1. **目录**

[1. 目录 1](#_Toc6237)

[2. 版本修订 2](#_Toc27503)

[3. 产品概览 2](#_Toc31564)

[4. 电气特性说明 3](#_Toc10797)

[5. 安装尺寸 5](#_Toc21777)

[6. 接线方式 6](#_Toc29948)

[6.1 电源和通讯线的连接 6](#_Toc7651)

[6.2 电磁铁或两线制电机接线方式 6](#_Toc2534)

[6.3 三线制电机接线方法 8](#_Toc20392)

[6.4 光幕接线方式 9](#_Toc26417)

[6.5 温度检测接线 9](#_Toc5024)

[6.6 开关量输出端口 9](#_Toc27876)

[6.7 开关量输入端口 9](#_Toc24147)

[7. 通讯参数 10](#_Toc1923)

[8. 指令格式 10](#_Toc29675)

[9. 详细指令模式 11](#_Toc5499)

[9.1 获取序列号(ID) 指令:01H 11](#_Toc24675)

[9.2 查询Run执行状态(Motor Poll)指令:03H 11](#_Toc1299)

[9.3 测试(Motor Scan)指令:04H 11](#_Toc2613)

[9.4 启动电机(Motor Run)指令：05H 11](#_Toc22250)

[9.5 读温度( Read Temp)指令 07H 12](#_Toc31594)

[9.6 开关量输出(Write DO)指令08H 12](#_Toc20798)

[9.7 读开关量状态(Read DI)指令09 12](#_Toc7225)

[9.8 设置板卡地址Set Address 指令 FF 13](#_Toc21137)

[10. 样例 13](#_Toc13213)

[11. CRC程序参考代码 14](#_Toc26223)

1. **版本修订**

2017-11-30 全新制定 Sinshine

1. **产品概览**

*  意法半导体芯片控制
*  精密逻辑控制电路
* 矩阵式电路结构,板载矩阵交叉电路，减少线束成本
* 支持100通道的出货机构驱动, 60电机 40 电磁锁
* 三线电机和两线电机混合使用
* 电机与电磁铁混合使用
* 支持最多255块控制卡并联扩展，实现无限扩展
* 增强型灵活通讯口，板载TTL,RS-232、RS-485接口
* 高抗干扰设计
* 掉货检测接口
* **开关量输出：8路开关量输出**
* **开关量输入：8路开关量输入**
* **温度采样： 1路温度采样**

1. **电气特性说明**

**供电电压：DC24V （12V可选）**

**静态电流：小于20mA**

**驱动电流：最大1A**

**通讯接口： RS-232**

1. **供电电源**

供电电压：DC24V （可定制DC12V版本）。电源功率不低于50W。

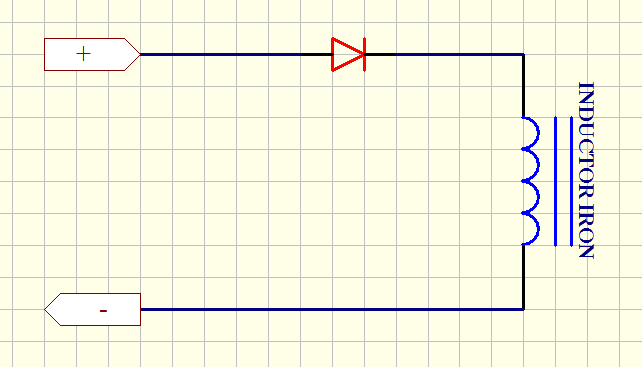
1. **负载**

支持3线制马达 80个

支持2限制马达100个

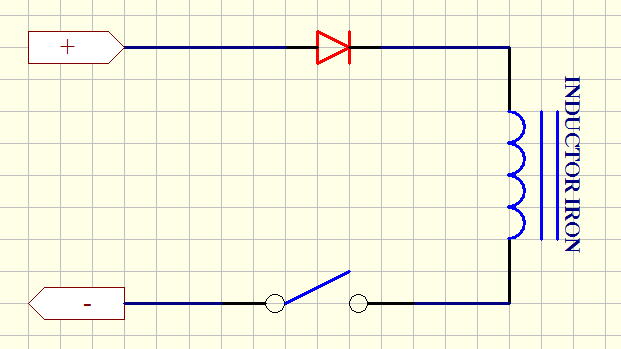
支持 电磁锁 100个

1. 无反馈电磁铁，可以驱动100路

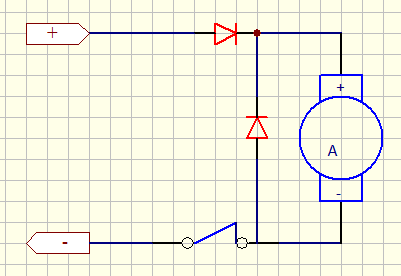


1. 有反馈电磁铁，可以驱动100路

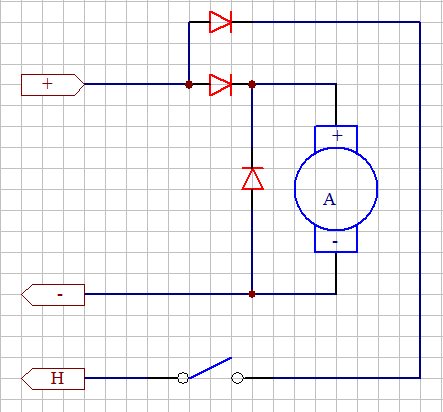
当电磁锁关闭时，开关处于闭合状态；当门弹开后，开关处于断开状态



1. 两线制电机，可以驱动100路



1. 三线制电机，可以驱动80路



1. **开关量输出**

带有8路开关量输出回路，系统id为0~7。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）。

1. **开关量输入**

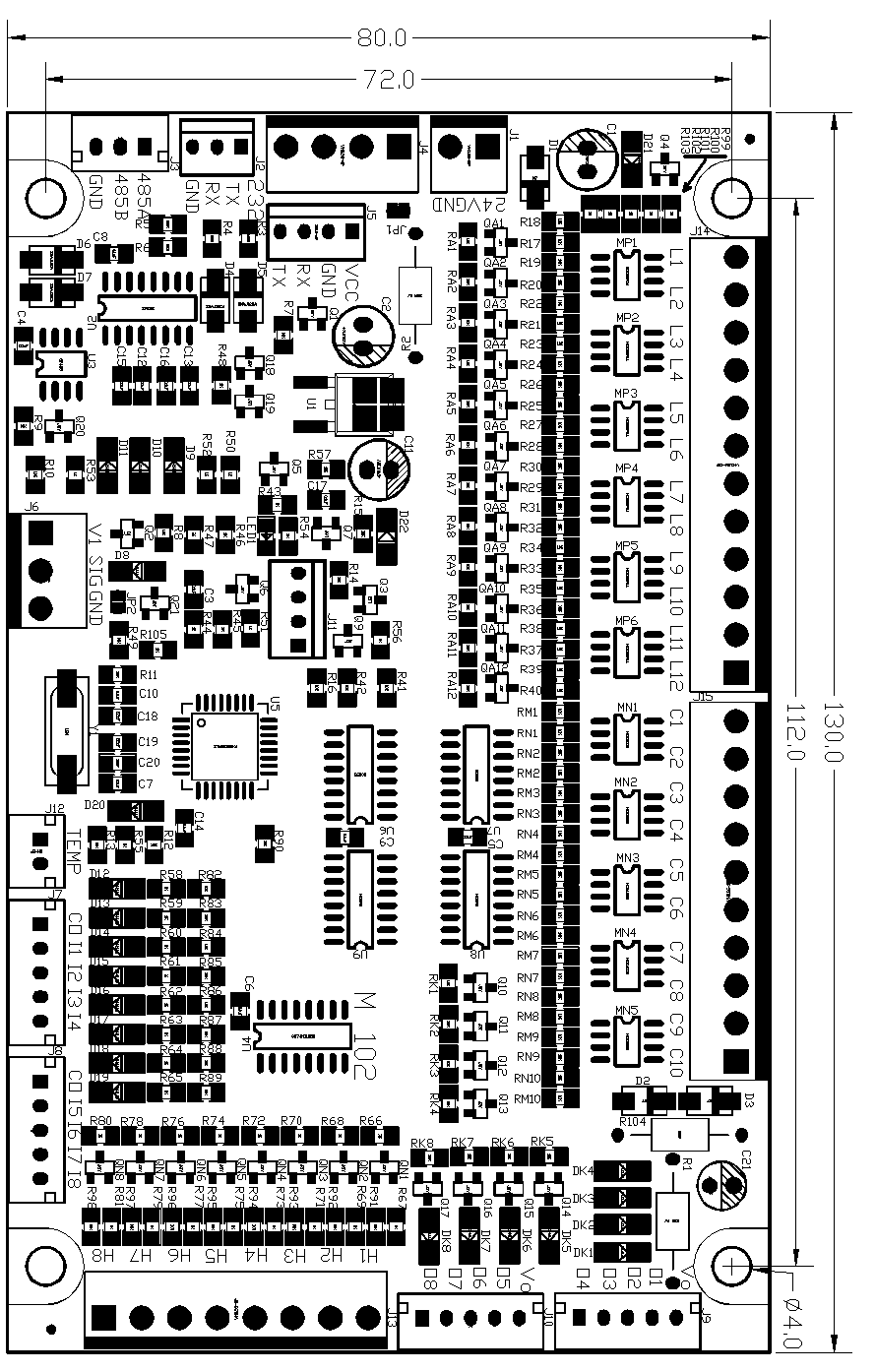
有8路开关量输入回路，编号为0~7。外部可以输入0~24V 或者开关按钮信号。

1. **测温回路**

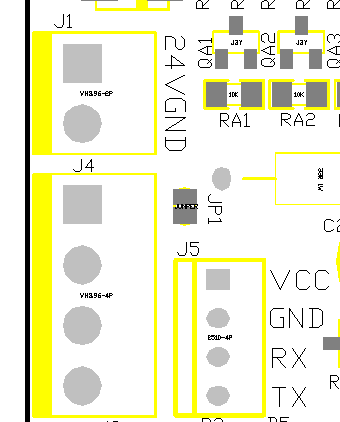
带有1路测温回路，配套传感器型号为NTC 10K@25℃ B=3950。温度测量范围为-50.0℃~50.0℃，未接传感器时系统返回-50.0℃。

1. **安装尺寸**

**单位(mm) 孔径4mm**



1. **接线方式**
   1. **电源和通讯线的连接**



供电电压24V，12V可选

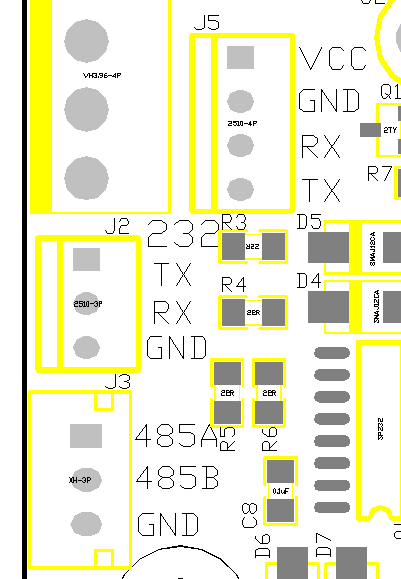
静态电流 小于40mA

最大电流取决于负载电流。

J1 供电

J4 为了兼容 M101G

J5 为了兼容 M101H\M101J



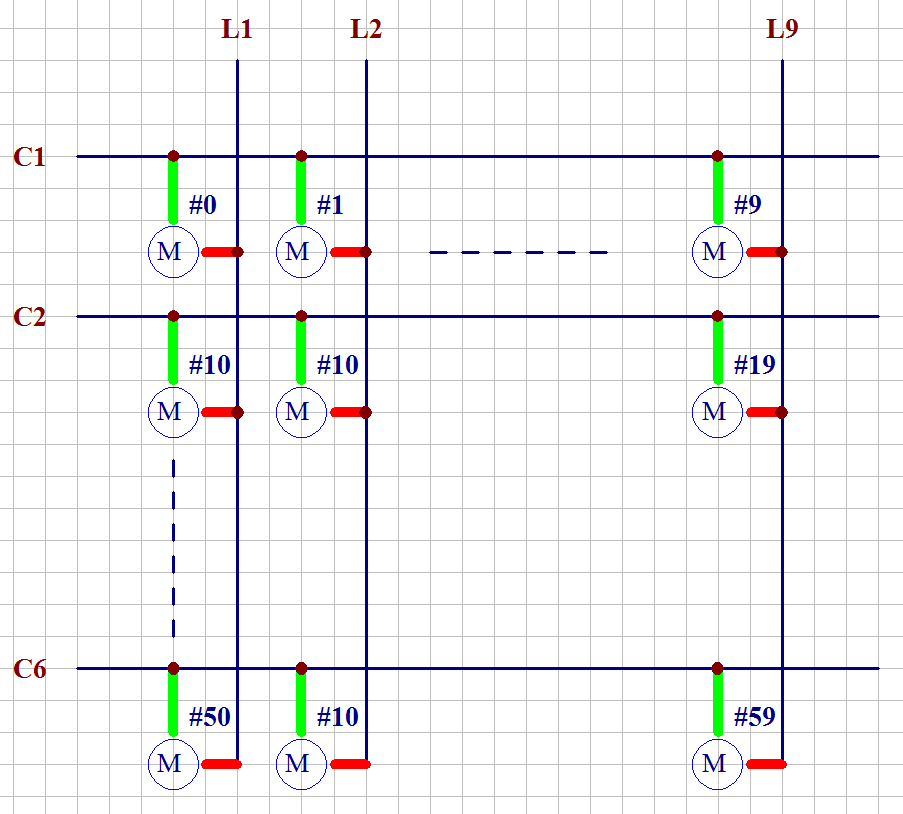
J5 TTL 通讯接口

J2 RS-232 接口

J3 RS-485 接口

* 1. **电磁铁或两线制电机接线方式**

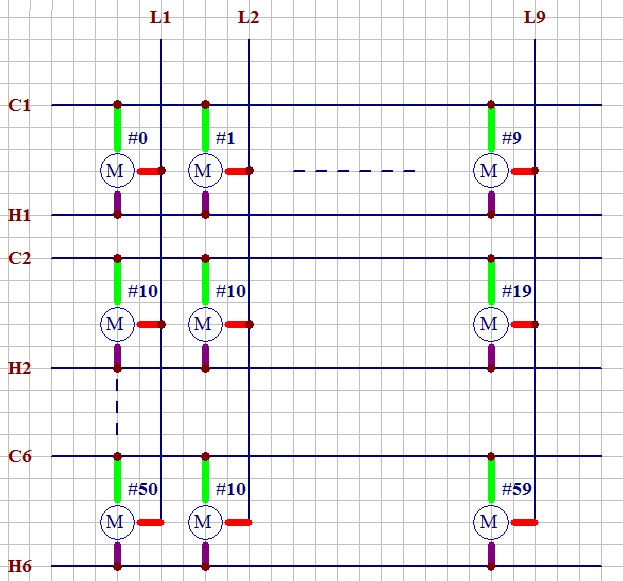
两线制接口需要使用到的信号是L1~L10 ; C1~C6 ，C7~C10 。 L 信号输出24V正极，C信号输出0V。L与C之间接负载形成回路。



**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，并非普通电机。**

* 1. **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24V正极，C信号输出0V。L与C之间接负载形成回路。H信号是电机到位的检测信号，当电机停止时是高阻抗，当电机运转过程中是24V。



**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，并非普通电机。**

* 1. **光幕接线方式**

光幕接口 信号 V1；SIG;GND。 V1给光幕供电，SIG光幕信号，GND是光幕的电源地线。

待机时：V1 无输出，光幕不供电。 此时 SIG需要是，高阻抗或者断路。

启动马达前，V1输出24V给光幕供电，此时SIG需要是0V（或者说与GND通路）。货物掉落经过光幕时，SIG信号是有高阻抗或者断路。

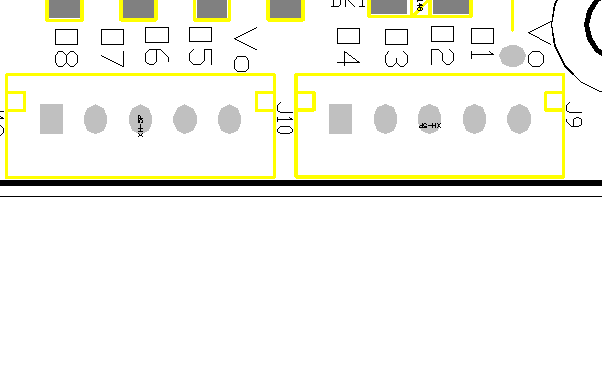
* 1. **温度检测接线**

TEMP接线，仅需要接上 ，规格符合热敏电阻即可。

* 1. **开关量输出端口**

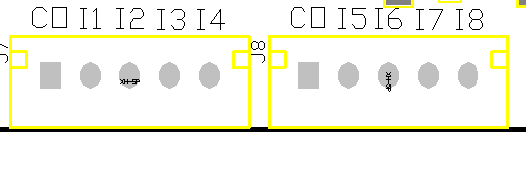
**Vo输出24V**

**O1，O2，O3，O4 ,O5,O6,O7,O8吸入电流（负载能力40mA）可以驱动24V继电器，驱动光耦时需要串联10K 1/4W电阻。**

****

* 1. **开关量输入端口**

**CO 端口是公共线，I1，I2，I3，I4,I5,I6,I7,I8 是每路的输入点，可以接入外部开关信号**



1. **通讯参数**

控制卡采用TTL串口方式进行通讯,可以方便转换成232或者485电平。

**通讯参数： 9600bps，1停止位，无奇偶校验，无流量控制**

在通讯过程中，上位机（安卓或者PC，称为主机）与控制卡（称为从机）采用主从方式

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 或者超时后才可以发起新的请求
3. 在一个通讯网络中可包含一台主机和若干台从机
4. 在一个通讯网络中可包含一台主机和若干台从机，通过个设备地址(1字节)，来区分这些设备。其中主机的地址为0，MS101G控制卡的地址为1-8。
5. 此外地址255作为广播地址。目前此地址只用作从机地址修改。**（慎重使用）**
6. 控制卡卡上的指示灯的连续闪烁次数即为它的通讯地址。

通讯流程如下：

1. 主机下发指令
2. 等待50ms
3. 接收从机响应(超时1秒)
4. **指令格式**

为了便于上位机程序的编写，通讯协议中使用了定长数据帧，一帧指令或响应的长度为20字节，包含了地址，指令,数据段及校验代码四部分。当数据段中有效数据小于16字节时，需要填0补充至16字节。

格式如下：

**指令：主机 -> 从机**

|  |  |  |  |
| --- | --- | --- | --- |
| 从机地址（1byte） | 指令（1byte） | 数据（16byte） | 校验代码(2byte) |

**数据 用 Y1~Y16表达**

**响应：从机->主机**

|  |  |  |  |
| --- | --- | --- | --- |
| 主机地址（1byte） | 指令（1byte） | 数据（0-16byte） | 校验代码(2byte) |

数据用Z1~Z16表达

**地址：**

从机地址 1~8，主机地址为0。

**指令：**

详见指令表

**数据：**

数据段可包含16字节，指令中的数据用Y1-Y16表达，响应中的数据用Z1-Z16表达。数据段中可包含**单字节和双字节**两种整数，其中16位整数的高字节在前，低字节在后。

**校验代码：**

采用CRC16校验方式，运算过程见附录。校验代码的低字节在前、高字节在后。

1. **详细指令模式**
   1. 获取序列号(**ID) 指令:01H**

|  |  |  |
| --- | --- | --- |
| Z1-Z12 | 控制卡的序列号 |  |
|  |  |  |

* 1. **查询Run执行状态(Motor Poll)指令:03H**

|  |  |  |
| --- | --- | --- |
| Z1 | Run指令执行状态 | 0 = ----  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。指超过7秒仍未检测到电机到位信号。（一般是由于负载过重、卡货或开关电源干扰引起）  4 = 光幕自检失败，未启动电机。  5= 有反馈电磁铁门未开 |
| 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） |

* 1. **测试(Motor Scan)指令:04H**

|  |  |  |
| --- | --- | --- |
| Y1 | 电机索引号 | 00~99 |
| Z1 | 结果 | AA=正常 BB=异常 CC=过载 |

* 1. **启动电机(Motor Run)指令：05H**

|  |  |  |
| --- | --- | --- |
| 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  6=强三履带 |
| 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。  2 =启动电机前，需光幕自检成功；当光幕检测到掉货后电机立即停止，未检测到掉货时电机，一直运转直到超时（7s） |
| 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) |
| Z1 | 指定执行结果 | 0=已启动；  1=无效的电机索引号；  2=另一台电机在运行； |

* 1. **读温度( Read Temp)指令 07H**

本驱动板卡带有一路温度采样回路

|  |  |  |
| --- | --- | --- |
| Z1-Z2 | 温度值 | -50.0℃~50.0℃，未安装传感器时为-50.0℃。 |

FE0C = -500 代表-50.0

01F4 = 500 代表 50.0读

* 1. **开关量输出(Write DO)指令08H**

本驱动卡带有4路开关量输出回路，编号为0~3。

|  |  |  |
| --- | --- | --- |
| Y1 | DO 索引 | 0-6 |
| Y2 | 操作 | 1=ON 0=OFF |
| Z1 | DO 索引 | 0-6 |
| Z2 | 操作结果 | Z2=Y2+F0 . 00->F0 01->F1 |

* 1. **读开关量状态(Read DI)指令09**

本驱动卡带有4路开关量输入回路，编号为0~3。

|  |  |  |  |
| --- | --- | --- | --- |
| 主机下发 | | | |
| 从机地址 | 指令 | 数值 | 校验代码 |
| 01~08 | 09 | Y1~Y16 | CRC-16 |

Y1=0

|  |  |  |  |
| --- | --- | --- | --- |
| 从机应答 | | | |
| 主机地址 | 指令 | 数据 | 校验代码 |
| 01~08 | 08 | Z1~Z16 | CRC-16 |

Z1~Z4 ==DI1~DI4 1=外部连接 0=外部断开

* 1. **设置板卡地址Set Address 指令 FF**

该指令用于修改驱动卡的通讯地址。按以下步骤操作：

1. 整个网络中只保留主机和一块驱动卡。
2. 发送设置设置指令，人工确认指示灯 闪烁次数。
3. 该网络恢复原样。

注意：该指令为的通讯地址采用广播地址（255）。

板卡地址 可以根据指示灯闪烁次数获得

|  |  |  |
| --- | --- | --- |
| Y1 | 驱动卡新地址 | 1-8 |
| Z1 | 驱动卡新地址 | 1-8 |

1. **样例**

***获取ID***

**01 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 71 88**

**00 01 00 64 00 3B 04 47 36 32 33 38 36 39 00 00 00 00 6F 8E**

***启动1号板卡 #0号马达 3线制 无光幕***

**01 05 00 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 70 48**

**00 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 62 B5**

***获取马达旋转结果***

**01 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 D0 E8**

**00 03 02 00 00 05 BA 00 77 0A 3C 00 00 00 00 00 00 00 98 46**

Z1=02 运转完毕

Z2= 00 当前操作 id 0

Z3=00 无故障

Z4,Z5=05BA 峰值电流 1466mA

Z6,Z7= 0077 平均电流 119mA

Z8,Z9=0A3C 运转耗时 2620mS

***启动2#板卡 #0号马达 3线制 无光幕***

**02 05 00 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 80 0C**

**00 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 62 B5**

***修改从机地址***

**FF 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 20 50**

**FF 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 D0 A0**

**FF 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 81 30**

**FF 04 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 33 01**

**FF 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 62 91**

**FF 06 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 92 61**

**FF 07 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 C3 F1**

**FF 08 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 F6 02**

***启动 #1路 DO***

**01 08 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 66 DA**

00 08 00 F1 **00 00 00 00 00 00 00 00 00 00 00 00 00 00 C7 62**

***启动 #2路 DO***

**01 08 01 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 A7 4A**

00 08 00 F1 **00 00 00 00 00 00 00 00 00 00 00 00 00 00 C7 62**

***关闭 #1路 DO***

**01 08 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 A7 DA**

00 08 00 F0 **00 00 00 00 00 00 00 00 00 00 00 00 00 00 06 62**

***读取DI***

**01 09 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 F6 4A**

00 09 00 00 **00 00 00 00 00 00 00 00 00 00 00 00 00 00 A7 B6**

1. **CRC程序参考代码**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

功能： CRC校验

pcMess: 待求数据的首指针

wLen : 待求数据的长度

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

unsigned int crcVal1(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;

}