

EM4469(EM4205/4305)读写协议说明

数据格式

1. 1 数据格式

数据格式（起始位，数据位，校验位，停止位）可以根据通讯的需要由软件设置，下面是设备支持的数据格式：

下面是默认设置：

波特率	数据位	起始位	停止位	校验位
9600	8	1	1	None

1. 2 数据包格式

数据包格式，命令包是由主机发送到读写器，返回包是由读写器返回主机。

命令包格式 (主机到读写器)：

STX	CARD ID	DATA LENGTH	CMD	DATA [0..N]	BCC	ETX
-----	---------	-------------	-----	-------------	-----	-----

(BCC) = CARD ID ⊕ DATALENGTH ⊕ CMD ⊕ DATA [0] ⊕ ... ⊕ DATA [n], where ⊕ is the "EOR".

返回包格式 (读写器到主机)

STX	CARD ID	DATA LENGTH	STATUS	DATA[0..N]	BCC	ETX
-----	---------	-------------	--------	------------	-----	-----

(BCC) = CARD ID ⊕ DATA LENGTH ⊕ STATUS ⊕ DATA [0] ⊕ ... ⊕ DATA [n], where ⊕ is the "EOR".

数据包中字节描述：

字段	长度	描述	备注
STX	1	0xAA – ‘起始字节’ – 标准控制字节. 表示一个数据包的开始	
CARD ID	1	卡型号代码 Manchester RF/64 CARD ID:0A Bi-phase RF/32 CRAD ID:0B	Em4469,EM4205 EM4305 或兼容的卡
DATALENGT H	1	数据包中数据字节的长度.包括 CMD/STATUS 和 DATA field,但不包括 BCC. LENGTH= 字节数 (CMD/STATUS + DATA[0.. N])	
CMD	1	命令字:由一个命令字节组成.	可以参照命令表 该字节只在发送包 中使用
STATUS	1	返回状态字节: 由读写器返回主机的状态	该字节只在返回包 中使用
DATA [0-N]	0-255	这是一个长度与命令字有关的数据流。也有 部分命令不需要附加数据.	
BCC	1	8bits 的校验字节.它包括除 STX, ETX 外所有 字节的异或校验.	

非接触式读卡模块通讯协议

ETX	1	0xBB:'终止字节' – 标准控制字节, 表示数据包的结束.	
-----	---	---------------------------------	--

COMMANDS (命令)

命 令 表		
命令字	名称	描述
系统命令 (0x80~0x8F)		
0x84	Write_UserInfo	写卡数据信息
0x85	Read_UserInfo	读取卡中指定块的数据信息
0x86	Wake-Up	登入
卡数据输出格式为 Manchester RF/64 CARD ID:0A 卡数据输出格式为 Bi-phase RF/32 CRAD ID:0B 在下面协议中都有两种格式一种是 0A 的一种是 0B 的,不再用文字注明		

3.1 Write_UserInfo (0x84)

发送数据:

- DATA[0]: 对卡进行写数据操作的区域号
- 0X00: 对卡的 Page0 进行写操作
 - 0x01: 对卡的 Page1 进行写操作
 - 0x02: 对卡的 Page2 进行写操作
 - 0x03: 对卡的 Page3 进行写操作
 - 0x04: 对卡的 Page4 进行写操作
 - 0x05: 对卡的 Page5 进行写操作
 - 0x06: 对卡的 Page6 进行写操作
 - 0x07: 对卡的 Page7 进行写操作
 - 0x08: 对卡的 Page8 进行写操作
 - 0x09: 对卡的 Page9 进行写操作
 - 0x0A: 对卡的 Page10 进行写操作
 - 0x0B: 对卡的 Page11 进行写操作
 - 0x0C: 对卡的 Page12 进行写操作
 - 0x0D: 对卡的 Page13 进行写操作

0x0E: 对卡的 Page14 进行写操作

0x0F: 对卡的 Page15 进行写操作

DATA[1-4] 要写入的数据信息.

正确返回:

STATUS: 0x00 – OK

DATA[0]: 0x80(表示操作成功)

错误返回:

STATUS: 0x01 –FAIL

DATA[0] 参考错误代码表

描述: 卡提供 15 个块 (每个块有 4 个字节),

比如: PAGE 1 写数据为 55 AA 55 AA

发送命令 : **AA 0A 06 84 01 55 AA 55 AA 89 BB**

或: **AA 0B 06 84 01 55 AA 55 AA 88 BB**

写成功后

回执数据 : **AA 0A 02 00 80 88 BB**

AA 0B 02 00 80 89 BB

写卡失败:

回执数据 : **AA 0A 02 01 83 8A BB**

AA 0B 02 01 83 8B BB

写卡举例:

CARD ID:0A

写 Page 0 数据:00 00 00 00 : (AA 0A 06 84 00 00 00 00 00 88 BB)

写 Page 1 数据:11 11 11 11 : (AA 0A 06 84 01 11 11 11 11 89 BB)

写 Page2 数据:22 22 22 22 : (AA 0A 06 84 02 22 22 22 22 8A BB)

写 Page 3 数据:33 33 33 33 : (AA 0A 06 84 03 33 33 33 33 8B BB)

写 Page4 数据:44 44 44 44 : (AA 0A 06 84 04 44 44 44 44 8C BB)

写 Page5 数据:55 55 55 55 : (AA 0A 06 84 05 55 55 55 55 8D BB)

写 Page 6 数据:66 66 66 66 : (AA 0A 06 84 06 66 66 66 66 8E BB)

写 Page 7 数据:77 77 77 77 : (AA 0A 06 84 07 77 77 77 77 8F BB)

写 Page8 数据: 88 88 88 88 (AA 0A 06 84 08 88 88 88 88 80 BB)

写 Page9 数据::99 99 99 99 (AA 0A 06 84 09 99 99 99 99 81 BB)

写 Page10 数据::AA AA AA AA (AA 0A 06 84 0A AA AA AA AA 82 BB)

写 Page11 数据::BB BB BB BB (AA 0A 06 84 0B AA AA AA AA 83 BB)

写 Page12 数据::CC CC CC CC (AA 0A 06 84 0C CC CC CC CC 84 BB)

写 Page13 数据::DD DD DD DD (AA 0A 06 84 0D DD DD DD DD 85 BB)

写 Page14 数据::EE EE EE EE (AA 0A 06 84 0E EE EE EE EE 86 BB)

写 Page15 数据::FF FF FF FF (AA 0A 06 84 0F FF FF FF FF 87 BB)

CARD ID:0B

写 Page 0 数据:00 00 00 00 : (AA 0B 06 84 00 00 00 00 00 89 BB)

写 Page 1 数据:11 11 11 11 : (AA 0B 06 84 01 11 11 11 11 88 BB)

写 Page2 数据:22 22 22 22 : (AA 0B 06 84 02 22 22 22 22 8B BB)

写 Page 3 数据:33 33 33 33 : (AA 0B 06 84 03 33 33 33 33 8A BB)

写 Page4 数据:44 44 44 44 : (AA 0B 06 84 04 44 44 44 44 8D BB)

写 Page5 数据:55 55 55 55 : (AA 0B 06 84 05 55 55 55 55 8C BB)

写 Page 6 数据:66 66 66 66 : (AA 0B 06 84 06 66 66 66 66 8F BB)

写 Page 7 数据:77 77 77 77 : (AA 0B 06 84 07 77 77 77 77 8E BB)

写 Page8 数据: 88 88 88 88 (AA 0B 06 84 08 88 88 88 88 81 BB)

写 Page9 数据::99 99 99 99 (AA 0B 06 84 09 99 99 99 99 80 BB)

写 Page10 数据::AA AA AA AA (AA 0B 06 84 0A AA AA AA AA 83 BB)

写 Page11 数据::BB BB BB BB (AA 0B 06 84 0B AA AA AA AA 82 BB)

写 Page12 数据::CC CC CC CC (AA 0B 06 84 0C CC CC CC CC 85 BB)

写 Page13 数据::DD DD DD DD (AA 0B 06 84 0D DD DD DD DD 84 BB)

写 Page14 数据::EE EE EE EE (AA 0B 06 84 0E EE EE EE EE 87 BB)

写 Page15 数据::FF FF FF FF (AA 0B 06 84 0F FF FF FF FF 86 BB)

3.2 Read_UserInfo (0x85)

发送数据:

DATA[0]: 对卡进行读数据操作的区域号

0X00: 对卡的 Page0 进行读操作

0x01: 对卡的 Page1 进行读操作

0x02: 对卡的 Page2 进行读操作

0x03: 对卡的 Page3 进行读操作

0x04: 对卡的 Page4 进行读操作

0x05: 对卡的 Page5 进行读操作

0x06: 对卡的 Page6 进行读操作

0x07: 对卡的 Page7 进行读操作

0x08: 对卡的 Page8 进行读操作

0x09: 对卡的 Page9 进行读操作

0x0A: 对卡的 Page10 进行读操作

0x0B: 对卡的 Page11 进行读操作

0x0C: 对卡的 Page12 进行读操作

0x0D: 对卡的 Page13 进行读操作

0x0E: 对卡的 Page14 进行读操作

0x0F: 对卡的 Page15 进行读操作

正确返回:

STATUS: 0x00 – OK

DATA{1..N} 读出的卡数据 4 byte

错误返回:

STATUS: 0x01 –FAIL

DATA[0] 参考错误代码表

描述: 读卡的 15 个块 (每个块 4 个字节) 的数据,

比如: 读对卡的 PAGE1 进行读操作

卡的数据: 55 AA 55 AA

发送命令: **AA 0A 02 85 01 8C BB**

或: **AA 0B 02 85 01 8D BB**

回执数据: **AA 0A 05 00 55 AA 55 AA 0F BB**

或: **AA 0B 05 00 55 AA 55 AA 0E BB**

没有读到卡:

回执数据: **AA 0A 02 01 83 8A BB**

或: **AA 0B 02 01 83 8B BB**

3. 登入 LOGIN (0x86)命令

发送数据: DATA[0-3] 4 个字节的密码

正确返回:

STATUS: 0x00 – OK

DATA[0]: 0x80(表示操作成功)

错误返回:

STATUS: 0x01 –FAIL

DATA[0] 参考错误代码表

描述: 密码登入

比如: 卡的密码是 0X55 , 0X55H , 0X55H

发送命令: **AA 0A 05 86 55 55 55 55 89 BB**

非接触式读卡模块通讯协议

或: AA **0B** 05 **86** 55 55 55 55 **88** BB

成功返回:

回执数据: AA 0A 02 00 **80** 88 BB

或: AA **0B** 02 00 80 89 BB

错误返回:

回执数据: AA 0A 02 01 83 8A BB

或: AA **0B** 02 01 83 8B BB

7 错误/状态 代码(STATUS)

一般代码：

0x00:	表示命令执行成功
0x01:	表示命令操作失败（具体说明参见函数）
0x80:	表示参数设置成功
0x81:	表示写卡失败
0x82:	表示读卡失败
0x83:	表示卡不存在(81,82,83代码可以通用都可以代表读或写失败)
0x84 符合)	表示卡和读卡器不匹配(就是要读卡的类型CARDID和读卡器不
0x87:	表示未知的错误
0x85:	表示输入参数错误,校验错误或者输入命令不存在
0x8f:	表示 输入的指令代码不存在