W2690 读写器用户手册

(V1.0)

北京握奇智能科技有限公司

目录

一 读写器的功能和性能介绍

W2690 多功能读写器支持 ISO14443 TypeA/TypeB 的非接触 CPU 卡和 Mifare one 卡，内置 4 个符合 ISO7816-3 的 PSAM 小 IC 卡，并且内嵌一个 ESAM 加密模块（选配），以及一个大的接触卡座，支持 USB 和串口两种通讯接口，并且支持无驱无软，操作简单，功能齐全，外观精美，性能稳定，质量可靠，适用于各种 IC 卡的应用系统。

1. 主要功能

- 支持符合 ISO14443 TypeA/TypeB 的非接触卡
- 支持 Mifare one 卡
- 支持大的接触卡（支持多速率通讯）
- 内置 4 个小 PSAM 卡座，支持符合 ISO7816 的 CPU 卡（支持多速率）
- 内嵌一个 ESAM 加密模块（具有 PK 卡的相关的加密算法，按照客户要求进行选配）
- 内嵌一个天线外，同时外部可以扩展出一个天线板（客户可以选配相应长度的同轴线）
- 支持串口通讯(USB 取电) （支持多速率）
- 支持全速 USB 通讯，速率 12Mbps
- 内置蜂鸣器，用户可以控制
- 3 个指示灯（蓝/红/绿）
- 读写器的固件程序可通过用户升级来增加新功能

2. 符合标准

- 〈非接触 IC 卡读写器技术规范〉
- 〈ISO14443-1/2/3/4〉
- 〈ISO7816-1/2/3/4〉
- 〈EIA-232-E 串口通讯标准〉
- 〈USB2.0 标准〉

二 主要技术指标

参数	指标
大卡座的接触卡通讯速率	9600/19200/38400/57600/115200bps
大卡座接触卡工作电压	3V、5V
小卡座的接触卡通讯速率	9600/19200/38400/57600/115200bps
小卡座接触卡工作电压	3V、5V
串口通讯	9600/19200/57600/115200bps
串口线	线长 1.2 米，DB9F/DB9F
工作电压	DC5V±0.5V
工作温度	$-20^{\circ}C$ ~ $+55^{\circ}C$
操作系统	Win 2003/ Win XP/ Win Vista/ Windows 7/ Windows 2008/ Win 8
USB 通讯	USB2.0 全速 12Mbps
USB 线	线长 1.2 米，A/B 接口
射频场强	0cm：6.5A/m　　5cm：2 A/m
谐振频率	13.56MHz±7kHz
平均无故障时间	5000 小时
驱动	无驱

三 指示灯和蜂鸣器的说明

1、IC 卡读写器指示灯与蜂鸣器的状态说明
红色灯：电源指示灯
蓝、绿色灯：工作指示灯，可以通过指令控制指示灯的状态
蜂鸣器：声音指示信号，在上电自检时响一声，可以通过指令控制开关进行控制
根据通讯方式的不同，LED 的状态定义有所区别
2、IC 卡读写器指示灯与蜂鸣器的控制说明
支持此命令的卡座为NAD=00
B017002301+data：data定义如下：
bit6 绿灯控制位 1 on 0 off
bit4 控制有效位 1 on 0 off
bit3 蓝灯控制位 1 on 0 off

四 读写器通讯协议

本通讯协议指的是 IC 卡读写器与上位机之间数据传输的格式，用户也可以按照此格式，通过不同的系统与 IC 卡读写器进行通讯连接。总体来说，该通讯协议就是在 ISO7816 协议的 APDU 指令基础上，在头尾各增加相应的数据，以保证通信数据的完整和正确性。

特别说明：本手册里与命令相关的的数字默认为十六进制。

1．发送到读写器的命令格式：

信息域	标识	字节长度	含义
通信数据头	NAD	1	0x12：大卡用户卡座 0x13：ESAM 加密模块 0x15/0x25：非接触卡 0x16：小 SAM 卡座 0x17：小 SAM 卡座 0x18：小 SAM 卡座 0x19：小 SAM 卡座
	PCB	1	HID 口：包编号 串口：LEN 的高位
	LEN	1	数据长度，包括 CLA INS P1 P2 Lc DATA
APDU 指令	CLA	1	指令类型
	INS	1	指令码
	P1	1	指令参数 1
	P2	1	指令参数 2
	Lc	1	输入数据长度或期望返回数据长度
	DATA	0-FF	输入数据
校验	XOR	1	XOR 校验值(从 NAD 开始的所有数据做异或计算的结果)

字节			USB 模式下不存在

例 1：CPU 卡选择主文件（3F00）命令

12 00 07 00 A4 00 00 02 3F 00 8C
↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
NAD PCB LEN CLA INS P1 P2 Lc (D A T A) XOR（串口存在，USB 口不存在）

例 2：CPU 卡复位命令

12 00 05 00 12 00 00 00 05
↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
NAD PCB LEN CLA INS P1 P2 Lc XOR（串口存在，USB 口不存在）

2.从读写器返回信息的格式

标识	字节长度	含义
NAD	1	发送命令 NAD 的半字节互换 例如发送 NAD=12，返回 NAD=21
PCB	1	默认为 00
LEN	1	数据长度，包括 DATA SW1 SW2
DATA	0-FF	返回数据
SW1	1	状态字节 1
SW2	1	状态字节 2
XOR	1	XOR 校验值(从 NAD 开始的所有数据做异或计算的结果) USB 模式下不存在

例 1 返回信息如下：

21 00 02 61 XX XOR
↓ ↓ ↓ ↓ ↓ ↓
NAD PCB LEN SW1 SW2 XOR（USB 时不存在）

例 2 返回信息如下

21 00 13 3B6D000057444624A864341301F131C12 90 00 7F
↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
NAD PCB LEN (←－－－－－－DATA－－－－－－→) SW1 SW2 XOR（串口存在，USB 口不存在）

其中 90 00 是读写器自动补上状态字节 SW1 SW2

五 动态库的说明

适应操作系统：

无限制

适用的 IC 卡：

*符合 7816 的 CPU 卡（T=0&T=1(3V/5V)）
*符合 14443 的 CPU 卡（Type A& Type B）

1、读写器及 CPU 卡读写接口函数

此部分是针对 CPU 卡的操作函数，推荐的函数调用顺序

CT_open	打开设备获得设备句柄
ICC_set_NAD	设置卡座 NAD，默认值为 00
ICC_reset	对 IC 卡复位
…	
ICC_tsi_api	对 IC 卡读写器进行操作
...	
CT_close	关闭打开设备的连接

1）打开 IC 卡终端端口

```
HANDLE WINAPI CT_open(
    char *name,
    unsigned int param1,
    unsigned char param2
    );
```

参数：

（A）串口读写器

name：读写器与 PC 相连的端口，可取 COM1 COM2 COM3 COM4 等

param1：波特率，可取 9600、19200、38400、57600 或 115200

param2：奇偶校验，仅仅支持"N"无校验

（B）USB 读写器

name: 可取"hid1","hid2","hidN"等,1<=N<=9

序号分配原则，有以下情况:

i) 系统上所有 N 个设备均处于断开状态，则调用"hid1","hid2",...,"hidN"依次打开设备，跟插入顺序无关。

ii) 系统上存在 N 个已建立连接的设备，插上一新设备并打开此设备，则需从"hid1"～"hid(N+1)"依次调用。

param1: 未使用,设为 0

param2: 未使用,设为 0 返回值：

INVALID_HANDLE_VALUE(-1)表示打开端口失败；

其他值(大于 0)为打开的端口句柄，用于卡操作函数的 fd

示例：

```
//以下的程序为以 9600 波特率，无校验方式打开与 COM1 口连接的串口读写器
HANDLE fDev;                //定义句柄，用于保存端口句柄
char devName[5];            //用于保存端口名称
strcpy(devName, "COM1");        //获得端口名称
fDev = CT_open(devName ,9600, 'N');   //以相应的格式打开端口，并得到端口句柄
if(fDev == INVALID_HANDLE_VALUE)       //判断端口是否正确打开
{
  MessageBox("Open Device Error!");
  return;
}
```

2）关闭与 IC 卡读写器相连的端口（必须用 CT_open 打开的端口）

```
int WINAPI CT_close(
    HANDLE fd
    );
```

参数：

fd ： 为函数 CT_open 所返回的句柄

返回值：

-1：失败 0：成功

示例：

```
//以下示例为关闭以 CT_open()函数打开的读写器
int ret;       //定义 int 变量用于保存关闭函数返回值
ret = CT_close( fDev );        //关闭端口，并获得结果
if(ret == -1)       //判断端口是否正确关闭
{
  MessageBox("Close Deivce Error");
}
```

3）对设备当前激活插槽中的 IC 卡进行复位

```
unsigned WINAPI ICC_reset(
    HANDLE fd,
    unsigned char *lenr,
    unsigned char *resp
    );
```

参数：

fd ： 已打开的端口描述符

lenr ： 为对 IC 卡复位所返回的复位数据的长度

resp ： 复位的数据结果

返回值：

0x9000 成功

0x6200 无卡

0x6201 协议不认识

0x6FF0 卡通讯失败或其它未知的错误

0xFFFF 通迅失败

说明：

此函数受 ICC_set_NAD 函数影响，对 W2690 读写器可设置为 12 与 13、15、16、17、18、19

示例：

```
//以下示例，为对 CPU 卡片进行复位
unsigned int sw;          //定义变量分别用于保存返回状态值
unsigned int lenr, resp[256];   //定义变量分别用于保存返回数据长度及数据
ICC_set_NAD(fDev,0x12); //设置NAD为0x12
sw = ICC_reset(fDev, &lenr, resp);      //执行复位指令
if(sw != 0x9000)          //判断是否执行成功
{
    … //操作失败后，用户的处理
}
```

4）从外设向 CPU 卡或读写器发送 APDU 命令并接收应答

comm的结构：CLA INS P1 P2 Lc DATA [Le] 其中DATA长度为Lc字节

resp 的结构：DATA 其中 DATA 长度为 Le 字节

```
unsigned WINAPI ICC_tsi_api(
    HANDLE fd,
    unsigned char len,
    unsigned char *comm,
    unsigned char *lenr,
    unsigned char *resp
    );
```

参数：

fd	：	已打开的端口描述符
len	：	命令 comm 的长度
comm	：	发向卡上的命令
lenr	：	从卡上接收到的数据长度
resp	：	从卡上接收到的数据

返回值：

0xFFFF 通迅失败（发送命令或接收返回的数据失败）

0x6FF0 卡通讯失败或其它未知的错误

其它为从卡上返回的状态 SW1 SW2

说明：

该函数适用于所有 CPU 卡操作，一次最多可读 255 个字节，写 255 个字节，并且切换串口波特率的指令只可以在此函数中实现

示例：

```
//以下示例为发送取版本号指令并显示
unsigned char lens;        //定义发送的数据长度变量
unsigned char lenr;        //定义保存返回数据长度变量
unsigned char comm[300];   //定义发送指令数组
```

```
unsigned char resp[300];        //定义接收数据数组
unsigned int sw;              //定义变量用于保存返回状态值
char tmpbuf[300];       //定义变量用于保存转化为字符型值的返回值
CString strDisplay;          //定义变量用于显示
ICC_set_NAD(fDev,0x12);     //设置NAD为0x12
memcpy(comm,"\x00\x19\x00\x00\x00",5);      //设置十六进制的指令
lens=5;                  //设置指令长度为5
sw=ICC_tsi_api(fDev,lens,comm,&lenr,resp);      //发送指令并取得返回值
if(sw!=0x9000)      //对指令执行是否成功进行判断
{
  MessageBox("Get Reader Firmware Version Error!");
}else
{
  BinToCHex((unsigned char *)tmpbuf,resp,lenr);          //将返回值转换
为字符以供显示
  tmpbuf[lenr*2]=0;
  strDisplay=CString(tmpbuf);
  MessageBox("Firmware Version is "+strDisplay);
}
```

5）从外设向 T=0 的 CPU 卡发送 APDU 命令并接收应答

```
unsigned WINAPI ICC_tsi_apiT0(
    HANDLE fd,
    unsigned int len,
    unsigned char *comm,
    unsigned int *lenr,
    unsigned char *resp
    );
```

参数：

fd	：	已打开的端口描述符
len	：	命令 comm 的长度
comm	：	发向卡上的命令
lenr	：	从卡上接收到的数据长度
resp	：	从卡上接收到的数据

返回值：

0xFFFF 通迅失败（发送命令或接收返回的数据失败）

0x6FF0 卡通讯失败或其它未知的错误

其它为从卡上返回的状态 SW1 SW2

说明：

该函数只适用于在 W2690 读写器读写 T=0 的支持 256 字节读写的 CPU 卡，使用该函数一次可读多达 256 字节（len=0），写多达 255 个字节的数据

示例：

同上，写入数据和返回值可达 255 字节

6）设置 CPU 卡读写地址 NAD

```
void WINAPI ICC_set_NAD(
    HANDLE fd,
    unsigned char nad
    );
```

参数：

fd ： 已打开的端口描述符

nad ： 读写地址

返回值：

无

说明：

该函数适用于 W2690 读写器，系统缺省值为 00

00 对主卡操作

12 对读写器或主卡操作

13 选择读写器的 ESAM 卡操作

……

示例：

```
ICC_set_NAD(fDev,0x12);    //设置NAD为0x12
```

7）检查读写器的主卡座否插入 IC 卡

```
unsigned WINAPI ICC_present(
    HANDLE fd
    );
```

参数：

fd ： 已打开的端口描述符

返回值：

0x9000 已插入 IC 卡

0x6200 没有插入卡或卡没插到位

示例：

```
//此例程为检查卡片是否存在
unsigned int sw;          //定义返回状态变量
sw = ICC_present(fDev); //检查是否插入卡操作
if(sw != 0x9000)          //判断是否执行成功
{
    … //操作失败后，用户的处理
}
```

8）写 CPU 卡的二进制文件

```
unsigned WINAPI ICC_write_file(
    HANDLE fd,
    unsigned int offset,
    unsigned int len,
    unsigned char *data
    );
```

参数：

fd ： 已打开的端口描述符

offset ： 二进制文件的偏移量
len ： 要写入卡上的数据长度
data ： 要写入卡上的数据

返回值：
0xFFFF 通迅失败（发送命令或接收返回的数据失败）
0x6FF0 卡通讯失败或其它未知的错误
其它为从卡上返回的状态 SW1 SW2

说明：
该函数适用于所有 CPU 卡操作，用户必须先选择要操作的二进制文件

示例：

```
//此示例为向 CPU 卡的二进制文件写入 3 个数据，请先对 CPU 卡片复位后，选择相应二进制文件
unsigned int offset = 0;          //定义写入数据地址的偏移量变量,并赋初值
unsigned int len;              //定义写入数据长度的变量
unsigned char data[3]={0, 1, 2};     //定义写入数据变量，并赋初值
len = 3;               //写入数据的长度为 3 字节
sw = ICC_write_file(fDev, offset, len, data);   //写入二进制文件 3 字节数据操作
if(sw != 0x9000)          //判断是否执行成功
{
    …  //操作失败后，用户的处理
}
```

9）读 CPU 卡的二进制文件

```
unsigned WINAPI ICC_read_file(
    HANDLE fd,
    unsigned int offset,
    unsigned int len,
    unsigned char *data
    );
```

参数：
fd ： 已打开的端口描述符
offset ： 二进制文件的偏移量
len ： 要读卡上的数据长度
data ： 要读卡上的数据

返回值：
0xFFFF 通迅失败（发送命令或接收返回的数据失败）
0x6FF0 卡通讯失败或其它未知的错误
其它为从卡上返回的状态 SW1 SW2

说明：
该函数适用于所有 CPU 卡操作，用户必须先选择要操作的二进制文件

示例：

```
//此示例为向 CPU 卡的二进制文件读取 3 个数据，请先对 CPU 卡片复位后，选择相应二进制文件
```

```
    unsigned int offset = 0;        //定义读取数据地址的偏移量变量,并赋初值
    unsigned int len;           //定义读取数据长度的变量
    unsigned char data[256];    //定义读取数据变量
    len = 3;            //读取数据的长度为 3 字节
    sw = ICC_read_file (fDev, offset, len, data);  //读取二进制文件 3 字节
数据操作
    if(sw != 0x9000)        //判断是否执行成功
    {
        …  //操作失败后，用户的处理
    }
```

六 W2690 读写器自定义指令

1. W2690 读写器自定义指令说明.

W2690 读写器指令表							
NAD	CLA	INS	P1	P2	Lc	Data	功能说明
00	B0	19	00	00	00		取读写器版本号
00	B0	16	RATE	00	00		修改读写器与 PC 间的串口通讯速率： 00:9600bps 01:19200bps 02:38400 bps 03:57600 bps 04:115200 bps 其它无效
	80	F2	P1	P2	00		蜂鸣器鸣叫，P1 P2 鸣叫时间参数，秒数（P1 为高位，P2 为低位）
	B0	17	00	23	01	data	LED 灯控制。Data: bit6 绿灯控制位：1 on 0 off bit4 控制有效位：1 on 0 off bit3 蓝灯控制位：1 on 0 off
	00	19	01	01	Le		读设备序列号，Le 的长度小于 20
12/16/17/18/19	00	12	ETU	00	00		接触卡 ETU=00 默认 9600 ETU!=0 指令通讯速率复位, 固定可选值: B5: 9600bps 58: 19200bps 29: 38400bps 1B: 57600bps 0B: 115200bps 上述设置后，掉电失效
12/16/17/18/19	B0	16	ETU	00	00		接触卡 ETU=00 默认 9600 ETU!=0 指令通讯速率复位, 固定可选值: B5: 9600bps 58: 19200bps 29: 38400bps 1B: 57600bps 0B: 115200bps 上述设置后，掉电仍然有效
15/25	80	11	P1	P2	Lc	data	参见<非接触卡的指令使用说明>

2．读写器通用指令返回状态 SW1SW2 的含义

9000　操作成功

6FF0 卡通讯失败或其它未知的错误

6D00 不识别的读写器命令

6200/6201　指定操作的卡不存在

70/75XX 非接触卡复位失败

76XX 通讯失败

FFFF(-1)　PC 与读写器通讯失败

七 W2690 读写器非接触卡指令

1. 读写器可以自动识别 TypeA 和 TypeB 非接触卡片，以及标准的 Mifare One 卡片。
2. 标准非接触的 CPU 卡使用流程：
   a) 设置 nad 为 0x15/0x25
   b) 发送 cpu 卡的命令: CLA INS P1 P2 LC DATA
3. Mifare One 卡片的操作

命令说明表：

CLA	INS	P1	P2	LC	COMMAND	命令描述
80	11	09	00	00	----------	按 mifare 1 复位卡片
80	11	02	00	08	密钥类型（1 字节，60：keyA；61：keyB）+待认证的块号（1 字节）+密钥值（6 字节）	利用指令中给定的密钥进行认证
80	11	03	00	01	经过认证的块号（1 字节）	读 mifare 卡
80	11	04	00	11	经过认证的块号（1 字节）+数据字节（16 字节）	写 mifare 卡
80	11	05	00	05	经过认证的块号（1 字节）+金额数（4 字节）	将 Mifare 1 卡的块初始化为其规定的钱包形式。 注：4 字节的金额数为低位在前，高位在后。
80	11	06	00	05	经过认证的块号（1 字节）+金额数（4 字节）	向 mifare 1 卡的钱包块内添加金额 注：4 字节的金额数为低位在前，高位在后。
80	11	07	00	05	经过认证的块号（1 字节）+金额数（4 字节）	在 Mifare 1 卡的钱包块中扣除给定的金额 注：4 字节的金额数为低位在前，高位在后。

注意：对 Mifare One 卡进行块读写等操作时，只要换块操作，就要在块操作之前对该块进行复位操作（即发送 8011090000），然后再验证块密码，以及读写等操作。错误码说明：

9000 正确
6982 没有经过认证
6983/6984 认证失败
6ff0 初始化失败
6ff3 写数据错误
6ff5 超时
6ff8 返回数据长度错误

八 注意事项

1）因为读写器工作频率为13.56MHz，所以在读写器安装现场不得有13MHz～15MHz之间强电磁场。
2）为了防止读写器发射磁场的相互影响，2台读写器的安装距离应大于10CM。
3）金属平面对电磁波有反射和屏蔽作用，因此读写器应尽量避免放置或安装在金属平面上。

九 声明

此产品为 A 级产品，在生活环境中，该产品可能会造成无线电干扰。在这种情况下，可能需要用户对其干扰采取切实可行的措施。