

居民身份证识读智能接口 Linux 版 软件说明手册 V2.0

2024-1-26

一、概述

本手册是操作居民身份证识读智能接口的数据结构定义、调用方法和返回值的说明。

二、定义

开发包含下列文件：

编号	文件名	说明
1	termbSDK.h, termbSDK.cpp	API 函数原型定义，动态库导入方法。
2	libtermb_###_**.so	API 函数库文件，需要执行权限 GCC 版本 (###) : glibc、uclibc 系统架构 (***) : arm、mips、x86 等
3	libwlt.so, license.bin	相关的后台库，需要执行权限

平台支持：

CPU 平台	系统版本
X86_64 (Intel/AMD/海光/兆芯等)	Ubuntu12.04+, Deepin OS V15.11+, UOS 20, 银河麒麟 V4/V10
ARM64(飞腾、鲲鹏等)	UOS 20, 麒麟系统 V4/V10
MIPS64(龙芯)	UOS 20, 麒麟系统 V4/V10
其他 Linux 系统/CPU 等	根据不同系统、不同 CPU 定制评估适配

适用开发工具语言：

GCC 4.8 及以后版本的 C/C++, Python3, Java(jna)等。

目 录

一、概述.....	1
二、定义.....	1
三、技术参数.....	3
四、数据结构定义.....	4
五、函数说明.....	8
附录一 返回错误代码说明：	31
附录二 居民身份证文字信息.....	32
附录三 2017 版外国人永久居留证.....	33
附录四 港澳台居民居住证文字信息.....	34
附录五 新版外国人永久居留证文字信息.....	34
附录六 民族代码对照表.....	36
附录七 性别代码对照表.....	36
附录八 推导 2017 版永居证证件号码.....	37

三、技术参数

3.1 支持设备列表

接入设备解压 SDK 直接重编译验证 SDK 是否可以运行, 如果没有运行问题, 直接忽略以下操作:

- 1) 接入设备,终端输入 lsusb 指令, 确认指令是否可用。如果报错需要联网并执行以下指令安装 libusb-dev 驱动: `sudo apt-get install libusb-dev`。
- 2) 将规则文件 "98-mkreader.rules" 上传至 /etc/udev/rules.d 目录, 执行以下命令:
`sudo service udev reload && sudo service udev restart`, 如果未生效, 尝试拔插 USB 或者重启系统。

设备名	VID	PID
MKR-100	0xC35A	0x0400
MKR-200	0x2B8B	0x2721

3.2 函数列表

本函数列表列出所有相关功能函数接口及支持产品类型。函数列为接口函数名称, 支持产品列为设备类别的简称: MKR-100 系列 (简称 1), MKR-200 系列 (简称 2)。

#	函数	说明	支持产品
读卡器操作函数			
1	MNK_InitComm	初始化端口	1/2
2	MNK_CloseComm	关闭端口	1/2
3	MNK_Authenticate	卡认证	1/2
4	MNK_Read_Content	读卡操作,信息文件存储在应用程序当前目录下	1/2

5	MNK_Read_Content_Path	读卡操作,信息文件存储在指定目录下	1/2
6	MNK_SetIDCardInfo	读卡信息解码	1/2
7	MNK_GetSAMID	获取安全模块编号	1/2
8	MNK_GetSAMIDToStr	获取 SAM 编号字符串	1/2
9	MNK_ActivateCard	激活(复位)卡片	2
10	MNK_TransCommand	读卡器指令交互	2
11	MNK_DecomposeIDCardInfo	获取身份证文字信息	1/2
12	MNK_DecomposeIDCardPhoto	获取身份证头像图片	1/2
13	MNK_StringConverter	字符串转码	1/2
14	MNK_ISO15693_ReadSingleBlock	读单块数据	2
15	MNK_ISO15693_WriteSingleBlock	写单块数据	2
16	MNK_ISO15693_ReadMultiBlocks	读多块数据	2
17	MNK_ISO15693_WriteMultipleBlocks	写多块数据	2
18	MNK_ISO15693_GetBlockSecStatus	获取数据块安全状态	2
19	MNK_ISO15693_GetSystemInfo	获取标签系统信息	2
20	MNK_ISO15693_LockMultipleBlocks	锁多块	2
21	MNK_ISO15693_WriteAFI	写 AFI	2
22	MNK_ISO15693_LockAFI	锁 AFI	2
23	MNK_ISO15693_WriteDSFID	写 DSFID	2
24	MNK_ISO15693_LockDSFID	锁 DSFID	2
25	MNK_ISO15693_Inventory	通过 AFI 寻卡	2
26	MNK_ISO14443_TransApdu	非接触卡 APDU 数据交换	2
27	MNK_MIFARE_Authentication	M1 卡认证	2
28	MNK_MIFARE_ReadData	M1 卡读数据	2
29	MNK_MIFARE_WriteData	M1 卡写数据	2
30	MNK_MIFARE_InitValue	M1 卡初始化值	2
31	MNK_MIFARE_IncrementValue	M1 卡加值	2
32	MNK_MIFARE_DecrementValue	M1 卡减值	2
33	MNK_MIFARE_ReadValue	M1 卡读值	2
34	MNK_MIFARE_WriteSectorTrailer	M1 卡写控制块信息	2

四、数据结构定义

MNK_IDCardInfo 结构说明见下表:

#	结构变量声明	说明
1	int document_type	证件类型:1-身份证, 2-2017 版外国人永久居留证, 3-港澳台居民居住证, 4-新版

		外国人永久居留证																																				
2	union DocumentInformati on	<p>DocumentInformation 为结构联合体声明见下面表格：</p> <p>在 linux 中，中文字段存储空间计算方法：HANZI_SIZE(a) = ((a+1) * 4)</p> <p>字符串字段存储空间计算方法：ASC_SIZE(a) = ((a+1) * 1)，单位 Byte。</p> <table><tr><td colspan="2">MNK_GATDoc GATDoc</td></tr><tr><td colspan="2">港澳台居民居住证信息定义</td></tr><tr><td>uchar name[HANZI_SIZE(15)]</td><td>姓名</td></tr><tr><td>uchar gender[ASC_SIZE(1)]</td><td>性别</td></tr><tr><td>uchar reserve1[ASC_SIZE(2)]</td><td>预留</td></tr><tr><td>uchar birth_date[ASC_SIZE(8)]</td><td>生日</td></tr><tr><td>uchar address[HANZI_SIZE(35)]</td><td>居住地址</td></tr><tr><td>uchar id[ASC_SIZE(18)]</td><td>证件号码</td></tr><tr><td>uchar issuing_authority[HANZI_SIZE(15)]</td><td>签发机关</td></tr><tr><td>uchar validity_date_from[ASC_SIZE(8)]</td><td>签发日期</td></tr><tr><td>uchar validity_date_to[ASC_SIZE(8)]</td><td>有效期</td></tr><tr><td>uchar travel_permit_number[ASC_SIZE(9)]</td><td>通行证号码</td></tr><tr><td>uchar issue_times[ASC_SIZE(2)]</td><td>签发次数</td></tr><tr><td>uchar reserve2[ASC_SIZE(3)]</td><td>预留</td></tr><tr><td>uchar document_type[ASC_SIZE(1)]</td><td>证件类型'J'</td></tr><tr><td>uchar reserve3[ASC_SIZE(3)]</td><td>预留</td></tr><tr><td colspan="2">MNK_PRDoc_Old PRDoc_Old</td></tr><tr><td colspan="2">2017 版外国人永久居留身份证信息定义</td></tr></table>	MNK_GATDoc GATDoc		港澳台居民居住证信息定义		uchar name[HANZI_SIZE(15)]	姓名	uchar gender[ASC_SIZE(1)]	性别	uchar reserve1[ASC_SIZE(2)]	预留	uchar birth_date[ASC_SIZE(8)]	生日	uchar address[HANZI_SIZE(35)]	居住地址	uchar id[ASC_SIZE(18)]	证件号码	uchar issuing_authority[HANZI_SIZE(15)]	签发机关	uchar validity_date_from[ASC_SIZE(8)]	签发日期	uchar validity_date_to[ASC_SIZE(8)]	有效期	uchar travel_permit_number[ASC_SIZE(9)]	通行证号码	uchar issue_times[ASC_SIZE(2)]	签发次数	uchar reserve2[ASC_SIZE(3)]	预留	uchar document_type[ASC_SIZE(1)]	证件类型'J'	uchar reserve3[ASC_SIZE(3)]	预留	MNK_PRDoc_Old PRDoc_Old		2017 版外国人永久居留身份证信息定义	
MNK_GATDoc GATDoc																																						
港澳台居民居住证信息定义																																						
uchar name[HANZI_SIZE(15)]	姓名																																					
uchar gender[ASC_SIZE(1)]	性别																																					
uchar reserve1[ASC_SIZE(2)]	预留																																					
uchar birth_date[ASC_SIZE(8)]	生日																																					
uchar address[HANZI_SIZE(35)]	居住地址																																					
uchar id[ASC_SIZE(18)]	证件号码																																					
uchar issuing_authority[HANZI_SIZE(15)]	签发机关																																					
uchar validity_date_from[ASC_SIZE(8)]	签发日期																																					
uchar validity_date_to[ASC_SIZE(8)]	有效期																																					
uchar travel_permit_number[ASC_SIZE(9)]	通行证号码																																					
uchar issue_times[ASC_SIZE(2)]	签发次数																																					
uchar reserve2[ASC_SIZE(3)]	预留																																					
uchar document_type[ASC_SIZE(1)]	证件类型'J'																																					
uchar reserve3[ASC_SIZE(3)]	预留																																					
MNK_PRDoc_Old PRDoc_Old																																						
2017 版外国人永久居留身份证信息定义																																						

		uchar english_name[ASC_SIZE(60)]	英文姓名
		uchar gender[ASC_SIZE(1)]	性别
		uchar id[ASC_SIZE(15)]	永久居留证号码
		uchar nationality[ASC_SIZE(3)]	国籍或所在地区代码
		uchar chinese_name[HANZI_SIZE(15)]	中文姓名
		uchar validity_date_from[ASC_SIZE(8)]	证件签发日期
		uchar validity_date_to[ASC_SIZE(8)]	证件终止日期
		uchar birth_date[ASC_SIZE(8)]	出生日期
		uchar doc_version[ASC_SIZE(2)]	证件版本号
		uchar issuing_authority[ASC_SIZE(4)]	当次申请受理机关代码
		uchar document_type[ASC_SIZE(1)]	证件类型标识'I'
		uchar reserve[ASC_SIZE(3)]	预留项
		MNK_PRDoc PRDoc	
		新版外国人永久居留身份证信息定义	
		uchar chinese_name[HANZI_SIZE(15)]	中文姓名
		uchar gender[ASC_SIZE(1)]	性别
		uchar renewal_times[ASC_SIZE(2)]	换证次数
		uchar birth_date[ASC_SIZE(8)]	出生日期
		uchar english_name[ASC_SIZE(35)]	英文姓名
		uchar id[ASC_SIZE(18)]	永久居留证号码
		uchar issuing_authority[HANZI_SIZE(15)]	预留项

		uchar validity_date_from[ASC_SIZE(8)]	证件签发日期
		uchar validity_date_to[ASC_SIZE(8)]	证件终止日期
		uchar alternate_english_name[ASC_SIZE(11)]	英文姓名备用
		uchar nationality[ASC_SIZE(3)]	国籍或所在地区代码
		uchar document_type[ASC_SIZE(1)]	证件类型标识'Y'
		uchar historical_id[ASC_SIZE(3)]	既往版本证件号码关联项
		MNK_SFZDoc SFZDoc 身份证信息定义	
		uchar name[HANZI_SIZE(15)]	姓名
		uchar gender[ASC_SIZE(1)]	性别
		uchar ethnic_group[ASC_SIZE(2)]	民族
		uchar birth_date[ASC_SIZE(8)]	出生日期
		uchar address[HANZI_SIZE(35)]	详细地址
		uchar id[ASC_SIZE(18)]	证件号码
		uchar issuing_authority[HANZI_SIZE(15)]	签发机关
		uchar validity_date_from[ASC_SIZE(8)]	证件签发日期
		uchar validity_date_to[ASC_SIZE(8)]	证件终止日期
		uchar reserve[ASC_SIZE(18)]	预留项
3	uchar photo [1024]	证件头像文件名	
4	uchar fingers [1024]	证件指纹文件名，如果无指纹为空。	

五、函数说明

5.1 初始化读卡器端口

原 型：int MNK_InitComm (int Port)

说 明：本函数用于读卡器端口初始化。

参 数：Port：端口编码

	值（10 进制）	意义
1	1~16	设备 ttyS1 到 ttyS16
2	20~36	设备 ttyUSB1 到 ttyUSB16
3	40~56	设备 ttySP1 到 ttySP16
2	1001~1016	设备 USB1 至 USB16
3	2401	设备 USB HID 端口

返 回 值：

值	意义
0	正确
其它	错误（具体见附录一）

5.2 关闭读卡器端口

原 型：int MNK_CloseComm (void)

说 明：本函数用于关闭已经打开的读卡器端口。

参 数：无

返 回 值：

值	意义
0	正确
其它	错误（具体见附录一）

5.3 卡认证

原 型：int MNK_Authenticate (void)

说 明：本函数用于读卡器和身份证之间的合法身份确认。

参 数：无

返 回 值：

值	意义	说明
0	正确	卡片正确放置时
其它	错误	未放卡或卡片放置不正确时

注意：若卡片放置后发生认证错误时，应移走卡片重新放置。

注意：若采用查询方式自动判断卡片是否放置，则间隔时间建议大于 300ms。

5.4 读卡操作

a) 原 型：int MNK_Read_Content (int Active);

说 明：本函数用于通过读卡器从非接触身份证卡中读取相应信息，
信息存在当前目录下。

参 数：Active 读取信息类型

Active 值	意义	说明
1	读基本信息	读取成功后保存在内存中，通过 MNK_DecodeIDCardInfo

		方法读取
2	只读文字信息	读取成功后保存在内存中, 通过 MNK_DecodeIDCardInfo 方法读取
3	读最新住址信息	读取成功后保存在内存中, 通过 MNK_DecodeIDCardInfo 方法读取
4	只读文字信息	形成文字信息文件 WZ.TXT
5	只读照片信息	形成相片文件 XP.WLT 和 ZP.BMP
6	读取基本信息和 指纹信息	形成文字信息文件 WZ.TXT、相片文件 XP.WLT 和 ZP.BMP, 指纹信息文件 FP.BIN

返回 值:

值	意义
0	正确
其它	错误 (具体见附录一)

b) 原 型: int MNK_Read_Content_Path (char* cPath, int Active);

说 明: 本函数用于通过读卡器从非接触身份证卡中读取相应信息, 信息存储在由 cPath 参数指定的路径下。

参 数: cPath 信息存储路径, 路径长度不能大于 240

Active 读取信息类型

Active 值	意义	说明
1	读基本信息	形成文字信息文件 WZ.TXT、相片文件 XP.WLT 和 ZP.BMP

2	只读文字信息	形成文字信息文件 WZ.TXT 和相片文件 XP.WLT
3	读最新住址信息	形成最新住址文件 NEWADD.TXT
4	只读文字信息	形成文字信息文件 WZ.TXT
5	只读照片信息	形成相片文件 XP.WLT 和 ZP.BMP
6	读取基本信息和 指纹信息	形成文字信息文件 WZ.TXT、相片文件 XP.WLT 和 ZP.BMP, 指纹信息文件 FP.BIN

返 回 值:

值	意义
0	正确
其它	错误 (具体见附录一)

c) 原 型: int MNK_SetIDCardInfo (char* sdtBuffer);

说 明: 本函数本函数将 SDT 返回的读卡信息解码,解码成功后可通过 MNK_DecodeIDCardInfo 获取文字信息通过 MNK_DecodeIDCardInfo 获取 BMP 格式头像信息。

参 数: sdtBuffer:指向 sdt 返回读卡原始数据数组指针。

返 回 值:

值	意义
0	正确
其它	错误 (具体见附录一)

注意:

1、读完基本信息后,若需要立即读取最新住址信息,在未移走卡片的情况下可以不用卡认证;

- 2、 单独读取最新住址信息，需要先进行卡认证；
- 3、 若卡片放置后发生读卡错误时，应移走卡片重新放置。
- 4、 当 Active 值为 6 时读取身份证信息，如果该证件无指纹信息将不会生成 FP.BIN 文件。

5.5 获取 SAM 编号

原 型：int MNK_GetSAMID(unsigned char *pucSAMID, int *nLen)

说 明：本函数用于读取 SAM 模块的编号。

参 数：pucSAMID:应用申请的数组指针，长度不少于 32 字节

nLen: 数组的长度

返 回 值：

值	意义
0	正确
其它	错误（具体见附录一）

5.6 获取 SAM 编号字符串

原 型：int MNK_GetSAMIDToStr (unsigned char *pucSAMID, int *nLen)

说 明：本函数用于读取 SAM 模块编号的字符串。

参 数：pucSAMID:应用申请的字符串指针，长度不少于 32 字节

nLen: 数组的长度

返 回 值：

值	意义
0	正确

其它	错误（具体见附录一）
----	------------

5.7 复位卡片

原 型：int MNK_ActivateCard (int nReaderSlot, unsigned char *pRecvDatas, int *nRecvLen)

说 明：本函数用于向读卡器发送卡片复位指令，并返回卡片复位信息。

参 数：int nReaderSlot：操作读卡器通讯选项

值（16 进制）	说明
0x01	ISO 14443 TypeA 协议
0x02	ISO 14443 TypeB 协议
0x03	ISO 15693 协议
0x04	读取身份证 DN 码
0x05	Mifare1 卡

unsigned char *pRecvDatas 接收读卡器命令返回数据

int *nRecvLen：接收返回数据长度

返回数据 pRecvDatas 内容格式说明：

TYPE A：UID 长度（1 字节）+UID+ATR 长度（1 字节）+ATR

TYPE B：UID 长度（1 字节）+UID+ATR 长度（1 字节）+ATR

Mifare1：UID（4 字节）

身份证：身份证 DN 码

返 回 值：

值	意义
---	----

0	正确
其它	错误（具体见附录一）

5.8 读卡器 APDU 命令通讯

原型：int MNK_ISO14443_TransApdu (int nReaderSlot, unsigned char *pSendDatas, int nSendLen, unsigned char *pRecvDatas, int *nRecvLen)

说明：本函数用于向读卡器发送 APDU 命令并接收返回数据。

参数：int nReaderSlot: 操作读卡器通讯选项

值（16 进制）	说明
0x01	ISO 14443 TypeA 协议
0x02	ISO 14443 TypeB 协议
0x03	ISO 15693 协议
0x04	读取身份证 DN 码
0x05	Mifare1 卡

unsigned char *pSendDatas: 发送命令数据（参见附录六）

int nSendLen: 发送命令数据长度

unsigned char *pRecvDatas 接收读卡器命令返回数据

int *nRecvLen: 接收返回数据长度

返回数据 pRecvDatas 内容格式说明：

成功：0x00 0x00+返回数据

失败：0xxx 0xxx （两个字节错误码）

返回值：

值	意义
0	正确
其它	错误（具体见附录一）

5.9 解码身份证文字信息

原型：int MNK_DecodeIDCardInfo (MNK_IDCardInfo *pInfo, int nFormat)

说明：本函数用于获取从身份证读取的文字信息。

参数：MNK_IDCardInfo *pInfo:具体数据字段说明参考第四部分数据结构定义。

int nFormat:0 为 UCS-2LE, 1 为 GBK, 2 为 UTF8。

返回值：

值	意义
0	正确
其它	错误（具体见附录一）

5.10 解码身份证头像图片

原型：int MNK_DecodeIDCardInfo (unsigned char *photo)

说明：本函数用于获取从身份证读取的头像数据。

参数：unsigned char *photo:身份证头像图片, Bmp 格式数组分配空间为 38864 字节。

int nFormat:该参数保留，传 0。

返回值：

值	意义
0	正确

其它	错误（具体见附录一）
----	------------

5.11 字符串转码

原 型：int MNK_StringConverter (int iType, unsigned char *pInStr, unsigned char *pOutStr)

说 明：本函数用于将 pInStr 的字符串数据转换成 iType 指定编码格式，结果保存在 pOutStr。

参 数：unsigned char * pInStr： 输入待转换的字符串数组指针

unsigned char * pOutStr： 输出编码的字符串

int iType： 转换类型：0-GBK->UCS-2LE, 1-UCS-2LE->GBK

2-GBK->UTF8, 3-UTF8->GBK

4-UCS-2LE->UTF8, 5-UTF8->UCS-2LE。

返 回 值：

值	意义
0	正确
其它	错误

5.12 读卡器命令数据交互

原 型：int MNK_TransCommand(unsigned char *pSendDatas, int nSendLen, unsigned char *pRecvDatas, int *nRecvLen)

说 明：本函数用于向读卡器发送指令并接收返回数据。

参 数： unsigned char *pSendDatas： 发送命令数据

int nSendLen: 发送命令数据长度

unsigned char *pRecvDatas 接收读卡器命令返回数据

int *nRecvLen: 接收返回数据长度

返回数据 pRecvDatas 内容格式说明:

成功: 0x00 0x00+返回数据

失败: 0xxx 0xxx (两个字节错误码)

返 回 值:

值	意义
0	正确
其它	错误 (具体见附录一)

5.13 ISO15693 读单块数据

原 型: int MNK_ISO15693_ReadSingleBlock(bool readSecSta, int nBlkAddr,
unsigned char bufBlockDat[], int nSize, int *bytesBlkDatRead)

说 明: 本函数用于读取单块数据。

参 数:

值	意义
bool readSecSta	是否读数据块安全状态 1 读数据块和安全状态字节, 安全状态字节(1 个字节)+数据块(4 个字节)=5; 0 不读安全状态字节, 只有 4 个字节的块数据

int nBlkAddr	要读的数据块地址
int nSize	缓冲区可保存的最大字节数
unsigned char *bufBlockDat	字节缓冲区，用于保存已读到的数据块
int *bytesBlkDatRead	已写入缓冲区的字节数

返 回 值：

值	意义
0	正确
其它	错误（具体见附录一）

注意：此方法只能在 MKR-2XX 系列设备上使用。

5.14 ISO15693 写单块数据

原 型：int MNK_ISO15693_WriteSingleBlock(int blkAddr,unsigned char newBlkData[],int bytesToWrite)

说 明：本函数用于写单块数据。

参 数：

值	意义
int nBlkAddr	要写的数据块地址
unsigned char newBlkData[]	新的数据块数据
int bytesToWrite	要写入的字节数

返 回 值：

值	意义
---	----

0	正确
其它	错误（具体见附录一）

注意：此方法只能在 MKR-2XX 系列设备上使用。

5.15 ISO15693 读多块数据

原型：int MNK_ISO15693_ReadMultiBlocks(bool readSecSta, int nBlkAddr, int numOfBlksToRead, int *numOfBlksRead, unsigned char bufBlockDat[], int nSize, int *bytesBlkDatRead)

说明：本函数用于读取多块数据。

参数：

值	意义
bool readSecSta	是否读数据块安全状态 1 读数据块和安全状态字节, 安全状态字节(1 个字节)+数据块(4 个字节)=5; 0 不读安全状态字节, 只有 4 个字节的块数据
int nBlkAddr	要读的数据块地址
int nSize	缓冲区可保存的最大字节数
unsigned char *bufBlockDat	字节缓冲区, 用于保存已读到的数据块
int *bytesBlkDatRead	已写入缓冲区的字节数
int *numOfBlksRead	已读到的数据块数量

返回值:

值	意义
0	正确
其它	错误（具体见附录一）

注意：此方法只能在 MKR-2XX 系列设备上使用。

5.16 ISO15693 写多块数据

原型：int MNK_ISO15693_WriteMultipleBlocks(int blkAddr,int numOfBlks,
unsigned char newBlkData[],int bytesToWrite)

说明：本函数用于写多块数据。

参数:

值	意义
int nBlkAddr	要写的数据块地址
int numOfBlks	要写入的数据块数量
unsigned char newBlkData[]	新的数据块数据
int bytesToWrite	要写入的字节数

返回值:

值	意义
0	正确
其它	错误（具体见附录一）

注意：此方法只能在 MKR-2XX 系列设备上使用。

5.17 ISO15693 获取数据块安全状态

原 型：int MNK_ISO15693_GetBlockSecStatus(int blkAddr, int numOfBlks, unsigned char bufBlkSecs[], int nSize, int *bytesSecRead)

说 明：本函数用于获取数据块安全状态。

参 数：

值	意义
int nBlkAddr	要读的数据块地址
int numOfBlks	要读的块数量
int nSize	缓冲区可保存的最大字节数
unsigned char *bufBlkSecs	节的缓冲区,用于保存安全状态字节
int *bytesBlkDatRead	已写入缓冲区的字节数

返 回 值：

值	意义
0	正确
其它	错误（具体见附录一）

注意：此方法只能在 MKR-2XX 系列设备上使用。

5.18 ISO15693 获取标签系统信息

原 型：int MNK_ISO15693_GetSystemInfo(unsigned char uid[], unsigned char *dsfid, unsigned char *afi, int *blkSize ,int *numOfBlocs, unsigned char* icRef)

说 明：本函数用于获取标签系统信息。

参 数:

值	意义
unsigned char uid[]	8 个字节的缓冲区, 用于保存标签系列号
unsigned char *dsfid	用于保存标签信息 DSFID
unsigned char *afi	用于保存标签信息 AFI
int *blkSize	用于保存标签信息单块长度
int *numOfBlks	已用于保存标签信息总体块数量
unsigned char* icRef	用于保存标签信息 IC 引用

返 回 值:

值	意义
0	正确
其它	错误 (具体见附录一)

注意: 此方法只能在 MKR-2XX 系列设备上使用。

5.19 ISO15693 锁多块

原 型: int MNK_ISO15693_LockMultipleBlocks(int blkAddr, int numOfBlks)

说 明: 本函数用于锁多块。

参 数:

值	意义
int nBlkAddr	要锁的数据块地址
int numOfBlks	要锁的数据块数量

返 回 值:

值	意义
0	正确
其它	错误（具体见附录一）

注意：此方法只能在 MKR-2XX 系列设备上使用。

5.20 ISO15693 写 AFI

原 型：int MNK_ISO15693_WriteAFI(unsigned char afi)

说 明：本函数用于写 AFI。

参 数：

值	意义
unsigned char afi	AFI 值

返 回 值：

值	意义
0	正确
其它	错误（具体见附录一）

注意：此方法只能在 MKR-2XX 系列设备上使用。

5.21 ISO15693 锁 AFI

原 型：int MNK_ISO15693_LockAFI()

说 明：本函数用于锁 AFI。

参 数：无

返 回 值：

值	意义
0	正确
其它	错误（具体见附录一）

注意：此方法只能在 MKR-2XX 系列设备上使用。

5.22 ISO15693 写 DSFID

原 型：int MNK_ISO15693_WriteDSFID(unsigned char dsfid)

说 明：本函数用于写 DSFID。

参 数：

值	意义
unsigned char dsfid	DSFID 值

返 回 值：

值	意义
0	正确
其它	错误（具体见附录一）

注意：此方法只能在 MKR-2XX 系列设备上使用。

5.23 ISO15693 锁 DSFID

原 型：int MNK_ISO15693_LockDSFID()

说 明：本函数用于锁 DSFID。

参 数：无

返 回 值：

值	意义
0	正确
其它	错误（具体见附录一）

注意：此方法只能在 MKR-2XX 系列设备上使用。

5.24 ISO15693 通过 AFI 查找标签

原 型：int MNK_ISO15693_Inventory(unsigned char afi, unsigned char bufUids[],
int nSize, int *bytesUidNums)

说 明：本函数用于通过 AFI 查找标签。

参 数：

值	意义
unsigned char afi	Afi 值
unsigned char bufUids[]	用于保存标签 UID
int nSize	保存空间最大长度
int *bytesUidNums	已找到标签数量

返 回 值：

值	意义
0	正确
其它	错误（具体见附录一）

注意：此方法只能在 MKR-2XX 系列设备上使用。

5.25 ISO14443 APDU 数据交互

原型：int MNK_ISO14443_TransApu (int nReaderSlot, unsigned char *pSendDatas, int nSendLen, unsigned char *pRecvDatas, int *nRecvLen)

说明：本函数用于向读卡器发送 APDU 命令并接收返回数据。

参数：int nReaderSlot: 操作读卡器通讯选项

值 (16 进制)	说明
0x01	ISO 14443 TypeA 协议
0x02	ISO 14443 TypeB 协议

unsigned char *pSendDatas: APDU 命令数据

int nSendLen: 发送命令数据长度

unsigned char *pRecvDatas 接收读卡器命令返回数据

int *nRecvLen: 接收返回数据长度

返回数据 pRecvDatas 内容格式说明：

成功：卡片返回数据

失败：0xxx 0xxx （卡片返回错误码）

返回值：

值	意义
0	正确
其它	错误（具体见附录一）

5.26 MIFARE 1 卡认证

原型：int MNK_MIFARE_Authentication(unsigned char bSectorNo, unsigned char

*pbKeyA, unsigned char *pbKeyB)

说 明：本函数用于对 M1 卡片进行认证。

参 数： bSectorNo：待认证的扇区号，S50 卡（0~15），S70 卡（0~39）

pbKeyA：密钥 A 数据，6 字节，如果没有传 NULL

pbKeyB：密钥 B 数据，6 字节，如果没有传 NULL

返 回 值：

值	意义
0	正确
其它	错误（具体见附录一）

5.27 MIFARE1 卡读数据

原 型：int MNK_MIFARE_ReadData(unsigned char bBlock, unsigned char *data, int nLen)

说 明：本函数用于在卡认证后读取 M1 卡指定块数据。

参 数： bBlock：块号，S50 卡（0~64），S70 卡（0~255）

data：读取数据数组

nLen：读取块数据长度（1~16）

返 回 值：

值	意义
0	正确
其它	错误（具体见附录一）

5.28 MIFARE1 卡写数据

原 型: int MNK_MIFARE_WrtieData (unsigned char bBlock, unsigned char *data, int nLen)

说 明: 本函数用于在卡认证后写 M1 卡指定块数据。

参 数: bBlock: 块号, S50 卡 (0~64) , S70 卡 (0~255)

data: 写入数据数组

nLen: 写入块数据长度 (1~16)

返 回 值:

值	意义
0	正确
其它	错误 (具体见附录一)

5.29 MIFARE 1 卡初始化值

原 型: int MNK_MIFARE_InitValue(unsigned char bBlock, unsigned int nValue)

说 明: 本函数用于在卡认证后初始化 M1 卡指定块值数据。

参 数: bBlock: 块号, S50 卡 (0~64) , S70 卡 (0~255)

nValue: 初始化值数据

返 回 值:

值	意义
0	正确
其它	错误 (具体见附录一)

5.30 MIFARE 1 卡充值

原 型：int MNK_MIFARE_IncrementValue (unsigned char bBlock, unsigned int nValue)

说 明：本函数用于在卡认证后对 M1 卡指定块充值。

参 数：bBlock：块号，S50 卡（0~64），S70 卡（0~255）

nValue：充值数据

返 回 值：

值	意义
0	正确
其它	错误（具体见附录一）

5.31 MIFARE 1 卡减值

原 型：int MNK_MIFARE_DecrementValue (unsigned char bBlock, unsigned int nValue)

说 明：本函数用于在卡认证后对 M1 卡指定块减值。

参 数：bBlock：块号，S50 卡（0~64），S70 卡（0~255）

nValue：减值数据

返 回 值：

值	意义
0	正确
其它	错误（具体见附录一）

5.32 MIFARE 1 卡读值

原 型: int MNK_MIFARE_ReadValue(unsigned char bBlock, unsigned int *pnValue)

说 明: 本函数用于在卡认证后读取指定块值数据。

参 数: bBlock: 块号, S50 卡 (0~64) , S70 卡 (0~255)

pnValue: 保存待读值

返 回 值:

值	意义
0	正确
其它	错误 (具体见附录一)

5.33 MIFARE 1 卡写控制块信息

原 型: int MNK_MIFARE_WriteSectorTrailer(unsigned char bBlock, unsigned char *pbKEYA, unsigned char *pbKEYB, unsigned char *pbAccessBits)

说 明: 本函数用于在卡认证后写入指定块控制数据。

参 数: bBlock: 块号, S50 卡 (0~64) , S70 卡 (0~255)

pbKeyA: 密钥 A 数据, 6 字节

pbKeyB: 密钥 B 数据, 6 字节

pbAccessBits: 存取控制位, 4 字节

返 回 值:

值	意义
0	正确
其它	错误 (具体见附录一)

附录一 返回错误代码说明：

错误代码	说明
-1	软件未授权
-2	授权文件长度不正确
-3	分配 BUFFER 失败
-4	发送失败
-5	接收失败
-6	连接失败
-7	读取客户 IP 地址
-8	未找到卡
-9	命令不可用
-10	未找到芯片信息
-11	系统错误
-12	串口不能打开
-13	串口不能写
-14	串口不能读
-15	串口超时
-16	未定义错误
-27	参数错误
-28	没有默认打印机
-29	没有最新住址信息

-30	相片解码错误
-31	没找到设备
-32	读取居民身份证操作失败
-33	芯片中无此数据
-34	加载库文件错误
-35	导出库文件 API 接口错误

附录二 居民身份证文字信息

居民身份证文字信息采用 GB 13000 的 UCS-2 进行存储，各项目分配如下：

项目	长度（字节）	说明
姓名	30	汉字
性别	2	代码
民族	4	代码
出生	16	年月日：YYYYMMDD
住址	70	汉字和数字
公民身份号码	36	数字
签发机关	30	汉字
有效期起始日期	16	年月日：YYYYMMDD
有效期截止日期	16	年月日：YYYYMMDD 有效期为长期时存储“长期”

最新住址	70	汉字和数字
------	----	-------

附录三 2017 版外国人永久居留证

文字信息采用 GB 13000 的 UCS-2 进行存储，各项目分配如下：

项目	长度 (字节)	说明
英文姓名	120	大写英文字母。英文姓与英文名用逗号隔开。
性别	2	符合 GB/T 2261.1 的规定
永久居留证号码	30	15 个字符
国籍或所在地区代码	6	符合 GB/T 2659-2000 的规定
中文姓名	30	
证件签发日期	16	年月日：YYYYMMDD
证件终止日期	16	年月日：YYYYMMDD
出生日期	16	年月日：YYYYMMDD
证件版本号	4	
当次申请受理机关代码	8	
证件类型标识	2	大写字母 "I"
预留项	6	

附录四 港澳台居民居住证文字信息

采用 GB 13000 的 UCS-2 进行存储，各项目分配如下：

项目	长度 (字节)	说明
姓名	30	汉字
性别代码	2	符合 GB/T 2261.1 的规定
预留	4	预留区
出生日期	16	年月日：YYYYMMDD
住址	70	汉字和数字
证件号码	36	数字
签发机关	30	汉字
有效起始日期	16	年月日：YYYYMMDD
有效期截止日期	16	年月日：YYYYMMDD
通行证号码	18	
签发次数	4	
预留	6	预留区
证件类型标识	2	大写字母 “J”

附录五 新版外国人永久居留证文字信息

采用 GB 13000 的 UCS-2 进行存储，各项目分配如下：

项目	长度 (字节)	说明
----	---------	----

中文姓名	30	如无中文姓名, 则存储“无中文姓名”标识信息。
性别代码	2	符合 GB/T 2261.1 的规定
换证次数	4	首次申领为 00,再次申领证件时换证次数加 1
出生日期	16	年月日: YYYYMMDD
英文姓名	70	存储英文姓名第 1 至 35 位,与英文姓名备用项组合构成完整英文姓名
证件号码	36	
预留	30	预留区
证件签发日期	16	年月日: YYYYMMDD
证件终止日期	16	年月日: YYYYMMDD
英文姓名备用	22	英文姓名长度超过 35 个字符时,其余部分存储在此区域
国籍或所在地区代码	6	符合 GB/T 2659-2000 的规定
证件类型标识	2	大写字母 “Y”
既往版本外国人永久居留证件号码关联项	6	存储持证人曾持有既往外国人永久居留证件号码与新版外国人永久居留身份证号码的关联关系。

附录六 民族代码对照表

代码	民族	代码	民族	代码	民族	代码	民族
01	汉	15	土家	29	柯尔克孜	43	乌孜别克
02	蒙古	16	哈尼	30	土	44	俄罗斯
03	回	17	哈萨克	31	达斡尔	45	鄂温克
04	藏	18	傣	32	佤	46	德昂
05	维吾尔	19	黎	33	羌	47	保安
06	苗	20	傈僳	34	布朗	48	裕固
07	彝	21	佯	35	撒拉	49	京
08	壮	22	畲	36	毛南	50	塔塔尔
09	布依	23	高山	37	仡佬	51	独龙
10	朝鲜	24	拉祜	38	锡伯	52	鄂伦春
11	满	25	水	39	阿昌	53	赫哲
12	侗	26	东乡	40	普米	54	门巴
13	瑶	27	纳西	41	塔吉克	55	珞巴
14	白	28	景颇	42	怒	56	基诺
				98	外国血统	99	其他

附录七 性别代码对照表

代码	性别	代码	性别
----	----	----	----

0	未知	2	女
1	男	9	未说明

附录八 推导 2017 版永居证证件号码

当既往版本永居证证件号码关联项非空时说明之前办理过 2017 版永居证；若办理过 2017 版永居证，可以根据相关数据推导出 2017 版的永居证号码。

数据项名称	位置	对应关系
国家/地区代码	第1-3位	国家/地区代码
省/自治区/直辖市行政区划数字代码	第4-5位	2023版永居证证件号第2-3位
市行政区划数字代码	第6-7位	既往版本永居证证件号码关联项第1-2位
出生日期	第8-13位	出生日期后6位数字
顺序码	第14位	既往版本永居证证件号码关联项第3位
校验码	第15位	按照七三一算法，由前面14位计算得出

七三一算法示例：

```
String oldId = "PAK31008001010";
byte[] bufOldId = oldId.getBytes();
for (int i=0;i<bufOldId.length;i++){
    if (bufOldId[i] >= '0' && bufOldId[i] <= '9') {
        bufOldId[i] = (byte) (bufOldId[i] - 0x30);
    }
    else{
        bufOldId[i] = (byte) (bufOldId[i] - 0x41 + 10); // 字母'A'对应数字 10...
    }
}
int sum = bufOldId[0] * 7 + bufOldId[1] * 3 + bufOldId[2] * 1 +
bufOldId[3] * 7 + bufOldId[4] * 3 + bufOldId[5] * 1 +
bufOldId[6] * 7 + bufOldId[7] * 3 + bufOldId[8] * 1 +
bufOldId[9] * 7 + bufOldId[10] * 3 + bufOldId[11] * 1 +
```

```
bufOldId[12] * 7 + bufOldId[13] * 3;  
oldId += String.valueOf(sum%10);  
// 结果为:"PAK310080010103"
```