



金卡读写器 JSC280 驱动接口说明

部门	研发部
文档用途	JSC280 驱动接口说明
文档编号	
文档版本号	V1.0
文档创建日期	2012/09/21
最后修改日期	
文档作者	皮艳良
项目负责人	
审核人	
项目名称	金卡读卡器

[illegible]



一、 驱动介绍

JSC280 具有三合一、五合一两个版本，其中包括磁条卡读写器、IC 卡读写器、二代证读卡器，由于不同的需求来进行组合。其中磁条卡读写器、IC 卡读写器、二代证用串口通讯方式。

二、 驱动名称及依赖关系

模块名称	动态库名	依赖文件
磁条卡读写器	CardReader.dll	
IC 卡读写器	IcReader.dll	
二代证读卡器	IDReader.dll	WltRS.dll,sdtapi.dll, termb.lic

三、 磁条卡读写器驱动接口说明

3.1 驱动功能

提供对磁条卡的读、写操作、读取状态及设置二磁道格式。

3.2 接口返回码定义

#define ERR_SUCCESS	1	//执行成功
#define ERR_MAX_BUFFER	-1	//超出最大缓存
#define ERR_NO_DATA_TWO	-2	//2 轨道无数据
#define ERR_NO_DATA_THREE	-3	//3 轨道无数据
#define ERR_NO_DATA_TWO_AND_THREE	-4	//2、3 轨道都无数据
#define ERR_NO_DATA_TWO_BUT_THREE	-5	//2 轨有数据，但 3 轨无数据
#define ERR_NO_DATA_THREE_BUT_TWO	-6	//2 轨无数据，但 3 轨有数据
#define ERR_INTPUT_PARAMETER	-7	//输入参数有误
#define ERR_STATUS_FAIL_TWO	-8	//2 磁道操作状态失败
#define ERR_STATUS_FAIL_THREE	-9	//3 磁道操作状态失败
#define ERR_STATUS_FAIL_TWO_AND_THREE	-10	//2、3 磁道操作状态失败
#define ERR_UNKNOW	-99	//未知错误



3.3 接口说明

3.3.1 打开串口通讯

函数原型: `int WINAPI ZtDevice_CR_Open(int nComPort, int nBaudRate);`

函数名	ZtDevice_CR_Open			
属性	参数名	参数类型	参数含义	备注
入参	nComPort	int	通讯串口号	1:COM1, 2:COM2...
入参	nBaudRate	int	通讯波特率	eg: 9600,19200,38400,57600,115200
出参				无
返回值		int		见:3.2
注: 使用此接口后与设备以串口方式进行通讯				

3.3.2 关闭串口通讯

函数原型: `int WINAPI ZtDevice_CR_Close();`

函数名	ZtDevice_CR_Close			
属性	参数名	参数类型	参数含义	备注
入参				无
出参				无
返回值		int		见:3.2
注: 在串口通讯情况下关闭通讯, 必须与 ZtDevice_CR_Open 成对使用				

3.3.3 打开 USB 通讯

函数原型: `int WINAPI ZtDevice_CR_UsbOpen(int nPid, int nVid);`

函数名	ZtDevice_CR_UsbOpen			
属性	参数名	参数类型	参数含义	备注
入参	nPid	int	USB 设备 PID	
入参	nVid	int	USB 设备 VID	
出参				无
返回值		int		见:3.2
注: 使用此接口后与设备以 USB 方式进行通讯				

3.3.4 关闭 USB 通讯

函数原型: `int WINAPI ZtDevice_CR_UsbClose();`

函数名	ZtDevice_CR_UsbClose			
属性	参数名	参数类型	参数含义	备注



入参				无
出参				无
返回值		int		见:3.2
注：在 USB 通讯情况下关闭通讯，必须与 ZtDevice_CR_UsbOpen 成对使用				

3.3.5 设备复位

函数原型：int WINAPI ZtDevice_CR_Reset();

函数名	ZtDevice_CR_Reset			
属性	参数名	参数类型	参数含义	备注
入参				无
出参				无
返回值		int		见:3.2
注：设备热复位				

3.3.6 读二磁道数据

函数原型：int WINAPI ZtDevice_CR_ReadTrack2(LPBYTE byTrackDataTwo,LPINT nDataTwoLen,int nTimeOut);

函数名	ZtDevice_CR_ReadTrack2			
属性	参数名	参数类型	参数含义	备注
出参	byTrackDataTwo	LPBYTE	二磁道数据	ASC 码
出参	nDataTwoLen	LPINT	二磁道数据长度	
入参	nTimeOut	int	超时时间	单位：秒
返回值		int		见:3.2

3.3.7 读三磁道数据

函数原型：int WINAPI ZtDevice_CR_ReadTrack3(LPBYTE byTrackDataThree,LPINT nDataThreeLen,int nTimeOut);

函数名	ZtDevice_CR_ReadTrack3			
属性	参数名	参数类型	参数含义	备注
出参	byTrackDataThree	LPBYTE	三磁道数据	ASC 码
出参	nDataThreeLen	LPINT	三磁道数据长度	
入参	nTimeOut	int	超时时间	单位：秒
返回值		int		见:3.2



3.3.8 合读二三磁道数据

函数原型: `int WINAPI ZtDevice_CR_ReadTrack23(LPBYTE byTrackDataTwo,LPINT nDataTwoLen,LPBYTE byTrackDataThree,LPINT nDataThreeLen,int nTimeOut);`

函数名	ZtDevice_CR_ReadTrack23			
属性	参数名	参数类型	参数含义	备注
出参	byTrackDataTwo	LPBYTE	二磁道数据	ASC 码
出参	nDataTwoLen	LPINT	二磁道数据长度	
出参	byTrackDataThree	LPBYTE	三磁道数据	ASC 码
出参	nDataThreeLen	LPINT	三磁道数据长度	
入参	nTimeOut	int	超时时间	单位: 秒
返回值		int		见:3.2

3.3.9 写二磁道数据

函数原型: `int WINAPI ZtDevice_CR_WriteTrack2(LPBYTE byTrackDataTwo,int nDataTwoLen,int nTimeOut);`

函数名	ZtDevice_CR_WriteTrack2			
属性	参数名	参数类型	参数含义	备注
入参	byTrackDataTwo	LPBYTE	二磁道数据	ASC 码
入参	nDataTwoLen	int	二磁道数据长度	
入参	nTimeOut	int	超时时间	单位: 秒
返回值		int		见:3.2

3.3.10 写三磁道数据

函数原型: `int WINAPI ZtDevice_CR_WriteTrack3(LPBYTE byTrackDataThree,int nDataThreeLen,int nTimeOut);`

函数名	ZtDevice_CR_WriteTrack3			
属性	参数名	参数类型	参数含义	备注
入参	byTrackDataThree	LPBYTE	三磁道数据	ASC 码
入参	nDataThreeLen	int	三磁道数据长度	
入参	nTimeOut	int	超时时间	单位: 秒
返回值		int		见:3.2



3.3.11 合写二三磁道数据

函数原型: `int WINAPI ZtDevice_CR_WriteTrack23(LPBYTE byTrackDataTwo,int
nDataTwoLen,LPBYTE byTrackDataThree,int nDataThreeLen,int nTimeOut);`

函数名	ZtDevice_CR_WriteTrack23			
属性	参数名	参数类型	参数含义	备注
入参	byTrackDataTwo	LPBYTE	二磁道数据	ASC 码
入参	nDataTwoLen	int	二磁道数据长度	
入参	byTrackDataThree	LPBYTE	三磁道数据	ASC 码
入参	nDataThreeLen	int	三磁道数据长度	
入参	nTimeOut	int	超时时间	单位: 秒
返回值		int		见:3.2

3.3.12 取二磁道操作状态

函数原型: `int WINAPI ZtDevice_CR_GetTrack2Status();`

函数名	ZtDevice_CR_GetTrack2Status			
属性	参数名	参数类型	参数含义	备注
入参				无
出参				无
返回值		int		见:3.2
注: 获取前一次对二磁道操作是否成功				

3.3.13 取三磁道操作状态

函数原型: `int WINAPI ZtDevice_CR_GetTrack3Status();`

函数名	ZtDevice_CR_GetTrack3Status			
属性	参数名	参数类型	参数含义	备注
入参				无
出参				无
返回值		int		见:3.2
注: 获取前一次对三磁道操作是否成功				

3.3.14 取二三磁道操作状态

函数原型: `int WINAPI ZtDevice_CR_GetTrack23Status();`

函数名	ZtDevice_CR_GetTrack3Status			
属性	参数名	参数类型	参数含义	备注



入参				无
出参				无
返回值		int		见:3.2
注：获取前一次对二三磁道操作是否成功				

3.3.15 设二磁道为 210BPI

函数原型：int WINAPI ZtDevice_CR_SetTrack2BPI();

函数名	ZtDevice_CR_SetTrack2BPI			
属性	参数名	参数类型	参数含义	备注
入参				无
出参				无
返回值		int		见:3.2

3.3.16 设二磁道为 75BPI

函数原型：int WINAPI ZtDevice_CR_SetTrack275BPI();

函数名	ZtDevice_CR_SetTrack2BPI			
属性	参数名	参数类型	参数含义	备注
入参				无
出参				无
返回值		int		见:3.2

四、 IC 卡读写器驱动接口说明

4.1 驱动功能

提供对接触式、非接触式 IC 卡上电、下电、APDU 等操作的接口。

4.2 接口返回码定义

各 API 函数返回值的信息定义

```
enum IC_RetStatus
```

```
{
```

```
    RetStatus_FALSE,                // 函数执行失败
```

```
    RetStatus_TURE,                 // 函数执行成功
```

```
    RetStatus_Not_Define,           // 未定义的返回值
```




```
RetStatus_TIMEOUT = 3, //接收超时
RetStatus_NotSupportBaudRate, //读写器不支持该串口波特率
RetStatus_ERROR_MEMORY_LIMIT, //内存不够
RetStatus_CardMemoryLimit, //待写数据超出存储卡的容量
RetStatus_Not_Support, // 10H,01H 不支持接触式用户卡
RetStatus_Not_In_Position, // 10H,02H 接触式用户卡未插到位
RetStatus_Power_Off, // 10H,04H 接触式用户卡未上电
RetStatus_Power_Failed, // 10H,05H 接触式用户卡上电失败
RetStatus_PSAM_Not_Support, // 20H,01H 不支持 PSAM 卡
RetStatus_PSAM_Power_Failed, // 20H,05H PSAM 卡上电失败
RetStatus_Contextless_Card_Not_Support, //30H,01H 不支持非接触用户卡
RetStatus_Contextless_Card_ActiveFailed, //30H,05H 非接触用户卡激活失败
RetStatus_WaitingCard_EnterTimeOut, // 30H,06H 等待卡进入感应区超时
RetStatus_TooManyCardInside, // 30H,09H 有多张卡在感应区
RetStatus_WaitingCard_ExitTimeOut, // 30H,06H 等待卡离开感应区超时
RetStatus_SetHalt_Failed, // 30H,08H 设置卡 Halt 状态失败
RetStatus_No_Response, // 10H,06H 操作接触式用户卡数据无回应
RetStatus_Failed, // 10H,07H 操作接触式用户卡数据出现错误
RetStatus_PSAM_Power_Off, // 20H,04H PSAM 卡未上电
RetStatus_PSAM_No_Response, // 20H,06H 操作 PSAM 卡数据无回应
RetStatus_PSAM_Failed, // 20H,07H 操作 PSAM 卡数据出现错误
RetStatus_Contextless_Card_Not_Active, // 30H,04H 非接触式用户卡未激活
RetStatus_Contextless_Card_No_Response, //30H,06H 操作非接触式用户卡数据无回应
RetStatus_Contextless_Card_Failed, // 30H,07H 操作非接触式用户卡数据出现错误
RetStatus_GetCardType_Error, //获取接触式存储卡类型失败
RetStatus_UndefineCardType, //未知的存储卡类型
RetStatus_GetAPDUBack_Err, //D006 读卡器从卡片中得到的指令格式出错
};
```

4.3 结构体定义说明

4.3.1 IC 卡读写器模块类型

```
enum ReaderType
{
    CONNECTED_IC_Card = 0, //接触式 IC 卡
    CONNECTLESS_IC_Card //非接触式 IC 卡
};
```

4.3.2 IC 卡读写器波特率

```
enum IcBaudRate
{
    B9600 = 0, //波特率为 9600
    B19200, //波特率为 19200
};
```



```
B38400,          //波特率为 38400
B57600,          //波特率为 57600
B115200         //波特率为 115200
```

```
};
```

4.3.3 LED 灯类型

```
enum LedType
{
    Led_Green ,      //绿灯
    Led_Red          //红灯
};
```

4.3.4 接触式用户卡状态

```
enum Ic_CardStatus
{
    Status_Not_Support,      //不支持接触式用户卡
    Status_Not_In_Position,  //接触式用户卡未插到位
    Status_Power_On,         //接触式用户卡已上电
    Status_Power_Off,        //接触式用户卡未上电
    Status_Power_Failed,     //上电失败
    Status_No_Response,      //操作接触式用户卡数据无回应
    Status_Falied,           //操作接触式用户卡数据出现错误
    Status_PSAM_Not_Support, //不支持 PSAM 卡
    Status_PSAM_Power_On,    //PSAM 卡已上电
    Status_PSAM_Power_Off,   //PSAM 卡未上电
    Status_PSAM_No_Response, //操作 PSAM 卡数据无回应
    Status_PSAM_Failed,      //操作 PSAM 卡数据出现错误
    Status_PSAM_Power_Failed, //PSAM 卡上电失败
    Status_Contectless_Card_Not_Support, //不支持非接触式用户卡
    Status_Contectless_Card_Not_Active,  //非接触式用户卡未激活
    Status_Contectless_Card_No_Response, //操作非接触式用户卡数据无回应
    Status_Contectless_Card_Failed       //操作非接触式用户卡数据出现错误
};
```

4.3.5 用户卡卡座

```
enum UserCardNo
{
    Connected_Ic_Card_1 = 0x00,      //接触式用户卡
    Connected_Ic_Card_2 ,            //接触式用户卡
    Connected_Ic_Card_3 ,            //接触式用户卡
    Connected_Ic_Card_4 ,            //接触式用户卡
    Connected_PSAM_1 = 0x10,         //PSAM 卡座
    Connected_PSAM_2,                //PSAM 卡座
    Connected_PSAM_3,                //PSAM 卡座
    Connected_PSAM_4,                //PSAM 卡座
};
```



```
Connectless_Ic_Card = 0xff //非接触式 IC 卡
```

```
};
```

4.3.6 接触式用户卡种类

```
enum UserCardType
```

```
{
```

```
    CardType_AT_ST_GPM = 0x01,
```

```
    CardType_45D041,
```

```
    CardType_SLE18_28,
```

```
    CardType_SLE32_42,
```

```
    CardType_AT24C32_64,
```

```
    CardType_93C46,
```

```
    CardType_AT153,
```

```
    CardType_AT102,
```

```
    CardType_AT1608
```

```
};
```

4.3.7 读写器版本信息

```
struct IcVersion
```

```
{
```

```
    BYTE CUP_Interface [64]; //由银联定义的读写器规范版本信息
```

```
    BYTE Acquirer_Interface[64]; //由受理方定义的版本信息
```

```
    BYTE Len; // ProInfomation 的长度
```

```
    BYTE ProInfomation[512]; //厂家自定义信息
```

```
};
```

4.3.8 非接触式 IC 卡激活时应答数据单元

```
struct ActiveActionResponse
```

```
{
```

```
    BYTE Type; //0AH A类卡; 0BH B类卡
```

```
    BYTE UIDLen; //卡 UID 长度
```

```
    BYTE UID[64]; //卡 UID
```

```
    BYTE Len; // ATR 长度
```

```
    BYTE ATR[512]; //卡上电成功的 ATR
```

```
};
```



4.4 接口说明

4.4.1 打开串口通讯

函数原型: `int __stdcall ZtDevice_IC_OpenComm(int nComPort, int nBaudRate);`

函数名	ZtDevice_IC_OpenComm			
属性	参数名	参数类型	参数含义	备注
入参	nComPort	int	通讯串口号	1:COM1, 2:COM2...
入参	nBaudRate	int	通讯波特率	eg: 9600,19200,38400,57600,115200
出参				无
返回值		int		见:4.2
注: 使用此接口后与设备以串口方式进行通讯				

4.4.2 关闭串口通讯

函数原型: `void __stdcall ZtDevice_IC_CloseComm()`

函数名	ZtDevice_IC_CloseComm			
属性	参数名	参数类型	参数含义	备注
入参				无
出参				无
返回值				无
注: 在串口通讯情况下关闭通讯, 必须与 ZtDevice_IC_OpenComm 成对使用				

4.4.3 打开 USB 通讯

函数原型: `int __stdcall ZtDevice_IC_OpenUsb(int nPid, int nVid);`

函数名	ZtDevice_IC_OpenUsb			
属性	参数名	参数类型	参数含义	备注
入参	nPid	int	USB 设备 PID	
入参	nVid	int	USB 设备 VID	
出参				无
返回值		int		见:4.2
注: 使用此接口后与设备以 USB 方式进行通讯				

4.4.4 关闭 USB 通讯

函数原型: `void __stdcall ZtDevice_IC_CloseUsb();`



函数名	ZtDevice_IC_CloseUsb			
属性	参数名	参数类型	参数含义	备注
入参				无
出参				无
返回值				无
注：在 USB 通讯情况下关闭通讯，必须与 ZtDevice_IC_OpenUsb 成对使用				

4.4.5 判断 USB 是否有连接

函数原型: `int __stdcall ZtDevice_IC_UsbIsConnective();`

函数名	ZtDevice_IC_UsbIsConnective			
属性	参数名	参数类型	参数含义	备注
入参				无
出参				无
返回值		int		True 已连接 False 未连接
注：				

4.4.6 关闭非接天线

函数原型: `int __stdcall ZtDevice_IC_ShutDownConnectless();`

函数名	ZtDevice_IC_ShutDownConnectless			
属性	参数名	参数类型	参数含义	备注
入参				无
出参				无
返回值		int		0[失败], 1[成功], 3[超时]...
注：				

4.4.7 设置通讯参数

函数原型: `short __stdcall ZtDevice_IC_SetParameter(ReaderType DeviceType, IcBaudRate BaudRate);`

函数名	ZtDevice_IC_SetParameter			
属性	参数名	参数类型	参数含义	备注
入参	DeviceType	ReaderType	读写器模块	
	BaudRate	IcBaudRate	通讯波特率	
出参				无
返回值		short		0[失败], 1[成功], 3[超时]...



注:

4.4.8 查看设备版本

函数原型: `short __stdcall ZtDevice_IC_GetVersion(ReaderType DeviceType, unsigned int Len, IcVersion *Version);`

函数名	ZtDevice_IC_GetVersion			
属性	参数名	参数类型	参数含义	备注
入参	DeviceType	ReaderType	读写器模块	
	Len	unsigned int	IcVersion 结构	成员 ProInfomation 分配的空间的长度
出参	Version	IcVersion *	版本信息	
返回值				0[失败], 1[成功], 3[超时]...
注:				

4.4.9 软复位

函数原型: `short __stdcall ZtDevice_IC_SoftReset(ReaderType DeviceType);`

函数名	ZtDevice_IC_SoftReset			
属性	参数名	参数类型	参数含义	备注
入参	DeviceType	ReaderType	读写器模块	
出参				无
返回值				0[失败], 1[成功], 3[超时]...
注:				

4.4.10 蜂鸣器控制

函数原型: `short __stdcall ZtDevice_IC_Beep(BYTE LastTime, BYTE StopTime, BYTE Counts);`

函数名	ZtDevice_IC_Beep			
属性	参数名	参数类型	参数含义	备注
入参	LastTime	BYTE	鸣叫持续时间(100ms)	
	StopTime	BYTE	鸣叫停止时间(100ms)	
	Counts	BYTE	鸣叫次数	
出参				无
返回值				0[失败], 1[成功], 3[超时]...
注:				



4.4.11 LED 控制

函数原型: `short __stdcall ZtDevice_IC_ControlLed(LedType Led, bool Control);`

函数名	ZtDevice_IC_ControlLed			
属性	参数名	参数类型	参数含义	备注
入参	Led	LedType	灯光选择	Led_Green[绿灯], Led_Red[红灯]
	Control	bool	灯开关控制	ture[灯亮], false[灯灭]
出参				无
返回值		short		0[失败], 1[成功], 3[超时]...
注:				

4.4.12 判断接触式卡片状态

函数原型: `short __stdcall ZtDevice_IC_GetStatus(UserCardNo CardNo, BYTE* Status);`

函数名	ZtDevice_IC_GetStatus			
属性	参数名	参数类型	参数含义	备注
入参	CardNo	UserCardNo	用户卡卡座	
出参	Status	BYTE*	卡片状态	返回值可以用枚举类型 Ic_CardStatus 赋值
返回值		short		0[失败], 1[成功], 3[超时]...
注:				

4.4.13 接触式卡片上电

函数原型: `short __stdcall ZtDevice_IC_PowerOn(unsigned short DelayTime, UserCardNo CardType, unsigned short *Len, BYTE *ATR);`

函数名	ZtDevice_IC_PowerOn			
属性	参数名	参数类型	参数含义	备注
入参	DelayTime	unsigned short	等待插卡时间(ms)	0 表示无需等待, 无卡直接返回
	CardType	UserCardNo	用户卡卡座	
出参	Len	unsigned short *	为 Atr 分配的空间的长度,	返回 ATR 的实际长度
	ATR	BYTE *	上电返回数据	ATR 的接收缓冲最小为字节
返回值		short		0[失败], 1[成功], 3[超时]...
注:				

4.4.14 接触式卡片下电

函数原型: `short __stdcall ZtDevice_IC_PowerOff(UserCardNo CardNo);`



函数名	ZtDevice_IC_PowerOff			
属性	参数名	参数类型	参数含义	备注
入参	CardNo	UserCardNo	用户卡卡座	
出参				无
返回值		short		0[失败], 1[成功], 3[超时]...
注:				

4.4.15 激活非接触式 IC 卡

函数原型: `short __stdcall ZtDevice_IC_ConnectlessActive(unsigned short DelayTime, unsigned short Len, ActiveActionResponse *Response);`

函数名	ZtDevice_IC_ConnectlessActive			
属性	参数名	参数类型	参数含义	备注
入参	DelayTime	unsigned short	等待进入感应区时间(ms)	0 表示无需等待, 无卡直接返回; 0xFFFF 表示一直等待
	Len	unsigned short		为结构 ActiveActionResponse 的成员 ATR 分配的空间的长度
出参	Response	ActiveActionResponse *	激活成功应答数据单元	
返回值		short		0[失败], 1[成功], 3[超时]...
注:				

4.4.16 激活非接触式 IC 卡

函数原型: `short __stdcall ZtDevice_IC_ConnectlessActive_B(unsigned short DelayTime, unsigned short *Len, BYTE *Recv);`

函数名	ZtDevice_IC_ConnectlessActive_B			
属性	参数名	参数类型	参数含义	备注
入参	DelayTime	unsigned short	等待进入感应区时间(ms)	0 表示无需等待, 无卡直接返回; 0xFFFF 表示一直等待
出参	Len	unsigned short *	为结构 Recv 的空间的长度	
	Recv	BYTE *	激活成功应答数据	
返回值		short		0[失败], 1[成功], 3[超时]...
注:				

4.4.17 设置非接触式 IC 卡为 HALT 状态

函数原型: `short __stdcall ZtDevice_IC_ConnectlessHalt(unsigned short DelayTime);`

函数名	ZtDevice_IC_ConnectlessHalt			
-----	-----------------------------	--	--	--



属性	参数名	参数类型	参数含义	备注
入参	DelayTime	unsigned short	等待离开感应区时间(ms)	0 表示无需等待，直接返回; 0xFFFF 表示一直等待
出参				无
返回值		short		0[失败], 1[成功], 3[超时]...
注:				

4.4.18 应用层命令传输 APDU:

函数原型: `short __stdcall ZtDevice_IC_SendApu(UserCardNo CardNo, BYTE * CApu, unsigned short Len, unsigned short * RecvLen, BYTE* RApu);`

函数名	ZtDevice_IC_SendApu			
属性	参数名	参数类型	参数含义	备注
入参	CardNo	UserCardNo	用户卡卡座	
	CApu	BYTE *	命令数据单元	
	Len	unsigned short	命令数据单元长度	
出参	RecvLe	unsigned short *	为 RApu 分配的空间的长度	
	RApu	BYTE*	应答数据单元	
返回值		short		0[失败], 1[成功], 3[超时]...
注:				

4.4.19 设置接触式存储卡种类

函数原型: `short __stdcall ZtDevice_IC_SetCardType(UserCardNo CardNo, UserCardType * CardType);`

函数名	ZtDevice_IC_SetCardType			
属性	参数名	参数类型	参数含义	备注
入参	CardNo	UserCardNo	用户卡卡座	
	CardType	UserCardType *	用户卡类型	
出参				无
返回值		short		0[失败], 1[成功], 3[超时]...
注:				

4.4.20 识别接触式存储卡种类

函数原型: `short __stdcall ZtDevice_IC_GetCardType(UserCardNo CardNo, BYTE * CardType);`

函数名	ZtDevice_IC_GetCardType			
属性	参数名	参数类型	参数含义	备注
入参	CardNo	UserCardNo	用户卡卡座	



深圳市吉盛科技有限公司

ShenZhen Jisheng Technology co.,Ltd.

出参	CardType	BYTE *	用户卡类型	返回值可以用枚举类型 UserCardType 赋值
返回值		short		0[失败], 1[成功], 3[超时]...
注:				

4.4.21 接触式存储卡密钥初始化

函数原型: `short __stdcall ZtDevice_IC_InitializeKey(UserCardNo CardNo, BYTE * Key, BYTE Len);`

函数名	ZtDevice_IC_InitializeKey			
属性	参数名	参数类型	参数含义	备注
入参	CardNo	UserCardNo	用户卡卡座	
	Key	BYTE *	密钥数据	
	Len	BYTE	密钥数据长度	
出参				无
返回值		short		0[失败], 1[成功], 3[超时]...
注:				

4.4.22 接触式存储卡密码校验

函数原型: `short __stdcall ZtDevice_IC_VerifyPwd(UserCardNo CardNo, BYTE * Key, BYTE Len);`

函数名	ZtDevice_IC_VerifyPwd			
属性	参数名	参数类型	参数含义	备注
入参	CardNo	UserCardNo	用户卡卡座	
	Key	BYTE *	密钥数据	
	Len	BYTE	密钥数据长度	
出参				无
返回值		short		0[失败], 1[成功], 3[超时]...
注:				

4.4.23 接触式存储卡读数据

函数原型: `short __stdcall ZtDevice_IC_ReadStoreData(UserCardNo CardNo, unsigned short Address, unsigned short Len, BYTE * Response);`

函数名	ZtDevice_IC_ReadStoreData			
属性	参数名	参数类型	参数含义	备注
入参	CardNo	UserCardNo	用户卡卡座	
	Address	unsigned short	起始地址	
	Len	unsigned short	待读数据长度	
出参	Response	BYTE *	应答数据	
返回值		short		0[失败], 1[成功], 3[超时]...



注:

4.4.24 接触式存储卡写数据

函数原型: `short __stdcall ZtDevice_IC_WriteStoreData(UserCardNo CardNo, unsigned short Address, BYTE *Data, unsigned short Len);`

函数名	ZtDevice_IC_WriteStoreData			
属性	参数名	参数类型	参数含义	备注
入参	CardNo	UserCardNo	用户卡卡座	
	Address	unsigned short	起始地址	
	Data	BYTE *	待写数据	
	Len	unsigned short	待写数据长度	
出参				无
返回值		short		0[失败], 1[成功], 3[超时]...
注:				

4.4.25 激活非接触式存储卡

函数原型: `short __stdcall ZtDevice_IC_ConnectlessActiveStoreCard(unsigned short DelayTime, ActiveActionResponse * Response);`

函数名	ZtDevice_IC_ConnectlessActiveStoreCard			
属性	参数名	参数类型	参数含义	备注
入参	DelayTime	unsigned short	等待进入感应区时间(ms)	.0 表示无需等待, 无卡直接返回; 0xFFFF 表示一直等待
出参	Response	ActiveActionResponse *		激活成功应答数据单元
返回值		short		0[失败], 1[成功], 3[超时]...
注:				

4.4.26 非接触式存储卡认证扇区

函数原型: `short __stdcall ZtDevice_IC_ConnectlessCertifySector(BYTE Sector, BYTE KeyType, BYTE* Key);`

函数名	ZtDevice_IC_ConnectlessCertifySector			
属性	参数名	参数类型	参数含义	备注
入参	Sector	BYTE	扇区号	
	KeyType	BYTE	密钥类型	
	Key	BYTE*	密钥数据 6 字节 密钥数据	
出参				无
返回值		short		0[失败], 1[成功], 3[超时]...



注:

4.4.27 非接触式存储卡读块

函数原型: `short __stdcall ZtDevice_IC_ConnectlessReadBlock(BYTE BlockNo, unsigned short *Len, BYTE* Response);`

函数名	ZtDevice_IC_ConnectlessReadBlock			
属性	参数名	参数类型	参数含义	备注
入参	BlockNo	BYTE	块号	
	Len	unsigned short *	为 Response 分配的空间大长度	
出参	Response	BYTE*	应答数据	
返回值		short		0[失败], 1[成功], 3[超时]...
注:				

4.4.28 非接触式存储卡写块

函数原型: `short __stdcall ZtDevice_IC_ConnectlessWriteBlock(BYTE BlockNo, BYTE* Data);`

函数名	ZtDevice_IC_ConnectlessWriteBlock			
属性	参数名	参数类型	参数含义	备注
入参	BlockNo	BYTE	块号	
	Data	BYTE*	待写块数据 16 字节数据	
出参				无
返回值		short		0[失败], 1[成功], 3[超时]...
注:				

4.4.29 非接触式存储卡读值块

函数原型: `short __stdcall ZtDevice_IC_ConnectlessReadValueBlock(BYTE BlockNo, float *fValue);`

函数名	ZtDevice_IC_ConnectlessReadValueBlock			
属性	参数名	参数类型	参数含义	备注
入参	BlockNo	BYTE	块号	
出参	fValue	float *	块中的数值	
返回值		short		0[失败], 1[成功], 3[超时]...
注:				



4.4.30 非接触式存储卡写值块

函数原型: `short __stdcall ZtDevice_IC_ConnectlessWriteValueBlock(BYTE BlockNo, float fValue);`

函数名	ZtDevice_IC_ConnectlessWriteValueBlock			
属性	参数名	参数类型	参数含义	备注
入参	BlockNo	BYTE	块号	
	fValue	float	待写入的数值	
出参				无
返回值		short		0[失败], 1[成功], 3[超时]...
注:				

4.4.31 非接触式存储卡加值

函数原型: `short __stdcall ZtDevice_IC_ConnectlessIncrease(BYTE BlockNo, float fValue);`

函数名	ZtDevice_IC_ConnectlessIncrease			
属性	参数名	参数类型	参数含义	备注
入参	BlockNo	BYTE	块号	
	fValue	float	待加数值	
出参				无
返回值		short		0[失败], 1[成功], 3[超时]...
注:				

4.4.32 非接触式存储卡减值

函数原型: `short __stdcall ZtDevice_IC_ConnectlessDecrease(BYTE BlockNo, float fValue);`

函数名	ZtDevice_IC_ConnectlessDecrease			
属性	参数名	参数类型	参数含义	备注
入参	BlockNo	BYTE	块号	
	fValue	float	待减数值	
出参				无
返回值		short		0[失败], 1[成功], 3[超时]...
注:				

4.4.33 非接触式存储卡写主密钥

函数原型: `short __stdcall ZtDevice_IC_ConnectlessWriteKey(BYTE Mode, BYTE SectorNo, BYTE* Key);`

函数名	ZtDevice_IC_ConnectlessWriteKey			
-----	---------------------------------	--	--	--



属性	参数名	参数类型	参数含义	备注
入参	Mode	BYTE	密钥模式	
	SectorNo	BYTE	扇区号	
	Key	BYTE*	密钥数据 6 字节 密钥数据	
出参				无
返回值		short		0[失败], 1[成功], 3[超时]...
注:				

4.4.34 设置 IC 卡序列号

函数原型: `short __stdcall ZtDevice_IC_SetSerialNo(BYTE *bSn, int iLen);`

函数名	ZtDevice_IC_SetSerialNo			
属性	参数名	参数类型	参数含义	备注
入参	bSn	BYTE *	序列号 asc 16 字节	
	iLen	int	序列号长度	
出参				无
返回值		short		0[失败], 1[成功], 3[超时]...
注:				

4.4.35 读取 IC 卡序列号

函数原型: `short __stdcall ZtDevice_IC_GetSerialNo(BYTE *bSn, int *iLen);`

函数名	ZtDevice_IC_GetSerialNo			
属性	参数名	参数类型	参数含义	备注
入参				无
出参	bSn	BYTE *	序列号 asc 16 字节	
	iLen	int *	序列号长度	
返回值		short		0[失败], 1[成功], 3[超时]...
注:				