LK-TECH Servo motor Protocol（CAN）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.

LK-TECH is the trademark of Shanghai Lingkong Technology Co.,LTD and its related companies.

All copyright of products and handbooks are reserved by LK-TECH. Reproduction in any form shall not be allowed without permission. Regarding the disclaimer the final interpretation right, all belongs to LK-TECH.

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

CONTENTS

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

Bus interface:CAN

Baud rate: 1Mbps

Identifier: 0x140 + ID (1 ~ 32)

Frame format: Data

Frame type: standard frame

Data bit:8

2. Single motor command list

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.		0x95
12.	Read motor status 1 and error flag command	0x9A
13.	Clear motor error flag command	0x9B
14.	Read motor status 2 command	0x9C
15.	Read motor status 3 command	0x9D
16.	Motor shutdown command	0x80
17.	Motor stop command	0x81
18.	Motor operation command	0x88
19.	Torque closed loop control command	0xA1
20.	Speed closed loop control command	0xA2
21.	Position closed loop control command 1	0xA3
22.	Position closed loop control command 2	0xA4
23.	Position closed loop control command 3	0xA5
24.	Position closed loop control command 4	0xA6
25.	Position closed-loop control command 5	0xA7
26.	Position closed-loop control command 6	0xA8

3. Single motor command description

(1) Read PID parameter command (1 frame)

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

Data Field	Instructions	Data

DATA[0]	Command byte	0x30
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	NULL	0x00
DATA[4]	NULL	0x00
DATA[4]	NULL	0x00
DATA[4]	NULL	0x00

Driver respond (1 frame)

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

Data Field	Instructions	Data
DATA[0]	Command byte	0x30
DATA[1]	NULL	0x00
DATA[2]	P parameter of position loop	DATA[2] = anglePidKp
DATA[3]	I parameter of position loop	DATA[3] = anglePidKi
DATA[4]	P parameter of speed loop	DATA[4] = speedPidKp
DATA[5]	I parameter of speed loop	DATA[5] = speedPidKi
DATA[6]	P parameter of torque loop	DATA[6] = iqPidKp
DATA[7]	I parameter of torque loop	DATA[7] = iqPidKi

(2) Write PID parameter to RAM command (1 frame)

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]	Command byte	0x31
DATA[1]	NULL	0x00
DATA[2]	P parameter of position loop	DATA[2] = anglePidKp
DATA[3]	I parameter of position loop	DATA[3] = anglePidKi
DATA[4]	P parameter of speed loop	DATA[4] = speedPidKp
DATA[5]	I parameter of speed loop	DATA[5] = speedPidKi
DATA[6]	P parameter of torque loop	DATA[6] = iqPidKp
DATA[7]	I parameter of torque loop	DATA[7] = iqPidKi

Driver respond (1 frame)

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

(3) Write PID parameter to ROM command (1 frame)

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]	Command byte	0x32
DATA[1]	NULL	0x00
DATA[2]	P parameter of position loop	DATA[2] = anglePidKp
DATA[3]	I parameter of position loop	DATA[3] = anglePidKi

DATA[4]	P parameter of speed loop	DATA[4] = speedPidKp
DATA[5]	I parameter of speed loop	DATA[5] = speedPidKi
DATA[6]	P parameter of torque loop	DATA[6] = iqPidKp
DATA[7]	I parameter of torque loop	DATA[7] = iqPidKi

Driver respond (1 frame)

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

(4) Read acceleration command (1 frame)

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

Data Field	Instructions	Data
DATA[0]	Command byte	0x33
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	NULL	0x00
DATA[7]	NULL	0x00

Driver respond(1 frame)

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]	Command byte	0x33
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	Acceleration low byte 1	DATA[4] = *(uint8_t *)(&Accel)
DATA[5]	Acceleration byte 2	DATA[5] = *((uint8_t *)(&Accel)+1)
DATA[6]	Acceleration byte 3	DATA[6] = *((uint8_t *)(&Accel)+2)
DATA[7]	Acceleration byte 4	DATA[7] = *((uint8_t *)(&Accel)+3)

(5) Write acceleration to RAM command (1 frame)

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]	Command byte	0x34
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	Acceleration low byte 1	DATA[4] = *(uint8_t *)(&Accel)
DATA[5]	Acceleration byte 2	DATA[5] = *((uint8_t *)(&Accel)+1)
DATA[6]	Acceleration byte 3	DATA[6] = *((uint8_t *)(&Accel)+2)

DATA[7]	Acceleration byte 4	DATA[7] = *((uint8_t *)(&Accel)+3)

Driver respond(1 frame)

The drive reply data is consistent with the received command parameters

(6) Read encoder command (1 frame)

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

Data Field	Instructions	Data
DATA[0]	Command byte	0x90
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	NULL	0x00
DATA[7]	NULL	0x00

Driver respond(1 frame)

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]	Command byte	0x90
DATA[1]	NULL	0x00
DATA[2]	Encoder data in low bytes	DATA[2] = *(uint8_t *)(&encoder)
DATA[3]	Encoder data in high bytes	DATA[3] = *((uint8_t *)(&encoder)+1)
DATA[4]	Encoder original position low byte	DATA[4] = *(uint8_t *)(&encoderRaw)
DATA[5]	Encoder original position high byte	DATA[5] = *((uint8_t *)(&encoderRaw)+1)
DATA[6]	Encoder zero low byte	DATA[6] = *(uint8_t *)(&encoderOffset)
DATA[7]	Encoder zero high byte	DATA[7] = *((uint8_t *)(&encoderOffset)+1)

(7) Write encoder value as motor zero point command (1 frame)

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]	Command byte	0x91
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	Encoder zero low byte	DATA[6] = *(uint8_t *)(&encoderOffset)

DATA[7]	Encoder zero high byte	DATA[7] = *((uint8_t *)(&encoderOffset)+1)

Driver respond(1 frame)

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 (1 frame)

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

Note:

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]	Command byte	0x19
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	NULL	0x00
DATA[7]	NULL	0x00

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

3E 19 01 00 58

Driver respond(1 frame)

The motor replies to the computer host after receiving the command, and EncoderOffset in the data is the 0 offset value set.

Data Field	Instructions	Data
DATA[0]	Command byte	0x19
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	Encoder zero low byte	DATA[6] = *(uint8_t *)(&encoderOffset)
DATA[7]	Encoder zero high byte	DATA[7] = *((uint8_t *)(&encoderOffset)+1)

(9) Read multi-loop Angle command (1 frame)

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

Data Field	Instructions	Data
DATA[0]	Command byte	0x92
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	NULL	0x00

DATA[5]	NULL	0x00
DATA[6]	NULL	0x00
DATA[7]	NULL	0x00

Drive respond(1 frame)

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]	Command byte	0x92
DATA[1]	Angle low byte 1	DATA[1] = *(uint8_t *)(&motorAngle)
DATA[2]	Angle byte 2	DATA[2] = *((uint8_t *)(& motorAngle)+1)
DATA[3]	Angle byte 3	DATA[3] = *((uint8_t *)(& motorAngle)+2)
DATA[4]	Angle byte 4	DATA[4] = *((uint8_t *)(& motorAngle)+3)
DATA[5]	Angle byte 5	DATA[5] = *((uint8_t *)(& motorAngle)+4)
DATA[6]	Angle byte 6	DATA[6] = *((uint8_t *)(& motorAngle)+5)
DATA[7]	Angle byte 7	DATA[7] = *((uint8_t *)(& motorAngle)+6)

(10) Read single-loop Angle command (1 frame)

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

Data Field	Instructions	Data
DATA[0]	Command byte	0x94
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	NULL	0x00
DATA[7]	NULL	0x00

Driver respond(1 frame)

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]	Command byte	0x94
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	Single loop Angle low byte 1	DATA[4] = *(uint8_t *)(& circleAngle)
DATA[5]	Single loop Angle byte 2	DATA[5] = *((uint8_t *)(& circleAngle)+1)
DATA[6]	Single loop Angle byte 3	DATA[6] = *((uint8_t *)(& circleAngle)+2)

DATA[7]	Single loop Angle high byte4	DATA[7] = *((uint8_t *)(& circleAngle)+1)

(11) The command to clear the motor angle (1 frame) . Not implemented yet

This command clears the multi turn and single turn angle data of the motor, and sets the current position as the zero point of the motor, which is invalid after power failure.

Note: this command will clear the control command data of all position rings at the same time.

Data Field	Instructions	Data
DATA[0]	Command byte	0x95
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	NULL	0x00
DATA[7]	NULL	0x00

Driver respond(1 frame)

The drive reply data is consistent with the received command parameters

(12) Read motor state 1 and error flag command (1 frame)

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

Data Field	Instructions	Data
DATA[0]	Command byte	0x9A
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	NULL	0x00
DATA[7]	NULL	0x00

Driver respond(1 frame)

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]	Command byte	0x9A
DATA[1]	Motor temperature	DATA[1] = *(uint8_t *)(&temperature)
DATA[2]	NULL	0x00
DATA[3]	voltage low byte	DATA[3] = *(uint8_t *)(&voltage)
DATA[4]	voltage high byte	DATA[4] = *((uint8_t *)(& voltage)+1)
DATA[5]	NULL	0x00
DATA[6]	NULL	0x00

DATA[7]	Error status byte	DATA[7]=errorState

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		

(13) Clear motor error mark command (1 frame)

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

Data Field	Instructions	Data
DATA[0]	Command byte	0x9B
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	NULL	0x00
DATA[7]	NULL	0x00

Driver respond(1 frame)

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]	Command byte	0x9B
DATA[1]	Motor temperature	0x9B
DATA[2]	NULL	0x01~0x20
DATA[3]	voltage low byte	0x07
DATA[4]	voltage high byte	From DATA[0] to DATA[3]checksum
DATA[5]	NULL	DATA[5] = *(uint8_t *)(&temperature)
DATA[6]	NULL	0x00
DATA[7]	Error status byte	DATA[7] = *(uint8_t *)(&voltage)

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.

(14) Read motor state 2 command (1 frame)

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

Data Field	Instructions	Data
DATA[0]	Command byte	0x9C
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	NULL	0x00
DATA[7]	NULL	0x00

Driver respond(1 frame)

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]	Command byte	0x9C
DATA[1]	Motor temperature	DATA[1] = *(uint8_t *)(&temperature)
DATA[2]	Torque current low byte	DATA[2] = *(uint8_t *)(&iq)
DATA[3]	Torque current high byte	DATA[3] = *((uint8_t *)(&iq)+1)
DATA[4]	Motor speed low bytes	DATA[4] = *(uint8_t *)(&speed)
DATA[5]	Motor speed high bytes	DATA[5] = *((uint8_t *)(&speed)+1)
DATA[6]	Encoder position low byte	DATA[6] = *(uint8_t *)(&encoder)
DATA[7]	Encoder position high byte	DATA[7] = *((uint8_t *)(&encoder)+1)

(15) Read motor state 3 command (1 frame)

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

Data Field	Instructions	Data
DATA[0]	Command byte	0x9D
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	NULL	0x00
DATA[7]	NULL	0x00

Driver respond(1 frame)

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]	Command byte	0x9D
DATA[1]	Motor temperature	DATA[1] = *(uint8_t *)(&temperature)
DATA[2]	A phase current low byte	DATA[2] = *(uint8_t *)(&iA)
DATA[3]	A phase current high byte	DATA[3] = *((uint8_t *)(& iA)+1)
DATA[4]	B phase current low byte	DATA[4] = *(uint8_t *)(&iB)
DATA[5]	B phase current high byte	DATA[5] = *((uint8_t *)(& iB)+1)
DATA[6]	C phase current low byte	DATA[6] = *(uint8_t *)(&iC)
DATA[7]	C phase current high byte	DATA[7] = *((uint8_t *)(& iC)+1)

(16) Motor shutdown command (1 frame)

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

Data Field	Instructions	Data
DATA[0]	Command byte	0x80
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	NULL	0x00
DATA[7]	NULL	0x00

Drive respond(1 frame)

It is same as computer host send.

(17) Motor stop command (1 frame)

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

Data Field	Instructions	Data
DATA[0]	Command byte	0x81
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	NULL	0x00
DATA[7]	NULL	0x00

Driver respond(1 frame)

It is same as computer host send.

(18) Motor operation command (1 frame)

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

Data Field	Instructions	Data
DATA[0]	Command byte	0x88
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	NULL	0x00
DATA[5]	NULL	0x00
DATA[6]	NULL	0x00
DATA[7]	NULL	0x00

Drive respond(1 frame)

It is same as computer host send.

(19) Torque open loop control command （1 frame）.The 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]	Command byte	0xA0
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	Output power control value low byte	DATA[5] = *(uint8_t *)(&powerControl)
DATA[5]	Output power control value high byte	DATA[6] = *((uint8_t *)(&powerControl)+1)
DATA[6]	NULL	0x00
DATA[7]	NULL	0x00

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(1 frame)

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]	Command byte	0xA0
DATA[1]	Motor temperature	DATA[1] = *(uint8_t *)(&temperature)

DATA[2]	Output power low byte	DATA[2] = *(uint8_t *)(& power)
DATA[3]	Output power high byte	DATA[3] = *((uint8_t *)(&power)+1)
DATA[4]	motor speed low byte	DATA[4] = *(uint8_t *)(&speed)
DATA[5]	motor speed high byte	DATA[5] = *((uint8_t *)(&speed)+1)
DATA[6]	encoder position low byte	DATA[6] = *(uint8_t *)(&encoder)
DATA[7]	encoder position high byte	DATA[7] = *((uint8_t *)(&encoder)+1)

(20) Torque closed-loop control command (1 frame). 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]	Command byte	0xA1
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	Torque current control value low byte	DATA[4] = *(uint8_t *)(&iqControl)
DATA[5]	Torque current control value high byte	DATA[5] = *((uint8_t *)(&iqControl)+1)
DATA[6]	NULL	0x00
DATA[7]	NULL	0x00

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(1 frame)

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]	Command byte	0xA1
DATA[1]	Motor temperature	DATA[1] = *(uint8_t *)(&temperature)
DATA[2]	Torque current low byte	DATA[2] = *(uint8_t *)(&iq)
DATA[3]	Torque current high byte	DATA[3] = *((uint8_t *)(&iq)+1)
DATA[4]	Motor speed low bytes	DATA[4] = *(uint8_t *)(&speed)
DATA[5]	Motor speed high bytes	DATA[5] = *((uint8_t *)(&speed)+1)
DATA[6]	Encoder data in low bytes	DATA[6] = *(uint8_t *)(&encoder)
DATA[7]	Encoder data in high bytes	DATA[7] = *((uint8_t *)(&encoder)+1)

(21) Speed closed-loop control command (1 frame)

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]	Command byte	0xA2
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	Motor speed low byte	DATA[4] = *(uint8_t *)(&speedControl)
DATA[5]	Motor speed	DATA[5] = *((uint8_t *)(&speedControl)+1)
DATA[6]	Motor speed	DATA[6] = *((uint8_t *)(&speedControl)+2)
DATA[7]	Motor speed high byte	DATA[7] = *((uint8_t *)(&speedControl)+3)

Remark:

1. The max torque current in this command (MF/MG) limited by max torque current in the LK-Motor Tool.
2. The max acceleration in this command limited by max acceleration in the LK-Motor Tool.

Drive respond(1 frame)

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]	Command byte	0xA2
DATA[1]	Motor temperature	DATA[1] = *(uint8_t *)(&temperature)
DATA[2]	Torque current low byte	DATA[2] = *(uint8_t *)(&iq)
DATA[3]	Torque current high byte	DATA[3] = *((uint8_t *)(&iq)+1)
DATA[4]	Motor speed low byte	DATA[4] = *(uint8_t *)(&speed)
DATA[5]	Motor speed high byte	DATA[5] = *((uint8_t *)(&speed)+1)
DATA[6]	Encoder data low byte	DATA[6] = *(uint8_t *)(&encoder)
DATA[7]	Encoder data high byte	DATA[7] = *((uint8_t *)(&encoder)+1)

(22) Multi position closed-loop control command 1 (1 frame)

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]	Command byte	0xA3
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00

DATA[3]	NULL	0x00
DATA[4]	Position control low byte	DATA[4] = *(uint8_t *)(&angleControl)
DATA[5]	Position control byte	DATA[5] = *((uint8_t *)(&angleControl)+1)
DATA[6]	Position control byte	DATA[6] = *((uint8_t *)(&angleControl)+2)
DATA[7]	Position control high byte	DATA[7] = *((uint8_t *)(&angleControl)+3)

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(1 frame)

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).

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]	Command byte	0xA3
DATA[1]	Motor temperature	DATA[1] = *(uint8_t *)(&temperature)
DATA[2]	Torque current low byte	DATA[2] = *(uint8_t *)(&iq)
DATA[3]	Torque current high byte	DATA[3] = *((uint8_t *)(&iq)+1)
DATA[4]	Motor speed low byte	DATA[4] = *(uint8_t *)(&speed)
DATA[5]	Motor speed high byte	DATA[5] = *((uint8_t *)(&speed)+1)
DATA[6]	Encoder data low byte	DATA[6] = *(uint8_t *)(&encoder)
DATA[7]	Encoder data high byte	DATA[7] = *((uint8_t *)(&encoder)+1)

(23) Multi position closed-loop control command 2 (1 frame)

The host computer sends the command to control the position of the motor (multi-turn Angle), the control value angleControl is int32_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]	Command byte	0xA4
DATA[1]	NULL	0x00

DATA[2]	Speed limit low byte	DATA[2] = *(uint8_t *)(&maxSpeed)
DATA[3]	Speed limit high byte	DATA[3] = *((uint8_t *)(&maxSpeed)+1)
DATA[4]	Position control low byte	DATA[4] = *(uint8_t *)(&angleControl)
DATA[5]	Position control	DATA[5] = *((uint8_t *)(&angleControl)+1)
DATA[6]	Position control	DATA[6] = *((uint8_t *)(&angleControl)+2)
DATA[7]	Position control high byte	DATA[7] = *((uint8_t *)(&angleControl)+3)

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(1 frame)

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).

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]	Command byte	0xA4
DATA[1]	Motor temperature	DATA[1] = *(uint8_t *)(&temperature)
DATA[2]	Torque current low byte	DATA[2] = *(uint8_t *)(&iq)
DATA[3]	Torque current high byte	DATA[3] = *((uint8_t *)(&iq)+1)
DATA[4]	Motor speed low byte	DATA[4] = *(uint8_t *)(&speed)
DATA[5]	Motor speed high byte	DATA[5] = *((uint8_t *)(&speed)+1)
DATA[6]	Encoder data low byte	DATA[6] = *(uint8_t *)(&encoder)
DATA[7]	Encoder data high byte	DATA[7] = *((uint8_t *)(&encoder)+1)

(24) Single position closed-loop control command 1 (1 frame)

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

1. 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°.

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

Data Field	Instructions	Data
DATA[0]	Command byte	0xA5
DATA[1]	Rotation direction byte	DATA[1] = spinDirection
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00

DATA[4]	Position control low byte	DATA[4] = *(uint8_t *)(&angleControl)
DATA[5]	Position control high byte	DATA[5] = *((uint8_t *)(&angleControl)+1)
DATA[6]	NULL	0x00
DATA[7]	NULL	0x00

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(1 frame)

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).

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]	Command byte	0xA5
DATA[1]	Motor temperature	DATA[1] = *(uint8_t *)(&temperature)
DATA[2]	Torque current low byte	DATA[2] = *(uint8_t *)(&iq)
DATA[3]	Torque current high byte	DATA[3] = *((uint8_t *)(&iq)+1)
DATA[4]	Motor speed low byte	DATA[4] = *(uint8_t *)(&speed)
DATA[5]	Motor speed high byte	DATA[5] = *((uint8_t *)(&speed)+1)
DATA[6]	Encoder data low byte	DATA[6] = *(uint8_t *)(&encoder)
DATA[7]	Encoder data high byte	DATA[7] = *((uint8_t *)(&encoder)+1)

(25) Single position closed-loop control command 2 (1 frame)

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 uint16_t type, corresponding to the actual speed of 0.01dps/LSB.

Data Field	Instructions	Data
DATA[0]	Command byte	0xA6
DATA[1]	Rotation direction byte	DATA[1] = spinDirection
DATA[2]	Speed limit low byte 1	DATA[2] = *(uint8_t *)(&maxSpeed)
DATA[3]	Speed limit byte 2	DATA[3] = *((uint8_t *)(&maxSpeed)+1)
DATA[4]	Position control low byte	DATA[4] = *(uint8_t *)(&angleControl)

DATA[5]	Position control high byte	DATA[5] = *((uint8_t *)(&angleControl)+1)
DATA[6]	NULL	0x00
DATA[7]	NULL	0x00

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(1 frame)

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).

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]	Command byte	0xA6
DATA[1]	Motor temperature	DATA[1] = *(uint8_t *)(&temperature)
DATA[2]	Torque current low byte	DATA[2] = *(uint8_t *)(&iq)
DATA[3]	Torque current high byte	DATA[3] = *((uint8_t *)(&iq)+1)
DATA[4]	Motor speed low byte	DATA[4] = *(uint8_t *)(&speed)
DATA[5]	Motor speed high byte	DATA[5] = *((uint8_t *)(&speed)+1)
DATA[6]	Encoder data low byte	DATA[6] = *(uint8_t *)(&encoder)
DATA[7]	Encoder data high byte	DATA[7] = *((uint8_t *)(&encoder)+1)

(26) Incremental closed-loop control command 1（1 frame）

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]	Command byte	0xA7
DATA[1]	NULL	0x00
DATA[2]	NULL	0x00
DATA[3]	NULL	0x00
DATA[4]	position control low byte	DATA[4] = *(uint8_t *)(&angleControl)
DATA[5]	position control byte	DATA[5] = *((uint8_t *)(&angleControl)+1)
DATA[6]	position control byte	DATA[6] = *((uint8_t *)(&angleControl)+2)
DATA[7]	position control high byte	DATA[7] = *((uint8_t *)(&angleControl)+3)

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(1 frame)

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).

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]	Command byte	0xA7
DATA[1]	Motor temperature	DATA[1] = *(uint8_t *)(&temperature)
DATA[2]	Torque current low byte	DATA[2] = *(uint8_t *)(&iq)
DATA[3]	Torque current high byte	DATA[3] = *((uint8_t *)(&iq)+1)
DATA[4]	motor speed low byte	DATA[4] = *(uint8_t *)(&speed)
DATA[5]	motor speed high byte	DATA[5] = *((uint8_t *)(&speed)+1)
DATA[6]	Encoder position low byte	DATA[6] = *(uint8_t *)(&encoder)
DATA[7]	Encoder position high byte	DATA[7] = *((uint8_t *)(&encoder)+1)

(27) Incremental closed-loop control command 2（1 frame）

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. 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 uint16_t type, which corresponds to the actual speed of 0.01dps / LSB.

Data Field	Instructions	Data
DATA[0]	Command byte	0xA8
DATA[1]	NULL	0x00
DATA[2]	speed limited low byte	DATA[2] = *(uint8_t *)(&maxSpeed)
DATA[3]	speed limited high byte	DATA[3] = *((uint8_t *)(&maxSpeed)+1)
DATA[4]	position control low byte	DATA[4] = *(uint8_t *)(&angleControl)
DATA[5]	position control byte	DATA[5] = *((uint8_t *)(&angleControl)+1)
DATA[6]	position control byte	DATA[6] = *((uint8_t *)(&angleControl)+2)
DATA[7]	position control high byte	DATA[7] = *((uint8_t *)(&angleControl)+3)

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(1 frame)

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).

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]	Command byte	0xA8
DATA[1]	Motor temperature	DATA[1] = *(uint8_t *)(&temperature)
DATA[2]	Torque current low byte	DATA[2] = *(uint8_t *)(&iq)
DATA[3]	Torque current high byte	DATA[3] = *((uint8_t *)(&iq)+1)
DATA[4]	motor speed low byte	DATA[4] = *(uint8_t *)(&speed)
DATA[5]	motor speed high byte	DATA[5] = *((uint8_t *)(&speed)+1)
DATA[6]	Encoder position low byte	DATA[6] = *(uint8_t *)(&encoder)
DATA[7]	Encoder position high byte	DATA[7] = *((uint8_t *)(&encoder)+1)

4. Multi- motor command data

The multi motor command needs to be opened in the setting software, and the multi motor command and single motor command cannot be used at the same time.

Multi motor torque closed loop control command (1 frame).

The message format used to send commands to multiple motors at the same time is as follows:

Identifier: 0x280

Frame format: DATA

Frame type: standard frame

Data bit:8

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).

The motor ID should be set to #1~#4, and cannot be repeated, corresponding to the 4 torque currents in the frame data

Data Field	Instructions	Data
DATA[0]	Torque current 1 control value low byte	DATA[0] = *(uint8_t *)(&iqControl_1)
DATA[1]	Torque current 1 control value high byte	DATA[1] = *((uint8_t *)(&iqControl_1)+1)
DATA[2]	Torque current 2 control value low byte	DATA[2] = *(uint8_t *)(&iqControl_2)
DATA[3]	Torque current 2 control value high byte	DATA[3] = *((uint8_t *)(&iqControl_2)+1)
DATA[4]	Torque current 3 control value low byte	DATA[4] = *(uint8_t *)(&iqControl_3)

DATA[5]	Torque current 3 control value high byte	DATA[5] = *((uint8_t *)(&iqControl_3)+1)
DATA[6]	Torque current 4 control value low byte	DATA[6] = *(uint8_t *)(&iqControl_4)
DATA[7]	Torque current 4 control value high byte	DATA[7] = *((uint8_t *)(&iqControl_4)+1)

Drive respond(1 frame)

The message format of each motor reply command is as follows:

Identifier: 0x140 + ID(1~4)

Frame format: DATA

Frame type: standard frame

Data bit:8