

H55 SERIES JOYSTICKS

Mate Joystick, Single axis joysticks, Panel Mounted



3- axis with one button
JS07 Handle



3-axis
JS30 handle



2-axis with one button
JS31 handle



2-axis
JS29 handle

DESCRIPTION

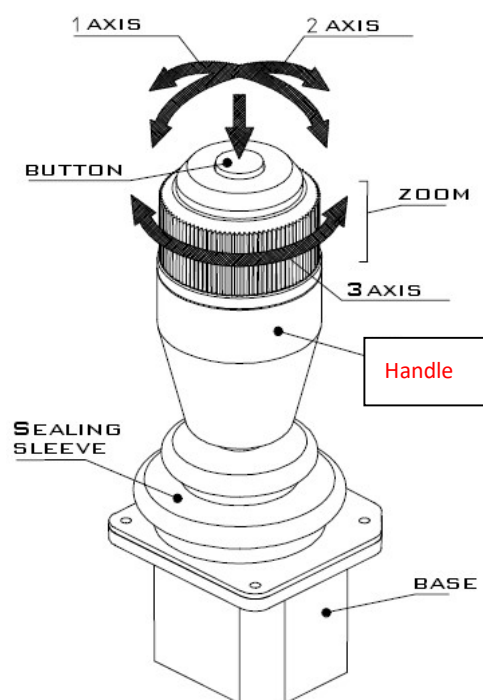
H55 contact-less joystick controller is designed for precision handle control applications where safety and long trouble-free life are primary requirements. It is available in two or three axis configurations and can accommodate a choice of handles, including push-button switch version, mounting in instrument board. The H55 series component of stainless steel and aluminum alloy, spring return and full temperature range linear correction, above IP54 protection. It is devised ergonomic handle with excellent operating experience.

SPECIFICATION

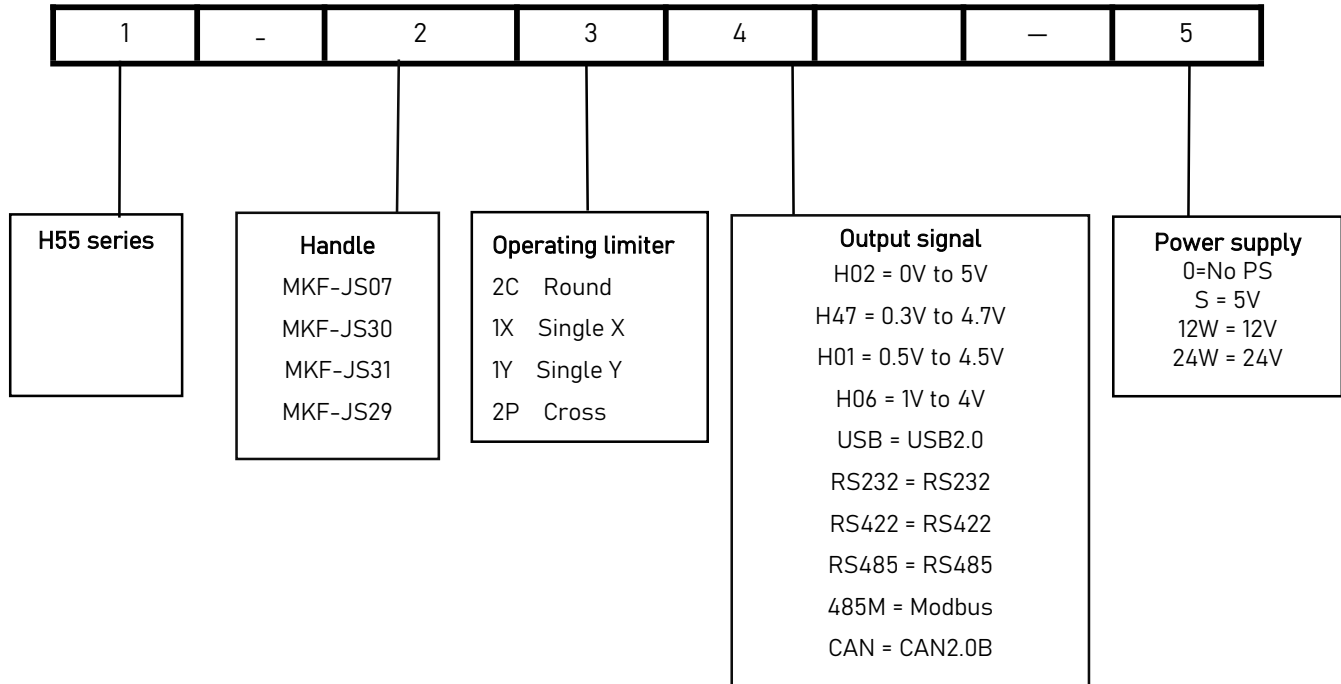
Material	Stainless steel + aluminum alloy +engineering plastics
Lever Action	Spring return
Lever Mechanical Angle	XY axis±25°, Z axis±36°
Sensor Type	Hall effect
Linearity	Less than 1%
Signal Output	Analog voltage 0-5V, analog current 0-20MA, RS485, RS422, RS232, USB, CAN
Operating force (X,Y axis)	0.98N{100gf}
Operating force (Z axis)	0.0067N.m{63gf.m}
Bearing force-max(X,Y axis)	120N(12.2kgf)
Bearing force-max(Z axis)	1.19N.m(12.2kgf.cm)
Power Supply	DC5V/12-24V
Operating Cycles	> 5 Million Cycles
Operating Temperature	-40℃~+70℃
Storage Temperature	-50℃~+80℃
Degree of Protection	Above the flange IP67
Base Dimension	55(L) x 55(W) x 38.5(H) mm
Weight	128g

Others:

Button: Switch value, press down it short-circuit with GND, When release it open-circuit with GND; Connect resistance (10K) when using. XYZ output load $\geq 1.0K\Omega$



PRODUCT CONFIGURATION

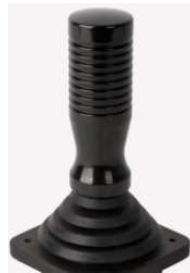


- When USB interface, no additional power supply
- A model handle only with all-round 3-axis, no optional for single and cross
- RS232,RS422, please contact factory for other communications protocol options
- CAN, baud rate, extended code and protocol can be customized
- Spring force: Available choices

HANDLE OPTIONAL



MKF-JS07 Handle
Material: Engineering plastic
1 button
3-axis



MKF-JS30 Handle
Material: Aluminium alloy
No button
2-axis



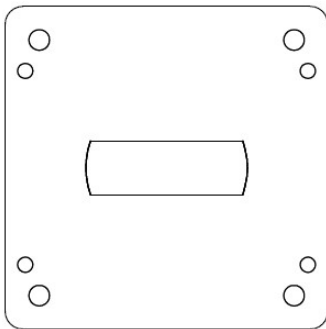
MKF-JS31 Handle
Material: Aluminium alloy
1 button
2-axis



MKF-JS29 Handle
Material: Aluminium alloy
No button
2-axis

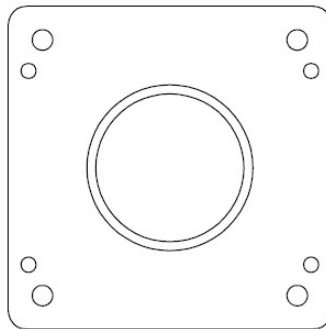
Lever gate profiles:

Gate (lever movement limiter)



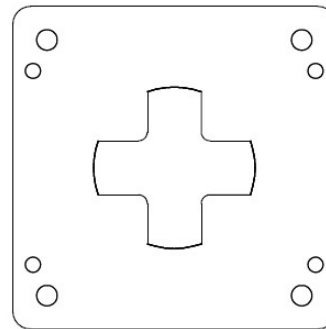
一字 (单轴)

1X/1Y= Single



全向

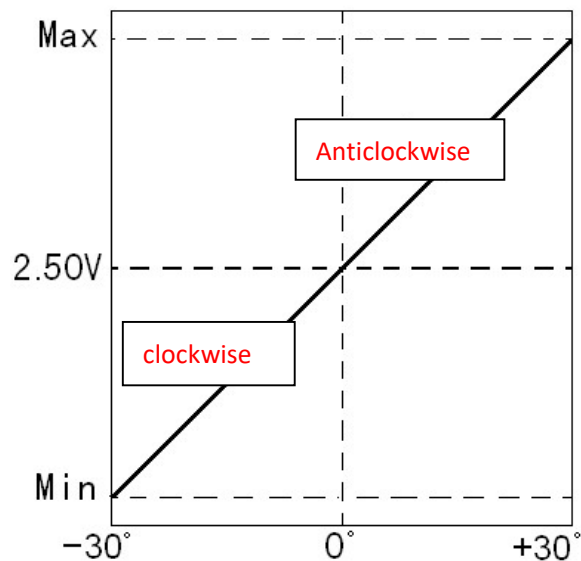
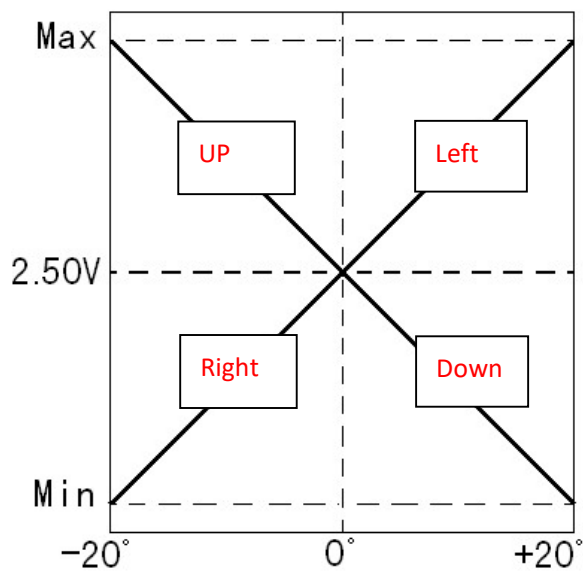
2C= Round



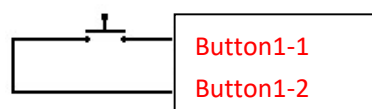
十字

2P= Cross

Analog voltage signal output:



Schematic diagram of The button:



Electrical parameters of a button: (Omron)

- Switch mode: Normal open
- Contact current: 50mA/5V
- IP Grade: IP40
- Operating cycles: >1 million operations

Electrical parameter:

Minimum working voltage: 3.05V (2 axes 5V PS), 7V(12-24V PS)

4.0V(3 axes 5V PS), 7V(12-24V PS)

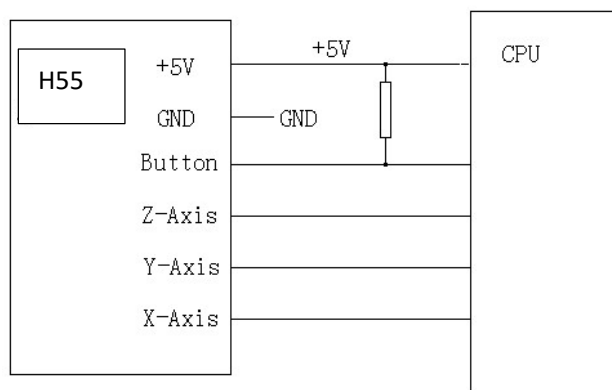
Maximum input voltage: 5.5V (5V PS),28V(12-24V PS)

Working current: 15mA(5V PS, during signal output of analog voltage)

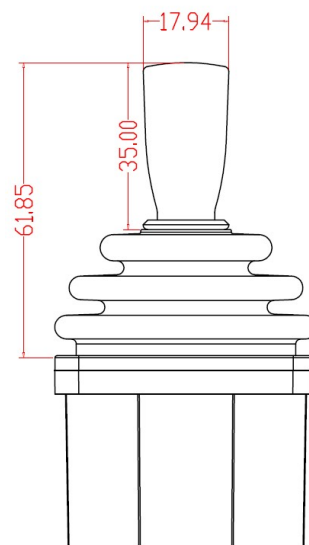
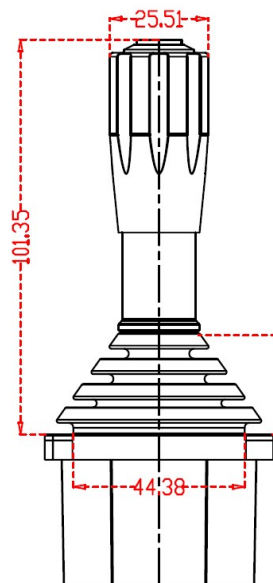
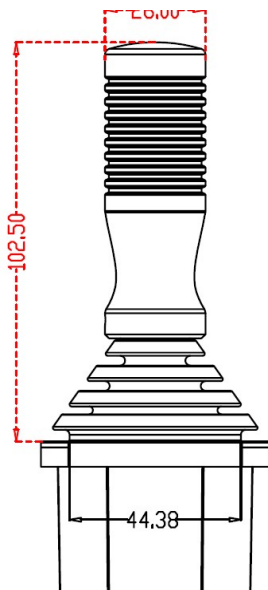
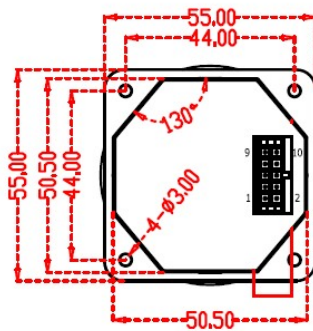
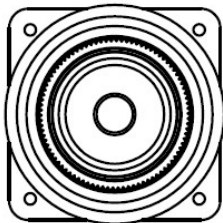
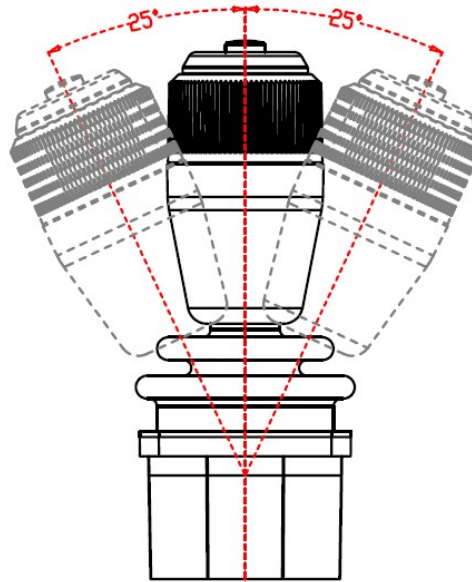
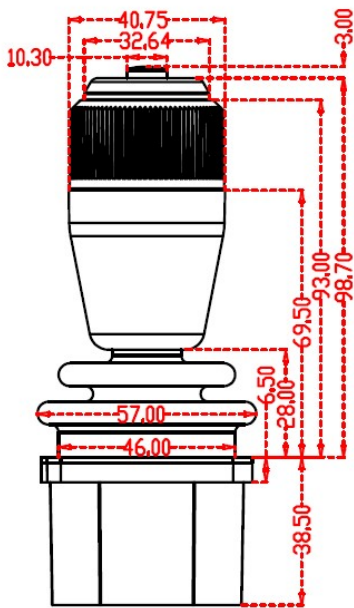
Analog voltage signal output load: >1K Ω

Central voltage of analog voltage signal output: 2.50V or 50%Vdd

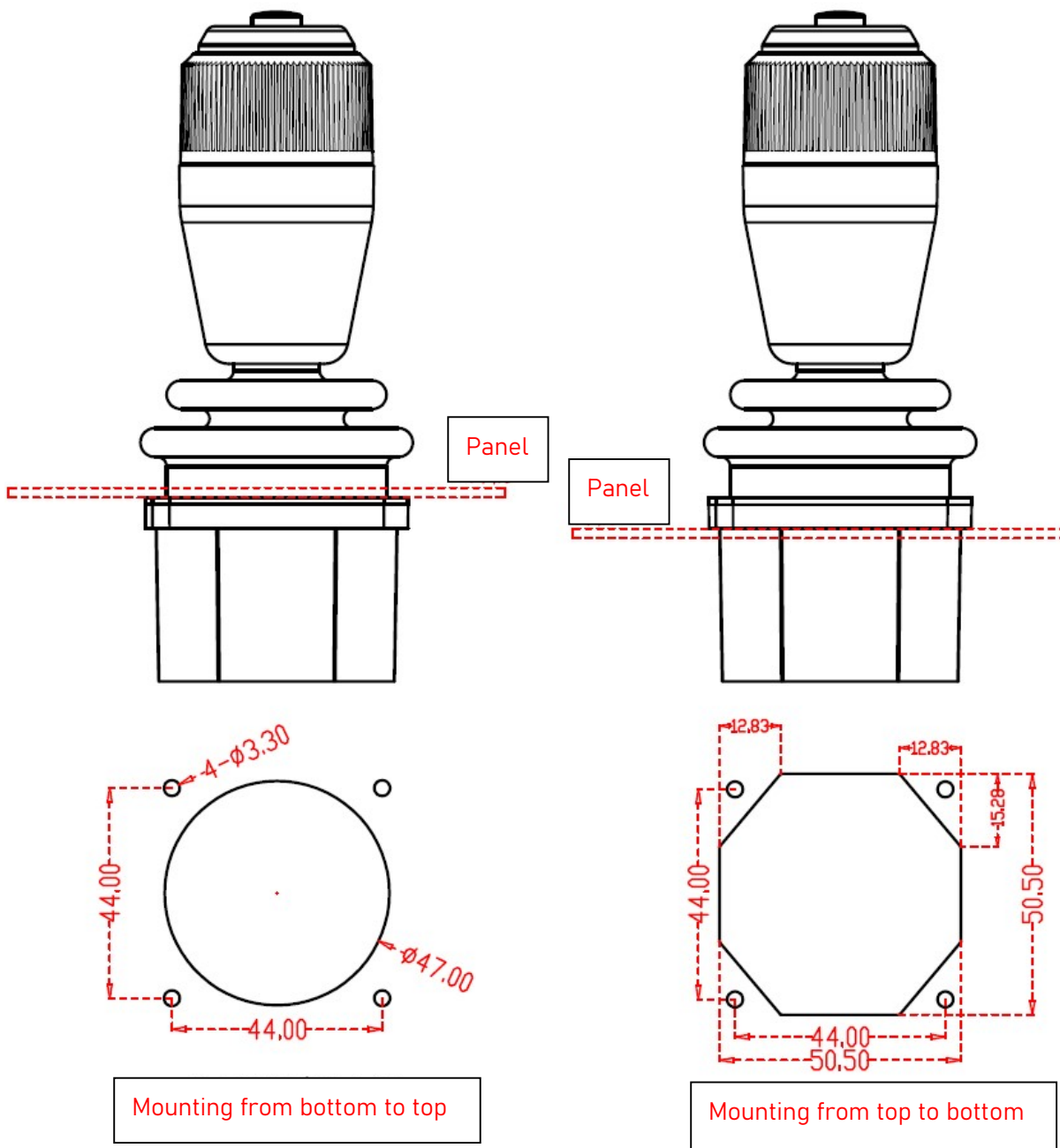
Wiring diagram



MECHANICAL DRAWING

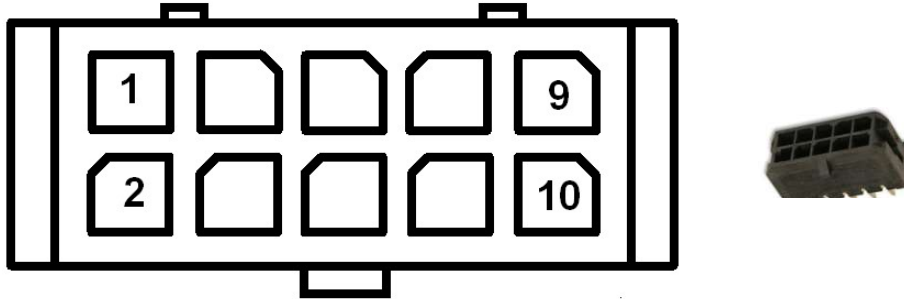


Mounting method:



Analog voltage signal output terminal diagram:

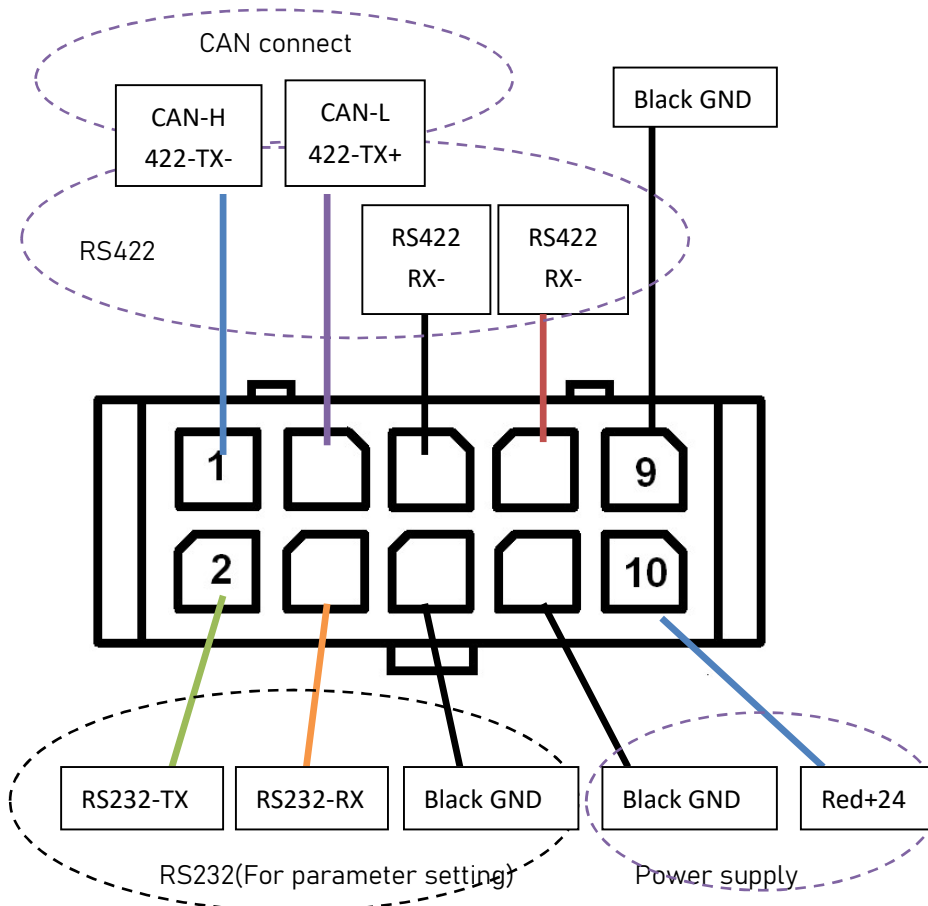
Connector model: Molex 43045-10000



Analogue voltage signal

Pin	Color	Function	Pin	Color	Function
1	White	Button 1-1	2		Null
3	White	Button 1-2	4		Null
5	Yellow	X Signal	6	Black	GND
7	Green	Y Signal	8	Black	Power-(GND)
9	Blue	Z Signal	10	Red	Power+(VCC)

Communication signal output terminal diagram



CAN signal

Pin	Color	Function	Pin	Color	Function
1	Blue	CAN-H/422TX-	2	Green	RX232-TX
3	White	CAN-L/422TX+	4	Yellow	RX232-RX
5		422RX-	6	Black	GND
7		422RX+	8	Black	Power-(GND)
9	Black	GND	10	Red	Power+(5 or12-24V)

RS422 signal output interface

Pin	Color	Function	Pin	Color	Function
1	Blue	RS422TX-	2		RX232-TX(CAN parameter setting)
3	White	RS422TX+	4		RX232-RX(CAN parameter setting)
5	Green	RS422RX-	6		GND
7	Yellow	RS422RX+	8	Black	Power -(GND)
9	Black	GND	10	Red	Power +(5-24V)

RS232 signal

Pin	Color	Function	Pin	Color	Function
1			2	Yellow	RX232-TX(CAN parameter setting)
3			4	Green	RX232-RX(CAN parameter setting)
5			6	Black	GND
7			8	Black	Power -(GND)
9	Black	GND	10	Red	Power +(5-24V)

Pin definition of connector -USB signal output:

Wire lead length 1.6M

USB plug: TAPE-A



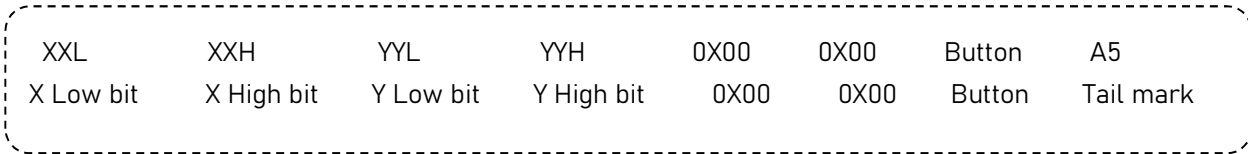
Pin	Symbol	Color	Function
1	+5V	Red	USB power+
2	D-	White	Data -
3	D+	Blue	Data +
4	GND	Black	USB power-GND

CAN bus communication:

- CAN2.0B
- Extended data frame: Default ID= extended code 0X00F0F101 (can modify these identification code via RS232)
- Baud rate: Default=250K (can modify baud rate via RS232)
- Transmit mode: Interval 45ms

1. Transmit data format: (16 system HEX)

BYTE0	XXL X axis low	X axis data 0X0080~0X0800~0X0F80
BYTE1	XXH X axis high	
BYTE2	YYL Y axis low	Y axis data 0X0080~0X0800~0X0F80
BYTE3	YYH Y axis high	
BYTE4	ZZL Z axis low	Z axis data 0X0080~0X0800~0X0F80
BYTE5	ZZH Z axis high	
BYTE6	Button	Button
BYTE7	0XA5	Tail mark



YYYYY axis angle

XXXX X axis angle

ZZZZ Z axis angle

Button Button

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

X axis parameter

MAX	Left	MIN	Stop	MIN	Right	MAX
0X0080-	-- --	0X07ff	0800	0X0801-	-- --	0X0F80

Y axis parameter

MAX	Down	MIN	Stop	MIN	Up	MAX
0X0080-	-- --	0X07ff	0800	0X0801-	-- --	0X0F80

Z axis parameter

MAX	WIDE	MIN	Stop	MIN	TELE	MAX
0X0080-	-- --	0X07ff	0800	0X0801-	-- --	0X0F80

Button parameter

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

1: button pressed, 0: non-button pressed

H55 Series RS232/RS422/485 communication protocol

Communication mode:

- Data varies then transmit, It keeps the same position then stops sending repetitive data
- Regular transmission, during 10-50MS, Time can be adjusted by user
<Noted communication mode when purchase>

General protocol

(Non-address bit, for RS232,RS422,RS485) factory default setting)

1. Joystick transmit data (9 byte) (joystick-PC)

Function: Transmit each axis location parameter of joystick

Baud rate: 9600.8.1.N

Joystick transmit data format: (16 system HEX)

FF	YYH	YYL	XXH	XXL	ZZH	ZZL	Button	CH
Head	Y high bit	Y low bit	X high bit	X low bit	Z high bit	Z low bit	Button	Checksum

YYYYY 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 parameter

MAX	Left	MIN	Stop	MIN	Right	MAX
0X0020-	-- --	0X01ff	0200	0X0201-	-- --	0X03DF

Y axis parameter

MAX	Down	MIN	Stop	MIN	Up	MAX
0X0020-	-- --	0X01ff	0200	0X0201-	-- --	0X03DF

Z axis parameter

MAX	WIDE	MIN	Stop	MIN	TELE	MAX
0X0020-	-- --	0X01ff	0200	0X0201-	-- --	0X03DF

Button parameter

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

Joystick button =1: button pressed, 0: non-button pressed

E.g. : FF 02 00 02 00 03 D0 00 D7

Modbus communication protocol Modbus (RTU mode)

Master station mode of joystick:

- Baud rate: 9600
- Data bits: 1 start bit, 8 data bits, 1 stop bit, no check bit
- Communication interface: Either RS485 or RS232
- Data format: Modbus
- Operating mode: Master station (the master station sends data to slave station 1)
- Operating mode: Joystick->slave station
- Transmit data frame regular, frame interval 17ms, about 20HZ/frame
- The slave station does not reply

Function	Data	Parameter range
Device address	0x01	Modbus station number
Function code	0x10	
1st register address- high-order	0x40	Register address
1st register address- low-order	0x01	
Number of register- high-order	0x00	
Number of register- low-order	0x04	
Data length	0x08	
Button high-order Bit15-Bit8	0x00	1-16 buttons (Bit0=button 1) 1=ON, 0=OFF
Button low-order Bit7-Bit0	0x00	
Joystick 1 X axis high-order	0x08	0x0081-----0x07FF, 0x0800, 0x0801-----0x0F7F (max Left min) Stop (min Right max)
Joystick 1 X axis low-order	0x00	
Joystick 1 Y axis high-order	0x08	0x0081-----0x07FF, 0x0800, 0x0801-----0x0F7F (max Down min) Stop (min Up max)
Joystick 1 Y axis low-order	0x00	
Joystick 1 Z axis high-order	0x08	0x0081-----0x07FF, 0x0800, 0x0801-----0x0F7F (max Anticlockwise min) Stop (min clockwise max)
Joystick 1 Z axis low-order	0x00	
CRC high bit	B1	
CRC low bit	91	

45ms send one frame regularly

Joystick slave station mode:

(Master-slave mode)

- Baud rate: 9600/115200
- Data bits: 1 start bit, 8 data bits, 1 stop bit, no check bit
- Communication interface: Either RS485 or RS232
- Data format: Modbus
- Operating mode: Slave station
- Operating mode: Master-slave mode(receive read command, return with 1 frame data)
- Register address: 60000

Modbus the format of mast read data and slave response (function code 03) (PC→Joystick)

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Address	Command	Begin address		Register number		CRC	
0x01	0x03	High	Low	High	Low	High	Low

E.g.: 01 03 60 00 00 04 5A 09

When the joystick receives the command, then the joystick switches to slave mode from master mode (Stop send data actively, joystick respond below data only when receiving this read data.)

Response of joystick (joystick→PC)

Function	Data	Parameter range
Device address	0x01	Device address
Function code	0x03	
Data length	0x08	
Button high-order Bit15-Bit8	0x00	1-16 Buttons (Bit0=Button 1) 1=ON, 0=OFF
Button low-order Bit7-Bit0	0X00	
Joystick 1 X axis high-order	0x08	0x0081-----0x07ff, 0x0800, 0x0801-----0x0F7F (max Left min) Stop (min Right max)
Joystick 1 X axis low-order	0x00	
Joystick 1 Y axis high-order	0x08	0x0081-----0x07ff, 0x0800, 0x0801-----0x0F7F (max Down min) Stop (min Up max)
Joystick 1 Y axis low-order	0x00	
Joystick 1 Z axis high-order	0x08	0x0081-----0x07ff, 0x0800, 0x0801-----0x0F7F (max Anticlockwise min) Stop (min clockwise max)
Joystick 1 Z axis low-order	0x00	
CRC high-order	91	
CRC low-order	3F	

E.g.:

PC→Joystick: Device address =1: 01 03 60 00 00 04 5A 09

Joystick→PC(Response): 01 03 08 00 00 08 00 08 00 08 00 91 3F

(PC→Joystick :Device address =2:02 03 60 00 00 04 5A 3A)

Joystick communication parameter setting

Ver: 17.11.20

Users can be set and adjust the communication parameters of the joystick (including CAN, RS232, RS422)
All the above "parameters" can be adjusted only through RS422 or RS232 ports of the joystick, including CAN parameters.

PC→Joystick (RS422, RS485 or RS232) PC (serial assistant) software send instruction to joystick

If no RS232 in PC (DB9 9 pin connector), then USB to RS232 convertor (standard convertor, not TTL lever)

RS422,RS485 or RS232 communication interface of joystick, default baud rate 9600.8.1.N

Basic instruction:

1 ACK confirmation (Joystick-PC)

AA 55 AF

It indicates that the joystick successfully receives instructions and executes them

2 Joystick ID address setting;

ID is in RS232/RS422 communication protocol, or CAN open

(PC→Joystick)

0xaf 0x0d 00 00 00 Add 0xf5

Head Command Data1 Data2 Data3 Data4 Tai

Add=0x01~0x7F Address 1-127

Add=0x00 Invalid

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 It can reset only when ADD same with joystick ADD

Add=0x00 Reset all joystick

Add Out of rang (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 (PC-> joystick)

This command is set up in factory, user can ignore it.

PC connect with RS422, baud rate 9600

0xaf	0x09	00	00	00	00	0xf5
Head	Command	Data1	Data2	Data3	Data4	End

Transmit these data to joystick, re-set stop position of joystick

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

5 Communication port selection: (PC->Joystick)

Joystick communication port RS232, RS422,CAN(select one); (Set in factory)

0xaf	0x05	XX	00	00	00	0xf5
Head	Command	Data1	Data2	Data3	Data4	End

XX=00 CAN port

XX=01 RS232 port

XX=02 RS422 port

XX=03 RS485 port (standard RS232/422/485 protocol)

XX=04 RS485 Modbus RTU protocol

E.g.:

af 05 00 00 00 00 f5	(HEX)	CAN port
af 05 01 00 00 00 f5	(HEX)	RS232 port
af 05 02 00 00 00 f5	(HEX)	RS422 port
af 05 03 00 00 00 f5	(HEX)	RS485 port (standard 485 protocol)
af 05 04 00 00 00 f5	(HEX)	RS485 Modbus RTU protocol

6 Refresh rate setting(PC->Joystick)

Refresh rate=frame interval of send data, e.g. setting 20ms(per 20MS send one frame data to master)

0xaf	0x11	00	00	00	Ref	0xf5
Head	Command	Data1	Data2	Data3	Data4	Tail

Ref =0x0A~0x64 (10-100)ms, Units is“Millisecond”; (Default:20ms)

Setting up this parameter will take effect after reset or restart

E.g.: Set refresh rate = 20MS (send one frame/ 20MS, send 50 times/ 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 instruction→reply ACK→resets the joystick

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

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

7 Communication mode (master-slave query, timed auto-send, include CAN and RS232/422) (PC->Joystick)

Master-slave query: Joystick is slave device, It will transmit data to master only when received the query command of master

Timed auto-send: Joystick will send data to master once power-up, sending rate reference "refresh rate setting"

The factory has been set up and stored permanently.

Format:

```

0xaf    0x08    00    00    00    Mode    0xf5
Head    Command  Data1  Data2  Data3  Data4    Tail
  
```

Mode=00 Timed send(master mode)

Mode=01 Master-slave query(slave mode)

Slave mode: Master needs to send query command for the joystick to respond

E.g.: (PC->joystick)

Auto-send mode(master mode) af 08 00 00 00 00 f5 (HEX)

Master-slave query mode(slave mode) af 08 00 00 00 01 f5 (HEX)

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

8 Number of joystick signal axis: (PC->Joystick) (Only for factory)

The factory has been set up, users no need set:

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

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

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

9 Query device information: (PC->Joystick)

af 20 00 00 00 00 f5 (HEX)

Reply format of joystick: Joystick->PC

```

FF    45    18    18 01 20    53 4D 43 34 35 41    ch
Head  Model Ver    Date      S M C 4 5 A    Checksum
  
```

Checksum = The sum of all the previous bytes except the head (FF), take the lower byte of the result

Set communication parameter of RS232, RS422 and RS485

10 Set baud rate of RS232, RS422 and RS485(PC->Joystick)

RS232 and RS422 are same baud rate, settings are valid concurrence

0xaf	0x0b	00	00	00	Baud	0xf5
Head	Command	Data1	Data2	Data3	Data4	End
	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)
Set succeed then joystick return ACK
```

11 Inquire the position of the joystick (PC-> Joystick)

This instruction is valid only when "master-slave query 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	Add	0xf5	
Head	Command	Data1	Data2	Data3	Data4	End
Addr =	0x01-0x7f	If address is correct, return send				

Joystick return send current position when receive this data

E.g. RS232 communication inquiry:

```
(PC->Joystick) af 07 00 00 00 01 f5  (HEX)
(Joystick ->PC) FF 01 08 00 70 00 00 00 00 79
Joystick return send current position when receive this data
```

CAN parameter setting:

CAN parameter setting via RS232 or RS422 port

12 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

13 CAN protocol setting: (PC->Joystick)

0xaf	0x0a	00	00	TP	SS	0xf5
Head	Command	Data1	Data2	Data3	Data4	Tail

SS=00 Standard protocol ID=Send node ID, Ref . (1) Joystick sending node ID default

SS=01 CAN open protocol ID=180+ID Ref.(2)Set joystick ID address
(The factory has set up)

TP in CAN OPEN protocol: TPDO

TP=00: TPDO1 send ID 0X0180+ID(Ref joystick ID address setting) default
TP=01: TPDO2 send ID 0X0280+ID(Ref joystick ID address setting)
TP=02: TPDO3 send ID 0X0380+ID(Ref joystick ID address setting)
TP=03: TPDO4 send ID 0X0480+ID(Ref joystick ID address setting)

Reg.: af 0a 00 00 00 00 f5 (HEX) Standard protocol
af 0a 00 00 00 01 f5 (HEX) CAN open protocol TPDO1

14 Joystick sending node ID setting: (PC->Joystick)

Only for "standard protocol", not for CAN open

0xaf	0x01	D1	D2	D3	D4	0xf5
Head	Command	Data1	Data2	Data3	Data4	Tail

D1.7=0 29 bits extend frame

D1.7=1 11 bits standard frame

- 29 bits extend frame: Data range 0X0-0X0FFFFFFF, Data D1-D4 corresponding "identification code"

E.g.: Set sending node identification code- Extend frame"0X00F0F101"

af 01 00 f0 f1 01 f5 (HEX)

- 11 bits standard frame: Data range 0X000-0X3FF, Data D3-D4 corresponding "Node identification code"

E.g.: Set sending node identification code - standard frame"0X181"

af 01 80 00 01 81 f5 (HEX)

- 15 Joystick “Receiving node ID” setting: (PC->Joystick)
Only for “standard protocol”, Not for CAN Open

0xaf	0x02	D1	D2	D3	D4	0xf5
Head	Command	Data1	Data2	Data3	Data4	Tail

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

- 29 bits extend frame: Data range 0X0-0X0FFFFFFF, data D1-D4 corresponding “identification code”

E.g.: Set receiving node identification code-extend frame“0X00F0F101”

af 02 00 f0 f1 01 f5 (HEX)

- 11 bits standard frame: Data range 0X000-0X3FF,data D3-D4 corresponding “Node identification code”

E.g.: Set receiving node identification code –standard frame“0X1E1”

af 02 80 00 01 E1 f5 (HEX)

- 16 Joystick “Shield node ID”setting: (PC->Joystick)

0xaf	0x03	D1	D2	D3	D4	0xf5
Head	Command	Data1	Data2	Data3	Data4	End

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

- 29 bits extend frame: Data range 0X0-0X0FFFFFFF, Data D1-D4 corresponding “identification code”

E.g.: Set Shield node identification code –extend frame“0X00002201”

af 03 00 00 22 01 f5 (HEX)

- 11 bits standard frame: Data range 0X000-0X3FF, data D3-D4 corresponding “identification code”

E.g.: Set Shield node identification code –standard frame“0X122”

af 03 80 00 01 22 f5 (HEX)

Communication parameters of Modbus RTU (RS485) setting:

(Modify time: 2018-01-19)

- 17 Set"Device address": factory default 1
Ref.(Set ID Address of the joystick)
- 18 Joystick operating mode (PC->Joystick)
Ref.(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. After restarting the joystick, it restores the master mode.

- 19 Register address setting (PC->Joystick):

Data format:

0xaf	0x18	D1	D2	D3	D4	0xf5
Head	Command	Data1	Data2	Data3	Data4	Tail

D1: Register address high byte

D2: Register address low byte

(D3&D4=0x00)

Register address default by factory=0x4001

After setting, it is permanently stored in joystick

E.g.: set the register address =0x4001(hexadecimal), if octal ,change to hexadecimal

ID=0X4001 af 18 40 01 00 00 f5

USB interface

USB communication protocol: USB 2.0 HID human-machine interface protocol standards, Support Microsoft , driving free; Support direct X .For related routine, Search online“joystick direct input”

Data format of USB keyboard sending (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 high-order,BYTE1 data low-order)

0X0020-0X01FF Left
0X0200 Stop
0X0201-0X03DF Right

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

0X0020-0X01FF Down
0X0200 Stop
0X0201-0X03DF Up

ZZZZ: Z axis data,,0000-03FF,(BYTE6 data high-order,BYTE5 data low-order)

0X0020-0X01FF Anticlockwise(wide)
0X0200 Stop
0X0201-0X03DF Clockwise(tele)

BB1:1st group of button

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

(Note:The company reserves the right to change specifications without notice.)

