LK-TECH Servo motor Protocol（RS485）V3.2

Disclaimer

Thank you for using the LK-M series motor drive system. Before use, please read this statement carefully. Once used, it will be regarded as acceptance of all contents of this statement.Please use the motor which strictly abide by the manual, product description and relevant laws, regulations, policies, installation guidelines. In the process of using the product, the user promises to be responsible for his behavior. Due to improper use, installation, modification caused by any loss, Shanghai Lingkong Technology Co.,(LK-TECH) will not bear legal responsibility.

This manual include all commands for differ LK-M series(MS,MF,MG)

CONTENTS

1. RS485 bus parameters and single motor command data frame format

Bus interface:RS485

Baud rate: 9600, 19200, 57600, 115200(default)，230400，460800，1Mbps，2Mbps

Data bit:8

Odd-even check: None

Stop bit: 1

2. Command frame description and single motor command list

Up to 32 drivers (depending on bus load) can be mounted on the same bus. To prevent bus collisions, each driver needs to set a different ID. Refer to the basic settings in the previous section.The main control sends control commands to the driver, and the driver associated ID analyzes the data after receiving the commands, and selects the control mode (Angle closed loop, speed closed loop and torque closed loop) according to the command type, and sends a reply to the main control after a period of time (within 0.5ms).

Each control command is composed of 2 parts: frame header + data, which are specified as follows:

Type	Data description	Length Of the Data	Instructions
Frame Command	Head Byte	1	Frame Head Identification,0x3E
	Command Byte	1	CMD
	ID Byte	1	1-32，Corresponding to the ID of the motor
	Data Length Byte	1	Describes the length of the data, depending on the different command
	Frame Header Check Byte	1	Header checksum
Frame Data	Data	0-60	Data attached to the command, depending on the different command
	Data Check Byte	0 or 1	Data checkSum

RS485 control commands supported by LK-TECH motor drive as following table:

Item	Name	Command Data
1.	Read the PID parameter command	0x30
2.	Write PID parameter to RAM command	0x31
3.	Write PID parameter to ROM command	0x32
4.	Read acceleration command	0x33
5.	Write acceleration to RAM command	0x34
6.	Read encoder command	0x90
7.	Writes the encoder value to ROM as the motor zero command	0x91
8.	Write the current position to ROM as the motor zero command	0x19
9.	Read multi -loop Angle command	0x92
10.	Read single -loop Angle command	0x94

11.	Read motor status 1 and error flag command	0x9A
12.	Clear motor error flag command	0x9B
13.	Read motor status 2 command	0x9C
14.	Read motor status 3 command	0x9D
15.	Motor shutdown command	0x80
16.	Motor stop command	0x81
17.	Motor operation command	0x88
18.	Open loop control command	0XA0
19.	Torque closed loop control command	0xA1
20.	Speed closed loop control command	0xA2
21.	Multi position closed loop control command 1	0xA3
22.	Multi position closed loop control command 2	0xA4
23.	Single position closed loop control command 1	0xA5
24.	Single position closed loop control command 2	0xA6
25.	Incremental closed-loop control command 1	0xA7
26.	Incremental closed-loop control command 2	0xA8
27.	Read driver and motor model commands	0x12

3. Single motor command description

(1) Read PID parameter command (5byte)

The computer host sends command to read the PID parameter of the current motor.

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E
DATA[1]	Command byte	0x30
DATA[2]	ID byte	0x01~0x20
DATA[3]	Data length byte	0x00
DATA[4]	Frame header check byte	DATA[0]~DATA[3]Byte checksum

Driver respond(12byte)

The data of driver respond contains PI parameters of each control loop

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E
DATA[1]	Command byte	0x30
DATA[2]	ID byte	0x01~0x20
DATA[3]	Data length byte	0x06
DATA[4]	Frame header check byte	DATA[0]~DATA[3]Byte checksum
DATA[5]		DATA[5] = anglePidKp
DATA[6]	I parameter of position ring	DATA[6] = anglePidKi
DATA[7]	P parameter of speed ring	DATA[7] = speedPidKp
DATA[8]	I parameter of speed ring	DATA[8] = speedPidKi
DATA[9]	P parameter of torque ring	DATA[9] = iqPidKp
DATA[10]	I parameter of torque ring	DATA[10] = iqPidKi
DATA[11]	Frame header check byte	DATA[5]~DATA[10]Byte checksum

(2)Write PID parameter to RAM command (12byte)

The computer host sends command to write PID parameters into RAM, and the parameters become invalid when power off.

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E
DATA[1]	Command byte	0x31
DATA[2]	ID byte	0x01~0x20
DATA[3]	Data length byte	0x06
DATA[4]	Frame header check byte	DATA[0]~DATA[3]Byte checksum
DATA[5]	P parameter of position ring	DATA[5] = anglePidKp
DATA[6]	I parameter of position ring	DATA[6] = anglePidKi
DATA[7]	P parameter of speed ring	DATA[7] = speedPidKp
DATA[8]	I parameter of speed ring	DATA[8] = speedPidKi
DATA[9]	P parameter of torque ring	DATA[9] = iqPidKp
DATA[10]	I parameter of torque ring	DATA[10] = iqPidKi
DATA[11]	Frame header check byte	DATA[5]~DATA[10]Byte checksum

Driver respond(12byte)

The drive reply data is consistent with the received command parameters.

(3) Write PID parameter to ROM command (12byte)

The computer host sends the command to write the PID parameter to RAM. It is still valid when power off.

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E
DATA[1]	Command byte	0x32
DATA[2]	ID byte	0x01~0x20
DATA[3]	Data length byte	0x06
DATA[4]	Frame header check byte	DATA[0]~DATA[3]Byte checksum
DATA[5]	P parameter of position ring	DATA[5] = anglePidKp
DATA[6]	I parameter of position ring	DATA[6] = anglePidKi
DATA[7]	P parameter of speed ring	DATA[7] = speedPidKp
DATA[8]	I parameter of speed ring	DATA[8] = speedPidKi
DATA[9]	P parameter of torque ring	DATA[9] = iqPidKp
DATA[10]	I parameter of torque ring	DATA[10] = iqPidKi
DATA[11]	Frame header check byte	DATA[5]~DATA[10]Byte checksum

Driver respond(12byte)

The driver reply data is consistent with the received command parameters.

(4) Read acceleration command (5byte)

The computer host sends this command to read the acceleration parameters of the current motor.

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E

DATA[1]	Command byte	0x33
DATA[2]	ID byte	0x01~0x20
DATA[3]	Data length byte	0x00
DATA[4]	Frame header check byte	DATA[0]~DATA[3]Byte checksum

Driver respond(10byte)

Acceleration parameters are included in the driver reply data. The acceleration data is of int32_t type, with a unit of 1dps/s.

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E
DATA[1]	Command byte	0x33
DATA[2]	ID byte	0x01~0x20
DATA[3]	Data length byte	0x04
DATA[4]	Frame header check byte	DATA[0]~DATA[3]Byte checksum
DATA[5]	Acceleration low byte 1	DATA[5] = *(uint8_t *)(&Accel)
DATA[6]	Acceleration byte 2	DATA[6] = *((uint8_t *)(&Accel)+1)
DATA[7]	Acceleration byte 3	DATA[7] = *((uint8_t *)(&Accel)+2)
DATA[8]	Acceleration byte 4	DATA[8] = *((uint8_t *)(&Accel)+3)
DATA[9]	Data check byte	DATA[5]~DATA[8]Byte checksum

(5) Write acceleration to RAM command (10byte)

The computer host sends the command to write acceleration parameters into RAM, and the parameters will lose when power off. Acceleration data is int32_t type, unit 1dps/s.

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E
DATA[1]	Command byte	0x34
DATA[2]	ID byte	0x01~0x20
DATA[3]	Data length byte	0x04
DATA[4]	Frame header check byte	DATA[0]~DATA[3]Byte checksum
DATA[5]	Acceleration low byte 1	DATA[5] = *(uint8_t *)(&Accel)
DATA[6]	Acceleration byte 2	DATA[6] = *((uint8_t *)(&Accel)+1)
DATA[7]	Acceleration byte 3	DATA[7] = *((uint8_t *)(&Accel)+2)
DATA[8]	Acceleration byte 4	DATA[8] = *((uint8_t *)(&Accel)+3)
DATA[10]	Data check byte	DATA[5]~DATA[8]Byte checksum

Driver respond(10byte)

The drive reply data is consistent with the received command parameters

(6) Read encoder command (5byte)

The computer host sends command to read the current position of the encoder.

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E
DATA[1]	Command byte	0x90
DATA[2]	ID byte	0x01~0x20
DATA[3]	Data length byte	0x00
DATA[4]	Frame header check byte	DATA[0]~DATA[3]Byte checksum

Driver respond(12byte)

The motor replies to the computer host after receiving the command, and the reply data contains the following parameters.

1.Encoder position encoder (uint16_t type, eg:14bit encoder value range 0~16383), which is the original position of encoder minus encoder offset.

2.Original position of encoder (uint16_t type, eg:14bit encoder value range 0~16383).

3.EncoderOffset (uint16_t type, eg:14bit encoder value range 0~16383), and this point is taken as the 0 point of the motor Angle.

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E
DATA[1]	Command byte	0x90
DATA[2]	ID byte	0x01~0x20
DATA[3]	Data length byte	0x06
DATA[4]	Frame header check byte	DATA[0]~DATA[3]Byte checksum
DATA[5]	Encoder data in low bytes	=*(uint8_t *)(&encoder)
DATA[6]	Encoder data in high bytes	=*((uint8_t *)(&encoder)+1)
DATA[7]	Encoder original position low byte	=*(uint8_t *)(&encoder Raw)
DATA[8]	Encoder original position high byte	=*((uint8_t *)(&encoder Raw)+1)
DATA[9]	Encoder zero low byte	= *(uint8_t *)(&encoder Offset)
DATA[10]	Encoder zero high byte	= *((uint8_t *)(&encoder Offset)+1)
DATA[11]	Data check byte	DATA[5]~DATA[10]Byte checksum

(7) Write encoder value as motor zero point command (8byte)

The computer host sends the command to set the encoder Offset , that the encoder Offset to be written is the type of uint16_t, and value range of the 14bit encoder is 0~16383.

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E
DATA[1]	Command byte	0x91
DATA[2]	ID byte	0x01~0x20
DATA[3]	Data length byte	0x02
DATA[4]	Frame header check byte	DATA[0]~DATA[3]Byte checksum
DATA[5]	Encoder zero low byte	= *(uint8_t *)(&encoderOffset)
DATA[6]	Encoder zero high byte	= *((uint8_t *)(&encoderOffset)+1)
DATA[7]	Data check byte	DATA[5]~DATA[6]Byte checksum

Driver respond(8byte)

The drive reply data is consistent with the received command parameters

(8) Write the current position to ROM as the zero point command of the motor (5byte)

Writes the current encoder position of the motor into ROM as the initial position.

Attention:

1. This command needs to restart to take effect.
2. This command will write zero point into ROM of the driver, multiple writing will affect the chip life,which is not recommended for frequent use

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E
DATA[1]	Command byte	0x19
DATA[2]	ID byte	0x01~0x20
DATA[3]	Data length byte	0x00
DATA[4]	Frame header check byte	DATA[0]~DATA[3]Byte checksum

For example, the computer host sends zero point setting command to the 1# driver (HEX) as following.

3E 19 01 00 58

Driver respond(26 byte)

The command is not open yet.

(9) Read multi-loop Angle command (5byte)

The computer host sends command to read the absolute multi-turn Angle of the current motor.

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E
DATA[1]	Command byte	0x92
DATA[2]	ID byte	0x01~0x20
DATA[3]	Data length byte	0x00
DATA[4]	Frame header check byte	DATA[0]~DATA[3]Byte checksum

Drive respond(14byte)

The motor replies to the computer host after receiving the command, and the frame data contains the following parameters:

1.Motor-angle, int64_t type data, positive value represents clockwise cumulative Angle, negative value represents counter clockwise cumulative Angle, unit 0.01° /LSB.

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E
DATA[1]	Command byte	0x92
DATA[2]	ID byte	0x01~0x20
DATA[3]	Data length byte	0x08
DATA[4]	Frame header check byte	From DATA[0] to DATA[3]checksum
DATA[5]	Angle low byte 1	DATA[5] = *(uint8_t *)(&motorAngle)
DATA[6]	Angle byte 2	DATA[6] = *((uint8_t *)(& motorAngle)+1)
DATA[7]	Angle byte 3	DATA[7] = *((uint8_t *)(& motorAngle)+2)
DATA[8]	Angle byte 4	DATA[8] = *((uint8_t *)(& motorAngle)+3)
DATA[9]	Angle byte 5	DATA[9] = *((uint8_t *)(& motorAngle)+4)
DATA[10]	Angle byte 6	DATA[10] = *((uint8_t *)(& motorAngle)+5)
DATA[11]	Angle byte 7	DATA[11] = *((uint8_t *)(& motorAngle)+6)
DATA[12]	Angle high byte 8	DATA[12] = *((uint8_t *)(& motorAngle)+6)
DATA[13]	Data check byte	From DATA[5] to DATA[12]checksum

(10) Read single-loop Angle command (5 byte)

The computer host sends command to read the absolute single-turn Angle of the current motor.

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E
DATA[1]	Command byte	0x94
DATA[2]	ID byte	0x01~0x20
DATA[3]	Data length byte	0x00
DATA[4]	Frame header check byte	From DATA[0] to DATA[3]checksum

Driver respond(10 byte)

The motor replies to the computer host after receiving the command, the frame data contains the following parameters:

1.The single-loop angle of the motor,uint32_t type data, which takes encoder zero point as the starting point, increases clockwise, and when it reaches zero again, the value returns to 0, unit 0.01° /LSB, and the value range is 0~36000*i-1（i：Reduction ratio）.

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E
DATA[1]	Command byte	0x94
DATA[2]	ID byte	0x01~0x20
DATA[3]	Data length byte	0x04
DATA[4]	Frame header check byte	From DATA[0] to DATA[3]checksum
DATA[5]	Single loop Angle low byte 1	DATA[5] = *(uint8_t *)(&circleAngle)
DATA[6]	Single loop Angle byte 2	DATA[6] = *((uint8_t *)(& circleAngle)+1)
DATA[7]	Single loop Angle byte 3	DATA[6] = *((uint8_t *)(& circleAngle)+1)
DATA[8]	Single loop Angle high byte 4	DATA[6] = *((uint8_t *)(& circleAngle)+1)
DATA[9]	Data check byte	From DATA[5] to DATA[8]checksum

(11) Read motor state 1 and error flag command (5byte)

This command reads the current motor's temperature, voltage, and error status flags.

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E
DATA[1]	Command byte	0x9A
DATA[2]	ID byte	0x01~0x20
DATA[3]	Data length byte	0x00
DATA[4]	Frame header check byte	Fromd DATA[0] to DATA[3]checksum

Driver respond(13byte)

The motor replies to the host after receiving the command, and the frame data contains the following parameters:

1. Motor temperature (int8_t type, unit 1℃/LSB).
2. Voltage (uint16_t, unit 0.1v /LSB).
3. ErrorState (uint8_t type, each bit represents different motor state)

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E
DATA[1]	Command byte	0x9A

DATA[2]	ID byte	0x01~0x20
DATA[3]	Data length byte	0x07
DATA[4]	Frame header check byte	From DATA[0] to DATA[3]checksum
DATA[5]	Motor temperature	DATA[5] = *(uint8_t *)(&temperature)
DATA[6]	NULL	0x00
DATA[7]	voltage low byte	DATA[7] = *(uint8_t *)(&voltage)
DATA[8]	voltage high byte	DATA[8] = *((uint8_t *)(& voltage)+1)
DATA[9]	NULL	0x00
DATA[10]	NULL	0x00
DATA[11]	Error status byte	DATA[11]=errorState
DATA[12]	Data check byte	From DATA[5] to DATA[11]checksum

Remark:

1. The specific status table of each bit of errorState is as follows.

errorState byte	State instructions	0	1
0	Voltage condition	Normal	Low voltage protection
1	NULL		
2	NULL		
3	Temperature condition	Normal	Over temperature protection
4	NULL		
5	NULL		
6	NULL		
7	NULL		

(12) Clear motor error mark command (5byte)

This command clears the current motor error state and the motor returns when it is received.

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E
DATA[1]	Command byte	0x9B
DATA[2]	ID byte	0x01~0x20
DATA[3]	Data length byte	0x00
DATA[4]	Frame header check byte	From DATA[0] to DATA[3]checksum

Driver respond(13byte)

The motor replies to the host after receiving the command, and the frame data contains the following parameters.

1.Motor temperature (int8_t type, unit 1℃/LSB).

2.Voltage (uint16_t, unit 0.1v /LSB).

3.ErrorState (uint8_t type, each bit represents different motor state)

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E
DATA[1]	Command byte	0x9B
DATA[2]	ID byte	0x01~0x20
DATA[3]	Data length byte	0x07
DATA[4]	Frame header check byte	From DATA[0] to DATA[3]checksum

DATA[5]	Motor temperature	DATA[5] = *(uint8_t *)(&temperature)
DATA[6]	NULL	0x00
DATA[7]	voltage low byte	DATA[7] = *(uint8_t *)(&voltage)
DATA[8]	voltage high byte	DATA[8] = *((uint8_t *)(& voltage)+1)
DATA[9]	NULL	0x00
DATA[10]	NULL	0x00
DATA[11]	Error status byte	DATA[11]=errorState
DATA[12]	Data check byte	From DATA[5] to DATA[11]checksum

Remark:

1. If the motor state is not restored to normal, the error mark cannot be removed.
2. The specific state of each bit of error state refers to reading motor state 1 and error flag command.

(13) Read motor state 2 command (5byte)

This command reads the current motor temperature, voltage, speed, encoder position.

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E
DATA[1]	Command byte	0x9C
DATA[2]	ID byte	0x01~0x20
DATA[3]	Data length byte	0x00
DATA[4]	Frame header check byte	From DATA[0] to DATA[3]checksum

Driver respond(13byte)

The motor replies to the host after receiving the command, and the frame data contains the following parameters.

1. Motor temperature (int8_t type, 1℃/LSB).
2. Torque current IQ of the motor (int16_t type, range -2048~2048, corresponding to the actual torque current range -33A ~33A).
3. Motor speed (int16_t type, 1dps/LSB).
4. Encoder position value (uint16_t type, the value range of 14bit encoder is 0~16383).

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E
DATA[1]	Command byte	0x9C
DATA[2]	ID byte	0x01~0x20
DATA[3]	Data length byte	0x07
DATA[4]	Frame header check byte	DATA[0]到 DATA[3]的校验和
DATA[5]	Motor temperature	DATA[5] = *(uint8_t *)(&temperature)
DATA[6]	Torque current low byte	DATA[6] = *(uint8_t *)(&iq)
DATA[7]	Torque current high byte	DATA[7] = *((uint8_t *)(&iq)+1)
DATA[8]	Motor speed low bytes	DATA[8] = *(uint8_t *)(&speed)
DATA[9]	Motor speed high bytes	DATA[9] = *((uint8_t *)(&speed)+1)
DATA[10]	Encoder position low byte	DATA[10] = *(uint8_t *)(&encoder)
DATA[11]	Encoder position high byte	DATA[11] = *((uint8_t *)(&encoder)+1)
DATA[12]	Data check byte	From DATA[5] to DATA[11]checksum

(14) Read motor state 3 command (5byte)

This command reads the current motor temperature and phase current data.

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E
DATA[1]	Command byte	0x9D
DATA[2]	ID byte	0x01~0x20
DATA[3]	Data length byte	0x00
DATA[4]	Frame header check byte	From DATA[0] to DATA[3]checksum

Driver respond(13byte)

The motor replies to the host after receiving the command, and the frame data contains the following parameters.

1. Motor temperature (int8_t type, 1℃/LSB).
2. Phase A current data, data type int16_t, corresponding to the actual phase current 1A/64LSB.
3. Phase B current data, data type int16_t, corresponding to the actual phase current 1A/64LSB.
4. Phase C current data, data type int16_t, corresponding to the actual phase current 1A/64LSB.

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E
DATA[1]	Command byte	0x9D
DATA[2]	ID byte	0x01~0x20
DATA[3]	Data length byte	0x07
DATA[4]	Frame header check byte	From DATA[0] to DATA[3]checksum
DATA[5]	Motor temperature	DATA[5] = *(uint8_t *)(&temperature)
DATA[6]	A phase current low byte	DATA[6] = *(uint8_t *)(&iA)
DATA[7]	A phase current high byte	DATA[7] = *((uint8_t *)(& iA)+1)
DATA[8]	B phase current low byte	DATA[8] = *(uint8_t *)(&iB)
DATA[9]	B phase current high byte	DATA[9] = *((uint8_t *)(& iB)+1)
DATA[10]	C phase current low byte	DATA[10] = *(uint8_t *)(&iC)
DATA[11]	C phase current high byte	DATA[11] = *((uint8_t *)(& iC)+1)
DATA[12]	Data check byte	From DATA[5] to DATA[11]checksum

(15) Motor shutdown command (5byte)

Turn off the motor and clear the motor running state and the control instruction received before.

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E
DATA[1]	Command byte	0x80
DATA[2]	ID byte	0x01~0x20
DATA[3]	Data length byte	0x00
DATA[4]	Frame header check byte	From DATA[0] to DATA[3]checksum

For example, the computer host sends the motor shutdown command to driver 1# as follows (HEX).

3E 80 01 00 BF

Drive respond(5byte)

It is same as computer host send.

(16) Motor stop command (5byte)

Stop the motor, but do not clear the motor running state and previous received control instructions.

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E
DATA[1]	Command byte	0x81
DATA[2]	ID byte	0x01~0x20
DATA[3]	Data length byte	0x00
DATA[4]	Frame header check byte	From DATA[0] to DATA[3]byte checksum

For example, the host sends the motor stop command to driver 1# as follows (HEX).

3E 81 01 00 C0

Driver respond(5byte)

It is same as computer host send.

(17) Motor operation command (5byte)

Restore motor operation from motor stop command (control mode before restoration stop).

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E
DATA[1]	Command byte	0x88
DATA[2]	ID byte	0x01~0x20
DATA[3]	Data length byte	0x00
DATA[4]	Frame header check byte	From DATA[0] to DATA[3]byte checksum

For example, the host sends the motor stop command to driver 1# as follows (HEX).

3E 88 01 00 C7

Drive respond(5byte)

It is same as computer host send.

（18）Torque open loop control command （8byte） This command only for MS series

The computer host sends this command to control the output power of open loop, and the control value is int16_t type, with the value range of -1000~ 1000, (the bus current and the actual torque of the motor vary with different motors).

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E
DATA[1]	Command byte	0xA0
DATA[2]	ID byte	0x01~0x20
DATA[3]	Data length byte	0x02
DATA[4]	Frame header check byte	DATA[0]~DATA[3] byte checksum
DATA[5]	Output power control value low byte	DATA[5] = *(uint8_t *)(&powerControl)
DATA[6]	Output power control value high byte	DATA[6] = *((uint8_t *)(&powerControl)+1)
DATA[7]	Data check byte	DATA[5]~DATA[6] byte checksum

Remark:

1. The control value power control in this command is not limited by the Max power value in the LK-Motor Tool.

Driver respond(13byte)

The motor replies to the computer host after receiving the command, and the frame data contains the following data:

1. Motor temperature (int8_t type, 1℃/LSB).

2.Motor power output value (int16_t type, range -1000~1000)

3.Motor speed (int16_t type, 1dps/LSB).

4.Encoder position value (uint16_t type, eg:14bit encoder value range 0~16383).

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E
DATA[1]	Command byte	0xA0
DATA[2]	ID byte	0x01~0x20
DATA[3]	Data length byte	0x07
DATA[4]	Frame header check byte	DATA[0]toDATA[3]data checksum
DATA[5]	Motor temperature	DATA[5] = *(uint8_t *)(&temperature)
DATA[6]	Output power low byte	DATA[6] = *(uint8_t *)(&power)
DATA[7]	Output power high byte	DATA[7] = *((uint8_t *)(&power)+1)
DATA[8]	motor speed low byte	DATA[8] = *(uint8_t *)(&speed)
DATA[9]	motor speed high byte	DATA[9] = *((uint8_t *)(&speed)+1)
DATA[10]	encoder position low byte	DATA[10] = *(uint8_t *)(&encoder)
DATA[11]	encoder position high byte	DATA[11] = *((uint8_t *)(&encoder)+1)
DATA[12]	Data check byte	DATA[5]toDATA[11]data checksum

(19)Torque closed-loop control command (8byte),the command for MF and MG

The computer host sends this command to control the torque current output of the motor, and the control value is int16_t type, with the value range of -2000~ 2000, corresponding to the actual torque current range of -32A ~32A (the bus current and the actual torque of the motor vary with different motors).

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E
DATA[1]	Command byte	0xA1
DATA[2]	ID byte	0x01~0x20
DATA[3]	Data length byte	0x02
DATA[4]	Frame header check byte	From DATA[0] to DATA[3]byte checksum
DATA[5]	Torque current control value low byte	DATA[5] = *(uint8_t *)(&iqControl)
DATA[6]	Torque current control value high byte	DATA[6] = *((uint8_t *)(&iqControl)+1)
DATA[7]	Data check byte	DATA[5]~DATA[6]byte checksum

Remark:

1.The control value iqControl in this command is not limited by the Max Torque Current value in

the LK-Motor Tool.

Drive respond(13byte)

The motor replies to the computer host after receiving the command, and the frame data contains the following data:

1.Motor temperature (int8_t type, 1℃/LSB).

2.Torque current IQ of the motor (int16_t type, range -2048~2048, corresponding to the actual torque current range -33A ~33A).

3.Motor speed (int16_t type, 1dps/LSB).

4.Encoder position value (uint16_t type, 14bit encoder value range 0~16383).

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E
DATA[1]	Command byte	0xA1
DATA[2]	ID byte	0x01~0x20
DATA[3]	Data length byte	0x07
DATA[4]	Frame header check byte	DATA[0]~DATA[3]byte checksum
DATA[5]	Motor temperature	DATA[5] = *(uint8_t *)(&temperature)
DATA[6]	Torque current low byte	DATA[6] = *(uint8_t *)(&iq)
DATA[7]	Torque current high byte	DATA[7] = *((uint8_t *)(&iq)+1)
DATA[8]	Motor speed low bytes	DATA[8] = *(uint8_t *)(&speed)
DATA[9]	Motor speed high bytes	DATA[9] = *((uint8_t *)(&speed)+1)
DATA[10]	Encoder data in low bytes	DATA[10] = *(uint8_t *)(&encoder)
DATA[11]	Encoder data in high bytes	DATA[11] = *((uint8_t *)(&encoder)+1)
DATA[12]	Data check byte	DATA[5]~DATA[11]byte checksum

(20)Speed closed-loop control command (10byte)

The computer host sends this command to control the speed of the motor with a speedControl of type int32_t corresponding to the actual speed of 0.01 DPS /LSB.

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E
DATA[1]	Command byte	0xA2
DATA[2]	ID byte	0x01~0x20
DATA[3]	Data length byte	0x04
DATA[4]	Frame header check byte	DATA[0]~DATA[3]byte checksum
DATA[5]	Motor speed low byte	DATA[5] = *(uint8_t *)(&speedControl)
DATA[6]	Motor speed	DATA[6] = *((uint8_t *)(&speedControl)+1)
DATA[7]	Motor speed	DATA[7] = *((uint8_t *)(&speedControl)+2)
DATA[8]	Motor speed high byte	DATA[8] = *((uint8_t *)(&speedControl)+3)
DATA[9]	Data check byte	DATA[5]~DATA[8]byte checksum

Remark:

1. Motor speed control in this command limited by max speed in RMD Assistant
2. The max acceleration in this command limited by max acceleration in RMD Assistant
3. The max torque current in this command (MF/MG) limited by max torque current in RMD Assistant ;max power of MS limited by max power in RMD Assistant

Drive respond(13byte)

The motor replies to the computer host after receiving the command, and the frame data contains the following data.

1.Motor temperature (int8_t type, 1℃/LSB).

2.Torque current IQ of the motor (int16_t type, range -2048~2048, corresponding to the actual torque current range -33A ~33A). Motor power output value (int16_t type, range -1000~1000)

3.Motor speed (int16_t type, 1dps/LSB).

4.Encoder position value (uint16_t type, 14bit encoder value range 0~16383).

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E
DATA[1]	Command byte	0xA2
DATA[2]	ID byte	0x01~0x20
DATA[3]	Data length byte	0x07
DATA[4]	Frame header check byte	DATA[0]~DATA[3]byte checksum
DATA[5]	Motor temperature	DATA[5] = *(uint8_t *)(&temperature)
DATA[6]	Torque current low byte	DATA[6] = *(uint8_t *)(&iq)
DATA[7]	Torque current high byte	DATA[7] = *((uint8_t *)(&iq)+1)
DATA[8]	Motor speed low byte	DATA[8] = *(uint8_t *)(&speed)
DATA[9]	Motor speed high byte	DATA[9] = *((uint8_t *)(&speed)+1)
DATA[10]	Encoder data low byte	DATA[10] = *(uint8_t *)(&encoder)
DATA[11]	Encoder data high byte	DATA[11] = *((uint8_t *)(&encoder)+1)
DATA[12]	Data check byte	DATA[5]~DATA[11]byte checksum

(21)Multi position closed-loop control command 1 (14byte)

The host computer sends the command to control the position of the motor (multi-turn Angle), the control value angleControl is int64_t, corresponding to the actual position is 0.01degree/LSB, that is 36000 represents 360° , and the motor rotation direction is determined by the difference between the target position and the current position.

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E
DATA[1]	Command byte	0xA3
DATA[2]	ID byte	0x01~0x20
DATA[3]	Data length byte	0x08
DATA[4]	Frame header check byte	DATA[0]~DATA[3]byte checksum
DATA[5]	Position control low byte 1	DATA[5] = *(uint8_t *)(&angleControl)
DATA[6]	Position control byte 2	DATA[6] = *((uint8_t *)(&angleControl)+1)
DATA[7]	Position control byte 3	DATA[7] = *((uint8_t *)(&angleControl)+2)
DATA[8]	Position control byte 4	DATA[8] = *((uint8_t *)(&angleControl)+3)
DATA[9]	Position control byte 5	DATA[9] = *((uint8_t *)(&angleControl)+4)
DATA[10]	Position control byte 6	DATA[10] = *((uint8_t *)(&angleControl)+5)
DATA[11]	Position control byte 7	DATA[11] = *((uint8_t *)(&angleControl)+6)
DATA[12]	Position control high byte 8	DATA[12] = *((uint8_t *)(&angleControl)+7)
DATA[13]	Data check byte	DATA[5]~DATA[13]byte checksum

Remark:

1. The control value angleControl under this command is limited by Max Angle value in the LK-Motor Tool.
2. The maximum Speed of the motor under this command is limited by the Max Speed value in the LK-Motor Tool.
3. In this control mode, the maximum Acceleration of the motor is limited by the Max Acceleration value of the LK-Motor Tool.
4. In this control mode, the maximum Torque Current of the motor is limited by Max Torque Current value in the LK-Motor Tool.

Driver respond(13byte)

The motor replies to the host after receiving the command, and the frame data contains the following data.

1.Motor temperature (int8_t type, 1℃/LSB).

2.Torque current IQ of the motor (int16_t type, range -2048~2048, corresponding to the actual torque current range -33A ~33A); Motor power output value (int16_t type, range -1000~1000)

3.Motor speed (int16_t type, 1dps/LSB).

4.Encoder position value (uint16_t type, 14bit encoder value range 0~16383).

Data Field	Instructions	Data	
DATA[0]	Head byte	0x3E	
DATA[1]	Command byte	0xA3	
DATA[2]	ID byte	0x01~0x20	
DATA[3]	Data length byte	0x07	
DATA[4]	Frame header check byte	DATA[0]~DATA[3]byte checksum	
DATA[5]	Motor temperature	DATA[5] = *(uint8_t *)(&temperature)	
DATA[6]	Torque current low byte	DATA[6] = *(uint8_t *)(&iq)	MF; MG
	Power output low byte	DATA[6] = *(uint8_t *)(&power)	MS
DATA[7]	Torque current high byte	DATA[7] = *((uint8_t *)(&iq)+1)	MF; MG
	Power output high byte	DATA[7] = *((uint8_t *)(&power)+1)	MS
DATA[8]	Motor speed low byte	DATA[8] = *(uint8_t *)(&speed)	
DATA[9]	Motor speed high byte	DATA[9] = *((uint8_t *)(&speed)+1)	
DATA[10]	Encoder data low byte	DATA[10] = *(uint8_t *)(&encoder)	
DATA[11]	Encoder data high byte	DATA[11] = *((uint8_t *)(&encoder)+1)	
DATA[12]	Data check byte	DATA[5]~DATA[11]byte checksum	

(22) Multi position closed-loop control command 2 (14byte)

The host computer sends the command to control the position of the motor (multi-turn Angle), the control value angleControl is int64_t, corresponding to the actual position is 0.01degree/LSB, that is 36000 represents 360° , and the motor rotation direction is determined by the difference between the target position and the current position. The control value maxSpeed limits the maximum speed of motor rotation, which is uint32_t type, corresponding to the actual speed of 0.01dps/LSB, namely 36000 represents 360dps.

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E
DATA[1]	Command byte	0xA4
DATA[2]	ID byte	0x01~0x20
DATA[3]	Data length byte	0x0C
DATA[4]	Frame header check byte	DATA[0]~DATA[3]byte checksum
DATA[5]	Position control low byte 1	DATA[5] = *(uint8_t *)(&angleControl)
DATA[6]	Position control byte 2	DATA[6] = *((uint8_t *)(&angleControl)+1)
DATA[7]	Position control byte 3	DATA[7] = *((uint8_t *)(&angleControl)+2)
DATA[8]	Position control byte 4	DATA[8] = *((uint8_t *)(&angleControl)+3)
DATA[9]	Position control byte 5	DATA[9] = *((uint8_t *)(&angleControl)+4)
DATA[10]	Position control byte 6	DATA[10] = *((uint8_t *)(&angleControl)+5)
DATA[11]	Position control byte 7	DATA[11] = *((uint8_t *)(&angleControl)+6)
DATA[12]	Position control high byte 1	DATA[12] = *((uint8_t *)(&angleControl)+7)
DATA[13]	Speed limit low byte 1	DATA[13] = *(uint8_t *)(&maxSpeed)
DATA[14]	Speed limit byte 2	DATA[14] = *((uint8_t *)(&maxSpeed)+1)
DATA[15]	Speed limit byte 3	DATA[15] = *((uint8_t *)(&maxSpeed)+2)
DATA[16]	Speed limit high byte 4	DATA[16] = *((uint8_t *)(&maxSpeed)+3)
DATA[17]	Data check byte	DATA[5]~DATA[16]byte checksum

Remark:

1. The control value angleControl under this command is limited by Max Angle value in the LK-Motor Tool.
2. In this control mode, the maximum Acceleration of the motor is limited by the Max Acceleration value of the LK-Motor Tool.
3. In this control mode, the maximum Torque Current of the motor is limited by Max Torque Current value in the LK-Motor Tool.

Driver respond(13byte)

The motor replies to the host after receiving the command, and the frame data contains the following data.

1.Motor temperature (int8_t type, 1℃/LSB).

2.Torque current IQ of the motor (int16_t type, range -2048~2048, corresponding to the actual torque current range -33A ~33A). Motor power output value (int16_t type, range -1000~1000)

3.Motor speed (int16_t type, 1dps/LSB).

4.Encoder position value (uint16_t type, 14bit encoder value range 0~16383).

Data Field	Instructions	Data	
DATA[0]	Head byte	0x3E	
DATA[1]	Command byte	0xA4	
DATA[2]	ID byte	0x01~0x20	
DATA[3]	Data length byte	0x07	
DATA[4]	Frame header check byte	DATA[0]~DATA[3]byte checksum	
DATA[5]	Motor temperature	DATA[5] = *(uint8_t *)(&temperature)	
DATA[6]	Torque current low byte	DATA[6] = *(uint8_t *)(&iq)	MF;MG

	Output Power low byte	DATA[6] = *(uint8_t *)(&power)	MS
DATA[7]	Torque current high byte	DATA[7] = *((uint8_t *)(&iq)+1)	MF;MG
	Output Power high byte	DATA[7] = *((uint8_t *)(&power)+1)	MS
DATA[8]	Motor speed low byte	DATA[8] = *(uint8_t *)(&speed)	
DATA[9]	Motor speed high byte	DATA[9] = *((uint8_t *)(&speed)+1)	
DATA[10]	Encoder data low byte	DATA[10] = *(uint8_t *)(&encoder)	
DATA[11]	Encoder data high byte	DATA[11] = *((uint8_t *)(&encoder)+1)	
DATA[12]	Data check byte	DATA[5]~DATA[11]byte checksum	

(23)Single position closed-loop control command 1 (10byte)

1. The host sends this command to control the position of the motor (single turn Angle).
2. The control value angleControl is uint16_t, with a range of 0~35999 and a corresponding actual position of 0.01degree/LSB, namely the actual Angle range of 0°~359.99°.

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E
DATA[1]	Command byte	0xA5
DATA[2]	ID byte	0x01~0x20
DATA[3]	Data length byte	0x03
DATA[4]	Frame header check byte	DATA[0]~DATA[3]byte checksum
DATA[5]	Rotation direction byte	DATA[5] = spinDirection
DATA[6]	Position control low byte 1	DATA[6] = *(uint8_t *)(&angleControl)
DATA[7]	Position control high byte 2	DATA[7] = *((uint8_t *)(&angleControl)+1)
DATA[8]	null	0x00
DATA[9]	Data check byte	DATA[5]~DATA[8]byte checksum

Remarks:

1.The maximum Speed of the motor under this command is limited by the Max Speed value in the LK-Motor Tool.

2.In this control mode, the maximum Acceleration of the motor is limited by the value of Max Acceleration in the LK-Motor Tool.

3.In this control mode, the maximum Torque Current of the motor is limited by Max Torque Current value in the LK-Motor Tool.

Driver respond(13byte)

The motor replies to the host after receiving the command, and the frame data contains the following data.

1.Motor temperature (int8_t type, 1℃/LSB).

2.Torque current IQ of the motor (int16_t type, range -2048~2048, corresponding to the actual torque current range -33A ~33A). Motor power output value (int16_t type, range -1000~1000)

3.Motor speed (int16_t type, 1dps/LSB).

4.Encoder position value (uint16_t type, 14bit encoder value range 0~16383).

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E
DATA[1]	Command byte	0xA5
DATA[2]	ID byte	0x01~0x20

DATA[3]	Data length byte	0x07	
DATA[4]	Frame header check byte	DATA[0]~DATA[3]byte checksum	
DATA[5]	Motor temperature	DATA[5] = *(uint8_t *)(&temperature)	
DATA[6]	Torque current low byte	DATA[6] = *(uint8_t *)(&iq)	MG;MF
	Output Power low byte	DATA[6] = *(uint8_t *)(&power)	MS
DATA[7]	Torque current high byte	DATA[7] = *((uint8_t *)(&iq)+1)	MG;MF
	Output Power high byte	DATA[7] = *((uint8_t *)(&power)+1)	MS
DATA[8]	Motor speed low byte	DATA[8] = *(uint8_t *)(&speed)	
DATA[9]	Motor speed high byte	DATA[9] = *((uint8_t *)(&speed)+1)	
DATA[10]	Encoder data low byte	DATA[10] = *(uint8_t *)(&encoder)	
DATA[11]	Encoder data high byte	DATA[11] = *((uint8_t *)(&encoder)+1)	
DATA[12]	Data check byte	DATA[5]~DATA[11]byte checksum	

(24)Single position closed-loop control command 2 (14byte)

The computer host sends this command to control the position of the motor (single turn Angle).

1. Control value spinDirection sets the direction of motor rotation as uint8_t type, 0x00 represents clockwise, 0x01 represents counterclockwise.

2. The control value angleControl is uint16_t, with a range of 0~35999 and a corresponding actual position of 0.01degree/LSB, namely the actual Angle range of 0°~359.99°.

3. The control value maxSpeed limits the maximum speed of motor rotation, which is uint32_t type, corresponding to the actual speed of 0.01dps/LSB, i.e. 36000 represents 360dps.

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E
DATA[1]	Command byte	0XA6
DATA[2]	ID byte	0x01~0x20
DATA[3]	Data length byte	0x08
DATA[4]	Frame header check byte	DATA[0]~DATA[3]byte checksum
DATA[5]	Rotation direction byte	DATA[5] = spinDirection
DATA[6]	Position control low byte 1	DATA[6] = *(uint8_t *)(&angleControl)
DATA[7]	Position controlhigh byte 2	DATA[7] = *((uint8_t *)(&angleControl)+1)
DATA[8]	NULL	0X00
DATA[9]	Speed limit low byte 1	DATA[8] = *(uint8_t *)(&maxSpeed)
DATA[10]	Speed limit byte 2	DATA[9] = *((uint8_t *)(&maxSpeed)+1)
DATA[11]	Speed limit byte 3	DATA[10] = *((uint8_t *)(&maxSpeed)+2)
DATA[12]	Speed limit byte 4	DATA[11] = *((uint8_t *)(&maxSpeed)+3)
DATA[13]	Data check byte	DATA[5]~DATA[12]byte checksum

Remarks:

1. In this control mode, the maximum Acceleration of the motor is limited by the value of Max Acceleration in the LK-Motor Tool.

2. In this control mode, the maximum Torque Current of the motor is limited by Max Torque Current value in the LK-Motor Tool.

Driver respond(13byte)

The motor replies to the host after receiving the command, and the frame data contains the

following data.

1.Motor temperature (int8_t type, 1℃/LSB).

2.Torque current IQ of the motor (int16_t type, range -2048~2048, corresponding to the actual torque current range -33A ~33A). Motor power output value (int16_t type, range -1000~1000)

3.Motor speed (int16_t type, 1dps/LSB).

4.Encoder position value (uint16_t type, 14bit encoder value range 0~16383).

Data Field	Instructions	Data	
DATA[0]	Head byte	0x3E	
DATA[1]	Command byte	0xA6	
DATA[2]	ID byte	0x01~0x20	
DATA[3]	Data length byte	0x07	
DATA[4]	Frame header check byte	DATA[0]~DATA[3]byte checksum	
DATA[5]	Motor temperature	DATA[5] = *(uint8_t *)(&temperature)	
DATA[6]	Torque current low byte	DATA[6] = *(uint8_t *)(&iq)	MG;MF
	Output power low byte	DATA[6] = *(uint8_t *)(&power)	MS
DATA[7]	Torque current high byte	DATA[7] = *((uint8_t *)(&iq)+1)	MG;MF
	Output power high byte	DATA[7] = *((uint8_t *)(&power)+1)	MS
DATA[8]	Motor speed low byte	DATA[8] = *(uint8_t *)(&speed)	
DATA[9]	Motor speed high byte	DATA[9] = *((uint8_t *)(&speed)+1)	
DATA[10]	Encoder data low byte	DATA[10] = *(uint8_t *)(&encoder)	
DATA[11]	Encoder data high byte	DATA[11] = *((uint8_t *)(&encoder)+1)	
DATA[12]	Data check byte	DATA[5]~DATA[11]byte checksum	

(25) Incremental closed-loop control command 1（10byte）

The host sends this command to control the incremental position of the motor

1.The control value of angle Increment is int32_t type, and the corresponding actual position is 0.01degree / LSB, 36000 represents 360 °. The direction of motor rotation is determined by the sign of this parameter.

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E
DATA[1]	Command byte	0xA7
DATA[2]	ID byte	0x01~0x20
DATA[3]	Data length byte	0x03
DATA[4]	Frame header check byte	DATA[0]~DATA[3]byte checksum
DATA[5]	incremental position control low byte 1	DATA[5] = *(uint8_t *)(&angle Increment)
DATA[6]	incremental position control low byte 2	DATA[6] = *((uint8_t *)(&angle Increment)+1)
DATA[7]	incremental position control low byte 3	DATA[7] = *((uint8_t *)(&angle Increment)+2)
DATA[8]	incremental position control low byte 4	DATA[8] = *((uint8_t *)(&angle Increment)+3)
DATA[9]	Data check byte	DATA[5]~DATA[8] byte checksum

Remarks:

1.The maximum Speed of the motor under this command is limited by the Max Speed value in

the LK-Motor Tool.

2.In this control mode, the maximum Acceleration of the motor is limited by the value of Max Acceleration in the LK-Motor Tool.

3.In this control mode, the maximum Torque Current of the motor is limited by Max Torque Current value in the LK-Motor Tool.

Driver respond(13byte)

The motor replies to the host after receiving the command, and the frame data contains the following data.

1.Motor temperature (int8_t type, 1℃/LSB).

2.Torque current IQ of the motor (int16_t type, range -2048~2048, corresponding to the actual torque current range -33A ~33A). Motor power output value (int16_t type, range -1000~1000)

3.Motor speed (int16_t type, 1dps/LSB).

4.Encoder position value (uint16_t type, 14bit encoder value range 0~16383)

Data Field	Instructions	Data	
DATA[0]	Head byte	0x3E	
DATA[1]	Command byte	0xA7	
DATA[2]	ID byte	0x01~0x20	
DATA[3]	Data length byte	0x07	
DATA[4]	Frame header check byte	DATA[0]to DATA[3] check sum	
DATA[5]	Motor temperature	DATA[5] = *(uint8_t *)(&temperature)	
DATA[6]	Torque current low byte	DATA[6] = *(uint8_t *)(&iq)	MF MG
	Output power low byte	DATA[6] = *(uint8_t *)(&power)	MS
DATA[7]	Torque current high byte	DATA[7] = *((uint8_t *)(&iq)+1)	MF MG
	Output power high byte	DATA[7] = *((uint8_t *)(&power)+1)	MS
DATA[8]	motor speed low byte	DATA[8] = *(uint8_t *)(&speed)	
DATA[9]	motor speed high byte	DATA[9] = *((uint8_t *)(&speed)+1)	
DATA[10]	Encoder position low byte	DATA[10] = *(uint8_t *)(&encoder)	
DATA[11]	Encoder position high byte	DATA[11] = *((uint8_t *)(&encoder)+1)	
DATA[12]	Data check byte	DATA[5]~DATA[11]check sum	

(26) Incremental closed-loop control command 2（14byte）

The host sends this command to control the incremental position of the motor

1.The control value of angle Increment is int32_t type, and the corresponding actual position is 0.01degree / LSB, 36000 represents 360 °. The direction of motor rotation is determined by the sign of this parameter.

2. The control value of maxSpeed limits the maximum speed of the motor. It is uint32_t type, which corresponds to the actual speed of 0.01dps / LSB, 36000 represents 360dps.

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E
DATA[1]	Command byte	0xA8
DATA[2]	ID byte	0x01~0x20

DATA[3]	Data length byte	0x08
DATA[4]	Frame header check byte	DATA[0]~DATA[3] byte checksum
DATA[5]	incremental position control low byte 1	DATA[5] = *(uint8_t *)(&angleIncrement)
DATA[6]	incremental position control byte 2	DATA[6] = *((uint8_t *)(&angleIncrement)+1)
DATA[7]	incremental position control byte 3	DATA[7] = *((uint8_t *)(&angleIncrement)+2)
DATA[8]	incremental position control high byte 4	DATA[8] = *((uint8_t *)(&angleIncrement)+3)
DATA[9]	speed limited low byte1	DATA[9] = *(uint8_t *)(&maxSpeed)
DATA[10]	speed limited byte 2	DATA[10] = *((uint8_t *)(&maxSpeed)+1)
DATA[11]	speed limited byte 3	DATA[11] = *((uint8_t *)(&maxSpeed)+2)
DATA[12]	speed limited high byte 4	DATA[12] = *((uint8_t *)(&maxSpeed)+3)
DATA[13]	Data check byte	DATA[5]~DATA[12]byte check sum

Remarks:

1.In this control mode, the maximum Acceleration of the motor is limited by the value of Max Acceleration in the LK-Motor Tool.

2.In this control mode, the maximum Torque Current of the motor is limited by Max Torque Current value in the LK-Motor Tool.

Driver respond(13byte)

The motor replies to the host after receiving the command, and the frame data contains the following data.

1.Motor temperature (int8_t type, 1℃/LSB).

2.Torque current IQ of the motor (int16_t type, range -2048~2048, corresponding to the actual torque current range -33A ~33A). Motor power output value (int16_t type, range -1000~1000)

3.Motor speed (int16_t type, 1dps/LSB).

4.Encoder position value (uint16_t type, 14bit encoder value range 0~16383)

Data Field	Instructions	Data	
DATA[0]	Head byte	0x3E	
DATA[1]	Command byte	0xA8	
DATA[2]	ID byte	0x01~0x20	
DATA[3]	Data length byte	0x08	
DATA[4]	Frame header check byte	DATA[0]~DATA[3]byte checksum	
DATA[5]	Motor temperature	DATA[5] = *(uint8_t *)(&temperature)	
DATA[6]	Torque current low byte	DATA[6] = *(uint8_t *)(&iq)	MF MG
	output power low byte	DATA[6] = *(uint8_t *)(&power)	MS
DATA[7]	Torque current high byte	DATA[7] = *((uint8_t *)(&iq)+1)	MF MG
	output power high byte	DATA[7] = *((uint8_t *)(&power)+1)	MS
DATA[8]	speed limited low byte1	DATA[8] = *(uint8_t *)(&speed)	
DATA[9]	speed limited byte 2	DATA[9] = *((uint8_t *)(&speed)+1)	
DATA[10]	speed limited byte 3	DATA[10] = *(uint8_t *)(&encoder)	
DATA[11]	speed limited high byte 4	DATA[11] = *((uint8_t *)(&encoder)+1)	
DATA[12]	Data check byte	DATA[5]~DATA[11] check sum	

(27)Read driver and motor model command (5byte)

This command is used to read the driver model, motor model, hardware version number, and firmware version number.

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E
DATA[1]	Command byte	0x12
DATA[2]	ID byte	0x01~0x20
DATA[3]	Data length byte	0x00
DATA[4]	Frame header check byte	DATA[0]~DATA[3]byte checksum

For example, the host sends the following command to the 1# driver (HEX)

3E 12 01 00 45

Drive respond(48byte)

Data Field	Instructions	Data
DATA[0]	Head byte	0x3E
DATA[1]	Command byte	0x12
DATA[2]	ID byte	0x01~0x20
DATA[3]	Data length byte	0x2A
DATA[4]	Frame header check byte	DATA[0]~DATA[3]byte checksum
DATA[5~46]	Drive device information	productInfo structural body
DATA[47]	Data check byte	DATA[5]~DATA[46]byte checksum

ProductInfo structure of driver device information is as follows.

```
Struct productInfo
{
Uint8_t driverName [20]. Driver name
Uint8_t motorName [20]. Name of motor
Uint8_t hardwareVersion; // drive the hardware version
Uint8_t firmwareVersion; // firmware version
};
```

Where, the driver hardwareVersion shown in the LK-Motor Tool = hardwareVersion/10.0f, and the firmwareVersion = firmwareVersion/10.0f