

版权申明

北京和利时电机技术有限公司保留所有权力。

说明书的内容参照了相关法律基准和行业基准。使用产品时，如对本说明书提供的内容有疑问，请向购买产品的销售人员咨询，或致电客户服务热线，或致信本公司邮箱。

由于产品的不断更新升级，和利时电机保留在不事先通知的情况下，修改本手册中的产品和产品规格参数等文件的权利，提示客户请使用最新版本的说明书。

和利时电机具有本产品及其软件的专利权、版权和其它知识产权。未经许可，不得直接或者间接地复制、制造、加工、使用本产品及其相关部分。

和利时电机具有本说明书的著作权，未经许可，不得修改、复制说明书的全部或部分内容。

一、通讯端口设置:

UDP 通讯需要打开端口, 本地端口为 8410, 目标端口为 7408, 从本地端口 8410 发给目标端口 7408。(8410 和 7408 是 MBOX 出厂默认端口, 可以修改, 详见 MBOX 说明书)

二、动作指令 MBOXUDP 数据区举例

01	00	00	00	00	55	aa	00	00	13	01	00	01	ff	ff	ff	ff	00	00	00	
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	12	34	56	78	ab	cd					

三、动作指令 UDP 数据区 分析

1、验证码

固定值, 0x55aa, 十六进制, 2 个 Byte。

2、通过码

固定值, 0x0000, 十六进制, 2 个 Byte。

3、功能码

分为两类 0x1301, 0x1401, 十六进制, 2 个 Byte。

其中绝对时间播放功能码是 0x1301, 相对播放时间功能码是 0x1401。

绝对时间 0x1301 是匹配影片的, 绝对时间播放时, 指令的执行时间是当前指令里的绝对时间减去上一条指令里的绝对时间, 比如你上一条指令的绝对时间是 1500ms, 现在发送的这一条是 1510ms, 那么现在这一条指令的执行完成的时间是 100ms (1510ms-1500ms)。

相对时间播放功能码 0x1401 是当前指令里的 Delta Time 就是当前指令的执行时间。

MBOX 是根据这个执行时间来计算电机的转速的。

如果用绝对时间播放功能码 1301, 时间码为 0, 那么每一帧数据的播放时间将按内部时间播放(默认 100ms); 如果用相对时间功能码, 时间码为 0, 那么每一帧数据的播放时间也将按内部时间播放(默认 100ms)。

但是建议不要把时间码设置成 0, 请根据实际的执行时间来设置。

4、通道号

十六进制, 2 个 Byte。

确定当前指令是 3 轴、6 轴、10 轴控制指令, 3 轴指令为 0x0000, 六轴指令为 0x0001, 10 轴指令为 0x0002。

5、谁接收, 谁回复

十六进制, 每个 2 个 Byte, 共 4 个 Byte。

每台 MBOX 都有 IP 地址, 出厂默认为 192.168.15.201(可以修改, 修改方法见 MBOX 说明书), 且支持对每条控制指令进行应答。

如发送的指令希望 IP 地址为 192.168.15.201 接收并回复, 则谁接收为 0x0fc9 (0x0f=15, 0xc9=201), 谁回复为 0x0fc9 (0x0f=15, 0xc9=201)。如同时控制多个平台, 希望每个平台都可以接收, 每个平台都回复, 则谁接收为 0xffff, 谁回复为 0xffff。将谁回复设置为 0x0000, 则平台不对接收到的指令进行回复; 将谁回复设置为某个固定值, 则对应的就回复 (如谁回复为 0x0510, 则对应 IP 地址后连个为 192.168.5.16 进行指令回复)。对于行和列接收回复详

批注 [z1]: 验证码, 2 个字节

批注 [z2]: 通过码, 2 个字节

批注 [z3]: 功能码, 2 个字节。分为播放绝对时间、播放相对时间

批注 [z4]: 确定是六个轴 2 个字节

批注 [z5]: 确定 UDP 数据由那个平台接收, 有那个平台回复, (全为 FF, 表示全接收, 全回复)。每个 2 个字节, 共计 4 个字节

批注 [z6]: 发送的 UDP 指令的序号, 2 个字节

批注 [z7]: 时间码, 4 个字节, 单位是毫秒, 格式为十六进制

批注 [z8]: 六个轴的数据, 每个轴 4 个字节, 单位是脉冲数, 格式十六进制

批注 [z9]: 2 个字节, 12 路开关量

批注 [z10]: 4 个字节, 两路模拟量

见 MBOX 说明书。

6、UDP 指令的序号

十六进制，共 4 个 Byte。
仅校验用，可以设置为 0x00000000。

7、时间码

十六进制，共 4 个 Byte。
单位是毫秒。

如果第 3 功能码选择为 0x1301，则时间码为绝对时间，例如希望本条指令 100 毫秒内执行完，则需要计算本条指令减去上一条指令的时间差值。

如果第 3 功能码选择为 0x1401，则时间码为相对对时间，例如希望本条指令 100 毫秒内执行完，则本条指令里的时间码需设置为 0x00000064。

绝对时间 0x1301 是匹配影片的，绝对时间播放时，指令的执行时间是当前指令里的绝对时间减去上一条指令里的绝对时间，比如你上一条指令的绝对时间是 15000ms，现在发送的这一条是 15100ms，那么现在这一条指令的执行完成的时间是 100ms (15100ms-15000ms)。

相对时间播放功能码 0x1401 是当前指令里的 Delta Time 就是当前指令的执行时间。

MBOX 是根据这个执行时间来计算电机的转速的。

如果用绝对时间播放功能码 1301，时间码为 0，那么每一帧数据的播放时间将按内部时间播放(默认 100ms)；如果用相对时间功能码，时间码为 0，那么每一帧数据的播放时间也将按内部时间播放(默认 100ms)。

但是建议不要把时间码设置成 0，请根据实际的执行时间来设置。

8、位置信息

十六进制，每个位置信息 4 个 Byte。
单位是脉冲个数。

有多少位置信息是由第 4 通道号来决定，如果通道号是 3 轴指令 0x0000，则需要些 3 个位置信息分别为 X、Y、Z (对应 1-3 轴)，共 12 个 Byte；如果通道号是 6 轴指令 0x0001，则需要些 6 个位置信息分别为 X、Y、Z、U、V、W (对应 1-6 轴)，共 24 个 Byte；如果通道号是 10 轴指令 0x0002，则需要些 10 个位置信息分别为 X、Y、Z、U、V、W、A、B、C、D (对应 1-10 轴)，共 40 个 Byte。

脉冲的计算方式：出厂默认电机转一圈的脉冲数是 10000 个

(1) 电动缸举例：

假如，电动缸行程是 5 毫米，一圈脉冲数是 10000。电机转一圈，电动缸升高一个行程。电动缸都从 0 到 25 毫米，计算所需脉冲数： $25/5*10000=50000$ ，转换成十六进制是 0x0000c350，(注：可以根据需要设置每个缸伸长的长度。)

(2) 旋转轴举例：

旋转轴一般都有减速，假如减速比是 1: 60，就是电机转 60 圈，转台转一圈。如果转台需要旋转 30 度，计算所需脉冲数 $30/360*(60*10000)=50000$ ，转换为十六进制是 0x0000c350。

9、特效输出

十六进制，共 2 个 Byte。

这是一个 16bit 的数据，高 4bit 保留，低 12bit 表示 12 路数字输出，每一个 bit 对应一路输出，1 位特效打开，0 为特效关闭。

B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
保留				12 路数字输出设定值											

举例如下：

如要控制 1、3、4 路输出，其他各路关闭，这对应如下

B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1

十六进制为 0x000D。

10、模拟量输出

输出模拟量，两路输出，每路 2 个字节，共 4 个字节，详见 MBOX 说明书

动作指令举例如下：

1、三轴数据且为绝对时间模式（功能码是 1301）

```

55 aa 00 00 13 01 00 00 ff ff ff ff 00 00 00
01 00 00 00 00 00 00 c3 50 00 00 c3 50 00 00 c3 50 12 34
56 78 ab cd

```

2、三轴数据且为相对时间模式（功能码是 1401）

```

55 aa 00 00 14 01 00 00 ff ff ff ff 00 00 00
01 00 00 00 64 00 00 c3 50 00 00 c3 50 00 00 c3 50 12 34
56 78 ab cd

```

3、六轴数据且为绝对时间模式（功能码是 1301）

```

55 aa 00 00 13 01 00 01 ff ff ff ff 00 00 00
01 00 00 00 00 00 00 c3 50 00 00 c3 50 00 00 c3 50 00 00
c3 50 00 00 c3 50 00 00 c3 50 12 34 56 78 ab cd

```

4、六轴数据且为相对时间模式（功能码是 1401）

```

55 aa 00 00 14 01 00 01 ff ff ff ff 00 00 00
01 00 00 00 64 00 00 c3 50 00 00 c3 50 00 00 c3 50 00 00
c3 50 00 00 c3 50 00 00 c3 50 12 34 56 78 ab cd

```

5、十轴数据且为绝对时间模式（功能码是 1301）

```

55 aa 00 00 13 01 00 02 ff ff ff ff 00 00 00
01 00 00 00 00 00 00 c3 50 00 00 c3 50 00 00 c3 50 00 00 c3
50 00 00 c3 50 00 00 c3 50 00 00 c3 50 00 00 c3 50 00 00 c3
50 00 00 c3 50 12 34 56 78 ab cd

```

6、十轴数据且为相对时间模式（功能码是 1401）

```

55 aa 00 00 14 01 00 02 ff ff ff ff 00 00 00
01 00 00 00 64 00 00 c3 50 00 00 c3 50 00 00 c3 50 00 00 c3
50 00 00 c3 50 00 00 c3 50 00 00 c3 50 00 00 c3 50 00 00 c3
50 00 00 c3 50 12 34 56 78 ab cd

```

四、读写参数指令

```

55 aa 00 00 12 01 00 00 ff ff ff ff 00 90 00
01 00 01

```

1、确认码

同 动作指令

批注 [z11]: 确认码，同动作指令

批注 [z12]: 通过码，同动作指令

批注 [z13]: 功能码，2 个字节

批注 [z14]: 对象通道为，两个字节

批注 [z15]: 谁接收，谁回复，同动作指令

批注 [z16]: 参数寄存器的起始地址，2 个字节

批注 [z17]: 寄存器个数，2 个字节

批注 [z18]: 读寄存器参数，无此数据；写寄存器参数，为写寄存的值，每个寄存器的值都是 2 个字节，总数跟寄存器个数有关

- 2、通过码
同 动作指令
- 3、功能码
十六进制，2 个 Byte。
分为读寄存器功能码 0x1101，写寄存器 0x1201
- 4、通道
十六进制，2 个 Byte。
由第 3 功能码确定。
如功能码读寄存器 0x1101，则通道为 0x0000 读 DN 参数，通道为 0x0001 读 FN 参数。
如功能码写寄存器 0x1201，则通道为 0x0000 写 FN 参数但不保存，通道为 0x0001 写 FN 参数并保存，通道为 0x0002 写 CX 控制参数。
- 5、谁接收，谁回复
同 动作指令
- 6、参数起始地址
十六进制，2 个 Byte。
支持单个读取和连续多个读取，因此为要读取参数的起始地址。
- 7、读取寄存器的个数
十六进制，2 个 Byte。
包括起始地址在内连续读多少个字节。
- 8、参数值
如为读取参数命令，则指令无参数值部分。如为写参数，则为寄存器的值。

读寄存器指令举例

读 DN 参数，起始地址为 0x0010（对应 DN10），连续读取 0x0010 个参数（对应连续读取 16 个参数），即连续读取（DN10、DN11、……、DN1F 共 16 个参数）

55 aa 00 00 11 01 00 00 ff ff ff ff 00 10 00

10

- 批注 [z19]: 参数寄存器的起始地址，2 个字节，起始地址为 16
- 批注 [z20]: 连续读取 16 个 DN 参数

读 FN 参数起始地址为 0x0001（对应 FN001），连续读取 0x0002 个参数（对应连续读取 2 个参数），即连续读取（FN001、FN002 共 2 个参数）

55 aa 00 00 11 01 00 01 ff ff ff ff 00 01 00

02

- 批注 [z21]: 参数寄存器的起始地址，2 个字节，起始地址为 1
- 批注 [z22]: 连续读取 2 个 FN 参数

DN 参数和 FN 参数的回复指令相同，仅通道不同。此处以 FN 参数为例。

55 aa 00 00 11 02 00 01 00 00 00 00 00 01 00

02 00 00 00 00

- 批注 [z23]: 为 0x0000 则回复 DN 的值，为 0x0001 则回复 FN 的值
- 批注 [z24]: 参数寄存器的起始地址，2 个字节，起始地址为 1
- 批注 [z25]: 连续回复 2 个参数
- 批注 [z26]: 回复的参数值，每个寄存器对应 2 个 Byte。依次为 FN01, FN02

写寄存指令举例

写 FN 参数起始地址为 0x0020 (对应 FN020), 连续写 0x0004 个参数 (对应连续写 4 个参数), 即连续写 (FN020、FN021、FN022、FN023 共 4 个参数)

04 | 00 c0 00 a8 00 0f 00 c9 | 55 aa 00 00 12 01 00 01 ff ff ff ff 00 20 00

批注 [z27]: 为 0x0000 则写 FN 不保存, 为 0x0001 写 FN 并保存, 为 0x0002 写 CX 参数

批注 [z28]: 起始地址为 FN020

批注 [z29]: 连续写 4 个参数

批注 [z30]: 每个参数为 2 个 Byte, 共 8 个 Byte

写参数常用指令举例

1、平台急停命令(写 FN 寄存器, 修改不保存)

01 | 00 01 | 55 aa 00 00 12 01 00 00 ff ff ff ff 00 90 00

批注 [z31]: 对象通道为 0, 表示修改参数寄存器, 但不保存

批注 [z32]: 参数寄存器的起始地址为 0x0090

批注 [z33]: 寄存器的长度为 1, 表示只访问一个寄存器

批注 [z34]: 参数寄存器的值, 即向地址 0x0090 写 0x0001, 使平台急停

2、平台取消急停命令(写 FN 寄存器, 修改不保存)

01 | 00 00 | 55 aa 00 00 12 01 00 00 ff ff ff ff 00 90 00

批注 [z35]: 对象通道为 0, 表示修改参数寄存器, 但不保存

批注 [z36]: 参数寄存器的起始地址为 0x0090

批注 [z37]: 寄存器的长度为 1, 表示只访问一个寄存器

3、平台复位命令(写 CX 寄存器)

01 | 00 00 | 55 aa 00 00 12 01 00 02 ff ff ff ff 00 00 00

批注 [z38]: 对象通道为 2, 表示命令寄存器

批注 [z39]: 命令寄存器起始地址为 0, 表示播放控制寄存器

批注 [z40]: 寄存器的长度为 1, 表示只对 1 个寄存器进行操作

批注 [z41]: 命令寄存器数据为: 0x0000, 实现复位操作

4、通过写寄存器控制特效(写 CX 寄存器)

01 ff ff 55 aa 00 00 12 01 00 02 ff ff ff ff 00 01 00

批注 [z42]: 对象通道为 2，表示命令寄存器

批注 [z43]: 命令寄存器起始地址为 1，表示特效控制寄存器

批注 [z44]: 寄存器的长度为 1，表示只对 1 个寄存器进行操作

批注 [z45]: 命令寄存器数据为：0xffff，特效全开