



Advanced Card Systems Ltd.
Card & Reader Technologies

ACR83 PINeasy



应用程序编程接口 V1.06



目录

1.0.	简介	3
2.0.	特性	4
3.0.	设备控制	5
3.1.	操作流程（PC/SC 2.0 第 10 部分）.....	5
3.2.	具体的 ScardControl 函数.....	6
3.3.	智能卡设备 IOCTL.....	6
3.3.1.	CM_IOCTL_GET_FEATURE_REQUEST.....	6
3.3.2.	FEATURE_VERIFY_PIN_DIRECT.....	7
3.3.3.	FEATURE_MODIFY_PIN_DIRECT.....	8
3.3.4.	FEATURE_IFD_PIN_PROP.....	10
3.3.5.	IOCTL_SMARTCARD_GET_FIRMWARE_VERSION.....	11
3.3.6.	IOCTL_SMARTCARD_DISPLAY_LCD_MESSAGE.....	12
3.3.7.	IOCTL_SMARTCARD_READ_KEY.....	12
	附录 A.设置 bKeyReturnCondition	14
	附录 B.响应错误代码	15
	附录 C.bmFormatStrings 说明	16
	附录 D.bmPINBlockString 说明	17
	附录 E.bmPINLength 格式	18

图目录

图 1	: 操作流程 图.....	5
------------	----------------------	----------



1.0. 简介

ACR83 PINeasy 是一款具有键盘和显示屏的 USB 接口读写器。它支持安全 PIN 码输入 (SPE) 功能, 用户可以在它自带的键盘上安全输入 PIN 码 (个人识别号码)。持卡人验证在设备内进行, 由于 PIN 码被输入到安全的 ACR83 PINeasy 内, 而非易受攻击的电脑或工作站中, 从而避免了中间人的攻击。

本 API 文档介绍了如何执行 PC/SC APDU 命令来对设备外设进行控制。有关接触式卡片的操作, 请参阅相关卡片的文档以及 PC/SC 标准。



2.0. 特性

- 14 键键盘
- 2 行 x 16 个字符的点阵式 LCD，每个字符 5x8 点
- 支持具有以下特性的 ISO-7816 CPU 卡：
 - A 类、B 类和 C 类（5 V、3 V、1.8 V）
 - T=0 和/或 T=1 协议
- 支持安全 PIN 码输入（SPE）
- 通过 EMV Level 1 认证
- USB 全速接口（12 Mbps）
- 符合下列标准：
 - PC/SC
 - WHQL
 - CCID
 - CE/FCC
 - RoHS

3.0. 设备控制

本节将对系统智能卡设备的 IOCTL 进行介绍。

3.1. 操作流程 (PC/SC 2.0 第 10 部分)

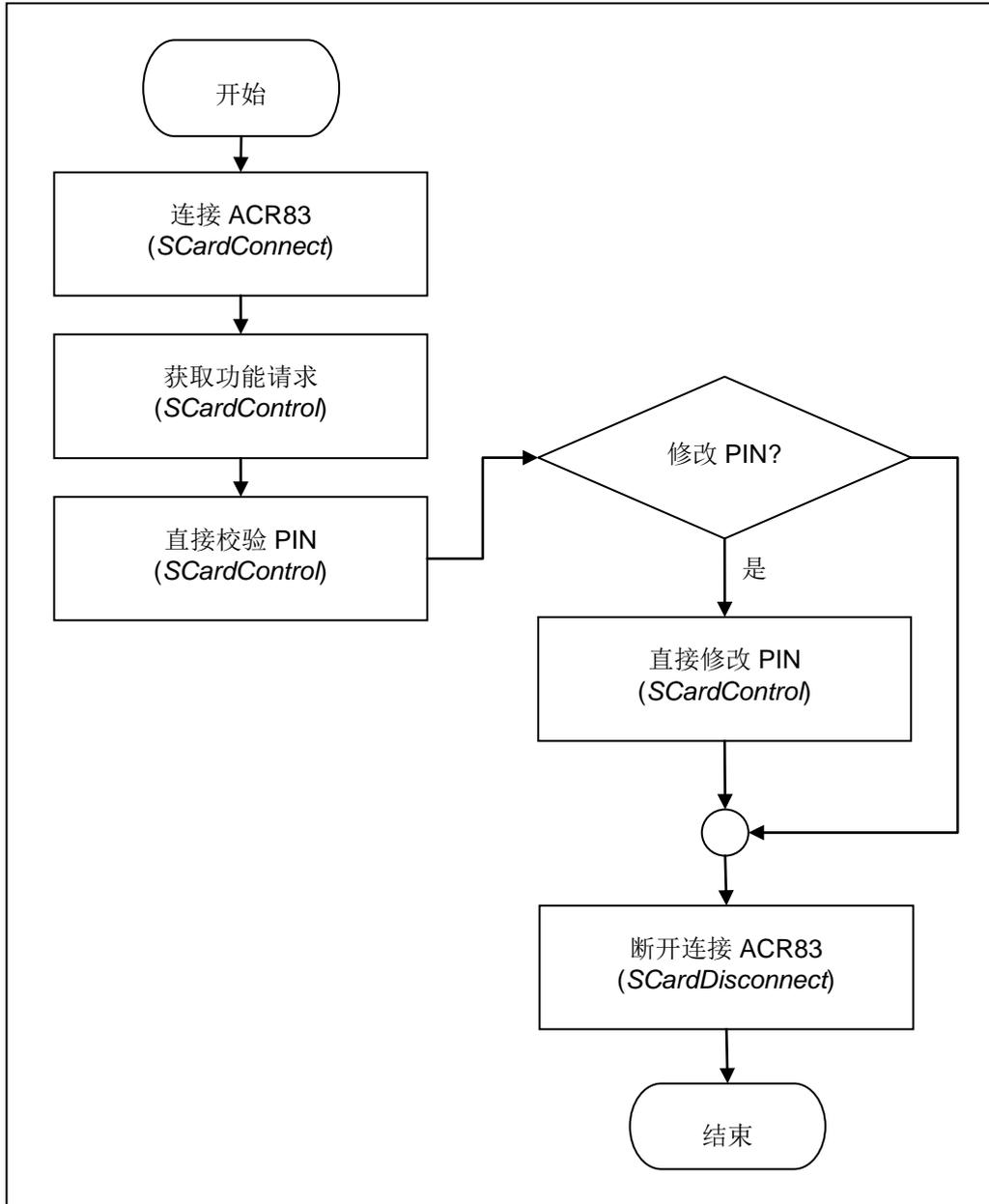


图1 : 操作流程图

为了使用 PIN 码校验和修改功能，*SCardControl* API 在调用时必须带有“获取功能请求”控制码，此 API 会返回读写器支持的功能列表。

ACR83 仅支持直接校验 PIN、直接修改 PIN 以及 IFD PIN 属性。要使用这些功能，您可以从此列表中获得相应的控制码。如需了解更多信息，请参考 PC/SC 2.0 规范第 10 部分。



3.2. 具体的ScardControl函数

```
LONG SCardControl(
    SCARDHANDLE hCard,
    DWORD dwControlCode,
    LPCVOID lpInBuffer,
    DWORD nInBufferSize,
    LPVOID lpOutBuffer,
    DWORD nOutBufferSize,
    LPDWORD lpBytesReturned
);
#define IOCTL_SMARTCARD_GET_FIRMWARE_VERSION SCARD_CTL_CODE(2078)
#define IOCTL_SMARTCARD_DISPLAY_LCD_MESSAGE SCARD_CTL_CODE(2079)
#define IOCTL_SMARTCARD_READ_KEY SCARD_CTL_CODE(2080)
// PC/SC 2.0 Part 10
#define CM_IOCTL_GET_FEATURE_REQUEST SCARD_CTL_CODE(3400)
```

注：数据以小端格式存储，最低有效字节（LSB）在前面。另外，必须在源代码中对 SCardControl 命令予以声明。

3.3. 智能卡设备IOCTL

3.3.1. CM_IOCTL_GET_FEATURE_REQUEST

CM_IOCTL_GET_FEATURE_REQUEST 会返回读写器支持的功能列表。

hCard	由函数 SCardConnect 返回的引用值
dwControlCode	CM_IOCTL_GET_FEATURE_REQUEST
lpInBuffer	NULL
nInBufferSize	必须是 lpInBuffer 的 sizeof(ULONG)
lpOutBuffer	根据 PC/SC 2.0 规范第 10 部分规定，定义了以下功能：

```
#define FEATURE_VERIFY_PIN_START 0x01
#define FEATURE_VERIFY_PIN_FINISH 0x02
#define FEATURE_MODIFY_PIN_START 0x03
#define FEATURE_MODIFY_PIN_FINISH 0x04
#define FEATURE_GET_KEY_PRESSED 0x05
#define FEATURE_VERIFY_PIN_DIRECT 0x06
#define FEATURE_MODIFY_PIN_DIRECT 0x07
#define FEATURE_MCT_READERDIRECT 0x08
#define FEATURE_MCT_UNIVERSAL 0x09
#define FEATURE_IFD_PIN_PROP 0x0A
#define FEATURE_ABORT 0x0B
```

ACR83 支持以下功能：

```
#define FEATURE_VERIFY_PIN_DIRECT 0x06
#define FEATURE_MODIFY_PIN_DIRECT 0x07
#define FEATURE_IFD_PIN_PROP 0x0A
```



如果使用的ACR83支持PC/SC 2.0第10部分规定，您将获得以下数据：

06 04 XX XX XX XX 07 04 XX XX XX XX 0A 04 XX XX XX XXh

其中，XX XX XX XXh 为功能的控制代码。

nOutBufferSize *lpOutBuffer* 的 **sizeof(ULONG)**

lpBytesReturned 指向一个 **DWORD** 变量的指针，该变量用于接收存储进缓冲区的数据的大小（字节数），而该缓冲区由 *lpOutBuffer* 指定。

3.3.2. FEATURE_VERIFY_PIN_DIRECT

hCard 由函数 *SCardConnect* 返回的引用值

dwControlCode CM_IOCTL_GET_FEATURE_REQUEST

lpInBuffer

偏移	数据域	大小	值	说明
0	<i>bTimeOut</i>	1	-	秒数。若值等于 00h，则使用默认值
1	<i>bTimeOut2</i>	1	00h	不支持。第一次按键后的秒数
2	<i>bmFormatString</i>	1	-	PIN 格式选项的几个参数，更多信息请参阅 附录 A 。
3	<i>bmPINBlockString</i>	1	-	定义 APDU 命令中 PIN 数据块的长度（字节数）。更多信息请参阅 附录 B 。
4	<i>bmPINLengthFormat</i>	1	-	允许在 APDU 命令中加入 PIN 码长度。更多信息请参阅 附录 C 。
5	<i>wPINMaxExtraDigit</i>	2	XXYYh	XXh: PIN 码最大长度（位数） Yyh: PIN 码最小长度（位数）
7	<i>bEntryValidationCondition</i>	1	-	该值是一个位 OR 运算 01h = 达到最大长度 02h = 按下确认键 04h = 出现超时
8	<i>bNumberMessage</i>	1	FFh	PIN 验证显示的消息数量
9	<i>wLangId</i>	2	0409h	消息的语言
11	<i>bMsgIndex</i>	1	00h	消息索引（应为 00h）
12	<i>bTeoPrologue</i>	3	000000h	要使用的 T=1 信息块（I-block）起始域（填写 00h）
15	<i>ulDataLength</i>	4	-	待发送至 ICC 的数据的长度。



偏移	数据域	大小	值	说明
19	<i>abData</i>	-	-	待发送至 ICC 的数据

nInBufferSize 19 + *uLDataLength*

lpOutBuffer

偏移	数据域	大小	值	说明
0	<i>abStatus</i>	2	-	<p>6400h: SPE 操作超时</p> <p>6401h: 通过“取消”按钮取消了 SPE 操作。</p> <p>6402h: 两次输入的“新 PIN”不一致, PIN 修改操作失败</p> <p>6403h: 用户输入的 PIN 太短或太长, 不符合最短/最长 PIN 码长度限制</p> <p>注: ACR83 在 PIN 输入过程中检查 PIN 的长度, 将不再返回此状态。</p> <p>6B80h: 传递的结构参数无效</p> <p>SW1SW2: 来自卡片的结果</p>

nOutBufferSize 2

lpBytesReturned 指向一个 *DWORD* 变量的指针, 该变量用于接收存储进缓冲区的数据的长度 (字节数), 而该缓冲区由 *lpOutBuffer* 指定。

3.3.3. FEATURE_MODIFY_PIN_DIRECT

hCard 由函数 *SCardConnect* 返回的引用值

dwControlCode CM_IOCTL_GET_FEATURE_REQUEST

lpInBuffer

偏移	数据域	大小	值	说明
0	<i>bTimeOut</i>	1	-	秒数。若值等于 00h, 则使用默认值
1	<i>bTimeOut2</i>	1	00h	不支持。第一次按键后的秒数
2	<i>bmFormatString</i>	1	-	PIN 码格式选项的几个参数, 更多信息请参阅 附录 A 。
3	<i>bmPINBlockString</i>	1	-	定义 APDU 命令中 PIN 数据块的字节长度。更多信息请参阅 附录 B 。



偏移	数据域	大小	值	说明
4	<i>bmPINLengthFormat</i>	1	-	允许在 APDU 命令中加入 PIN 码长度。更多信息请参阅附录 C。
5	<i>bInsertionOffsetOld</i>	1	-	当前 PIN 码的插入位置偏移（字节）
6	<i>bInsertionOffsetNew</i>	1	-	新 PIN 码的插入位置偏移（字节）
7	<i>wPINMaxExtraDigit</i>	2	XXYYh	XXh: PIN 码最大长度（位数） Yyh: PIN 码最小长度（位数）
9	<i>bConfirmPIN</i>	1	00h, 01h, 02h, 03h	表示是否要在新的 PIN 码生效前进行确认（意思是用户要输入两次新 PIN 后，该 PIN 才会生效） 表示是否要在相同的 APDU 域输入并设置当前 PIN 码 b0: (0/1) 如果为 0 = 无需进行确认 如果为 1 = 需要进行确认 b1: (0/1) 如果为 0 = 无需输入当前 PIN（在此情况下， <i>bInsertinoOffsetOld</i> 的值不能被考虑在内） 如果为 1 = 需要输入当前 PIN b2 – b7: RFU
10	<i>bEntryValidationCondition</i>	1	-	该值是一个位 OR 运算 01h = 达到最大长度 02h = 按下确认键 04h = 出现超时
11	<i>bNumberMessage</i>	1	FFh	PIN 验证显示的消息数量
12	<i>wLangId</i>	2	0409h	消息的语言
14	<i>bMsgIndex1</i>	1	00h	第一条提示信息的索引
15	<i>bMsgIndex2</i>	1	01h	第二条提示信息的索引
16	<i>bMsgIndex3</i>	1	02h	第三条提示信息的索引
17	<i>bTeoPrologue</i>	3	000000h	要使用的 T=1 信息块（I-block）起始域（填写 00h）。
20	<i>ulDataLength</i>	4	-	待发送至 ICC 的数据的长度。
24	<i>abData</i>		-	待发送至 ICC 的数据



nInBufferSize 24 + ulDataLength

lpOutBuffer

偏移	数据域	大小	值	说明
0	abStatus	2	-	<p>6400h: SPE 操作超时</p> <p>6401h: 通过“取消”按钮取消了 SPE 操作。</p> <p>6402h: 两次输入的“新 PIN”不一致, PIN 修改操作失败</p> <p>6403h: 用户输入的 PIN 太短或太长, 不符合最短/最长 PIN 码长度限制</p> <p>注: ACR83 在 PIN 输入过程中检查 PIN 的长度, 将不再返回此状态。</p> <p>6B80h: 传递的结构参数无效</p> <p>SW1SW2: 来自卡片的结果</p>

nOutBufferSize 2

lpBytesReturned 指向一个 *DWORD* 变量的指针, 该变量用于接收存储进缓冲区的数据的大小 (字节数), 而该缓冲区由 *lpOutBuffer* 指定。

3.3.4. FEATURE_IFD_PIN_PROP

hCard 由函数 *SCardConnect* 返回的引用值。

dwControlCode 由 *CM_IOCTL_GET_FEATURE_REQUEST* 返回。

lpInBuffer NULL

LpOutBuffer

偏移	数据域	大小	值	说明
0	wLcdLayout	2	0210h	显示特性: 2 行, 每行 16 个字符
2	bEntryValidationCondition	1	07h	支持超时时间到、达到 PIN 码最大长度以及按下确认按钮这三种条件
3	bTimeOut2	1	00h	<p>0 = IFD 不能识别 <i>bTimeOut</i> 和 <i>bTimeOut2</i></p> <p>1 = IFD 可以识别 <i>bTimeOut</i> 和 <i>bTimeOut2</i></p>

nOutBufferSize 4

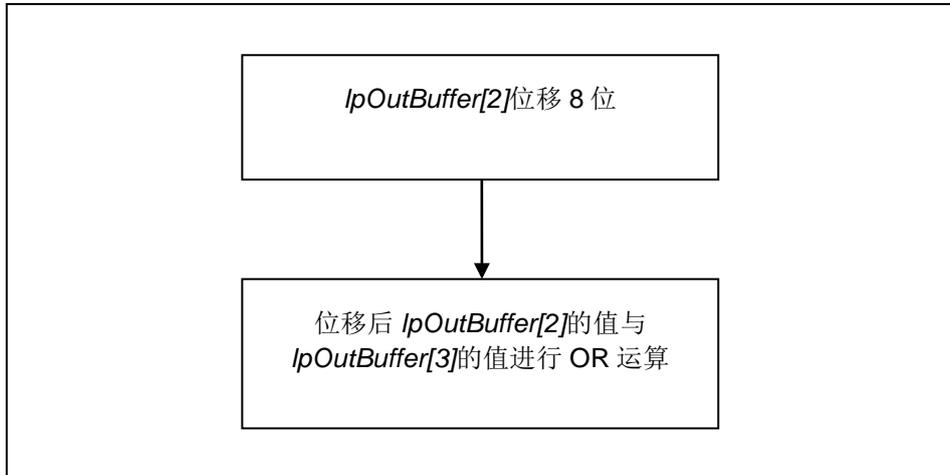
lpBytesReturned 指向一个 *DWORD* 变量的指针, 该变量用于接收存储进缓冲区的数据的大小 (字节数), 而该缓冲区由 *lpOutBuffer* 指定。

3.3.5. IOCTL_SMARTCARD_GET_FIRMWARE_VERSION

IOCTL_SMARTCARD_GET_FIRMWARE_VERSION 用于启用 *Get Firmware Version* 命令。

3.3.5.1. 固件版本号

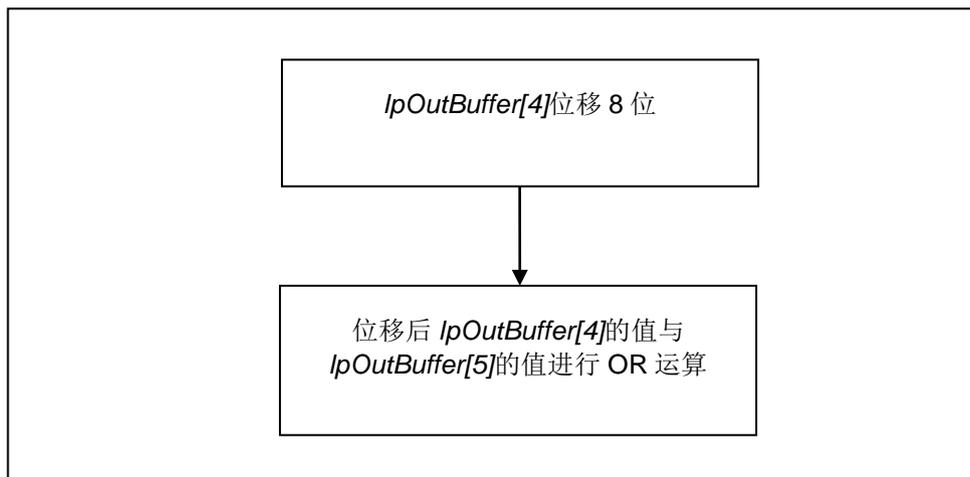
为了获得设备的固件版本号，要将接收到的缓冲数据的第三个数据元位移 8 位，之后再将位移后的结果与缓冲数据的第四个数据元进行 OR 运算。



例如：`Firmware_Version = (Common.RecvBuff[2] << 8) | Common.RecvBuff[3];`

3.3.5.2. LCD

为了获得设备 LCD 的信息，要将接收到的缓冲数据的第五个数据元位移 8 位，之后再将位移后的结果与缓冲数据的第六个数据元进行 OR 运算。



输入数据:

hCard 由函数 *SCardConnect* 返回的引用值
dwControlCode IOCTL_SMARTCARD_GET_FIRMWARE_VERSION

输出数据:

lpOutBuffer 命令输出的值



nOutBufferSize *lpOutBuffer* 的 **sizeof(ULONG)**

lpBytesReturned 指向一个 **DWORD** 变量的指针，该变量用于接收存储进缓冲区的数据的大小（字节数），而该缓冲区由 *lpOutBuffer* 指定。

偏移	数据域	大小	值	说明
0	<i>abStatus</i>	2	0000h	成功
2	<i>wACR83Firmware</i>	2		-
4	<i>LCD</i>	2		-

3.3.6. IOCTL_SMARTCARD_DISPLAY_LCD_MESSAGE

IOCTL_SMARTCARD_DISPLAY_LCD_MESSAGE 用于启用 *Display LCD Message* 命令。

hCard 由函数 *SCardConnect* 返回的引用值

dwControlCode IOCTL_SMARTCARD_DISPLAY_LCD_MESSAGE

lpInBuffer 设置 *Display LCD Message* 选项的值

nInBufferSize *lpInBuffer* 的 **sizeof(ULONG)**

偏移	数据域	大小	值	说明
0	<i>abLCDmessage</i>	0-32	-	LCD 消息（最多 32 个字符）

输出数据：

lpOutBuffer 命令输出的值

nOutBufferSize *lpOutBuffer* 的 **sizeof(ULONG)**

lpBytesReturned 指向一个 **DWORD** 变量的指针，该变量用于接收存储进缓冲区的数据的大小（字节数），而该缓冲区由 *lpOutBuffer* 指定。

偏移	数据域	大小	值	说明
0	<i>abStatus</i>	2	0000h 0001h	成功 BAD_PARAMETER

3.3.7. IOCTL_SMARTCARD_READ_KEY

IOCTL_SMARTCARD_READ_KEY 用于启用 *Read Key* 命令。

输入数据：

hCard 由函数 *SCardConnect* 返回的引用值

dwControlCode IOCTL_SMARTCARD_READ_KEY



lpInBuffer 设置 Display LCD Message 选项的值

nInBufferSize *lpInBuffer* 的 **sizeof(ULONG)**

偏移	数据域	大小	值	说明
0	<i>bTimeOut</i>	1	-	秒数若值等于 00h，则使用默认值。
1	<i>wPINMaxExtraDigit</i>	2	XXYYh	XXh: PIN 码最大长度 (位数) YYh: PIN 码最小长度 (位数)
3	<i>bKeyReturnCondition</i>	1	-	该值是一个位 OR 运算 01h: 达到最大长度 02h: 按下了按键[E] 04h: 发生超时 08h: 按下了按键[C]
4	<i>bEchoLCDStartPosition</i>	1	-	开始位置 (0 – 31)
5	<i>bEchoLCDMode</i>	1	-	00h: 通过按键输入的数据在 LCD 上显示为 ASCII 格式 01h: 通过按键输入的数据在 LCD 上显示为星号“*”。

输出数据:

lpOutBuffer 命令输出的值

nOutBufferSize *lpOutBuffer* 的 **sizeof(ULONG)**

lpBytesReturned 指向一个 *DWORD* 变量的指针，该变量用于接收存储进缓冲区的数据的大小 (字节数)，而该缓冲区由 *lpOutBuffer* 指定。

偏移	数据域	大小	值	说明
0	<i>abStatus</i>	2	0000h 0001h	成功 BAD_PARAMETER
2	<i>bKeyReturnCondition</i>	1	31h 32h 33h 34h	达到最大长度 按下了按键[E] 发生超时 按下了按键[C]
3	<i>abNumericInputKeys</i>	0-32	-	-



附录A. 设置 bKeyReturnCondition

<i>bKeyReturnCondition</i>	OR 操作数
若达到 PIN 码最大长度	0x01h
若按下 ACR83 设备的 KEY_E	0x02
若 ACR83 会话超时时间到	0x04h
若按下 ACR83 设备的 KEY_C	0x08h
若按下 ACR83 设备的 KEY_BACK	0x10h
若按下 ACR83 设备的 KEY_FN	0x20h

注：将值设为对特定的 OR 操作数进行 OR 运算。



附录B. 响应错误代码

下表汇总了 ACR83 (CCID) 可能返回的错误代码:

错误代码	状态
0001h	BAD_PARAMETER
0083h	SLOTERROR_LCDCOMMANDERROR
0084h	SLOTERROR_WRONGCONFIRMPIN
0085h	SLOTERROR_UNKNOWN_LCD
0086h	SLOTERROR_MAXPIN_SIZE_EQUAL_ZERO
00EFh	SLOTERROR_PIN_CANCELLED
00F0h	SLOTERROR_PIN_TIMEOUT



附录C. **bmFormatStrings** 说明

位号	说明
Bit 7	系统单位的类型指示符 若为 0h: 系统的单位是比特 (Bit) 若为 1h: 系统的单位是字节 (Byte) 该位可量化下一个参数 (单位移动)
Bit 6 – 3	定义格式化后 PIN 在 APDU 命令中的位置 (相对于 Lc 后的第一个数据)。该位置基于系统单位的类型指示符来确定 (最大值为 1111, 15 个系统单位)
Bit 2	PIN 对齐的位掩码 若为 0h: 数据左对齐 若为 1h: 数据右对齐
Bit 1-0	PIN 格式类型 00h: 二进制 01h: BCD 10h: ASCII



附录D. **bmPINBlockString** 说明

位号	说明
Bit 7 - 4	加入 APDU 命令的 PIN 长度的大小，单位为比特。（如果该值为 0，则不在 APDU 命令中加入有效的 PIN 长度）。
Bit 3 - 0	PIN 长度信息：经过对齐和格式化后 PIN 数据块的大小，单位为字节



附录E. bmPINLength 格式

位号	说明
Bit 7-5	RFU
Bit 4	系统单位的类型指示符 若为 0h: 系统的单位是比特 (Bit) 若为 1h: 系统的单位是字节 (Byte)
Bit 3 - 0	根据前面所讲的参数表示 PIN 长度在 APDU 命令中的位置 (最大值为 1111, 15 个系统单位)。