

# 电插锁控制板跟上位机通讯协议

(2022 年 10 月修订)



## 1、概述

- 1) 板子的通讯接口为 RS485 通讯口；
- 2) 板子可以接 1 把或者 2 把电插锁；
- 3) 板子波特率默认为 9600；

## 2、数据帧格式

名称	板地址	指令字	数据长度	数据域	校验
长度(字节)	1	1	2	n	2

板地址 : 1 个字节, 板子默认的地址为 01;

指令字 : 1 个字节, 不同的指令字对应不同的功能;

数据长度: 2 个字节, 指数据域里面的字节数量;

数据域 : n 个字节, 数据域里面的内容由指令字决定, 不同的指令字数数据域的内容不同;

校验 : 2 个字节, 除了校验位, 其他所有字节都要校验, 采取 CRC16 校验。

## 3、指令字列表

指令字	说明	功能	注意事项
0x06	开锁	开单个锁	无需发落锁指令落锁
		开 1 号和 2 号锁	无需发落锁指令落锁
		同时开 2 个锁	需要发落锁指令落锁
0x05	落锁	落单个锁	落锁后, 不允许拿开门磁铁
		同时落 2 个锁	落锁后, 不允许拿开门磁铁
0x03	查询门磁和锁舌状态	查询单个门磁和锁舌状态	
		查询 1 号和 2 号电插锁的门磁和锁舌状态	

#### 4、数据帧详解

##### 4.1 开锁 (0x06)

注意: 不允许锁舌落下后, 然后去拿开门磁铁, 不符合逻辑, 会造成 5 秒内跳锁的现象出现, 出现此现象是由于不按逻辑开门造成, 属于正常现象。

**4.1.1 开单个锁 (开锁默认延时 5 秒, 5 秒后感应到门磁关闭会自动落锁) 无需发落锁指令**  
服务器发送:

板地址	指令字	数据长度	数据域	校验
0x01	0x06	0x00 0x03	开锁数量 (1) + 锁号 (1) + 开锁延时时间 (1)	CRC16

设备端回复:

板地址	指令字	数据长度	数据域	校验
0x01	0x06	0x00 0x01	开锁结果 (1)	CRC16

**数据长度:** 指数据域里面内容的字节数量;

**开锁数量:** 如果是开单个锁, 开锁数量就是 1 个, 即 0x01;

**锁号:** 指需要开 1 号锁还是开 2 号锁, 开 1 号锁就是 0x01, 开 2 号锁就是 0x02;

**开锁延时时间:** 这里默认开锁延时时间为 5 秒, 即 0x05; 需要说明一下, 所谓的开锁延时时间, 就是锁打开后, 如果门磁铁没有移走, 5 秒后就自动落锁, 如果门磁铁移走了, 5 秒后, 只要锁感应到门磁铁, 就会马上落锁;

**开锁结果:** 0x01 代表开锁成功; 0x00 代表开锁失败。

**Demo1:** 开 1 号电插锁

服务器发送: 01 06 00 03 01 01 05 9B B1

设备端回复: 01 06 00 01 01 D9 18

**Demo2:** 开 2 号电插锁

服务器发送: 01 06 00 03 01 02 05 9B 41

设备端回复: 01 06 00 01 01 D9 18

**4.1.2 开 1 号和 2 号锁 (开锁延时 5 秒, 5 秒后感应到门磁关闭自动落锁) 无需发落锁指令**

服务器发送:

板地址	指令字	数据长度	数据域	校验
0x01	0x06	0x00 0x05	开锁数量 (1) + 锁号 (1) + 开锁延时时间 (1) + 锁号 (1) + 开锁延时时间 (1)	CRC16

设备端回复:

板地址	指令字	数据长度	数据域	校验
0x01	0x06	0x00 0x02	1 号锁开锁结果 (1) + 2 号锁开锁结果 (1)	CRC16

**数据长度:** 指数据域里面内容的字节数量;

**开锁数量:** 如果是开 2 个锁, 开锁数量就是 2 个, 即 0x02;

**锁号:** 指需要开 1 号锁还是开 2 号锁, 开 1 号锁就是 0x01, 开 2 号锁就是 0x02;

**开锁延时时间:** 这里默认开锁延时时间为 5 秒, 即 0x05; 需要说明一下, 所谓的开锁延时时间, 就是锁打开后, 如果门磁铁没有移走, 5 秒后就自动落锁, 如果门磁铁移走了, 5 秒后, 只要锁感应到门磁铁, 就会马上落锁;

**开锁结果:** 0x01 代表开锁成功; 0x00 代表开锁失败。

**Demo1: 开 1 号和 2 号锁**

服务器发送: 01 06 00 05 02 01 05 02 05 2E 41

设备端回复: 01 06 00 02 01 01 E8 5A

**4.1.3 同时开 2 个锁（开锁一直延时，不会自动落锁）需要发落锁指令才能落锁**

服务器发送:

板地址	指令字	数据长度	数据域（固定）	校验（固定）
0x01	0x06	0x00 0x03	0xFF 0xFF 0x00	0x7A 0x22

设备端回复:

板地址	指令字	数据长度	数据域	校验
0x01	0x06	0x00 0x02	1 号锁开锁结果(1)+2 号锁开锁结果(1)	CRC16

**数据长度:** 指数据域里面内容的字节数量;

**开锁结果:** 0x00 代表开锁成功; 0x01 代表开锁失败。

**Demo1: 同时开 2 个锁**

服务器发送: 01 06 00 03 FF FF 00 7A 22

设备端回复: 01 06 00 02 00 00 28 0A

**4.2 落锁（0x05）**

**注意:** 发落锁指令后，只有在感应到门磁关闭的时候，才会落锁

**4.2.1 落单个锁**

服务器发送:

板地址	指令字	数据长度	数据域	校验
0x01	0x05	0x00 0x03	落锁数量（1）+锁号（1）+预留数据位（1）	CRC16

设备端回复:

板地址	指令字	数据长度	数据域	校验
0x01	0x05	0x00 0x01	落锁结果（1）	CRC16

**数据长度:** 指数据域里面的字节数量;

**落锁数量:** 如果是落单个锁，落锁数量就是 1 个，即 0x01;

**锁号:** 指需要落 1 号锁还是落 2 号锁，落 1 号锁就是 0x01，落 2 号锁就是 0x02;

**预留数据位:** 固定为 0x00;

**落锁结果:** 0x00 代表落锁成功; 0x01 代表落锁失败。

**Demo1: 落 1 号电插锁**

服务器发送: 01 05 00 03 01 01 00 5B 81

设备端回复: 01 05 00 01 00 18 9C

**Demo2: 落 2 号电插锁**

服务器发送: 01 05 00 03 01 02 00 5B 71

设备端回复: 01 05 00 01 00 18 9C

#### 4.2.2 同时落 2 个锁

服务器发送:

板地址	指令字	数据长度	数据域	校验
0x01	0x05	0x00 0x05	落锁数量 (1) + 锁号 (1) + 预留数据位 (1) + 锁号 (1) + 预留数据位 (1)	CRC16

设备端回复:

板地址	指令字	数据长度	数据域	校验
0x01	0x05	0x00 0x02	1 号锁落锁结果 (1) + 2 号锁落锁结果 (1)	CRC16

**数据长度:** 指数据域里面的字节数量;

**落锁数量:** 如果是落 2 个锁, 落锁数量就是 2 个, 即 0x02;

**锁号:** 指需要落 1 号锁还是落 2 号锁, 落 1 号锁就是 0x01, 落 2 号锁就是 0x02;

**预留数据位:** 固定为 0x00;

**落锁结果:** 0x00 代表落锁成功; 其他指令表示落锁失败。

**Demo1:** 同时落 2 个锁

服务器发送: 01 05 00 05 02 01 00 02 00 BE 56

设备端回复: 01 05 00 02 00 00 6C 0A

#### 4.3 查询门磁和锁舌状态 (0x03)

##### 4.3.1 查询单个门磁和锁舌状态

服务器发送:

板地址	指令字	数据长度	数据域	校验
0x01	0x03	0x00 0x03	锁号 (1)	CRC16

设备端回复:

板地址	指令字	数据长度	数据域	校验
0x01	0x03	0x00 0x01	门磁和锁舌状态 (1)	CRC16

**数据长度:** 指数据域里面内容的字节数量;

**锁号:** 指需要查询 1 号锁还是查询 2 号锁, 查询 1 号锁就是 0x01, 查询 2 号锁就是 0x02;

**门磁和锁舌状态:** 1 个字节, 高四位代表门磁状态, 低四位代表锁舌状态, 0 代表关, 1 代表开; 例如 0x0**0**, 高四位的 **0** 表示门磁关闭, 低四位的 **0** 表示锁舌落下; 再例如 0x**1****1**, 高四位的 **1** 表示门磁打开, 低四位的 **1** 表示锁舌打开。

**Demo1:** 查询 1 号锁的门磁和锁舌状态 (门磁关闭, 锁舌落下)

服务器发送: 01 03 00 01 01 D9 D4

设备端回复: 01 03 00 01 00 18 14

**Demo2:** 查询 1 号锁的门磁和锁舌状态 (门磁打开, 锁舌打开)

服务器发送: 01 03 00 01 01 D9 D4

设备端回复: 01 03 00 01 11 D8 18

**Demo3:** 查询 1 号锁的门磁和锁舌状态 (门磁关闭, 锁舌打开)

服务器发送: 01 03 00 01 01 D9 D4

设备端回复: 01 03 00 01 01 D9 D4

**Demo4:** 查询 2 号锁的门磁和锁舌状态（门磁关闭，锁舌落下）

服务器发送: 01 03 00 01 02 99 D5

设备端回复: 01 03 00 01 00 18 14

**Demo5:** 查询 2 号锁的门磁和锁舌状态（门磁打开，锁舌打开）

服务器发送: 01 03 00 01 02 99 D5

设备端回复: 01 03 00 01 11 D8 18

**Demo6:** 查询 2 号锁的门磁和锁舌状态（门磁关闭，锁舌打开）

服务器发送: 01 03 00 01 02 99 D5

设备端回复: 01 03 00 01 01 D9 D4

#### 4.3.2 查询 1 号和 2 号电插锁的门磁和锁舌状态

服务器发送:

板地址	指令字	数据长度	数据域	校验
0x01	0x03	0x00 0x02	锁号 (1) + 锁号 (1)	CRC16

设备端回复:

板地址	指令字	数据长度	数据域	校验
0x01	0x03	0x00 0x02	1 号电插锁的门磁和锁舌状态 (1) + 2 号电插锁的门磁和锁舌状态 (1)	CRC16

**数据长度:** 指数据域里面内容的字节数量;

**锁号:** 指需要查询 1 号锁还是查询 2 号锁, 查询 1 号锁就是 0x01, 查询 2 号锁就是 0x02;

**门磁和锁舌状态:** 1 个字节, 高四位代表门磁状态, 低四位代表锁舌状态, 0 代表关, 1 代表开; 例如 0x00, 高四位的 0 表示门磁关闭, 低四位的 0 表示锁舌落下; 再例如 0x11, 高四位的 1 表示门磁打开, 低四位的 1 表示锁舌打开。

**Demo1:** 查询 1 号和 2 号电插锁的门磁和锁舌状态

服务器发送: 01 03 00 02 01 02 64 5B

设备端回复: 01 03 00 02 00 00 E4 0A (1 号和 2 号电插锁的门磁都关闭, 锁舌都落下)

**Demo2:** 查询 1 号和 2 号电插锁的门磁和锁舌状态

服务器发送: 01 03 00 02 01 02 64 5B

设备端回复: 01 03 00 02 11 11 28 56 (1 号和 2 号电插锁的门磁都打开, 锁舌都打开)

**Demo3:** 查询 1 号和 2 号电插锁的门磁和锁舌状态

服务器发送: 01 03 00 02 01 02 64 5B

设备端回复: 01 03 00 02 00 11 24 06 (1 号电插锁的门磁关闭, 锁舌落下; 2 号电插锁的门磁打开, 锁舌打开)

**Demo4:** 查询 1 号和 2 号电插锁的门磁和锁舌状态

服务器发送: 01 03 00 02 01 02 64 5B

设备端回复: 01 03 00 02 11 00 E8 5A (1 号电插锁的门磁打开, 锁舌打开; 2 号电插锁的门磁关闭, 锁舌落下)

**Demo5:** 查询 1 号和 2 号电插锁的门磁和锁舌状态

服务器发送: 01 03 00 02 01 02 64 5B

设备端回复: 01 03 00 02 01 01 24 5A (1 号电插锁的门磁关闭, 锁舌打开; 2 号电插锁的门磁关闭, 锁舌打开)