

H38 SERIES

Mate Industrial Finger Joystick, Hall effect, Multi axis, Panel Mounted



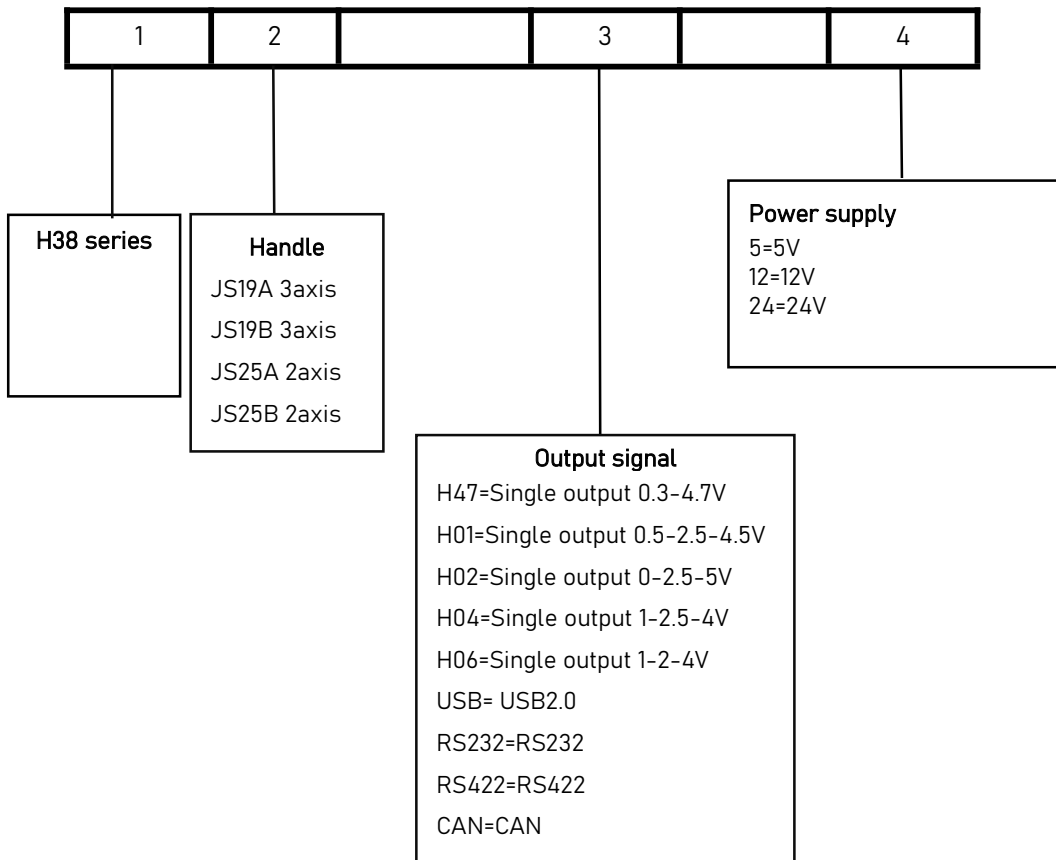
DESCRIPTION

H38 Series Hall joystick, 2-axis or 3-axis, instrument panel installation method, stainless steel and aluminum alloy materials, spring automatic return structure, German high-precision Hall sensor, linear correction in the full temperature range, IP65 or above protection level, Smoother operating feel, ergonomic mechanical design.

SPECIFICATION

Material	stainless steel + aluminum alloy
Protection	IP65 above the part above the panel
Positioning	Spring returns automatically
Operating angle	XY axis $\pm 20^\circ$, Z axis $\pm 30^\circ$
Operating range	360 degrees in all direction
Sensor	Hall sensor
Power supply	DC5V, DC12-24V
Power consumption	9-25MA (5V power supply), the signal output and the number of axes
Signal output	Analog voltage, RS485, RS422, RS232, USB
Operating life	more than 5 million times
Working temperature	$-40^\circ\text{C} \sim +70^\circ\text{C}$
Storage temperature	$-50^\circ\text{C} \sim +80^\circ\text{C}$
Protection level	IP65 or above
Base Dimension	38.0 x 38.0 x 31.5
Installation Dimension	28.0 x 28.0

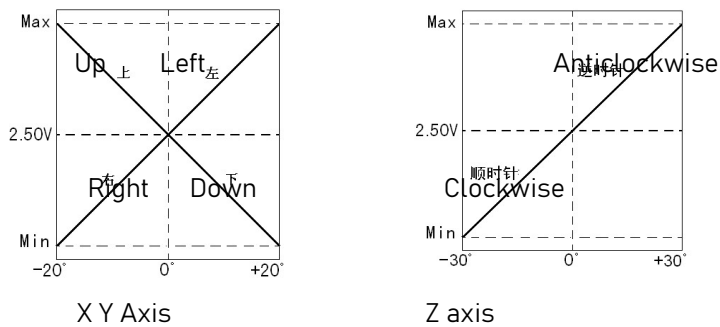
PRODUCT CONFIGURATION



Note:

- RS232, RS422, and other communication protocols, please consult with us
- CAN communication, baud rate, extension code and protocol can be customized
- Spring strength can be customized, please consult with us

Signal output of analog voltage



Schematic diagram of The button:



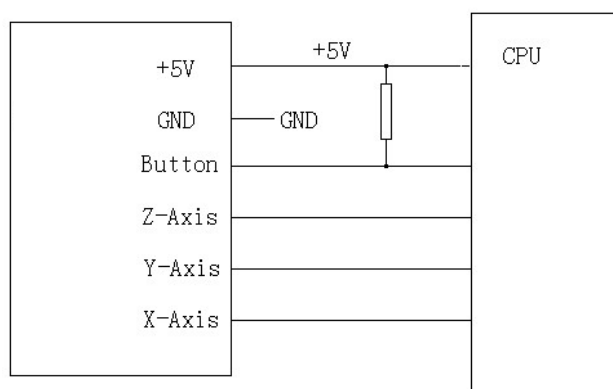
Electrical parameters of a button:

- ✧ Normal open
- ✧ Current: 50mA/5V
- ✧ IP Grade: IP66

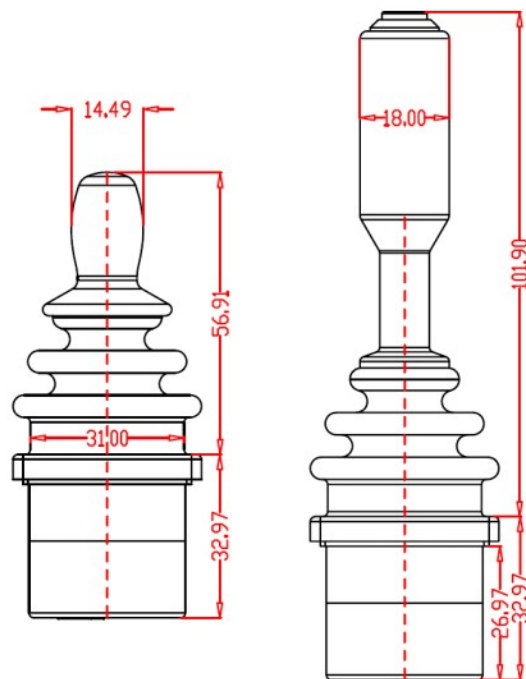
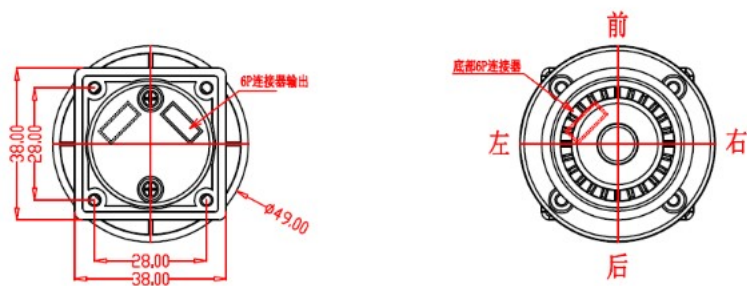
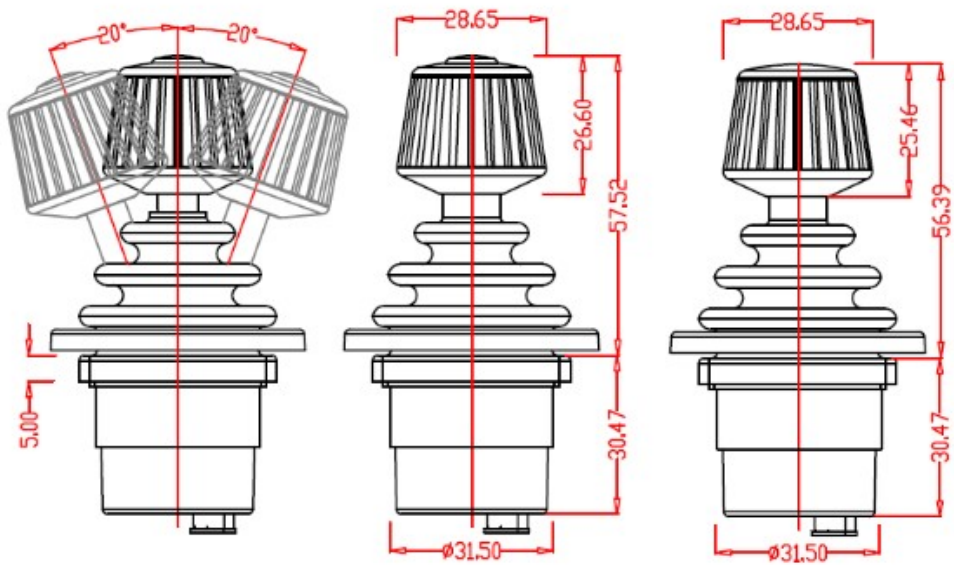
XYZ Electrical parameter:

- ✧ Minimum working voltage: 3.8V (5V PS) or 8V (12-24V PS)
- ✧ Maximum input voltage: 5.5V (5V PS) or 28V (12-24V PS)
- ✧ Working current: less than 10mA (5V PS, during signal output of analog voltage)
- ✧ Analog voltage signal output load: more than 1K Ω
- ✧ Central voltage of analog voltage signal output: 2.50V or 50%V_{dd}
- ✧ Analog voltage output signal: 0V~5V / 0.3V ~4.7V / 0.5V ~ 4.5V / 1V~ 4V

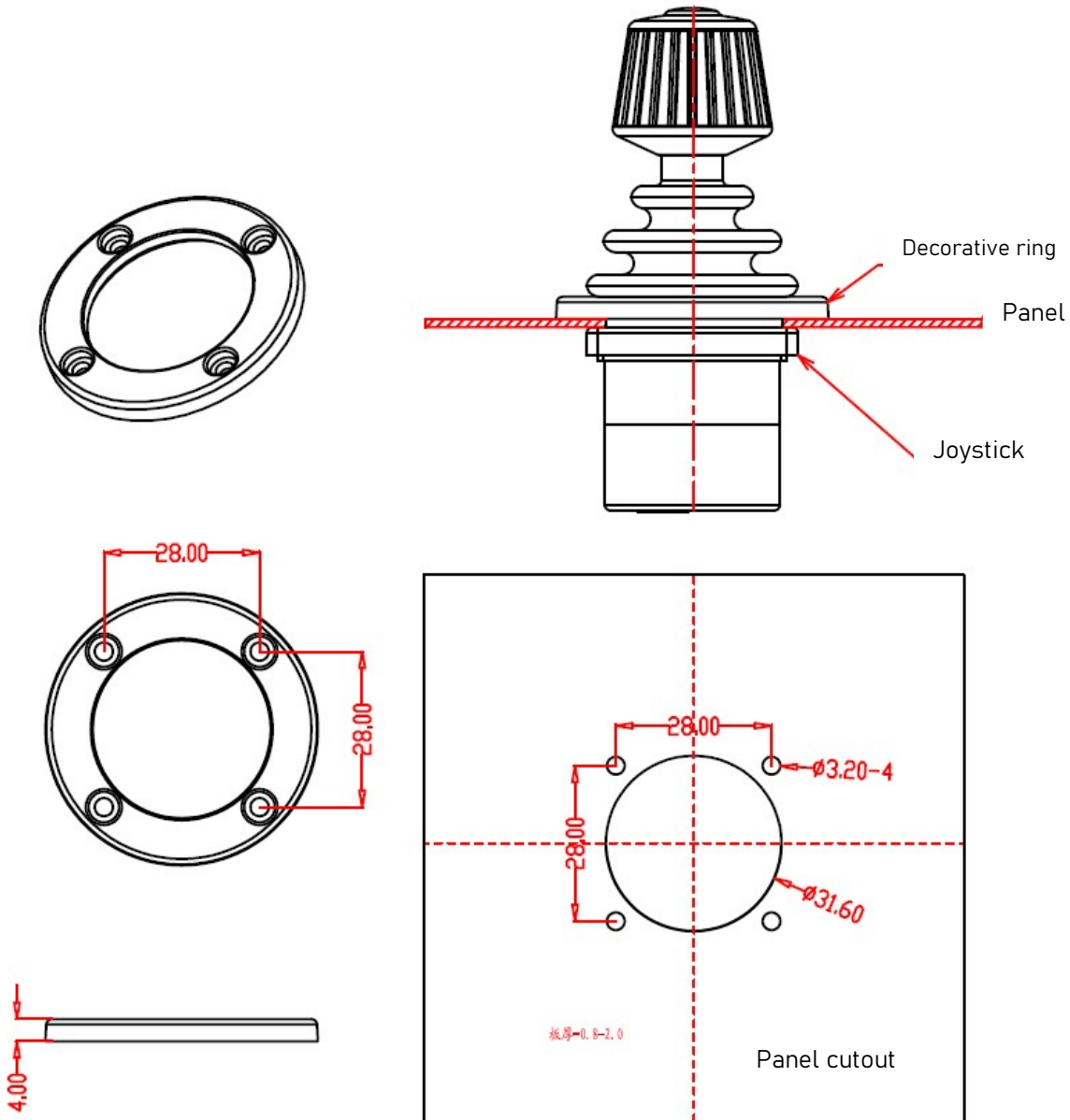
WIRING DIAGRAM



MECHANICAL DRAWING



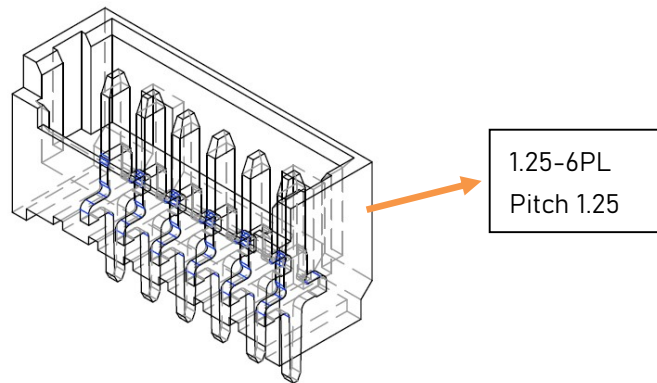
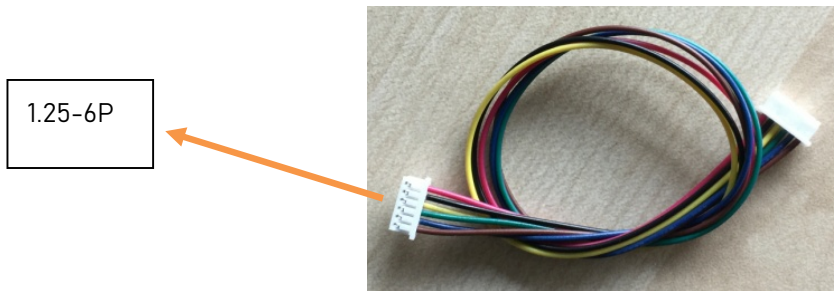
MOUNTING SIZE



WIRE LEADS

Lead length: 180mm (including terminals)

Terminal model: 1.25-6P



Connector Pin Definition - Analog Voltage Signal Output

Pin	Symbol	Color	Remark
1	+5V	Red	+5V (Power +)
2	GND	Black	GND (Power -)
3	Button	Yellow	Button
4	Z	Green	Z
5	X	Blue	X
6	Y	Brown	Y

Connector Pin Definition - RS232 Signal Output

Pin	Symbol	Color	Remark
1	+5V	Red	+5V (Power +)
2	GND	Black	GND (Power -)
3	NC	Yellow	
4	GND	Green	RS232 GND
5	RS232-TX	Blue	RS232-Transmit
6	RS232-RX	Brown	RS232-Receive

Connector Pin Definition - RS422 Signal Output

Pin	Symbol	Color	Remark
1	+5V	Red	+5V (Power +)
2	GND	Black	GND (Power -)
3	(RS422)RX+	Yellow	RS422 receive +
4	(RS422)RX-	Green	RS422 receive -
5	(RS422)TX+	Blue	RS422 transmit +
6	(RS422)TX-	Brown	RS422 transmit -

Connector Pin Definition - RS485 Signal Output

Pin	Symbol	Color	Remark
1	+5V	Red	+5V (Power +)
2	GND	Black	GND (Power -)
3		Yellow	
4		Green	
5	RS485 A+	Blue	RS485 A+
6	RS485 B-	Brown	RS485 B-

Connector Pin Definition - USB Signal Output:

Lead length: 1.6M

USB plug: TAPY-A



Pin	Symbol	Color	Remark
1	+5V	Red	USB Power +
2	D-	White	D-
3	D+	Blue	D+
4	GND	Black	GND Power-

Connector Pin Definition - CAN Signal Output

Pin	Symbol	Color	Remark
1	+5V	Red	+5V (Power +)
2	GND	Black	GND (Power -)
3	CAN-L	Yellow	CAN-L
4	CAN-H	Green	CAN-H
5	RS485 A+	Blue	RS485 A+
6	RS485 B-	Brown	RS485 B-

Note: RS485 is used to set internal parameters

CAN COMMUNICATION

CAN2.0B

- ✧ Can ID : Extended Frames ID, Standard Frames ID (the user is allowed to change ID through the rs485 Port)
- ✧ Interval 5-200ms scheduled sending, 20ms by default (can be modified through rs485)
- ✧ baud rate: 125K/250K/500K/1000K default=250K (change baud rate via RS485)

Data message format (HEX)

BYTE0	XXL X axis low	X axis data 0X01C0~0X0800~0X0E40
BYTE1	XXH X axis high	
BYTE2	YYL Y axis low	Y axis data 0X01C0~0X0800~0X0E40
BYTE3	YYH Y axis high	
BYTE4	0X00 Z axis low	Z axis data 0X01C0~0X0800~0X0E40
BYTE5	0X00 Z axis high	
BYTE6	Button	Button
BYTE7	0XA5	Tail

XXL	XXH	YYL	YYH	0X00	0X00	Button	A5
X low	X high	Y low	Y high	0X00	0X00	Button	Tail

YYYY Y axis angle

XXXX X axis angle

ZZZZ Z axis angle

Button Button

X axis data

MAX Left MIN	Stop	MIN Right MAX
0X01C0- -- -- 0X07ff	0800	0X0801- -- -- 0X0E40

Y axis data

MAX Lower MIN	Stop	MIN Upper MAX
0X01C0- -- -- 0X07ff	0800	0X0801- -- -- 0X0E40

Z axis data

MAX WIDE MIN	Stop	MIN TELE MAX
0X0100- -- -- 0X07ff	0800	0X0801- -- -- 0X0E40

Button data

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	0	Button	0	0000			

1: a button is pressed; 0: no button is pressed

E.g. 00 08 00 08 00 00 00 A5

RS232/RS422/RS485 COMMUNICATION PROTOCOL

I. Way of communication:

- If the data changes, it will be sent, and if it stops in one position, it will not send repeated data;
- Timing sending, can be sent regularly within 5-50MS, this time can be modified according to customer requirements

II. General protocol

(no address byte, applicable to RS232,RS422 and RS485) factory setting)

1. Joystick Transmit data (9 bytes joystick→PC) :

Baud rate: 9600.8.1.N

Data format (HEX)

FF	YYH	YYL	XXH	XXL	ZZH	ZZL	Button	CH
Head	Y high	Y low	X high	X low	Z high	Z low	button	checksum

YYYY Y axis angle

XXXX X axis angle

ZZZZ Z axis angle

BB joystick button

CH =XXH+XXL+YYH+YYL+ZZH+ZZL+Button (00-FF)

X axis data

MAX	Left	MIN	Stop	MIN	Right	MAX
0X0020-	-- --	0X01ff	0200	0X0201-	-- --	0X03E0

Y axis data

MAX	Lower	MIN	Stop	MIN	Upper	MAX
0X0020-	-- --	0X01ff	0200	0X0201-	-- --	0X03E0

Z axis data

MAX	WIDE	MIN	Stop	MIN	TELE	MAX
0X0020-	-- --	0X01ff	0200	0X0201-	-- --	0X03E0

Button data

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	0	Joystick button	0	0000			

Joystick button =1: a button is pressed; 0: no button is pressed;

E.G. FF 02 00 02 00 03 D0 00 07

2. Data sent by joystick with address bite (10 bytes) (joystick→PC)

Baud rate: 9600.8.1.N

FF	Addr	YYH	YYL	XXH	XXL	ZZH	ZZL	Button	CH
Head	Address	Y high	Y low	X high	X low	Z high	Z low	button	checksum

Adr address (01-7F)

YYYY Y-axis angle

XXXX X-axis angle

ZZZZ Z-axis angle

BB joystick button button on the joystick

CH =Adr+XXH+XXL+YYH+YYL+ZZH+ZZL+Button (00-FF) Low-order byte of the sum (1 byte)

E.g. FF 01 02 00 02 00 02 00 00 07

Address is 01 Checksum is 07

MODBUS COMMUNICATION PROTOCOL MODBUS (RTU MODE)

Master station mode of joystick:

1. Baud rate: 9600
2. Data bits: 1 starting bit, 8 data bits and 1 stop bit, no check bit
3. Communication interface: either RS485 or RS232
4. Data format: Modbus
5. Working mode: master station (the master station sends data to slave station 1)
6. Working mode: joystick->slave station

- ◆ Send data at a regular interval of 17ms between neighbouring frames, about 20HZ/frame;
- ◆ The frame interval can be changed by modifying the refresh rate (see 1.5 Refresh rate settings for details);
- ◆
- ◆ The slave station does not reply;

Function	Data	Data range
Equipment address	0x01	Modbus number
Function code	0x10	
Register 1 -high byte	0x40	Register address
Register 1 -low byte	0x01	
Number of register-high	0x00	
Number of register-low	0x04	
Data length	0x08	
Button high Bit15-Bit8	0x00	1-16 buttons (Bit0=button 1) 1=ON, 0=OFF
Button low Bit7-Bit0	0x00	
Joystick X axis high byte	0x02	0x0020-----0x01FF, 0x0100, 0x0201-----0x03E0 (max right min) Stop (min left max)
Joystick X axis low	0x00	
Joystick Y axis high	0x02	0x0020-----0x01FF, 0x0100, 0x0201-----0x03E0 (max down min) Stop (min Up max)
Joystick Y axis low	0x00	
Joystick Z axis high	0x02	0x0020-----0x01FF, 0x0100, 0x0201-----0x03E0 (max anticlockwise min) Stop (min clockwise max)
Joystick Z axis low	0x00	
CRC high	B4	
CRC low	43	

Send a frame / 45ms

E.g. 10 40 01 00 04 08 00 00 02 00 02 00 02 00 B4 43

SLAVE STATION MODE OF JOYSTICK

1. Baud rate: 9600/115200
2. Data bits: 1 starting bit, 8 data bits and 1 stop bit, no check bit
3. Communication interface: either RS485 or RS232
4. Data format: Modbus
5. Working mode: slave station
6. Working mode: master-slave mode (receive read command and return with 1 frame data)

Register address: 4001(HEX)

Modbus Read data format of host and answer format of slave (function code 03) (PC→joystick)

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Address	command	Start address		Number of register		CRC	
0x01	0x03	High	Low	High	Low	High	Low

Example: 01 03 40 01 00 04 00 09

When the joystick receives the command, the joystick changes from master mode to slave mode. (The joystick stops sending data actively and responses with below data when it receives the read data);

Response of joystick (joystick→PC)

Function	Data	Data range
Equipment address	0x01	Equipment address
Function code	0x03	
Data length	0x08	
Button high Bit15-Bit8	0x00	1-16 buttons (Bit0= button 1) 1=ON, 0=OFF
Button low Bit7-Bit0	0x00	
Joystick 1 X axis high	0x02	0x0020-----0x01ff, 0x0200, 0x0201-----0x03E0 (max Right min) Stop (min Left max)
Joystick 1 X axis low	0x00	
Joystick 1 Y axis high	0x02	0x0020-----0x01ff, 0x0200, 0x0201-----0x03E0 (max Lower min) Stop (min Upper max)
Joystick 1 Y axis low	0x00	
Joystick 1 Z axis high	0x02	0x0020-----0x01ff, 0x0200, 0x0201-----0x03E0 (max Anticlockwise min) Stop (min Clockwise max)
Joystick 1 Z axis low	0x00	
CRC high	94	
CRC low	ED	

E.g.

Host→joystick: Device address=1: 01 03 40 01 00 04 00 09

Joystick→host (response): 01 03 08 00 00 02 00 02 00 02 00 94 ED

Communication parameter of Joystick setting

Ver: 17.11.20

- ✧ Users can set and modify the communication parameters of the joystick (including CAN, RS232, RS422);
- ✧ All the above "parameters" can be modified only through RS422 or RS232 ports of the joystick, including CAN parameters.
- ✧ PC→joystick (RS422, RS485 or RS232) ,PC (serial debugging tool) software sends instructions to the joystick(RS422,RS485 or RS232).
- ✧ (If there is no serial debugging tool software, you can ask our technician for it.)
- ✧ If PC does not have COM port RS232 (DB9 9 pin head), you can use USB-RS232 converter (standard converter DB9 head, not TTL level converter)
- ✧ RS422, RS485 or RS232 communication interface on the joystick, default baud rate 9600.8.1.N

I. Basic parameter settings

(1). ACK Confirmation (joystick-PC)

AA 55 AF

It indicates that the joystick successfully receives instructions and executes them.

(2). Setting the joystick ID

ID is the Address in RS232/RS422 or CANopen communication protocol (PC-> joystick)

Format

0xaf 0x0d 00 00 00 Address 0xf5

Head Command Data1 Data 2 Data 3 Data 4 End

Add=0x01~0x7F address:1-127

Add=0x00 Null

E.g.

Setting ID=1: af 0d 00 00 00 01 f5 (HEX)
 Setting ID=2: af 0d 00 00 00 02 f5 (HEX)
 The Joystick return ACK

(3). Reset Joystick (PC->Joystick)

0xaf	0x15	00	00	00	Add	0xf5
Head	Command	Data1	Data 2	Data 3	Data 4	End
Add=0x01~0x7f						

(Only if the ADD has the same address as the joystick, it can reset the joystick.)

Add=0x00 Reset all joysticks
 Add out of range (0-0x7f) invalid

E.g.

Reset all joysticks: af 15 00 00 00 00 f5 (HEX)
 Reset joysticks(ID=1): af 15 00 00 00 01 f5 (HEX)
 Reset joysticks(ID=2): af 15 00 00 00 02 f5 (HEX)

(4). Setting the center position of the joystick

(Setting Joystick centered position) (PC - > joystick)

It has set up this feature when the factory. Users can ignore the instructions.

0xaf	0x09	00	00	00	00	0xf5
Head	Command	Data1	Data 2	Data 3	Data 4	End
Joystick receiving this data, it's current location as a center point						

E.g. af 09 00 00 00 00 f5 (HEX)

(5). Communication port selection: (PC - > joystick)

Communication ports RS232/RS422/CAN one of them
 (The factory has chosen this port)

0xaf	0x05	XX	00	00	00	0xf5
Head	Command	Data1	Data 2	Data 3	Data 4	End
XX=00 CAN Port;						
XX=01 RS232 Port;						
XX=02 RS422 Port;						
XX=03 RS485 Port; (RS232/422/485 Protocol)						
XX=04 RS485 Port Modbus RTU Protocol;						

E.g. af 05 00 00 00 00 f5 (HEX) CAN
 af 05 01 00 00 00 f5 (HEX) RS232
 af 05 02 00 00 00 f5 (HEX) RS422
 af 05 03 00 00 00 f5 (HEX) RS485
 af 05 04 00 00 00 f5 (HEX) RS485 Modbus RTU

(6).Refresh Rate Settings (PC -> joystick)

Refresh rate = the cycle time of sending 1 frame of data, such as setting 20 ms (sending 1 frame of data to the host every 20MS)

Format:

```
0xaf 0x11 00 00 00 Ref 0xf5
Head Command Data1 Data 2 Data 3 Data 4 End
```

Ref =0x0A~0x64 (10-100)ms, The unit is "milliseconds"

(Default: 20ms) Setting this parameter will take effect after reset or restart.

E.g. Set refresh rate = 20MS (send one frame per 20MS, send 50 times per second)

20MS af 11 00 00 00 14 f5 (HEX)

25MS af 11 00 00 00 19 f5 (HEX)

33MS af 11 00 00 00 21 f5 (HEX)

50MS af 11 00 00 00 32 f5 (HEX)

The joystick receives this instruction→ returns ACK→resets the joystick

Note: If the baud rate is low, the frame interval time will be longer

Default refresh rate 20ms (CAN baud rate 250K, RS232 and RS422 baud rate 9600)

(7).Communication mode(Including CAN and RS232/422)

Master-slave : 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"

The factory has been set up and stored permanently.

Format:

```
0xaf 0x08 00 00 00 Mode 0xf5
Head Command Data1 Data 2 Data 3 Data 4 End
```

Mode=00 Master (Timely Send)

Mode=01 Slave

E.g. (PC->Joystick)

Master mode: af 08 00 00 00 00 f5 (HEX)

Slave mode: af 08 00 00 00 01 f5 (HEX)

Joystick return ACK (AA 55 AF) (joystick->PC)

(8). Setting Number of axis of joystick: (PC-> joystick) only set by manufacturer

The factory has been set up and the user does not need:

E.g. (PC->Joystick)

2 axis: af 0c 00 00 00 00 f5 (HEX)

3 axis: af 0c 01 00 00 00 f5 (HEX)

4 axis: af 0c 02 00 00 00 f5 (HEX)

2. Set Communication parameter of RS232,RS422 and RS485

(9). Set the baud rates of RS232, RS422 and RS485 (PC - > joystick)

The RS232 and RS422 baud rates are the same, and the settings are valid at the same time.

0xaf	0x0b	00	00	00	Baud	0xf5
Head	Command	Data1	Data 2	Data 3	Data 4	End
	Baud=0X00	Baud rate=9600		Default		
	Baud=0X01	Baud rate =19200				
	Baud=0X02	Baud rate =57600				
	Baud=0X03	Baud rate =115200				
	Baud=0X04	Baud rate =2400				
	Baud=0X05	Baud rate =4800				

E.g. (PC->Joystick)

Baud rate =9600	af 0b 00 00 00 00 f5	(HEX)	Default
Baud rate =19200	af 0b 00 00 00 01 f5	(HEX)	
Baud rate =57600	af 0b 00 00 00 02 f5	(HEX)	
Baud rate =115200	af 0b 00 00 00 03 f5	(HEX)	
Baud rate =2400	af 0b 00 00 00 04 f5	(HEX)	
Baud rate =4800	af 0b 00 00 00 05 f5	(HEX)	

Joystick return ACK (joystick->PC)

(10). Inquire the position of the joystick (PC - > joystick)

This instruction is valid only when "slave mode"

When the joystick does not receive the inquiry instruction, it does not send data, and when it receives the inquiry instruction, the joystick returns a frame of data.

0xaf	0x07	00	00	00	Addr	0xf5
Head	Command	Data1	Data 2	Data 3	Data 4	End

◆ Addr = 0x01-0x7f

(PC->joystick) af 07 00 00 00 01 f5 (HEX)

(joystick->PC) FF 01 08 00 70 00 00 00 00 79

3. Parameter Setting of Remote Control Camera

(11). Set the address of the camera (PC - > joystick)

0xaf	0x0f	Cam	00	00	00	0xf5
Head	Command	Data1	Data 2	Data 3	Data 4	End
Cam=0x00-0x7f						
E.g. (PC->Joystick)						
		Camera ID=01:	af 0f 01 00 00 00 f5			
		Camera ID=02:	af 0f 02 00 00 00 f5			

(12). Setting up communication protocol (only for RS232, RS422, RS485)

(PC - > joystick)

0xaf	0x0e	Pro	00	00	00	0xf5
Head	Command	Data1	Data 2	Data 3	Data 4	End
Pro =0X00 Standard 9 bytes/10 bytes or Modbus protocol						
Pro =0X01 Camera remote control protocol PELCO-D						
Pro =0X02 Camera remote control protocol PELCO-P						
		Standard Protocol:	af 0e 00 00 00 00 f5	(HEX)	Default	
		Pelco-D Protocol:	af 0e 01 00 00 00 f5	(HEX)		
		Pelco-P Protocol:	af 0e 02 00 00 00 f5	(HEX)		

4. The parameter setting of CAN communication:

It is set by RS232 or RS422 port.

(13). Set CAN port baud rate: (PC->joystick)

0xaf	0x06	XX	00	00	00	0xf5
Head	Command	Data1	Data 2	Data 3	Data 4	End
XX=00 125K						
XX=01 250K (default)						
XX=02 500K						
XX=03 1000K						
E.g.	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			

(14).CAN protocol settings: (PC - > joystick)

The protocol defines the rules of CAN ID, and Data frames have no effect.

0xaf	0x0a	00	00	00	SS	0xf5
Head	Command	Data1	Data 2	Data 3	Data 4	End

SS = 00 Standard protocol, ID = setting ID (see (11) joystick sending node ID setting)

SS = 01 CANopen protocol ID = 180 + ID (see (2) Setting the joystick ID address)

The factory has been set up.

E.g. af 0a 00 00 00 00 f5 (HEX) Can Standard protocol,
 af 0a 00 00 00 01 f5 (HEX) CANopen Protocol

(15). Set the sending node ID of the joystick: (PC - > joystick)

It only applies to CAN Standard Protocol, but CANopen Protocol does not.

0xaf	0x01	D1	D2	D3	D4	0xf5
Head	Command	Data1	Data 2	Data 3	Data 4	End

D1.7=0 29 bits extended frame

D1.7=1 11 bits standard frame

- 29 bits extended frame: data range 0X0-0X0FFFFFFF, D1-D4 are corresponding to node identifications

E.g. Set the joystick's ID= identification-extended frame "0X00F0F101"

af 01 00 f0 f1 01 f5 (HEX)

- 11 bits standard frame: data range 0X000-0X3FF,D3-D4 are corresponding to node identifications

E.g. Set the joystick's ID= identification- standard frame "0X181"

af 01 80 00 01 81 f5 (HEX)

(16). Set the ID of the joystick to receive data: (PC - > joystick)

It is only used for "standard protocols" and CANopen protocol is not used.

0xaf	0x02	D1	D2	D3	D4	0xf5
Head	Command	Data1	Data 2	Data 3	Data 4	End

D1.7=0 29 bits extended frame

D1.7=1 11 bits standard frame

- 29 bits extended frame: data range 0X0-0X0FFFFFFF, D1-D4 are identification code(HEX)

E.g. Set the receiving node identification code-extended frame "0X00F0F101"

af 02 00 f0 f1 01 f5 (HEX)

- 11 bits standard frame: data range 0X000-0X3FF,D3-D4 are identification code(HEX);

E.g. Set the receiving node identification code- standard frame "0X1E1"

af 02 80 00 01 E1 f5 (HEX)

(17). Set the shielded node identification of joystick (PC->joystick)

0xaf	0x03	D1	D2	D3	D4	0xf5
Head	Command	Data1	Data 2	Data 3	Data 4	End

D1.7=0 29 bits of extended frame

D1.7=1 11 bits of standard frame

- 29 bits extended frame: data range 0X0-0X0FFFFFFF, D1-D4 are corresponding to node identification code;

E.g. Set the shielded node identification code-extended frame "0X00002201"

af 03 00 00 22 01 f5 (HEX)

- 11 bits standard frame: data range 0X000-0X3FF, D3-D4 are corresponding to node identification code;

E.g. Set the shielded node identification code-standard frame "0X122"

af 03 80 00 01 22 f5 (HEX)

5.Setting the communication parameters of Modbus RTU (RS485):

(18). Setting "Device Address": Default 1

Reference to "I,(2), Setting the ID Address of the joystick"

(19). Operating mode of joystick (PC-> joystick):

Reference to "I, (6), Communication Mode"

When in master mode, the joystick receives the correct "host reads data" instruction (01 03 40 01 00 04 0009), and the working mode is automatically changed to slave mode, but after restarting the joystick, it restores the master mode.

(20). Set the register address (PC - > joystick):

Data format:

0xaf	0x18	D1	D2	D3	D4	0xf5
Head	Command	Data1	Data 2	Data 3	Data 4	End

D1: Register address high byte

D2: Low byte register address

D3 and D4 = constant 0x00

Factory default register address = 0x4001

Permanent storage in joystick after installation

E.g.:

set the register address = 0x4001 (hexadecimal), if the octal to hexadecimal.

ID=0X4001 af 18 40 01 00 00 f5

After setting up a successful function, the joystick returns a "Reader Code" (with CRC Check) for testing.

01 03 40 01 00 04 00 09

Testing:

(PC Sent) 01 03 40 01 00 04 00 09

(Joystick return) 01 03 08 00 00 02 00 02 00 02 00 94 ED

USB interface & USB HID Protocol

USB 2.0 HID human-machine interface protocol standard

Microsoft operation system is supported, driving free; directX is supported, for related routine, inquire "joystick directx input" online;

I. Data format (7-byte HEX):

Angle of 3-axis joystick and keyboard status data sent by USB keyboard

byte1	byte2	byte3	byte4	byte5	byte6	byte7
XXL	XXH	YYL	YYH	ZZL	ZZH	BB1

XXXX: X axis data, 0000-03FF,(BYTE2 data is high, BYTE1 data is low)

0X0020-0X01FF LEFT

0X0200 Stop

0X0200-0X03E0 RIGHT

YYYY: Y axis data, 0000-03FF,(BYTE4 data is high,BYTE3 data is low)

0X0020-0X01FF LOWER

0X0200 Stop

0X0200-0X03E0 UPPER

ZZZZ: Z axis data, 0000-03FF,(BYTE6 data is high,BYTE5data is low)

0X0020-0X01FF anticlockwise (wide)

0X0200 Stop

0X0200-0X03E0 clockwise (tele)

BB1: the first group of buttons

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Button 8	Button 7	Button 6	Button 5	Button 4	Button 3	Button 2	Button 1

1. USB set center point command (send command through USB), the joystick receives the command from the host

f5 00 00 00 00 01 55 56

The factory has been set, the user can not use this command;

2. Set the low-order byte of the USB PID (send commands through USB)

Format: f5 0A Pid 00 00 00 CH (CH checksum: low-order byte of 2-6byte summation)

E.g.:

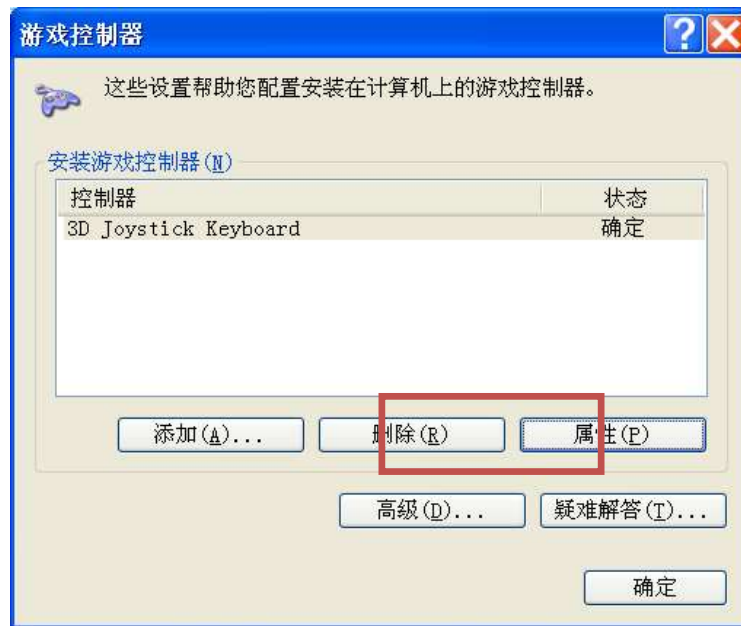
PID=0061 F5 0a 61 00 00 00 00 6b (default)

PID=0062 F5 0a 62 00 00 00 00 6c

PID=0063 F5 0a 63 00 00 00 00 6d

PID=0064 F5 0a 64 00 00 00 00 6e

PID=0065 F5 0a 65 00 00 00 00 6f



The joysticks are USB 2.0 HID compliant "game controllers", these utilize the DirectX API and are "plug-and-play" in MS Windows.