

## Myactuator joint module CAN-EMCP control protocol

### Disclaimer

Thank you for purchasing myactuator RMD series motor drive system. Please read this statement carefully before use. Once used, it is deemed to be recognition and acceptance of the entire content of this statement. Please install and use this product in strict compliance with the manual, product instructions, and relevant laws, regulations, policies, and guidelines. In the process of using the product, users promise to be responsible for their own actions and all consequences arising therefrom. Myactuator will not be held legally responsible for any losses caused by improper use, installation, or modification by users.

Myactuator is a trademark of SUZHOU MICRO ACTUATOR TECHNOLOGYCO LTD and its affiliated companies. The product names and brands appearing in this article are the trademarks or registered trademarks of their respective companies.

This product and manual are the copyright of myactuator. Reproduction in any form is prohibited without permission. The final right to interpret the disclaimer belongs to Myactuator.

### contents

Disclaimer.....	1
1. Communication bus parameters and message format.....	5
1.1. CAN bus.....	5
1.1.1. Parameters.....	5
1.1.2. Message format.....	5
2. Instruction description.....	5
2.1. Emergency stop (command address: 00000).....	5
2.1.1. Instruction description.....	5
2.1.2. Send data field definition.....	5
2.1.3. Reply data field definition.....	5
2.1.4. Communication example.....	6
2.2. Set motor status (command address: 00001).....	6
2.2.1. Instruction description.....	6
2.2.2. Send data field definition.....	6
2.2.3. Function index description.....	6
2.2.4. Reply data field definition.....	6
2.2.5. Communication example.....	7
2.3. Read motor status (command address: 00010).....	7
2.3.1. Instruction description.....	7
2.3.2. Send data field definition.....	7
2.3.3. Reply data field definition.....	7
2.3.4. Function index description.....	7
2.3.5. Communication example.....	8
2.4. Set motor mode (command address: 00011).....	8
2.4.1. Instruction description.....	8
2.4.2. Send data field definition.....	8
2.4.3. Function index description.....	8
2.4.4. Reply data field definition.....	8
2.4.5. Communication example.....	9
2.5. Read motor mode (command address: 00100).....	9
2.5.1. Instruction description.....	9
2.5.2. Send data field definition.....	9
2.5.3. Reply data field definition.....	9
2.5.4. Function index description.....	9
2.5.5. Communication example.....	10
2.6. Set the current position as zero point (command address: 00101).....	10
2.6.1. Instruction description.....	10
2.6.2. Send data field definition.....	10
2.6.3. Reply data field definition.....	10
2.6.4. Communication example.....	10
2.7. Set PID parameters (command address: 00110).....	11
2.7.1. Instruction description.....	11
2.7.2. Send data field definition.....	11
2.7.3. Function index description.....	11

2.7.4. Reply data field definition.....	12
2.7.5. Communication example.....	12
Send command:.....	12
2.8. Read PID parameters (command address: 00111) .....	13
2.8.1. Instruction description.....	13
2.8.2. Send data field definition.....	13
2.8.3. Function index description.....	13
2.8.4. Reply data field definition.....	14
2.8.5. Communication example.....	14
2.9. Set restriction parameters (command address: 01000) .....	15
2.9.1. Instruction description.....	15
2.9.2. Send data field definition.....	15
2.9.3. Function index description.....	15
2.9.4. Reply data field definition.....	16
2.9.5. Communication example.....	16
2.10. Read restriction parameters (command address: 01001) .....	16
2.10.1. Instruction description.....	16
2.10.2. Send data field definition.....	17
2.10.3. Function index description.....	17
2.10.4. Reply data field definition.....	17
2.10.5. Communication example.....	18
2.11. Single point operation (instruction address: 01010) .....	18
2.11.1. Instruction description.....	18
2.11.2. Send data field definition.....	18
2.11.3. Reply data field definition.....	18
2.11.4. Communication example.....	19
2.12. Single point trajectory (command address: 01011) .....	19
2.12.1. Instruction description.....	19
2.12.2. Send data field definition.....	19
2.12.3. Reply data field definition.....	19
2.12.4. Communication example.....	20
2.13. Set continuous trajectory position data (command address: 01100).....	20
2.13.1. Instruction description.....	20
2.13.2. Send data field definition.....	20
2.13.3. Reply data field definition.....	20
2.13.4. Communication example.....	20
2.14. Set continuous trajectory speed data (command address: 01101).....	21
2.14.1. Instruction description.....	21
2.14.2. Send data field definition.....	21
2.14.3. Reply data field definition.....	21
2.14.4. Communication example.....	21
2.15. Set continuous trajectory moment flow data (command address: 01110).....	22
2.15.1. Instruction description.....	22
2.15.2. Send data field definition.....	22

2.15.3. Reply data field definition.....	22
2.15.4. Communication example.....	22
2.16. Specify trajectory data operation (command address: 01111) .....	23
2.16.1. Instruction description.....	23
2.16.2. Send data field definition.....	23
2.16.3. Reply data field definition.....	23
2.16.4. Communication example.....	23
2.17. Record current trajectory data (command address: 10000) .....	24
2.17.1. Instruction description.....	24
2.17.2. Send data field definition.....	24
2.17.3. Reply data field definition.....	24
2.17.4. Communication example.....	24
2.18. Read operating data (command address: 10001) .....	25
2.18.1. Instruction description.....	25
2.18.2. Send data field definition.....	25
2.18.3. Function index description.....	25
2.18.4. Reply data field definition.....	26
2.18.5. Communication example.....	26
2.19. Set CAN ID (command address: 10010) .....	26
2.19.1. Instruction description.....	26
2.19.2. Send data field definition.....	26
2.19.3. Reply data field definition.....	26
2.19.4. Communication example.....	27
2.20. Restore settings (command address: 10011) .....	27
2.20.1. Instruction description.....	27
2.20.2. Send data field definition.....	27
2.20.3. Reply data field definition.....	27
2.20.4. Communication example.....	27
2.21. Start OTA service (command address: 10100) .....	28
2.21.1. Instruction description.....	28
2.21.2. Send data field definition.....	28
2.21.3. Reply data field definition.....	28
2.21.4. Communication example.....	28
3. Appendix.....	28
3.1.1. Indicator light status description.....	28
3.1.2. Online hexadecimal conversion link.....	29
4. Version revision information.....	29

## 1. Communication bus parameters and message format

### 1.1. CAN bus

#### 1.1.1. Parameters

Bus interface: CAN

Baud rate: 1Mbps

Communication method: EMCP-CAN adopts instruction address communication method and supports 64 nodes and 128 instruction addresses.

DLC: Variable length DLC

#### 1.1.2. Message format

Identifier: start frame: 0

Device address: 00000~11110 Broadcast address: 11111 (Default device address is: 00001)

Instruction address: 00000~11111

Reply flag: 0: No reply required, 1: Reply required

Example: When the ID is 1, 0 00001 00000 1 is converted into hexadecimal 0x41.

Start frame device address command address reply flag

Frame format: data frame

Frame type: standard frame

## 2. Instruction description

### 2.1. Emergency stop (command address: 00000)

#### 2.1.1. Instruction description

Emergency stops the motor and keeps it enabled.

#### 2.1.2. Send data field definition

ID frame: 0000100000

Reply flag: 0: No reply required, 1: Reply required

Command parameters: none

#### 2.1.3. Reply data field definition

ID frame: 0000100000

Reply flag: 0 – failure, 1 – success

Reply data: None

#### 2.1.4. Communication example

Example 1:

Send command: Reply command:

**CAN:**

ID
0x41

**CAN:**

ID
0x41

illustrate:

Send a command to make the motor stop running in an emergency and keep it in the enabled state.

If the reply command is 0x41, the reply is successful, otherwise the reply fails.

## 2.2. Set motor status (command address: 00001)

### 2.2.1. Instruction description

Set several status commands for the motor.

### 2.2.2. Send data field definition

ID frame: 0000100001

Reply flag: 0: No reply required, 1: Reply required

Instruction parameters: Function index (1 Byte, Data[0])

### 2.2.3. Function index description

index value	Command name	Function Description
0x00	disabled motor	Put the motor into a disabled state.
0x01	Enable motor	Put the motor into the enable state.
0x02	Motor restart	Restart the main control.
0x03	Parameter reset	Restore motor parameters to default values.
0x04	reset error status	Reset motor error status

### 2.2.4. Reply data field definition

ID frame: 0000100001

Reply flag: 0 – failure, 1 – success

Reply data: None

### 2.2.5. Communication example

Example 1:

Send command: Reply command:

CAN: CAN:

ID	Data[0]
0x43	0x00

ID
0x43

illustrate:

Send instructions to set the motor status.

Data[0] represents the index value, and sending 0x00 disables the motor.

If the reply command ID is 0x43, the reply is successful, otherwise the reply fails.

## 2.3. Read motor status (command address: 00010)

### 2.3.1. Instruction description

Read several status instructions of the motor.

### 2.3.2. Send data field definition

ID frame: 0000100010

Reply flag: Default reply

Command parameters: none

### 2.3.3. Reply data field definition

ID frame: 0000100010

Reply flag: 0 – failure, 1 – success

Reply data: Function index, Byte type, 1 byte

### 2.3.4. Function index description

index value	Command name	Function Description
0x00	No alarm	No error alarm
0x81	Overtemperature (motor)	The motor temperature exceeds the limit value.
0x83	overvoltage	The voltage exceeds the maximum voltage limit

0x84	Undervoltage	The voltage is lower than the minimum voltage limit
------	--------------	---

### 2.3.5. Communication example

Example 1:

Send command: Reply command:

CAN: CAN:

ID	ID	Data[0]
0x45	0x45	0x00

illustrate:

Send a command to read the motor status.

If the reply command ID is 0x45, it is successful, otherwise the reply fails.

Data[0] represents the index value, and replying 0x00 means there is no error alarm.

If a false alarm occurs, such as overvoltage, the alarm status will not be automatically eliminated after the voltage is adjusted to normal, and a command needs to be sent manually to eliminate it or the motor needs to be powered off and restarted.

## 2.4. Set motor mode (command address: 00011)

### 2.4.1. Instruction description

Set several mode instructions for the motor.

### 2.4.2. Send data field definition

ID frame: 0000100011

Reply flag: 0: No reply required, 1: Reply required

Instruction parameters: Function index (1 Byte, Data[0])

### 2.4.3. Function index description

index value	Command name	Function Description
0x00	Torque mode	Current mode rotation.
0x01	speed mode	Turn in current limiting mode.
0x02	Position mode	Rotate in speed-limited and current-limited modes.

### 2.4.4. Reply data field definition

ID frame: 0000100011

Reply flag: 0 - failure, 1 - success

Reply data: None



#### 2.4.5. Communication example

Example 1:

Send command: Reply command:

CAN: CAN:

ID	Data[0]
0x47	0x00

ID
0x47

illustrate:

Send instructions to set the motor operating mode.

Data[0] represents the index value. Sending 0x00 sets the motor operating mode to torque mode.

If the reply command ID is 0x47, the reply is successful, otherwise the reply fails.

## 2.5. Read motor mode (command address: 00100)

### 2.5.1. Instruction description

Read several mode instructions of the motor.

### 2.5.2. Send data field definition

ID frame: 0000100100

Reply flag: Default reply

Command parameters: none

### 2.5.3. Reply data field definition

ID frame: 0000100100

Reply flag: 0 – failure, 1 – success

Reply data: Function index, Byte type, 1 byte

### 2.5.4. Function index description

index value	Command name	Function Description
0x00	Torque mode	Current mode rotation
0x01	speed mode	Turn in current limiting mode.
0x02	Position mode	Rotate in speed-limited and current-limited modes.

#### 2.5.5. Communication example

Example 1:

Send command: Reply command:

CAN: CAN:

ID
0x49

ID	Data[0]
0x49	0x00

illustrate:

Send a command to read the motor operating mode.

If the reply command ID is 0x49, the reply is successful, otherwise the reply fails. Data[0] represents the index value, and returning 0x00 means that the read motor operating mode is torque mode.

### 2.6. Set the current position as zero point (command address: 00101)

#### 2.6.1. Instruction description

Set the current position as zero point and reset it after power off.

#### 2.6.2. Send data field definition

ID frame: 0000100101

Reply flag: 0: No reply required, 1: Reply required

Command parameters: none

#### 2.6.3. Reply data field definition

ID frame: 0000100101

Reply flag: 0 - failure, 1 - success

Reply data: None

#### 2.6.4. Communication example

Example 1:

Send command: Reply command:

**CAN:**

ID
0x4B

**CAN:**

ID
0x4B

illustrate:

Send a command to set the current position of the motor to zero.

If the reply command ID is 0x4B, the reply is successful, otherwise the reply fails.

## 2.7. Set PID parameters (command address: 00110)

### 2.7.1. Instruction description

Set PID parameter instructions.

### 2.7.2. Send data field definition

ID frame: 0000100110

Reply flag: 0: No reply required, 1: Reply required

Instruction parameters: Parameter 1: PID parameter address index (1 Byte, Date[0])

Parameter 2: PID value (Float, 4 Byte, Date[1] is the low bit, Date[3] is the high bit)

### 2.7.3. Function index description

index value	Command name	Function Description
0x00	position ring P	(float)
0x01	position ring I	(float)
0x02	position ring D	(float)
0x03	position loop slope	(float)
0x04	Position loop filter period	(float)
0x05	Speed ring P	(float)
0x06	Speed loop I	(float)
0x07	Speed ring D	(float)
0x08	speed loop slope	(float)
0x09	Speed loop filter period	(float)
0x0A	Q-axis current loop P	(float)

0x0B	Q-axis current loop I	(float)
0x0C	Q-axis current loop D	(float)
0x0D	Q-axis current loop slope	(float)
0x0E	Q-axis current loop filter period	(float)
0x0F	D axis current loop P	(float)
0x10	D-axis current loop I	(float)
0x11	D axis current loop D	(float)
0x12	D-axis current loop slope	(float)
0x13	D-axis current loop filter period	(float)

#### 2.7.4. Reply data field definition

ID frame: 0000100110

Reply flag: 0 - failure, 1 - success

Reply data: None

#### 2.7.5. Communication example

Example 1:

Send command:

Reply instructions

CAN:

ID	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]
0x4D	0x00	0x00	0x00	0xC8	0x42

CAN:

ID
0x4D

illustrate:

Send instructions to set motor PID parameters.

Data[0] represents the index value, Data[1]-Data[4] data from low to high, input the converted hexadecimal number.

The corresponding value of 00 00 C8 42 is 100. For the hex conversion tool, see Appendix 3.1.2 Online Hex Conversion Link.

If the reply command ID is 0x4D, the reply is successful, otherwise the reply fails.

## 2.8. Read PID parameters (command address: 00111)

### 2.8.1. Instruction description

Read PID parameter instructions.

### 2.8.2. Send data field definition

ID frame: 0000100111

Reply flag: Default reply

Instruction parameters: Parameter 1: PID parameter address index (1 Byte, Data[0])

### 2.8.3. Function index description

index value	Command name	Function Description
0x00	position ring P	(float)
0x01	position ring I	(float)
0x02	position ring D	(float)
0x03	position loop slope	(float)
0x04	Position loop filter period	(float)
0x05	Speed ring P	(float)
0x06	Speed loop I	(float)
0x07	Speed ring D	(float)
0x08	speed loop slope	(float)
0x09	Speed loop filter period	(float)

0x0A	Q-axis current loop P	(float)
0x0B	Q-axis current loop I	(float)
0x0C	Q-axis current loop D	(float)
0x0D	Q-axis current loop slope	(float)
0x0E	Q-axis current loop filter period	(float)
0x0F	D axis current loop P	(float)
0x10	D-axis current loop I	(float)
0x11	D axis current loop D	(float)
0x12	D-axis current loop slope	(float)
0x13	D-axis current loop filter period	(float)

#### 2.8.4. Reply data field definition

ID frame: 0000100111

Reply flag: 0 - failure, 1 - success

Reply data: PID value (Float, 4 Byte, Date[0] is the low bit, Date[3] is the high bit)

#### 2.8.5. Communication example

Example 1:

Send command: Reply command:

**CAN:**

**CAN:**

ID	Data[0]
0x4F	0x00

ID	Data[0]	Data[1]	Data[2]	Data[3]
0x4F	0x00	0x00	0xC8	0x42

illustrate:

Send instructions to read PID parameters.

Data[0] represents the index value, and sending 0x00 is to read the position ring P parameter.

If the reply command ID is 0x4F, the reply is successful, otherwise the reply fails. The reply data Data[0]-Data[3] is the converted hexadecimal number from low to high. The corresponding value of 00 00 C8 42 is 100. For the hex conversion tool, see Appendix 3.1.2 Online Hex Conversion Link.

## 2.9. Set restriction parameters (command address: 01000)

### 2.9.1. Instruction description

Set limit parameter command.

### 2.9.2. Send data field definition

ID frame: 0000101000

Reply flag: 0: No reply required, 1: Reply required

Instruction parameters: Parameter 1: PID parameter address index (1 Byte, Date[0])

Parameter 2: Limit parameter value (Float, 4 Byte, Date[1] is the low bit, Date[4] is the high bit)

### 2.9.3. Function index description

index value	Command name	Function Description
0x01	Motor temperature limit	Temperature setting value: 50-100 (float)
0x02	voltage limit	Voltage limit (0.1-supply voltage/2) (float)
0x03	current limit	Single point operating current limit ( $\pm 100.0$ ) (float)
0x04	speed limit	Single point operating speed limit ( $\pm 6000.0$ ) (float)
0x05	Position	Minimum angle to limit rotation (float)

	limit-minimum	
0x06	position limit-max	Limit the maximum angle of rotation (float)
0x07	Brake start	Brake start duty cycle setting parameters (parameter range: 0-100) (uint32)
0x08	Brake maintenance	Brake maintenance duty cycle setting parameters (parameter range: 0-100) (uint32)
0x09	Overvoltage value	Overvoltage error status setting parameters (parameter range 12-40) (float)
0x0A	Reduction ratio	(uint16)
0x0B	Motor number	(uint16)
0x0C	Factory time	(uint32)

#### 2.9.4. Reply data field definition

ID frame: 0000101000

Reply flag: 0 - failure, 1 - success

Reply data: None

#### 2.9.5. Communication example

Example 1:

Send command: Reply command:

**CAN:**

ID	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]
0x51	0x00	0x00	0x00	0xC8	0x42

**CAN:**

ID
0x51

illustrate:

Send instructions to set motor limit parameters.

Data[0] represents the index value, Data[1]-Data[4] data from low to high, input the converted hexadecimal number.

The corresponding value of 00 00 C8 42 is 100. For the hex conversion tool, see Appendix 3.1.2 Online Hex Conversion Link.

If the reply command ID is 0x51, the reply is successful, otherwise the reply fails.

### 2.10. Read restriction parameters (command address: 01001)

#### 2.10.1. Instruction description



Read limit parameter command.

#### 2.10.2. Send data field definition

ID frame: 0000101001

Reply flag: 0: No reply required, 1: Reply required

Instruction parameters: PID parameter address index (1 Byte, Data[0])

#### 2.10.3. Function index description

index value	Command name	Function Description
0x01	Motor temperature limit	50-100 Temperature setting value: 50-100
0x02	voltage limit	Voltage limit (0.1-supply voltage/2)
0x03	current limit	Single point operating current limit ( $\pm 100.0$ )
0x04	speed limit	Single point operating speed limit ( $\pm 6000.0$ )
0x05	Position limit-minimum	Limit the minimum angle of rotation
0x06	positionlimit-max	Limit the maximum angle of rotation
0x07	Brake start	Brake start duty cycle setting parameters (parameter range: 0-100) (uint32)
0x08	Brake maintenance	Brake maintenance duty cycle setting parameters (parameter range: 0-100) (uint32)
0x09	Overvoltage value	Overvoltage error status setting parameters (parameter range 12-40) (float)
0x0A	Reduction ratio	(uint16)
0x0B	Motor number	(uint16)
0x0C	Factory time	(uint32)

When the minimum and maximum values of the position limit parameter are the same, it means there is no position limit.

#### 2.10.4. Reply data field definition

ID frame: 0000101001

Reply flag: 0 – failure, 1 – success

Reply data: Limit parameter value (Float, 4 Byte, Data[0] is the low bit, Data[3] is the high bit)

#### 2.10.5. Communication example

Example 1:

Send command: Reply command:

**CAN:**

ID	Data[0]
0x53	0x01

**CAN:**

ID	Data[0]	Data[1]	Data[2]	Data[3]
0x53	0x00	0x00	0xA0	0x42

illustrate:

Send the command to read the restriction parameters.

Data[0] represents the index value, and sending 0x01 is to read the motor temperature limit parameter.

If the reply command ID is 0x53, the reply is successful, otherwise the reply fails. The reply data Data[0]–Data[3] is the converted hexadecimal number from low to high. The corresponding value of 00 00 A0 42 is 80. For the hex conversion tool, see Appendix 3.1.2 Online Hex Conversion Link.

## 2.11. Single point operation (instruction address: 01010)

### 2.11.1. Instruction description

Run the motor according to the target value according to the current mode.

### 2.11.2. Send data field definition

ID frame: 0000101010

Reply flag: 0: No reply required, 1: Reply required

Instruction parameters: target value (Float, 4 Byte, Data[0] is the low bit, Data[3] is the high bit)

### 2.11.3. Reply data field definition

ID frame: 0000101010

Reply flag: 0 – failure, 1 – success

Reply data: None

#### 2.11.4. Communication example

Example 1:

Send command: Reply command:

**CAN:**

ID	Data[0]	Data[1]	Data[2]	Data[3]
0x55	0x00	0x00	0xC8	0x42

**CAN:**

ID
0x55

illustrate:

Send instructions to make the motor run according to the target value according to the current mode. (If it is torque mode, the input data represents the current; if it is speed mode, the input data represents the speed; if it is position mode, the motor will run to the specified position at the limited speed.)

Data[0]—Data[3] data is from low bit to high bit, input the converted hexadecimal number.

The corresponding value of 00 00 C8 42 is 100. For the hex conversion tool, see Appendix 3.1.2 Online Hex Conversion Link.

If the reply command ID is 0x55, the reply is successful, otherwise the reply fails.

## 2.12. Single point trajectory (command address: 01011)

### 2.12.1. Instruction description

Run the motor to the specified position according to the specified speed and limited current parameters.

### 2.12.2. Send data field definition

ID frame: 0000101011

Reply flag: 0: No reply required, 1: Reply required

Instruction parameters: Position (Float, 4 Byte, Data[0] is the low bit, Data[3] is the high bit)

Speed (Float, 4 Byte, Data[4] is the low bit, Data[7] is the high bit)

### 2.12.3. Reply data field definition

ID frame: 0000101011

Reply flag: 0 - failure, 1 - success

Reply data: None

#### 2.12.4. Communication example

Example 1:

Send command: Reply command:

**CAN:**

ID	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]
0x57	0x00	0x00	0xC8	0x42	0x00	0x00	0xC8	0x42

**CAN:**

ID
0x57

illustrate:

Send instructions to make the motor run to the specified position according to the specified speed and limited current parameters. (This command can only be used in position mode.)

The value of Data[0]—Data[3] represents the position the motor has reached. The data is from low to high. Enter the converted hexadecimal number.

Data[4]—The value of Data[7] represents the speed of the motor. The data is from low to high. Enter the converted hexadecimal number.

The corresponding value of 00 00 C8 42 is 100. For the hex conversion tool, see Appendix 3.1.2 Online Hex Conversion Link.

If the reply command ID is 0x57, the reply is successful, otherwise the reply fails.

### 2.13. Set continuous trajectory position data (command address: 01100)

#### 2.13.1. Instruction description

Set position data parameters for continuous trajectory runs.

#### 2.13.2. Send data field definition

ID frame: 0000101100

Reply flag: 0: No reply required, 1: Reply required

Instruction parameters: trajectory data index (Int16, 2 Byte)

Position (Float, 4 Byte, Data[2] is the low bit, Data[5] is the high bit)

#### 2.13.3. Reply data field definition

ID frame: 0000101100

Reply flag: 0 – failure, 1 – success

Reply data: None

#### 2.13.4. Communication example

Example 1:

Send command: Reply command:

**CAN:**

ID	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]
0x59	0x00	0x00	0x00	0x00	0xC8	0x42

**CAN:**

ID
0x59

illustrate:

Send instructions to set position data parameters for continuous trajectory operation.

Data[0]-Data[1] represents the index value, 00 00 is the data for setting point 0, Data[2]-Data[5] data is from low to high, input the converted hexadecimal number. (If you set the data of point 1, the index value should be 01 00, and so on, up to 1000)

The corresponding value of 00 00 C8 42 is 100. For the hex conversion tool, see Appendix 3.1.2 Online Hex Conversion Link.

If the reply command ID is 0x59, the reply is successful, otherwise the reply fails.

## 2.14. Set continuous trajectory speed data (command address: 01101)

### 2.14.1. Instruction description

Set the speed data parameters for continuous trajectory operation.

### 2.14.2. Send data field definition

ID frame: 0000101101

Reply flag: 0: No reply required, 1: Reply required

Instruction parameters: trajectory data index (Int16, 2 Byte)

Speed (Float, 4 Byte, Data[2] is the low bit, Data[5] is the high bit)

### 2.14.3. Reply data field definition

ID frame: 0000101101

Reply flag: 0 - failure, 1 - success

Reply data: None

### 2.14.4. Communication example

Example 1:

Send command: Reply command:

**CAN:**

ID	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]
0x5B	0x00	0x00	0x00	0x00	0xC8	0x42

**CAN:**

ID
0x5B

illustrate:

Send instructions to set the speed data parameters for continuous trajectory operation.

Data[0]-Data[1] represents the index value, 00 00 is the data for setting point 0, Data[2]-Data[5] data is from low to high, input the converted hexadecimal number. (If you set the data of point 1, the index value should be 01 00, and so on, up to 1000)

The corresponding value of 00 00 C8 42 is 100. For the hex conversion tool, see Appendix 3.1.2 Online Hex Conversion Link.

If the reply command ID is 0x5B, the reply is successful, otherwise the reply fails.

## 2.15. Set continuous trajectory moment flow data (command address: 01110)

### 2.15.1. Instruction description

Sets the moment flow data parameters for continuous trajectory runs.

### 2.15.2. Send data field definition

ID frame: 0000101110

Reply flag: 0: No reply required, 1: Reply required

Instruction parameters: trajectory data index (Int16, 2 Byte)

Torque flow (Float, 4 Byte, Data[2] is the low bit, Data[5] is the high bit)

### 2.15.3. Reply data field definition

ID frame: 0000101110

Reply flag: 0 - failure, 1 - success

Reply data: None

### 2.15.4. Communication example

Example 1:

Send command: Reply command:

**CAN:**

ID	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]
0x5D	0x00	0x00	0x00	0x00	0xC8	0x42

**CAN:**

ID
0x5D

illustrate:

Send instructions to set the torque flow data parameters for continuous trajectory operation.

Data[0]-Data[1] represents the index value, 00 00 is the data for setting point 0, Data[2]-Data[5] data is from low to high, input the converted hexadecimal number. (If you set the data of point 1, the index value should be 01 00, and so on, up to 1000)

The corresponding value of 00 00 C8 42 is 100. For the hex conversion tool, see Appendix 3.1.2 Online Hex Conversion Link.

If the reply command ID is 0x5D, the reply is successful, otherwise the reply fails.

## 2.16. Specify trajectory data operation (command address: 01111)

### 2.16.1. Instruction description

Run the motor according to the specified trajectory data.

### 2.16.2. Send data field definition

ID frame: 0000101111

Reply flag: 0: No reply required, 1: Reply required

Instruction parameters: trajectory data index (Int16, 2 Byte)

### 2.16.3. Reply data field definition

ID frame: 0000101111

Reply flag: 0 - failure, 1 - success

Reply data: None

### 2.16.4. Communication example

Example 1:

Send command: Reply command:

**CAN:**

ID	Data[0]	Data[1]
0x5F	0x00	0x00

**CAN:**

ID
0x5F

illustrate:

Send instructions to make the motor run according to the specified trajectory data.

(Position, speed, and current data must be set)

Data[0]—Data[1] represents the index value. 00 00 runs according to the value set at point 0, 01 00 runs according to the value set at point 1, and so on, with a maximum of 1,000 settings.

If the reply command ID is 0x5F, the reply is successful, otherwise the reply fails.

## 2.17. Record current trajectory data (command address: 10000)

### 2.17.1. Instruction description

Record current position, velocity and moment flow data to the specified data index location.

### 2.17.2. Send data field definition

ID frame: 0000110000

Reply flag: 0: No reply required, 1: Reply required

Instruction parameters: trajectory data index (Int16, 2 Byte)

### 2.17.3. Reply data field definition

ID frame: 0000110000

Reply flag: 0 – failure, 1 – success

Reply data: None

### 2.17.4. Communication example

Example 1:

Send command: Reply command:

**CAN:**

ID	Data[0]	Data[1]
0x61	0x00	0x00

**CAN:**

ID
0x61



illustrate:

Send instructions to record current position, velocity and moment flow data to the specified data index location.

Data[0]—Data[1] represents the index value, 00 00 means recording to point 0, 01 00 means recording to point 1, and so on, up to 1000 can be set.

If the reply command ID is 0x61, the reply is successful, otherwise the reply fails.

## 2.18. Read operating data (command address: 10001)

### 2.18.1. Instruction description

Read the data when the motor is running.

### 2.18.2. Send data field definition

ID frame: 0000110001

Reply flag: Default reply

Instruction parameters: data address index (1 Byte, Data[0])

### 2.18.3. Function index description

index value	Command name
0x00	Current location (float)
0x01	Current speed (float)
0x02	Q-axis current (float)
0x03	Q-axis voltage (float)
0x04	D-axis current (float)
0x05	D-axis voltage (float)
0x06	Current motor temperature (float)
0x07	Program version (uint32)
0x0A	Position + speed (float)
0x0B	Q-axis voltage + Q-axis current (float)
0x0C	D-axis voltage + D-axis current (float)

0x0D	Motor name (ASCII code) (uint64)
------	----------------------------------

#### 2.18.4. Reply data field definition

ID frame: 0000110001

Reply flag: 0 – failure, 1 – success

Reply data: operating data (Float, 4 Byte or 8 Byte, Date[0] is the low bit, Date[3] is the high bit, Date[4] is the low bit, Date[7] is the high bit)

#### 2.18.5. Communication example

Example 1:

Send command: Reply command:

**CAN:**

ID	Data[0]
0x63	0x00

**CAN:**

ID	Data[0]	Data[1]	Data[2]	Data[3]
0x63	0x00	0x00	0x00	0x00

illustrate:

Send a command to read the data when the motor is running.

Data[0] represents the index value. 0x00 means reading the current position data of the motor

If the reply command ID is 0x63, the reply is successful, otherwise the reply fails. Data[0]—Data[3] data are from low bit to high bit and are converted hexadecimal numbers.

The corresponding value of 00 00 00 00 is 0. For the base conversion tool, see Appendix 3.1.2 for the online base conversion link.

## 2.19. Set CAN ID (command address: 10010)

#### 2.19.1. Instruction description

Set CAN ID, device address from 00001~11110.

#### 2.19.2. Send data field definition

ID frame: 0000110010

Reply flag: 0: No reply required, 1: Reply required

Instruction parameters: ID parameter (1 Byte, Date[0])

#### 2.19.3. Reply data field definition

ID frame: 0000110010

Reply flag: 0 - failure, 1 - success

Reply data: None

#### 2.19.4. Communication example

Example 1:

Send command: Reply command:

**CAN:**

ID	Data[0]
0x65	0x01

**CAN:**

ID
0x65

illustrate:

Send instructions to set CAN ID.

Data[0] represents the set motor serial number, and the corresponding data in hexadecimal should be entered. For example, motor No. 11 should be 0x0b.

0x01 means setting the motor ID to 57, 0x02 means setting the motor ID to 97.

0x03 means setting the motor ID to d7, and so on. Every time the serial number increases by 1, the corresponding ID increases by 4 (ID is hexadecimal).

If the reply command ID is 0x65, the reply is successful, otherwise the reply fails.

## 2.20. Restore settings (command address: 10011)

### 2.20.1. Instruction description

Restore the motor parameters, restart the motor and recalibrate. You need to wait for about 3 to 5 minutes.

### 2.20.2. Send data field definition

ID frame: 0000110011

Reply flag: 0: No reply required, 1: Reply required

Instruction parameters: ID parameter (1 Byte, Data[0])

### 2.20.3. Reply data field definition

No reply data field

### 2.20.4. Communication example

Example 1: Send command:

**CAN:**

ID
----

0x67

## 2.21. Start OTA service (command address: 10100)

### 2.21.1. Instruction description

Start the online update service.

### 2.21.2. Send data field definition

ID frame: 0000110100

Reply flag: 0: No reply required, 1: Reply required

Instruction parameters: ID parameter (1 Byte, Data[0])

### 2.21.3. Reply data field definition

No reply data field

### 2.21.4. Communication example

Example 1:

Send command:

CAN:

ID

0x69

## 3. Appendix

### 3.1.1. Indicator light status description

The blue indicator light stays on to indicate that the motor is running normally;

The flashing blue indicator light indicates that the motor is over-voltage or under-voltage;

A long red indicator light indicates that the motor is overheated;

The green indicator light flashes to indicate that the CAN bus is disconnected incorrectly;

The yellow indicator light stays on to indicate that the motor is being calibrated;

### 3.1.2. Online hexadecimal conversion link

<http://www.speedfly.cn/tools/hexconvert/>

Need to use IEEE 754 floating point to hexadecimal conversion

The base conversion process is shown in Figure 3.1.2.1:

For example, in position mode, when the motor needs to be moved 120 ° at a speed of 5RPM,  
The input command is shown in the following figure:

The conversion method is shown in the following figure:

For positional data, the unit of decimal data is degrees (°)

For speed data, the units of decimal data are RPM

Therefore, the final instruction data sent is 00 00 F0 42 00 00 A0 40

## 4. Version revision information

2023.08.21:

1. Add restoration parameters and restart calibration functions.

2023.08.29:

1. Add the function of recording trajectory data

2. Add the function of reading track position data
3. Add the function of reading trajectory speed data
4. Add the function of reading trajectory moment flow data

2023.9.15

1. Add CANID reply
2. Change the problem of setting the status of the motor to rotate after the motor stops.
3. Delete the main temperature control function

2023.9.22

1. Add CAN communication bus disconnection alarm green light flashing at high frequency
2. Add the function of reading program version

2023.9.28

1. Add the brake function, enable the command to automatically open the brake, enable the automatic close of the brake
2. Set the limit parameters to increase the brake opening and maintenance duty cycle parameters.

2023.10.08

1. Fix read mode reply error
2. Change of user zero setting function
3. Add error status reset function

2023.10.20

1. Added parameter reading and setting functions for motor number, motor name, reduction ratio, and factory time.
2. Add the function of automatically sending data during power-on calibration, and the process of lights turning on and off during the calibration process.
3. The units of angle and speed are changed from radians to degrees.

2023.10.23

1. Add the function of recording current trajectory data and adjust the original 10000-10010 command address.

2. Add OTA command

send feedback

Sidebar

history record

Saved

Give suggestion

