

目录

一、动态库简介	2
二、易用函数	2
激活CPU卡	2
初始化CPU卡	3
创建文件	3
修改文件密码	4
读文件数据	4
写文件数据	5
控制蜂鸣器响声	5
三、可作为软件加密狗，轻松实现自己软件的产权保护。	5

OUR_MIFARE.dll 动态库使用说明

一、动态库简介

动态库 OUR_MIFARE.dll 用 VC 开发，编译成 32 位 Release 实体或 64 位 Release 实体，适用于 WIN2000、XP、2003、win7 的 32 或 64 位、win8 编译和运行环境。本文档下面的源代码例子是 C++ 的调用格式。

本动态库是本公司 USB 接口 FM1208CPU 卡读写器的配套文件，必须和读写器一起使用。

OUR_MIFARE.dll 支持在软件运行中可以随时更换 USB 接口。本公司免驱型和有驱型读写器使用的 OUR_MIFARE.dll 不同，但函数的调用一样，如果客户程序开发好了，免驱型和有驱型互换时，只需更换对应的 OUR_MIFARE.dll。

推荐使用动态调用的方法使用本动态库。

二、易用函数

我们把复杂的 CPU 卡认证、读、写过程集成转化为更简单的函数调用操作，即使是从未接触过一卡通开发的软件开发人员，写起程序来也毫不困难，一点都不浪费开发时间。开发人员不再需要花大量时间去了解 CPU 卡的内部机制，我们将它封装成五个函数：1、CPU 卡 CPU 卡激活；2、初始化 CPU 卡；3、创建文件和删除文件；4、修改文件密码；5、读写文件。全国首创，只需十分钟就可轻松搞定 CPU 卡！。

第一步：CPU 卡激活

函数名：cpurequest

功能	复位 CPU 卡成功,CPU 卡进入 14443A-4 的协议模式，可以接着重复操作第二步进行调试了。	
VB 声明	Declare Function cpurequest Lib "OUR_MIFARE.dll" (ByVal serial As Long, ByVal param As Long, ByVal cosver As Long, ByVal code As Long) As Byte	
输入	1、serial 只需指向一个至少已分配了 4 个 char 空间的可写数组 unsigned char * 指针，serial 的下标由 0 开始。如果在控制字中没有指定 NEEDSERIAL，则 Serial 数组的内容无需赋值，因为此数组仅用于返回值。如果指定了 NEEDSERIAL，则必须为数组的内容赋值。 2、param 只需指向一个至少已分配了 4 个 char 空间的可写数组 unsigned char * 指针，param 的下标由 0 开始。如果在控制字中没有指定 NEEDSERIAL，则 param 数组的内容无需赋值，因为此数组仅用于返回值。如果指定了 NEEDSERIAL，则必须为数组的内容赋值。 3、myver 一个字节空间，返回版本号。 4、myver 一个字节空间，返回厂商代码号。	
返回		
返回值说明	0	复位 CPU 卡成功。
	8	寻卡错误，根本就没有卡在感应区，*serial 无效。
	50	RATS 错误，厂家调试代码，用户不需理会。
	51	PPS 错误，厂家调试代码，用户不需理会
	52	已进入了 14443-4 协议状态，可进行 CPU 卡功能所有操作了。
	53	CPU 卡功能通讯错误。

	其他	见函数返回代码表:
--	----	-----------

第二步：初始化 CPU 卡：

函数名：cpursinit

功能	初始化函数, (ctrlword 是否需要先清空卡, 不需要清空的话, 可以需输入卡密码, 卡密码长度) ctrlword_0 是否先清空卡, 分配空间 (字节数)	
VB 声明	Declare Function cpursinit Lib "OUR_MIFARE.dll" (ByVal ctrlword As Byte, ByVal key As Long, ByVal keylen As Byte, ByVal customsize As Long) As Byte	
输入	1、ctrlword 0 表示不清空已创建的文件, 1 表示将已创建的文件清除。 2、key 卡密码, (卡密码和文件没有任何关系, 卡密码只是用来清空卡, 读和写文件是无效的)。是指向下标个数 ≤ 16 的字节数组, 用于存放卡密码。 3、keylen 卡密码的长度。 4、customsize 设定应用空间的大小, 要 ≤ 6121 。	
返回	返回 unsigned char 值, 并将卡分配成可用区间为 customsize 个字节, 并可以在此空间内创建文件, 所有创建的文件大小总和不能大于 customsize。	
返回值说明	0	操作成功, 写卡数据有效。
	其他	见函数返回代码表:
例子及注释	请要本公司网站下载各种开发工具的例子源代码, 本函数的声明和输入参数格式可以直接套用, 谢谢使用	

第三步：创建文件

函数名：cpursfileadd

功能	创建好的文件不能删除, 只能重新初始化卡来清除所有文件,	
VB 声明	Declare Function cpursfileadd Lib "OUR_MIFARE.dll" (ByVal fileno As Byte, ByVal readonlykey As Long, ByVal readonlykeylen As Byte, ByVal writekey As Long, ByVal writekeylen As Byte, ByVal customsize As Long) As Byte	
输入	fileno 为文件号, 取值 0-5; readonlykey 指向存放只读密码的数组(\geq readonlykeylen 个 char 的密码数组); readonlykeylen 只读密码的长度; writekey 指向存放读写密码的数组(\geq writekeylen 个 char 的密码数组); writekeylen 读写密码长度 customsize 文件大小;	
返回	返回 unsigned char 值。	
返回值说明	0	操作成功,。
	60	当前文件号已经存在, 不能再次创建。
	其他	见函数返回代码表:

第四步：修改文件密码

函数名：cpursfilekeychg

功能	修改文件密码，修改后，请妥善保管，否则该文件将无法再用。	
VB 声明	Declare Function cpursfilekeychg Lib "OUR_MIFARE.dll" (ByVal fileno As Byte, ByVal keytype As Byte, ByVal oldkey As Long, ByVal oldkeylen As Byte, ByVal newkey As Long, ByVal newkeylen As Byte) As Byte	
输入	Fileno 为文件号，取值 0-5； Keytype 认证密码 + 更改密码类型，0 更改只读密码，1 更改读写密码，0 用只读密码认证，2 用读写密码认证； Oldkey 指向存放旧文件密码的数组(≥Oldkeylen 个 char 的密码数组)； Oldkeylen 旧密码长度 Newkey 指向存放新文件密码的数组(≥Newkeylen 个 char 的密码数组)； Newkeylen 新密码长度	
返回	返回 unsigned char 值。	
返回值说明	0	操作成功。
	其他	见函数返回代码表：

第五步：读写文件内容

一、读卡函数名：cpursfiledataread

功能	读文件数据。	
VB 声明	Declare Function cpursfiledataread Lib "OUR_MIFARE.dll" (ByVal fileno As Byte, ByVal keytype As Byte, ByVal key As Long, ByVal keylen As Byte, ByVal startaddr As Long, ByVal databuf As Long, ByVal datalen As Long) As Byte	
输入	Fileno 为文件号，取值 0-5； Keytype 认证密码类型，0 用只读密码认证，2 用读写密码认证； Key 指向存放文件密码的数组(≥Keylen 个 char 的密码数组)； Keylen 文件密码长度； Startaddr 文件起始地址，≥0 Databuf 是指向下标个数大于 Datalen 的字节数组，用于存放读取上来的数据， Datalen 本次读取的长度，≥1	
返回	返回 unsigned char 值。	
返回值说明	0	操作成功。
	其他	见函数返回代码表：

二、写卡函数名：cpursfiledatawrite

功能	读文件数据。	
VB 声明	Declare Function cpursfiledatawrite Lib "OUR_MIFARE.dll" (ByVal fileno As Byte, ByVal keytype As Byte, ByVal key As Long, ByVal keylen As Byte, ByVal startaddr As Long, ByVal databuf As Long, ByVal datalen As Long) As Byte	
输入	Fileno 为文件号，取值 0-5； Keytype 认证密码类型，0 用只读密码认证，2 用读写密码认证； Key 指向存放文件密码的数组(\geq Keylen 个 char 的密码数组)； Keylen 文件密码长度； Startaddr 文件起始地址， ≥ 0 Databuf 是指向下标个数大于 Datalen 的字节数组，用于存放要写入的数据， Datalen 本次写卡的长度， ≥ 1	
返回	返回 unsigned char 值。	
返回值说明	0	操作成功。
	其他	见函数返回代码表：

让读写器发出声音

函数名：pcdbeep

功能	让读写器发出声响。(暂不开放对指示灯的控制，如果有特殊要求请致电本公司)	
原始声明	unsigned char __stdcall pcdbEEP(unsigned long xms)	
输入	xms 为响声的时间长度，单位为 2 毫秒	
返回	返回 unsigned char 值。	
返回值说明	0	操作成功。
	22	动态库或驱动程序异常，解决方法是退出程序，拔出 IC 卡读写器，重装驱动程序再插上 IC 卡读写器重试，或者重新拷贝动态库 OUR_MIFARE.dll 到正确的位置。
	24	操作超时。可能是电脑中毒导致 USB 帧传递调度缓慢，或者是 IC 卡读写器有问题，解决方法是重启电脑或重新拔插 IC 卡读写器。
	27	USB 传输不稳定导致传输的字符不全。不需理会这个错误，因为基本上是不会出现这个错误的。
	28	USB 传输不稳定导致 CRC 校验错。不需理会这个错误，因为基本上是不会出现这个错误的。
	其他	未知错误
例子及注释	请要本公司网站下载各种开发工具的例子源代码，本函数的声明和输入参数格式可以直接套用，谢谢使用	

三、可作为软件加密狗，轻松实现自己软件的产权保护。

函数名：pcdgetdevicenum

功能	返回本读写器独一无二的设备编号，此编号固化在芯片中，并通过加密的方式传输。根据此编号可在本公司网站查询是否真正为本公司的质保产品。因为可以返回全球唯一的设备编号，所以只需增加少量的算法，本读写器也可作软件加密狗用。	
原始声明	unsigned char __stdcall pcdgetdevicenumber(unsigned char *devicenumber)	
输入	devicenumber 只需指向一个至少已分配了 4 个 char 空间的可写数组 unsigned char *指针， serial 的下标由 0 开始。因为此数组仅用于返回设备编号。	
返回	返回 unsigned char 值，并将寻到的卡的序列号传值到*serial 数组。	
返回值说明	0	操作成功，* devicenumber 数组中的数据有效。
	12	读取设备编号失败。
	9	有多张卡在感应区，寻卡过程中防冲突失败，*serial 无效。
	10	该卡可能已被休眠，无法选中，但卡序列号已被读出，*serial 数组中的数据有效。
	22	动态库或驱动程序异常，解决方法是退出程序，拔出 IC 卡读写器，重装驱动程序再插上 IC 卡读写器重试，或者重新拷贝动态库 OUR_MIFARE.dll 到正确的位置。
	24	操作超时。可能是电脑中毒导致 USB 帧传递调度缓慢，或者是 IC 卡读写器有问题，解决方法是重启电脑或重新拔插 IC 卡读写器。
	28	USB 传输不稳定导致 CRC 校验错。不需理会这个错误，因为基本上是不会出现这个错误的。
	其他	未知错误
例子及注释	请要本公司网站下载各种开发工具的例子源代码，本函数的声明和输入参数格式可以直接套用，谢谢使用	

函数返回代码表:

RetCode = 0 "操作成功"
 RetCode = 8 "请重新拿开卡后再放到感应区"
 RetCode = 50 "RATS 错误，厂家调试代码，用户不需理会"
 RetCode = 51 "PPS 错误，厂家调试代码，用户不需理会"
 RetCode = 52 "已进入 14443-4 协议状态，可进行 CPU 卡功能所有操作了"
 RetCode = 53 "CPU 卡功能通讯错误"
 RetCode = 54 "数据不足，需要接着发送未完成的数据至卡上"
 RetCode = 55 "发送 ACK 指令给卡，让卡接着发送数据回来"
 RetCode = 56 "清空根目录失败"
 RetCode = 57 "卡片不支持功能"
 RetCode = 58 "卡片初始化失败"
 RetCode = 59 "分配的空间不足"
 RetCode = 60 "本次操作的实体已存在"
 RetCode = 61 "无足够空间"
 RetCode = 62 "文件不存在"
 RetCode = 63 "权限不足,有可能是用只读密码认证，导致无法更改读写密码或无法写文件"
 RetCode = 64 "密码不存在，或密钥文件未创建"

RetCode = 65 "传送长度错误"

RetCode = 66 "Le 错误，即接收的数据长度指定过大"

RetCode = 67 "功能不支持或卡中无 MF 或卡片已锁定"

RetCode = 68 "密码认证错误次数过多，该密码已被锁死"

RetCode = 86 "更改后的密码长度必须和创建时的长度一致"

RetCode = 87 "应用目录不存在"

RetCode = 88 "应用文件不存在"

RetCode = 89 "文件号不能超过 5"

RetCode = 90 "读取文件时返回的长度不足，数据可能不正确"

RetCode = 91 "一次读文件的长度不能超过 255"

RetCode = 92 "一次写文件的长度不能超过 247"

RetCode >= 70 And RetCode <= 85

"密码错误，剩余次数为" & CStr(RetCode - 70) & "，如果为 0，该密码将锁死，无法再认证"

"未知错误"