Hiwonder

LX-15D Bus Servo User Manual

V1.0

1. Product Introduction

1.1 Instruction

LX-15D serial bus servo incorporates a servo driver, motor, and bus servo signal into one unit. It is controlled by serial commands with a baud rate of 115200. Users can send corresponding commands to the servo based on the communication protocol provided by our company to control the servo's rotation, read servo information, or switch to stepper motor mode.

The interface is a half-duplex UART asynchronous serial interface. Each servo has three interfaces, allowing them to be daisy-chained. Theoretically, up to 253 bus servos can be connected in series. This servo is widely used in the joint design of various bionic robots.

1.2 Working Principle

A servo typically consists of a housing, a set of reduction gears, a motor, a potentiometer, and a control circuit.

The communication method of the asynchronous serial bus (data lines in series, parallel in the circuit) requires the master controller to send corresponding instructions to the servo to achieve control. In serial communication, each transmission needs to be packaged and parsed according to a specific format.

The ID number is used to distinguish different devices. The data length indicates the length of the data to be sent, where the data length plus 3 equals the length of the entire instruction packet, from the frame header to the checksum. The instruction field is used for various servo commands, such as position and speed control. The checksum is used to verify data integrity.

When the bus servo engages in serial communication, it first sends a request to the master control board along with the corresponding parameters (such as position). The command packet includes the target servo's ID, and only the servo with the matching ID will execute the command, encapsulating the execution result and related status information in feedback information returned to the master control board. The master control board parses the command and performs the corresponding action.

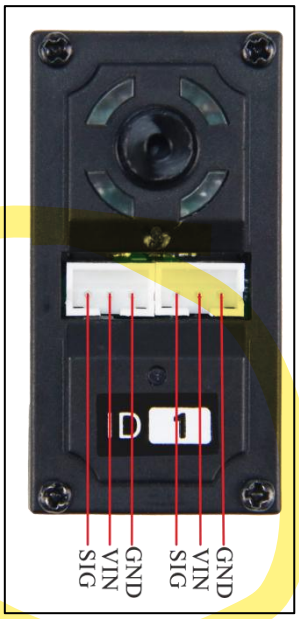
This characteristic of bus servos enables the coordinated control and data exchange of multiple servos, greatly enhancing the system's scalability and flexibility, playing a crucial role in the components of robots.

Regarding the bus servo communication protocol, please refer to “**Hiwonder Bus Servo Communication Protocol**”.

1.3 Servo Horn Installation & Pin Instruction

Align the servo horn with the red “+” to install, please refer to the following picture:

The port distribution and instruction refer to the following picture and table.



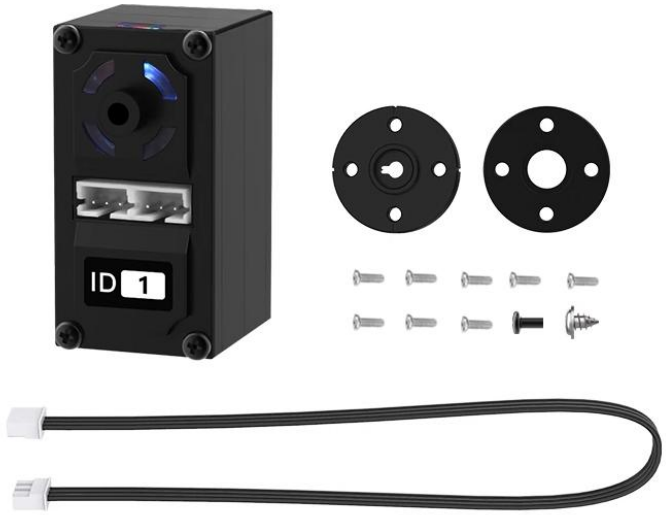
Pin (from left to right)	Instruction
GND	Power ground
VIN	Power input
SIG	Signal terminal, (half-duplex

	UART asynchronous serial interface)

Note: When the servo and the microcontroller are not using the same power supply, ensure that the two power supplies share a common ground.

1.4 Packing List

The servo package will include accessories such as plastic servo horns, the screw bag, and a 20cm servo wire, as shown below:



2. Parameter

Name	LX-15D bus servo	**Control method**	UART serial command
Brand	Hiwonder	**Communication baud rate**	115200
Weight	43g	**Memory**	Servo settings are automatically saved when power off

Size	44.02 x 22.92 x 35.12mm	**Servo ID**	0-253 for user setting, ID 1 is default
Working voltage	Rated at 7.4V, ranging 6-8.4V	**Readback function**	Support angle readback
Rotation range	0.22sec/60° 7.4V	**Protection**	Avoid stalling and overheat
Stall torque	15kg.cm 7.4V	**Parameter feedback**	Temperature, voltage and position
Rotation range	0-1000 corresponds to 0°-240°	**Working mode**	Servo mode and gear motor mode
No-load current	100mA	**Gear type**	Alloy Gear
Stall current	2.4-3A	**Servo wire**	20cm, other lengths can be selected
Servo accuracy	0.3°	**Connector model**	PH2.0-3P
Control angle range	0-1000 corresponds to 0-240°	**Application**	All kinds of bionic robot joints

3. Notice

Please carefully read the following precautions before operations:

① The operating voltage of this bus servo is 6-8.4V. Please select a stable power supply within this range. Do not use over-voltage as it will damage the servo. Conversely, insufficient voltage will prevent the servo from operating.

② The default ID of the bus servo is 1 by default. Before use, preset the servo ID according to your requirements.

③ Before setting the ID number for servo, connect each servo individually; otherwise, all servos will be set to the same ID. Connect servos in series after setting the IDs.

④ The bus servo is a high-precision accessory. After powering on, do not manually force the servo arm, servo horn, etc., to avoid internal damage to the servo.

⑤ The bus servo will heat up after prolonged operation. Allow the servo to cool down after running for a while to prevent overheating, which can affect performance and service life.

⑥ During continuous rotation, the load on the bus servo should not exceed its stall torque. It is recommended that the load be 1/3 to 1/5 of the stall torque.

Stall torque refers to the maximum torque the servo can provide when it is unable to rotate (i.e., stalled or blocked). Due to energy loss during servo operation, control the load appropriately to avoid exceeding the servo's rated load. Excessive load will overwork the servo, increasing energy consumption and reducing efficiency.

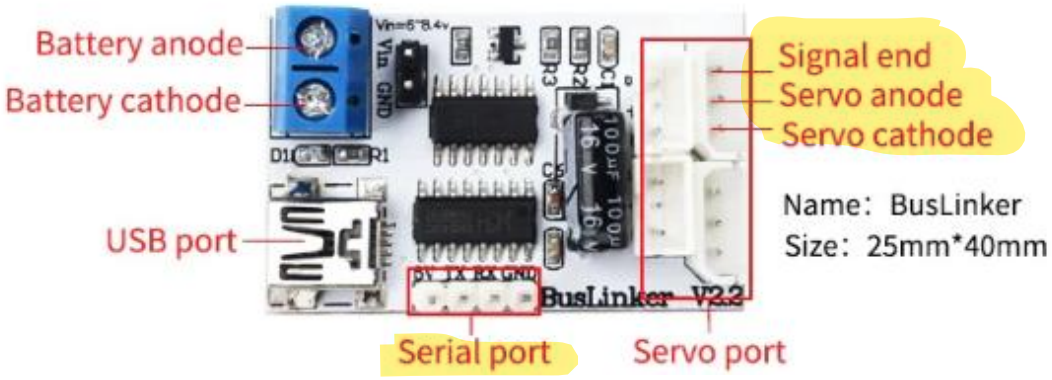
⑦ The communication lines of the bus servo are susceptible to electromagnetic interference. To ensure stable and reliable communication, avoid placing the bus servo near high-power power lines, motor lines, or other potential sources of electromagnetic interference.

⑧ Set the angle control range and speed of the bus servo reasonably based on the actual project requirements to ensure safe operation. Avoid exceeding mechanical limits or moving too quickly to prevent accidental collisions or damage.

4. Test

4.1 Debugging Software Introduction

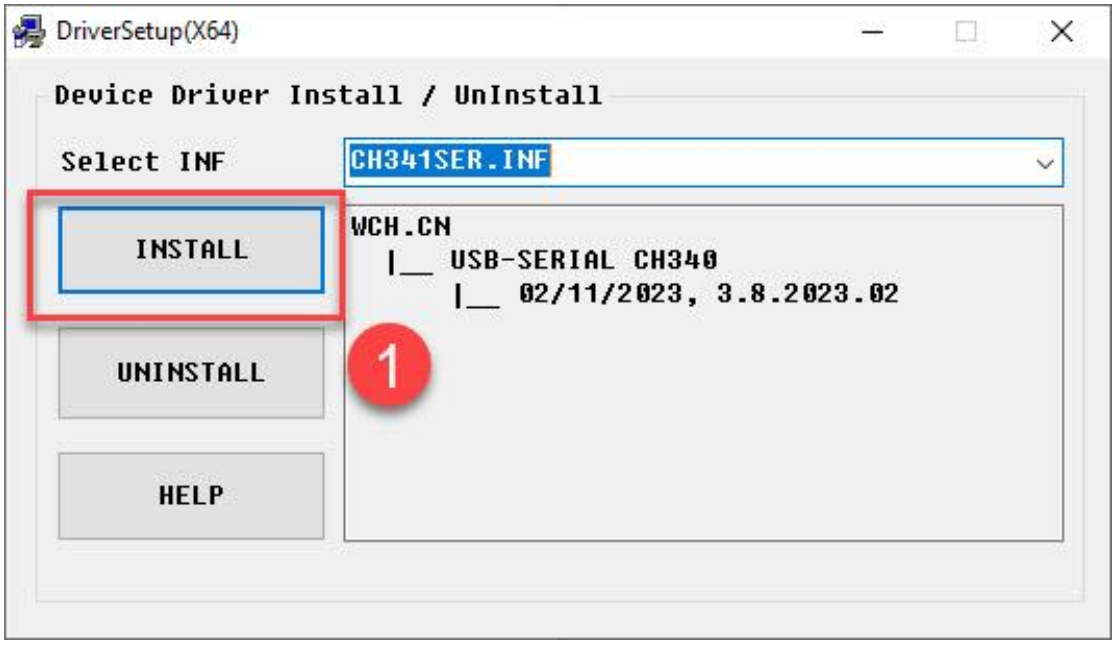
TTL/USB debug board is a tool for debugging servos. With the provided Bus Servo Terminal software, it can test servos and set servo parameters. Additionally, the debug board can communicate via the serial port with a microcontroller to control the servos.

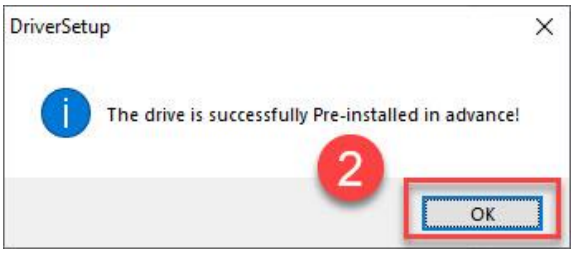


4.1.1 Install Software for Debug Board

(1) Locate the “ch341ser.exe” driving package in “**Servo Parameter Setting Software(Debug Board)/ BusLinker Debug Board Driver**”

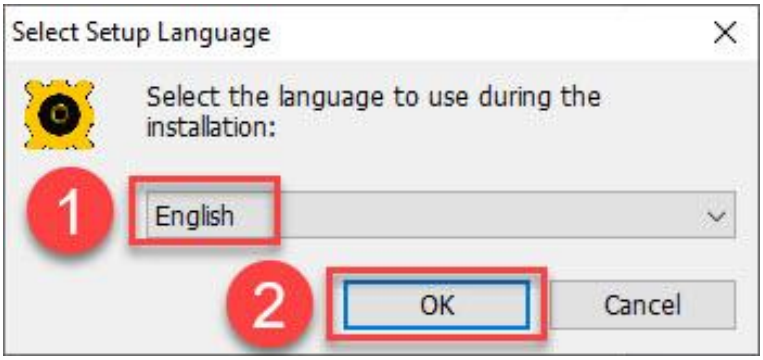
(2) Install the software according to the installation prompts.

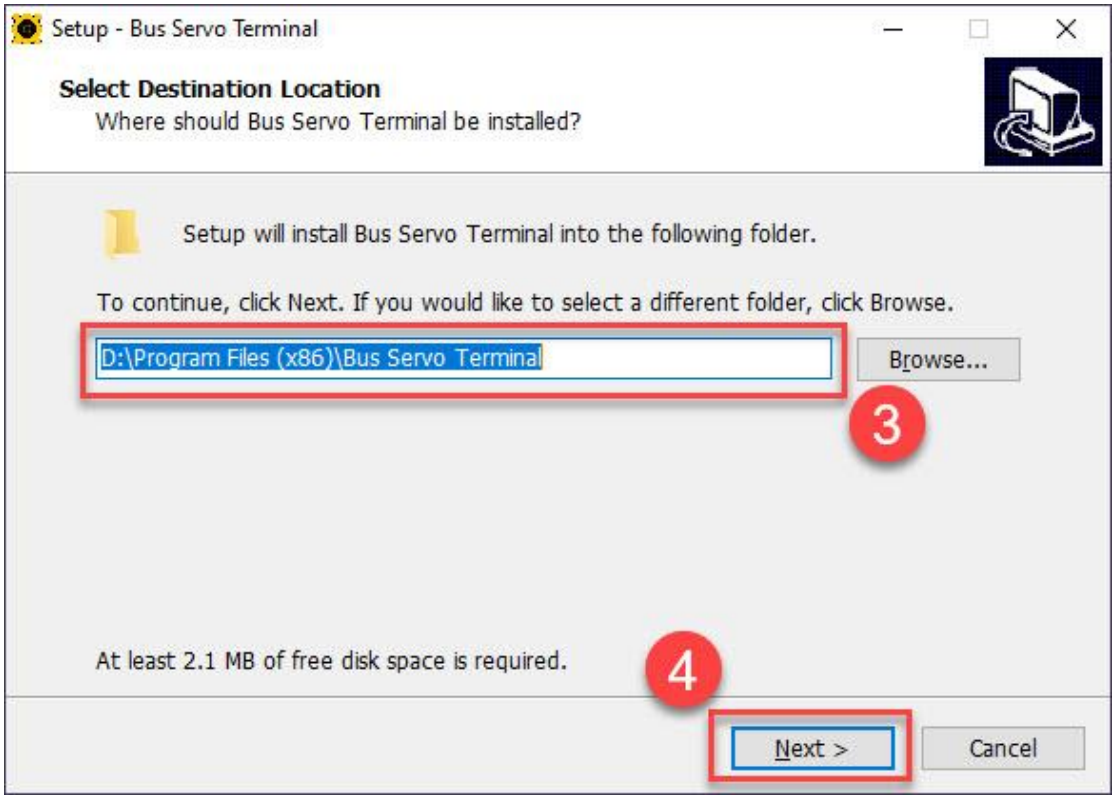


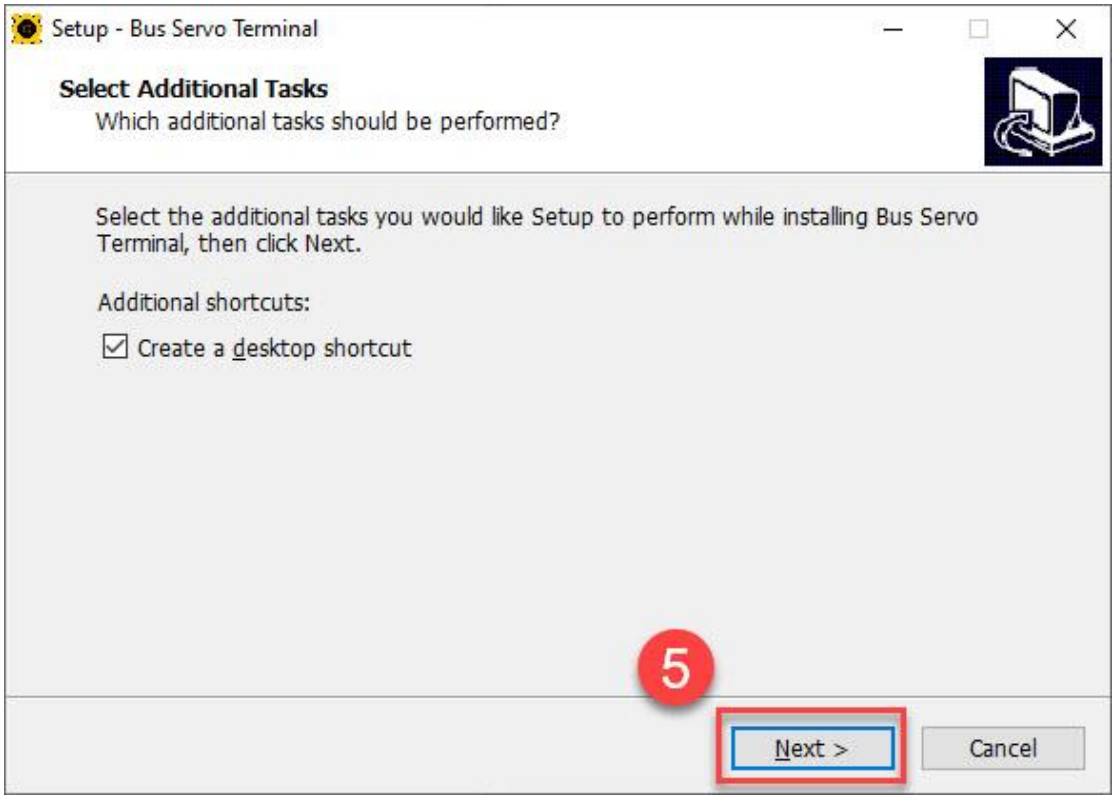


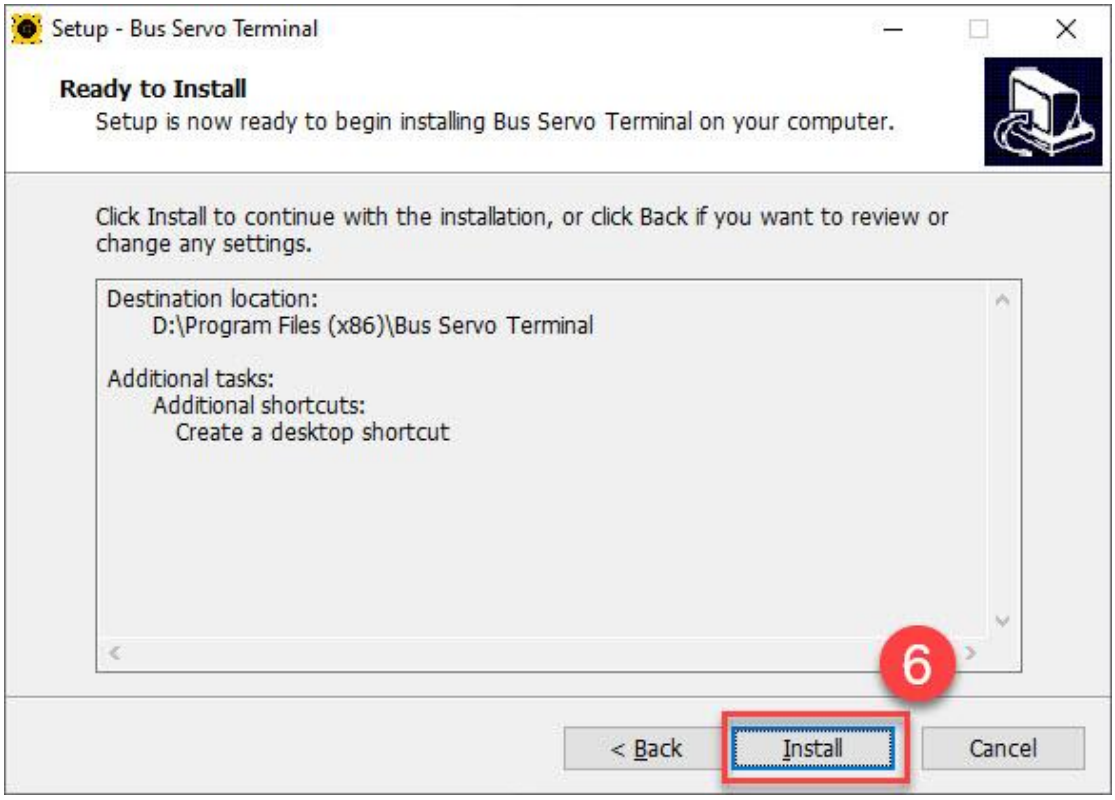
(3) After the driving software installation is installed, locate the "**Bus Servo Terminal setup V2.3.exe**" in "**Servo Parameter Setting Software (Debug Board)/ BusLinker Debug Board Software**"

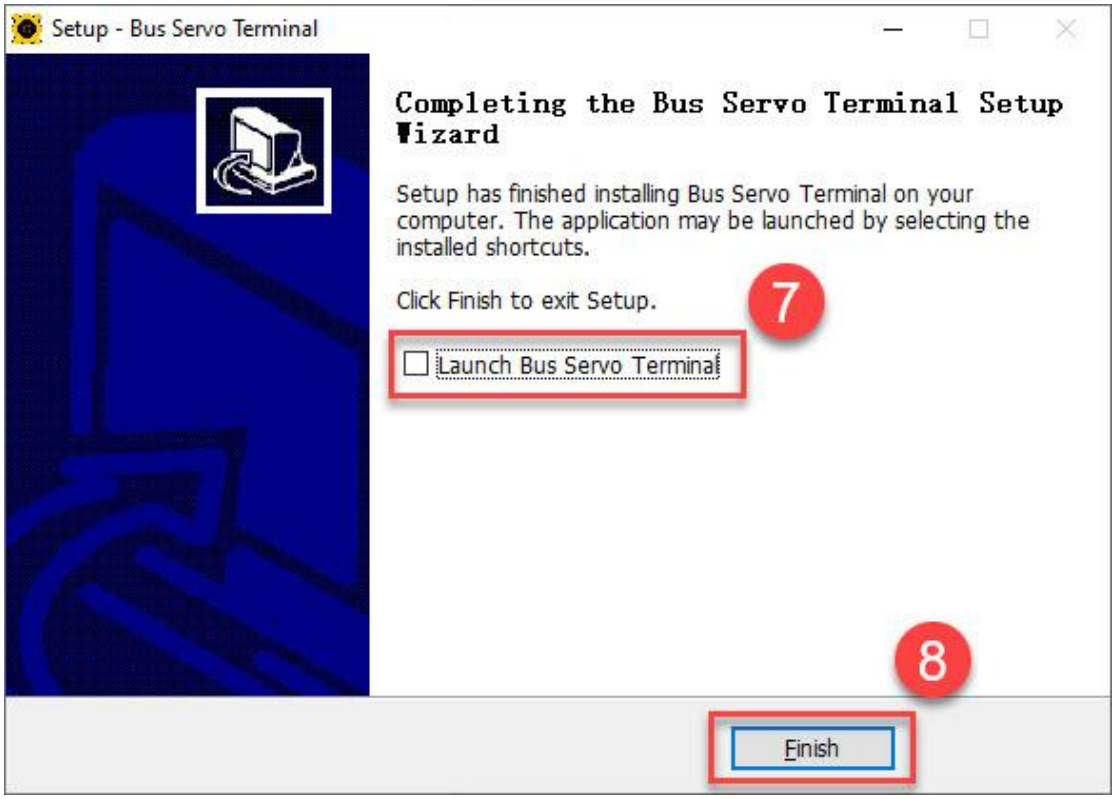
(4) Install the software according to the installation prompts.




Setup - Bus Servo Terminal
Select Destination Location
Where should Bus Servo Terminal be installed?
Setup will install Bus Servo Terminal into the following folder.
To continue, click Next. If you would like to select a different folder, click Browse.
D:\Program Files (x86)\Bus Servo Terminal
Browse...
3
At least 2.1 MB of free disk space is required.
4
Next >
Cancel

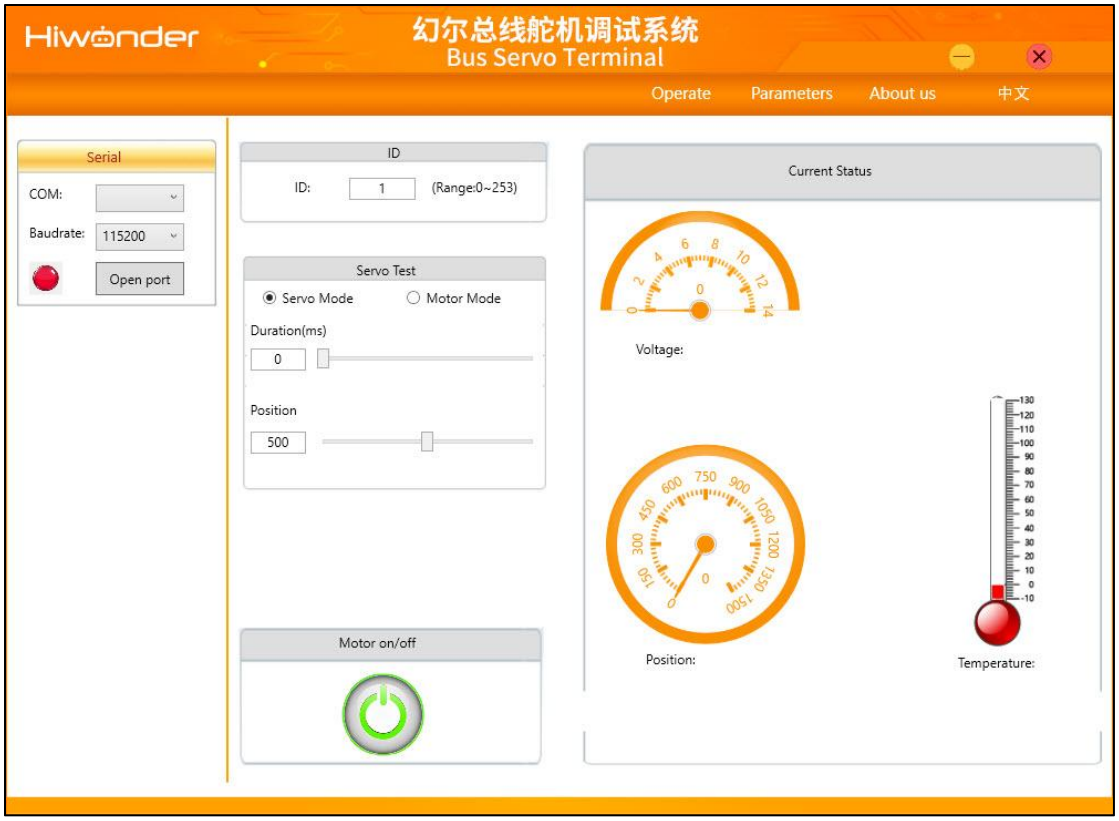

Setup - Bus Servo Terminal
Select Additional Tasks
Which additional tasks should be performed?
Select the additional tasks you would like Setup to perform while installing Bus Servo Terminal, then click Next.
Additional shortcuts:
Create a desktop shortcut
5
Next >
Cancel





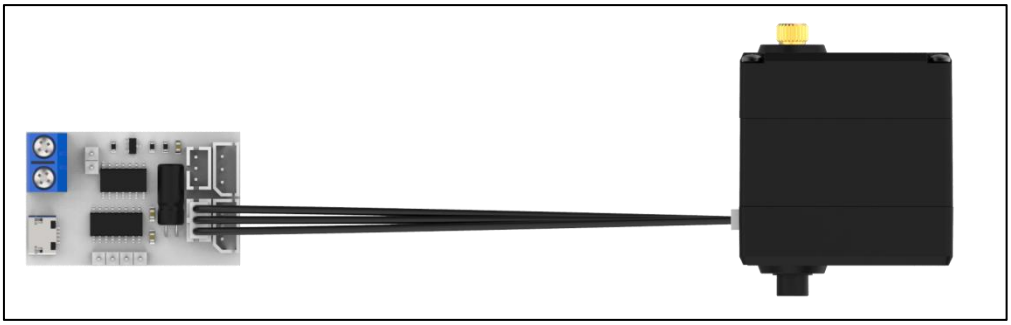
(5) After opening the software, the interface layout is as follow. The

specific operations can refer to “**Servo Debug Board Software Instruction**”.

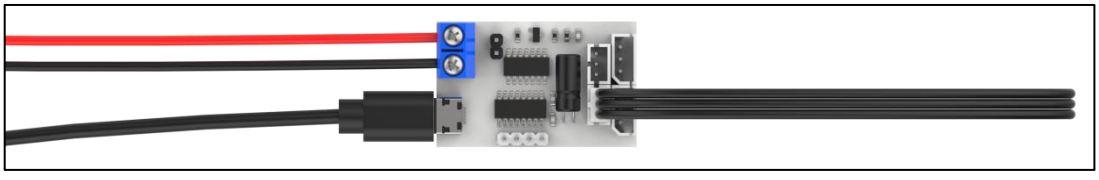


4.1.2 Hardware Connection

(1) Connect the servo to any bus servo port of the debug board.

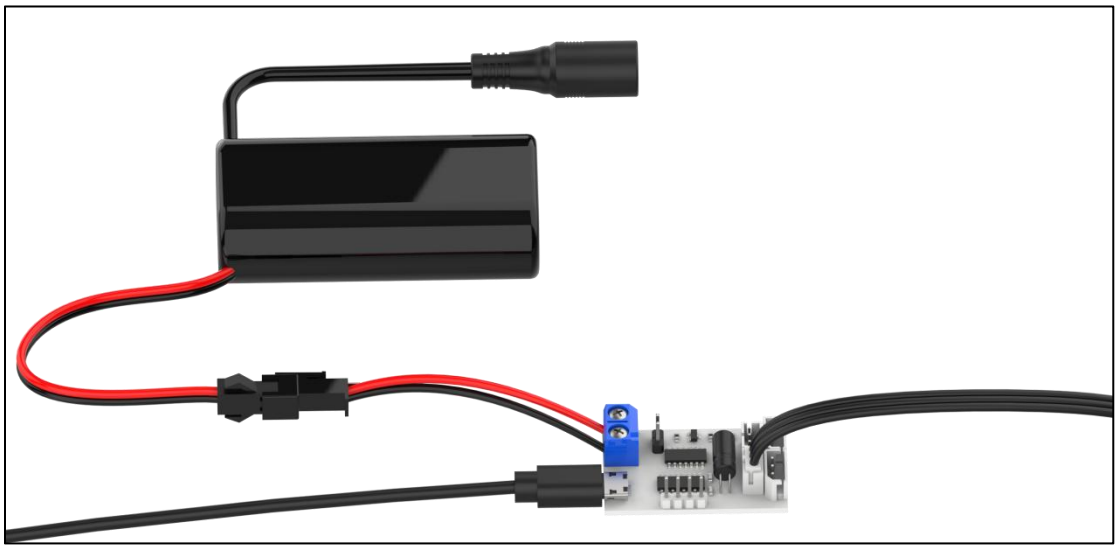


(2) Connect the red wire to the “+”, the black wire to “-”, then connect USB cable.



(3) Connect the lithium battery to the power matching cable. (The power matching cable adopts anti-reverse plug, do not force the cable)

If you do not have a lithium battery, please prepare yourself a 12V power source.



4.2 Test Instruction

4.2.1 Introduction to Servo State

The three images represent the three basic positions of the servo (central position, maximum angle, and minimum angle). For demonstration, a servo arm is inserted as a reference (with the label on the right side of the servo arm as the initial position).

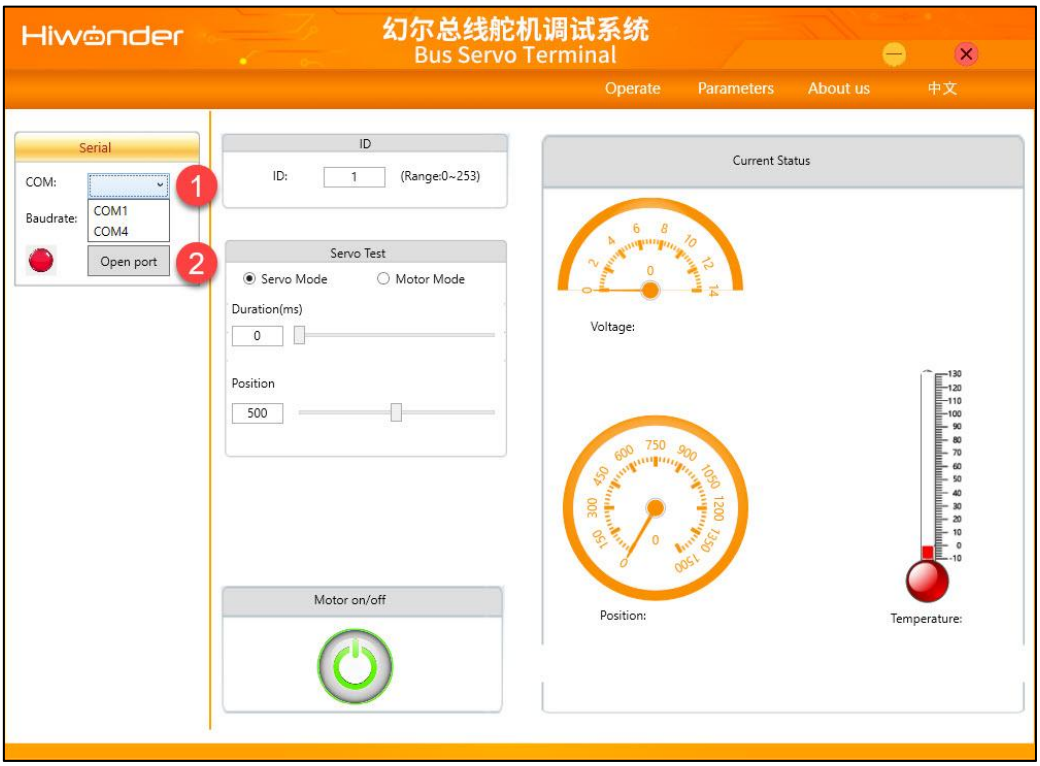
Central position: 120°servo angle (corresponding to the position of 500)

Maximum position:240° servo angle (corresponding to the position of 1000)

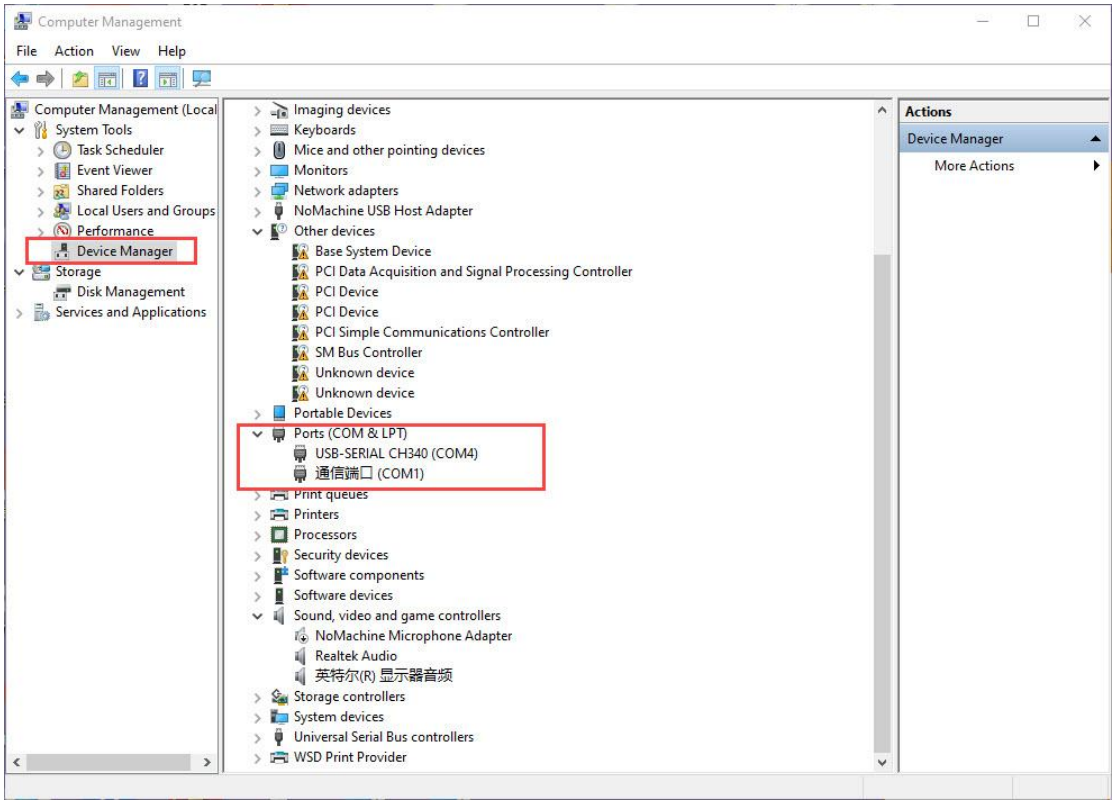
Minimum position: 0° servo angle (corresponding to the position of 0)

4.2.3 Retrieve Servo Information

(1) Open Bus Servo Terminal software, select the serial number on the left, then click “Open Serial Port”. The baud rate defaults to “115200”.



Note: If the serial port cannot be found, go to the Control Panel, locate Device Manager, and check if the computer has recognized the CH340 port.



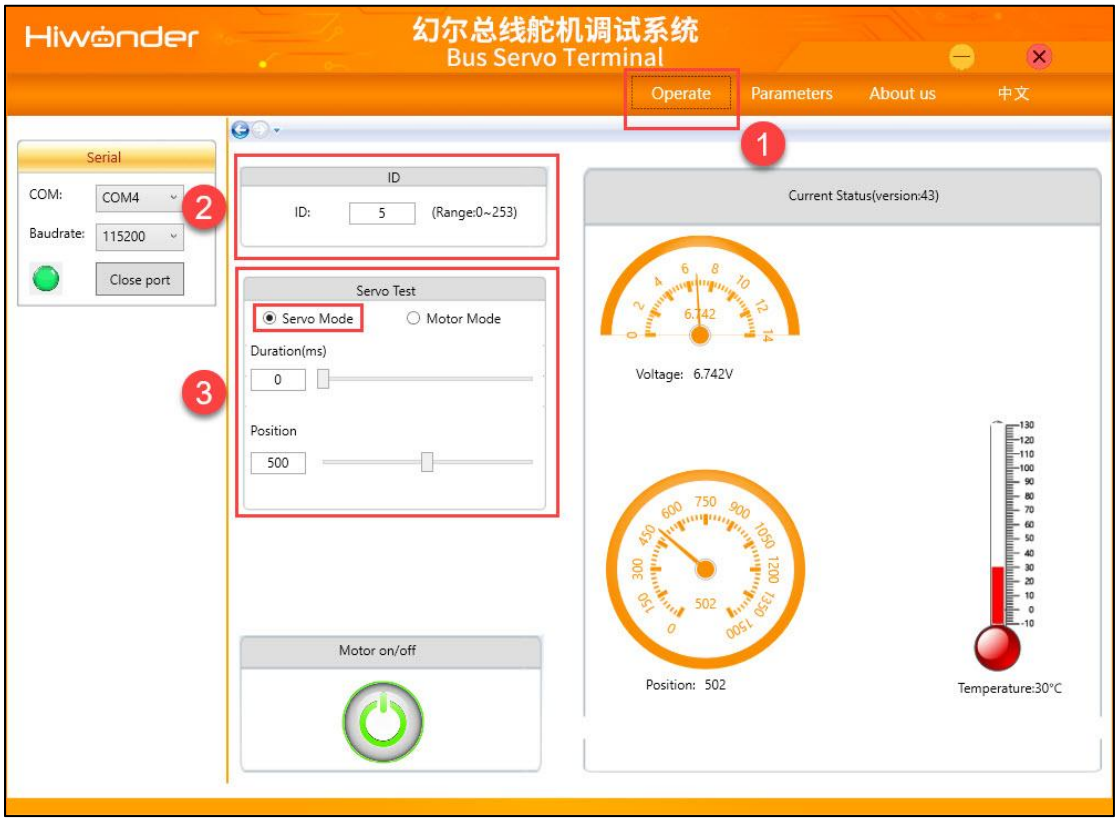
(2) Click “**Parameters**”, then click “**Read**” and wait for the completion prompt as follow：

From the above image, the information about the connected servo can be viewed, including ID, deviation value, position range, etc.

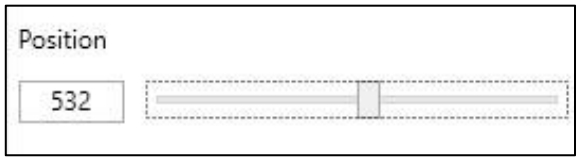
4.2.4 Test Servo Rotation

Once the position range is read, you can perform simple debugging on the servo according to the following operation steps.

(1) Click “**Basic Operation**”, and enter the servo ID number to be controlled. Here uses an example of “5” ID number. Then select “**Servo mode**”.



Control the servo rotation by dragging the slider in the "Position" column. (by clicking and dragging with the mouse or using “←→” keys).



At the central position, which corresponds to 500, the corresponding angle is 120°. At the minimum position (0), the corresponding angle is 0°, and at the maximum position (1000), the corresponding angle is 240°.

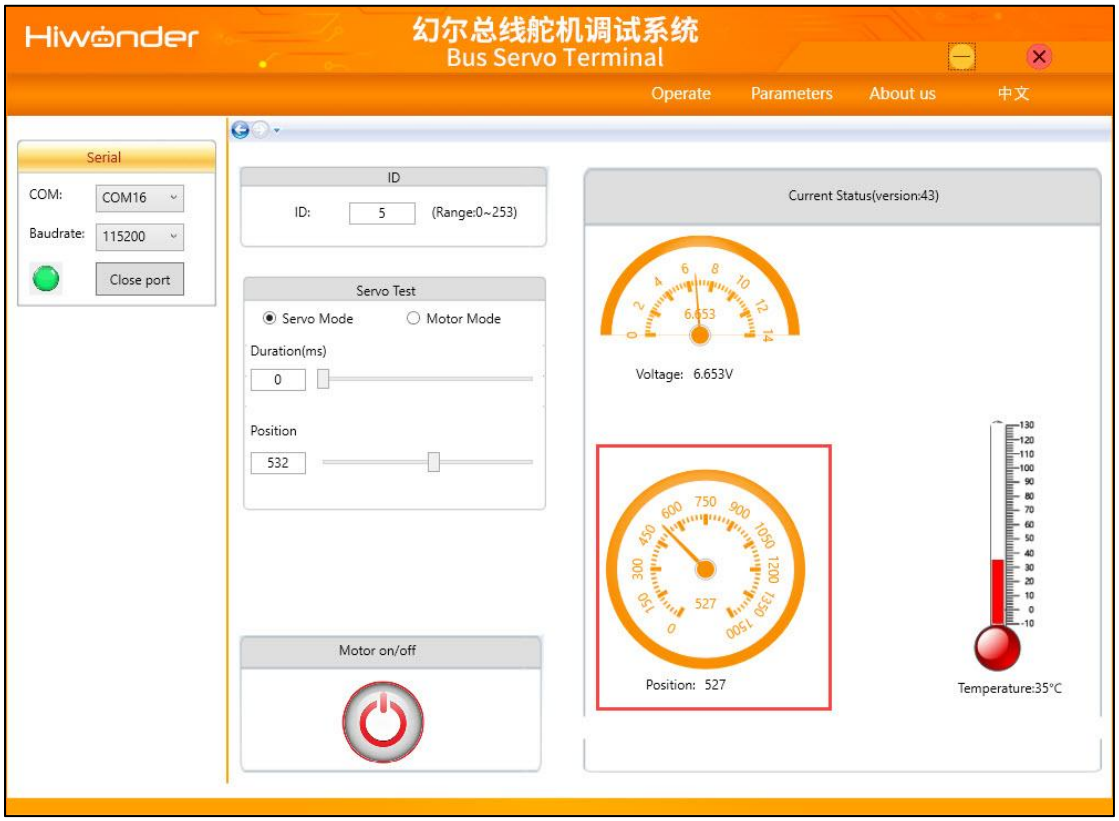
Given that the servo rotates 240° after power-up, the calculation method for servo angle and position is: 240/1000 = 0.24.

For example, if you need the servo to rotate to 90°, the corresponding position calculation would be: 90/0.24 = 375.

Note: If the servo has limit settings enabled, dragging the slider beyond the limits will result in the servo being unable to rotate. For detailed information, please refer to "**5.2 Set Servo Limit**".

(2) In "**Servo**" mode, click "Power ON/Off" to control the servo power state.

After power-off, you can manually rotate the servo arm on servo. Simultaneously, the real-time servo position information can be monitored at right side of the interface.



(3) When clicking "**Motor mode**", the servo serves as DC geared motor and its speed can be adjusted by dragging the slider or inputting data.

(4) The speed ranges from -1000 to 1000. When the value is positive, the servo rotates counterclockwise; when the value is negative, the servo rotate clockwise. The larger the value, the faster the rotation.

5. Servo Setting Method

5.1 Set Servo ID

Note：Due to the the characteristics of the bus connection, when setting the ID, please ensure that only one servo is connected to the debugging board at a time. If multiple servos are connected, all the servos will be set to the same ID!

(1) Reference to "**4.2.1 Hardware Connection**" to connect the hardware. Open "Bus Servo Terminal" and connect the serial port.

(2) Click "**Parameter settings**" and then click "**Read**". You will see that the current servo ID is 0.

(3) Enter the number directly in the ID setting box, with ID range being 0 to

253. Here, let's use ID 5 as an example to demonstrate:

(4) Click "**Apply**" and wait for the progress bar at the bottom to complete.

(5) After the settings are complete, open the software and click "**Read**" to

check the settings result.

5.2 Set Servo Limit

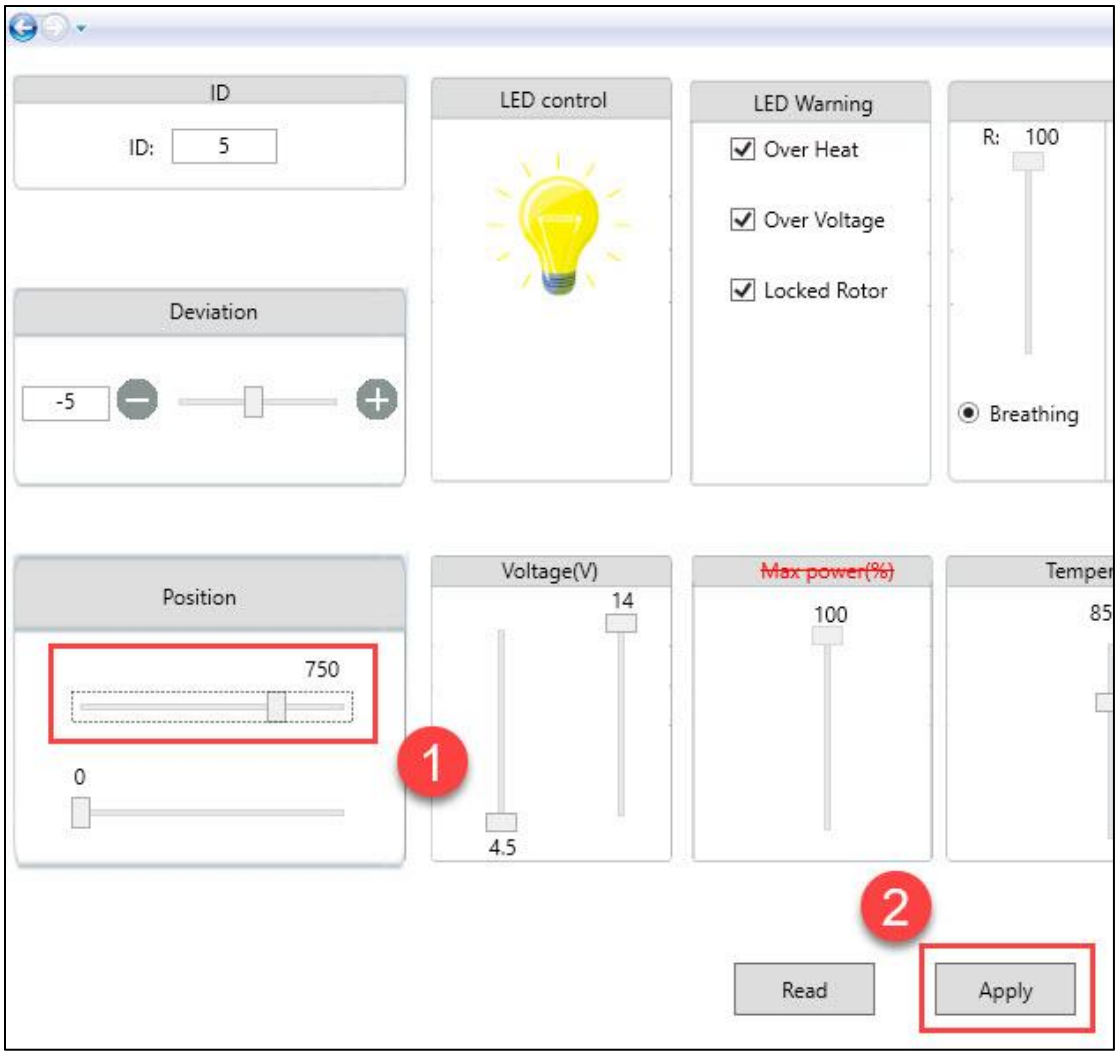
Setting the servo limits serves two purposes. Firstly, it prevents the servo from moving beyond its mechanical limits, thus avoiding potential damage. Secondly, when the servo is installed in a robot, setting limits helps prevent mechanical components from interfering with each other, which could cause the robot to get stuck and lead to servo stalling. When a servo stalls, the current can increase significantly, up to 3A, leading to elevated servo temperature and a risk of damage.

(1) Open "**Bus Servo Terminal**", and locate "**Position range adjustment**" in "**Parameters**".

(2) Adjust the position by dragging the slider.

For this servo, the rotation angle after power-on is 240° . So, the calculation method for servo angle and position is: 240/1000 = 0.24.

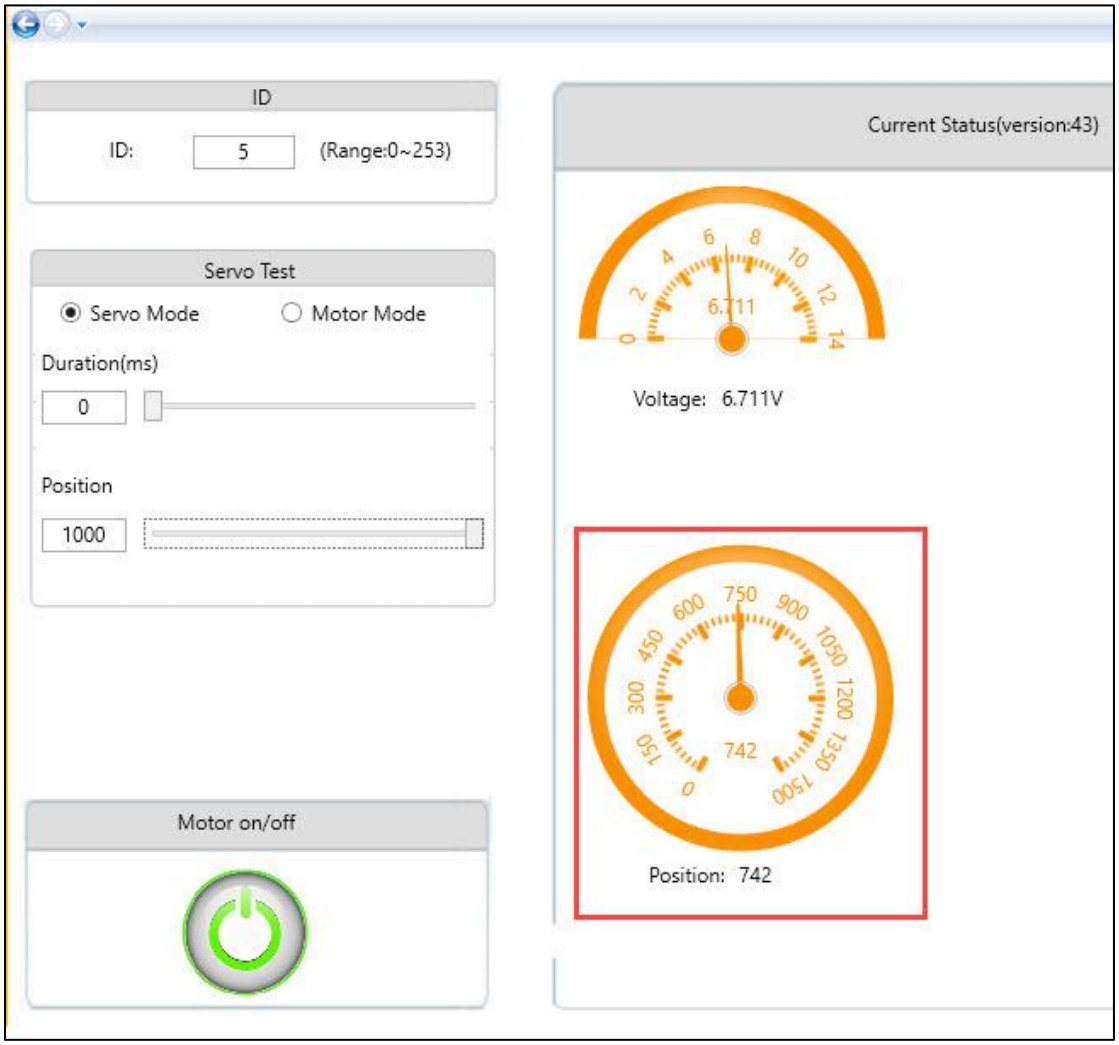
For example: if you need to limit the maximum rotation angle of the servo to 180° , the corresponding position calculation is: 180/0.24 = 750. Click on the slider for adjusting the position range, as shown in the figure below, and adjust it to 750 using the mouse or the "←→" arrows on keyboard.



After setting is completed, you can test it in the "**Basic Operations**" interface. You can observe that when the servo angle exceeds the set limit, it cannot continue to rotate further.

The slider range is [0, 1000], and our company has implemented position protection for the gripper, with a set safe range of [400, 650]. This range is also the effective adjustment range. Any movement outside this range will not result in visible action of the gripper!

Prolonged tight closure of the gripper can lead to an increase in servo temperature. Therefore, when in use, please adjust the closed position based on the actual items. A range adjustment of approximately ±50 around position 50 should be sufficient.



5.3 Adjust Servo Deviation

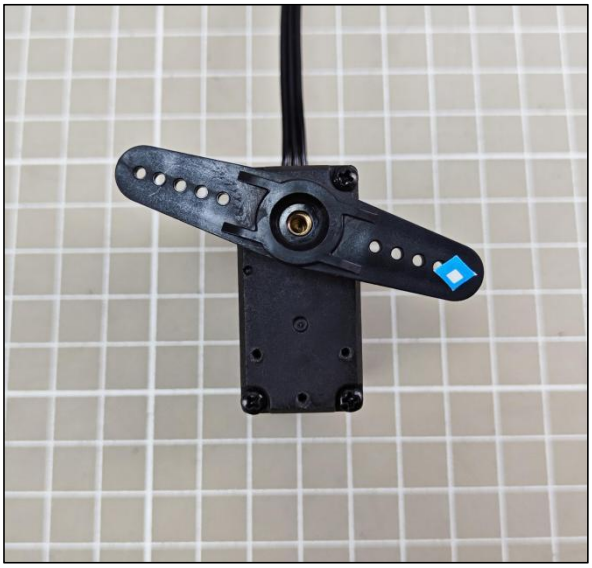
The deviation is caused by the spacing of the splines on the servo arm when assembling with the external bracket. To ensure accurate control of the servo at the desired angle, the servo deviation must be adjusted.

Deviation adjustment must be done when the servo is in the central position. The standard central posture is shown in the figure below (using the label on

the right side of the servo arm as the initial position). As shown, the servo arm and the servo are in a parallel state.

Central position: servo angle 120°（corresponding to the position of 500）

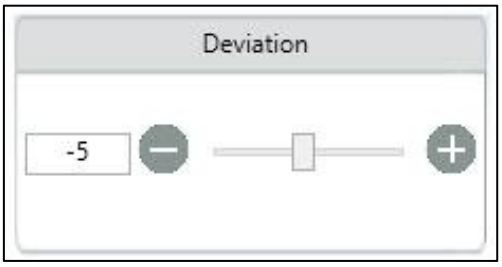
1）When the servo is in the central angle position and it does not align with the standard posture as shown above, refer to the example in the figure below:



2) Open “**Bus Servo Terminal**”. Click “**Read**” in “**Parameter Settings**” and wait for the reading to complete.

3) Locate the “**Deviation adjustment**” section and adjust the deviation by dragging the slider bar.

Note: You can also click on the slider with the mouse and use the "←→" arrow keys for fine adjustments. The maximum adjustment range through the software is ±30°.



4) When the slider is adjusted to the central posture, click "**Set**" to write the deviation value.

5) In the "**Basic Operations**" interface, rerun the central position and check if the deviation adjustment aligns with the standard posture.

5.4 Adjust Servo Speed

We can adjust the servo speed using the "**Bus Servo Control**" software, or by using a microcontroller program. Here, we will only introduce the method of adjustment via "**Bus Servo Control**" software.

For the program adjustment method, please refer to the "2. **Bus Servo Control Development Method**". For detailed information on the host "**Bus Servo Control**" software, please refer to "**2 Software**".

1) Open the software "**Bus Servo Control**" and connect the servo to the PC via the bus servo control board.

For example, the ID of the servo is 1. When it is at the center position (500), it takes 1200ms to move to position 800, and it takes 300ms to move to position 200. We want to increase the operating speed.

	Index	Time(ms)	Action
	1	1000	#1 P500
	2	1200	#1 P800
▶	3	300	#1 P200

2) Click "**Time (ms)**" and change the time to move to the position 800 to 500ms, and modify the time to move to position 200 to 100ms.

	Index	Time(ms)	Action
	1	1000	#1 P500
	2	500	#1 P800
▶	3	100	#1 P200

3) Then run the action to observe the effect, and you'll notice that the servo speed increases. Similarly, if you need to slow down the operating speed, you can achieve this by increasing the delay.

6. Servo Control Development Tutorial

Apart from basic setup and testing, microcontrollers, Jetson Nano, Raspberry Pi, etc., can directly control servo motors. Detailed information can be found in the "**2. Bus Servo Control Development Tutorial**" section.

If you need to design your own servo control circuit, you can refer to the content in the "3 Servo Manual and Drawings\Servo Self-design Circuit Notes" document.