

## 一、命令格式

### 命令例子：

自 2015 年 11 月 27 日起，本司所发 RS485 泊位锁均为一对多（即一个充电桩管理多个泊位锁）模式，协议修改如下：（以下命令均为 16 进制）

包头	地址	长度	命令	数据	校验码	包尾
0x55/0x5A/0x5B	ADDR	N+1	CMD	N	CRC-8	0xAA

包头：固定 0x55, 0x5A, 0x5B

地址：同一网关下每个锁地址唯一

长度：数据长度 N 字节+命令 1=N+1

命令：CMD

数据：长度 N

校验码：CRC-8/MAXIM X8+X5+X4+1，CRC 校验范围：长度+命令+数据，代码详见附录

包尾：固定 0xAA

### 命令及返回：

命令格式：0x55 + 地址 ADDR+长度 LEN + 命令 CMD + 数据 DATA + 校验码 CRC + AA

返回正确：0x5A + 地址 ADDR+长度 LEN + 命令 CMD + 数据 DATA + 校验码 CRC + AA

返回故障：0x5B + 地址 ADDR+长度 LEN + 命令 CMD + 故障码 DATA + 校验码 CRC + AA

序号	错误类	故障码
1	数据错误	01
8	执行失败	08

### 通信流程及升级内容：

1. 后台和各泊位锁（485 从机）之间采用严格的询问—应答的方式，泊位锁不会主动上传信息；
2. 后台控制各泊位锁（485 从机）开锁、闭锁之后，泊位锁（485 从机）返回应答（表示已接收），但不会主动上传操作结果；后台可以一段时间后查询锁状态获取开锁、闭锁结果；
3. 添加 ADDR 设置、查询指令，设置或者查询 ADDR 时建议单个操作
4. 添加波特率设置指令：波特率 bps 0—9600, 1—4800, 2—2400, 3—1200, 4—600
5. 添加声音报警输出设置、查询指令，可关闭声音报警，防止扰民
6. 添加软硬件版本号
7. 添加超声波车位探测的相关指令

### 具体命令：

1. **开锁命令：** 55 ADDR 01 01 CRC AA  
 接收成功返回 5A ADDR 02 01 01 CRC AA  
 接收失败返回 5B ADDR 03 01 01 00 CRC AA
2. **闭锁命令：** 55 ADDR 01 02 CRC AA  
 接收成功返回 5A ADDR 02 02 01 CRC AA  
 接收失败返回 5B ADDR 03 02 01 00 CRC AA
3. **读取锁状态：** 55 ADDR 01 06 CRC AA  
 执行成功返回 5A ADDR 02 06 STATUS CRC AA  
 接收失败返回 5B ADDR 03 06 01 00 CRC AA

注：00 闭锁状态，01 开锁状态，02 下降遇阻，03 上升遇阻并且恢复，88 运动状态

**10 当前开锁状态并且检测到上面无车（地锁请求无车升起）——重要**

4. **设置超声波检测周期：** 55 ADDR 02 07 DATA CRC AA（S 为单位时间）

执行成功返回 5A ADDR 02 07 00 CRC AA

5. **读超声波检测周期：** 55 ADDR 01 08 CRC AA

执行成功返回 5A ADDR 02 08 DATA CRC AA

6. **设置超声波滤波时间：** 55 ADDR 02 09 DATA CRC AA（S 为单位时间）

执行成功返回 5A ADDR 02 09 00 CRC AA

注：DATA—滤波时间，例如 60 表示连续检测到无车 60s 后才会执行自动升起流程

7. **读取超声波滤波时间：** 55 ADDR 01 0A CRC AA

执行成功返回 5A ADDR 02 0A DATA CRC AA

注：DATA—滤波时间，例如 60 表示连续检测到无车 60s 后才会执行自动升起流程

8. **超声波运行参数查询：** 55 ADDR 01 14 CRC AA（调试时应用）

执行成功返回 5A ADDR 03 14 CSB\_TIME CSB\_NUM CRC AA

注：CSB\_TIME 超声波的检测周期计时，CSB\_NUM 连续检测上面无车计时，定义详见本文第二部分—超声波运行逻辑

9. **蜂鸣器输出设置/查询：** 55 ADDR 02 15 DATA CRC AA

55 ADDR 02 15 00 5C AA

55 ADDR 02 15 01 02 AA

55 ADDR 02 15 02 E0 AA

执行成功返回 5A ADDR 02 15 00/01 CRC AA

接收/执行失败返回 5B ADDR 03 15 01/08 00 CRC AA

注：00—蜂鸣器不输出，01—蜂鸣器输出，02—蜂鸣器输出查询

10. **软硬件版本号读取：** 55 ADDR 01 1A CRC AA

55 ADDR 01 1A 27 AA

执行成功返回 5A ADDR 03 1A SW HW CRC AA

接收失败返回 5B ADDR 03 1A 01 00 CRC AA

（程序固化时由客户提供）

11. **设置超声波探测开关/查询：** 55 ADDR 02 1B DATA CRC AA

55 00 02 1B 00 80 AA

55 00 02 1B 01 DE AA

55 00 02 1B 02 3C AA

执行成功返回 5A ADDR 02 1B 00/01 CRC AA

执行失败返回 5B ADDR 03 1B 08 00 CRC AA

注：当超声波失效之后，可以选择关闭超声波用于恢复到手动关锁，00—打开，01—关闭，02—查询

12. **ADDR 设置：** 55 ADDR 02 1C DATA CRC AA

55 00 02 1C 01 B0 AA

55 00 02 1C 02 52 AA

执行成功返回 5A ADDR 02 1C 00 CRC AA

- 接收/执行失败返回 5B ADDR 03 1C 01/08 00 CRC AA
13. **ADDR 查询:** 55 ADDR(FF) 01 1D CRC AA  
 55 FF 01 1D A4 AA  
 执行成功返回 5A ADDR 02 1D DATA(ADDR) CRC AA
14. **波特率设置:** 55 ADDR 02 1E DATA CRC AA  
 55 00 02 1E 01 21 AA  
 执行成功返回 5A ADDR 02 1E 00 CRC AA  
 接收/执行失败返回 5B ADDR 03 1E 01/08 00 CRC AA
15. **网络 MAC 地址查询:** 55 ADDR 01 22 CRC AA  
 执行成功返回 5A ADDR 07 22 XX XX XX XX XX XX CRC AA (MAC 地址 6byte)

**注 1:** 后台设备控制建议:

1. RS485 数据发出后, 等待, 若 2s 内无接收返回, 则重发;
2. 开锁关锁由于需要一定时间, 第一条快速应答表示已收到, 充电桩接收到应答后可间隔查询锁状态, 以判断当前操作成功与否;

## 附录: CRC8 校验源代码:

```
code unsigned char code CRC8_TAB[256] =
{
    0x00, 0x5e, 0xbc, 0xe2, 0x61, 0x3f, 0xdd, 0x83,
    0xc2, 0x9c, 0x7e, 0x20, 0xa3, 0xfd, 0x1f, 0x41,
    0x9d, 0xc3, 0x21, 0x7f, 0xfc, 0xa2, 0x40, 0x1e,
    0x5f, 0x01, 0xe3, 0xbd, 0x3e, 0x60, 0x82, 0xdc,
    0x23, 0x7d, 0x9f, 0xc1, 0x42, 0x1c, 0xfe, 0xa0,
    0xe1, 0xbf, 0x5d, 0x03, 0x80, 0xde, 0x3c, 0x62,
    0xbe, 0xe0, 0x02, 0x5c, 0xdf, 0x81, 0x63, 0x3d,
    0x7c, 0x22, 0xc0, 0x9e, 0x1d, 0x43, 0xa1, 0xff,
    0x46, 0x18, 0xfa, 0xa4, 0x27, 0x79, 0x9b, 0xc5,
    0x84, 0xda, 0x38, 0x66, 0xe5, 0xbb, 0x59, 0x07,
    0xdb, 0x85, 0x67, 0x39, 0xba, 0xe4, 0x06, 0x58,
    0x19, 0x47, 0xa5, 0xfb, 0x78, 0x26, 0xc4, 0x9a,
    0x65, 0x3b, 0xd9, 0x87, 0x04, 0x5a, 0xb8, 0xe6,
    0xa7, 0xf9, 0x1b, 0x45, 0xc6, 0x98, 0x7a, 0x24,
    0xf8, 0xa6, 0x44, 0x1a, 0x99, 0xc7, 0x25, 0x7b,
    0x3a, 0x64, 0x86, 0xd8, 0x5b, 0x05, 0xe7, 0xb9,
    0x8c, 0xd2, 0x30, 0x6e, 0xed, 0xb3, 0x51, 0x0f,
    0x4e, 0x10, 0xf2, 0xac, 0x2f, 0x71, 0x93, 0xcd,
    0x11, 0x4f, 0xad, 0xf3, 0x70, 0x2e, 0xcc, 0x92,
    0xd3, 0x8d, 0x6f, 0x31, 0xb2, 0xec, 0x0e, 0x50,
    0xaf, 0xf1, 0x13, 0x4d, 0xce, 0x90, 0x72, 0x2c,
```

```
0x6d, 0x33, 0xd1, 0x8f, 0x0c, 0x52, 0xb0, 0xee,  
0x32, 0x6c, 0x8e, 0xd0, 0x53, 0x0d, 0xef, 0xb1,  
0xf0, 0xae, 0x4c, 0x12, 0x91, 0xcf, 0x2d, 0x73,  
0xca, 0x94, 0x76, 0x28, 0xab, 0xf5, 0x17, 0x49,  
0x08, 0x56, 0xb4, 0xea, 0x69, 0x37, 0xd5, 0x8b,  
0x57, 0x09, 0xeb, 0xb5, 0x36, 0x68, 0x8a, 0xd4,  
0x95, 0xcb, 0x29, 0x77, 0xf4, 0xaa, 0x48, 0x16,  
0xe9, 0xb7, 0x55, 0x0b, 0x88, 0xd6, 0x34, 0x6a,  
0x2b, 0x75, 0x97, 0xc9, 0x4a, 0x14, 0xf6, 0xa8,  
0x74, 0x2a, 0xc8, 0x96, 0x15, 0x4b, 0xa9, 0xf7,  
0xb6, 0xe8, 0x0a, 0x54, 0xd7, 0x89, 0x6b, 0x35
```

```
};
```

```
//precrc 为 CRC-8 初始因子，如无特殊规定，则默认为 0x00
```

```
unsigned char CRC8_Tab( unsigned char *ptr, unsigned char len, unsigned char precrc )  
{  
    unsigned char index;  
    unsigned char crc8=precrc;  
    while( len-- )  
    {  
        index = crc8 ^ ( *(ptr++) );  
        _nop_();  
        _nop_();  
        crc8 =CRC8_TAB[ index ];  
        _nop_();  
        _nop_();  
    }  
    _nop_();  
    _nop_();  
    return( crc8 );  
}
```