

读 ID 卡号协议

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	
CARDID	1	卡型号代码, 125KHZ 的 ID 卡代码是 01	EM4001 或兼容卡
DATALENGTH	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.	
ETX	1	0xBB: '终止字节' - 标准控制字节, 表示数据包的结束.	

2:COMMANDS (命令)

命令表		
命令字	名称	描述
系统命令 (0x80~0x8F) 读 125KHZ 的 EMID(EM4001 或兼容的卡)只有一个命令		
0x85	Read_ID	读取卡中的数据信息
如果命令发送后 1 秒内没有收到读卡器的返回数据就停止接收并提示命令发送失败		

3:Read-ID (0x85)

发送数据：无

正确返回：

STATUS: 0x00 – OK

DATA[1-N]: DATA1-N 是这个卡的具体数据

错误返回：

STATUS: 0x01 –FAIL

DATA[0] 参考错误代码表

描述： 读 125KHZ 的 EMID 卡(EM4001 或兼容卡)

比如:卡号数据: 02 00 B0 97 44

发送命令： AA 01 01 85 85 BB

回执数据： AA 01 06 00 02 00 B0 97 44 66 BB

没有读到卡：

回执数据： AA 01 02 01 83 81 BB

CARD: EMID卡 (EM4001)或兼容的卡 (RF/64 Manchester)

读卡1区	读卡2区
读卡1区适用于软件点一下读卡后ID读卡器 再发送数据的ID读卡器	读卡2区适用于主动发送数据的ID读卡器
卡 号 HEX <input style="width: 100%;" type="text" value="02 00 B0 97 44"/>	卡 号 HEX <input style="width: 100%;" type="text" value="02 00 B0 97 44"/>
10位10进制 <input style="width: 100%;" type="text" value="0011573060"/>	10位10进制 <input style="width: 100%;" type="text" value="0011573060"/>
WG26(韦根26) <input style="width: 100%;" type="text" value="176, 38724"/>	WG26(韦根26) <input style="width: 100%;" type="text" value="176, 38724"/>
<input type="button" value="读卡"/> <input type="button" value="清除数据"/>	<input type="button" value="读卡/关闭"/> <input type="button" value="清除数据"/>

读卡1区是软件点一下读卡后ID读卡器返回数据,不会主动读卡
 读卡2区是读卡器读到数据后主动发数据给软件

读卡1区和读卡2区两个软件不能同时工作.

先讲读卡1区:

在读卡1区点一下”读卡”按键,软件发送: : AA 01 01 85 **85** BB
 ID读卡器如果没读到卡就返回: AA 01 02 01 **83 81** BB
 软件中提示”读卡失败”

如读到的卡号是: **02 00 B0 97 44**
 返回: AA **01 06 00 02 00 B0 97 44 66** BB

在软件的 HEX 框中显示:02 00 B0 97 44

在 10 位 10 进制中显示:0011573060 (这个数据是由后 4 个 16 进制转换而来 00 B0 97 44, 数据不够 10 位前面添 0 补充)

WG26 中显示:176,38724 (这个由后 3 个字节转换而来: B0 , 9744 . 前三位由 B0 转换 10 进制,位数不够前面添 0,后 5 位由 9744 转换后来,位数不够前面添 0,前三位和后 5 位之间加一个”,”号上去把两个数据隔离开)

读卡2区:

在读卡2区点一下”读卡/关闭”后,软件不发送数据,一直处于接收状态,ID读卡器发送数据后软件直接显示数据,显示数据方式和读卡1区一样.当读卡器再发送一次数据后,软件更新显示数据,再点”读卡/关闭”后软件就不接收数据了.读卡器发的数据是和接收命令后有卡发送的数据格式是一样的,没卡是不发送数据.

10 位 10 进制位数不够前面添 0 解释:

比如 4 个字节是 00 00 11 89 转换成 10 进制是 4489,因为需要 10 位,而 4489 才 4 位,缺了 6 位,要在前面添加 6 个 0,补齐 10 位所以 10 位 10 进:0000004489

WG26 时位数不够前面添 0 解释:

比如后 3 个字节是 33 0F E9 转换后数据的前三位是 33 转换而来的,0X33 转换后是 51 只有两位,要在前面添个 0 补齐 3 位:051

后 5 位是 0F E9 转换而后,0FE9 转换后数据是 4073,只有 4 位,要在前面添个 0 补齐 5 位:04073

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	表示卡和读卡器不匹配(就是要读卡的类型CARDID和读卡器不符合)
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