINGAIN

请参见 [AGCMINGAINDOWNLINK](#)

3.1.2.12.7.3 AGCMAXGAINUPLINK

请参见 [AGCMINGAINDOWNLINK](#)

3.1.2.12.7.4 AGCMINGAINUPLINK

请参见 [AGCMINGAINDOWNLINK](#)

3.1.2.12.7.5 AGCMAXGAINDOWNLINK

请参见 [AGCMINGAINDOWNLINK](#)

3.1.2.12.7.6 AGCMINGAINDOWNLINK

配置项	AGCMAXGAIN AGCMINGAIN AGCMAXGAINUPLINK AGCMINGAINUPLINK AGCMAXGAINDOWNLINK AGCMINGAINDOWNLINK
节	[BoardId=x]
书写格式	AGCMAXGAIN=G _{max} AGCMINGAIN=G _{min} AGCMAXGAINUPLINK=G _{max} AGCMINGAINUPLINK=G _{min} AGCMAXGAINDOWNLINK=G _{max} AGCMINGAINDOWNLINK=G _{min}
取值范围	G _{max} : 1≤G _{max} ≤255, 缺省值为 255。当设为最大值 255 时, AGC 的最大增益为 24dB。这个参数用于控制 AGC 对小信号的最大放大倍数。参数越大, 越能提升小信号的电平, 但是背景噪声也将被放大。 G _{min} : 1≤G _{min} ≤255, 缺省值为 6。当设为缺省值 6 时, AGC 的最小增益为-8.5dB。这个参数用于控制 AGC 对大信号的最大衰减。如果期望 AGC 控制器只有放大作用, 没有衰减作用, 则本参数应大于或等于 16。
功能描述	用于控制 DSP 增益系数的变化范围的参数。
注意事项	<ul style="list-style-type: none">● AGC 的原始增益系数为 16, 对应的增益为 0dB(既不放大也不衰减)。当输出信号电平小于预期电平时, 增益系数将上升, 反之则下降。AGC 的实际电平增益为 $20\log(\text{增益系数}/16)$ dB。本配置项为高级配置项。大多数情况下, 使用缺省值就可以满足系统要求。如果确实需要修改此配置项, 请在本公司技术人员的指导下进行。● <u>AGCMAXGAINUPLINK</u>、<u>AGCMINGAINUPLINK</u>、<u>AGCMAXGAINDOWNLINK</u> 以及 <u>AGCMINGAINDOWNLINK</u> 配置项是 DST-24B/PCI、DST-24B/PCI+、DST-24B/PCIe、DST-24B/PCIe+卡的专用 AGC 配置项, 用于对上下行 AGC 参数分别进行设置。

3.1.2.12.7.7 AGCMAXLEVEL

请参见 [AGCMINLEVELDOWNLINK](#)

3.1.2.12.7.8 AGCMINLEVEL

请参见 [AGCMINLEVELDOWNLINK](#)

3.1.2.12.7.9 AGCMAXLEVELUPLINK

请参见 [AGCMINLEVELDOWNLINK](#)

3.1.2.12.7.10 AGCMINLEVELUPLINK

请参见 [AGCMINLEVELDOWNLINK](#)

3.1.2.12.7.11 AGCMAXLEVELDOWNLINK

请参见 [AGCMINLEVELDOWNLINK](#)

3.1.2.12.7.12 AGCMINLEVELDOWNLINK

配置项	AGCMAXLEVEL AGCMINLEVEL AGCMAXLEVELUPLINK AGCMINLEVELUPLINK AGCMAXLEVELDOWNLINK AGCMINLEVELDOWNLINK
节	[BoardId=x]
书写格式	AGCMAXLEVEL=L_max AGCMINLEVEL=L_min AGCMAXLEVELUPLINKL=L_max AGCMINLEVELUPLINKL=L_min AGCMAXLEVELDOWNLINKL=L_max AGCMINLEVELDOWNLINKL=L_min
取值范围	L _{max} : AGCMINLEVEL≤L _{max} ≤25600000, 缺省值为3200000。输出电平的上限, 如果实际的输出电平大于L _{max} , AGC模块开始降低增益, 直到到达AGCMINGAIN设定的最小增益。 L _{min} : 0≤L _{min} ≤AGCMAXLEVEL, 缺省值为2560000。输出电平的下限, 如果实际输出电平小于L _{min} , 并且保持时间一定时间后, AGC模块开始增大增益, 直到到达配置项AGCMAXGAIN设定的最大增益。
功能描述	用于控制AGC输出电平期望值的参数。
注意事项	<ul style="list-style-type: none">● AGC通过调节增益, 将输出信号尽可能控制在指定范围内。如果希望输出结果的电平尽可能平稳, 则可以将两个参数设为接近的值。但最好不要设为相同的值, 否则将导致增益频繁调整, 使得在稳定的输入电平情况下, AGC输出电平反而上下波动。电平值的满幅度=25600000, 如果假定满幅度电平为0dB, 则设定电平=10log(设定值/25600000) dB, 缺省值3200000和2560000分别相当于-9dB和-10dB。本配置项为高级配置项。大多数情况下, 使用缺省值就可以满足系统要求。如果确实需要修改此配置项, 请在本公司技术人员的指导下进行。● AGCMAXLEVELUPLINK、AGCMINLEVELUPLINK、AGCMAXLEVELDOWNLINK以及AGCMINLEVELDOWNLINK配置项是DST-24B/PCI, DST-24B/PCI+, DST-24B/PCIe, DST-24B/PCIe+卡的专用AGC配置项, 用于对上行AGC参数分别进行设置。

3.1.2.12.7.13 AGCDOWNRATIO

请参见 [AGCKEEPETIME](#)

3.1.2.12.7.14 AGCUPRATIO

请参见 [AGCKEEPETIME](#)

3.1.2.12.7.15 AGCDOWNRATIOUPLINK

请参见 [AGCKEEPETIME](#)

3.1.2.12.7.16 AGCUPRATIOUPLINK

请参见 [AGCKEEPETIME](#)

3.1.2.12.7.17 AGCDOWNRATIODOWNLINK

请参见 [AGCKEEPETIME](#)

3.1.2.12.7.18 AGCUPRATIODEDOWNLINK

请参见 [AGCKEEPETIME](#)

3.1.2.12.7.19 AGCKEEPETIME

配置项	AGCDOWNRATIO AGCUPRATIO AGCDOWNRATIOUPLINK AGCUPRATIOUPLINK AGCDOWNRATIODEDOWNLINK AGCUPRATIODEDOWNLINK AGCKEEPETIME
节	[BoardId=x]
书写格式	AGCDOWNRATIO=R _{down} AGCUPRATIO=R _{up} AGCDOWNRATIOUPLINK=R _{down} AGCUPRATIOUPLINK=R _{up} AGCDOWNRATIODEDOWNLINK=R _{down} AGCUPRATIODEDOWNLINK=R _{up} AGCKEEPETIME=t

取值范围	<p>R_{down}: $0 \leq R_{down} \leq 100$, 缺省值为 35。自动衰减增益的步长, 5 表示 5%。控制增益下降速度。通常情况下应该设定较快的下降速度, 从而减少输出信号的碰顶失真。</p> <p>R_{up}: $0 \leq R_{up} \leq 100$, 缺省值为 1。自动提升增益的步长, 5 表示 5%。控制增益上升速度。通常情况下应该设定较慢的上升速度, 从而减少输出信号的背景噪声。但设得过小将使小信号得不到及时放大。</p> <p>t: $0 \leq t \leq 32768$, 缺省值为 0。保持时间, 每个单位表示 8 毫秒。增益由下降转为上升时, 可以保持一段时间再开始。保持时间=(设定值+1)*8ms。设定本参数, 有助于减少话音间隔的背景噪声, 但会延迟对小信号的提升。增益下降没有保持时间。</p>
功能描述	用于控制 AGC 增益变化速度的参数。
注意事项	<ul style="list-style-type: none"> ● DOWN_TIME 和 UP_TIME 所设定的增益变化速度 = $20\log(1/(1-\text{设定值}/100)) \text{ dB}/16\text{ms}$。缺省的下降速度为 3.7dB/16ms, 也就是大约经过 100ms 时间, 增益可以从最大值降为 0dB。缺省的上升速度为 0.087dB/16ms, 也就是大约经过 4 秒的时间, 增益可以从 0dB 到达最大值。本配置项为高级配置项。大多数情况下, 使用缺省值就可以满足系统要求。如果确实需要修改此配置项, 请在本公司技术人员的指导下进行。 ● AGCDOWNRATIOUPLINK、AGCUPRATIOUPLINK、AGCDOWNRATIODOWNLINK 以及 AGCUPRATIODOWNLINK 配置项是 DST-24B/PCI、DST-24B/PCI+、DST-24B/PCIe、DST-24B/PCIe+卡的专用 AGC 配置项, 用于对上下行 AGC 参数分别进行设置。

3.1.2.12.7.20 OpenRecEnAndPlayEnOnIdle

配置项	OpenRecEnAndPlayEnOnIdle
节	[BoardId=x]
书写格式	格式 1: OpenRecEnAndPlayEnOnIdle=b,b,b,b 格式 2: OpenRecEnAndPlayEnOnIdle=b,b,b,b,b,b,b,b 格式 3: OpenRecEnAndPlayEnOnIdle=b,b,b,b,b,b,b,b,b,b,b,b,b
取值范围	b=0: 否 (缺省值); b=1: 是。
功能描述	设置模拟中继线通道在空闲状态下进行录音操作时是否开启 AGC 功能。
注意事项	<ul style="list-style-type: none"> ● 本配置项只适用于 SHT 系列板卡; ● 本配置项要求 SynCTI 驱动程序的版本为 4.7.2.0 或更高。

3.1.2.12.7.21 AutoRecAgcSwitch

配置项	AutoRecAgcSwitch
节	[BoardId=x]
书写格式	AutoRecAgcSwitch=a

取值范围	<p>a=0: 关闭 (缺省值); a>0: 对于 DST 系列 B 型数字电话录音卡, a 的 Bit0、Bit1、Bit2、Bit3 以及 Bit4 有效: Bit0=0, 下行 AGC 关闭; Bit0=1, 下行 AGC 打开; Bit1=0, 上行 AGC 关闭; Bit1=1, 上行 AGC 打开; Bit2=0 及 Bit3=0, 直流分离关闭; Bit2=1 及 Bit3=1, 直流分离打开; Bit4=1, 打开立体声开关; Bit4=0, 关闭立体声开关。 对于其他型号板卡, a 的 Bit0 有效: Bit0=0, AGC 关闭; Bit0=1, AGC 打开</p>
功能描述	设置通道在进行录音操作时是否自动开启 AGC 功能。
注意事项	<ul style="list-style-type: none"> ● 本配置项要求 SynCTI 驱动程序的版本为 5.3.1.1 或更高; ● SHT 板卡如果需要一直开启 AGC, 除了对本配置项进行设置外, 还需要修改配置项 OpenRecEnAndPlayEnOnIdle 的值为 1。

3.1.2.12.8 设置录音操作的其它参数

3.1.2.12.8.1 RecEliDTDurTalking

配置项	RecEliDTDurTalking
节	[BoardId=x]
书写格式	RecEliDTDurTalking =b,b,b,b,b,b,b,b,b,b,b,b,b,b
取值范围	b=0: 否 (缺省值); b=1: 是。
功能描述	是否在录音时消除通话过程中的 DTMF 码。
注意事项	本配置项只适用于 SHT-16C/PCI/EC 卡。

3.1.2.12.9 设置 MP3 格式录音的比特率

3.1.2.12.9.1 Mp3EncodingBitRate

配置项	Mp3EncodingBitRate
节	[SystemConfig]
书写格式	Mp3EncodingBitRate=b
取值范围	b=1: MP3 录音采用 8kbps 的比特率 (缺省值); b=2: MP3 录音采用 16kbps 的比特率; b=3: MP3 录音采用 32kbps 的比特率。
功能描述	设置 MP3 录音采用的比特率。

3.1.2.13 设置录音/放音的共用参数

3.1.2.13.1 设置乒乓缓冲区的最小值

3.1.2.13.1.1 RecordAndPlayUseAsIP

配置项	RecordAndPlayUseAsIP
节	[SystemConfig]
书写格式	RecordAndPlayUseAsIP=b
取值范围	b=0: 否 (缺省值); b=1: 是。
功能描述	设置录音、放音操作是否支持微型缓冲区。更多信息请参见函数 SsmPlayMemBlock 、 SsmRecordMemBlock 的说明。

3.1.2.13.2 设置乒乓缓存放音的模式

3.1.2.13.2.1 BlockPlayType

配置项	BlockPlayType
节	[SystemConfig]
书写格式	BlockPlayType=b
取值范围	b=0: 饥饿模式 (缺省值); b=1: 非饥饿模式。
功能描述	设置乒乓缓存放音操作是否采用饥饿模式。饥饿模式：在应用程序提交的数据将驱动内部缓存区填满之前，驱动将根据计时器不断调用回调函数。非饥饿模式：在应用程序提交的数据放完之后，驱动才根据计时器调用回调函数。

3.1.2.13.3 设置软件编码的 GSM 格式

3.1.2.13.3.1 GsmCodecEnable

配置项	GsmCodecEnable
节	[SystemConfig]
书写格式	GsmCodecEnable=b

取值范围	b=0: 否(缺省值); b=1: 是。
功能描述	驱动程序是否自动加载 macmcvt.dll 模块。macmcvt.dll 是 SynCTI 驱动程序提供的音频编码/解码器模块(ACM)，可以实现 GSM、MP3 的软件编码/解码，以及 G.729A 的软件解码。如果应用程序希望在 Synway 板卡上实现 GSM、MP3 或 G.729A 格式的录音或放音，而板卡上不具备相应的硬件编码器/解码器，可以通过设置本配置项，将外挂的 ACM 程序加载到 SynCTI 驱动程序中，从而使 SynCTI 驱动程序具有了软件方式的语音编码和解码能力。
注意事项	<ul style="list-style-type: none"> ● 如果 b 设置为 1，必须保证 Windows 系统中已经安装了相应的音频编解码器(ACM)驱动程序； ● 本配置项只适用于 Windows 操作系统。对于 Linux 操作系统，只支持 GSM、MP3 和 G.729A 格式的硬件编码，关于硬件编码设置详见配置项 DspCoder； ● 当配置项 BusPlayListen=0 时，GsmCodecEnable=0 设置无效，驱动内部仍将加载软件录放音引擎，以支持压缩格式监听。

3.1.2.14 设置 DTMF 检测器

3.1.2.14.1 启动/关闭 DTMF 检测器

3.1.2.14.1.1 AlwaysEnableRxDtmf

配置项	AlwaysEnableRxDtmf
节	[BoardId=x]
书写格式	AlwaysEnableRxDtmf=m
取值范围	<p>m=0: DTMF 检测器的工作状态由驱动程序根据通道状态的迁移情况自动控制，当通道进入通话状态时自动开启，离开通话状态时自动关闭，但应用程序也可以通过 SsmEnableRxDtmf 函数进行控制；</p> <p>m=1: DTMF 检测器一直处于开启状态。</p>
功能描述	设置 DTMF 检测器 的工作方式。
注意事项	<ul style="list-style-type: none"> ● 本配置项支持 SHD、DTP 系列板卡； ● m=0 时，振铃状态下，ISDN 可以收到 DTMF，ISUP 和 TUP 不能收到 DTMF。

3.1.2.14.1.2 RcvDtmfOnIdle

配置项	RcvDtmfOnIdle
节	[BoardId=x]
书写格式	<p>格式 1: RcvDtmfOnIdle=b,b,b,b</p> <p>格式 2: RcvDtmfOnIdle=b,b,b,b,b,b,b</p> <p>格式 3: RcvDtmfOnIdle=b,b,b,b,b,b,b,b,b,b,b</p> <p>格式 4: RcvDtmfOnIdle=[0,29]:b</p>

取值范围	b=0: 关闭 (缺省值); b=1: 开启。
功能描述	当坐席通道处于空闲状态时, 设置 DTMF 检测器 的工作状态。
注意事项	<ul style="list-style-type: none">● 本配置项仅支持坐席通道;● 格式 1 适用于采用 USB 总线方式的 SHT 系列语音盒;● 格式 2 适用于 8 通道规格的 SHT 系列;● 格式 3 适用于 16 通道规格的 SHT 系列板卡;● 格式 4 适用于 30 通道规格的远端坐席接入模块。

3.1.2.14.2 设置 DTMF 检测器的缓冲区大小

3.1.2.14.2.1 RxDtmfBufSize

配置项	RxDtmfBufSize
节	[SystemConfig]
书写格式	RxDtmfBufSize=k
取值范围	k: 缓冲区的大小, 单位为字节, 缺省值为 200。
功能描述	设置 DTMF 检测器 的缓冲区大小。

3.1.2.14.3 设置 DTMF 滤波器的参数

3.1.2.14.3.1 DualAndAllFreqEnScale

配置项	DualAndAllFreqEnScale
节	[BoardId=x]
书写格式	格式 1: DualAndAllFreqEnScale=n,n,n,n 格式 2: DualAndAllFreqEnScale=n,n,n,n,n,n,n,n 格式 3: DualAndAllFreqEnScale=n,n,n,n,n,n,n,n,n,n,n 格式 4: DualAndAllFreqEnScale=m
取值范围	0≤n≤58, 单位为%, 缺省值为 32。 0≤m≤58, 单位为%, 默认为 32。
功能描述	设置 DTMF 信号的带内能量占全频能量的百分比判定门限。有关此配置项的详细说明请参见第 1 章中 DTMF 滤波器 部分。

注意事项	<ul style="list-style-type: none"> ● 格式 1 适用于采用 USB 总线方式的 SHT 系列语音盒、ATP 系列录音盒； ● 格式 2 适用于 8 通道规格的 SHT 系列、ATP 系列板卡； ● 格式 3 适用于 16 通道规格的 SHT 系列、ATP、DST 系列板卡； ● 格式 4 适用于 SHD 系列、DTP 系列、SHN 系列板卡，但下列型号的板卡要求 SynCTI Ver. 4.5.1.7 或更高： <ul style="list-style-type: none"> ◆ SHD-240A-CT/cPCI ◆ SHD-240S-CT/cPCI ◆ SHD-480A-CT/cPCI ◆ SHD-480S-CT/cPCI ● 函数 SsmSetFlag（携带 F_DualAndAllFreqEnScale 参数）可以实现相同功能； ● 8B、16B、8C、16C 系列板卡接收 abcd 时，该配置项设为 15。
------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3.1.2.14.3.2 HighAndLowFreqEnScale

配置项	HighAndLowFreqEnScale
节	[BoardId=x]
书写格式	格式 1: HighAndLowFreqEnScale=n,n,n,n 格式 2: HighAndLowFreqEnScale=n,n,n,n,n,n,n,n 格式 3: HighAndLowFreqEnScale=n,n,n,n,n,n,n,n,n,n,n,n,n 格式 4: HighAndLowFreqEnScale=m
取值范围	0≤n≤100，单位为%，缺省值为 10。 0≤m≤100，单位为%，缺省值为 20。
功能描述	设置 DTMF 信号中，高频分量部分能量与低频分量部分能量之间的比例。只有当 DTMF 信号的高频分量部分能量与低频分量部分能量之间的比例大于本配置项的设定值时，驱动程序才认为是真正的 DTMF 信号。
注意事项	<ul style="list-style-type: none"> ● 格式 1 适用于采用 USB 总线方式的 SHT 系列语音盒、ATP 系列录音盒； ● 格式 2 适用于 8 通道规格的 SHT 系列、ATP 系列板卡； ● 格式 3 适用于 16 通道规格的 SHT 系列、ATP 系列、DST 系列板卡； ● 格式 4 适用于 SHD 系列、DTP 系列板卡、SHN 系列板卡； ● 函数 SsmSetFlag（携带 F_HighAndLowFreqEnScale 参数）可以实现相同功能。

3.1.2.14.3.3 DtmfEnergy

配置项	DtmfEnergy
节	[BoardId=x]

书写格式	格式 1: DtmfEnergy=n,n,n,n 格式 2: DtmfEnergy=n,n,n,n,n,n,n 格式 3: DtmfEnergy=n,n,n,n,n,n,n,n,n,n,n,n,n 格式 4: DtmfEnergy=m
取值范围	n≥0, 缺省值为 0。设置方法请参考配置项 MinimumVoiceDetermineEnergy 的说明。
功能描述	设置 DTMF 信号的带内能量的绝对值。只有当 DTMF 信号的带内能量大于本配置项的设定值时，驱动程序才认为是真正的 DTMF 信号。
注意事项	<ul style="list-style-type: none"> ● 格式 1 适用于采用 USB 总线方式的 SHT 系列语音盒、ATP 系列录音盒； ● 格式 2 适用于 8 通道规格的 SHT 系列、ATP 系列板卡； ● 格式 3 适用于 16 通道规格的 SHT 系列、ATP 系列、DST 系列板卡； ● 格式 4 适用于 SHD 系列、DTP 系列板卡、SHN 系列板卡。

3.1.2.14.3.4 DTMFFreqOffset

配置项	DTMFFreqOffset
节	[BoardId=x]
书写格式	格式 1: DTMFFreqOffset=n,n,n,n 格式 2: DTMFFreqOffset=n,n,n,n,n,n,n,n 格式 3: DTMFFreqOffset=n,n,n,n,n,n,n,n,n,n,n,n,n,n 格式 4: DTMFFreqOffset=m
取值范围	SHT 系列: 0≤n≤100, 单位为%, 缺省值为 16; ATP 系列: 0≤m≤30, 单位为%, 缺省值为 15; SHD 系列: 0≤m≤100, 单位为%, 缺省值为 18。
功能描述	设置 DTMF 信号的最大偏移误差。(偏移误差= (检测到的频率-标准频率) *1024/检测到的频率)
注意事项	<ul style="list-style-type: none"> ● 格式 1 适用于采用 USB 总线方式的 SHT 系列语音盒； ● 格式 2 适用于 8 通道规格的 SHT 系列板卡； ● 格式 3 适用于 16 通道规格的 SHT 系列板卡； ● 格式 4 适用于 ATP 系列板卡, SHD 系列卡。

3.1.2.14.3.5 DtmfModel

配置项	DtmfModel
节	[BoardId=x]

书写格式	格式 1: DtmfModel=n,n,n,n 格式 2: DtmfModel=n,n,n,n,n,n,n,n 格式 3: DtmfModel=n,n,n,n,n,n,n,n,n,n,n,n,n,n,n
取值范围	n=0: 否 (缺省值); n=1: 是。
功能描述	设置 DTMF 检测时是否仅使用 697HZ~1633HZ 范围内的能量作为参考全频能量。
注意事项	<ul style="list-style-type: none"> ● 格式 1 适用于采用 USB 总线方式的 SHT 系列语音盒; ● 格式 2 适用于 8 通道规格的 SHT 系列板卡; ● 格式 3 适用于 16 通道规格的 SHT 系列板卡;

3.1.2.14.4 设置 DTMF 脉宽滤波器的参数

3.1.2.14.4.1 ReceiveDtmfSensitive

配置项	ReceiveDtmfSensitive
节	[BoardId=x]
书写格式	ReceiveDtmfSensitive=n
取值范围	<p>n 包含 8 个比特 (记为 Bit7...Bit0), 各比特含义如下:</p> <p>Bit3~Bit0: 高电平最小保持时间, 取值范围为 1~6, 单位为 16 毫秒, 缺省值为 3;</p> <p>Bit7~Bit4: 低电平最小保持时间, 取值范围为 0~6, 单位为 16 毫秒, 缺省值为 0。</p> <p>注意: 如果 Bit7~Bit4 取值为 0, 那么低电平最小保持时间由高电平最小保持时间决定。如果高电平最小保持时间小于 3, 那么低电平最小保持时间取值为 1; 否则, 低电平最小保持时间取值为 2。</p> <p>n 采用 16 进制表示。例如: ReceiveDtmfSensitive=0x43, 表示高电平、低电平的最小保持时间分别为 48、64 毫秒。ReceiveDtmfSensitive=0x03, 表示高电平、低电平的最小保持时间分别为 48、32 毫秒。</p>
功能描述	设置 DTMF 信号的高电平、低电平的最小保持时间。更多信息请参见第 1 章中 DTMF 脉宽滤波器 部分。
注意事项	<p>适用于 SHD 系列、SHT 系列板卡、ATP 系列板卡、DTP 系列板卡, 但 SHD 系列的下列型号的板卡要求 SynCTI Ver. 4.5.1.7 或更高:</p> <ul style="list-style-type: none"> ◆ SHD-240A-CT/cPCI ◆ SHD-240S-CT/cPCI ◆ SHD-480A-CT/cPCI ◆ SHD-480S-CT/cPCI

3.1.2.14.4.2 OmitABCD

配置项	OmitABCD
节	[BoardId=x]
书写格式	格式 1: OmitABCD=b,b,b,b 格式 2: OmitABCD=b,b,b,b,b,b,b,b 格式 3: OmitABCD=b,b,b,b,b,b,b,b,b,b,b,b 格式 4: OmitABCD=b
取值范围	b=0: 丢弃, 缺省值为 0; b=1: 不丢弃。
功能描述	设置是否丢弃 DTMF 检测器收到的'a'、'b'、'c'、'd'字符。有关此配置项的详细说明请参见第 1 章中 DTMF 脉宽滤波器 部分。
注意事项	<ul style="list-style-type: none"> ● 格式 1 适用于采用 USB 总线方式的 SHT 系列语音盒、ATP 系列录音盒; ● 格式 2 适用于 8 通道规格的 SHT、ATP 系列板卡; ● 格式 3 适用于 16 通道规格的 SHT、ATP、DST 系列板卡; ● 格式 4 适用于 SHD 系列、DTP 系列、SHN 系列板卡; ● 开启接收 abcd 码时, 需要调整 DualAndAllFreqEnScale 值, 开启时驱动默认调整为 15, 若仍接收不到, 请继续降低该值进行测试; ● 对于 SHN 系列板卡, 本配置项仅对带内方式下的 DTMF 检查起作用, 对 RFC2833 和信令方式下的 DTMF 检查无效。

3.1.2.14.4.3 DtrmOnDtmfHighLevel

配置项	DtrmOnDtmfHighLevel
节	[BoardId=x]
书写格式	格式 1: DtrmOnDtmfHighLevel=n,n,n,n 格式 2: DtrmOnDtmfHighLevel=n,n,n,n,n,n,n,n 格式 3: DtrmOnDtmfHighLevel=n,n,n,n,n,n,n,n,n,n,n,n,n 格式 4: DtrmOnDtmfHighLevel=n
取值范围	n=0: DTMF 信号的低电平保持时间超过配置项 ReceiveDtmfSensitive 指定的最小保持时间后才输出; n=1: DTMF 信号的高电平保持时间超过配置项 ReceiveDtmfSensitive 指定的最小保持时间后立即输出。 缺省值为 0。

功能描述	设置 DTMF 检测器 输出 DTMF 字符的时机。有关此配置项的详细说明请参见第 1 章中 DTMF 脉宽滤波器 部分。
注意事项	<ul style="list-style-type: none">● 格式 1 适用于采用 USB 总线方式的 SHT 系列语音盒、ATP 系列录音盒；● 格式 2 适用于 8 通道规格的 SHT 系列、ATP 系列板卡；● 格式 3 适用于 16 通道规格的 SHT 系列、ATP 系列、DST 系列板卡；● 格式 4 适用于 SHD 系列、DTP 系列板卡、SHN 系列板卡。

3.1.2.15 设置 DTMF 发生器(CTI 系列)

3.1.2.15.1 设置发送缓冲区的大小

3.1.2.15.1.1 TxDtmfBufSize

配置项	TxDtmfBufSize
节	[SystemConfig]
书写格式	TxDtmfBufSize=n
取值范围	$n \geq 1$, 缺省值为 50 个字符。
功能描述	设置 DTMF 发生器的发送缓冲区大小。

3.1.2.15.2 设置 DTMF 发生器的参数

3.1.2.15.2.1 TxDtmfTimePara

注意事项	<ul style="list-style-type: none">● 格式 1 适用于 SHT 系列的 USB 语音盒；● 格式 2、格式 3 分别适用于 8 通道、16 通道规格的 SHT 系列；● 格式 4 适用于 SHD 系列、SHN 系列板卡。
------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------

3.1.2.15.2.2 DefaultTxDelayTime

配置项	DefaultTxDelayTime
节	[SystemConfig]
书写格式	DefaultTxDelayTime=n
取值范围	n: 保持静默的时间，单位为毫秒，缺省值为 2000。
功能描述	设置发送 DTMF 时延时字符的延时时间。有关此参数的含义请参见 SsmTxDtmf 函数的说明。

3.1.2.15.2.3 TxDtmfAmp

配置项	TxDtmfAmp
节	[BoardId=x]
书写格式	TxDtmfAmp=0xxxx
取值范围	数字卡默认 0x3f3f，模拟卡默认 0x8976。
功能描述	设置发送 DTMF 信号的幅度，高 8 位设置低频率的幅度，低 8 位设置高频率的幅度。功能同函数 SsmSetFlag （携带 F_TXDTMFAMP 参数）设置。

3.1.2.16 设置回波抵消器(CTI 系列)

3.1.2.16.1 设置工作状态

3.1.2.16.1.1 EnableEchoCancellor

配置项	EnableEchoCancellor
节	[BoardId=x]
书写格式	格式 1: EnableEchoCancellor=b,b,b,b 格式 2: EnableEchoCancellor=b,b,b,b,b,b,b,b 格式 3: EnableEchoCancellor=b,b,b,b,b,b,b,b,b,b,b 格式 4: EnableEchoCancellor=b

取值范围	b=0: 关闭; b=1: 开启（缺省值）。
功能描述	设置回波抵消器的工作状态。详细内容请参见第1章中 回波抵消器 部分。
注意事项	<ul style="list-style-type: none">● 格式1适用于采用USB总线方式的SHT系列语音盒；● 格式2适用于8通道规格的SHT系列板卡；● 格式3适用于16通道规格的SHT系列；● 格式4适用于SHD系列板卡、SHN系列板卡。

3.1.2.16.2 设置工作参数

3.1.2.16.2.1 EchoCancelDelaySize

配置项	EchoCancelDelaySize
节	[BoardId=x]
书写格式	EchoCancelDelaySize=k
取值范围	k: 滤波延迟系数，取值范围及缺省值取决于配置项 LoadFskBin 的设置： LoadFskBin=0: 0≤k≤7, 缺省值: 6; LoadFskBin>0: 0≤k≤31, 缺省值: 12。
功能描述	设置回波抵消器的滤波延迟系数。
注意事项	只支持SHD系列板卡和SHT系列的USB语音盒。

3.1.2.17 设置 Barge-in 检测器 (CTI 系列)

3.1.2.17.1 设置 Barge in 检测器的启动方式

3.1.2.17.1.1 AlwaysDetectBargeln

配置项	AlwaysDetectBargeln
节	[BoardId=x]
书写格式	AlwaysDetectBargeln=b

取值范围	b=0: 由驱动程序自动控制。驱动程序的调度机制为：当通道状态迁移到 S_CALL_TALKING，或者以动态发言的身份加入会议室时，自动开启 BargelIn 检测器；当通道状态迁出 S_CALL_TALKING，或者具有动态发言身份的成员退出会议室时，自动停止 BargelIn 检测器； b=1: 总是开启。 缺省值为 0。
功能描述	设置 Barge in 检测器 的启动方式。
注意事项	本配置项适用于 SHD、SHN、DTP、SHV 系列板卡。对于 DTP、SHV 系列板卡，该配置项设置成 0 表示完全关闭 Barge in 检测器，驱动不会自动控制检测器开关。更多信息请参见第 1 章中“ Barge in 检测器 ”部分内容。

3.1.2.17.2 设置 Barge in 检测器的工作参数

3.1.2.17.2.1 VoiceEnergyMinValue

配置项	VoiceEnergyMinValue
节	[BoardId=x]
书写格式	VoiceEnergyMinValue=k
取值范围	k: 噪声判定门限，缺省值：100,000。
功能描述	设置 Barge in 检测器的噪声判定门限。

3.1.2.17.2.2 BargelNSensitive

配置项	BargelNSensitive
节	[BoardId=x]
书写格式	格式 1: BargelNSensitive=k,k,k,k 格式 2: BargelNSensitive=k,k,k,k,k,k,k,k 格式 3: BargelNSensitive=k,k,k,k,k,k,k,k,k,k,k,k,k,k,k 格式 4: BargelNSensitive=k
取值范围	0≤k≤31，值越大越灵敏，缺省值：6。
功能描述	设置 Barge In 检测器的灵敏度。

注意事项	<ul style="list-style-type: none"> ● 格式 1 适用于采用 USB 总线方式的 SHT 系列语音盒、ATP 系列录音盒； ● 格式 2 适用于 8 通道规格的 SHT、ATP 系列板卡； ● 格式 3 适用于 16 通道规格的 SHT、DST 系列板卡； ● 格式 4 适用于 SHV、SHD、SHN 和 DTP 系列板卡； ● 详细内容请参见函数 SsmSetBargeInSens 的说明。 																																																																																																																														
对应的 DB 值	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>BargeInSensitive 设置值</th> <th>0</th> <th>1</th> <th>2</th> <th>3</th> <th>4</th> <th>5</th> </tr> </thead> <tbody> <tr> <td>ATP-24A/PCI+ 对应 DB 值</td> <td>-</td> <td>-</td> <td>-</td> <td>-2.7</td> <td>-5.9</td> <td>-8.9</td> </tr> <tr> <td>ATP-24A/PCIe 对应 DB 值</td> <td>-</td> <td>-</td> <td>-</td> <td>-2.9</td> <td>-5.9</td> <td>-8.9</td> </tr> <tr> <th>BargeInSensitive 设置值</th> <th>6</th> <th>7</th> <th>8</th> <th>9</th> <th>10</th> <th>11</th> </tr> <tr> <td>ATP-24A/PCI+ 对应 DB 值</td> <td>-11.9</td> <td>-14.9</td> <td>-18</td> <td>-21</td> <td>-24</td> <td>-27</td> </tr> <tr> <td>ATP-24A/PCIe 对应 DB 值</td> <td>-12</td> <td>-15</td> <td>-18</td> <td>-21.1</td> <td>-24.1</td> <td>-27</td> </tr> <tr> <th>BargeInSensitive 设置值</th> <th>12</th> <th>13</th> <th>14</th> <th>15</th> <th>16</th> <th>17</th> </tr> <tr> <td>ATP-24A/PCI+ 对应 DB 值</td> <td>-29.9</td> <td>-32.9</td> <td>-35.2</td> <td>-37.1</td> <td>-41.4</td> <td>-43.7</td> </tr> <tr> <td>ATP-24A/PCIe 对应 DB 值</td> <td>-30</td> <td>-32.9</td> <td>-35.3</td> <td>-37.2</td> <td>-41.5</td> <td>-43.8</td> </tr> <tr> <th>BargeInSensitive 设置值</th> <th>18</th> <th>19</th> <th>20</th> <th>21</th> <th>22</th> <th>23</th> </tr> <tr> <td>ATP-24A/PCI+ 对应 DB 值</td> <td>-45.5</td> <td>-46.6</td> <td>-47.3</td> <td>-48</td> <td>-48</td> <td>-48</td> </tr> <tr> <td>ATP-24A/PCIe 对应 DB 值</td> <td>-45.6</td> <td>-46.8</td> <td>-47.6</td> <td>-48</td> <td>-48</td> <td>-48.2</td> </tr> <tr> <th>BargeInSensitive 设置值</th> <th>24</th> <th>25</th> <th>26</th> <th>27</th> <th>28</th> <th>29</th> </tr> <tr> <td>ATP-24A/PCI+ 对应 DB 值</td> <td>-48.1</td> <td>-48.1</td> <td>-48.1</td> <td>-48.1</td> <td>-48.1</td> <td>-48.1</td> </tr> <tr> <td>ATP-24A/PCIe 对应 DB 值</td> <td>-48.2</td> <td>-48.2</td> <td>-48.2</td> <td>-48.3</td> <td>-48.3</td> <td>-48.3</td> </tr> <tr> <th>BargeInSensitive 设置值</th> <th>30</th> <th>31</th> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>ATP-24A/PCI+ 对应 DB 值</td> <td>-48.1</td> <td>-48.1</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>ATP-24A/PCIe 对应 DB 值</td> <td>-48.3</td> <td>-48.3</td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table> <p>注：此表中只列出了 ATP-24A/PCI+ 和 ATP-24A/PCIe 的对应 DB 值。</p>	BargeInSensitive 设置值	0	1	2	3	4	5	ATP-24A/PCI+ 对应 DB 值	-	-	-	-2.7	-5.9	-8.9	ATP-24A/PCIe 对应 DB 值	-	-	-	-2.9	-5.9	-8.9	BargeInSensitive 设置值	6	7	8	9	10	11	ATP-24A/PCI+ 对应 DB 值	-11.9	-14.9	-18	-21	-24	-27	ATP-24A/PCIe 对应 DB 值	-12	-15	-18	-21.1	-24.1	-27	BargeInSensitive 设置值	12	13	14	15	16	17	ATP-24A/PCI+ 对应 DB 值	-29.9	-32.9	-35.2	-37.1	-41.4	-43.7	ATP-24A/PCIe 对应 DB 值	-30	-32.9	-35.3	-37.2	-41.5	-43.8	BargeInSensitive 设置值	18	19	20	21	22	23	ATP-24A/PCI+ 对应 DB 值	-45.5	-46.6	-47.3	-48	-48	-48	ATP-24A/PCIe 对应 DB 值	-45.6	-46.8	-47.6	-48	-48	-48.2	BargeInSensitive 设置值	24	25	26	27	28	29	ATP-24A/PCI+ 对应 DB 值	-48.1	-48.1	-48.1	-48.1	-48.1	-48.1	ATP-24A/PCIe 对应 DB 值	-48.2	-48.2	-48.2	-48.3	-48.3	-48.3	BargeInSensitive 设置值	30	31					ATP-24A/PCI+ 对应 DB 值	-48.1	-48.1					ATP-24A/PCIe 对应 DB 值	-48.3	-48.3				
BargeInSensitive 设置值	0	1	2	3	4	5																																																																																																																									
ATP-24A/PCI+ 对应 DB 值	-	-	-	-2.7	-5.9	-8.9																																																																																																																									
ATP-24A/PCIe 对应 DB 值	-	-	-	-2.9	-5.9	-8.9																																																																																																																									
BargeInSensitive 设置值	6	7	8	9	10	11																																																																																																																									
ATP-24A/PCI+ 对应 DB 值	-11.9	-14.9	-18	-21	-24	-27																																																																																																																									
ATP-24A/PCIe 对应 DB 值	-12	-15	-18	-21.1	-24.1	-27																																																																																																																									
BargeInSensitive 设置值	12	13	14	15	16	17																																																																																																																									
ATP-24A/PCI+ 对应 DB 值	-29.9	-32.9	-35.2	-37.1	-41.4	-43.7																																																																																																																									
ATP-24A/PCIe 对应 DB 值	-30	-32.9	-35.3	-37.2	-41.5	-43.8																																																																																																																									
BargeInSensitive 设置值	18	19	20	21	22	23																																																																																																																									
ATP-24A/PCI+ 对应 DB 值	-45.5	-46.6	-47.3	-48	-48	-48																																																																																																																									
ATP-24A/PCIe 对应 DB 值	-45.6	-46.8	-47.6	-48	-48	-48.2																																																																																																																									
BargeInSensitive 设置值	24	25	26	27	28	29																																																																																																																									
ATP-24A/PCI+ 对应 DB 值	-48.1	-48.1	-48.1	-48.1	-48.1	-48.1																																																																																																																									
ATP-24A/PCIe 对应 DB 值	-48.2	-48.2	-48.2	-48.3	-48.3	-48.3																																																																																																																									
BargeInSensitive 设置值	30	31																																																																																																																													
ATP-24A/PCI+ 对应 DB 值	-48.1	-48.1																																																																																																																													
ATP-24A/PCIe 对应 DB 值	-48.3	-48.3																																																																																																																													

3.1.2.17.2.3 BargeInDtrmTime

配置项	BargeInDtrmTime
节	[BoardId=x]
书写格式	格式 1: BargeInDtrmTime=k,k,k,k 格式 2: BargeInDtrmTime=k,k,k,k,k,k,k 格式 3: BargeInDtrmTime=k,k,k,k,k,k,k,k,k,k,k,k,k 格式 4: BargeInDtrmTime=k
取值范围	$k \geq 16$, 必须是 16ms 的整数倍, 单位: 毫秒, 缺省值: 32。
功能描述	设置 Barge In 的最小保持时间。

注意事项	<ul style="list-style-type: none">● 格式 1 适用于采用 USB 总线方式的 SHT 系列语音盒、ATP 系列录音盒；● 格式 2 适用于 8 通道规格的 SHT 系列板卡；● 格式 3 适用于 16 通道规格的 SHT、DST 系列板卡；● 格式 4 适用于 SHD、DTP 系列板卡、SHN 系列板卡；● 详细内容请参见函数 SsmSetIsBargeInDtrmTime 的说明
------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3.1.2.17.2.4 IsNoSoundDtrmTime

配置项	IsNoSoundDtrmTime
节	[SystemConfig]
书写格式	IsNoSoundDtrmTime=k
取值范围	最小保持时间，单位：毫秒，取值范围： ≥ 16 ，必须为 16 毫秒的整数倍，缺省值为 5000 毫秒。
功能描述	设置静默的最小保持时间。更多信息请参见函数 SsmSetNoSoundDtrmTime 的说明。
使用范例	IsNoSoundDtrmTime= 5000

3.1.2.17.2.5 DefaultDtmfIsSound

配置项	DefaultDtmfIsSound
节	[BoardId=x]
书写格式	<p>格式 1: DefaultDtmfIsSound=b,b,b,b 格式 2: DefaultDtmfIsSound=b,b,b,b,b,b,b 格式 3: DefaultDtmfIsSound=b,b,b,b,b,b,b,b,b,b,b,b 格式 4: DefaultDtmfIsSound=b</p>
取值范围	<p>b=0: 否； b=1: 是（缺省值）。</p>
功能描述	设置是否将来话信号中的 DTMF 信号视为语音信号。
注意事项	<ul style="list-style-type: none">● 格式 1 适用于采用 USB 总线方式的 SHT 系列语音盒、ATP 系列录音盒；● 格式 2 适用于 8 通道规格的 SHT、ATP 系列板卡；● 格式 3 适用于 16 通道规格的 SHT、DST 系列板卡；● 格式 4 适用于 SHD、DTP 系列板卡、SHN 系列板卡；● 本配置项要求 SynCTI Ver. 4.5.2.9 及以后版本。

3.1.2.18 设置 TDM 总线连接

3.1.2.18.1 InVoiceToBus

配置项	InVoiceToBus
节	[BoardId=x]
书写格式	格式 1: InVoiceToBus=b,b,b,b 格式 2: InVoiceToBus=b,b,b,b,b,b,b,b 格式 3: InVoiceToBus=b,b,b,b,b,b,b,b,b,b,b,b 格式 4: InVoiceToBus=b
取值范围	b=0: 否; b=1: 是 (缺省值)。
功能描述	系统初始化时, 是否将来话信号连接到 TDM 交换总线。函数 SsmSetFlag (携带 F_InVoiceToBus 参数) 可以实现相同功能。
注意事项	<ul style="list-style-type: none">● 格式 1 适用于采用 USB 总线方式的 SHT 系列语音盒、ATP 系列录音盒;● 格式 2 适用于 8 通道规格的 SHT、ATP 系列板卡;● 格式 3 适用于 16 通道规格的 SHT、DST 系列板卡;● 格式 4 适用于 SHD、DTP、SHN 系列板卡以及 24 通道规格的 DST 系列板卡;● 本配置项要求 SynCTI Ver. 4.0 及以后版本。

3.1.2.18.2 LinkFromStopPlayAndTone

配置项	LinkFromStopPlayAndTone
节	[BoardId=x]
书写格式	LinkFromStopPlayAndTone=b
取值范围	b=1: 立即终止放音任务或信号音发生器任务 (缺省值); b=0: 等待放音任务终止后, 再执行下总线操作命令。
功能描述	详细内容请参见第 1 章中 “ SHD 系列板卡的原理框图 ” 部分内容。
注意事项	<ul style="list-style-type: none">● 本配置项只适用于 30 通道、60 通道和 120 通道规格的 SHD 系列板卡;● 本配置项要求 SynCTI Ver. 4.5.2.9 及以后版本;● 当配置项 UsageMode 设置为 1 时, 本配置项无效并使用缺省值。

3.1.2.18.3 DefaultSpeakVolume

配置项	DefaultSpeakVolume
节	[BoardId=x]
书写格式	格式 1: DefaultSpeakVolume=k,k,k,k 格式 2: DefaultSpeakVolume=k,k,k,k,k,k,k,k 格式 3: DefaultSpeakVolume=k,k,k,k,k,k,k,k,k,k,k,k,k,k,k,k 格式 4: DefaultSpeakVolume=k
取值范围	-7≤k≤6, 乘以 3 即为分贝, 大于 0 表示音量提升, 小于 0 表示音量衰减, 缺省值为 0。
功能描述	设置本通道语音信号经过总线下到另一个通道的下总线混音器 M2 前的音量, 详细内容请参见第 1 章中板卡的工作原理框图部分。
注意事项	格式 1 适用于 SHT 系列的 USB 语音盒、ATP 系列的 USB 录音盒; 格式 2 适用于 8 通道规格的 SHT 系列、ATP 系列板卡; 格式 3 适用于 16 通道规格的 SHT 系列、ATP 系列、DST 系列板卡; 格式 4 适用于 SHD 系列、SHN 系列板卡。

3.1.2.18.4 EnableCommonTimeSlot

配置项	EnableCommonTimeSlot
节	[SystemConfig]
书写格式	EnableCommonTimeSlot=b
取值范围	b=0: 默认不开放; b=1: 开放语音卡 TDM 总线上所有时隙。
功能描述	开放 TDM 总线上的所有公共时隙给应用程序。
注意事项	● 本配置项只适用于将 Synway 板卡与第 3 方板卡进行互联的场合; ● 本配置项要求 SynCTI Ver. 5.2.0.0 及以后版本。

3.1.2.19 设置模拟电话线的 Caller ID 检测器（SHT+ATP 系列）

3.1.2.19.1 设置 Caller ID 检测器的工作模式

3.1.2.19.1.1 CallerIdStyle

配置项	CallerIdStyle
-----	---------------

节	[BoardId=x]
书写格式	格式 1: CallerIdStyle=m,m 格式 2: CallerIdStyle=m,m,m,m 格式 3: CallerIdStyle=m,m,m,m,m,m,m,m,m 格式 4: CallerIdStyle=m,m,m,m,m,m,m,m,m,m,m,m,m,m,m
取值范围	m=0: DTMF 制式, 配置项 DtmfCallerIDStyleLength 和 DtmfCallerIDInterTimeout 有效; m=1: FSK 制式, 配置项 CloseCallerIdOnReceived 和 FSKCallerIdDtrmTime 有效。 缺省值为 1。
功能描述	设置对端交换机在模拟电话线上发送主叫号码的模式。详细信息请参见第 1 章中的“ 模拟电话线上的主叫号码 ”部分内容。
注意事项	<ul style="list-style-type: none"> ● 本配置项只适用于 SHT 系列板卡的模拟中继线通道; ● 格式 1、格式 2 分别适用于 2 通道/4 通道规格 USB 语音盒和录音盒, 格式 3 适用于 8 通道规格的板卡, 格式 4 适用于 16 通道规格的板卡; ● 本配置项实现的功能也可以通过调用函数 SsmSetFlag (携带 F_CALLERIDTYPE 参数) 实现; ● 对 ATP-24A 系列板卡录音通道无效。

3.1.2.19.2 设置 FSK 制式的工作参数

3.1.2.19.2.1 CloseCallerIdOnReceived

配置项	CloseCallerIdOnReceived
节	[SystemConfig]
书写格式	CloseCallerIdOnReceived=b
取值范围	b=0: 不关闭; b=1: 关闭 (缺省值)。
功能描述	如果 Caller ID 检测器工作在 FSK 制式, 当驱动程序完成 FSK 制式的主叫号码的接收后, 本配置项设置是否由驱动程序自动关闭 Caller ID 检测器。更多信息请参见第 1 章中“ 模拟电话线上的主叫号码 ”部分内容。
注意事项	本配置项只适用于 SHT、ATP 系列板卡。

3.1.2.19.2.2 FSKCallerIdDtrmTime

配置项	FSKCallerIdDtrmTime
节	[BoardId=x]

书写格式	格式 1: FSKCallerIdDtrmTime=k,k 格式 2: FSKCallerIdDtrmTime=k,k,k,k 格式 3: FSKCallerIdDtrmTime=k,k,k,k,k,k,k,k 格式 4: FSKCallerIdDtrmTime=k,k,k,k,k,k,k,k,k,k,k,k,k,k
取值范围	k: 结束的判断时间, 单位为毫秒, $100 \leq k \leq 500$, 缺省值为 500ms。
功能描述	FSK 制式的 Caller ID 检测器启动后, 由于 FSK 制式的 Caller ID 号码在传送时没有明确的结束标志, 因此, 驱动程序只能依靠时间与数据的相关性来判断传送过程的结束, 即, 当收到的 Caller ID 字符的长度在经过了本参数设定的时间后, 仍然保持不变, 则驱动程序认为 FSK 制式的 Caller ID 信号传送结束。
注意事项	<ul style="list-style-type: none"> ● 本配置项只适用于 SHT 系列板卡的模拟中继线通道和 ATP 系列板卡的录音通道; ● 格式 1/格式 2 分别适用于 2 通道/4 通道规格 USB 语音盒和录音盒, 格式 3 适用于 8 通道规格的板卡, 格式 4 适用于 16 通道规格的板卡。

3.1.2.19.2.3 FilterInvalidCID

配置项	FilterInvalidCID
节	[BoardId=x]
书写格式	格式 1: FilterInvalidCID=b,b,b,b,b,b,b 格式 2: FilterInvalidCID=b,b,b,b,b,b,b,b,b,b,b,b,b
取值范围	b=0: 否; b=1: 是(缺省值)。
功能描述	设置是否开启“过滤错误的 FSK 方式主叫号码”功能。驱动程序收到的 FSK 方式主叫号码可能会发生错误: <ul style="list-style-type: none"> ◆ 号码并非对端交换机发送, 而是由于声音中的回波而形成的假号码; ◆ 号码的确是由对端交换机发送, 但在传输过程中由于干扰等因素, 信号被破坏, 形成了不完整的号码。
注意事项	<ul style="list-style-type: none"> ● 格式 1、格式 2 分别适用于 8 通道、16 通道规格的 SHT/ATP 系列板卡; ● 本配置项要求 SynCTI 驱动程序的版本为 Ver. 4.7.1.8 或更高。

3.1.2.19.2.4 FSKMinGate

配置项	FSKMinGate
节	[BoardId=x]

书写格式	格式 1: FSKMinGate =b,b,b,b,b,b,b 格式 2: FSKMinGate =b,b,b,b,b,b,b,b,b,b,b 格式 3: FSKMinGate =b,b,b,b,b,b,b,b,b,b,b,b,b,b,b,b,b 格式 4: FSKMinGate =b
取值范围	b=n(n>0): 能量门限; b=150: 缺省值 (SHT/ATP 系列板卡); b=60: 缺省值 (SHD 系列板卡)。
功能描述	设置接收 FSK 的能量门限值。
注意事项	<ul style="list-style-type: none"> • 格式 1、格式 2、格式 3 分别适用于 8 通道、16 通道、24 通道规格的 SHT/ATP 系列板卡; • 格式 4 适用于 SHD 系列板卡; • 本配置项要求 SynCTI 驱动程序的版本为 Ver. 5.0.2.0 或更高版本, 格式 4 要求 Ver. 5.3.1.3 或更高版本。

3.1.2.19.2.5 FskNoPhase

配置项	FskNoPhase
节	[BoardId=x]
书写格式	FskNoPhase=m
取值范围	m=0: 前导码为 0/1 翻转波形, 然后是全 1 标志码 (缺省值); m=1: 前导码为 0804, 然后是全 1 标志码。
功能描述	设置 FSK 主叫号码信号中前导码的格式。
注意事项	本配置项只适用于 SHT 系列板卡的模拟中继线通道和 ATP 系列板卡的录音通道。

3.1.2.19.3 设置 DTMF 制式的工作参数

3.1.2.19.3.1 DtmfCallerIDStyleLength

请参见 [DtmfCallerIDInterTimeOut](#)

3.1.2.19.3.2 DtmfCallerIDInterTimeOut

配置项	DtmfCallerIDStyleLength DtmfCallerIDInterTimeOut
节	[SystemConfig]

书写格式	DtmfCallerIDStyleLength=n DtmfCallerIDInterTimeOut=t
取值范围	n: 字符的最小个数, $1 \leq n \leq 29$, 缺省值为 1。 t: 时间间隔, 单位为毫秒, $t \geq 300$, 缺省值为 500。
功能描述	当 Caller ID 检测器工作在 DTMF 制式时, 如果驱动程序收到的主叫号码个数小于或等于 n, 已经收到的主叫号码将被抛弃; 每当驱动程序收到一个 DTMF 字符时, 如果此字符与上一个字符之间的时间间隔大于 t, 则以前收到的字符都将被抛弃。
注意事项	<ul style="list-style-type: none"> ● DtmfCallerIDInterTimeOut 要求驱动程序的版本为 4.5.2.1 或更高; ● 本配置项只适用于 SHT、ATP 系列板卡。

3.1.2.20 设置铃流信号检测器 (SHT+ATP 系列)

3.1.2.20.1 RingDetectFilterPara

配置项	RingDetectFilterPara
节	[SystemConfig]
书写格式	RingDetectFilterPara=t _H , t _L
取值范围	<p>t_H: 高电平的最小保持时间, 单位为 8 毫秒的倍数, $t_H \geq 1$, 缺省值为 6, 即 48 毫秒。只有当铃流信号保持高电平, 并且保持时间大于或等于 t_H 时, 驱动程序才会认为检测到了一次铃流信号。t_H 越大, 铃流信号检测器的灵敏度越低。</p> <p>t_L: 低电平的最小保持时间, 单位为 8 毫秒的倍数, $t_L \geq 1$, 缺省值为 40, 即 320 毫秒。只有当铃流信号保持低电平, 并且保持时间大于或等于 t_L 时, 驱动程序才会断定铃流信号消失。</p>
功能描述	设置铃流信号检测器的滤波参数。模拟中继线通道和磁石通道检测振铃的参数。驱动程序每隔 8 毫秒对模拟中继线通道和磁石通道的振铃信号进行采样。当实际低电平计数值大于或等于低电平判定值(第 2 个参数)时认为一次振铃信号结束; 当驱动程序检测到模拟中继线通道和磁石通道的振铃信号计数大于或等于高电平判定值(第 1 个参数)时判决为一次振铃。加大振铃信号高电平判定值会使振铃检测灵敏度降低, 反之增加。
注意事项	本配置项只适用于 SHT 系列的模拟中继线通道和磁石通道, 以及 ATP 系列的模拟电话线录音通道。

3.1.2.20.2 CallBackRingDetectFilterPara

配置项	CallBackRingDetectFilterPara
节	[SystemConfig]
书写格式	CallBackRingDetectFilterPara=t _H , t _L
取值范围	<p>t_H: 高电平的最大保持时间, 单位为 8 毫秒的倍数, $t_H \geq 1$, 缺省值为 25, 即 200 毫秒。</p> <p>t_L: 低电平的最大保持时间, 单位为 8 毫秒的倍数, $t_L \geq 1$, 缺省值为 40, 即 320 毫秒。</p>

功能描述	设置回呼铃流信号检测器的滤波参数。模拟中继线通道和磁石通道检测振铃的参数。驱动程序每隔8毫秒对模拟中继线通道和磁石通道的振铃信号进行采样。当实际高电平和低电平计数值满足 RingDetectFilterPara 配置同时满足高电平的保持时间小于t _H 并且低电平的保持时间小于t _L 时判决为一次回呼振铃，此功能可以用来检测两种不同的振铃信号。同时会抛出 E_CHG_RingCount 和 E_CHG_CallBackRingCount 事件。
注意事项	<ul style="list-style-type: none">本配置项只适用于SHT系列的模拟中继线通道和磁石通道，以及ATP系列的模拟电话线录音通道；RingDetectFilterPara用于振铃参数配置；CallBackRingDetectFilterPara用于回呼振铃参数配置。回呼振铃是一种特殊的振铃，即满足“RingDetectFilterPara的高电平保持时间<声音的高电平保持时间<CallBackRingDetectFilterPara的高电平保持时间，并且RingDetectFilterPara的低电平保持时间<声音的低电平保持时间<CallBackRingDetectFilterPara的低电平保持时间”的一种振铃。

3.1.2.20.3 RingEndDtrTime

配置项	RingEndDtrTime
节	[SystemConfig]
书写格式	RingEndDtrTime=k
取值范围	k>0：表示铃流是否已结束的最大等待时间。当铃流结束时间超过最大等待时间时，铃流需要重新开始计数。K的缺省值为6000，单位为毫秒。
功能描述	用来设置低电平最大保持时间，铃流信号的低电平的最大保持时间为“t _L +RingEndDtrTime”，其中t _L 为低电平最小保持时间，可通过配置项中的“RingDetectFilterPara=tH, tL”获得。即要能够检测到铃流信号，那么铃流信号的低电平持续时间必须满足“t _L ≤低电平持续时间≤t _L +RingEndDtrTime”这个条件才能被检测到。因此当铃流的周期太长时，需要增大该配置项来防止丢失铃流信号。
注意事项	本配置项只适用于SHT系列的模拟中继线通道和磁石通道，以及ATP系列的模拟电话线录音通道。

3.1.2.20.4 AlwaysToRingingOnRingCntX

配置项	AlwaysToRingingOnRingCntX
节	[SystemConfig]
书写格式	AlwaysToRingingOnRingCntX=k
取值范围	k=0：驱动程序检测到2个（Caller ID检测器设置为DTMF模式或FSK模式）完整的铃流信号后迁移到“振铃”状态； k≥1：无论Caller ID检测器工作于什么模式，驱动程序检测到k个完整的铃流信号后就迁移到“振铃”状态。 缺省值为0。
功能描述	设置铃流信号检测器的工作参数。详细信息请参见第1章中“ 模拟电话线上的铃流信号 ”部分内容。
注意事项	<ul style="list-style-type: none">本配置项可能会被配置项ChToRingingOnRingCnt所覆盖，详细信息请参见第1章中“模拟电话线上的铃流信号”部分内容；本配置项设定的参数也可以通过函数SsmSetFlag（携带F_ChToRingingOnRingCnt参数）进行设置；本配置项只支持SHT系列板卡的模拟中继线通道。

3.1.2.20.5ChToRingingOnRingCnt

配置项	ChToRingingOnRingCnt
节	[BoardId=x]
书写格式	格式 1: ChToRingingOnRingCnt=n,n,n,n 格式 2: ChToRingingOnRingCnt=n,n,n,n,n,n,n,n 格式 3: ChToRingingOnRingCnt=n,n,n,n,n,n,n,n,n,n,n,n,n
取值范围	n=0: 驱动程序检测到 2 个 (Caller ID 检测器设置为 DTMF 模式或 FSK 模式) 完整的铃流信号后 迁移到“振铃”状态; n≥1: 无论 Caller ID 检测器工作于什么模式, 驱动程序检测到 n 个完整的铃流信号后就迁移到“振 铃”状态。 缺省值为 0。
功能描述	设置铃流信号检测器的工作参数。详细信息请参见第 1 章中“ 模拟电话线上的铃流信号 ”部分内容。
注意事项	<ul style="list-style-type: none">本配置项只适用于 SHT 系列板卡的模拟中继线通道;如果在某个通道上指定了本配置项目 n>0, 该通道基于配置项 AlwaysToRingingOnRingCntX 的 参数将被覆盖, 详细信息请参见第 1 章中“模拟电话线上的铃流信号”部分内容;本配置项设定的参数也可以通过函数 SsmSetFlag (携带 F_ChToRingingOnRingCnt 参数) 进 行设置;格式 1 适用于模拟外线系列的 USB 语音盒;格式 2 适用于 8 通道规格的模拟外线系列板卡;格式 3 适用于 16 通道规格的模拟外线系列板卡。

3.1.2.21 设置 FSK 数据收发器(CTI 系列)

3.1.2.21.1 设置 FSK 数据发送器的工作参数

3.1.2.21.1.1 FreqBit0

请参见 [MdlAmp](#)

3.1.2.21.1.2 FreqBit1

请参见 [MdlAmp](#)

3.1.2.21.1.3 Baudrate

请参见 [MdlAmp](#)

3.1.2.21.1.4 MdlAmp

配置项	FreqBit0 FreqBit1 Baudrate MdlAmp
节	[FskConfig]
书写格式	$\text{FreqBit0} = f_0$ $\text{FreqBit1} = f_1$ $\text{Baudrate} = k$ $\text{MdlAmp} = a$
取值范围	f_0 : 比特 0 的载波频率, 单位为 Hz, $300 \leq f_0 \leq 3400$, 缺省值为 2200Hz。 f_1 : 比特 1 的载波频率, 单位为 Hz, $300 \leq f_1 \leq 3400$, 缺省值为 1200Hz。 k : 波特率, 单位为 bps, $300 \leq k \leq 2400$, 缺省值为 1200bps。 a : 信号的幅度, $0 \leq a \leq 255$, 缺省值为 128。
功能描述	FreqBit0 、 FreqBit1 分别设置比特 0 和比特 1 的载波频率; Baudrate 设置波特率; MdlAmp 设置调制信号的幅度。有关 FSK 数据发送器的更多信息请参见第 1 章中“ FSK 数据收发器 ”部分内容。

3.1.2.22 设置 FSK 数据接收器(CTI 系列)

3.1.2.22.1 FskMarkSignal

配置项	FskMarkSignal
节	[BoardId=x]
书写格式	格式 1: $\text{FskMarkSignal} = m_0, m_1, m_2, m_3, m_4, m_5, m_6, m_7$ 格式 2: $\text{FskMarkSignal} = m_0, m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9, m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}$ 格式 3: $\text{FskMarkSignal} = m$
取值范围	m, m_i : 标志码的个数 (下标 i 表示通道的物理编号), 可以使用的值为: $=0$: 标志码为连续 12 个 1 (缺省值); $=1$: 标志码为连续 8 个 1。
功能描述	设置 FSK 数据流中标志码 (前导信号) 的格式。

注意事项	<ul style="list-style-type: none">● 格式 1、格式 2 分别适用于 8 通道、16 通道规格的 SHT 系列板卡；● 格式 3 适用于 SHD 系列板卡。
------	------------------------------------------------------------------------------------------------------------------------

3.1.2.22.2 FskFrameMode

配置项	FskFrameMode
节	[BoardId=x]
书写格式	FskFrameMode=m
取值范围	m: 可以使用的值为： =0: 采用格式“标志码+数据帧”； =1: 采用格式“同步码+标志码+数据帧”（缺省值）； 关于 FSK 帧格式请参考“ FSK 数据收发器（CTI 系列） ”
功能描述	设置 FSK 数据接收器默认接收的帧格式。
注意事项	本配置项适用于所有 SHT 系列和 SHD 系列板卡。

3.1.2.22.3 FskEchoCancelDelay

配置项	FskEchoCancelDelay
节	[BoardId=x]
书写格式	格式 1: FskEchoCancelDelay=b ₀ ,b ₁ ,b ₂ ,b ₃ ,b ₄ ,b ₅ ,b ₆ ,b ₇ ,b ₈ ,b ₉ ,b ₁₀ ,b ₁₁ ,b ₁₂ ,b ₁₃ ,b ₁₄ ,b ₁₅ 格式 2: FskEchoCancelDelay=b ₀ ,b ₁ ,b ₂ ,b ₃ ,b ₄ ,b ₅ ,b ₆ ,b ₇
取值范围	b _i =0: 关闭； b _i =1: 开启（缺省值）。 注：下标 i 为通道的物理编号。
功能描述	当 FSK 接收器处于工作状态时，是否关闭回波抵消器。
注意事项	<ul style="list-style-type: none">● 格式 1、格式 2 分别适用于 8 通道、16 通道规格的 SHT 系列板卡；● 当通道上启动了 FSK 接收器后，如果本配置项配置为 1，则强制关闭回波抵消器；否则，回波抵消器的工作状态由配置项 EnableEchoCancellor 决定；● 本配置项要求 SynCTI 的版本为 Ver4.5.6.2 或更高；● 函数 SsmSetFlag（携带 F_EchoCancelInFsk 参数）可以实现相同功能。

3.1.2.23 设置传真资源通道的工作参数

3.1.2.23.1 MaxSpeed

配置项	MaxSpeed
节	[Fax]
书写格式	MaxSpeed=k
取值范围	k: 可用的传真速率, 单位为 bps, 可以使用的值为 4800、9600、14400、33600 缺省值为 9600。
功能描述	设置接收或发送传真时使用的最大速率。

3.1.2.23.2 RcvDisTime

配置项	RcvDisTime
节	[Fax]
书写格式	RcvDisTime=n
取值范围	0≤n≤10, 缺省值为 4。
功能描述	传真过程开始后, 如果出现检测不到 Dis 信号或 Dis 信号无效的情况, 本配置项设置重复检测 Dis 信号的次数。该配置项在 Ver4.7.2.0 版本上支持。
注意事项	本配置项要求 SynCTI 驱动程序的版本为 4.7.2.0 或更高。

3.1.2.23.3 ResendForRTN

配置项	ResendForRTN
节	[Fax]
书写格式	ResendForRTN=k
取值范围	k=0: 收到 RTN 消息后不重发页面数据 (缺省值); k=1: 收到 RTN 消息后重发页面数据。
功能描述	设置在传真发送过程中收到 RTN 消息时是否进行页面重发。

3.1.2.23.4 ResetRcvForRTN

配置项	ResetRcvForRTN
节	[Fax]
书写格式	ResetRcvForRTN =k
取值范围	k=0: 传真接收到效果不好的页面时, 发送 RTN 消息(缺省值); k=1: 传真接收到效果不好的页面时, 不发送 RTN, 直接挂断。
功能描述	在传真接收过程中接收到效果不好的页面数据时, 判断是发送 RTN 还是直接挂断。

3.1.2.23.5 KeepPageForRTN

配置项	KeepPageForRTN
节	[Fax]
书写格式	KeepPageForRTN =k
取值范围	k=0: 传真接收到效果不好的页面时, 不保留此页面; k=1: 传真接收到效果不好的页面时, 保留此页面(缺省值)。
功能描述	传真接收到效果不好的页面数据时, 判断是否保留此页面。

3.1.2.23.6 PercentageForRTN

配置项	PercentageForRTN
节	[Fax]
书写格式	PercentageForRTN =n
取值范围	0≤n≤100, 缺省值为 30。
功能描述	当接收到的页面数据大于此百分比时, 认为页面效果不好, 否则认为页面数据正常。

3.1.2.23.7 FaxCodeMode

配置项	FaxCodeMode
节	[Fax]
书写格式	FaxCodeMode =k
取值范围	k=0: 最高支持 MH 格式 (缺省值); k=1: 最高支持 MR 格式; k=2: 最高支持 MMR 格式。
功能描述	设置接收或发送传真时使用的最高的编解码格式。

3.1.2.23.8 FaxSendDisTime

配置项	FaxSendDisTime
节	[Fax]
书写格式	FaxSendDisTime =n
取值范围	3000≤n≤6000, 单位为 ms, 缺省值为 6000。
功能描述	设置 DIS 的重发时间间隔。

3.1.2.24 设置电压检测器

3.1.2.24.1 忽略电压检测器的检测结果

3.1.2.24.1.1 IgnoreLineVoltage

配置项	IgnoreLineVoltage
节	[BoardId=x]
书写格式	格式 1: IgnoreLineVoltage=b,b,b,b 格式 2: IgnoreLineVoltage=b,b,b,b,b,b,b 格式 3: IgnoreLineVoltage=b,b,b,b,b,b,b,b,b,b,b 格式 4: IgnoreLineVoltage=b,b,b,b,b,b,b,b,b,b,b,b,b,b,b

取值范围	b=0: 否（缺省值）; b=1: 是。
功能描述	设置是否忽略线路上的电压检测结果。详细内容请参见第 1 章中的“ 模拟电话线的电压变化 ”部分。
注意事项	<ul style="list-style-type: none">● 本配置项只适用于 ATP 系列板卡的录音通道（包括模拟线路录音模块和麦克风模块）；● 格式 1 适用于 ATP 系列的 USB 语音盒/录音盒；● 格式 2 适用于 8 通道规格的 ATP 系列板卡；● 格式 3 适用于 16 通道规格的 ATP 系列板卡；● 格式 4 适用于 24 通道规格的 ATP 系列板卡。
相关函数	SsmSetIgnoreLineVoltage

3.1.2.24.2 设置振铃检测门限值

3.1.2.24.2.1 JudgeLineVoltage

配置项	JudgeLineVoltage
节	[SystemConfig]
书写格式	JudgeLineVoltage=k
取值范围	0<k<256：振铃时，线路当前电压与前 8ms 的线路电压的差值，单位为 0.5 伏 (0.5V)，缺省值为 7V($15 \times 0.5V$)。
功能描述	设置用来调节适应模拟线路上不同振铃的电压伏值。
注意事项	<ul style="list-style-type: none">本配置项适用于 ATP/SHT 系列板卡的录音通道（包括模拟线路录音模块和外线模块）。

3.1.2.24.3 设置摘机/挂机判决条件

3.1.2.24.3.1 IsHangupDtrmVoltage

配置项	IsHangupDtrmVoltage
节	[BoardId=x]
书写格式	格式 1: IsHangupDtrmVoltage=k,k,k,k 格式 2: IsHangupDtrmVoltage=k,k,k,k,k,k,k,k 格式 3: IsHangupDtrmVoltage=k,k,k,k,k,k,k,k,k,k,k,k,k,k
取值范围	k: 判定电压, 单位为伏 (V), 缺省值为 26V。

功能描述	设置 ATP 系列板卡的录音通道的摘/挂机判定电压。详细内容请参见第 1 章中“ 模拟电话线的电压变化 ”部分。
注意事项	<ul style="list-style-type: none">● 本配置项只适用于 ATP 系列板卡的录音通道（包括模拟线路录音模块和麦克风模块）；● 格式 1 适用于 ATP 系列的 USB 语音盒/录音盒；● 格式 2 适用于 8 通道规格的 ATP 系列板卡；● 格式 3 适用于 16 通道规格的 ATP 系列板卡。
相关函数	SsmSetDtrmLineVoltage

3.1.2.24.3.2 LineOncnt

配置项	LineOncnt
节	[SystemConfig]
书写格式	LineOncnt=t
取值范围	t: 最小保持次数为 1，取值范围是 $1 \leq t \leq 65536$ ，每次表示 8 毫秒，缺省值为 25 次（即 200 毫秒）。
功能描述	当线路电压符合摘机判定电压的条件，并且保持本配置项设定的最小保持时间后，驱动程序才认为线路上发生了摘机动作。更多信息请参见第 1 章中“ 模拟电话线的电压变化 ”部分内容。
注意事项	<ul style="list-style-type: none">● 本配置项只适用于 ATP 系列板卡，或 SHT 系列板卡上的录音模块；● 本配置项要求驱动程序的版本为 Ver. 4.7.2.0 或更高。

3.1.2.24.4 设置断线监测的判定条件

3.1.2.24.4.1 OffLineSet

配置项	OffLineSet
节	[SystemConfig]
书写格式	OffLineSet=k
取值范围	k: 电压门限值，是实际电压值的 2 倍，缺省值为 5，表示实际电压为 2.5V。
功能描述	设置模拟电话线的断线判定电压门限值。详细内容请参见第 1 章中“ 模拟电话线的电压变化 ”部分。
注意事项	本配置项只适用于模拟中继线的录音通道。

3.1.2.24.4.2 OffLineDetermineTime

配置项	OffLineDetermineTime
-----	-----------------------------

节	[SystemConfig]
书写格式	OffLineDetermineTime=t
取值范围	t: 最小保持时间, 单位为毫秒, 缺省值为 200。
功能描述	设置判定模拟电话线断开的最小保持时间。详细内容请参见第 1 章中“ 模拟电话线的电压变化 ”部分。
注意事项	本配置项只适用于 ATP 系列板卡的录音通道和 SHT 系列板卡的模拟中继线通道。

3.1.2.24.5 设置极性反转信号是否影响通道的状态迁移

3.1.2.24.5.1 DisablePolarReverse

配置项	DisablePolarReverse
节	[BoardId=x]
书写格式	格式 1: DisablePolarReverse=b,b,b,b 格式 2: DisablePolarReverse=b,b,b,b,b,b,b 格式 3: DisablePolarReverse=b,b,b,b,b,b,b,b,b,b,b 格式 4: DisablePolarReverse= [0,29]:b
取值范围	b=0: 影响 (缺省值); b=1: 不影响。
功能描述	设置极性反转信号是否影响通道的状态迁移。详细内容请参见第 1 章中“ 模拟电话线的电压变化 ”部分。
注意事项	<ul style="list-style-type: none"> ● 本配置项只适用 SHT 系列板卡的模拟中继线通道; ● 格式 1 适用于 SHT 系列的 USB 语音盒/录音盒; ● 格式 2 适用于 8 通道规格的 SHT 系列板卡; ● 格式 3 适用于 16 通道规格的 SHT 系列板卡; ● 格式 4 适用于 30 通道规格的远端坐席接入模块。

3.1.2.24.6 设置极性反转的线路干扰忽略电压

3.1.2.24.6.1 PolarIgnore

配置项	PolarIgnore
节	[BoardId=x]
书写格式	PolarIgnore =b
取值范围	b>=0, 缺省值为 0。

功能描述	设置在极性反转线路上忽略干扰电压的值，避免驱动将干扰电压误判为极性反转信号。PolarIgnore=b 时，电压值小于等于 b 的都会被忽略，而不会被认为是极性反转信号。详细内容请参见第 1 章中“ 模拟电话线的电压变化 ”部分。
注意事项	本配置项适用 ATP 系列的模拟电话线录音通道和 SHT 系列板卡的模拟中继线通道。

3.1.2.25 设置 USB 录音盒/语音盒中的内置板载扬声器

3.1.2.25.1 USBLLine0Output

配置项	USBLLine0Output
节	[BoardId=x]
书写格式	USBLLine0Output=m
取值范围	m=0: 将声音送到板载扬声器进行播放; m=1: 将声音送到板载扬声器插座，由外置扬声器进行播放。
功能描述	开启或关闭 USB 录音盒/语音盒中的内置板载扬声器。更多信息请参见第 1 章中“ ATP 系列板卡的工作原理图 ”或“ SHT 系列板卡的工作原理图 ”部分内容。
注意事项	<ul style="list-style-type: none">● 本配置项只适用于 ATP 系列的 USB 录音盒和 SHT 系列的 USB 语音盒；● 本配置项要求 SynCTI 驱动程序的版本为 4.7.1.7 或更高。

3.1.2.26 SHT 系列板卡常用配置项(CTI 系列)

3.1.2.26.1 模拟中继通道专用配置项

3.1.2.26.1.1 设置模拟中继通道为录音通道

3.1.2.26.1.1.1 SetAnalogChToRecCh

配置项	SetAnalogChToRecCh
节	[Boardid=x]
书写格式	格式 1: SetAnalogChToRecCh =b,b,b,b 格式 2: SetAnalogChToRecCh =b,b,b,b,b,b,b,b 格式 3: SetAnalogChToRecCh =b,b,b,b,b,b,b,b,b,b,b,b
取值范围	b=1: 通道开启，将外线通道作为录音通道； b=0: 通道关闭(缺省值)，此配置项无效。
功能描述	强制设置模拟中继通道为录音通道类型。

注意事项	<ul style="list-style-type: none">该配置项对 5.0.1.0 及以上版本的驱动才有效；本配置项只适用模拟中继通道，包括安装了模拟中继模块的 SHT 系列板卡（SHT D 型卡除外）和 USB 语音盒；格式 1 适用于 SHT 系列的 USB 语音盒；格式 2 适用于 8 通道规格的 SHT 系列板卡；格式 3 适用于 16 通道规格的 SHT 系列板卡。
------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3.1.2.26.1.2 设置去话呼叫的参数

3.1.2.26.1.2.1 MaxWaitDialToneTime

配置项	MaxWaitDialToneTime
节	[SystemConfig]
书写格式	MaxWaitDialToneTime=t
取值范围	t: 最大等待时间，单位为秒， $0 \leq t \leq 30$ ，缺省值为 3 秒。
功能描述	设置等待拨号音的最大等待时间。 应用程序在调用函数 SsmAutoDial 函数启动 AutoDial 任务后，驱动程序需要首先检测拨号音。如果在本配置项指定的时间内不能检测到拨号音，AutoDial 任务失败。 有关 AutoDial 任务的执行过程请参见第 1 章中“ 模拟中继线通道的状态转移 ”部分内容。
注意事项	本配置项只适用于模拟中继线通道。

3.1.2.26.1.3 设置模拟中继通道是否开启校准

3.1.2.26.1.3.1 AdjustImmediately

配置项	AdjustImmediately
节	[BoardId=x]
书写格式	AdjustImmediately =b
取值范围	b=0: 开启（缺省值）; b=1: 关闭。
功能描述	设置模拟外线通道摘机时是否启动校准。

3.1.2.26.2 设置闪断信号及保持时间

3.1.2.26.2.1 DefaultTxFlashChar

配置项	DefaultTxFlashChar
-----	---------------------------

节	[SystemConfig]
书写格式	DefaultTxFlashChar =s
取值范围	s: 闪断表示符, 缺省值为“!”, 不能为标准 DTMF 字符或“? ”。
功能描述	设置发送闪断的表示符。当对函数 SsmTxDtmf 传入该字符后, 即发送一个闪断信号。

3.1.2.26.2.2 DefaultTxFlashTime

配置项	DefaultTxFlashTime
节	[SystemConfig]
书写格式	DefaultTxFlashTime=t
取值范围	32≤t≤1000, 单位为毫秒, 缺省值为 500。
功能描述	设置使用!字符作为参数调用函数 SsmTxDtmf 在模拟中继线上产生闪断信号时, 闪断信号的保持时间。详细内容请参见第 1 章中“ 在模拟线路上产生闪断信号 ”部分内容。

3.1.2.26.2.3 设置闪断后是否保持通道状态

3.1.2.26.2.3.1 AfterFlashNotAffectChState

配置项	AfterFlashNotAffectChState
节	[SystemConfig]
书写格式	AfterFlashNotAffectChState=b
取值范围	b=0: 否 (缺省值); b=1: 是。
功能描述	设置闪断后是否保持通道状态。
注意事项	本配置项只适用于模拟中继线通道。

3.1.2.26.2.4 远端摘机检测器

3.1.2.26.2.4.1 设置普通的远端摘机检测器的工作参数

3.1.2.26.2.4.1.1 WaitAfterDialTime

配置项	WaitAfterDialTime
节	[SystemConfig]

书写格式	WaitAfterDialTime=t
取值范围	t: 最大等待时间, 单位为毫秒, $16 \leq t \leq 60000$, 缺省值为 18000。
功能描述	应用程序在模拟中继线通道上调用 SsmAutoDial 函数后, 如果驱动程序向对端交换机发送完全部被叫号码后没有在线路上检测到回铃音, 也没有在本配置项设定的时间内检测到任何语音活动, 驱动程序会输出 E_CHG_ToneAnalyze 事件 (携带 CHKTONE_NOVOICE 参数)。更多信息请参见第 1 章中 “ 普通的远端摘机检测器 ” 部分内容。
注意事项	本配置项适用于 SHT 系列板卡的模拟中继线通道。 如果数字卡/IP 卡的 DefaultToneCheckState = 1, 那么此配置项的功能与在 SHT 系列板卡下功能类似。

3.1.2.26.2.4.1.2 MaxWaitVocAfterEcho

配置项	MaxWaitVocAfterEcho
节	[SystemConfig]
书写格式	MaxWaitVocAfterEcho=t
取值范围	t: 最大等待时间, 单位为秒, 缺省值为 10。
功能描述	应用程序在模拟中继线通道上调用 SsmAutoDial 函数后, 如果驱动程序向对端交换机发送完全部被叫号码后, 在线路上检测到了回铃音, 但没有在本配置项设定的时间内检测到任何语音活动, 驱动程序会输出 E_CHG_ToneAnalyze 事件 (携带 CHKTONE_ECHO_NOVOICE 参数)。更多信息请参见第 1 章中 “ 普通的远端摘机检测器 ” 部分内容。
注意事项	本配置项只适用于 SHT 系列板卡的模拟中继线通道。

3.1.2.26.2.4.1.3 VoiceOnDetermineTime

配置项	VoiceOnDetermineTime
节	[SystemConfig]
书写格式	VoiceOnDetermineTime=t
取值范围	$16 \leq t \leq 1024$, 单位为毫秒, 必须为 16 的整数倍, 缺省值为 96ms。
功能描述	设置语音活动信号的最小保持时间。只有当线路上连续出现语音活动信号, 并且保持的时间大于本配置项的设定值时, 驱动程序才会确认线路上出现了真正的语音活动信号。更多信息请参见第 1 章中 “ 普通的远端摘机检测器 ” 部分内容。
注意事项	本配置项适用于 SHT 系列板卡的模拟中继线通道。 如果数字卡/IP 卡的 DefaultToneCheckState = 1, 那么此配置项的功能与在 SHT 系列板卡下功能类似。

3.1.2.26.2.4.1.4 EchoNoVoiceToTalking

配置项	EchoNoVoiceToTalking
节	[SystemConfig]
书写格式	EchoNoVoiceToTalking=N

取值范围	N=0: 不启用该功能(缺省值); N=1: 启动该功能。
功能描述	当遇到线路上回铃过后的摘机脉冲过小或者没有时,会抛出“回铃后无声”事件。当“回铃后无声”事件产生时,设置本配置项,可使通道自动进入通话状态。适用于一些特殊场合。更多信息请参见第1章中“ 普通的远端摘机检测器 ”部分内容。
注意事项	本配置项只适用于SHT系列板卡的模拟中继线通道。

3.1.2.26.2.4.2 设置增强的远端摘机检测器

3.1.2.26.2.4.2.1 RelativeEngyHookDetect

配置项	RelativeEngyHookDetect
节	[SystemConfig]
书写格式	RelativeEngyHookDetect=b
取值范围	b=0: 关闭; b=1: 开启(缺省值)。
功能描述	设置模拟中继通道上“增强的远端摘机检测器”的工作状态。更多信息请参见第1章中“ 增强的远端摘机检测器 ”部分内容。
注意事项	<ul style="list-style-type: none">● 本配置项只适用于SHT系列板卡的模拟中继线通道,并且要求SynCTI的版本为3.5.2.4或更高;● 本配置项实现的功能也可以通过调用函数SsmSetFlag(携带F_RELATIVEENGYHOOKDETECT参数)实现。

3.1.2.26.2.4.2.2 HookEngyConfigMulti

请参见[HookValidEngyCnt](#)

3.1.2.26.2.4.2.3 HookValidEngyCnt

配置项	HookEngyConfigMulti HookValidEngyCnt
节	[SystemConfig]
书写格式	HookEngyConfigMulti=k HookValidEngyCnt=t
取值范围	k: 灵敏度, $2 \leq k \leq 65535$, 值越大, 灵敏度越低, 即漏检测的概率增加, 而误检测的概率减小。缺省值为6。 t: 最小保持时间, 单位为毫秒, $32 \leq t \leq 65535$, 必须为16的整数倍, 缺省值为48ms。值越大, 灵敏度越低, 即漏检测的概率增加, 而误检测的概率减小。

功能描述	设置模拟中继线通道上的“增强的远端摘机检测器”的工作参数，HookEngyConfigMulti 设置摘机检测的灵敏度，HookValidEngyCnt 设置最小保持时间，以消除信号抖动对判断结果造成的干扰。只有当通道上检测到了瞬时的摘机动作，并且保持时间大于或等于 t，驱动程序才会判定线路上发生了真正的摘机动作。更多信息请参见第 1 章中“ 增强的远端摘机检测器 ”部分内容。
注意事项	<ul style="list-style-type: none">本配置项只适用于 SHT 系列板卡的模拟中继线通道，只有当配置项 RelativeEngyHookDetect 设置为 1 时有效；本配置项要求 SynCTI 的版本为 3.5.2.4 或更高。

3.1.2.26.3 坐席通道专用配置项

3.1.2.26.3.1 AutoSendDialTone

配置项	AutoSendDialTone
节	[BoardId=x]
书写格式	格式 1: AutoSendDialTone=b,b,b,b 格式 2: AutoSendDialTone=b,b,b,b,b,b,b 格式 3: AutoSendDialTone=b,b,b,b,b,b,b,b,b,b,b,b 格式 4: AutoSendDialTone=[S ₁ ,E ₁]:b+...+[S _m ,E _m]:b
取值范围	b=1: 开启； b=0: 关闭(缺省值)。 S ₁ : 起始通道编号，缺省值为 0。 E ₁ : 结束通道编号，缺省值为板卡上的总通道数。 m: 参数段的数量，最小为 1，最大不能超过板卡上的总通道数。
功能描述	设置当检测到与坐席通道相连的话机摘机后，是否由驱动程序自动发送拨号音的功能。
注意事项	<ul style="list-style-type: none">本配置项只适用坐席通道，包括安装了坐席模块的 SHT 系列板卡和连接到远端坐席接入模块的 SHD 系列板卡；格式 1 适用于 SHT 系列的 USB 语音盒；格式 2 适用于 8 通道规格的 SHT 系列板卡；格式 3 适用于 16 通道规格的 SHT 系列板卡；格式 4 适用于连接到远端坐席接入模块的 SHD 系列板卡，详细信息请参见第 1 章中“连接到远端坐席接入模块”部分内容。
使用范例	SHD-30A-CT/PCI/SS1 板卡连接到了远端坐席接入模块，希望开启前 15 个通道的本项功能，关闭后 15 个通道的本项功能，可以按照下列方式设置： AutoSendDialTone=[0,14]:1+[15,29]:0 如果全部 30 个通道都设置为开启，可以按照下列方式设置： AutoSendDialTone=[0, 29]:1

3.1.2.26.3.2 StopSendDialToneOnDtmf

配置项	StopSendDialToneOnDtmf
节	[BoardId=x]
书写格式	<p>格式 1: StopSendDialToneOnDtmf=b,b,b,b</p> <p>格式 2: StopSendDialToneOnDtmf=b,b,b,b,b,b,b</p> <p>格式 3: StopSendDialToneOnDtmf=b,b,b,b,b,b,b,b,b,b,b,b</p> <p>格式 4: StopSendDialToneOnDtmf =[S₁,E₁]:b+...+[S_m,E_m]:b</p>
取值范围	<p>b=1: 开启 (缺省值);</p> <p>b=0: 关闭。</p> <p>S₁: 起始通道编号, 缺省值为 0。</p> <p>E₁: 结束通道编号, 缺省值为板卡上的总通道数。</p> <p>m: 参数段的数量, 最小为 1, 最大不能超过板卡上的总通道数。</p>
功能描述	当驱动程序正在向话机发送信号音的时候, 如果检测到了 DTMF 字符, 本配置项设置是否启用由驱动程序自动停止发送拨号音的功能。
注意事项	<ul style="list-style-type: none"> 本配置项只适用坐席通道, 包括安装了坐席模块的 SHT 系列板卡和连接到远端坐席接入模块的 SHD 系列板卡; 格式 1 适用于 SHT 系列的 USB 语音盒; 格式 2 适用于 8 通道规格的 SHT 系列板卡; 格式 3 适用于 16 通道规格的 SHT 系列板卡; 格式 4 适用于连接到远端坐席接入模块的 SHD 系列板卡, 详细信息请参见第 1 章中 “连接到远端坐席接入模块” 部分内容。配置范例请参考配置项 AutoSendDialTone 中的范例。

3.1.2.26.3.3 MaxLocalFlashTime

配置项	MaxLocalFlashTime
节	[SystemConfig]
书写格式	MaxLocalFlashTime=t
取值范围	32≤t≤2000, 单位为毫秒, 缺省值为 700。
功能描述	设置判断闪断信号的最大保持时间。
注意事项	本函数只适用于坐席通道。

3.1.2.26.3.4 UserOnHookFilterTime

配置项	UserOnHookFilterTime
节	[BoardId=x]
书写格式	格式 1: UserOnHookFilterTime=t,t,t,t 格式 2: UserOnHookFilterTime=t,t,t,t,t,t,t,t 格式 3: UserOnHookFilterTime=t,t,t,t,t,t,t,t,t,t,t,t 格式 4: UserOnHookFilterTime =[S ₁ ,E ₁]:t+...+ [S _m ,E _m]:t
取值范围	t: 摘机信号的最小保持时间, 单位为毫秒, 必须为 16 毫秒的整数倍, 值越小越敏感。缺省值为 64。 S ₁ : 起始通道编号, 缺省值为 0。 E ₁ : 结束通道编号, 缺省值为板卡上的总通道数。 m: 参数段的数量, 最小为 1, 最大不能超过板卡上的总通道数。
功能描述	设置摘机信号的最小保持时间。
注意事项	<ul style="list-style-type: none"> ● 本配置项只适用坐席通道, 包括安装了坐席模块的 SHT 系列板卡和连接到远端坐席接入模块的 SHD 系列板卡; ● 格式 1 适用于 SHT 系列的 USB 语音盒; ● 格式 2 适用于 8 通道规格的 SHT 系列板卡; ● 格式 3 适用于 16 通道规格的 SHT 系列板卡; ● 格式 4 适用于连接到远端坐席接入模块的 SHD 系列板卡, 详细信息请参见第 1 章中 “连接到远端坐席接入模块” 部分内容。配置范例请参考配置项 AutoSendDialTone 中的范例; ● 本配置项的缺省值已经能够适应绝大多数的应用场合, 通常无须设置。

3.1.2.26.3.5 UserChGenerateRingMode

配置项	UserChGenerateRingMode
节	[BoardId=x]
书写格式	UserChGenerateRingMode=m
取值范围	m=0: 1 秒高电平、4 秒低电平方式 (缺省值); m=1: 持续高电平, 直到应用程序发出停止命令。
功能描述	设置坐席通道或磁石话机通道产生振铃信号的方式。

3.1.2.26.3.6 UserSendPolar

配置项	UserSendPolar
-----	----------------------

节	[BoardId=x]
书写格式	格式 1: UserSendPolar=b,b,b,b 格式 2: UserSendPolar=b,b,b,b,b,b,b,b 格式 3: UserSendPolar=b,b,b,b,b,b,b,b,b,b,b,b
取值范围	b=0: 否 (缺省值); b=1: 是。
功能描述	设置坐席通道是否支持产生极性反转信号的功能。详细信息请参见第 1 章中“在电话线上产生极性反转信号”部分内容。
注意事项	<ul style="list-style-type: none"> ● 本配置项只适用 SHT 系列板卡的坐席通道; ● 格式 1 适用于 SHT 系列的 USB 语音盒; ● 格式 2 适用于 8 通道规格的 SHT 系列板卡; ● 格式 3 适用于 16 通道规格的 SHT 系列板卡。

3.1.2.26.3.7 LocalHookFilterTime

配置项	LocalHookFilterTime
节	[SystemConfig]
书写格式	LocalHookFilterTime=t
取值范围	16≤t≤MaxLocalFlashTime, 单位为毫秒, 缺省值为 16。
功能描述	设置判定闪断信号的最小保持时间。
注意事项	本函数只适用于坐席通道。

3.1.2.26.4 传真通道专用配置项

3.1.2.26.4.1 MaxFaxChannel

配置项	MaxFaxChannel
节	[BoardId=x]
书写格式	MaxFaxChannel=n
取值范围	取值范围及缺省值与板卡的型号有关, 详细内容请参见第 1 章中“ 设置传真资源通道的数量 ”部分内容。
功能描述	设置传真资源通道的总数。

3.1.2.26.4.2 DSP3WORKMODE

配置项	DSP3WORKMODE
节	[BoardId=x]
书写格式	DSP3WORKMODE=n
取值范围	n=1: 设置 SHT-16B-CT/PCI/FAX 卡具备传真资源能力, 针对 SHT-16B-CT/PCI/FAX 卡 (默认); n=2: 设置 SHT-16B-CT/PCI/FAX 卡具备硬件 GSM 录音能力, 针对 SHT-16B-CT/PCI/FAX 卡; n=3: 设置 SHT-16B-CT/PCI/MP3 和 SHT-16B-CT/cPCI/MP3 卡具备硬件 MP3 录音能力, 针对 SHT-16B-CT/PCI/MP3 和 SHT-16B-CT/cPCI/MP3 卡。
功能描述	如果是针对 SHT-16B-CT/PCI/FAX 卡, DSP3WORKMODE 配置项用于设置 SHT-16B-CT/PCI/FAX 卡是具有传真资源能力还是具有硬件 GSM 录音能力。如果是针对 SHT-16B-CT/PCI/MP3 和 SHT-16B-CT/cPCI/MP3 板卡, DSP3WORKMODE 配置项用于设置 SHT-16B-CT/PCI/MP3 和 SHT-16B-CT/cPCI/MP3 具备硬件 MP3 录音能力。更多信息请参见第 1 章中 “ SHT 系列板卡简介 ” 部分内容。
注意事项	SHT-16B-CT/PCI/MP3 在配置工具中显示的卡型号为 SHT-16B-CT/PCI/FAX, 正确使用 SHT-16B-CT/PCI/MP3 需要配置 DSP3WORDMODE=3。

3.1.2.26.5 联合模块的专用配置项

3.1.2.26.5.1 UnimoduleState

配置项	UnimoduleState
节	[BoardId=x]
书写格式	格式 1: UnimoduleState=b,b,b,b 格式 2: UnimoduleState=b,b,b,b,b,b,b 格式 3: UnimoduleState=b,b,b,b,b,b,b,b,b,b,b
取值范围	b=0: 是; b=1: 否 (缺省值)。
功能描述	设置是否将联合模块上的模拟中继线通道直接连接到坐席通道。
注意事项	<ul style="list-style-type: none">● 格式 1 适用于 SHT 系列的 USB 语音盒;● 格式 2 适用于 8 通道规格的 SHT 系列板卡;● 格式 3 适用于 16 通道规格的 SHT 系列板卡。

3.1.2.26.6 无业务模块通道的专用配置项

3.1.2.26.6.1 NoModuleChBusRec

配置项	NoModuleChBusRec
节	[BoardId=x]
书写格式	格式 1: NoModuleChBusRec=b,b,b,b 格式 2: NoModuleChBusRec=b,b,b,b,b,b,b,b 格式 3: NoModuleChBusRec=b,b,b,b,b,b,b,b,b,b,b,b
取值范围	b=0: 关闭 (缺省值); b=1: 开启。
功能描述	设置是否将来自总线的语音信号, 而不是来话信号, 作为 Barge in 检测器 的输入信号源。更多信息请参见第 1 章中 “ 使用未安装业务模块的通道 ” 部分内容。
注意事项	<ul style="list-style-type: none">● 格式 1 适用于 SHT 系列的 USB 语音盒;● 格式 2 适用于 8 通道规格的 SHT 系列板卡;● 格式 3 适用于 16 通道规格的 SHT 系列板卡。

3.1.2.26.7 磁石通道

3.1.2.26.7.1 IsMagnetModule

配置项	IsMagnetModule
节	[BoardId=x]
书写格式	格式 1: IsMagnetModule=b,b,b,b 格式 2: IsMagnetModule=b,b,b,b,b,b,b,b 格式 3: IsMagnetModule=b,b,b,b,b,b,b,b,b,b,b,b
取值范围	b=0: 模拟中继线模块 (缺省值); b=1: 磁石模块。
功能描述	指定 SHT 系列板卡上安装的是模拟中继线模块还是磁石模块。如果安装的为模拟中继模块, 该配置项需要设置为 0; 如果是磁石模块, 该配置项需要设置为 1。更多信息请参见第 1 章中 “ 磁石电话机通道 ” 部分内容。
注意事项	<ul style="list-style-type: none">● 格式 1 适用于 SHT 系列的 USB 语音盒;● 格式 2 适用于 8 通道规格的 SHT 系列板卡;● 格式 3 适用于 16 通道规格的 SHT 系列板卡。

3.1.2.27 SHD 系列板卡常用配置项(CTI 系列)

3.1.2.27.1 设置来话呼叫的收号规则

3.1.2.27.1.1 DefaultRcvPhoNumLen

请参见 [RcvPhoNumCfgLen](#)

3.1.2.27.1.2 DefaultRcvCallerID

请参见 [RcvPhoNumCfgLen](#)

3.1.2.27.1.3 RcvPhoNumCfgLen

配置项	DefaultRcvPhoNumLen DefaultRcvCallerID RcvPhoNumCfgLen
节	[TUP] / [ISUP] / [ISDN] / [SS1Config]
书写格式	DefaultRcvPhoNumLen=n DefaultRcvCallerID=m RcvPhoNumCfgLen=b
取值范围	n: 收号模式, n=0: 使用入局字冠方式的收号规则。入局字冠方式的收号规则由配置项 MaxPhoNumRule 、 Rule 进行设置; 1≤n≤40: 使用指定长度的收号规则, n 为被叫号码的长度, 在来话呼叫时, 如果本端收齐了 n 个被叫号码, 就认为收号过程结束。此时, 配置项 DefaultRcvCallerID 和 RcvPhoNumCfgLen 有效。 n 的缺省值为 0。 m: 是否需要接收主叫方号码, m=0: 不需要 (缺省值); m=1: 需要。 b: 被叫号码的保存模式, b=0: 保存实际接收到的全部被叫号码 (缺省值); b=1: 只保存前面的部分号码, 个数由 DefaultRcvPhoNumLen 指定, 多余部分被丢弃。
功能描述	DefaultRcvPhoNumLen 设置来话呼叫的收号规则的工作模式。 DefaultRcvCallerID 设置是否需要接收主叫方的电话号码。该配置项不支持 ISDN 协议。 RcvPhoNumCfgLen 设置驱动程序将接收到的被叫号码保存到内部缓冲区时的操作策略。

注意事项	<ul style="list-style-type: none">● 只有当配置项 DefaultRcvPhoNumLen 设置为定长方式时，配置项 DefaultRcvCallerID 才有效；● RcvPhoNumCfgLen 只适用于 TUP 通道；● 本配置项只适用于 TUP 通道、ISUP 通道、ISDN 通道、SS1 通道；● SS1 通道使用 [SS1Config] 配置节；● ISUP 通道使用 [ISUP] 配置节；● TUP 通道使用 [TUP] 配置节；● ISDN 通道使用 [ISDN] 配置节。对于 ISDN 通道，如果驱动程序工作在汇接局方式，本配置项设置的收号规则将被忽略。更多信息请参见第 1 章中“端局方式与汇接局方式”部分内容。
------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3.1.2.27.1.4 MaxPhoNumRule

请参见 [Rule](#)

3.1.2.27.1.5 Rule

配置项	MaxPhoNumRule Rule
节	[TUP] / [ISUP] / [ISDN] / [SS1Config]
书写格式	MaxPhoNumRule=N 格式 1: Rule[i] = "a...x",n,f,y,z 格式 2: Rule[i] = "a...x",n,f,y 格式 3: Rule[i] = "a...x",n,f
取值范围	N: 收号规则的个数。如果配置项的 DefaultRcvPhoNumLen 设定值大于 0，可以不设置本配置项；否则，N 必须大于 0，并且配置项 Rule 必须从 i=0 开始，重复设置 N 次。N 的缺省值为 0。 i: 收号规则的编号，从 0 开始编号， $0 \leq i < N$ 。 "a...x": 入局字冠的字符串，如“114”、“127”等。 n: 设置入局字冠的电话号码长度，n 必须大于或等于入局字冠字符串的长度。除了 SS1 通道和 ISDN 通道，只有当 y 设置为 0 时，本参数才有效。 f: 是否需要接收主叫方的电话号码， =1: 需要； =0: 不需要。 y: 电话号码的长度是否为固定值， =1: 不固定，参数 n 和 z 无效。驱动程序完成局号的接收后，不再控制收号过程，将控制权交给应用程序。当应用程序完成全部号码的接收后，可以通过调用函数 SsmSetKB 来拒绝或接受来话呼叫，并将控制权交还给应用程序； =0: 固定（缺省值），参数 n 和 z 有效。 z: 被叫号码的保存模式， z=0: 保存实际接收到的全部被叫号码（缺省值）； z=1: 只保存前面的部分号码，个数由 n 指定，多余部分被丢弃。
功能描述	MaxPhoNumRule 设置收号规则的个数。 Rule 设置具体的收号规则，格式 1 只适用于 TUP 通道，格式 2 只适用于 TUP 通道和 ISUP 通道，格式 3 适用于 SS1 通道和 ISDN 通道。

注意事项	<ul style="list-style-type: none">● 只有当配置项 DefaultRcvPhoNumLen 设置为入局字冠方式时，本配置项才有效；● 本配置项只适用于 TUP 通道、ISUP 通道、ISDN 通道、SS1 通道；● SS1 通道使用[SS1Config] 配置节；● ISUP 通道使用[ISUP] 配置节；● TUP 通道使用[TUP] 配置节；● ISDN 通道使用[ISDN] 配置节。对于 ISDN 通道，如果驱动程序工作在汇接局方式，本配置项设置的收号规则将被忽略。更多信息请参见第 1 章中“端局方式与汇接局方式”部分内容。
------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3.1.2.27.1.6 MfcLenCtrlPos

请参见 [CallerIdEnTable](#)

3.1.2.27.1.7 MfcLengthTable

请参见 [CallerIdEnTable](#)

3.1.2.27.1.8 CallerIdEnTable

配置项	MfcLenCtrlPos MfcLengthTable CallerIdEnTable
节	[SS1Config]
书写格式	MfcLenCtrlPos=n MfcLengthTable=m,m,m,m,m,m,m,m,m,m CallerIdEnTable=b,b,b,b,b,b,b,b,b,b
取值范围	n: 被叫号码长度控制位索引, n≥1; m: 被叫号码收号长度; b: 是否需要接收主叫方的电话号码, b=1: 需要; b=0: 不需要。
功能描述	MfcLenCtrlPos 设置入中继通道被叫号码收号长度控制位。 MfcLengthTable 设置入中继通道的 MFC 收号长度控制表。 CallerIdEnTable 设置入中继通道接收主叫号码的使能控制表；以接收到的被叫号码的第 MfcLenCtrlPos 位作为索引，在收号长度控制表 MfcLengthTable 中查找相应的收号长度；同样，以此作为索引在主叫号码使能控制表 CallerIdEnTable 中可以查找到该号码是否需要接收主叫号码。
注意事项	本配置项只适用于使用 SS1 协议的通道。

3.1.2.27.2 设置“自动应答来话呼叫”功能

3.1.2.27.2.1 AutoSendKB

请参见 [NetSideAutoSendAck](#)

3.1.2.27.2.2 AutoSendACM

请参见 [NetSideAutoSendAck](#)

3.1.2.27.2.3 UserSideAutoSendAck

请参见 [NetSideAutoSendAck](#)

3.1.2.27.2.4 NetSideAutoSendAck

配置项	AutoSendKB AutoSendACM UserSideAutoSendAck NetSideAutoSendAck
节	[SS1Config] / [TUP] / [ISUP] / [ISDN]
书写格式	AutoSendKB=b // 使用[SS1Config]节 AutoSendACM=b // 使用[TUP] 或 [ISUP]节 UserSideAutoSendAck=b // 使用[ISDN]节 NetSideAutoSendAck=b // 使用[ISDN]节
取值范围	b=0: 否; b=1: 是 (缺省值)。
功能描述	设置是否开启“自动应答来话呼叫”功能。 AutoSendKB 用于设置 SS1 通道, 更多信息请参见第 1 章中“ 中国 1 号信令的状态转移 ”部分内容; AutoSendACM 用于设置 ISUP 通道和 TUP 通道, 更多信息请参见第 1 章中“ ISUP 通道的状态转移 ”或“ TUP 通道的状态转移 ”部分内容; UserSideAutoSendAck 用于设置 ISDN 通道 (用户侧), NetSideAutoSendAck 用于设置 ISDN 通道 (网络侧), 更多信息请参见第 1 章中“ ISDN 通道的状态转移图 ”中的 k2 开关。
注意事项	函数 SsmEnableAutoSendKB 可以实现相同功能。

3.1.2.27.3 设置去话呼叫的主叫号码

3.1.2.27.3.1 TxCallerId

请参见 [CalloutCallerId](#)

3.1.2.27.3.2 CalloutCallerId

配置项	TxCallerId CalloutCallerId
节	[SS1Config] / [TUP] / [ISUP] / [ISDN]

书写格式	格式 1: TxCallerId=s 格式 2: TxCallerId[m]=s 格式 1: CalloutCallerId=s 格式 2: CalloutCallerId[m]=s
取值范围	<p>s: 存放主叫方电话号码的 ASCII 码形式的字符串, ISDN 通道的最大字符数为 20, TUP 通道的最大字符数为 15, 其它通道的最大字符数为 50, 只能由数字字符'0'~'9'组成, 其它类型的字符将被忽略。s 为空表示去话呼叫时不向对端交换机发送主叫号码。</p> <p>m: 为应用程序中的数字中继线的逻辑编号, 从 0 开始编号, 取值范围: 0≤m≤M (M 为配置项 TotalPcm 的设定值)。</p> <p>缺省值为空。</p>
功能描述	<p>去话呼叫时, 设置本端的电话号码, 即主叫方电话号码。</p> <p>TxCallerId 用于设置 SS1 通道 (使用[SS1Config]配置节); CalloutCallerId 适用于 ISUP 通道 (使用[ISUP]配置节)、TUP 通道 (使用[TUP]配置节)、ISDN 通道 (使用[ISDN]配置节)。格式 1 用于设置全局通道, 格式 2 用于设置指定 pcm 编号的所属通道。</p>
注意事项	<ul style="list-style-type: none"> 对于 TUP 协议和 ISUP 协议, 某些交换机要求主叫方发送的主叫号码字符串中应该包含 ST 信号 (脉冲终了信号)。配置项 SetSTSignal 可以用来设置此项功能; 如果格式 1 和格式 2 两者同时存在, 则格式 2 优先于格式 1。

3.1.2.27.3.3 SetSTSignal

配置项	SetSTSignal
节	[TUP] / [ISUP]
书写格式	SetSTSignal=b
取值范围	<p>b=0: 否 (缺省值);</p> <p>b=1: 是。</p>
功能描述	去话呼叫时, 设置本端向对端交换机发送的主叫号码字符串中是否包含 ST 信号 (脉冲终了信号)。
注意事项	本配置项只适用于使用 TUP 协议和 ISUP 协议的通道, ISUP 通道使用 [ISUP]配置节, TUP 通道使用[TUP] 配置节。

3.1.2.27.4 设置去话呼叫的被叫号码

3.1.2.27.4.1 SetCalledSTSignal

配置项	SetCalledSTSignal
节	[TUP] / [ISUP]
书写格式	SetCalledSTSignal=b

取值范围	b=0: 否(缺省值); b=1: 是。
功能描述	去话呼叫时, 设置本端向对端交换机发送的被叫号码字符串中是否包含 ST 信号(脉冲终了信号)。
注意事项	本配置项只适用于使用 TUP 协议和 ISUP 协议的通道, ISUP 通道使用 [ISUP] 配置节, TUP 通道使用[TUP] 配置节。

3.1.2.27.5 连接到远端坐席接入模块

3.1.2.27.5.1 UsageMode

配置项	UsageMode
节	[BoardId=x]
书写格式	UsageMode=b
取值范围	b=0: 否(缺省值); b=1: 是。
功能描述	设置是否将 SHD 系列板卡连接到远端坐席接入模块(Channel bank)。详细信息请参见第 1 章中“ 连接到远端坐席接入模块 ”部分内容。

	<p>本配置项只适用于 SHD 系列的下列型号板卡：</p> <ul style="list-style-type: none">✓ SHD-30A-CT/PCI/SS1✓ SHD-30A-CT/PCI/ISDN✓ SHD-30A-CT/PCI/SS7✓ SHD-30A-CT/cPCI✓ SHD-30B-CT/PCI/FAX✓ SHD-60A-CT/PCI/SS1✓ SHD-60A-CT/PCI/ISDN✓ SHD-60A-CT/PCI/SS7✓ SHD-60A-CT/cPCI✓ SHD-60B-CT/PCI/SS7/FAX✓ SHD-60B-CT/cPCI/FAX✓ SHD-120A-CT/PCI/SS1✓ SHD-120A-CT/PCI/ISDN✓ SHD-120A-CT/PCI/SS7✓ SHD-120A-CT/cPCI✓ SHD-120D-CT/PCI/CAS✓ SHD-240D-CT/PCI/CAS✓ SHD-30E-CT/PCIe✓ SHD-30E-CT/PCIe/FAX✓ SHD-30E-CT/PCIe/EC✓ SHD-60E-CT/PCIe✓ SHD-60E-CT/PCIe/FAX✓ SHD-60E-CT/PCIe/EC✓ SHD-120E-CT/PCIe
注意事项	

	<ul style="list-style-type: none"> ✓ SHD-120E-CT/PCIe/FAX ✓ SHD-120E-CT/PCIe/EC ✓ SHD-240E-CT/PCIe ✓ SHD-240E-CT/PCIe/FAX ✓ SHD-240E-CT/PCIe/EC ✓ SHD-240E-CT/PCIe/VAR ✓ SHD-30E-CT/PCI(SSW) ✓ SHD-30E-CT/PCI/FAX(SSW) ✓ SHD-30E-CT/PCI/EC(SSW) ✓ SHD-60E-CT/PCI(SSW) ✓ SHD-60E-CT/PCI/FAX(SSW) ✓ SHD-60E-CT/PCI/EC(SSW) ✓ SHD-120E-CT/PCI(SSW) ✓ SHD-120E-CT/PCI/FAX(SSW) ✓ SHD-120E-CT/PCI/EC(SSW) ✓ SHD-240E-CT/PCI(SSW) ✓ SHD-240E-CT/PCI/FAX(SSW) ✓ SHD-240E-CT/PCI/EC(SSW)
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3.1.2.27.5.2 CBProtocolType

配置项	CBProtocolType
节	[BoardId=x]
书写格式	CBProtocolType=m
取值范围	m=0: CCS 方式; m=1: CAS 方式 (缺省值)。
功能描述	设置将 SHD 系列板卡连接到远端坐席接入模块时使用的信令方式。
注意事项	本配置项只有在配置项 UsageMode 设置为 1 时有效。

3.1.2.27.5.3 CBChannelType

配置项	CBChannelType
节	[BoardId=x]

书写格式	CBChannelType=m
取值范围	m=0: 不使用此通道; m=1: 坐席通道 (缺省值); m=2: E/M 通道。
功能描述	设置将 SHD 系列板卡连接到远端坐席接入模块时的通道类型。
注意事项	本配置项只有在配置项 UsageMode 设置为 1 时有效。

3.1.2.27.5.4 CBChangeChannelType

配置项	CBChangeChannelType
节	[BoardId=x]
书写格式	CBChangeChannelType=m
取值范围	m=0: 不使用此功能 (缺省值); m=1: 通道类型初始化失败时强制设置通道类型为坐席通道。
功能描述	一般大容量盒子间通过光纤线路对接, 初始化通道类型为-1 时, 可以使用本配置项将 SHD 系列板卡连接到远端坐席接入模块时的通道类型强制设置为坐席通道。
注意事项	本配置项只有在配置项 UsageMode 设置为 1 时有效。

3.1.2.27.6 设置板卡工作模式

3.1.2.27.6.1 RunAsSpy

配置项	RunAsSpy
节	[BoardId=x]
书写格式	RunAsSpy =b
取值范围	b=0: 正常模式 (缺省值); b=1: 监控模式。
功能描述	设置 SHD 系列板卡的工作模式。
注意事项	<ul style="list-style-type: none"> ● 对于信令监控卡, 始终处于监控模式, 本配置对它无效; ● 本配置项只支持 A 型和 C 型数字卡。

3.1.2.27.7 SS1 信令的高级设置

3.1.2.27.7.1 选择国家

3.1.2.27.7.1.1 mfcr2_Protocol

配置项	mfcr2_Protocol
节	[SS1Config]
书写格式	mfcr2_Protocol=s

	s 为字符串，其含义如下表所示：																																							
	<table border="1"> <thead> <tr> <th>s 的取值</th><th>支持的国家/协议</th></tr> </thead> <tbody> <tr><td>al</td><td>阿尔及利亚 ALGERIA</td></tr> <tr><td>ar</td><td>阿根廷 ARGENTINA</td></tr> <tr><td>bh</td><td>巴林 BAHRAIN</td></tr> <tr><td>bo</td><td>玻利维亚 BOLIVIA</td></tr> <tr><td>br</td><td>巴西 BRAZIL</td></tr> <tr><td>cl</td><td>智利 CHILE</td></tr> <tr><td>cn</td><td>中国 CHINA</td></tr> <tr><td>co-land</td><td>哥伦比亚陆地 COLOMBIA_LAND</td></tr> <tr><td>co-cell</td><td>哥伦比亚蜂窝 COLOMBIA_CELL</td></tr> <tr><td>cz</td><td>捷克 CZECH</td></tr> <tr><td>cd</td><td>刚果民主 DEMO_CONGO</td></tr> <tr> <td rowspan="14">取值范围</td><td>eg 埃及 EGYPT</td></tr> <tr> <td>gn 加纳 GHANA</td></tr> <tr> <td>hd 洪都拉斯 HONDURAS</td></tr> <tr> <td>in 印度 INDIA</td></tr> <tr> <td>id 印度尼西亚 INDONESIA</td></tr> <tr> <td>iq 伊拉克 IRAQ</td></tr> <tr> <td>pa 巴拿马 PANAMA</td></tr> <tr> <td>ph 菲律宾 PHILIPINNES</td></tr> <tr> <td>ro 罗马尼亚 ROMANIA</td></tr> <tr> <td>sa 沙特阿拉伯 SAUDI_ARABIA</td></tr> <tr> <td>sg 新加坡 SINGAPORE</td></tr> <tr> <td>th 泰国 THAILAND</td></tr> <tr> <td>ve 委内瑞拉 VENEZUELA</td></tr> <tr> <td>s 的缺省值为 cn。</td></tr> </tbody> </table>	s 的取值	支持的国家/协议	al	阿尔及利亚 ALGERIA	ar	阿根廷 ARGENTINA	bh	巴林 BAHRAIN	bo	玻利维亚 BOLIVIA	br	巴西 BRAZIL	cl	智利 CHILE	cn	中国 CHINA	co-land	哥伦比亚陆地 COLOMBIA_LAND	co-cell	哥伦比亚蜂窝 COLOMBIA_CELL	cz	捷克 CZECH	cd	刚果民主 DEMO_CONGO	取值范围	eg 埃及 EGYPT	gn 加纳 GHANA	hd 洪都拉斯 HONDURAS	in 印度 INDIA	id 印度尼西亚 INDONESIA	iq 伊拉克 IRAQ	pa 巴拿马 PANAMA	ph 菲律宾 PHILIPINNES	ro 罗马尼亚 ROMANIA	sa 沙特阿拉伯 SAUDI_ARABIA	sg 新加坡 SINGAPORE	th 泰国 THAILAND	ve 委内瑞拉 VENEZUELA	s 的缺省值为 cn。
s 的取值	支持的国家/协议																																							
al	阿尔及利亚 ALGERIA																																							
ar	阿根廷 ARGENTINA																																							
bh	巴林 BAHRAIN																																							
bo	玻利维亚 BOLIVIA																																							
br	巴西 BRAZIL																																							
cl	智利 CHILE																																							
cn	中国 CHINA																																							
co-land	哥伦比亚陆地 COLOMBIA_LAND																																							
co-cell	哥伦比亚蜂窝 COLOMBIA_CELL																																							
cz	捷克 CZECH																																							
cd	刚果民主 DEMO_CONGO																																							
取值范围	eg 埃及 EGYPT																																							
	gn 加纳 GHANA																																							
	hd 洪都拉斯 HONDURAS																																							
	in 印度 INDIA																																							
	id 印度尼西亚 INDONESIA																																							
	iq 伊拉克 IRAQ																																							
	pa 巴拿马 PANAMA																																							
	ph 菲律宾 PHILIPINNES																																							
	ro 罗马尼亚 ROMANIA																																							
	sa 沙特阿拉伯 SAUDI_ARABIA																																							
	sg 新加坡 SINGAPORE																																							
	th 泰国 THAILAND																																							
	ve 委内瑞拉 VENEZUELA																																							
	s 的缺省值为 cn。																																							

功能描述	设置使用 No.1 信令的国家。
注意事项	<ul style="list-style-type: none">● 只有当配置项 ProtocolType 设置为 0 时，本配置项才有效；● 本配置项也适用于 DTP 系列板卡的 SS1 监控通道。

3.1.2.27.7.2 设置接续过程的 R2 参数

当配置项 `mfcr2_Protocol` 不为 `cn`（中国 CHINA）时，可以用以下配置项来定义接续过程的 R2 参数来扩充 SS1 R2MFC 协议变体。

3.1.2.27.7.2.1 tonesgroupA

配置项	tonesgroupA
节	[SS1Config]
书写格式	<code>tonesgroupA=k</code>
取值范围	<code>k</code> : 取值 16bit，每四个 bit（一个 16 进制位）代表一种请求，从低位到高位定义如下： <ol style="list-style-type: none">1. 发送下一位被叫(bit0-3);2. 发送 I 组主叫用户类别(bit4-7);3. 发送下一位主叫(bit8-11);4. 发送 II 组信号且转至 B 组信号接收(bit12-15)。
功能描述	设置后向 A 组信号，驱动使用这些参数在 MFC 互控中发送后向 A 组请求。
注意事项	本配置项只适用于 SHD 系列板卡的 SS1 通道。
使用范例	若 R2MFC 协议定义：A1 发送下一位被叫，A5 发送 I 组主叫用户类别，A5 发送下一位主叫，A3 发送 II 组信号且转至 B 组信号接收，则设置 <code>tonesgroupA=0x3551</code> 。

3.1.2.27.7.2.2 tonesgroupB

配置项	tonesgroupB
节	[SS1Config]
书写格式	<code>tonesgroupB=k</code>
取值范围	<code>k</code> : 取值 16bit，每四个 bit（一个 16 进制位）代表一种指示。从低位到高位定义如下： <ol style="list-style-type: none">1. 拥塞(bit0-3);2. 空号(bit4-7);3. 忙(bit8-11);4. 故障(bit12-15)。
功能描述	设置后向信号结束 MFC 互控指示。
注意事项	本配置项只适用于 SHD 系列板卡的 SS1 通道。

3.1.2.27.7.2.3 Tonesendofinfo

配置项	Tonesendofinfo
节	[SS1Config]
书写格式	tonesendofinfo=k
取值范围	k: 取值 16bit, 每四个 bit (一个 16 进制位) 从低位到高位定义如下: 1. 被叫结束信号(bit0-3)。若协议不存在此信号, 用 0 表示; 2. 保留备用(bit4-7); 3. 主叫可用, 主叫结束信号(bit8-11); 4. 限制提供主叫, 主叫结束信号(bit12-15)。若协议不存在此信号, 用 0 表示。
功能描述	设置前向信号指示某类信号结束或不可用。
注意事项	本配置项只适用于 SHD 系列板卡的 SS1 通道。

3.1.2.27.7.2.4 Tonesanswer

配置项	Tonesanswer
节	[SS1Config]
书写格式	tonesanswer=k
取值范围	k: 取值 16bit, 每四个 bit (一个 16 进制位) 从低位到高位定义如下: 1. B 组接受呼叫-计费; 2. B 组接受呼叫-免费; 3. A 组接受呼叫; 4. 备用。
功能描述	设置后向信号指示接收呼叫。
注意事项	本配置项只适用于 SHD 系列板卡的 SS1 通道。

3.1.2.27.7.2.5 Tonesrepeatrequest

配置项	Tonesrepeatrequest
节	[SS1Config]
书写格式	tonesrepeatrequest=k

取值范围	k: 取值 16bit, 每四个 bit (一个 16 进制位) 从低位到高位定义如下: 1. 由前一位发起; 2. 由前二位发起; 3. 由前三位发起; 4. 由第一位发起。
功能描述	设置后向 A 组信号, 呼入端请求呼出端重发。
注意事项	本配置项只适用于 SHD 系列板卡的 SS1 通道。

3.1.2.27.7.2.6 TonesAnswerA

配置项	TonesAnswerA
节	[SS1Config]
书写格式	TonesAnswerA=k
取值范围	k: 取值 16bit, 每四个 bit (一个 16 进制位) 从低位到高位定义如下: 1. A 组网络忙; 2. A 组未定义的号码; 3. B 组特殊信号音; 4. II 组 xxx。
功能描述	设置 R2 信令部分参数。
注意事项	本配置项只适用于 SHD 系列板卡的 SS1 通道。

3.1.2.27.7.3 设置接续过程的基本工作参数

3.1.2.27.7.3.1 TxCas_CD

配置项	TxCas_CD
节	[SS1Config]
书写格式	TxCas_CD=k
取值范围	k: 高 6 位比特保留, 应设置为 0; 低 2 比特为 C/D 位码: Bit1: C 的信令码, 缺省值为 1; Bit0: D 的信令码, 缺省值为 1。
功能描述	设置本端向对端交换机发送 ABCD 信令码时, CD 信令码的值。
注意事项	本配置项只适用于 SHD 系列板卡的 SS1 通道, 只有当配置项 ProtocolType 为 1、2、3、4、5 时有效。

3.1.2.27.7.3.2 RxCASFilterTime

配置项	RxCASFilterTime
节	[SS1Config]
书写格式	RxCASFilterTime=t
取值范围	t: 对端交换机送出的 ABCD 信令的最小保持时间, 单位为毫秒, 必须为 8 的整数倍, 缺省值为 0。
功能描述	设置对端交换机送出的 ABCD 信令的最小保持时间。只有当线路上 ABCD 信令码发生变化, 并且新值的保持时间超过本配置项的设定值, 驱动程序才会确认 ABCD 信令码的变化, 否则认为是干扰抖动信号。
注意事项	<ul style="list-style-type: none">● 本配置项只适用于 SHD 系列板卡的 SS1 通道;● 本参数适用于信号质量较差的 E1/T1 线路, 当信令抖动非常严重时, 往往会导致驱动程序误判远端挂机, 本配置项的功能是设定一个最小保持时间以消除抖动。注意: 对于去话呼叫, 本配置项的设定值如果过大, 可能会有副作用, 即当被叫发送“被叫应答”信令(即 A_b/B_b=0/1)后, 又立即发送“被叫先挂机”信令(即 A_b/B_b=1/1), 则“被叫应答”信令有可能被滤波器当成抖动信号而过滤掉。

3.1.2.27.7.3.3 MaxWaitMfcTime

配置项	MaxWaitMfcTime
节	[SS1Config]
书写格式	MaxWaitMfcTime=t
取值范围	t: 最大等待时间, 单位为秒, 缺省值为 10 秒。
功能描述	设置 SS1 状态机的 T2 定时器。 有关 T2 的更多信息请参见第 1 章中“ 中国 1 号信令的状态转移 ”部分内容。
注意事项	本配置项只适用于 SHD 系列板卡的 SS1 通道。

3.1.2.27.7.3.4 RxR2FilterTime

配置项	RxR2FilterTime
节	[BoardId=x]
书写格式	RxR2FilterTime=t
取值范围	t: R2 信号的最小保持时间, 单位为毫秒, 取值范围: 16~96, 必须为 16 的整数倍, 缺省值为 16。
功能描述	设置 MFC R2 信号的最小保持时间。
注意事项	<ul style="list-style-type: none">● 本配置项只适用于 SHD 系列板卡的 SS1 通道;● 本配置项实现的功能也可以通过调用函数 SsmSetFlag (携带 F_RXR2FILTERNODE 参数) 实现。

3.1.2.27.7.3.5 LSRingTimeout

配置项	LSRingTimeout
节	[SS1Config]
书写格式	LSRingTimeout =t
取值范围	t: 单位为毫秒, 取值范围: 1000~20000, 缺省值为 6000。
功能描述	设置振铃状态的超时时间。当处于振铃的通道在此时间内未收到任何接续消息, 通道将进入“等待释放”状态。
注意事项	<ul style="list-style-type: none">● 本配置项只适用于 SHD 系列板卡的 SS1 通道;● 本配置项只对 SS1 处于 LineSide (ProtocolType=1) 模式下有效。

3.1.2.27.7.4 设置状态机中通道的工作方式

3.1.2.27.7.4.1 EnableAutoCall

配置项	EnableAutoCall
节	[BoardId=x]
书写格式	EnableAutoCall[n]=b
取值范围	n: 数字中继线的板卡内部编号, 0 ≤ n ≤ M-1, (M 为板卡的数字中继线总数); b: 是否使用 SynCTI 驱动程序提供的状态机, b=1: 是 (缺省值); b=0: 否。
功能描述	设置 SS1 通道状态机的工作状态。
注意事项	本配置项只适用于 SHD 系列的 SS1 通道。

3.1.2.27.7.4.2 AutoCallInTimeSlot

配置项	AutoCallInTimeSlot
节	[BoardId=x]
书写格式	AutoCallInTimeSlot[n]=i,k
取值范围	n: 数字中继线的板卡内部编号, 0 ≤ n ≤ M-1, (M 为板卡的数字中继线总数)。 i: 设置为来话方向的起始时隙。 k: 设置为来话方向的通道 (时隙) 数量。
功能描述	将第 n 条 PCM 上从 i 时隙开始的 k 个时隙设置为来话方向, 其余时隙设置为去话方向。
注意事项	本配置项只适用于 SHD 系列的 SS1 通道。

3.1.2.27.7.5 设置中国 1 号信令状态机的参数

3.1.2.27.7.5.1 MfcKB

配置项	MfcKB
节	[SS1Config]
书写格式	MfcKB=k
取值范围	1≤k≤6，缺省值随着配置项 mfcr2_Protocol 的不同而不同：沙特阿拉伯、墨西哥、中国、泰国、巴西缺省值为 1，科威特缺省值为 5；其余国家为 6；当 k 的取值为缺省值时，表示被叫用户空闲，接受来话呼叫。有关 KB 取值的物理含义请参见函数 SsmSetKB 的说明以及相关国家 SS1 协议。
功能描述	设置当 SS1 通道自动应答来话呼叫时向对端交换机发送的 KB 信号值。 详细内容请参见第 1 章中“ 中国 1 号信令的状态转移 ”部分内容。
注意事项	本配置项只有当配置项 AutoSendKB 设置为 1 时才有效。

3.1.2.27.7.5.2 MaxWaitSetKBTime

配置项	MaxWaitSetKBTime
节	[SS1Config]
书写格式	MaxWaitSetKBTime=t
取值范围	t：最大等待时间，单位为秒，缺省值为 3 秒。
功能描述	设置等候应用程序设置 KB 信号的最大时间。 设置 SS1 通道状态机中应用程序应答本次来话呼叫请求的最大等待时间，即 T4 定时器。有关 T4 的更多信息请参见第 1 章中“ 中国 1 号信令的状态转移 ”部分内容。

3.1.2.27.7.5.3 MaxWaitKDTime

配置项	MaxWaitKDTime
节	[SS1Config]
书写格式	MaxWaitKDTime=t
取值范围	t：最大等待时间，单位为秒，缺省值为 60 秒。
功能描述	设置 SS1 通道状态机中等待对端交换机发送 KD 信号的最大等待时间，即 T3 定时器。有关 T3 的更多信息请参见第 1 章中“ 中国 1 号信令的状态转移 ”部分内容。

3.1.2.27.7.5.4 PcmSyncMask

配置项	PcmSyncMask

节	[SS1Config]
书写格式	PcmSyncMask=x
取值范围	x: 线路同步值掩码, 缺省值为 0x206。
功能描述	设置是否屏蔽 E1 线上的 PCM 链路同步状态, PcmSyncMask 为 0x0206 时, 表示将忽略除 1、2、9bit 位以外的 PCM 链路工作状态。更多信息请参考 SsmGetPcmLinkStatus 。

3.1.2.27.7.5.5 PhoNumHoldup

请参见 [A3pTime](#)

3.1.2.27.7.5.6 A1ToA3pWaitTime

请参见 [A3pTime](#)

3.1.2.27.7.5.7 A3pTime

配置项	PhoNumHoldup A1ToA3pWaitTime A3pTime
节	[SS1Config]
书写格式	PhoNumHoldup=m A1ToA3pWaitTime=t _{wait} A3pTime=t _{keep}

	<p>m: “被叫号码拦截”功能的控制开关，其取值及含义如下表所示：</p> <table border="1"> <thead> <tr> <th>m</th><th>描述</th></tr> </thead> <tbody> <tr> <td>0</td><td>关闭“被叫号码拦截”功能</td></tr> <tr> <td>1</td><td> <p>开启“被叫号码拦截”功能，只拦截1位被叫号码。当驱动程序按照设定的收号规则收到最后一个被叫号码后，不是向对端交换机发送互控方式的A3信号，而是发送A1信号，继续等待后续被叫号码。</p> <ul style="list-style-type: none"> ◆ 如果在 t_{wait} 时间内收到了后续号码，驱动程序将该被叫号码存入被叫号码缓冲区，接收被叫号码的操作完成，驱动程序恢复向对端交换机发送互控方式的A3信号，并恢复正常接续过程； ◆ 如果超过 t_{wait} 时间后仍未收到后续被叫号码，驱动程序向对端交换机发送脉冲方式的A3信号，并恢复正常接续过程。 </td></tr> <tr> <td>2</td><td> <p>开启“被叫号码拦截”功能，拦截多位被叫号码。当驱动程序按照设定的收号规则收到最后一个被叫号码后，继续发送A1信号，等待后续被叫号码。如果在 t_{wait} 内收到了后续号码，驱动程序将该被叫号码存入被叫号码缓冲区，并重复上述过程。直到超过 t_{wait} 后仍未收到下一个被叫号码，驱动程序才向对端交换机发送脉冲方式的A3信号，恢复正常接续过程。</p> </td></tr> </tbody> </table> <p>m 的缺省值为 0。</p> <p>t_{wait}: 最大等待时间，单位为毫秒，缺省值为 1000, m>0 时有效。</p> <p>t_{keep}: 脉冲宽度，单位为毫秒，缺省值为 150, m>0 时有效。</p>	m	描述	0	关闭“被叫号码拦截”功能	1	<p>开启“被叫号码拦截”功能，只拦截1位被叫号码。当驱动程序按照设定的收号规则收到最后一个被叫号码后，不是向对端交换机发送互控方式的A3信号，而是发送A1信号，继续等待后续被叫号码。</p> <ul style="list-style-type: none"> ◆ 如果在 t_{wait} 时间内收到了后续号码，驱动程序将该被叫号码存入被叫号码缓冲区，接收被叫号码的操作完成，驱动程序恢复向对端交换机发送互控方式的A3信号，并恢复正常接续过程； ◆ 如果超过 t_{wait} 时间后仍未收到后续被叫号码，驱动程序向对端交换机发送脉冲方式的A3信号，并恢复正常接续过程。 	2	<p>开启“被叫号码拦截”功能，拦截多位被叫号码。当驱动程序按照设定的收号规则收到最后一个被叫号码后，继续发送A1信号，等待后续被叫号码。如果在 t_{wait} 内收到了后续号码，驱动程序将该被叫号码存入被叫号码缓冲区，并重复上述过程。直到超过 t_{wait} 后仍未收到下一个被叫号码，驱动程序才向对端交换机发送脉冲方式的A3信号，恢复正常接续过程。</p>
m	描述								
0	关闭“被叫号码拦截”功能								
1	<p>开启“被叫号码拦截”功能，只拦截1位被叫号码。当驱动程序按照设定的收号规则收到最后一个被叫号码后，不是向对端交换机发送互控方式的A3信号，而是发送A1信号，继续等待后续被叫号码。</p> <ul style="list-style-type: none"> ◆ 如果在 t_{wait} 时间内收到了后续号码，驱动程序将该被叫号码存入被叫号码缓冲区，接收被叫号码的操作完成，驱动程序恢复向对端交换机发送互控方式的A3信号，并恢复正常接续过程； ◆ 如果超过 t_{wait} 时间后仍未收到后续被叫号码，驱动程序向对端交换机发送脉冲方式的A3信号，并恢复正常接续过程。 								
2	<p>开启“被叫号码拦截”功能，拦截多位被叫号码。当驱动程序按照设定的收号规则收到最后一个被叫号码后，继续发送A1信号，等待后续被叫号码。如果在 t_{wait} 内收到了后续号码，驱动程序将该被叫号码存入被叫号码缓冲区，并重复上述过程。直到超过 t_{wait} 后仍未收到下一个被叫号码，驱动程序才向对端交换机发送脉冲方式的A3信号，恢复正常接续过程。</p>								
功能描述	设置“被叫号码拦截”功能。 <code>PhoNumHoldup</code> 设置“被叫号码拦截”功能的控制开关， <code>A1ToA3pWaitTime</code> 设置等待后续被叫号码的时间门限， <code>A3pTime</code> 设置脉冲形式的A3信号的脉冲宽度。								
注意事项	PhoNumHoldup 的功能也可以通过调用函数 SsmSetFlag （携带 F_RCVPHONUMHOLDUP 参数）实现。								

3.1.2.27.7.5.8 设置状态机中去话呼叫的参数

3.1.2.27.7.5.8.1 MaxWaitOccupyAckTime

配置项	MaxWaitOccupyAckTime
节	[SS1Config]
书写格式	MaxWaitOccupyAckTime=t
取值范围	$t \geq 1$, 单位为秒, 缺省值为 60。
功能描述	设置 T5 定时器的值。 有关 T5 定时器的详细信息请参见第 1 章中“ 中国 1 号信令的状态迁移 ”部分内容。

3.1.2.27.7.5.8.2 MfcKD

配置项	MfcKD
节	[SS1Config]

书写格式	MfcKD=k
取值范围	1≤k≤6，缺省值为3（市内电话）。有关KD信号的详细说明请参见函数 SsmSetKD 的说明。
功能描述	设置去话呼叫时的发端业务类别，即KD信号。

3.1.2.27.7.5.8.3 MfcKA

配置项	MfcKA
节	[SS1Config]
书写格式	MfcKA=k
取值范围	1≤k≤10，缺省值为1（普通/定期）。有关KA信号的详细说明请参见函数 SsmSetKA 的说明。
功能描述	设置去话呼叫时KA信号（本端的主叫类别）。

3.1.2.27.7.5.8.4 MaxWaitKBTime

配置项	MaxWaitKBTime
节	[SS1Config]
书写格式	MaxWaitKBTime=t
取值范围	t：最大等待时间，单位为秒，缺省值为60秒。
功能描述	设置等待对端交换机发送KB信号的最大等待时间。

3.1.2.27.7.5.9 与 Dialogic 的 SS1 通道对接

3.1.2.27.7.5.9.1 ToRingingDelayTime

配置项	ToRingingDelayTime
节	[BoardId=x]
书写格式	ToRingingDelayTime=t
取值范围	t：延迟时间，单位为毫秒，缺省值为0。
功能描述	设置延迟时间。在 Synway 板卡与 Dialogic 板卡的 SS1 通道对联时，如果 Synway 板卡的 SS1 通道作为呼入端，当完成全部被叫号码的接收后，需要在通道进入振铃状态之前插入一段等待时间。
注意事项	本配置项仅在将 Synway 的 SHD 系列板卡与 Dialogic 的板卡互联时使用。

3.1.2.27.7.5.9.2 RepeatPhoNumOn1stR2bwdsA5

配置项	RepeatPhoNumOn1stR2bwdsA5
节	[BoardId=x]
书写格式	RepeatPhoNumOn1stR2bwdsA5=m
取值范围	m=0: 驱动程序设置挂起原因为 SS1OUT_BWD_A5, 将通道迁移到 S_CALL_PENDING 状态; m=1: 如果本端只发送了第 1 位被叫号码就收到对端的 A5 信号, 则重新发送第 1 位号码; 否则, 驱动程序设置挂起原因为 SS1OUT_BWD_A5, 将通道迁移到 S_CALL_PENDING 状态。 缺省值为 1。
功能描述	去话呼叫时, 如果收到对端的 A5 信号(空号), 本配置项选择驱动程序的后续动作。
注意事项	本配置项仅在将 Synway 的 SHD 系列板卡与 Dialogic 的板卡互联时使用。

3.1.2.27.7.5.10 应用程序退出时是否向闭塞对端

3.1.2.27.7.5.10.1 IsBlockSS1In

配置项	IsBlockSS1In
节	[SS1Config]
书写格式	IsBlockSS1In=b
取值范围	b=1: 发送(缺省值); b=0: 不发送。
功能描述	当应用程序退出时, 本配置项用于设置是否由驱动程序自动向对端交换机发送闭塞命令, 以通知对端交换机不要再向本端发起呼叫。
注意事项	本配置项只适用于 SHD 系列板卡的 SS1 通道。

3.1.2.27.7.5.11 输出调试信息

3.1.2.27.7.5.11.1 MfcR2ToRxCallerIdBuf

配置项	MfcR2ToRxCallerIdBuf
节	[SS1Config]
书写格式	MfcR2ToRxCallerIdBuf=b

取值范围	b=0: 否(缺省值); b=1: 是。
功能描述	设置是否将呼叫过程中对端交换机发送的 R2 信号保存到 Caller ID 扩展缓冲区中, 以便于对呼叫过程进行观察。如果开启此功能, 每当对端交换机发送的 R2 信号发生变化, 包括 R2 信号出现、R2 信号消失, 驱动程序都会将此事件放入 RxCallerIdExBuf 缓冲区中, 同时向应用程序抛出 E_CHG_CIDExBuf 事件。 函数 SsmGetCallerIdEx 用于取出该缓冲区中的字符串。

3.1.2.27.7.6 设置 LineSide 协议的参数

3.1.2.27.7.6.1 LSWaitPickupTime

配置项	LSWaitPickupTime
节	[SS1Config]
书写格式	LSWaitPickupTime=t
取值范围	16≤t≤10000, 单位为毫秒, 必须为 8 的整数倍, 缺省值为 96。如果设置的值超过取值范围, 驱动程序自动将其修改为 96 毫秒。
功能描述	设置 LineSide 协议的状态机中的 T1 定时器。有关 T1 定时器的详细信息请参见第 1 章中 “ Line side 协议的状态机描述 ” 部分内容。
注意事项	本配置项不适用于 1 号信令, 当配置项 ProtocolType 设置为 1、2、3、4 时有效。

3.1.2.27.7.6.2 Ss1SendIdleState

配置项	Ss1SendIdleState
节	[SS1Config]
书写格式	Ss1SendIdleState=m
取值范围	当配置项 ProtocolType 设置为 1、3 时: m=0: 使用 CAS 00xx; m=1: 使用 CAS 01xx; (阿尔卡特交换机使用本值) 当配置项 protocolType 设置为 2 时: m=0: 使用 CAS 00xx; m=1: 使用 CAS 01xx; m=2: 使用 CAS CD01; 其中 CD 位的值为配置项 TxCas_CD 设定值。
功能描述	设置向对端交换机发送空闲指示标志时使用的 ABCD 位码值。
注意事项	本配置项适用于使用 Line side 协议的 SS1 通道、ASB 协议的通道和 ASBEL7 协议的通道, 仅当配置项 ProtocolType 设置为 1、2、3、5 时有效。

3.1.2.27.7.6.3 LSTxCas

配置项	LSTxCas
节	[SS1Config]
书写格式	LSTxCas=0xabcd
取值范围	=0: 本配置项不起作用; =其他值: 每个数字代表 CAS 的 ab 位。从低到高依次代表: d (空闲时发送的 ab 位); c (摘机时发送的 ab 位); b (闪断开始时的 ab 位); a (闪断结束后的 ab 位)。
功能描述	设置向对端交换机发送在空闲, 摘机, 闪断开始, 闪断结束状态时的 CAS 值的 ab 位。
注意事项	本配置项适用于使用 Line side 协议的 SS1 通道, 仅当配置项 ProtocolType 设置为 1、3 时有效。

3.1.2.27.8 SS7 信令的高级设置

3.1.2.27.8.1 设置数字中继线第 16 时隙的属性

3.1.2.27.8.1.1 UseTS16AsCircuit

请参见 [Ss7SignalingTS](#)

3.1.2.27.8.1.2 Ss7SignalingTS

配置项	UseTS16AsCircuit Ss7SignalingTS
节	[BoardId=x]
书写格式	UseTS16AsCircuit[n]=b Ss7SignalingTS[n]=k

取值范围	<p>n: 数字中继线的物理编号, 取值范围为 0~M-1, M 为配置项 PcmNumber 的设定值。</p> <p>b: 本数字中继线内是否包含 SS7 信令链路,</p> <p> b=0: 是(缺省值)。具体提供信令链路的时隙编号由配置项 Ss7SignalingTS 决定;</p> <p> b=1: 否。配置项 Ss7SignalingTS 所指定的时隙既不能用于信令, 也不能用于话路。</p> <p>k: 时隙编号, 缺省值为 16。</p> <p>1) 如果 SynCTI 驱动程序的版本低于 4.5.8.0, k 固定为 16;</p> <p>2) 如果驱动版本高于或等于 4.5.8.0 且低于 4.8.0.0, k 的取值与板卡型号有关。</p> <p> 若板卡型号为 SHD 系列中的下列型号之一:</p> <ul style="list-style-type: none">✧ SHD-240A-CT/cPCI✧ SHD-240S-CT/cPCI✧ SHD-480A-CT/cPCI✧ SHD-480S-CT/cPCI <p> k 只能设置为 16; 如果是其余型号, k 可以选择 1 或 16。</p> <p>3) 如果驱动版本高于或等于 4.8.0.0</p> <p> 若板卡型号为 SHD 系列中的下列型号之一:</p> <ul style="list-style-type: none">✧ SHD-240A-CT/cPCI✧ SHD-240S-CT/cPCI✧ SHD-480A-CT/cPCI✧ SHD-480S-CT/cPCI <p> k 只能设置为大于等于 0, 且小于等于 32; 取值等于 0 或 32 时本数字中继线 1-31 时隙全部用于话路, 其他值为信令链路时隙编号;</p> <p> 若板卡型号为 SHD 系列中的下列型号之一:</p> <ul style="list-style-type: none">✧ SHD-240D-CT/PCI✧ SHD-240D-CT/PCI/EC✧ SHD-120D-CT/PCI✧ SHD-120D-CT/PCI/EC✧ SHD-60B-CT/PCI/FJ✧ 所有 E 型系列数字卡 <p> k 只能设置为大于 0, 且小于 33。</p> <p> 具体如下:</p> <p> E1 模式下, k=32 指明该 PCM 中继 31 时隙全部走话路。如果为 SHD-60B-CT/PCI/FJ, 则表示被监控的 PCM 中继 31 时隙全部是走话路的, 该特性目前仅支持 E1、SS7。(注意: SS1 和 ISDN 默认 k 为 16, 且不可修改。)</p> <p> T1/J1 模式下, 1≤k≤25, k=25 指明该 PCM 中继 24 时隙全部走话路。(注意: SS1 默认 k 为 25; ISDN 默认 k 为 24, 且不可修改。)</p> <p> 其它 SHD 系列板卡 k 可以选择 1 或 16。</p>
------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

功能描述	UseTS16AsCircuit 设置本数字中继线内是否包含 SS7 信令链路。
注意事项	<ul style="list-style-type: none">● 本配置项为高级配置项，只适用于 SS7 信令；● 如果配置项 PcmNumber 大于 1，应为每条数字中继线对本配置项进行设置。

3.1.2.27.8.2 设置地址信号中备用值的对应字符

3.1.2.27.8.2.1 AddressSignal

配置项	AddressSignal
节	[TUP] / [ISUP]
书写格式	AddressSignal[n]=c
取值范围	n: TUP/ISUP 消息中地址信号使用的值, 10≤n≤14。 c: n 对应的编码字符, 不能为 ‘0’ ~ ‘9’中的任何一个字符; 如果 c 超过一个字符, 则 n 对应的编码字符为 c 的第一个字符。
功能描述	设置地址信号中备用值的对应字符。 更多信息请参见第 1 章中 “ 设置地址信号中备用值的对应字符 ” 部分内容。
注意事项	<ul style="list-style-type: none">● 本配置项只适用于 TUP/ISUP 通道;● ISUP 通道使用[ISUP]配置节, TUP 通道使用[TUP]配置节。● 使用范例: AddressSignal[11]=b。

3.1.2.27.8.3 设置 SS7 信令服务器的 IP 地址

3.1.2.27.8.3.1 Ss7ServerIP

请参见 [SecondServerIP](#)

3.1.2.27.8.3.2 SecondServerIP

配置项	Ss7ServerIP SecondServerIP
节	[SS7]
书写格式	Ss7ServerIP=a.b.c.d SecondServerIP=a.b.c.d
取值范围	a.b.c.d: SS7 信令服务器的 IP 地址, 缺省值为 127.0.0.1 (表示 SS7 信令服务器只有一个, 并且与应用程序一起, 都运行于本机上)。
功能描述	设置 SS7 信令服务器 IP 地址。

注意事项	如果 SS7 信令服务器程序与应用程序都在本机上运行，则无需设置这 2 个配置项。如果系统中包含了 2 个 SS7 信令服务器，则 Ss7ServerIP、SecondServerIP 分别用于设置主信令服务器、从信令服务器的 IP 地址。
使用范例	Ss7ServerIP=201.123.123.1

3.1.2.27.8.4 设置本机的 IP 地址

3.1.2.27.8.4.1 LocalIP

配置项	LocalIP
节	[SS7]
书写格式	LocalIP=a.b.c.d
取值范围	a.b.c.d: 本机的 IP 地址，缺省值为 127.0.0.1，表示应用程序与 SS7 信令服务器都在本机运行。
功能描述	设置 SS7 信令服务器或网关的 IP 地址。
使用范例	LocalIP=201.123.123.5

3.1.2.27.8.5 SS7 信令服务器输出语音信道

3.1.2.27.8.5.1 LoadShp_a3AsSIU

配置项	LoadShp_a3AsSIU
节	[SystemConfig]
书写格式	LoadShp_a3AsSIU=b
取值范围	b=0: 否(缺省值); b=1: 是。
功能描述	设置 SS7 信令服务器是否输出语音时隙。当 SS7 信令服务器上不运行任何业务系统，只是提供 SS7 信令服务时，它只处理数字中继线上的信令时隙，不处理语音时隙。如果接入到 SS7 信令服务器的数字中继线上除了信令链路外，还包含语音时隙，需要 SS7 信令服务器提供将语音时隙交换输出的功能。输出语音时隙的原理为：将物理编号为 0 的数字中继线上的全部语音时隙分别与物理编号为 1 的数字中继线上的对应语音时隙进行双向连接。更多信息请参见第 4 章中“ 为第三方板卡提供 SS7 信令服务 ”部分内容。
注意事项	只适用于 SHD-60A-CT/PCI/SS7 型号的板卡。

3.1.2.27.8.6 设置数字中继线 SS7 电路

3.1.2.27.8.6.1 Ss7CircuitMap[pcm]

配置项	Ss7CircuitMap[pcm]
节	[BoardId=x]

书写格式	<code>Ss7CircuitMap[pcm]=b</code>
取值范围	pcm: 数字中继线在板卡上的物理编号, 更多信息请参见相关手册的说明。取值范围: $0 \leq \text{pcm} \leq 31$; b: DWORD 32bit, 从 bit0-bit31 分别指示数字中继 PCM 的 TS0-TS31 是否用作 SS7 电路, 0 不用, 1 使用。在 E1 系统中 TS0 及处理 SS7 信令的时隙将被忽略。本配置项缺省值 0xFFFFFFFF。
功能描述	设置数字中继线 PCM 的 SS7 电路与 PCM 时隙的对应关系。对于设置为不用作 SS7 电路的时隙, 驱动将不对其进行复原, 并且阻断对端有关该时隙的呼入, 该时隙对应的通道状态为“不可用”。
使用范例	<code>Ss7CircuitMap[0]= 0xFFFFFFFF</code>

3.1.2.27.8.7 TUP 协议高级配置项

3.1.2.27.8.7.1 设置电路群消息的电路范围

3.1.2.27.8.7.1.1 SendGRMRange

配置项	SendGRMRange
节	[TUP]
书写格式	<code>SendGRMRange=m</code>
取值范围	<p>m=0: 使用全 0 范围编码字段。电路群消息的 CIC 字段的时隙为 1, 范围字段为 0, 表示该电路群消息覆盖了时隙 1~31;</p> <p>m=1: 使用非全 0 范围编码字段。电路群消息覆盖的时隙范围与配置项 Ss7SignalingTS 有关。假定 Ss7SignalingTS 的设定值为 N, 则:</p> <ul style="list-style-type: none"> ◇ 如果 N=1, 电路群消息的 CIC 字段的时隙为 2, 范围字段为 29; ◇ 如果 N=16, 电路群消息会被拆分成 2 条, 其中一条消息的 CIC 字段的时隙为 1, 范围字段为 14, 另外一条消息的 CIC 字段的时隙为 17, 范围字段为 14。
功能描述	当本端驱动程序向对端交换机发送电路群消息时, 本配置项设置电路群消息所覆盖的时隙范围。
注意事项	本配置项会影响所有的电路群消息, 包括电路群复原消息、电路群闭塞/解除闭塞。

3.1.2.27.8.7.1.2 HangupRingSendCBK

配置项	HangupRingSendCBK
节	[TUP]
书写格式	<code>HangupRingSendCBK=m</code>
取值范围	<p>m=0: 发送 CFL 消息 (缺省值);</p> <p>m=1: 发送 CBK 消息。</p>
功能描述	当通道处于 S_CALL_RINGING 状态时, 如果应用程序调用 SsmHangup 或 SsmHangupEx 函数, 会导致驱动程序向对端交换机发送呼叫拒绝消息。呼叫拒绝消息可以是 CBK 或 CFL 消息, 具体使用哪一个, 由本配置项决定。
注意事项	本配置项要求驱动程序的版本为 4.7.1.5 或更高。

3.1.2.27.8.7.2 来话呼叫：定制 ACM 消息

3.1.2.27.8.7.2.1 DefaultACM

配置项	DefaultACM																																																				
节	[TUP]																																																				
书写格式	DefaultACM=0xab																																																				
取值范围	<p>ab: 8 比特数据，以 16 进制表示，8 个比特从高到低表示为 HGFEDCBA:</p> <table border="1"> <thead> <tr> <th>比特</th> <th>含义</th> <th>取值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>H</td> <td>备用</td> <td>0</td> <td>任何通道</td> </tr> <tr> <td>F</td> <td>信令通道表示语</td> <td>1</td> <td>全部是 SS7 信令方式通道</td> </tr> <tr> <td>G</td> <td></td> <td>0</td> <td>呼叫不前转</td> </tr> <tr> <td>E</td> <td>呼叫前转表示语</td> <td>1</td> <td>呼叫前转</td> </tr> <tr> <td>D</td> <td>来话回声抑制表示语</td> <td>0</td> <td>未包括来话半回声抑制器</td> </tr> <tr> <td>F</td> <td></td> <td>1</td> <td>包括来话半回声抑制器</td> </tr> <tr> <td>C</td> <td>用户闲表示语</td> <td>0</td> <td>未指示</td> </tr> <tr> <td>B</td> <td></td> <td>1</td> <td>用户闲</td> </tr> <tr> <td>A</td> <td>地址全信号类型表示语</td> <td>00</td> <td>地址全信号</td> </tr> <tr> <td></td> <td></td> <td>01</td> <td>地址全信号、计费</td> </tr> <tr> <td></td> <td></td> <td>10</td> <td>地址全信号、免费</td> </tr> <tr> <td></td> <td></td> <td>11</td> <td>地址全信号、付费电话</td> </tr> </tbody> </table> <p>缺省值为 0x05。</p>	比特	含义	取值	描述	H	备用	0	任何通道	F	信令通道表示语	1	全部是 SS7 信令方式通道	G		0	呼叫不前转	E	呼叫前转表示语	1	呼叫前转	D	来话回声抑制表示语	0	未包括来话半回声抑制器	F		1	包括来话半回声抑制器	C	用户闲表示语	0	未指示	B		1	用户闲	A	地址全信号类型表示语	00	地址全信号			01	地址全信号、计费			10	地址全信号、免费			11	地址全信号、付费电话
比特	含义	取值	描述																																																		
H	备用	0	任何通道																																																		
F	信令通道表示语	1	全部是 SS7 信令方式通道																																																		
G		0	呼叫不前转																																																		
E	呼叫前转表示语	1	呼叫前转																																																		
D	来话回声抑制表示语	0	未包括来话半回声抑制器																																																		
F		1	包括来话半回声抑制器																																																		
C	用户闲表示语	0	未指示																																																		
B		1	用户闲																																																		
A	地址全信号类型表示语	00	地址全信号																																																		
		01	地址全信号、计费																																																		
		10	地址全信号、免费																																																		
		11	地址全信号、付费电话																																																		
功能描述	来话呼叫时，当本端完成被叫号码等信息的接收后，需要向对端交换机发送 ACM 消息，在 ACM 消息中应该包含指示被叫用户状态的消息表示语字段，本配置项用于设置此字段。更多信息请参加第 1 章中“ TUP 通道的状态转移 ”部分内容。																																																				
使用范例	DefaultACM=0x05																																																				

3.1.2.27.8.7.3 来话呼叫：定制 GRQ 消息

3.1.2.27.8.7.3.1 ReqTypeIndicators

配置项	ReqTypeIndicators
节	[TUP]
书写格式	ReqTypeIndicators=0xk

取值范围	k: 8 比特数据，以 16 进制表示，8 个比特从高到低表示为 HGFEDCBA:						
	比特	含义	取值	描述			
	H	备用					
	G	回声抑制器请求表示语	0	未请求			
	F	(也适用于回声消除器)	1	请求			
	E	请求保持表示语	0	未请求			
			1	请求			
	D	恶意呼叫识别表示语	0	未遇到			
			1	遇到			
	C	原被叫地址表示语	0	未请求			
			1	请求			
	B	主叫用户线标识表示语	0	未请求			
			1	请求			
	A	主叫用户类别表示语	0	未请求			
			1	请求			
缺省值为 0x03。							
功能描述	设置 GRQ 消息中的请求类型指示语。						
使用范例	ReqTypeIndicators=0x03						

3.1.2.27.8.7.4 去话呼叫：定制 IAI/IAM 消息

3.1.2.27.8.7.4.1 ConnectReqMsg

配置项	ConnectReqMsg
节	[TUP]
书写格式	ConnectReqMsg=m
取值范围	<p>m=0: 由驱动程序自动选择。如果驱动程序内部的去话主叫号码缓冲区（可通过函数 SsmSetTxCallerId 或配置项 TxCallerId 设置）或原被叫号码缓冲区（可通过函数 SsmSetTxOriginalCallerID 设置）不为空，使用 IAI 消息；否则使用 IAM 消息；</p> <p>m=1: 使用 IAM 消息；</p> <p>m=2: 使用 IAI 消息。</p> <p>缺省值为 0。</p>
功能描述	本端发起去话呼叫时，本配置项设置使用的初始地址消息的类型。

3.1.2.27.8.7.4.2 CalloutIAM_CAT

配置项	CalloutIAM_CAT
节	[TUP]

书写格式	CalloutIAM_CAT=0xk																																																																																																																
取值范围	<p>k: 8 比特数据，以 16 进制表示，8 个比特从高到低表示为 HGFEDCBA:</p> <table border="1"> <thead> <tr> <th colspan="4">低位字节: HGFEDCBA</th> </tr> <tr> <th>比特</th> <th>含义</th> <th>取值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>HG</td> <td>备用</td> <td>00</td> <td></td> </tr> <tr> <td></td> <td></td> <td>000000</td> <td>来源未知，在国际半自动接续中使用</td> </tr> <tr> <td></td> <td></td> <td>000001</td> <td>话务员，法语，在国际半自动接续中使用</td> </tr> <tr> <td></td> <td></td> <td>000010</td> <td>话务员，英语，在国际半自动接续中使用</td> </tr> <tr> <td></td> <td></td> <td>000011</td> <td>话务员，德语，在国际半自动接续中使用</td> </tr> <tr> <td></td> <td></td> <td>000100</td> <td>话务员，俄语，在国际半自动接续中使用</td> </tr> <tr> <td></td> <td></td> <td>000101</td> <td>话务员，西班牙语，在国际半自动接续中使用</td> </tr> <tr> <td></td> <td></td> <td>000110</td> <td>双方协商采用的语言(汉语)，在国际半自动接续中使用</td> </tr> <tr> <td></td> <td></td> <td>000111</td> <td>双方协商采用的语言，在国际半自动接续中使用</td> </tr> <tr> <td></td> <td></td> <td>001000</td> <td>双方协商采用的语言(日语)，在国际半自动接续中使用</td> </tr> <tr> <td></td> <td></td> <td>001001</td> <td>国内话务员(具有插入性能)</td> </tr> <tr> <td></td> <td></td> <td>001010</td> <td>普通用户、在长(国际)--长，长(国际)--市，局间使用</td> </tr> <tr> <td></td> <td></td> <td>001011</td> <td>优先用户、在长(国际)--长，长(国际)--市，市--市局间使用</td> </tr> <tr> <td></td> <td></td> <td>001100</td> <td>数据呼叫</td> </tr> <tr> <td></td> <td></td> <td>001101</td> <td>测试呼叫</td> </tr> <tr> <td></td> <td></td> <td>001110</td> <td>至 001111 备用</td> </tr> <tr> <td></td> <td></td> <td>010000</td> <td>普通、免费，在市--长(国际)局间使用</td> </tr> <tr> <td></td> <td></td> <td>010001</td> <td>普通、定期，在市--长(国际)局间使用</td> </tr> <tr> <td></td> <td></td> <td>010010</td> <td>普通、用户表、立即，在市--长(国际)局间使用</td> </tr> <tr> <td></td> <td></td> <td>010011</td> <td>普通、打印机、立即，在市--长(国际)局间使用</td> </tr> <tr> <td></td> <td></td> <td>010100</td> <td>优先、免费，在市--长(国际)局间使用</td> </tr> <tr> <td></td> <td></td> <td>010101</td> <td>优先、定期，在市--长(国际)局间使用</td> </tr> <tr> <td></td> <td></td> <td>010110</td> <td>备用</td> </tr> <tr> <td></td> <td></td> <td>010111</td> <td>备用</td> </tr> <tr> <td></td> <td></td> <td>011000</td> <td>普通用户，在市--市局间使用</td> </tr> <tr> <td></td> <td></td> <td>011001</td> <td>至 111111 备用</td> </tr> </tbody> </table> <p>缺省值为 0x18。</p>	低位字节: HGFEDCBA				比特	含义	取值	描述	HG	备用	00				000000	来源未知，在国际半自动接续中使用			000001	话务员，法语，在国际半自动接续中使用			000010	话务员，英语，在国际半自动接续中使用			000011	话务员，德语，在国际半自动接续中使用			000100	话务员，俄语，在国际半自动接续中使用			000101	话务员，西班牙语，在国际半自动接续中使用			000110	双方协商采用的语言(汉语)，在国际半自动接续中使用			000111	双方协商采用的语言，在国际半自动接续中使用			001000	双方协商采用的语言(日语)，在国际半自动接续中使用			001001	国内话务员(具有插入性能)			001010	普通用户、在长(国际)--长，长(国际)--市，局间使用			001011	优先用户、在长(国际)--长，长(国际)--市，市--市局间使用			001100	数据呼叫			001101	测试呼叫			001110	至 001111 备用			010000	普通、免费，在市--长(国际)局间使用			010001	普通、定期，在市--长(国际)局间使用			010010	普通、用户表、立即，在市--长(国际)局间使用			010011	普通、打印机、立即，在市--长(国际)局间使用			010100	优先、免费，在市--长(国际)局间使用			010101	优先、定期，在市--长(国际)局间使用			010110	备用			010111	备用			011000	普通用户，在市--市局间使用			011001	至 111111 备用
低位字节: HGFEDCBA																																																																																																																	
比特	含义	取值	描述																																																																																																														
HG	备用	00																																																																																																															
		000000	来源未知，在国际半自动接续中使用																																																																																																														
		000001	话务员，法语，在国际半自动接续中使用																																																																																																														
		000010	话务员，英语，在国际半自动接续中使用																																																																																																														
		000011	话务员，德语，在国际半自动接续中使用																																																																																																														
		000100	话务员，俄语，在国际半自动接续中使用																																																																																																														
		000101	话务员，西班牙语，在国际半自动接续中使用																																																																																																														
		000110	双方协商采用的语言(汉语)，在国际半自动接续中使用																																																																																																														
		000111	双方协商采用的语言，在国际半自动接续中使用																																																																																																														
		001000	双方协商采用的语言(日语)，在国际半自动接续中使用																																																																																																														
		001001	国内话务员(具有插入性能)																																																																																																														
		001010	普通用户、在长(国际)--长，长(国际)--市，局间使用																																																																																																														
		001011	优先用户、在长(国际)--长，长(国际)--市，市--市局间使用																																																																																																														
		001100	数据呼叫																																																																																																														
		001101	测试呼叫																																																																																																														
		001110	至 001111 备用																																																																																																														
		010000	普通、免费，在市--长(国际)局间使用																																																																																																														
		010001	普通、定期，在市--长(国际)局间使用																																																																																																														
		010010	普通、用户表、立即，在市--长(国际)局间使用																																																																																																														
		010011	普通、打印机、立即，在市--长(国际)局间使用																																																																																																														
		010100	优先、免费，在市--长(国际)局间使用																																																																																																														
		010101	优先、定期，在市--长(国际)局间使用																																																																																																														
		010110	备用																																																																																																														
		010111	备用																																																																																																														
		011000	普通用户，在市--市局间使用																																																																																																														
		011001	至 111111 备用																																																																																																														
功能描述	设置 IAI/IAM 消息中的主叫用户类别字段。																																																																																																																
使用范例	CalloutIAM_MSGPNT=0xabcd																																																																																																																

3.1.2.27.8.7.4.3 CalloutIAM_MsgPntr

配置项	CalloutIAM_MsgPntr
节	[TUP]
书写格式	CalloutIAM_MsgPntr=0xabcd

取值范围	abcd: 16 比特数据, 以 16 进制表示, 低 12 比特有效, Bit11~Bit0 依次记为 LKJIHGFEDCBA, 各比特的含义为:			
	比特	含义	取值	描述
	H	国际来话呼叫表示语	0	不是国际来话呼叫
			1	是国际来话呼叫
	G	去话回声抑制器表示语	0	未包括去话回声抑制器
			1	包括了去话回声抑制器
			00	不需要进行导通检验
	FE	导通检验表示语	01	该段电路需要进行导通检验
			10	在前段电路进行了导通检验
			11	备用
	DC	电路性质表示语	00	在接续中无卫星电路
			01	在接续中有卫星电路
			其它	备用
	BA	地址性质表示语	00	市话用户号码
			01	备用
			10	国内有效号码
			11	国际号码
	L	备用	0	任何通道
	K	信号通道表示语	1	全部是 NO.7 信号方式通道
	J	需要全部是数字通道表示语	0	普通呼叫
			1	需要全数字通道
	I	改发呼叫表示语	0	非改发呼叫
			1	改发呼叫
缺省值为 0x0000。				
功能描述	设置 IAI/IAM 消息中的消息表示语字段。			
使用范例	CalloutIAM_MsgPntr=0x0000			

3.1.2.27.8.7.4.4 CallingIndicatorBit

配置项	CallingIndicatorBit
节	[TUP]
书写格式	CallingIndicatorBit=n
取值范围	n=0x10: 使用比特 E; n=0x04: 使用比特 C (缺省值)。
功能描述	在 IAI 消息中, 有一个第 1 表示语八位位组(octet), 从 Bit7~Bit0 依次记为 HGFEDCBA, 其中, 比特 C 为附加主叫用户信息表示语, 比特 E 为主叫用户线标识表示语。如果在 IAI 消息中包含了主叫号码, 有的交换机要求使用比特 E, 有的交换机要求使用比特 C。具体使用哪一个, 应根据对端交换机的实际需要, 通过本配置项进行设置。
使用范例	CallingIndicatorBit=0x10

3.1.2.27.8.7.4.5 OriginalCalleeAddrInd

配置项	OriginalCalleeAddrInd
-----	------------------------------

节	[TUP]																								
书写格式	OriginalCalleeAddrInd=0xn																								
取值范围	<p>n: 16 进制的地址表示语, 共 8 比特, 低 4 比特有效。从 Bit3~Bit0, 依次记为 DCBA, 各比特的含义为:</p> <table border="1"> <thead> <tr> <th>比特</th> <th>含义</th> <th>取值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>DC</td> <td>备用</td> <td>00</td> <td></td> </tr> <tr> <td></td> <td></td> <td>00</td> <td>市内用户号码</td> </tr> <tr> <td>BA</td> <td>地址性质表示语</td> <td>01</td> <td>国内备用</td> </tr> <tr> <td></td> <td></td> <td>10</td> <td>国内有效号码</td> </tr> <tr> <td></td> <td></td> <td>11</td> <td>国际号码</td> </tr> </tbody> </table> <p>缺省值为 0x02。</p>	比特	含义	取值	描述	DC	备用	00				00	市内用户号码	BA	地址性质表示语	01	国内备用			10	国内有效号码			11	国际号码
比特	含义	取值	描述																						
DC	备用	00																							
		00	市内用户号码																						
BA	地址性质表示语	01	国内备用																						
		10	国内有效号码																						
		11	国际号码																						
功能描述	设置 IAI 消息中原被叫地址字段的地址表示语。																								
使用范例	OriginalCalleeAddrInd=0x02																								

3.1.2.27.8.7.4.6 CallerAddrInd

配置项	CallerAddrInd																																				
节	[TUP]																																				
书写格式	CallerAddrInd =0xn																																				
取值范围	<p>n: 16 进制的地址表示语, 共 8 比特, 低 4 比特有效。从 Bit3~Bit0, 依次记为 DCBA, 各比特的含义为:</p> <table border="1"> <thead> <tr> <th>比特</th> <th>含义</th> <th>取值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>D</td> <td>主叫号码不全表示语</td> <td>0</td> <td>未表示不全</td> </tr> <tr> <td></td> <td></td> <td>1</td> <td>主叫号码不全</td> </tr> <tr> <td>C</td> <td>提供主叫用户号码表示语</td> <td>0</td> <td>未限制提供主叫号码</td> </tr> <tr> <td></td> <td></td> <td>1</td> <td>限制提供主叫号码</td> </tr> <tr> <td>BA</td> <td>地址性质表示语</td> <td>00</td> <td>市内用户号码</td> </tr> <tr> <td></td> <td></td> <td>01</td> <td>国内备用</td> </tr> <tr> <td></td> <td></td> <td>10</td> <td>国内有效号码</td> </tr> <tr> <td></td> <td></td> <td>11</td> <td>国际号码</td> </tr> </tbody> </table> <p>缺省值为 0x02。</p>	比特	含义	取值	描述	D	主叫号码不全表示语	0	未表示不全			1	主叫号码不全	C	提供主叫用户号码表示语	0	未限制提供主叫号码			1	限制提供主叫号码	BA	地址性质表示语	00	市内用户号码			01	国内备用			10	国内有效号码			11	国际号码
比特	含义	取值	描述																																		
D	主叫号码不全表示语	0	未表示不全																																		
		1	主叫号码不全																																		
C	提供主叫用户号码表示语	0	未限制提供主叫号码																																		
		1	限制提供主叫号码																																		
BA	地址性质表示语	00	市内用户号码																																		
		01	国内备用																																		
		10	国内有效号码																																		
		11	国际号码																																		
功能描述	设置 IAI 消息中主叫用户线标识字段的地址表示语。																																				
使用范例	CallerAddrInd =0x02																																				
注意事项	<ul style="list-style-type: none"> 本配置项要求 SynCTI Ver. 5.3.1.2 及以上版本。 																																				

3.1.2.27.8.7.5 去话呼叫：设置应答 GRQ 消息的模式

3.1.2.27.8.7.5.1 AutoSendGSM

配置项	AutoSendGSM
节	[TUP]
书写格式	AutoSendGSM=m
取值范围	m=0: 由应用程序接管，通道迁移到“挂起”状态，应用程序调用 SsmSetTxCallerId 设置主叫号码后恢复去话呼叫过程； m=1: 由驱动程序自动应答（缺省值）。
功能描述	去话呼叫时，如果收到对端交换机请求主叫号码的 GRQ 消息，本配置项用于决定是否由驱动程序自动应答 GSM 消息。更多信息请参见第 1 章中“ TUP 通道的状态转移 ”部分内容。

3.1.2.27.8.7.6 不使用 SynCTI 提供的 TUP 协议状态机

3.1.2.27.8.7.6.1 AutoHandleTup

配置项	AutoHandleTup
节	[SS7]
书写格式	AutoHandleTup=b
取值范围	b=0: 否； b=1: 是（缺省值）。
功能描述	设置是否使用 SynCTI 驱动程序提供的 TUP 协议状态机。

3.1.2.27.8.7.7 输出调试信息

3.1.2.27.8.7.7.1 DebugViewTupCallProc

配置项	DebugViewTupCallProc
节	[SS7]
书写格式	DebugViewTupCallProc=b
取值范围	b=0: 否（缺省值）； b=1: 是。
功能描述	设置是否将 TUP 协议状态机的调试信息输出到 DebugView 工具软件中。如果选择是，需要先运行 DebugView 工具软件。

注意事项	开启 TUP 协议的调试功能后，会降低应用系统的运行效率，因此仅用于调试的目的。
------	------------------------------------------

3.1.2.27.8.8 ISUP 协议高级配置项

3.1.2.27.8.8.1 来话呼叫：定制本端接受来话呼叫时应答的消息类型

3.1.2.27.8.8.1.1 DefaultCalledPickupMsg

配置项	DefaultCalledPickupMsg
节	[ISUP]
书写格式	DefaultCalledPickupMsg=k
取值范围	k=0x09: 使用 ANM 消息（应答）; k=0x07: 使用 CON 消息（地址全并摘机）。 缺省值为 0x09。
功能描述	当 ISUP 通道处于“振铃”状态时，如果应用程序调用 SsmPickup 或 SsmPickupANX 函数，本配置项设置向对端交换机发送的消息类型。
注意事项	本配置项要求 SynCTI Ver. 4.7.1.8 及更高。
使用范例	DefaultCalledPickupMsg=0x09

3.1.2.27.8.8.2 来话呼叫：定制后向呼叫表示语

3.1.2.27.8.8.2.1 DefaultBackwardCallInd

配置项	DefaultBackwardCallInd
节	[ISUP]
书写格式	DefaultBackwardCallInd=k

取值范围	k: 后向呼叫表示语字段的参数值，包含 2 个字节，从最高位比特（Bit15）到最低位比特（Bit0）记为 PONMLKJIHGFEDCBA，各比特含义如下：		
	比特组合	含义	取值范围及描述
	BA	计费表示语	00: 无指示 01: 不计费 10: 计费 11: 备用
	DC	被叫用户状态表示语	00: 无指示 01: 用户空 10: 空时连接 11: 备用
	FE	被叫用户类别表示语	00: 无指示 01: 普通用户 10: 付费电话 11: 备用
	HG	端至端方式表示语 (注)	00: 端至端方式不可用 (只有逐段转发方式可用) 01: 传递方式可用 10: SCCP 方式可用 11: 传递方式和 SCCP 方式可用
	I	互通表示语(注)	0: 未遇到互通 1: 遇到互通
	J	端至端信息表示语(注)	0: 无端至端信息可用 1: 端至端信息可用
	K	ISDN 用户部分表示语 (注)	0: 不是所有方向都使用 ISDN 用户部分 1: 所有方向都使用 ISDN 用户部分
	L	保持表示语	0: 未请求保持 1: 请求保持
功能描述	设置 ACM 和 CON 消息中的后向呼叫表示语字段。		
	注意事项 本配置项要求 SynCTI Ver. 4.7.1.8 及更高。		
	使用范例 DefaultBackwardCallInd=0x1416		

3.1.2.27.8.8.3 来话呼叫：定制 REL 消息

3.1.2.27.8.8.3.1 DefaultHangupRELInd

配置项	DefaultHangupRELInd
节	[ISUP]
书写格式	DefaultHangupRELInd=0xk

	k 的取值范围及含义如下表所示:																											
取值范围	<table border="1"> <thead> <tr> <th>取值</th> <th>shpa3api.h 中的宏</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0x01</td> <td>C_ISUP_REL_UNN</td> <td>空号</td> </tr> <tr> <td>0x10</td> <td>C_ISUP_REL_NORMAL_REL</td> <td>正常拆线</td> </tr> <tr> <td>0x11</td> <td>C_ISUP_REL_BUSY</td> <td>用户忙</td> </tr> <tr> <td>0x12</td> <td>C_ISUP_REL_NOREPLY</td> <td>无应答</td> </tr> <tr> <td>0x15</td> <td>C_ISUP_REL_DENY</td> <td>拒绝</td> </tr> <tr> <td>0x1c</td> <td>C_ISUP_REL_LACKADDR</td> <td>地址不全</td> </tr> <tr> <td>0x1f</td> <td>C_ISUP_REL_NORMAL</td> <td>正常</td> </tr> <tr> <td>0x2a</td> <td>C_ISUP_REL_BLOCK</td> <td>交换设备拥塞</td> </tr> </tbody> </table> <p>缺省值为 0x15(C_ISUP_REL_DENY)。</p>	取值	shpa3api.h 中的宏	描述	0x01	C_ISUP_REL_UNN	空号	0x10	C_ISUP_REL_NORMAL_REL	正常拆线	0x11	C_ISUP_REL_BUSY	用户忙	0x12	C_ISUP_REL_NOREPLY	无应答	0x15	C_ISUP_REL_DENY	拒绝	0x1c	C_ISUP_REL_LACKADDR	地址不全	0x1f	C_ISUP_REL_NORMAL	正常	0x2a	C_ISUP_REL_BLOCK	交换设备拥塞
取值	shpa3api.h 中的宏	描述																										
0x01	C_ISUP_REL_UNN	空号																										
0x10	C_ISUP_REL_NORMAL_REL	正常拆线																										
0x11	C_ISUP_REL_BUSY	用户忙																										
0x12	C_ISUP_REL_NOREPLY	无应答																										
0x15	C_ISUP_REL_DENY	拒绝																										
0x1c	C_ISUP_REL_LACKADDR	地址不全																										
0x1f	C_ISUP_REL_NORMAL	正常																										
0x2a	C_ISUP_REL_BLOCK	交换设备拥塞																										
功能描述	在通道有电话呼入，还没有摘机的条件下，当应用程序调用 SsmHangup 函数而引发驱动程序向对端交换机发送 REL 消息时，本配置项设置 REL 消息中携带的释放原因值。																											
注意事项	<ul style="list-style-type: none"> 如果设定值不在取值范围之内，驱动程序会自动调整为 0x15； 本配置项要求 SynCTI Ver. 4.7.1.5 或更高。 																											
使用范例	DefaultHangupRELInd=0x15																											

3.1.2.27.8.8.3.2 DefaultCauseInd

配置项	DefaultCauseInd																																																																																																
节	[ISUP]																																																																																																
书写格式	DefaultCauseInd=n																																																																																																
取值范围	<p>n: 16 比特数据，以 16 进制表示，各比特的含义为：</p> <table border="1"> <thead> <tr> <th colspan="4">低位字节: HGFEDCBA</th> </tr> <tr> <th>比特</th> <th>含义</th> <th>取值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>H</td> <td>扩充表示语</td> <td>0</td> <td>八位位组延续到下一个八位位组（建议八位位组）</td> </tr> <tr> <td></td> <td></td> <td>1</td> <td>最后的八位位组</td> </tr> <tr> <td>GF</td> <td>编码标准</td> <td>00</td> <td>ITU-T 标准编码</td> </tr> <tr> <td></td> <td></td> <td>其它</td> <td>保留</td> </tr> <tr> <td>E</td> <td>备用</td> <td>0</td> <td>应设置为 0</td> </tr> <tr> <td></td> <td></td> <td>0000</td> <td>用户 (U)</td> </tr> <tr> <td></td> <td></td> <td>0001</td> <td>为本地用户提供服务的专用网 (LPN)</td> </tr> <tr> <td></td> <td></td> <td>0010</td> <td>为本地用户提供服务的公用网 (LN)</td> </tr> <tr> <td></td> <td></td> <td>0011</td> <td>转接网络 (TN)</td> </tr> <tr> <td>DCBA</td> <td>位置</td> <td>0100</td> <td>为远端用户提供服务的公用网 (RLN)</td> </tr> <tr> <td></td> <td></td> <td>0101</td> <td>为远端用户提供服务的专用网 (RPN)</td> </tr> <tr> <td></td> <td></td> <td>0111</td> <td>国际网络 (INTL)</td> </tr> <tr> <td></td> <td></td> <td>1010</td> <td>互通点以外的网路 (BI)</td> </tr> <tr> <td></td> <td></td> <td>其它</td> <td>备用</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th colspan="4">高位字节: HGFEDCBA</th> </tr> <tr> <th>比特</th> <th>含义</th> <th>取值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>H</td> <td>扩充表示语</td> <td>1</td> <td>最后的八位位组</td> </tr> <tr> <td></td> <td></td> <td>0000000</td> <td>Q.931</td> </tr> <tr> <td></td> <td></td> <td>0000011</td> <td>X.21</td> </tr> <tr> <td>GFEDCBA</td> <td>建议</td> <td>0000100</td> <td>X.25</td> </tr> <tr> <td></td> <td></td> <td>0000101</td> <td>公共陆地移动网, Q.1031/Q.1051</td> </tr> <tr> <td></td> <td></td> <td>其它</td> <td>保留</td> </tr> </tbody> </table> <p>缺省值为 0x0000。</p>	低位字节: HGFEDCBA				比特	含义	取值	描述	H	扩充表示语	0	八位位组延续到下一个八位位组（建议八位位组）			1	最后的八位位组	GF	编码标准	00	ITU-T 标准编码			其它	保留	E	备用	0	应设置为 0			0000	用户 (U)			0001	为本地用户提供服务的专用网 (LPN)			0010	为本地用户提供服务的公用网 (LN)			0011	转接网络 (TN)	DCBA	位置	0100	为远端用户提供服务的公用网 (RLN)			0101	为远端用户提供服务的专用网 (RPN)			0111	国际网络 (INTL)			1010	互通点以外的网路 (BI)			其它	备用	高位字节: HGFEDCBA				比特	含义	取值	描述	H	扩充表示语	1	最后的八位位组			0000000	Q.931			0000011	X.21	GFEDCBA	建议	0000100	X.25			0000101	公共陆地移动网, Q.1031/Q.1051			其它	保留
低位字节: HGFEDCBA																																																																																																	
比特	含义	取值	描述																																																																																														
H	扩充表示语	0	八位位组延续到下一个八位位组（建议八位位组）																																																																																														
		1	最后的八位位组																																																																																														
GF	编码标准	00	ITU-T 标准编码																																																																																														
		其它	保留																																																																																														
E	备用	0	应设置为 0																																																																																														
		0000	用户 (U)																																																																																														
		0001	为本地用户提供服务的专用网 (LPN)																																																																																														
		0010	为本地用户提供服务的公用网 (LN)																																																																																														
		0011	转接网络 (TN)																																																																																														
DCBA	位置	0100	为远端用户提供服务的公用网 (RLN)																																																																																														
		0101	为远端用户提供服务的专用网 (RPN)																																																																																														
		0111	国际网络 (INTL)																																																																																														
		1010	互通点以外的网路 (BI)																																																																																														
		其它	备用																																																																																														
高位字节: HGFEDCBA																																																																																																	
比特	含义	取值	描述																																																																																														
H	扩充表示语	1	最后的八位位组																																																																																														
		0000000	Q.931																																																																																														
		0000011	X.21																																																																																														
GFEDCBA	建议	0000100	X.25																																																																																														
		0000101	公共陆地移动网, Q.1031/Q.1051																																																																																														
		其它	保留																																																																																														

功能描述	设置 REL 消息中的原因表示语字段的编码标准位置及建议参数，其中建议参数可选。若不选择建议参数，则低 8 比特的扩充位必须置 1，驱动将忽略高 8 比特值；若选择建议参数，则低 8 比特的扩充位必须置 0。
使用范例	DefaultCauseInd=0x0000

3.1.2.27.8.8.4 来话呼叫：设置其它参数

3.1.2.27.8.8.4.1 SaveRGNTo1stPhoNumStr

配置项	SaveRGNTo1stPhoNumStr
节	[ISUP]
书写格式	SaveRGNTo1stPhoNumStr=b
取值范围	b=0: 否; b=1: 是（缺省值）。
功能描述	设置函数 SsmGet1stPhoNumStr 或 SsmGet1stPhoNumStrA 是否只返回 IAM 消息中的原被叫号码信息。更多信息请参见上述函数的说明。
注意事项	<ul style="list-style-type: none"> ● 本配置项仅适用于 ISUP 通道； ● 本配置项要求 SynCTI 驱动程序的版本为 Ver 4.7.1.7 或更高。

3.1.2.27.8.8.5 去话呼叫：定制 IAM 消息

3.1.2.27.8.8.5.1 DefaultNatureOfConnectionInd

配置项	DefaultNatureOfConnectionInd
节	[ISUP]
书写格式	DefaultNatureOfConnectionInd=0xk
取值范围	k: 16 进制形式的连接性质表示码，缺省值为 0x00。具体取值请参见 附录 2 ISUP 可选参数列表和部分参数介绍 ，通常由对端交换机根据其具体配置提供实际数值。
功能描述	设置 IAM 消息中的连接性质表示语。
使用范例	DefaultNatureOfConnectionInd=0x00

3.1.2.27.8.8.5.2 DefaultIAM_ForceCallInd

配置项	DefaultIAM_ForceCallInd
节	[ISUP]

书写格式	DefaultIAM_ForwardCallInd=0xk
取值范围	k: 16 进制形式的前向呼叫指示语，缺省值为 0x0040。具体取值请参见 附录 2 ISUP 可选参数列表和部分参数介绍 ，通常由对端交换机根据其具体配置提供实际数值。
功能描述	设置 IAM 消息中的前向呼叫指示语。
使用范例	DefaultIAM_ForwardCallInd=0x0040

3.1.2.27.8.8.5.3 DefaultIAM_CAT

配置项	DefaultIAM_CAT
节	[ISUP]
书写格式	DefaultIAM_CAT=0xk
取值范围	k: 16 进制形式的主叫用户类别，缺省值为 0x0a。具体取值请参见 附录 2 ISUP 可选参数列表和部分参数介绍 ，通常由对端交换机根据其具体配置提供实际数值。
功能描述	设置 IAM 消息中的主叫用户类别。
使用范例	DefaultIAM_CAT=0x0a

3.1.2.27.8.8.5.4 DefaultIAM_TransmissionMediumRequirment

配置项	DefaultIAM_TransmissionMediumRequirment
节	[ISUP]
书写格式	格式 1: DefaultIAM_TransmissionMediumRequirment=0xk 格式 2: DefaultIAM_TransmissionMediumRequirment[N]=0xk
取值范围	k: 16 进制形式的传输媒介要求，缺省值为 0x00。具体取值请参见 附录 2 ISUP 可选参数列表和部分参数介绍 ，通常由对端交换机根据其具体配置提供实际数值。 N: 为所在链路 PCM 编号。当整个系统使用同样的传输媒介时，使用格式 1；当系统中的链路使用不同的传输媒介时，需使用格式 2，但同时需要保留格式 1，并将 k 值设置为 0xff。
功能描述	设置 IAM 消息中的传输媒介要求。
注意事项	在使用格式 2 时，一定要保留格式 1，并将 K 设置为 0xff。

使用范例	格式 1: DefaultIAM_TransmissionMediumRequirement=0x02
	格式 2: DefaultIAM_TransmissionMediumRequirement=0xff ; 必须要设置
	DefaultIAM_TransmissionMediumRequirement[0]=0x02 ; PCM 0,1 使用 0x02
	DefaultIAM_TransmissionMediumRequirement[1]=0x02
	DefaultIAM_TransmissionMediumRequirement[2]=0x04 ; PCM 2,3 使用 0x04
	DefaultIAM_TransmissionMediumRequirement[3]=0x04

3.1.2.27.8.8.5.5 DefaultIAM_CalleeParam

配置项	DefaultIAM_CalleeParam																																																						
节	[ISUP]																																																						
书写格式	DefaultIAM_CalleeParam=0xk																																																						
取值范围	<p>k: 16 比特数据，以 16 进制表示，各比特的含义为：</p> <table border="1"> <thead> <tr> <th colspan="4">低位字节：HGFEDCBA</th> </tr> <tr> <th>比特</th> <th>含义</th> <th>取值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td rowspan="2">H</td> <td rowspan="2">奇偶表示语</td> <td>0</td> <td>地址为偶数个</td> </tr> <tr> <td>1</td> <td>地址为奇数个</td> </tr> <tr> <td rowspan="5">GFEDCBA</td> <td rowspan="5">地址性质表示语</td> <td>0000001</td> <td>用户号码</td> </tr> <tr> <td>0000011</td> <td>国内号码</td> </tr> <tr> <td>0000100</td> <td>国际号码</td> </tr> <tr> <td>其它</td> <td>备用</td> </tr> <tr> <td></td> <td></td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th colspan="4">高位字节：HGFEDCBA</th> </tr> <tr> <th>比特</th> <th>含义</th> <th>取值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td rowspan="2">H</td> <td rowspan="2">内部网络号码表示语</td> <td>0</td> <td>允许路由到内部网络号码</td> </tr> <tr> <td>1</td> <td>不允许路由到内部网络号码</td> </tr> <tr> <td rowspan="5">GFE</td> <td rowspan="5">号码计划表示语</td> <td>001</td> <td>ISDN(电话)号码计划(建议 E.164, E.163)</td> </tr> <tr> <td>011</td> <td>数据号码计划 (建议 X.164)</td> </tr> <tr> <td>100</td> <td>用户电报号码计划 (建议 F.164)</td> </tr> <tr> <td>其它</td> <td>备用</td> </tr> <tr> <td>DCBA</td> <td>备用</td> <td></td> <td></td> </tr> </tbody> </table> <p>缺省值为 0x1003。</p>	低位字节：HGFEDCBA				比特	含义	取值	描述	H	奇偶表示语	0	地址为偶数个	1	地址为奇数个	GFEDCBA	地址性质表示语	0000001	用户号码	0000011	国内号码	0000100	国际号码	其它	备用			高位字节：HGFEDCBA				比特	含义	取值	描述	H	内部网络号码表示语	0	允许路由到内部网络号码	1	不允许路由到内部网络号码	GFE	号码计划表示语	001	ISDN(电话)号码计划(建议 E.164, E.163)	011	数据号码计划 (建议 X.164)	100	用户电报号码计划 (建议 F.164)	其它	备用	DCBA	备用		
低位字节：HGFEDCBA																																																							
比特	含义	取值	描述																																																				
H	奇偶表示语	0	地址为偶数个																																																				
		1	地址为奇数个																																																				
GFEDCBA	地址性质表示语	0000001	用户号码																																																				
		0000011	国内号码																																																				
		0000100	国际号码																																																				
		其它	备用																																																				
高位字节：HGFEDCBA																																																							
比特	含义	取值	描述																																																				
H	内部网络号码表示语	0	允许路由到内部网络号码																																																				
		1	不允许路由到内部网络号码																																																				
GFE	号码计划表示语	001	ISDN(电话)号码计划(建议 E.164, E.163)																																																				
		011	数据号码计划 (建议 X.164)																																																				
		100	用户电报号码计划 (建议 F.164)																																																				
		其它	备用																																																				
		DCBA	备用																																																				
功能描述	设置 IAM 消息中被叫用户号码字段的被叫参数。																																																						
使用范例	DefaultIAM_CalleeParam=0x1003																																																						

3.1.2.27.8.8.5.6 DefaultIAM_CallerParam

配置项	DefaultIAM_CallerParam
-----	------------------------

节	[ISUP]																																																																				
书写格式	DefaultIAM_CallerParam=0xk																																																																				
取值范围	<p>k: 16 比特数据, 以 16 进制表示, 各比特的含义为:</p> <table border="1"> <thead> <tr> <th colspan="4">低位字节: HGFEDCBA</th> </tr> <tr> <th>比特</th> <th>含义</th> <th>取值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td rowspan="7">GFEDCBA</td> <td rowspan="7">地址性质表示语</td> <td>0</td> <td>地址为偶数个</td> </tr> <tr> <td>1</td> <td>地址为奇数个</td> </tr> <tr> <td>0000001</td> <td>用户号码</td> </tr> <tr> <td>0000011</td> <td>国内号码</td> </tr> <tr> <td>0000100</td> <td>国际号码</td> </tr> <tr> <td>其它</td> <td>备用</td> </tr> <tr> <td></td> <td></td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th colspan="4">高位字节: HGFEDCBA</th> </tr> <tr> <th>比特</th> <th>含义</th> <th>取值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td rowspan="7">GFE</td> <td rowspan="7">号码计划表示语</td> <td>0</td> <td>全</td> </tr> <tr> <td>1</td> <td>不全</td> </tr> <tr> <td>001</td> <td>ISDN(电话)号码计划(建议 E.164, E.163)</td> </tr> <tr> <td>011</td> <td>数据号码计划 (建议 X.164)</td> </tr> <tr> <td>100</td> <td>用户电报号码计划 (建议 F.164)</td> </tr> <tr> <td>其它</td> <td>备用</td> </tr> <tr> <td></td> <td></td> </tr> <tr> <td rowspan="4">DC</td> <td rowspan="4">地址提供限制表示语</td> <td>00</td> <td>不限制提供</td> </tr> <tr> <td>01</td> <td>限制提供</td> </tr> <tr> <td>10</td> <td>地址不可获得</td> </tr> <tr> <td>其它</td> <td>备用</td> </tr> <tr> <td rowspan="4">BA</td> <td rowspan="4">显示表示语</td> <td>00</td> <td>用户提供, 不验证; (国内用)</td> </tr> <tr> <td>01</td> <td>用户保留, 验证和通过</td> </tr> <tr> <td>10</td> <td>用户提供, 验证和失败; (国内用)</td> </tr> <tr> <td>11</td> <td>网络提供</td> </tr> </tbody> </table> <p>缺省值为 0x1001。</p>	低位字节: HGFEDCBA				比特	含义	取值	描述	GFEDCBA	地址性质表示语	0	地址为偶数个	1	地址为奇数个	0000001	用户号码	0000011	国内号码	0000100	国际号码	其它	备用			高位字节: HGFEDCBA				比特	含义	取值	描述	GFE	号码计划表示语	0	全	1	不全	001	ISDN(电话)号码计划(建议 E.164, E.163)	011	数据号码计划 (建议 X.164)	100	用户电报号码计划 (建议 F.164)	其它	备用			DC	地址提供限制表示语	00	不限制提供	01	限制提供	10	地址不可获得	其它	备用	BA	显示表示语	00	用户提供, 不验证; (国内用)	01	用户保留, 验证和通过	10	用户提供, 验证和失败; (国内用)	11	网络提供
低位字节: HGFEDCBA																																																																					
比特	含义	取值	描述																																																																		
GFEDCBA	地址性质表示语	0	地址为偶数个																																																																		
		1	地址为奇数个																																																																		
		0000001	用户号码																																																																		
		0000011	国内号码																																																																		
		0000100	国际号码																																																																		
		其它	备用																																																																		
高位字节: HGFEDCBA																																																																					
比特	含义	取值	描述																																																																		
GFE	号码计划表示语	0	全																																																																		
		1	不全																																																																		
		001	ISDN(电话)号码计划(建议 E.164, E.163)																																																																		
		011	数据号码计划 (建议 X.164)																																																																		
		100	用户电报号码计划 (建议 F.164)																																																																		
		其它	备用																																																																		
DC	地址提供限制表示语	00	不限制提供																																																																		
		01	限制提供																																																																		
		10	地址不可获得																																																																		
		其它	备用																																																																		
BA	显示表示语	00	用户提供, 不验证; (国内用)																																																																		
		01	用户保留, 验证和通过																																																																		
		10	用户提供, 验证和失败; (国内用)																																																																		
		11	网络提供																																																																		
功能描述	设置 IAM 消息中主叫用户号码字段的主叫参数。																																																																				
使用范例	DefaultIAM_CallerParam=0x1001																																																																				

3.1.2.27.8.8.5.7 DefaultIAM_OriginalCalleeParam

配置项	DefaultIAM_OriginalCalleeParam
节	[ISUP]
书写格式	DefaultIAM_OriginalCalleeParam=0xk
取值范围	k: 16 进制形式的 16 比特数据, 缺省值为 0x1001。具体取值请参见 附录 2 ISUP 可选参数列表和部分参数介绍 。
功能描述	设置 IAM 消息中原被叫的头两字节信息, 包括地址性质表示语、编号计划、限制提供。具体取值请参见 ISUP 协议的相关说明, 通常由对端交换机根据其具体配置提供实际数值。
使用范例	DefaultIAM_OriginalCalleeParam=0x1001

3.1.2.27.8.8.5.8 DefaultIAM_RedirectingNumber

配置项	DefaultIAM_RedirectingNumber
节	[ISUP]
书写格式	DefaultIAM_RedirectingNumber=0xk
取值范围	k: 16 进制形式的 16 比特数据，缺省值为 0x1001。具体取值请参见 附录 2 ISUP 可选参数列表和部分参数介绍 。
功能描述	设置 IAM 消息中改发的号码的头两字节信息，包括地址性质表示语、编号计划、限制提供。具体取值请参见 ISUP 协议的相关说明，通常由对端交换机根据其具体配置提供实际数值。
使用范例	DefaultIAM_RedirectingNumber=0x1001

3.1.2.27.8.8.5.9 bSubscriberSI

请参见 [SubscriberSI](#)

3.1.2.27.8.8.5.10 SubscriberSI

配置项	bSubscriberSI SubscriberSI
节	[ISUP]
书写格式	bSubscriberSI=k SubscriberSI=0xk ₁ , 0xk ₂ , ..., 0xk _i , ..., 0xk ₁₁
取值范围	k: =0: 不包含（缺省值）; =1: 包含，用户业务信息的具体值在配置项 SubscriberSI 中指定。 k _i : 16 进制形式的用户业务信息，最多可以配置 11 个，缺省值为 0x80,0x90,0xa3，适用于华为交换机。k _i 的具体取值请参见 附录 2 ISUP 可选参数列表和部分参数介绍 ，通常由对端交换机根据其具体配置提供实际数值。
功能描述	bSubscriberSI 设置 IAM 消息中是否包含用户业务信息内容。 SubscriberSI 设置用户业务信息的内容，只有当 bSubscriberSI 设置为 1 时才有效。
使用范例	bSubscriberSI=1 SubscriberSI=0x80,0x90,0xa3

3.1.2.27.8.8.5.11 bOptionalFCI

请参见 [OptionalFCI](#)

3.1.2.27.8.8.5.12 OptionalFCI

配置项	bOptionalFCI OptionalFCI																																								
节	[ISUP]																																								
书写格式	bOptionalFCI=b OptionalFCI=0xk																																								
取值范围	<p>b: =0: 不包含 (缺省值); =1: 包含。</p> <p>k: 为 16 进制形式的 8 比特数据, 按照从高比特位到低比特位的顺序记为 HGFEDCBA, 各比特的含义为:</p> <table border="1"> <thead> <tr> <th>比特</th> <th>含义</th> <th>取值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>H</td> <td>被连接线识别请求表示语</td> <td>0</td> <td>未请求</td> </tr> <tr> <td></td> <td></td> <td>1</td> <td>请求</td> </tr> <tr> <td>Gfed</td> <td>备用</td> <td></td> <td></td> </tr> <tr> <td>C</td> <td>简单分段表示语</td> <td>0</td> <td>将不发送附加的信息</td> </tr> <tr> <td></td> <td></td> <td>1</td> <td>将在分段消息中发送附加的信息</td> </tr> <tr> <td></td> <td></td> <td>00</td> <td>非 CUG 呼叫</td> </tr> <tr> <td></td> <td></td> <td>01</td> <td>备用</td> </tr> <tr> <td>BA</td> <td>闭合用户群呼叫表示语</td> <td>10</td> <td>闭合用户群呼叫, 允许出接口</td> </tr> <tr> <td></td> <td></td> <td>11</td> <td>闭合用户群呼叫, 不允许出接口</td> </tr> </tbody> </table> <p>缺省值为 0。如果需要配置, 通常由对端交换机根据其具体配置提供实际数值。</p>	比特	含义	取值	描述	H	被连接线识别请求表示语	0	未请求			1	请求	Gfed	备用			C	简单分段表示语	0	将不发送附加的信息			1	将在分段消息中发送附加的信息			00	非 CUG 呼叫			01	备用	BA	闭合用户群呼叫表示语	10	闭合用户群呼叫, 允许出接口			11	闭合用户群呼叫, 不允许出接口
比特	含义	取值	描述																																						
H	被连接线识别请求表示语	0	未请求																																						
		1	请求																																						
Gfed	备用																																								
C	简单分段表示语	0	将不发送附加的信息																																						
		1	将在分段消息中发送附加的信息																																						
		00	非 CUG 呼叫																																						
		01	备用																																						
BA	闭合用户群呼叫表示语	10	闭合用户群呼叫, 允许出接口																																						
		11	闭合用户群呼叫, 不允许出接口																																						
功能描述	bOptionalFCI 设置 IAM 消息中是否包含任选前向呼叫表示语。如果包含, 任选前向呼叫表示语的值在配置项 OptionalFCI 中指定。 OptionalFCI 任选前向呼叫表示语的具体值。																																								
使用范例	bOptionalFCI=1 OptionalFCI=0x01																																								

3.1.2.27.8.8.5.13 Usr2UsrInfo

配置项	Usr2UsrInfo
节	[ISUP]
书写格式	Usr2UsrInfo=s ₁ ,s ₂ ,...,s _i ,...,s _N
取值范围	s _i : 十六进制字符, 最多可以设置 131 个 (即 N 最大为 131)。缺省值为空, 即不包含该字段。
功能描述	设置 IAM 消息中的用户至用户信息字段。本配置项设置的功能也可以通过函数 SsmSetIsupParameter 进行设置。
使用范例	Usr2UsrInfo=a0,31,6c

3.1.2.27.8.8.5.14 LocationNumber

配置项	LocationNumber
节	[ISUP]
书写格式	LocationNumber=s ₁ ,s ₂ ,...,s _i ,...,s _N
取值范围	s _i : 十六进制字符, 最多可以设置 50 个 (即 N 最大为 50)。缺省值为空, 即不包含该字段。
功能描述	设置 IAM 消息中的 LocationNumber 信息字段。本配置项设置的功能也可以通过函数 SsmSetIsupParameter 进行设置。
使用范例	LocationNumber=0x04,0x13,0x19,0x89,0x45,0x90,0x09,0x20

3.1.2.27.8.8.6 去话呼叫: 设置应答 INF 消息的模式

3.1.2.27.8.8.6.1 AutoSendINF

配置项	AutoSendINF
节	[ISUP]
书写格式	AutoSendINF=m
取值范围	m=0: 由应用程序接管; m=1: 由驱动程序自动应答 (缺省值)。
功能描述	去话呼叫时, 如果收到对端交换机请求主叫号码的 INR 消息, 本配置项用于决定是否由驱动程序自动应答 INF 消息。详细信息请参见第 1 章中 “ ISUP 通道的状态转移 ” 部分内容。

3.1.2.27.8.8.7 去话呼叫: 设置等待 ACM 消息的超时时间

3.1.2.27.8.8.7.1 MaxWaitACMTime

配置项	MaxWaitACMTime
节	[ISUP]
书写格式	MaxWaitACMTime=n
取值范围	n≥0: 驱动程序等待对方 ACM 消息最大时间 (缺省值 25), 单位: 秒。
功能描述	去话呼叫时, 本配置项决定等待对方回复 ACM 消息的最大时间。

3.1.2.27.8.8.8 设置挂起状态自动迁移到空闲状态的超时时间

3.1.2.27.8.8.8.1 MaxWaitPendingTidleTime

配置项	MaxWaitPendingTidleTime
节	[ISUP]
书写格式	MaxWaitPendingTidleTime=n
取值范围	n≥0: 驱动程序从挂起状态自动迁移到空闲状态的等待时间（缺省值 60），单位：秒。
功能描述	本配置项决定驱动程序从挂起状态自动迁移到空闲状态的等待时间。

3.1.2.27.8.8.9 设置驱动 WaitRlc 状态修改为 Pending 状态

3.1.2.27.8.8.9.1 WaitRlcToPending

配置项	WaitRlcToPending
节	[ISUP]
书写格式	WaitRlcToPending=n
取值范围	n=0: 驱动程序使用 WaitRlc 状态（缺省值）； n=1: 驱动程序使用 Pending 状态替代 WaitRlc 状态。
功能描述	本配置项决定驱动程序是否将 WaitRlc 状态修改为 Pending 状态。

3.1.2.27.8.8.10 不使用 SynCTI 提供的 ISUP 协议状态机

3.1.2.27.8.8.10.1 AutoHandleIsup

配置项	AutoHandleIsup
节	[SS7]
书写格式	AutoHandleIsup=b
取值范围	b=0: 否； b=1: 是（缺省值）。
功能描述	设置是否使用 SynCTI 驱动程序提供的 ISUP 协议状态机。

3.1.2.27.8.8.10.2 CircuitReset

配置项	CircuitReset
节	[ISUP]
书写格式	CircuitReset=m
取值范围	m=0: SS7 业务开通后, 电路不发复原消息就进入空闲。 m=1: SS7 业务开通后, 电路发送复原消息, 并进入空闲(缺省值)。这是一个“布尔值”配置项, 不支持 TUP。
功能描述	用于控制 ISUP 业务开通后, 电路是进入空闲状态还是复原状态。

3.1.2.27.8.9 SCCP 协议的高级配置项

3.1.2.27.8.9.1 AutoHandleSccp

配置项	AutoHandleSccp
节	[SS7]
书写格式	AutoHandleSccp=b
取值范围	b=0: 否; b=1: 是(缺省值)。
功能描述	设置是否由 SynCTI 驱动程序处理 SCCP 消息。
注意事项	本配置项要求 SynCTI 驱动程序的版本为 Ver. 4.7.1.7 或更高。

3.1.2.27.8.10 设置 SS7 MSU 输出方式

3.1.2.27.8.10.1 GetMsuOnAutoHandle

配置项	GetMsuOnAutoHandle
节	[SS7]
书写格式	GetMsuOnAutoHandle=b
取值范围	b=0: 自动接续(使用驱动状态机处理用户层协议)时不输出原始的 SS7 MSU, 缺省值; b=1: 可在自动接续(使用驱动状态机处理用户层协议)时获取原始的 SS7 MSU。
功能描述	设置 SS7 MSU 输出方式。若不使用驱动状态机处理用户层协议, 即自动接续关闭时, 驱动输出原始的 SS7 MSU; 若使用驱动状态机处理用户层协议, 即自动接续打开时, 由配置项 GetMsuOnAutoHandle 指定驱动是否输出 SS7 MSU。
注意事项	本配置项要求 SynCTI 驱动程序的版本为 Ver. 5.0.3.0 或更高。Ver. 5.0.3.0 以下版本驱动只能按 GetMsuOnAutoHandle=0 方式运行。

3.1.2.27.8.11 设置 SS7 MTP2 MSU 处理方式

3.1.2.27.8.11.1 AppHandleMtp2Msu

配置项	AppHandleMtp2Msu
节	[SS7]
书写格式	AppHandleMtp2Msu=b
取值范围	b=0: 驱动处理 SS7 MTP2 MSU (缺省值); b=1: 应用程序处理 SS7 MTP2 MSU; b=2: 应用程序处理 SS7 MTP2 MSU 以及 SS7 MTP2 链路命令和状态。
功能描述	设置 SS7 MTP2 MSU 处理方式。
注意事项	本配置项要求 SynCTI 驱动程序的版本为 Ver. 5.1.0.0 或更高。Ver. 5.1.0.0 以下版本驱动只能按 AppHandleMtp2Msu=0 方式运行。Ver5.3.2.1 以上版本驱动开始支持配置 AppHandleMtp2Msu=2。

3.1.2.27.8.12 SS7 高负荷系统相关配置项

3.1.2.27.8.12.1 RefreshWatchDogInSys

配置项	RefreshWatchDogInSys
节	[BoardId=x]
书写格式	RefreshWatchDogInSys=b
取值范围	b=0: 驱动上层刷新看门狗 (缺省值); b=1: SYS 层刷新看门狗。
功能描述	设置 SS7 链路看门狗刷新方式。
注意事项	本配置项要求 SynCTI 驱动程序的版本为 Ver. 5.3.1.5 或更高。当 SS7 系统经常出现本端处理机故障时, 请设置本配置项为 1.

3.1.2.27.8.12.2 ReplySLTAInSys

配置项	ReplySLTAInSys
节	[BoardId=x]
书写格式	ReplySLTAInSys=b

取值范围	b=0: 驱动上层回复 SLTA 消息（缺省值）； b=1: SYS 层回复 SLTA 消息。
功能描述	设置 SLTA 消息回复方式。
注意事项	本配置项要求 SynCTI 驱动程序的版本为 Ver. 5.3.1.5 或更高。当 SS7 系统经常出现对方交换机中断 SS7 信令链路时，请设置本配置项为 1。目前仅支持 SHD 系列的 D 型、E 型卡。.

3.1.2.27.8.13 SS7 协议类型

3.1.2.27.8.13.1 Ss7Type

配置项	Ss7Type
节	[SS7]
书写格式	Ss7Type=b
取值范围	b=0: ITU 协议类型（缺省值）； b=1: ANSI 协议类型。
功能描述	设置 SS7 使用的协议类型。
注意事项	要求 SynCTI Ver. 5.4.0.0 及以上版本。

3.1.2.27.9 ISDN 信令的高级设置

3.1.2.27.9.1 设置接续模式

3.1.2.27.9.1.1 AutoHandleIsdn

配置项	AutoHandleIsdn
节	[ISDN]
书写格式	AutoHandleIsdn=b
取值范围	b=0: 否； b=1: 是（缺省值）。
功能描述	设置是否由驱动程序提供的 ISDN 状态机来处理 ISDN 消息。更多信息请参见第 1 章中“ ISDN 高级编程接口 ”部分内容。
注意事项	本配置项为高级配置，只有当应用程序希望自行处理 ISDN 消息时，才需要设置本配置项。

3.1.2.27.9.2 设置用户侧/网络侧数字中继线的工作参数

3.1.2.27.9.2.1 UserCrcMode

请参见 [NetCrcMode](#)

3.1.2.27.9.2.2 NetCrcMode

配置项	UserCrcMode NetCrcMode
节	[ISDN]
书写格式	UserCrcMode[n]=b NetCrcMode[n]=b
取值范围	n: 用户侧（对于 UserCrcMode）或网络侧（对于 NetCrcMode）数字中继线的编号。 b: CRC 校验功能开关， b=0: 关闭； b=1: 开启。
功能描述	设置是否开启数字中继线上的 CRC 校验功能。UserCrcMode 适用于用户侧，NetCrcMode 适用于网络侧。
注意事项	<ul style="list-style-type: none">● 本配置项为高级配置；● 如果配置项 TotalUserLinker 的设定值(假定为 N)大于 0, UserCrcMode 必须设置 N 次；如果配置项 TotalNetLinker 的设定值(假定为 N)大于 0, NetCrcMode 必须设置 N 次。

3.1.2.27.9.2.3 UserChIdentify

请参见 [NetChIdentify](#)

3.1.2.27.9.2.4 NetChIdentify

配置项	UserChIdentify NetChIdentify
节	[ISDN]
书写格式	UserChIdentify[n]=k NetChIdentify[n]=k

取值范围	n: 用户侧（对于 UserChIdentify）或网络侧（对于 NetChIdentify）数字中继线的编号； k: 通路识别表示方法， k=0x11: 时隙图； k=0x10: 号码(缺省值)。 注意: k 的缺省值取决于 SynCTI 驱动程序的版本：如果版本低于 Ver. 4.7.3.1，k 的缺省值为 0x11；否则为 0x10。
功能描述	设置数字中继线上的通路识别表示方法。UserChIdentify 适用于用户侧，NetChIdentify 适用于网络侧。有关通路识别表示方法的更多信息请参见第 1 章中 “ 通路识别表示方法 ” 部分内容。

3.1.2.27.9.2.5 UserVoiceFormat

请参见 [NetVoiceFormat](#)

3.1.2.27.9.2.6 NetVoiceFormat

配置项	UserVoiceFormat NetVoiceFormat
节	[ISDN]
书写格式	UserVoiceFormat[n]=k NetVoiceFormat[n]=k
取值范围	n: 用户侧（UserVoiceFormat）或网络侧（NetVoiceFormat）数字中继线的编号。 k: 语音编码格式， k=6: A-law (缺省值)； k=7: μ-law。
功能描述	设置数字中继线上语音信道的语音编码格式。 UserVoiceFormat 适用于用户侧，NetVoiceFormat 适用于网络侧。

3.1.2.27.9.2.7 UserRestartTime

请参见 [NetRestartTime](#)

3.1.2.27.9.2.8 NetRestartTime

配置项	UserRestartTime NetRestartTime
节	[ISDN]
书写格式	UserRestartTime=t NetRestartTime=t

取值范围	100≤t≤120，单位为秒，缺省值为 100。
功能描述	设置本端响应对端链路重启命令的时间，即从本端链路收到对端的重启命令开始，到完成重启动作的时间。如果在本配置项设定的时间内，链路未能完成重启，则相关的数字中继线上全部通道将进入不可用状态（即 S_CALL_UNAVAILABLE）。 UserRestartTime 适用于用户侧，NetRestartTime 适用于网络侧。
注意事项	<ul style="list-style-type: none">● 本配置项为高级配置项，大多数情况下无需进行设置；● 本配置项对系统中所有数字中继线都起作用。

3.1.2.27.9.2.9 UserEstablishTime

请参见 [NetEstablishTime](#)

3.1.2.27.9.2.10 NetEstablishTime

配置项	UserEstablishTime NetEstablishTime
节	[ISDN]
书写格式	UserEstablishTime=t NetEstablishTime=t
取值范围	5≤t≤10，单位为秒，缺省值为 5。
功能描述	设置驱动程序尝试将本端与对端建立链路连接的时间间隔。如果不能建立链路连接，在本配置项设置的时间后，驱动程序会自动开始下一次尝试。 UserEstablishTime 适用于用户侧，NetEstablishTime 适用于网络侧。
注意事项	<ul style="list-style-type: none">● 本配置项为高级配置项，大多数情况下无需进行设置；● 本配置项对系统中所有的用户侧或网络侧数字中继线都起作用。

3.1.2.27.9.3 设置去话呼叫的工作参数

3.1.2.27.9.3.1 UserCalledNoSet

请参见 [NetCallingNoSet](#)

3.1.2.27.9.3.2 NetCalledNoSet

请参见 [NetCallingNoSet](#)

3.1.2.27.9.3.3 UserCallingNoSet

请参见 [NetCallingNoSet](#)

3.1.2.27.9.3.4 NetCallingNoSet

配置项	UserCalledNoSet NetCalledNoSet UserCallingNoSet NetCallingNoSet
节	[ISDN]
书写格式	UserCalledNoSet[n]=k NetCalledNoSet[n]=k UserCallingNoSet[n]=g NetCallingNoSet[n]=g
取值范围	n: 用户侧（对于 UserCalledNoSet、UserCallingNoSet）或网络侧（对于 NetCalledNoSet、NetCallingNoSet）数字中继线的编号。 k: 号码类型及编号方案，16 进制形式， k=0xa1: 国内号码（缺省值）; k=0x91: 国际号码; k=0xc1: 用户号码; k=0xb1: 特定号码; k=0x81: 未知。 g: 号码类型及编号方案，16 进制形式， g=0x21: 国内号码（缺省值）; g=0x11: 国际号码; g=0x41: 用户号码; g=0x31: 特定号码; g=0x01: 未知。
功能描述	去话呼叫时，设置 setup 消息中主叫号码或被叫号码的号码类型及编号方案。 UserCalledNoSet、NetCalledNoSet 分别用于设置用户侧、网络侧的被叫号码，UserCallingNoSet、NetCallingNoSet 分别用于设置用户侧、网络侧的主叫号码。
注意事项	<ul style="list-style-type: none">● 本配置项为高级配置；● 如果配置项 TotalUserLinker 的设定值（假定为 N）大于 0，UserCalledNoSet 和 UserCallingNoSet 必须设置 N 次；如果配置项 TotalNetLinker 的设定值（假定为 N）大于 0，NetCalledNoSet 和 NetCallingNoSet 必须设置 N 次。

3.1.2.27.9.3.5 UserNumIsFull

请参见 [NetNumIsFull](#)

3.1.2.27.9.3.6 NetNumIsFull

配置项	UserNumIsFull NetNumIsFull
节	[ISDN]
书写格式	UserNumIsFull=b NetNumIsFull=b
取值范围	b=0: 否 (缺省值); b=1: 是。
功能描述	去话呼叫时, 是否在 SETUP 消息中包含“被叫号码全”参数。 UserNumIsFull 适用于用户侧, NetNumIsFull 适用于网络侧。
注意事项	本配置项为高级配置。

3.1.2.27.9.3.7 UserChPreference

请参见 [NetChPreference](#)

3.1.2.27.9.3.8 NetChPreference

配置项	UserChPreference NetChPreference
节	[ISDN]
书写格式	UserChPreference=b NetChPreference=b
取值范围	b=0: 否 (缺省值); b=1: 是。
功能描述	是否需要优选通路类型。 UserChPreference 适用于用户侧, NetChPreference 适用于网络侧。
注意事项	本配置项为高级配置。

3.1.2.27.9.3.9 UserHighLayerCompatible

请参见 [NetHighLayerCompatible](#)

3.1.2.27.9.3.10 NetHighLayerCompatible

配置项	UserHighLayerCompatible NetHighLayerCompatible
节	[ISDN]
书写格式	UserHighLayerCompatible=b NetHighLayerCompatible=b
取值范围	b=0: 否; b=1: 是（缺省值）。
功能描述	设置 SETUP 消息中是否包含“高层兼容性”字段。 UserHighLayerCompatible 适用于用户侧，NetHighLayerCompatible 适用于网络侧。
注意事项	本配置项为高级配置。

3.1.2.27.9.3.11 UserLowLayerCompatible

请参见 [NetLowLayerCompatible](#)

3.1.2.27.9.3.12 NetLowLayerCompatible

配置项	UserLowLayerCompatible NetLowLayerCompatible
节	[ISDN]
书写格式	UserLowLayerCompatible=b NetLowLayerCompatible=b
取值范围	b=0: 否; b=1: 是（缺省值）。
功能描述	设置 SETUP 消息中是否包含“低层兼容性”字段。 UserLowLayerCompatible 适用于用户侧，NetLowLayerCompatible 适用于网络侧。
注意事项	本配置项为高级配置。

3.1.2.27.9.3.13 UserDialTime

请参见 [NetDialTime](#)

3.1.2.27.9.3.14 NetDialTime

配置项	UserDialTime NetDialTime
节	[ISDN]
书写格式	UserDialTime=t NetDialTime=t
取值范围	t: 等待时间, 单位为秒, 缺省值为 60 秒。
功能描述	当应用程序在某个 ISDN 通道上调用函数 SsmSearchIdleCallOutCh 后, 驱动程序会启动一个定时器, 应用程序应当在定时器溢出前开始去话呼叫。如果定时器溢出, 驱动程序将自动回收该通道, 将其迁移到 S_CALL_STANDBY 状态。本配置项用于设置此定时器的值。更多信息请参见第 1 章中 “ ISDN 通道的状态转移图 ” 部分内容。 UserDialTime 适用于用户侧, NetDialTime 适用于网络侧。
注意事项	本配置项为高级配置。

3.1.2.27.9.3.15 TransferCapability

配置项	TransferCapability
节	[ISDN]
书写格式	TransferCapability=n
取值范围	n=0: 语音, 默认值; n=1: 3.1k 音频。
功能描述	设置信令消息中“负载能力”字段的内容。
注意事项	本配置项为高级配置。

3.1.2.27.9.3.16 PresentNumber

配置项	PresentNumber
节	[ISDN]
书写格式	PresentNumber =n
取值范围	n=0: 不允许显示号码。 n=1: 允许显示号码, 默认值。
功能描述	设置信令消息中主叫号码是否“允许显示”字段的内容。

注意事项	本配置项为高级配置。
------	------------

3.1.2.27.9.3.17 UserT303

请参见 [NetT303](#)

3.1.2.27.9.3.18 NetT303

配置项	UserT303 NetT303
节	[ISDN]
书写格式	UserT303=t NetT303=t
取值范围	t: 定时器的时长，单位为秒，缺省值为 4 秒。
功能描述	设置定时器 T303。UserT303 适用于用户侧，NetT303 适用于网络侧。有关 T303 的更多信息请参见第 1 章中“ ISDN 通道的状态转移图 ”部分内容。

3.1.2.27.9.3.19 UserT304

请参见 [NetT304](#)

3.1.2.27.9.3.20 NetT304

配置项	UserT304 NetT304
节	[ISDN]
书写格式	UserT304=t NetT304=t
取值范围	t: 定时器的时长，单位为秒，缺省值为 30 秒。
功能描述	设置定时器 T304。UserT304 适用于用户侧，NetT304 适用于网络侧。有关 T304 的更多信息请参见第 1 章中“ ISDN 通道的状态转移图 ”部分内容。

3.1.2.27.9.3.21 UserWaitAfterCallProceeding

配置项	UserWaitAfterCallProceeding
节	[ISDN]
书写格式	UserWaitAfterCallProceeding=n

取值范围	n: 等待时间, 缺省值为 15s。
功能描述	在去话呼叫过程中, 本端等待对端发送确认消息时的最大等待时间, 如果超时后仍没收到对端的确认消息, 本端将主动拆线。
注意事项	本配置项为高级配置, 只适用于用户侧去话呼叫。

3.1.2.27.9.3.22 NetWaitAfterCallProceeding

配置项	NetWaitAfterCallProceeding
节	[ISDN]
书写格式	NetWaitAfterCallProceeding=n
取值范围	n: 等待时间, 缺省值为 20s。
功能描述	在去话呼叫过程中, 本端等待对端发送确认消息时的最大等待时间, 如果超时后仍没收到对端的确认消息, 本端将主动拆线。
注意事项	本配置项为高级配置, 只适用于网络侧去话呼叫。

3.1.2.27.9.3.23 UserTxCallingPartyNum

配置项	UserTxCallingPartyNum
节	[ISDN]
书写格式	UserTxCallingPartyNum = b
取值范围	b=0: 呼出时 isdn user setup 消息不发送 6c 单元(主叫用户号码); b=1: 呼出时 isdn user setup 消息发送 6c 单元(主叫用户号码)(缺省值)。
功能描述	设置呼出时 isdn user setup 消息是否发送 6c 单元(主叫用户号码)。

3.1.2.27.9.3.24 NetTxCallingPartyNum

配置项	NetTxCallingPartyNum
节	[ISDN]
书写格式	NetTxCallingPartyNum = b
取值范围	b=0: 呼出时 isdn net setup 消息不发送 6c 单元(主叫用户号码); b=1: 呼出时 isdn net setup 消息发送 6c 单元(主叫用户号码)(缺省值)。
功能描述	设置呼出时 isdn net setup 消息是否发送 6c 单元(主叫用户号码)。

3.1.2.27.9.4 设置来话呼叫的工作参数

3.1.2.27.9.4.1 UserSideDefaultAckTimer

请参见 [NetSideDefaultAck](#)

3.1.2.27.9.4.2 UserSideDefaultAck

请参见 [NetSideDefaultAck](#)

3.1.2.27.9.4.3 NetSideDefaultAckTimer

请参见 [NetSideDefaultAck](#)

3.1.2.27.9.4.4 NetSideDefaultAck

配置项	UserSideDefaultAckTimer UserSideDefaultAck NetSideDefaultAckTimer NetSideDefaultAck																		
节	[ISDN]																		
书写格式	UserSideDefaultAckTimer=t UserSideDefaultAck=k NetSideDefaultAckTimer=t NetSideDefaultAck=k																		
取值范围	t: 定时器的设定值, 单位为秒, t<120, 缺省值为 20。 k: 设置定时器溢出后驱动程序的动作, 其取值及其含义如下表所示: <table border="1"><thead><tr><th>K</th><th>含义</th><th>驱动程序动作</th></tr></thead><tbody><tr><td>1</td><td>被叫用户空闲</td><td>发送 ALERTING 消息</td></tr><tr><td>2</td><td>被叫用户忙</td><td>发送 DISCONNECT 消息, 原因=被叫用户忙</td></tr><tr><td>4</td><td>呼叫拒绝</td><td>发送 DISCONNECT 消息, 原因=呼叫拒绝</td></tr><tr><td>5</td><td>无应答</td><td>发送 DISCONNECT 消息, 原因=被叫用户无应答</td></tr><tr><td>其它</td><td>备用</td><td></td></tr></tbody></table> k 的缺省值为 1。	K	含义	驱动程序动作	1	被叫用户空闲	发送 ALERTING 消息	2	被叫用户忙	发送 DISCONNECT 消息, 原因=被叫用户忙	4	呼叫拒绝	发送 DISCONNECT 消息, 原因=呼叫拒绝	5	无应答	发送 DISCONNECT 消息, 原因=被叫用户无应答	其它	备用	
K	含义	驱动程序动作																	
1	被叫用户空闲	发送 ALERTING 消息																	
2	被叫用户忙	发送 DISCONNECT 消息, 原因=被叫用户忙																	
4	呼叫拒绝	发送 DISCONNECT 消息, 原因=呼叫拒绝																	
5	无应答	发送 DISCONNECT 消息, 原因=被叫用户无应答																	
其它	备用																		
功能描述	来话呼叫时, 当驱动程序完成全部号码的接收后, 本配置项设置是否由驱动程序自动发送应答消息。 更多信息请参见第 1 章中 “ ISDN 通道的状态转移图 ” 部分内容。 UserSideDefaultAckTimer、UserSideDefaultAck 适用于用户侧, NetSideDefaultAckTimer、NetSideDefaultAck 适用于网络侧。																		
注意事项	本配置项为高级配置。																		

3.1.2.27.9.4.5 UserIsReceivePhoNum

请参见 [NetIsReceivePhoNum](#)

3.1.2.27.9.4.6 NetIsReceivePhoNum

配置项	UserIsReceivePhoNum NetIsReceivePhoNum
节	[ISDN]
书写格式	UserIsReceivePhoNum=b NetIsReceivePhoNum=b
取值范围	b=0: 汇接局方式; b=1: 端局方式 (缺省值)。
功能描述	设置来话呼叫时, 驱动程序的工作方式。更多信息请参见第 1 章中“ 端局方式与汇接局方式 ”部分内容。 UserIsReceivePhoNum 适用于用户侧, NetIsReceivePhoNum 适用于网络侧。
注意事项	<ul style="list-style-type: none">如果对端交换机为阿尔卡特或 AVAYA, 通常应将本配置项设置为 0;较早版本的驱动程序支持的配置项 UserTieStationMode 具有相同功能, 但只支持用户侧, 可以用 UserIsReceivePhoNum 来替代。

3.1.2.27.9.4.7 UserIsSendChIdentify

请参见 [NetIsSendChIdentify](#)

3.1.2.27.9.4.8 NetIsSendChIdentify

配置项	UserIsSendChIdentify NetIsSendChIdentify
节	[ISDN]
书写格式	UserIsSendChIdentify=b NetIsSendChIdentify=b
取值范围	b=0: 否; b=1: 是 (缺省值)。
功能描述	来话呼叫时, 当本端收到对端交换机的 SETUP 消息后, 在相应的应答信号中 (如 CALL PROCEEDING、ALERT 等) 是否包含通路识别信息。 UserIsSendChIdentify 适用于用户侧, NetIsSendChIdentify 适用于网络侧。

3.1.2.27.9.4.9 UserT302

请参见 [NetT302](#)

3.1.2.27.9.4.10 NetT302

配置项	UserT302 NetT302
节	[ISDN]
书写格式	UserT302=t NetT302=t
取值范围	t: 定时器的时长, 单位为秒, 缺省值为 15 秒。
功能描述	设置定时器 T302。UserT302 适用于用户侧, NetT302 适用于网络侧。有关 T302 的更多信息请参见第 1 章中 “ ISDN 通道的状态转移图 ” 部分内容。

3.1.2.27.9.4.11 UserT313

配置项	UserT313
节	[ISDN]
书写格式	UserT313=t
取值范围	t: 定时器的时长, 单位为秒, 缺省值为 4 秒。
功能描述	设置定时器 T313。有关 T313 的更多信息请参见第 1 章中 “ ISDN 通道的状态转移图 ” 部分内容。
注意事项	本配置项只适用于用户侧。

3.1.2.27.9.4.12 ProgressExt

配置项	ProgressExt
节	[ISDN]
书写格式	ProgressExt=n

	<p>n: 进展表示语内容。缺省值为 0, 表明不包含进展表示语。</p> <p>n 的值用 8 位表示, 各 bit 的含义如下:</p> <p>Bit8=1</p> <p>Bit7, Bit 6=00----ITU-T 编码标准</p> <p>Bit5=0----备用</p> <p>Bit4~ Bit1=</p> <p>0000----用户</p> <p>0001----为本地用户提供服务的专用网</p> <p>0010----为本地用户提供服务的公用网</p> <p>0011----转接网络</p> <p>0100----为远端用户提供服务的公用网</p> <p>0101----为远端用户提供服务的专用网</p> <p>1010----互通点以外的网络</p>
功能描述	在来话呼叫过程中, 当 ProgressExt 的值不为 0 且本端有发送 ALERT 或 SETUP 消息时, 在 ALERT 或 SETUP 消息里将包含进展表示语的内容。
注意事项	本配置项为高级配置, 只适用来话呼叫。

3.1.2.27.9.5 设置定时器

3.1.2.27.9.5.1 UserT305

请参见 [NetT306](#)

3.1.2.27.9.5.2 NetT305

请参见 [NetT306](#)

3.1.2.27.9.5.3 NetT306

配置项	UserT305 NetT305 NetT306
节	[ISDN]
书写格式	UserT305=t NetT305=t NetT306=t
取值范围	t: 定时器的时长, 单位为秒, 缺省值为 30 秒。

功能描述	设置定时器 T305 和 T306。UserT305 适用于用户侧，NetT305、NetT306 适用于网络侧。 ◆ 对于用户侧，当本端向对端交换机发送 DISCONNECT 消息后，驱动程序会启动 T305；当本端收到对端交换机的 RELEASE 或 DISCONNECT 消息后，驱动程序会自动关闭 T305。 ◆ 对于网络侧，当本端向对端交换机发送 DISCONNECT 消息后，如果 DISCONNECT 消息中包含了进展表示语字段，并且进展表示语字段的值等于 8，驱动程序会启动 T306，否则启动 T305。当驱动程序收到对端交换机的 RELEASE 或 DISCONNECT 消息后，会自动关闭 T305 或 T306。 如果 T305 或 T306 溢出，驱动程序会发送 RELEASE 消息。
------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3.1.2.27.9.5.4 UserT308

请参见 [NetT308](#)

3.1.2.27.9.5.5 NetT308

配置项	UserT308 NetT308
节	[ISDN]
书写格式	UserT308=t NetT308=t
取值范围	t：定时器的时长，单位为秒，缺省值为 4 秒。
功能描述	设置定时器 T308。UserT308 适用于用户侧，NetT308 适用于网络侧。 当本端向对端交换机发送 RELEASE 消息后，驱动程序会启动 T308；当本端收到对端交换机的 RELEASE 或 RELEASE COMPLETE 消息后，驱动程序会自动关闭 T308。如果 T308 溢出，驱动程序会重发 RELEASE 消息。

3.1.2.27.9.6 设置其它参数

3.1.2.27.9.6.1 WaitHangupTime

配置项	WaitHangupTime
节	[ISDN]
书写格式	WaitHangupTime=b
取值范围	20< b <10000，缺省值为 1600，单位：毫秒。
功能描述	设置 ISDN 挂起等待时间。
注意事项	本配置项为高级配置。

3.1.2.27.9.6.2 UserStatusReason

请参见 [NetStatusReason](#)

3.1.2.27.9.6.3 NetStatusReason

配置项	UserStatusReason NetStatusReason
节	[ISDN]
书写格式	UserStatusReason=m NetStatusReason=m
取值范围	m=1: 3字节格式(缺省值); m=0: 2字节格式。
功能描述	当本端收到对端交换机发送的 STATUS ENQUIRY 消息(状态查询)时,驱动程序会自动应答 STATUS 消息。本配置项设置 STATUS 消息中的原因值字段是采用 2 字节格式还是 3 字节格式。具体采用哪一种格式,需要根据对端交换机的参数进行设置。UserStatusReason 适用于用户侧,NetStatusReason 适用于网络侧。
注意事项	<ul style="list-style-type: none">● 本配置项为高级选项;● 本配置项要求 SynCTI 驱动程序的版本为 Ver. 4.7.1.8 或更高。

3.1.2.27.9.6.4 EnAutoAlert02

请参见 [EnAutoAlert03](#)

3.1.2.27.9.6.5 EnAutoAlert03

配置项	EnAutoAlert02 EnAutoAlert03
节	[ISDN]
书写格式	EnAutoAlert02=m EnAutoAlert03=m
取值范围	m=0: 禁用该功能(缺省值); m=1: 启动该功能。
功能描述	当配置项 EnAutoAlert02 为 1 时,表示收到 02(CALL PROCEEDING) 消息,进展指示为 8 或 1 时,自动进入提醒状态。 当配置项 EnAutoAlert03 为 1 时,表示收到 03(PROGRESS) 消息,进展指示为 8 或 1 时,自动进入提醒状态。
注意事项	<ul style="list-style-type: none">● 本配置项为高级配置项;● 本配置项要求 SynCTI 驱动程序的版本为 Ver. 5.2.0.0 或更高。

3.1.2.27.9.6.6 GetRedirectionReason

配置项	GetRedirectionReason
节	[ISDN]
书写格式	GetRedirectionReason=m
取值范围	m=0: 获取信息单元 Facility (0x1c) 的转呼原因 (缺省值); m=1: 获取信息单元 Redirecting Number (0x74) 的转呼原因。
功能描述	当配置项 GetRedirectionReason 为 1 时, 表示 SsmGetRedirectionInfReason 获取到信息单元 0x74 字段的呼转原因值。 当配置项 GetRedirectionReason 为 0 时, 表示 SsmGetRedirectionInfReason 获取到信息单元 0x1c 字段的呼转原因值。
注意事项	<ul style="list-style-type: none">本配置项为高级配置项;本配置项要求 SynCTI 驱动程序的版本为 Ver. 5.3.2.3 或更高。

3.1.2.27.9.7 NetCallingNoSet

配置项	UserRedirectingNoSet NetRedirectingNoSet
节	[ISDN]
书写格式	UserRedirectingNoSet[n]=k NetRedirectingNoSet[n]=k
取值范围	n: 用户侧 (对于 UserCalledNoSet、UserCallingNoSet) 或网络侧 (对于 NetCalledNoSet、NetCallingNoSet) 数字中继线的编号。 k: 号码类型及编号方案, 16 进制形式 , k=0x21: 国内号码 (缺省值); k=0x11: 国际号码; k=0x41: 用户号码; k=0x31: 特定号码; k=0x01: 未知。
功能描述	去话呼叫时, 设置 setup 消息中主叫号码或被叫号码的号码类型及编号方案。 分别用于修改用户侧、网络侧 0x74 字段的号码属性。
注意事项	本配置项为高级配置。

3.1.2.27.10 设置 DSP 芯片的工作方式

3.1.2.27.10.1 LoadFskBin

配置项	LoadFskBin
节	[BoardId=x]
书写格式	LoadFskBin=m
取值范围	<p>m=0: 增强的回波抵消器方式。语音通道不再具有录音功能和 FSK 收发器功能，闲置出来的 DSP 资源被用来增强回波抵消器的效果；</p> <p>m=1: 正常运行方式（缺省值）；</p> <p>m=2: 16 时隙录音方式。驱动程序将 16 时隙上的原始信令数据通过录音的方式直接传递给应用程序进行处理。当 m=1 时，板卡上的每片 DSP 芯片处理 15 个语音时隙，每条数字中继线上 30 个时隙分别由 2 片 DSP 芯片进行处理，一片处理 TS₁~TS₁₅，另外一片处理 TS₁₇~TS₃₁。当 m=2 时，原来处理时隙 TS₁₇~TS₃₁ 的 DSP 芯片转而处理时隙 TS₁₆~TS₃₀，TS₃₁ 上的语音数据将被抛弃。注意：通道的物理编号 15 所对应的时隙不再是 TS₁₇，而是 TS₁₆，其余通道依此类推。</p>
功能描述	选择 SHD 系列板卡上 DSP 芯片的工作程序。
注意事项	本配置项是高级配置项，只适用于 30 通道、60 通道和 120 通道规格的 SHD 系列 A 型板卡。

3.1.2.27.11 设置信号录制模式

3.1.2.27.11.1 ShdDEToneRec

配置项	ShdDEToneRec
节	[BoardId=x]
书写格式	ShdDEToneRec=b
取值范围	b=0: 否（缺省值）； b=1: 是。
功能描述	设置 SHD 数字卡 D、E 系列板卡发送的信号音和 DTMF 信号是否进入录音缓冲区。
注意事项	本配置只适用于 SHD 系列 D 型和 E 型板卡。

3.1.2.27.12 设置底噪过滤门限

3.1.2.27.12.1 NoiseFilteringMinGate

配置项	NoiseFilteringMinGate
节	[BoardId=x]
书写格式	NoiseFilteringMinGate=b
取值范围	b=0: 关闭 (缺省值); b=n (0<n≤1000): 能量门限值 (如需配置, 建议设置为 300)。
功能描述	设置底噪过滤门限值。
注意事项	<ul style="list-style-type: none">适用于 SHD 系列 D 和 E 型板卡;当该配置项的值在 1~1000 的范围内时, 可能出现无法发送传真的情况;若配置项的值不在 1~1000 的范围内时, 则认为该配置关闭。

3.1.2.27.13 设置指定 PCM 工作状态

3.1.2.27.13.1 PcmPowerDown

配置项	PcmPowerDown
节	[BoardId=x]
书写格式	PcmPowerDown[i]=d
取值范围	i 表示数字中继线的逻辑编号, 取值范围: 0~N-1 (N 为配置项 TotalPcm 的设定值, 可以通过函数 SsmGetMaxPcm 获取) d=0: 表示正常工作模式 (默认值) ; d=1: 表示休息模式 (类似于该 PCM 未接线) 。
功能描述	设置指定的 PCM 进行工作状态, 可以将其设置为工作或者休息状态。
注意事项	<ul style="list-style-type: none">适用于 SHD 系列 D 和 E 型板卡。

3.1.2.28 SHN 系列板卡常用配置项(CTI 系列)

3.1.2.28.1 SHN 高级通用项配置

3.1.2.28.1.1 EnableSIPStun

请参见 [HeartInterval](#)

3.1.2.28.1.2 SIPStunServerIP

请参见 [HeartInterval](#)

3.1.2.28.1.3 LogLevel

请参见 [HeartInterval](#)

3.1.2.28.1.4LogFile

请参见 [HeartInterval](#)

3.1.2.28.1.5 EventThreadNum

请参见 [HeartInterval](#)

3.1.2.28.1.6 SipCallCheckInterval

请参见 [HeartInterval](#)

3.1.2.28.1.7 SendOptionsOutCallIP

请参见 [HeartInterval](#)

3.1.2.28.1.8 SendOptionsOutCallInterval

请参见 [HeartInterval](#)

3.1.2.28.1.9 RemoteCrashCheckInterval

请参见 [HeartInterval](#)

3.1.2.28.1.10 IgnoreUserName

请参见 [HeartInterval](#)

3.1.2.28.1.11 AutoDetectRemoteRTPAddress

请参见 [HeartInterval](#)

3.1.2.28.1.12 HeartInterval

注：5.1.0.0 版本前该配置项名为 UdpHeartTime。

配置项	EnableSIPStun SIPStunServerIP StunMaskAddr LogLevel EventThreadNum SipCallCheckInterval SendOptionsOutCallIP SendOptionsOutCallInterval RemoteCrashCheckInterval IgnoreUserName AutoDetectRemoteRTPAddress HeartInterval
节	[SIP]
书写格式	EnableSIPStun=p SIPStunServerIP=y StunMaskAddr=d LogLevel=m EventThreadNum=j SipCallCheckInterval=n SendOptionsOutCallIP=k SendOptionsOutCallInterval=q RemoteCrashCheckInterval=i IgnoreUserName=l AutoDetectRemoteRTPAddress=z HeartInterval=x

	<p>p: p=0, 关闭 SIP 的 stun 功能; p=1, 开启 SIP 的 stun 功能。</p> <p>y: SIP 的 stun 服务器地址。</p> <p>d: IP 卡所在网络的网络地址。</p> <p>m: 设置输出的日志等级, 具体为:</p> <ul style="list-style-type: none"> m=1: SEGMENT, 严重错误等级, 默认值; m=2: FATAL, 协议栈消息级错误; m=3: ERROR, 逻辑错误; m=4: WARNING, 逻辑警告; m=5: INFO1, 逻辑信息; m=6: INFO2, SIP 消息信息; m=7: INFO3, RTP 消息信息; m=8: INFO4, SIP 原始消息。 <p>j: 设置捕获 SIP 协议事件的线程池大小, 默认值为 3, 范围为: 0<j<6。</p> <p>取值范围</p> <p>n: 设置检测对方异常挂机的时间间隔, 单位秒, n=0: 表示不使用该功能。</p> <p>k: 非注册情况下自动发送信令 options 的目的 IP 地址。</p> <p>q: 发送信令 options 的时间间隔, 单位秒, q=0, 表示关闭该功能。</p> <p>i: 设置通过 RTP 来检测对方异常挂机的时间间隔, 单位秒, i=0, 表示不使用该功能。注: 只有当 n=0 时, 该功能才有效。C 型 IP 卡可以设置的最长时间是 10s</p> <p>l: 设置是否开启忽略对用户名的判断功能。l=0 关闭 (默认值); l=1 开启。此功能开启时用于 SIP 通道注册到服务器上且作为呼入端时, 只通过信令中的域名查找对应的空闲通道, 而忽略对用户名的判断。</p> <p>z: 设置是否开启 RTP 远端地址自适应功能。z=0, 关闭; z=1, 开启 (默认值)。此功能用于当远端发送的信令消息中所携带的 RTP 接收地址或端口与实际情况不符时, 自动更新其为实际的 RTP 接收地址或端口, 提高板卡 RTP 发送的自适应能力; 480C 板卡实现了以下功能: 以 OCT 收到的第一个 RTP 的源地址作为发送 RTP 包的目的地址。</p> <p>x: 设置 UDP 端口保持心跳的时间。取值范围为: 0≤x≤1000, 单位为 ms。x=0: 不使用心跳功能 (默认值)。</p>
功能描述	设置其他参数。
注意事项	只有当配置项 ProtocolType 设置为 1 时, 本配置项才有效。

3.1.2.28.1.13 UseRport

配置项	UseRport
节	[SIP]
书写格式	UseRport=z
取值范围	z=1, Invite 消息中携带 rport 字段 (默认值); z=0, Invite 消息中不携带 rport 字段。
功能描述	设置 Invite 消息中是否携带 rport 字段。

3.1.2.28.1.14 SipTransportProtocol

配置项	SipTransportProtocol
节	[SIP]
书写格式	SipTransportProtocol =z
取值范围	z=1, 表示 SIP 协议使用 TCP 进行传输; z=0, 表示 SIP 协议使用 UDP 进行传输（默认值）。
功能描述	设置 SIP 协议使用 TCP 还是 UDP。

3.1.2.28.1.15 RetransmitLost200OK

配置项	RetransmitLost200OK
节	[SIP]
书写格式	RetransmitLost200OK=z
取值范围	z=1, 板卡发送 200OK 消息以后不需要等待 ACK 消息; z=0, 板卡发送 200OK 消息以后需要等待 ACK 消息（默认值）。
功能描述	设置板卡发送 200OK 消息以后是否需要等待收到 ACK 消息后才建立通话。

3.1.2.28.1.16 RegisterTimeOutSpace

配置项	RegisterTimeOutSpace
节	[SIP]
书写格式	RegisterTimeOutSpace=q
取值范围	q 为时间间隔, q>0, 单位为秒, 默认值为 10。
功能描述	不同注册消息的时间间隔。
注意事项	不同注册消息的时间间隔指的是注册超时以后自动产生新的注册消息的间隔时间。

3.1.2.28.1.17 EnableSetUsername

配置项	EnableSetUsername
节	[SIP]

书写格式	EnableSetUsername=n
取值范围	n=1, 表示注册后可以对 username 进行修改; n=0, 表示注册后不能对 username 进行修改（默认值）。
功能描述	设置 SHN 卡在注册后能否对 username 进行修改。
注意事项	该配置项只能在一些特定的 SIP 服务器上才能使用，一般服务器配置不起作用。

3.1.2.28.1.18 UserName

请参见 [SipTotalMultiRegNum](#)

3.1.2.28.1.19 RegPassword

请参见 [SipTotalMultiRegNum](#)

3.1.2.28.1.20 Domain

请参见 [SipTotalMultiRegNum](#)

3.1.2.28.1.21 RegRealm

请参见 [SipTotalMultiRegNum](#)

3.1.2.28.1.22 RegExpires

请参见 [SipTotalMultiRegNum](#)

3.1.2.28.1.23 RegStartCh

请参见 [SipTotalMultiRegNum](#)

3.1.2.28.1.24 RegEndCh

请参见 [SipTotalMultiRegNum](#)

3.1.2.28.1.25 SipTotalMultiRegNum

配置项	UserName RegPassword Domain RegRealm RegExpires RegStartCh RegEndCh RegOutBoundProxyAddr RegAuthUserName RegDisplayName SipTotalMultiRegNum
节	[SIP]
书写格式	UserName[m]=s RegPassword[m]=j Domain[m]=x RegRealm[m]=r RegExpires[m]=q RegStartch[m]=o RegEndCh[m]=e RegOutBoundProxyAddr[m]=b RegAuthUserName[m]=a RegDisplayName[m]=d SipTotalMultiRegNum=n

取值范围	<p>m: 表示执行多通道操作的编号，必须从 0 开始，取值范围 $0 \leq m \leq n$(n 为配置项中 SipTotalMultiRegNum 的设定值)。</p> <p>s: 用户名，最大长度为 50 的字符串。 当 $m \neq 0$ 时，s 表示注册用户名；当 $m=0$ 时，s 表示 SIP URI 中 username 字段的内容。</p> <p>j: 注册用户密码。 当 $m \neq 0$ 时，j 有效，表示注册用户密码。</p> <p>x: 注册服务器。 当 $m \neq 0$ 时，x 有效，表示注册服务器的地址。格式为 host[:port]。</p> <p>r: 服务器别名(一般不用填)。</p> <p>q: 多通道操作的注册有效期。表示板卡每隔多长时间向注册服务器刷新注册消息。默认为 3600s，最小值为 10s。</p> <p>o: 此次多通道操作的起始通道号。</p> <p>e: 此次多通道操作的结束通道号，当 o 等于 e 时，此次多通道注册操作等同于单通道注册操作。</p> <p>b: 多通道注册的外绑地址，用于 IMS 网络环境。配置以后，注册消息和 invite 消息都会发送到外绑地址。</p> <p>a: 多通道注册的认证用户名，用来设置认证信息中的认证用户名字段。</p> <p>d: 多通道注册的显示用户名，用来设置注册消息中的显示用户名。</p> <p>n: 表示执行多通道操作的次数。</p>
功能描述	通过配置项来执行多组多通道注册操作，即相当于执行 n 次的 SsmSipRegister 操作。
注意事项	只有当配置项 SipTotalMultiRegNum 的设置值大于 0 时，本节中其他配置项才有效。

使用范例	<pre>假设系统中安装了一块 SHN-60B-CT/PCI+板卡，需要注册的通道是：0~19 通道（以 1001 的用户名注册到 192.168.1.100:5200 上）、20~39 通道（以 1002 的用户名注册到 192.168.1.101:5200 上）、40~59 通道（IMS 网络以 1001 的用户名注册到 ims.test.com 服务器上，外绑地址为 192.168.1.102:5200，认证用户名为+1001@ims.test.com）。 [SIP] //在 SIP 字段下 SipTotalMultiRegNum = 3 //表示执行多通道操作的次数 //以下三组配置项是分别三次多通道注册的参数，各个配置项所表示的内容可以参考板卡注册的配置项 //第 1 组多通道注册的相关配置项 Domain[0]=192.168.1.100:5200 RegExpires[0]=3600 UserName[0]=1001 RegPassword[0]=1001 RegRealm[0]= RegStartCh[0]=0 //第 1 次多通道操作的起始通道，这个通道值按整个程序中所在的通道计算 RegEndCh[0]=19 //第 1 次多通道操作的结束通道，这个通道值按整个程序中所在的通道计算 //第 2 组多通道注册的相关配置项 Domain[1]=192.168.1.101:5200 RegExpires[1]=3600 UserName[1]=1002 RegPassword[1]=1002 RegRealm[1]= RegStartCh[1]=20 RegEndCh[1]=39 //第 3 组多通道注册的相关配置项 Domain[2]= ims.test.com RegExpires[2]=3600 UserName[2]=1001 RegPassword[2]=1001 RegRealm[2]= RegStartCh[2]=40 RegEndCh[2]=59 RegOutBoundProxyAddr[0]= 192.168.1.102:5200 RegAuthUserName[0]= +1001@ims.test.com RegDisplayName[0]=1001 (可选)</pre>
------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3.1.2.28.1.26 UseGroupByIp

配置项	UseGroupByIp
节	[SIP]
书写格式	UseGroupByIp=z
取值范围	z=1，开启 SIP Trunk 功能； z=0，关闭 SIP Trunk 功能（默认值）。
功能描述	开启 SIP Trunk 功能后，使用多通道注册函数或单通道注册函数将通道和 IP 地址绑定，当板卡做被叫时，可以通过主叫服务器 IP 地址搜索到通道。

3.1.2.28.1.27 UseSipKB

配置项	UseSipKB
-----	-----------------

节	[SIP]
书写格式	UseSipKB=z
取值范围	$z=2$, 板卡作为注册服务器, 注册账号呼入, 应用程序自行回复 180 消息; 非注册账号呼入, 驱动自行回复 180 消息; $z=1$, 应用程序自行回复 180 消息; $z=0$, 驱动自行回复 180 消息(默认值)。
功能描述	设置由驱动还是应用程序回复 180 消息。
注意事项	<ul style="list-style-type: none">如果本配置项设置为 1, 通道收到 invite 消息后会进入挂起状态;如果本配置项设置为 2, 只有板卡作为注册服务器, 注册账号呼入, 通道才会进入挂起状态, 其他非注册服务器上的注册账号呼入, 通道进入振铃状态;如果本配置项设置为 1, 通道收到 invite 消息后会进入挂起状态驱动配置了注册服务器功能时, 即 SipRegMode=1 或者 2, 如果本配置项为默认值 0, 那么驱动内部将其自动设置成 2, 否则影响注册服务器的功能。

3.1.2.28.1.28 SipProxyAddr

配置项	SipProxyAddr
节	[SIP]
书写格式	SipProxyAddr=z
取值范围	$z=a.b.c.d$, SIP 代理服务器的地址, 默认值为 0.0.0.0, 表示不使用 SIP 代理服务器。
功能描述	设置 SIP 代理服务器的地址。

3.1.2.28.1.29 SipProxyPort

配置项	SipProxyPort
节	[SIP]
书写格式	SipProxyPort=n
取值范围	$0 \leq n \leq 65535$, SIP 代理服务器的端口, 默认值为 5060。
功能描述	设置 SIP 代理服务器的端口。

3.1.2.28.1.30 ims

配置项	ims
-----	------------

节	[SIP]
书写格式	ims=b
取值范围	b=0, 对方 SIP 服务器不使用 ims 网络（默认值）; b=1, 对方 SIP 服务器使用 ims 网络。
功能描述	设置对方 SIP 服务器是否使用 ims 网络。如果对方使用 ims 网络，需要设置本配置项为 1，否则会注册不上。

3.1.2.28.1.31 SipNewRegisterMode

配置项	SipNewRegisterMode
节	[SIP]
书写格式	SipNewRegisterMode=b
取值范围	b=0, 通道和注册信息绑定（默认值）; b=1, 通道和注册信息分离。
功能描述	设置是否绑定通道和注册信息。

3.1.2.28.1.32 SipBuildBusyReplaceForbidden

配置项	SipBuildBusyReplaceForbidden
节	[SIP]
书写格式	SipBuildBusyReplaceForbidden=b
取值范围	b=0, 使用 403、415 消息（默认值）; b=1, 使用 486 代替 403 或 415 消息。
功能描述	设置是否使用 486 代替 403 或 415 消息。

3.1.2.28.1.33 SipFindRingChFromPreRingCh

配置项	SipFindRingChFromPreRingCh
节	[SIP]
书写格式	SipFindRingChFromPreRingCh=b
取值范围	b=0, 从 0 通道开始从前往后依次查询振铃通道（默认值）; b=1, 从前一次振铃通道的下一个通道开始从前往后依次查询振铃通道。

功能描述	设置是否从前一次振铃的通道的下一个通道开始查询振铃通道。
------	------------------------------

3.1.2.28.1.34 SipDomain

配置项	SipDomain
节	[SIP]
书写格式	SipDomain=b
取值范围	b=0, 查询通道注册情况时不解析注册服务器域名; b=1, 查询通道注册情况时解析注册服务器域名（默认值）。
功能描述	查询通道注册情况时是否解析注册服务器域名。

3.1.2.28.1.35 SipSearchChInRegisterChannel

配置项	SipSearchChInRegisterChannel
节	[SIP]
书写格式	SipSearchChInRegisterChannel=b
取值范围	b=0, 点对点呼入时只在非注册通道上查询空闲通道（默认值）; b=1, 点对点呼入时在所有通道（已注册和非注册）上查询空闲通道。
功能描述	设置点对点呼入时是否在已经注册的通道上查询空闲通道。

3.1.2.28.1.36 SipRegNumberAreaLen

配置项	SipRegNumberAreaLen
节	[SIP]
书写格式	SipRegNumberAreaLen=z
取值范围	0<z ≤20, 表示注册号码前缀位数; z=0, 表示注册号码不使用前缀（默认值）。
功能描述	设置注册号码前缀。
注意事项	本配置用于兼容注册时的号码的前缀和实际呼入中的号码的前缀有出入的情况。

3.1.2.28.1.37 SipAddRTPChkSum

配置项	SipAddRTPChkSum
节	[SIP]
书写格式	SipAddRTPChkSum =b
取值范围	b=0, RTP 数据包不带校验和（默认值）; b=1, RTP 数据包带校验和。
功能描述	设置 B 型 IP 卡的 RTP 数据包是否带校验和。

3.1.2.28.1.38 StunMaskAddr

请参见 [HeartInterval](#)

3.1.2.28.1.39 SipMsgHeaderName

请参见 [SipMsgHeaderValue](#)

3.1.2.28.1.40 SipMsgHeaderValue

配置项	SipMsgHeaderName SipMsgHeaderValue
节	[SIP]
书写格式	SipMsgHeaderName=a SipMsgHeaderValue=b
取值范围	a: 表示需要添加的字段名称; b: 表示需要添加字段的具体内容。
功能描述	设置添加字段的名称和内容, 函数 SsmSipMsgSetHeader 可以实现相同功能。

3.1.2.28.1.41 SipAutoAdaptRtpSrc

配置项	SipAutoAdaptRtpSrc
节	[SIP]
书写格式	SipAutoAdaptRtpSrc=b
取值范围	b=0, 不处理; (默认值); b=1, 处理。

功能描述	设置 B 型 IP 卡是否处理源（RTP 包的一个字段）不断变换的 RTP 包。
------	------------------------------------------

3.1.2.28.1.42 SipSpecialReg

配置项	SipSpecialReg
节	[SIP]
书写格式	SipSpecialReg=z
取值范围	z=1, 注册消息中 from 和 to 字段不带服务器端口; z=0, 注册消息中 from 和 to 字段带服务器端口（默认值）。
功能描述	设置注册消息中的 from 和 to 字段是否携带服务器端口。

3.1.2.28.1.43 SipTotalWhiteListNum

配置项	SipTotalWhiteListNum SipTotalBlackListNum SipWhiteList SipBlackList
节	[SIP]
书写格式	SipTotalWhiteListNum=n SipTotalBlackListNum=m SipWhiteList[i]=x SipBlackList[i]=x
取值范围	n: 白名单 IP 地址数目； m: 黑名单 IP 地址数目； x: IP 地址。
功能描述	配置黑白名单。

使用范例	<p>设置白名单:</p> <p>SipTotalWhiteListNum=3</p> <p>SipWhiteList[0]=201.123.115.100</p> <p>SipWhiteList[1]=201.123.115.200</p> <p>SipWhiteList[2]=201.123.115.300</p> <p>设置黑名单:</p> <p>SipTotalBlackListNum=3</p> <p>SipBlackList[0]=201.123.115.10</p> <p>SipBlackList[1]=201.123.115.20</p> <p>SipBlackList[2]=201.123.115.30</p>
------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3.1.2.28.1.44 SipTotalBlackListNum

请参见 [SipTotalWhiteListNum](#)

3.1.2.28.1.45 SipWhiteList

请参见 [SipTotalWhiteListNum](#)

3.1.2.28.1.46 SipBlackList

请参见 [SipTotalWhiteListNum](#)

3.1.2.28.1.47 SipSrtp

配置项	SipSrtp
节	[SIP]
书写格式	SipSrtp=b
取值范围	b=0, 关闭 Srtp 加密 (默认值); b=1, 打开 Srtp 加密。
功能描述	设置 Srtp 加密。

3.1.2.28.1.48 SipSessionExpiresMin

配置项	SipSessionExpiresMin
节	[SIP]
书写格式	SipSessionExpiresMin =b

取值范围	b=0, 不指定会话刷新最长时间（默认值）； b>=90, 指定会话刷新最长时间。
功能描述	指定会话刷新最长时间，收到的 invite 消息中的会话刷新时间必须大于本配置，否则驱动将回复 422 拒绝该 invite 消息。
注意事项	本配置项必须和 SipSessionExpires 配置项一起使用才有效。

3.1.2.28.1.49 SipSessionExpires

配置项	SipSessionExpires
节	[SIP]
书写格式	SipSessionExpires =b
取值范围	b=0, 不指定会话刷新时间（默认值）； b>=会话刷新最长时间（ SipSessionExpiresMin ），指定会话刷新时间。
功能描述	设置是否在 invite 消息和 200 ok 中带上本配置项内容的相关字段。
注意事项	本配置项必须和 SipSessionExpiresMin 配置项一起使用才有效。

3.1.2.28.1.50 SipNetMultiIP

配置项	SipNetMultiIP
节	[SIP]
书写格式	SipNetMultiP[i]=[n,m]-x
取值范围	i: 下标, 0-7; n: 起始通道号; m: 结束通道号; x: IP 地址。
功能描述	设置通道呼叫时使用的 IP 地址。
注意事项	<ul style="list-style-type: none">下标必须从 0 开始，顺序递增；使用前必须启用双网卡功能，即配置 SipMultiIpOn=1。

3.1.2.28.1.51 SipTransferRTPFromHost

配置项	SipTransferRTPFromHost
-----	-------------------------------

节	[SIP]
书写格式	SipTransferRTPFromHost =b
取值范围	b=0, OCT 网口 RTP 和外部终端 RTP 包直接交互（默认值）； b=1, OCT 网口 RTP 和外部终端 RTP 包经过主机转发。
功能描述	设置 RTP 包是否经过主机。

3.1.2.28.1.52 SDPAnswerSpecified

配置项	SDPAnswerSpecified
节	[SIP]
书写格式	SDPAnswerSpecified=z
取值范围	z=1, 表示响应 SIP 消息的 SDP 中只带上一个支持的语音编码； z=0, 表示响应 SIP 消息的 SDP 中带上所有支持的语音编码（默认值）。
功能描述	配置响应 SIP 消息的 SDP（200、183 等）中是否只带一个支持的语音编码。

3.1.2.28.1.53 SipMultilpOn

配置项	SipMultilpOn
节	[SIP]
书写格式	SipMultilpOn =b
取值范围	b=0, 不启用双网卡功能（默认值）； b=1, 启用双网卡功能。
功能描述	设置是否启用双网卡功能。
注意事项	双网卡功能涉及的配置项（ SipNetMultiIP ）和函数（ SsmSipSetMutilNetIP ）只有启用了本配置项之后才有效。

3.1.2.28.1.54 SipRestartAudioFor183

配置项	SipRestartAudioFor183
节	[SIP]
书写格式	SipRestartAudioFor183 =z

取值范围	$z=1$, 表示收到 200OK 时, 重启因 183 触发的语音 (默认值); $z=0$, 表示收到 200OK 时, 不重启因 183 触发的语音。
功能描述	在收到 200OK 时是否重启因 183 触发的语音。

3.1.2.28.1.55 SipSingleProtocol

配置项	SipSingleProtocol
节	[SIP]
书写格式	SipSingleProtocol =z
取值范围	$z=1$, 只支持 TCP 传输 SIP 消息或者只支持 UDP 传输 SIP 消息; $z=0$, 同时支持 TCP 和 UDP 传输 SIP 消息 (默认值)。
功能描述	设置是否只支持 TCP 或 UDP 传输 SIP 消息。
注意事项	本配置项是在 SipTransportProtocol 配置的基础上指定传输协议类型, 如果 SipTransportProtocol 配置成 UDP, 那么只使用 UDP 传输 SIP 消息; 如果 SipTransportProtocol 配置成 TCP, 那么只使用 TCP 传输 SIP 消息。

3.1.2.28.1.56 SipSendRequestToSrcIP

配置项	SipSendRequestToSrcIP
节	[SIP]
书写格式	SipSendRequestToSrcIP=z
取值范围	$z=1$, 请求消息向会话源地址发送 (默认值); $z=0$, 请求消息向会话消息中的 contact 字段发送。
功能描述	设置请求消息是否向会话源地址发送。

3.1.2.28.1.57 SipExpiresPercent

配置项	SipExpiresPercent
节	[SIP]
书写格式	SipExpiresPercent =m
取值范围	m 代表注册刷新有效期百分比, $0 < m \leq 100$, 缺省为 100。

功能描述	设置注册刷新有效期百分比。
------	---------------

3.1.2.28.1.58 UserAgent

配置项	UserAgent
节	[SIP]
书写格式	UserAgent =a
取值范围	a: 用户代理名，最大长度为 20 个字符。缺省值默认为“Synway/版本号”，如“Synway/5.3.4.0”。
功能描述	用于设置 SHN 系列板卡的用户代理名。
注意事项	本配置项要求 SynCTI 驱动程序的版本为 Ver. 5.3.1.5 或更高。
使用范例	某些特殊的 sip 服务器只支持特定软终端的注册。通过配置注册消息里的本字段来完成三汇板卡的注册。 假设系统中有一块型号为 SHN 的板卡需要注册到华为软交换上，可进行如下配置： 手动增加一个配置项 UserAgent // [SIP] 节下 UserAgent=HUAWEI OpenEye v3.1 // 华为软交换所识别的终端

3.1.2.28.1.59 LocalTLSPort

配置项	LocalTLSPort
节	[SIP]
书写格式	LocalTLSPort =n
取值范围	n: 设置 TLS 倾听端口，默认为 5061，取值范围为：1024~65536。
功能描述	设置 TLS 倾听端口。

3.1.2.28.1.60 SipRegRefreshTime

配置项	SipRegRefreshTime
节	[SIP]
书写格式	SipRegRefreshTime=m
取值范围	m 代表注册失败之后重新发起注册的时间间隔， $0 \leq m \leq 3600$ ，缺省为 0，单位：秒。
功能描述	设置注册失败之后重新发起注册的时间间隔。
注意事项	<ul style="list-style-type: none">• 本配置只在注册失败之后有效；• 对 4xx 响应的注册失败消息有效。

3.1.2.28.1.61 SipRegMode

配置项	SipRegMode
节	[SIP]
书写格式	SipRegMode =b
取值范围	b=0, 关闭注册服务（默认值）; b=1, 独立模式, 驱动程序自动处理终端的注册请求, 无需应用程序参与; b=2, 交互模式, 应用程序可以通过注册服务 API 对终端的注册请求进行响应并获取终端的实时状态, 对终端进行动态管理, 以实现 VoIP 代理服务器的相关功能。
功能描述	设置板卡的注册服务模式。

3.1.2.28.1.62 SipDscp

配置项	SipDscp
节	[SIP]
书写格式	SipDscp =m
取值范围	m: 设置 Sip 信令包 IP 头中的 Dscp 值, $0 \leq m \leq 63$, 缺省值为 0。m 值越大, 网络优先级越高。
功能描述	通过设置 Sip 信令包 IP 头中的 Dscp 值, 提高 Sip 消息在网络中的优先级。
注意事项	对于 Windows 2000, Windows XP, Windows 2003 系统, 需要将配置项 DisableUserTOSSetting=0 加在 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TcpIp\Parameters 位置; 此配置项对 Windows 2003 版本之后系统无效。

3.1.2.28.1.63 SipListOnlyForOptions

配置项	SipListOnlyForOptions
节	[SIP]
书写格式	SipListOnlyForOptions =z
取值范围	z=0, 过滤全部黑名单（默认值）; z=1, 只过滤黑名单中 Options 消息。
功能描述	设置是否过滤黑名单中 Options 消息。

3.1.2.28.1.64 SipTCPConnectTime

配置项	SipTCPConnectTime
节	[SIP]
书写格式	SipTCPConnectTime =b
取值范围	b>0, TCP 连接超时时间, 单位 ms, 默认值 1000。
功能描述	设置 Sip 呼叫的 TCP 连接超时时间。

3.1.2.28.1.65 SipCallInTickCountLimit

配置项	SipCallInTickCountLimit
节	[SIP]
书写格式	SipCallInTickCountLimit =b
取值范围	b>=0, 通道呼入的间隔时间, 单位 ms, 默认值 0。
功能描述	设置 Sip 通道的呼入间隔时间。

3.1.2.28.1.66 SipExpiresPercent

配置项	SipExpiresPercent
节	[SIP]
书写格式	SipExpiresPercent =m
取值范围	注册刷新有效期百分比, 0<m<=100。
功能描述	设置注册刷新有效期百分比。

3.1.2.28.1.67 SipLearnNetFromVia

配置项	SipLearnNetFromVia
节	[SIP]
书写格式	SipLearnNetFromVia =z
取值范围	z=0, 关闭网络地址学习功能 (默认值); z=1, 开启网络地址学习功能。

功能描述	打开或者关闭网络地址学习功能。
注意事项	<ul style="list-style-type: none">只支持 UDP 呼叫；必须开启 rport；以通道为单位学习，无法对通道的第一次呼叫起作用；学习之后，等同于通道上调用了 SsmSipSetMultiNetIP 函数。

3.1.2.28.1.68 ContactSection

配置项	ContactSection
节	[SIP]
书写格式	ContactSection=d
取值范围	IP 卡通信 IP 地址和端口号
功能描述	指定通道发送 invite、180 和 200 消息中 Contact 字段的 host 和 port。
注意事项	和 SsmSipSetContactSection 功能相同。

3.1.2.28.2 SHN A 系列卡

3.1.2.28.2.1 SHN A 系列卡的高级设置

3.1.2.28.2.1.1 RTP 的 NAT 穿透及设置注册服务器

3.1.2.28.2.1.1.1 EnableRTPStun

请参见 [RegRealm](#)

3.1.2.28.2.1.1.2 StunServerIP

请参见 [RegRealm](#)

3.1.2.28.2.1.1.3 Register

请参见 [RegRealm](#)

3.1.2.28.2.1.1.4 UserName

请参见 [RegRealm](#)

3.1.2.28.2.1.1.5 RegPassword

请参见 [RegRealm](#)

3.1.2.28.2.1.1.6 Domain

请参见 [RegRealm](#)

3.1.2.28.2.1.1.7 RegExpires

请参见 [RegRealm](#)

3.1.2.28.2.1.1.8 MapIP

请参见 [RegRealm](#)

3.1.2.28.2.1.1.9 RegRealm

配置项	EnableRTPStun StunServerIP Register UserName RegPassword Domain RegExpires MapIP RegRealm
节	[BoardId=x]
书写格式	EnableRTPStun=y StunServerIP=p Register=m UserName=s RegPassword=j Domain=x RegExpires=q MapIP=k RegRealm=r

	<p>y: 是否启用 RTP 的 stun 探测功能。</p> <p>p: RTP 的 stun 服务器地址。</p> <p>m: 表示是否启动注册功能。</p> <p>$m=0$: 不启动注册功能;</p> <p>$m \neq 0$: 启动注册功能。</p> <p>s: 用户名, 最大长度为 50 的字符串。</p> <p>当 $m \neq 0$ 时, s 表示注册用户名; 当 $m=0$ 时, s 表示 SIP URI 中 username 字段的内容。</p>
取值范围	<p>j: 注册用户密码。</p> <p>当 $m \neq 0$ 时, j 有效, 表示注册用户密码。</p> <p>x: 注册服务器。</p> <p>当 $m \neq 0$ 时, x 有效, 表示注册服务器的地址。格式为 host[:port]。</p> <p>q: 注册有效期。表示板卡每隔多长时间向注册服务器刷新注册消息。</p> <p>k: 板卡映射的 IP 地址。</p> <p>函数 SsmSipSetConnectionInforOfSDP 可以实现相同功能。</p> <p>r: Sip 服务器别名。</p>
功能描述	设置注册服务器。
注意事项	只有当配置项 ProtocolType 设置为 1 时, 本配置项才有效。

3.1.2.28.3 SHN B/ SHN C 系列卡

3.1.2.28.3.1 SHN B/ SHN C 系列卡的高级设置

3.1.2.28.3.1.1 RTP 的 NAT 穿透及设置注册服务器

3.1.2.28.3.1.1.1 EnableRTPStun

请参见 [RegRealm](#)

3.1.2.28.3.1.1.2 StunServerIP

请参见 [RegRealm](#)

3.1.2.28.3.1.1.3 Register

请参见 [RegRealm](#)

3.1.2.28.3.1.1.4 UserName

请参见 [RegRealm](#)

3.1.2.28.3.1.1.5 RegPassword

请参见 [RegRealm](#)

3.1.2.28.3.1.1.6 Domain

请参见 [RegRealm](#)

3.1.2.28.3.1.1.7 RegExpires

请参见 [RegRealm](#)

3.1.2.28.3.1.1.8 MapIP

请参见 [RegRealm](#)

3.1.2.28.3.1.1.9 RegRealm

配置项	EnableRTPStun StunServerIP Register UserName RegPassword Domain RegExpires MapIP RegRealm
节	[BoardId=x]
书写格式	EnableRTPStun=y StunServerIP=p Register=m UserName=s RegPassword=j Domain=x RegExpires=q MapIP=k RegRealm=r

取值范围	y: 是否启用 RTP 的 stun 探测功能。
	p: RTP 的 stun 服务器地址。
	m: 表示是否启动注册功能。 m=0: 不启动注册功能; m≠0: 启动注册功能。
	s: 用户名, 最大长度为 50 的字符串。 当 m≠0 时, s 表示注册用户名; 当 m=0 时, s 表示 SIP URI 中 username 字段的内容。
	j: 注册用户密码。 当 m≠0 时, j 有效, 表示注册用户密码。
	x: 注册服务器。 当 m≠0 时, x 有效, 表示注册服务器的地址。格式为 host[:port]。
	q: 注册有效期。表示板卡每隔多长时间向注册服务器刷新注册消息。
	k: 板卡映射的 IP 地址。
	r: Sip 服务器别名。
功能描述	设置注册服务器。
注意事项	只有当配置项 ProtocolType 设置为 1 时, 本配置项才有效。

3.1.2.28.3.1.1.10 DscpFlag

请参见 [DscpValue](#)

3.1.2.28.3.1.1.11 DscpValue

配置项	DscpFlag DscpValue
节	[SIP]
书写格式	DscpFlag=p DscpValue=m
取值范围	p: 设置 B 型或 C 型 IP 卡发送的 RTP 数据包是否携带 Dscp 值。 p=1: 是; p=0: 否 (缺省值)。 m: 设置数据包的 Dscp 值, 0≤m≤63, 缺省值为 0。m 值越大优先级越高。
功能描述	通过设置数据包的 Dscp 值, 提高数据包在网络中的优先级。

3.1.2.28.3.1.2.2 设置 RTP 语音包负载大小

3.1.2.28.3.1.2.2.1 SizeG711A

请参见 [SizeG729](#)

3.1.2.28.3.1.2.2.2 SizeG711U

请参见 [SizeG729](#)

3.1.2.28.3.1.2.2.3 SizeG729

配置项	SizeG711A SizeG711U SizeG729
节	[BoardId=x]
书写格式	SizeG711A=m SizeG711U=n SizeG729=x
取值范围	m=20 或者 m=30 n=20 或者 n=30 x=20 或者 x=30
功能描述	设置 RTP 语音包负载大小，单位是毫秒，同时也能调节板卡语音处理能力：每隔多少毫秒取多大的包。
注意事项	<ul style="list-style-type: none">● 本配置项只能填写 20 或者 30，如果不填或者填写其它值，则默认是 20ms；● 本配置项只对 SHN-B 和 SHN-C 型卡有效。

3.1.2.28.3.1.2.4 JitterTime

配置项	JitterTime
节	[BoardId=x]
书写格式	JitterTime=x, y
取值范围	40≤x≤800, 160≤y≤1200, 单位为毫秒，缺省值为 x=160, y=400。
功能描述	设置 RTP 语音 JitterBuffer 的缓冲大小，x 为最小缓冲时间，y 为最大缓冲时间。当 RTP 包接收速度大于最大缓冲或小于最小缓冲时都会造成语音质量不好的情况。
注意事项	在没有特殊需求的情况下请不要随意修改这个配置，否则容易出现语音质量问题。

3.1.2.28.3.1.3 开启 OCT 语音复位功能

配置项	OctTDMIPReSet
节	[BoardId=x]
书写格式	OctTDMIPReSet=p
取值范围	p=0: 否; p=1: 是(默认值)。
功能描述	设置是否开启 OCT 语音复位功能。

3.1.2.29 DST 系列板卡常用配置项(REC 系列)

3.1.2.29.1 SupplyBoardClockLine

配置项	SupplyBoardClockLine
节	[BoardId=x]
书写格式	SupplyBoardClockLine=n
取值范围	n: 板卡通道号, 从 0 开始编号。
功能描述	设置板卡上提供参考时钟的电话线。若该配置项不配置或者配置值不是合法值, 板卡则自动选择参考时钟。
注意事项	本配置项是高级配置项, 只适用于 DST 系列板卡。

3.1.2.29.2 DEventUpdates

配置项	DEventUpdates
节	[SystemConfig]
书写格式	DEventUpdates=n
取值范围	n=0: 过滤事件(默认值); n=1: 不过滤事件。
功能描述	事件过滤开关。如果打开过滤开关, 板卡会过滤掉多余的重复事件。

注意事项	本配置项只适用于 DST 系列板卡。
------	--------------------

3.1.2.30 ATP 系列板卡常用配置项(REC 系列)

3.1.2.30.1 设置输入端的增益

3.1.2.30.1.1 MicGain

配置项	MicGain
节	[BoardId=x]
书写格式	格式 1: MicGain=g,g,g,g 格式 2: MicGain=g,g,g,g,g,g,g 格式 3: MicGain=g,g,g,g,g,g,g,g,g,g,g,g,g
取值范围	g=0: 正常增益; g=1: 高增益。
功能描述	设置模拟录音通道上输入信号的增益。如果语音信号来自模拟电话线路，应将输入增益设置为 0 正常增益(0 DB); 如果来自自动圈式话筒，使用高增益(20DB)。
注意事项	<ul style="list-style-type: none">● 格式 1 适用于 ATP 系列的 USB 录音盒;● 格式 2 适用于 8 通道规格的 ATP 系列板卡;● 格式 3 适用于 16 通道规格的 ATP 系列板卡;● 本配置项不适用于 ATP-24A 系列板卡。

3.1.2.30.2 设置空闲通道的能量检测

3.1.2.30.2.1 EnableIdleChTA

配置项	EnableIdleChTA
节	[BoardId=x]
书写格式	格式 1: EnableIdleChTA =b,b,b,b 格式 2: EnableIdleChTA =b,b,b,b,b,b,b 格式 3: EnableIdleChTA =b,b,b,b,b,b,b,b,b,b,b
取值范围	b=0: 关闭 (缺省值); b=1: 开启。

功能描述	设置模拟录音通道在空闲时是否驱动能量并接收 DTMF 码。
注意事项	<ul style="list-style-type: none">● 格式 1 适用于 ATP 系列的 USB 录音盒；● 格式 2 适用于 8 通道规格的 ATP 系列板卡；● 格式 3 适用于 16 通道规格的 ATP 系列板卡。

3.1.2.30.3 选择多种录音格式

3.1.2.30.3.1 DspCoder

配置项	DspCoder
节	[BoardId=x]
书写格式	DspCoder=n
取值范围	N=131: 加载 G.729A 编码器; N=49: 加载 GSM 编码器; N=85: 加载 MP3 8kbps 编码器; N=86: 加载 MP3 16kbps 编码器。
功能描述	设置板卡上资源 Dsp 加载的编码器，使板卡可以选择更多的录音格式。
注意事项	<ul style="list-style-type: none">● 本配置项是高级配置项，只适用于以下板卡（括号内为其支持的编码格式）： SHD-30B-CT/PCI/FJ (G.729A, GSM) SHD-60B-CT/PCI/FJ (G.729A, GSM) DTP-30C/PCI+ (G.729A, GSM, MP3 8kbps, MP3 16kbps) DTP-60C/PCI+ (G.729A, GSM, MP3 8kbps, MP3 16kbps) DTP-120C/PCI+ (G.729A, GSM, MP3 8kbps, MP3 16kbps) DTP-30C/PCIe+ (G.729A, GSM, MP3 8kbps, MP3 16kbps) DTP-60C/PCIe+ (G.729A, GSM, MP3 8kbps, MP3 16kbps) DTP-120C/PCIe+ (G.729A, GSM, MP3 8kbps, MP3 16kbps) ATP-24A-PCI+ (G.729A, GSM, MP3 8kbps, MP3 16kbps) ATP-24A-PCIe+ (G.729A, GSM, MP3 8kbps, MP3 16kbps) DST-24B/PCI+ (G.729A, GSM, MP3 8kbps, MP3 16kbps) DST-24B/PCI+(SSW) (G.729A, GSM, MP3 8kbps, MP3 16kbps) DST-24B/PCIe+ (G.729A, GSM, MP3 8kbps, MP3 16kbps)● 本配置项要求 SynCTI 驱动程序的版本为 5.0.2.0 或更高。

3.1.2.30.4 设置预录音功能

3.1.2.30.4.1 PrerecordEnable

配置项	PrerecordEnable
节	[SystemConfig]

书写格式	PrerecordEnable=b
取值范围	b=0: 关闭 (缺省值); b=1: 开启。
功能描述	开启或关闭预录音功能。
注意事项	<ul style="list-style-type: none"> 本函数适用于 SHT、SHF 系列板卡的外线通道、ATP 系列板卡和 SHD 系列板卡的 TUP 通道、ISUP 通道、ISDN 通道； 要求 SynCTI Ver. 5.3.3.0 或更高。

3.1.2.30.4.2 PrerecordMode

配置项	PrerecordMode
节	[SystemConfig]
书写格式	PrerecordMode=m
取值范围	m=0: 当 Barge in 检测器检测到 Barge in 时开始将录音数据写入到内部缓冲区 (缺省值); m=1: 当模拟录音通道在线路上检测到摘机动作时开始将录音数据写入到内部缓冲区。 缺省值为 0。
功能描述	设置预录音功能的工作模式。
注意事项	<ul style="list-style-type: none"> 本函数适用于 SHT、SHF 系列板卡的外线通道、ATP 系列板卡和 SHD 系列板卡的 TUP 通道、ISUP 通道、ISDN 通道； 要求 SynCTI Ver. 5.3.3.0 或更高。

3.1.2.30.4.3 PrerecordInsertTime

配置项	PrerecordInsertTime
节	[SystemConfig]
书写格式	PrerecordInsertTime=t
取值范围	0<t<n, 单位为毫秒, 缺省值为 500, 最大值 n 由配置项 RecordBufSize 决定, n=RecordBufSize/16。
功能描述	预录音功能启动后, 当应用程序启动文件录音任务时, 本参数设置将预录音的数据写入到文件中的时间长度。
注意事项	<ul style="list-style-type: none"> 本函数适用于 SHT、SHF 系列板卡的外线通道、ATP 系列板卡和 SHD 系列板卡的 TUP 通道、ISUP 通道、ISDN 通道； 要求 SynCTI Ver. 5.3.3.0 或更高。

3.1.2.30.4.4 PrerecordCodec

配置项	PrerecordCodec
节	[SystemConfig]
书写格式	PrerecordCodec=m
取值范围	m=1: PCM8; m=6: A-law (缺省值); m=7: μ-law; m=17: IMA ADPCM (注: 要求板卡具有硬件编码器); m=49: GSM 6.10; m=85: MP3。
功能描述	设置预录音的编码格式。
注意事项	<ul style="list-style-type: none">本函数适用于 SHT、SHF 系列板卡的外线通道、ATP 系列板卡和 SHD 系列板卡的 TUP 通道、ISUP 通道、ISDN 通道;要求 SynCTI Ver. 5.3.3.0 或更高。

3.1.2.30.5 检测脉冲方式的按键信号

3.1.2.30.5.1 EnablePulseKeyDetect

配置项	EnablePulseKeyDetect
节	[SystemConfig]
书写格式	EnablePulseKeyDetect=b
取值范围	b=0: 否 (缺省值); b=1: 是。
功能描述	设置是否需要检测脉冲方式的按键信号。
注意事项	本配置项只适用于模拟中继线录音通道。

3.1.2.30.6 设置闪断信号的最大保持时间

3.1.2.30.6.1 MaxRecChFlashFilterTime

配置项	MaxRecChFlashFilterTime
节	[SystemConfig]

书写格式	MaxRecChFlashFilterTime=t
取值范围	t: 保持时间, 单位: 毫秒, 缺省值为 200 毫秒。
功能描述	设置录音通道判断闪断信号的过滤时间。更多信息请参见第 1 章中“ 模拟电话线的电压变化 ”部分内容。

3.1.2.30.7 设置振铃信号计数器的清零保护时间

3.1.2.30.7.1 RecChClearRingDelayTime

配置项	RecChClearRingDelayTime
节	[SystemConfig]
书写格式	RecChClearRingDelayTime=t
取值范围	$t \geq 0$, 单位为秒, 缺省值为 0。
功能描述	设置录音通道在检测到振铃信号并摘机之后, 振铃信号计数器的清零保护时间。

3.1.2.30.8 设置软交换功能

3.1.2.30.8.1 BusPlayListen

配置项	BusPlayListen
节	[SystemConfig]
书写格式	BusPlayListen=b
取值范围	b=0: 关闭 CT-BUS 交换功能 (即启动软交换功能); b=1: 打开 CT-BUS 交换功能 (缺省值)。
功能描述	驱动程序是否开启 CT-BUS 总线交换功能。对于无 CT-BUS 总线板卡, 该项不起作用, 自动启动软交换功能, 比如 ATP 系列卡。更多信息请参见第 1 章中“ ATP 系列板卡的工作原理框图 ”部分内容。
注意事项	● 当 BusPlayListen=0 时, GsmCodecEnable=0 设置无效, 驱动内部仍将加载软件录放音引擎, 以支持压缩格式监听。

3.1.2.30.9 设置 MF 检测器

3.1.2.30.9.1 AlwaysEnableRxMF

配置项	AlwaysEnableRxMF
节	[BoardId=x]

3.1.2.30.9.2 MFDualAndAllFreqEnScale

配置项	MFDualAndAllFreqEnScale
节	[BoardId=x]
书写格式	HighAndLowFreqEnScale=m
取值范围	0≤m≤100，单位为%，缺省值为32。
功能描述	设置MF信号的带内能量占全频能量的百分比判定门限
注意事项	本配置项只支持ATP系列板卡。

3.1.2.30.9.3 MFFreqOffset

配置项	MFFreqOffset
节	[BoardId=x]
书写格式	MFFreqOffset=m
取值范围	0≤m≤30，单位为%，缺省值为15。
功能描述	设置MF信号的最大偏移误差。(偏移误差=(检测到的频率-标准频率)*1024/检测到的频率)
注意事项	本配置项只支持ATP系列板卡。

3.1.2.31 DTP 系列板卡常用配置项(REC 系列)

3.1.2.31.1 将监控消息保存到文件

请参见“[3.1.3.5 设置 Ss1 监控日志](#)、[3.1.3.6 设置 Ss7 监控日志](#)、[3.1.3.7 设置 ISDN 监控日志](#)、[3.1.3.8 设置创建日志相关的控制信息](#)”。

3.1.2.31.2 SS7 信令状态机同步输出原始信令消息

3.1.2.31.2.1 bOpenSpySS7Msu

配置项	bOpenSpySS7Msu
节	[SS7Spy]
书写格式	bOpenSpySS7Msu=b
取值范围	b=0: 否 (缺省值); b=1: 是。
功能描述	SynCTI 驱动程序的 ISUP/TUP 协议呼叫状态机在处理收到的 ISUP/TUP 消息时, 是否同时将原始消息输出给应用程序。更多信息请参见第 1 章中 “ DTP 系列板卡的工作原理框图 ” 部分内容。
注意事项	<ul style="list-style-type: none">● 本配置项只适用于 DTP 系列板卡的 SS7 信令;● 本配置项要求 SynCTI 驱动程序的版本为 4.7.2.0 或更高。

3.1.2.31.3 ISDN 监控支持 CorNet 协议

3.1.2.31.3.1 ISDNProtocolType

配置项	ISDNProtocolType
节	[SpyPcm]
书写格式	ISDNProtocolType=b
取值范围	b=0: 否 (缺省值); b=1: 是。
功能描述	设置 ISDN 监控是否支持 CorNet 协议 (该协议与 ISDN Q.931 协议相似)。
注意事项	<ul style="list-style-type: none">● 本配置项只适用于 DTP 系列板卡的 ISDN 信令;● 本配置项要求 SynCTI 驱动程序的版本为 5.1.0.0 或更高。

3.1.2.31.4 E1 SS1 监控支持无 R2 信令链路

3.1.2.31.4.1 SpySs1NoR2

配置项	SpySs1NoR2
节	[SS1Config]

书写格式	SpySs1NoR2=b
取值范围	b=0: 否(缺省值); b=1: 是。
功能描述	设置 E1 SS1 监控是否支持无 R2 线路(DTMF 方式传送号码)。
注意事项	<ul style="list-style-type: none">● 本配置项只适用于 DTP 系列板卡的 E1 SS1 信令;● 本配置项要求 SynCTI 驱动程序的版本为 5.3.1.3 或更高。

3.1.2.31.5 链路不同步到通道不可用状态需要的时间

3.1.2.31.5.1 SpyWaitUnavailableTime

配置项	SpyWaitUnavailableTime
节	[SystemConfig]
书写格式	SpyWaitUnavailableTime=t
取值范围	单位为毫秒, 缺省值为 1280。
功能描述	链路不同步时间超过本配置项的设定值, 则相关的数字中继线上全部通道将进入不可用状态(即 S_CALL_UNAVAILABLE)。
注意事项	<ul style="list-style-type: none">● 本配置项只适用于 DTP 系列板卡;● 本配置项要求 SynCTI 驱动程序的版本为 5.3.1.3 或更高。

3.1.2.31.6 设置板卡消息解码方式

3.1.2.31.6.1 UseHdLC

配置项	UseHdLC
节	[BoardId=x]
书写格式	UseHdLC=b
取值范围	b=0: 否(缺省值); b=1: 是。
功能描述	设置 DTP-C 系列卡是否采用 HDLC 方式进行消息解码.默认采用 dsp 解码。

注意事项	<ul style="list-style-type: none">● 本配置项只适用于 DTP C 系列板卡；● 本配置项要求 SynCTI 驱动程序的版本为 5.3.1.3 或更高。
------	----------------------------------------------------------------------------------------------------------------------

3.1.2.31.7 SS1 监控支持 RBS 协议

3.1.2.31.7.1 RBSType

配置项	RBSType
节	[SS1Config]
书写格式	RBSType=b
取值范围	b=0: 不使用 RBS 协议（缺省值）; b=1: 使用 E&M WINK 方式； b=2: 使用 E&M 方式； b=3: 使用 E&M 2 方式。
功能描述	设置 T1 SS1 监控是支持哪种 RBS 协议。
注意事项	<ul style="list-style-type: none">● 本配置项只适用于 DTP 系列板卡的 T1 SS1 线路；● 本配置项要求 SynCTI 驱动程序的版本为 5.3.1.3 或更高。

3.1.2.31.8 ISDN 监控支持输出第 2 层（L2）原始消息事件

3.1.2.31.8.1 bOpenSpyIsdnL2

配置项	bOpenSpyIsdnL2
节	[ISDN]
书写格式	bOpenSpyIsdnL2=k
取值范围	k=0: 不输出 E_RCV_IsdnL2SpyMsu 事件（缺省值）； k=1: 输出 E_RCV_IsdnL2SpyMsu 事件。
功能描述	设置 DTP 板卡是否输出 E_RCV_IsdnL2SpyMsu 事件。当收到该事件时应用程序可调用 SsmGetIsdnL2SpyMsu 获取消息内容。

3.1.2.32 IPR 系列产品常用配置项(REC 系列)

3.1.2.32.1 SynIPRecorder 系列产品 Master 端的高级通用项配置

3.1.2.32.1.1 RcvTimeSpan

配置项	RcvTimeSpan
节	[BoardId=x]
书写格式	RcvTimeSpan =m
取值范围	m: m 为数字, 默认为 20。
功能描述	接收线程的工作周期, Slaver 端多长时间接收一次 RTP 数据。

3.1.2.32.1.2 CodecInBufferSize

配置项	CodecInBufferSize
节	[BoardId=x]
书写格式	CodecInBufferSize =m
取值范围	m: m 为数字, 默认为 4096。
功能描述	临时存放等待解码的数据的缓冲区的大小, 单位为字节。

3.1.2.32.1.3 SlaverLogLevel

配置项	SlaverLogLevel
节	[BoardId=x]
书写格式	SlaverLogLevel =m
取值范围	m=0: 系统级错误; m=1: 普通出错; m=2: 警告 (默认值); m=3: 普通信息; m=4: 详细信息, 包括函数调用栈。 m=5: 最详细信息, 包括收发数据的内容。 m=6: 调试信息。

功能描述	Slaver 端的日志等级。
------	----------------

3.1.2.32.1.4 SlaverLogType

配置项	SlaverLogType
节	[BoardId=x]
书写格式	SlaverLogType =m
取值范围	m=0: 控制台输出; m=1: 在 Slaver 端的 DbgView 输出; m=2: 在 Slaver 端的 log 文件输出; m=3: 传回 Master (默认值)。
功能描述	Slaver 端的日志显示方式。

3.1.2.32.1.5 RecorderMixerType

配置项	RecorderMixerType
节	[BoardId=x]
书写格式	RecorderMixerType =m
取值范围	m=0: 只录 Primary 的声音; m=1: 只录 Secondary 的声音; m=2: 两边的声音都录 (默认值)。
功能描述	设置混音录音的模式, 对所有 SynIPRecorder 的通道均有效。

3.1.2.32.1.6 RecorderJitterBuffer

配置项	RecorderJitterBuffer
节	[BoardId=x]
书写格式	RecorderJitterBuffer =m,n
取值范围	m: m 为数字, 表示最小 Jitter 延迟时间, 默认值为 1280; n: n 为数字, 表示最大 Jitter 延迟时间, 默认值为 3200。
功能描述	录音模块 JitterBuffer 部分的几个配置项对应的数值, 每个配置项间用,号隔开, x 表示最小 Jitter 延迟时间, y 表示最大 Jitter 延迟时间, 单位为 byte, 即采样点数。

3.1.2.32.1.7 RecorderVolume

配置项	RecorderVolume
节	[BoardId=x]
书写格式	RecorderVolume =m,n
取值范围	m: m 为数字, 为 primary 的音量, 取值范围为[-7, +6], 默认值为 0; n: n 为数字, 为 secondary 的音量, 取值范围为[-7, +6], 默认值为 0。
功能描述	设置录音音量。

3.1.2.32.2 SynIPAnalyzer 系列产品的 IP 交换机监控端口项配置

3.1.2.32.2.1 SIP

配置项	MonitorPortNo MonitorPort MonitorPortType ProxyIPAddress NonStationAddressNo NonStationAddress Special MixCSProtocol
节	[IPRSIP]
书写格式	MonitorPortNo =m MonitorPort[x] = p MonitorPortType[x] = t ProxyIPAddress = i NonStationAddressNo = n NonStationAddress[y] = j Special = k MixCSProtocol = q

取值范围	<p>m: 监控端口的数量，默认值为 0，表示没有监控 SIP 协议，上限为 20;</p> <p>x: x 表示第几个该配置项，范围为 0~19;</p> <p>p: 监控的端口，默认端口为 5060;</p> <p>t: 监控的端口类型 (0: UDP, 1: TCP)，默认值为 0;</p> <p>i: i 为字符串，表示 SIP 的代理地址，取值形式为 IPv4 地址，默认值为 0.0.0.0;</p> <p>n: 非 Station 端的地址个数，这些地址不会被识别成 Station，默认值 0，上限为 20;</p> <p>y: 表示第几个该配置项，取值范围 0~19;</p> <p>j: 非 Station 端的 IP 地址，取值形式为 IPv4 地址;</p> <p>k: 对于同时存在 SIP 消息和交换机私有协议等一些特殊情况，需要对信令解析进行特殊处理。k 取值为 1，用于支持 SIP/AASP 协议；k 取值为 2，按分机号目标监控时 IPA 解析模块从 REQUEST 和 CONTACT 字段中获取分机号;</p> <p>q: 当呼叫建立过程和呼叫释放过程使用不同的传输协议，如呼叫建立过程使用了 TCP，而呼叫释放过程使用了 UDP 时，需要将此配置项设置为 1。</p>
功能描述	对 SIP 协议的监控。

3.1.2.32.2.2 Cisco SCCP

配置项	MonitorPort MonitorPortType
节	[IPRSCCP]
书写格式	MonitorPort = p MonitorPortType = t
取值范围	<p>p: 监控的端口，默认值为 0，表示不监控。如果对 SCCP 协议的监控开启，则常用端口为 2000;</p> <p>t: 监控的端口类型 (0: UDP, 1: TCP)，默认值为 1。</p>
功能描述	对 SCCP 协议的监控。

3.1.2.32.2.3 Avaya H323

配置项	H225CSPortNO H225CSPort H225CSPortType H225RASPortNO H225RASPort H225RASPortType NonStationAddressNo NonStationAddress
节	[IPRAvayaH323]
书写格式	H225CSPortNo =m H225CSPort[x] = p H225CSPortType[x] = t H225RASPortNo =n H225RASPort[x] = q H225RASPortType[x] = u NonStationAddressNo = o NonStationAddress[y] = j
取值范围	<p>m: 监控端口的数量, 默认值为 0, 表示没有监控 Avaya H323 协议, 上限为 20;</p> <p>x: x 表示第几个该配置项, 范围为 0~19;</p> <p>p: CS 端口, 默认端口为 1720;</p> <p>q: RAS 端口, 此端口为可选配置项, 如果交换机没有使用该端口请设置为默认值 0;</p> <p>t: 监控的端口类型 (0: UDP, 1: TCP), 默认值为 1;</p> <p>u: 监控的端口类型 (0: UDP, 1: TCP), 默认值为 0;</p> <p>o: 非 Station 端的地址个数, 这些地址不会被识别成 Station, 默认值 0, 上限为 20;</p> <p>y: 表示第几个该配置项, 取值范围 0~19;</p> <p>j: 非 Station 端的 IP 地址, 取值形式为 IPv4 地址。</p>
功能描述	对 Avaya H323 协议的监控。

3.1.2.32.2.4 Shortel MGCP

配置项	CallAgentPort CallAgentPortType GatewayPort GatewayPortType
节	[IPRShortelMGCP]
书写格式	CallAgentPort = p CallAgentPortType = t GatewayPort=q GatewayPortType=u
取值范围	p: CallAgent 端的端口, 默认值为 0, 表示不监控。如果对 Shortel MGCP 的监控开启, 则常用端口为 2727; q: Gateway 端的端口, 默认值为 0, 表示不监控, 如果对 Shortel MGCP 的监控开启, 则常用端口为 2427; t, u: 监控的端口类型 (0: UDP, 1: TCP), 默认值为 0。
功能描述	对 Shortel MGCP 协议的监控。

3.1.2.32.2.5 H323

配置项	H225CSPortNO H225CSPort H225CSPortType H225RASPortNO H225RASPort H225RASPortType NonStationAddressNo NonStationAddress
节	[IPRH323]

书写格式	H225CSPortNo =m H225CSPort[x] = p H225CSPortType[x] = t H225RASPortNo =n H225RASPort[x] = q H225RASPortType[x] = u NonStationAddressNo = o NonStationAddress[y] = j
取值范围	m: 监控端口的数量, 默认值为 0, 表示没有监控 H323 协议, 上限为 20; x: x 表示第几个该配置项, 范围为 0~19; p: CS 端口, 默认端口为 1720; q: RAS 端口, 此端口为可选配置项, 如果交换机没有使用该端口请设置为默认值 0; t: 监控的端口类型 (0: UDP, 1: TCP), 默认值为 1; u: 监控的端口类型 (0: UDP, 1: TCP), 默认值为 0; o: 非 Station 端的地址个数, 这些地址不会被识别成 Station, 默认值 0, 上限为 20; y: 表示第几个该配置项, 取值范围 0~19; j: 非 Station 端的 IP 地址, 取值形式为 IPv4 地址。
功能描述	对 H323 协议的监控。

3.1.2.32.2.6 Panasonic MGCP

配置项	CallAgentPort CallAgentPortType GatewayPort GatewayPortType
节	[IPRPanasonicMGCP]
书写格式	CallAgentPort = p CallAgentPortType = t GatewayPort=q GatewayPortType=u

	p: CallAgent 端的端口，默认值为 0，表示不监控。如果对 Panasonic MGCP 的监控开启，则常用端口为 2727； q: Gateway 端的端口，默认值为 0，表示不监控。如果对 Panasonic MGCP 的监控开启，则常用端口为 2427； t, u: 监控的端口类型（0: UDP, 1: TCP），默认值为 0。
功能描述	对 Panasonic MGCP 协议的监控。

3.1.2.32.2.7 Toshiba MEGACO

配置项	MonitorPort MonitorPortType
节	[IPRToshibaMEGACO]
书写格式	MonitorPort = p MonitorPortType = t
取值范围	p: 监控的端口，默认值为 0，表示不监控。如果对 Toshiba MEGACO 协议的监控开启，则常用端口为 2944； t: 监控的端口类型（0: UDP, 1: TCP），默认值为 1。
功能描述	对 Toshiba MEGACO 协议的监控。

3.1.2.32.2.8 Siemens H323

配置项	H225CSPort H225CSPortType ProprietaryPort ProprietaryPortType
节	[IPRSiemensH323]
书写格式	H225CSPort = p H225CSPortType = t ProprietaryPort = q ProprietaryPortType = u
取值范围	p: CS 端口，默认端口为 1720； q: Siemens 交换机专用协议消息端口，默认端口为 4060； t, u: 监控的端口类型（0: UDP, 1: TCP），默认值为 1。

功能描述	对 Siemens H323 协议的监控。
------	-----------------------

3.1.2.32.2.9 Alcatel

配置项	MonitorPort MonitorPortType
节	[IPRALCATEL]
书写格式	MonitorPort = p MonitorPortType = t
取值范围	p: 监控的端口，默认值为 0，表示不监控。如果对 Alcatel 协议的监控开启，则常用端口为 32640、7775 以及 32128； t: 监控的端口类型 (0: UDP, 1: TCP)，默认值为 0。
功能描述	对 Alcatel 协议的监控。

3.1.2.32.2.10 Mitel

配置项	MonitorPort MonitorPortType
节	[IPRMITEL]
书写格式	MonitorPort = p MonitorPortType = t
取值范围	p: 监控的端口，默认值为 0，表示不监控。如果对 Mitel 协议的监控开启，则常用端口为 6800； t: 监控的端口类型 (0: UDP, 1: TCP)，默认值为 1。
功能描述	对 Mitel 协议的监控。

3.1.2.32.2.11 LG Nortel

配置项	LTPSPort LTPSPortType PhonePort PhonePortType NonStationAddressNo NonStationAddress
-----	----------------------------------------------------------------------------------------------------------------------------------------------

节	[IPRLGNortel]
书写格式	LTPSPort = m LTPSPortType = t PhonePort = n PhonePortType = q NonStationAddressNo = o NonStationAdress[y] = j
取值范围	m: LTPS 端的端口, 默认值为 0, 表示不监控。如果对 LG Nortel 协议的监控开启, 则常用端口为 5588; n: Phone 端的端口, 默认值为 0, 表示不监控。如果对 LG Nortel 协议的监控开启, 则常用端口为 5588; t, q: 监控的端口类型 (0: UDP, 1: TCP), 默认值为 0。 o: 非 Station 端的地址个数, 这些地址不会被识别成 Station, 默认值 0, 上限为 20; y: 表示第几个该配置项, 取值范围 0~19; j: 非 Station 端的 IP 地址, 取值形式为 IPv4 地址。
功能描述	对 LG Nortel 协议的监控。

3.1.2.32.2.12 Samsung

配置项	MonitorPort MonitorPortType
节	[IPRSAMSUNG]
书写格式	MonitorPort = p MonitorPortType = t
取值范围	p: 监控的端口, 默认值为 0, 表示不监控。如果对 Samsung 协议的监控开启, 则常用端口为 6000; t: 监控的端口类型 (0: UDP, 1: TCP), 默认值为 0。
功能描述	对 Samsung 协议的监控。

3.1.2.32.2.13 Tadicom MGCP

配置项	CallAgentPort CallAgentPortType GatewayPort GatewayPortType
节	[IPRTadicomMGCP]

书写格式	CallAgentPort = p CallAgentPortType = t GatewayPort = q GatewayPortType = u
取值范围	p: CallAgent 端的端口, 默认值为 0, 表示不监控。如果对 Tadicom MGCP 的监控开启, 则常用端口为 2727; q: Gateway 端的端口, 默认值为 0, 表示不监控。如果对 Tadicom MGCP 的监控开启, 则常用端口为 2427; t, u: 监控的端口类型 (0: UDP, 1: TCP), 默认值为 0。
功能描述	对 Tadicom MGCP 协议的监控。

3.1.2.32.2.14 Zenitel

配置项	MonitorPort MonitorPortType
节	[IPRZENITEL]
书写格式	MonitorPort = p MonitorPortType = t
取值范围	p: 监控的端口, 默认值为 0, 表示不监控。如果对 Zenitel 协议的监控开启, 则常用端口为 50001; t: 监控的端口类型 (0: UDP, 1: TCP), 默认值为 1。
功能描述	对 Zenitel 协议的监控。

3.1.2.32.2.15 Nortel UNIStim

配置项	MonitorPort MonitorPortType
节	[IPRNortelUNIStim]
书写格式	MonitorPort = p MonitorPortType = t
取值范围	p: 监控的端口, 默认值为 0, 表示不监控。如果对 Nortel UNIStim 协议的监控开启, 则常用端口为 5000; t: 监控的端口类型 (0: UDP, 1: TCP), 默认值为 0。
功能描述	对 Nortel UNIStim 的监控。

3.1.2.32.3 SynIPAnalyzer 系列产品高级通用项配置

3.1.2.32.3.1 SynIPAnalyzer 声控录音

3.1.2.32.3.1.1 RTPCtrlRec

配置项	RTPCtrlRec
节	[BoardId=x]
书写格式	RTPCtrlRec=m
取值范围	m=0: 关闭 RTP 声控录音模式 (默认值); m=1: 开启 RTP 声控录音模式, 则当前交换机协议端口配置自动失效。
功能描述	设置 SynIPAnalyzer 的 RTP 声控录音模式开关。
注意事项	当开启 RTP 声控录音模式后, 当前交换机协议端口配置将自动失效, 即声控录音模式优先级高于交换机协议端口配置。

3.1.2.32.3.2 SynIPAnalyzer 分析日志

3.1.2.32.3.2.1 IPALogType

配置项	IPALogType
节	[BoardId=x]
书写格式	IPALogType=m
取值范围	m=0: 控制台输出 (默认值) ; m=1: 在 DbgView 输出; m=2: 在 log 文件输出。
功能描述	设置 SynIPAnalyzer 日志显示方式。

3.1.2.32.3.2.2 IPALogLevel

配置项	IPALogLevel
节	[BoardId=x]
书写格式	IPALogLevel=m
取值范围	m=0: 系统级错误; m=1: 普通出错; m=2: 警告 (默认值) ; m=3: 普通信息; m=4: 详细信息, 包括函数调用栈; m=5: 最详细信息, 包括收发数据的内容; m=6: 调试信息。
功能描述	设置 SynIPAnalyzer 的日志等级。

3.1.2.32.3.3 SynIPAnalyzer 数据抓包

3.1.2.32.3.3.1 DumpIPData

配置项	DumpIPData
节	[BoardId=x]
书写格式	DumpIPData=m
取值范围	m=0: 关闭 SynIPAnalyzer IP 数据抓包（默认值）； m=1: 开启 SynIPAnalyzer IP 数据抓包。
功能描述	SynIPAnalyzer IP 数据抓包开关。

3.1.2.32.3.3.2 DumpPackNumPerFile

配置项	DumpPackNumPerFile
节	[BoardId=x]
书写格式	DumpPackNumPerFile=n
取值范围	n: n 为数字, 表示单个抓包保存的数据量, 默认为 2000000 条。
功能描述	SynIPAnalyzer IP 数据抓包开启时单个抓包的数据量。

3.1.2.32.3.3.3 DumpFileReserveNum

配置项	DumpFileReserveNum
节	[BoardId=x]
书写格式	DumpFileReserveNum=n
取值范围	n: n 为数字, 如果值为 0 则驱动保存所有的 IP 数据抓包文件; 如果值为正值, 则驱动只保留指定数量的最近的 IP 数据抓包文件, 其余的文件会被自动删除, n 默认为 0。
功能描述	SynIPAnalyzer IP 数据抓包开启时设置保留的文件个数。

3.1.2.32.3.3.4 DumpFilePath

配置项	DumpFilePath
节	[BoardId=x]
书写格式	DumpFilePath=s
取值范围	s 是存放 IP 数据抓包文件的路径, 长度小于 256 个字符, 缺省值与 API 日志路径相同。
功能描述	SynIPAnalyzer IP 数据抓包文件存放的路径。

3.1.2.33 SHV 系列板卡常用配置项

3.1.2.33.1 OldVar

配置项	OldVar
节	[BoardId=x]

书写格式	OldVar=n
取值范围	n=0: 新算法(默认值); n=1: 老算法。
功能描述	设置 SHN-120B-CT/PCIe/VAR 和 SHD-240E-CT/PCIe/VAR 型变声卡支持的变声算法。
注意事项	<ul style="list-style-type: none">对于 SHN-120B-CT/PCIe/VAR 变声卡, 新变声算法支持 60 路变声资源, 老变声算法支持 120 路变声资源;对于 SHD-240E-CT/PCIe/VAR 变声卡, 新变声算法支持 120 路变声资源, 老变声算法支持 240 路变声资源;本配置项要求 SynCTI 驱动程序的版本为 5.4.0.0 或更高。

3.1.3 高级配置项

3.1.3.1 驱动程序支持多个进程

3.1.3.1.1 MultiCardMultiProcess

配置项	MultiCardMultiProcess
节	[SystemConfig]
书写格式	MultiCardMultiProcess=b
取值范围	b=0: 否(缺省值); b=1: 是。
功能描述	设置 SynCTI 驱动程序是否支持多进程方式的应用程序。

3.1.3.2 输出 API 函数调用的调试信息

3.1.3.2.1 ApiLogEnable

配置项	ApiLogEnable
节	[DebugView]
书写格式	ApiLogEnable =b

取值范围	b = 1: 将 API 函数调用信息输出到 DebugView 文件; b = 2: 将 API 函数调用信息输出到 LOG 文件; b = 3: 将事件信息输出到 LOG 文件; b = 4: 将 API 函数调用信息和事件信息同时输出到 LOG 文件; b = 5: 将 API 函数调用信息和事件信息同时输出到 DebugView 文件; b<1 或者 b>: 5 (缺省值): 不输出调试信息。
功能描述	设置是否将应用程序调用 API 函数的调试信息输出。如果允许驱动程序输出 API 函数的调用信息，驱动程序会将应用程序发出的所有 API 函数调用的函数名称、入口参数、函数执行结果等信息以字符串的形式进行输出。 更多信息请参见第 1 章中“ 使用驱动程序提供的调试功能 ”部分内容。
注意事项	开启此功能会消耗主机的 CPU 资源，因此通常只用于调试应用程序。当应用系统正式运行时，请关闭此项功能。

3.1.3.2.2 ApiLogSetEventRange

配置项	ApiLogSetEventRange
节	[DebugView]
书写格式	ApiLogSetEventRange =n..m
取值范围	n: 表示写 LOG 的起始事件号, n=-1(缺省值)表示输出<=m 的事件。 m: 表示写 LOG 的结束事件号,m=-1(缺省值)表示输出>=n 的事件。
功能描述	设置写入 LOG 文件的事件输出的范围, 将 n 至 m 之间的事件输出都写到 LOG 中(含 n 和 m), 如果 n 和 m 都是-1, 表示所有事件输出都写入 LOG 中。
注意事项	只有当配置项 ApiLogEnable 设置为输出时, 本配置项才有效。

3.1.3.2.3 ApiLogSetChRange

配置项	ApiLogSetChRange
节	[DebugView]
书写格式	ApiLogSetChRange =n..m
取值范围	n: 表示 API 函数和事件输出写 LOG 的起始通道号, n=-1(缺省值)表示输出通道范围为<=m。 m: 表示 API 函数和事件输出写 LOG 的结束通道号, m=-1(缺省值)表示输出通道范围为>=n。
功能描述	设置写入 LOG 的 API 函数和事件的通道范围, 从 n 到 m 之间的通道都具备事件输出写 Log 能力(含 n 和 m), 如果 n 和 m 都是-1, 表示所有通道都具备事件输出到 LOG 能力。

注意事项	只有当配置项 ApiLogEnable 设置为输出时，本配置项才有效。
------	-------------------------------------

3.1.3.2.4 ApiLogCreateMode

配置项	ApiLogCreateMode
节	[DebugView]
书写格式	ApiLogCreateMode =n
取值范围	n=0: 所有通道日志写在一个文件中（缺省值）; n=1: 每个通道单独生成一个日志文件。
功能描述	设置 API 函数和事件日志创建的方式。
注意事项	1. 只有当配置项 ApiLogEnable 设置为输出到 LOG 时，本配置项才有效。 2. 日志文件的命名规则： （1）ApiLogCreateMode =0，输出日志的文件名由“日志类型（ShCtiApi）+年+月+日+时”组成。 例如：ShCtiApi_2012041915.log。 （2）ApiLogCreateMode =1，输出日志的文件名由“日志类型（ShCtiApi）+通道编号+年+月+日+时”组成。例如：ShCtiApi_Ch0015_2012041916.log。

3.1.3.2.5 Mask_SsmGetNoSoundTime

请参见 [Mask_SsmGetRecOffset](#)

3.1.3.2.6 Mask_SsmGetChStateKeepTime

请参见 [Mask_SsmGetRecOffset](#)

3.1.3.2.7 Mask_SsmGetPlayedTime

请参见 [Mask_SsmGetRecOffset](#)

3.1.3.2.8 Mask_SsmGetPlayedPercentage

请参见 [Mask_SsmGetRecOffset](#)

3.1.3.2.9 Mask_SsmGetPlayOffset

请参见 [Mask_SsmGetRecOffset](#)

3.1.3.2.10 Mask_SsmGetRecTime

请参见 [Mask_SsmGetRecOffset](#)

3.1.3.2.11 Mask_SsmGetRecOffset

配置项	Mask_SsmGetNoSoundTime Mask_SsmGetChStateKeepTime Mask_SsmGetPlayedTime Mask_SsmGetPlayedPercentage Mask_SsmGetPlayOffset Mask_SsmGetRecTime Mask_SsmGetRecOffset Mask_SsmGetChState
节	[DebugView]
书写格式	Mask_SsmGetNoSoundTime=b Mask_SsmGetChStateKeepTime=b Mask_SsmGetPlayedTime=b Mask_SsmGetPlayedPercebtage=b Mask_SsmGetPlayOffset=b Mask_SsmGetRecTime=b Mask_SsmGetRecOffset=b Mask_SsmGetChState=b
取值范围	b=0: 不输出（缺省值）; b=1: 输出。
功能描述	设置驱动程序是否输出一些查询类型的 API 函数的调用信息。在配置项的名称中，“Mask_”后面的字符串就表示函数名称。 更多信息请参见第 1 章中“ 使用驱动程序提供的调试功能 ”部分内容。
注意事项	只有当配置项 APILogEnable 设置为 API 日志输出时，本配置项才有效。

3.1.3.2.12 Mask_SsmGetChState

请参见 [Mask_SsmGetRecOffset](#)

3.1.3.3 设置 ISDN 接续日志

3.1.3.3.1 IsdnLogEnable

配置项	IsdnLogEnable
节	[DebugView]

书写格式	<code>IsdnLogEnable =n</code>
取值范围	=0: 不输出 (缺省值); =1: 输出到 LOG 文件; =2: 输出到 DebugView; =3: 输出到 PCAP 文件。
功能描述	设置是否记录 ISDN 协议数据链路层 1 帧消息。

3.1.3.3.2 DecodeIsdnMsg

配置项	DecodeIsdnMsg
节	[DebugView]
书写格式	<code>DecodeIsdnMsg=b</code>
取值范围	<code>b=0</code> : 否 (缺省值); <code>b=1</code> : 是。
功能描述	设置是否由驱动程序在输出 ISDN 调试消息前先将 ISDN 消息进行解码。 原始的 ISDN 消息是二进制形式，不方便阅读。解码后的 ISDN 消息是文本形式，可以直接阅读。同时适用于网络侧和用户侧。

3.1.3.3.3 IsdnDebugLog

配置项	IsdnDebugLog
节	[ISDN]
书写格式	<code>IsdnDebugLog=b</code>
取值范围	<code>b=0</code> : 否 (缺省值); <code>b=1</code> : 是。
功能描述	在配置项 <code>IsdnLogEnable</code> 打开的情况下，设置是否输出 ISDN 信令的调试信息，以便开发人员对运行结果进行观测。如果 <code>IsdnLogEnable</code> 关闭则不输出任何信息。
注意事项	本配置项为高级配置。

3.1.3.4 设置 Ss1 接续日志

3.1.3.4.1 Ss1LogEnable

配置项	Ss1LogEnable
节	[DebugView]
书写格式	Ss1LogEnable=n
取值范围	=0: 不输出 (缺省值); =1: 输出到 LOG 文件; =2: 输出到 DebugView。
功能描述	设置是否将接收和发送的 ABCD 位码和 R2 输出到日志文件。

3.1.3.4.2 Ss1LogCreateMode

配置项	Ss1LogCreateMode
节	[DebugView]
书写格式	Ss1LogCreateMode=b
取值范围	b=0: 所有通道日志写在一个文件中 (缺省值); b=1: 每个通道单独生成一个日志文件。
功能描述	设置 SS1 接续日志的创建方式。
注意事项	只有当配置项 Ss1LogEnable 设置为输出到 LOG 时, 本配置项才有效。

3.1.3.5 设置 Ss1 监控日志

3.1.3.5.1 SpySs1LogEnable

配置项	SpySs1LogEnable
节	[DebugView]
书写格式	SpySs1LogEnable=n
取值范围	=0: 不输出 (缺省值); =1: 输出到 LOG 文件; =2: 输出到 DebugView。
功能描述	设置是否将驱动程序收到的一号信令相关的 CAS 和 R2 的信息输出到日志文件。

3.1.3.5.2 SpySs1CreateMode

配置项	SpySs1CreateMode
节	[DebugView]
书写格式	SpySs1CreateMode=b
取值范围	b=0: 所有通道日志写在一个文件中 (缺省值); b=1: 每个通道单独生成一个日志文件。
功能描述	设置 SS1 监控日志的创建方式。
注意事项	只有当配置项 SpySs1LogEnable 设置为输出到 LOG 时, 本配置项才有效。

3.1.3.6 设置 Ss7 监控日志

3.1.3.6.1 SpySs7LogEnable

配置项	SpySs7LogEnable
节	[DebugView]
书写格式	SpySs7LogEnable=n
取值范围	=0: 不输出 (缺省值); =1: 输出到 LOG 文件; =2: 输出到 DebugView; =3: 输出到 PCAP 文件。
功能描述	设置是否将驱动程序收到的七号信令相关的信令信息输出到日志文件。

3.1.3.7 设置 ISDN 监控日志

3.1.3.7.1 SpylsdnLogEnable

配置项	SpyIsdnLogEnable
节	[DebugView]
书写格式	SpyIsdnLogEnable=n
取值范围	=0: 不输出 (缺省值); =1: 输出到 LOG 文件; =2: 输出到 DebugView; =3: 输出到 PCAP 文件。

功能描述	设置是否将驱动程序收到的 ISDN 信令（包含 DASS2、DPNSS 信令）相关信息输出到日志文件。 (注：此文档中默认情况下 ISDN 信令指的是 Euro-ISDN 信令，如果包含 DASS2、DPNSS 信令会另行说明。)
------	------------------------------------------------------------------------------------------------------------------------

3.1.3.8 设置系统信息监控日志

3.1.3.8.1 SystemInfoLogEnable

配置项	SystemInfoLogEnable
节	[DebugView]
书写格式	SystemInfoLogEnable =n
取值范围	=0: 不输出; =1: 输出到 LOG 文件（缺省值）; =2: 输出到 DebugView。
功能描述	设置是否记录链路同步信息、本端阻抗、E1/T1、CRC-4 和其它板卡运行信息。
注意事项	该日志为系统运行日志，将一直记录，不受配置项 LogMaxPeriod 控制。

3.1.3.9 设置数字电话监控信令日志

3.1.3.9.1 DSTLogEnable

配置项	DSTLogEnable
节	[DebugView]
书写格式	DSTLogEnable =n
取值范围	=0: 不输出（缺省值）; =1: 输出到 LOG 文件; =2: 输出到 DebugView。
功能描述	设置是否将驱动程序收到的数字电话的相关信令信息输出到日志文件。
注意事项	此配置项只有在驱动版本 5314 及以上才生效。

3.1.3.9.2 DSTLogCreateMode

配置项	DSTLogCreateMode
节	[DebugView]

书写格式	DSTLogCreateMode=b
取值范围	b=0: 所有通道日志写在一个文件中(缺省值); b=1: 每个通道单独生成一个日志文件。
功能描述	设置数字电话监控信令日志的创建方式。
注意事项	只有当配置项 DSTLogEnable 设置为输出到 LOG 时, 本配置项才有效。

3.1.3.10 设置传真交互命令日志

3.1.3.10.1 FaxLogEnable

配置项	FaxLogEnable
节	[DebugView]
书写格式	FaxLogEnable =n
取值范围	=0: 不输出(缺省值); =1: 输出到 LOG 文件; =2: 输出到 DebugView。
功能描述	设置是否将驱动上层收到的传真相关交互信息输出到日志文件。

3.1.3.10.2 FaxLogCreateMode

配置项	FaxLogCreateMode
节	[DebugView]
书写格式	FaxLogCreateMode=b
取值范围	b=0: 所有通道日志写在一个文件中(缺省值); b=1: 每个通道单独生成一个日志文件。
功能描述	设置传真交互信息日志的创建方式。
注意事项	只有当配置项 FaxLogEnable 设置为输出到 LOG 时, 本配置项才有效。

3.1.3.11 设置 Oct 模块调试信息和错误信息日志

3.1.3.11.1 OctLogEnable

配置项	OctLogEnable
节	[DebugView]

书写格式	<code>OctLogEnable =n</code>
取值范围	=0: 不输出(缺省值); =1: 输出到 LOG 文件; =2: 输出到 DebugView。
功能描述	设置是否将 Oct 模块的相关调试信息和错误信息输出到日志文件。

3.1.3.12 设置创建日志相关的控制信息

3.1.3.12.1 LogCreatePeriod

配置项	LogCreatePeriod																
节	[DebugView]																
书写格式	<code>LogCreatePeriod=n</code>																
取值范围	$n > 0$, 单位: 小时, 缺省值为 24 小时。																
功能描述	设置生成日志的周期																
注意事项	本配置项只对如下日志有效: <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">ShCtiApiLog</td> <td style="padding: 2px;">记录 API 函数调用和驱动抛出的事件的相关信息</td> </tr> <tr> <td style="padding: 2px;">ShCtiSs1Log</td> <td style="padding: 2px;">记录接收和发送的 ABCD 位码和 R2</td> </tr> <tr> <td style="padding: 2px;">ShCtiIsdnLog</td> <td style="padding: 2px;">记录 ISDN 协议数据链路层 1 帧消息</td> </tr> <tr> <td style="padding: 2px;">ShCtiSpySs7Log</td> <td style="padding: 2px;">记录驱动程序收到的 1 帧消息</td> </tr> <tr> <td style="padding: 2px;">ShCtiSpySs1Log</td> <td style="padding: 2px;">记录驱动程序收到的与一号信令相关的 CAS 和 R2</td> </tr> <tr> <td style="padding: 2px;">ShCtiSpyIsdnLog</td> <td style="padding: 2px;">记录驱动程序收到的 1 帧消息</td> </tr> <tr> <td style="padding: 2px;">ShCtiSystemInfoLog</td> <td style="padding: 2px;">记录系统信息, 如链路同步信息、本端阻抗、E1/T1、CRC-4 和其它板卡运行信息等</td> </tr> <tr> <td style="padding: 2px;">ShCtiDstLog</td> <td style="padding: 2px;">记录数字电话监控信令信息</td> </tr> </table>	ShCtiApiLog	记录 API 函数调用和驱动抛出的事件的相关信息	ShCtiSs1Log	记录接收和发送的 ABCD 位码和 R2	ShCtiIsdnLog	记录 ISDN 协议数据链路层 1 帧消息	ShCtiSpySs7Log	记录驱动程序收到的 1 帧消息	ShCtiSpySs1Log	记录驱动程序收到的与一号信令相关的 CAS 和 R2	ShCtiSpyIsdnLog	记录驱动程序收到的 1 帧消息	ShCtiSystemInfoLog	记录系统信息, 如链路同步信息、本端阻抗、E1/T1、CRC-4 和其它板卡运行信息等	ShCtiDstLog	记录数字电话监控信令信息
ShCtiApiLog	记录 API 函数调用和驱动抛出的事件的相关信息																
ShCtiSs1Log	记录接收和发送的 ABCD 位码和 R2																
ShCtiIsdnLog	记录 ISDN 协议数据链路层 1 帧消息																
ShCtiSpySs7Log	记录驱动程序收到的 1 帧消息																
ShCtiSpySs1Log	记录驱动程序收到的与一号信令相关的 CAS 和 R2																
ShCtiSpyIsdnLog	记录驱动程序收到的 1 帧消息																
ShCtiSystemInfoLog	记录系统信息, 如链路同步信息、本端阻抗、E1/T1、CRC-4 和其它板卡运行信息等																
ShCtiDstLog	记录数字电话监控信令信息																

3.1.3.12.2 LogMaxKeep

配置项	LogMaxKeep
节	[DebugView]
书写格式	<code>LogMaxKeep=n</code>
取值范围	$n > 0$ 或者 $n = -1$, 缺省值为: $n = 7$ 。
功能描述	生成日志文件数目大于本参数设置的值时, 最先生成的多出的日志将自动被删除。如果 $n = -1$, 则表不删除, 生成日志文件的数目无限制。

注意事项	1. 本配置项功能实现是以日志输出开启后，应用程序没有再重启过为前提。
	2. 本配置项只对如下日志有效：
	ShCtiApiLog 记录 API 函数调用和驱动抛出的事件的相关信息
	ShCtiSs1Log 记录接收和发送的 ABCD 位码和 R2
	ShCtiIsdnLog 记录 ISDN 协议数据链路层 1 帧消息
	ShCtiSpySs7Log 记录驱动程序收到的 1 帧消息
	ShCtiSpySs1Log 记录驱动程序收到的与一号信令相关的 CAS 和 R2
	ShCtiSpyIsdnLog 记录驱动程序收到的 1 帧消息
	ShCtiSystemInfoLog 记录系统信息，如链路同步信息、本端阻抗、E1/T1、CRC-4 和其它板卡运行信息等
	ShCtiDstLog 记录数字电话监控信令信息

3.1.3.12.3 LogMaxPeriod

配置项	LogMaxPeriod
节	[DebugView]
书写格式	LogMaxPeriod=n
取值范围	n>0 或者 n=-1，缺省值为：n=30。
功能描述	累计日志创建周期数达到本参数设置的次数后，自动停止输出日志。如果 n= -1，则表无限制。
注意事项	1. 本配置项功能实现是以日志输出开启后，应用程序没有再重启过为前提。
	2. 本配置项只对如下日志有效：
	ShCtiApiLog 记录 API 函数调用和驱动抛出的事件的相关信息
	ShCtiSs1Log 记录接收和发送的 ABCD 位码和 R2
	ShCtiIsdnLog 记录 ISDN 协议数据链路层 1 帧消息
	ShCtiSpySs7Log 记录驱动程序收到的 1 帧消息
	ShCtiSpySs1Log 记录驱动程序收到的与一号信令相关的 CAS 和 R2
	ShCtiSpyIsdnLog 记录驱动程序收到的 1 帧消息
	ShCtiDstLog 记录数字电话监控信令信息

3.1.3.12.4 LogFilePath

配置项	LogFilepath
节	[DebugView]
书写格式	LogFilepath=S
取值范围	S 是存放日志文件的路径。缺省值为当前应用程序所在的工作目录。
功能描述	设置日志存放的路径，必须是已存在路径，如：“D:\”，否则函数调用失败。如果设置为 NULL，则是当前工作路径，如果只是写一个名字，则在当前工作路径下创建此文件夹。

注意事项	本配置项只对如下日志有效：	
	ShCtiApiLog	记录 API 函数调用和驱动抛出的事件的相关信息
	ShCtiSs1Log	记录接收和发送的 ABCD 位码和 R2
	ShCtiIsdnLog	记录 ISDN 协议数据链路层 1 帧消息
	ShCtiSpySs7Log	记录驱动程序收到的 1 帧消息
	ShCtiSpySs1Log	记录驱动程序收到的与一号信令相关的 CAS 和 R2
	ShCtiSpyIsdnLog	记录驱动程序收到的 1 帧消息
	ShCtiSystemInfoLog	记录系统信息，如链路同步信息、本端阻抗、E1/T1、CRC-4 和其它板卡运行信息等
	ShCtiDstLog	记录数字电话监控信令信息
	ShCtiSipLog	记录 IP 卡的信令以及处理信息

3.1.3.12.5 LogOverWrite

配置项	LogOverWrite	
节	[DebugView]	
书写格式	LogOverWrite=n	
取值范围	n=0：重启应用程序原有日志内容不被覆盖，在原有内容后面追加新内容（缺省值）； n=1：重启应用程序原有日志内容被覆盖。	
功能描述	设置重启应用程序后是否覆盖原有日志内容。	
注意事项	本配置项只对如下日志有效：	
	ShCtiApiLog	记录 API 函数调用和驱动抛出的事件的相关信息
	ShCtiSs1Log	记录接收和发送的 ABCD 位码和 R2
	ShCtiIsdnLog	记录 ISDN 协议数据链路层 1 帧消息
	ShCtiSpySs7Log	记录驱动程序收到的 1 帧消息
	ShCtiSpySs1Log	记录驱动程序收到的与一号信令相关的 CAS 和 R2
	ShCtiSpyIsdnLog	记录驱动程序收到的 1 帧消息
	ShCtiSystemInfoLog	记录系统信息，如链路同步信息、本端阻抗、E1/T1、CRC-4 和其它板卡运行信息等
	ShCtiDstLog	记录数字电话监控信令信息
	ShCtiSipLog	记录 SIP 日志信息

3.1.3.12.6 CreateDumpWhenCrash

配置项	CreateDumpWhenCrash	
节	[SystemConfig]	
书写格式	CreateDumpWhenCrash=n	
取值范围	n=0：驱动崩溃时不产生 dump 文件； n=1：驱动崩溃时产生 dump 文件(缺省值)。	
功能描述	设置是否开启驱动崩溃时产生 dump 文件功能。Windows 下如果驱动崩溃，将在应用程序当前目录产生文件名为 Crash-日期时间.dmp 文件。VC 平台在应用程序或驱动出现非法时都能输出 dump 文件，其他平台下只有在驱动出现非法时才会输出 dump 文件，Windows 2000 及以下系统不支持此功能；Linux 下如果驱动崩溃，将在应用程序当前目录产生文件名为“core.PID”文件，PID 为应用程序进程号。	

3.1.3.13 设置 DSP 四字节访问功能

3.1.3.13.1 DspAddr4

配置项	DspAddr4
节	[BoardId=x]
书写格式	DspAddr4=n
取值范围	n=0: 以双字节方式访问 DSP (缺省值)。 n=1: 以四字节方式访问 DSP。
功能描述	设置是否开启 DSP 四字节访问功能。目前已经发现机器 IBM X3650 M3 和研华 610L，当以双字节方式访问 DSP 时会导致机器出现蓝屏，用本配置项设置为四字节访问即可解决蓝屏问题。

3.1.3.14 设置 DSP 的回波抵消效果

3.1.3.14.1 DspEc

配置项	DspEc
节	[BoardId=x]
书写格式	DspEc=n
取值范围	n=0: 不开启 DSP 的 EC (Echo Cancellation, 回波抵消) 功能 (缺省值)； n≠0: 开启 DSP 的 EC 功能，并以该值作为 EC 功能的参数。
功能描述	设置是否开启 DSP EC 功能及 DSP EC 功能开启时的回波抵消效果。参数 DspEc 需要根据实际声音情况配置，具体步骤为： 1、先将 DspEc 设成 0，录制数字卡来话； 2、在录制的声音文件中选取一段稳定的回波残留，用 cooledit 选中一小段分析极值（假设极值为 val），按如下公式计算 DspEc=[(val/4)*(val/4)]*32。
注意事项	本配置项仅对 SHD EC 卡有效。

3.1.3.15 设置 CPU 多核分担功能

3.1.3.15.1 EnableSetDpc

配置项	EnableSetDpc
节	[SystemConfig]
书写格式	EnableSetDpc=b
取值范围	b=0: 否 (缺省值)； b=1: 是。
功能描述	设置 SynCTI 驱动程序是否打开 cpu 多核分担功能 (仅适用 windows 系统下)。

3.1.3.16 设置过滤直流偏离

3.1.3.16.1 DcOnOffValue

配置项	DcOnOffValue
节	[BoardId=x]
书写格式	DcOnOffValue=v

取值范围	对 SHT/ATP 系列板卡，取值范围为： $0 \leq v \leq 100$ ；SHT-16B-CT/PCI/FAX 板卡缺省值为 0，其他 SHT/ATP 系列卡缺省值为 40； 对 SHD-30C-CT/PCI、SHD-60C-CT/PCI、SHD-30C-CT/PCI/Fax、SHD-60C-CT/PCI/Fax 板卡，取值为 0 或 1，默认为 0。
功能描述	对 SHT/ATP 系列板卡来说是设置 DC 偏离阈值； 对 SHD-30C-CT/PCI、SHD-60C-CT/PCI、SHD-30C-CT/PCI/Fax、SHD-60C-CT/PCI/Fax 板卡为是否打开去直流开关，打开去直流开关后的能量将会乘以 2，以保持能量不衰减。

3.1.3.16.2 BusDcOnOffValue

配置项	BusDcOnOffValue
节	[Boardid=x]
书写格式	BusDcOnOffValue=b
取值范围	取值为 0 或 1，默认为 0。
功能描述	是否打开语音 DSP 串口 1 上总线去直流开关（0：关闭；1：打开）。
注意事项	本配置项仅对 SHT-16D-CT/PCIe 卡有效。

3.1.3.17 设置驱动事件处理方式

3.1.3.17.1 MultiBoardClock

配置项	MultiBoardClock
节	[SystemConfig]
书写格式	MultiBoardClock=b
取值范围	b=0：否(缺省值)； b=1：是。
功能描述	设置是否启用多线程处理驱动事件。

3.2 预加载语音配置文件 ShIndex.ini (CTI 系列)

3.2.1 [System]节

3.2.1.1 MaxIndexSeg

配置项	MaxIndexSeg
节	[System]
书写格式	MaxIndexSeg=n
取值范围	$0 \leq n \leq 65535$ ，缺省值为 0。（注：“ $n \times$ 板卡总通道数 $\times 306$ ”的值必须小于实际可用的电脑物理内存，否则会初始化失败）。
功能描述	设置单通道预加载语音片段的总数。
注意事项	MaxIndexSeg 决定了[SegNo=x]节的个数。

3.2.2 [SegNo=x]节

设置每段内存索引语音片段的文件信息，其中 x 为段号。如果有多个语音片段，需要配置多个[SegNo=x]节。

3.2.2.1 FileName

配置项	FileName
节	[SegNo=x]
书写格式	FileName=s
取值范围	s 为包含语音片段的文件名，可以是标准的 wav 格式文件，也可以是没有文件头的无格式文件。如果是标准的 wav 文件，配置项 CodecFormat 指定的格式将被忽略；否则被视为无格式文件，编码格式由配置项 CodecFormat 指定。 如果不指定全路径，驱动程序在应用程序的当前目录下查找该文件。
功能描述	设置语音文件名。
使用范例	FileName=d0.wav

3.2.2.2 Alias

配置项	Alias
节	[SegNo=x]
书写格式	Alias=s
取值范围	s 为别名，长度不能超过 20 个字符，不能以数字'0'~'9'作为别名字符串的首字符。为空表示不设置别名。 缺省值为空。
功能描述	为语音片段分配别名。
使用范例	Alias=D0

3.2.2.3 CodecFormat

配置项	CodecFormat
节	[SegNo=x]
书写格式	CodecFormat=n

取值范围	n=6: A-Law (缺省值); n=7: U-Law; n=17: IMA ADPCM。
功能描述	设置语音片段的语音编码格式。
注意事项	所有的预加载语音片段都必须具有相同的语音编码格式。

3.2.2.4 StartOffset

配置项	StartOffset
节	[SegNo=x]
书写格式	StartOffset=n
取值范围	n: 语音片段数据在文件中的起始位置偏移量, 缺省值为 0, 对于 ADPCM 格式应该设置为 256 的整数倍。
功能描述	设置从文件中实际的语音数据部分开始计算的语音片段的起始位置。对于标准的 wav 文件, 起始位置为 0 是指文件头后的第 1 个语音数据字节; 对于无格式文件, 是从文件的头部开始计算。

3.2.2.5 Length

配置项	Length
节	[SegNo=x]
书写格式	Length=n
取值范围	N: 语音片段的长度, 单位为字节, 缺省值为-1。
功能描述	设置本语音片段的数据长度。语音片段的字节长度与时间长度之间的换算值取决于编码格式, 相关内容请参见第 1 章中的 “ Synway 板卡支持的语音编码/解码格式 ” 部分。
注意事项	<ul style="list-style-type: none">如果本参数大于配置项 StartOffset 到文件尾的数据长度, 则实际加载的语音片段长度为 StartOffset;如果 Length 设置为-1, 表示从 IStartPos 装载到文件尾。

3.3 SynIPRecorder Slaver 端配置文件 record_slaver.ini (SynIPR 产品 only)

运行驱动安装包, 选择“完全安装”或者“特殊安装”并正确完成 SynCTI 驱动程序的安装后, 安装程序会在安装目录下生成一个文件名称为 record_slaver.ini 的配置文件, 此配置文件用于保存 SynIPRecorder 系列产品 Slaver 端的配置项。开发人员可以通过文本编辑软件直接编写或修改配置文件。

配置文件中, 配置信息以节(Section)为单位, 每个节可以包含一个或多个配置项(Item)。节的名称出现在一对

方括号中，后续的每一个文本行就是一个配置项。节的作用范围为从本节的后面一行开始，直到下一个节的上面一行止。每一个配置项都具有“ $x=y$ ”的形式，等号左边是配置项的名称，右边是配置内容。

3.3.1 Slaver 端的必须配置项

3.3.1.1 ActiveConnect

配置项	ActiveConnect
节	[SlaverConfig]
书写格式	ActiveConnect=m
取值范围	m: m 为数字，Slaver 的连接模式，默认值为 1，取值范围为： m=0: 表示 Slaver 处于被动连接模式下，此时 Slaver 不会主动连接 Master，多个 Master 可以连接到 Slaver 上，常用配置项中 SlaverIp , SlaverPort , MasterIp , MasterPort 不起作用， SlaverListenIP , SlaverListenPort 起作用。 m=1: 表示 Slaver 处于主动连接模式下，此时 Slaver 会主动连接 Master，Master 并不能连接 Slaver，常用配置项中 SlaverIp, SlaverPort, MasterIp, MasterPort 起作用，SlaverListenIP, SlaverListenPort 不起作用。
功能描述	设置 Slaver 的连接模式。

3.3.1.2 SlaverIP

配置项	SlaverIP
节	[SlaverConfig]
书写格式	SlaverIP =m
取值范围	m: m 为字符串，取值形式为 IPv4 地址，默认值为 127.0.0.1。
功能描述	Slaver 端的 IP 地址。

3.3.1.3 SlaverPort

配置项	SlaverPort
节	[SlaverConfig]
书写格式	SlaverPort =m
取值范围	m: m 为数字，默认为 9885。
功能描述	Slaver 端的端口。

3.3.1.4 MasterIP

配置项	MasterIP
节	[SlaverConfig]
书写格式	MasterIP =m
取值范围	m: m 为字符串, 取值形式为 IPv4 地址, 默认值为 127.0.0.1。
功能描述	SynIPRecorder Master 的主机 IP 地址。

3.3.1.5 MasterPort

配置项	MasterPort
节	[SlaverConfig]
书写格式	MasterPort =m
取值范围	m: m 为数字, 默认为 9888。
功能描述	SynIPRecorder Master 的主机监听端口。

3.3.1.6 SlaverListenIP

配置项	SlaverListenIP
节	[SlaverConfig]
书写格式	SlaverListenIP =m
取值范围	m: m 为字符串, 取值形式为 IPv4 地址, 默认值为 127.0.0.1。
功能描述	被动连接模式下 Slaver 的监听 IP 地址。

3.3.1.7 SlaverListenPort

配置项	SlaverListenPort
节	[SlaverConfig]
书写格式	SlaverListenPort=m
取值范围	m: m 为数字, 默认为 9887。
功能描述	被动连接模式下 Slaver 端监听的端口。

3.3.2 Slaver 端的高级通用项配置

3.3.2.1 KeepAliveTime

配置项	KeepAliveTime
节	[SlaverConfig]
书写格式	KeepAliveTime =m
取值范围	m: m 为数字, 默认为 3。
功能描述	Slaver 端与 Master 端的心跳时间, 单位为秒, 即每隔多少时间与 Master 自动通讯一次来保持链接, 当检测到链接断开时, 它自动会每隔这么多的时间再尝试链接一次。若该值为 0, 则当断开链接时不再自动尝试链接。

3.3.2.2 RTPPortRange

配置项	RTPPortRange
节	[SlaverConfig]
书写格式	RTPPortRange =m,n
取值范围	m: m 为数字, RTP 起始接收端口, 默认值为 6000; n: n 为数字, RTP 终止接收端口, 默认值为 10000。
功能描述	RTP 的接收端口范围。

3.4 HMP Client 端配置文件 HMPCodec.ini (SynHMP 产品 only)

3.4.1 HMP 系列产品 Client 端的必须配置项

3.4.1.1 NICNum

配置项	NICNum
节	[HMPCodec]
书写格式	NICNum=n
取值范围	n: 配置 HMP Client 所在主机上所使用的网口数。取值范围: 0<n≤8, 默认值为 1。
功能描述	配置 HMP Client 单机支持多网口, 对于 HMP Server 来说相当于一台主机运行了多个 HMP Client。由于每个网口处理 RTP 包时存在带宽限制, 当 HMP Client 处理更多 RTP 资源时, 如果主机存在多个网口, 可以配置多个的网口来使用。

3.4.1.2 LocalIP

配置项	LocalIP
节	[HMPCodec]
书写格式	LocalIP[n] =m
取值范围	n: 主机网口编号, 取值 0~ NICNum -1。 m: m 为字符串, 取值形式为 IPv4 地址, 默认值为 127.0.0.1。 如果配置多网卡时, 也就是 NICNum 值大于 1 时, m 值必须配置为对应的网口 IP 地址。
功能描述	HMP Client 所在的主机 IP 地址, 接收来自 HMP Server 的 RTP 数据, 同时发送 RTP 数据给 HMP Server, 用于 HMP Server 与 HMP Client 之间的内部 RTP 收发。

3.4.1.3 RemoteIP

配置项	RemoteIP
节	[HMPCodec]
书写格式	RemoteIP[n] =m
取值范围	n: 主机网口编号, 取值 0~ NICNum -1。 m: m 为字符串, 取值形式为 IPv4 地址, 默认值为 127.0.0.1。 如果配置多网卡时, 也就是 NICNum 值大于 1 时, m 值必须配置为对应的网口 IP 地址。
功能描述	HMP server 所在的主机 IP 地址。

3.4.1.4 LocalPort

配置项	LocalPort
节	[HMPCodec]
书写格式	LocalPort[n] =m
取值范围	n: 主机网口编号, 取值 0~ NICNum -1。 m: 端口号, 默认值为 5050。
功能描述	HMP Client 所在的主机端口号。

3.4.1.5 RemotePort

配置项	RemotePort

节	[HMPCodec]
书写格式	RemotePort[n] =m
取值范围	n: 主机网口编号, 取值 0~ NICNum -1。 m: 端口号, 默认值为 5051。
功能描述	HMP server 所在主机的端口号。

3.4.1.6 LogType

配置项	LogType
节	[HMPCodec]
书写格式	LogType =m
取值范围	m=0, 不输出 (Linux 默认值) m=1, 输出到终端 (Windows 默认值); m=2, 输出到 HMPCodecx.log 日志上。
功能描述	日志输出类型。

3.4.1.7 LogLocation

配置项	LogLocation
节	[HMPCodec]
书写格式	LogLocation =m
取值范围	m 是存放 HMP Client 日志的路径。缺省路径为应用程序当前目录。
功能描述	设置 HMP Client 日志输出路径。

3.4.1.8 MaxRtpThread

配置项	MaxRtpThread
节	[HMPCodec]
书写格式	MaxRtpThread =m
取值范围	m: 配置 HMP Client RTP 编解码的线程数。取值范围: 0<m≤16, 默认值为 4。

功能描述	HMP Client 处理更多 RTP 资源时，需要使用更多线程来处理 RTP 的编解码与转发，可以根据 CPU 核数和 RTP 资源数来合理分配线程数。
------	-------------------------------------------------------------------------------

4 SS7 信令方式 (CTI 系列)

4.1 SS7 信令方式的基本概念

4.1.1 OPC 和 DPC

信令点(Signaling Point, SP)是信令网中处理控制消息的节点。产生消息的信令点称为该消息的起源点 OPC (Originating Point Code)，在配置项 OPC 中进行设置；消息到达的信令点称为该消息的目的点 DPC (Destination Point Code)。OPC、DPC 均在 SS7 信令服务器的 Ss7Server.ini 文件中设置。

信令点的编码方式包括中国国内网的信令点编码和国际信令网的信令点编码。对于国际信令点编码，CCITT 建议采用 14bit 编码，编码方式见下图：

3bit	8bit	3bit
大区或洲 (ZONE)	国家或地区 (AREA)	信令点 (POINT)

对于中国的信令点编码采用 24bit 方式，各比特的含义如下图所示：

8bit	8bit	8bit
主信令区编码	分信令区编码	信令点编码

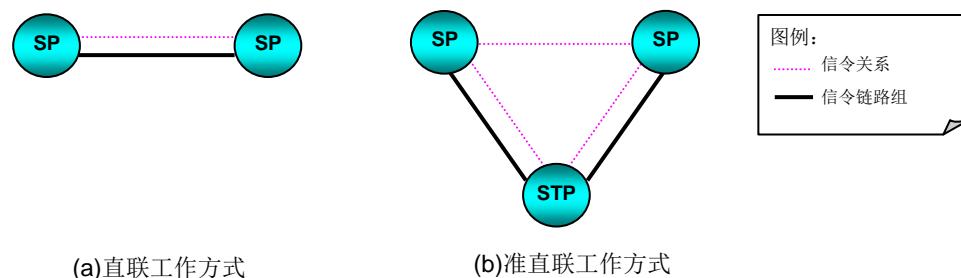
Synway 的 SHD 系列板卡同时支持国际信令点编码方式和中国信令点编码方式，可通过配置项 [SpCodeLen](#) 中进行设置。

4.1.2 直联和准直联

两个信令点之间的信令消息，通过直接连接两个信令点的信令链路进行传递，称为直联工作方式。

属于某信令关系的消息，经过两个或多个串接的信令链路传送，中间要经过一个或几个信令转接点，但通过信令网的消息所取的通路在一定时间内是预先确定的和固定的，称为准直联。

下图表示了直联和准直联工作方式的概念：



SynCTI 驱动程序支持直联和准直联的工作方式，可通过配置配置项 [MaxDPC/DPC](#) 及 [MaxUP_DPC/UP_DPC](#) 进行设置。更多信息请参见本章中“[SS7 信令服务器的配置](#)”部分内容。

4.1.3 信令链路和链路组

在两个信令点之间传送信令消息的链路称为信令链路。直接连接两个信令点的一束信令链路构成一个信令链路组。

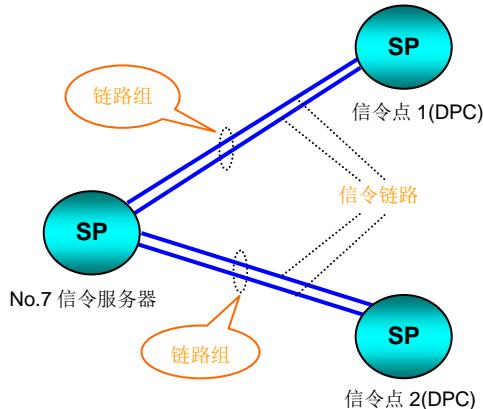
SynCTI 驱动程序最多 48 个信令链路组，可通过配置项 [MaxLinkSet](#) 和 [LinkSet](#) 进行设置；每个信令链路组最多支持 16 条 64kbps 信令链路，可通过配置项 [MaxSs7Pcm](#) 和 [Ss7PcmLink](#) 进行设置。如果信令链路组包含 16 条信令链路，这 16 条信令链路均工作于负荷分担方式。

SS7 信令服务器最多可以连接 48 个信令点或信令转接点。

4.2 Synway 板卡的 SS7 信令应用系统

SynCTI 驱动程序以 SS7 信令服务器的方式向应用程序提供 SS7 信令的服务，应用程序作为客户端程序，可以有一个或多个，可以位于一台或多台计算机上，甚至可以与服务器运行于同一台计算机上。

SS7 信令服务器为使用中国 SS7 信令的数字中继电路提供信令服务，支持本地信令点到多个信令点的连接。有关信令链路、链路组、信令点编码 DPC/OPC、信令点路由的关系如下图所示：



本 SS7 信令服务器程序具有下列特点：

- 一条数字中继线可以同时传送信令消息和语音信息。通常使用数字中继线的 16 时隙来传送信令消息，时隙 1~15、17~31 这 30 个时隙用来传送语音信息；
- 每个链路组最多可以包括 16 条 64kbps 信令链路，这 16 条信令链路之间采用负荷分担方式，由服务器自动进行负荷管理；
- 本机信令点最多可以同时连接到 48 个信令点（SP 或 STP），并且本机信令点与其它直联信令点之间只能使用 1 个链路组（即 1 条路由）；
- 采用 C/S 架构，通过 TCP/IP 协议支持分布式信令链路/客户端结构，但单机使用（服务器与板卡信令链路在同一台计算机上）时可以不需要网卡；
- 如果 SS7 信令服务器上包含了物理信令链路，服务器程序在启动时，会自动为 SHD 系列板卡加载 SynCTI 驱动程序。**注意：在这种情况下，SHD 系列板卡上数字中继线的话路时隙无法再作为话路使用。**

SS7 信令服务器包括下列文件：

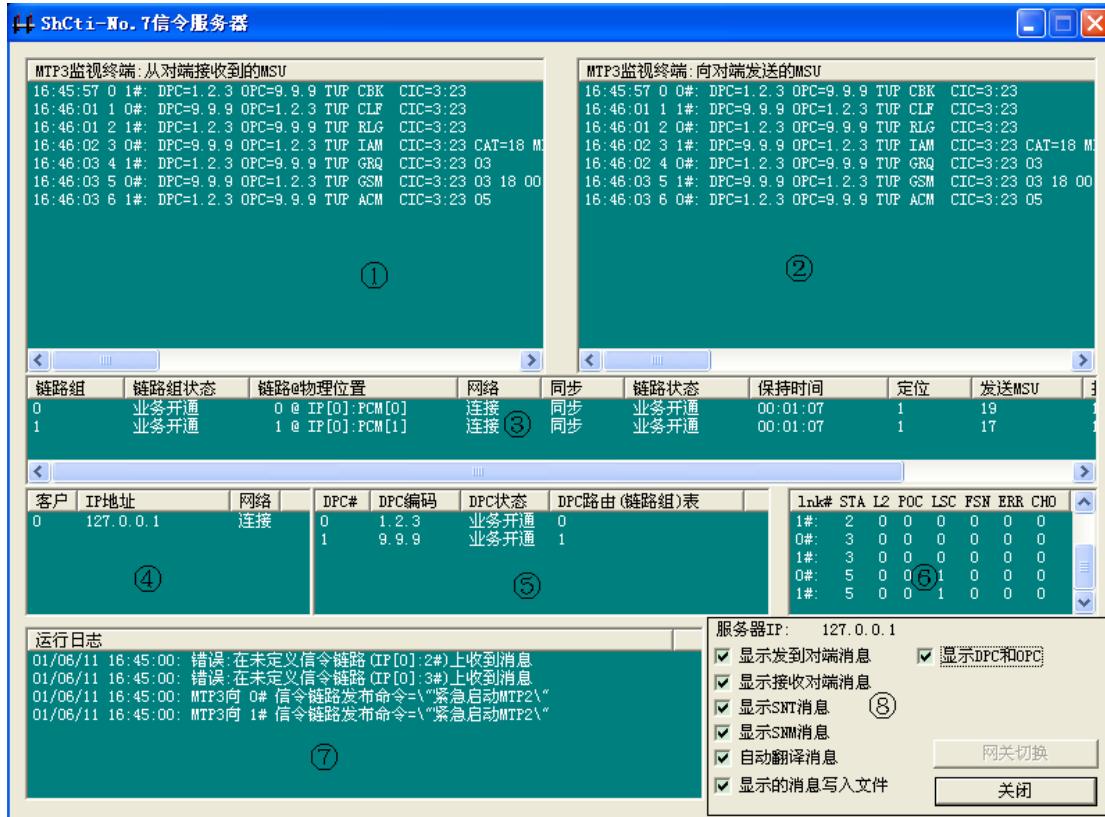
Ss7Server.ini	SS7 信令服务器的配置文件
Ss7Monitor.exe	SS7 信令服务器主程序

Ss7Server.dll	SS7 信令调度模块
Mtp3.dll	SS7 信令 MTP3 层模块
TcpServer.dll	SS7 信令服务器与客户端通信模块 1
MmfServer.dll	SS7 信令服务器与客户端通信模块 2

4.3 SS7 信令服务器的配置和说明

SS7 信令服务器的执行程序为 Ss7Monitor.exe，它使用的配置文件为 Ss7server.ini。

运行 Ss7Monitor.exe 后，将出现如下界面：



程序界面上共有 7 个状态栏和一个按钮区域。

1、①②：接收消息/发送消息列表

为了应用程序调试方便，本程序可以将发出的消息和收到的消息在上述窗口中显示出来。接收/发送消息列表的属性受⑧按钮区域中复选框的控制：

- 显示发到对端消息：只有选中才会将发送到对端消息显示出来
- 显示接收对端消息：只有选中才会将从对端接收消息显示出来
- 显示 SNT 消息：选中时显示 SNT 消息
- 显示 SNM 消息：选中时显示 SNM 消息
- 自动翻译消息：选中时自动翻译接收/发送消息内容；否则以十六进制方式显示接收/发送的原始数据
- 显示的消息写入文件：只有选中才会将显示的消息输出到文件，文件名为 msu_rcv+序号, msu_send+序号
- 显示 DPC 和 OPC：只有选中才会将 DPC 和 OPC 显示出来

当选中“自动翻译消息”时，每一条接收/发送消息通常具有下列格式：

时间 总条数 信令链路编号# SIO 消息内容

对于 TUP 消息，SIO 为 “TUP” (0x84)，后面是消息内容，通常为下列形式：

标题码 CIC=PCM:TS 消息体

2、③：链路组/信令链路信息

链路组/信令链路信息给出本系统配置的信令链路信息。

- ◆ 链路组: 链路组的编号, 来源于配置文件 Ss7Server.ini 的[LinkSetInfo]节。
- ◆ 链路组状态: 链路组业务状态, 包括“业务开通”和“业务中断”。只要本链路组中有 1 条信令链路处于“业务开通”状态时本链路组就会处于“业务开通”状态。
- ◆ 链路@物理位置: 信令链路编号及其物理位置, 如“0 @ IP[0]:PCM[0]”表示本网关上第 0 条信令链路的物理位置为第 0 个客户端上本地 PCM 编号为 0 的 E1。本列中的信息来自于配置文件 Ss7Server.ini 的[Ss7PcmLinkInfo]节。
- ◆ 网络: 本信令链路是否已经注册到本程序, 包括“连接”和“断开”(或无显示)。信令链路进入正常使用时必须在“连接”状态。
- ◆ 同步: 本信令链路的基本帧同步状态(0 时隙), 有“同步”和“失步”2 种状态。信令链路只有在“同步”状态下才可能投入使用。
- ◆ 保持时间: 本信令链路最后一次进入业务开通后所持续的时间。
- ◆ 定位: 本信令链路从程序启动后发生定位的次数。
- ◆ 发送 MSU: 自从程序启动后在本信令链路上发送消息的总数。
- ◆ 接收 MSU: 自从程序启动后在本信令链路上收到消息的总数。
- ◆ 倒换 MSU: 自从程序启动后在本信令链路上倒换消息的总数。

3、④：客户端信息

客户端信息显示程序配置的客户端 IP 地址及连接状态的信息, 来源于配置文件 Ss7Server.ini 的[Ss7ClientInfo]节。共有下述列:

- ◆ 客户: 客户端编号
- ◆ IP 地址: 客户端计算机的 IP 地址
- ◆ 网络: 客户端是否已经成功连接到了本程序。

4、⑤：DPC 信息

DPC 信息显示程序配置的 DPC 信息, 来源于配置文件 Ss7Server.ini 的[DPCInfo]节。

- ◆ DPC#: 从 0 开始的 DPC 编号
- ◆ DPC 编码: DPC 的信令点编码, 由电信局分配
- ◆ DPC 状态: 指示到该信令点的路由是否可用。包括“业务开通”和“业务中断”。只有到 DPC 的路由处于“业务开通”时才能向该 DPC 发送消息。只要到达 DPC 的所有链路组中有 1 条链路组业务开通, 该 DPC 即进入“业务开通”状态。
- ◆ DPC 路由(链路组)表: 到达该 DPC 的路由, 即链路组编号。

5、⑥：链路信息

本状态栏显示所有信令链路状态的详细信息, 通常用于查找信令链路业务中断的原因。

Link #	STA	L2	POC	LSC	FSN	ERR	CHO
链路号	链路状态 0-6	链路故障原因(中断)	处理机故障 0-3	业务未开通 0-1	本端正在发送消息的前面序号	未使用	未使用
	0. 上载但未启动	0. 正常	0. 未出现异常	0. 业务未开通			
	1. 业务中断	1.BSNR非法	1. 本端处理机故障	1. 业务开通			

	2. 初始定位	2.FIBR 非法	2. 对端处理器故障				
	3. 已定位/准备好	3.T2 超时	3. 两端处理器故障				
	4. 已定位/未准备好	4.T6 超时, 对端忙					
	5. 业务开通	5.L3 发送停止命令					
	6. 处理机故障	6. 信号差错率过高					
		7. 初始定位过程中, 未能进入正常定位					
		8. 定时器 1 超时					
		9. 已定位准备好, 收到对端中断信号					
		10. 未准备好状态, 收到对端中断信号					
		11. 开通状态时收到对端中断信号					
		12. 处理机故障状态收到对端中断信号					

6、⑦：运行日志

运行日志随时记录 MTP3 发送的命令及记录运行过程中的各种错误信息。

本状态栏显示程序启动后的日志记录。

4.3.1 设置信令服务器的工作参数

4.3.1.1 OPC

配置项	OPC
节	[Ss7SystemConfig]
书写格式	OPC=a.b.c

取值范围	a: 主信号区编码，以十进制表示。 b: 分信号区编码，以十进制表示。 c: 信号点编码，以十进制表示。
功能描述	设置信令服务器的信令点编码，通常由电信局分配。
注意事项	本配置项为必须配置项。
使用范例	OPC=1.2.13

4.3.1.2 SpCodeLen

配置项	SpCodeLen
节	[Ss7SystemConfig]
书写格式	SpCodeLen=n
取值范围	n: 信令点的编码方式， n=24 (缺省值): 使用中国的信令点编码方式； n=14: 使用国际的信令点编码方式。
功能描述	设置信令点的编码方式。
注意事项	中国的用户通常无须设置本配置项。

4.3.1.3 TEJ

配置项	TEJ
节	[Ss7SystemConfig]
书写格式	TEJ=n
取值范围	n: 7号信令服务器的信令方式， n=0 (缺省值): 表示 E1; n=1, 表示 T1。
功能描述	设置 7 号信令服务器的信令方式。
注意事项	要求 SynCTI Ver. 5.4.0.0 及以上版本。

4.3.2 设置信令服务器的 IP 地址

4.3.2.1 ServerIP

请参见 [4.3.2.1 SecondServerIP](#)

4.3.2.2 SecondServerIP

配置项	ServerIP SecondServerIP
节	[Ss7SystemConfig]
书写格式	ServerIP=a.b.c.d SecondServerIP=a.b.c.d.
取值范围	a.b.c.d: IP 地址。ServerIP 的缺省值为 127.0.0.1。
功能描述	设置 SS7 信令服务器的 IP 地址。ServerIP 用于设置信令服务器本机 IP 地址，SecondServerIP 用于设置与本机互为热备的信令服务器机器 IP 地址。
注意事项	如果只有一个信令服务器，无须设置 SecondServerIP。如果设置了 SecondServerIP，本信令服务器将以热备份服务器的方式启动。

4.3.2.3 Port

配置项	Port
节	[Ss7SystemConfig]
书写格式	Port=n
取值范围	n: 端口号，缺省值为 5150。
功能描述	设置 SS7 信令服务器的侦听端口号。
注意事项	如果只有一个信令服务器，无须设置本配置项。

4.3.3 设置 MTP3 的工作模式和工作参数

4.3.3.1 ConfigMtp3AsSTP

配置项	ConfigMtp3AsSTP
节	[Ss7SystemConfig]
书写格式	ConfigMtp3AsSTP=m

取值范围	m=0: 信令点（缺省值），Mtp3 层对接收消息的 DPC 与本机该链路组的 OPC 作一致性检查； m=1: 信令转接点，Mtp3 层对接收消息的 DPC 不作检查。
功能描述	设置信令服务器中 MTP3 的工作模式。

4.3.3.2 SendSNT

配置项	SendSNT
节	[Ss7SystemConfig]
书写格式	SendSNT=n
取值范围	n=0: 不发送（缺省值）; n=1: 发送。
功能描述	设置 MTP3 的工作参数：是否定期向对端交换机性发送信令链路测试消息(SLTM)。

4.3.3.3 SubServicefield

配置项	SubServicefield
节	[Ss7SystemConfig]
书写格式	SubServicefield=0xn

取值范围	<p>n: 十六进制形式的子业务编码。</p> <p>低四 bit (Bit3~Bit0 分别记为 DCBA):</p> <ul style="list-style-type: none"> DC: 网络表示语 <ul style="list-style-type: none"> 00: 国际网路 01: 国际备用 10: 国内网路 11: 国内备用 BA: 备用 <ul style="list-style-type: none"> 高四 bit (Bit7~Bit4 分别记为 DCBA): DC: 备用 n 的缺省值为 8。 BA: 测试消息类型 <ul style="list-style-type: none"> 00: 备用 01: SNT 消息 10: MTNS 消息 11: 备用 <p>n 的缺省值为 0x18。</p>
功能描述	设置 SS7 子业务编码。
使用范例	SubServicefield=0x18
注意事项	SynCTI Ver. 5.4.0.0 及以上版本时，高四 bit 位有效。

4.3.3.4 SubServicefield

配置项	SubServicefield
节	[LinkSetInfo]
书写格式	SubServicefield[k]=0xn

	<p>k: 链路组序号。</p> <p>n: 十六进制形式的子业务编码。</p> <p>低四 bit (Bit3~Bit0 分别记为 DCBA):</p> <p>DC: 网络表示语</p> <ul style="list-style-type: none"> 00: 国际网路 01: 国际备用 10: 国内网路 11: 国内备用 <p>取值范围 BA: 备用</p> <p>高四 bit (Bit7~Bit4 分别记为 DCBA):</p> <p>DC: 备用</p> <p>BA: 测试消息类型</p> <ul style="list-style-type: none"> 00: 备用 01: SNT 消息 10: MTNS 消息 11: 备用 <p>n 的缺省值为 0x18。</p>
功能描述	设置指定链路组的 SS7 子业务编码。
使用范例	SubServicefield[0]=0x18 SubServicefield[1]=0x18
注意事项	SynCTI Ver. 5.4.0.0 及以上版本时，高四 bit 位有效。

4.3.4 设置客户端的信息

4.3.4.1 MaxSs7Client

请参见 [IP](#)

4.3.4.2 IP

配置项	MaxSs7Client IP
节	[Ss7ClientInfo]
书写格式	MaxSs7Client=M IP[m]=a.b.c.d

取值范围	M: 使用 SS7 信令服务器的客户端计算机的总数, 缺省值为 0。M 最大为 10。如果 M 大于 0, IP 配置项必须出现 M 次。 m: 客户端的逻辑编号, 从 0 开始编号, $0 \leq m < M$ 。 a.b.c.d: 第 m 个客户端计算机的 IP 地址。
功能描述	MaxSs7Client 设置使用 SS7 信令的客户端计算机的总数, IP 逐项设置每个客户端计算机的 IP 地址。单机应用可以设置本配置项为 1
使用范例	如果 SS7 信令服务器与客户端程序运行在同一台计算机上: [Ss7ClientInfo] MaxSs7Client=1 IP[0]=127.0.0.1 如果有 2 台客户端计算机连接到 SS7 信令服务器: [Ss7ClientInfo] MaxSs7Client=2 IP[0]=203.203.203.200 IP[1]= 203.203.203.201

4.3.5 设置信令链路的物理位置

4.3.5.1 MaxSs7Pcm

请参见 [Ss7PcmLink](#)

4.3.5.2 Ss7PcmLink

配置项	MaxSs7Pcm Ss7PcmLink
节	[Ss7PcmLinkInfo]
书写格式	MaxSs7Pcm=M Ss7PcmLink[m]=IP[n].LocalPCM[k]
取值范围	M: 信令链路总数。SS7 信令服务器最多支持 48 个链路组, 每个链路组最多支持 16 条信令链路, 因此 M 的取值范围为 $0 < M \leq 768$ 。 m: 信令链路在服务器中的逻辑编号 (从 0 开始编号), $0 \leq m < M$ 。 n: 包含第 m 条信令链路的客户端计算机的逻辑编号, 该逻辑编号在配置项 IP 中定义。 k: 服务器上第 m 条信令链路在该链路所在客户端计算机上的逻辑编号。k 可以通过系统配置文件 ShConfig.ini 中的配置项 TotalPcm 和 Pcm 进行设置。
功能描述	MaxSs7Pcm 设置 64kbps 信令链路的总数, Ss7PcmLink 设置 SS7 信令服务器中信令链路在客户端计算机中的物理位置。

使用范例	<p>假设应用系统中包含 1 台 SS7 信令服务器和 2 台客户端计算机，总共有 2 条信令链路，第 1 条信令链路的物理位置为第 1 台客户端计算机，其本地编号为 0；第 2 条信令链路的物理位置为第 2 台客户端计算机，其本地编号为 3；则相关配置项的设置如下：</p> <p>[Ss7ClientInfo]</p> <pre>MaxSs7Client=2 IP[0]=203.203.203.200 IP[1]= 203.203.203.201</pre> <p>[Ss7PcmLinkInfo]</p> <pre>MaxSs7Pcm=2 Ss7PcmLink[0]=IP[0],LocalPCM[0] Ss7PcmLink[1]=IP[1],LocalPCM[3]</pre> <p>如果 2 条信令链路均位于第 1 台客户端计算机，并且它们的本地编号分别为 0 和 1，则 MaxSs7Pcm 和 Ss7PcmLink 的设定值为：</p> <pre>Ss7PcmLink[0]=IP[0],LocalPCM[0] Ss7PcmLink[1]=IP[0],LocalPCM[1]</pre>
------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

4.3.6 设置信令链路组

4.3.6.1 MaxLinkSet

请参见 [LinkSet](#)

4.3.6.2 LinkSet

配置项	MaxLinkSet LinkSet
节	[LinkSetInfo]
书写格式	MaxLinkSet=N 格式 1: LinkSet[n]=Ss7PcmLink[i] (链路组中只有 1 条信令链路) 格式 2: LinkSet[n]=Ss7PcmLink[i]+...+Ss7PcmLink[j] (链路组中有 2 条及 2 条以上信令链路) 格式 3: LinkSet[n]=Ss7PcmLink[i],a.b.c (链路组中只有 1 条信令链路) 格式 4: LinkSet[n]=Ss7PcmLink[i]+...+Ss7PcmLink[j],a.b.c (链路组中有 2 条及 2 条以上信令链路)
取值范围	N: 服务器上信令链路组的总数, $1 \leq N \leq 48$, N 决定了 LinkSet 配置项的数目。 n: 链路组的逻辑序号, 从 0 开始编号, $0 \leq n < N$ 。 i,j: 第 n 个信令链路组中包含的信令链路在服务器上的逻辑编号, 在配置项 Ss7PcmLink 中指定, 一个链路组最多可指定 16 条信令链路。 a.b.c: 为第 n 个信令链路组分配的 OPC。
功能描述	MaxLinkSet 设置连接到本服务器的信令链路组的总数, LinkSet 设置链路组由哪些信令链路组成。
注意事项	<ul style="list-style-type: none">● 格式 1 和格式 3 适用于链路组中只有 1 条信令链路的情况, 格式 2 和格式 4 适用于链路组中有 2 条及 2 条以上信令链路的情况;● 格式 3 和格式 4 适用于多个不同链路组有不同的 OPC;● 链路组的 OPC 由配置项 LinkSet[n] 优先指定。如果配置项 LinkSet 中未指定 OPC, 则该链路组的 OPC 由配置项 OPC 指定。
使用范例	MaxLinkSet=2 LinkSet[0]=Ss7PcmLink[1],1.1.1 LinkSet[1]=Ss7PcmLink[2]+ Ss7PcmLink[0],1.1.2

4.3.7 设置 DPC

4.3.7.1 MaxDPC

请参见 [DPC](#)

4.3.7.2 DPC

配置项	MaxDPC DPC
节	[DPCInfo]
书写格式	MaxDPC=N DPC[n]=x.y.z,LinkSet[k]
取值范围	N: DPC 的总数, $0 \leq N \leq 48$ 。 n: DPC 的逻辑编号, 从 0 开始编号, $0 \leq n < N$ 。 x.y.z: 第 n 个 DPC 的信令点编码 (十进制), x 为主信号区编码, y 为分信号区编码, z 为信号点编码, 由电信局分配。 k: 连接本信令服务器与第 n 个 DPC 的信令链路组的逻辑编号。信令链路组的逻辑编号在配置项 MaxLinkSet 和 LinkSet 中指定。
功能描述	配置项 MaxDPC 设置连接到本服务器的 DPC 总数(SP 或 STP); 配置项 DPC 设置 DPC 的编码, 以及到达该 DPC 的路由信息。
注意事项	<ul style="list-style-type: none">如果 $N \geq 1$, DPC 必须从 $n=0$ 开始连续配置 N 次;本配置项只适用于直联方式。准直联方式请使用配置项 MaxUP_DPC 和 UP_DPC。
使用范例	MaxDPC=2 DPC[0]=1.1.8,LinkSet[0] DPC[1]=1.1.9,LinkSet[1]

4.3.8 设置用户层消息的目的点码

4.3.8.1 MaxUP_DPC

请参见 [UP_DPC](#)

4.3.8.2 UP_DPC

配置项	MaxUP_DPC UP_DPC
节	[UP_DPCInfo]

书写格式	<p>MaxUP_DPC=N</p> <p>格式 1: UP_DPC[n]=x.y.z,LinkSet[i] 格式 2: UP_DPC[n]= x.y.z,LinkSet[i]+LinkSet[j] 格式 3: UP_DPC[n]= x.y.z,LinkSet[i]+LinkSet[j],m</p>
取值范围	<p>N: 配置项 UP_DPC 的个数, $0 \leq N \leq 48$;</p> <p>n: UP_DPC 的逻辑编号, 从 0 开始编号, $0 \leq n < N$;</p> <p>x.y.z: 第 n 个 UP_DPC 的信令点编码 (十进制), x 为主信号区编码, y 为分信号区编码, z 为信号点编码, 由电信局分配;</p> <p>i,j: 连接本信令服务器与第 n 个 UP_DPC 之间的信令链路组的逻辑编号。信令链路组的逻辑编号在配置项 <u>MaxLinkSet</u> 和 <u>LinkSet</u> 中指定;</p> <p>m: 如果连接本信令服务器与第 n 个 UP_DPC 之间的信令链路组有 2 组, m 用于指定服务器使用信令链路组 i 和 j 的工作方式:</p> <ul style="list-style-type: none"> $m=0$: 负荷分担 (缺省值, 此时格式 2 与格式 3 等效); $m=1$: 第 i 组信令链路组为正常路由, 第 j 路由为迂回路由。
功能描述	MaxUP_DPC 设置通过信令转接点 (STP) 与本系统相联的信令业务点 (SP) 总数, 它的设定值决定了配置项 UP_DPC[n] 出现的行数。UP_DPC 设置第 n 个信令点的编码, 以及到达该信令点的路由信息。
注意事项	<ul style="list-style-type: none"> ● 如果 $N \geq 1$, UP_DPC 必须从 $n=0$ 开始连续配置 N 次; ● 本配置项可以用于直联、准直联、直联/准直联混合环境; ● 如果将本配置应用于直联信令点, 必须与配置项 <u>MaxDPC</u>、<u>DPC</u> 的配置参数保持一致, 即 UP_DPC 与相应的 <u>DPC</u> 应指向同一 SP。 ● 设置 UP_DPC 的格式取决于本信令服务器与第 n 个 UP_DPC 之间的信令链路组的数量: <ul style="list-style-type: none"> ◆ 只有 1 组: 使用格式 1 ◆ 有 2 组: 使用格式 2 或格式 3。
使用范例	<p>MaxUP_DPC=2</p> <p>UP_DPC[0]=1.2.8,LinkSet[0]</p> <p>UP_DPC[1]=1.2.9,LinkSet[1]</p>

4.3.9 设置 TUP/ISUP 消息的分发路由

4.3.9.1 UP_DPC

请参见 [DPC](#)

4.3.9.2 CIC_PCM

请参见 [DPC](#)

4.3.9.3 DPC

配置项	UP_DPC,CIC_PCM, DPC
节	[TUPRouter] / [ISUPRouter]
书写格式	<p>格式 1: UP_DPC[i],CIC_PCM[p]=IP[j],LocalPCM[k] //适用于 TUP/ISUP 协议</p> <p>格式 2: DPC[i],CIC_PCM[p]=IP[j],LocalPCM[k] //适用于 TUP/ISUP 协议</p>
取值范围	<p>i: DPC 或 UP_DPC 的逻辑编号。格式 1 指定 UP_DPC，此时配置项 MaxUP_DPC 必须大于 0，应用于直联或准直联环境；格式 2 指定 DPC，此时可以不设置配置项 MaxUP_DPC 或将其设置为 0，只应用于直联环境。</p> <p>p: CIC 字段中的数字中继线编号，由电信局分配的电路 CIC 起始编号除以 32 后得到的值。不必从 0 开始编号。</p> <p>j: 客户端计算机的 IP 地址的逻辑编号。更多信息请参见配置项 MaxSs7Client 和 IP 的说明。</p> <p>k: 数字中继线在客户端计算机上的逻辑编号。更多信息请参见配置项 TotalPcm 和 Pcm 的说明。</p>
功能描述	<p>UP_DPC,CIC_PCM 设置 TUP/ISUP 消息的派发路由。</p> <p>SS7 信令服务器分发 TUP/ISUP 消息的过程为：</p> <ul style="list-style-type: none"> ◆ 每当 SS7 信令服务器从第 i 个 UP_DPC / DPC 收到一条 TUP/ISUP 消息时，服务器会从 CIC 字段中获取数字中继线的编号 p。如果能够为此 TUP/ISUP 消息找到设定的路由，服务器会按照配置项 UP_DPC,CIC_PCM 设定的路由，将此 TUP/ISUP 消息的 CIC 字段中的数字中继线编号 p 修改为 k，并将此 TUP 消息转发给客户端 j；如果不能找到，此 TUP/ISUP 消息将被丢弃。 ◆ 每当 SS7 信令服务器从客户端 j 收到一条 TUP/ISUP 消息时，服务器会从 CIC 字段中获取数字中继线的编号 k，并按照配置项 UP_DPC,CIC_PCM 设定的路由，将此 TUP/ISUP 消息的 CIC 字段中的数字中继线编号 k 修改为 p，然后将此 TUP/ISUP 消息转发给第 i 个 UP_DPC / DPC。如果不能找到匹配的路由，此 TUP/ISUP 消息将被丢弃。
注意事项	<ul style="list-style-type: none"> ● TUP 协议使用 [TUPRouter] 配置节，ISUP 协议使用 [ISUPRouter] 配置节； ● 如果没有设置 [UP_DPCInfo] 节，或者 MaxUP_DPC=0，则应使用 DPC,CIC_PCM : DPC[i],CIC_PCM[p]=IP[j],LocalPCM[k]； ● 客户端的应用程序如果使用驱动程序提供的 TUP/ISUP 协议，必须设置本配置项； ● 必须用本配置项对电信局分配的每条数字中继线逐个进行设置。
使用范例	<p>[ISUPRouter]</p> <p>DPC[0]:CIC_PCM[0]=IP[0],LocalPCM[0]</p> <p>DPC[1]:CIC_PCM[1]=IP[0],LocalPCM[1]</p>

4.3.10 设置服务器的启动特性

4.3.10.1 ConfigAsGateway

配置项	ConfigAsGateway
节	[Monitor]
书写格式	ConfigAsGateway=n

取值范围	n=0: 否 (缺省值) n=1: 是
功能描述	当 SS7 信令服务器启动时, 是否加载 SynCTI 驱动程序。如果服务器中没有安装任何 Synway 板卡, 应当将本配置项设置为 0; 如果安装了提供信令链路的 SHD 系列板卡, 必须将此配置项设置为 1, 当服务器启动时, 服务器会自动加载 SynCTI 驱动程序。 在加载驱动程序时, SS7 信令服务器使用的配置文件名为 ShConfig.ini 和 ShIndex.ini。服务器首先在当前目录下查找这两个配置文件。如果找不到, 则到 SynCTI 驱动程序安装目录下查找。如果还是找不到, 服务器启动失败, 会返回出错信息。有关 ShConfig.ini 和 ShIndex.ini 的配置方法请参考第 4 章。
注意事项	如果本配置项设置为 1, 则在服务器中运行的其它应用程序 (如果有) 不能加载 SynCTI 驱动程序。

4.3.11 设置服务器的显示特性

4.3.11.1 RcvMsuListMaxItem

请参见 [TxMsuListMaxItem](#)

4.3.11.2 TxMsuListMaxItem

配置项	RcvMsuListMaxItem TxMsuListMaxItem
节	[Monitor]
书写格式	RcvMsuListMaxItem=n //适用于用户部分 (客户端) 到服务器的消息 TxMsuListMaxItem=m //适用于信令点 (SP/STP) 到服务器的消息
取值范围	n,m: 允许显示的最大消息数量, 缺省值为 50 条。
功能描述	SS7 信令服务器可以将来自信令点 (SP/STP) 或用户部分 (客户端) 的消息显示在屏幕上, 本配置项用于设置可显示消息的最大值。 TxMsuListMaxItem 用于设置用户部分 (客户端) 到服务器的消息, RcvMsuListMaxItem 用于设置信令点 (SP/STP) 到服务器的消息。

4.3.12 Windows 7 号信令服务器专用配置项

4.3.12.1 EnDisplayL32Msu

配置项	EnDisplayL32Msu
节	[Monitor]
书写格式	EnDisplayL32Msu =n

取值范围	n=0: 关闭 (默认值) n=1: 开启
功能描述	显示发到对端消息

4.3.12.2 EnDisplayL23Msu

配置项	EnDisplayL23Msu
节	[Monitor]
书写格式	EnDisplayL23Msu =n
取值范围	n=0: 关闭 (默认值) n=1: 开启
功能描述	显示接收对端消息

4.3.12.3 ShowSNT

配置项	ShowSNT
节	[Monitor]
书写格式	ShowSNT=n
取值范围	n=0: 关闭 n=1: 开启 (默认值)
功能描述	显示 SNT 消息

4.3.12.4 ShowSNM

配置项	ShowSNM
节	[Monitor]
书写格式	ShowSNM=n
取值范围	n=0: 关闭 n=1: 开启 (默认值)
功能描述	显示 SNM 消息

4.3.12.5 AutoTranslate

配置项	AutoTranslate
节	[Monitor]
书写格式	AutoTranslate=n
取值范围	n=0: 关闭 (默认值) n=1: 开启
功能描述	自动翻译消息

4.3.12.6 WriteToFile

配置项	WriteToFile
节	[Monitor]
书写格式	WriteToFile=n
取值范围	n=0: 关闭 (默认值) n=1: 开启
功能描述	显示的消息写入文件

4.3.12.7 ShowDPCOPC

配置项	ShowDPCOPC
节	[Monitor]
书写格式	ShowDPCOPC=n
取值范围	n=0: 关闭 (默认值) n=1: 开启
功能描述	显示 DPC 和 OPC

4.3.12.8 LogToPcap

配置项	LogToPcap
节	[Monitor]

书写格式	LogToPcap=n
取值范围	n=0: 关闭 (默认值) n=1: 开启
功能描述	输出日志的同时输出一个由 MSU 文件转换得到的 PCAP 文件

4.3.12.9 HeartTimeOut

配置项	HeartTimeOut
节	[Ss7SystemConfig]
书写格式	HeartTimeOut=n
取值范围	n>0, 单位为秒, 默认值为 60
功能描述	设置 Windows 单机客户端和服务器之间的心跳超时时间。当应用系统比较繁忙时, 应尽量将该值改大, 以防客户端和服务器的连接主动断开。

4.3.12.10 HandleWithoutCic

配置项	HandleWithoutCic
节	[Ss7SystemConfig]
书写格式	HandleWithoutCic=b
取值范围	b=0: 否 (缺省值); b=1: 是。
功能描述	设置 SS7 不配置任何 CIC 时, 能否调用 SsmGetDecodeSs7Msu 获取解码信息。

4.3.12.11 OutputTimerStateToLog

配置项	OutputTimerStateToLog
节	[Monitor]
书写格式	OutputTimerStateToLog =b
取值范围	b=0: 否; b=1: 是 (缺省值)。
功能描述	设置 7 号信令服务器内部时间信息是否输出到日志中。

4.3.13 Linux 7 号信令服务器专用配置项

4.3.13.1 EnDisplayL32Msu

配置项	EnDisplayL32Msu
节	[Monitor]
书写格式	EnDisplayL32Msu=n
取值范围	n=0: 关闭 (默认值); n=1: 开启。
功能描述	显示发到对端消息。

4.3.13.2 EnDisplayL23Msu

配置项	EnDisplayL23Msu
节	[Monitor]
书写格式	EnDisplayL23Msu=n
取值范围	n=0: 关闭 (默认值); n=1: 开启。
功能描述	显示接受对端消息。

4.3.13.3 ShowMSU

配置项	ShowMSU
节	[Monitor]
书写格式	ShowMSU=n
取值范围	n=0: 关闭 (默认值); n=1: 开启。
功能描述	记录 MSU 消息到 msu.0.log 功能。

4.3.13.4 ShowStatus

配置项	ShowStatus
节	[Monitor]

书写格式	ShowStatus=n
取值范围	n=0: 关闭 (默认值); n=1: 开启。
功能描述	记录 Link 状态消息到 Ss7RunInfo.0.log 功能。

4.3.13.5 ShowSNT

配置项	ShowSNT
节	[Monitor]
书写格式	ShowSNT=n
取值范围	n=0: 关闭; n=1: 开启 (默认值)。
功能描述	使能 SNT 记录功能。

4.3.13.6 ShowSNM

配置项	ShowSNM
节	[Monitor]
书写格式	ShowSNM=n
取值范围	n=0: 关闭; n=1: 开启 (默认值)。
功能描述	使能 SNM 记录功能。

4.3.13.7 AutoTranslate

配置项	AutoTranslate
节	[Monitor]
书写格式	AutoTranslate=n
取值范围	n=0: 关闭 (默认值); n=1: 开启。
功能描述	使能 msu 翻译功能。

4.3.13.8 LogFileSize

配置项	LogFileSize
节	[Monitor]
书写格式	LogFileSize=n
取值范围	n>0 且 n 为整数， 默认值为 50。
功能描述	配置 msu.x.log 文件大小，单位 M，当文件大小超过 LogFileSize 就会写入下一个文件，文件个数由 TotalLogFiles 决定。

4.3.13.9 TotalLogFiles

配置项	TotalLogFiles
节	[Monitor]
书写格式	TotalLogFiles=n
取值范围	n>0 且 n 为整数， 默认值为 5。
功能描述	配置 msu.x.log 文件个数最大值 (n+1)，当写入的文件个数超出时，就会覆盖之前写入的文件。

4.3.13.10 LogPath

配置项	LogPath
节	[Monitor]
书写格式	LogPath=S
取值范围	S 是存放日志文件的路径，长度小于 256 个字符，缺省值为 /var/log/ss7log。
功能描述	设置 msu、pcap、Ss7RunInfo 日志存放的路径，必须是已存在路径。

4.3.13.11 LogToPcap

配置项	LogToPcap
节	[Monitor]
书写格式	LogToPcap=n

取值范围	n=0: 关闭 (默认值); n=1: 开启。
功能描述	记录 msu 消息到 PCAP 文件功能。

4.3.13.12 WriteToFile

配置项	WriteToFile
节	[Monitor]
书写格式	WriteToFile=n
取值范围	n=0: 关闭 (默认值); n=1: 开启。
功能描述	显示的消息写入 msu 文件。

4.3.13.13 ShowDPCOPC

配置项	ShowDPCOPC
节	[Monitor]
书写格式	ShowDPCOPC=n
取值范围	n=0: 关闭 (默认值); n=1: 开启。
功能描述	显示 DPC 和 OPC。

4.3.14 一个 Ss7Monitor 程序支持多个 Client 客户端的应用

数字卡使用 SS7(TUP/ISUP)时，需要同时开启 Test 程序(客户应用程序)和 SS7MONITOR 程序，其中 SS7MONITOR 程序负责 SS7 底层 MTP1-3 的传输。

4.3.14.1 客户端与服务器在同一台机器上

我们可以通过 [MultiCardMultiProcess](#) 这个配置项设置 SynCTI 驱动程序是否支持多进程方式的应用程序。如果 [MultiCardMultiProcess](#) 配成 1，那么在一台机器上可以启动多个 Test 程序(客户应用程序)。

具体的应用如下：

假设一台机器上插两块 SHD-120D-CT/PCI 板卡，设置为多卡多进程，[MultiCardMultiProcess](#) 配成 1。这时应该有两个独立的 shconfig.ini 和一个 ss7server.ini。首先按照一般的 ISUP 自环配置。然后修改配置文件：

1、Windows 系统配置：

ss7server.ini 修改前：

[Ss7ClientInfo]

MaxSs7Client=2

IP[0]=127.0.0.1

IP[1]=127.0.0.1

.....

ss7server.ini 修改后:

[Ss7ClientInfo]

MaxSs7Client=2

IP[0]=127.0.0.1-0

IP[1]=127.0.0.1-1

.....

Client[0]的 shconfig.ini 修改前:

[SS7]

Ss7ServerIP=127.0.0.1

LocalIP=127.0.0.1

修改后:

[SS7]

Ss7ServerIP=127.0.0.1

LocalIP=127.0.0.1-0

Client[1]的 shconfig.ini 修改前:

[SS7]

Ss7ServerIP=127.0.0.1

LocalIP=127.0.0.1

修改后:

[SS7]

Ss7ServerIP=127.0.0.1

LocalIP=127.0.0.1-1

2、Linux 系统配置:

Ss7Server.ini 修改后:

.....

[Ss7ClientInfo]

MaxSs7Client=2

IP[0]=127.0.0.1-0

IP[1]=127.0.0.1-1

.....

Client[0]的 shconfig.ini 修改后:

.....

[SS7]

Ss7ServerIP=127.0.0.1:5150-0

LocalIP=127.0.0.1-0

.....

Client[1]的 shconfig.ini 修改后:

.....

```
[SS7]
Ss7ServerIP=127.0.0.1:5150-1
LocalIP=127.0.0.1-1
....
```

注：Linux 系统配置时，需在 shconfig.ini 文件[SS7]小节下的配置项 Ss7ServerIP 的 IP 后增加“:端口号-客户端 ID”，如 127.0.0.1:5150-1。

修改完配置后保存配置，分别开启 ss7monitor.exe 和两个 Test。

4.3.14.2 客户端与服务器在不同的机器上

具体的应用及配置情况参考“[4.4.1.3 多机系统](#)”。

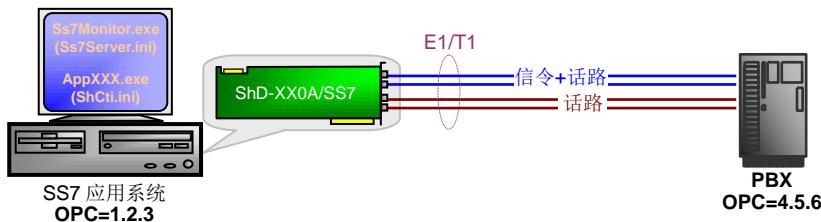
注：Linux 系统配置时，需在 shconfig.ini 文件[SS7]小节下的配置项 Ss7ServerIP 的 IP 后增加“:端口号-客户端 ID”，如 127.0.0.1:5150-1。

4.4 SS7 信令服务器的应用及配置实例

4.4.1 单 OPC/单 DPC

4.4.1.1 单机单卡的应用系统

在一台计算机上安装 1 片 SHD 系列板卡的最小应用系统，如下图所示：

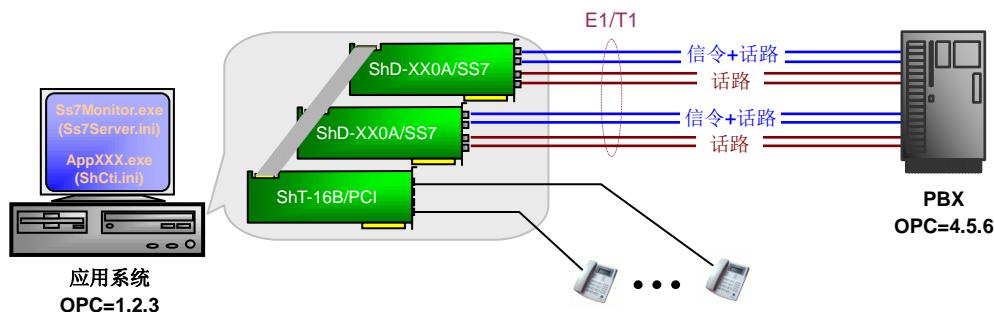


假设上述系统使用 SHD-60A-CT/SS7/PCI 板卡，第 1 条数字中继线的 16 时隙作为信令链路，其余 30 个时隙作为语音信道，第 2 条数字中继线的 30 个时隙均为话路，16 时隙不使用。以 TUP 协议为例，系统的配置为：

Ss7Server.ini	ShConfig.ini
<p>[Ss7SystemConfig]</p> OPC=1.2.3 ServerIP=127.0.0.1 [Ss7ClientInfo] MaxSs7Client=1 IP[0]=127.0.0.1 [Ss7PcmLinkInfo] MaxSs7Pcm=1 Ss7PcmLink[0]=IP[0],LocalPCM[0]	<p>.....</p> <p>[SS7]</p> Ss7ServerIP=127.0.0.1 //服务器的 IP 地址 LocalIP=127.0.0.1 //本机的 IP 地址

[LinkSetInfo] MaxLinkSet=1 LinkSet[0]=Ss7PcmLink[0] PcmNumber=2 PcmSSx[0]=7 //使用 SS7 信令方式
[DPCInfo] MaxDPC=1 DPC[0]=4.5.6,LinkSet[0]	PcmClockMode[0]=0 //主时钟，线路同步方式 Ss7SignalingTS[0]=16 //16 时隙作为信令时隙 UseTS16AsCircuit[0]=0
[TUPRouter] DPC[0]:CIC_PCM[0]=IP[0],LocalPCM[0] DPC[0]:CIC_PCM[1]=IP[0],LocalPCM[1]	PcmSSx[1]=7 //使用 SS7 信令方式 PcmClockMode[1]=2 //从时钟 Ss7SignalingTS[1]=16 //16 时隙作为信令时隙 UseTS16AsCircuit[1]=1

4.4.1.2 单机多卡的应用系统

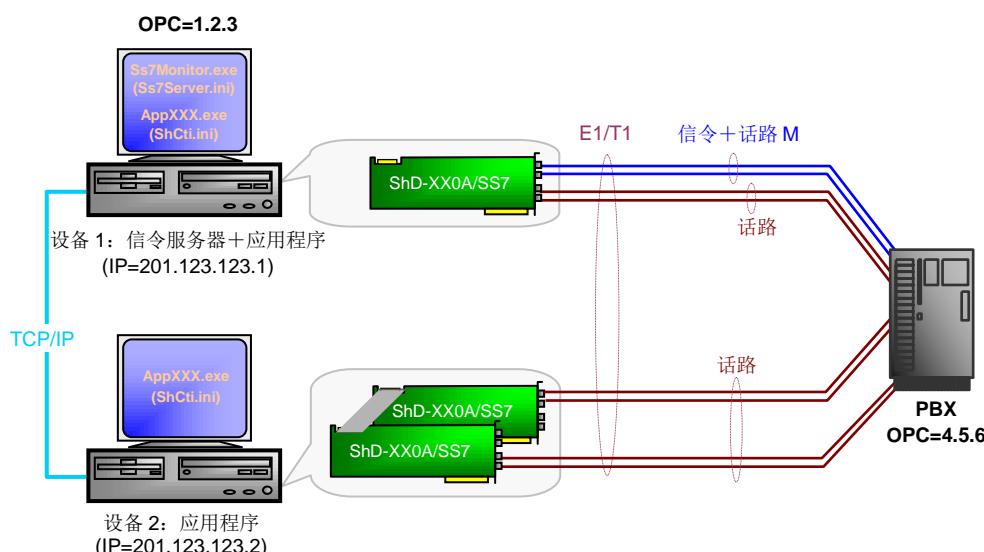


假设该系统包含 4 条 E1/T1 链路（由 2 块卡组成），1 块 16 路坐席卡，第 1 块卡的第 1 个 PCM 为信令/话路复用链路（信令时隙为 16 时隙），第 2 条 PCM 为纯话路；第 2 块卡第 1 个 PCM 为信令/话路复用链路（信令时隙为 16 时隙），第 2 条 PCM 为纯话路，以 TUP 为例，配置方法为：

Ss7Server.ini	ShConfig.ini
[Ss7SystemConfig] OPC=1.2.3 ServerIP=127.0.0.1 [SS7] Ss7ServerIP=127.0.0.1

MaxDPC=1	PcmClockMode[0]=0 //主时钟，线路同步方式
DPC[0]=4.5.6,LinkSet[0]	PcmSSx[1]=7
[TUPRouter]	PcmClockMode[1]=2 //从时钟
DPC[0]:CIC_PCM[0]=IP[0],LocalPCM[0]	[BoardId=1]
DPC[0]:CIC_PCM[1]=IP[0],LocalPCM[1]
DPC[0]:CIC_PCM[2]=IP[0],LocalPCM[2]	PcmNumber=2
DPC[0]:CIC_PCM[3]=IP[0],LocalPCM[3]	PcmSSx[0]=7
	PcmClockMode[0]=2 //从时钟
	PcmSSx[1]=7
	PcmClockMode[1]=2 //从时钟
	[BoardId=2] //设置 SHT-16B-CT/PCI

4.4.1.3 多机系统



假设应用系统系统包括 2 台设备，设备 1 上同时运行 SS7 信令服务器和应用程序，设备 2 为设备 1 的客户端。设备 1 包含 2 条数字中继线，其中，第 1 条数字中继线的 16 时隙为 SS7 信令链路，其余 30 个时隙为语音信道；第 2 条数字中继线的 16 时隙不使用，其余 30 个时隙为语音信道。设备 2 包含 2 条数字中继线，全部 30×2 个时隙都为语音信道。以 TUP 为例，配置方法如下：

- 设备 1：信令服务器和应用程序的配置文件

Ss7Server.ini	ShConfig.ini
[Ss7SystemConfig] OPC=1.2.3 [SS7]

ServerIP=201.123.123.1 [Ss7ClientInfo] MaxSs7Client=2 IP[0]= 201.123.123.1 IP[1]= 201.123.123.2 [Ss7PcmLinkInfo] MaxSs7Pcm=1 Ss7PcmLink[0]=IP[0],LocalPCM[0] [LinkSetInfo] MaxLinkSet=1 LinkSet[0]=Ss7PcmLink[0] [DPCInfo] MaxDPC=1 DPC[0]=4.5.6,LinkSet[0] [TUPRouter] DPC[0]:CIC_PCM[0]=IP[0],LocalPCM[0] DPC[0]:CIC_PCM[1]=IP[0],LocalPCM[1] DPC[0]:CIC_PCM[2]=IP[1],LocalPCM[0] DPC[0]:CIC_PCM[3]=IP[1],LocalPCM[1]	Ss7ServerIP=201.123.123.1 //服务器的 IP 地址 LocalIP=201.123.123.1 //本机的 IP 地址
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------

➤ 设备 2：应用程序的配置文件 ShConfig.ini

.....

[SS7]

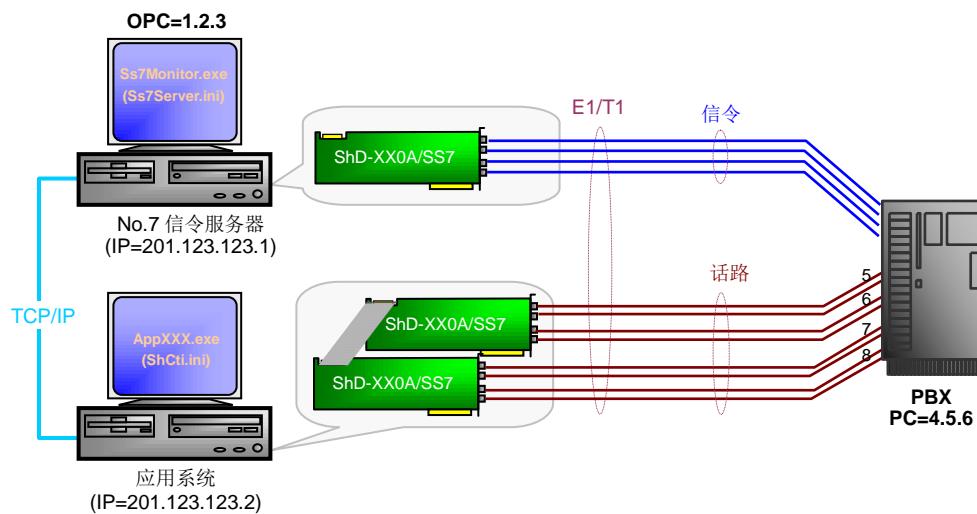
Ss7ServerIP=201.123.123.1 //服务器的 IP 地址

LocalIP=201.123.123.2 //本机的 IP 地址

.....

4.4.1.4 使用专用信令服务器的多机系统

如果应用系统对可靠性要求比较高，建议单独使用一台计算机来处理信令，应用程序在其它的一台或多台计算机上运行，如下图所示：



上图中，设备 1 只处理信令，不处理话路，因此，信令服务器上除了运行 SS7 信令服务器，还需要为只提供信令链路的板卡加载 SynCTI 驱动程序；应用系统作为客户端。假设信令服务器包含 2 条数字中继线，只使用其中的 16 时隙；设备 2 包含 4 条数字中继线，电信局在 TUP 协议中为其分配的编号为 5、6、7、8。配置方法为：

➤ SS7 信令服务器的配置

Ss7Server.ini	ShConfig.ini
<p>[Ss7SystemConfig]</p> <p>OPC=1.2.3</p> <p>ServerIP=201.123.123.1 //信令服务器的 IP 地址</p> <p>[Ss7ClientInfo]</p> <p>MaxSs7Client=2</p> <p>IP[0]= 201.123.123.1</p> <p>IP[1]= 201.123.123.2</p> <p>[Ss7PcmLinkInfo]</p> <p>MaxSs7Pcm=2</p> <p>Ss7PcmLink[0]=IP[0],LocalPCM[0]</p> <p>Ss7PcmLink[1]=IP[1],LocalPCM[1]</p> <p>[LinkSetInfo]</p> <p>MaxLinkSet=1</p> <p>LinkSet[0]=Ss7PcmLink[0]+ Ss7PcmLink[1]</p> <p>[DPCInfo]</p> <p>MaxDPC=1</p> <p>DPC[0]=4.5.6,LinkSet[0]</p> <p>[TUPRouter]</p> <p>DPC[0]:CIC_PCM[5]=IP[1],LocalPCM[0]</p> <p>DPC[0]:CIC_PCM[6]=IP[1],LocalPCM[1]</p> <p>DPC[0]:CIC_PCM[7]=IP[1],LocalPCM[2]</p>	<p>.....</p> <p>[SS7]</p> <p>Ss7ServerIP=201.123.123.1 //服务器的 IP 地址</p> <p>LocalIP=201.123.123.1 //客户端程序的 IP 地址</p> <p>.....</p>

DPC[0]:CIC_PCM[8]=IP[1],LocalPCM[3]	
[Monitor]	
ConfigAsGateway=1 //需要加载 SynCTI 驱动程序	

➤ 应用系统的配置文件 ShConfig.ini

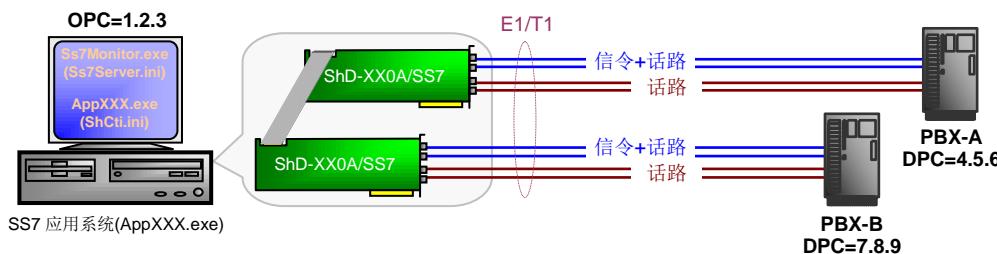
.....

[SS7]

Ss7ServerIP=201.123.123.1 //服务器的 IP 地址
 LocalIP=201.123.123.2 //本机的 IP 地址

.....

4.4.2 单 OPC/多 DPC

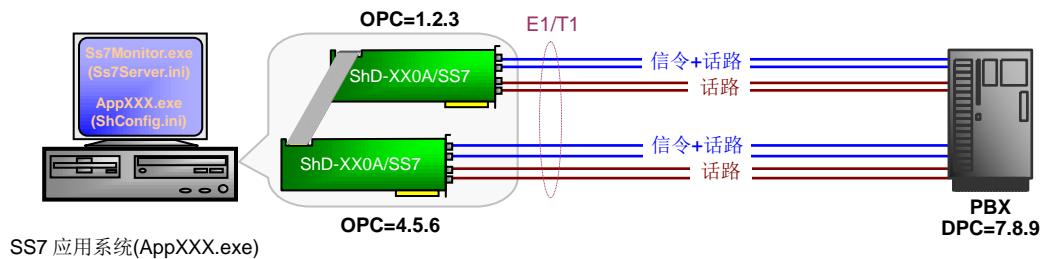


假设应用系统包含 2 片 SHD-60A-CT/PCI/SS7 板卡。每片板卡上的 2 条数字中继线的时隙 1~15、17~31 均为语音信道，第 1 条数字中继线上的第 16 时隙为信令链路，第 2 条数字中继线上的第 16 时隙不使用。2 片板卡分别连接到了具有不同 DPC 的 2 台交换机。以 TUP 为例，配置方法为：

Ss7Server.ini	ShConfig.ini
[Ss7SystemConfig] OPC=1.2.3 ServerIP=127.0.0.1 [Ss7ClientInfo] MaxSs7Client=1 IP[0]= 127.0.0.1 //客户端程序的 IP 地址 [Ss7PcmLinkInfo] MaxSs7Pcm=2 Ss7PcmLink[0]=IP[0],LocalPCM[0] Ss7PcmLink[1]=IP[0],LocalPCM[2] [LinkSetInfo] MaxLinkSet=2 LinkSet[0]=Ss7PcmLink[0] LinkSet[1]=Ss7PcmLink[1] [DPCInfo] MaxDPC=2 [PcmInfo] TotalPcm=4 Pcm[0]=0,0 Pcm[1]=0,1 Pcm[2]=1,0 Pcm[3]=1,1 [BoardId=0] PcmNumber=2 PcmSSx[0]=7 PcmSSx[1]=7 PcmClockMode[0]=0 //主时钟，线路同步方式 PcmClockMode[1]=2 //从时钟 [BoardId=1]

DPC[0]=4.5.6,LinkSet[0]
DPC[1]=7.8.9,LinkSet[1]	PcmNumber=2
[TUPRouter]	PcmSSx[0]=7
DPC[0]:CIC_PCM[0]=IP[0],LocalPCM[0]	PcmSSx[1]=7
DPC[0]:CIC_PCM[1]=IP[0],LocalPCM[1]	PcmClockMode[0]=2 //从时钟
DPC[1]:CIC_PCM[0]=IP[0],LocalPCM[2]	PcmClockMode[1]=2 //从时钟
DPC[1]:CIC_PCM[1]=IP[0],LocalPCM[3]	

4.4.3 多 OPC 的 SS7 应用系统



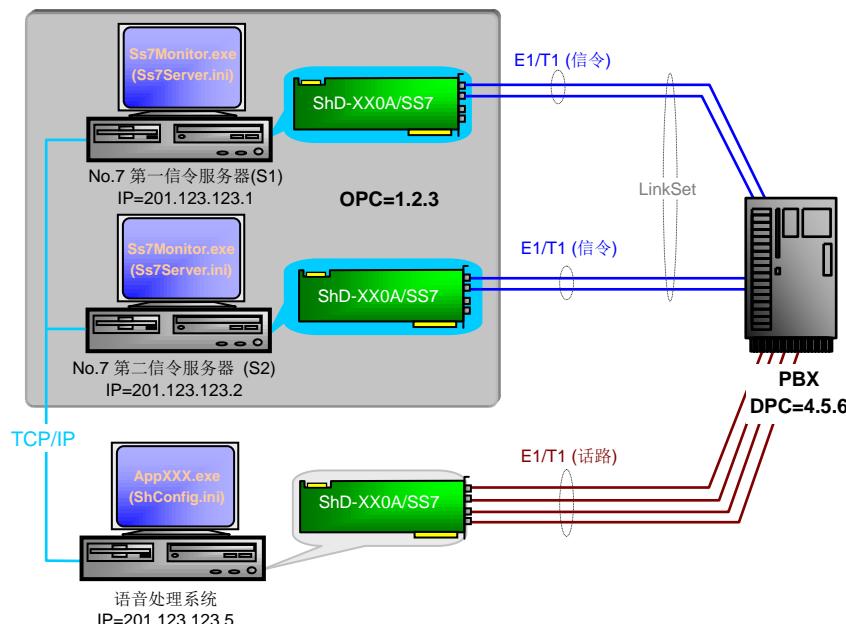
假设应用系统包含 2 片 SHD-60A-CT/PCI/SS7 板卡。第 1 个 PCM 为信令/话路复用链路(信令时隙为 16 时隙), 第 2 个 PCM 为纯话路, 第 1、2PCM 对应一个 OPC, 第 3 个 PCM 为信令/话路复用链路 (信令时隙为 16 时隙), 第 4 个 PCM 为纯话路, 第 3、4PCM 对应一个 OPC。以 TUP 协议为例, 系统的配置方法为:

Ss7Server.ini	ShConfig.ini
<p>[Ss7SystemConfig]</p> <p>OPC=1.2.3</p> <p>ServerIP=127.0.0.1</p> <p>[Ss7ClientInfo]</p> <p>MaxSs7Client=1</p> <p>IP[0]=127.0.0.1</p> <p>[Ss7PcmLinkInfo]</p> <p>MaxSs7Pcm=2</p> <p>Ss7PcmLink[0]=IP[0],LocalPCM[0]</p> <p>Ss7PcmLink[1]=IP[0],LocalPCM[2]</p> <p>[LinkSetInfo]</p> <p>MaxLinkSet=2</p> <p>LinkSet[0]=Ss7PcmLink[0],1.2.3</p> <p>LinkSet[1]=Ss7PcmLink[1],4.5.6</p> <p>[DPCInfo]</p> <p>MaxDPC=2</p> <p>DPC[0]=7.8.9,LinkSet[0]</p> <p>DPC[1]=7.8.9,LinkSet[1]</p>	<p>.....</p> <p>[PcmInfo]</p> <p>TotalPcm=4</p> <p>Pcm[0]=0,0</p> <p>Pcm[1]=0,1</p> <p>Pcm[2]=1,0</p> <p>Pcm[3]=1,1</p> <p>[BoardId=0]</p> <p>.....</p> <p>PcmNumber=2</p> <p>PcmSSx[0]=7</p> <p>PcmSSx[1]=7</p> <p>PcmClockMode[0]=0 //主时钟, 线路同步方式</p> <p>PcmClockMode[1]=2 //从时钟</p> <p>[BoardId=1]</p> <p>.....</p> <p>PcmNumber=2</p>

[TUPRouter] DPC[0]:CIC_PCM[0]=IP[0],LocalPCM[0] DPC[0]:CIC_PCM[1]=IP[0],LocalPCM[1] DPC[1]:CIC_PCM[0]=IP[0],LocalPCM[2] DPC[1]:CIC_PCM[1]=IP[0],LocalPCM[3]	PcmSSx[0]=7 PcmSSx[1]=7 PcmClockMode[0]=2 //从时钟 PcmClockMode[1]=2 //从时钟
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------

4.4.4 高可靠性的应用系统

Synway 推出的 SS7 信令服务器支持双机热备份的运行方式，可以为应用系统提供电信级的可靠性。一个典型的高可靠性应用系统如下图所示：



上图中，SS7 信令服务器有 2 台，每台均安装一片 SHD 系列板卡，但只使用数字中继线的信令时隙，不处理话路。2 台服务器同时工作，共用一个点码（即 **OPC=1.2.3**），为语音处理系统提供信令服务，其中一台作为主服务器(Master)，另外一台作为从服务器(Slave)。当主服务器出现故障时，从服务器会自动接管全部的信令处理业务。

主、从信令服务器的确定规则如下：

- 1、先启动的信令服务器为主服务器，后启动的信令服务器为从服务器。
- 2、主、从服务器均正常工作时，如果主服务器出现故障，从服务器自动转变为主服务器，待原主服务器故障恢复后再次投入使用时，会自动变为从服务器。
- 3、主、从服务器均正常工作时，可在主/从服务器之间进行手工切换，切换结束后，原主服务器变为从服务器，原从服务器变为主服务器。

语音处理系统由一台或多台安装 SHD 系列板卡的计算机组成，通过局域网从服务器获得信令服务，因此，客户端计算机上无需运行 SS7 信令服务器程序。

- 第一信令服务器(简称为 S1)的配置 (包括 **Ss7Server.ini** 和 **ShConfig.ini**)

Ss7Server.ini	ShConfig.ini
<p>[Ss7SystemConfig]</p> <p>OPC=1.2.3 ServerIP=201.123.123.1 //本服务器的 IP 地址 SecondServerIP=201.123.123.2 //S2 的 IP 地址</p> <p>[Ss7ClientInfo]</p> <p>MaxSs7Client=3 IP[0]=201.123.123.1 //信令链路处理程序 IP[1]=201.123.123.2 //信令链路处理程序 IP[2]=201.123.123.3 //客户端程序的 IP 地址</p> <p>[Ss7PcmLinkInfo]</p> <p>MaxSs7Pcm=2 Ss7PcmLink[0]=IP[0],LocalPCM[0] Ss7PcmLink[1]=IP[1],LocalPCM[0]</p> <p>[LinkSetInfo]</p> <p>MaxLinkSet=1 LinkSet[0]=Ss7PcmLink[0]+ Ss7PcmLink[1]</p> <p>[DPCInfo]</p> <p>MaxDPC=1 DPC[0]=4.5.6,LinkSet[0]</p> <p>[TUPRouter] //如不使用 TUP 协议则无需配置</p> <p>DPC[0]:CIC_PCM[0]=IP[2],LocalPCM[0] DPC[0]:CIC_PCM[1]=IP[2],LocalPCM[1]</p> <p>[ISUPRouter] //如不使用 ISUP 协议则无需配置</p> <p>DPC[0]:CIC_PCM[0]=IP[2],LocalPCM[0] DPC[0]:CIC_PCM[1]=IP[2],LocalPCM[1]</p> <p>[Monitor]</p> <p>ConfigAsGateway=1 //需要加载 SynCTI 驱动程序</p> <p>.....</p>	<p>.....</p> <p>[PcmInfo]</p> <p>TotalPcm=1 Pcm[0]=0,0</p> <p>[BoardId=0]</p> <p>.....</p> <p>PcmNumber=1 PcmSSx[0]=7 PcmClockMode[0]=0 //主时钟，线路同步方式</p> <p>[SS7]</p> <p>Ss7ServerIP=201.123.123.1 //S1 的 IP 地址 SecondServerIP=201.123.123.2 //S2 的 IP 地址 LocalIP= 201.123.123.1 //本机的 IP 地址</p>

➤ 第二信令服务器(简称为 S2)的配置 (包括 Ss7Server.ini 和 ShConfig.ini)

Ss7Server.ini	ShConfig.ini
<p>[Ss7SystemConfig]</p> <p>OPC=1.2.3 ServerIP=201.123.123.2 //本服务器的 IP 地址 SecondServerIP=201.123.123.1 //S1 的 IP 地址</p> <p>[Ss7ClientInfo]</p> <p>MaxSs7Client=3 IP[0]= 201.123.123.1 //信令链路处理程序 IP[1]= 201.123.123.2 //信令链路处理程序 IP[2]= 201.123.123.3 //客户端程序的 IP 地址</p> <p>[Ss7PcmLinkInfo]</p> <p>MaxSs7Pcm=2 Ss7PcmLink[0]=IP[1],LocalPCM[0] Ss7PcmLink[1]=IP[0],LocalPCM[0]</p>	<p>.....</p> <p>[PcmInfo]</p> <p>TotalPcm=1 Pcm[0]=0,0</p> <p>[BoardId=0]</p> <p>.....</p> <p>PcmNumber=1 PcmSSx[0]=7 PcmClockMode[0]=0 //主时钟，线路同步方式</p>

[LinkSetInfo] MaxLinkSet=1 LinkSet[0]=Ss7PcmLink[0]+ Ss7PcmLink[1] [DPCInfo] MaxDPC=1 DPC[0]=4.5.6,LinkSet[0] [TUPRouter] //如不使用 TUP 协议则无需配置 DPC[0]:CIC_PCM[0]=IP[2],LocalPCM[0] DPC[0]:CIC_PCM[1]=IP[2],LocalPCM[1] [ISUPRouter] //如不使用 ISUP 协议则无需配置 DPC[0]:CIC_PCM[0]=IP[2],LocalPCM[0] DPC[0]:CIC_PCM[1]=IP[2],LocalPCM[1] [Monitor] ConfigAsGateway=1 //需要加载 SyncTI 驱动程序 	[SS7] Ss7ServerIP=201.123.123.1 //S1 的 IP 地址 SecondServerIP=201.123.123.2 //S2 的 IP 地址 LocalIP= 201.123.123.2 //本机的 IP 地址
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------

➤ 客户端程序的配置 (ShConfig.ini)

.....

[PcmInfo]

TotalPcm=1
 Pcm[0]=0,0

[BoardId=0]

.....

PcmNumber=2

PcmSSx[0]=7

PcmSSx[1]=7

PcmClockMode[0]=0 //主时钟, 线路同步方式

PcmClockMode[1]=2 //从时钟

[SS7]

Ss7ServerIP=201.123.123.1 //S1 的 IP 地址
 SecondServerIP=201.123.123.2 //S2 的 IP 地址
 LocalIP=201.123.123.3 //本机的 IP 地址

.....

对于 Linux 系统, 默认是使用 TCP/IP 协议进行信息传递的, 如需使用 SCTP, 则需在完成上面的配置后, 增加下面两个操作:

1、加载操作系统的 Sctp 模块

命令: modprobe sctp

如果加载成功, 使用 lsmod 命令便可看到 sctp 模块, 如:

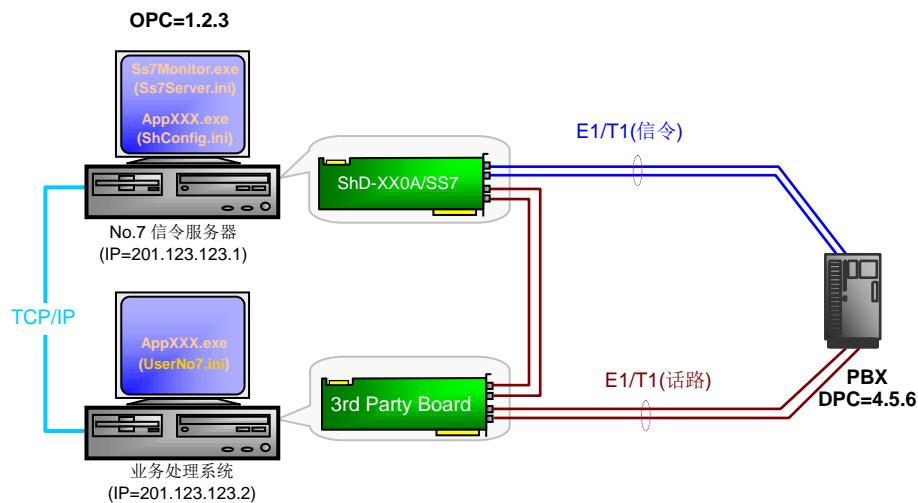
```
[root@localhost out]# lsmod |grep sctp
sctp                  153737  1 [unsafe]
```

2、修改配置文件

在 ShConfig.ini 文件 [SS7] 小节下添加 UseSctp=1

在 Ss7Server.ini 文件 [Ss7SystemConfig] 小节下添加 UseSctp=1

4.4.5 为第三方板卡提供 SS7 信令服务



上图中，SS7 信令服务器使用 Synway 板卡处理信令，业务处理系统使用第三方板卡，信令信息通过 TCP/IP 协议从服务器获得。假设接入到 SS7 信令服务器的数字中继线中还包含语音时隙，可以通过板卡的语音交换功能，通过设置配置项 [LoadShp_a3AsSIU](#)，将语音时隙交换到另一个数字中继线接口，并连接到应用系统，如图中虚线所示。以 TUP 协议为例，系统的配置方法为：

- SS7 信令服务器的配置（Ss7Server.ini 和 ShConfig.ini）

Ss7Server.ini	ShConfig.ini
<p>[Ss7SystemConfig]</p> <p>OPC=1.2.3</p> <p>ServerIP=201.123.123.1</p> <p>[Ss7ClientInfo]</p> <p>MaxSs7Client=2</p> <p>IP[0]= 201.123.123.1</p> <p>IP[1]= 201.123.123.2</p> <p>[Ss7PcmLinkInfo]</p> <p>MaxSs7Pcm=1</p> <p>Ss7PcmLink[0]=IP[0],LocalPCM[0]</p> <p>[LinkSetInfo]</p> <p>MaxLinkSet=1</p> <p>LinkSet[0]=Ss7PcmLink[0]</p> <p>[DPCInfo]</p> <p>MaxDPC=1</p> <p>DPC[0]=4.5.6,LinkSet[0]</p> <p>[TUPRouter]</p>	<p>.....</p> <p>[SystemConfig]</p> <p>LoadShp_a3AsSIU=1 //输出语音时隙</p> <p>.....</p> <p>[PcmInfo]</p> <p>TotalPcm=1</p> <p>Pcm[0]=0,0</p> <p>[BoardId=0]</p> <p>PcmNumber=1</p> <p>PcmSSx[0]=7</p> <p>PcmClockMode[0]=0 //主时钟，线路同步方式</p>

DPC[0]:CIC_PCM[0]=IP[1],LocalPCM[0]	
[Monitor]	
ConfigAsGateway=1 //需要加载 SynCTI 驱动程序	

➤ 基于第三方板卡的应用程序使用 SS7 信令服务器的客户端 SDK 时的配置文件 UserNo7.ini:

[SystemConfig]

```
TotalPcm=2
DefaultPcmLinkStatus=0x0000
.....
```

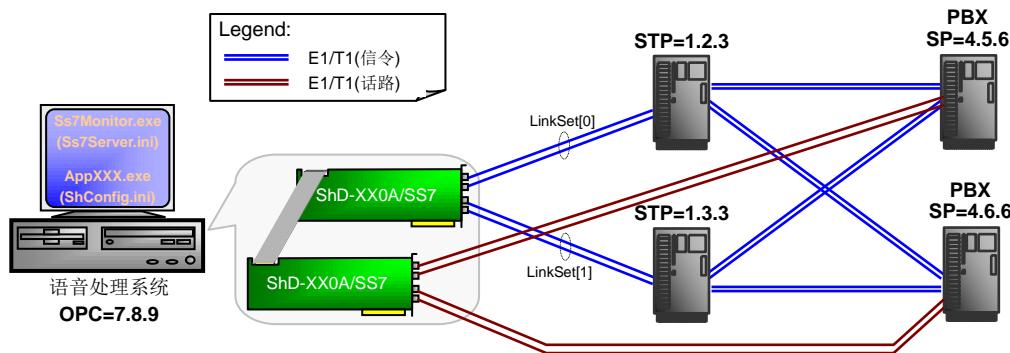
[AppChToPcmTsTable]

```
TotalAppCh=60
AppCh[0]=0,1..15
AppCh[15]=0,17..31
AppCh[30]=1,1..15
AppCh[45]=1,17..31
```

[SS7]

```
Ss7ServerIP=201.123.123.1 //信令服务器 IP
LocalIP=201.123.123.2 //本地 IP
.....
```

4.4.6 准直联的应用系统



上图中，假设应用系统由 2 片 SHD-120A/SS7 板卡组成，第 1 片板卡只处理 4 条数字中继线上的信令（PCM 的逻辑编号为 0、1、2、3，并且信令时隙均为 16 时隙），第 2 片板卡处理 4 条数字中继线上的话路。以 TUP 为例，应用系统的配置如下：

Ss7Server.ini	ShConfig.ini
[Ss7SystemConfig] OPC=7.8.9 ServerIP=127.0.0.1 [Ss7ClientInfo] MaxSs7Client=1 [PcmInfo] TotalPcm=8

<p>IP[0]=127.0.0.1</p> <p>[Ss7PcmLinkInfo]</p> <p>MaxSs7Pcm=4 Ss7PcmLink[0]=IP[0],LocalPCM[0] Ss7PcmLink[1]=IP[0],LocalPCM[1] Ss7PcmLink[2]=IP[0],LocalPCM[2] Ss7PcmLink[3]=IP[0],LocalPCM[3]</p> <p>[LinkSetInfo]</p> <p>MaxLinkSet=2</p> <p>LinkSet[0]=Ss7PcmLink[0]+ Ss7PcmLink[1] LinkSet[1]=Ss7PcmLink[2]+ Ss7PcmLink[3]</p> <p>[DPCInfo]</p> <p>MaxDPC=2 DPC[0]=1.2.3,LinkSet[0] DPC[1]=1.3.3,LinkSet[1]</p> <p>[UP_DPCInfo]</p> <p>MaxUP_DPC=2 UP_DPC[0]=4.5.6,LinkSet[0]+LinkSet[1] //负荷分担 UP_DPC[1]=4.6.6,LinkSet[1]+LinkSet[0],1 //第 1 组信令链路组为正常路由，第 0 路由为迂回路由</p> <p>[TUPRouter]</p> <p>UP_DPC[0]:CIC_PCM[1]=IP[0],LocalPCM[4] UP_DPC[0]:CIC_PCM[2]=IP[0],LocalPCM[5] UP_DPC[1]:CIC_PCM[1]=IP[0],LocalPCM[6] UP_DPC[1]:CIC_PCM[2]=IP[0],LocalPCM[7]</p> <p>.....</p>	<p>Pcm[0]=0,0</p> <p>Pcm[1]=0,1</p> <p>Pcm[2]=0,2</p> <p>Pcm[3]=0,3</p> <p>Pcm[4]=1,0</p> <p>Pcm[5]=1,1</p> <p>Pcm[6]=1,2</p> <p>Pcm[7]=1,3</p> <p>[BoardId=0]</p> <p>PcmNumber=4</p> <p>PcmSSx[0]=7</p> <p>PcmSSx[1]=7</p> <p>PcmSSx[2]=7</p> <p>PcmSSx[3]=7</p> <p>PcmClockMode[0]=0 //主时钟，线路同步方式</p> <p>PcmClockMode[1]=2 //从时钟</p> <p>PcmClockMode[2]=2 //从时钟</p> <p>PcmClockMode[3]=2 //从时钟</p> <p>Ss7SignalingTS[0]=16 //16 时隙作为信令时隙</p> <p>UseTS16AsCircuit[0]=0</p> <p>Ss7SignalingTS[1]=16 //16 时隙作为信令时隙</p> <p>UseTS16AsCircuit[1]=0</p> <p>Ss7SignalingTS[2]=16 //16 时隙作为信令时隙</p> <p>UseTS16AsCircuit[2]=0</p> <p>Ss7SignalingTS[3]=16 //16 时隙作为信令时隙</p> <p>UseTS16AsCircuit[3]=0</p> <p>.....</p> <p>[BoardId=1]</p> <p>PcmNumber=4</p> <p>PcmSSx[0]=7</p> <p>PcmSSx[1]=7</p> <p>PcmSSx[2]=7</p> <p>PcmSSx[3]=7</p> <p>PcmClockMode[0]=2 //从时钟</p> <p>PcmClockMode[1]=2 //从时钟</p> <p>PcmClockMode[2]=2 //从时钟</p> <p>PcmClockMode[3]=2 //从时钟</p> <p>.....</p>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

附录1 ISDN 通道释放原因值表

返回值	取值含义	支持协议
1	未分配的号码	DSS1, ISUP
2	无路由到达规定的转接网络	DSS1, ISUP
3	无路由到达终点	DSS1, ISUP
4	发送特殊的信息音	ISUP
5	转接前缀拨号错误	ISUP
6	不可接受的通路	DSS1, ISUP
7	呼叫已经给出并正在已建立的通路上递交	DSS1, ISUP
8	先占	DSS1, ISUP
9	先占电路留作重新使用	ISUP
16	正常的呼叫清除	DSS1, ISUP
17	用户忙	DSS1, ISUP
18	用户无响应	DSS1, ISUP
19	用户无应答（用户已提醒）	DSS1, ISUP
20	用户缺席	DSS1, ISUP
21	呼叫拒绝	DSS1, ISUP
22	号码变更	DSS1, ISUP
26	清除未选择的用户	DSS1, ISUP
27	终点故障	DSS1, ISUP
28	无效的号码格式	DSS1, ISUP
29	性能被拒绝	DSS1, ISUP
30	对 STATUS ENQUIRY 的响应	DSS1
31	正常未规定	DSS1, ISUP
34	无可用的电路/通路	DSS1, ISUP
38	网络失序	DSS1, ISUP
39	永久帧方式连接未开放业务	DSS1
40	永久帧方式连接可运营	DSS1
41	临时故障	DSS1, ISUP
42	交换设备拥塞	DSS1, ISUP
43	接入信息被丢弃	DSS1, ISUP
44	请求的电路/通路不可用	DSS1, ISUP
46	优先呼叫阻塞	DSS1, ISUP
47	资源不可用，未规定	DSS1, ISUP
49	业务质量不可用	DSS1
50	未预订所请求的性能	DSS1, ISUP
53	CUG 内呼出呼叫阻塞	DSS1, ISUP
55	CUG 内呼入呼叫阻塞	DSS1, ISUP
57	承载能力未认可	DSS1, ISUP

58	目前尚无可用的承载能力	DSS1, ISUP
62	分配的呼出接入信息与用户级别不一致	DSS1, ISUP
63	无适用的业务或任选项目，未规定	DSS1, ISUP
65	承载能力未实施	DSS1, ISUP
66	通路类型未实施	DSS1
69	请求的性能未实施	DSS1, ISUP
70	只有受限的数字信息承载能力可用	DSS1, ISUP
79	业务或任选未实施，未规定	DSS1, ISUP
81	无效的呼叫参考值	DSS1
82	所标识的通路不存在	DSS1
83	存在暂停的呼叫，但无所用的呼叫身份	DSS1
84	呼叫身份在使用	DSS1
85	无呼叫暂停	DSS1
86	具有所请求的呼叫身份的呼叫已被清除	DSS1
87	用户不是 CUG 成员	DSS1, ISUP
88	不兼容的终点	DSS1, ISUP
90	不存在的 CUG	DSS1, ISUP
91	无效的转接网络选择	DSS1, ISUP
95	无效的消息，未规定	DSS1, ISUP
96	必选信息单元丢失	DSS1
97	消息类型不存在或未实施	DSS1, ISUP
98	消息与呼叫状态不符或消息类型不存在或未实施	DSS1
99	信息单元不存在或未实施	DSS1, ISUP
100	无效的信息单元内容	DSS1
101	消息与呼叫状态不符	DSS1
102	定时器超时的恢复	DSS1, ISUP
103	参数不存在或未实施，通过	ISUP
110	带不可识别参数的消息，丢弃	ISUP
111	协议差错，未规定	DSS1, ISUP
127	互通，未规定	DSS1, ISUP
128	T303 超时	Synway Define
129	T304 超时	Synway Define
130	T310 超时	Synway Define
131	T308 第二次超时	Synway Define
132	网络忙	Synway Define
133	电路重启	Synway Define
134	临时故障	Synway Define
135	数据链路故障	Synway Define
136	摘机后连接失败	Synway Define

137	摘机超时	Synway Define
-----	------	---------------

附录2 ISUP 可选参数列表和部分参数介绍

参数名	编码	说明															
被叫用户号码	0x04		8	7	6	5	4	3	2	1							
		1	奇/偶	地址性质表示语													
		2	INN 表示语	编号计划			备用										
		3	第 2 个地址信号			第 1 个地址信号											
															
		n	填充码 (如果需要)				第 n 个地址信号										
		1) 奇偶表示语															
		0: 地址信号为偶数															
		1: 地址信号为奇数															
		2) 地址性质表示语															
		0000001: 用户号码															
		0000011: 国内号码															
		0000100: 国际号码															
		3) 内部网号码表示语 (INN 表示语)															
		0: 允许选路到内部网号码															
		1: 不允许选路到内部网号码															
		4) 编号计划表示语															
		001: ISDN(电话)编号计划 (建议 E.164, E.163)															
		011: 数据编号计划 (建议 X.164)															
		100: 用户电报编号计划 (建议 F.164)															
		5) 地址信号															
		从第 3 个字节开始表示号码															
		6) 填充码															
		如果地址信号为奇数，则在最后一个地址信号后插入填充码 (0000)															

主叫用户号码	0x0a		8	7	6	5	4	3	2	1											
		1	奇/偶	地址性质表示语																	
		2	号码不全	编号计划表示语			提供表示语		鉴别表示语												
		3	第 2 个地址信号			第 1 个地址信号															
																			
		n	填充码（如果需要）			第 n 个地址信号															
1) 奇偶表示语																					
0: 地址信号为偶数																					
1: 地址信号为奇数																					
2) 地址性质表示语																					
0000001: 用户号码																					
0000011: 国内号码																					
0000100: 国际号码																					
3) 主叫用户号码不全表示语 (NI)																					
0: 号码全																					
1: 号码不全																					
4) 编号计划表示语																					
001: ISDN(电话)编号计划 (建议 E.164, E.163)																					
011: 数据编号计划 (建议 X.164)																					
100: 用户电报编号计划 (建议 F.164)																					
5) 限制地址提供表示语																					
00: 允许提供																					
01: 限制提供																					
10: 地址不可用																					
6) 鉴别表示语																					
00: 用户提供, 不核实																					
01: 用户提供, 核实通过																					
10: 用户提供, 核实失败																					
11: 网络提供																					
7) 地址信号																					
从第 3 个字节开始表示号码																					
8) 填充码																					
如果地址信号为奇数, 则在最后一个地址信号后插入填充码 (0000)																					

后向呼叫表示语	0x11		8	7	6	5	4	3	2	1
		1	H	G	F	E	D	C	B	A
		2	P	O	N	M	L	K	J	I
1) 比特 BA: 计费表示语 00: 无指示 01: 不计费 10: 计费 2) 比特 DC: 被叫用户状态表示语 00: 无指示 01: 用户空 10: 空时连接 3) 比特 FE: 被叫用户类别表示语 00: 无指示 01: 普通用户 10: 付费电话 4) 比特 HG: 端至端方式表示语 00: 端至端方式不可用 (只有逐段转发方式可用) 01: 传递方式可用 10: SCCP 方式可用 11: 传递方式和 SCCP 方式可用 5) 比特 I: 互通表示语 0: 未遇到互通 1: 遇到互通 6) 比特 J: 端至端信息表示语 0: 无端至端信息可用 1: 端至端信息可用 7) 比特 K: ISDN 用户部分表示语 0: 不是所有方向都使用 ISDN 用户部分 1: 所有方向都使用 ISDN 用户部分 8) 比特 L: 保持表示语 0: 未请求保持 1: 请求保持 9) 比特 M: ISDN 接入表示语 0: 终端接入非 ISDN 1: 终端接入 ISDN 10) 比特 N: 回声控制装置表示语 0: 未包括来话半回声控制装置 1: 包括来话半回声控制装置 11) 比特 PO: SCCP 方式表示语 00: 无指示 01: 无连接方式可用 10: 面向连接方式可用 11: 无连接和面向连接方式可用										

呼叫变更信息	0x36	<table border="1"> <tr><td></td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr> <tr><td>1</td><td>H</td><td>G</td><td>F</td><td>E</td><td>D</td><td>C</td><td>B</td><td>A</td></tr> </table>		8	7	6	5	4	3	2	1	1	H	G	F	E	D	C	B	A																							
	8	7	6	5	4	3	2	1																																			
1	H	G	F	E	D	C	B	A																																			
<p>1) 比特 CBA: 通知预定任选</p> <p>000: 不知道</p> <p>001: 提供不允许</p> <p>010: 提供允许, 带有改发号码</p> <p>011: 提供允许, 不带有改发号码</p> <p>2) 比特 GFED: 改发原因</p> <p>0000: 不知道</p> <p>0001: 用户忙</p> <p>0010: 无应答</p> <p>0011: 无条件</p> <p>0100: 在提示期间转向</p> <p>0101: 转向立即响应</p> <p>0110: 移动用户不可达</p>																																											
主叫用户类别	0x09	<table border="1"> <tr><td></td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr> <tr><td>1</td><td colspan="8">主叫用户类别</td></tr> </table>		8	7	6	5	4	3	2	1	1	主叫用户类别																														
	8	7	6	5	4	3	2	1																																			
1	主叫用户类别																																										
<p>00001001: 国内话务员 (国内用)</p> <p>00001010: 普通用户用于长途局和长途局, 以及长途局和市话局之间</p> <p>00001011: 优先主叫用户</p> <p>00001100: 数据呼叫 (音频带数据)</p> <p>00001101: 测试呼叫</p> <p>00001111: 付费电话</p> <p>11111000: 普通主叫用户, 用于市话局之间</p>																																											
原因表示语	0x12	<table border="1"> <tr><td></td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr> <tr><td>1</td><td colspan="2">扩展表示语</td><td colspan="2">编码标准</td><td>备用</td><td colspan="4" rowspan="2">位置</td></tr> <tr><td>2</td><td colspan="2">扩展表示语</td><td colspan="7">原因值</td></tr> <tr><td>3</td><td colspan="8" rowspan="3">诊断 (如果存在)</td><td rowspan="3"></td></tr> <tr><td>...</td></tr> <tr><td>n</td></tr> </table>		8	7	6	5	4	3	2	1	1	扩展表示语		编码标准		备用	位置				2	扩展表示语		原因值							3	诊断 (如果存在)									...	n
	8	7	6	5	4	3	2	1																																			
1	扩展表示语		编码标准		备用	位置																																					
2	扩展表示语		原因值																																								
3	诊断 (如果存在)																																										
...																																											
n																																											
<p>1) 扩展表示语</p> <p>0: 八位位组延续到下一个八位位组</p> <p>1: 最后的八位位组</p> <p>2) 编码标准</p> <p>00: CCITT 标准编码</p> <p>01: ISO/IEC 标准</p> <p>10: 国内标准</p> <p>11: 专用于所识别的位置的标准</p> <p>3) 位置</p> <p>0000: 用户</p> <p>0001: 为本地用户服务的专用网</p> <p>0010: 为本地用户服务的公用网</p> <p>0011: 转接网</p>																																											

	0100: 为远端用户服务的公用网 0101: 为远端用户服务的专用网 0111: 国际网 1010: 超出互通点的范围 4) 原因值(部分) 000 0001: 未分配的号码 000 0010: 无路由到指定的转接网 000 0011: 无路由到目的地 000 0100: 发送专用信息音 000 0101: 误拨长途字冠 001 0000: 正常的呼叫拆线 001 0001: 用户忙 001 0010: 用户未响应 001 0011: 用户未应答 001 0101: 用户拒收 001 0110: 号码改变 001 1011: 目的地不可达 001 1100: 无效的号码格式 001 1101: 性能拒绝 001 1111: 正常 010 0010: 无电路可工作 010 0110: 网络未正常工作 010 1001: 临时故障 010 1010: 交换设备拥塞 010 1100: 所请求的电路不可用 010 1111: 资源不可用 101 1000: 不兼容的目的地
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

回声控制信息	0x37	<table border="1"><tr><td></td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr><tr><td>1</td><td>H</td><td>G</td><td>F</td><td>E</td><td>D</td><td>C</td><td>B</td><td>A</td></tr></table>		8	7	6	5	4	3	2	1	1	H	G	F	E	D	C	B	A	1) 比特 BA : 去话半回声控制装置响应表示语 00: 无信息 01: 未包括去话半回声控制装置 10: 包括去话半回声控制装置
	8	7	6	5	4	3	2	1													
1	H	G	F	E	D	C	B	A													
	2) 比特 DC : 来话半回声控制装置响应表示语 00: 无信息 01: 未包括来话半回声控制装置 10: 包括来话半回声控制装置																				
			3) 比特 FE : 去话半回声控制装置请求表示语 00: 无信息 01: 去话半回声控制装置激活请求 10: 去话半回声控制装置取消激活请求																		
			4) 比特 HG : 来话半回声控制装置请求表示语 00: 无信息 01: 来话半回声控制装置激活请求 10: 来话半回声控制装置取消激活请求																		

前向呼叫表示语	0x07		8	7	6	5	4	3	2	1
		1	H	G	F	E	D	C	B	A
		2	P	O	N	M	L	K	J	I
<p>1) 比特 A: 国内/国际呼叫表示语 0: 呼叫按国内呼叫处理 1: 呼叫按国际呼叫处理</p> <p>2) 比特 CB: 端到端方式表示语 00: 无端到端方式可用 01: 传递方式可用 10: SCCP 方式可用 11: 传递方式和 SCCP 方式可用</p> <p>3) 比特 D: 互通表示语 0: 未遇到互通 1: 遇到互通</p> <p>4) 比特 E: 端到端信息表示语 0: 无端到端信息可用 1: 端到端信息可用</p> <p>5) 比特 F: ISDN 用户部分表示语 0: 不是所有方向都使用 ISDN 用户部分 1: 所有方向都使用 ISDN 用户部分</p> <p>6) 比特 HG: ISDN 用户部分优选表示语 00: 所有方向都优选 ISDN 用户部分 01: 不是所有方向都需要 ISDN 用户部分 10: 所有方向都需要 ISDN 用户部分</p> <p>7) 比特 I: ISDN 接入表示语 0: 始发接入非 ISDN 1: 始发接入 ISDN</p> <p>8) 比特 KJ: SCCP 方式表示语 00: 无指示 01: 无连接方式可用 10: 面向连接方式可用 11: 不连接和面向连接方式可用</p>										

通用通知表示语	0x2c		8	7	6	5	4	3	2	1
		1	扩充	通知表示语						

1) 扩充表示语

0: 信息在下一个八位位组中继续

1: 最后的八位位组

2) 通知表示语

0000000: 用户暂停

0000001: 用户恢复

0000010: 承载业务改变

0000100: 呼叫完成延时

1000010: 会议建立

1000011: 会议断开

1000100: 增加他方

1000101: 隔离

1000110: 重新连接

1001000: 他方隔离

1001001: 他方分离

1001011: 会议浮动

1100000: 呼叫是一个等待呼叫

1101000: 变更已激活

1101001: 呼叫变换, 提示

1101010: 呼叫变换, 执行

1111001: 远端保持

1111010: 远端恢复

1111011: 呼叫正在变更

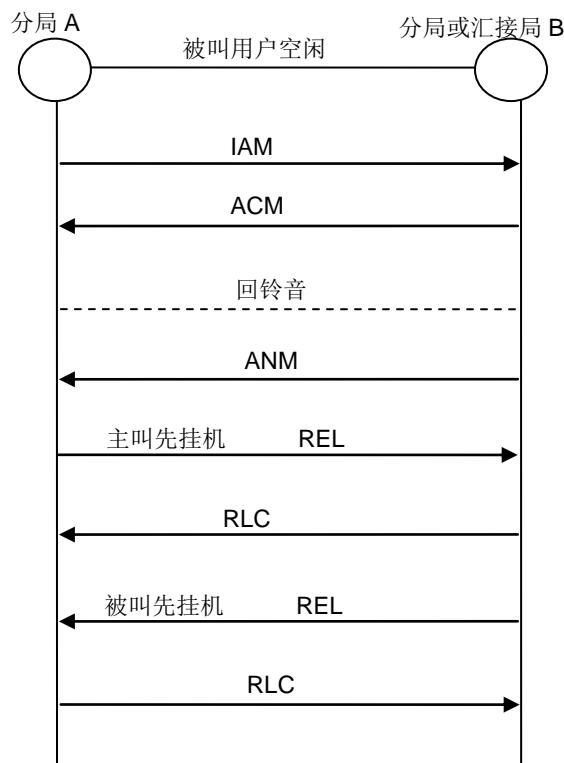
位置号码	0x3f		8	7	6	5	4	3	2	1							
		1	奇/偶	地址性质表示语													
		2	INN/NI 表示语	编号计划表示语			提供表示语		鉴别表示语								
		3	第 2 个地址信号			第 1 个地址信号											
															
		n	填充码（如果需要）			第 n 个地址信号											
1) 奇/偶表示语 0: 地址信号为偶数 1: 地址信号为奇数 2) 地址性质表示语 0000011: 国内号码 0000100: 国际号码 3) 内部网号表示语 (INN) 0: 选路到内部号码允许 1: 选路到内部号码不允许 4) 编号计划表示语 001: ISDN 编号计划 (建议 E.164) 011: 数据编号计划 (建议 X.121) 100: 用户电报编号计划 (建议 F.69) 101: 专用编号计划 5) 提供表示语 00: 提供允许 01: 提供限制 10: 地址不可用 6) 鉴别表示语 01: 用户提供, 核实和通过 11: 网络提供 7) 地址信号 从第 3 个字节开始表示号码 8) 填充码 如果地址信号为奇数, 则在最后一个地址信号后插入填充码 (0000)																	

连接性质表示语	0x06	<table border="1"> <tr><td></td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr> <tr><td>1</td><td>H</td><td>G</td><td>F</td><td>E</td><td>D</td><td>C</td><td>B</td><td>A</td></tr> </table>		8	7	6	5	4	3	2	1	1	H	G	F	E	D	C	B	A	1) 比特 BA: 卫星表示语 00: 连接中无卫星电路 01: 连接中有一段卫星电路 10: 连接中有两段卫星电路 2) 比特 DC: 导通检验表示语 00: 不需要导通检验 01: 在该电路上需要导通检验 10: 在前一电路上完成导通检验 3) 比特 E: 回声控制装置表示语 0: 未包括去话半回声控制装置 1: 包括去话半回声控制装置
	8	7	6	5	4	3	2	1													
1	H	G	F	E	D	C	B	A													
<table border="1"> <tr><td></td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr> <tr><td>1</td><td>H</td><td>G</td><td>F</td><td>E</td><td>D</td><td>C</td><td>B</td><td>A</td></tr> </table>		8	7	6	5	4	3	2	1	1	H	G	F	E	D	C	B	A	1) 比特 A: 带内信息表示语 0: 无指示 1: 带内信息或适当的码型目前可用 2) 比特 B: 呼叫变更可能发生表示语 0: 无指示 1: 呼叫变更可能发生 3) 比特 C: 简单分段表示语 0: 将不发送附加的信息 1: 将在分段消息中发送附加的信息 4) 比特 D: MLPP 用户表示语 0: 无指示 1: MLPP 用户		
	8	7	6	5	4	3	2	1													
1	H	G	F	E	D	C	B	A													
任选前向呼叫表示语	0x08	<table border="1"> <tr><td></td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr> <tr><td>1</td><td>H</td><td>G</td><td>F</td><td>E</td><td>D</td><td>C</td><td>B</td><td>A</td></tr> </table>		8	7	6	5	4	3	2	1	1	H	G	F	E	D	C	B	A	1) 比特 BA: 闭合用户群呼叫表示语 00: 非 CUG 呼叫 10: 闭合用户群呼叫, 允许出接口 11: 闭合用户群呼叫, 不允许出接口 2) 比特 C: 简单分段表示语 0: 将不发送附加的信息 1: 将在分段消息中发送附加的信息 3) 比特 H: 被连接线识别请求表示语 0: 未请求 1: 请求
	8	7	6	5	4	3	2	1													
1	H	G	F	E	D	C	B	A													

转接网选择	0x23		8	7	6	5	4	3	2	1					
		1	奇/偶	网络识别的类型			网络识别计划								
		2	网络识别												
		...													
		n													
			1) 奇/偶表示语 0: 偶数 1: 奇数 2) 网络识别的类型 000: CCITT-标准化的识别 010: 国内网识别 3) 网络识别计划 (对于 CCITT 标准化的识别) 0000: 未知 0011: 公用数据网识别码 0110: 公用陆地移动网识别码												
传输媒介请求	0x00	00000000: 语音 00000001: 备用 00000010: 64kb/s 不受限 00000011: 3.1KHZ 音频 00000100: 语音 (业务 2) /64kb/s 不受限 (业务 1) 交替 00000101: 64kb/s 不受限 (业务 1) /语音 (业务 2) 交替 00000110: 64kb/s 优选 00000111: 2*64kb/s 不受限 00001000: 384 kb/s 不受限 00001001: 备用 00001010: 1920 kb/s 不受限													
接入转送	0x03														
呼叫历史信息	0x2d														
电路群监视消息类型表示语	0x15														
电路状态表示语	0x26														
闭合用户群连锁编码	0x1a														
被连接的号码	0x21														
导通表示语	0x10														
任选参数结束	0x00														
事件信息	0x24														
性能表示语	0x18														
通用数字	0xc1														
通用号码	0xc0														
信息表示语	0x0f														
信息请求表示语	0x0e														
MCID 请求表示语	0x3b														
MCID 响应表示语	0x3c														

消息兼容性信息	0x38
网络专用性能	0x2f
原被叫号码	0x28
始发 ISC 点编码	0x2b
参数兼容性信息	0x39
传播时延计数器	0x31
范围和状态	0x16
改发的号码	0x0b
改发信息	0x13
改发号码	0x0c
改发号码限制	0x40
后续号码	0x05
暂停/恢复表示语	0x22
信令点编码	0x1e
所用的传输媒介	0x35
用户业务信息	0x1d
用户终端业务信息	0x34
用户-用户表示语	0x2a
用户-用户信息	0x20
计费信息	0xfe

注：更多信息请查阅 ITU-T Q.763

**ISUP 几种典型接续的信号过程：**

- 1、呼叫遇被叫用户空闲的接续：

2、呼叫遇被叫用户忙等的接续

