ADP Z-axis Manual

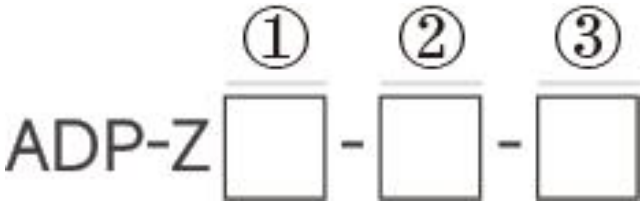
Shenzhen Keyto Fluid Technology Co., Ltd.

Contents

1. Product Overview

The Z-axis is used to carry the high precision movement of ADP products in the vertical direction, maximum 180mm stroke.The Z-axis is generally integrated with the ADP, supporting RS232 and CAN communication, and connecting the ADP also supports RS485 communication. In order to ensure the reliability of communication, it is recommended to use CAN communication.

1.1 How to order



Note code	Note type	Name	Product naming instructions
①	Stroke	120	120mm
		180	180mm
②	Screw type	B	Ball Screw
		S	Slide Screw
③	Customized code	0	Standard
		1	Customized adapter board
		2	Add brake function

1.2 Z-axis Main Features

- Use Z-axis with Keyto ADP, can achieve vertical movement with minimum center spacing of 9mm.
- Movement speed can be up to 180 mm/s
- Provide a vertical thrust of not less than 3kg for picking up the tips
- Loaded with a 410g weight, it is capable of achieving vertical self-locking in the event of a power failure.
- Movement accuracy is within ± 0.2mm.
- Support RS232 and CAN communication, using with ADP can achieve RS485 communication.
- Support single-channel, 2-channel, 4-channel and 8-channel application with Keyto ADP16
- Support single-channel, 2-channel and 4-channel application with Keyto ADP18
- Support single-channel application with Keyto ADP20 or Keyto ADP28

1.3 Definition

- Pipettor: Air Displacement Pipettor is a keyto based on pneumatic aspirat and dispense technology using with TIPs to realize the liquid handling module, Keyto series: SP16, SP18, SP20, SP28 and others;
- Master: client, sends commands to the Z-axis;
- Send: Direction is from the master to the Z-axis;
- Receive: Direction is from Z-axis to master;
- Device: Z-axis (ADPZ).

2. Product Specifications

2.1 Parameter table

Parameter table

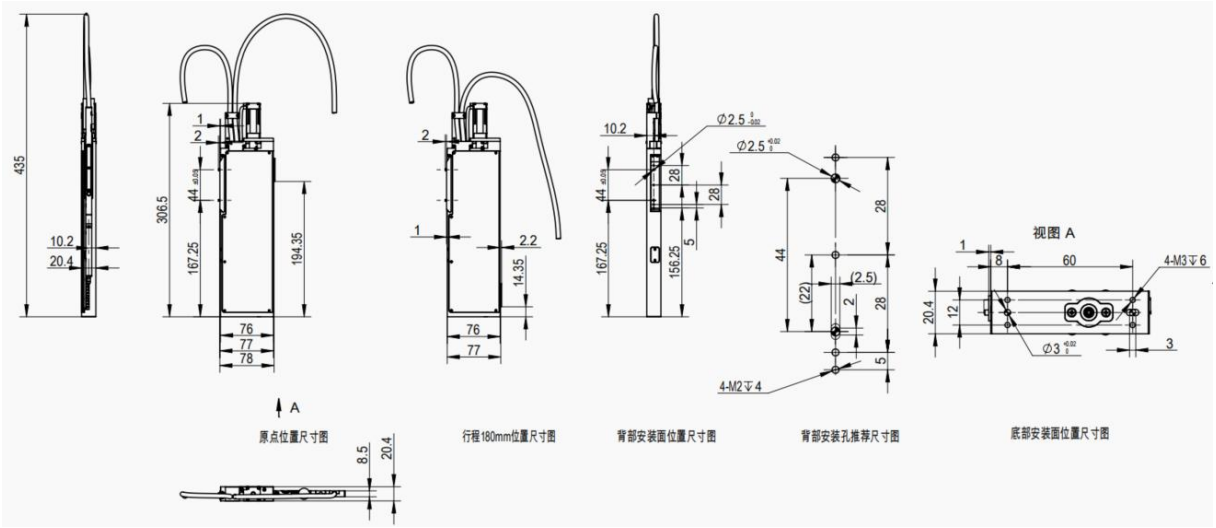
Product	ADP-Z180-S-0
Maximum speed	180mm/s
Maximum stroke	180mm
Weight	350g/410g(wire included)
Maximum noise	≤65dB（A）
Dimension	302.5mm*78mm*20.4mm（no wire）
Multi-channel center distance	Integrate with SP16, minimum 9mm

	Integrate with SP18, minimum 18mm
Communication	RS232、CAN、RS485（Use with keyto ADP series）
Baud rate	Serial port：9600，19200，38400（default），115200 CAN：100K，125K，250K，500K（default），1000K
Maximum thrust	3kg
Driver Design	步进电机加滑动丝杆 Stepper motor with sliding screw
Operating temperature	+15~+35°C
Voltage input	24V DC
Life cycle	100 万次 One million times
Maximum load	410g

2.2 Z180 Series Product Appearance

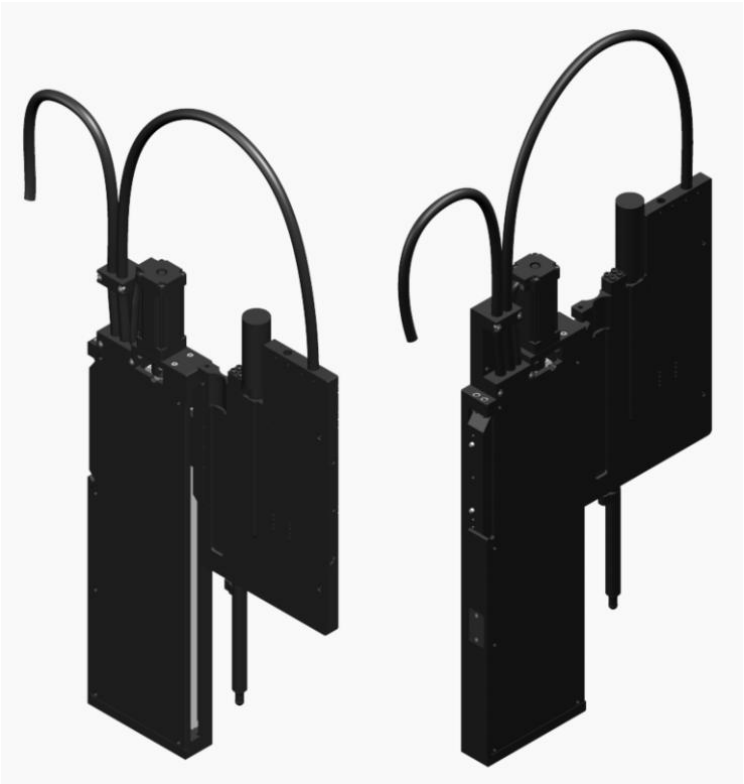
Product appearance

Overall Dimension and Installation

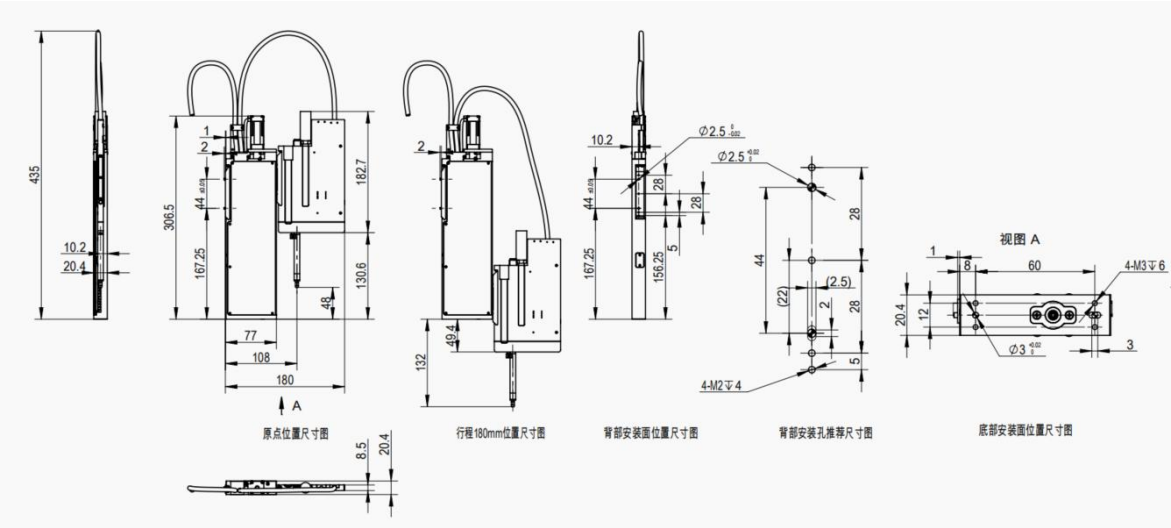


2.3 SP16+Z180 Series Product Appearance

Product appearance



Overall Dimension and Installation



Overall Dimension and Installation (Single channel)

3. Product Electrical Interface

3.1 Z-axis Interface Lead Definition

The Z-axis input power supply is 24VDC ±5%, with a peak current of under 900mA and a rated current of under 600mA. When you are connecting or disconnecting the cable, The power must be cut off!

The Z-axis integrate with the ADP and shares RS232, 485 and CAN communication. When using the device, communication lines which are not needed should insulate treatment to prevent functional abnormalities caused by the core contact.

Caution:

Communication lines which are not needed should insulate treatment

When you are connecting or disconnecting the cable, The power must be cut off!

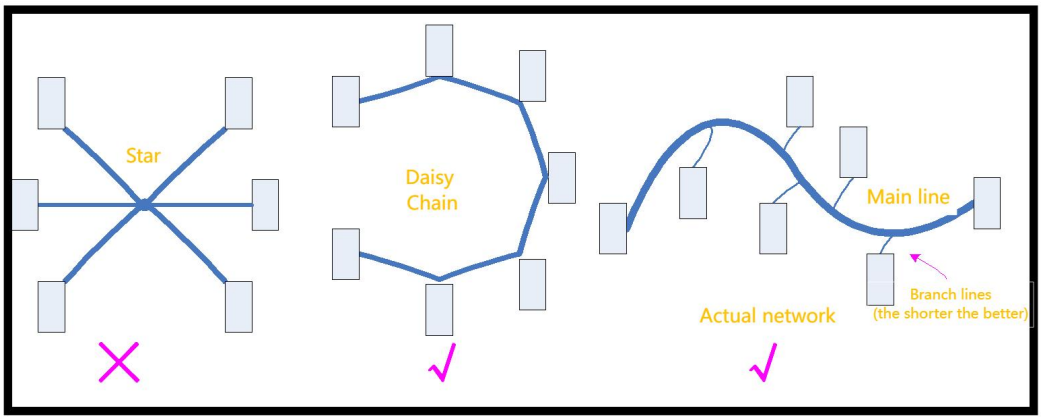
Using highly flexible drag chain cable to connect to the external control system with AWG26 cores. The following table defines the communication cable:

Wiring Definition

Function	Cable color	Remarks
DC 24V+	Red	Power input 24V±5%，≥900mA+500mA（SP16）
GND	Black	Grounding
RS232-RX	Green	Communication Interface, Z-axis input
RS232-TX	Blue	Communication Interface, Z-axis output
RS485A	Orange	Communication Interface
RS485B	White	Communication Interface
CAN L	Yellow	Communication Interface
CAN H	Yellow-green	Communication Interface

3.2 RS485/CAN Connection Topology Diagram

For RS485, it is recommended to reduce the number of bus devices, favoring one-to-one communication, which helps enhance communication efficiency and reliability.



4. Installation and commissioning

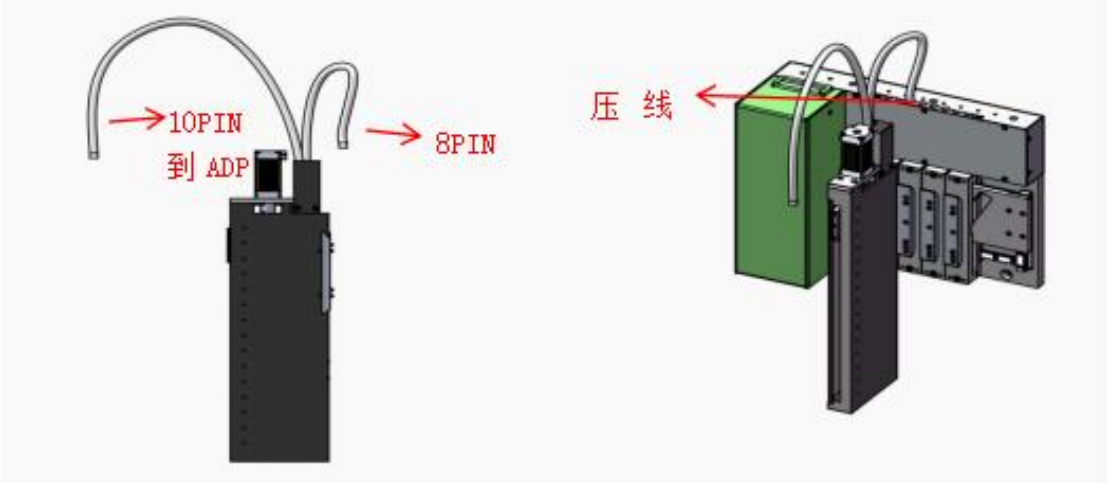
4.1 Installation of Z-axis and movement mechanism

As shown in Figure 4-1, install the Z-axis in the movement mechanism or vertical plane. When using the Z-axis seprately, you can use the equipped 8-core high-flex cable wired according to Table 4-1. When using the Z-axis with keyto movement mechanism, the Z-axis 8-core cable CAN cable and power cable into the movement mechanism, the relevant accessories are packed with the goods.

Caution:

When you are connecting or disconnecting the cable, The power must be cut off!

The movement mechanism crimp plate must be installed with the 8-core wire sheath compressed!

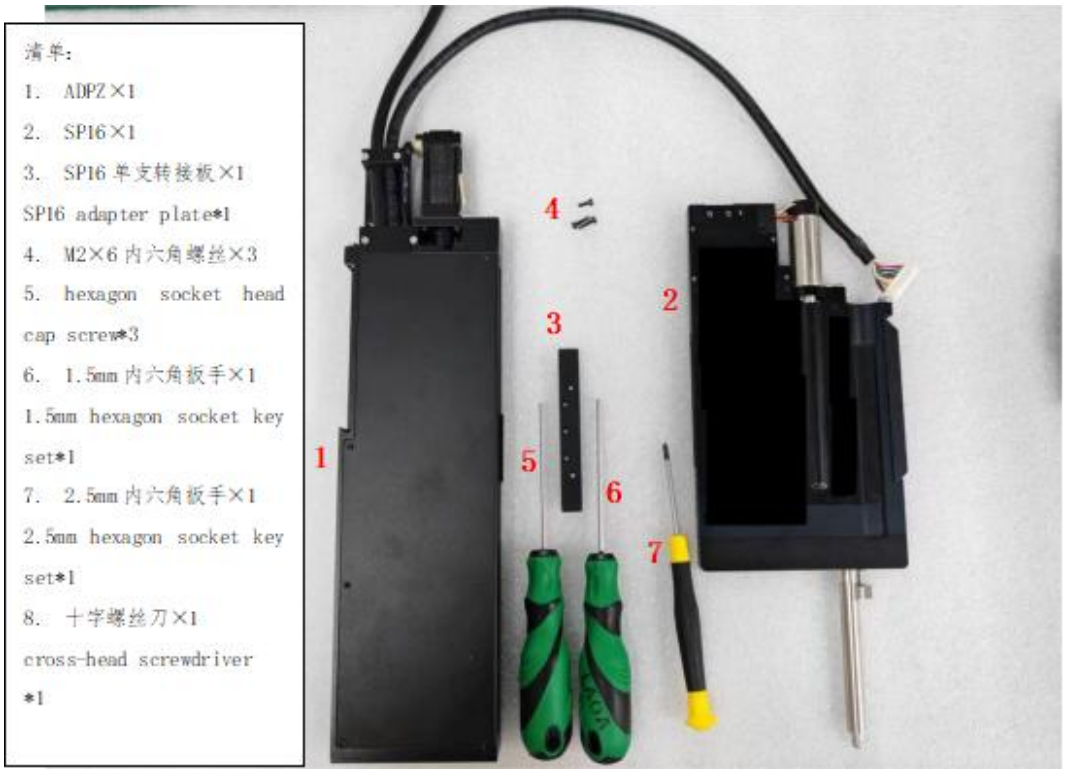


Z-axis installation Figure

4.2 ADPZ and ADP Installation Process

Single channel Installation process of ADPZ and SP16

Step1: Prepare the following tools and equipment.

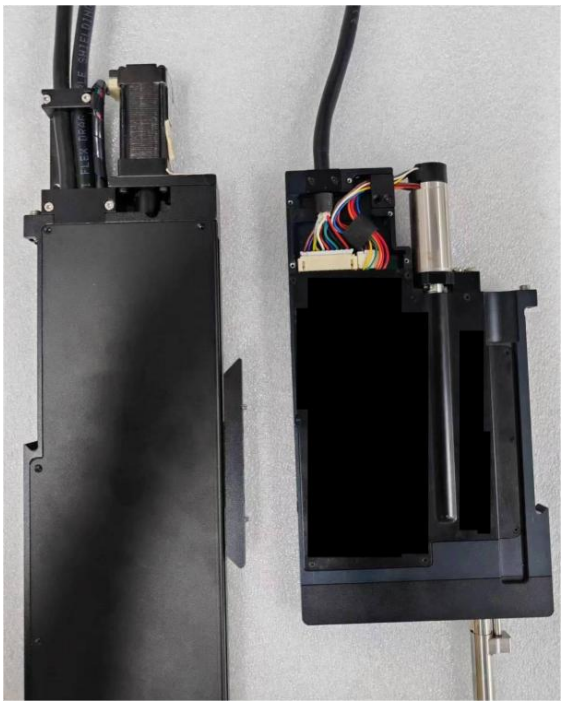


Step 2: Remove the mounting plate of SP16, Install the SP16 single adapter plate with ADPZ,large rounded corner facing down.

Step 3: Remove the cover plate and wire pressboard from ADP16, insertconnection cable between SP16 and ADPZ into SP16, then install the wire pressboard (please note that the pressboard should be level with the surface of SP16)

Caution:

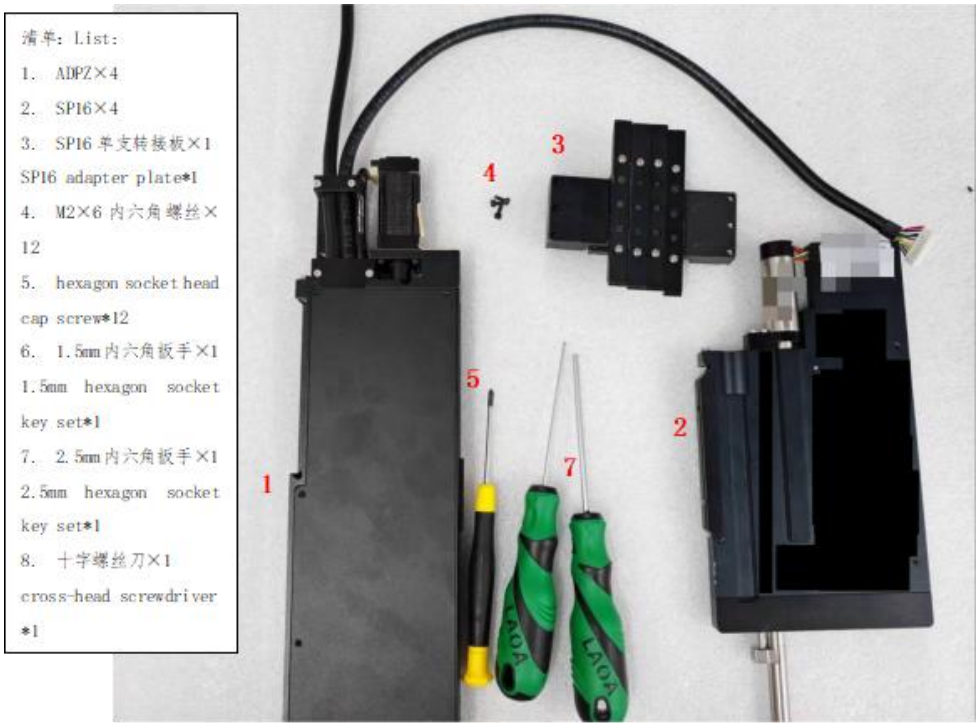
Attention to the wire pressboard need flush with SP16 surface.



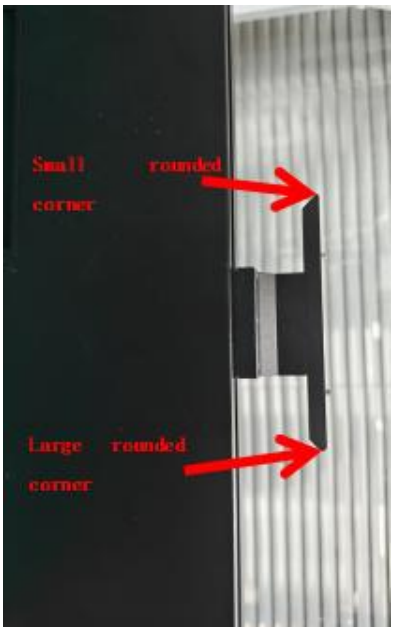
Step 4: Fix SP16 on SP16 single adapter plates (the pin of SP16 single adapter plate should be aligned with the hole of SP16 during installation)

Installation Process of ADPZ and ADP16 Fixed in 4-channel Movement Mechanism

Step 1: Prepare the following tools and equipment (The picture below shows the installation steps for just one set)



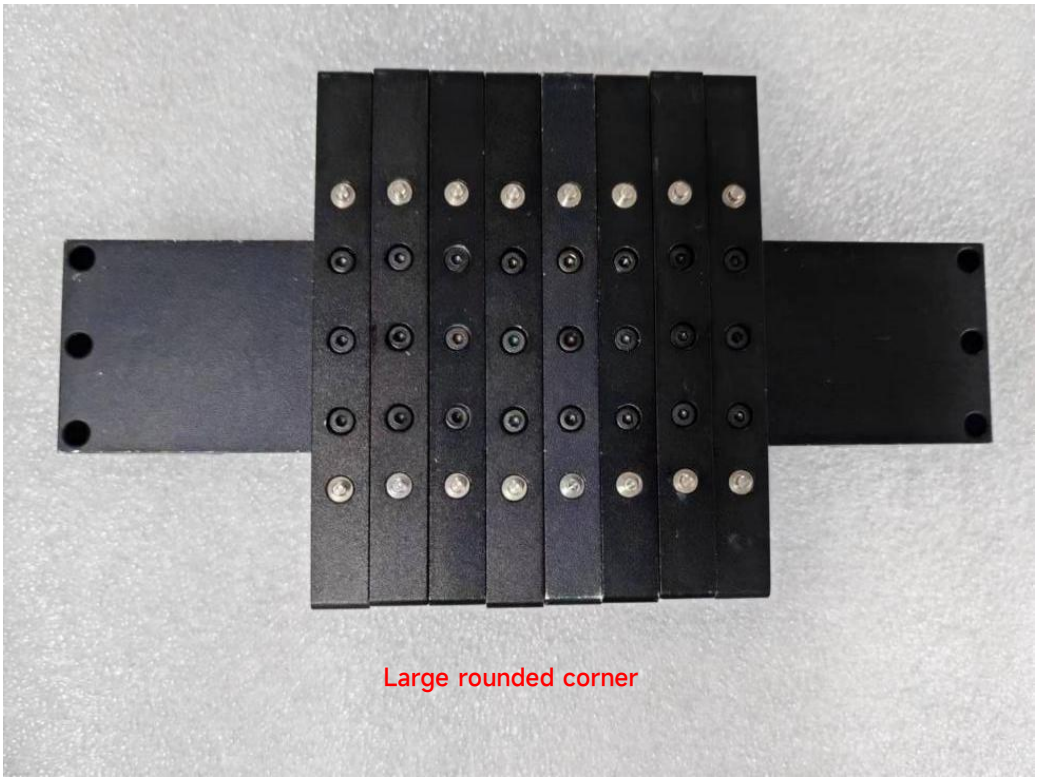
Step 2: Arrange 4 adapter plates in order and install them on ADPZ (Remark: When you install the 4-channel pipettors, in order SP16-A/SP16-)B/SP16-A/SP16-B or SP16-B/SP16-A/SP16-B/SP16-A



For other installation steps, please refer to the section Single channel Installation process of ADPZ and SP16.

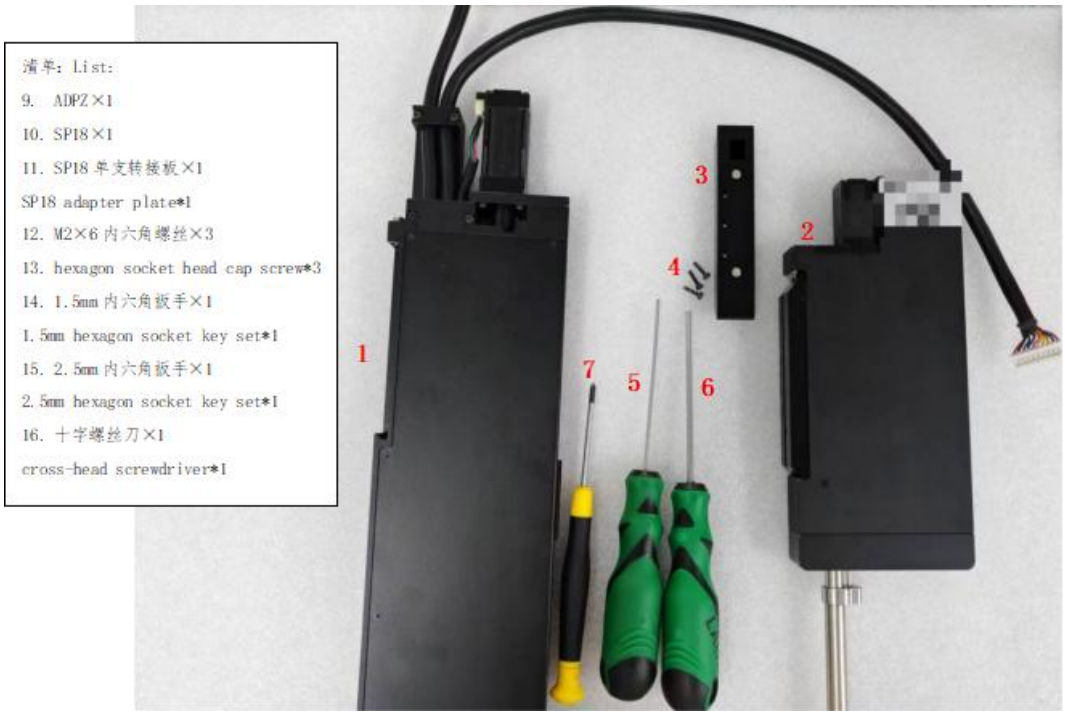
Installation Process of 8-channel ADPZ and ADP16

The installation steps please refer to section Installation Process of ADPZ and ADP16 Fixed in 4-channel Movement Mechanism.



Installation Process of ADPZ and SP18

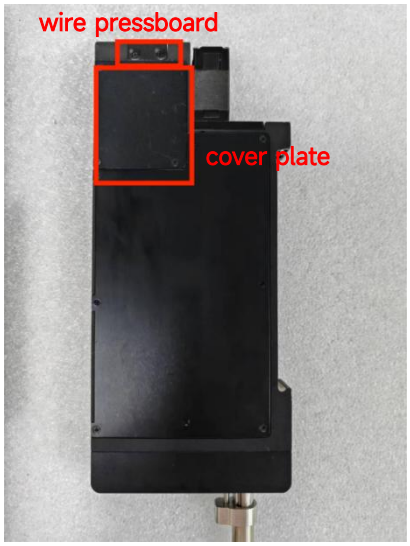
Step 1: Prepare the following tools and equipment.



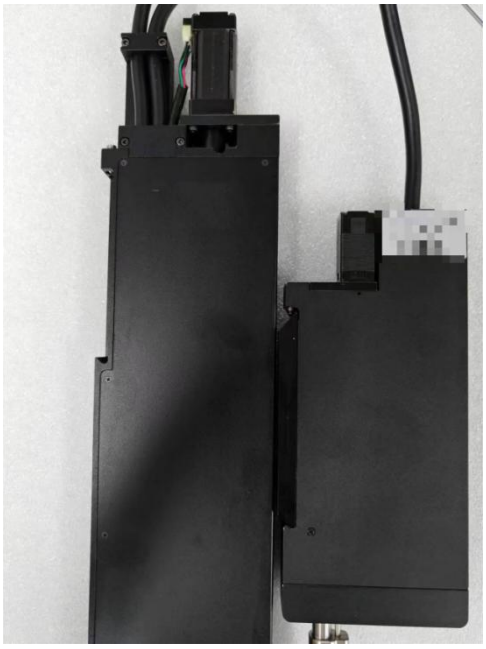
Step 2: Remove the mounting plate of SP18, Install the SP16 single adapter plate with ADPZ,large rounded corner facing down.

Step 3: Remove the cover plate and wire pressboard from ADP16, insertconnection cable between SP16 and ADPZ into SP16, then install the wire pressboard (please note that the pressboard should be level with the surface of SP16)

Caution:
Attention to the wire pressboard need flush with SP18 surface.

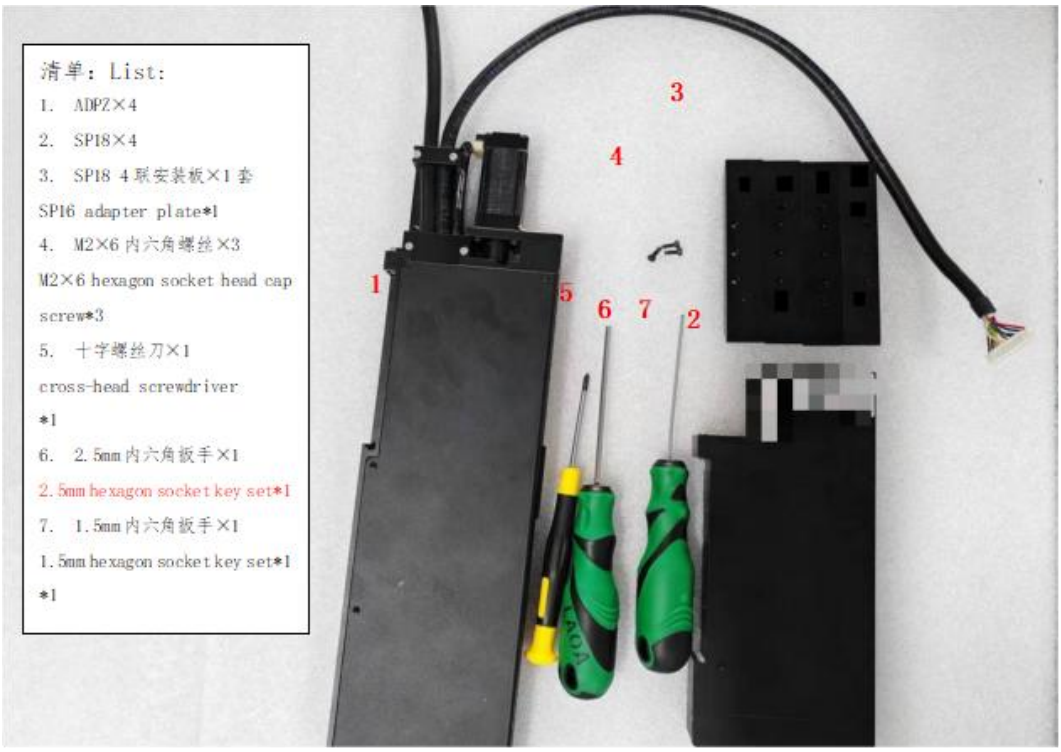


Fix SP18 on SP18 single adapter plates (the pin of SP18 single adapter plate should be aligned with the hole of SP18 during installation)



Installation Process of ADPZ and ADP18 Fixed in 4-channel Movement Mechanism

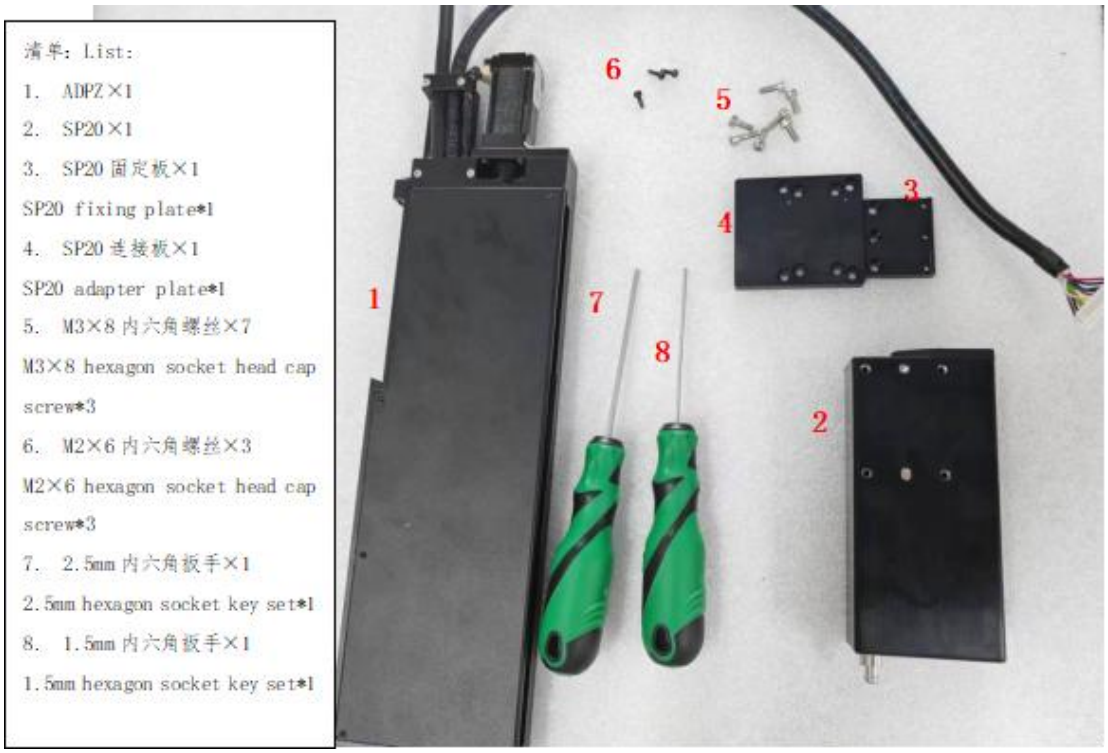
Step 1: Prepare the following tools and equipment (The picture below shows the installation steps for just one set)



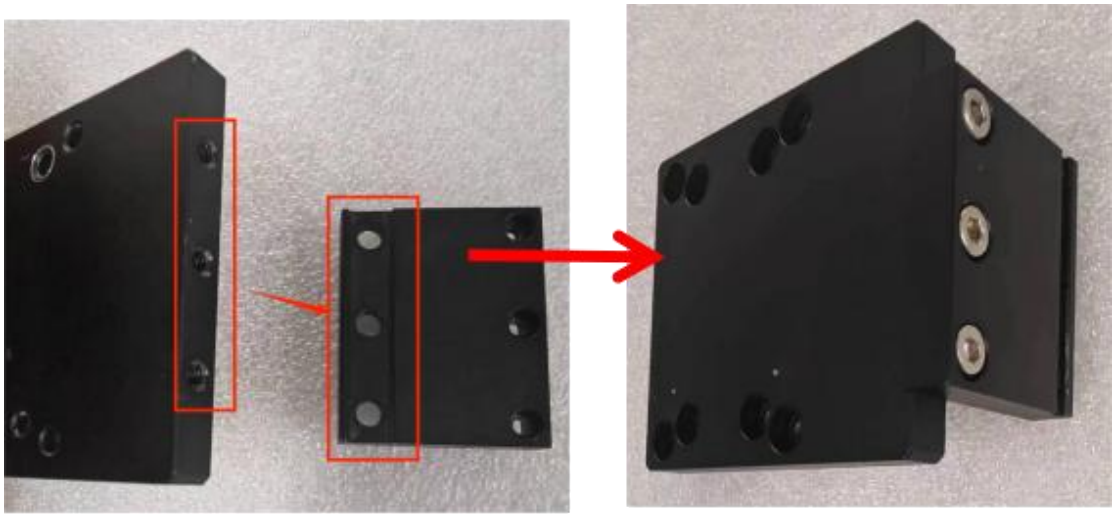
Step 2: Arrange 4 adapter plates in order and install them on ADPZ, Refer to section Installation Process of ADPZ and SP18 for the remaining installation steps.

Installation Process of ADPZ and SP20

Step 1: Prepare the following tools and equipment.



Step 2: Connecting the SP20's adapter plate to the fixing plate



Step 3:Install the connected adapter plate to ADP Z.

Step 4: Install the connected ADP Z and connection wire to SP20.

Product Electrical Interface

When using Keyto single ADP, connect the 8-core high-flex cable of ADPZ to the power supply and communication cable according to Table 4-1, and refer to Section 5 for debugging:

NOTE: Cables must be connected or disconnected with the power off!

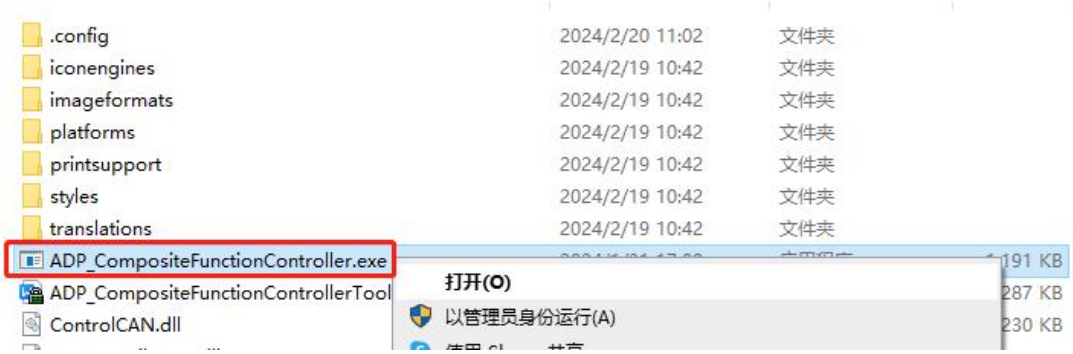
Debugging Wiring

Function	Cable color	Description
DC 24V+	Red	Power input 24V ±5%, ≥≥900mA (Z axis)+500mA(SP16)
GND	Black	Grounding
RS232-RX	绿色	Communication Interface, Z axis input
RS232-TX	蓝色	Communication Interface, Z axis output

5. Description of Host Computer Test Software

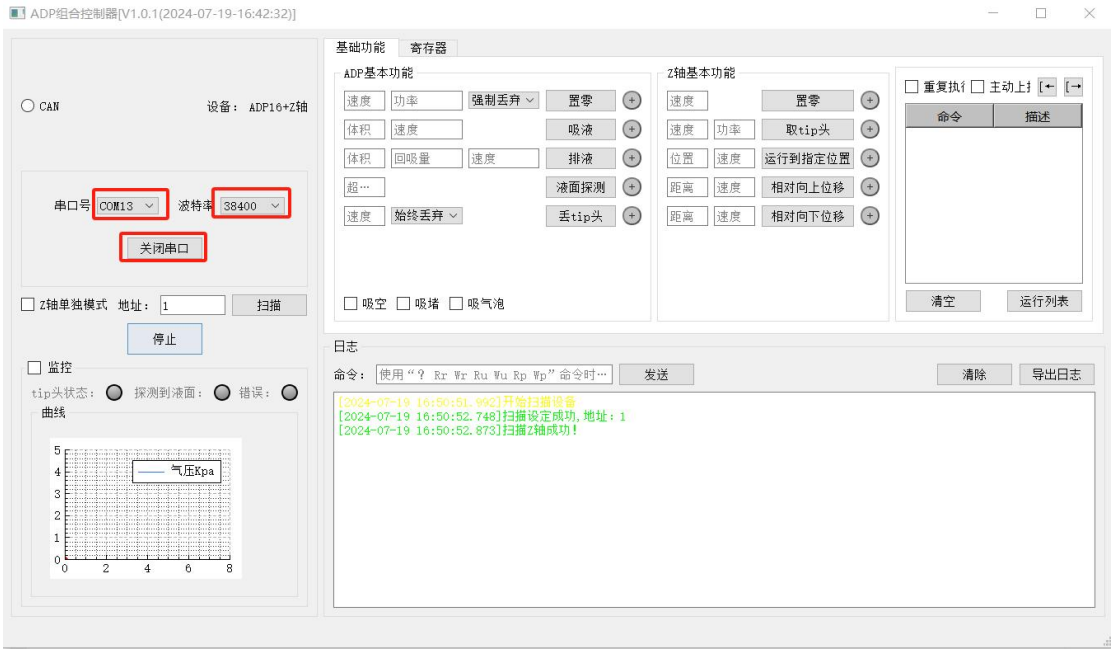
5.1 Open the host computer test software

Install the single Z-axis and SP16 according to section 4, connect the wiring and power up the unit, and open ADP_CompositeFunctionController.exe to test the software:



5.2 Serial port, Baud Rate Selection

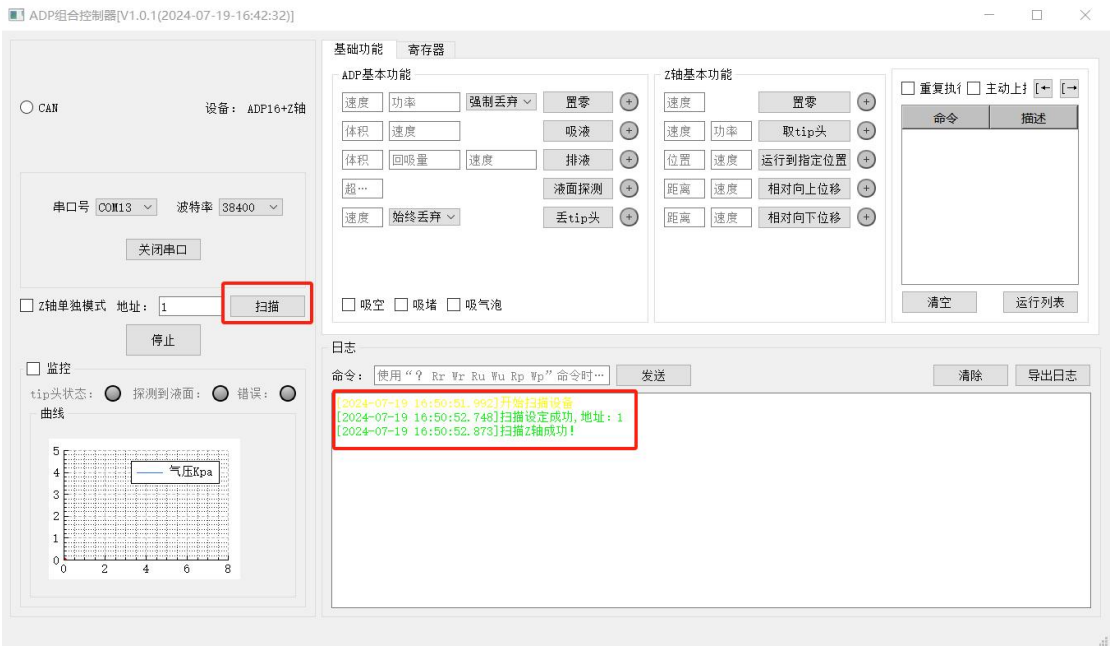
Select the corresponding serial port number to RS232, and the baud rate of 38400 (factory default 38400). Click "open serial port" buttons, after the operation, the button becomes “close serial port".



5.3 Scan Device Address

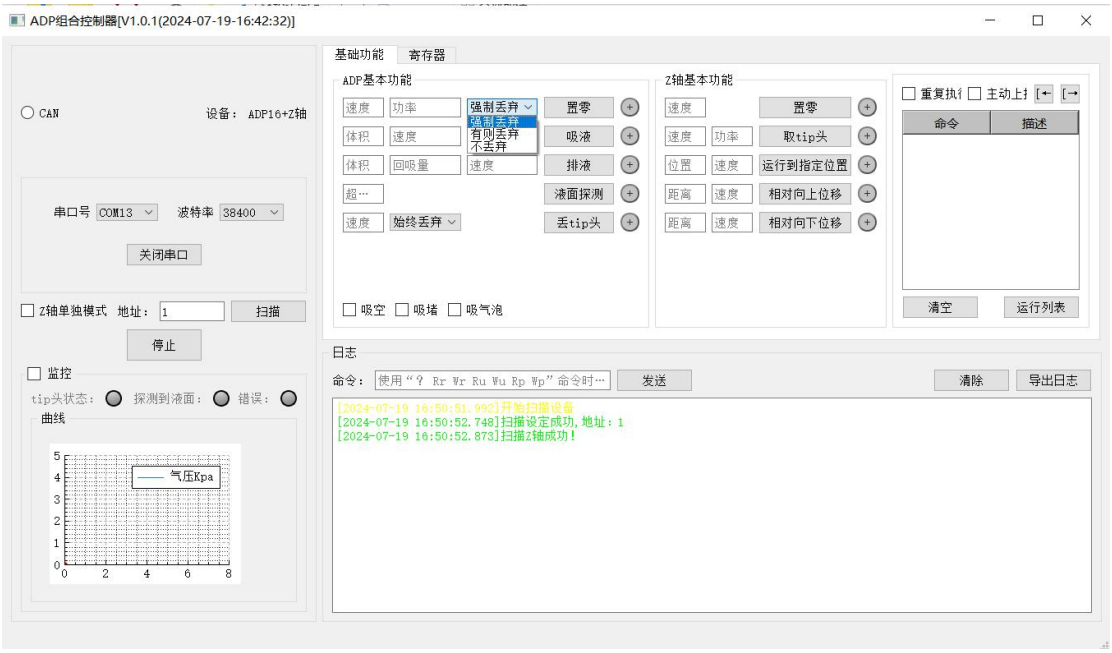
Click the "scan device" button, the default address is 1, the address bar will be automatically populated with the scanned address when the scan is complete.Note that other operations are executed only after the address is scanned.

When the Z-axis is not connected to ADP, after 10 seconds of scanning, the message "No connected device scanned" appears in the status bar, and subsequent operations cannot be performed.



5.4 Basic operation

In the Basic Function page, set the action parameters (default value if empty), and then click each action button, the status bar will display the contents of the commands sent to ADPZ and ADP, as well as the answer data and the description of the answer data of ADPZ and ADP, and refer to section 9.3 for the specific command parameters.

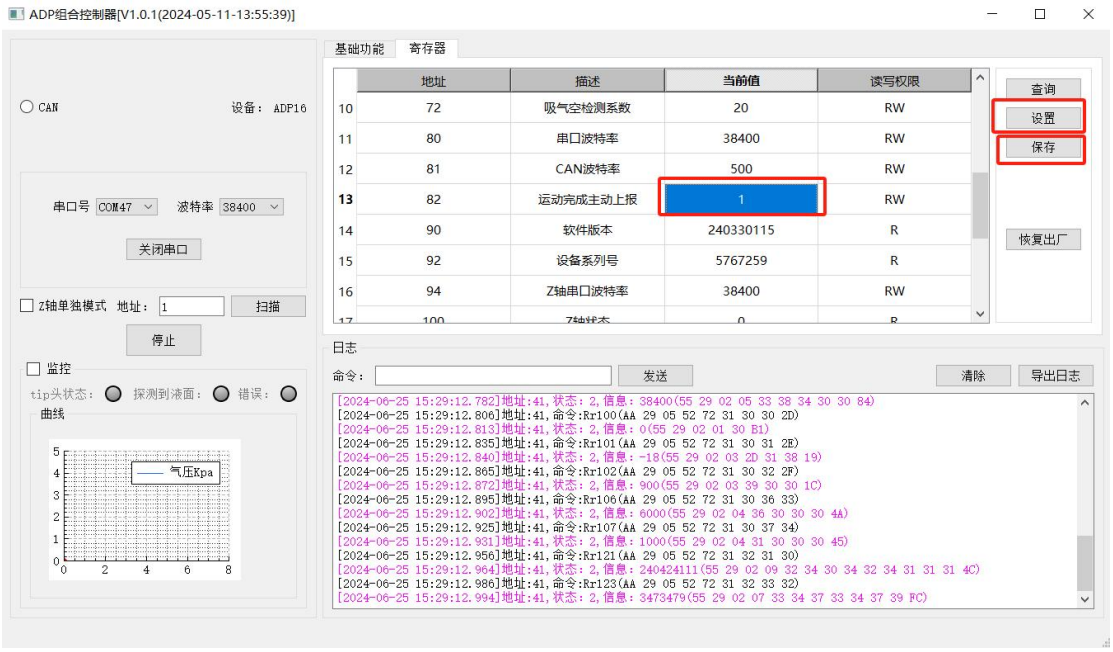


5.5 Register Query

Click ‘Register’bottom, to switch to the register interface. Click ‘Query’bottom, the host computer will query the device registers of ADPZ and ADP,and automatically updated‘Current Value’.

5.6 Register Parameter Settings

Double-click the “current value”in the register parameter line,which needs to be modified(RW can only be modified),fill in the modified value, press the Enter key or click other areas to uncheck, click the "Settings" button, click the "Save" button to save the set parameters to the ADPZ or ADP and will not be lost after power failure.Note that some parameters need to be rebooted to take effect, see Table 9-7 for the parameters of ADPZ and the corresponding specification for the parameters of ADP.

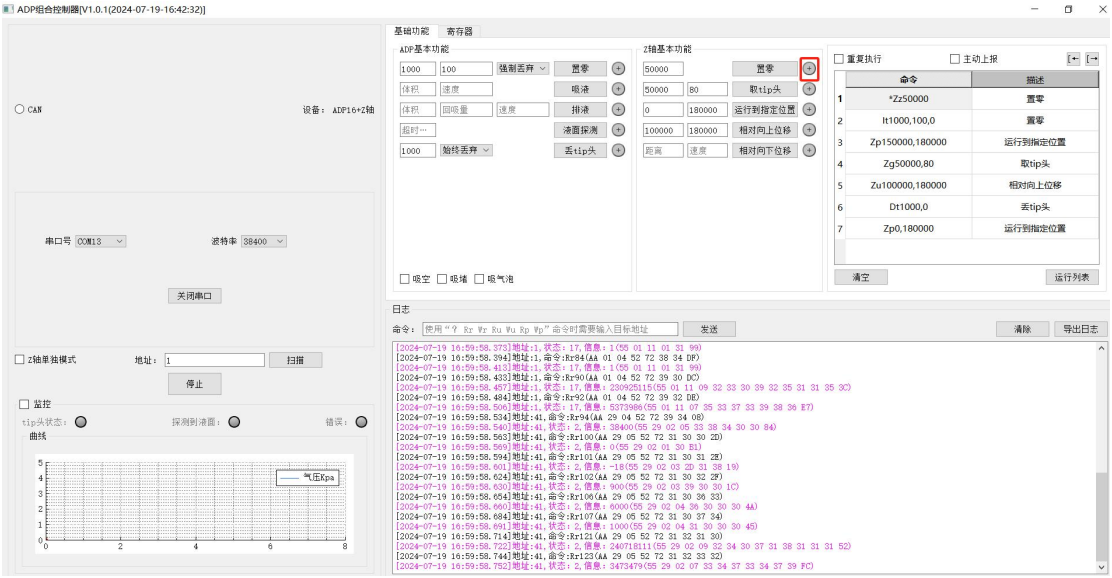


5.7 Restore to factory settings

Click‘Restore Factory’, all register settings will be restored to factory settings,settings require a device reboot to take effect.After clicking the "Restore Factory Settings" button, two messages with the command "M123456" will appear in the status bar, and two messages with the status "2" will appear after successfully restoring the factory settings.After successfully restoring the factory settings, two messages with status "2" will appear.

5.8 Command set

In section 5.4, the basic operation of single action control was described. This section describes how to operate multi-action control. As shown in the figure below, click the “+” button to add each single action command to the command set:



1. Zz50000: Z-axis is set to initialization at 50000 um/s ("*": the current instruction and the next instruction are executed at the same time, you need to edit manually to add "*" to the front of the instruction);

2. It1000,100,0: ADP is set to initialization at 1000um/s, 100 power to enject TIP, no back TIP action;

3. Zp150000,180000: The Z-axis moves at 180,000 um/s to the 150,000 um position;

4. Zg50000,80: Z-axis at 50,000um/s, 80% of the power is taken as TIP;

5. Zu100000,180000: The Z-axis moves 100,000um upward at 180,000um/s;

6. Dt1000,0:ADP enject the TIP by enjecting TIP at a speed of 1000um/s.

7. Zp0,180000: Z-axis moves to position 0 with a speed of 180,000 um/s.

After adding, as shown in the figure below, click the "Execute List" button to execute the instruction set, ADPZ and ADP will execute each instruction in order.If you want to save the instruction set, click the Export button labeled "2" in the figure below, and if you want to import the saved instruction set, click the Import button labeled "1" in the figure below:

Besides, you can modify the added commands manually.As shown in the figure below, first add the Z-axis zero command three times, then change the command column in the second line to "L3000", and change the description column to "Delay 3 seconds".At this time, click the "Execute List" button, the following actions will be executed sequentially:

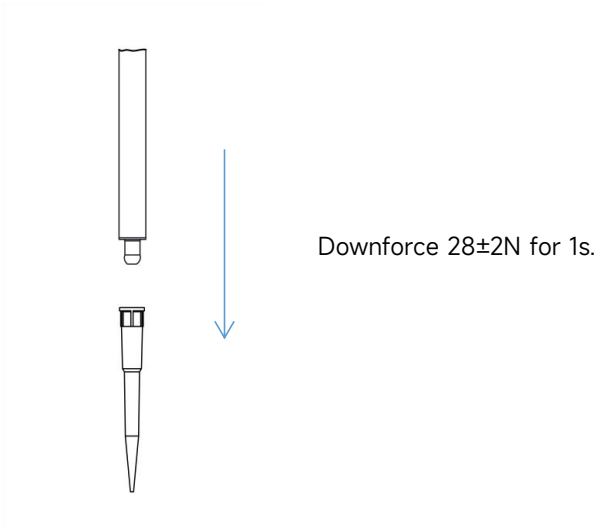
1. Z-axis is set to zero;
2. 3s delay;
3. Z-axis zeroing.

6. Application chapters

6.1 Application process

Application process : TIP pickup and liquid level detection. Recommended communication frameworks and actual communication data are provided in the development process examples in section 8.4. If you have questions about pipettes, you can refer to the datasheet of each pipette.

Pick up TIP



1. Control the Z-axis to move the pipette tip to the position 15-20mm directly above the TIP, available speed movement of 180,000um/s;

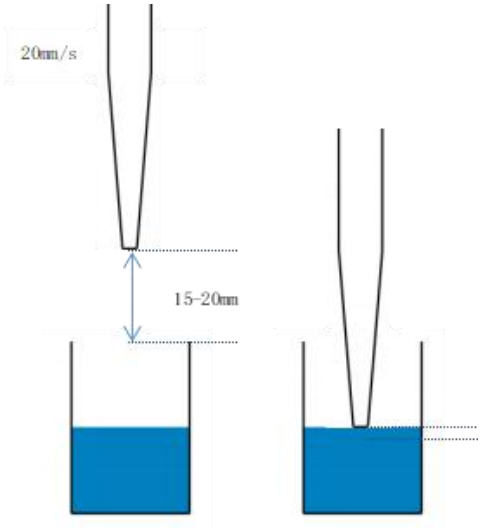
2. Send the command to fetch TIP (Zg: KT_DT protocol, torque mode), suggested parameters: speed 50000um/s (default value), power 80% (default value), maximum position 180000um (default value).

Note: The movement time of fetching TIP should be ⩾0.3s, otherwise it may lead to the failure of fetching TIP; when fetching TIP, the ADP gun head is 15-20mm away from the TIP port, and the power is used at 80%, at this time, the force of fetching TIP is about 28±2N or so. When the power setting is lower than 80%, the program all defaults to use the minimum 80% power, you can modify the minimum power through register 135, see section 9.3.3.1 for details.

Warning:
TIP fetching in location mode is not recommended (Zp, Zd commands - KT_DT protocol)

Liquid level detection

Fast descending pipette, when the TIP tip is located 15-20mm above the test tube mouth Z axis becomes slow (20000um/s) descending, after 500ms start liquid level detection, when the TIP tip touches the liquid level the pipette will feedback signal to make the Z axis stop automatically.



6.2 Blocking Detection

Configure the Z-axis register 110 to 1 to enable the blocking detection function, when the Z-axis movement is jammed or stuck, the Z-axis will stop and report an error.

When the position mode is used to fetch TIP, the blocking detection function cannot be turned on, otherwise the Z-axis will report an error and the fetching of TIP may fail.

When TIP is taken in torque mode, the blocking detection function will be ignored, and if there is stagnation or jamming, Z-axis will not report an error and the TIP is considered to be successful, and the master control will query the TIP in-position detection register of the pipette to determine whether the TIP is successful or not.

7. Communication Protocol

7.1 Communication Method

Communication interface

The Z-axis is controlled with the same interface as the ADP, you can use the following communication method:

- RS232
- RS485
- CAN

Baud rate:

Serial: 9600, 19200, 38400 (default), 115200, with 8 data bits, no parity, 1 stop bit;

CAN: 100K, 125K, 250K, 500K (default), 1000K.

Notice: When using serial communication, the suggested transmission interval is 10ms, following a request-response communication mode.

Protocol

Z-axis supports both serial and CAN communication. The serial port is divided into RS232, RS485 (Only connect with ADP together), both supportKT_DT and KT_OEM Protocol.Both protocols can be used to control command flow, enabling compatibility with ADP. The protocol format is the same as ADP.

Protocol lock: The protocol of the first command received is used as the working protocol, and ADPZ needs to be restarted to switch the protocol. Support KT_CAN_DIC, KT_DT, and KT_OEM protocols.

- KT_DT protocol
- KT_OEM protocol
- KT_CAN_DIC protocol

Notice: When using serial communication, the suggested transmission interval is 10ms, following a request-response communication mode.

KT_DT Protocol

This protocol is based on RS232 and RS485 communication, the communication protocol has no check bit, you can easily use the serial interface debugging tool to send a string to control the Z-axis. Data is sent and received as ASCII string, which is convenient for users to debug. The communication protocol has no check bit so there is a risk of data loss.

KT_OEM Protocol (Recommend)

This protocol is based on RS232 and RS485 communication, the communication protocol includes communication sequences and check byte, which can effectively prevent data loss. During the working period, the controller polls the Z-axis status and analyses the queried status to decide whether the Z-axis completed commands execution or there is an error, please refer to the KT_OEM protocol for details.

KT_CAN_DIC Protocol (Recommend)

This communication protocol is used for local area network CAN communication. It enable the control of the Z-axis and the setting of parameters by reading and writing registers without checking the Z-axis status. The Z-axis will automatically upload the current status whenever there is a change, please refer to the KT_CAN_DIC protocol for details.

7.2 KT_CAN_DIC Protocol Format

Sending data format:

Frame type: data frame

Message identifier type: extended frame

ID: use extended ID, please refer to the following table

DLC: data length 8

Data field: the data length is the fixed 8 bytes, including communication Sequence number, register address, and data as shown in the table below

ID data format:

KT_CAN_DIC Message ID area data format

bit28 ~ 16	bit15 ~ 8	bit7 ~ 0
Commands	Master address	Target device address

Data area data format:

KT_CAN_DIC Message data area data format

byte0	byte1 ~ 2	byte3	byte4 ~ 7
Communication serial number	Object index	Object sub-index	4 bytes of data

ID data:

1. Commands: Used to indicate the type of operation to change the frame data, as shown in KT_CAN_DIC Message Commands:

KT_CAN_DIC Message Commands

Commands	Function	Description
0x0000	Responding	Read and write data return
0x0001	Write	Write object dictionary Return value: status, see status table
0x0002	Read	Read object dictionary, if there is no corresponding object dictionary, no data is returned.
0x0003	Process data	Used for uploading real time data that does not require an answer, e.g. status change active uploading send should via this commands
0x0004	Heartbeat	The heartbeat data uploaded at regular intervals can be used to detect whether the device is online or not, the heartbeat uploaded data is the node status, the status information is shown in the status table
0x0080	Alarm	The device error is actively reported in the commands, the error information is shown in the status table

2. Original address: The sender address.
3. Target address: The receiver address.

数据区数据包括：

1. Communication sequence number: It is used to distinguish which frame of data is sent and received. It is recommended that users add 1 to the communication sequence number before each frame of data is sent so that each frame of data is different.

2. Object index: The object index is 16-bit index address data, which is used to define the object dictionary table, the function of the object dictionary table is shown in the object dictionary table, and the details are shown in the CAN Object Dictionary Chapter

3. Object sub-index: The sub-index is an 8-bit sub-index address data, which is used together with the object index to define the object dictionary table.More details, see table 10-1 KT_CAN_DIC control command.

Each control instruction has a unique primary index and a number of sub-indexes, the first to send the sub-index of non-zero frame data, the last to send the sub-index of zero frame data, the Z-axis begins to move when receiving the sub-index of zero command

4. Data: A data length of 32 bits represents the communication data, which is a signed integer. More details, see table 10-1 KT_CAN_DIC control command.

7.3 KT_OEM Protocol Format

KT_OEM protocol is based on RS232 or RS485 communication, and the string part of the data area is consistent with KT_DT protocol.The communication protocol includes communication sequences and check byte, which can effectively prevent data loss, so it is recommended to use OEM protocol to communicate with Z-axis. During operation, the controller queries the Z-axis status by sending a blank command, analyzes the queried status to determine whether the Z-axis executes a completed command or an error occurs, and the commands to send the data in the data area is empty can query the status.

KT_OEM Send instruction protocol format

Data fields	Data type	Number of bytes	Description
Frame header	Uint8	1	Fixed value 0XAA, indicating the start of data
Indexes	Uint8	1	Instruction index, value ranges 0x80~0xFE. If the index of the current instruction is the same as that of the previous instruction, the current instruction is not executed and only answered
Address	Uint8	1	The value ranges from 0x01 to 0x0F. All devices can receive and execute the command. The broadcast address command does not answer
Data Area length	Uint8	1	Data area data length, range 0x01 ~ 0xFF
Data Area	Byte	n	ASCII commands strings, see 9.3 operation commands for details
Checksum	Uint8	1	The 8-bit checksum is calculated from the frame header to the end byte of the data area. The value obtained takes the last 8 bits of data

KT_OEM return data protocol format

Data field	Data type	Number of bytes	Description
Frame header	Uint8	1	Fixed value 0X55, indicating the start of data
Indexes	Uint8	1	Same as the received instruction index
Address	Uint8	1	Communication address, each drive on each bus should be set to a unique address number, the address is the original address of the device when returning data
Status	Uint8	1	Current module status, see status table

Data length	Uint8	1	The data length of the data area, when the data area is empty, the data length is 0
Data	Char	n	ASCII code return data, see operation commands for details
Checksum	Uint8	1	The 8-bit checksum is calculated from the header of the frame to the end byte of the data area. The value obtained takes the last 8 bits of data

7.4 KT_DT protocol format

Each time the Z-axis receives a string, it parses the string, verifies that the address matches, the syntax is correct, and returns the status of the first instruction execution.

KT_DT protocol send format

Serial number	Function	Number of bytes	Description
1 ~ 2	Address	2	The target communication address should be set as a unique address number for each drive on the bus, with a value range of 1-32. Normal communication occurs only when the address matches; otherwise, received commands will be ignored
3	Direction	1	The character > indicates sending data
3+n	Data	n	ASCII commands strings, see operation commands for details.
4+n	Terminator	1	The carriage return character 0x0D, every communication data should end with a carriage return.

KT_DT protocol return format

Serial number	Function	Number of bytes	Description
1 ~ 2	Address	2	The original device address, which matches the address sent, indicates which address returned the data.
3	Direction	1	The character < indicates return data.
	Status	2	See status table
		1	':' if no return data this symbol is not displayed

	Special symbols:		
4+n	Data	n	Return data string, see the corresponding return data of the operation commands for details
5+n	Terminator	1	The carriage return character 0x0D, every communication data should end with a carriage return.

8. Communication process

Note: For serial communication, the KT_DT protocol does not have data verification. Therefore, you are advised to use the KT_OEM protocol only for debugging. The format of the KT_OEM protocol in the data area is the same as that of the KT_DT protocol, except that the format of the frame header and the frame end data are different. When the KT_OEM protocol is applied, check the status of each step to ensure that the command execution is complete before performing the next step. In the following example, the Z-axis address is 41 (if the z-axis is not connected to the pipette, the address is "1", and the Z-axis address after the pipette is connected is the address of the corresponding pipette plus "40").

8.1 KT_CAN_DIC Protocol single message

Note: See 10.1 Control Commands section

Each control instruction has a unique primary index and a number of sub-indexes, the first to send the sub-index of non-zero frame data, the last to send the sub-index of zero frame data, the Z-axis begins to move when receiving the sub-index of zero command

KT_CAN_DIC 8.1 Protocol single message

Function	Direction	ID	Data	Description
Zero setting	Send	00010029	00 41 00 00 00 00 C3 50	ID: 0001: write commands; 00: source device address;29: target device address; Data: 00: communication sequence number; 41 00: main index (initialization control) 00:sub-index((Initialization process power setting); 00 00 C3 50 data 50000（um/s）
	Receive	00002900	00 41 00 00 00 00 00 02	ID: 0000: answer data; 29: answer device address; 00: receive answer data device

				address; Data: 00: communication sequence (consistent with the sequence in the received commands); 41 00: main index; 00: sub-index; 00 00 00 02: status (commands executed successfully, others see status table)
calibration	Send	00010029	01 90 00 00 00 00 00 00	ID: 00 01: write commands; 00: source device address; 29: target device address; Data: 01: communication sequence number;90 00: main index (calibration); 00: sub-index ; 00 00 00 00： Data.
	Receive	00002900	01 90 00 00 00 00 00 02	ID: 0000: answer data; 29: answer device address; 00: receive answer data device address; 01: communication sequence(consistent with the sequence in the received commands);90 00: main index; 00: sub-index; 00 00 00 02: status (commands executed successfully, others see status table)
Move to designated position	Send	00010029	02 41 01 01 00 02 BF 20	ID: 00 01: write commands; 00: source device address; 29: target device address; Data: 02: communication sequence number;41 01: main index (Move to position); 01: sub-index(movement velocity) ; 00 02 BF 20： Data 180000（um/s）.
	Receive	00002900	02 41 01 01 00 00 00 02	ID: 0000: answer data; 29: answer device address; 00: receive answer data device address; 02: communication sequence(consistent with the sequence in the received commands);41 01: main index; 01: sub-index; 00 00 00 02: status (commands executed successfully, others see status table)
	Re-Send	00010029	03 41 01 00	ID:

			00 01 FB D0	00 01: write commands; 00: source device address; 29: target device address; Data: 03: communication sequence number;41 01: main index (Move to designated position); 00: sub-index(designated position) ; 00 02 BF 20： Data 130000（um）.
	Receive	00002900	03 41 01 00 00 00 00 02	ID: 0000: answer data; 29: answer device address; 00: receive answer data device address; 03: communication sequence(consistent with the sequence in the received commands);41 01: main index; 00: sub-index; 00 00 00 02: status (commands executed successfully, others see status table)
Downward motion in relative position	Send	00010029	04 41 03 01 00 02 BF 20	ID: 00 01: write commands; 00: source device address; 29: target device address; Data: 04: communication sequence number;41 03: main index (Downward motion in relative position); 00: sub-index(movement velocity) ; 00 02 BF 20： Data 180000（um/s）.
	Receive	00010029	04 41 03 01 00 00 00 02	ID: 0000: answer data; 29: answer device address; 00: receive answer data device address; 04: communication sequence(consistent with the sequence in the received commands);41 03: main index; 00: sub-index; 00 00 00 02: status (commands executed successfully, others see status table)
	Re-send	00010029	05 41 03 00 00 00 4E 20	ID: 00 01: write commands; 00: source device address; 29: target device address; Data: 05: communication sequence number;41 03: main index (Downward motion in

				relative position); 00: sub-index(the distance of downward motion) ; 00 00 4E 20：Data 20000（um）
	Receive	00010029	05 41 03 00 00 00 00 02	ID: 0000: answer data; 29: answer device address; 00: receive answer data device address; 05: communication sequence(consistent with the sequence in the received commands);41 03: main index; 00: sub-index; 00 00 00 02: status (commands executed successfully, others see status table)
Move down to pick up the TIP	Send	00010029	06 41 04 01 00 00 00 50	ID: 00 01: write commands; 00: source device address; 29: target device address; Data: 06: communication sequence number;41 04: main index (Downward motion to pick up TIP); 01: sub-index(power) ; 00 00 00 50：Data 80%.
	Receive	00010029	06 41 04 01 00 00 00 02	ID: 0000: answer data; 29: answer device address; 00: receive answer data device address; 06: communication sequence(consistent with the sequence in the received commands);41 04: main index; 01: sub-index; 00 00 00 02: status (commands executed successfully, others see status table)
	Send	00010029	07 41 04 02 00 02 BF 20	ID: 00 01: write commands; 00: source device address; 29: target device address; Data: 07: communication sequence number;41 04: main index (Downward motion to pick up TIP); 02: sub-index(posision) ; 00 02 BF 20：Data 180000(um).
	Receive	00010029	07 41 04 02 00 00 00 02	ID: 0000: answer data; 29: answer device address; 00: receive answer data device address; 07: communication sequence(consistent with the sequence

				in the received commands);41 04: main index; 02: sub-index; 00 00 00 02: status (commands executed successfully, others see status table)
	Re-send	00010029	08 41 04 00 00 00 C3 50	ID: 00 01: write commands; 00: source device address; 29: target device address; Data: 08: communication sequence number;41 04: main index (Downward motion to pick up TIP); 00: sub-index(The velocity of picking up TIP) ; 00 00 C3 50： Data 50000um/s.
	Receive	00010029	08 41 04 00 00 00 00 02	ID: 0000: answer data; 29: answer device address; 00: receive answer data device address; 07: communication sequence(consistent with the sequence in the received commands);41 04: main index; 02: sub-index; 00 00 00 02: status (commands executed successfully, others see status table)
Upward motion in relative position	Send	00010029	09 41 02 01 00 02 BF 20	ID: 00 01: write commands; 00: source device address; 29: target device address; Data: 09: communication sequence number;41 02: main index (Upward motion at relative position); 01: sub-index(posision) ; 00 02 BF 20： Data 180000(um).
	Receive	00010029	09 41 02 01 00 00 00 02	ID: 0000: answer data; 29: answer device address; 00: receive answer data device address; 09: communication sequence(consistent with the sequence in the received commands);41 02: main index; 00: sub-index; 00 00 00 02: status (commands executed successfully, others see status table)
	Re-send	00010029	0A 41 02 00 00 01 FB D0	ID: 00 01: write commands; 00: source device address;

				29: target device address; Data: 09: communication sequence number;41 02: main index (the Upward motion at relative position); 00: sub-index(the distance of upward motion) ; 00 01 FB D0：Data 130000（um）.
	Receive	00010029	0A 41 02 00 00 00 00 02	ID: 0000: answer data; 29: answer device address; 00: receive answer data device address; 0A: communication sequence(consistent with the sequence in the received commands);41 02: main index; 00: sub-index; c。
Check status	Send	00020029	0B 20 00 01 00 00 00 00	ID: 00 02: write commands; 00: source device address; 29: target device address; Data: 0B: communication sequence number;20 00: main index (Register read and write); 01: sub-index(Status) ; 00 00 00 00：Data.
		00000029	0B 20 00 01 00 00 00 00	ID: 0000: answer data; 29: answer device address; 00: receive answer data device address; 0B: communication sequence(consistent with the sequence in the received commands);20 00: main index; 01: sub-index; 00 00 00 00：Register value (free, see status table for others).
Check register	Send	00020029	0C 20 00 5A 00 00 00 00	ID: 00 02: write commands; 00: source device address; 29: target device address; Data: 0C: communication sequence number;20 00: main index (Register read and write); 5A: sub-index(Register address) ; 00 00 00 00：Data.
		00000029	0C 20 00 5A 00 00 00 29	ID: 0000: answer data; 29: answer device address; 00: receive answer data device address; 0C: communication

				sequence(consistent with the sequence in the received commands);20 00: main index; 5A: sub-index; 00 00 00 29: Register value.
Write register	Send	00010029	0D 20 00 83 00 00 00 01	ID: 00 01: write commands; 00: source device address; 29: target device address; Data: 0D: communication sequence number;20 00: main index (Register read and write); 83: sub-index(Register address) ; 00 00 00 01: Data:Write register value
		00000029	0D 20 00 83 00 00 00 02	ID: 0000: answer data; 29: answer device address; 00: receive answer data device address; 0D: communication sequence(consistent with the sequence in the received commands);20 00: main index; 83: sub-index; 00 00 00 02: Status (instruction executed successfully, see the status table for others)
power-down store	Send	00010001	0E 9F 10 00 00 00 00 00	ID: 00 01: write commands; 00: source device address; 29: target device address; Data: 0E: communication sequence number;9F 10: main index (Register); 00: sub-index; 00 00 00 00: Data 0
	Receive	00002900	0E 9F 10 00 00 00 00 02	ID: 0000: answer data; 29: answer device address; 00: receive answer data device address; Data: 0E: communication sequence;9F 10: main index; 00: sub-index(Register address 110: locked-rotor detection); 00 00 00 02: Data 2 (status 2 executed successfully, see the status table for others)

8.2 KT_OEM Example of OEM Protocol Application (HEX ModeSend)

Note: The KT_OEM protocol is an encapsulation of the DT protocol, and the command string and return string are described in Section 9.3.

KT_OEM Protocol Single Command Examples

Function	Direction	Message	Description
Zero setting	Send	AA 80 29 07 5A 7A 35 30 30 30 30 23	AA: frame header; 80: index; 29: target device address; 07: command string length; 5A 7A 35 30 30 30 30: string command "Zz50000"; 23: frame checksum
	Receive	55 80 29 02 00 00	55: frame header;80: index; 29: target device address; 02: status of successful commands execution (others see status table); 00: return string length; 00: frame checksum
calibration	Send	AA 81 29 02 5A 63 13	AA: frame header; 81: index; 29: target device address; 02: command string length; 5A 63: string command "Zc"; 13: frame checksum
	Receive	55 81 29 02 00 01	55: frame header;80: index; 29: target device address; 02: status of successful commands execution (others see status table); 00: return string length; 01: frame checksum
Move to designated position	Send	AA 82 29 0F 5A 70 31 33 30 30 30 30 2C 31 38 30 30 30 30 A7	AA: frame header; 82: index; 29: target device address; 0F command string length; 5A 70 31 33 30 30 30 30 2C 31 38 30 30 30 30:string command "Zp130000,180000"; A7: frame checksum
	Receive	55 82 29 02 00 02	55: frame header;82: index; 29: target device address; 02: status of successful commands execution (others see status table); 00: return string length; 02: frame checksum
Downward motion in relative position	Send	AA 83 29 0E 5A 64 32 30 30 30 30 2C 31 38 30 30 30 30 69	AA: frame header; 29: target device address; 0E command string length; 5A 64 32 30 30 30 30 2C 31 38 30 30 30 30:string command "Zd20000,180000"; 69: frame checksum
	Receive	55 83 29 02 00 03	55: frame header;83: index; 29: target device address; 02: status of successful commands execution (others see status table); 00: return string length; 03: frame checksum
Pick up the TIP	Send	AA 84 29 11 5A 67 35 30 30 30 30 2C	AA: frame header;84: index; 29: target device address; 11 command string length; 5A 67 35 30 30 30 30 2C 38 30 2C 31 38 30 30 30 30:string

		38 30 2C 31 38 30 30 30 30 07	command “Zg50000,80,180000”; 07: frame checksum
	Receive	55 84 29 02 00 04	55: frame header;84: index; 29: target device address; 02: status of successful commands execution (others see status table); 00: return string length; 04: frame checksum
Upward motion in relative position	Send	AA 85 29 0F 5A 75 31 33 30 30 30 30 2C 31 38 30 30 30 30 AF	AA: frame header;84: index; 29: target device address; 0F command string length; 5A 75 31 33 30 30 30 30 2C 31 38 30 30 30 30 2A: string command “Zu130000,180000”; AF: frame checksum
	Receive	55 85 29 02 00 05	55: frame header;85: index; 29: target device address; 02: status of successful commands execution (others see status table); 00: return string length; 05: frame checksum
Check status	Send	AA 86 29 01 3F 99	AA: frame header;86: index; 29: target device address; 01 command string length; 3F:string command “?”; 99: frame checksum
	Receive	55 86 29 00 00 04	55: frame header;85: index; 29: target device address; 00: idle state 00: return string length; 0:4: frame checksum
Check register	Send	AA 87 29 04 52 72 39 30 8B	AA: frame header;87: index; 29: target device address; 04: command string length; 52 72 39 30:string command “Rr90”; 8B: frame checksum
	Receive	55 87 29 02 02 34 31 6E	55: frame header;87: index; 29: target device address; 02: status of successful commands execution (others see status table); 02: return string length; 34 31: The query result: 41 6E: frame checksum
Write register	Send	AA 88 29 07 57 72 31 33 31 2C 31 1D	AA: frame header;88: index; 29: target device address; 07: command string length; 57 72 31 33 31 2C 31: string command “Wr131,1”; 1D: frame checksum
	Receive	55 88 29 02 00 08	55: frame header;88: index; 29: target device address; 02: status of successful commands execution (others see status table); 00: return string length; 08: frame checksum
power-down store	Send	AA 89 29 01 53 B0	AA: frame header;89: index; 29: target device address; 01: command string length; 53: string command “S”; B0: frame checksum
	Receive	55 89 29 02 00 09	55: frame header;89: index; 29: target device address; 02: status of successful commands execution ; 00: return string length; 09: frame

			checksum

8.3 Example for KT DT Single Command (Sending in String Mode)

KT DT Single Command

Function	Direction	Message	Description
Zero setting	Send	41>Zz50000	41: target device address;50000:Zero operation speed of 50,000 um/s
	Receive	41<2	41: target device address; 2: status of successful commands execution (others see status table)
calibration	Send	41>Zc	41: target device address;
	Receive	41<2	41: target device address; 2: status of successful commands execution (others see status table)
Move to designated position	Send	41>Zp100000 ,180000	41: target device address; 100000 moving to 100000 um position; 180000： The speed of motion to the specified position is 180,000 um/s.
	Receive	41<2	41: target device address; 2: status of successful commands execution (others see status table)
Downward motion in relative position	Send	41>Zd20000, 180000	41: target device address; 20000: Current location moves to 20000 um position; 180000： The speed of motion to the specified position is 180,000 um/s.
	Receive	41<2	41: target device address; 2: status of successful commands execution (others see status table)
Pick up the TIP	Send	41>Zg50000, 80,180000	41: target device address; 50000: The speed of motion to the specified position is 50,000 um/s;80:power;180000:The maximum position to which TIP is moved.
	Receive	41<2	41: target device address; 2: status of successful commands execution (others see status table)
Upward motion in relative position	Send	41>Zu50000, 130000	41: target device address; 50000: Current location moves to 50000 um position; 130000： The speed of motion to the specified position is 130000 um/s.
	Receive	41<2	41: target device address; 2: status of successful commands execution (others see status table)
Check status	Send	41>?	41: target device address; ?:Used to query the current Z-axis status, equivalent to read the parameter Settings of the register address 1.
	Receive	41<0	41: target device address; 0:idle state (others see

			status table)
Check register	Send	41>Rr90	41: target device address; 90: Query the value of register 90.
	Receive	41<2:41	41: target device address; 2: status of successful commands execution (others see status table); 41: The value of the 121 register.
Set parameters	Send	41>Wr131,1	1: target device address; Wr131: Write 131 register,set holding force mode (other register address view parameter Settings); 1: Write the value of register 131 to 1.
	Receive	41<2	1: target device address; 2: status of successful commands execution (others see status table)
power-down store	Send	41>S	41: target device address; S:Power down save command.
	Receive	41<2	1: target device address; 2: status of successful commands execution (others see status table)

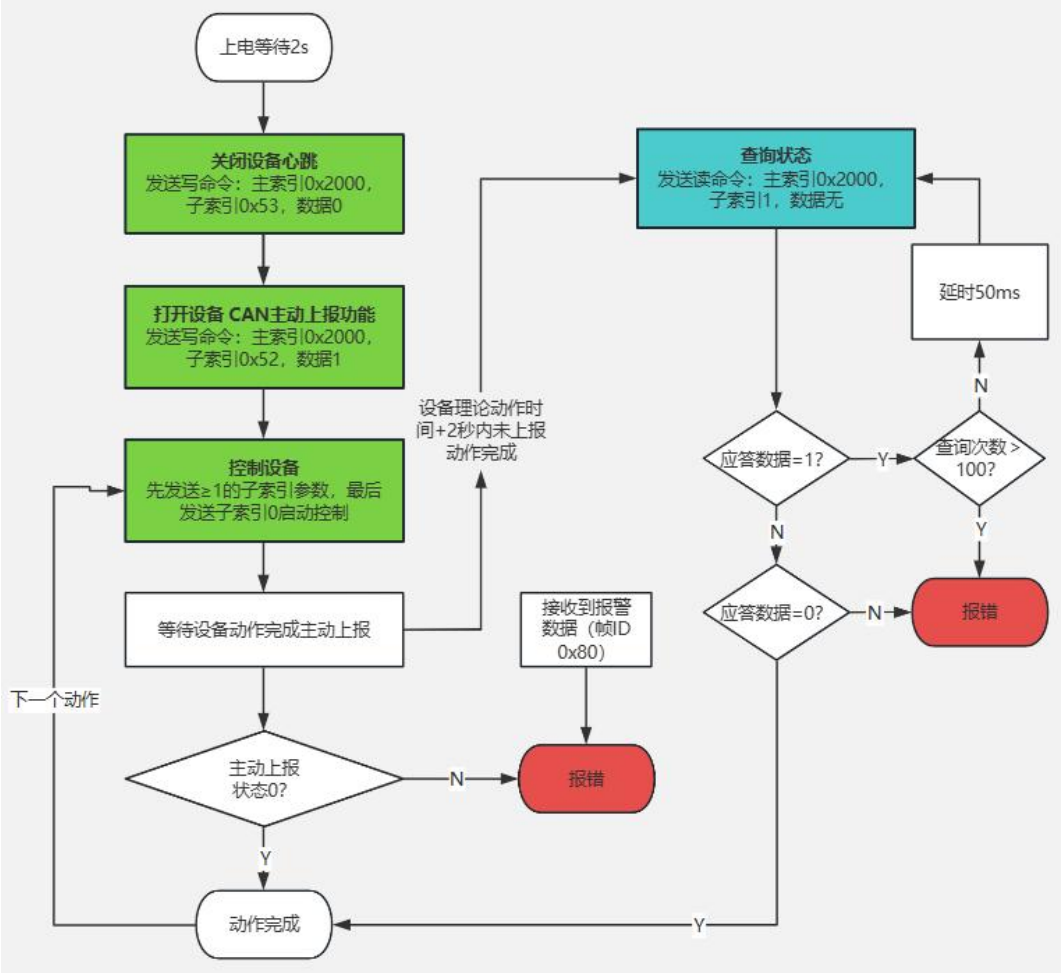
Note: The sending direction refers to the Z axis from the main control, and the receiving direction refers to the Z axis to the main control device.

8.4 Development Process Example

CAN Communication flow

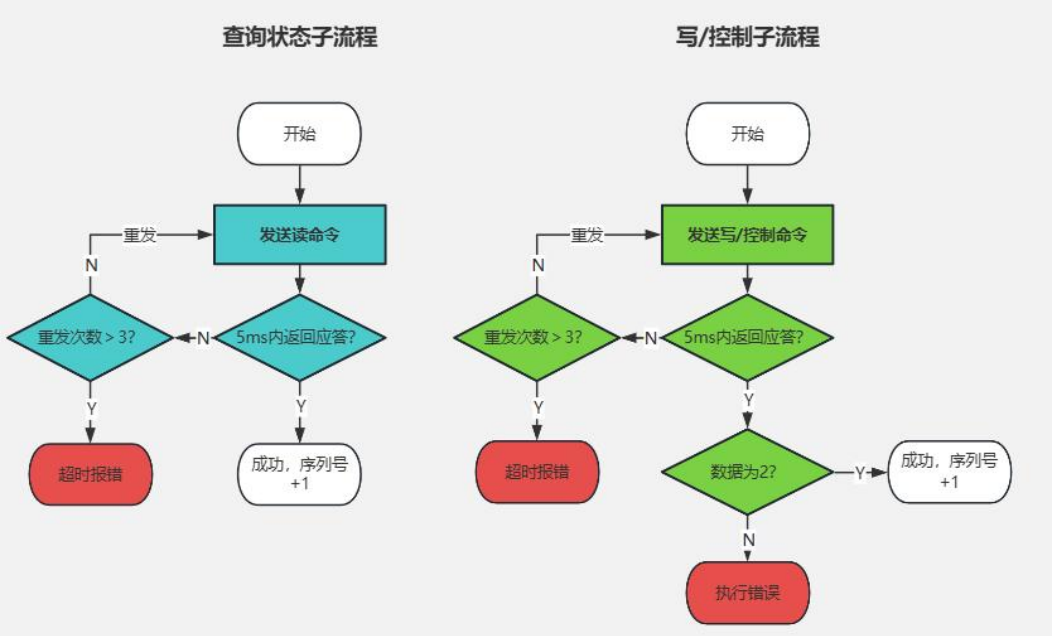
Note: Recommend the communication method be in question-and-answer mode, i.e. each time you send a command, wait until it is answered before sending a new command.

Configure automatically reporting via main index 0x2000 sub-index 0x52 to use the automatically reporting communication mode.



CAN Protocol Communication Framework

The green box indicates writing registers and controlling SP20 operation, see Figure 8-2 for details. The blue box indicates querying the status of SP20 and reading the registers, see KT_CAN_DIC Protocol Communication Sub flow Framework for details.

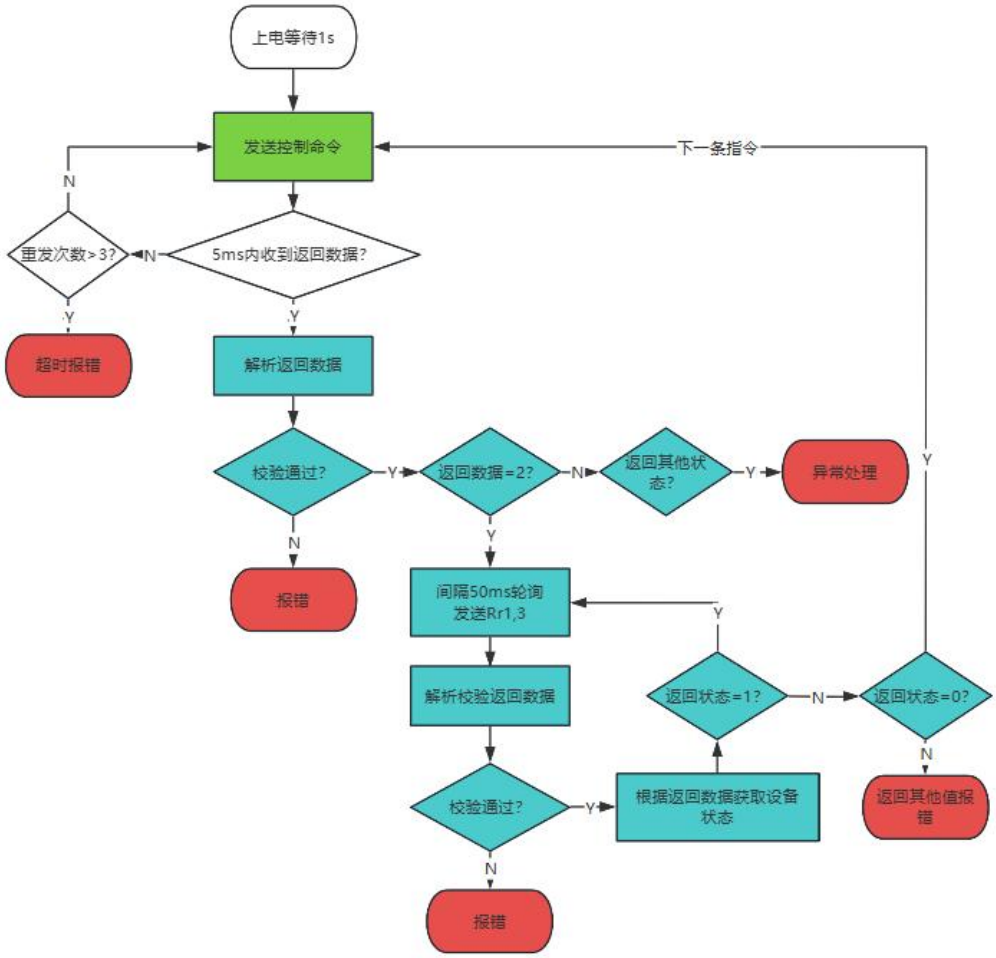


KT_CAN_DIC Protocol Communication Subflow Framework

Serial Port Communication Process

Note: When using serial communication, it is recommended to send the next frame of data in 10ms interval after the command is answered to avoid bus interference; the communication mode adopts one-question-one-answer mode, i.e., each time you send a command, wait until it is answered and then send a new command.

See KT_OEM Protocol Communication Framework for a question-and-answer model:



KT_OEM Protocol Communication Framework

KT_CAN_DIC CAN Development Process Monitoring Example

KT_CAN_DIC Development Process

No.	Transmission direction	Description	Frame ID	Format	Type	DLC	Data
0	Receive	Power-up default upload heartbeat	0x00040100	Data frame	Extended frame	0x08	00 00 00 00 00 00 00 00
1	Receive	Power-up default upload heartbeat	0x00040100	Data frame	Extended frame	0x08	00 00 00 00 00 00 00 00
2	Receive	Power-up default upload heartbeat	0x00042900	Data frame	Extended frame	0x08	01 00 00 00 00 00 00 00
3	Receive	Power-up default upload heartbeat	0x00040100	Data frame	Extended frame	0x08	01 00 00 00 00 00 00 00
4	Receive	Power-up default upload heartbeat	0x00042900	Data frame	Extended frame	0x08	02 00 00 00 00 00 00 00
5	Send	Turn off Z-axis heartbeat	0x00010029	Data frame	Extended frame	0x08	01 20 00 6B 00 00 00 00
6	Receive		0x00002900	Data frame	Extended frame	0x08	01 20 00 6B 00 00 00 02
7	Send	Turn off SP20 heartbeat	0x00010001	Data frame	Extended frame	0x08	01 20 00 53 00 00 00 00
8	Receive		0x00000100	Data frame	Extended frame	0x08	01 20 00 53 00 00 00 02
9	Send	Turn on SP20, Complete active reporting	0x00010001	Data frame	Extended frame	0x08	02 20 00 52 00 00 00 01
10	Receive		0x00000100	Data frame	Extended frame	0x08	02 20 00 52 00 00 00 02
11	Send	Turn on Z-axis, Complete active reporting	0x00010029	Data frame	Extended frame	0x08	02 20 00 52 00 00 00 01
12	Receive		0x00002900	Data frame	Extended frame	0x08	02 20 00 52 00 00 00 02
13	Send	Configure pipettor initialisation power	0x00010001	Data frame	Extended frame	0x08	03 40 00 01 00 00 00 64

		100					
14	Receive		0x00000100	Data frame	Extended frame	0x08	03 40 00 01 00 00 00 02
15	Send	Configure pipettor initialisation to eject TIPs	0x00010001	Data frame	Extended frame	0x08	04 40 00 02 00 00 00 00
SP16	Receive		0x00000100	Data frame	Extended frame	0x08	04 40 00 02 00 00 00 02
17	Send	Perform pipettor initialisation	0x00010001	Data frame	Extended frame	0x08	05 40 00 00 00 00 FA 00
18	Receive		0x00000100	Data frame	扩展帧 Extended frame	0x08	05 40 00 00 00 00 00 02
19	Receive	Pipettor operation completion frame	0x00030100	Data frame	Extended frame	0x08	** 70 02 00 00 00 00 00
20	Send	Z-axis initialisation	0x00010029	Data frame	Extended frame	0x08	** 41 00 00 00 00 C3 50
21	Receive		0x00002900	Data frame	Extended frame	0x08	** 41 00 00 00 00 00 02
22	Receive	Z-axis operation completion frame	0x00032900	Data frame	Extended frame	0x08	** 70 02 00 00 00 00 00
23	Send	Configure Pipettor aspiration speed 100	0x00010001	Data frame	Extended frame	0x08	** 40 01 01 00 00 00 64
24	Receive		0x00000100	Data frame	Extended frame	0x08	** 40 01 01 00 00 00 02
25	Send	Configure Pipettor with a aspiration interruption speed of 0	0x00010001	Data frame	Extended frame	0x08	** 40 01 02 00 00 00 00
26	Receive		0x00000100	Data frame	Extended frame	0x08	** 40 01 02 00 00 00 02
27	Send	Perform Pipettor aspiration empty/air of 30.00ul	0x00010001	Data frame	Extended frame	0x08	** 40 01 00 00 00 0B B8
28	Receive		0x00000100	Data frame	Extended frame	0x08	** 40 01 00 00 00 00 02
29	Receive		0x00030100	数据帧 Data frame	扩展帧 Extended frame	0x08	** 70 02 00 00 00 00 00
30	Send	Configure Z-axis to pick up TIPs of	0x00010029	Data frame	Extended frame	0x08	** 41 04 01 00 00 00 50

		80% power					
31	Receive		0x00002900	Data frame	Extended frame	0x08	** 41 04 01 00 00 00 02
32	Send	Execute Z-axis to pick up TIPs	0x00010029	Data frame	Extended frame	0x08	** 41 04 00 00 00 4E 20
33	Receive		0x00002900	Data frame	Extended frame	0x08	** 41 04 00 00 00 00 02
34	Receive	Pipettor picked up the TIP	0x00030100	Data frame	Extended frame	0x08	** 70 01 00 00 00 00 01
35	Receive	Z-axis operation completion frame	0x00032900	Data frame	Extended frame	0x08	** 70 02 00 00 00 00 00
36	Send	Configure Z-axis moving speed 30000	0x00010029	Data frame	Extended frame	0x08	** 41 01 01 00 00 75 30
37	Receive		0x00002900	Data frame	Extended frame	0x08	** 41 01 01 00 00 00 02
38	Send	Execute Z-axis move position 0	0x00010029	Data frame	Extended frame	0x08	** 41 01 00 00 00 00 00
39	Receive		0x00002900	Data frame	Extended frame	0x08	** 41 01 00 00 00 00 02
40	Receive	Z-axis operation completion frame	0x00032900	Data frame	Extended frame	0x08	** 70 02 00 00 00 00 00
41	Send	Configure Z-axis moving speed 30000	0x00010029	Data frame	Extended frame	0x08	** 41 01 01 00 00 75 30
42	Receive		0x00002900	Data frame	Extended frame	0x08	** 41 01 01 00 00 00 02
43	Send	Execute Z-axis move position 180000	0x00010029	Data frame	Extended frame	0x08	** 41 01 00 00 02 BF 20
44	Receive		0x00002900	Data frame	Extended frame	0x08	** 41 01 00 00 00 00 02
45	Send	Configuring SP18 liquid level detection without detecting timeout	0x00010001	Data frame	Extended frame	0x08	** 40 07 01 00 00 00 00
46	Receive		0x00000100	Data frame	Extended frame	0x08	** 40 07 01 00 00 00 02
47	Send	Perform pipettor liquid level detection	0x00010001	Data frame	Extended frame	0x08	** 40 07 00 00 00 00 00
48	Receive		0x00000100	Data frame	Extended frame	0x08	** 40 07 00 00 00 00 02
49	Receive	Z-axis operation	0x00032	Data	Extende	0x08	** 70 02 00

		completion frame	900	frame	d frame		00 00 00 00
50	Receive	Pipettor operation completion frame	0x00030100	Data frame	Extended frame	0x08	** 70 02 00 00 00 00 00
		...Perform mixing of aspiration					
51	Send	Perform mixing of aspirates 100.00ul	0x00010001	Data frame	Extended frame	0x08	** 40 01 00 00 00 27 10
52	Receive		0x00000100	Data frame	Extended frame	0x08	** 40 01 01 00 00 00 02
53	Receive	Pipettor operation completion frame	0x00030100	Data frame	Extended frame	0x08	** 70 02 00 00 00 00 00
54	Send	Perform Pipettor initialisation zero	0x00010001	Data frame	Extended frame	0x08	** 40 00 00 00 00 FA 00
55	Receive		0x00000100	Data frame	Extended frame	0x08	** 40 00 00 00 00 00 02
56	Receive	Pipettor operation completion frame	0x00030100	Data frame	Extended frame	0x08	** 70 02 00 00 00 00 00
		...Cycle 51~56 steps for mixing					
54	Send	Configuration pipettor aspiration anomaly detection	0x00010001	Data frame	Extended frame	0x08	** 20 00 3C 00 00 00 05
55	Receive		0x00000100	Data frame	Extended frame	0x08	** 20 00 3C 00 00 00 02
56	Send	Configure pipettor to liquid level following the test tube area 90	0x00010001	Data frame	Extended frame	0x08	** 20 00 68 00 00 00 5A
57	Receive		0x00000100	Data frame	Extended frame	0x08	** 20 00 68 00 00 00 02
58	Send	Configure pipettor for a aspiration speed of 100	0x00010001	Data frame	Extended frame	0x08	** 40 01 01 00 00 00 64
59	Receive		0x00000100	Data frame	Extended frame	0x08	** 40 01 01 00 00 00 02
60	Send	Configure pipettor aspiration interruption speed 0	0x00010001	Data frame	Extended frame	0x08	** 40 01 02 00 00 00 00
61	Receive		0x00000100	Data frame	Extended frame	0x08	** 40 01 02 00 00 00 02
62	Send	Perform pipettor aspiration liquid	0x00010001	Data frame	Extended frame	0x08	** 40 01 00 00 00 27 10

		100.00ul					
63	Receive		0x00000100	Data frame	Extended frame	0x08	** 40 01 00 00 00 00 02
64	Receive	Z-axis operation completion frame	0x00032900	Data frame	Extended frame	0x08	** 70 02 00 00 00 00 00
65	Receive	Pipettor operation completion frame	0x00030100	Data frame	Extended frame	0x08	** 70 02 00 00 00 00 00
66	Send	Configure Pipettor aspiration liquid level following closure	0x00010001	Data frame	Extended frame	0x08	** 20 00 68 00 00 00 00
67	Receive		0x00000100	Data frame	Extended frame	0x08	** 20 00 68 00 00 00 02
68	Send	Configuration Z-axis move speed 40,000	0x00010029	Data frame	Extended frame	0x08	** 41 01 01 00 00 9C 40
69	Receive		0x00002900	Data frame	Extended frame	0x08	** 41 01 01 00 00 00 02
70	Send	Configuration Z-axis moving position 0	0x00010029	Data frame	Extended frame	0x08	** 41 01 00 00 00 00 00
71	Receive		0x00002900	Data frame	Extended frame	0x08	** 41 01 00 00 00 00 02
72	Receive	Z-axis operation completion frame	0x00032900	Data frame	Extended frame	0x08	** 70 02 00 00 00 00 00
73	Send	Configure pipettor re-aspiration volume 0.00ul during dispensing liquid	0x00010001	Data frame	Extended frame	0x08	** 40 02 01 00 00 00 00
74	Receive		0x00000100	Data frame	Extended frame	0x08	** 40 02 01 00 00 00 02
75	Send	Configure pipettor for a dispensing speed of 400ul/s	0x00010001	Data frame	Extended frame	0x08	** 40 02 02 00 00 01 90
76	Receive		0x00000100	Data frame	Extended frame	0x08	** 40 02 02 00 00 00 02
77	Send	configure pipettor aspiration interruption speed 0ul/s	0x00010001	Data frame	Extended frame	0x08	** 40 02 03 00 00 00 00
78	Receive		0x00000100	Data frame	Extended frame	0x08	** 40 02 03 00 00 00 02

79	Send	Perform pipettor dispensing liquid 130.00ul	0x00010001	Data frame	Extended frame	0x08	** 40 02 00 00 00 32 C8
80	Receive		0x00000100	Data frame	Extended frame	0x08	** 40 02 00 00 00 00 02
81	Receive	Pipettor operation completion frame	0x00030100	Data frame	Extended frame	0x08	** 70 02 00 00 00 00 00
82	Send	Close pipettor aspiration liquid anomaly detection	0x00010001	Data frame	Extended frame	0x08	** 20 00 3C 00 00 00 00
83	Receive		0x00000100	Data frame	Extended frame	0x08	** 20 00 3C 00 00 00 02
84	Send	Configure pipettor to always eject TIP	0x00010001	Data frame	Extended frame	0x08	** 40 06 01 00 00 00 00
85	Receive		0x00000100	Data frame	Extended frame	0x08	** 40 06 01 00 00 00 02
86	Send	Pipettor execute eject TIP	0x00010001	Data frame	Extended frame	0x08	** 40 06 00 00 00 7D 00
87	Receive		0x00000100	Data frame	Extended frame	0x08	** 40 06 00 00 00 00 02
88	Receive	Pipettor operation completion frame	0x00030100	Data frame	Extended frame	0x08	** 70 02 00 00 00 00 00

**: It is recommended that the user +1 the communication serial number before sending each frame of data to make sure each frame is different and the device will reply with the same serial number for that frame each time it answers.

串口开发流程监控实例

Z 轴和 SP18 串口开发流程

方向	消息	功能	指令字符串 ASCII
Send	AA29075A7A31303030309F	Z-axis initialisation	Zz10000
Response	5529020080		
Send	AA29013F13	Query Z-axis status	
Response	552901007F	Z-axis status 01 busy	
	.....	Omitted (continue polling status until idle)	
Send	AA29013F13	Query Z-axis status	
Response	552900007e	Z-axis state 0 idle	
Send	AA010D4974363430303302C3130302C3088	Pipettor initialisation	It64000,100,0

Response	5501020058		
Send	AA01013FEB	Query Pipettor status	
Response	5501010057	Pipettor status 01 busy	
	.....	Omitted (continue polling status until idle)	
Send	AA01013FEB	Query Pipettor status	
Response	5501000056	Pipettor status 0 idle	
Send	AA290A5A6732303030302C383024	Z-axis downward movement pick up TIP	Zg50000,80
Response	5529020080		
Send	AA29013F13	Query Z-axis status	
Response	552900007e	Z-axis state 0 idle	
Send	AA0103527233A5	Query TIP status	Rr3
Response	55010201318A		
Send	AA010B57723130302C313030303032D	Configuration of liquid level detection following	Wr100,10000
Response	5501020058		
	.....	Omitted	
Response	5501000056	Idle	
Send	AA010C4961333030302C3130302C303D	Pipettor aspirate air 30.00ul	la3000,100,0
Response	5501020058		
	.....	Omitted	
Response	5501000056	Idle	
Send	AA01054C64302C30EC	Turn on Pipettor liquid level detection	Ld0,0
Response	5501020058		
	.....	Omitted	
Response	5501000056	Idle	
Send	AA011D7B4961313030303032C3130302C3049743634303030302C3130302C327D357B	Pipettor aspiration 100.00ul-initial emptying, cycle 5 times for mixing	{la10000,100,0lt64000,100,2}5
Response	5501020058		
Send	AA0106577236302C3541	Setting of aspiration abnormality detection	Wr60,5
Response	5501020058		
Send	AA010D4961313030303032C3130302C306C	100.00ul Pipettor aspiration 100.00ul	la10000,100,0
Response	5501020058		
	.....	Omitted	
Response	5501000056	Idle	
Send	AA29095A70302C3830303030FA	0 Z-axis rises to position 0	Zp0,80000

Response	5529290200a9		
	.....	Omitted	
Response	552900007e	Idle	
Send	AA010F446131333030302C302C3130302C30C8	Pipettor dispense liquid 130ul	Da13000,0,100,0
Response	5501020058		
	.....	Omitted	
Response	5501000056	Idle	
	AA0106577236302C303C	Turn off aspiration anomaly detection	Wr60,0
	5501020058		
Send	AA0109447436343030302C30C2	TIP Pipettor execute eject TIP	Dt64000,0
Response	5501020058		
	.....	Omitted	
Response	5501000056	Idle	

9. Serial Interface Commands

This chapter describes the data format in partial data area of the KT_OEM、KT_DT protocol, i.e. the format of the operation commands. The data is an ASCII string and multiple commands sets can be sent at the same time, Z-axis and Pipettor will parse and execute the command sets one by one. These commands are classified by function as follows:

- Initialization commands
- Control commands
- Parameter read/write commands
- Query commands
- System control commands

9.1 Commands Syntax

Send multiple commands to Z-axis, as follows:

<CMD><n1,n2,n3><CMD><n1,n2,n3><CMD><n1,n2,n3>

Returns data ASCII format as follows:

<n1,n2,n3>

Among them:

<CMD>: stands for commands, represented by the letters a-z and A-Z, up to two letters, please refer to commands details.

<n1,n2,n3>: indicates commands parameters, commands parameters are separated by ' , ' signs,

and the commands without parameters can be empty. If some of the parameters need to be by default, you can fill in the parameters before the corresponding semicolon as empty. For example, the second parameter will be empty commands ID1000,,2. If the next part of the parameters are empty, it can be omitted. For example, the last two parameters are empty commands ID1000 means that the last two parameters are empty.

Caution:

<> is used to differentiate data blocks and does not need to be sent

The commands are case-sensitive;

We agree that instruction letters are up to 2 letters long, and that two-letter instructions are uppercase letters following a lowercase letter, one-letter instructions are in uppercase, and special characters?are query instructions, {} are loop control instructions, instructions with one capital letter alone are system control, and instructions with a capital letter followed by a lowercase letter are run control instructions.

9.2 Status

Each command has a return status, which indicates whether the commands are executed successfully, whether the device is error, busy, idle, etc. The status is 1 byte of hexadecimal data.

Our convention:

0-9：Working status.

10-19：Commands execution error status.

≥50：Equipment failures

84：未校准错误

Status Table

Value	Function	Description
0	Free	Device is in idle status
1	Busy	The device is operating
2	Executed successfully	Commands executed successfully
…		
10	Parameter overrun	Commands parameters out of range
11	Parameter error	Commands parameter error
12	Syntax error	Commands syntax error
13	Invalid commands	This command is not supported
14	Address error	Read and write register address error
15	Prohibit writing	This address is inhibited from being written to
16	Prohibit reading	This address is inhibited from being readout
18	Z-axis not initialised	Z-axis not initialised

19	Z-axis not connected	Z-axis not connected
…		
80	Z-axis Motor blockage error	Need to re-initialize and troubleshoot
81	Z-axis motor drive failure	Need to re-initialize and troubleshoot
82	Z-axis motor optocoupler error	Need to re-initialize and troubleshoot
83	Z-axis storage error	Need to re-initialize and troubleshoot
84	Z-axis not calibrated error	Need to re-initialize then calibration

9.3 Commands Details

This section describes the detailed individual commands, and the installation functions are divided into the following categories.

- Initialization commands
- Control commands
- Parameter setting commands
- Query commands
- Circulation control commands

Caution:

[] represents an optional parameter, if the optional parameter is empty it will be executed according to the default parameter in the protocol.

The content inside <> is a command, the mark <> does not need to be sent, it is just used to distinguish between letters and commands.

Z-axis initialization commands

<Zz>n1 Z-axis initialization

Power-up and initialize the Z-axis to find the top 0 position. Other movements are only allowed after the initialization is completed. It is recommended to use the default values for theinitialization speed.

Initialization commands

Command value	Parameters	Data range	Unit	Default value	Description
Zz	n1	0~180000	um/s	50000	Initialization speed

Return data: see Status Table

Control Commands

<Zp>n1,[n2] Move to specified position

Control the Z-axis move to the specified position.

Move to specified position

Command value	Parameters	Data range	Unit	Default value	Description
Zp	n1	0~180000	um	0	Position
	[n2]	0~180000	um/s	50000	Speed

Return data: see Status Table

<Zu>n1,[n2] Move upward by a relative distance

Control the Z-axis move upward to the specified position relative to the current position.

Move upward by a relative distance

Command value	Parameters	Data range	Unit	Default value	Description
Zu	n1	0~180000	um	0	Relative distance position
	[n2]	0~180000	um/s	50000	Operating speed

Return data: see Status Table

<Zd>n1,[n2] Move downward by a relative distance

Control the Z-axis move upward to the specified position relative to the current position.

Move downward by a relative distance

Command value	Parameters	Data range	Unit	Default value	Description
Zd	n1	0~180000	um	0	Relative distance position
	[n2]	0~180000	um/s	50000	Operating speed

Return data: see Status Table

<Zg>n1,[n2],[n3] Move downward to pick up TIP

Control the Z-axis to move downward, stop when picked up the TIP.

Pick up TIP command

Command value	Parameters	Data range	Unit	Default value	Description
Zg	n1	0~180000	um	50000	Operating speed
	[n2]	0~100		80	Power percentage

Return data: see Status Table

<Zt> Stop Z-axis

Immediately stop Z-axis movement.

<Zc> Z-axis calibration

Calibrate the Z-axis.

Attention:

The Z-axis will move at full travel when this command is sent, make sure that there are no obstacles underneath the pipette carried by this Z-axis before sending this command.

Parameter Setting

Register

The registers are used for user configuration and viewing of ADP parameters, facilitating flexible application by users.

Register

Register Address	R/W	Data range	Unit	Default value	Description
81	R/W	100K/125K/250K /500K/1000K	Baud	500	CAN baud rate, requires a reboot to take effect.
82	R/W	0 ~ 1		0	Active reporting after motion completion 0: Do not report 1: Report
94	R/W	9600\19200\384 00\115200	Baud	38400	Serial Port Baud Rate
100	R			0	Z-axis current status (see status table)
101	R		um		Z-axis current position
107	R/W		ms	1000	Heartbeat interval time

110	R/W	0 ~ 1		0	Motor stall detection 0: Turn off motor blockage detection. 1: Turn on motor blockage detection.
120	R/W	0 ~ 255		1	Device Address
121	R				Software version
122	R				Device model, same for each ADP-Z
123	R				Device sequence number
124	R				Hardware version number
131	R/W	0 ~ 1		0	Self-locking mode 0: for open relay 1: for open holding current 2: for fully closed without holding force
134	R/W	1 ~ 5		1	Execute picking up TIP command, and continue to move down after picked up TIP: move down a distance after picking up TIP, which can increase the reliability of picking up TIP. Example: 1. Write 134 register as 1, then Z-axis moves downward 1/3mm distance after picking up TIP. 2. Write 134 register as 5, then the Z-axis moves down a distance of 5/3mm after picking up the TIP.

<Wr>n1,n2 Write register

Write register

Write register command

Command value	Parameters	Data range	Unit	Default value	Description
Wr	n1	81~135	N/A	N/A	Register address
	n2				Write data

<Rr>n1,[n2] Read register

Read registers, read the specified number of registers from the starting address, read registers larger than 1, return data separated by ‘,’.

Rread register command

Command value	Parameters	Data range	Unit	Default value	Description
Rr	n1	81~131	N/A	N/A	Start address
	n2				Number of registers

Return data: the status part is shown in the status table, the data area part is the data to be fetched, and the return data of multiple addresses are separated by ‘,’.

System operation commands

<?> Query Status

Query status via command ?

Return data: see status table

{}[n1] Loop control command

The loop control commands are used to control the cycle of the command string, the loop can be nested, a maximum of 20 cycles including nested cycles are supported in a command string.

Loop control command

Command value	Parameters	Data range	Unit	Default value	Description
{	N/A		N/A	0	Start loop
}	[n1]			0	End of cycle position No parameter or 0: infinite loop Other values: loop times

Note: Loops can be nested up to 20 levels

<L>[n1] Delay

The delay is used for internal system delay, mainly as a delay between two command executions.

Delay command

Command value	Parameters	Data range	Unit	Default value	Description
L	n1	0 ~ 2147483647	ms	N/A	Delay time

Return data: see status table

<U>n1 Rest command

This command is used to reset the device.

Rest command

Command value	Parameters	Data range	Unit	Default value	Description
U	n1			123456	The parameter must be 123456 for the successful reset

<M>n1 Restore factory settings command

After executing the commands, the device needs to be restarted to take effect.

Restore factory settings command

Command value	Parameters	Data range	Unit	Default value	Description
M	n1			123456	The parameter must be 123456 for the successful restoration to the factory settings.

<S> Power down to save setting parameters

After executing the commands, the modified register parameters will be saved when power down.

10. KT_CAN_DIC Object Dictionary

The object dictionary consists of a 16-bit main index and an 8-bit sub-index, which indicates different operations and parameters to read and write by means of ordered index numbers, all data including read and write permissions.Control instructions including zero, movement to absolute position, take TIP instructions, etc., are in accordance with different main indexes indicate different instructions, sub-indexes for different parameters, the number of parameters and the number of parameters of the serial port instruction is the same, for example, the serial port instruction Zz [n1], such as the implementation of the instruction before the need to modify the default parameters.

10.1 Z-axis Control Commands

Each control command has a unique main index and multiple sub-indexes. Frame data with non-zero sub-indexes are sent first, and frame data with zero sub-indexes are sent last, and the Z-axis starts to move when it receives a command with zero sub-indexes.

KT_CAN_DIC Control Commands

Function	Main index	Sub-index	R/W permissions	Data range	Default value	Description
Initialization	0x4100	0	W	1 ~ 180000 (um/s)	Required	Initialization, speed during parameter initialization process
运动到指定位置	0x4101	0	W	0 ~ 180000(um)	Required	Counting from the top to the bottom after initialization
		1	RW	0 ~ 180000 (um/s)	50000	Move to appointed position speed
相对位置向上运动	0x4102	0	W	1 ~ 180000 (um)	Required	Move up from current position
		1	RW	0 ~ 180000 (um/s)	50000	Upward motion speed
相对位置向下运动	0x4103	0	W	1 ~ 180000 (um)	Required	Move down from current position
		1	RW	0 ~ 180000 (um/s)	50000	Downward motion speed
向下运动获取TIP	0x4104	0	W	1 ~ 180000 (um/s)	Required	Downward motion speed
		1	RW	0 ~ 100	80	Power percentage
急停	0x4108	0	W	0	0	Stop movement

校准	0x9000	0	W	0	0	zero-calibration and precision
掉电保存	0x9F10	0	W	0	0	Power down to save data
恢复出厂设置	0x9F10	1	W	123456	0	Restore factory settings
运动完成主动上报	0x9F00	5	RW	0 ~ 1	0	Active reporting after completing movement
复位重启	0x9F00	3	W	123456	0	Reset and restart

10.2 General Instructions

KT_CAN_DIC Control Commands

Function	Main index	Subindex	Read and write permissions	Data range	Default value	Function description
Query status	0x2000	1	RW			node status, 0: Idle, 1: Busy. Other values: different error codes depending on the equipment, see the specifications of different equipment for details
Device model	0x9F00	0	R			Unique number for each type of equipment
急停	0x9F00	1	W			Stop this axis movement
Heartbeat interval time	0x9F00	2	RW			CAN active upload heartbeat interval ms 0: No heartbeat data Other values: Timed Upload Interval in ms
Reopen node	0x9F00	3	W			Reopen
software version	0x9F00	4	R			software version
active	0x9F00	5	RW			0: Close active reporting

reporting					1: Turn on the active reporting
					Motion completion CAN active upload, via KT_CAN_DIC command 03, main index 0x7002 sub-index 0 uploads data 0, indicating motion completion.Other values are error states with the same error code as the node error code
Power down to save data	0x9F10	0	W	123456	Fill in 123456 in the data area for Power down to save data
Restore Factory Settings	0x9F10	1	W	123456	Fill in 123456 in the data area for Restoring Factory Settings

10.3 Register Read and Write

It is specified that the main index 0x2000 is the Z-axis internal register, the sub-index register address, the register address is shown in Register

10.4 Process Data

The Z-axis automatically uploads process data via commands 0x03, and the process data dictionary is as below KT_CAN_DIC

KT_CAN_DIC Process Data

Function	Main index	Subindex	Read and write permissions	Data range	Default value	Function description
change of state	0x7002	0	/	0 ~ 255	0	Open register 82. the movement completes the active uploaded data 0: normal Other data:Error status, see status table

10.5 Heart Rate Data

Z-axis sends heartbeat data via command 0x04, the master device can be used to detect whether the device is online

10.6 Alarm Data

Z-axis sends alarm messages via command 0x08. The alarm messages are shown in the status table.

10.7 LED Indicator

The current status of the Z-axis is indicated by LED flashes at different frequencies, as shown in LED Status Description below:

LED Status Description

LED Status	Description
No flashing	idle
Blue light blinks once and stops for 1s	Motor blocking error or Z-axis in crypto mode
Blue light blinks twice and stops for 1s	Driver Error
Blue light blinks 3 times and stops for 1s	Optocoupler error
Blue light blinks 4 times and stops for 1s	Storage error
Blue light blinks 5 times and stops for 1s	Uncalibrated
Blue light always on	Z-axis in motion or Z-axis in download mode

11. Common Faults and Q&A

11.1 Common Faults and Troubleshooting Methods

Note: The following descriptions are troubleshooting methods for faults arisen when performed in a compliant operating environment and under rated operating conditions.Here are four types of common faults:

Communication related (Communication Faults)

Communication Faults

Faults	Cause	Suggested Troubleshooting Method
Z-axis power on but the indicator doesn't flash	The cable plug part is not fixed. Z-axis moves up and down causing the cable to bend and break at the terminal connection.	Consider replacing the cable if poor contact occurs when wiggling the cable, and ensure that the cable cover is properly pressed against the cable sheath. For more details, refer to section 4.1, "Z-axis Installation."
	short circuit of power cables	Use a multimeter to check if the Z-axis power supply is short-circuited. If a short circuit is detected, please return the device to Keyto for repair.
	Loose cable joint	Power off then reconnect. For more details, refer to section 4.1, "Z-axis Installation"
	Incorrect cable connection	Connect the cables correctly as described in 3.2 "Lead Definition"
Failed to communicate with ADP	Incorrect baud rate	The baud rate defaults to 38400, please test the communications in accordance with Chapter 5, "Composite Function Controller Tool Usage Instructions."
	Incorrect wiring sequence	Check RS232 cables: Rx→Tx, Tx→Rx, GND→GND.
	Incompatible serial port tool	Replace the USB to serial cable or RS485 adapter with a different brand.
	Incorrect instruction format	Check the instruction format in accordance with Chapter 8 "Communication Process ".
Unstable communication	Noise on communication lines	1.Make sure the communication cable is as short as possible. 2.Dial terminating resistor to ON if necessary. 3.Separate the communication cable from the high-current cable. 4.Keep the resistance between the two phases of CAN and RS485 at 60Ω.

	Unstable communication mode	1.Communicate at a lower baud rate. 2.Adopt the question-and-answer communication mode, make sure transmission interval is≥10ms. 3.Use of retransmission mechanisms to ensure communication reliability.

11.2 Q & A

Q1: Why did the ADP work yesterday and suddenly stop working today? /Why the communication doesn't work after being connected?

1. Check the wire connection, supply voltage or power.
2. Measure the resistance between the 24V(read) and GND(black), if it is short-circuited, it means the board is damaged, please contact Keyto for further maintenance.
3. Check the wire sequencing and configuration for serial port or CAN.
4. Refer to Chapter 5, using correct address(default address 1)to send command.
5. Please keep the resistance between two phrase of CAN and RS485 as 60Ω, minimizing the transmitting distance.
6. Ensure proper RS232 wiring as Rx→Tx, Tx→Rx, GND→GND

Q2: How can I use the ADP with the Z-axis?/How does liquid level detection work?

1. When using our Composite Function Controller Tool to control both the Pipette and Z-axis, the tool will automatically detect if the Z-axis is connected after device scanning.
2. Enter the required parameters step by step. Send "pipette initialization, Z-axis initialization, Z-axis descent, Z-axis descent", after 500ms, then click to execute the liquid level detection command.
3. There are currently no command for Z-axis to follow descent when liquid detecting and aspirating in the Controller tool.
4. The Z-axis will automatically stop once the Pipette detects the liquid surface.

Q3: Is there a DEMO database?

1. Currently available STM32 microcontroller, C / C # upper computer reference source code

Q4: Which communication is recommended?

1. Suggest to use CAN communication. After setting the active report mode at motion completion, motion completion and abnormal status will be reported actively.
2. If the serial port is used, it is recommended to use OEM protocol. When the communication receives the response, please send the next frame at least 10ms interval.

Q5: How can I make sure each command is successfully received?

1. The CAN command sequence number response and the send command should correspond to each other.

2. Wait for a response to each command before sending the next frame. If a timeout occurs without a response, re-send the command.

Q6: How to determine motion completion? /Can the flag bit be reported when the movement is complete?

1. The query status is "busy" when the device is in motion, and "idle" when the motion is complete.
2. The CAN heartbeat status changes from 1 (busy) to 0 (idle).
3. Refer to Register 82 for report after motion completion.

Q7: Why do the parameters remain the same after restarting the controller tool after parameter setting?

1. Refer to 5.6 Register Parameter Setting Instructions for the steps of setting parameters (please do not modify the system parameters arbitrarily)
2. If you use a switch power supply, make sure the power is completely off before restarting.
3. Part of the parameters do not support the power-down save.

Q8: What should I do when an error is reported?

Record the status of the device feedback and the current execution process when an error is reported. If the following solution measures do not provide effective help, please contact us. Below are some of the status trigger scenarios and their corresponding solutions:

1. Error status 10-16 (DEC the same below): Check the text-transform parameters and the writable range of the parameters.
2. Error status 18: Requires device initialization to control actions;
3. Error status 80: Requires device initialization and troubleshoot;
4. Error status 81: Requires device initialization and troubleshoot;
5. Error status 82: Requires device initialization and troubleshoot;
6. Error status 83: Requires device initialization and troubleshoot;
7. Error status 84: Requires device initialization then calibration.

Q9: What should I do if the Z-axis zero position is not the same when using multiple channels?

1.You need to contact the manufacturer to solve the problem.

Q10: What should I do if the Z-axis slips after the motion is completed?

1. Change the Z-axis holding mode to motor holding force self-locking mode (register 131).For details, see Table 9-7 Register Table.

12. Environmental Conditions

Item	Unit	Value
Operating environmental temperature	Degrees Celsius	+5°C ~ +40°C
Operating environmental humidity	RH%	30% ~ 80%无冷凝
Storage temperature	Degrees Celsius	-20°C ~ +60°C
Storage humidity	RH%	10% ~ 85%

13. Safety precautions

For the personal safety of you and other users and to prevent damage caused by improper operation, please read the safety precautions carefully.
This manual uses the following symbols. Please fully understand what they represent before continuing.

Warning	Where the content with the mark is related to the safe use of the product and the personal safety of the user, it must be operated in strict accordance with the requirements, otherwise it may cause damage to the product or endanger the personal safety of the user.
Caution	The content with the mark is the part that users must pay attention to, otherwise it will cause product damage or other losses due to improper operation.

Caution
Confirm the specification: Please fully consider the use, fluid, environment and other conditions of use, and use within the scope of the specification to avoid damage to the product;
Material selection: For the media tolerance is not clear, first do the corresponding experiment to confirm the material, and then confirm whether the selected model is suitable;
The electromagnetic coil of this product has no waterproof function. If water or liquid drops accidentally splash on the solenoid valve during use, please wipe it as soon as possible to avoid water penetrating into the coil and causing short circuit. If you need to contact with water frequently, please configure waterproof protective devices;
Special fluorinated rubber and perfluorinated rubber are greatly affected by temperature, please avoid high frequency use at low temperatures;

Warning
1. Installation position: Do not put the inlet and outlet of the valve towards the position where the human body may contact, to avoid high pressure air impact on the human body after the solenoid valve is accidentally loosened;
2. Installation operation: The rotation of the coil assembly of this product will cause poor product performance. Do not use hands or tools to twist or rotate the proportional valve coil when installing this product;
3. Installation environment: Avoid using the solenoid valve near the heat source with high temperature;
4. Maintenance and repair: For abnormal occurrence, please contact the manufacturer first, do not directly disassemble and assemble, to avoid irreparable damage, and avoid failure to confirm the cause of the abnormal.

Keyto 垦拓

Contact Us At

Tel: 0755-29516669

E-mail: info@keyto.com

Web: en.keyto.com

Add: 2 Yuandong East Road, Fuhai Street, Baoan District, Shenzhen

LinkedIn

YouTube

VERSION NUMBER 1.0