

Contents

Statement	4
SDK Overview	4
Function interface description	6
1 Initialize and unregister SDK	6
1.1 Initialize SDK: IPCSdkInit()	6
1.2 Unregister : IPCSdkUnInit ()	6
1.3 Get SDK version: IPCSDKVersion ()	6
2 Login and logout of the device	7
2.1 Login device: IPCSdkLogin (IPCSdkConnetParam *param, PSTR_USER_LOGIN pUserLogin, int nsec)	7
2.2 Log out of the device: IPCSdkLogout (int iparam)	9
3 Video Preview	9
3.1 Request video: IPCStartVideoEx(int iparam, PSTR_REAL_VIDEO_QUERY_EX ptVideoEx)	9
3.2 Disconnect video: IPCStopVideoEx (int iparam, PSTR_REAL_VIDEO_STOP_EX pArray)	11
4 Parameter configuration	11
4.1 Get system parameter information: IPCGetSysVersion(int iparam, PSTR_SYS_RELEASE_QUERY ptSysInfo)	11
4.2 Get encoding information: IPCVideoParamGet(int iparam, PSTR_VCHANN_VIDEO_PARAM_QUERY ptVideoParam, int nMaxCount)	13
4.3 Set encoding information: IPCVideoParamSet(int iparam, PSTR_VCHANN_VIDEO_PARAM_QUERY ptVideoParam, int nCount)	14
4.4 Get video subtitle information: IPCExOsdGet(int iparam, PSTR_EX_OSD_QUERY ptExOsdParam, int nMaxCount)	15
4.5 Set video subtitle information: IPCExOsdSet(int iparam, PSTR_EX_OSD_QUERY ptExOsdParam, int nCount)	17
4.6 Get GB 28181 information: IPCGB28181_setupGet(int iparam, PSTR_GB28181_CON_QUERY ptGB28181)	17
4.7 Set GB 28181 information: IPCGB28181_setupSet(int iparam, PSTR_GB28181_CON_QUERY ptGB28181)	19
4.8 Get FTP image storage information: IPCFtpGet(int iparam, PSTR_FTP_INFO_QUERY ptFtpInfo)	20
4.9 Set FTP image storage information: IPCFtpSet(int iparam, PSTR_FTP_INFO_QUERY ptFtpInfo)	21
4.10 Get the national standard 1400 protocol information: IPCGB1400Get(int iparam, PSTR_WEB_SERVICE_INFO_QUERY ptWebServiceInfo)	22
4.11 Set the national standard 1400 protocol information: IPCGB1400Set(int iparam, PSTR_WEB_SERVICE_INFO_QUERY ptWebServiceInfo)	23
4.12 Get system time: IPCSystemTimeGet(int iparam, PSTR_SYS_TIME_QUERY	

ptSysTimeInfo)	24
4.13 Set system time: IPCSystemTimeSet(int iparam, PSTR_SYS_TIME_QUERY ptSysTimeInfo)	25
4.14 Get NTP network time: IPCNtpConfGet(int iparam, PSTR_NTP_SERVER_QUERY ptNtpConfInfo)	25
4.15 Set NTP network time synchronization: IPCNtpConfSet(int iparam, PSTR_NTP_SERVER_QUERY ptNtpConfInfo)	26
4.16 Set the tracking type: IPCTargetTrackSet(int iparam, PSTR_CAMERA_TARGET_TRACK ptTargetTrack)	27
4.17 Ball camera positioning linkage: IPCMagnifyApply(int iparam, PSTR_CAMERA_PTZ_LOCATE ptPTZLocate)	28
4.18 Get the startup action configuration: IPCGetStartOperationInfo(int iparam, LPS_START_OPERATION_INFO ptStartOperationInfo)	29
4.19 Set the startup action configuration: IPCSetStartOperationInfo(int iparam, LPS_START_OPERATION_INFO ptStartOperationInfo)	30
4.20 Get device IP: IPCDevNetWorkGet(int iparam, PSTR_DEV_NET_INFO ptNetConfInfo)	31
4.21 Set device IP: IPCDevNetWorkSet(int iparam, PSTR_DEV_NET_INFO ptNetConfInfo)	32
4.22 Cross cursor: IPCCrossSet(int iparam, PSTR_CROSS cross)	32
4.23 Get visible light image enhancement parameters: IPCImagelspImageEnhanceGet(int iparam, PSTR_IMAGE_ENH_PARAM_INFO_QUERY image_isp)	33
4.24 Set visible light image enhancement parameters: IPCImagelspImageEnhanceSet(int iparam, PSTR_IMAGE_ENH_PARAM_INFO_QUERY image_isp)	34
4.25 Get the visible light image day and night switching parameters: IPCImagelspDayNightGet(int iparam, PSTR_DAY_NIGHT_TRANSF_INFO_QUERY image_isp)	35
4.26 Set the day and night switching parameters of the visible light image: IPCImagelspDayNightSet(int iparam, PSTR_DAY_NIGHT_TRANSF_INFO_QUERY image_isp)	36
5 PTZ control	37
5.1 Get the dome camera control parameters: IPCPTZParamGet(int iparam, PSTR_YTCT_CON_ADDR ptCameraParam, int nMaxCount)	37
5.2 Set the dome camera control parameters: IPCPTZParamSet(int iparam, PSTR_YTCT_CON_ADDR ptCameraParam, int nCount)	38
5.3 Control the movement of the ball camera: IPCPTZCtrl(int iparam, PSTR_YTCT_CON_CTRL ptCameraCtrl)	39
5.4 Get preset point: IPCGetPresetPoint(int iparam, PSTR_YTCT_CON_NAME_STR ptPresetName)	41
5.5 Clear preset point: IPCClearPresetPoint(int iparam, PSTR_YTCT_CON_NAME_STR pptPresetName)	42
5.6 Set preset point: IPCSetPresetPoint(int iparam, PSTR_YTCT_CON_NAME	

ptPresetName)	42
5.7 Get device PTZ timer reset information: IPCGetRestartInfo2(int iparam, PSTR_RESTART_TIME_QUERY2 ptRestartInfo)	43
5.8 Set the device PTZ timer reset information: IPCSetRestartInfo2 (int iparam, PSTR_RESTART_TIME_QUERY2 ptRestartInfo)	44
5.9 PTZ reset immediately : IPCPtzReboot (int iparam)	45
5.10 Get PTZ value: IPCPTZValueGet(int iparam, PSTR_YTCT_PTZ_VAL ptPTZValueInfo)	45
5.11 Set PTZ value: IPCPTZValueSet(int iparam, PSTR_YTCT_PTZ_VAL ptPTZValueInfo)	46
5.12 Get cruise information: IPCCruiseinfoGet(int iparam, LPS_PTZ_CRUISE_GROUP_INFO ptzCruiseinfo)	47
5.13 Set cruise information: IPCCruiseInfoSet(int iparam, LPS_PTZ_CRUISE_GROUP_INFO ptz_cruiseinfo)	48
5.14 Set the cruise status: IPCCruiseSet(int iparam, PSTR_START_CRUISE ptCruise)	49
5.15 PTZ pass-through : IPCDevPassThrough(int iparam, PSTR_PASS_THROUGH ptPassThrough)	50
5.16 Wiper: IPCWiperSet(int iparam, PSTR_WIPER wiper)	50
5.17 Laser: IPLaserSet(int iparam, PSTR_LASER laser)	51
5.18 Synchronous zoom: IPCZoomSysncSet(int iparam, PSTR_ZOOMSYSNC zoom)	52
6 Remote equipment maintenance	53
6.1 Get device scheduled restart information: IPCGetRestartInfo(int iparam, PSTR_RESTART_TIME_QUERY ptRestartInfo)	53
6.2 Set device scheduled restart information: IPCSetRestartInfo(int iparam, PSTR_RESTART_TIME_QUERY ptRestartInfo)	54
6.3 Reboot the device immediately : IPCReboot (int iparam)	55
7 Reporting Information	55
7.1 Manual capture: IPCSIMUSNAP(int iparam, PSTR_SIMUL_CAPTURE ptInfo, int blockLabel, unsigned char * pBuf , unsigned int ibufLen, int stayTime)	55
7.2 GPS information: IPCGpsInfo(int iparam, PSTR_GPS_INFO_QUERY ptInfo, int blockMark, unsigned char *pBuf, unsigned int ibufLen, int stayTime)	58
7.3 Distance measurement: IPCDistantInfoSet(int iparam, PSTR_DISTANT_PARAM ptInfo, int blockMark, unsigned char *pBuf, unsigned int ibufLen, int stayTime)	59
7.4 Report data callback function: SetReportCallbackFunc(DataReportCallbackFunc callBack)	61
7.5 Video callback function: SetVideoFrameCallBack(VideoFrameCallBack callBack)	63
7.6 Start receiving video frames: StartVideoRequest(int iparam)	64
7.7 Stop receiving video frames: StopVideoRequest(int iparam)	65
7.8 Get control authority: IPCFireModeGet(int iparam, PSTR_FIRE_TEC_MODE ModeSet)	65
7.9 Set control rights: IPCFireModeSet(int iparam, PSTR_FIRE_TEC_MODE ModeSet)	66
7.10 Fireworks alarm parameter information:	67
7.11 Exception callback function:SetExceptionCallBack(DisconnectionCallBack callBack)	69
8.1 Force exit: IPC_FORCE_EXIT(int iparam)	70
9.1 Error code definition	70

Statement

Thank you very much for purchasing our products. If you have any questions or needs, please feel free to contact us.

We have tried our best to ensure the completeness and accuracy of the contents of the manual, but technical inaccuracies, inconsistencies with product functions and operations, or printing errors are inevitable. If you have any questions or disputes, please refer to our final interpretation.

Products and manuals will be updated in real time without prior notice.

The contents in this manual are only for reference and guidance. Please refer to the actual contents of the SDK .

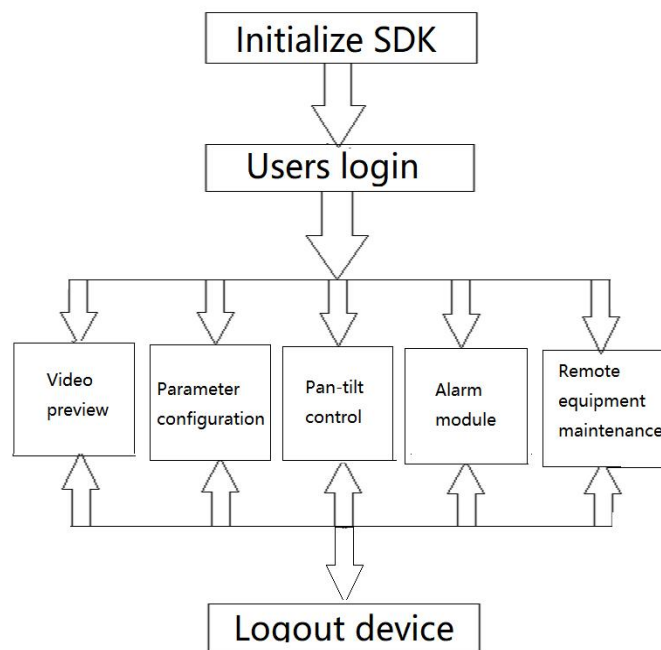
SDK Overview

The device network SDK is developed based on the device's private network communication protocol. It is a supporting module for embedded network products and is used for secondary development of remote access and control device software.

Device Network SDK Main functions:

It is mainly used for real-time stream preview, PTZ control, alarm, equipment maintenance management, parameter configuration (system configuration, channel configuration, alarm configuration, user configuration) and obtaining equipment capability sets.

The main process of SDK basic calling



Initialize SDK : Initialize the entire network SDK system, pre-allocate memory, and perform other operations.

Preview module: obtains real-time code stream from the device, decodes , displays and plays it. For details, see the preview module process.

PTZ control module: realizes the basic operation, preset points, cruise and trajectory control of the PTZ.

General parameter configuration module: obtains and sets device parameters, mainly including device parameters, network parameters, user parameters and other parameter information.

Remote device maintenance module: implements maintenance tasks such as restarting the device, restoring default values, remote upgrading, and importing / exporting configuration files.

Function interface description

1 Initialize and unregister SDK

1.1 Initialize SDK: IPCSdkInit()

Function definition	IPCSdkInit()
Function	SDK Initialization
Return value	A value of 0 indicates success; a value of -1 indicates failure
Remark	

1.2 Unregister : IPCSdkUnInit ()

Function definition	IPCSdkUnInit()
Function	Logout
Return value	A value of 0 indicates success; a value of -1 indicates failure
Remark	

1.3 Get SDK version: IPCSDKVersion ()

Function	IPCSDKVersion ()
----------	------------------

definition	
Function	Get SDK version
Return value	A value of 0 indicates success; a value of -1 indicates failure
Remark	

2 Login and logout of the device

2.1 Login device: IPCSdkLogin (IPCSdkConnetParam *param, PSTR_USER_LOGIN pUserLogin, int nsec)

Function definition	IPCSdkLogin(IPCSdkConnetParam *param, PSTR_USER_LOGIN pUserLogin, int nsec)	
Function	Login device	
Parameter	Variable	Illustrate
	param	Connection device parameters
	nsec	Connection timeout wait time (seconds)
	pUserLogin	User login structure parameters
Return value	If the value is greater than 0, success; if the value is less than or equal to 0, failure	

Remark	
--------	--

```
typedef enum IPCSDKNetMode
{
    SDK_NET_UDP = 0x01,
    SDK_NET_TCP = 0x02
}IPCSdk_Net_Mode;
```

Member Name	Describe
SDK_NET_UDP	Value 1 UDP connection
SDK_NET_TCP	Value is 2 TCP connection

```
typedef struct sdkConnetParam
{
    IPCSdk_Net_Mode netMode;

    unsigned int uiPort;

    union
    {
        char argLocatName[64];
        char serverIp[64];
    };
}IPCSdkConnetParam;
```

Member Name	Describe
netMode	Network connection mode
uiPort	The port number
argLocatName	Local Linux connection
serverIp	IP address

```
typedef struct tagStrUserLogin
{
    char cUserName[32];
    char cUserPasswd[32];
    unsigned int uiDecConfVer;
    int iUserManage;
    char cResv[88];
} _PACKED_ STR_USER_LOGIN , *PSTR_USER_LOGIN;
```

Member Name	Describe
cUserName	username

cUserPasswd	password
uiDecConfVer	Client version
iUserManage	User Management
cResv[88]	Reserved Bytes

2.2 Log out of the device: IPCSdkLogout (int iparam)

Function definition	IPCSdkLogout(int iparam)	
Function	Exit device	
Parameter	Variable	Illustrate
	iparam	User ID
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

3 Video Preview

3.1 Request video: IPCStartVideoEx(int iparam, PSTR_REAL_VIDEO_QUERY_EX ptVideoEx)

Function definition	IPCStartVideoEx(int iparam, PSTR_REAL_VIDEO_QUERY_EX ptVideoEx)	
Function	Request Video	
Parameter	Variable	Illustrate
	iparam	User ID

	ptVideoConn	Request video structure parameters
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

typedef struct tagStrRealVQuery

```
{
    unsigned int    UserID;
    char            bConnectOK;
    char            bReason;
    char            VideoChannel;
    char            VideoConnMode;
    unsigned int    ReceiveIPAddress;
    unsigned short  ReceivePort;
    char            bActivate;
    char            bStream;
    unsigned int    nSessionIdLen;
    unsigned int    nPackLen;
    unsigned short  SendBindPort;
    unsigned char   ucResv[2];
    char            *ucPackBuf;
} _PACKED_ STR_REAL_VIDEO_QUERY, *PSTR_REAL_VIDEO_QUERY;
```

Member Name	Describe
UserID	User ID
bConnectOK	Whether the connection is successful,the success monitoring value is 1,and the monitoring failure value is 0
bReason	Connection failed, indicate the reason
VideoChannel	Video channel number
VideoConnMode	Video connection mode, UDP connection value is 1, TCP connection value is 3
ReceiveIPAddress	Receiving address
ReceivePort	Receiving Port
bActivate	Activate Package Options
bStream	Video or audio/video composite stream format
nSessionIdLen	Session ID length

nPackLen	Activation packet length, enabled when bActivate is 2
SendBindPort	Send Port
ucResv[2]	Reserved Bytes
ucPackBuf	Activation package data

3.2 Disconnect video: IPCStopVideoEx (int iparam, PSTR_REAL_VIDEO_STOP_EX pArray)

Function definition	IPCStopVideoEx (int iparam, PSTR_REAL_VIDEO_STOP_EX pArray)	
Function	Disconnect Video	
Parameter	Variable	Illustrate
	iparam	User ID
	pArray	Disconnect video structure parameters
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

4 Parameter configuration

4.1 Get system parameter information: IPCGetSysVersion(int iparam, PSTR_SYS_RELEASE_QUERY ptSysInfo)

Function definition	IPCGetSysVersion(int iparam, PSTR_SYS_RELEASE_QUERY ptSysInfo)
---------------------	---

Function	Get system parameter information	
Parameter	Variable	Illustrate
	iparam	User ID
	ptSysInfo	Get system parameter structure parameters
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

```
typedef struct tagStrSysReleaseQuery
{
    unsigned int uiProcType;
    char cSerialNum[16];
    unsigned int uiHardwareVer;
    unsigned int uiSoftwareVer;
    unsigned int uiProtocolVer;
    char          szcMac[18];
    unsigned short usActWidth;
    unsigned short usActHeight;
    short          usPTZModel;
    short          usPTZVer;
    char cProBuildTime[18];
    unsigned int   uiDpuVer;
    char cDevName[16];
    char cRev[72];
}_PACKED_STR_SYS_RELEASE_QUERY, *PSTR_SYS_RELEASE_QUERY;
```

Member Name	Describe
uiProcType	Product number
cSerialNum[16]	Serial number
uiHardwareVer	Hardware version number
uiSoftwareVer	Software version number
uiProtocolVer	Protocol version number
szcMac[18]	MAC Address
usActWidth	Actual resolution width
usActHeight	Actual resolution height

usPTZModel	Ball machine model
usPTZVer	Program version number
cProBuildTime[18]	Dome program creation time
uiDpuVer	Dpu module version
cDevName[16]	Device Name
cRev[72]	Reserved Bytes

4.2 Get encoding information: IPCVideoParamGet(int iparam, PSTR_VCHANN_VIDEO_PARAM_QUERY ptVideoParam, int nMaxCount)

Function definition	IPCVideoParamGet(int iparam, PSTR_VCHANN_VIDEO_PARAM_QUERY ptVideoParam, int nMaxCount)	
Function	Get encoding information	
Parameter	Variable	Illustrate
	iparam	User ID
	ptVideoParam	Get the encoding information structure parameters
	nMaxCount	Maximum number
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

```
typedef struct tagStrVChannelVParamQuery
{
    unsigned short    usChannel;
    char              cVQuality;
    char              cFrameRate;
```

```

        unsigned int    uiBitRate;
        char            cKeyFInterval;
        char            cImageResol;
        char            cEncodeType;
        char            cVBR;
        char            cRev[120];
    }_PACKED_ STR_VCHANN_VIDEO_PARAM_QUERY, *PSTR_VCHANN_VIDEO_PARAM_QUERY;

```

Member Name	Describe
usChannel	Video channel number
cVQuality	Video Quality
cFrameRate	Frame rate
uiBitRate	Bit rate
cKeyFInterval	Keyframe Interval
cImageResol	Image Resolution
cEncodeType	Encoding Type
cVBR	Whether to change the bit rate
cRev[120]	Reserved Bytes

4.3 Set encoding information: IPCVideoParamSet(int iparam, PSTR_VCHANN_VIDEO_PARAM_QUERY ptVideoParam, int nCount)

Function definition	IPCVideoParamSet(int iparam, PSTR_VCHANN_VIDEO_PARAM_QUERY ptVideoParam, int nCount)	
Function	Set encoding information	
Parameter	Variable	Illustrate
	iparam	User ID
	ptVideoParam	Set the encoding information structure parameters
	nCount	Maximum number

Return value	A value of 0 indicates success; a value of -1 indicates failure
Remark	

4.4 Get video subtitle information: IPCEXosdGet(int iparam, PSTR_EX_OSD_QUERY ptExOsdParam, int nMaxCount)

Function definition	IPCEXosdGet(int iparam, PSTR_EX_OSD_QUERY ptExOsdParam, int nMaxCount)	
Function	Get video subtitle information	
Parameter	Variable	Illustrate
	iparam	User ID
	ptExOsdParam	Get video subtitle structure parameters
	nMaxCount	Maximum number
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

```
typedef struct tagStrOsdContent
{
    char cShowOsd;
    char cShowTime;
    char cShowCompass;
    char cShowZ;
    char cShowPT;
    char cShowYaw;
    char cShowAlgoType;
```

```

char ucOsdSize;
char cExOsdEn[4];
char szContent[64];
unsigned short usChannel;
unsigned short usXPos;
unsigned short usYPos;
unsigned short usTimeXPos;
unsigned short usTimeYPos;
STR_OSD_TEXT_INFO exOsdInfo[4];
unsigned int ucShowLocalGPS;
}_PACKED_STR_EX_OSD_QUERY, *PSTR_EX_OSD_QUERY;

```

Member Name	Describe
cShowOsd	Title display enable
cShowTime	Time display enable
cShowCompass	Compass display enable
cShowZ	Zoom display enable
cShowPT	PT display enable
cShowYaw	Roll angle display enable
cShowAlgoType	Working status display enable
ucOsdSize	Subtitle size
cExOsdEn[4]	Custom subtitle display
szContent[64]	title content
usChannel	Video channel number
usXPos	Title X coordinate
usYPos	Title Y coordinate
usTimeXPos	Time X coordinate
usTimeYPos	Time Y coordinate
exOsdInfo[4]	Custom subtitle information
ucShowLocalGPS	Local GPS display enabled

```

typedef struct tagStrOsdText
{
    unsigned short usXPos;
    unsigned short usYPos;
    char szContent[64];
}_PACKED_STR_OSD_TEXT_INFO, *PSTR_OSD_TEXT_INFO;

```

Member Name	Describe
usXPos	The x value of the subtitle's coordinates on the screen
usYPos	The y value of the subtitle's coordinates on the screen

szContent[64]	Subtitle content
---------------	------------------

4.5 Set video subtitle information: IPCEX0sdSet(int iparam, PSTR_EX_OSD_QUERY ptExOsdParam, int nCount)

Function definition	IPCEX0sdSet(int iparam, PSTR_EX_OSD_QUERY ptExOsdParam, int nCount)	
Function	Set video subtitle information	
Parameter	Variable	Illustrate
	iparam	User ID
	ptExOsdParam	Set the video subtitle structure parameters
	nCount	Maximum number
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

4.6 Get GB 28181 information: IPCGB28181_setupGet(int iparam, PSTR_GB28181_CON_QUERY ptGB28181)

Function definition	IPCGB28181_setupGet(int iparam, PSTR_GB28181_CON_QUERY ptGB28181)	
Function	Get information about GB 28181	
Parameter	Variable	Illustrate

	iparam	User ID
	ptGB28181	Get the national standard 28181 information structure parameters
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

```
typedef struct tagSTR_GB28181_CON_QUERY
{
    char          szServerId[32];
    char          szServerDomain[32];
    char          szServerIp[32];
    unsigned int  uiServerPort;
    unsigned int  uiRegisterExpiry;
    unsigned int  uiKeepLiveTime;
    unsigned int  uiConMode;
    char          szClientId[32];
    char          szClientUserName[32];
    char          szClientPasswd[32];
    char          cEnFlag;
    char          cSyncTime;
    char          cRsv[2];
    char          szChnID1[32];
    char          szChnID2[32];
    char          ChanelName1[64];
    char          ChanelName2[64];
}__PACKED__ STR_GB28181_CON_QUERY, *PSTR_GB28181_CON_QUERY;
```

Member Name	Describe
szServerId[32]	Server id
szServerDomain[32]	Server Domain
szServerIp[32]	Server ip
uiServerPort	Server port
uiRegisterExpiry	Registration validity period
uiKeepLiveTime	Heartbeat cycle
uiConMode	Connection method ; 0 - udp ; 1 - tcp

szClientId[32]	Device ID
szClientUserName[32]	Device User Name
szClientPasswd[32]	Device password
cEnFlag	Platform Enablement
cSyncTime	Platform synchronization
cRsv[2]	Reserved Bytes
szChnID1[32]	Video channel code ID1
szChnID2[32]	Video channel code ID2
ChanelName1[64]	Channel Number Name 1
ChanelName2[64]	Channel Number Name 2

4.7 Set GB 28181 information: IPCGB28181_setupSet(int iparam, PSTR_GB28181_CON_QUERY ptGB28181)

Function definition	IPCGB28181_setupSet(int iparam, PSTR_GB28181_CON_QUERY ptGB28181)	
Function	Set national standard 28181 information	
Parameter	Variable	Illustrate
	iparam	User ID
	ptGB28181	Set the national standard 28181 information structure parameters
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

4.8 Get FTP image storage information: IPCFtpGet(int iparam, PSTR_FTP_INFO_QUERY ptFtpInfo)

Function definition	IPCFtpGet(int iparam, PSTR_FTP_INFO_QUERY ptFtpInfo)	
Function	Get FTP image storage information	
Parameter	Variable	Illustrate
	iparam	User ID
	ptFtpInfo	Get FTP image storage information structure parameters
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

```
typedef struct tagSTR_FTP_INFO_QUERY
{
    unsigned int uiEnable;
    unsigned int uiFtpIP;
    unsigned short usFtpPort;
    char    cResv1[18];
    char    szUserName[32];
    char    szUserPasswd[32];
    char    szRoadName[64];
    char    szDirection[32];
    char    cResv2[32];
} _PACKED_ STR_FTP_INFO_QUERY, *PSTR_FTP_INFO_QUERY;
```

Member Name	Describe
uiEnable	FTP Enable
uiFtpIP	FTP server address

usFtpPort	FTP server port
cResv1[18]	Reserved Bytes
szUserName[32]	FTP server username
szUserPasswd[32]	FTP server password
szRoadName[64]	Scene Name
szDirection[32]	Direction Name
cResv2[32]	Reserved Bytes

4.9 Set FTP image storage information: IPCFtpSet(int iparam, PSTR_FTP_INFO_QUERY ptFtpInfo)

Function definition	IPCFtpSet(int iparam, PSTR_FTP_INFO_QUERY ptFtpInfo)	
Function	Set FTP image storage information	
Parameter	Variable	Illustrate
	iparam	User ID
	ptFtpInfo	Set the FTP image storage information structure parameters
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

4.10 Get the national standard 1400 protocol information:

**IPCGB1400Get(int iparam, PSTR_WEB_SERVICE_INFO_QUERY
ptWebServiceInfo)**

Function definition	IPCGB1400Get(int iparam, PSTR_WEB_SERVICE_INFO_QUERY ptWebServiceInfo)	
Function	Get GB 1400 protocol information	
Parameter	Variable	Illustrate
	iparam	User ID
	ptWebServiceInfo	Get the national standard 1400 protocol information structure parameters
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

```
typedef struct tagSTR_WEB_SERVICE_INFO_QUERY
{
    unsigned int  uiEnable;
    unsigned int isBase64;
    unsigned int uiProtocolType;
    char          szServerId[32];
    char          szUserName[32];
    char          szPassWord[32];
    unsigned int  uiServerIpAddr;
    unsigned short usServerPort;
    char          cResv[122];
} _PACKED_ STR_WEB_SERVICE_INFO_QUERY, *PSTR_WEB_SERVICE_INFO_QUERY;
```

Member Name	Describe
uiEnable	Whether to enable
isBase64	Encryption
uiProtocolType	Agreement type
szServerId[32]	Server address
szUserName[32]	Username
szPassWord[32]	User password
uiServerIpAddr	Server IP
usServerPort	Server Port
cResv[122]	Reserved Bytes

4.11 Set the national standard 1400 protocol information: IPCGB1400Set(int iparam, PSTR_WEB_SERVICE_INFO_QUERY ptWebServiceInfo)

Function definition	IPCGB1400Set(int iparam, PSTR_WEB_SERVICE_INFO_QUERY ptWebServiceInfo)	
Function	Set the national standard 1400 protocol information	
Parameter	Variable	Illustrate
	iparam	User ID
	ptWebServiceInfo	Set the national standard 1400 protocol information structure parameters
Return value	A value of 0 indicates success; a value of -1 indicates failure	

Remark	
--------	--

4.12 Get system time: IPCSystemTimeGet(int iparam, PSTR_SYS_TIME_QUERY ptSysTimeInfo)

Function definition	IPCSystemTimeGet(int iparam, PSTR_SYS_TIME_QUERY ptSysTimeInfo)	
Function	Get system time	
Parameter	Variable	Illustrate
	iparam	User ID
	ptSysTimeInfo	Get system time structure parameters
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

```
typedef struct tagStrSysTimeQuery
{
    unsigned int uiSysTime;
    char cRev[128];
}_PACKED_STR_SYS_TIME_QUERY, *PSTR_SYS_TIME_QUERY;
```

Member Name	Describe
uiSysTime	system time
cRev[128]	Reserved Bytes

4.13 Set system time: IPCSystemTimeSet(int iparam, PSTR_SYS_TIME_QUERY ptSysTimeInfo)

Function definition	IPCSystemTimeSet(int iparam, PSTR_SYS_TIME_QUERY ptSysTimeInfo)	
Function	Setting the system time	
Parameter	Variable	Illustrate
	iparam	User ID
	ptSysTimeInfo	Set system time structure parameters
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

4.14 Get NTP network time: IPCNtpConfGet(int iparam, PSTR_NTP_SERVER_QUERY ptNtpConfInfo)

Function definition	IPCNtpConfGet(int iparam, PSTR_NTP_SERVER_QUERY ptNtpConfInfo)	
Function	Obtain NTP network time synchronization	
Parameter	Variable	Illustrate
	iparam	User ID
	ptNtpConfInfo	Get NTP network

		synchronization structure parameters
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

```
typedef struct tagStrNtpServerQuery
{
    unsigned int uiEnable;
    unsigned int uiServerIp;
    unsigned short usJudgeCycle;
    unsigned char ucJudegPrecision;
    char cRev[125];
} _PACKED_ STR_NTP_SERVER_QUERY, *PSTR_NTP_SERVER_QUERY;
```

Member Name	Describe
uiEnable	Whether to enable
uiServerIp	Server IP
usJudgeCycle	Time synchronization period (unit: minutes)
ucJudegPrecision	Time synchronization accuracy (unit: seconds)
cRev[125]	Reserved Bytes

4.15 Set NTP network time synchronization: IPCNtpConfSet(int iparam, PSTR_NTP_SERVER_QUERY ptNtpConfInfo)

Function definition	IPCNtpConfSet(int iparam, PSTR_NTP_SERVER_QUERY ptNtpConfInfo)	
Function	Set up NTP network time synchronization	
Parameter	Variable	Illustrate
	iparam	User ID

	ptNtpConfInfo	Set the NTP network synchronization structure parameters
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

4.16 Set the tracking type: IPCTargetTrackSet(int iparam, PSTR_CAMERA_TARGET_TRACK ptTargetTrack)

Function definition	IPCTargetTrackSet(int iparam, PSTR_CAMERA_TARGET_TRACK ptTargetTrack)	
Function	Set the trace type	
Parameter	Variable	Illustrate
	iparam	User ID
	ptTargetTrack	Set the trace type structure parameters
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

```
typedef struct cameraTargetTrack
{
    unsigned char ucTrackType;
}_PACKED_STR_CAMERA_TARGET_TRACK, *PSTR_CAMERA_TARGET_TRACK;
```

Member Name	Describe
ucTrackType	0 means stop tracking , 1 means start tracking

4.17 Ball camera positioning linkage: IPCMagnifyApply(int iparam, PSTR_CAMERA_PTZ_LOCATE ptPTZLocate)

Function definition	IPCMagnifyApply(int iparam, PSTR_CAMERA_PTZ_LOCATE ptPTZLocate)	
Function	Ball camera positioning linkage	
Parameter	Variable	Illustrate
	iparam	User ID
	ptPTZLocate	Ball camera positioning linkage structure parameters
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

```
typedef struct cameraPTZLocate
{
    strAreaPoint_t strAreaPoint[2];
    unsigned char uiChanID;
    unsigned char szResv[123];
}_PACKED_STR_CAMERA_PTZ_LOCATE, *PSTR_CAMERA_PTZ_LOCATE;
```

Member Name	Describe
strAreaPoint[2]	Save the ball camera positioning linkage point
uiChanID	The value of video channel 1 is 1; the value of video channel 2 is 2
szResv[123]	Reserved Bytes

```
typedef struct strAreaPoint_struct
{
```

```

        unsigned short usPointX;
        unsigned short usPointY;
    }strAreaPoint_t;

```

Member Name	Describe
usPointX	The x-coordinate of the point
usPointY	The y-coordinate of the point

4.18 Get the startup action configuration:

**IPCGetStartOperationInfo(int iparam,
LPS_START_OPERATION_INFO ptStartOperationInfo)**

Function definition	IPCGetStartOperationInfo(int iparam, LPS_START_OPERATION_INFO ptStartOperationInfo)	
Function	Get the boot action configuration	
Parameter	Variable	Illustrate
	iparam	User ID
	ptStartOperationInfo	Get the boot action configuration structure parameters
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

```

typedef struct tagStartOperationInfo
{
    unsigned char ucEnable;
    unsigned char ucStartOperationNo;
}

```

```

    unsigned char ucIndex;
    unsigned char ucPresetNo;
    unsigned char ucStartTime;
    unsigned char resv[123];
}_PACKED_ S_START_OPERATION_INFO, *LPS_START_OPERATION_INFO;

```

Member Name	Describe
ucEnable	Whether to enable
ucStartOperationNo	Power-on action number, 0: guard position , 1: cruise, 2: tracking
ucIndex	Preset point or cruise group index
ucPresetNo	Preset point number
ucStartTime	Starting time
resv[123]	Reserved Bytes

4.19 Set the startup action configuration:

**IPCSetStartOperationInfo(int iparam,
LPS_START_OPERATION_INFO ptStartOperationInfo)**

Function definition	IPCSetStartOperationInfo(int iparam, LPS_START_OPERATION_INFO ptStartOperationInfo)	
Function	Set the boot action configuration	
Parameter	Variable	Illustrate
	iparam	User ID
	ptStartOperationInfo	Set the startup action configuration structure parameters
Return value	A value of 0 indicates success; a value of -1 indicates failure	

Remark	
--------	--

4.20 Get device IP: IPCDevNetWorkGet(int iparam, PSTR_DEV_NET_INFO ptNetConfInfo)

Function definition	IPCDevNetWorkGet(int iparam, PSTR_DEV_NET_INFO ptNetConfInfo)	
Function	Get device IP information	
Parameter	Variable	Illustrate
	iparam	User ID
	ptNetConfInfo	Get device IP structure parameters
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

```
typedef struct tagSTR_DEV_NET_INFO
{
    unsigned int    uiIPAddr;
    unsigned int    uiNetMask;
    unsigned int    uiGateWay;
    char            cResv[128];
} _PACKED_ STR_DEV_NET_INFO, *PSTR_DEV_NET_INFO;
```

Member Name	Describe
uiIPAddr	IP address
uiNetMask	Subnet Mask
uiGateWay	Gateway
cResv[128]	Reserved Bytes

4.21 Set device IP: IPCDevNetWorkSet(int iparam, PSTR_DEV_NET_INFO ptNetConfInfo)

Function definition	IPCDevNetWorkSet(int iparam, PSTR_DEV_NET_INFO ptNetConfInfo);	
Function	Set the device IP	
Parameter	Variable	Illustrate
	iparam	User ID
	ptNetConfInfo	Set the device IP structure parameters
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

4.22 Cross cursor: IPCCrossSet(int iparam, PSTR_CROSS cross)

Function definition	IPCCrossSet(int iparam, PSTR_CROSS cross)	
Function	Crosshair settings	
Parameter	Variable	Illustrate
	iparam	User ID
	cross	Crosshair cursor structure parameters

Return value	A value of 0 indicates success; a value of -1 indicates failure
Remark	

```
typedef struct cross
{
    int cross_stat;
    char cRsv[127];
}_PACKED_STR_CROSS, *PSTR_CROSS;
```

Member Name	Describe
cross_stat	0-turn off the cross cursor; 1-turn on the cross cursor
cRsv[127]	Reserved Bytes

4.23 Get visible light image enhancement parameters: **IPCImageIspImageEnhanceGet(int iparam, PSTR_IMAGE_ENH_PARA_INFO_QUERY image_isp)**

Function definition	IPCImageIspImageEnhanceGet(int iparam, PSTR_IMAGE_ENH_PARA_INFO_QUERY image_isp)	
Function	Get visible light image enhancement parameters	
Parameter	Variable	Illustrate
	iparam	User ID
	image_isp	Get visible light image enhancement structure parameters
Return	A value of 0 indicates success; a value of -1	

value	indicates failure
Remark	

```
typedef struct imageEnhParaInfoQuery
```

```
{
```

```
    char cFilter;
```

```
    char cFilterLevel;
```

```
    char cAntiFogEn;
```

```
    char cAntiShakeEn;
```

```
    char cAntiFogLevel;
```

```
    char cRsv[127];
```

```
}_PACKED_STR_IMAGE_ENH_PARA_INFO_QUERY,*PSTR_IMAGE_ENH_PARA_INFO_QUERY;
```

Member Name	Describe
cFilter	Digital noise reduction;0-Off; 1-Normal noise reduction; 2-3D noise reduction
cFilterLevel	Noise reduction level ; 0-100
cAntiFogEn	Defogging mode;0-on;1-off;2-auto; 3-intelligent
cAntiShakeEn	0-turn on electronic image stabilization ; 1-turn off electronic image stabilization
cAntiFogLevel	Mist penetration level : 1-17
CRsv[127]	Reserved Bytes

4.24 Set visible light image enhancement parameters:

IPCImageIspImageEnhanceSet(int iparam, PSTR_IMAGE_ENH_PARA_INFO_QUERY image_isp)

Function definition	IPCImageIspImageEnhanceSet(int iparam, PSTR_IMAGE_ENH_PARA_INFO_QUERY image_isp)	
Function	Set visible light image enhancement parameters	
Parameter	Variable	Illustrate
	iparam	User ID

	image_isp	Set visible light image enhancement structure parameters
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

4.25 Get the visible light image day and night switching parameters: `IPCImageIspDayNightGet(int iparam, PSTR_DAY_NIGHT_TRANSF_INFO_QUERY image_isp)`

Function definition	<code>IPCImageIspDayNightGet(int iparam, PSTR_DAY_NIGHT_TRANSF_INFO_QUERY image_isp)</code>	
Function	Get the day and night switching parameters	
Parameter	Variable	Illustrate
	iparam	User ID
	image_isp	Get the day and night switching structure parameters
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

```
typedef struct dayNightTransfInfoQuery
{
    char cMode;
```

```

char cSensitivity;
char cRsv1[2];
unsigned int uiStartTime;
unsigned int uiEndTime;
char cRsv2[116];
}_PACKED_STR_DAY_NIGHT_TRANSF_INFO_QUERY,*PSTR_DAY_NIGHT_TRANSF_INFO_QUERY;

```

Member Name	Describe
cMode	Day and night switching mode;0-Day;1-Night; 2-Automatic;3-Timed switching; 4-Alarm input trigger; 5-Photoresistor
cSensitivity	Sensitivity in automatic mode ; 0-100
cRsv1[2]	Reserved Bytes
uiStartTime	Scheduled switching start time
uiEndTime	Scheduled switching end time
cRsv2[116]	Reserved Bytes

4.26 Set the day and night switching parameters of the visible light image: IPCImageIspDayNightSet(int iparam, PSTR_DAY_NIGHT_TRANSF_INFO_QUERY image_isp)

Function definition	IPCImageIspDayNightSet(int iparam, PSTR_DAY_NIGHT_TRANSF_INFO_QUERY image_isp)	
Function	Set day and night switching parameters	
Parameter	Variable	Illustrate
	iparam	User ID
	image_isp	Set the day and night switching structure parameters
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

5 PTZ control

5.1 Get the dome camera control parameters:

IPCPTZParamGet(int iparam, PSTR_YTCT_CON_ADDR ptCameraParam, int nMaxCount)

Function definition	IPCPTZParamGet(int iparam, PSTR_YTCT_CON_ADDR ptCameraParam, int nMaxCount)	
Function	Get the dome camera control parameters	
Parameter	Variable	Illustrate
	iparam	User ID
	ptCameraParam	Get the dome camera control structure parameters
	nMaxCount	Maximum number
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

```
typedef struct tagStrYtctConAddr
{
    unsigned char ucComPort;
    unsigned char ucDataBit;
    unsigned char ucStopBit;
    unsigned char ucParityBit;
    unsigned int uiBaudrate;
    unsigned char ucRs485Addr;
    unsigned char ucCameraType;
```

```

char          resv[118];
}_PACKED_STR_YTCT_CON_ADDR, *PSTR_YTCT_CON_ADDR;

```

Member Name	Describe
ucComPort	485 Channel Number
ucDataBit	Data bits
ucStopBit	Stop bits
ucParityBit	Check Digit
uiBaudrate	Baud rate
ucRs485Addr	485 Address
ucCameraType	agreement type
resv[118]	Reserved Bytes

5.2 Set the dome camera control parameters:

IPCPTZParamSet(int iparam, PSTR_YTCT_CON_ADDR ptCameraParam, int nCount)

Function definition	IPCPTZParamSet(int iparam, PSTR_YTCT_CON_ADDR ptCameraParam, int nCount)	
Function	Set the dome camera control parameters	
Parameter	Variable	Illustrate
	iparam	User ID
	ptCameraParam	Set the dome camera control structure parameters
	nCount	Maximum number
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

5.3 Control the movement of the ball camera: IPCPTZCtrl(int iparam, PSTR_YTCT_CON_CTRL ptCameraCtrl)

Function definition	IPCPTZCtrl(int iparam, PSTR_YTCT_CON_CTRL ptCameraCtrl)	
Function	Control the ball machine action	
Parameter	Variable	Illustrate
	iparam	User ID
	ptCameraCtrl	Control ball machine action structure parameters
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

```
typedef struct tagStrYtctConCtrl
{
    char          uChanId;
    char          uPlateId;
    char          uCommand;
    unsigned char uParam;
    char          uSpeed;
    char          resv[127];
} _PACKED_ STR_YTCT_CON_CTRL, *PSTR_YTCT_CON_CTRL;
```

Member Name	Describe
uChanId	Video channel number
uPlateId	PTZ 485 address number
uCommand	Command word
uParam	parameter
uSpeed	speed
resv[127]	Reserved Bytes

```

typedef enum
{
    e_PROTO_LENS_UP = 0x01,
    e_PROTO_LENS_DOWN,
    e_PROTO_LENS_LEFT,
    e_PROTO_LENS_RIGHT,
    e_PROTO_LENS_NEAR,
    e_PROTO_LENS_FAR,
    e_PROTO_LENS_STOP,
    e_PROTO_LENS_PRESET_GOTO,
    e_PROTO_LENS_PRESET_SET,
    e_PROTO_LENS_PRESET_DEL,
    e_PROTO_LENS_PRESET_CLEAR,
    e_PROTO_LENS_SETPAN_SPEED,
    e_PROTO_LENS_SETTILT_SPEED,
    e_PROTO_LENS_FOCUSNEAR,
    e_PROTO_LENS_FOCUSFAR,
    e_PROTO_LENS_DIAPHRAGM_LARGE,
    e_PROTO_LENS_DIAPHRAGM_SMALL,
    e_PROTO_LENS_LEFT_UP,
    e_PROTO_LENS_LEFT_DOWN,
    e_PROTO_LENS_RIGHT_UP,
    e_PROTO_LENS_RIGHT_DOWN,
}YTCT_COMMAND_CODE;

```

Member Name	Describe
e_PROTO_LENS_UP	Up
e_PROTO_LENS_DOWN	Down
e_PROTO_LENS_LEFT	Left
e_PROTO_LENS_RIGHT	Right
e_PROTO_LENS_NEAR	Near
e_PROTO_LENS_FAR	Far
e_PROTO_LENS_STOP	Stop
e_PROTO_LENS_PRESET_GOTO	Call preset
e_PROTO_LENS_PRESET_SET	Set preset
e_PROTO_LENS_PRESET_DEL	Delete preset
e_PROTO_LENS_PRESET_CLEAR	Clear preset
e_PROTO_LENS_SETPAN_SPEED	Temporarily unused
e_PROTO_LENS_FOCUSNEAR	Focus near
e_PROTO_LENS_FOCUSFAR	Focus far
e_PROTO_LENS_DIAPHRAGM_LARGE	Aperture increase
e_PROTO_LENS_DIAPHRAGM_SMALL	Aperture reduction
e_PROTO_LENS_LEFT_UP	Upleft
e_PROTO_LENS_LEFT_DOWN	Left down

e_PROTO_LENS_RIGHT_UP	Upper right
e_PROTO_LENS_RIGHT_DOWN	Right down

5.4 Get preset point: IPCGetPresetPoint(int iparam, PSTR_YTCT_CON_NAME_STR ptPresetName)

Function definition	IPCGetPresetPoint(int iparam, PSTR_YTCT_CON_NAME_STR ptPresetName)	
Function	Get preset points	
Parameter	Variable	Illustrate
	iparam	User ID
	ptPresetName	Get the preset point structure parameters
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

```
typedef struct tagStrYtctConNameSTR
{
    STR_YTCT_CON_NAME strPresetInfo[MAX_PRESET_POINT];
} _PACKED_ STR_YTCT_CON_NAME_STR, *PSTR_YTCT_CON_NAME_STR;
```

Member Name	Describe
strPresetInfo[MAX_PRESET_POINT]	Preset point information

```
typedef struct tagStrYtctConName
{
    char uChanId;
    char uPlateId;
    unsigned char uPresetNum;
    char ulsSet;
    char chPreName[32];
}
```

```
}_PACKED_STR_YTCT_CON_NAME, *PSTR_YTCT_CON_NAME;
```

Member Name	Describe
uChanId	Video channel number
uPlateId	PTZ 485 address number
uPresetNum	Preset point number
uIsSet	Have you set
chPreName[32]	Preset point name

5.5 Clear preset point: IPCClearPresetPoint(int iparam, PSTR_YTCT_CON_NAME_STR pptPresetName)

Function definition	IPCClearPresetPoint(int iparam, PSTR_YTCT_CON_NAME_STR pptPresetName)	
Function	Clear preset points	
Parameter	Variable	Illustrate
	iparam	User ID
	pptPresetName	Clear the preset point structure parameters
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

5.6 Set preset point: IPCSetPresetPoint(int iparam, PSTR_YTCT_CON_NAME ptPresetName)

Function definition	IPCSetPresetPoint(int iparam, PSTR_YTCT_CON_NAME ptPresetName)	
---------------------	---	--

Function	Set preset points	
Parameter	Variable	Illustrate
	iparam	User ID
	ptPresetName	Set the preset point structure parameters
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

5.7 Get device PTZ timer reset information: IPCGetRestartInfo2(int iparam, PSTR_RESTART_TIME_QUERY2 ptRestartInfo)

Function definition	IPCGetRestartInfo2(int iparam, PSTR_RESTART_TIME_QUERY2 ptRestartInfo)	
Function	Get the device PTZ timing reset information	
Parameter	Variable	Illustrate
	iparam	User ID
	ptRestartInfo	Get the device PTZ timing reset structure parameters
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

```
typedef struct tagStrRestartTimeQuery2
{
    unsigned int uiRestartTime2;
    unsigned int uiRestartFreq2;
    char cEnFlag2;
    char cRev[119];
} _PACKED_ STR_RESTART_TIME_QUERY2, *PSTR_RESTART_TIME_QUERY2;
```

Member Name	Describe
uiRestartTime2	Scheduled restart time
uiRestartFreq2	Restart frequency, 0-6 from Sunday to Saturday, 7 per day
cEnFlag2	Timed restart enable
cRev[119]	Reserved Bytes

5.8 Set the device PTZ timer reset information:

IPCSetRestartInfo2 (int iparam, PSTR_RESTART_TIME_QUERY2 ptRestartInfo)

Function definition	IPCSetRestartInfo2(int iparam, PSTR_RESTART_TIME_QUERY2 ptRestartInfo)	
Function	Set the PTZ timing reset information of the device	
Parameter	Variable	Illustrate
	iparam	User ID
	ptRestartInfo	Set the device PTZ timing reset structure parameters
Return value	A value of 0 indicates success; a value of -1 indicates failure	

Remark	
--------	--

5.9 PTZ reset immediately : IPCPtzReboot (int iparam)

Function definition	IPCPtzReboot(int iparam)	
Function	PTZ immediate reset	
Parameter	Variable	Illustrate
	iparam	User ID
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

5.10 Get PTZ value: IPCPTZValueGet(int iparam, PSTR_YTCT_PTZ_VAL ptPTZValueInfo)

Function definition	IPCPTZValueGet(int iparam, PSTR_YTCT_PTZ_VAL ptPTZValueInfo)	
Function	Get PTZ value	
Parameter	Variable	Illustrate
	iparam	User ID
	ptPTZValueInfo	Get PTZ value structure parameters
Return	A value of 0 indicates success; a value of -1	

value	indicates failure
Remark	

```
typedef struct tagStrYtctPtzVal
{
    unsigned short    usPanVal;
    unsigned short    usTiltVal;
    unsigned short    usZoomVal;
    char              resv[128];
}_PACKED_STR_YTCT_PTZ_VAL, *PSTR_YTCT_PTZ_VAL;
```

Member Name	Describe
usPanVal	Horizontal Angle
usTiltVal	Vertical Angle
usZoomVal	Magnification
resv[128]	Reserved Bytes

5.11 Set PTZ value: IPCPTZValueSet(int iparam, PSTR_YTCT_PTZ_VAL ptPTZValueInfo)

Function definition	IPCPTZValueSet(int iparam, PSTR_YTCT_PTZ_VAL ptPTZValueInfo)	
Function	Setting PTZ Values	
Parameter	Variable	Illustrate
	iparam	User ID
	ptPTZValueInfo	Set the PTZ value structure parameters
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

5.12 Get cruise information: IPCCruiseinfoGet(int iparam, LPS_PTZ_CRUISE_GROUP_INFO ptzCruiseinfo)

Function definition	IPCCruiseinfoGet(int iparam, LPS_PTZ_CRUISE_GROUP_INFO ptzCruiseinfo)	
Function	Get cruise information	
Parameter	Variable	Illustrate
	iparam	User ID
	ptzCruiseinfo	Get cruise information structure parameters
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

```
typedef struct tagPtzCruiseGroupInfo
{
    unsigned char ucCruiseGroupNum;
    unsigned char resv[3];
    S_PTZ_CRUISE_INFO strCruiseInfo[MAX_CRUISE_GROUP_NUM];
} _PACKED_ S_PTZ_CRUISE_GROUP_INFO, *LPS_PTZ_CRUISE_GROUP_INFO;
```

Member Name	Describe
ucCruiseGroupNum	Number of cruise groups
resv[3]	Reserved Bytes
strCruiseInfo[MAX_CRUISE_GROUP_NUM]	Cruise Group Information

```
typedef struct tagPtzCruiseInfo
{
    char chCruiseName[32];
    unsigned char ucPresetPosNum;
    unsigned char resv[3];
    unsigned char uPresetNum[MAX_CRUISE_PRESET_NUM];
```

```

    unsigned char uStayTime[MAX_CRUISE_PRESET_NUM];
    unsigned char uSpeed[MAX_CRUISE_PRESET_NUM];
    unsigned char speed;
    unsigned char resv2[83];
} _PACKED_ S_PTZ_CRUISE_INFO, *LPS_PTZ_CRUISE_INFO;

```

Member Name	Describe
chCruiseName[32]	Cruise track name
ucPresetPosNum	Number of preset points
resv[3]	Reserved Bytes
uPresetNum[MAX_CRUISE_PRESET_NUM]	Preset point number
uStayTime[MAX_CRUISE_PRESET_NUM]	Dwell time (in seconds)
uSpeed[MAX_CRUISE_PRESET_NUM]	Preset speed
speed	speed
resv2[83]	Reserved Bytes

5.13 Set cruise information: IPCCruiseInfoSet(int iparam, LPS_PTZ_CRUISE_GROUP_INFO ptz_cruiseinfo)

Function definition	IPCCruiseInfoSet(int iparam, LPS_PTZ_CRUISE_GROUP_INFO ptz_cruiseinfo)	
Function	Set cruise information	
Parameter	Variable	Illustrate
	iparam	User ID
	ptz_cruiseinfo	Set the cruise information structure parameters
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

5.14 Set the cruise status: IPCCruiseSet(int iparam, PSTR_START_CRUISE ptCruise)

Function definition	IPCCruiseSet(int iparam, PSTR_START_CRUISE ptCruise)	
Function	Set cruise status	
Parameter	Variable	Illustrate
	iparam	User ID
	ptCruise	Set the cruise status structure parameters
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

```
typedef struct startCruise
{
    unsigned char ucCruise;
    unsigned char ucCruisenum;
    unsigned char szResv[122];
}_PACKED_ STR_START_CRUISE, *PSTR_START_CRUISE;
```

Member Name	Describe
ucCruise	0 - Stop cruising ; 1 - Start cruising
ucCruisenum	Cruise track number
szResv[122]	Reserved Bytes

5.15 PTZ pass-through : IPCDevPassThrough(int iparam, PSTR_PASS_THROUGH ptPassThrough)

Function definition	IPCDevPassThrough(int iparam, PSTR_PASS_THROUGH ptPassThrough)	
Function	PTZ transparent transmission	
Parameter	Variable	Illustrate
	iparam	User ID
	ptPassThrough	PTZ transparent transmission structure parameters
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

```
typedef struct tagPassThrough
{
    int comType;
    int comId;
    char szAction[64];
} STR_PASS_THROUGH, * PSTR_PASS_THROUGH;
```

Member Name	Describe
comType	1: RS485, 2: RS232
comId	Port ID, numbered starting from 1
szAction[64]	Transfer content

5.16 Wiper: IPCWiperSet(int iparam, PSTR_WIPER wiper)

Function	IPCWiperSet(int iparam, PSTR_WIPER wiper)
----------	---

definition		
Function	Wipers	
Parameter	Variable	Illustrate
	iparam	User ID
	ptPassThrough	Wiper structure parameters
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

```
typedef struct wiper
{
    int action;
}_PACKED_STR_WIPER, *PSTR_WIPER;
```

Member Name	Describe
action	1 means turn on the wipers, 2 means turn off the wipers, 3 means turn on and off the wipers automatically

5.17 Laser: IPLaserSet(int iparam, PSTR_LASER laser)

Function definition	IPLaserSet(int iparam, PSTR_LASER laser)	
Function	Laser Switch	
Parameter	Variable	Illustrate
	iparam	User ID
	PSTR_LASER	Laser structure parameters
Return	A value of 0 indicates success; a value of -1	

value	indicates failure
Remark	

```
typedef struct laser
{
    int action;
} _PACKED_ STR_LASER, *PSTR_LASER;
```

Member Name	Describe
action	1 means turn on the laser, 2 means turn off the laser, 3 means turn on and off the laser automatically

5.18 Synchronous zoom: IPCZoomSysncSet(int iparam, PSTR_ZOOMSYSNC zoom)

Function definition	IPCZoomSysncSet(int iparam, PSTR_ZOOMSYSNC zoom)	
Function	Synchronous zoom setting	
Parameter	Variable	Illustrate
	iparam	User ID
	zoom	Synchronous zoom parameters
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

```
typedef struct zoomsysnc
{
    unsigned char action;
    unsigned char channel;
    char cRsv[126];
}
```

```
_PACKED_STR_ZOOMSYSNC, *PSTR_ZOOMSYSNC;
```

Member Name	Describe
action	1-Turn on synchronized zoom; 2-Turn off synchronized zoom
channel	Video channel number
cRsv[126]	Reserved Bytes

6 Remote equipment maintenance

6.1 Get device scheduled restart information:

IPCGetRestartInfo(int iparam, PSTR_RESTART_TIME_QUERY ptRestartInfo)

Function definition	IPCGetRestartInfo(int iparam, PSTR_RESTART_TIME_QUERY ptRestartInfo)	
Function	Get device scheduled restart information	
Parameter	Variable	Illustrate
	iparam	User ID
	ptRestartInfo	Get the device scheduled restart information structure parameters
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

```
typedef struct tagStrRestartTimeQuery
{
    unsigned int uiRestartTime;
```

```

        unsigned int    uiRestartFreq;
        char            cEnFlag;
        char cRev[119];
    }_PACKED_STR_RESTART_TIME_QUERY, *PSTR_RESTART_TIME_QUERY;

```

Member Name	Describe
uiRestartTime	Scheduled restart time
uiRestartFreq	Restart frequency, 0-6 from Sunday to Saturday, 7 per day
cEnFlag	Timed restart enable
cRev[119]	Reserved Bytes

6.2 Set device scheduled restart information:

IPCSetRestartInfo(int iparam, PSTR_RESTART_TIME_QUERY ptRestartInfo)

Function definition	IPCSetRestartInfo(int iparam, PSTR_RESTART_TIME_QUERY ptRestartInfo)	
Function	Set device restart schedule	
Parameter	Variable	Illustrate
	iparam	User ID
	ptRestartInfo	Set the device timer restart structure parameters
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

6.3 Reboot the device immediately : IPCReboot (int iparam)

Function definition	IPCReboot(int iparam)	
Function	Restart the device now	
Parameter	Variable	Illustrate
	iparam	User ID
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

7 Reporting Information

7.1 Manual capture: IPCSIMUSNAP(int iparam, PSTR_SIMUL_CAPTURE ptInfo, int blockLabel, unsigned char * pBuf , unsigned int ibufLen, int stayTime)

Function definition	IPCSIMUSNAP(int iparam, PSTR_SIMUL_CAPTURE ptInfo, int blockLabel, unsigned char *pBuf, unsigned int ibufLen, int stayTime)	
Function	Manual capture	
Parameter	Variable	Illustrate
	iparam	User ID
	ptInfo	Manual capture structure

	blockLabel	1-Obtaining manually captured data in a blocking manner
	pBuf	Snapshot image information data
	ibuflen	Length of captured image information data
	stayTime	Waiting time (seconds)
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

```
typedef struct strSimulCapture
{
    unsigned char ucSceneId;
    unsigned char ucRsv[3];
    unsigned char ucUserData[32];
    unsigned char szResv[96];
} _PACKED_ STR_SIMUL_CAPTURE,*PSTR_SIMUL_CAPTURE;
```

Member Name	Describe
ucSceneId	Scene number, starting from 1
ucRsv[3]	Reserved Bytes
ucUserData[32]	User defined fields
szResv[96]	Reserved Bytes

```
typedef struct _tagSnapInfrom_new
{
    unsigned int uiPicNum;
    unsigned char resv[128] ;
    unsigned int *uiParamLen;
    char *pucParamData;
    unsigned int *auiPicLen [2] ;
```

```

char * pszPicData [2] ;
}_PACKED_ SNAP_INFORM_NEW, *PSNAP_INFORM_NEW;

```

Member Name	Describe
uiPicNum	Number of captured pictures
resv[128]	Reserved Bytes
uiParamLen	Length of parameter information for a single image
pucParamData	Single image parameters
auPicLen [2]	Single picture data length
pszPicData [2]	Single image data

```

typedef struct _tagMANUAL_CAPTURE
{
    unsigned int uiSubVersion;
    unsigned int uiGroupId;
    unsigned char ucChanId;
    unsigned char ucSceneId;
    unsigned char ucGroupPicSum;
    unsigned char ucGroupPicSeq ;
    unsigned int uiAlarmType;
    unsigned short usPicWidth ;
    unsigned short usPicHeight ;
    unsigned int uiPicSnapSec;
    unsigned int uiPicSnapMSec ;
    unsigned char ucUserData[32];
    unsigned char ucResv[196];
}_PACKED_ sManualCaptureReport;

```

Member Name	Describe
uiSubVersion	version number
uiGroupId	Group No
ucChanId	Video channel number
ucSceneId	Scene number
ucGroupPicSum	The total number of photos in this group
ucGroupPicSeq	The serial number of this group of photos
uiAlarmType	alarm type
usPicWidth	Image width
usPicHeight	Image height
uiPicSnapSec	Capture time, in seconds
uiPicSnapMSec	Capture time, in milliseconds
ucUserData[32]	User data
ucResv[196]	Reserved Bytes

7.2 GPS information: IPCGpsInfo(int iparam, PSTR_GPS_INFO_QUERY ptInfo, int blockMark, unsigned char *pBuf, unsigned int ibufLen, int stayTime)

Function definition	IPCGpsInfo(int iparam, PSTR_GPS_INFO_QUERY ptInfo, int blockMark, unsigned char *pBuf, unsigned int ibufLen, int stayTime)	
Function	Set target GPS and local GPS	
Parameter	Variable	Illustrate
	iparam	User ID
	ptInfo	Set GPS structure parameters
	blockMark	1- Obtain GPS data in a blocking manner
	pBuf	GPS data buf
	ibufLen	GPS data buf length
	stayTime	Waiting time (seconds)
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

```
typedef struct gpsInfoQuery
{
    int action;
    int duration;
    char cRsv[120];
}STR_GPS_INFO_QUERY, *PSTR_GPS_INFO_QUERY;
```

Member Name	Describe
action	0 is the target GPS, 1 is the local GPS
duration	Temporarily unused
cRsv[120]	Reserved Bytes

```
typedef struct tag_rf_gps_info_t
{
    int lat;
    int latD;
    int latM;
    int lon;
    int lonD;
    int lonM;
    int rxlev;
    float latS;
    float lonS;
} _PACKED_ RF_GPS_INFO_T, *PRF_GPS_INFO_T;
```

Member Name	Describe
lat	latitude
latD	Degrees of latitude
latM	Latitude
lon	longitude
lonD	Degrees of longitude
lonM	Minutes of longitude
rxlev	Signal Strength
latS	Seconds of latitude
lonS	Seconds of longitude

7.3 Distance measurement: IPCDistantInfoSet(int iparam, PSTR_DISTANT_PARAM ptInfo, int blockMark, unsigned char *pBuf, unsigned int ibufLen, int stayTime)

Function	IPCDistantInfoSet(int iparam,
definition	PSTR_DISTANT_PARAM ptInfo, int blockMark, unsigned char *pBuf, unsigned int ibufLen, int

	stayTime)	
Function	Set single and multiple distance measurement	
Parameter	Variable	Illustrate
	iparam	User ID
	ptInfo	Set the ranging structure parameters
	blockMark	1- Obtaining ranging data in a blocking manner
	pBuf	Distance measurement data buf
	ibufLen	Distance measurement data buf length
	stayTime	Waiting time (seconds)
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

```
typedef struct distant_param
{
    char distant_type;
    char distant_time;
    char cRsv[126];
}_PACKED_STR_DISTANT_PARAM, * PSTR_DISTANT_PARAM;
```

Member Name	Describe
distant_type	1 is single distance measurement, 2 is multiple distance measurement
distant_time	Temporarily unused

cRsv[126]	Reserved Bytes
-----------	----------------

```
typedef struct
{
    unsigned int distance;
}Distance_Report_Info;
```

Member Name	Describe
distance	Measuring distance, in cm

7.4 Report data callback function: SetReportCallbackFunc(DataReportCallbackFunc callBack)

Function definition	SetReportCallbackFunc (DataReportCallbackFunc callBack)	
Function	Obtaining reported data	
Function Pointers	typedef int (*DataReportCallbackFunc) (int nType, void* hDev, int nMsgType, char* szData, int nLen, void* pCustomData)	
Parameter	Variable	Illustrate
	nType	Data type: 0x1001 Report local latitude and longitude data; 0x1002 Report target latitude and longitude data; 0x1003 Report high temperature target; 0x1004 Report

		distance measurement data; 0xA002 Report user manual snapshot; 0xA003 Automatic snapshot of fire alarm; 0xD003 Report fire alarm information.
	hDev	Device IP
	nMsgType	Message Type
	szData	Report data content
	nLen	Report data length
	pCustomData	User defined data
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

```
typedef struct
{
    unsigned int uiTargetType;
    unsigned short usStartX;
    unsigned short usStartY;
    unsigned short usWidth;
    unsigned short usHeight;
}_PACKED_ALARM_RECT;
```

Member Name	Describe
uiTargetType	Alarm box type
usStartX	Alarm box starting coordinate X
usStartY	Alarm box starting coordinate Y
usWidth	Alarm frame width
usHeight	Alarm frame height

```
typedef struct
{
    unsigned int uiChanId;
    unsigned int uiRectNum;
    ALARM_RECT pAlarmRect[6];
} _PACKED_ Alarm_Report_Info;
```

Member Name	Describe
uiChanId	Video channel number
uiRectNum	Number of alarm boxes
pAlarmRect[6]	Alarm box information

7.5 Video callback function:

SetVideoFrameCallBack(VideoFrameCallBack callBack)

Function definition	SetVideoFrameCallBack(VideoFrameCallBack callBack)	
Function	Get video preview data	
Function Pointers	typedef int (*VideoFrameCallBack)(int iparam, void* pBuf)	
Parameter	Variable	Illustrate
	iparam	User ID
	pBuf	Video frame buf
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

```
typedef struct VideoFrameInfo
{
    unsigned int uiFrameHead;
```

```

unsigned int      uiChanId;
unsigned int      uiFrameLen;
unsigned int      uiSequence;
unsigned long long ullTimeStamp;
unsigned short    usEncW;
unsigned short    usEncH;
unsigned char     bKeyFrame;
unsigned char     bStreamType;
unsigned char     ucFrameRate;
unsigned char     ucVideoFormat;
unsigned char     ucResv[120];
unsigned char     *pData;
unsigned int      uiFrameTail;
}VideoFrameInfo_S,*LPVideoFrameInfo;

```

成员名称	Describe
uiFrameHead	Frame header identifier
uiChanId	Video channel number
uiFrameLen	Frame length
uiSequence	sequence
ullTimeStamp	Timestamp
usEncW	Video Width
usEc	Video Height
bKeyFrame	1- Video I frame, 0 - Video P frame
bStreamType	1- Video, 2 - Audio, 3 - Audio/Video Mix
ucFrameRate	Frame rate
ucVideoFormat	Video format ; 1 - NTSC , 2 - PAL
ucResv[120]	Reserved Bytes
pData	Video Data
uiFrameTail	Frame end mark

7.6 Start receiving video frames: StartVideoRequest(int iparam)

Function definition	StartVideoRequest(int iparam)	
Function	Enable video frame reception	
Parameter	Variable	Illustrate

	iparam	User ID
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

7.7 Stop receiving video frames: StopVideoRequest(int iparam)

Function definition	StopVideoRequest(int iparam)	
Function	Stop receiving video frames	
Parameter	Variable	Illustrate
	iparam	User ID
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

7.8 Get control authority: IPCFireModeGet(int iparam, PSTR_FIRE_TEC_MODE ModeSet)

Function definition	IPCFireModeGet(int iparam, PSTR_FIRE_TEC_MODE ModeSet)	
Function	Get control permissions	
Parameter	Variable	Illustrate

	iparam	User ID
	ModeSet	Control mode structure parameters
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

```
typedef struct firedetecmode
{
    char mode;
    char cRsv[63];
}_PACKED_ STR_FIRE_TEC_MODE, *PSTR_FIRE_TEC_MODE;
```

Member Name	Describe
mode	Control mode; 0-active, 1-passive
cRsv[63]	Reserved Bytes

7.9 Set control rights: IPCFireModeSet(int iparam, PSTR_FIRE_TEC_MODE ModeSet)

Function definition	IPCFireModeSet(int iparam, PSTR_FIRE_TEC_MODE ModeSet)	
Function	Set control permissions	
Parameter	Variable	Illustrate
	iparam	User ID
	ModeSet	Control mode structure parameters
Return	A value of 0 indicates success; a value of -1	

value	indicates failure
Remark	

7.10 Fireworks alarm parameter information:

```
typedef struct _tagFireSmokeAlarm
{
    unsigned int uiSubVersion;
    unsigned int uiGroupId;
    unsigned char ucChannel;
    unsigned char ucGroupPicSum;
    unsigned char    ucAlarmType;
    unsigned char    ucAlarmChan;
    unsigned int    uiPicSnapSec;
    unsigned int    uiPicSnapMSec;
    unsigned int    uiDistance;
    rf_gps_info_t    localGPS;
    rf_gps_info_t    targetGPS;
    PTZ              strPtz;
    RECT             strRect;
    unsigned char ucResv[128];
    unsigned int    *auiPicLen;
    char            *pszPicData;
}_PACKED_ FireSmokeAlarm;
```

Member Name	Describe
uiSubVersion	Version number
uiGroupId	Group No
ucChannel	Video channel number
ucGroupPicSum	The total number of pictures in this group
ucAlarmType	Alarm type:1-smoke,2-fire,3-pyrotechnics, 4-high temperature, etc.
ucAlarmChan	Visible light or thermal imaging alarm channel
uiPicSnapSec	Capture time (seconds)
uiPicSnapMSec	Capture time (milliseconds)
uiDistance	Measuring distance
localGPS	Local GPS
targetGPS	Target GPS
strPtz	PTZ value
strRect	Visible light alarm box, thermal imaging alarm box

ucResv[128]	Reserved Bytes
auiPicLen	Length of captured image data
pszPicData	Captured image data

```
typedef struct _RECT
{
    unsigned int left;
    unsigned int top;
    unsigned int right;
    unsigned int bottom;
} RECT;
```

Member Name	Describe
left	Alarm box starting coordinate X
top	Alarm box starting coordinate Y
right	Alarm box end coordinate X
bottom	Alarm box end coordinate Y

```
typedef struct _PTZ
{
    unsigned int pan;
    unsigned int tilt;
    unsigned int zoom;
    unsigned int pan_view;
    unsigned int tilt_view;
} PTZ;
```

Member Name	Describe
pan	Horizontal Angle
tilt	Vertical Angle
zoom	Magnification
pan_view	Horizontal field of view
tilt_view	Vertical field of view

```
typedef struct
{
    int lat;
    int latD;
    int latM;
    int lon;
    int lonD;
    int lonM;
    int rxlev;
    float latS;
```

```
float lonS;
}rf_gps_info_t;
```

Member Name	Describe
lat	Latitude
latD	Degrees of latitude
latM	Latitude
lon	longitude
lonD	Degrees of longitude
lonM	Minutes of longitude
rxlev	Signal Strength
latS	Seconds of latitude
lonS	Seconds of longitude

7.11 Exception callback

**function:SetExceptionCallBack(DisconnectionCallback
callBack)**

Function definition	SetExceptionCallBack(DisconnectionCallback callBack)	
Function	Exception callback	
Function Pointers	typedef int (*DisconnectionCallback) (int iDeviceIP, int iparam, void* pCustomData)	
Parameter	Variable	Illustrate
	iDeviceIP	Device IP
	iparam	User ID
	pCustomData	User-defined data
Return value	A value of 0 indicates success; a value of -1 indicates failure	

Remark	
--------	--

8.1 Force exit: IPC_FORCE_EXIT(int iparam)

Function definition	IPC _FORCE_EXIT (int iparam)	
Function	Force quit	
Parameter	Variable	Illustrate
	iparam	User ID
Return value	A value of 0 indicates success; a value of -1 indicates failure	
Remark		

9.1 Error code definition

Error Name	Error Value	Illustrate
SDK_REPEAT_INIT	0x80001001	Repeated initialization
SDK_UNINIT	0x80001002	Uninitialized
DEV_REPEAT_INIT	0x80001003	Device duplicate login
DEV_UNLOGIN	0x80001004	Device not logged in
SDK_LOGINERR	0x80001005	Device login

		failed
SDK_TIMEOUT	0x80001006	Time out
SDK_NETERR	0x80001007	Network anomaly
SDK_CONNECT_FAIL	0x80001008	Failed to connect to device
SDK_CONNECT_CLOSE	0x80001009	Connection closed
SDK_PARAMERR	0x80002001	Parameter error
SDK_PROTOERR	0x80002002	Protocol exception
SDK_NOMEM	0x80002003	Insufficient memory length
SDK_OVER_MAXLINK	0x80002004	The number of users connected to the device exceeds the maximum value