MODEL NO.: MKF-JS61 Joystick

This joystick controller is suitable for camera remote control, robot teaching, three-dimensional, laser welding and other equipment. Ergonomic design, using the most advanced 3D Hall sensor in the world, high reliability.

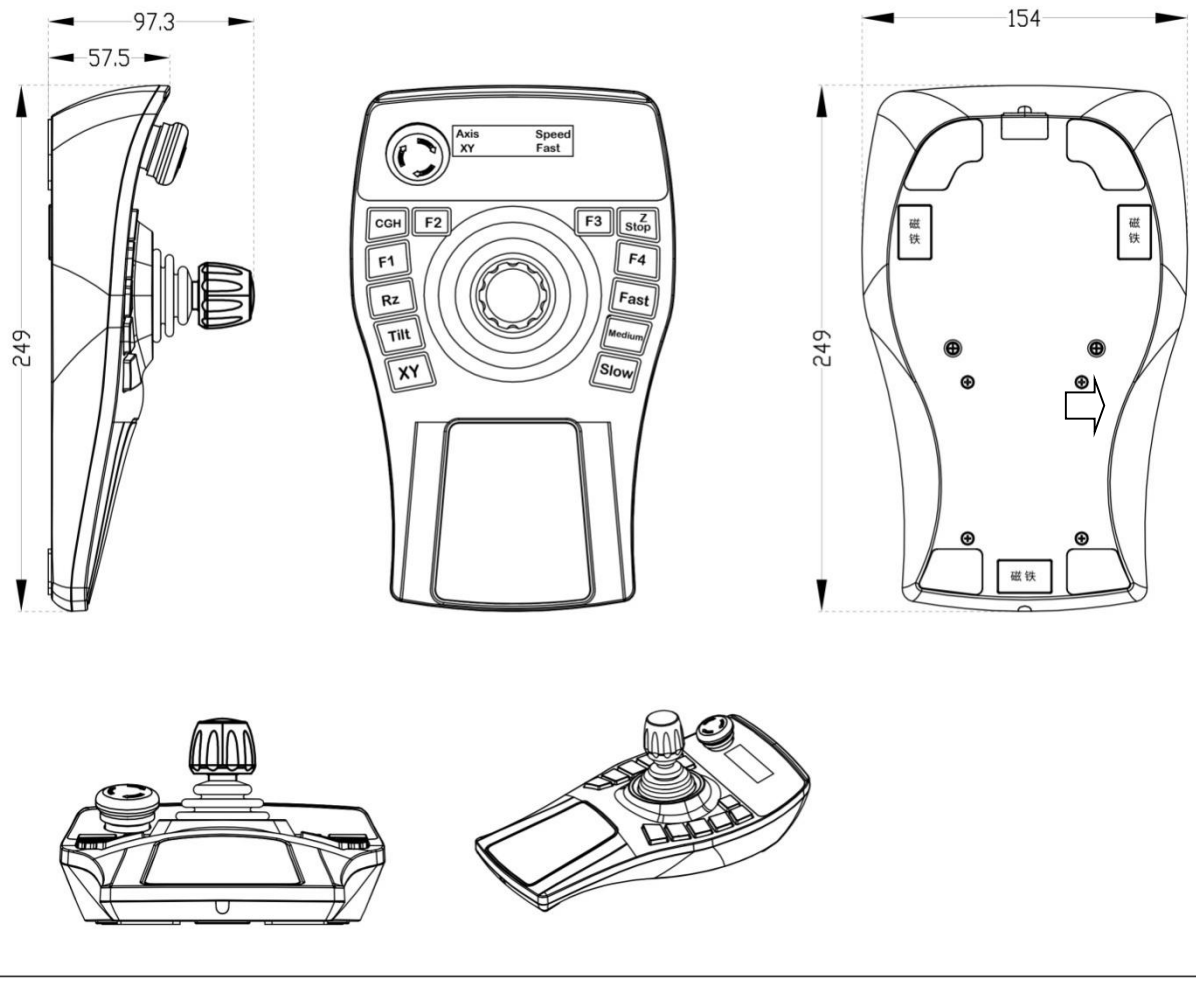
Features:

- Ergonomic design
- It has a strong magnetic adsorption function, which can be directly attached to the surface of the device to prevent it from falling and breaking.
- Integrated with a 3-axis Hall type joystick
- With 12 backlighting buttons, layout can be customized
- With one Emergency stop switch
- With one OLED display
- Signal output interface: RS232 / CAN
- Power supply: 10-36V
- Factory default communication mode: CAN/250K/30mS timing sending

Technical parameters:

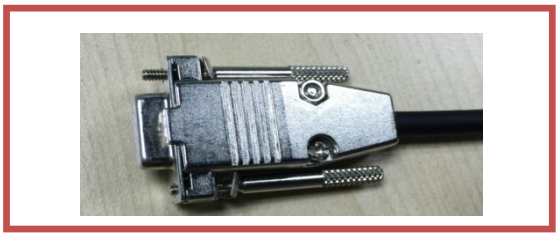
- Joystick: 3-axis, Hall sensor
- Display: OLED display 128X32
- Buttons: 12 buttons
- Power supply: DC10-36V
- Power consumption: less than 2W
- Signal output: RS232 or CAN interface
- Operating life: more than 1 million times
- Connector: DB9 male, cable length 3M
- Weight: 1.0KG

DRAWING:

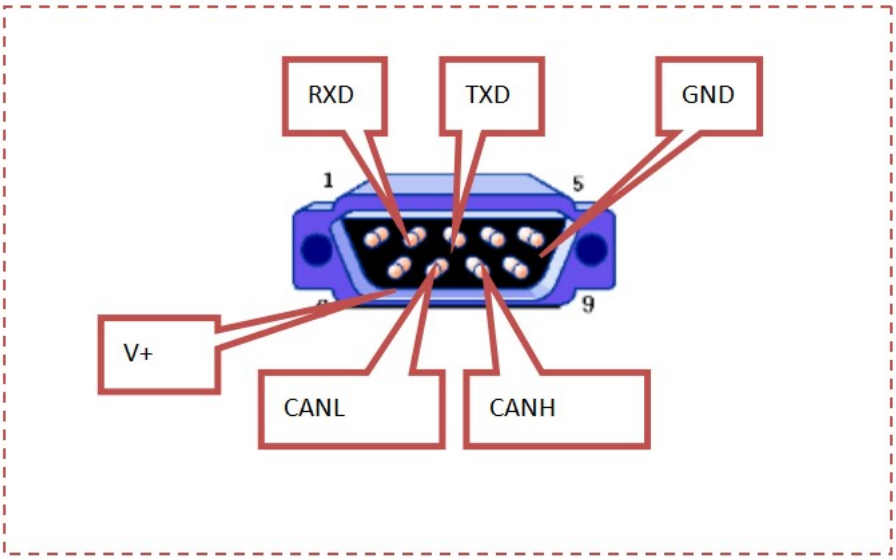

97.3
57.5
249
Axis
XY
Speed
Fast
CGH
F2
F3
Z
Stop
F1
F4
Rz
Fast
Tilt
Medium
XY
Slow
154
249
磁铁
磁铁
磁铁

Definition of RS232 communication

6-core shielded wire, 3M , the connector is as shown in the figure below: (DB9 male connector)



Schematic diagram of the connector



Pin definition: DB9 male

DB9 male	Symbol	Specification	Inner color	Remark
1	NC			
2	RXD	RS232 Joystick data reception	Grey Red	Connect to host TXD
3	TXD	RS232 Joystick data transmission	Grey Black	Connect to host RXD
4	NC			
5	GND	GND	Orange Black	GND
6	V+	Positive power VDD	Orange Red	Input Power: 10-36V
7	CAN L	CAN L	White Black	
8	CAN H	CAN H	White Red	
9	NC			
SHEILD		Shield		

CAN bus communication:

- CAN2.0B
- Frame ID: standard frame ID and extended frame ID, default ID=0X0181 (this ID can be modified through RS232)
- Baud rate: 125K, 250K, 500K, 1000K (default 250K)
- Sending method: continuous sending at intervals of 5-100ms (default 30ms)

CAN protocol format

Byte	Biyt7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Description
Byte0	00=work 01=E-stop								E-STOP
Byte1			CCW	CW	Right	Left	after	before	Joystick direction
Byte2	A7	A6	A5	A4	A3	A2	A1	A0	X angle 0-250
Byte3	A7	A6	A5	A4	A3	A2	A1	A0	Y angle 0-250
Byte4	A7	A6	A5	A4	A3	A2	A1	A0	Z angle 0-250
Byte5					Button F3	Button Z-STOP	Button F4	Button FAST	Button
Byte6	Button MEDIUM	Button SLOW	Button XY	Button TILT	Button RZ	Button F1	Button CGH	Button F2	Button
Byte7	0x00								STANDBY

RS232 data format (9 bytes) （Joystick→PC）:Hexadecimal HEX data

Byte	Biyt7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Description
Byte0	0XFF								Header
Byte1	00=work 01=E-stop								E-stop
Byte2			CCW	CW	Right	Left	after	before	Joystick direction
Byte3	A7	A6	A5	A4	A3	A2	A1	A0	X angle 0-250
Byte4	A7	A6	A5	A4	A3	A2	A1	A0	Y angle 0-250
Byte5	A7	A6	A5	A4	A3	A2	A1	A0	Z angle 0-250
Byte6					Button F3	Button Z-STOP	Button F4	Button FAST	Button
Byte7	Button MEDIUM	Button SLOW	Button XY	Button TILT	Button RZ	Button F1	Button CGH	Button F2	Button
Byte8	CH=Byte1+ Byte2+ Byte3+ Byte4+ Byte5+ Byte6+ Byte7 Except for the header, the remaining bytes are added, and the low byte of the added sum is taken								Sum check

eg: FF 02 00 00 00 00 00 A0 A2

Joystick communication parameter setting

The user may need to set and modify the communication parameters of the joystick (including CAN, RS232, RS422);

All the above "parameter modification" can only be done through the RS422 interface or RS232 interface of the joystick, including CAN parameters.

PC→Joystick (RS232) PC (serial port assistant) software sends commands to the joystick.

(If there is no serial port assistant software, you can ask our technical staff for it)

If the host PC does not have RS232 (DB9 9-pin connector), there is a USB to RS232 converter (standard converter, not a TTL level converter).

The RS232 communication interface on the joystick, the factory default baud rate is 9600.8.1.N

Basic instructions:

ACK confirmation (joystick-PC)

AA 55AF

Indicates that the joystick successfully receives the address setting command and executes it.

Set the joystick ID address

ID refers to the ID in the RS232 communication protocol, or the ID in the CAN open protocol

(PC->Joystick)

0xaf	0x0d	00	00	00	Add	0xf5
Head	Command	Data 1	Data 2	Data 3	Data 4	Tail

Add=0x01～0x7F Address 1-127

Add=0x00 is invalid (when address=0, there is no address bit in RS232 or RS422)

E.g:

Set address 1 af 0d 00 00 00 01 f5 (HEX)

Set address 2 af 0d 00 00 00 02 f5 (HEX)

The joystick receives this command, after execution, it will reply ACK

Reset Joystick (PC->Joystick)

0xaf	0x15	00	00	00	Add	0xf5
Head	Command	Data 1	Data 2	Data 3	Data 4	Tail

Add=0x01～0x7f The address must be the same as the joystick address to reset

Add=0x00 reset the joystick for all addresses, any address is reset

Add range is not valid from 0-0x7f

E.g:
Reset all address joysticks: af 15 00 00 00 00 f5 (HEX)
Reset address joystick 1: af 15 00 00 00 01 f5 (HEX)
Reset address joystick 2: af 15 00 00 00 02 f5 (HEX)

Set the center point of the joystick (for correcting the center point position) (PC->Joystick)
When it leaves the factory, it has been set, you can ignore this command
The PC is connected to the RS232 port of the joystick, the baud rate is 9600

0xaf	0x09	00	00	00	00	0xf5
Head	Command	Data 1	Data 2	Data 3	Data 4	Tail

Send some data to the joystick to reset the stop position (center point) of the joystick
Example: af 09 00 00 00 00 f5 (HEX)

Communication port selection: (PC->Joystick)
Joystick communication port RS232, CAN choose one; (the factory has been set for the customer)

0xaf	0x05	XX	00	00	00	0xf5
Head	Command	Data 1	Data 2	Data 3	Data 4	Tail

XX=00 CAN communication;
XX=01 RS232 communication
XX=02 RS232 Modbus RTU communication

For example: af 05 00 00 00 00 f5 (HEX) CAN communication
af 05 01 00 00 00 f5 (HEX) RS232 communication
af 05 02 00 00 00 f5 (HEX) RS232 communication Modbus RTU communication

Refresh Rate Settings (PC->Joystick)
Refresh rate = frame interval for sending data, for example, set 20ms (send one frame of data to the host every 20MS)

0xaf	0x11	00	00	00	Ref	0xf5
Head	Command	Data 1	Data 2	Data 3	Data 4	Tail

Ref =0x0A～0x64 (10-100)ms, the unit is "millisecond" (factory default: 20ms)
After setting this parameter, reset or restart takes effect
For example: Set the refresh rate of data sent by the joystick to 20MS (one frame of data is sent every 20MS, 50 times per second)
Set 20MS af 11 00 00 00 14 f5 (HEX)
Set 25MS af 11 00 00 00 19 f5 (HEX)
Set 33MS af 11 00 00 00 21 f5 (HEX)
Set 50MS af 11 00 00 00 32 f5 (HEX)
The joystick receives this command→reply ACK→reset the joystick
Note: The lower the baud rate, the longer the corresponding frame interval
Factory default: refresh rate 20ms (CAN baud rate 250K, RS232 and RS422 baud rate 9600)

Communication mode (master-slave query, timing automatic transmission, including CAN and RS232 communication) (PC->joystick)

Master-slave query: The joystick is a slave device, and it sends data back to the host only after receiving the query command from the host.
Timed automatic sending: the joystick sends data to the host when the joystick is turned on, and the sending rate refers to "refresh rate setting"
This parameter joystick is stored forever (the factory has been set for the customer)

Format:

0xaf	0x08	00	00	00	Mode	0xf5
Head	Command	Data 1	Data 2	Data 3	Data 4	Tail

Mode=00 master mode (timed transmission)
Mode=01 Slave mode (query mode)

Example: (PC->Joystick)
Timing Send (Master Mode) af 08 00 00 00 00 f5 (HEX)
Master-slave query (slave mode) af 08 00 00 00 01 f5 (HEX)
After the setting is successful, the joystick returns to ACK (AA 55 AF) (joystick->PC)

Backlight brightness (PC->joystick)

Format:

0xaf	0x12	00	00	BL	BF	0xf5
Head	Command	Data 1	Data 2	Backlight Brightness	Function Brightness	Tail

BL backlight brightness 00-0F
BF function brightness 00-0F

Example: (PC->Joystick)
af 12 00 00 08 0e f5 Backlight 8, function E

Query device information: (PC->Joystick)
af 20 00 00 00 00 f5 (HEX)

RS232 communication parameter setting

Set the RS232 baud rate (PC->joystick)

The baud rate of RS232 is the same, and the settings are valid at the same time

0xaf	0x0b	00	00	00	Baud	0xf5
Head	Command	Data 1	Data 2	Data 3	Data 4	Tail

Baud=0X00 Baud rate=9600
Baud=0X01 Baud rate=19200
Baud=0X02 Baud rate=57600
Baud=0X03 Baud rate=115200

E.g:

set 9600 af 0b 00 00 00 00 f5 (HEX)
set 19200 af 0b 00 00 00 01 f5 (HEX)
set 57600 af 0b 00 00 00 02 f5 (HEX)
set 115200 af 0b 00 00 00 03 f5 (HEX)
The joystick receives this command, after execution, it will reply ACK

Check the position of the patrol joystick (PC->joystick)

This command is valid only when the "master-slave query" can be trusted mode
When there is no query command, the joystick does not output any data, check the joystick once and return it once.

0xaf	0x07	00	00	00	Addr	0xf5
Head	Command	Data 1	Data 2	Data 3	Data 4	Tail

Addr address = 0x01-0x7f When the address is correct, loopback

When the joystick receives this data, it will send back the current position, check the joystick once and return it once, no data will be sent if it is not checked
For example, query during RS232 communication:
(PC->Joystick) af 07 00 00 00 01 f5 (HEX)
(Joystick -> PC) FF 01 08 00 70 00 00 00 00 00 79

When the joystick receives this data, it will send back the current position

Parameter setting of CAN communication

The parameter setting of CAN also goes through the RS232 port

CAN port baud rate: (PC->joystick)

0xaf	0x06	XX	00	00	00	0xf5
Head	Command	Data 1	Data 2	Data 3	Data 4	Tail

XX=00 125K
XX=01 250K (default)
XX=02 500K
XX=03 1000K
XX=04 100K

For example: af 06 00 00 00 00 f5 (HEX) CAN baud rate=125K
af 06 01 00 00 00 f5 (HEX) CAN baud rate=250K (default)
af 06 02 00 00 00 f5 (HEX) CAN baud rate=500K
af 06 03 00 00 00 f5 (HEX) CAN baud rate=1000K
af 06 04 00 00 00 f5 (HEX) CAN baud rate=100K

CAN protocol settings: (PC->joystick)

0xaf	0x0a	00	Pro	TP	SS	0xf5
Head	Command	Data 1	Data 2	Data 3	Data 4	Tail

Pro: Protocol format

Pro=00
Pro=01
Pro=02

SS: Canopen protocol

The Canopen protocol is different from the ordinary protocol mainly in CAN ID, and the format of the data has not changed

SS=01 CANopen protocol ID=180+ID (refer to (2) Set joystick ID address)
The factory has been set for the customer
SS=00 Non-CanOpen ID=sending node ID, see (11 Joystick sending node ID setting)
Default: Not CanOpen

TP: is the transmission channel TPDO in the CAN OPEN protocol
TP=00: TPDO1 sends ID 0X0180+ID (see 1, 2, set joystick ID address) default
TP=01: TPDO2 sends ID 0X0280+ID (see 1, 2, set joystick ID address)
TP=02: TPDO3 sends ID 0X0380+ID (see 1, 2, set joystick ID address)
TP=03: TPDO4 sends ID 0X0480+ID (see 1, 2, set joystick ID address)

For example: af 0a 00 00 00 00 f5 (HEX) normal protocol, protocol format XLDDQ
af 0a 00 00 00 01 f5 (HEX) CANopen protocol, TPDO1, protocol format XLDDQ

Joystick "Send Node ID" setting: (PC->Joystick)

Only applicable to "normal protocol", CANopen protocol does not use this command

0xaf	0x01	D1	D2	D3	D4	0xf5
Head	Command	Data 1	Data 2	Data 3	Data 4	Tail

D1.7=0 Extended frame 29 bits
D1.7=1 standard frame 11 bits

29-bit extended frame: data range 0X0-0X0FFFFFFF, data D1-D4 corresponds to "node identification code"
For example: Set the sending node identification code - extended frame "0X00F0F101"
af 01 00 f0 f1 01 f5 (HEX)

11-bit standard frame: data range 0X000-0X3FF, data D3-D4 corresponds to "node identification code"
For example: Set the sending node identification code - standard frame "0X181"
af 01 80 00 01 81 f5 (HEX)

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

Only applicable to "normal protocol", CANopen protocol does not use this command

0xaf	0x02	D1	D2	D3	D4	0xf5
Head	Command	Data 1	Data 2	Data 3	Data 4	Tail

D1.7=0 Extended frame 29 bits
D1.7=1 standard frame 11 bits

29-bit extended frame: data range 0X0-0X0FFFFFFF, data D1-D4 corresponds to "node identification code"
For example: Set the receiving node identification code - extended frame "0X00F0F101"
af 02 00 f0 f1 01 f5 (HEX)

11-bit standard frame: data range 0X000-0X3FF, data D3-D4 corresponds to "node identification code"
For example: Set the receiving node identification code - standard frame "0X1E1"
af 02 80 00 01 E1 f5 (HEX)

Joystick "Shield Node ID" setting: (PC->Joystick)

0xaf	0x03	D1	D2	D3	D4	0xf5
Head	Command	Data 1	Data 2	Data 3	Data 4	Tail

D1.7=0 Extended frame 29 bits
D1.7=1 standard frame 11 bits
29-bit extended frame: data range 0X0-0X0FFFFFFF, data D1-D4 corresponds to "node identification code"

For example: set the mask node identification code - extended frame "0X00002201"
af 03 00 00 22 01 f5 (HEX)
11-bit standard frame: data range 0X000-0X3FF, data D3-D4 corresponds to "node identification code"
For example: Set the mask node identification code - standard frame "0X122"
af 03 80 00 01 22 f5 (HEX)