MKF-JS12

Mate Joystick, Multi-axis joysticks, Left and Right hand, Panel Mounted

MKF-JS12A
6 pushbuttons

MKF-JS12B
3 pushbuttons
1 rocker switch

MKF-JS12C
2 pushbuttons
1 thumbwheel

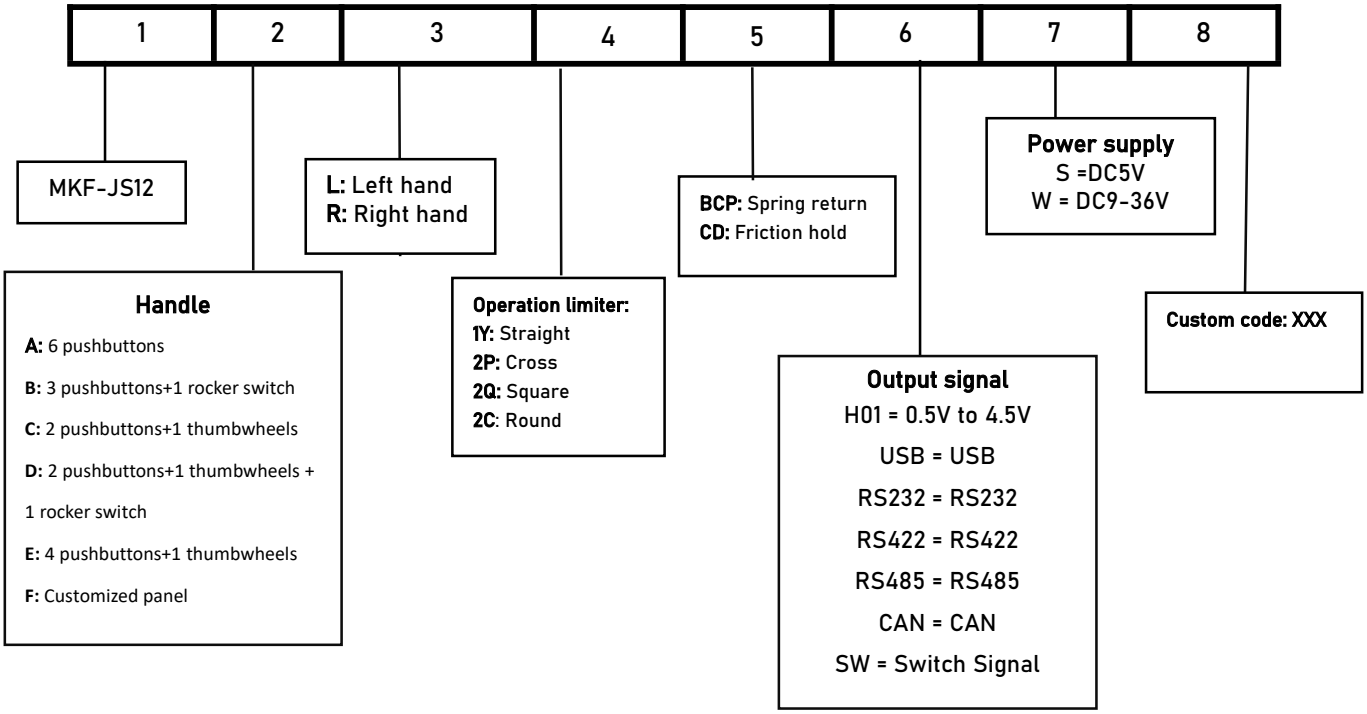
MKF-JS12D
2 pushbuttons
1 thumbwheel
1 rocker switch

MKF-JS12E
4 pushbuttons
1 thumbwheel

DESCRIPTION

Spring return, Friction positioning and friction+spring return
Developed for applications where ergonomics and system integrity are paramount, the MKF-JS12 is a multi-axis Hall joystick, with component of stainless steel and aluminum alloy, spring return, and full temperature range linear correction. It is sealed to IP66 to enable it to operate in extreme environments.

PRODUCT CONFIGURATION



HANDLE OPTIONAL

MKF-JS12A

MKF-JS12B

MKF-JS12C

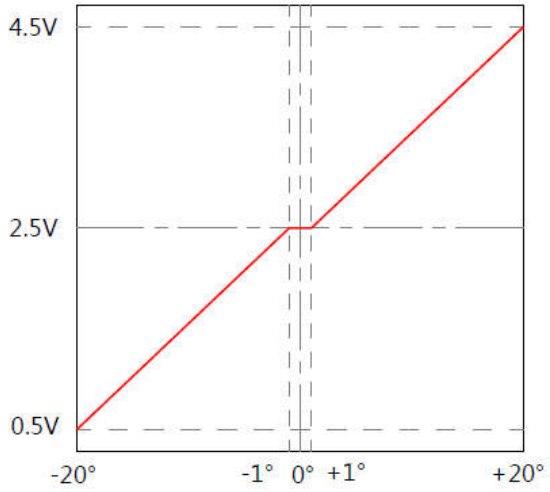
MKF-JS12D

MKF-JS12E

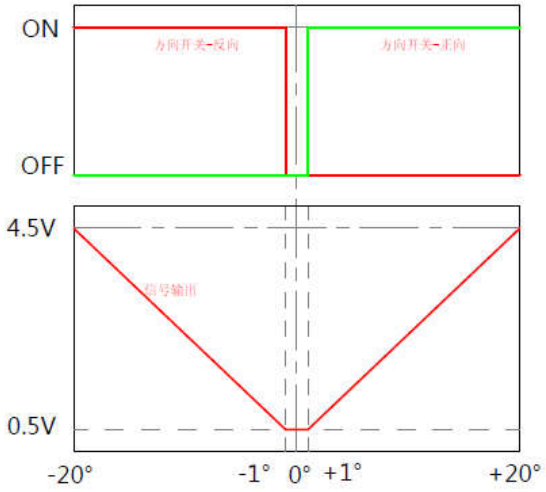
SPECIFICATION

Material	Stainless steel + aluminum alloy
Lever Action	Spring return / Friction hold / Front and back friction positioning / Left and right spring return
Lever Mechanical Angle	Either ±30° or ±20°
Sensor Type	Hall effect
Operating Range	1 axis, 2 axis, 3 axis, multiple axis
Signal Output	Analog voltage, RS422, RS485, RS232, CAN, USB, Switch
Power Supply	DC5V/ DC9-36V
Operating Cycles	> 10 Million Cycles
Operating Temperature	-40℃~+70℃
Storage Temperature	-50℃~+80℃
Degree of Protection	Above the flange IP65
Power Consumption	Less than 32mA
Electronic Parameters	
Lowest Working Voltage	4.2V(5V) / 9V (9-36V)
Max. Input Voltage	36V(9-36V) / 5.5V(5V)
Working Current	Less than 50mA (5V power supply)
Button Switch Capacity	1A/24V
Limit Switch Capacity	1A/24V
Analog Voltage Signal Output Load	> 1KΩ
Analog Voltage Signal Output Center Voltage	2.50V or 50%Vdd
Analog Voltage Output Signal	0.3-4.7V/0.5V-4.5V/1V-4V

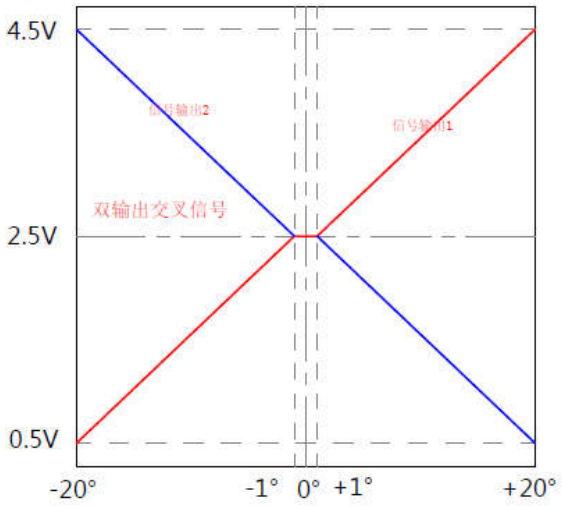
VOLTAGE SIGNAL OUTPUT CURVE



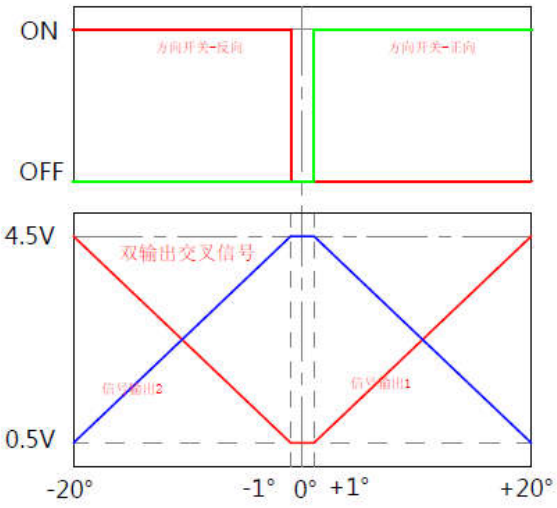
Voltage signal: Output curve 1



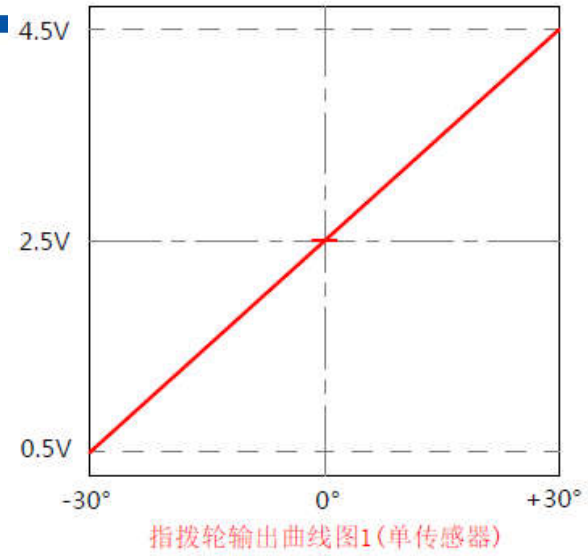
Voltage signal: Output curve 2



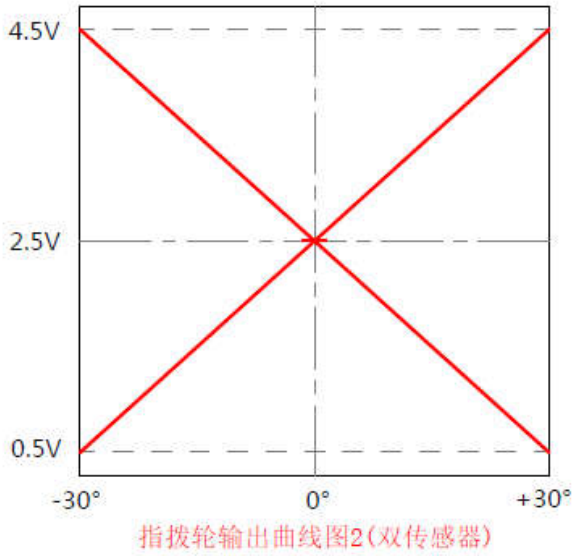
Voltage signal: Output curve 3



Voltage signal: Output curve 4

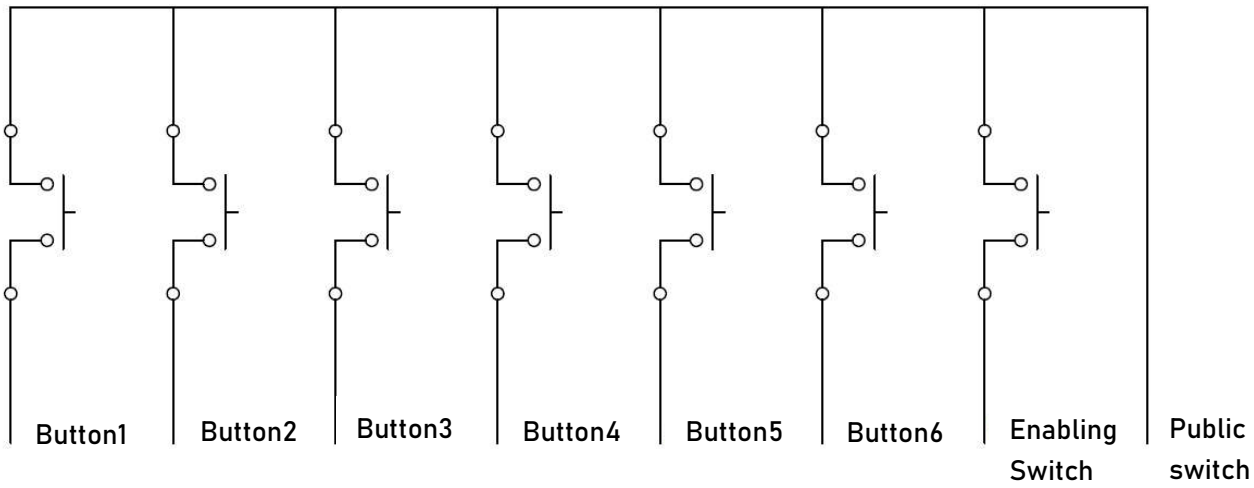


Thumb wheel output curve(single sensor)

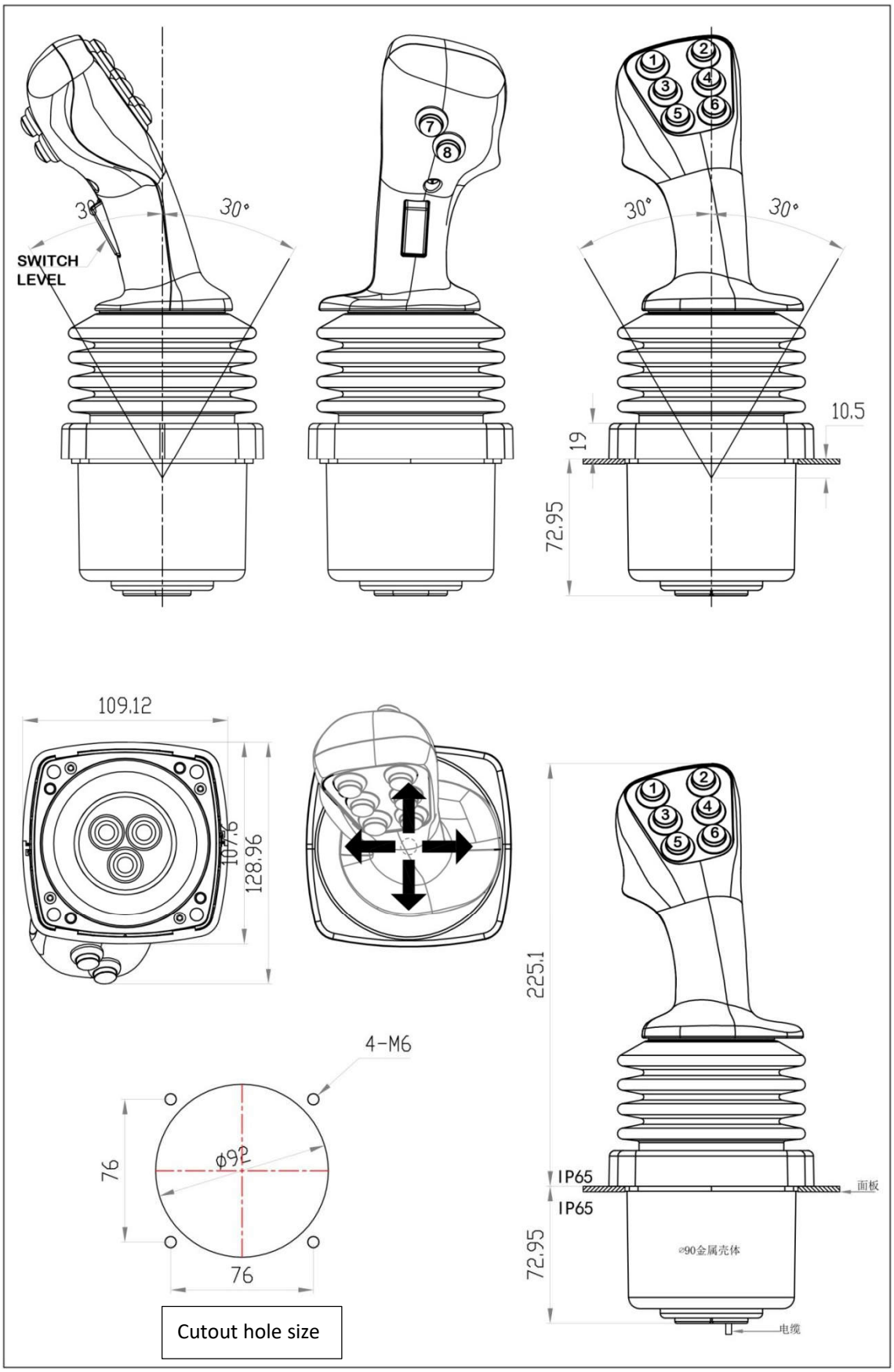


Thumb wheel output curve(double sensor)

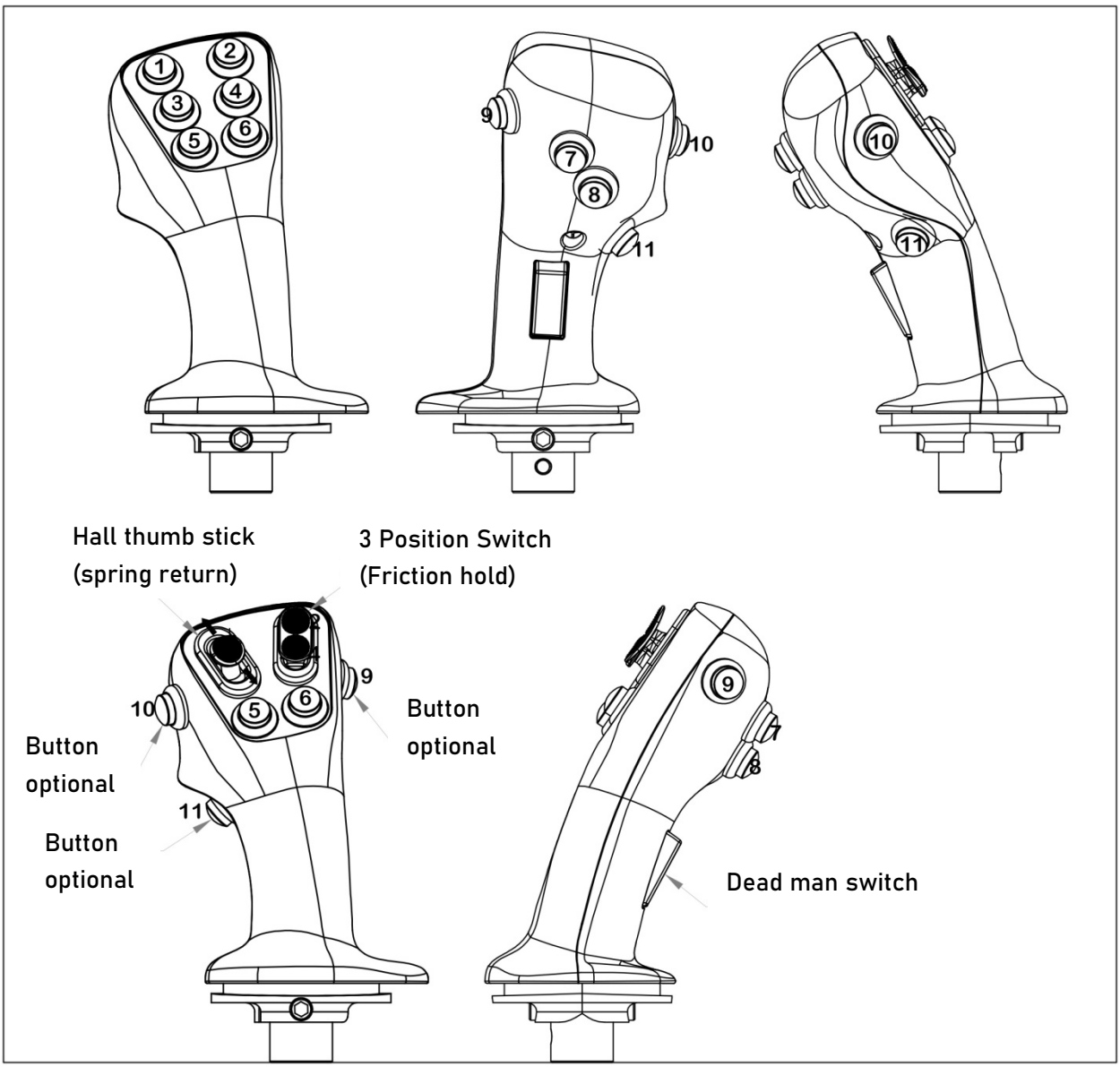
SCH of joystick direction switch and button



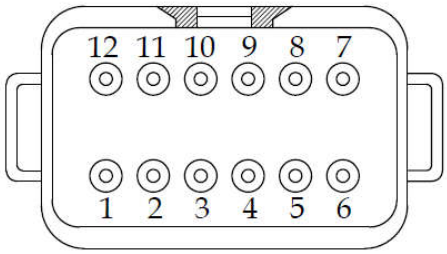
MECHANICAL DRAWING MKF-JS12A HANDLE



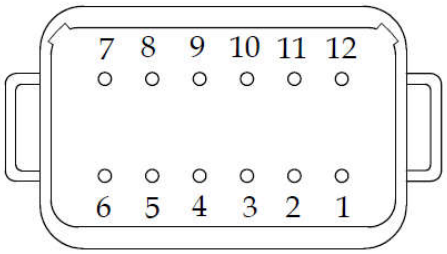
MECHANICAL DRAWING MKF-JS12A HANDLE



CONNECTOR LEAD

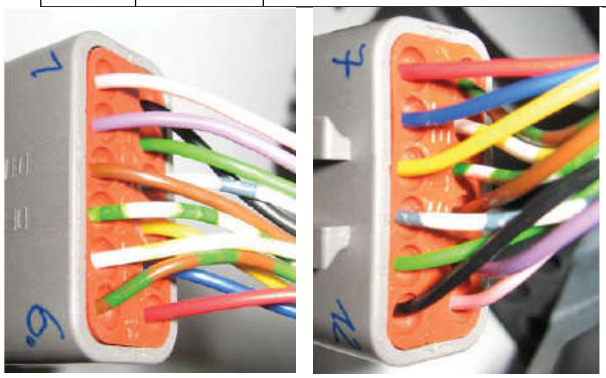


Back view of connector



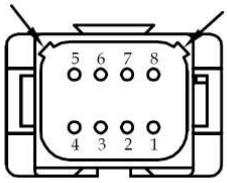
Front view of connector

Pin	Symbol	Feature		Pin	Symbol	Feature
1	V+	Power + 9-32V		12	RX-	RS422 RX-
2	V-	Power - 0V		11	RX+	RS422 RX+
3				10	TX-	RS422 TX-
4	CAN-H	CAN-H		9	TX+	RS422 TX+
5	CAN-L	CAN-L		8	RXD	RS232 RxD
6	GND	RS232 Signal GND		7	TXD	RS232 Rxd



Back view of connector

CAN port 8 cores connector DT04-8P



Front view

Wiring table description

Pin	Symbol	Color	Description		Pin	Symbol	Color	Description
1	V+	Red	Power+ 9-32V		8	GND	Brown	RS232 Signal GND
2	V-	Black	Power- 0V		7	RXD	Yellow	RS232 RXD
3			Null		6	TXD	Green	RS232 TxD
4	CAN-H	Blue	CAN-H		5	CAN-L	White	CAN-L

CAN communication, RS232 set the communication parameters for the joystick

Analog voltage interface lead table (without connector)

1. Sensor 1 channel

+5V	Red	
0V(GND)	Black	
X1	Yellow	(left –small, right-big)
Y1	Green	(down-small, up-big)

2. Sensor 2 channel

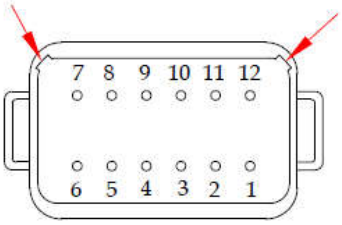
+5V	Red	
0V(GND)	Black	
X2	Yellow	(left-big, right-small)
Y2	Green	(down-big, up-small)

3. Button switch

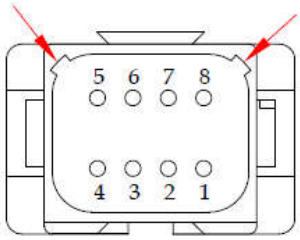
Button 1	White
Button 2	White
Button 3	White
Button 4	White
Button 5	White
Button 6	White
Enable	Gray

Button common port: Brown

Analog voltage port connector:



Connector Front View
DT04-12P (Male)



Connector Front View
DT04-8P (Male)

12P connector DT04-12P		
Pin	**Color**	**Feature**
1		Button 1
2		Button 2
3		Button 3
4		Button 4
5		Button 5
6		Button 6
7	Gray	Enable switch
8		Button 7
9		Button 8
10		Button 9
11		Button 10
12	Brown	Button switch common port

8P connector DT04-8P		
Pin	**Color**	**Feature**
1	Red	Sensor 1 Power+5V
2	Black	Sensor 1 Power 0V
3	Yellow	Sensor 1 Signal X (left/small)
4	Green	Sensor 1 Signal Y (down/small)
5	Red	Sensor 2 Power +5V
6	Black	Sensor 2 Power 0V
7	Yellow	Sensor 2 Signal X (left/big)
8	Green	Sensor 2 Signal Y (down/big)
Neutral position signal: 2.5V		

CAN Bus communication mode

- CAN2.0B
- Frame ID: Standard frame ID and extend frame ID, default ID=0X0101 (can modify these identification code via RS232)
- Baud rate：125K, 250K, 500K, 1000K (default 250K)
- Transmit mode：Interval 5-100ms continuous send (default 30ms)

2-4 axes transmit CAN communication protocol(Joystick→PC)

CAN baud rate and ID setting in joystick of"sending model D"

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Feature
Byte0	Z2+	Z2-	Z1-	Z1+	Right	Left	Backward	Forward	
Byte1	A7	A6	A5	A4	A3	A2	A1	A0	Axis 1 (value) Y
Byte2	A7	A6	A5	A4	A3	A2	A1	A0	Axis 2 (value) X
Byte3	A7	A6	A5	A4	A3	A2	A1	A0	Axis 3 (value) Z1
Byte4	A7	A6	A5	A4	A3	A2	A1	A0	Axis 4 (value) Z2
Byte5				Button 11	Button 10	Button 13	Button 12	Button 9	Digital inputs (palm grip)
Byte6	Button8	Enable Level / Button 7	Button 6	Button 5	Button 4	Button 3	Button 2	Button 1	Digital inputs (palm grip)
Byte7	CNT7	CNT6	CNT5	CNT4	CNT3	CNT2	CNT1	CNT0	Message counter(8bit) 0-255

Note: ①2 axes AXIS3,AXIS4 Value=0x00 Axis1(Y), Axis2(X) data range 0X00-0XFF (0-255)

②3 axes AXIS4 Value =0x00 Axis3(Z1-Y), Axis4(Z1-X) data range 0X00-0XF0 (0-240)

③Direction: Valid=1, Invalid=0 Up (forward) +, down (backward)-, right+, left-

Joystick receive CAN data format（PC→Joystick）-for LED and handle vibration

The receive frame ID in parameter setting

Only when the frame ID of the received data is the same as the "received frame ID " of the board, LED and handle vibration control be carried out

Data frame(8 bytes HEX):

	Symbol	Feature	
BYTE0	Vibrat	Vibation	00=off, 01=open
BYTE1	V-Level	Vibation level	1-10 level（Frequency10-30HZ）
BYTE2	00		
BYTE3	00		
BYTE4	00		
BYTE5	00		
BYTE6	00		
BYTE7	F5	0xF5	0xF5

1=led ON 0=led OFF

5 axes CAN communication protocol

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Feature
Byte0	0	0	0	0	Right	Left	backward	forward	Axis1,2 Direction
Byte1	A7	A6	A5	A4	A3	A2	A1	A0	Axis 1 (value) Y
Byte2	-	A6	A5	A4	A3	A2	A1	A0	Axis 2 (value) X
Byte3	S10	S09	Z3-	Z3+	Z2-	Z2+	Z1-	Z1+	Axis3,4,5 Direction Button 9, 10
Byte4	A7	A6	A5	A4	A3	A2	A1	A0	Axis 3(value) Z1
Byte5	A7	A6	A5	A4	A3	A2	A1	A0	Axis 4 (value) Z2
Byte6	A7	A6	A5	A4	A3	A2	A1	A0	Axis 5 (value) Z3
Byte7	Button 8	Enable Level / Button 7	Button 6	Button 5	Button 4	Button 3	Button 2	Button 1	Digital inputs (palm grip)

Note: ①2 axes AXIS3, AXIS4 value=0x00 Axis1(Y), Axis2(X) Data range 0X00-0XFF (0-255)

②3 axes AXIS4 value=0x00 Axis3(Z1-Y), Axis4(Z1-X) Data range 0X00-0XF0 (0-240)

③Direction: Valid=1, Invalid=0 Up（forward）+, down (backward) -, right+, left-

2-4 axes joystick RS232/RS422/485 communication protocol

Default baud rate 9600.8.1.N, No address bit by factory default

Function:Send the position parameter of each axis of the joystick

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	State
Byte0	0xFF								
Byte1	Z2_R+G	Z2_L-G	Z1_D-G	Z1_U+G	Right	Left	backward	forward	
Byte2	A7	A6	A5	A4	A3	A2	A1	A0	Axis 1 (value) Y
Byte3	A7	A6	A5	A4	A3	A2	A1	A0	Axis 2 (value) X
Byte4	A7	A6	A5	A4	A3	A2	A1	A0	Axis 3 (value) Z1
Byte5	A7	A6	A5	A4	A3	A2	A1	A0	Axis 4 (value) Z2
Byte6				Button 11	Button 10	Button 13	Button 12	Button 9	Digital inputs (palm grip)
Byte7	Button 8	Enable Level/ Button 7	Button 6	Button 5	Button 4	Button 3	Button 2	Button 1	Digital inputs (palm grip)
Byte8	CNT7	CNT6	CNT5	CNT4	CNT3	CNT2	CNT1	CNT0	Message counter(8bit) 0-255
Byte9	CH=Byte1+ Byte2+ Byte3+ Byte4+ Byte5+ Byte6+ Byte7+ Byte8 The lowest byte of the result								Checksum

Note:①2 axes AXIS3, AXIS4 value=0x00 Axis1(Y), Axis2(X) Data range 0X00-0XFF (0-255)

②3 axes AXIS4, value=0x00 Axis3(Z1-Y), Axis4(Z1-X) Data range 0X00-0XF0 (0-240)

③Direction: Valid=1, Invalid=0 UP (forward)+, Down(Backward)-, Right+, Left- (G Thumb rocker)

5 axes joystick RS232/RS422/485 communication protocol

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	State
Byte0	0xFF								
Byte1	0	0	0	0	Right	Left	backward	forward	Axis1,2 direction
Byte2	A7	A6	A5	A4	A3	A2	A1	A0	Axis 1 (value) Y (0-255)
Byte3	-	A6	A5	A4	A3	A2	A1	A0	Axis 2 (value) X(0-255)
Byte4	S10	S09	Z3-	Z3+	Z2_RG	Z2_LG	Z1_DG	Z1_UG	Axis3,4,5 direction Button 9，10
Byte5	A7	A6	A5	A4	A3	A2	A1	A0	Axis 3(value) Z1 (0-240)
Byte6	A7	A6	A5	A4	A3	A2	A1	A0	Axis 4 (value) Z2 (0-240)
Byte7	A7	A6	A5	A4	A3	A2	A1	A0	Axis 5 (value) Z3 (0-240)
Byte8	Button8	Enable Level/ Button 7	Button 6	Button 5	Button 4	Button 3	Button 2	Button 1	Digital inputs (palm grip)
Byte9	CH=Byte1+ Byte2+ Byte3+ Byte4+ Byte5+ Byte6+ Byte7+ Byte8 The lowest byte of the result								Checksum CH

Note: Direction: Valid=1, Invalid=0 DIR: UP(forward) +, Down(Backward)-, Right+, Left-

Those with address bits add an address bits after FF , so become 11 bytes.

Format as follows： FF Addr Dir1 Axis1 Axis2 Dir2 Axis3 Axis4 Axis5 CH

6 axes joystick RS232/RS422/485 communication protocol

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	State
Byte0	0xFF								
Byte1	0	0	0	0	Right	Left	backward	forward	Axis1, 2 Direction
Byte2	A7	A6	A5	A4	A3	A2	A1	A0	Axis 1 (value) Y (0-255)
Byte3	-	A6	A5	A4	A3	A2	A1	A0	Axis 2 (value) X(0-255)
Byte4	Z4-	Z4+	Z3-	Z3+	Z2_RG	Z2_LG	Z1_DG	Z1_UG	Axis3,4,5 Direction Button 9, 10
Byte5	A7	A6	A5	A4	A3	A2	A1	A0	Axis 3(value) Z1 (0-240)
Byte6	A7	A6	A5	A4	A3	A2	A1	A0	Axis 4 (value) Z2 (0-240)
Byte7	A7	A6	A5	A4	A3	A2	A1	A0	Axis 5 (value) Z3 (0-240)
Byte8	A7	A6	A5	A4	A3	A2	A1	A0	Axis 6 (value) Z4 (0-240)
Byte9	--	Enable Level/ Button 7	Button 6	Button 5	Button 4	Button 3	Button 2	Button 1	Digital inputs (palm grip)
Byte10	CH=Byte1+ Byte2+ Byte3+ Byte4+ Byte5+ Byte6+ Byte7+ Byte8+ Byte9 The lowest byte of the result								CH

Note: Direction: Valid=1, Invalid=0; DIR: UP(forward)+, Down(Backward)-, Right+, Left-

Modbus communication protocol

Modbus RTU Master station mode:

1. Baud rate: 9600(can be modified)
2. Data bits: 1 start bit, 8 data bits, 1 stop bit, no check bit
3. Communication interface：Either RS485 or RS232
4. Data format: Modbus
5. Operating mode: Master station(the master station sends data to slave station 1)
6. 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	
I st register address- high-order	0x40	Register address
I st register address- low-order	0x01	
Number of register- high-order	0x00	Register quality
Number of register- low-order	0x04	
Data length	0x08	Byte length
IO high-order Bit15-Bit8	0x00	Bit7 button 10, bit6 button 9; Bit5-0 direction of joystick
IO low-order Bit7-Bit0	0X00	1-8 buttons(Bit0= button 1) 1=ON, 0=OFF
Joystick Y axis high-order	00	Constant 0x00
Joystick Y axis low-order	0x00-0xff	Y axis 0-255 angle value
Joystick X axis high-order	0x00	Constant 0x00
Joystick X axis low-order	0x00-0xff	X axis 0-255 angle value
Joystick Z axis high-order	00	Constant 0x00
Joystick Z axis low-order	0x00-0xff	Z axis 0-255 angle value
CRC high-order		
CRC low-order		

Note：Direction: Valid=1, Invalid=0; DIR: UP (forward)+, Down(Backward)-, Right+, Left-

Digital value high-order （1=Valid, 0=Invalid）

Biyt7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Button10	Button9	Z-	Z+	RIGHT	LEFT	DOWN	UP
Button 10	Button 9			Right	Left	backward	forward

Timing send 25ms/frame (refresh rate can be set)

01 10 40 01 00 04 08 00 00 08 00 08 00 08 00 B1 91

Modbus RTU slave station mode：

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

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 40 01 00 04 00 09

When the joystick receives the command, then the joystick switches to slave mode from master mod (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	
IO high-order Bit Bit15-Bit8	0x00	Bit7 Buttons 10，bit6 Buttons 9；Bit5-0 Direction of joystick
IO low-order Bit Bit7-Bit0	0X00	1-8 Buttons（Bit0= Buttons 1)1=ON, 0=OFF)
Joystick Y axis high-order	00	Constant 0x00
Joystick Y axis low-order	0x00-0xff	Y axis 0-255 Angle value
Joystick X axis high-order	0x00	Constant 0x00
Joystick X axis low-order	0x00-0xff	X axis 0-255 Angle value
Joystick Z axis high-order	00	Constant 0x00
Joystick Z axis low-order	0x00-0xff	Z axis 0-255 Angle value
CRC high-order		
CRC low-order		

Digital value high-order （1=Valid, 0=Invalid)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Button10	Button9	Z-	Z+	RIGHT	LEFT	Back	Forward
				Direction	Direction	Direction	Direction

Note: Direction：Valid=1, Invalid=0；DIR：UP（forward）+, Down (Backward)-, Right+, Left-

E.g:

PC→Joystick: 01 03 40 01 00 04 00 09

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

Joystick communication parameter setting

User 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 converter (standard converter, not TTL lever)

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

1. Basic instruction:

2. ACK confirmation (Joystick-PC)

AA 55 AF

It indicates that the joystick successfully receives instructions and executes them

3. Joystick ID address setting;

ID is in RS232/RS422 communication protocol, or CANopen (PC-> Joystick)

0xaf	0x0d	00	00	00	Add	0xf5
Head	Command	Data1	Data 2	Data 3	Data 4	Tail

Add=0x01～0x7F Address 1-127

Add=0x00 Invalid (Address =0, RS232 or RS422 no addrss bit)

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

4. 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 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)

5. Setting the center position of the joystick (PC-> joystick)

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

PC connect with RS422, baud rate 9600

0xaf	0x09	00	00	00	00	0xf5
Head	Command	Data1	Data 2	Data 3	Data 4	Tail

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

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

6. 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	Data 1	Data 2	Data 3	Data 4	Tail

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

7. 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	Data 2	Data 3	Data 4	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)

Set communication parameter of RS232, RS422 and RS485

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	Data	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)

Set succeed then joystick return ACK

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	Data 2	Data 3	Data 4	Tail

- Add = 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 00 79

Joystick return send current position when receive this data

CAN parameter setting:

CAN parameter setting via RS232or RS422 port

CAN port baud rate: (PC->joystick)

0xaf	0x06	XX	00	00	00	0xf5
Head	Command	Data1	Data 2	Data 3	Data 4	Tail

XX=00 125K

XX=01 250K (Default)

XX=02 500K

XX=03 1000K

XX=04 100K

E.g.: af 06 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 setting: (PC->Joystick)

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

Pro: protocol format

Pro=00 Factory protocol Xldq

SS: Canopen protocol

The differet Canopen with common protocl which is CAN ID , the data format no change

SS=01 CANopen protocol ID=180+ID （Reg.（2）Setting joystick ID address）

(The factory has set up)

SS=00 non CanOpen ID=Sending node ID, Reg.(11joystick sending node ID setting)

Default: non CanOpen

TP: TP in CAN OPEN protocol: TPDO

TP=00: TPDO1 send ID 0X0180+ID(Reg.一、2、Setting joystick ID address)Default

TP=01: TPDO2 send ID 0X0280+ID(Reg.一、2、Setting joystick ID address)

TP=02: TPDO3 send ID 0X0380+ID(Reg.一、2、Setting joystick ID address)

TP=03: TPDO4 send ID 0X0480+ID(Reg.一、2、Setting joystick ID address)

e.g.: af 0a 00 00 00 00 f5 (HEX) Common protocol, protocol format XLDDQ

af 0a 00 04 00 00 f5 (HEX) 6 axis A10 protocol format

af 0a 00 00 00 01 f5 (HEX) CANopen protocol, TPDO1, protocol format XLDDQ

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

Only for"standard protocol", not for CANopen

0xaf	0x01	D1	D2	D3	D4	0xf5
Head	Command	Data 1	Data 2	Data 3	Data 4	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 " indentification code "

E.g.: Set sending node indentification 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 indentification code"

E.g.: Set sending node indentification code – standard frame "0X181"

af 01 80 00 01 81 f5 (HEX)

Joystick "Receiving node ID"setting: (PC->Joystick)

Only for"standard protocol", Not for CANopen

0xaf	0x02	D1	D2	D3	D4	0xf5
Head	Command	Data 1	Data 2	Data 3	Data 4	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 "indentification code"

E.g.: Set receiving node indentification code–extend frame "0X00F0F101"

af 02 00 f0 f1 01 f5 (HEX)

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

E.g.: Set receiving node indentification 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 29 bits extend frame

D1.7=111 bits standard frame

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

E.g.: Set Shield node indentification code –extend frame "0X00002201"

af 03 00 00 22 01 f5 (HEX)

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

E.g.: Set Shield node indentification code –standard frame"0X122"

af 03 80 00 01 22 f5 (HEX)

Communication parameters of Modbus RTU (RS485) Setting

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.

Register address setting (PC->Joystick):

Data format:

0xaf	0x18	D1	D2	D3	D4	0xf5
Head	Command	Data 1	Data 2	Data 3	Data 4	Tail

D1: Register address high byte

D2: Register address low byte

(D3 and 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

Note: Technical data subject to change without notice!