



# 电机驱动板 使用说明书

## 概述

本文档描述了电机驱动板的使用、配置和控制方法。

## 目录

一、 产品简介.....	1
二、 硬件功能.....	3
2.1 硬件接口.....	3
2.2 电磁铁或两线制电机接线方式.....	4
2.3 三线制电机接线方法.....	5
三、 通讯协议.....	6
3.1 通讯参数.....	6
3.2 指令格式.....	6
3.3 详细指令格式.....	7
3.4 CRC 校验参考代码.....	10

## 一、产品简介

杭州浩聚致力于物联网设备的开发，有着多年物联网网关的技术积累，形成了产品软硬件开发、硬件设备生产供应为主的科技化公司，对于各种网络制式、网络通讯协议、设备接口协议有着丰富的理解和经验，诚心期待您的合作。

浩聚电机驱动板主要是用于格子柜类产品电机和控制锁的驱动控制，控制板通过串口、485 接口、232 接口与主控制器进行数据通讯，最多可以驱动 100 负载，负载的类型可以是：无反馈电磁锁、有反馈电磁锁、两线制售货机电机、三线制售货机电机。

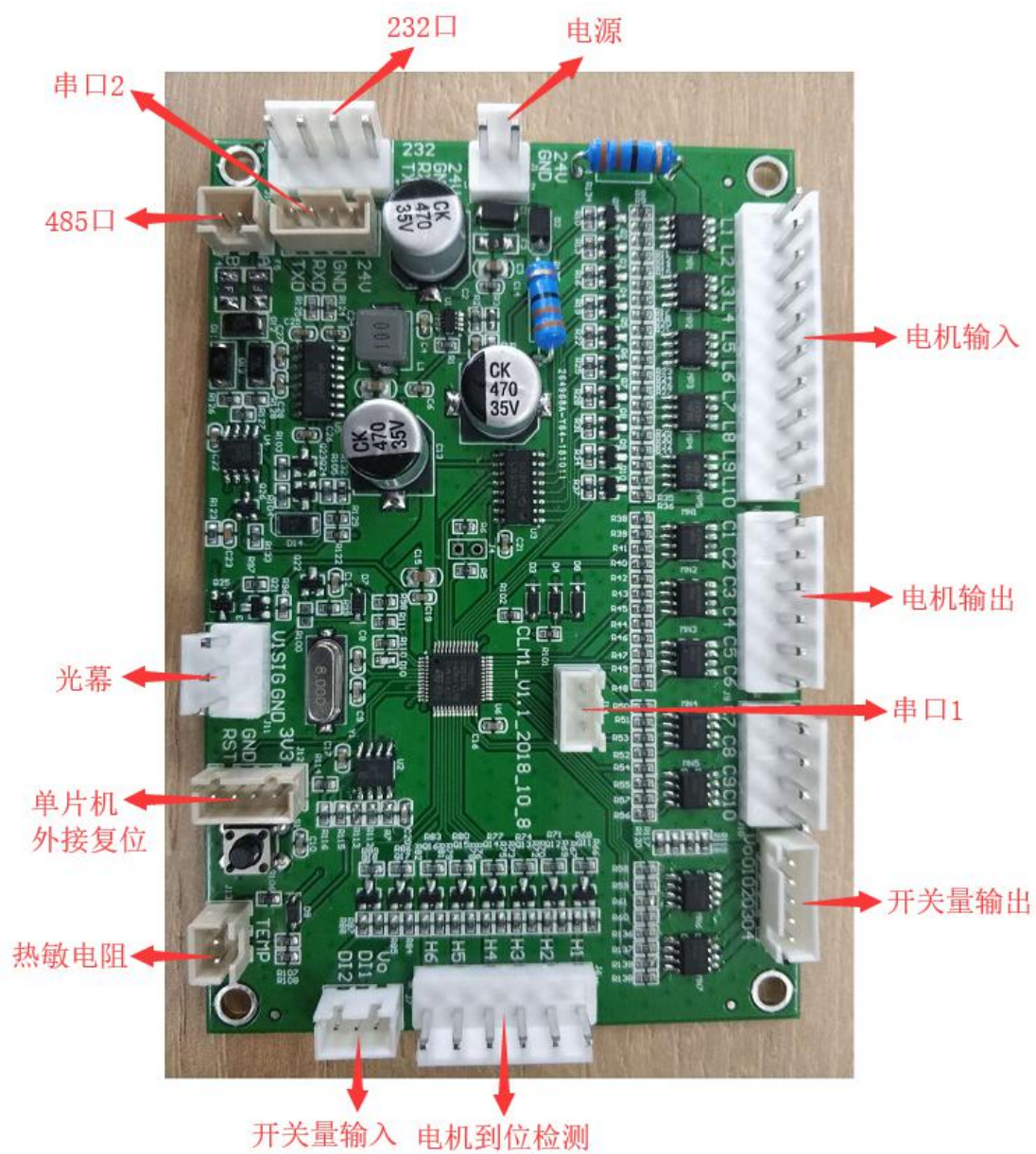
电机驱动板硬件接口配置如下：

型号	CM
通讯接口	2 路串口/1 路 485/1 路 232
电机接口	10 路输入/10 路输出/6 路到位检测
测温接口	1 路热敏电阻（NTC）接口
开关量接口	2 路输入/4 路输出
光幕接口	1 路
电源	12V/24V

浩聚电机驱动板的特色如下：

- ⇒ 意法半导体芯片控制
- ⇒ 矩阵式电路结构
- ⇒ 支持 100 通道的出货机构驱动，60 电机 40 电磁锁
- ⇒ 三线制电机和两线制电机混合使用
- ⇒ 电机与电磁锁混合使用
- ⇒ 支持 1 路温度检测，匹配 NTC 检测单元
- ⇒ 增强型灵活通讯口，可以方便支持串口、RS-232、RS-485
- ⇒ 开关量输出：4 路开关量输出
- ⇒ 开关量输入：2 路开关量输入
- ⇒ 温度采样：1 路温度采样

各接口示意图如下：



## 二、硬件功能

### 2.1 硬件接口

#### (1) 供电电源

供电电压：DC12V(兼容 12V)。电源功率不低于 50W。

#### (2) 负载

最多可以驱动 100 负载，每路最大电流 3A。负载的类型可以是：无反馈电磁锁、有反馈电磁锁、两线制售货机电机、三线制售货机电机。

#### (3) 开关量输出

驱动板支持 4 路开关量输出回路，编号依次为 01、02、03、04，对外输出 0V 或者高阻抗。支持吸入电流 50mA 每路（足够驱动普通 24V 继电器）。

#### (4) 开关量输入

驱动板支持两路开关量输入回路，编号为 DI1、DI2，可以输入 12V（24V）或者开关按钮信号。

#### (5) 测温回路

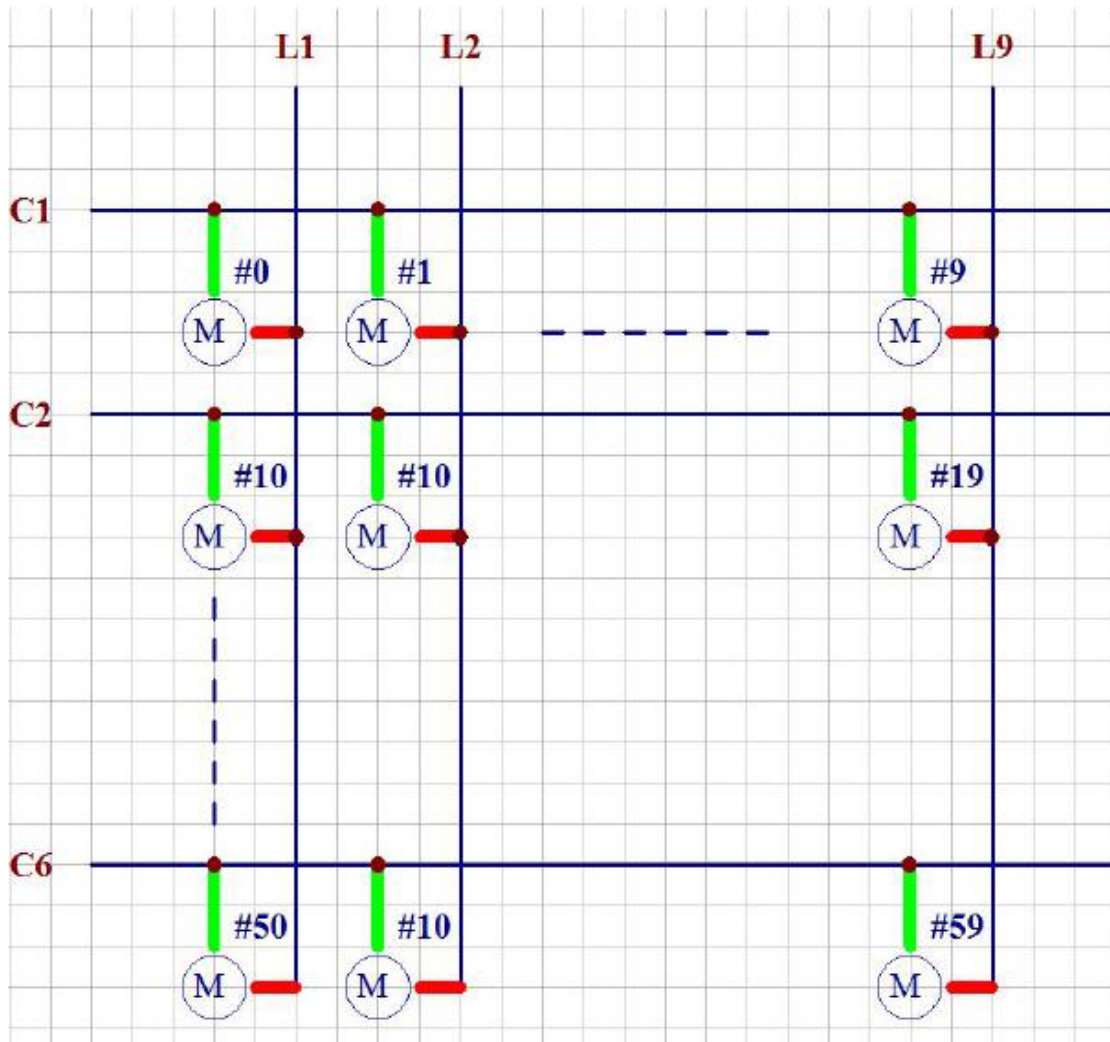
带有 1 路测温回路，配套的传感器信号为 NTC 10K@25℃ B=3950。温度测量范围为-50℃~50℃, 未接传感器时系统返回-50℃。系统测量 NTC 电阻阻值，然后通过查表获取当前温度值。

#### (6) 光幕控制

驱动板带有一路光幕接口，信号线有 V1、SIG、GND。V1 用来给光幕供电，SIG 光幕信号，GND 是光幕的电源线。

## 2.2 电磁铁或两线制电机接线方式

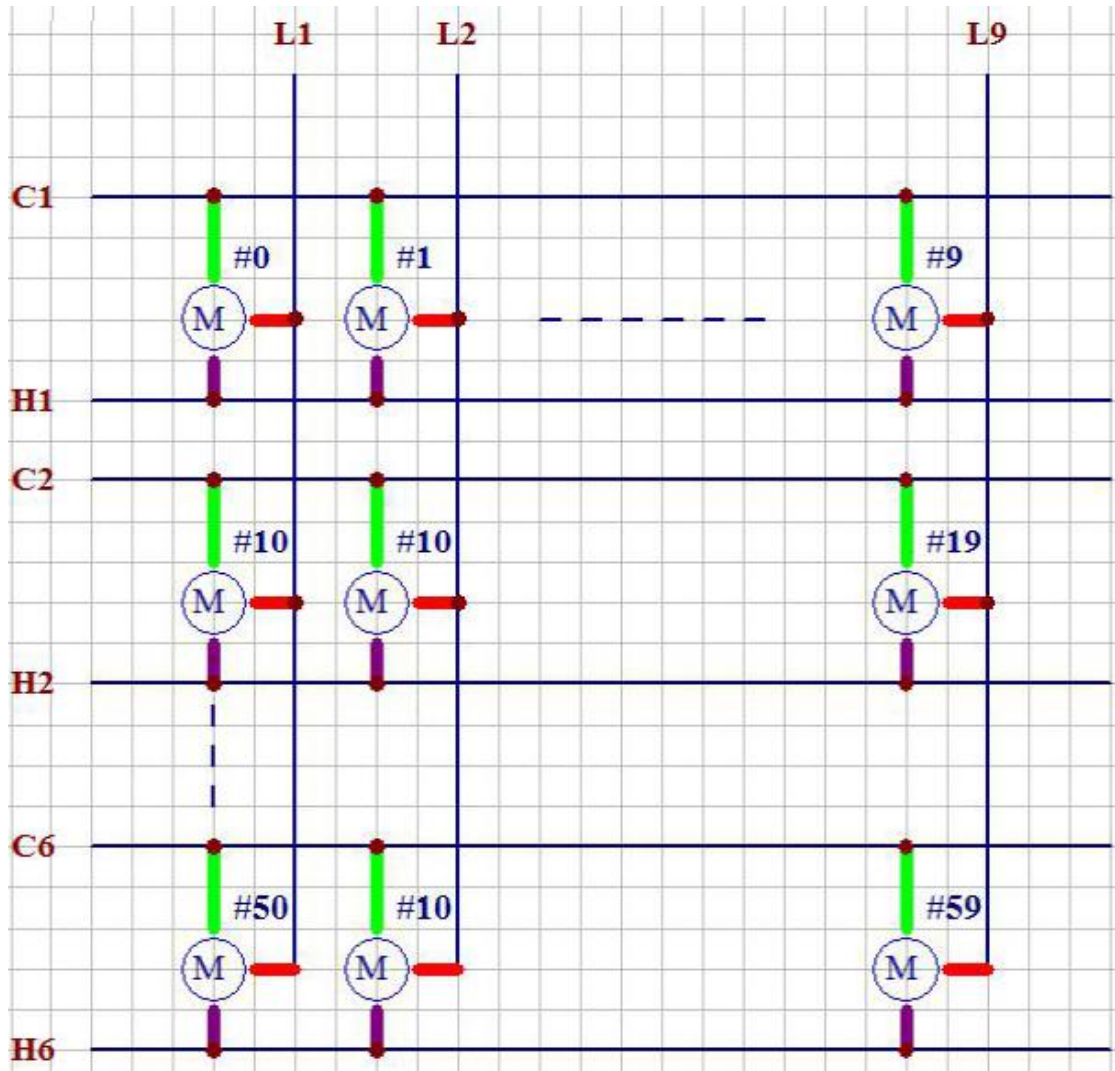
两线制接口需要使用到的信号是 L1~L10, C1~C6, C7~C10。L 信号输出 24V 正极, C 信号输出 0V 负极。L 和 C 之间连接负载形成回路。两线制电机接线方式如下图所示:



图中的电机, 实际是安装过选通二极管、续流二极管、位置检测微动开关的组合, 并非普通的电机。

## 2.3 三线制电机接线方法

三线制接口需要使用到的信号是 L1~L10, C1~C6, H1~H6。L 信号输出 12V 正极，C 信号输出 0V 负极。L 和 C 之间连接负载形成回路。H 信号是电机到位检测信号，当 H 端检测到电机到位信号时，H 端信号由高电平变为低电平。三线制电机接线方式如下图所示：



## 三、通讯协议

### 3.1 通讯参数

控制板可以通过 TTL/RS-485/RS-232 的方式与上位机进行通讯。通讯参数规定为：**9600bps，1 停止位，无奇偶校验，无流量控制。**

在通讯过程中，采用主从方式上位机与驱动板进行通讯，通讯遵循以下规范：

- (1) 每次通讯由主机发起请求，从机应答
- (2) 主机收到上次请求的应答或者超时后才可发起新的请求
- (3) 在一个通讯网络中可以包含一台主机和若干台从机，通过设备地址来区分这些设备，其中主机的地址为 0，驱动板的地址为 1~8。
- (4) 地址 255 作为广播地址，此地址作为从机地址设置使用。
- (5) 驱动板上电开机后，板子上指示灯闪烁的次数即为它的通讯地址。

### 3.2 指令格式

功能码列表

功能码	功能名称
01	获取序列号
03	查询 Run 执行状态
04	测试电机
05	启动电机
07	读取温度
08	开关量输出
09	读开关量状态
FF	设置地址

数据包定义

名称	格式	长度	内容
从机地址	Int	1byte	主机为 0，从机为 1~8
指令功能码	Int	1byte	
数据	Int	16byte	不足 16 位，后面补 0
校验位	Int	2byte	CRC 校验

补充：主机->从机，数据用 Y1~Y16 表示；从机->主机，数据用 Z1~Z16。  
16 字节整数的高字节在前，低字节在后。

### 3.3 详细指令格式

1) 获取序列号

功能码: 01

上位机发起

数据包为空

响应:

上位机地址	指令	数据	校验
0	01	Z1~Z12:序列号	CRC 校验

2) 查询 Run 执行状态

功能码: 03

上位机发起

从机地址	指令	数据	校验
1~8	03	Y1~Y16:0	CRC 校验

响应:

上位机地址	指令	数据	校验
0	03	Z1~Z10:查询结果	CRC 校验

Z1 Run 执行状态, 0: 空闲, 1: 执行中, 2: 执行完毕;

Z2 正在运转的电机编号, 00~59 (共 60 个);

Z3 执行结果, 0: 成功, 1: 过流 (负载过重或者卡货), 2: 欠流 (一般是电线断开或者负载未安装), 3: 超时 (7 秒仍未检测到电机到位信号), 4: 光幕自检失败 (未启动电机), 5: 有反馈电磁门未开;

Z4~Z5 峰值电流, 单位 mA;

Z6~Z7 平均电流, 单位 mA;

Z8~Z9 运行时间, 单位 ms;

Z10 光幕状态, 0: 无掉货, 1~200 表示货物经过光栅的时间, 单位 ms;

3) 测试电机

功能码: 04

上位机发起

从机地址	指令	数据	校验
1~8	04	Y1: 电机索引号, Y2~Y16:0	CRC 校验

响应:

从机地址	指令	数据	校验
0	04	Z1: 测试结果, Z2~Z16:0	CRC 校验

Z1 AA:正常, BB: 异常, CC 过载

#### 4) 启动电机

功能码：05

从机地址	指令	数据	校验
1~8	05	Y1~Y6:启动方式	CRC 校验

Y1 电机索引号：00~59

Y2 电机类型，0：无反馈电磁铁，1：有反馈电磁铁，2：两线制电机，3：三线制电机；

Y3 光幕工作模式，0：电机运转不参考光幕，1：启动电机前，需光幕自检成功，电机到位后停止；

Y4 电机过流阈值，1~255 代表 10mA~2550mA, 0 表示默认值；

Y5 电机欠流阈值，1~255 代表 10mA~2550mA, 0 表示默认值；

Y6 电机到位超时时间，1~100 代表 0.1S~10.0S, 0 表示默认值（7 秒）；

响应：

从机地址	指令	数据	校验
0	05	Z1: 执行结果	CRC 校验

Z1 0：已启动，1：无效的电机索引号，2：另一台电机正在运行。

#### 5) 读温度

功能码：07

上位机发起，数据包为空

响应：

从机地址	指令	数据	校验
0	07	Z1~Z2: 温度值	CRC 校验

温度范围-50℃~50℃，未安装传感器时为-50。

#### 6) 开光量输出（4 路）

功能码：08

上位机发起：

从机地址	指令	数据	校验
1~8	08	Y1: D0 索引（0~3）， Y2: 操作（1: ON, 0: OFF）	CRC 校验

响应：

从机地址	指令	数据	校验
0	08	Z1: D0 索引（0~3）， Z2: 操作结果	CRC 校验

7) 读取开关量状态 (2 路)

功能码: 09

上位机下发, 数据包为空

响应:

从机地址	指令	数据	校验
0	09	Z1~Z2: 开关量状态	CRC 校验

Z1~Z2==DI1~DI2    1:外部连接, 0: 外部断开

8) 设置板卡地址

功能码: FF

上位机下发

从机地址	指令	数据	校验
FF	FF	Y1: 驱动板新地址	CRC 校验

响应:

从机地址	指令	数据	校验
0	FF	Z1: 驱动板新地址	CRC 校验

设置板卡地址步骤:

- 整个网络中只保留主机和一块驱动板;
- 上位机发送设置地址指令, 人工确认指示灯闪烁次数;
- 该网络恢复原样。

注意: 该指令的通讯地址采用广播地址 (0xFF)。

### 3.4 CRC 校验参考代码

```
/******  
功能： CRC 校验  
pcMess：待求数据的首指针  
wLen ：待求数据的长度  
*****/  
unsigned int crcVall(unsigned char *pcMess,unsigned int wLen)  
{  
    long MSBInfo;  
    int i,j ;  
    unsigned int nCRCData;  
    nCRCData = 0xffff;  
  
    for(i = 0; i < wLen;i++)  
    {  
        pcMess[i]&=0x00ff;  
        nCRCData = nCRCData ^ pcMess[i] ;  
  
        for(j= 0 ; j < 8 ;j ++)  
        {  
            MSBInfo = nCRCData & 0x0001;  
            nCRCData = nCRCData >> 1;  
  
            if(MSBInfo != 0 )  
                nCRCData = nCRCData ^ 0xa001;  
        }  
    }  
  
    return nCRCData;  
}
```