



关键词

- MUR-500 读卡器
- Mifare 1 S50, S70
- 扇区 密钥
- 一卡多用 一卡通
- 非接触式 智能卡
- USB

摘要

MUR-500 读卡器是 USB 接口的基于 13.56MHz 频率的 Mifare 卡读写设备，符合 ISO14443A 标准，可支持 Mifare1 S50、Mifare1 S70、Mifare Light、Mifare UltraLight、华虹 SHC1102。

本文详细介绍了 MUR-500 的使用方法。

目 录

| | |
|-------------------------------------|----|
| 1. MUR-500 简介 | 3 |
| 1.1 技术参数..... | 3 |
| 1.2 上电及状态指示说明..... | 3 |
| 1.2.1 上电状态..... | 3 |
| 1.2.2 红灯指示..... | 3 |
| 1.2.3 绿灯 | 3 |
| 1.2.4 蜂鸣器..... | 3 |
| 2. MUR-500 的函数 | 4 |
| 2.1 新版函数列表..... | 4 |
| 2.2 旧版函数列表..... | 4 |
| 2.3 函数返回状态值列表..... | 5 |
| 2.4 函数描述..... | 6 |
| 2.4.1 装载密钥—Load_Key | 6 |
| 2.4.2 请求—Request..... | 6 |
| 2.4.3 带层级设置的防碰撞—Casc_Anticoll | 7 |
| 2.4.4 带层级设置的选择—Casc_Select | 8 |
| 2.4.5 证实 2—Auth2..... | 9 |
| 2.4.6 直接密码证实—Auth_Key | 9 |
| 2.4.7 写—Write..... | 9 |
| 2.4.8 写值块操作—Write_Val..... | 10 |
| 2.4.9 读—Read | 10 |
| 2.4.10 读值块操作—Read_Val | 11 |
| 2.4.11 带内部自动传送的值操作—Value..... | 11 |
| 2.4.12 写 UltraLight—Ul_Write | 11 |
| 2.4.13 挂起—Halt..... | 11 |
| 2.4.14 置位控制位—Green_Off..... | 12 |
| 2.4.15 清除控制位—Green_On | 12 |
| 2.4.16 输出信号—Alarm..... | 12 |
| 3. Mifare 卡工作状态介绍..... | 13 |
| 4. 修订历史..... | 14 |
| 5. 销售信息..... | 14 |

1. MUR-500 简介

1.1 技术参数

表 1 MUR-500 技术参数表

| | |
|-------|---|
| 产品型号 | MUR-500 |
| 功 耗 | 90 毫安/直流 5V; |
| 工作频率 | 13.56 兆赫兹 |
| 读卡距离 | 50~100 毫米 (mifare1 卡) |
| 接口方式 | USB |
| 支持卡类型 | mifare1 S50、mifare1 S70、mifare UltraLight、Mifare Light 、SHC1102 |
| 尺 寸 | 124mm (长) ×97mm (宽) ×31mm (高) |

1.2 上电及状态指示说明

1.2.1 上电状态

通过 A-B 的 USB 将 MUR-500 连接到 PC 机，MUR-500 即可上电，可通过观察读卡器上 LED 显示，判断读卡器所处状态。

正常情况下，上电时红灯和绿灯同时点亮，约 0.5 秒后，绿灯灭，红灯继续亮，读卡器进入工作状态。可启动 PC 机程序对其进行操作；读卡器上电稳定后，红、绿灯的任何其它可能的状态组合均为故障。

1.2.2 红灯指示

红灯用来指示 Config()函数的执行状态，若该函数执行成功，则红灯亮。若执行了 Close()函数，则红灯灭。

1.2.3 绿灯

有四种情况可能改变绿灯的状态：

1、执行 Config()函数和 Close()函数

若 Config()函数执行成功，则绿灯灭，否则绿灯亮。若读卡器执行了 Close() 函数，则绿灯灭。

读卡器上电时，将自动执行 Config()函数，因此读卡器稳定后，红灯亮、绿灯灭。

2、当读卡器上电进入正常工作状态后（此时红灯亮、绿灯灭），在 PC 机未发出任何命令之前，若读卡器感应区内有卡，绿灯将亮，否则绿灯灭。在 PC 机已发出任何一条命令后，此功能消失。因此在未启动 PC 机程序之前，本读卡器可以指示感应区内是否有卡。

3、执行 Green_Off()函数将熄灭绿灯，执行 Green_On()函数将点亮绿灯。

4、执行 Alarm()函数将可使绿灯闪烁。

1.2.4 蜂鸣器

执行 Alarm ()函数可使直流蜂鸣器发声，可控制蜂鸣器发声的持续时间、间歇时间及重复次数。

2. MUR-500 的函数

2.1 新版函数列表

表 2 新版函数列表

| 函数名称 | 功能描述 |
|-------------------|--|
| TX2_Load_Key | 改变存贮在 MUR-500 内密钥区中的密钥 |
| TX2_Request | 请求卡，检查在有效范围内是否有卡 |
| TX2_Casc_Anticoll | 可实现三层防碰撞协议 |
| TX2_Casc_Select | 可实现三层选择 |
| TX2_Auth2 | 使用 MUR-500 内部密钥区中的密码对指定的卡的扇区 Sector 进行验证 |
| TX2_Auth_Key | 直接密码验证 |
| TX2_Write_Enc | 向卡中指定块写入一 16 字节的数据块 |
| TX2_ULwrite_Enc | 向 UltraLight 卡中相应块写入 4 字节数据 |
| TX2_Write_Val | 向卡中指定块写入值块 |
| TX2_Read_Val | 读卡中指定块的值块 |
| TX2_Read_Enc | 从卡中指定块中读出一个 16 字节的块 |
| TX2_Value | 包含加、减、恢复函数，并可进行不同块之间的自动传送 |
| TX2_Init | 发卡器初始化 |
| TX2_Alarm | 输出驱动蜂鸣器、绿灯信号，能控制动作时间、间隙时间和重复次数 |
| TX2_Halt | 挂起当前卡片 |
| TX2_Green_Off | 将控制位置为高电平 |
| TX2_Green_On | 将控制位置为低电平 |

表 2 为新版函数列表，参数和返回值统一用 UINT（unsigned int）或者 字符串(char*) 传递，更加方便接口。新版函数在 MUR500USB.h 中声明。 **强烈建议采用新版函数。**

所有函数的返回值都是该函数执行后的状态结果，具体返回值请查看各函数说明。如果各函数有返回数据，则都是以指针的形式返回。

2.2 旧版函数列表

表 3 旧版函数列表

| 函数名称 | 功能描述 |
|------------------|--|
| TX_Config | 复位且配置 MUR-500 |
| TX_Close | 关闭 MUR-500 |
| TX_Get_Info | 读取 RC500 序列号 |
| TX_Load_Key | 改变存贮在 MUR-500 内密钥区中的密钥 |
| TX_Request | 请求卡，检查在有效范围内是否有卡 |
| TX_Casc_Anticoll | 可实现三层防碰撞协议 |
| TX_Casc_Select | 可实现三层选择 |
| TX_Auth2 | 使用 MUR-500 内部密钥区中的密码对指定的卡的扇区 Sector 进行验证 |
| TX_Auth_Key | 直接密码验证 |
| TX_Shc1102_Auth | Shc1102 卡验证 |
| TX_Write_Enc | 向卡中指定块写入一 16 字节的数据块 |
| TX_ULwrite_Enc | 向 UltraLight 卡中相应块写入 4 字节数据 |
| TX_Shc1102_Write | 往 SHC1102 卡指定块写入数据 |
| TX_Write_Val | 向卡中指定块写入值块 |

| | |
|-----------------|--------------------------------|
| TX_Read_Val | 读卡中指定块的值块 |
| TX_Check_Write | 将传送的数据和上一次所写数据比较 |
| TX_Read_Enc | 从卡中指定块中读出一个 16 字节的块 |
| TX_Shc1102_Read | 读 SHC1102 卡指定块的数据 |
| TX_Value | 包含加、减、恢复函数，并可进行不同块之间的自动传送 |
| TX_Read_E2 | 读 MUR-500 内 EEPROM 的内容 |
| TX_Write_E2 | 写数据到 MUR-500 内 EEPROM |
| TX_Green_Off | 将控制位置为高电平 |
| TX_Green_On | 将控制位置为低电平 |
| TX_Alarm | 输出驱动蜂鸣器、绿灯信号，能控制动作时间、间隙时间和重复次数 |

表 3 为旧版本函数列表，只是为了兼容之前的应用程序而保留。旧版函数在 mur500_old.h 中声明。

2.3 函数返回状态值列表

表 4 MUR-500 库函数的用到的状态值列表

| 名称 | 值 | 描述 |
|-----------------|------|-----------------------|
| OK, COMM_OK | 0 | 函数调用成功 |
| NO_TAG_ERR | -1 | 在有效区域内没有卡 |
| CRC_ERR | -2 | 从卡中接收到了错误的 CRC 校验和 |
| EMPTY | -3 | 值溢出 |
| AUTH_ERR | -4 | 不能验证 |
| PARITY_ERR | -5 | 从卡中接收到了错误的校验位 |
| CODE_ERR | -6 | 通信错误 |
| SENDER_ERR | -8 | 在防冲突时读到了错误的串行码 |
| KEY_ERR | -9 | 证实密码错 |
| NOT_AUTH_ERR | -10 | 卡没有验证 |
| BIT_COUNT_ERR | -11 | 从卡中接收到了错误数量的位 |
| BYTE_COUNT_ERR | -12 | 从卡中接收了错误数量的字节 |
| TRANS_ERR | -14 | 调用 Transfer 函数出错 |
| WRITE_ERR | -15 | 调用 Write 函数出错 |
| INCR_ERR | -16 | 调用 Increment 函数出错 |
| DECR_ERR | -17 | 调用 Decrment 函数出错 |
| READ_ERR | -18 | 调用 Read 函数出错 |
| COLL_ERR | -24 | 冲突错 |
| ACCESS_TIMEOUT | -27 | 访问超时 |
| QUIT | -30 | 上一次了送命令时被打断 |
| CHK_WR_OK | 0 | Check Write 正确 |
| CHK_WR_FAILED | -1 | Check Write 出错 |
| CHK_WR_COMP_ERR | -2 | Check Write:写出错（比较出错） |
| COMM_ERR | 255 | 串行通信错误 |
| MI_WRONG_VALUE | -123 | 值块格式错误 |

2.4 函数描述

2.4.1 装载密钥—Load_Key

函数原型: UINT __stdcall TX2_Load_Key(UINT mode, UINT secnr, char *key)

输入参数: mode: 密钥类型。可取值为 mode=0x00 (KEYA) —密钥 A, 或 mode=0x04 (KEYB) —密钥 B。

secnr: MUR-500 内的密钥区号, 取值范围: 0~15

key: 需要装载的密钥 (6 字节十六进制数据, 对应 12 个字符)

输出参数: 无

函数返回: 参见函数返回值列表。

例子: TX2_Load_Key(0x00, 0x01, "112233445566")

功能描述: 此函数的作用是将指定的密码装载到 MUR-500 内指定的密钥区, 并非改变 Mifare1 卡内扇区的密码。本函数只对 MUR-500 进行操作, MUR-500 与卡之间没有数据传输。

MUR-500 内有 16 个密码区 (区号 0——15), 称它为密钥区号。每个区分密钥 A 和密钥 B 两个, 总共 32 个密码。装载成功后, 可用该密钥对 Mifare1 卡进行验证。

在 M1 卡中也有 16 个存储区, 称它为扇区号。若要改变 Mifare1 卡内的密钥, 可在用原密码验证通过后, 直接用写块数据 TX_Write_Enc()函数, 将密码块改写。Mifare 卡出厂后的初始密钥为 6 个 FFH, A 和 B 密钥都一样。

2.4.2 请求—Request

函数原型: UINT __stdcall TX2_Request(UINT mode, UINT *tagtype);

输入参数: mode: 请求模式 ReqCode 取值为 1 或 0

mode=0 (IDLE), 请求天线范围内 IDLE 状态的卡 (HALT 状态的除外)

mode=1 (ALL), 请求天线范围内的所有卡。

输出参数: tagtype: 卡片类型

表 5 *TagType 的含义

| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|-----|-----|-----|-----|-----|-----|----|----|---|----|----------------------|----|----|----|----|----|
| 保留 | | | | | | | | 序列号大小 00:4bytes 01:7bytes 10:10bytes | 保留 | 任何位为 1 时, 则为比特帧防碰撞方式 | | | | | |

各种常用卡返回的数据如下表 6 所示:

表 6 常用卡的*TagType 值

| 卡类型 | *TagType | 卡类型 | *TagType |
|-------------------|----------|---------|----------|
| Mifare1 S50 | 0x0004 | SHC1101 | 0x0004 |
| Mifare1 S70 | 0x0002 | SHC1102 | 0x3300 |
| Mifare Light | 0x0010 | 11RF32 | 0x0004 |
| Mifare UltraLight | 0x0044 | | |

函数返回: 参见函数返回值列表。

功能描述: 检查在 MUR-500 有效范围内是否有卡存在。卡片进入天线区域后, 能获能量, 从而得电复位处于 IDLE 状态。TX2_Request 函数可用 ALL 或 IDLE 任意一种模式进行请求, 卡片均能响应, 并返回卡片类型号 TagType。在选择一张新的卡时必须调用该函数。

若对某一张卡成功进行了挂起操作（TX2_Halt 命令），卡片将进入 Halt 状态，此时的卡只能响应 ALL 模式的请求，除非该卡离开天线区域然后再重新进入。

注意：对同一张卡（不进入 HALT 状态）连续进行请求时，总是一次成功一次失败。

补充说明：

后面即将介绍的 TX_Halt()命令函数使被选中的卡进入 HALT 模式，进入该模式的卡只能用 ALL 方式进行请求，也即它对 IDLE 方式的请求不响应。若要使它再次响应 IDLE 请求，唯一方式是使卡复位，如执行 TX_Reset()函数，或使卡离开天线感应区后再进入。

可以配合使用 TX_Request()和 TX_Halt()函数，进行一次性扣费，如卡进入感应区后只扣一次（一元钱），离开后，下次进入再扣一次，若卡在感应区内停留时间较长，不得扣多于一次。方法如下：

```
while(1)
{
    while(TX2_Request(IDLE, databuf) != OK);
    ...
    扣费操作
    ...
    TX2_Halt();
}
```

2.4.3 带层级设置的防碰撞—Casc_Anticoll

函数原型：UINT __stdcall TX2_Casc_Anticoll(UINT bcnt, UINT code, ULONG *snr);

输入参数：code: 防碰撞层级编码：一层（ANTICOLL1）—0x93；二层（ANTICOLL2）—0x95；三层（ANTICOLL3）—0x97。

bcnt: 预选卡已经知道的序列号的位数，通常都设置 Bcnt=0。

输出参数：snr: 卡的序列号。若卡的序列号超过 4 个字节（如 Mifare UltraLight），序列号不完整，则最低字节的值为 0x88，表示需要进行更高一级的防碰撞 TX2_Casc_Anticoll（0x95）。

函数返回：参见函数返回值列表。

功能描述：可选级数的防碰撞操作。该函数必须在调用 TX2_Request 命令后立即调用。当多个卡片位于天线区域时，执行函数后，能得到序列号最大的那个卡片的序列号中的 4 个字节。卡的序列号长度有三种：4 字节、7 字节和 10 字节。4 字节的只要用一级防碰撞即可得到完整的序列号，如 Mifare1 S50 S70 等；7 字节的要求进行二级防碰撞才能得到完整的序列号，前一级所得到的序列号的最低字节为级联标志 0x88，在序列号内只后 3 字节可用，后一级防碰撞能得到 4 字节序列号，两者按顺序连接即为 7 字节序列号，如 UltraLight 和 DesFire 等；10 字节的以此类推，但至今没有此类卡。需要进行二级防碰撞操作，可通过修改参数 Select_Code 来实现。第一级使用 Select_Code=0x93，第二级使用 Select_Code=0x95，对于现在所有的 Mifare 卡，不需要进行三级防碰撞，因为所有 Mifare 卡的序列号最大为 7 个字节。如果已知卡片的序列号，则没有必要执行 AntiColl 函数，在调用 TX2_Request 后直接调用 TX2_Select 函数即可。

补充说明：

符合 ISO14443A 标准卡的序列号都是全球唯一的，正是由于唯一性，才能实现防碰撞

的算法逻辑。如果有多张卡同时在天线感应区内则这个函数能够找到一张序列号较大的卡来操作。实际上由于天线辐射的磁场能量有限，同时在天线感应区内的所有卡都要从辐射场中吸收，因此同时在天线感应区内的卡不能太多，否则没有一张卡能获得足够的能量来正常工作。

在执行选择 TX2_Select 命令之前，若已知所要操作的卡的序列号，则可跳过此步，直接执行选择 TX2_Select 命令；若不知道卡的序列号，则必须调用防碰撞函数，得到感应区内卡的序列号。

对于 MIFARE 1 卡片来说，返回某一张卡片的有效序列号 SNR（4 个字节）可能为：007e0a42h。

2.4.4 带层级设置的选择—Casc_Select

函数原型：UINT __stdcall TX2_Casc_Select(UINT code,unsigned long snr, UINT *sak);

输入参数：code：防碰撞层级编码：一层（ANTICOLL1）—0x93；二层（ANTICOLL2）—0x95；三层（ANTICOLL3）—0x97。该参数应该与之前的 TX2_Casc_Anticoll()中的 Select_Code 参数相同。

snr：前一次防碰撞返回的卡的序号，或已知的卡的序列号。如果卡得序号号大于 4 个字节，则 snr 的最低字节值为 0x88，表示需要进行更高一级的防碰撞。

输出参数：sak：是否选择成功的应答,其意义如表 7 所示：

表 7 *Sak 含义

| | | | | | | | |
|-----|----|----|-----|----|----------|-----|----|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| RFU | | | RFU | | 判断是否选择完成 | RFU | |

bit2 位用来判断是否还有序列号未读出，bit2=0 时，表示所有序列号输入完毕，选择成功，bit2=1 时表示序列号没有全部输入，选择没有完全成功，还要进行下一级的防碰撞和选择操作。

当*Sak=xxxxx1xx 时，序列号没有完成，还要进行下一级的防碰撞和选择操作。

当*Sak=xx1xx0xx 时，选择成功，该卡符合 ISO/IEC 14443-4 标准。

当*Sak=xx0xx0xx 时，选择成功，该卡不符合 ISO/IEC 14443-4 标准。

各种常用卡返回的数据如下所表 8 所示：

表 8 常用卡的*Sak 的值

| 卡类型 | *Sak | 卡类型 | *Sak |
|---------------|------|--------------------|------|
| Mifare1 S50 | 0x08 | Mifare0 UltraLight | 0x04 |
| Mifare1 S70 | 0x18 | SHC1101 | 0x22 |
| Mifare1 Light | 0x01 | 11RF32 | 0x08 |

函数返回：参见函数返回值列表。

功能描述：该函数选择某一个序列号的卡，与之建立通信连接，并返回是否选择成功的应答 sak。该函数一般与 TX2_Casc_Anticoll()配对使用，进行多级防碰撞后卡的选择。参数 code 表示防碰撞的级数，应该与该函数执行前的 TX2_Casc_Anticoll()中的 code 相同。在任意一个防碰撞函数成功执行后，必须使用 TX2_Casc_Select 函数，以建立与所选卡的通信。卡的序列号长度有三种：4 字节、7 字节和 10 字节。4 字节的只要用一级选择即可,如 Mifare1 S50 S70 等；7 字节的要求用二级选择才能完成，如 UltraLight 和 DesFire 等。如果需要进行二级选择，第一级选择输入的序列号为第一级防碰撞所得到的序列号（其中最低字节为级联标志 0x88，之后 3 字节可用），第二级选择输入的序列号为第二级防碰撞得到 4 字节序列号。

在程序中可用 sak 的 bit2 位来判断是否还有序列号未读出，如 if(sak & 0x04){...}。

2.4.5 证实 2—Auth2

函数原型: UINT __stdcall TX2_Auth2(UINT mode,UINT secnr,UINT keynr);

输入参数: mode: 密钥类型。可取值为 mode=0x00 (KEYA), 利用密钥 A 进行验证; 或 mode=0x04 (KEYB), 利用密钥 B 进行验证。

secnr: 所要验证的卡扇区号 (也即将要访问的卡的扇区号), 取值范围 0~39, 能用于 S70 卡。

keynr: MUR-500 内的密钥区号: 取值范围 0~15。

输出参数: 无

函数返回: 参见函数返回值列表。

功能描述: 使用 MUR-500 内部密钥区 keynr 中的密码对指定的卡的扇区 secnr 进行验证, 若卡 secnr 区中的密码与存储在 MUR-500 内 secnr 中的密码相同, 则验证成功, 返回 OK。

该函数依赖于 TX2_Load_Key 函数曾经成功执行过, 因为 MUR-500 内部密码区 (keynr) 中的密码要由 TX2_Load_Key 函数事先装载。该函数适用于对于所有卡来说密码相同的应用, 密钥的装载可以在一个安全的场合一次性装入。

2.4.6 直接密码证实—Auth_Key

函数原型: UINT __stdcall TX2_Auth_Key(UINT mode, char *key,UINT secnr);

输入参数: mode: 密钥类型。可取值为 mode=0x00 (KEYA), 利用密钥 A 进行验证; 或 mode=0x04 (KEYB), 利用密钥 B 进行验证。

key: 用于证实的密码 (6 字节十六进制数据, 对应 12 个字符);

secnr: 所要验证的卡扇区号 (也即将要访问的卡的扇区号), 取值范围 0~39, 能用于 S70 卡。

输出参数: 无

函数返回: 参见函数返回值列表。

例 子: TX2_Auth_Key(0x00, "112233445566", 1)

功能描述: TX2_Auth2 函数必须要依赖于 TX2_Load_Key()函数的曾经的成功执行, 在进行证实之前, 一定要确认正确的密钥已经存于 MUR-500 的密钥区内。而 TX2_Auth_Key 函数则直接将密钥传送到 MUR 内对卡片进行验证。TX2_Auth_Key 函数执行时不对密钥区进行操作, 因此也称为直接密码证实。若卡中的密钥与所传输的密码相匹配, 则证实成功, 函数将返回 OK。

直接密码证实一般用于对每一张卡来说密钥都不同的应用, 如在使用安全模块 (PSAM 卡) 的消费应用中, 消费机首先将卡的序列号读出, 然后与 PSAM 卡中消费主密钥一起生成导出密钥, 然后直接用导出密钥与卡的一个应用扇区相互证实。

2.4.7 写—Write

函数原型: UINT __stdcall TX2_Write_Enc(UINT addr, char *data)

输入参数: addr: 卡块号: S50: 1~63; S70: 1~255

data: 要写入的数据 (16 字节的十六进制数据, 对应 32 个字符)

输出参数: 无

函数返回: 参见函数返回值列表。

例 子: TX2_Write_Enc(1, "00112233445566778899AABBCCDDEEFF")

功能描述: 对卡内某一块进行验证成功后,即可对同一扇区的各个块进行写操作(只要访问条件允许),其中包括位于扇区尾的密码块,这是更改密码的唯一方法。

Mifare 卡中一个块的数据是 16 字节,因此读写一次均是 16 个字节。所读块号必须与之前所验证的块号在同一个扇区内,mifare1 卡从块号 0 开始按顺序每 4 个块 1 个扇区。

补充说明:

在将数据写到卡片上的某一扇区时,一定要小心。因为有些 block 中存储了密码数据以及存储允许使能数据。特别是每一个扇区的 Block3 中存放了该扇区的存取条件,包含有 KEYA, KEYB 及该扇区的控制字。Mifare 1 卡片出厂时的 Block3 有缺省值,为:“a0a1a2a3a4a5ff078069b0b1b2b3b4b5”,共 16 个 Bytes。

涉及 Mifare 1 卡片的存储结构等信息,请参考 Mifare 1 卡片数据手册。

程序员在使用 Mifare 1 卡片做应用时,一定要清清楚楚记住每一个扇区的 Block3 的数据,这样也就记住了扇区的密码和存取控制字。否则,扇区的存储空间将不执行 Read/Write 等操作而失效。

任何人试图用任何方式来读写不知密码的卡片或某一扇区都是徒劳无益的。

卡片应放在安全的地方,即不要放在离 MUR-500 天线较近的地方。因为当 MUR-500 对其它卡片执行某些指令时,有可能无意间对这一卡片进行了读/写等操作,从而造成卡片的失效。

2.4.8 写值块操作—Write_Val

函数原型: UINT __stdcall TX2_Write_Val(UINT addr,long value);

输入参数: addr --卡块号: S50: 1~63
S70: 1~255

value: 4 字节数据,用来存储减少值或增加值,当进行恢复操作时,该值为空值。

输出参数: 无

函数返回: 参见函数返回值列表。

功能描述: 往 Mifare 卡中相应块写入值块格式的数据(只要访问条件允许)。

2.4.9 读—Read

函数原型: UINT __stdcall TX2_Read_Enc(UINT addr, char *data)

输入参数: addr: 卡块号: S50: 0~63; S70: 0~255

输出参数: data: 为读回数据的首地址。读回的为 16 字节的十六进制数据,对应 32 个字符。例如读取到"00112233445566778899AABBCCDDEEFF"。

函数返回: 参见函数返回值列表。

功能描述: 在验证成功后,使用该函数读 Mifare 卡中相应块的数据。Mifare 卡中一个块的数据是 16 字节,因此读写一次均是 16 个字节。

所读块号必须与之前所验证的块号在同一个扇区内,mifare1 卡从块号 0 开始按顺序每 4 个块 1 个扇区。密码数据不能被读取。

2.4.10 读值块操作—Read_Val

函数原型: UINT __stdcall TX2_Read_Val(UINT addr, unsigned long *value);

输入参数: addr --卡块号: S50: 1~63
S70: 1~255

输出参数: value: 4 字节值块值。

函数返回: 参见函数返回值列表。

功能描述: 从 Mifare 卡中相应块读出值块格式的数据。

2.4.11 带内部自动传送的值操作—Value

函数原型: UINT __stdcall TX2_Value(UINT mode,UINT addr,long value,UINT trans_addr);

输入参数: mode: 0xC0—减; 0xC1—加; 0xC2—恢复

addr: 卡内块地址, 对该块进行值操作, 也就是源地址。取值范围: S50: 1~63; S70: 1~255

value: 4 字节数据, 用来存储减少值或增加值, 当进行恢复操作时, 该值为空值。

trans_addr: 传输块地址, 也就是结果保存地址。取值范围: S50: 1~63; S70: 1~255。

输出参数: 无

函数返回: 参见函数返回值列表。

功能描述: 此函数对卡内的某一块进行加、减或数据备份, 该块必须为值块格式。若卡块号与传输块号相同, 则将操作后的结果写入原来的块内; 若卡块号与传输块号不相同, 则将操作后的结果写入传输块内, 而原块内的值不变。当模式为“恢复”时, “值”无意义。

2.4.12 写 UltraLight—UI_Write

函数原型: UINT __stdcall TX2_ULwrite_Enc(UINT addr, char* data);

输入参数: addr: 卡块号: 0~15

data: 要写入的数据 (4 字节的十六进制数据, 对应 8 个字符)

输出参数: 无

函数返回: 参见函数返回值列表。

例子: TX2_ULwrite_Enc(1, "11223344")

功能描述: 对 UltraLight 卡写入一个 4 字节的数据。此命令只对 UltraLight 有效。对 UltraLight 进行读操作与 mifare1 一样。

2.4.13 挂起—Halt

函数原型: UINT __stdcall TX2_Halt(void)

输入参数: 无

输出参数: 无

函数返回: 参见函数返回值列表。

功能描述: 将天线区所选择卡置为挂起状态。如果要进行重新选择, 则应用 ALL 模式调用 TX2_Request 命令。如果要进行重新选择, 也可以将卡离开天线操作区再进入。

可以配合使用 TX2_Request()和 TX2_Halt()函数, 进行一次性扣费, 如卡进入感应区后只扣一次 (一元钱), 离开后, 下次进入再扣一次, 若卡在感应区内停留时间较长, 也不会扣多一次。

2.4.14 置位控制位—Green_Off

函数原型: UINT __stdcall TX2_Green_Off();

输入参数: 无

输出参数: 无

函数返回: 参见函数返回值列表。

功能描述: 此函数设置 MUR-500 的绿灯灭。

2.4.15 清除控制位—Green_On

函数原型: UINT __stdcall TX2_Green_On();

输入参数: 无

输出参数: 无

函数返回: 参见函数返回值列表。

功能描述: 此函数设置 MUR-500 的绿灯亮。

2.4.16 输出信号—Alarm

函数原型: UINT __stdcall TX2_Alarm(UINT contrl,UINT opentm,UINT closetm,UINT repcnt);

输入参数: contrl: 控制字, Control = 0x01 时, 控制对象为蜂鸣器, Control = 0x02 时, 控制对象为绿灯, Control = 0x03 时, 绿灯和蜂鸣器同时控制。

Opentm: 方波输出持续时间, 取值 (0~255), 10ms 的分辨率

Closetm: 间隙时间, 取值 (0~255), 10ms 的分辨率

Repcnt: 重复次数

输出参数: 无

函数返回: 参见函数返回值列表。

功能描述: 此函数输出一方波用于驱动蜂鸣器、绿灯, 驱动的频率、持续时间、间隙时间和重复次数可设定。

3. Mifare 卡工作状态介绍

Mifare 卡工作状态如表 9 所列。

表 9 Mifare 卡工作状态列表

| 状 态 | 描 述 |
|-------------------|---|
| POWER OFF (断电) | 卡片不位于读卡器有效区域,由于缺少射频磁场能量而处于断电状态,卡片不工作。 |
| IDLE (空闲) | 卡片进入读卡起有效区域内,被电磁场能量激活,延迟数毫秒后将进入 IDLE 状态。在该状态下,卡片能够接收读卡器传来的命令,并能对读卡器的 Request (以 IDLE 或 ALL 方式) 命令进行应答,应答后返回卡片的类型。 |
| READY (就绪) | 卡片对读卡器的 Request 命令进行应答后,就进入了 READY 状态。在该状态中,可以采用比特帧防冲突算法。当卡片的唯一序列号被读卡器发来的 Selection 命令选中时,就退出本状态。 |
| ACTIVE (激活) | 当卡片的唯一序列号被读卡器选中时就进入 ACTIVE 状态。在该状态中,卡片完成本次应用所要求的全部操作。 |
| HALT (停止) | 卡片应用完成后,读卡器可通过发送 Halt 命令,使卡片进入 HALT 状态。在该状态中,卡片只对读卡器以 ALL 方式发送的 Request 命令进行应答(或被唤醒),从而又进入 READY 状态。 |

图 1 为 Mifare 卡的状态转换图。

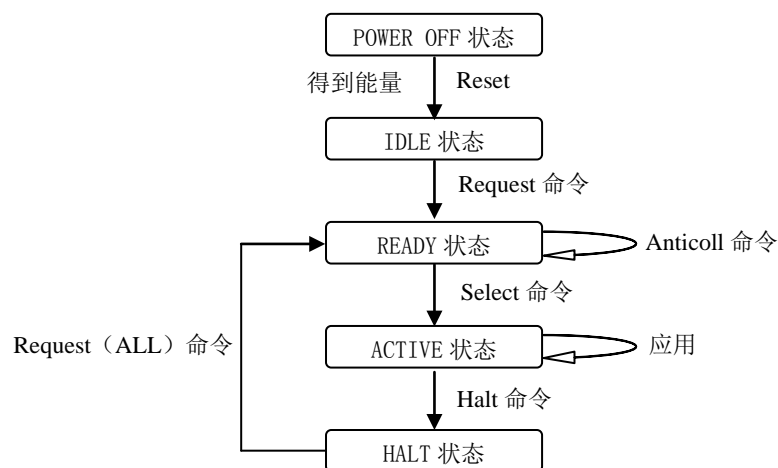


图 1 Mifare 卡状态转换示意图

4. 修订历史

| 版本 | 日期 | 原因 |
|-------|------------|-------|
| V1.00 | 2009/03/11 | 创建文档。 |
| | | |
| | | |
| | | |
| | | |
| | | |

5. 销售信息

同欣智能

地 址：广东省东莞市石碣镇沙腰管理区林屋洲

邮 编：523292

电 话：+86 (0769) 26672410; 15916881410

传 真：+86 (0769) 88709173

网 址：http:// www.TXRFID.com

E-mail: sales@TXRFID.com support@TXRFID.com