

RF、EYE 系列
IC 卡读写器
ActiveX 控件函数说明文档
第 2.1 版



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

文档版本：2.1

控件版本：2.0.0.2

此版本在 1.0 基础上进行了大量修改,包括函数名称,参数个数及类型等,也修改了一些 BUG.故 1.0 版本与此版本存在不兼容情况.

作 者：王宇杰

部 门：客服部

生效日期：2015.6.25

注： 此控件只是在明华公司内部测试通过,但在客户的应用中可能出现我们没有考虑到的情况,或是控件在某些情况下出现一些 BUG,故不能保证所有用户使用此控件没有问题.如若出现问题,请及时联系我们.

目录

1 概述	5
1.1 运行环境	5
1.2 硬件环境	5
1.3 ActiveX 控件说明	5
1.4 安装调试 ActiveX 控件	6
1.4.1 控件安装注册	6
1.4.2 浏览器设置:	7
1.4.3 在 WEB 网页中调用 mwReaderRf 控件	8
1.4.4 通过 WEB 服务器进行调试	9
2 安全建议	9
2.1 密钥保护措施	9
2.2 数据防篡改	9
3 ActiveX 控件函数说明	10
3.1 设备操作函数	10
3.1.1 打开读卡器	10
3.1.2 关闭读卡器	10
3.1.3 控制设备蜂鸣器	11
3.1.4 加载密码	11
3.1.5 设置时间	11
3.1.6 设置 LED 显示控制方式	11
3.1.7 设置 LED 显示模式	12
3.1.8 设置 LED 显示数字	12
3.2 M1 卡操作函数	13
3.2.1 打开卡片	13
3.2.2 校验密码	13
3.2.3 校验密码	13
3.2.4 校验密码	14
3.2.5 读数据	14

3.2.6	以 16 进制字符串格式读数据.....	15
3.2.7	写数据.....	15
3.2.8	以 16 进制字符串格式写数据	15
3.2.9	初始化值	15
3.2.10	增值	16
3.2.11	减值	16
3.2.12	读值	16
3.2.13	更改密码.....	16
3.3	非接触式 CPU 卡操作函数	17
3.3.1	CPU 卡复位	17
3.3.2	向 Cpu 卡发送命令串	17
3.4	工具函数	17
3.4.1	将二进制数据转换为 16 进制字符串	17
3.4.2	将 16 进制字符串转换为二进制数据.....	18
4	错误代码	18

1 概述

1.1 运行环境

接触式 IC 卡读写器 **ActiveX** 控件是针对我公司 RF、EYE 系列非接触 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,由基于 WEB 的二次开发者使用.

1.2 硬件环境

RF、EYE 系列 IC 卡读写器通过 PC 机 USB 口或串口与 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 控件为 VC++ 开发的 MFC 控件,只能使用在 IE 内核的浏览器下.

ActiveX 控件包含接触 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** 脚本语言调用控件的方法,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,由于 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.在向读取或者写卡的操作函数中,涉及到的二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(**LastRet** 属性),状态码含义如下:

==0 正确

<>0 错误

1.4 安装调试 ActiveX 控件

1.4.1 控件安装注册

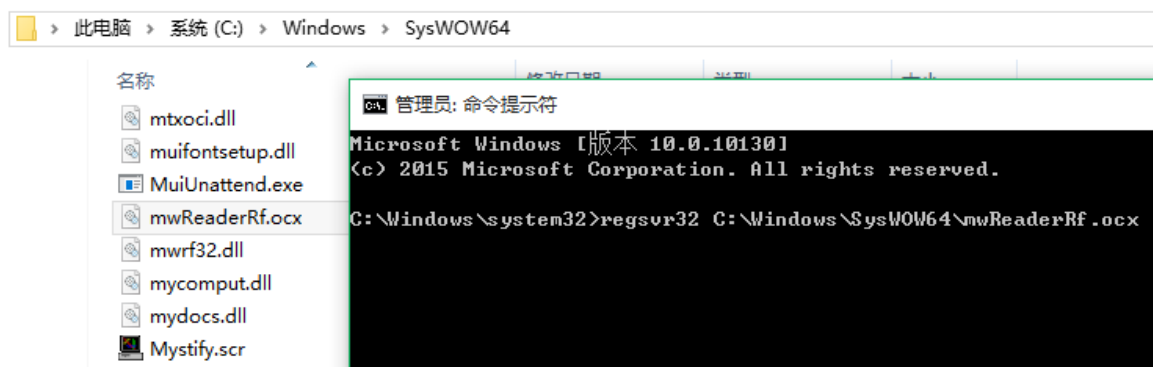
将我们提供的 mwReaderRf.ocx 及 mwrf32.dll 拷贝到系统文件夹下面并注册 mwReaderRf.ocx.

32 位系统文件夹位置：系统盘-->Windows-->system32

64 位系统文件夹位置：系统盘-->Windows-->SysWOW64

注册 mwReaderRf.ocx 方法：

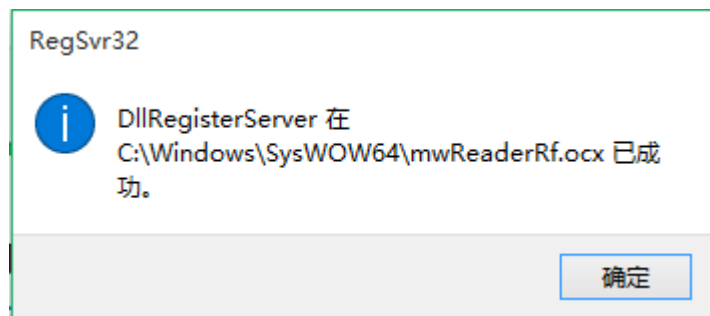
在开始菜单-->附件中找到"命令提示符", 点击右键-->以管理员身份运行, 在弹出的命令提示符对话框中输入 regsvr32, 然后加一个空格, 后面输入 mwReaderRf.ocx 的绝对路径, 完成后按回车键即可。如下图:(以下图片均为 64 位 win10 预览版 10130 截图)



按回车键后, 如果装有安全卫士, 会提示安全风险, 点击允许程序所有操作, 如下图:(360 安全卫士)



当操作完成后会弹出注册成功对话框,如下图:



注: 以上是我们提供的手动注册控件,二次 web 开发人员可以自己定制注册方法,如使用 .bat 批处理文件或是编写安装程序等方法自动注册控件。

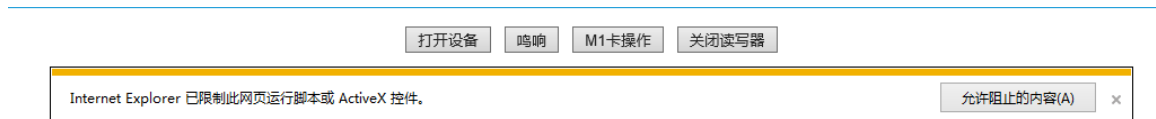
1.4.2 浏览器设置:

打开 IE 浏览器,点菜单工具—Internet 选项—安全—自定义级别,做以下设置:

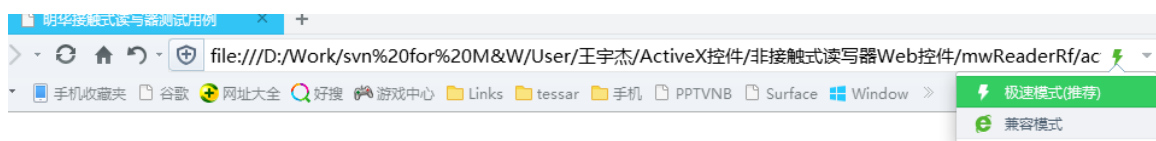
“对没有标记为安全的 ActiveX 控件进行初始化和脚本运行” 设置为“提示”;

“下载未签名的 ActiveX 控件” 设置为“提示”;

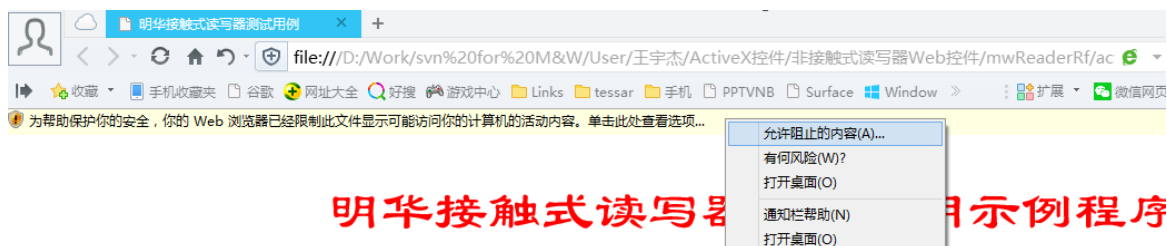
根据 IE 浏览器版本不同可能出现不同情况,建议将浏览器版本升级到最新版本 IE11. 在打开浏览器后,会在下面弹出提示"Internet Explorer 已限制此网页运行脚本或 ActiveX 控件",点击"允许阻止内容(A)"即可.如下图:



如果使用 360 浏览器,请选择兼容模式,如下图:



当选择兼容模式后,在最上方会弹出"为帮助保护你的安全,你的 Web 浏览器已经限制此文件显示可能访问你的计算机的活动内容。单击此处查看选项...",在上面点击右键,选择"允许阻止的内容(A)"即可。如下图:



1.4.3 在 WEB 网页中调用 mwReaderRf 控件

我们提供了两种调用方式:

1、使用 ActiveXObject 函数。

```
var MWRFATL =
new ActiveXObject("MWREADERRF.mwReaderRfCtrl.1");
```

参数:控件的对象名称。

具体代码请查看 htm 示例 activeXObjectTest.htm 源码。

2、使用<object> 标签。

```
<object id="MWRFATL" classid=
"CLSID: 856964B5-F42F-447B-A37D-ED07E8973ED2"></object>
```

参数:

id: mwReaderRf.ocx 控件的实例变量

classid: mwReaderRf.ocx 控件的 CLASSID

具体代码请参照 htm 示例 objectClassIdTest.htm 源码。

注: 我们提供的上述两种方法需要先安装注册控件。还有一种方法是制作 cab 包嵌入到网页中,这样网页会自动下载控件并注册,这种方法我们不提供,请自行查找相关技术文档。

1.4.4 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 RF、EYE 系列 IC 卡读写器的资源文件.客户端必须设置 IE 的安全选项,否则不能使用控件.客户端必须安装并且注册控件,才可以访问读卡器设备.

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管理办法.

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,由于我公司提供的控件为单步操作流程,会将密码或数据暴露在网页中,故还是希望有能力的开发工程师自己封装控件,将数据封装在函数内部,防止数据泄露。

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

控件只支持 S50、S70 卡片及标准非接触式 CPU 卡。除打开读写器函数返回版本号及功能函数外，其它都是将二进制数据转换成 16 进制字符串返回。写入的数据也要提前转换成 16 进制字符串写入。

属 性：LastRet

功 能：所有函数的执行结果。

返回值：==0 正确。

<>0 错误。

3.1 设备操作函数

3.1.1 打开读卡器

函 数：BSTR openReader(SHORT port, LONG baud);

功 能：打开读写器

参 数：USB 口读写器此参数无用，可设为 0, 9600

Port: 端口号，0 代表 com1, 1 代表 com2…。依此类推

Baud: 波特率

返回值：设备版本 ReaderInfo，长度为 18 个字节

说 明：使用读写器前调用此函数打开串口并连接读写器

3.1.2 关闭读卡器

函 数：void closeReader(void);

功 能：关闭读写器

说 明：该函数与打开读写器函数配套使用，用于关闭串口并断开读写器。

3.1.3 控制设备蜂鸣器

函 数: `void readerBeep(SHORT time);`

功 能: 让读写器鸣叫一段时间

参 数:

`time`: 蜂鸣时间,以 10 毫秒为一单位

3.1.4 加载密码

函 数: `void readerLoadKey(SHORT mode, SHORT sector, LPCTSTR key);`

功 能: 将卡片的密码装载到读写器中,用于密码验证。

参 数:

`mode`: 装载密码模式

0---装载 A 密码

4---装载 B 密码

`sector`: 要装载的扇区号

`key`: 密码

3.1.5 设置时间

函 数: `void readerSetTime(LPCTSTR time);`

功 能: 设置读写器的日期和时间

参 数:

`time`: 要设置的日期和时间。14 位 16 进制字符串,每两位为一单位,分别对应年月日星期时分秒。

例: "15080503122315"

15 年 8 月 5 日星期三 12 点 23 分 15 秒

3.1.6 设置 LED 显示控制方式

函 数: `void readerLedCtrlMode(SHORT mode);`

功 能: 设置 LED 显示控制方式

参 数:

mode: 0——数码管显示受计算机控制

1——数码管显示受读写器控制（默认设置）

3.1.7 设置 LED 显示模式

函 数: void readerLedShowMode(SHORT mode);

功 能: 设置读写器数码管显示模式, 关机后可保存设置值

参 数:

mode: 0—日期, 格式为年-月-日 (yy-mm-dd)

1—时间, 格式为时-分-秒 (hh-nn-ss)

3.1.8 设置 LED 显示数字

函 数: void readerLedShow(SHORT bePoint,
LPCTSTR numberData);

功 能: 设置 LED 显示数字字符串

参 数:

bePoint: 设置 numberData 中是否有小数点

0--无小数点

1--有小数点

numberData: 要显示的数字字符串

例: readerLedShow(0, "123"); //数码管显示 123

readerLedShow(1, "123.456"); //数码管显示 123.456

说明: 由于数码管不能显示灭状态, 故示例中实际显示的为
00000123 和 00123.456

3.2 M1 卡操作函数

3.2.1 打开卡片

函 数: `void openCard(SHORT openMode, SHORT cardMode);`

功 能: 打开卡片

参 数:

openMode: 打开模式

0---表示 IDLE 模式,只能寻卡一次

1---表示 ALL 模式,可多次寻卡

cardMode: 返回卡号模式

10---以 10 进制字符串返回

16---以 16 进制字符串返回

3.2.2 校验密码

函 数: `void cardVerifyPassword(SHORT mode,
SHORT sector);`

功 能: 验证卡片密码(加载密码认证,需要先将卡片密码用
readerLoadKey 加载到读写器中,S50 卡专用函数)

参 数:

mode: 验证密码模式

0---验证 A 密码

4---验证 B 密码

sector: 要验证的扇区号

3.2.3 校验密码

函 数: `void cardVerifyPassword2(SHORT mode,
SHORT keyAdr, SHORT block);`

功 能: 验证卡片密码(加载密码认证,需要先将卡片密码用
readerLoadKey 加载到读写器中,S70 卡专用函数)

参 数:

mode: 验证密码模式

0---验证 A 密码

4---验证 B 密码

keyAdr: 对应读写器中的扇区号

sector: 要验证的块号。验证成功后,此块所在的扇区中的任意块都可操作,不需要再次验证。

注: 此函数可选择读写器中加载的任意扇区的密码来验证卡片的密码, 如果读写器中选择的密码与卡片密码相匹配, 密码验证通过。主要用于验证扇区号大于 15 的扇区。

3.2.4 校验密码

函 数: void cardDirVerifyPassword(SHORT keyMode, SHORT block, LPCTSTR key);

功 能: 验证卡片密码(直接密码认证,不需要加载密码。EYE 读写器专用函数,RF 系列读写器无此功能, S50、S70 卡均适用)

参 数:

mode: 验证密码模式

0---验证 A 密码

4---验证 B 密码

block: 要验证的块号。验证成功后,此块所在的扇区中的任意块都可操作,不需要再次验证。

key: 密码

3.2.5 读数据

函 数: BSTR cardRead(SHORT block);

功 能: 读出指定块的数据

参 数:

block: 要读取数据的块号

返回值: 读取到的数据。

3.2.6 以 16 进制字符串格式读数据

函 数: `BSTR cardReadHex(SHORT block);`

功 能: 读出指定块的数据

参 数:

`block`: 要读取数据的块号

返回值: 读取到的数据,以 16 进制字符串格式显示。

3.2.7 写数据

函 数: `void cardWrite(SHORT block, LPCTSTR data);`

功 能: 向指定的块写数据

参 数:

`block`: 要写入数据的块号

`data`: 要写入的数据。只能写入 16 位(字节)数据,
汉字只能写入 8 个。

3.2.8 以 16 进制字符串格式写数据

函 数: `void cardWrite(SHORT block, LPCTSTR data);`

功 能: 向指定的块写数据

参 数:

`block`: 要写入数据的块号

`data`: 要写入的数据,只能写入 16 进制格式数据,且
长度必须为 32 位。

3.2.9 初始化值

函 数: `void cardInitValue(SHORT block, LONG value);`

功 能: 将指定的块初始化为值操作,将要初始化的值写入此块中

参 数:

`block`: 要初始化值的块号

`value`: 要初始化的值

3.2.10 增值

函 数: `void cardIncrementValue (SHORT block, LONG value);`

功 能: 向指定块中的值进行加值操作

参 数:

`block`: 要加值的块号

`value`: 要增加的值

3.2.11 减值

函 数: `void cardDecrementValue (SHORT block, LONG value);`

功 能: 向指定块中的值进行减值操作

参 数:

`block`: 要减值的块号

`value`: 要减少的值

3.2.12 读值

函 数: `LONG cardReadValue (SHORT block);`

功 能: 读出指定块中的值

参 数:

`block`: 要读值的块号

返 回: 读取到的值

3.2.13 更改密码

函 数: `cardChangeKey(SHORT sector, LPCTSTR keyA, LPCTSTR access, LPCTSTR keyB);`

功 能: 更改指定扇区的密码

参 数:

`sector`: 要更改密码的扇区

`keyA`: 密码 A

access: 存取控制,一般默认为"ff078069",可根据存取控制规则更改。

keyB: 密码 B

3.3 非接触式 CPU 卡操作函数

3.3.1 CPU 卡复位

函 数: BSTR cpuCardReset();

功 能: CPU 卡复位

返回值: 复位应答信息

3.3.2 向 Cpu 卡发送命令串

函 数: BSTR cpuCardSendCmd(BSTR send_data);

功 能: 对卡片发送命令串

参 数:

send_data: 发送给 Cpu 卡的命令串

返回值: CPU 卡返回的应答数据

3.4 工具函数

3.4.1 将二进制数据转换为 16 进制字符串

函 数: BSTR binToHexString(LPCTSTR binData);

功 能: 将二进制数据转换为 16 进制字符串

参 数:

binData: 要转换的二进制数据(asc 码)

返 回: 转换后的字符串

此函数可将二进制数据转换为 16 进制字符串,如可将 0x31 转换为字符串"31".

说明:慎用此函数。如果要转换的一个字节的二进制数不在 ASC 码范围(0x00~0x80)内,或两个字节的二进制数的组合不在 GBK 码范围(0x8140-0xFEFE)内,转换会出错。如 {0xbe}或

{0x81,0x31}都会转换出错。即函数可以执行成功,但返回的数据是错误的。

3.4.2 将 16 进制字符串转换为二进制数据

函 数: BSTR hexStringToBin(LPCTSTR hexString);

功 能: 将 16 进制字符串转换为二进制数据

参 数:

hexString: 要转换的 16 进制字符串

返 回: 转换后的二进制数据 (asc 码)

此函数可将 16 进制字符串转换为二进制数据,如可将字符串"31"转换为 0x31;

说明:慎用此函数如果两个字符组合成的二进制数不在 ASC 码范围 (0x00~0x80) 内,或四个字符组成的两个二进制数的组合不在 GBK 码范围 (0x8140-0xFEFE) 内,则转换会出错。如字符串 "be"转换成的二进制数为{0xbe},或字符串"8131"转换成的二进制数为{0x81,0x31},这两者都会出错。即函数可以执行成功,但返回的数据是错误的。

4 错误代码

函数错误类型代码

返回值	错误类型
0	正确
1	无卡
2	CRC 校验错
3	值溢出
4	未验证密码
5	奇偶校验错

6	通讯出错
8	错误的序列号
10	验证密码失败
11	接收的数据位错误
12	接收的数据字节错误
15	写失败
16	加值失败
17	减值失败
18	读失败
-0x10	PC 与读写器通讯错误
-0x11	通讯超时
-0x20	打开通信口失败
-0x24	串口已被占用
-0x30	地址格式错误
-0x31	该块数据不是值格式
-0x32	长度错误
-0x33	参数错误
-0x40	值操作失败
-0x50	卡中的值不够减