

## LK-TECH伺服单元不同系列对比表

LK-TECH SERVO UNIT DIFFERENT SERIES COMPARISON TABLE ENGLISH

电机系列	MS系列	MF系列	MG系列	MH系列
电机图片				
电机优势	低速稳定,过滑环线	高速,高精度	内置减速机, 小背隙	大中孔
输入电压	7.4-24V	12-36V	24-48V	12-24V
电流	0-4A	0-9A	0-14A	0-4A
速度范围	0-1000rpm	0-3000rpm	0-2000rpm	0-3000rpm
驱动类型	SVPWM控制	FOC控制	FOC控制	FOC控制
编码器精度	12bit/18bit	14bit/18bit	18bit	14bit
通讯方式	RS-485/CAN BUS	RS-485/CAN BUS	RS-485/CAN BUS	RS-485/CAN BUS
控制模式	速度模式/位置模式	力矩模式/速度模式/ 位置模式	力矩模式/速度模式/ 位置模式	力矩模式/速度模式/ 位置模式
保护类型	温度保护/低压保护	温度保护/低压保护	温度保护/低压保护	温度保护/低压保护
应景场景	云台、吊舱	云台、转盘、电力 工业巡检机械臂、 激光雷达	足式机器人、外骨 骼机器人	云台、吊舱、转盘、 激光雷达

### 免责声明

感谢您购买瓴控科技 M 系列电机及驱动系统。在使用之前,请仔细阅读本声明,一旦使用,即被视为对本声明全部内容的认可和接受。请严格遵守手册、产品说明和相关的法律法规、政策、准则安装和使用该产品。在使用产品过程中,用户承诺对自己的行为及因此而产生的所有后果负责。因用户不当使用、安装、改装造成的任何损失,瓴控科技将不承担法律责任。

瓴控科技是上海瓴控科技有限公司及其关联公司的商标。本文出现的产品名称、品牌等,均为其所属公司的商标或注册商标。

本产品及手册为瓴控科技版权所有。未经许可,不得以任何形式复制翻印。关于免责声明的最终解释权,归瓴控科技所有。

## 目录

1 RS485参数总线.....	3
2 命令帧说明及单电机命令列表.....	3
3 单电机命令说明.....	4
(1) 读取PID参数命令.....	4
(2) 写入PID参数到RAM命令.....	4
(3) 写入PID参数到ROM命令.....	5
(4) 读取加速度命令.....	5
(5) 写入加速度到RAM命令.....	6
(6) 读取编码器命令.....	6
(7) 写入编码器值作为电机零点命令.....	7
(8) 写入当前位置到ROM作为电机零点命令.....	7
(9) 读取多圈角度命令.....	8
(10) 读取单圈角度命令.....	8
(11) 读取电机状态1和错误标志命令.....	9
(12) 清除电机错误标志命令.....	10
(13) 读取电机状态2命令.....	10
(14) 读取电机状态3命令.....	11
(15) 电机关闭命令.....	12
(16) 电机停止命令.....	12
(17) 电机运行命令.....	12
(18) 开环控制命令.....	13
(19) 转矩闭环控制命令.....	13
(20) 速度闭环控制命令.....	14
(21) 多圈位置闭环控制命令1.....	15
(22) 多圈位置闭环控制命令2.....	16
(23) 单圈位置闭环控制命令1.....	17
(24) 单圈位置闭环控制命令2.....	18
(25) 增量位置闭环控制命令1.....	20
(26) 增量位置闭环控制命令2.....	21
(27) 读取驱动和电机型号命令.....	22

## RS485 总线通讯协议

### 1. RS485 总线参数

总线接口：RS485

波特率：9600, 19200, 38400, 57600, 115200(默认), 230400, 460800, 1Mbps, 2Mbps

数据位：8

奇偶校验：无

停止位：1

### 2. 命令帧说明及单电机命令列表

同一总线上共可以挂载多达 32（视总线负载情况而定）个驱动，为了防止总线冲突，每个驱动需要设置不同的 ID，具体可参考上一节中的基础设定。

主控向驱动发送控制命令，对应 ID 的驱动接收到命令后解析数据，根据命令类型选择控制方式（角度闭环、速度闭环、扭矩闭环），并在一段时间后（0.5ms 内）向主控发送回复。

每条控制命令都是由 2 部分组成：帧头 + 数据，具体说明如下：

类型	数据描述	数据长度	说明
帧命令	头字节	1	帧头识别, 0x3E
	命令字节	1	CMD
	ID 字节	1	1~32, 对应电机的 ID
	数据长度字节	1	描述数据长度, 视不同命令而定
	帧头校验字节	1	Header checksum
帧数据	数据	0~60	命令附带的数据, 视不同命令而定
	数据校验字节	0 或 1	Data checkSum

目前 M 系列电机驱动支持的 RS485 控制命令如下表：

名称	命令数据
读取 PID 参数命令	0x30
写入 PID 参数到 RAM 命令	0x31
写入 PID 参数到 ROM 命令	0x32
读取加速度命令	0x33
写入加速度到 RAM 命令	0x34
读取编码器命令	0x90
写入编码器值到 ROM 作为电机零点命令	0x91
写入当前位置到 ROM 作为电机零点命令	0x19
读取多圈角度命令	0x92
读取单圈角度命令	0x94
清除电机角度命令（设置电机初始位置）	0x95
读取电机状态 1 和错误标志命令	0x9A
清除电机错误标志命令	0x9B
读取电机状态 2 命令	0x9C

读取电机状态 3 命令	0x9D
电机关闭命令	0x80
电机停止命令	0x81
电机运行命令	0x88
开环控制命令	0xA0
转矩闭环控制命令	0xA1
速度闭环控制命令	0xA2
多圈位置闭环控制命令 1	0xA3
多圈位置闭环控制命令 2	0xA4
单圈位置闭环控制命令 1	0xA5
单圈位置闭环控制命令 2	0xA6
增量位置闭环控制命令 1	0xA7
增量位置闭环控制命令 2	0xA8
读取驱动和电机型号命令	0x12

### 3. 单电机命令说明

#### (1) 读取 PID 参数命令 (5byte)

主机发送该命令读取当前电机的 PID 的参数

数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0x30
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x00
DATA[4]	帧头校验字节	DATA[0]~DATA[3]字节校验和

#### 驱动回复 (12byte)

驱动回复数据中包含了各个控制环路的 PI 参数

数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0x30
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x06
DATA[4]	帧头校验字节	DATA[0]~DATA[3]字节校验和
DATA[5]	位置环 P 参数	DATA[5] = anglePidKp
DATA[6]	位置环 I 参数	DATA[6] = anglePidKi
DATA[7]	速度环 P 参数	DATA[7] = speedPidKp
DATA[8]	速度环 I 参数	DATA[8] = speedPidKi
DATA[9]	转矩环 P 参数	DATA[9] = iqPidKp
DATA[10]	转矩环 I 参数	DATA[10] = iqPidKi
DATA[11]	帧头校验字节	DATA[5]~DATA[10]字节校验和

#### (2) 写入 PID 参数到 RAM 命令 (12byte)

主机发送该命令写入 PID 参数到 RAM 中，断电后写入参数失效

数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0x31
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x06
DATA[4]	帧头校验字节	DATA[0]~DATA[3]字节校验和
DATA[5]	位置环 P 参数	DATA[5] = anglePidKp
DATA[6]	位置环 I 参数	DATA[6] = anglePidKi
DATA[7]	速度环 P 参数	DATA[7] = speedPidKp
DATA[8]	速度环 I 参数	DATA[8] = speedPidKi
DATA[9]	转矩环 P 参数	DATA[9] = iqPidKp
DATA[10]	转矩环 I 参数	DATA[10] = iqPidKi
DATA[11]	帧头校验字节	DATA[5]~DATA[10]字节校验和

#### 驱动回复 (12byte)

驱动回复数据和接收到命令参数一致

#### (3) 写入 PID 参数到 ROM 命令 (12byte)

主机发送该命令写入 PID 参数到 RAM 中，断电仍然有效

数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0x32
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x06
DATA[4]	帧头校验字节	DATA[0]~DATA[3]字节校验和
DATA[5]	位置环 P 参数	DATA[5] = anglePidKp
DATA[6]	位置环 I 参数	DATA[6] = anglePidKi
DATA[7]	速度环 P 参数	DATA[7] = speedPidKp
DATA[8]	速度环 I 参数	DATA[8] = speedPidKi
DATA[9]	转矩环 P 参数	DATA[9] = iqPidKp
DATA[10]	转矩环 I 参数	DATA[10] = iqPidKi
DATA[11]	帧头校验字节	DATA[5]~DATA[10]字节校验和

#### 驱动回复 (12byte)

驱动回复数据和接收到命令参数一致

#### (4) 读取加速度命令 (5byte)

主机发送该命令读取当前电机的加速度参数

数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0x33
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x00
DATA[4]	帧头校验字节	DATA[0]~DATA[3]字节校验和

#### 驱动回复 (10byte)

驱动回复数据中包含了加速度参数，加速度数据 Accel 为 int32\_t 类型，单位 1dps/s

数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0x33
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x04
DATA[4]	帧头校验字节	DATA[0]~DATA[3]字节校验和
DATA[5]	加速度低字节 1	DATA[5] = *((uint8_t *)&Accel)
DATA[6]	加速度字节 2	DATA[6] = *((uint8_t *)&Accel)+1
DATA[7]	加速度字节 3	DATA[7] = *((uint8_t *)&Accel)+2
DATA[8]	加速度字节 4	DATA[8] = *((uint8_t *)&Accel)+3
DATA[9]	数据校验字节	DATA[5]~DATA[8]字节校验和

#### (5) 写入加速度到 RAM 命令 (10byte)

主机发送该命令写入加速度参数到 RAM 中，断电后写入参数失效。加速度数据 Accel 为 uint32\_t 类型，单位 1dps/s

数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0x34
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x04
DATA[4]	帧头校验字节	DATA[0]~DATA[3]字节校验和
DATA[5]	加速度低字节 1	DATA[5] = *((uint8_t *)&Accel)
DATA[6]	加速度字节 2	DATA[6] = *((uint8_t *)&Accel)+1
DATA[7]	加速度字节 3	DATA[7] = *((uint8_t *)&Accel)+2
DATA[8]	加速度字节 4	DATA[8] = *((uint8_t *)&Accel)+3
DATA[10]	数据校验字节	DATA[5]~DATA[8]字节校验和

#### 驱动回复 (10byte)

驱动回复数据和接收到命令参数一致

#### (6) 读取编码器命令 (5byte)

主机发送该命令以读取当前编码器的当前位置

数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0x90
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x00
DATA[4]	帧头校验字节	DATA[0]~DATA[3]字节校验和

#### 驱动回复 (12byte)

电机在收到命令后回复主机，回复数据中包含了以下参数。

1. 编码器位置 encoder (uint16\_t 类型, 14bit 编码器的数值范围 0~16383)，为编码器原始位置减去编码器零偏后的值。
2. 编码器原始位置 encoderRaw (uint16\_t 类型, 14bit 编码器的数值范围 0~16383)。
3. 编码器零偏 encoderOffset (uint16\_t 类型, 14bit 编码器的数值范围 0~16383)，该点作为电机角度的 0 点。



数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0x90
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x06
DATA[4]	帧头校验字节	DATA[0]~DATA[3]字节校验和
DATA[5]	编码器数据低字节	=*(uint8_t*)&encoder
DATA[6]	编码器数据高字节	=*((uint8_t*)&encoder)+1
DATA[7]	编码器原始位置低字节	=*(uint8_t*)&encoderRaw
DATA[8]	编码器原始位置高字节	=*((uint8_t*)&encoderRaw)+1
DATA[9]	编码器零偏低字节	=*(uint8_t*)&encoderOffset
DATA[10]	编码器零偏高字节	=*((uint8_t*)&encoderOffset)+1
DATA[11]	数据校验字节	DATA[5]~DATA[10]字节校验和

#### (7) 写入编码器值作为电机零点命令 (8byte)

主机发送该命令以设置编码器的零偏，其中，需要写入的编码器值 encoderOffset 为 uint16\_t 类型，14bit 编码器的数值范围 0~16383

数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0x91
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x02
DATA[4]	帧头校验字节	DATA[0]~DATA[3]字节校验和
DATA[5]	编码器零偏低字节	=*(uint8_t*)&encoderOffset
DATA[6]	编码器零偏高字节	=*((uint8_t*)&encoderOffset)+1
DATA[7]	帧头校验字节	DATA[5]~DATA[6]字节校验和

#### 驱动回复 (8byte)

驱动回复的数据和主机发送的命令相同。

#### (8) 写入当前位置到 ROM 作为电机零点命令 (5byte)

将电机当前编码器位置作为初始位置写入到 ROM

注意：

1. 该命令需要重新上电后才能生效
2. 该命令会将零点写入驱动的 ROM，多次写入将会影响芯片寿命，不建议频繁使用

数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0x19
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x00
DATA[4]	帧头校验字节	DATA[0]~DATA[3]字节校验和

例如，主机向 1#驱动发送设置零点命令如下(HEX)

3E 19 01 00 58

#### 驱动回复 (26byte)

该命令暂未开放。

### (9) 读取多圈角度命令 (5byte)

主机发送该命令以读取当前电机的多圈绝对角度值

数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0x92
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x00
DATA[4]	帧头校验字节	DATA[0]~DATA[3]字节校验和

#### 驱动回复 (14byte)

电机在收到命令后回复主机，该帧数据中包含了以下参数：

1. 电机角度 **motorAngle**，为 **int64\_t** 类型数据，正值表示顺时针累计角度，负值表示逆时针累计角度，单位  $0.01^{\circ}/\text{LSB}$ 。

数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0x92
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x08
DATA[4]	帧头校验字节	DATA[0]到 DATA[3]的校验和
DATA[5]	角度低字节 1	DATA[5] = *((uint8_t *)&motorAngle)
DATA[6]	角度字节 2	DATA[6] = *((uint8_t *)&motorAngle)+1
DATA[7]	角度字节 3	DATA[7] = *((uint8_t *)&motorAngle)+2
DATA[8]	角度字节 4	DATA[8] = *((uint8_t *)&motorAngle)+3
DATA[9]	角度字节 5	DATA[9] = *((uint8_t *)&motorAngle)+4
DATA[10]	角度字节 6	DATA[10] = *((uint8_t *)&motorAngle)+5
DATA[11]	角度字节 7	DATA[11] = *((uint8_t *)&motorAngle)+6
DATA[12]	角度高字节 8	DATA[12] = *((uint8_t *)&motorAngle)+6
DATA[13]	数据校验字节	DATA[5]到 DATA[12]的校验和

### (10) 读取单圈角度命令 (5byte)

主机发送该命令以读取当前电机的单圈绝对角度值

数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0x94
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x00
DATA[4]	帧头校验字节	DATA[0]到 DATA[3]的校验和

#### 驱动回复 (10 byte)

电机在收到命令后回复主机，该帧数据中包含了以下参数：

1. 电机单圈角度 **circleAngle**，为 **uint32\_t** 类型数据，以编码器零点为起始点，顺时针增加，再次到达零点时数值回 0，单位  $0.01^{\circ}/\text{LSB}$ ，数值范围  $0 \sim 36000 \times \text{减速比} - 1$ 。

数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0x94
DATA[2]	ID 字节	0x01~0x20



DATA[3]	数据长度字节	0x04
DATA[4]	帧头校验字节	DATA[0]到 DATA[3]的校验和
DATA[5]	单圈角度低字节 1	DATA[5] = *(uint8_t *)&circleAngle
DATA[6]	单圈角度字节 2	DATA[6] = *((uint8_t *)& circleAngle)+1
DATA[7]	单圈角度字节 3	DATA[7] = *((uint8_t *)& circleAngle)+2
DATA[8]	单圈角度高字节 4	DATA[8] = *((uint8_t *)& circleAngle)+3
DATA[9]	数据校验字节	DATA[5]到 DATA[8]的校验和

#### (11) 读取电机状态 1 和错误标志命令 (5byte)

该命令读取当前电机的温度、电压和错误状态标志

数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0x9A
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x00
DATA[4]	帧头校验字节	DATA[0]到 DATA[3]的校验和

#### 驱动回复 (13byte)

电机在收到命令后回复主机，该帧数据包含了以下参数：

1. 电机温度 temperature (int8\_t 类型，单位 1°C/LSB)。
2. 电压 voltage (uint16\_t 类型，单位 0.1V/LSB)。
3. 错误标志 errorState (为 uint8\_t 类型，各个位代表不同的电机状态)

数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0x9A
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x07
DATA[4]	帧头校验字节	DATA[0]到 DATA[3]的校验和
DATA[5]	电机温度	DATA[5] = *(uint8_t *)&temperature
DATA[6]	NULL	0x00
DATA[7]	电压低字节	DATA[7] = *(uint8_t *)&voltage
DATA[8]	电压高字节	DATA[8] = *((uint8_t *)& voltage)+1
DATA[9]	NULL	0x00
DATA[10]	NULL	0x00
DATA[11]	错误状态字节	DATA[11]=errorState
DATA[12]	数据校验字节	DATA[5]到 DATA[11]的校验和

备注：

1. errorState 各个位具体状态表如下

errorState 位	状态说明	0	1
0	电压状态	电压正常	低压保护
1	无效		
2	无效		
3	温度状态	温度正常	过温保护
4	无效		
5	无效		

6	无效		
7	无效		

#### (12) 清除电机错误标志命令 (5byte)

该命令清除当前电机的错误状态，电机收到后返回

数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0x9B
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x00
DATA[4]	帧头校验字节	DATA[0]到 DATA[3]的校验和

#### 驱动回复 (13byte)

电机在收到命令后回复主机，该帧数据包含了以下参数：

1. 电机温度 `temperature` (`int8_t` 类型，单位 1°C/LSB)。
2. 电压 `voltage` (`uint16_t` 类型，单位 0.1V/LSB)。
3. 错误标志 `errorState` (为 `uint8_t` 类型，各个位代表不同的电机状态)。

数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0x9B
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x07
DATA[4]	帧头校验字节	DATA[0]到 DATA[3]的校验和
DATA[5]	电机温度	DATA[5] = *( <code>uint8_t</code> *)&temperature
DATA[6]	NULL	0x00
DATA[7]	电压低字节	DATA[7] = *( <code>uint8_t</code> *)&voltage
DATA[8]	电压高字节	DATA[8] = *(( <code>uint8_t</code> *)&voltage)+1
DATA[9]	NULL	0x00
DATA[10]	NULL	0x00
DATA[11]	错误状态字节	DATA[11]=errorState
DATA[12]	数据校验字节	DATA[5]到 DATA[11]的校验和

备注：

1. 电机状态没有恢复正常时，错误标志无法清除。
2. `errorState` 各个位具体状态参考读取电机状态 1 和错误标志命令。

#### (13) 读取电机状态 2 命令 (5byte)

该命令读取当前电机的温度、电机转矩电流 (MF、MG) / 电机输出功率 (MS)、转速、编码器位置。

数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0x9C
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x00
DATA[4]	帧头校验字节	DATA[0]到 DATA[3]的校验和

#### 驱动回复 (13byte)

电机在收到命令后回复主机，该帧数据中包含了以下参数。

1. 电机温度 `temperature` (`int8_t` 类型,  $1^{\circ}\text{C}/\text{LSB}$ )。
2. MF、MG 的转矩电流值 `iq` 或 MS 的输出功率值 `power`。 `iq` 为 `int16_t` 类型, 范围-2048~2048, 对应实际转矩电流范围-33A~33A; `power` 为 `int16_t` 类型, 范围-1000~1000。
3. 电机转速 `speed` (`int16_t` 类型,  $1\text{dps}/\text{LSB}$ )。
4. 编码器位置值 `encoder` (`uint16_t` 类型, 14bit 编码器的数值范围 0~16383)。

数据域	说明	数据	
DATA[0]	头字节	0x3E	
DATA[1]	命令字节	0x9C	
DATA[2]	ID 字节	0x01~0x20	
DATA[3]	数据长度字节	0x07	
DATA[4]	帧头校验字节	DATA[0]到 DATA[3]的校验和	
DATA[5]	电机温度	DATA[5] = *(uint8_t *)&temperature)	
DATA[6]	转矩电流低字节	DATA[6] = *(uint8_t *)&iq)	MF MG
	输出功率低字节	DATA[6] = *(uint8_t *)&power)	MS
DATA[7]	转矩电流高字节	DATA[7] = *((uint8_t *)&iq)+1)	MF MG
	输出功率高字节	DATA[7] = *((uint8_t *)&power)+1)	MS
DATA[8]	电机速度低字节	DATA[8] = *(uint8_t *)&speed)	
DATA[9]	电机速度高字节	DATA[9] = *((uint8_t *)&speed)+1)	
DATA[10]	编码器位置低字节	DATA[10] = *(uint8_t *)&encoder)	
DATA[11]	编码器位置高字节	DATA[11] = *((uint8_t *)&encoder)+1)	
DATA[12]	数据校验字节	DATA[5]到 DATA[11]的校验和	

#### (14) 读取电机状态 3 命令 (5byte)

该命令仅在 MF 和 MG 上实现

该命令读取当前电机的温度和相电流数据

数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0x9D
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x00
DATA[4]	帧头校验字节	DATA[0]到 DATA[3]的校验和

#### 驱动回复 (13byte)

电机在收到命令后回复主机, 该帧数据包含了以下数据:

1. 电机温度 `temperature` (`int8_t` 类型,  $1^{\circ}\text{C}/\text{LSB}$ )
2. A 相电流数据, 数据类型为 `int16_t` 类型, 对应实际相电流为 1A/64LSB。
3. B 相电流数据, 数据类型为 `int16_t` 类型, 对应实际相电流为 1A/64LSB。
4. C 相电流数据, 数据类型为 `int16_t` 类型, 对应实际相电流为 1A/64LSB。

数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0x9D
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x07
DATA[4]	帧头校验字节	DATA[0]到 DATA[3]的校验和
DATA[5]	电机温度	DATA[5] = *(uint8_t *)&temperature)

DATA[6]	A 相电流低字节	DATA[6] = *(uint8_t *)&iA)
DATA[7]	A 相电流高字节	DATA[7] = *((uint8_t *)&iA)+1)
DATA[8]	B 相电流低字节	DATA[8] = *(uint8_t *)&iB)
DATA[9]	B 相电流高字节	DATA[9] = *((uint8_t *)&iB)+1)
DATA[10]	C 相电流低字节	DATA[10] = *(uint8_t *)&iC)
DATA[11]	C 相电流高字节	DATA[11] = *((uint8_t *)&iC)+1)
DATA[12]	数据校验字节	DATA[5]到 DATA[11]的校验和

#### (15) 电机关闭命令 (5byte)

关闭电机，同时清除电机运行状态和之前接收的控制指令

数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0x80
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x00
DATA[4]	帧头校验字节	DATA[0]~DATA[3]字节校验和

例如，主机向 1#驱动发送电机关闭命令如下(HEX)

3E 80 01 00 BF

#### 驱动回复 (5byte)

和主机发送相同

#### (16) 电机停止命令 (5byte)

停止电机，但不清除电机运行状态和之前接收的控制指令

数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0x81
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x00
DATA[4]	帧头校验字节	DATA[0]~DATA[3]字节校验和

例如，主机向 1#驱动发送电机停止命令如下(HEX)

3E 81 01 00 C0

#### 驱动回复 (5byte)

和主机发送相同

#### (17) 电机运行命令 (5byte)

从电机停止命令中恢复电机运行（恢复停止前的控制方式）

数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0x88
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x00
DATA[4]	帧头校验字节	DATA[0]~DATA[3]字节校验和

例如，主机向 1#驱动发送电机停止命令如下(HEX)

3E 88 01 00 C7

#### 驱动回复 (5byte)

和主机发送相同

#### (18) 开环控制命令（8byte）

该命令仅在 MS 上实现

主机发送该命令以控制电机的开环输出功率，控制值 powerControl 为 int16\_t 类型，数值范围-1000~1000，（电机母线电流和扭矩因不同电机而异）。

数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0xA0
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x02
DATA[4]	帧头校验字节	DATA[0]~DATA[3]字节校验和
DATA[5]	输出功率控制值低字节	DATA[5] = *((uint8_t *)&powerControl)
DATA[6]	输出功率控制值高字节	DATA[6] = *((uint8_t *)&powerControl)+1
DATA[7]	数据校验字节	DATA[5]~DATA[6]字节校验和

备注：

1. 该命令中的控制值 powerControl 不受上位机(LK-Motor Tool)中的 Max Power 值限制。

#### 驱动回复（13byte）

电机在收到命令后回复主机，该帧数据中包含了以下参数。

1. 电机温度 temperature（int8\_t 类型，1℃/LSB）。
2. 电机输出功率值 power 为 int16\_t 类型，范围-1000~1000。
3. 电机转速 speed（int16\_t 类型，1dps/LSB）。
4. 编码器位置值 encoder（uint16\_t 类型，14bit 编码器的数值范围 0~16383）。

数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0xA0
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x07
DATA[4]	帧头校验字节	DATA[0]到 DATA[3]的校验和
DATA[5]	电机温度	DATA[5] = *((uint8_t *)&temperature)
DATA[6]	输出功率低字节	DATA[6] = *((uint8_t *)&power)
DATA[7]	输出功率高字节	DATA[7] = *((uint8_t *)&power)+1
DATA[8]	电机速度低字节	DATA[8] = *((uint8_t *)&speed)
DATA[9]	电机速度高字节	DATA[9] = *((uint8_t *)&speed)+1
DATA[10]	编码器位置低字节	DATA[10] = *((uint8_t *)&encoder)
DATA[11]	编码器位置高字节	DATA[11] = *((uint8_t *)&encoder)+1
DATA[12]	数据校验字节	DATA[5]到 DATA[11]的校验和

#### (19) 转矩闭环控制命令（8byte）

该命令仅在 MF 和 MG 上实现

主机发送该命令以控制电机的转矩电流输出，控制值 iqControl 为 int16\_t 类型，数值范围-2000~2000，对应实际转矩电流范围-32A~32A（母线电流和电机的实际扭矩因不同电机而异）。

数据域	说明	数据
DATA[0]	头字节	0x3E

DATA[1]	命令字节	0xA1
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x02
DATA[4]	帧头校验字节	DATA[0]~DATA[3]字节校验和
DATA[5]	转矩电流控制值低字节	DATA[5] = *(uint8_t *)&iqControl
DATA[6]	转矩电流控制值高字节	DATA[6] = *((uint8_t *)&iqControl)+1
DATA[7]	数据校验字节	DATA[5]~DATA[6]字节校验和

备注:

1. 该命令中的控制值 iqControl 不受上位机(LK-Motor Tool)中的 Max Torque Current 值限制。

#### 驱动回复 (13byte)

电机在收到命令后回复主机，该帧数据中包含了以下参数。

1. 电机温度 temperature (int8\_t 类型, 1°C/LSB)。
2. 电机的转矩电流值 iq。iq 为 int16\_t 类型, 范围-2048~2048, 对应实际转矩电流范围-33A~33A。
3. 电机转速 speed (int16\_t 类型, 1dps/LSB)。
4. 编码器位置值 encoder (uint16\_t 类型, 14bit 编码器的数值范围 0~16383)。

数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0xA1
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x07
DATA[4]	帧头校验字节	DATA[0]到 DATA[3]的校验和
DATA[5]	电机温度	DATA[5] = *(uint8_t *)&temperature
DATA[6]	转矩电流低字节	DATA[6] = *(uint8_t *)&iq
DATA[7]	转矩电流高字节	DATA[7] = *((uint8_t *)&iq)+1
DATA[8]	电机速度低字节	DATA[8] = *(uint8_t *)&speed
DATA[9]	电机速度高字节	DATA[9] = *((uint8_t *)&speed)+1
DATA[10]	编码器位置低字节	DATA[10] = *(uint8_t *)&encoder
DATA[11]	编码器位置高字节	DATA[11] = *((uint8_t *)&encoder)+1
DATA[12]	数据校验字节	DATA[5]到 DATA[11]的校验和

#### (20) 速度闭环控制命令 (10byte)

主机发送该命令以控制电机的速度，控制值 speedControl 为 int32\_t 类型, 对应实际转速为 0.01dps/LSB。

数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0xA2
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x04
DATA[4]	帧头校验字节	DATA[0]~DATA[3]字节校验和
DATA[5]	电机速度低字节	DATA[5] = *(uint8_t *)&speedControl
DATA[6]	电机速度	DATA[6] = *((uint8_t *)&speedControl)+1
DATA[7]	电机速度	DATA[7] = *((uint8_t *)&speedControl)+2
DATA[8]	电机速度高字节	DATA[8] = *((uint8_t *)&speedControl)+3
DATA[9]	数据校验字节	DATA[5]~DATA[8]字节校验和

备注:



1. 该命令下电机的 speedControl 由上位机(LK-Motor Tool)中的 Max Speed 值限制。
2. 该控制模式下，电机的最大加速度由上位机(LK-Motor Tool)中的 Max Acceleration 值限制。
3. 该控制模式下，MF 和 MG 电机的最大转矩电流由上位机(LK-Motor Tool)中的 Max Torque Current 值限制；MS 电机的最大功率由上位机中的 Max Power 值限制。

#### 驱动回复 (13byte)

电机在收到命令后回复主机，该帧数据中包含了以下参数。

1. 电机温度 temperature (int8\_t 类型，1°C/LSB)。
2. MF、MG 的转矩电流值 iq 或 MS 的输出功率值 power。iq 为 int16\_t 类型，范围-2048~2048，对应实际转矩电流范围-33A~33A；power 为 int16\_t 类型，范围-1000~1000。
3. 电机转速 speed (int16\_t 类型，1dps/LSB)。
4. 编码器位置值 encoder (uint16\_t 类型，14bit 编码器的数值范围 0~16383)。

数据域	说明	数据	
DATA[0]	头字节	0x3E	
DATA[1]	命令字节	0xA2	
DATA[2]	ID 字节	0x01~0x20	
DATA[3]	数据长度字节	0x07	
DATA[4]	帧头校验字节	DATA[0]到 DATA[3]的校验和	
DATA[5]	电机温度	DATA[5] = *(uint8_t *)&temperature	
DATA[6]	转矩电流低字节	DATA[6] = *(uint8_t *)&iq	MF MG
	输出功率低字节	DATA[6] = *(uint8_t *)&power	MS
DATA[7]	转矩电流高字节	DATA[7] = *((uint8_t *)&iq)+1	MF MG
	输出功率高字节	DATA[7] = *((uint8_t *)&power)+1	MS
DATA[8]	电机速度低字节	DATA[8] = *(uint8_t *)&speed	
DATA[9]	电机速度高字节	DATA[9] = *((uint8_t *)&speed)+1	
DATA[10]	编码器位置低字节	DATA[10] = *(uint8_t *)&encoder	
DATA[11]	编码器位置高字节	DATA[11] = *((uint8_t *)&encoder)+1	
DATA[12]	数据校验字节	DATA[5]到 DATA[11]的校验和	

#### (21) 多圈位置闭环控制命令 1 (14byte)

主机发送该命令以控制电机的位置（多圈角度），控制值 angleControl 为 int64\_t 类型，对应实际位置为 0.01degree/LSB，即 36000 代表 360°，电机转动方向由目标位置和当前位置的差值决定。

数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0xA3
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x08
DATA[4]	帧头校验字节	DATA[0]~DATA[3]字节校验和
DATA[5]	位置控制低字节 1	DATA[5] = *(uint8_t *)&angleControl
DATA[6]	位置控制字节 2	DATA[6] = *((uint8_t *)&angleControl)+1
DATA[7]	位置控制字节 3	DATA[7] = *((uint8_t *)&angleControl)+2
DATA[8]	位置控制字节 4	DATA[8] = *((uint8_t *)&angleControl)+3
DATA[9]	位置控制字节 5	DATA[9] = *((uint8_t *)&angleControl)+4
DATA[10]	位置控制字节 6	DATA[10] = *((uint8_t *)&angleControl)+5
DATA[11]	位置控制字节 7	DATA[11] = *((uint8_t *)&angleControl)+6

DATA[12]	位置控制高字节 8	DATA[12] = *((uint8_t *)&angleControl)+7)
DATA[13]	数据校验字节	DATA[5]~DATA[13]字节校验和

备注:

1. 该命令下的控制值 `angleControl` 受上位机(LK-Motor Tool)中的 `Max Angle` 值限制。
2. 该命令下电机的最大速度由上位机(LK-Motor Tool)中的 `Max Speed` 值限制。
3. 该控制模式下, 电机的最大加速度由上位机(LK-Motor Tool)中的 `Max Acceleration` 值限制。
4. 该控制模式下, MF 和 MG 电机的最大转矩电流由上位机(LK-Motor Tool)中的 `Max Torque Current` 值限制; MS 电机的最大功率由上位机中的 `Max Power` 值限制。

#### 驱动回复 (13byte)

电机在收到命令后回复主机, 该帧数据中包含了以下参数。

1. 电机温度 `temperature` (`int8_t` 类型,  $1^{\circ}\text{C}/\text{LSB}$ )。
2. MF、MG 的转矩电流值 `iq` 或 MS 的输出功率值 `power`。 `iq` 为 `int16_t` 类型, 范围-2048~2048, 对应实际转矩电流范围-33A~33A; `power` 为 `int16_t` 类型, 范围-1000~1000。
3. 电机转速 `speed` (`int16_t` 类型,  $1\text{dps}/\text{LSB}$ )。
4. 编码器位置值 `encoder` (`uint16_t` 类型, 14bit 编码器的数值范围 0~16383)。

数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0xA3
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x07
DATA[4]	帧头校验字节	DATA[0]到 DATA[3]的校验和
DATA[5]	电机温度	DATA[5] = *((uint8_t *)&temperature)
DATA[6]	转矩电流低字节	DATA[6] = *((uint8_t *)&iq)
	输出功率低字节	DATA[6] = *((uint8_t *)&power)
DATA[7]	转矩电流高字节	DATA[7] = *((uint8_t *)&iq)+1)
	输出功率高字节	DATA[7] = *((uint8_t *)&power)+1)
DATA[8]	电机速度低字节	DATA[8] = *((uint8_t *)&speed)
DATA[9]	电机速度高字节	DATA[9] = *((uint8_t *)&speed)+1)
DATA[10]	编码器位置低字节	DATA[10] = *((uint8_t *)&encoder)
DATA[11]	编码器位置高字节	DATA[11] = *((uint8_t *)&encoder)+1)
DATA[12]	数据校验字节	DATA[5]到 DATA[11]的校验和

#### (22) 多圈位置闭环控制命令 2 (18byte)

主机发送该命令以控制电机的位置 (多圈角度), 控制值 `angleControl` 为 `int64_t` 类型, 对应实际位置为  $0.01^{\circ}/\text{LSB}$ , 即 36000 代表  $360^{\circ}$ , 电机转动方向由目标位置和当前位置的差值决定。控制值 `maxSpeed` 限制了电机转动的最大速度, 为 `uint32_t` 类型, 对应实际转速  $0.01\text{dps}/\text{LSB}$ , 即 36000 代表  $360\text{dps}$ 。

数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0xA4
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x0C
DATA[4]	帧头校验字节	DATA[0]~DATA[3]字节校验和
DATA[5]	位置控制低字节 1	DATA[5] = *((uint8_t *)&angleControl)
DATA[6]	位置控制字节 2	DATA[6] = *((uint8_t *)&angleControl)+1)

DATA[7]	位置控制字节 3	DATA[7] = *((uint8_t *)&angleControl)+2)
DATA[8]	位置控制字节 4	DATA[8] = *((uint8_t *)&angleControl)+3)
DATA[9]	位置控制字节 5	DATA[9] = *((uint8_t *)&angleControl)+4)
DATA[10]	位置控制字节 6	DATA[10] = *((uint8_t *)&angleControl)+5)
DATA[11]	位置控制字节 7	DATA[11] = *((uint8_t *)&angleControl)+6)
DATA[12]	位置控制高字节 8	DATA[12] = *((uint8_t *)&angleControl)+7)
DATA[13]	速度限制低字节 1	DATA[13] = *(uint8_t *)&maxSpeed)
DATA[14]	速度限制字节 2	DATA[14] = *((uint8_t *)&maxSpeed)+1)
DATA[15]	速度限制字节 3	DATA[15] = *((uint8_t *)&maxSpeed)+2)
DATA[16]	速度限制高字节 4	DATA[16] = *((uint8_t *)&maxSpeed)+3)
DATA[17]	数据校验字节	DATA[5]~DATA[16]字节校验和

备注:

1. 该命令下的控制值 angleControl 受上位机(LK-Motor Tool)中的 Max Angle 值限制。
2. 该控制模式下, 电机的最大加速度由上位机(LK-Motor Tool)中的 Max Acceleration 值限制。
3. 该控制模式下, MF 和 MG 电机的最大转矩电流由上位机(LK-Motor Tool)中的 Max Torque Current 值限制; MS 电机的最大功率由上位机(LK-Motor Tool)中的 Max Power 值限制。

#### 驱动回复 (13byte)

电机在收到命令后回复主机, 该帧数据中包含了以下参数。

1. 电机温度 temperature (int8\_t 类型, 1°C/LSB)。
2. MF、MG 的转矩电流值 iq 或 MS 的输出功率值 power。iq 为 int16\_t 类型, 范围-2048~2048, 对应实际转矩电流范围-33A~33A; power 为 int16\_t 类型, 范围-1000~1000。
3. 电机转速 speed (int16\_t 类型, 1dps/LSB)。
4. 编码器位置值 encoder (uint16\_t 类型, 14bit 编码器的数值范围 0~16383)。

数据域	说明	数据	
DATA[0]	头字节	0x3E	
DATA[1]	命令字节	0xA4	
DATA[2]	ID 字节	0x01~0x20	
DATA[3]	数据长度字节	0x07	
DATA[4]	帧头校验字节	DATA[0]到 DATA[3]的校验和	
DATA[5]	电机温度	DATA[5] = *((uint8_t *)&temperature)	
DATA[6]	转矩电流低字节	DATA[6] = *((uint8_t *)&iq)	MF MG
	输出功率低字节	DATA[6] = *((uint8_t *)&power)	MS
DATA[7]	转矩电流高字节	DATA[7] = *((uint8_t *)&iq)+1)	MF MG
	输出功率高字节	DATA[7] = *((uint8_t *)&power)+1)	MS
DATA[8]	电机速度低字节	DATA[8] = *((uint8_t *)&speed)	
DATA[9]	电机速度高字节	DATA[9] = *((uint8_t *)&speed)+1)	
DATA[10]	编码器位置低字节	DATA[10] = *((uint8_t *)&encoder)	
DATA[11]	编码器位置高字节	DATA[11] = *((uint8_t *)&encoder)+1)	
DATA[12]	数据校验字节	DATA[5]到 DATA[11]的校验和	

#### (23) 单圈位置闭环控制命令 1 (10byte)

主机发送该命令以控制电机的位置 (单圈角度)。

1. 控制值 spinDirection 设置电机转动的方向, 为 uint8\_t 类型, 0x00 代表顺时针, 0x01 代表逆时针
2. 控制值 angleControl 为 uint16\_t 类型, 数值范围 0~35999, 对应实际位置为 0.01degree/LSB, 即实际角度范围 0° ~359.99°。

数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0xA5
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x04
DATA[4]	帧头校验字节	DATA[0]~DATA[3]字节校验和
DATA[5]	转动方向字节	DATA[5] = spinDirection
DATA[6]	位置控制低字节 1	DATA[6] = *(uint8_t *)&angleControl)
DATA[7]	位置控制高字节 2	DATA[7] = *((uint8_t *)&angleControl)+1)
DATA[8]	NULL	0x00
DATA[9]	数据校验字节	DATA[5]~DATA[8]字节校验和

备注:

1. 该命令下电机的最大速度由上位机(LK-Motor Tool)中的 Max Speed 值限制。
2. 该控制模式下, 电机的最大加速度由上位机(LK-Motor Tool)中的 Max Acceleration 值限制。
3. 该控制模式下, MF 和 MG 电机的最大转矩电流由上位机(LK-Motor Tool)中的 Max Torque Current 值限制; MS 电机的最大功率由上位机(LK-Motor Tool)中的 Max Power 值限制。

#### 驱动回复 (13byte)

电机在收到命令后回复主机, 该帧数据中包含了以下参数。

1. 电机温度 temperature (int8\_t 类型, 1°C/LSB)。
2. MF、MG 的转矩电流值 iq 或 MS 的输出功率值 power。iq 为 int16\_t 类型, 范围-2048~2048, 对应实际转矩电流范围-33A~33A; power 为 int16\_t 类型, 范围-1000~1000。
3. 电机转速 speed (int16\_t 类型, 1dps/LSB)。
4. 编码器位置值 encoder (uint16\_t 类型, 14bit 编码器的数值范围 0~16383)。

数据域	说明	数据	
DATA[0]	头字节	0x3E	
DATA[1]	命令字节	0xA5	
DATA[2]	ID 字节	0x01~0x20	
DATA[3]	数据长度字节	0x07	
DATA[4]	帧头校验字节	DATA[0]到 DATA[3]的校验和	
DATA[5]	电机温度	DATA[5] = *(uint8_t *)(&temperature)	
DATA[6]	转矩电流低字节	DATA[6] = *(uint8_t *)(&iq)	MF MG
	输出功率低字节	DATA[6] = *(uint8_t *)(&power)	MS
DATA[7]	转矩电流高字节	DATA[7] = *((uint8_t *)(&iq)+1)	MF MG
	输出功率高字节	DATA[7] = *((uint8_t *)(&power)+1)	MS
DATA[8]	电机速度低字节	DATA[8] = *(uint8_t *)(&speed)	
DATA[9]	电机速度高字节	DATA[9] = *((uint8_t *)(&speed)+1)	
DATA[10]	编码器位置低字节	DATA[10] = *(uint8_t *)(&encoder)	
DATA[11]	编码器位置高字节	DATA[11] = *((uint8_t *)(&encoder)+1)	
DATA[12]	数据校验字节	DATA[5]到 DATA[11]的校验和	

#### (24) 单圈位置闭环控制命令 2 (14byte)

主机发送该命令以控制电机的位置 (单圈角度)。

1. 控制值 spinDirection 设置电机转动的方向, 为 uint8\_t 类型, 0x00 代表顺时针, 0x01 代表逆时针
2. 控制值 angleControl 为 uint16\_t 类型, 数值范围 0~35999, 对应实际位置为 0.01degree/LSB, 即实际角度范围 0° ~359.99°。

3. 控制值 `maxSpeed` 限制了电机转动的最大速度，为 `uint32_t` 类型，对应实际转速 0.01dps/LSB，即 36000 代表 360dps。

数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0xA6
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x08
DATA[4]	帧头校验字节	DATA[0]~DATA[3]字节校验和
DATA[5]	转动方向字节	DATA[5] = <code>spinDirection</code>
DATA[6]	位置控制低字节 1	DATA[6] = <code>*(uint8_t *)&amp;angleControl</code>
DATA[7]	位置控制高字节 2	DATA[7] = <code>*((uint8_t *)&amp;angleControl)+1</code>
DATA[8]	NULL	0x00
DATA[9]	速度限制低字节 1	DATA[9] = <code>*(uint8_t *)&amp;maxSpeed</code>
DATA[10]	速度限制字节 2	DATA[10] = <code>*((uint8_t *)&amp;maxSpeed)+1</code>
DATA[11]	速度限制字节 3	DATA[11] = <code>*((uint8_t *)&amp;maxSpeed)+2</code>
DATA[12]	速度限制高字节 4	DATA[12] = <code>*((uint8_t *)&amp;maxSpeed)+3</code>
DATA[13]	数据校验字节	DATA[5]~DATA[12]字节校验和

备注：

1. 该控制模式下，电机的最大加速度由上位机(LK-Motor Tool)中的 `Max Acceleration` 值限制。
2. 该控制模式下，MF 和 MG 电机的最大转矩电流由上位机(LK-Motor Tool)中的 `Max Torque Current` 值限制；MS 电机的最大功率由上位机中的 `Max Power` 值限制。

#### 驱动回复 (13byte)

电机在收到命令后回复主机，该帧数据中包含了以下参数。

1. 电机温度 `temperature` (`int8_t` 类型，1°C/LSB)。
2. MF、MG 的转矩电流值 `iq` 或 MS 的输出功率值 `power`。`iq` 为 `int16_t` 类型，范围-2048~2048，对应实际转矩电流范围-33A~33A；`power` 为 `int16_t` 类型，范围-1000~1000。
3. 电机转速 `speed` (`int16_t` 类型，1dps/LSB)。
4. 编码器位置值 `encoder` (`uint16_t` 类型，14bit 编码器的数值范围 0~16383)。

数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0xA6
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x07
DATA[4]	帧头校验字节	DATA[0]到 DATA[3]的校验和
DATA[5]	电机温度	DATA[5] = <code>*(uint8_t *)&amp;temperature</code>
DATA[6]	转矩电流低字节	DATA[6] = <code>*(uint8_t *)&amp;iq</code>
	输出功率低字节	DATA[6] = <code>*(uint8_t *)&amp;power</code>
DATA[7]	转矩电流高字节	DATA[7] = <code>*((uint8_t *)&amp;iq)+1</code>
	输出功率高字节	DATA[7] = <code>*((uint8_t *)&amp;power)+1</code>
DATA[8]	电机速度低字节	DATA[8] = <code>*(uint8_t *)&amp;speed</code>
DATA[9]	电机速度高字节	DATA[9] = <code>*((uint8_t *)&amp;speed)+1</code>
DATA[10]	编码器位置低字节	DATA[10] = <code>*(uint8_t *)&amp;encoder</code>
DATA[11]	编码器位置高字节	DATA[11] = <code>*((uint8_t *)&amp;encoder)+1</code>
DATA[12]	数据校验字节	DATA[5]到 DATA[11]的校验和



(25) 增量位置闭环控制命令 1 (10byte)

主机发送该命令以控制电机的增量位置。

1. 控制值 angleIncrement 为 int32\_t 类型，对应实际位置为 0.01degree/LSB，即 36000 代表 360°，电机转动方向由该参数的符号决定。

数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0xA7
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x03
DATA[4]	帧头校验字节	DATA[0]~DATA[3]字节校验和
DATA[5]	增量位置控制低字节 1	DATA[5] = *((uint8_t *)&angleIncrement)
DATA[6]	增量位置控制字节 2	DATA[6] = *((uint8_t *)&angleIncrement)+1
DATA[7]	增量位置控制字节 3	DATA[7] = *((uint8_t *)&angleIncrement)+2
DATA[8]	增量位置控制高字节 4	DATA[8] = *((uint8_t *)&angleIncrement)+3
DATA[9]	数据校验字节	DATA[5]~DATA[8]字节校验和

备注：

1. 该命令下电机的最大速度由上位机(LK-Motor Tool)中的 Max Speed 值限制。
2. 该控制模式下，电机的最大加速度由上位机(LK-Motor Tool)中的 Max Acceleration 值限制。
3. 该控制模式下，MF 和 MG 电机的最大转矩电流由上位机(LK-Motor Tool)中的 Max Torque Current 值限制；MS 电机的最大功率由上位机(LK-Motor Tool)中的 Max Power 值限制。

驱动回复 (13byte)

电机在收到命令后回复主机，该帧数据中包含了以下参数。

1. 电机温度 temperature (int8\_t 类型，1°C/LSB)。
2. MF、MG 的转矩电流值 iq 或 MS 的输出功率值 power。iq 为 int16\_t 类型，范围-2048~2048，对应实际转矩电流范围-33A~33A；power 为 int16\_t 类型，范围-1000~1000。
3. 电机转速 speed (int16\_t 类型，1dps/LSB)。
4. 编码器位置值 encoder (uint16\_t 类型，14bit 编码器的数值范围 0~16383)。

数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0xA7
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x07
DATA[4]	帧头校验字节	DATA[0]到 DATA[3]的校验和
DATA[5]	电机温度	DATA[5] = *((uint8_t *)&temperature)
DATA[6]	转矩电流低字节	DATA[6] = *((uint8_t *)&iq)
	输出功率低字节	DATA[6] = *((uint8_t *)&power)
DATA[7]	转矩电流高字节	DATA[7] = *((uint8_t *)&iq)+1
	输出功率高字节	DATA[7] = *((uint8_t *)&power)+1
DATA[8]	电机速度低字节	DATA[8] = *((uint8_t *)&speed)
DATA[9]	电机速度高字节	DATA[9] = *((uint8_t *)&speed)+1
DATA[10]	编码器位置低字节	DATA[10] = *((uint8_t *)&encoder)
DATA[11]	编码器位置高字节	DATA[11] = *((uint8_t *)&encoder)+1
DATA[12]	数据校验字节	DATA[5]到 DATA[11]的校验和



**(26) 增量位置闭环控制命令 2 (14byte)**

主机发送该命令以控制电机的增量位置。

1. 控制值 `angleIncrement` 为 `int32_t` 类型，对应实际位置为 `0.01degree/LSB`，即 36000 代表  $360^\circ$ ，电机转动方向由该参数的符号决定。
2. 控制值 `maxSpeed` 限制了电机转动的最大速度，为 `uint32_t` 类型，对应实际转速 `0.01dps/LSB`，即 36000 代表 360dps。

数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0xA8
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x08
DATA[4]	帧头校验字节	DATA[0]~DATA[3]字节校验和
DATA[5]	增量位置控制低字节 1	DATA[5] = *((uint8_t *)&angleIncrement)
DATA[6]	增量位置控制字节 2	DATA[6] = *((uint8_t *)&angleIncrement)+1
DATA[7]	增量位置控制字节 3	DATA[7] = *((uint8_t *)&angleIncrement)+2
DATA[8]	增量位置控制高字节 4	DATA[8] = *((uint8_t *)&angleIncrement)+3
DATA[9]	速度限制低字节 1	DATA[9] = *((uint8_t *)&maxSpeed)
DATA[10]	速度限制字节 2	DATA[10] = *((uint8_t *)&maxSpeed)+1
DATA[11]	速度限制字节 3	DATA[11] = *((uint8_t *)&maxSpeed)+2
DATA[12]	速度限制高字节 4	DATA[12] = *((uint8_t *)&maxSpeed)+3
DATA[13]	数据校验字节	DATA[5]~DATA[12]字节校验和

备注：

1. 该控制模式下，电机的最大加速度由上位机(LK-Motor Tool)中的 Max Acceleration 值限制。
2. 该控制模式下，MF 和 MG 电机的最大转矩电流由上位机(LK-Motor Tool)中的 Max Torque Current 值限制；MS 电机的最大功率由上位机(LK-Motor Tool)中的 Max Power 值限制。

**驱动回复 (13byte)**

电机在收到命令后回复主机，该帧数据中包含了以下参数。

1. 电机温度 `temperature` (`int8_t` 类型， $1^\circ\text{C}/\text{LSB}$ )。
2. MF、MG 的转矩电流值 `iq` 或 MS 的输出功率值 `power`。`iq` 为 `int16_t` 类型，范围-2048~2048，对应实际转矩电流范围-33A~33A；`power` 为 `int16_t` 类型，范围-1000~1000。
3. 电机转速 `speed` (`int16_t` 类型，`1dps/LSB`)。
4. 编码器位置值 `encoder` (`uint16_t` 类型，14bit 编码器的数值范围 0~16383)。

数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0xA8
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x08
DATA[4]	帧头校验字节	DATA[0]到 DATA[3]的校验和
DATA[5]	电机温度	DATA[5] = *((uint8_t *)&temperature)
DATA[6]	转矩电流低字节	DATA[6] = *((uint8_t *)&iq)
	输出功率低字节	DATA[6] = *((uint8_t *)&power)
DATA[7]	转矩电流高字节	DATA[7] = *((uint8_t *)&iq)+1
	输出功率高字节	DATA[7] = *((uint8_t *)&power)+1

DATA[8]	电机速度低字节	DATA[8] = *(uint8_t *)&speed)
DATA[9]	电机速度高字节	DATA[9] = *((uint8_t *)&speed)+1)
DATA[10]	编码器位置低字节	DATA[10] = *(uint8_t *)&encoder)
DATA[11]	编码器位置高字节	DATA[11] = *((uint8_t *)&encoder)+1)
DATA[12]	数据校验字节	DATA[5]到 DATA[11]的校验和

#### (27) 读取驱动和电机型号命令 (5byte)

该命令用来读取驱动型号、电机型号、硬件版本号和固件版本号

数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0x12
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x00
DATA[4]	帧头校验字节	DATA[0]~DATA[3]字节校验和

例如，主机向 1#驱动发送该命令如下(HEX)

3E 12 01 00 51

#### 驱动回复 (48byte)

数据域	说明	数据
DATA[0]	头字节	0x3E
DATA[1]	命令字节	0x12
DATA[2]	ID 字节	0x01~0x20
DATA[3]	数据长度字节	0x2A
DATA[4]	帧头校验字节	DATA[0]~DATA[3]字节校验和
DATA[5~46]	驱动设备信息	productInfo 结构体
DATA[47]	数据校验字节	DATA[5]~DATA[46]字节的校验和

驱动设备信息的 productInfo 结构体如下：

```

struct productInfo
{
    uint8_t driverName[20];           // 驱动名称
    uint8_t motorName[20];           // 电机名称
    uint8_t hardwareVersion;          // 驱动硬件版本
    uint8_t firmwareVersion;          // 固件版本
};

```

其中，上位机中显示的驱动硬件版本= hardwareVersion/10.0f，固件版本= firmwareVersion/10.0f