

**M&W 非接触 IC 卡读写器**  
**ActiveX 控件函数说明文档**



深圳市明华澳汉电子科技有限公司

地址：深圳市福田区华强北上步工业区 202 栋南方大厦 569 室

电话：（086-755）83345003

传真：（086-755）26010111

邮编： 518028

摘    要    本文对 mifare 标准非接触 IC 卡读写器 ActiveX 控件函数使用进行了说明。

更新日期    2013-10-23

作    者    王宇杰

部    门    客服部

|        |                                |    |
|--------|--------------------------------|----|
| 1      | 概述.....                        | 4  |
| 1.1    | 运行环境.....                      | 4  |
| 1.2    | 硬件环境.....                      | 4  |
| 1.3    | ActiveX 控件说明.....              | 4  |
| 1.4    | 安装调试 ActiveX 控件.....           | 4  |
| 1.4.1  | 在 WEB 网页中调用 MWRFReader 控件..... | 4  |
| 1.4.2  | 通过 WEB 服务器进行调试.....            | 4  |
| 2      | 安全建议.....                      | 5  |
| 2.1    | 密钥保护措施.....                    | 5  |
| 2.2    | 数据防篡改.....                     | 5  |
| 3      | ActiveX 控件函数说明 .....           | 5  |
| 3.1    | 设备操作函数组.....                   | 5  |
| 3.1.1  | 复位 RF（射频）模块.....               | 5  |
| 3.1.2  | 打开读卡器.....                     | 6  |
| 3.1.3  | 关闭读卡器.....                     | 6  |
| 3.1.4  | 控制设备蜂鸣器.....                   | 6  |
| 3.2    | 射频卡操作函数组.....                  | 6  |
| 3.2.1  | 打开射频卡片.....                    | 6  |
| 3.2.2  | 关闭射频卡片.....                    | 6  |
| 3.2.3  | RF 读写器装载密码.....                | 6  |
| 3.2.4  | 验证卡片密码.....                    | 7  |
| 3.2.5  | 读取数据.....                      | 7  |
| 3.2.6  | 以 16 进制读数据 .....               | 7  |
| 3.2.7  | 写数据.....                       | 7  |
| 3.2.8  | 以 16 进制写数据 .....               | 7  |
| 3.2.9  | 初始化某一块的值.....                  | 8  |
| 3.2.10 | 读出指定值操作块的当前值 .....             | 8  |
| 3.2.11 | 对值操作的块进行增值操作 .....             | 8  |
| 3.2.12 | 对值操作的块进行减值操作 .....             | 8  |
| 3.2.13 | 改写密码 .....                     | 8  |
| 3.3    | 非接 CPU 操作函数.....               | 9  |
| 3.3.1  | CPU 卡复位.....                   | 9  |
| 3.3.2  | 向 CPU 卡发送命令.....               | 9  |
| 3.4    | 工具函数.....                      | 9  |
| 3.4.1  | 将 16 进制数转换为 ASCII 字符.....      | 9  |
| 3.4.2  | 将 ASCII 字符转换为 16 进制数.....      | 10 |

# 1 概述

## 1.1 运行环境

非接触 IC 卡读写器 ActiveX 控件是针对我公司非接触 IC 卡读写器产品开发的 ActiveX 控件，为 32 位 windows ActiveX 控件，适用于 Windows XP(sp3)及 以上系统，采用安装包形式，由基于 WEB 的二次开发者使用。

## 1.2 硬件环境

非接触 IC 卡读写器通过 PC 机串口或 USB 口与 PC 机通信，使用时根据读写器使用手册说明，将读写器与 PC 机相连。

## 1.3 ActiveX 控件说明

ActiveX 控件包含非接触 IC 读写器的设备操作函数、射频卡操作函数和智能卡函数组成，提供了 WEB 开发中的 ActiveX 控件函数接口，同时本文最后提供了用 JavaScript 脚本语言调用 MWRFReader 的方法，以及提供了常规读写器操作的示例。

ActiveX 控件的函数参数分为数字数据类型和字符串数据类型两种，由 WEB 页面中脚本语言数据类型是弱类型，因此在调用之前需要转换为正确的数据类型后传递才可以使用。在向读取或者写射频卡的操作函数中，涉及到的 16 进制数据全部以待操作数据的 16 进制字符表现形式传递，在二次开发时需要将字符形式表示的 16 进制串转换为真正的二进制数据。大部分函数返回一个状态码(WEB 中为字符串)，状态码含义如下：

==0 正确；

<0 错误；

说明：详细错误代码请参照读写器 API 函数错误代码。

## 1.4 安装调试 ActiveX 控件

### 1.4.1 在 WEB 网页中调用 MWRFReader 控件

代码请详见 MwReaderTest.htm 源码。

参数：

classid : ActiveX 控件的 CLASSID

Id : ActiveX 控件的实例变量，通过脚本语言访问 可以通过网页脚本语言 JavaScript 和 VBScript 引用变量 id 进行调用，执行 ActiveX 控件的读卡器功能。

### 1.4.2 通过 WEB 服务器进行调试

当所有二次开发的 WEB 页面完成之后，将 WEB 页面程序放置到 IIS 服务器

下，任何客户端机器都可以通过 IE 浏览器访问 IIS 服务器上的读卡器的资源文件。客户端必须设置 IE 的安全选项，否则不能下载控件。客户端 IE 浏览器自动下载控件后，自动安装并且注册控件，便可以访问读卡器设备了（操作说明详见 WEB 控件安装说明文档）。

## 2 安全建议

根据我公司多年从事 IC 卡行业项目经验，IC 卡交易安全性是系统设计的重点；系统安全分为 IC 卡密钥保护和数据防篡改。系统基于 WEB 模式，所有的网页脚本和交易数据都是开放。建议密钥保护和数据安全方法。

### 2.1 密钥保护措施

在网页中用脚本调用控件方法计算 IC 卡密钥，把 IC 密钥明文暴露在网页中，非常危险。基于 IC 卡密钥不能明文传递原则，网页脚本不能使用控件函数传递密钥。我公司设计密钥装载导入方法：

把 PSAM 卡计算 IC 卡密钥的流程，植入读写器主控制器中，算出的密钥值直接用于 IC 卡验证，不经过网页脚本导入设备，这样密钥不会被用户非法截获。

### 2.2 数据防篡改

数据在传输方式上有三种类型：明文方式、明文校验方式和密文校验方式。对以明文方式进行传输的数据由传输管理器直接送给命令处理模块。当数据以校验或密文校验方式传输时需要加密运算器对数据做处理。

在网络上进行卡交易我们应采用明文校验方式或密文校验方式。用户在网页上输入交易数据 ABC，先传送给服务器确认数据的合法性，服务器采用 DES 或 3DES 算法对数据增加校验码 MAC 后，回传终端，把数据 ABC+MAC 当作一个参数传递给控件校验，用 Psam 卡的密钥验证数据的合法性，再写入 IC 卡中。

具体的实现方法要通过双方讨论实现，本文没有实现控件方法。

## 3 ActiveX 控件函数说明

### 3.1 设备操作函数组

#### 3.1.1 复位 RF（射频）模块

函数：SHORT Reader\_reset(short \_msec)  
 参数：\_msec：复位时间（0~ 500ms）  
 返回值：=0 成功。

### 3.1.2 打开读卡器

函数: BSTR openReader(SHORT port, LONG baud)

参数:

port: 端口号, 0 表示端口 1, 1 表示端口 2, 以此类推

baud: 波特率

USB 口读写器参数可默认为 0, 9600;

返回值: 设备版本号, 长度为 18 个字节

### 3.1.3 关闭读卡器

函数: SHORT CloseReader(void)

返回值: =0 成功。

### 3.1.4 控制设备蜂鸣器

函数: SHORT ReaderBeep(SHORT m\_time)

参数: m\_time: 蜂鸣时间, 单位: 毫秒

返回值: =0 成功。

## 3.2 射频卡操作函数组

### 3.2.1 打开射频卡片

函数: BSTR openCard(SHORT mode)

参数: mode 寻卡模式

0: IDLE 模式, 一次只操作一张卡

1: ALL 模式, 一次可操作多张卡

返回值: 返回 4 字节的卡片序列号

### 3.2.2 关闭射频卡片

函数: SHORT CloseCard(void);

功能: 关闭卡片

返回值: =0 成功

### 3.2.3 RF 读写器装载密码

函数: SHORT Loadkey(SHORT mode, SHORT sector, BSTR key)

功能: 装载密钥

参数:

mode: 密码类型

0 — KEY A

4 — KEY B

sector: 须装载密码的扇区号(0~15)

key: 写入读写器的 12 字节新密码

例如 FFFFFFFFFFFF 共计 12 个字符, 表示 6 个 0xFF 字节数据

返回值: =0 成功

### 3.2.4 验证卡片密码

函数: SHORT mifare\_authentication(SHORT mode, SHORT sector)

功能: 验证卡片密码

参数:

mode: 验证密码类型:

0 – 用 KEY A 验证

4 – 用 KEY B 验证

sector: 将要验证的卡片扇区号(0~15)

返回值: =0 成功;

### 3.2.5 读取数据

函数: BSTR mifare\_read(SHORT block)

参数:

blocknr: 读取数据的块(0~63)

返回值: 返回读取的数据。

### 3.2.6 以 16 进制读数据

函数: BSTR mifare\_readHex(SHORT block)

参数:

blocknr: 读取数据的块(0~63)

返回值: 返回读取的数据。

说明: 卡片中某块存储的数据为 {0xa0, 0xb1, 0xc2, 0xd3, 0xe4, 0xf5}  
则读出的数据为字符串 “a0b1c2d3e4f5”

### 3.2.7 写数据

函数: SHORT mifare\_write(SHORT block, BSTR data\_buff)

参数:

block: 写入数据的块地址 (1~63)

data\_buff: 要写入数据, 长度<=16

返回值: =0 成功

### 3.2.8 以 16 进制写数据

函数: SHORT mifare\_writeHex(SHORT block, BSTR data\_buff)

参数:

block: 写入数据的块地址 (1~63)

`data_buff`: 要写入数据，长度必须为 32，如果长度不够 32，函数会失败  
返回值: =0 成功

说明: 要写入的字符串为 “a0b1c2d3e4f5”，该函数先将其转换为 16 进制再写入卡片，写入卡片的数据为 {0xa0, 0xb1, 0xc2, 0xd3, 0xe4, 0xf5}

### 3.2.9 初始化某一块的值

函数: `SHORT mifare_initValue(SHORT block, LONG value)`

参数:

`block`: 块地址

`value`: 初始化的值

返回值: =0 成功

### 3.2.10 读出指定值操作块的当前值

函数: `LONG mifare_value(SHORT block)`

参数:

`block`: 值操作的块地址

返回值: 返回操作块的当前值

### 3.2.11 对值操作的块进行增值操作

函数: `SHORT mifare_increment(SHORT block, LONG value)`

参 数:

`block`: 值操作的块地址(1-63)

`value`: 增加的值

返回值: =0 成功。

### 3.2.12 对值操作的块进行减值操作

函数: `SHORT mifare_decrement(SHORT block, LONG value)`

参 数:

`block`: 值操作的块地址

`value`: 减少的值

返回值: =0 成功。

### 3.2.13 改写密码

函数: `SHORT ChangeKey(SHORT sector, BSTR KeyA, SHORT B0, SHORT B1, SHORT B2, SHORT B3, BSTR KeyB)`

参数:

`sector`: 要改写密码的扇区

`KeyA`: A 密码

`B0` : 块 0 控制字，低 3 位 (D2D1D0) 对应 C10、C20、C30



B1 : 块 1 控制字, 低 3 位 (D2D1D0) 对应 C11、C21、C31  
 B2 : 块 2 控制字, 低 3 位 (D2D1D0) 对应 C12、C22、C32  
 B3 : 块 3 控制字, 低 3 位 (D2D1D0) 对应 C13、C23、C33  
 KeyB: B 密码

返回值: =0 成功。

例:

```
__int16 st=ChangeKey(0,"ffffffffffff",0,0,0,1,"ffffffffffff");
```

### 3.3 非接 CPU 操作函数

CPU 卡在操作前要先调用 openCard 函数打开卡片。

#### 3.3.1 CPU 卡复位

函数: BSTR NCpu\_Reset(void);

功能: 非接 CPU 卡复位

返回: 复位信息

#### 3.3.2 向 CPU 卡发送命令

函数: BSTR NCpu\_Send(BSTR sendCmd);

参数:

sendCmd: 发送的命令

返回: CPU 卡返回的应答数据

### 3.4 工具函数

#### 3.4.1 将 16 进制数转换为 ASCII 字符

函数: BSTR hexToAsc(BSTR hex, SHORT hexLen);

功 能: 将 16 进制数转换为 ASCII 字符。

参 数:

hex: 16 进制数

hexLen: 16 进制数的长度

返 回: 转换后的字符串

例如: {0x12, 0x34, 0x56} 转换后为字符串 “123456”

说明: 慎用此函数。如果要转换的一个字节的 16 进制数不在 ASC 码范围 (0x00~0x80) 内, 或两个字节的 16 进制数的组合不在 GBK 码范围 (0x8140~0xFEFE) 内, 转换会出错。如 {0xbe} 或 {0x81, 0x31} 都会转换出错。

### 3.4.2 将 ASCII 字符转换为 16 进制数

函数: BSTR ascToHex(BSTR asc, SHORT ascLen)

功能: 将 ASCII 字符转换为 16 进制数。

参 数:

asc: ASCII 字符

ascLen: ASCII 字符的长度

返 回: 转换后的16进制数

例如: 字符串 “12345678” 转换后为 {0x12, 0x34, 0x56, 0x78}

说明: 慎用此函数。如果两个字符组合成的 16 进制数不在 ASC 码范围 (0x00~0x80) 内, 或四个字符组成的两个 16 进制数的组合不在 GBK 码范围 (0x8140-0xFEFE) 内, 则转换会出错。如字符串 "be" 转换成 16 进制数为 {0xbe}, 或字符串 “8131” 转换成的 16 进制数 {0x81, 0x31} 都会出错。