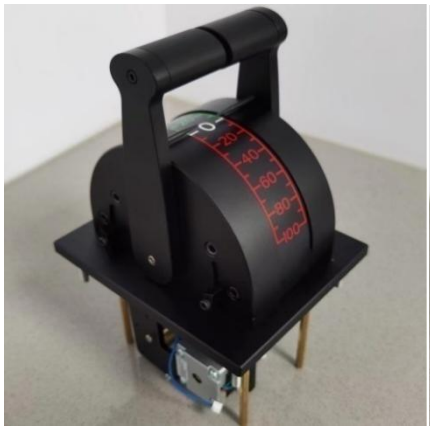
Dual Engine Throttle Controller

Car telegraph controller

MKF-JS72

Ver3.0 (ST32)

Independent PCBs for left and right steering wheels



MKF-JS72 Dual Engine Throttle Controller . Suitable for throttle control of marine engines, it can control two engines and features throttle and gear control functions. It has an electronic gear selector, adjustable PVC scale backlight, and uses a Hall sensor with full-temperature range linear compensation . It has an aluminum alloy housing with an anodized surface and uses nano-friction positioning materials for a good tactile feel.

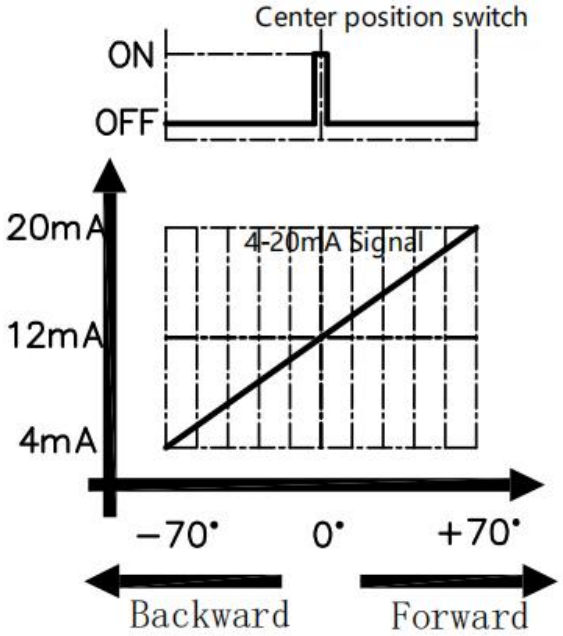
MATE TECHNOLOGY SHENZHEN LIMITED

2024-09-12

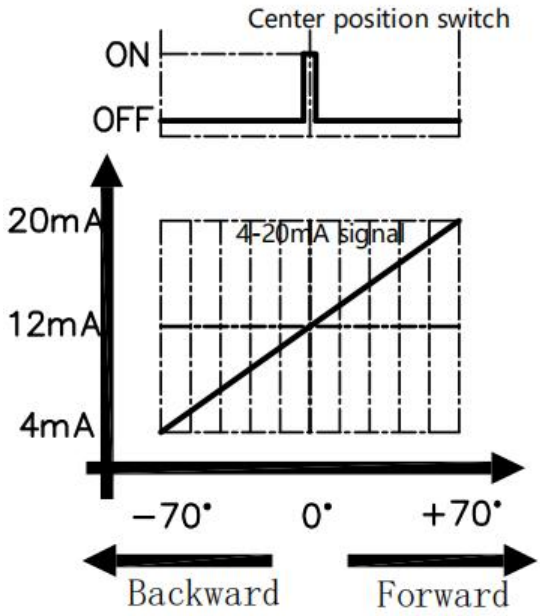
Edited on 2026-03-06

I. Characteristics:

- Sensor: Hall sensor
- Operating angle: 140 ° (± 70 °)
- Gears: friction brake, with center mechanical gear
- Synchronous control: Dual servo stepper motors, with an internal electromagnetic clutch mechanism for excellent handling feel.
  Synchronization control can be achieved through data communication or external input terminals, and remote synchronization control is also possible.
- Backlight: Adjustable backlight dimming via 0-24V signal input, dimming via communication data.
- Lifespan : Mechanical lifespan greater than 1 million cycles ;
- Signal output : 4-20mA current (12mA at midpoint), 2 channels on each side.
  Voltage signal: 0.5-2.5-4.5V or ± 10V, one channel each.
  CAN2.0 , 1 channel on each side
  RS485 , 2 channels on each side
- Power supply: DC 24V (DC10V /DC36V)
  Left and right engine power supply isolation, signal integrity isolation
- Operating temperature: -40 ℃ to +70 ℃
- Defend Protection : IP 67 above the panel



Left rudder signal output



Right rudder signal output

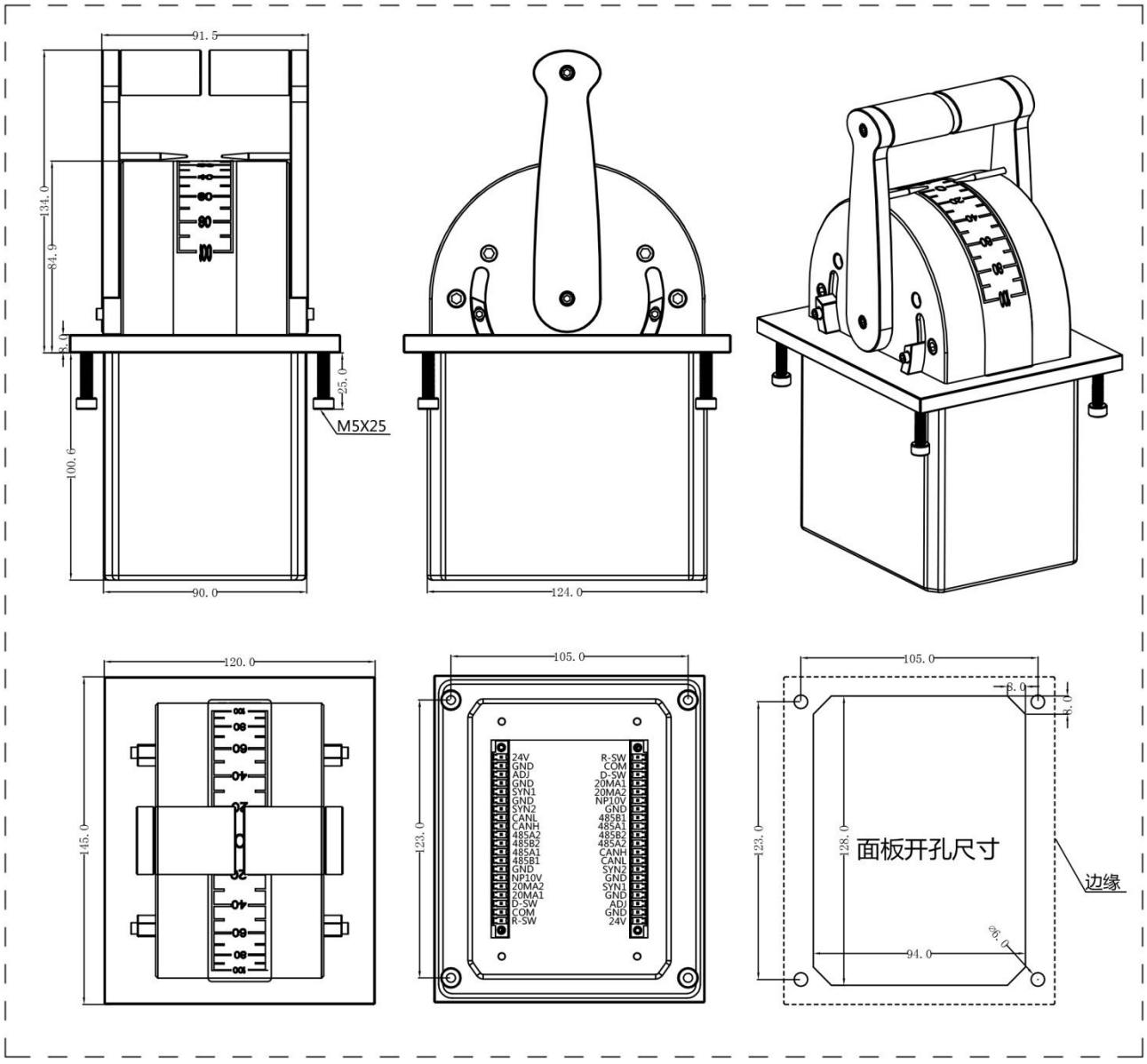
CAN interface : Supports J1939 and CAN open
RS485 Interface : Two isolated RS485 interfaces
Backlight : The backlight on the panel uses electroluminescent surface emission technology, providing uniform light and adjustable brightness from 0-100%.
Follow-up function : The follow-up function is an automatic synchronization function. The

controller has two follow-up switch interfaces. To activate the follow-up function,

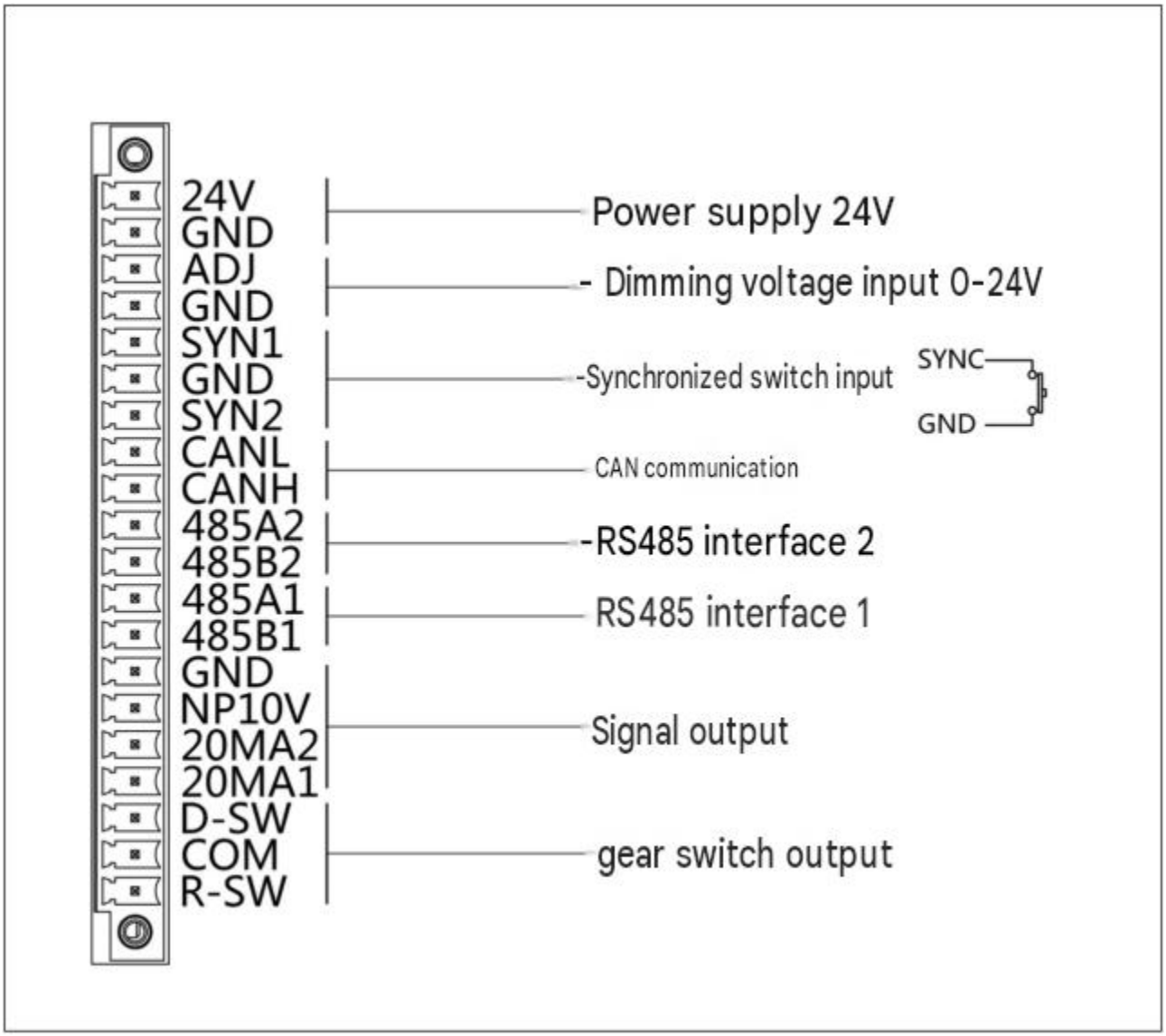
The internal servo motor automatically rotates the left and right rudder handles to ensure that the left and right signal outputs are the same, and one of the rudder handles...

As one rudder handle moves, the other automatically follows.

II. External Dimensions Drawing



External wiring diagram:



The diagram above shows the wiring for one side; the wiring for both sides is the same.
Connect a synchronous switch between SYNC1, SYNC2 and GND; connecting either one will work.
The 4-20mA signal output is valid; the NP10V standby output (± 10V) is not used.
4-20mA signal output, recommended load resistance: 120 Ω

III. Modbus RTU Communication Protocol

(1) Slave mode:

1. Baud rate: 9600
2. Data bits: 1 start bit, 8 data bits, 1 stop bit, and a parity bit.
3. Data format: Modbus
4. Working mode: Slave station
   The controller receives the read command and replies with 1 frame of data.

Modbus master data reading and slave response format (function code **03)** (PLC → joystick)

Byte 1	2 bytes	3 bytes	4 bytes	5 bytes	6 bytes	7 bytes	8 bytes
address	Read command	Register start address		Number of registers		CRC	
0x01	0x03	high	Low	High position 00	Low position 02	high	Low

Example: Left rudder "register start address" = 4001 01 03 4 0 01 00 02 80 0B
Right rudder control register start address = 4003 01 03 4 0 03 00 02 21 CB

When the controller receives this command, it switches from master mode to slave mode. (It stops actively sending data; the joystick will only respond with the following data after receiving this read data.)

The slave station returns 9 bytes of data – the joystick's response (joystick → PLC).

Function	data	Parameter range
Device address	0x01	Device address
Function code	0x03	
Data length	0x04	4 bytes
rudder shift	0x01	00 = Median 01 = Forward 02 = Back
rudder angle	0x64	0 - 100 / median = 0
Follow-up switch status	0x01	Bit0 = Input 1, bit1 = Input 2 1 = valid, 0 = invalid
Backlight brightness	0x1e	0-100
CRC high bits		
CRC low bits		

The device ID can be modified via commands, see section 4.(4)

For example: (Reading 4 registers)

PLC → Controller : 01 03 4 0 01 00 01 D5 CA

Joystick → PLC : 01 03 0 4 01 64 01 1e 3B 88

Modbus (RTU mode)

(1) Controller Master Station Mode:

1. Baud rate: 9600
2. Data bits: 1 start bit, 8 data bits, 1 stop bit, and a parity bit.
3. Data format: Modbus
4. Working mode: Main site

Main site data format (15 bytes)

Function	data	Parameter range
Device address	0x01	Modbus station number
Function code	0x10	0x10
First register address - high byte	0x00	Register address
First register address - low byte	0x01	
Number of registers - high bits	0x00	Number of registers
Number of registers - low bit	0x03	
Data length	0x06	6 bytes
Left tiller position	0x01	00=median 01=Forward 02=Back
left tiller angle	0x64	0 - 100 / median = 0
Right tiller position	0x01	00=median 01=Forward 02=Back
Right tiller angle	0x64	0 - 100 / median = 0
Follow-up switch status	0x08	
Backlight brightness	0x1e	0–100
CRC high bit		
CRC low bit		

The device ID can be modified via commands, see section 4(4).

For example: 011000 01 00 03 06 01 64 01 64 08 1e 00 B7

Modbus RTU

Write to register: (PLC → Controller)

(1) Remote control of the steering wheel

Write (8 bytes)

Function	data	Parameter range
Device address	0x01	Modbus station number
Function code	0x06	
Register address - high byte	0x40	Left handle=4005 Right handle=4007
Register address - low byte	0x05	
Gear shifter	81	See the table below for explanation.
Angle	twenty three	Throttle angle: 0-100 (0x00-0x64)
CRC high bits		
CRC low bits		

The register address is fixed and cannot be modified, but the device ID can be modified via instructions. See section 4.(4)

For example, ID=1: 01 06 40 05 81 23 AD 82

ID=2: 02 06 40 05 81 23 AD B1

① Gear

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1 = Motor Enable 0 = Motor disabled						Reversing	go ahead

Note: 1 = valid, 0 = invalid

Bit1-0 00 = Park / 01 = Forward / 02 = Reverse

② Throttle Angle: 0-100 (0x00-0x64);

When the current is 0, the output is 12mA; when moving forward 100, the output is 20mA; when moving backward 100, the output is 4mA.

③ When the controller receives this command, it automatically rotates the rudder to the corresponding angle.

Write to register: (PLC → Controller)

(2) Set the backlight brightness

Write (8 bytes)

Function	data	Parameter range
Device address	0x01	Modbus station number
Function code	0x06	
Register address - high byte	0x40	Register address 0X4009
Register address - low byte	0x09	
Backlight switch	0x01	00 = Turn off backlight 01 = Backlight on
Backlight brightness percentage	0x1e	Backlight brightness: 0-100 (0x00-0x64)
CRC high bits		
CRC low bits		

The register address is fixed at 0x4009 and cannot be modified. The device ID can be modified via instructions. See section 4.(4)

For example, ID=1: 01 06 40 09 01 1E CD 90

This command does not store parameters; after power is off, it will adjust the light intensity using the potentiometer.
When the controller receives this instruction, the backlight brightness will be based on the instruction. If the brightness knob is rotated, the brightness will be based on the knob's setting.

IV. RS485 Protocol Format (Non-Modbus)

Data bits: 8 bytes
Baud rate: 9600
Start bit 1, Stop bit 1, Parity bit

Sending method: Send data only if parameters have changed; otherwise, do not send data if the data is the same as the previous one.

Simultaneous transmission of 2 RS485 channels

Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
0xFF head	left handle gear 00 = Median	left handle angle 0-100	right handle gear 00 = Median	right handle angle 0-100	Follow-up switch status	Backlight brightness 0-100	CH Validity and

	01 = Forward 02 = Back		01 = Forward 02 = Back				

Note: CH = the low byte of the sum of Byte1 + Byte2 + Byte3 + Byte4 + Byte5 + Byte6 excluding the header.

Gear Parameter Explanation

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	0	0	0	0	0	R gear Accelerator	D gear Forward throttle

Note: Valid = 1, Invalid = 0

N gear = 0x00 Parking
D gear = 0x01 Forward throttle
R gear = 0x02 to accelerate.

Throttle angle value
Parameter range: 0x00-0x64 (hexadecimal) or 0-100 (decimal)
A value of 0 indicates 0% throttle.
A value of 100 indicates 100% throttle.

Follow-up (synchronous) switch status

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
						Enter 2	Enter 1

Note: 1 = Synchronization Start, 0 = Synchronization Disable. If either input 1 or 2 is 1, the synchronization function will be activated.

Backlight brightness: 0-100, which is adjusted from 0-100 by the system acquiring an external input voltage of 0-24V and using an external dimming potentiometer.

V. CAN Protocol Section

MKF-JS72 controller sends (Controller → Device)
Data frame: 8 bytes
CAN ID: Supports extended frames and standard frames (see CAN ID settings below).
Baud rate: 250K

Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
0X01 Fixed value	handle gear 00 = Median 01 = Forward	handle angle 0-100	Follow switch state	Backlight brightness 0-100	0x00	Cyclic counter 00-FF	0xA5 Fixed value

	02 = Back						

The loop counter increments by 1 for each frame of data sent, cycling through values from 0 to 255.

Right Handle =ID= 18fdd600 +01(address=01)

Left Handle = ID = 18fdd600 +02(address=02)

Gear Parameter Explanation

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	0	0	0	0	0	R gear Accelerator	D gear Forward throttle

Note: Valid = 1, Invalid = 0

N gear = 0x00 Parking

D gear = 0x01 Forward throttle

R gear = 0x02 to accelerate.

Throttle angle value

Parameter range: 0x00-0x64 (hexadecimal) or 0-100 (decimal)

A value of 0 indicates 0% throttle.

A value of 100 indicates 100% throttle.

Follow-up (synchronous) switch status

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
						Enter 2	Enter 1

Note: 1 = Synchronization Start, 0 = Synchronization Disable. If either input 1 or 2 is 1, the synchronization function will be activated.

Follow-up switch: The follow-up function is activated when either input 1 or input 2 is closed.

When starting automatically, it can only be controlled by the motor; manual operation of the handle is not allowed.

Backlight brightness: 0-100, based on the system's external input voltage of 0-24V.

If no external backlight control voltage is connected, the system defaults to 50% brightness.

1. Remote control of the steering wheel via CAN

Data frame: 8 bytes

CAN ID: Supports extended frames and standard frames (see CAN ID settings below).

(The received CAN ID can be changed)

Baud rate: 250K

Data command format: (HEX)

0x af	0x22	Gear	Angle	00	00	00	0x f5
head	Order	gear	angle	00	00	00	tail

Right Handle =ID= 18fdd 7 00 +01(address=01)
Left Handle = ID = 18fdd 7 00 +02(address=02)
Right handle address=0x01
Left handle address=0x02

Gear shifter

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1 = Motor Enable 0 = Motor disabled						Reversing	go ahead

Note: 1 = valid, 0 = invalid
Bit1-0 00 = Park / 01 = Forward / 02 = Reverse

Throttle angle: 0-100 (0x00-0x64);
When the current is 0, the output is 12mA; when moving forward 100, the output is 20mA; when moving backward 100, the output is 4mA.
When the controller receives this command, it automatically rotates the rudder to the corresponding angle.

For example: af 22 81 28 00 00 00 f5 (HEX)

2. Remotely control the backlight via CAN (do not store, restore default values upon disconnection).
canid=18fdd700+address
Right handle address=0x01
Left handle address=0x02

Command format:

0x af head	0x12 Order	Brightness value 0-100	00	00	00	00	0x f5 tail

Once the controller receives this command, it will display the brightness value specified in the command.
This function is only effective when there is no external dimming voltage (command dimming).

For example: af 12 64 00 00 00 f5 (HEX) 100% brightness

VI. System Control Commands:

.

Internal parameter settings

Users may need to set and modify the controller's communication parameters (CAN and

RS485).

All of the above-mentioned "parameter modifications" can only be set through the joystick's RS485 interface, including CAN parameters.

RS485 data format settings:

The PC sends commands to the joystick via the controller (RS485) host computer (serial port assistant software).

If the host PC does not have an RS232 port (DB9 9-pin connector), you will need to purchase a USB to RS485 adapter; if the host PC has an RS232 port, you will need to purchase an RS232 to RS485 adapter. The factory default is 115200.8.1.N.

Instruction data format: 8 bytes

af	CMD	D1	D2	D3	D4	D5	f5
Header	Command	Data 1	Data 2	Data 3	Data 4	Data 5	Tail

CAN sets the data format:

Baud rate = 250K, data length 8, CanID = 00ED0512 extended frame, data format as above RS485 communication: 115200.8.1.N

Response data format: CAN ACK: 80 00 00 00 00 00 00 00

For example: Set the switching cycle to 25ms

RS485 (9600.8.1.N): Send: a f 11 19 00 00 00 00 f5
Response: 80 00 00 00 00 00 00 00

CAN (250K, ID= 00ED0512) : Send: a f 11 19 00 00 00 00 f5
Response: 80 00 00 00 00 00 00 00

I. Basic Instructions:

1. ACK Confirmation (Joystick - PC)

Response: 80 00 00 00 00 00 00 00

This indicates that the message was successfully received and executed.

2. Configure the left and right joysticks (PC -> Joystick) Factory settings

Right handle: af 0 a 00 00 00 00 00 f5
Left handle: af 0 a 01 00 00 00 00 f5
Single controller: af 0 a 02 00 00 00 00 f5

First, set the left or right handle. After testing and confirming it works well, then set it to single or dual handle (dual test lights will illuminate).

Double handle: af 0 c 00 00 00 00 00 f5 (Default)

Single controller: af 0 c 01 00 00 00 00 f5

3. Configure the motor option (PC -> joystick) Factory settings

No motor: af 0 b 00 00 00 00 00 f5

There is a motor: af 0 b 01 00 00 00 00 f5

4. Calibrate the median

① Place the handle in the middle position

② Begin calibrating the median: af 0 9 01 00 00 00 00 f5

Slightly move the handle to collect the mechanical gap.

Calibrate sensor angle

③ Calibrate the sensor midpoint: af 0 9 02 00 00 00 00 f5

Slightly move the handle to collect the mechanical gap.

④ Store calibration values: af 0 9 03 00 00 00 00 f5

Calibration complete

5. Analog signal median error adjustment settings (PC -> joystick) Factory settings

Median signal calibration:

4-20mA Output 1: Midpoint 12mA Fine-tuning +1: af 11 01 01 00 00 00 f5

4-20mA Output 1: Midpoint 12mA Fine-tuning -1: af 11 01 02 00 00 00 f5

4-20mA Output 2: Midpoint 12mA Fine-tuning +1: af 11 02 01 00 00 00 f5

4-20mA Output 2: Midpoint 12mA Fine-tuning -1: af 11 02 02 00 00 00 f5

Voltage signal reference: Median voltage fine-tuning +1: af 11 03 01 00 00 00 f5

Voltage signal reference: Median voltage fine-tuning -1: af 11 03 02 00 00 00 f5

Store the above parameters: af 16 00 00 00 00 00 f5

6. Calibrate the 4mA and 204mA current settings at both ends (PC -> joystick) Factory settings

Output signal boundary calibration

4-20mA signal 1

4-20mA Output 1: 4mA Fine-tuning +1: af 12 01 01 00 00 00 f5

4-20mA Output 1: 4mA Fine-tuning -1: af 12 01 02 00 00 00 f5

4-20mA Output 1: 20mA Fine-tuning +1: af 12 02 01 00 00 00 f5

4-20mA Output 1: 20mA Fine-tuning -1: af 12 02 02 00 00 00 f5

4-20mA signal 2

4-20mA Output 2: 4mA Fine-tuning +1: af 12 03 01 00 00 00 f5

4-20mA Output 2: 4mA Fine-tuning -1: af 12 03 02 00 00 00 f5

4-20mA Output 2: 20mA Fine-tuning +1: af 12 04 01 00 00 00 f5
4-20mA Output 2: 20mA Fine-tuning -1: af 12 04 02 00 00 00 f5
Store the above parameters: af 16 00 00 00 00 00 f5

7. Set address(ID)

af 0 6 ID 00 00 00 00 f5

ID=1 af 0 6 01 00 00 00 00 f5 default
ID=2 af 0 6 02 00 00 00 00 f5
ID=3 af 0 6 03 00 00 00 00 f5
ID=4 af 0 6 04 00 00 00 00 f5

8. Set the address of the Modbus Reg register.

af 0f DH DL 00 00 00 f5
DH high position
DL low position
For example: Register address = 0x1234 af 0f 12 34 00 00 00 f5
Register address = 0x0001 af 0f 00 01 00 00 00 f5 (default)
Register address = 0x0002 af 0f 00 02 00 00 00 f5

9. Communication Interface Selection: (PC -> Controller)

af 08 00 00 00 00 00 f5 CAN and Modbus slave default
af 08 03 00 00 00 00 f5 RS485modbus Main Station

10. Sending cycle settings (PC -> joystick)

0x af 0x05 Cycle 00 00 Ref 0x f5
Header Command Data 1 Data 2 Data 3 Data 4 Tail
Cycle = 5-229 ms, unit is "milliseconds"; (factory default: 20ms)
Setting this parameter will take effect after a reset or restart.

Set 20MS a f 05 14 00 00 00 00 f5 (HEX)
Set 25MS a f 05 19 00 00 00 00 f5 (HEX)
Set 33MS a f 05 21 00 00 00 00 f5 (HEX)
Set 50MS a f 05 32 00 00 00 00 f5 (HEX)

11. Protocol selection: (PC -> Controller)

Select analog signal or CAN protocol function: Format af 07 pro 00 00 00 00 f5
PRO protocol number

af 07 00 00 00 00 00 f5 4-20mA (default)
af 07 01 00 00 00 00 f5 ±10V
af 07 02 00 00 00 00 f5 0.5V-4.5V (Remove R27 and R28, replace R29 with 0R)

12. Store parameters: (PC->Controller)
af 16 00 00 00 00 00 f5

13. Reset Controller: (PC->Controller)
af 17 00 00 00 00 00 f5

14. Erase all settings: (PC -> Controller)
af 27 00 00 00 00 00 f5

15. Test the control rudder.
CANID=00ED0512 250K For factory testing only
af 22 81 28 00 00 00 f5
af 22 82 28 00 00 00 f5
af 22 80 00 00 00 00 f5

16. Set the RS485 baud rate (PC -> joystick)
0x af 0x 0 4 Baud 00 00 00 00 0x f5

Set 9600 af 0 4 00 00 00 00 00 f5 (HEX)
Set 19200 af 0 4 01 00 00 00 00 f5 (HEX)
Set 38400 af 0 4 02 00 00 00 00 f5 (HEX)
Set 57600 af 0 4 03 00 00 00 00 f5 (HEX)
Set 115200 af 0 4 04 00 00 00 00 f5 (HEX)
After setting this parameter, the new baud rate will take effect immediately.

17. CAN Port Baud Rate: (PC -> Joystick)
af 03 00 00 00 00 00 f5 (HEX) CAN baud rate = 100K
af 03 01 00 00 00 00 f5 (HEX) CAN baud rate = 125K
af 03 02 00 00 00 00 f5 (HEX) CAN baud rate = 250K (default)
af 03 03 00 00 00 00 f5 (HEX) CAN baud rate = 500K
af 03 04 00 00 00 00 f5 (HEX) CAN baud rate = 1000K
After setting this parameter, the new baud rate will take effect immediately.

18. Joystick "Send Node ID" setting: (PC->Joystick)
29-bit extended frame:
af 01 18 fd d6 00 00 f5 CanId=18fdd600
CANID for transmitting data = CanId + address
Note that you need to set the address first.

11-bit standard frame:
af 01 80 00 00 01 80 f5 canid=0x0180
CANID for transmitting data = CanId + address
Note that you need to set the address first.

19. Joystick "Receive Node ID" setting: (PC->Joystick)

29-bit extended frame:

af 02 18 fd d7 00 00 f5 canid=18fdd700

CANID for received data = CanId + address

Note that you need to set the address first.

11-bit standard frame:

af 02 80 00 00 02 80 f5 canid=0x0280

CANID for received data = CanId + address

Note that you need to set the address first.

MATE TECHNOLOGY SHENZHEN LIMITED
www.matend.com
info@matend.com