RMD-L servo motor manual V1.2

Disclaimer

Thank you for purchasing GYEMS RMD-L series servo motor system. Please read this statement carefully before use. Once used, it is considered as acceptance and acceptance of the entire contents of this statement. Please strictly follow the manual, product description and related laws, regulations, policies and guidelines to install and use the product. In the process of using the product, the user is committed to being responsible for his or her actions and all consequences arising therefrom. GYEMS will not be liable for any loss caused by improper use, installation or modification by the user. GYEMS is a trademark of Guangyu Electromechanical (Shanghai) Co., Ltd. and its affiliates. Product names, brands, etc. appearing in this document are trademarks or registered trademarks of their respective companies. This product and manual are copyrighted by GYEMS. Reproduction may not be allowed in any form without permission. The final interpretation of the disclaimer is owned by GYEMS .

Introduction

The RMD-L servo motor system used a 32-bit high-performance MCU、High bandwidth op amp 、Low internal resistance flat MOSFET and combined with an optimized version of the FOC control technology, equipped with a high-performance brushless motor of the DM series, designed for high-precision, high-response, high-torque applications. The integrated design of the motor and the driver facilitates ,easily apply for system integration. The driver integrates a high-precision absolute encoder with an easy-to-use closed-loop control algorithm that greatly improves the accuracy of position ,speed feedback and torque output.

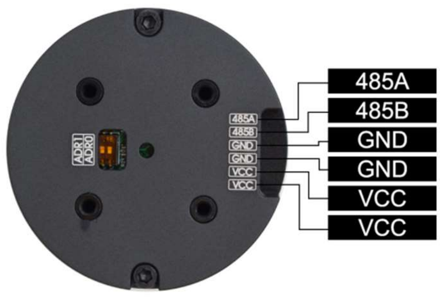
1. Electrical driver parameter

Input voltage	DRC06-S6	7.4V~24V
	DRC10-S6	7.4V~36V
Normal current	DRC06-S6	6A
	DRC10-S6	10A
Maxium current	DRC06-S6	8A （10s）
	DRC10-S6	15A （10s）
PWM Frequency	24KHz	
Torque loop control frequency	24KHz	
Torque loop control bandwidth	0.4KHz~2.8KHz（Depending on differ motor and torque）	
Speed loop control frequency	8KHz	
Position loop control frequency	8KHz	
Encoder	14bit	
Bus type	RS485（Non-isolation）	

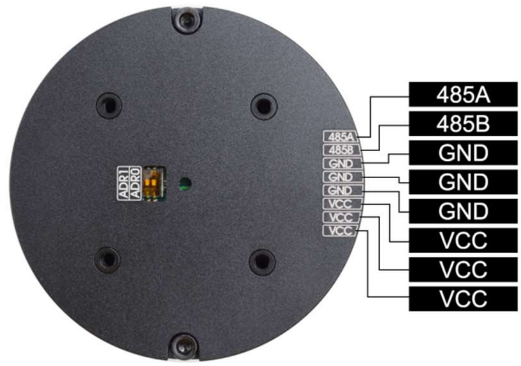
2. Motor electrical parameters

Check RMD motor data sheet to get detail motor information

3. Driver interface



40&50 MOTOR



70&90 MOTOR

Interface	Instruction
GND	Negative power supply
485B	RS485-B
485A	RS485-A
VCC	Positive power supply

4. Bus feature

Bus interface chip:MAX485

Baud rate： 9600, 19200, 57600, 115200(default)

Data bit： 8

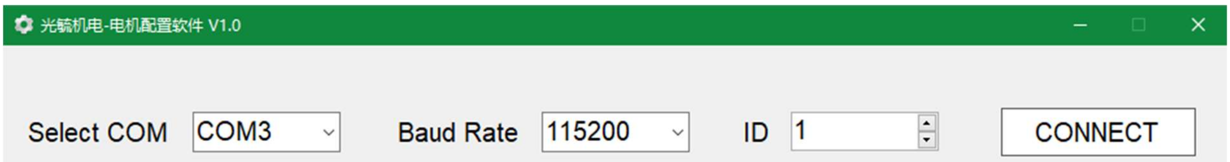
Parity bit： None

Spot bit： 1

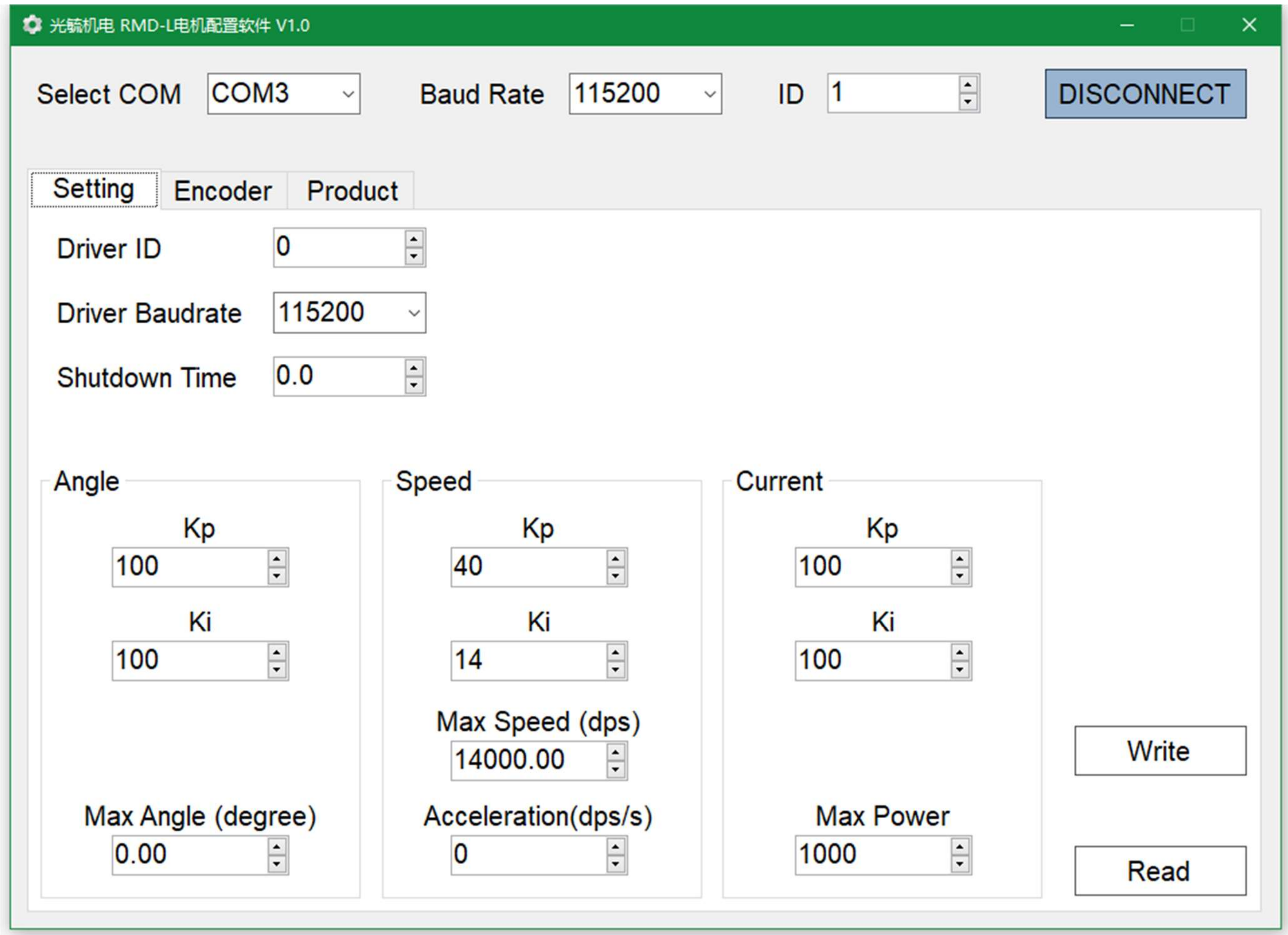
5. Setting

- **PC connection**

  The motor drive and the host computer can be connected via USB to RS485 module. The default baud rate is 115200. The default ID is generally 1 (set by the DIP switch) .Therefore, the settings before the host computer is connected as follows (where COM is selected according to the actual situation), after clicking the CONNECT button, connect the device.



- **Basic settings**, in the Settings page, click the Read button to read the motor and encoder information



- ✓ Driver ID：Set the ID number of the driver. When set to 0, the ID is selected by the dial switch, and the corresponding relationship between the two is as follows:

Switch 1	Switch 2	Switch 3	ID
OFF	OFF	OFF	#1
ON	OFF	OFF	#2
OFF	ON	OFF	#3
ON	ON	OFF	#4
OFF	OFF	ON	#5
ON	OFF	ON	#6
OFF	ON	ON	#7
ON	ON	ON	#8

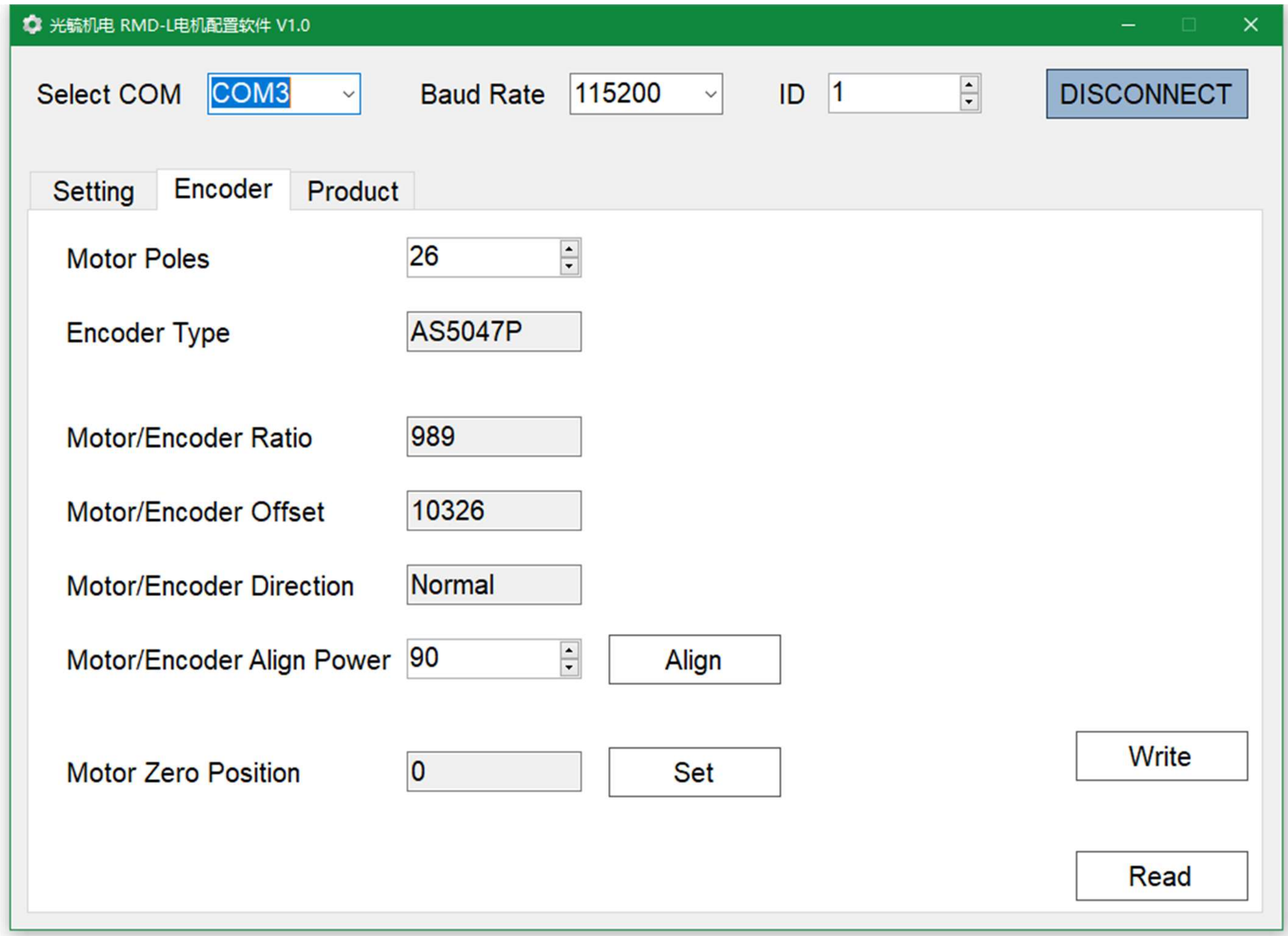
- ✓ Driver Baudrate：Set the baud rate of the drive. Note that the new baud rate needs to be powered back on after the setup is complete.
- ✓ Shutdown Time：Set the motor off time. When the control command is not received within this time, the motor will be turned off; when set to 0, the motor will never turn off. Note that the new shutdown time after the setup is complete needs to be powered back on to take effect
- ✓ Angle：Angle loop control parameters. Kp and Ki modify the PI parameter of the angle ring. Max Angle is used to limit the maximum rotation angle of the motor. For example, when set to 3600, the maximum rotation angle of the motor is ±3600°,10 turns.
- ✓ Speed：Speed loop control parameters. Kp and Ki modify the PI parameter of the speed loop. Max Speed is used to limit the maximum rotation speed of the motor. For example, when set to 720, the maximum angular velocity of the motor is ±720°/S, which is 2 turns per second.

✓ Current： Torque loop control parameters. Kp and Ki modify the PI parameter of the torque loop， Max Power is used to limit the ouput power to motor

Note:

1. Acceleration option does not take effect in the current version of the drive, the actual acceleration of the motor depends on the PI parameters, motor load and drive voltage.
2. After the parameters are modified, click the Write button to save the parameters to the driver.
3.After setting ,new parameters will be valid when power on again

➢ Encoder settings, in the Encoder page, click the Read button to read the motor and encoder information



✓ Motor Poles： Set the number of magnetic poles of the motor, usually setted before leaving the factory

✓ Encoder Type： Read-only parameter

✓ Motor/Encoder Ratio： Read-only parameter, generally around 1000, the closer to 1000, the better the calibration effect.

✓ Motor/Encoder Offset： Read-only parameter and generally has no effect on motor drive performance

✓ Motor/Encoder Direction： Read-only parameter and generally has no effect on motor drive performance.

✓ Motor/Encoder Align Power： Generally use the default parameters, when the load is large, you can increase the calibration to improve the calibration effect.

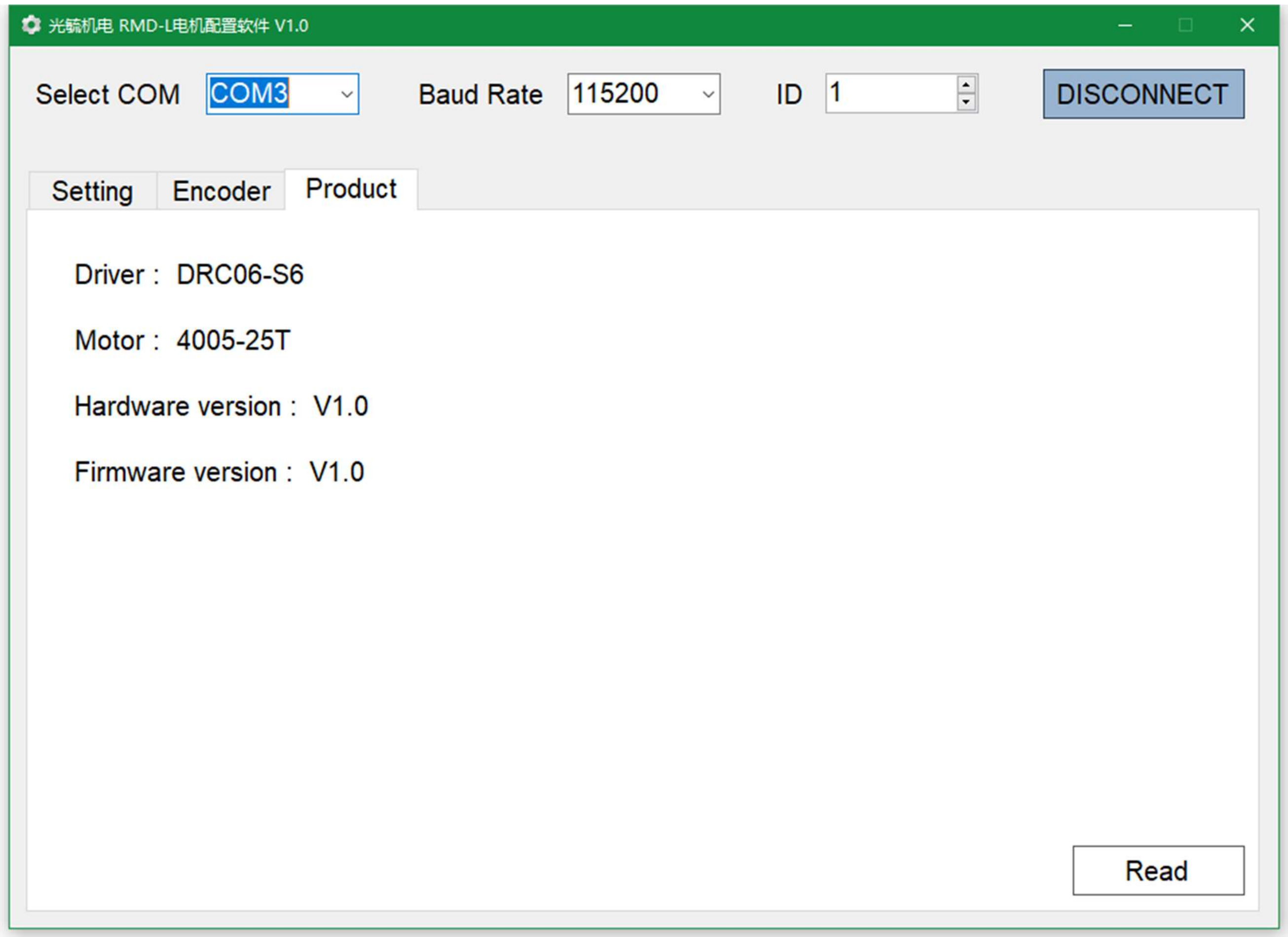
✓ Align button：Start calibration of the motor and encoder. Before this step, you need to ensure that the number of poles of the motor is set correctly and select the appropriate calibration power. After clicking the Align button, the motor will rotate back and forth to perform calibration. After the calibration is completed, the parameters will be automatically saved to the drive.

✓ Motor Zero Position：After clicking the Set button, the drive will save the current position as the starting position of the motor.

Note:

1. Suggest calibrating the motor and encoder under no-load conditions. If the motor does not rotate smoothly during the calibration rotation, check the motor fault or mechanical friction.
2. After the parameters are modified, click the Write button to save the parameters to the driver

- Product information: in the Product page, click the Read button to read the hardware and software version of the product



6. Control commands

Upto 32 drivers (depending on the bus load) can be mounted on the bus. To prevent bus collisions, each driver needs to be set with a different ID. For details, refer to the basic settings in the previous section. The master sends a control command to the driver, and the corresponding ID driver parses the data after receiving the command, selects the control mode according to the command type (angle closed loop, speed closed loop, torque loop), and sends reply command to the master after a period of time (within 0.5 ms)

7. Each control command consists of 2 parts: frame header + data, as specified below

	Data description	Data length	
Frame command	Head byte	1	Frame header recognition，0x3E
	Command byte	1	CMD
	ID byte	1	1~32
	Data length byte	1	Description of Data length 0~60
	Frame header check byte	1	Header check sum
Frame data	data	0~60	data stream attached to the command
	Data check byte	0 or 1	Data check sum

Control commands supported by the RMD-L drive motor as follows:

Motor off command	0x80
Motor stop command	0x81
Motor running command	0x88
Zero position command	0x19
Read encoder data command	0x90
Torque loop control command	0xA1
Speed loop control command	0xA2
Position loop control command1	0xA3
Position loop control command2	0xA4
Position loop control command3	0xA5
Position loop control command4	0xA6

- **Motor off command**, Turn off the motor and clear the motor running status and the previously received control commands. The total length of the command: 5byte

		Instruction	Memo
1	Head byte	0x3E	
2	Command byte	0x80	CMD
3	ID byte	0x01~0x20	#1~#32
4	Data length	0x00	
5	Head check byte	1~4byte check sum	

Eg, the host sends motor off command to 1# driver as follows (HEX)

3E 80 01 00 BF

- **Motor stop command**, stop the motor but does not clear the motor running status and the previously received control command. The total length of the command: 5byte

		Instruction	Memo
1	Head byte	0x3E	
2	Command byte	0x81	CMD
3	ID byte	0x01~0x20	#1~#32
4	Data length	0x00	
5	Head check byte	1~4 byte check sum	

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

3E 81 01 00 C0

- **Motor running command,** recover motor running from motor stop command (control mode same as before recovery), total command length: 5byte

		Instruction	Memo
1	Head byte	0x3E	
2	Command byte	0x88	CMD
3	ID byte	0x01~0x20	#1~#32
4	Data length	0x00	
5	Head check byte	1~4 byte check sum	

Eg, the host sends motor running command to 1# driver as follows (HEX)

3E 88 01 00 C7

- **Zero Position command**, set current motor position as zero position, total length of command is 5byte

  Notice：

  1. The command will be valid after power on again.
  2. The command will write the zero point to the drive FLASH memory. Multiple writes will affect the chip life. It is not recommended to use it frequently.

		Instruction	memo
1	Head byte	0x3E	
2	Command byte	0x19	CMD
3	ID byte	0x01~0x20	#1~#32
4	Data length byte	0x00	
5	Head check byte	1~4byte check sum	

Eg, the host sends motor zero position command to 1# driver as follows (HEX)

3E 19 01 00 58

- **Read the encoder command**, this command will not change the current state of the motor, the total length of the command: 5byte

		Instruction	Memo
1	Head byte	0x3E	
2	Command byte	0x88	CMD
3	ID byte	0x01~0x20	#1~#32
4	Data length	0x00	
5	Head check byte	1~4 byte check sum	

Eg: the host sends a command to read the encoder to 1# driver as follows (HEX)

3E 90 01 00 CF

Driver reply, data length: 8byte

		Instruction	memo
1	Head byte	0x3E	
2	Command byte	0x90	CMD
3	ID byte	0x01~0x20	#1~#32
4	Data length byte	0x02	
5	Head check byte	1~4byte check sum	
6	Encoder data low byte	=*(int8_t *)(&encoder)	The encoder data is 16bit shaped data, and the data range is related to the encoder accuracy, which is generally 0~16383(14bit).
7	Encoder data high byte	=*((int8_t *)(&encoder)+1)	

8	Data check byte	6~7 byte check sum	

Eg: the data that the 1# driver replies after receiving the command to read the encoder ,as follows (HEX)
3E 90 01 02 D1 CF 0F DE

- **Torque closed-loop control command,** which contains a control parameter (motor torque parameter), total command length: 8byte

		Instruction	memo
1	Head byte	0x3E	
2	Command byte	0xA1	CMD
3	ID byte	0x01~0x20	#1~#32
4	Data length byte	0x02	
5	Head check byte	1~4byte check sum	
6	Motor torque parameter low byte	=*(int8_t *)(&power)	This parameter represents the motor output torque ratio (actual torque output varies from motor to motor) which is 16-bit shaped data . Range: -5000 ~ +5000. The direction of motor rotation (torque output direction) is determined by the sign of this parameter
7	Motor torque parameter high byte	=*((int8_t *)(&power)+1)	
8	Data check byte	6~7byte check sum	

Eg: the host sends torque command to the 1# driver with the POWER value of 256 ;as follows (HEX)
3E A1 01 02 E2 00 01 01
the host sends torque command to the 1# driver with the POWER value of 512 ;as follows (HEX)
3E A1 01 02 E2 00 02 02

Driver reply, data length: 8byte

		Instruction	memo
1	Head byte	0x3E	
2	Command byte	0xA0	CMD
3	ID byte	0x01~0x20	#1~#32
4	Data length byte	0x02	
5	Head check byte	1~4 byte check sum	
6	Encoder data low byte	=*(int8_t *)(&encoder)	The encoder data is 16bit shaped data, and the data range is related to the encoder accuracy, which is generally 0~16383 (14bit).
7	Encoder data high byte	=*((int8_t *)(&encoder)+1)	
8	Data check byte	6~7byte check sum	

Eg: The command that the driver replies after receiving torque control data , as follows (HEX)
3E A1 01 02 E1 E8 03 EB

- **Speed closed-loop control**, the command contains a control parameter, this parameter defines the running speed of the motor, the total length of the command: 10byte

		Instruction	Memo
1	Head byte	0x3E	

2	Command byte	0xA2	CMD
3	ID byte	0x01~0x20	#1~#32
4	Data length byte	0x04	
5	Head check byte	1~4byte check sum	
6	Motor speed low byte	=*(int8_t *)(&speed)	The motor speed represents the angular velocity of the motor, which is 32bit shaped data. The actual speed ratio is 0.01 dps/LSB, 36000 represents 360 dps. The direction of motor rotation is determined by the sign of the speed value
7	Motor speed	=*((int8_t *)(&speed)+1)	
8	Motor speed	=*((int8_t *)(&speed)+2)	
9	Motor speed high byte	=*((int8_t *)(&speed)+3)	
10	Data check byte	6~9byte check sum	

Eg: the host sends a command with an angular velocity of 720 dps to the 1# drive as follows (HEX)
3E A2 01 04 E5 40 19 01 00 5A
the host sends a command with an angular velocity of 360 dps to the 1# drive as follows (HEX)
3E A2 01 04 E5 A0 8C 00 00 2C
the host sends a command with an angular velocity of 180 dps to the 1# drive as follows (HEX)
3E A2 01 04 E5 50 46 00 00 96
the host sends a command with an angular velocity of 90 dps to the 1# drive as follows (HEX)
3E A2 01 04 E5 28 23 00 00 4B
the host sends a command with an angular velocity of 0 dps to the 1# drive as follows (HEX)
3E A2 01 04 E5 00 00 00 00 00
the host sends a command with an angular velocity of -90 dps to the 1# drive as follows (HEX)
3E A2 01 04 E5 D8 DC FF FF B2
Driver reply, data length: 8byte

		Instruction	Memo
1	Head byte	0x3E	
2	Command byte	0xA2	CMD
3	ID byte	0x01~0x20	#1~#32
4	Data length byte	0x02	
5	Head check byte	1~4byte check sum	
6	Encoder data low byte	=*(int8_t *)(&encoder)	The encoder data is 16bit shaped data, and the data range is related to the encoder accuracy, which is generally 0~16383 (14bit).
7	Encoder data high byte	=*((int8_t *)(&encoder)+1)	
8	Data check byte	6~7byte check sum	

Eg: the command to reply after receiving the speed closed loop control data , as follows (HEX):
3E A2 01 02 E3 E8 03 EB

- **Position closed loop control 1,** the command contains a control parameter, this parameter defines the target position of the motor (multi-turn angle cumulative value), the maximum speed of motor rotation in this mode is determined by the Max Speed in the set value, the total command Length: 14byte

		Instrcution	Memo
1	Head byte	0x3E	

2	Command byte	0xA3	CMD
3	ID byte	0x01~0x20	#1~#32
4	Data length byte	0x08	
5	Head check byte	1~4byte check sum	
6	Motor position low byte	=*(int8_t *)(&angle)	The motor angle represents the angle of rotation of the motor, which is 64bit shaped data. The actual angle ratio is 0.01degree/LSB,36000 represents 360°. The direction of motor rotation is determined by the difference between the target position and the current position. Note that the maximum motor control angle limited by MAX_ANGLE in the setting tab.
7	Motor position	=*((int8_t *)(&angle)+1)	
8	Motor position	=*((int8_t *)(&angle)+2)	
9	Motor position	=*((int8_t *)(&angle)+3)	
10	Motor position	=*((int8_t *)(&angle)+4)	
11	Motor position	=*((int8_t *)(&angle)+5)	
12	Motor position	=*((int8_t *)(&angle)+6)	
13	Motor position high byte	=*((int8_t *)(&angle)+7)	
14	Data check byte	6~13byte check sum	

Eg:The host sends angle of 360°to 1# drive as follows (HEX)
3E A3 01 08 EA A0 8C 00 00 00 00 00 00 2C
The host sends angle of 180°to 1# drive as follows (HEX)
3E A3 01 08 EA 50 46 00 00 00 00 00 00 96
The host sends angle of -180°to 1# drive as follows (HEX)
3E A3 01 08 EA B0 B9 FF FF FF FF FF FF 63
The host sends angle of 90°to 1# drive as follows (HEX)
3E A3 01 08 EA 28 23 00 00 00 00 00 00 4B
The host sends angle of 0°to 1# drive as follows (HEX)
3E A3 01 08 EA 00 00 00 00 00 00 00 00 00
Driver reply, data length: 8byte

		instruction	memo
1	Head byte	0x3E	
2	Command byte	0xA3	CMD
3	ID byte	0x01~0x20	#1~#32
4	Data length byte	0x02	
5	Head check byte	1~4 byte check sum	
6	Encoder data low byte	=*(int8_t *)(&encoder)	The encoder data is 16bit shaped data, and the data range is related to the encoder accuracy, which is generally 0~16383 (14bit).
7	Encoder data high byte	=*((int8_t *)(&encoder)+1)	
8	Data check byte	6~7byte check sum	

Eg: Drive reply after receiving the position closed loop control data, as follows (HEX):
3E A3 01 02 E4 E8 03 EB

- **Position closed-loop control 2**, the command contains two control parameters, parameters respectively define the motor's target position (multi-turn angle cumulative value) and the maximum speed of motor rotation during the arrival of this target position, the total command length: 18byte

		Instruction	Memo

1	Head byte	0x3E	
2	Command byte	0xA4	CMD
3	ID byte	0x01~0x20	#1~#32
4	Data length byte	0x08	
5	Head check byte	1~4 byte check sum	
6	Motor position low byte	=*(int8_t *)(&angle)	The motor speed represents the angular velocity of the motor, which is 32bit shaped data. The actual speed ratio is 0.01 dps/LSB,36000 represents 360 dps, and the speed symbol is invalid. the difference between the target position and the current position. Note that the maximum motor control angle limited by MAX_ANGLE in the setting tab.
7	Motor position	=*((int8_t *)(&angle)+1)	
8	Motor position	=*((int8_t *)(&angle)+2)	
9	Motor position	=*((int8_t *)(&angle)+3)	
10	Motor position	=*((int8_t *)(&angle)+4)	
11	Motor position	=*((int8_t *)(&angle)+5)	
12	Motor position	=*((int8_t *)(&angle)+6)	
13	Motor position high byte	=*((int8_t *)(&angle)+7)	
14	Motor speed low byte	=*(int8_t *)(&speed)	The motor speed represents the angular velocity of the motor, which is 32bit shaped data. The actual speed ratio is 0.01 dps/LSB,36000 represents 360 dps, and the speed symbol is invalid.
15	Motor speed	=*((int8_t *)(&speed)+1)	
16	Motor speed	=*((int8_t *)(&speed)+2)	
17	Motor speed high byte	=*((int8_t *)(&speed)+3)	
18	Data check byte	6~17byte check sum	

Eg: The host sends angle of 360°& angular velocity of 90dps to 1# drive as follows (HEX)

3E A4 01 08 EF A0 8C 00 00 00 00 00 00 28 23 00 00 77

The host sends angle of 180°& angular velocity of 90dps to 1# drive as follows (HEX)

3E A4 01 08 EF 50 46 00 00 00 00 00 00 28 23 00 00 E1

The host sends angle of 90°& angular velocity of 90dps to 1# drive as follows (HEX)

3E A4 01 08 EF 28 23 00 00 00 00 00 00 28 23 00 00 96

The host sends angle of 0°& angular velocity of 90dps to 1# drive as follows (HEX)

3E A4 01 08 EF 00 00 00 00 00 00 00 00 28 23 00 00 4B

Driver reply, data length: 8byte

		Instruction	Memo
1	Head byte	0x3E	
2	Command byte	0xA4	CMD
3	ID byte	0x01~0x20	#1~#32
4	Data length byte	0x02	
5	Head check byte	1~4 byte check sum	
6	Encoder data low byte	=*(int8_t *)(&encoder)	The encoder data is 16bit shaped data, and the data range is related to the encoder accuracy, which is generally 0~16383 (14bit).
7	Encoder data high byte	=*((int8_t *)(&encoder)+1)	
8	Data check byte	6~7byte check sum	

Eg: Drive reply after receiving the position closed loop control data, as follows (HEX):
3E A4 01 02 E5 E8 03 EB

- **Position closed loop control 3**, the command contains two control parameters, respectively defining the direction of rotation of the motor and the target position (single turn angle value), the maximum speed of motor rotation in this mode is determined by the Max Speed in the set value, the command Total length: 10byte

		Instruction	Memo
1	Head byte	0x3E	
2	Command byte	0xA5	CMD
3	ID byte	0x01~0x20	#1~#32
4	Data length byte	0x08	
5	Head check byte	1~4byte check sum	
6	Direction of rotation byte	0x00 OR 0x01	0:clockwise,1 :counterclockwise
7	Motor position low byte	=*(uint8_t *)(&angle)	Motor angle represents the rotation angle of the motor, which is 32bit unsigned shaped data. The actual angle ratio is 0.01degree/LSB,data range is 0~35999,represents the angle range from 0°to 359.99°
8	Motor position	=*((uint8_t *)(&angle)+1)	
9	Motor position high byte	=*((uint8_t *)(&angle)+2)	
10	Data check byte	6~9byte check sum	

Eg :the host sends a clockwise rotation of 315° to the 1# drive as follows (HEX)
3E A5 01 04 E8 00 0C 7B 00 87
the host sends a clockwise rotation of 180° to the 1# drive as follows (HEX)
3E A5 01 04 E8 00 50 46 00 96
the host sends a clockwise rotation of 90° to the 1# drive as follows (HEX)
3E A5 01 04 E8 00 28 23 00 4B
the host sends a counterclockwise rotation of 315° to the 1# drive as follows (HEX)
3E A5 01 04 E8 01 0C 7B 00 88
the host sends a counterclockwise rotation of 180° to the 1# drive as follows (HEX)
3E A5 01 04 E8 01 50 46 00 97
the host sends a counterclockwise rotation of 90° to the 1# drive as follows (HEX)
3E A5 01 04 E8 01 28 23 00 4C
Driver reply, data length: 8byte

		Instruction	Memo
1	Head byte	0x3E	
2	Command byte	0xA5	CMD
3	ID byte	0x01~0x20	#1~#32
4	Data length byte	0x02	
5	Head check byte	1~4byte check sum	
6	Encoder data low byte	=*(int8_t *)(&encoder)	The encoder data is 16bit shaped data, and the data range is related to the encoder accuracy, which is generally 0~16383 (14bit).
7	Encoder data high byte	=*((int8_t *)(&encoder)+1)	

8	Data check byte	6~7byte check sum	

Eg: Drive reply after receiving the position closed loop control data, as follows (HEX):
3E A5 01 02 E6 00 0A 0A

- **Position closed-loop control 4,** the command contains three control parameters, respectively defining the direction of rotation of the motor, the target position (single-turn angle value) and the maximum speed of the motor during the process of reaching the target position. The total length of the command: 14byte

		Instruction	memo
1	Head byte	0x3E	
2	Command byte	0xA6	CMD
3	ID byte	0x01~0x20	#1~#32
4	Data length byte	0x08	
5	Head check byte	1~4byte check sum	
6	Direction of rotation byte	0x00 or 0x01	0:clockwise,1 :counterclockwise
7	Motor position low byte	=*(uint8_t *)(&angle)	Motor angle represents the rotation angle of the motor, which is 32bit unsigned shaped data. The actual angle ratio is 0.01degree/LSB, data range is 0~35999,represents the angle range from 0°to 359.99°°
8	Motor position	=*((uint8_t *)(&angle)+1)	
9	Motor position high byte	=*((uint8_t *)(&angle)+2)	
10	Motor speed low byte	=*(int8_t *)(&speed)	The motor speed represents the angular velocity of the motor, which is 32bit shaped data. The actual speed ratio is 0.01 dps/LSB,36000 represents 360 dps, and the speed symbol is invalid
11	Motor speed	=*((int8_t *)(&speed)+1)	
12	Motor speed	=*((int8_t *)(&speed)+2)	
13	Motor speed high byte	=*((int8_t *)(&speed)+3)	
14	Data check byte	6~13 byte check sum	

Eg:the host sends a clockwise rotation of 180° &angular velocity of 10dps to the 1# drive as follows (HEX)
3E A6 01 08 ED 00 50 46 00 E8 03 00 00 81
the host sends a clockwise rotation of 90° &angular velocity of 90dps to the 1# drive as follows (HEX)
3E A6 01 08 ED 00 28 23 00 28 23 00 00 96
the host sends counterclockwise rotation of 315° &angular velocity of 180dps to the 1# drive as follows (HEX)
3E A6 01 08 ED 01 0C 7B 00 50 46 00 00 1E
the host sends a counterclockwise rotation of 45° &angular velocity of 45dps to the 1# drive as follows (HEX)
3E A6 01 08 ED 01 94 11 00 94 11 00 00 4B

8byte Driver reply, data length: 8byte

		Instruction	Memo
1	Head byte	0x3E	
2	Command byte	0xA6	CMD
3	ID byte	0x01~0x20	#1~#32
4	Data length byte	0x02	

5	Head check byte	1~4byte check sum	
6	Encoder data low byte	=*(int8_t *)(&encoder)	The encoder data is 16bit shaped data, and the data range is related to the encoder accuracy, which is generally 0~16383 (14bit).
7	Encoder data high byte	=*((int8_t *)(&encoder)+1)	
8	Data check byte	6~7byte check sum	

Eg: Drive reply after receiving the position closed loop control data, as follows (HEX):

3E A6 01 02 E7 00 0A 0A

- **Products application：**