

动态库说明

Model No.	CRT-580 读卡器
Date	2008 /09/22
Ver.	v20080922
Page	1/58

CRT-580 读写器动态库接口函数说明

目 录

1.	CRT 读卡器动态库文件说明	2
2.	动态库版本函数	2
3.	串口操作基本函数	3
4.	CRT-580 专用函数	5
5.	设置卡机串口通讯波特率	9
6.	IC 卡操作公用函数	10
7.	Mafare 1 函数	12
8.	CPU 卡函数 (T=0/T=1)	20
9.	SIM 卡函数 (T=0/T=1)	24
10.	. SLE4428 函数	27
11.	. SLE4442 函数	31
12.	. 24Cxx 系列卡函数	34
13.	. AT88SC102 函数	36
14.	. AT88SC1604 函数	42
14.	. AT88SC1608 函数	48
15.	. AT45D041 函数	53
16.	. 磁卡函数	55



动态库说明

Model No.	CRT-580 读写器
Date	200 8/09/ 22
Ver.	v20080922
Page	2/58

文件名称: CRT_580.dll 内部版本: v20080922

功能简述:针对 CRT 读卡器系列产品开发,完成磁卡、射频卡和常规的几种接触式 IC 卡、CPU 卡的读写操作。

支持卡类型:

● 飞利浦 M1 射频卡

- SLE4428、SLE4442
- 24C01、24C02、24C04、24C08、24C16、24C32、24C64
- AT88SC102、AT88SC1604、AT45D041、AT88SC1608
- 接触式 CPU 卡 (T=0/T=1)
- SAM/SIM 卡 (T=0/T=1)
- 磁卡

协议中未作特殊说明的错误状态返回请参考下表:

错误字E	描叙
0x01	未定义的命令参数
0x02	命令不能执行
0x04	命令数据错误
0x05	输入电源电压不在卡机工作范围内
0x06	读卡器内有非标准长度的异常长度卡 (短卡或长卡)

获取内部版本信息。

1. int APIENTRY GetSysVerion(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, char *strVerion);

//参数:

// ComHandle: 串口句柄。

// strVerion: 版本信息字符串。

//返回:

// 如果函数调用成功,返回值为0。

// 如果函数调用失败,返回值不为0。



动态库说明

Model No.	CRT-580 读写器
Date	200 8/09/ 22
Ver.	v20080922
Page	3/58

打开串口

1. HANDLE APIENTRY CommOpen(char *Port);

//参数:

//Port:串口号字符串

//例如:CommOpen("Com1");

//返回:

//串口文件句柄

//备注:必须先调用此函数,获得指定串口的串口文件句柄,才可调用其他函数。

// 可以同时打开多个串口,获得多个串口文件句柄,但不能多次打开同一个串口。

// 使用完毕后,必须调用 CommClose()关闭串口。

按指定的波特率打开串口 (该函数完成的功能= CommOpen 函数+ CommSetting 函数)

2. HANDLE APIENTRY CommOpenWithBaut(char *Port, unsigned int _data);

//参数:

//Port:串口号字符串

//_data 指定波特率

// 波特率=9600, 19200, 38400, 57600。

//例如:CommOpen("Com1",9600);

//返回:

//串口文件句柄

//备注:必须先调用此函数,获得指定串口的串口文件句柄,才可调用其他函数。

// 可以同时打开多个串口,获得多个串口文件句柄,但不能多次打开同一个串口。

// 使用完毕后,必须调用 CommClose()关闭串口。

关闭指定串口

3. int APIENTRY CommClose(HANDLE ComHandle);

//参数:

// ComHandle: 串口句柄。

//返回:

//0=成功

//备注:与 CommOpen()函数配套使用,并且在使用串口完毕后必须调用此函数关闭串口。

设置上位机串口波特率。

4. int APIENTRY CommSetting(HANDLE ComHandle,char *ComSeting)

//参数:

// ComHandle: 串口句柄。

// ComSeting: 设置信息字符串。格式 "9600,n,8,1"



动态库说明

Model No.	CRT-580 读写器
Date	200 8/09/ 22
Ver.	v20080922
Page	4/58

// 波特率=9600, 19200, 38400, 57600。

	-
_// i 万 lr	٦ŀ٠

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。



动态库说明

Model No.	CRT-580 读写器
Date	2008/09/22
Ver.	v20080922
Page	5/58

CRT580 复位读卡机。

1. int APIENTRY CRT580_Reset(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl);

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址,低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

//返回值:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。
- 注: (1) 该命令即使在掉电保护等特殊情况下也可被执行;
 - (2) 本命令和以下各命令如执行失败就按非正常返回方式

从读卡机读取状态信息。

2. int APIENTRY CRT580_GetStatus(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, BYTE *_SS5,BYTE *_SS4, BYTE *_SS3,BYTE *_SS2,BYTE *_SS1,BYTE *_SS0)

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

// _SS5, _SS4, _SS3, _SS2, _SS1: 状态信息

	状态	状态说明
_ SS 5=0x30	停卡位为前端不持卡	读卡器进卡后的停卡位为读卡器卡口不持卡位
_ SS 5=0x31	停卡位为前端持卡	读卡器进卡后的停卡位为读卡器卡口持卡位
_ SS 5=0x32	停卡位为射频卡(卡机内)操作位	读卡器进卡后的停卡位为射频卡(卡机内)操作位
_ SS 5=0x33	停卡位为 IC 卡操作位	读卡器进卡后的停卡位为 IC 卡操作位
_ SS 4=0x31	读卡器卡口禁止进卡	读卡器不允许从卡口处进卡
_ SS 4=0x32	读卡器卡口允许磁卡进卡	读卡器只允许磁卡从卡口处进卡
_ SS 4=0x33	读卡器卡口允许 IC 卡进卡	读卡器允许 IC 卡,磁卡从卡口处进卡
_ SS 4=0x34	读卡器卡口允许磁信号进卡	允许磁信号进卡



动态库说明

Model No.	CRT-580 读写器
Date	2008/09/22
Ver.	v20080922
Page	6/58

_ SS 3=0x30	读卡器内无卡	读卡器中无卡,可以进行发卡到读卡器
_SS 3=0x31	读卡器内有卡(射频卡操作位)	读卡器内有卡 (射频卡操作位)
_ SS 3=0x32	读卡器 IC 卡操作位有卡	读卡器 IC 卡操作位有卡
_ SS 3=0x33	读卡器卡口持卡位有卡	读卡器卡口持卡位置有卡
_ SS 3=0x34	读卡器卡口不持卡位有卡	读卡器卡口不持卡位置有卡
_ SS 3=0x35	读卡器有长卡	读卡器内读卡器有长卡
_ SS 3=0x36	读卡器有短卡	读卡器内读卡器有短卡
_ SS 3=0x37	后端持卡位置有卡	读卡器内后端持卡位置有卡
_ SS 3=0x38	后端不持卡位置有卡	后端不持卡位置有卡
_ SS 2=0x30	发卡通道无卡	发卡通道预发卡位置,发卡通道扩展操作位置都无卡时为发卡通道无卡
_ SS 2=0x31	发卡通道预发卡位置有卡	
_ SS 2=0x32	发卡通道扩展操作位置有卡	
_ SS 2=0x33	发卡通道非正常位置有卡	卡在错误的位置上
_ SS 1=0x30	发卡箱内卡足	发卡箱中有足够的卡,允许进行发卡
_ SS 1=0x31	发卡箱内卡少	发卡箱中的卡不多,应及时给卡箱加卡
_ SS 1=0x32	发卡箱内无卡	发卡箱中无卡,卡已发完
_ SS 0=0x30	收卡箱卡不满	允许将卡收到收卡箱
_ SS 0=0x31	收卡箱卡已满	收卡箱中卡已收满, 应及时将收卡箱中卡清空

//返回值:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。

读感应器状态

3. int APIENTRY CRT580_SensorStatus(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl,BYTE *_PH1,BYTE *_PH2,BYTE *_PH3,BYTE *_PH4,BYTE *_PH5,BYTE *_PH6,BYTE *_KSW,BYTE *_CTSW, BYTE *_PSS1,BYTE *_PSS2, BYTE *_PSS3,BYTE *_PSS4,BYTE *_PSS5)

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

// 检测各传感器状态。状态信息包为 13 byte., 状态字节为 0x31,表示检测有卡, 为 0x30 表示无卡。

_PH1 发卡通道 1#光感状态



动态库说明

Model No.	CRT-580 读写器
Date	2008/09/22
Ver.	v20080922
Page	7/58

PH2 发卡通道 2#光感状态

PH3 发卡通道 3#光感状态

_PH4 发卡箱卡空传感器状态

PH5 发卡箱卡少传感器状态

PH6 收卡箱卡满传感器状态

_KSW 读卡器卡口传感器状态

CTSW 闸门状态

PSS1 读卡器 1#光感状态

PSS2 读卡器 2#光感状态

_PSS3 读卡器 3#光感状态

PSS4 读卡器 4#光感状态

_PSS5 读卡器 5#光感状态

注:

_CTSW: 闸门状态信息 _CTSW=0X30 表示闸门已关闭,

_CTSW=0X31 表示闸门已打开。

_KSW: 开关进卡传感受器状态 _KSW=0X30 表示开关没有检测到卡片插入闸门,

_KSW=0X31 表示开关检测到有卡片插入闸门。

_PSS1—PSS5: 红外传感器状态: _PSS1 (2…5) =0X30 表示此传感器位置上未探测到卡片;

_PSS1 (2···5) =0X31 表示探测到有卡片。

_PH1—PH6: 红外传感器状态: __PH1 (2···6) =0X30 表示此传感器位置上未探测到卡片;

PH1 (2···6) =0X31 表示探测到有卡片。

//返回值:

// 如果函数调用成功,返回值为0。

// 如果函数调用失败,返回值不为0。

卡机进卡使能控制操作

4. int APIENTRY CRT580_CardSetting(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl,BYTE _CardIn, BYTE _StopPosition)

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

// _CardIn: 前端进卡方式

=0x31 禁止前端进卡

=0x32 磁卡方式(磁信号+开关同时有效)进卡使能,只允许磁卡从前端开闸门进卡



动态库说明

Model No.	CRT-580 读写器
Date	2008/09/22
Ver.	v20080922
Page	8/58

=0x33 开关方式进卡使能,允许磁卡,IC卡,M1射频卡,双界面卡从前端开闸门进卡

// _EnableBackIn: 设置从读卡器前端进卡后的停卡位置

=0x30 停卡位为读卡器卡口不持卡位

=0x31 停卡位为读卡器卡口持卡位

=0x32 停卡位为射频卡(卡机内)操作位

=0x33 停卡位为 IC 卡操作位

//返回值:

// 如果函数调用成功,返回值为0。

// 如果函数调用失败,返回值不为0。

注: CRT580 上电复位是禁止读卡器卡口进卡,设定的进卡后的停卡位为射频卡操作位(卡机内)。

卡机卡走位操作

5. int APIENTRY CRT580_MoveCard(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, BYTE _ToPosition, BYTE _FromPosition)

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

// _ToPosition:

=0x30 发卡到发卡通道预操作位置

=0x31 发卡到射频卡操作(卡机内)位置

=0x32 发卡到 IC 卡操作位置

=0x33 发卡到卡口持卡位置

=0x34 发卡到卡口不持卡位置

=0x35 收卡到回收箱//返回值:

// _FromPosition:

=0x30 将发卡箱的卡,发到发卡通道中扩展操作位置。

=0x31 将读卡器的卡,发到发卡通道中扩展操作位置。

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。



动态库说明

Model No.	CRT-580 读写器
Date	200 8/09/ 22
Ver.	v20080922
Page	9/58

设置卡机串口通讯波特率

1. int APIENTRY CRT580_SetBaudRate(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, unsigned int _data)

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址,低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'.

例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

// _data: 波特率=9600, 19200, 38400。57600

//返回:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。

注意:成功设置后,需要关闭上位机串口,然后用对应的波特率重新打开上微机串口,否则不能通讯。



动态库说明

Model No.	CRT-580 读写器
Date	2008/09/22
Ver.	v20080922
Page	10/58

自动测试读卡机中的卡片类型。

2. int APIENTRY CRT_IC_DetectCard(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl,BYTE *_CardType,BYTE *_CardInfor)

//参数:

// ComHandle: 串口句柄。

// _AddrH: 卡机地址的高字节地址, 低字节地址。

// _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'.

例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

// CardType: 卡片类型。 // CardInfor: 卡片信息。

CardType	CardInfor	说明				
'N'	'0'	卡机内无卡				
	'1'	未知卡类型				
'0'	'0'	卡为非接触式射频卡				
'1'	'0'	卡为 T=0 接触式 CPU 卡				
	'1'	卡为 T=1 接触式 CPU 卡				
' 2'	'0'	卡为 24C01 卡				
	'1'	卡为 24C02 卡				
	'2'	卡为 24C04 卡				
	'3'	卡为 24008 卡				
	'4'	卡为 24C16 卡				
	' 5'	卡为 24C32 卡				
	'6'	卡为 24C64 卡				
' 3'	'0'	卡为 SL4442 卡				
	'1'	卡为 SL4428 卡				
' 4'	'0'	卡为 AT88S102 卡				
	'1'	卡为 AT88S1604 卡				
	'2'	卡为 AT45D041 卡				
	'3'	卡为 AT88S1608 卡				

//返回值:

// 如果函数调用成功,返回值为0。

// 如果函数调用失败,返回值不为0。

应用程序在操作接触式 IC 卡之前,需调用此函数给卡片上电。

3. int APIENTRY CRT_IC_CardOpen(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl);



动态库说明

Model No.	CRT-580 读写器
Date	200 8/09/ 22
Ver.	v20080922
Page	11/58

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

//返回值:

- // 如果函数调用成功,返回值为0,
- // 如果函数调用失败,返回值不为0。

应用程序在操作完接触式IC后,在用户拔出卡片之前,需调用此函数以给卡片下电。

4. int APIENTRY CRT_IC_CloseCard(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl);

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

//返回值:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。



动态库说明

Model No.	CRT-580 读写器
Date	2008/09/22
Ver.	v20080922
Page	12/58

检测是否 M1 卡片。

int APIENTRY RF_DetectCard(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl);

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址,低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'.

例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

//返回值:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。
 - = 'N' (0X4E) 寻卡不成功
 - = 'E' (0X45) 卡机内无卡
 - = 'W'(0X57) 卡不在允许操作的位置上。

获取 M1 卡片的序列号。

2. int APIENTRY RF_GetCardID(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, BYTE _CardID[4]);

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

// _CardID[4]:M1 卡片号码。

//返回值:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。
 - = 'N' (0X4E) 获取卡序列号失败,并返回空序列号(0X00,0X00,0X00,0X00)
 - = 'E' (0X45) 卡机内无卡

验证扇区的密码。

int APIENTRY RF_LoadSecKey(HANDLE ComHandle,BYTE _AddrH, BYTE _AddrH, BYTE _Sec, BYTE _KEYType, BYTE _KEY[6]);

//参数:

- // ComHandle: 串口旬柄。
- // _Sec: 扇区号。
- // _KEYType: 密码类型选择。0=KEYA; 1=KEYB



动态库说明

Model No.	CRT-580 读写器
Date	2008/09/22
Ver.	v20080922
Page	13/58

// _KEY: 密码,6个字节。

//返回值

// 如果函数调用成功,返回值为0。

// 如果函数调用失败,返回值不为0。

= '0' (0X30) 寻不到射频卡

= '3' (0X33) 密码错误

= 'E' (0X45) 卡机内无卡

= 'W'(0X57) 卡不在允许操作的位置上。

注 1: 扇区号= 0x00 0x01 0x020x28

S50 卡片扇区号是 0x00 0x01 0x020x0F, S70 卡片扇区号是 0x00 0x01 0x020x28

更改扇区的密码。

4. int APIENTRY RF_ChangeSecKey(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, BYTE _Sec, BYTE _KEY[6]);

//参数:

// ComHandle: 串口句柄。

// _AddrH: 卡机地址的高字节地址, 低字节地址。

// _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

// _Sec: 扇区号。

// _KEY: 密码。

//返回值 P:

// 如果函数调用成功,返回值为0。

// 如果函数调用失败,返回值不为0。

= '0' (0X30) 寻不到射频卡

= '3' (0X33) 密码错误

= 'E' (0X45) 卡机内无卡

= 'W'(0X57) 卡不在允许操作的位置上。

注 1: 扇区号= 0x00 0x01 0x020x28

S50 卡片扇区号是 0x00 0x01 0x020x0F, S70 卡片扇区号是 0x00 0x01 0x020x28

注 2: 执行该命令只能对 KEYA 的密码更改操作,并对 KEYB 密码的改写成: "0xFF, 0xFF, 0xF

读扇区指定块的数据。

5. int APIENTRY RF_ReadBlock(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, BYTE _Sec, BYTE _Block, BYTE _BlockData[16]);

//参数:

// ComHandle: 串口句柄。



11011	
ıH	

Model No.	CRT-580 读写器
Date	2008/09/22
Ver.	v20080922
Page	14/58

动态库说明

- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'.

例:卡机地址为'15',则 Addrh=0x31, addrl=0x35

- // _Sec: 扇区号。
- // _Block: 块号。
- // _BlockData: 数据块内容。

//返回值 P:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。
 - = '0 '(0X30) 寻不到 RF 卡
 - = '1'(0X31) 操作扇区号错(不是验证密码后的扇区)
 - = '2' (0X32) 操作的卡序列号错
 - = '3' (0X33) 密码验证错
 - = '4' (0X34) 读数据错
 - = 'E' (0X45) 卡机内无卡
 - = 'W' (0X57) 卡不在允许操作
 - 注: 扇区号= 0x00 0x01 0x020x28

S50 卡片扇区号是 0x00 0x01 0x020x0F, S70 卡片扇区号是 0x00 0x01 0x020x28

块号= 0x00 0x01 0x020x0F

S50 卡片每个扇区有 4 个地块, 块号分别是 0x00 0x01 0x02 0x03,

S70 卡片第 0-31 扇区中每一扇区有 4 个块, 块号分别是 0x00 0x01 0x02 0x03,

第 32-39 扇区每一扇区有 16 个块, 块号分别是 0x00 0x01 0x02......0x0F

写扇区指定块的数据。

6. int APIENTRY RF_WriteBlock(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, BYTE _Sec, BYTE _Block, BYTE _BlockData[16]);

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

- // _Sec: 扇区号。
- // _Block: 块号。
- // _BlockData: 数据块内容。

//返回值 P:

// 如果函数调用成功,返回值为0。



动态库说明

Model No.	CRT-580 读写器
Date	2008/09/22
Ver.	v20080922
Page	15/58

// 如果函数调用失败,返回值不为0。

= '0 '(0X30) 寻不到 RF 卡

= '1'(0X31) 操作扇区号错(不是验证密码后的扇区)

= '2' (0X32) 操作的卡序列号错

= '3' (0X33) 密码验证错

= '4' (0X34) 校验写入块数据错

= 'E' (0X45) 卡机内无卡

= 'W'(0X57) 卡不在允许操作的位置上。

注: 扇区号= 0x00 0x01 0x020x28

S50 卡片扇区号是 0x00 0x01 0x020x0F, S70 卡片扇区号是 0x00 0x01 0x020x28

块号= 0x00 0x01 0x020x0F

S50 卡片每个扇区有 4 个地块, 块号分别是 0x00 0x01 0x02 0x03,

S70 卡片第 0-31 扇区中每一扇区有 4 个块, 块号分别是 0x00 0x01 0x02 0x03,

第 32-39 扇区每一扇区有 16 个块, 块号分别是 0x00 0x01 0x02......0x0F

初始化值。

7. int APIENTRY RF_InitValue(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, BYTE _Sec, BYTE _Block, BYTE _Data[4]);

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

- // _Sec: 扇区号。
- // _Block: 块号。
- // _Data: 初始值。

//返回值 P:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。
 - = '0 '(0X30) 寻不到 RF 卡
 - = '1'(0X31) 操作扇区号错(不是验证密码后的扇区)
 - = '2' (0X32) 操作的卡序列号错
 - = '3' (0X33) 密码验证错
 - = '4'(0X34) 块数据格式错误(该块存贮数据没有写成值数据形式)
 - = '5' (0X35) 增值溢出
 - = 'E' (0X45) 卡机内无卡
 - = 'W'(0X57) 卡不在允许操作的位置上。



动态库说明

Model No.	CRT-580 读写器
Date	2008/09/22
Ver.	v20080922
Page	16/58

//注意:在进行'读值/增值/减值操作前,首先要使用该函数进行指定扇区内数据块的初始化,否则会产生错误!

初始化值:按 MIFARE 卡块数据格式进行写入 16 byter 数据,其格式如下:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Value		/Value	e			Value				Adr	/Adr	Adr	/Adr		

Value: 要初始化 4 byte 值(初始化函数初始值),注值的低字节在前,高字节在后

/Value: 要初始化 4 byte 值取反

Adr: 所要初始化值的块地址: Adr= 扇区号 X 4+ 块号

/Adr: 所要初始化值的块地址的取反

例:将第五扇区块 0 初始化值为 10,所要写入 16 byte 扇区块数据为:

"0x0A, 0x00, 0x00, 0x00, 0xF5, 0xFF,0xFF,0xFF,0xFF,0x0A, 0x00, 0x00, 0x00, 0x14, 0xEB, 0x14, 0xEB"

int P:

BYTE value[4];

memset(value,0,4);

value[0]=0xA;

P=RF_InitValue(hCom,5,0,value);

注: 在进行值操作时, S50, S70 第 0-31 扇区中每个扇区的第 3 块, S70 第 32-39 扇区中第 15 块是 KEYA、控制字、KEYB 的存储区

域,是不能作值段数据存贮。初始化值,增值,减值,读值时应注意操作扇区块地址范围。

读值。

8. int APIENTRY RF_ReadValue(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, BYTE _Sec, BYTE _Block, BYTE _Data[4]);

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址,低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'.

例: 卡机地址为 '15', 则 Addrh=0x31, addr1=0x35

- // _Sec: 扇区号。
- // _Block: 块号。
- // _Data: 初始值。

//返回值 P:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。
 - = '0 '(0X30) 寻不到 RF 卡
 - = '1'(0X31) 操作扇区号错(不是验证密码后的扇区)
 - = '2' (0X32) 操作的卡序列号错
 - = '3' (0X33) 密码验证错
 - = '4'(0X34) 块数据格式错误(该块存贮数据没有写成值数据形式)



动态库说明

Model No.	CRT-580 读写器
Date	2008/09/22
Ver.	v20080922
Page	17/58

= '5' (0X35) 增值溢出

= 'E' (0X45) 卡机内无卡

= 'W'(0X57) 卡不在允许操作的位置上。

注 1: 在进行读值操作前,首先要使用初始化函数 RF InitValue,否则会产生错误!

注 2: 扇区号= 0x00 0x01 0x020x28

S50 卡片扇区号是 0x00 0x01 0x020x0F, S70 卡片扇区号是 0x00 0x01 0x020x28

块号= 0x00 0x01 0x020x0F

S50 卡片每个扇区有 4 个地块, 块号分别是 0x00 0x01 0x02 0x03,

S70 卡片第 0-31 扇区中每一扇区有 4 个块, 块号分别是 0x00 0x01 0x02 0x03,

第 32-39 扇区每一扇区有 16 个块, 块号分别是 0x00 0x01 0x02......0x0F

注 3: 在进行值操作时, S50, S70 第 0-31 扇区中每个扇区的第 3 块, S70 第 32-39 扇区中第 15 块是 KEYA、控制字、KEYB 的存储 区域,是不能作值段数据存贮。初始化值,增值,减值,读值时应注意操作扇区块地址范围。

增值。

9. int APIENTRY RF_Increment(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, BYTE _Sec, BYTE _Block, BYTE _Data[4]);

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址,低字节地址。
- // _Addrl: 卡机地址的高字节地址,低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

- // _Sec: 扇区号。
- // _Block: 块号。
- // _Data: 数值。

//返回值 P:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。
 - = '0 '(0X30) 寻不到 RF 卡
 - = '1'(0X31) 操作扇区号错(不是验证密码后的扇区)
 - = '2' (0X32) 操作的卡序列号错
 - = '3' (0X33) 密码验证错
 - = '4'(0X34) 块数据格式错误(该块存贮数据没有写成值数据形式)
 - = '5' (0X35) 增值溢出
 - = 'E' (0X45) 卡机内无卡
 - = 'W'(0X57) 卡不在允许操作的位置上。
- 注 1: 在进行读值操作前,首先要使用初始化函数 RF_InitValue, 否则会产生错误!
- 注 2: 扇区号= 0x00 0x01 0x020x28



动态库说明

Model No.	CRT-580 读写器
Date	2008/09/22
Ver.	v20080922
Page	18/58

S50 卡片扇区号是 0x00 0x01 0x020x0F, S70 卡片扇区号是 0x00 0x01 0x020x28

块号= 0x00 0x01 0x020x0F

S50 卡片每个扇区有 4 个地块, 块号分别是 0x00 0x01 0x02 0x03,

S70 卡片第 0-31 扇区中每一扇区有 4 个块, 块号分别是 0x00 0x01 0x02 0x03,

第 32-39 扇区每一扇区有 16 个块, 块号分别是 0x00 0x01 0x02......0x0F

注 3: 在进行值操作时, S50, S70 第 0-31 扇区中每个扇区的第 3 块, S70 第 32-39 扇区中第 15 块是 KEYA、控制字、KEYB 的存储 区域,是不能作值段数据存贮。初始化值,增值,减值,读值时应注意操作扇区块地址范围。

减值。

10. int APIENTRY RF_Decrement(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, BYTE _Sec, BYTE _Block, BYTE _Data[4]);

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

- // _Sec: 扇区号。
- // _Block: 块号。
- // _Data: 数值。

//返回值 P:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。
 - = '0 '(0X30) 寻不到 RF 卡
 - = '1'(0X31) 操作扇区号错(不是验证密码后的扇区)
 - = '2' (0X32) 操作的卡序列号错
 - = '3' (0X33) 密码验证错
 - = '4'(0X34) 块数据格式错误(该块存贮数据没有写成值数据形式)
 - = '5' (0X35) 增值溢出
 - = 'E' (0X45) 卡机内无卡
 - = 'W'(0X57) 卡不在允许操作的位置上。

注 1: 在进行读值操作前,首先要使用初始化函数 RF_InitValue,否则会产生错误!

注 2: 扇区号= 0x00 0x01 0x020x28

S50 卡片扇区号是 0x00 0x01 0x020x0F, S70 卡片扇区号是 0x00 0x01 0x020x28

块号= 0x00 0x01 0x020x0F

S50 卡片每个扇区有 4 个地块, 块号分别是 0x00 0x01 0x02 0x03,

S70 卡片第 0-31 扇区中每一扇区有 4 个块, 块号分别是 0x00 0x01 0x02 0x03,

第 32-39 扇区每一扇区有 16 个块, 块号分别是 0x00 0x01 0x02......0x0F



动态库说明

Model No.	CRT-580 读写器
Date	200 8/09/ 22
Ver.	v20080922
Page	19/58

注 3: 在进行值操作时, S50, S70 第 0-31 扇区中每个扇区的第 3 块, S70 第 32-39 扇区中第 15 块是 KEYA、控制字、KEYB 的存储区域,是不能作值段数据存贮。初始化值,增值,减值,读值时应注意操作扇区块地址范围。



Model No.

Date

Ver.

CRT-580 读写器 2008/09/22 v20080922

动态库说明

Page 20/58

CPU 卡复位。

int APIENTRY CPU_Reset(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, BYTE *_CPUType,BYTE _exData[], int
 exdataLen):

```
//参数:
   ComHandle: 串口句柄。
   _AddrH: 卡机地址的高字节地址, 低字节地址。
//
   _Addrl: 卡机地址的高字节地址,低字节地址。
   Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'.
    例: 卡机地址为 '15', 则 Addrh=0x31, addr1=0x35
  _CPUType: 返回的 CPU 卡类型
        =0x0
             T=0
        =0x0
             T=1
   _exData: 返回数据。
   _exdataLen: 返回数据的长度。
//返回值:
   如果函数调用成功,返回值为0。
   如果函数调用失败,返回值不为0。
   = 'N' (0X4E) 操作不成功
   = 'E' (0X45) 卡机无卡
   = 'W'(0X57) 卡不在允许操作的位置上。
   举例:
   int rc;
   BYTE exData[300]; //预设长度为 300, 实际返回的有效长度依&1enASK 中的值为准。
   int lenASK;
   BYTE _CPUType;
   BYTE _Addh, _Add1;
   Addh=0x31;
   Add1=0x35;
   rc=CPU_Reset(hCom, _Addh, _Addl, &_CPUType, exData, &lenASK);
   //复位成功即 rc=0
   //则
   //lenASK 保存返回数据长度
   //exData 数组保存复位后返回的数据
   CString str, t;
   for (int i=0; i < lenASK; i++)
       t. Format ("%02x ", exData[i]);
```



动态库说明

Model No.	CRT-580 读写器
Date	2008/09/22
Ver.	v20080922
Page	21/58

```
str += t;
t="";
}

t. Format("CPU 卡类型为 T%d \n:",_CPUType);
m_list1. AddString (t);
t. Format("冷复位时返回数据长度为%d \n 如下所示:",lenASK);
m_list1. AddString (t);
m_list1. AddString (str); //字符串 str 中的数据即为实际返回的 16 进制字符串数据
m_list1. AddString ("\n");
```

CPU 卡热复位。

2. int APIENTRY CPU_WarmReset(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, BYTE *_CPUType,BYTE _exData[], int * exdataLen):

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

// _CPUType: 返回的 CPU 卡类型

=0x0 T=0

=0x0 T=1

- // _exData: 返回数据。
- // _exdataLen: 返回数据的长度。

//返回值:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。
 - = 'N' (0X4E) 操作不成功
 - = 'E' (0X45) 卡机无卡
 - = 'W'(0X57) 卡不在允许操作的位置上。

举例:请参考 CPU 复位命令函数

CPU(T=0)卡 C-APDU 命令。

3. int APIENTRY CPU_T0_C_APDU(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, int _dataLen, BYTE _APDUData[], BYTE _exData[], int *_exdataLen);;

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址,低字节地址。



Date Ver.

动态库说明

```
Model No.
               CRT-580 读写器
                  2008/09/22
                  v20080922
                    22/58
  Page
```

```
_Addrl: 卡机地址的高字节地址, 低字节地址。
    Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'.
    例: 卡机地址为'15',则 Addrh=0x31, addrl=0x35
    _dataLen: C-APDU 命令长度。
//
    _APDUData: CUP 卡 C-APDU 命令。
   _exData: 返回数据。
    _exdataLen: 返回数据的长度。
//
//返回值:
    如果函数调用成功,返回值为0。
    如果函数调用失败,返回值不为0。
    = 'N' (0X4E) 操作不成功
    = 'E' (0X45)
                卡机无卡
    = 'W' (0X57)
                卡不在允许操作的位置上。
    举例:
        BYTE DataBlock[5];
                         //存放 C-APDU 命令,该命令长度为 5 个字节。
        BYTE exData[300]; //用来存放执行上面的 C-APDU 命令后返回的数据,
                         //执行上面的 C-APDU 命令后实际返回的数据长度
      int lenASK;
        DataBlock[0]=0x00;
        DataBlock[1]=0x84;
        DataBlock[2]=0x00;
        DataBlock[3]=0x00;
        DataBlock[4]=0x08;
        rc=CPU_C_APDU(hCom,5,DataBlock,exData,&lenASK);
        if(rc==1) AfxMessageBox("No Find Card!");
        else if(rc!=0) AfxMessageBox("ERR!");
            else
                 {
                    CString str;
                    for(int i=0; i<lenASK; i++)
                         {
                            t.Format("%02x",exData[i]);
                            str += t;
                            t="";
                         }
```

//字符串 str 中的数据即为实际返回的 16 进制字符串数据



动态库说明

Model No.	CRT-580 读写器
Date	200 8/09/ 22
Ver.	v20080922
Page	23/58

CPU(T=1)卡 C-APDU 命令。

4. int APIENTRY CPU_T1_C_APDU(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, int _dataLen, BYTE _APDUData[], BYTE _exData[], int *_exdataLen);

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址,低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。

例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

- // _dataLen: C-APDU 命令长度。
- // _APDUData: CUP 卡 C-APDU 命令。
- // _exData: 返回数据。
- // _exdataLen: 返回数据的长度。

//返回值:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。
 - = 'N' (0X4E) 操作不成功
 - = 'E' (0X45) 卡机无卡
 - = 'W'(0X57) 卡不在允许操作的位置上。

举例: 请参考 CPU (T=0) 中 C-APDU 命令函数



动态库说明

Model No.	CRT-580 读写器
Date	2008/09/22
Ver.	v20080922
Page	24/58

SIM 卡复位。

1. int APIENTRY SIM_Reset(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl,BYTE _VOLTAGE,BYTE _SIMNo, BYTE *_SIMTYPE,BYTE _exData[], int *_exdataLen);

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

- // _VOLTAGE
 - = 0x2E 对工作电压是 1.8 V 的 SIM 卡进行复位操作
 - = 0x2F 对工作电压是 3.0 V 的 SIM 卡进行复位操作
 - = 0x30 对工作电压是 5.0 V 的 SIM 卡进行复位操作
- // SIMNo: SIM 卡座号码。
- SIM 卡座号=0x30 操作 SIM 卡 1
 - =0x31 操作 SIM 卡 2
 - =0x32 操作 SIM 卡 3
 - =0x33 操作 SIM 卡 4
 - =0x34 操作 SIM 卡 5
 - =0x35 操作 SIM 卡 6
 - =0x36 操作 SIM 卡 7
 - =0x37 操作 SIM 卡 8
- // _SIMTYPE: 返回的 SIM 卡类型
 - =0x0 T=0
 - =0x0 T=1
- // _exData: 返回数据。
- // _exdataLen: 返回数据的长度。

//返回值:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。

2. SIM 卡(T=0)C-APDU 命令。

int APIENTRY SIM_T0_C_APDU(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, BYTE SIMNo, int _dataLen, BYTE _APDUData[], BYTE _exData[], int *_exdataLen);

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址,低字节地址。



Model No.	CRT-580 读写器
Date	2008/09/22
Ver.	v20080922
Page	25/58

// _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

// SIMNo: SIM 卡座号码。

SIM 卡座号=0x30 操作 SIM 卡 1

=0x31 操作 SIM 卡 2

=0x32 操作 SIM 卡 3

=0x33 操作 SIM 卡 4

=0x34 操作 SIM 卡 5

=0x35 操作 SIM 卡 6

=0x36 操作 SIM 卡 7

=0x37 操作 SIM 卡 8

// _dataLen: C-APDU 命令长度。

// _APDUData: CUP 卡 C-APDU 命令。

// _exData: 返回数据。

// _exdataLen: 返回数据的长度。

//返回值:

// 如果函数调用成功,返回值为0。

// 如果函数调用失败,返回值不为0。

SIM 卡(T=1)C-APDU 命令。

3. int APIENTRY SIM_T1_C_APDU(HANDLE ComHandle,BYTE _AddrH, BYTE _AddrH, BYTE SIMNo, int _dataLen, BYTE _APDUData[], BYTE _exData[], int *_exdataLen);)

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址,低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

// SIMNo: SIM 卡座号码。

SIM 卡座号=0x30 操作 SIM 卡 1

=0x31 操作 SIM 卡 2

=0x32 操作 SIM 卡 3

=0x33 操作 SIM 卡 4

=0x34 操作 SIM 卡 5

=0x35 操作 SIM 卡 6

=0x36 操作 SIM 卡 7

=0x37 操作 SIM 卡 8



SP	FC	IFI	$C\Delta$	T	\cap	N
\mathbf{o}	ر حانا	шт	$\mathbf{C}_{\mathbf{\Gamma}}$	711	w	ΤN

动态库说明

Model No.	CRT-580 读写器
Date	200 8/09/ 22
Ver.	v20080922
Page	26/58

// _dataLen: C-APDU 命令长度。

// _APDUData: CUP \dagger C-APDU 命令。

// _exData: 返回数据。

// _exdataLen: 返回数据的长度。

//返回值:

// 如果函数调用成功,返回值为0。

// 如果函数调用失败,返回值不为0。



动态库说明

Model No.	CRT-580 读写器
Date	200 8/09/ 22
Ver.	v20080922
Page	27/58

SEL4428 复位。

1. int APIENTRY SLE4428_Reset(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl)

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址,低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'.

例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

//返回值:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。
 - = 'E' (0X45) 卡机无卡
 - = 'W'(0X57) 卡不在允许操作的位置上。

此函数从卡片内的_Address 地址起读取_dataLen 字节长的数据,并存储在_BlockData 指针内返回给应用程序。

2. int APIENTRY SLE4428_Read(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, int _Address, BYTE _dataLen, BYTE _BlockData[])

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

- // _ Address
- // 指定要读取的数据在卡片存储区内的起始地址。
- // _ dataLen
- // 指定要读取的数据长度。SLE4428 存储区的长度为 1024 字节。
- // _ BlockData
- // 字符型指针,指向从卡片存储区内读到的数据并返回给应用程序。

//返回值:

- // 如果函数调用成功,返回值为 0,且_bReadData 的内容为从卡片读取到的数据。
- // 如果函数调用失败,返回值不为0。
 - = 'E' (0X45) 卡机无卡
 - = 'W'(0X57) 卡不在允许操作的位置上。

此函数将_BlockData 指向的数据写入到卡片内_Address 地址,共_dataLen 个字节。为了保证写入的正确性,应用程序应确保_BlockData 指针指向的内容及长度。



动态库说明

Model No.	CRT-580 读写器
Date	2008/09/22
Ver.	v20080922
Page	28/58

3. int APIENTRY SLE4428_Write(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, int _Address, BYTE _dataLen, BYTE _BlockData[])

//参数:

- // ComHandle: 串口句柄。
- // AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

- // _ Address
- // 指定卡片存储区内被重写的起始地址。
- // _ dataLen
- // 指定要写入的数据长度。SLE4428 存储区的长度为 1024 字节。
- // BlockData
- // 字符型指针。应用程序在调用此函数前,将要写入到卡片内的数据存于_BlockData 指针内,该内容会覆盖卡片内_Address 地址起的_dataLen 个字节。

//返回值:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。
 - = 'E' (0X45) 卡机无卡
 - = 'W'(0X57) 卡不在允许操作的位置上。

此函数读取 SLE4428 卡片的保护区内容。0x01 表示相应数据区的内容未被保护,只要密码校验正确即可改写; 0x00 表示相应数据区的内容已被保护,不能再被改写。

5. int APIENTRY SLE4428_ReadP(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, int _Address, BYTE _dataLen, BYTE _BlockData[])

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

- // _ Address
- // 指定要读取的保护区的起始地址。
- // _ dataLen
- // 指定要读取的保护区的长度。SLE4428 保护区的长度为 1024 字节。
- // _ BlockData
- // 字符型指针,指向从卡片内读到的保护区内容并返回给应用程序。

//返回值:



动态库说明

Model No.	CRT-580 读写器
Date	2008/09/22
Ver.	v20080922
Page	29/58

- // 如果函数调用成功,返回值为 0,且_bPReadData 的内容为从卡片读取到的字符串。
- // 如果函数调用失败,返回值不为0。
 - = 'E' (0X45) 卡机无卡
 - = 'W'(0X57) 卡不在允许操作的位置上。

此函数将_BlockData 指向的字符串写入到卡片内_Address 地址,共_dataLen 个字节,同时写相应的保护位。

6. int APIENTRY SLE4428_WriteP(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, int _Address, BYTE _dataLen, BYTE _BlockData[])

//参数:

- // ComHandle: 串口旬柄。
- // AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

- // _ Address
- // 指定卡片存储区内被重写的起始地址。
- // _ dataLen
- // 指定要写入的数据长度。SLE4428 存储区受保护字节范围为字节地址 0 至 1023, 共 1024 字节。
- // _ BlockData
- // 字符型指针。应用程序在调用此函数前,将要写入到卡片内的数据存于_ BlockData 指针内,该串的内容会覆盖卡片内_ Address 地址起的_ dataLen 个字节。

//返回值:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。
 - = 'E' (0X45) 卡机无卡
 - = 'W'(0X57) 卡不在允许操作的位置上。

应用程序在改写 SLE4428 卡片的存储区之前,需验证密码。只有正确验证密码之后,对存储区的写入才有效。

$7. \quad int\ APIENTRY\ SLE4428_VerifyPWD(HANDLE\ ComHandle, BYTE\ _AddrH,\ BYTE\ _AddrH,\ BYTE\ _PWData[2])$

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

// _ PWData: 密码字节。

//返回值:

// 如果函数调用成功,返回值为0。



动态库说明

Model No.	CRT-580 读写器
Date	200 8/09/ 22
Ver.	v20080922
Page	30/58

- // 如果函数调用失败,返回值不为0。
 - = 'E' (0X45) 卡机无卡
 - = 'W'(0X57) 卡不在允许操作的位置上。

此函数改写 SLE4428 卡片的密码。

8. int APIENTRY SLE4428_WritePWD(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, BYTE _PWDataOld[2],BYTE _PWDataNew[2])

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'.

例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

- // PWDataOld: 旧密码字节。
- // PWDataNew: 新密码字节。

//返回值:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。
 - = 'E' (0X45) 卡机无卡
 - = 'W'(0X57) 卡不在允许操作的位置上。



动态库说明

CRT-580 读写器
2008/09/22
v20080922
31/58

SEL4442 复位。

1. int APIENTRY SLE4442_Reset(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl);

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'.

例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

//返回值:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。

SEL4442 读数据。

2. int APIENTRY SLE4442_Read(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, BYTE _Address, BYTE _dataLen, BYTE _BlockData[]);

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址,低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

- // _Address: 地址。
- // _dataLen: 数据长度。
- // _BlockData: 数据。

//返回值:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。

SEL4442 读保护数据。

3. int APIENTRY SLE4442_ReadP(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, BYTE _BlockData[32]);

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址,低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

// _BlockData: 数据。



动态库说明

Model No.	CRT-580 读写器
Date	2008/09/22
Ver.	v20080922
Page	32/58

//返回值:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。

SEL4442 读安全数据。

4. int APIENTRY SLE4442_ReadS(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, BYTE _BlockData[4]);

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

// _BlockData: 数据。

//返回值:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。

SEL4442 验证密码。

5. int APIENTRY SLE4442 VerifyPWD(HANDLE ComHandle,BYTE _AddrH, BYTE _AddrH, BYTE _PWData[3]);

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

// _PWData: 密码。

//返回值:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。

SEL4442 写数据。

6. int APIENTRY SLE4442_Write(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, BYTE _Address, BYTE _dataLen, BYTE _BlockData[]);

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址,低字节地址。
 Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。



动态库说明

Model No.	CRT-580 读写器
Date	200 8/09/ 22
Ver.	v20080922
Page	33/58

例:卡机地址为'15',则 Addrh=0x31, addrl=0x35

- // _Address: 地址。
- // _dataLen: 数据长度。
- // _BlockData: 数据。

//返回值:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。

SEL4442 写保护数据。

7. int APIENTRY SLE4442_WriteP(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, BYTE _BlockData[32]);

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址,低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

// _BlockData: 数据。

//返回值:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。

SEL4442 更改密码。

8. int APIENTRY SLE4442_WritePWD(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, BYTE _PWData[3]);

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

// _PWData: 密码。

//返回值:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。



动态库说明

Model No.	CRT-580 读写器
Date	200 8/09/ 22
Ver.	v20080922
Page	34/58

检测 24Cxx 卡类型。

1. int APIENTRY IC_DetectCard(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl,BYTE *_CardType);

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'.

例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

// _CardType: 卡类型。

=0 设置卡为 24C01 128BYTE ADR=0x0000—0x007F

=1 设置卡为 24C02 256BYTE ADR=0x0000—0x00FF

=2 设置卡为 24C04 512BYTE ADR=0x0000—0x01FF

=3 设置卡为 24C08 1K BYTE ADR=0x0000—0x03FF

=4 设置卡为 24C16 2K BYTE ADR=0x0000—0x07FF

=5 设置卡为 24C32 4K BYTE ADR=0x0000—0x0FFF

=6 设置卡为 24C64 8K BYTE ADR=0x0000—0x1FFF

//返回值:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。

P= 'N' (0X4E) 失败

P= 'E' (0X45) 卡机无卡

P = W'(0X57) 卡不在允许操作的位置上。

读指定地址数据。

int APIENTRY IC_ReadBlock(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, BYTE _CardType, int _Address, BYTE _dataLen, BYTE _BlockData[]);

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址,低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

- // _Address: 起始地址。
- // _dataLen: 数据长度。
- // _ BlockData: 数据。

//返回值:

// 如果函数调用成功,返回值为0。



动态库说明

Model No.	CRT-580 读写器
Date	2008/09/22
Ver.	v20080922
Page	35/58

// 如果函数调用失败,返回值不为0。

写指定地址数据。

3. int APIENTRY IC_WriteBlock(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, BYTE _CardType, int _Address, BYTE _dataLen, BYTE _BlockData[]);

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址,低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

- // _Address: 起始地址。
- // _dataLen: 数据长度。
- // _ BlockData: 数据。

//返回值:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。



动态库说明

Model No.	CRT-580 读写器
Date	2008/09/22
Ver.	v20080922
Page	36/58

复位。

1. int APIENTRY AT88SC102_Reset(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl);

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址,低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

//返回值:P

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。

P='N'(0X4E) 失败

P= 'E' (0X45) 卡机无卡

P = W'(0X57) 卡不在允许操作的位置上。

验证主密码。

int APIENTRY AT88SC102_VerifyPWD(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, BYTE _PWData[2]);

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

// _ PWData: 主密码。

//返回值 **P**:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。

P= 'N' (0X4E) 失败

P= 'E' (0X45) 卡机无卡

P='F'(0X46) 卡已报废(密码验证失败超过允许次数后卡锁死报废)

P='W'(0X57) 卡不在允许操作的位置上。

在安全级别1模式下验证主密码后所有单元均可读出。

在安全级别2模式下验证主密码成功除密码存贮单元读不出外,其余单元均可读出。

更改密码。

int APIENTRY AT88SC102_WritePWD(HANDLE ComHandle,BYTE _AddrH, BYTE _AddrH, BYTE _PWIndex, BYTE _PWData[]);



动态库说明

Model No.	CRT-580 读写器
Date	2008/09/22
Ver.	v20080922
Page	37/58

//参数:

- // ComHandle: 串口旬柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

- // _PWIndex: 密码类型选择。0=主密码; 1=擦除密码一; 2=擦除密码二
- // _ PWData: 密码。
- 区号 =0 修改控制区密码 密码数据为 2 byte
 - =1 修改应用区一密码 密码数据为 6 byte
 - =2 修改应用区二密码 密码数据为 4 byte

//返回值 P:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。
 - P= 'N' (0X4E) 失败
 - P= 'E' (0X45) 卡机无卡
 - P='W'(0X57) 卡不在允许操作的位置上。

修改密码只能在安全模式1下验证主密码后才能修改,进入安全模式2后不能再修改所有密码,只能验证。

读取存储区指定地址的数据。

4. int APIENTRY AT88SC102_Read(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, BYTE _Index, BYTE _Address, BYTE _dataLen, BYTE _BlockData[]);

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

- // _Index:数据区选择。0=控制区;1=应用区一;2=应用区二
- // _Address: 起始地址。
- // _ dataLen:数据长度。
- // _ BlockData: 数据。

//返回值 P:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。
 - P='N'(0X4E) 失败
 - P= 'E' (0X45) 卡机无卡



动态库说明

Model No.	CRT-580 读写器
Date	2008/09/22
Ver.	v20080922
Page	38/58

P='W'(0X57) 卡不在允许操作的位置上。

安全模式1下擦除存贮区。

5. int APIENTRY AT88SC102_Security1Clear(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, BYTE _Index, BYTE _Address, BYTE _dataLen);

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址,低字节地址。 Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。

例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

- // _Index:数据区选择。0=控制区;1=应用区一;2=应用区二
- // _ Address: 起始地址。
- // _ dataLen: 数据长度。

//返回值 P:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。
 - P='N'(0X4E) 失败
 - P= 'E' (0X45) 卡机无卡
 - P='W'(0X57) 卡不在允许操作的位置上。



动态库说明

Model No.	CRT-580 读写器
Date	200 8/09/ 22
Ver.	v20080922
Page	39/58

安全模式2下擦除应用区一。

6. int APIENTRY AT88SC102_Security2ClearApp1(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl,BYTE _Security2App1PW[6]);

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址,低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

// _ Security2App1PW:擦除密码。

//返回值 P:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。

P= 'N' (0X4E) 失败

P= 'E' (0X45) 卡机无卡

P='W'(0X57) 卡不在允许操作的位置上。

安全模式2下擦除应用区二。

7. int APIENTRY AT88SC102_Security2ClearApp2(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl,BYTE _EC2,BYTE _Security2App2PW[4]);

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

// EC2: 擦除熔丝状态操作字

EC2 = 0 擦除熔丝未熔断应用区二擦除

EC2 = 1 擦除熔丝已熔断应用区二擦除

// _Security2App2PW:擦除密码。

//返回值 P:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。

P='N'(0X4E) 失败

P= 'E' (0X45) 卡机无卡

P= 'F' (0X46) 二区已报废,只能读不能再进行擦写。

P='W'(0X57) 卡不在允许操作的位置上。



动态库说明

Model No.	CRT-580 读写器
Date	2008/09/22
Ver.	v20080922
Page	40/58

写指定地址的数据。

8. int APIENTRY AT88SC102_Write(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, BYTE _Index, BYTE _Address, BYTE _dataLen, BYTE _BlockData[]);

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

- // _Index:数据区选择。0=控制区;1=应用区一;2=应用区二
- // _ Address: 起始地址。
- // _ dataLen: 数据长度。
- // _ BlockData:数据。

//返回值 P:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。
 - P= 'N' (0X4E) 失败
 - P= 'E' (0X45) 卡机无卡
 - P='W'(0X57) 卡不在允许操作的位置上。

个人化操作, 使卡进入安全级别模式 2。

9. int APIENTRY AT88SC102_InitSecurity2(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, BYTE _CtrlMode);

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

// _CtrlMode: 操作模式。

_CtrlMode =0x30 使卡模拟进入安全级别模式 2,可供测试,

_CtrlMode =0x31 使模拟进入安全级别模式 2 的卡恢复到安全级别模式 1。

_CtrlMode =0x32 使卡完全进入安全级别模式 2 ,一旦将卡操作成安全模式 2,

将无法再恢复到安模式1。

//返回值 P:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。

P='N'(0X4E) 失败

P= 'E' (0X45) 卡机无卡



动态库说明

Model No.	CRT-580 读写器
Date	2008/09/22
Ver.	v20080922
Page	41/58

P = W'(0X57) 卡不在允许操作的位置上。

进入安全模式 2 前一定设定好应用区一,二的密码,应用区一第一字节(0x16)应用区二第一字节(0x5C) 不能轻易修改, 是控制这些区单元的读写使能。

进入安全模式 2 后要写这些应用区,卡进行擦除操作时是对这些应用区整块擦除,应注意写入新数据前应先读出保存,以防数据丢失。同时这些应用区受熔丝计数器的控制。使熔丝计数器有效则写入 128 次后不能再写入。使其无效后则写入次数为卡的最大有效操作数(100,000次).

二区擦除计数器操作字 EC2 设置成无效操作。

10. int APIENTRY AT88SC102_DisableEC2(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl);

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

//返回值 P:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。
 - P= 'N' (0X4E) 失败
 - P= 'E' (0X45) 卡机无卡
 - P = W'(0X57) 卡不在允许操作的位置上。

要使卡在安全模式2下应用区二擦除次数不受限则要在进入个人化操作前执行此操作。

否则在卡设置完成模式 2 后,卡默认应用区二在模式 2 下擦除次数受限有效(只能擦除 128 次)。再要取消应用二区擦写不受限,则无法取消二区擦除受限次数(128 次)。同样设置成卡在模式 2 下擦写次数不受限后不能再设置成擦写受限。同时用户也要对 EC2 操作状态保存,卡在模式 2 下应用时要擦写应用区二时(验证应用区二的擦除密码)应注意相应的参数。



动态库说明

Model No.	CRT-580 读写器
Date	2008/09/22
Ver.	v20080922
Page	42/58

复位。

1. int APIENTRY AT88SC1604_Reset(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl)

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

//返回值 P:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。

P='N'(0X4E) 失败

P= 'E' (0X45) 卡机无卡

P='W'(0X57) 卡不在允许操作的位置上。

验证密码

2. int APIENTRY AT88SC1604_VerifyPWD(HANDLE ComHandle,BYTE _AddrH, BYTE _AddrH, BYTE _PWIndex,BYTE _PWData[])

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

// PWIndex:密码类型索引。

其中密码类型号: = 0 验证主密码

- =1 验证应用一区密码
- = 2 验证应用一区擦除密码
- = 3 验证应用二区密码
- = 4 验证应用二区擦除密码
- = 5 验证应用三区密码
- = 6 验证应用三区擦除密码
- = 7 验证应用四区密码
- = 8 验证应用四区擦除密码
- // _ PWData: 密码。

//返回值 P:

// 如果函数调用成功,返回值为0。



动态库说明

Model No.	CRT-580 读写器
Date	200 8/09/ 22
Ver.	v20080922
Page	43/58

// 如果函数调用失败,返回值不为0。

P='N'(0X4E) 失败

P='E'(0X45) 卡机无卡

P='W'(0X57) 卡不在允许操作的位置上。

P= 'F' (0X46) 卡已报废或应用块报废



动态库说明

Model No.	CRT-580 读写器
Date	2008/09/22
Ver.	v20080922
Page	44/58

模式1下修改密码

3. int APIENTRY AT88SC1604_WritePWD(HANDLE ComHandle,BYTE _AddrH, BYTE _AddrH, BYTE _PWIndex,BYTE _PWData[])

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

// PWIndex: 密码类型选择。

其中密码类型号: = 0 验证主密码

- =1 验证应用一区密码
- = 2 验证应用一区擦除密码
- = 3 验证应用二区密码
- = 4 验证应用二区擦除密码
- = 5 验证应用三区密码
- = 6 验证应用三区擦除密码
- = 7 验证应用四区密码
- = 8 验证应用四区擦除密码

// _ PWData: 密码。

//返回值 P:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。

P='N'(0X4E) 失败

P= 'E' (0X45) 卡机无卡

P='W'(0X57) 卡不在允许操作的位置上。

读指定地址的数据。

4. int APIENTRY AT88SC1604_Read(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, BYTE _Index, int _Address, BYTE _dataLen, BYTE _BlockData[])

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址,低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

// Index:数据区选择。



动态库说明

Model No.	CRT-580 读写器
Date	2008/09/22
Ver.	v20080922
Page	45/58

```
区号: =0 一区(0x020 --- 0x21A)

=1 二区(0x21B --- 0x420)

=2 三区(0x421 ---- 0x621)

=3 四区(0x622 ---- 0x7F5)
```

= 4 其它区(除一,二,三,区以外的区域)

// _Address: 起始地址。 // _dataLen: 数据长度。 // _BlockData: 返回的数据

//返回值 P:

// 如果函数调用成功,返回值为0。

// _BlockData: 返回的数据

// 如果函数调用失败,返回值不为0。

P='N'(0X4E) 失败

P= 'E' (0X45) 卡机无卡

P='W'(0X57) 卡不在允许操作的位置上。

写指定地址的数据。

5. int APIENTRY AT88SC1604_Write(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, BYTE _Index, int _Address, BYTE _dataLen, BYTE _BlockData[])

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

// _Index:数据区选择。

// _ Address: 起始地址。

// _ dataLen: 数据长度。

// _BlockData: 返回的数据

//返回值 P:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。

P='N'(0X4E) 失败



动态库说明

Model No.	CRT-580 读写器
Date	2008/09/22
Ver.	v20080922
Page	46/58

P= 'E' (0X45) 卡机无卡

P='W'(0X57) 卡不在允许操作的位置上。

擦除指定地址的数据。

6. int APIENTRY AT88SC1604_Clear(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, BYTE _Index, int _Address, BYTE _dataLen)

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

// _Index: 数据区选择。

- // _Address: 起始地址。
- // _ dataLen: 数据长度。

//返回值 P:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。

P='N'(0X4E) 失败

P= 'E' (0X45) 卡机无卡

P='W'(0X57) 卡不在允许操作的位置上。

个人化操作(卡进入安全模式2)。

7. int APIENTRY AT88SC1604_Personalization(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, BYTE _data);

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址,低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

- // _data: 操作号 = 0x30 软个人化操作(模拟个人化操作使卡进入安全模式 2,供测试)
 - = 0x31 退出软个人化操作
 - = 0x32 完全个人化操作不可再恢复



动态库说明

Model No.	CRT-580 读写器
Date	200 8/09/ 22
Ver.	v20080922
Page	47/58

//返回值 P:

// 如果函数调用成功,返回值为0。

// 如果函数调用失败,返回值不为0。

P= 'N' (0X4E) 失败

P='E'(0X45) 卡机无卡

P='W'(0X57) 卡不在允许操作的位置上。



动态库说明

Model No.	CRT-580 读写器
Date	2008/09/22
Ver.	v20080922
Page	48/58

//复位

1. int APIENTRY AT88SC1608_Reset(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl);

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址,低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'.

例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

//返回值 P:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。
 - = 'N' (0X4E) 失败
 - = 'E' (0X45) 卡机无卡
 - = 'W'(0X57) 卡不在允许操作的位置上。

//验证密码

int APIENTRY AT88SC1608_VerifyPWD(HANDLE ComHandle,BYTE _AddrH, BYTE _AddrH, BYTE _PWIndex, BYTE _PWData[3]);

//参数:

- // ComHandle: 串口句柄。
- // PWIndex:密码类型索引。
 - =0 验证应用一 区读密码
 - =1 验证应用二 区读密码
 - = 2 验证应用三 区读密码
 - = 3 验证应用四 区读密码
 - = 4 验证应用五 区读密码
 - = 5 验证应用六 区读密码
 - =6 验证应用七 区读密码
 - = 7 验证应用八 区读密码
 - = 8 验证应用一 区写密码
 - = 9 验证应用二 区写密码
 - =10 验证应用三 区写密码
 - =11 验证应用四 区写密码
 - =12 验证应用五 区写密码
 - = 13 验证应用六 区写密码
 - = 14 验证应用七 区写密码/验证主密码



动态库说明

Model No.	CRT-580 读写器
Date	2008/09/22
Ver.	v20080922
Page	49/58

= 15 验证应用八 区写密码

// _ PWData: 密码。

//返回值 P:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。
 - = 'N' (0X4E) 验证密码失败
 - = 'F'(0X46) 卡已报废或应用块报废
 - = 'E' (0X45) 卡机无卡
 - = 'W'(0X57) 卡不在允许操作的位置上。

注意:每一个区密码只有允许验证 8 次, 8 次校验错语后卡锁死,就表明卡这个块区不能读或写。

//读数据

3. int APIENTRY AT88SC1608_Read(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, BYTE _Index, BYTE _Address, BYTE _dataLen, BYTE _BlockData[]);

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

- // _Index: 区号。
 - = 0 应用一区 (len=0x01—0x80)
 - = 1 应用二区 (len=0x01—0x80)
 - = 2 应用三区 (len=0x01—0x80)
 - = 3 应用四区 (len=0x01—0x80)
 - = 4 应用五区 (len=0x01—0x80)
 - = 6 应用七区 (len=0x01—0x80)

应用六区 (len=0x01—0x80)

- = 7 应用八区 (len=0x01—0x80)
- = 8 设置区 (len=0x01—0x80)
- // _Address: 起始地址。操作地址范围: 0x00----0xFF(除设置区外)
- // dataLen: 数据长度。操作长度范围: 0x01----0xFF(除设置区外)
- // BlockData: 数据。

要对应用区进行读时请校验该区读密码正确后才能进行读,否则读的数据无效,设置区数据只有密码区域(0x40---0x7F)是 受密码保护,只能校验正确后才能正确读出

//返回值 P:

// 如果函数调用成功,返回值为0。

= 5

// 如果函数调用失败,返回值不为0。



动态库说明

Model No.	CRT-580 读写器
Date	2008/09/22
Ver.	v20080922
Page	50/58

= 'N' (0X4E) 失败

= 'E' (0X45) 卡机无卡

= 'W'(0X57) 卡不在允许操作的位置上。

//写数据

4. int APIENTRY AT88SC1608_Write(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, BYTE _Index, BYTE _Address, BYTE _dataLen, BYTE _BlockData[]);

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

- // _Index: 区号。
 - = 0 应用一区 (len=0x01—0x80)
 - = 1 应用二区 (len=0x01—0x80)
 - = 2 应用三区 (len=0x01—0x80)
 - = 3 应用四区 (len=0x01—0x80)
 - = 4 应用五区 (len=0x01—0x80)
 - = 5 应用六区 (len=0x01—0x80)
 - = 6 应用七区 (len=0x01—0x80)
 - = 7 应用八区 (len=0x01—0x80)
 - = 8 设置区 (len=0x01—0x80)
 - // _Address: 起始地址。操作地址范围: 0x00----0xFF(除设置区外)
- // dataLen: 数据长度。操作长度范围: 0x01----0xFF(除设置区外)
- // BlockData: 数据。

要对应用区进行写时请校验该区读密码正确后才能进行写,否则写的数据无效,设置区的密码,厂家的固化的信息,用卡商固化信息,访问权限控制区,认证区,密钥区受相关条件才能进行写操作,只有符合写操作才能进行写。请使用时详细参阅相关资料。

//返回值 P:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。
 - = 'N' (0X4E) 失败
 - = 'E' (0X45) 卡机无卡
 - = 'W'(0X57) 卡不在允许操作的位置上。

//读熔丝状态

5. int APIENTRY AT88SC1608_GetFUSE(HANDLE ComHandle,BYTE _AddrH, BYTE _AddrH, BYTE * _PER, BYTE * _CMA,



动态库说明

Model No.	CRT-580 读写器
Date	200 8/09/ 22
Ver.	v20080922
Page	51/58

$BYTE * _FAB);$

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

- // PER: 厂商熔丝 (FAB =0X30 已熔断, FAB=0X31 末熔断)。
- // _ CMA: 卡商熔丝 (CMA =0X30 已熔断, CMA =0X31 末熔断)
- // _FAB: 发行商熔丝 (PER =0X30 已熔断, PER =0X31 末熔断)

//返回值 P:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。
 - = 'N' (0X4E) 失败
 - = 'E' (0X45) 卡机无卡
 - = 'W'(0X57) 卡不在允许操作的位置上。

//设置熔丝状态

int APIENTRY AT88SC1608 SetFUSE(HANDLE ComHandle, BYTE AddrH, BYTE Addrl);

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

//返回值 **P**:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。
 - = 'N' (0X4E) 失败
 - = 'E' (0X45) 卡机无卡
 - = 'W'(0X57) 卡不在允许操作的位置上。

//初始化验证区

7. int APIENTRY AT88SC1608_InitAuth(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, BYTE _Q[8]);

//参数:

- // ComHandle: 串口旬柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址,低字节地址。



动态库说明

Model No.	CRT-580 读写器
Date	200 8/09/ 22
Ver.	v20080922
Page	52/58

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

// _ Q[8]: 8 byte 随机数 Q0、Q1、Q2、Q3、Q4、Q5、Q6、Q7

初始化认证区是先读取卡中的 Nc, Ci, 通过 F1 或 F2 算法, 计算出 Gc=F1(Ks, Nc) 得到的随机数 Q0~Q7, 送入 AT88SC1608 卡中, 完成进行初始化认证区。

//返回值 P:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。
 - = 'N' (0X4E) 失败
 - = 'E' (0X45) 卡机无卡
 - = 'W'(0X57) 卡不在允许操作的位置上。

//校验验证区

8. int APIENTRY AT88SC1608_VerifyAuth(HANDLE ComHandle,BYTE_AddrH, BYTE_Addrl, BYTE_Q[8]);

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

// _ **Q[8]**: 8 byte 随机数 Q0、Q1、Q2、Q3、Q4、Q5、Q6、Q7

校验认证区是在进行初始化认证区操作后,按 F2 算法完成 Q1=F2(Gc, Ci, Q0), 分别生成的 Q0, Q1, Q2, Q3, Q4, Q5, Q6, Q7 送入 AT88SC1608 卡中由卡中来完成校验认证区,进行此验证。

注: Nc: 识别码,通常用作卡的唯一标识——卡号。个人化前定义。

Ci: 密文,个人化前可写一随机数,认证卡时使用,每次认证会被自动改写。

Gc:密钥,64位的保密种子,由 Nc 通过 F1 公式推算出来,在个人化前,写入卡

中。个人化后不可访问,认证时作为该卡的 F2 公式的参数。(详细用法参见认证协议)

//返回值 P:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值不为0。
 - = 'N' (0X4E) 失败
 - = 'E' (0X45) 卡机无卡
 - = 'W'(0X57) 卡不在允许操作的位置上。



动态库说明

Model No.	CRT-580 读写器
Date	2008/09/22
Ver.	v20080922
Page	53/58

复位。

1. int APIENTRY AT45D041_Reset(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl)

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

//返回值:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值为-1。

读指定地址页的数据。

2. int APIENTRY AT45D041_Read(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, int __Address, BYTE _BlockData[]) //参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

- // _ Address: 页地址。页地址: 0x0000---0x07FF
- // _ BlockData: 读取的数据

//返回值 P:

- // 如果函数调用成功,返回值为 0,且_BlockData 的内容为从卡片读取到的字符串。
- // 如果函数调用失败,返回值不为0。
 - = 'E' (0X45) 卡机无卡
 - = 'W'(0X57) 卡不在允许操作的位置上。

写指定地址页的数据。

3. int APIENTRY AT45D041_Write(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, int __Address, BYTE _BlockData[]) //参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址,低字节地址。
- // _Addrl: 卡机地址的高字节地址,低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

// _ Address: 页地址。页地址: 0x0000---0x07FF



动态库说明

Model No.	CRT-580 读写器
Date	200 8/09/ 22
Ver.	v20080922
Page	54/58

// _ BlockData:写入的数据 _BlockData的内容为要写入卡片的字符串。

//返回值 P

- // 如果函数调用成功,返回值为0
- // 如果函数调用失败,返回值不为0。
 - = 'E' (0X45) 卡机无卡
 - = 'W'(0X57) 卡不在允许操作的位置上。



动态库说明

M - 1-1 M -	ODT 500 法信用
Model No.	CRT-580 读写器
Date	200 8/09/ 22
Ver.	v20080922
Page	55/58

遗磁勃数据

1. int APIENTRY MC_ReadTrack(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl, BYTE _mode, BYTE _track, BYTE _TrackData[], int *_TrackDataLen)

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

// _mode: 数据模式。

读卡模式: 0x30 以ASCII码读卡数据

0x31 以二进制码读卡数据

// _track: 磁轨。

指定轨道号: 0x30 磁卡三轨都不读

0x31 读磁卡一轨

0x32 读磁卡二轨

0x33 读磁卡三轨

0x34 读磁卡一二轨

0x35 读磁卡二三轨

0x36 读磁卡一三轨

0x37 读磁卡一二三轨

// __TrackData: 三个轨磁道数据内容(一轨数据包+二轨数据包+三轨数据包)。

其中每轨数据包格式如下:

轨数据起始字+读卡状态字+卡轨道数据

轨道起始字: 0x1F

读卡状态字: 0x59 读该轨数据读正确,卡轨道数据为该轨信息数据

0x4E 读卡不正确,卡轨道数据为错误信息

0x4F 该轨道不读,卡轨道数据为 0xE0;

错误信息:

0xE1 读该轨数据错误,没有起始位 STX

0xE2 读该轨数据错误,没有结束位 ETX

0xE3 读该轨数据错误,位校验错误 VRC

0xE4 读该轨数据错误,字节校验位错误 LRC

0xE5 读该轨数据错误,该轨是空白信息磁道

*二进制读卡传送的数据格式是:

一轨: b0,b1,b2,b3,b4,P

二轨,三轨: b0,b1,b2 b3,P



动态库说明

Model No.	CRT-580 读写器
Date	200 8/09/ 22
Ver.	v20080922
Page	56/58

*注意:

当设置以 ASCII 码读卡时将卡每一轨信息的别换成一个字节 ASCII 码上传达卡数据。

如: 一轨数据第一字节为: 0x03 (HEX)

上传数据时卡轨道数据包为: 0x33(ASCII)

当设置成二进制读卡时将卡每一轨信息的每一字节数据按每4位转成一个字节以ASCII 码形式上传数据。

如:一轨数据第一字节为: 0x03 (HEX)

上传数据时卡轨道数据包为: 0x30 0x3

// _TrackDataLen: 三个轨磁道数据内容(一轨数据包+二轨数据包+三轨数据包)的长度字节数

//返回值:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值为非0。

= 'E' (0x45) 卡机无卡

= 'W' (0x57) 卡不在允许操作的位置上

按指定方式重新上传指字轨道磁卡数据。

2. int APIENTRY MC_UploadTrack(HANDLE ComHandle,BYTE _AddrH, BYTE _AddrH, BYTE _mode, BYTE _track, BYTE _TrackData[], int *_TrackDataLen)

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

Addrh, addrl= '00', '01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15'。 例: 卡机地址为 '15', 则 Addrh=0x31, addrl=0x35

// _mode: 数据模式。

读卡模式: 0x30 以 ASCII 码读卡数据

0x31 以二进制码读卡数据

// _track: 磁轨。

指定轨道号: 0x30 磁卡三轨都不读

0x31 读磁卡一轨

0x32 读磁卡二轨

0x33 读磁卡三轨

0x34 读磁卡一二轨

0x35 读磁卡二三轨

0x36 读磁卡一三轨

0x37 读磁卡一二三轨

// _TrackData: 三个轨磁道数据内容(一轨数据包+二轨数据包+三轨数据包)。

其中每轨数据包格式如下:



动态库说明

Model No.	CRT-580 读写器
Date	2008/09/22
Ver.	v20080922
Page	57/58

轨数据起始字+读卡状态字+卡轨道数据

轨道起始字: 0x1F

读卡状态字: 0x59 读该轨数据读正确,卡轨道数据为该轨信息数据

0x4E 读卡不正确,卡轨道数据为错误信息

0x4F 该轨道不读,卡轨道数据为 0xE0;

错误信息:

0xE1 读该轨数据错误,没有起始位 STX

0xE2 读该轨数据错误,没有结束位 ETX

0xE3 读该轨数据错误,位校验错误 VRC

0xE4 读该轨数据错误,字节校验位错误 LRC

0xE5 读该轨数据错误,该轨是空白信息磁道

*二进制读卡传送的数据格式是:

一轨: b0,b1,b2,b3,b4,P

二轨,三轨: b0,b1,b2 b3,P

*注意:

当设置以 ASCII 码读卡时将卡每一轨信息的别换成一个字节 ASCII 码上传达卡数据。

如:一轨数据第一字节为: (

0x03 (HEX)

上传数据时卡轨道数据包为:

0x33 (ASCII)

当设置成二进制读卡时将卡每一轨信息的每一字节数据按每4位转成一个字节以ASCII 码形式上传数据。

如:一轨数据第一字节为:

0x03 (HEX)

上传数据时卡轨道数据包为:

0x30 0x3

// _TrackDataLen: 三个轨磁道数据内容(一轨数据包+二轨数据包+三轨数据包)的长度字节数

//返回值:

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值为非0。

= 'E' (0x45) 卡机无卡

= 'W' (0x57) 卡不在允许操作的位置上

清除磁卡数据。

3. int APIENTRY MC_ClearTrack(HANDLE ComHandle,BYTE _AddrH, BYTE _Addrl)

//参数:

- // ComHandle: 串口句柄。
- // _AddrH: 卡机地址的高字节地址, 低字节地址。
- // _Addrl: 卡机地址的高字节地址, 低字节地址。

例: 卡机地址为'15',则 Addrh=0x31, addr1=0x35

//返回值:



动态库说明

Model No.	CRT-580 读写器
Date	200 8/09/ 22
Ver.	v20080922
Page	58/58

- // 如果函数调用成功,返回值为0。
- // 如果函数调用失败,返回值为非0。