UFACTORY

UFACTORY xARM

UFACTORY Linear Motor

SHENZHEN UFACTORY CO,. LTD

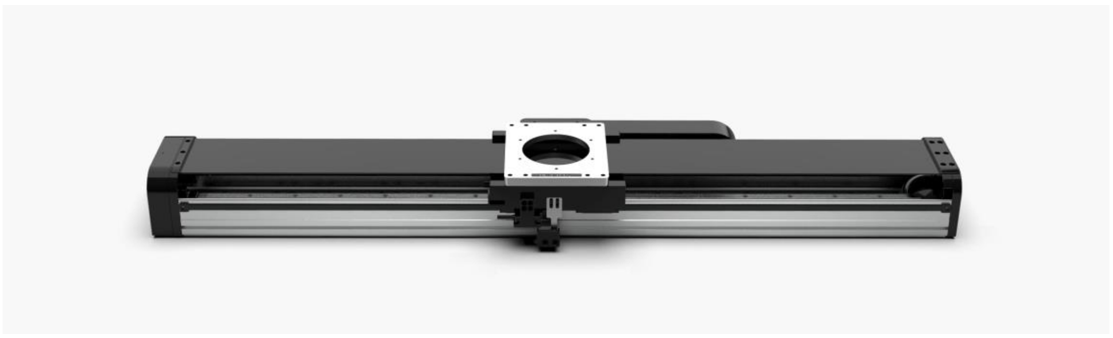
V 2.0.0

Table

1. General Presentation

1.1. UFactory Linear Motor Introduction

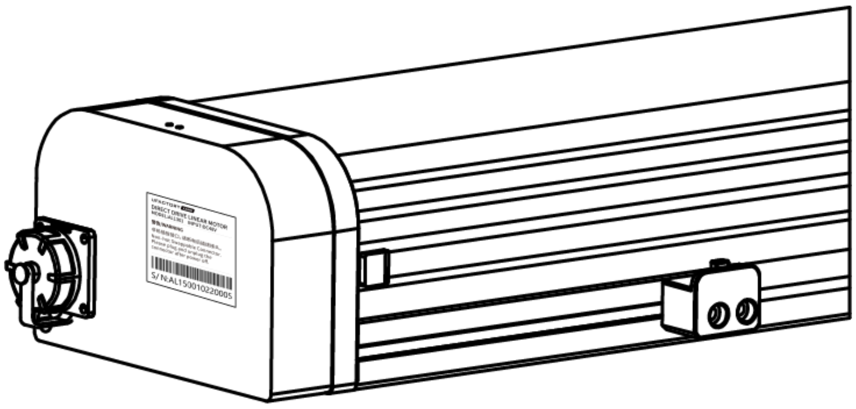
UFactory Linear Motor needs to be used with AC Control Box Pro, it mainly supports and guides the moving parts(Robotic Arm) to move smoothly according to the given direction, which significantly increases the working range of xArm.



UFactory Linear Motor

1.2. Linear Motor Model

There are two models of UFactory Linear Motor, which can be divided according to Serial Number(SN). SN can be found at the end plate of Linear Motor, see the figure below.



Zero Position: position - 0.

Speed Range: 1 to 1000(mm/s).

Position Range: depend on the model(SN) of Linear Motor.

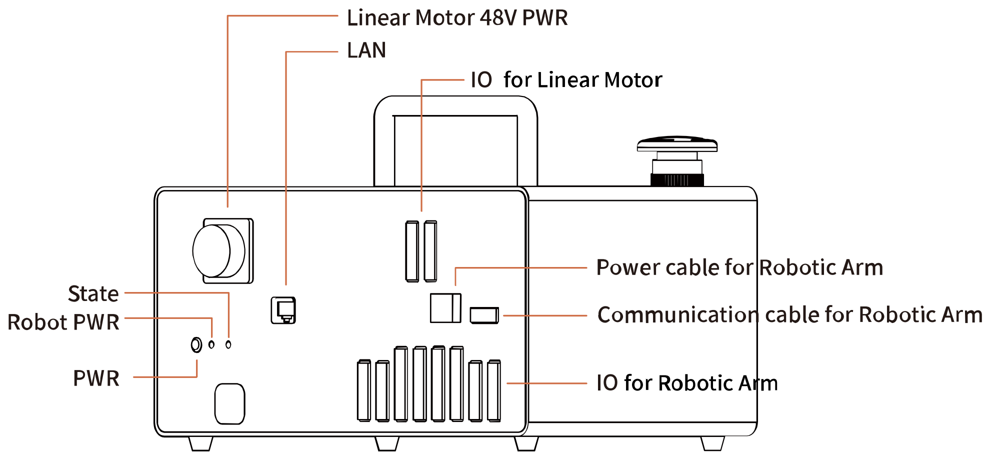
SN - AL0700: 0 to 700(mm)

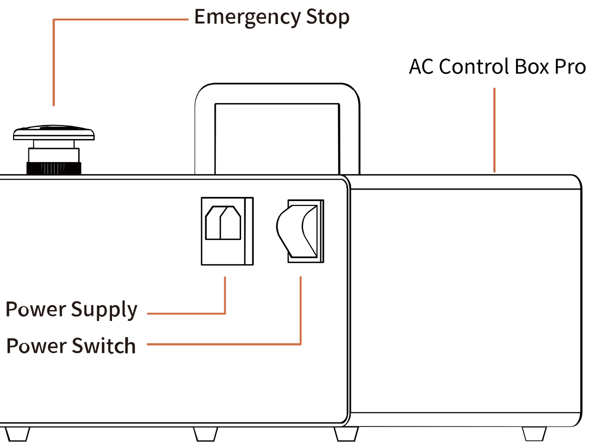
SN - AL1000: 0 to 1000(mm)

SN - AL1500: 0 to 1500(mm)

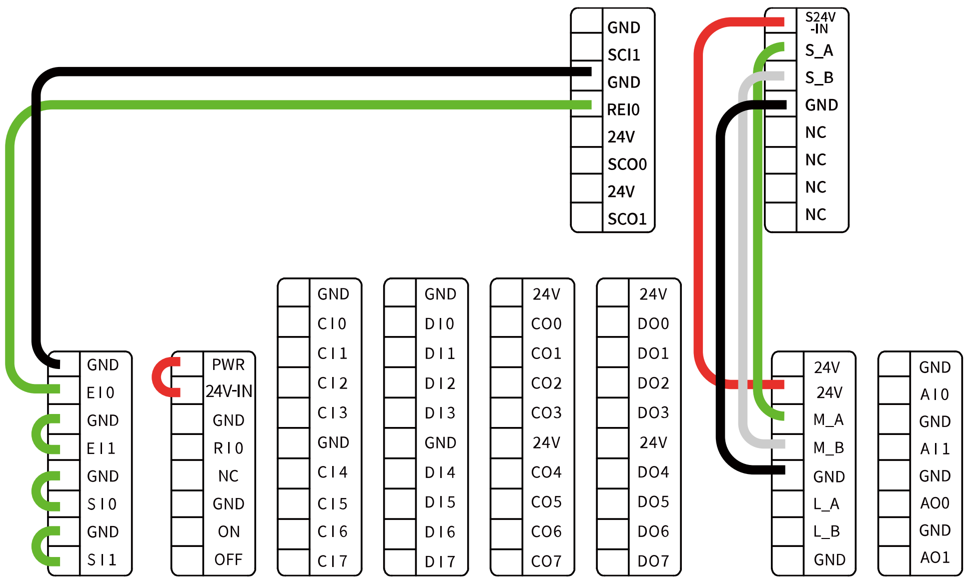
1.3. AC Control Box Pro

UFactory Linear Motor comes with AC Control Box Pro.





Control IO connection is as shown in the figure below, and it has been connected before shipping.



1.4. Safety

The operator must read and understand all the instructions below before running the Linear Motor.

1.4.1. Warning

1. The Linear Motor needs to be properly installed before operating.
2. Do not install or operate a Linear Motor that is damaged or lacking parts.
3. Never supply Linear Motor with an alternative current (AC) source.
4. Make sure all cord sets are always secured at both ends, Linear Motor end & Robot end.
5. Always satisfy the recommended Mechanical Installation.
6. Be sure nothing is in the robot and Linear Motor path before initializing the Linear Motor.

7. Set the Linear Motor speed and position accordingly, based on your application.

8. Keep magnetic scales away from strong magnets

Caution

The term "operator" refers to anyone responsible for any of the following operations on the Linear Motor:

- Installation
- Control
- Maintenance
- Inspection
- Programming
- Decommissioning

This documentation explains the various components of the Linear Motor and general operations regarding the whole life-cycle of the product from installation to operation and decommissioning.

The drawings and photos in this documentation are representative examples and differences may exist between them and the delivered product.

1.4.2. Risk Assessment and Final Application

The Linear Motor is meant to be used on an industrial robot. The robot, Linear Motor and any other equipment used in the final application must be evaluated with a risk assessment. The robot integrator must ensure that all local safety measures and regulations are respected. Depending on the application, there may be risks that need additional protection/safety measures, for example, the work-piece the Linear Motor is manipulating may be inherently dangerous to the operator.

1.4.3. Validity and Responsibility

The Linear Motor is designed for supporting and guiding the moving parts(Robotic Arm), according to the given direction of smooth reciprocating linear motion.

Caution

The product can be installed horizontally only.

No debris should be placed on the surface of the product.

The photoelectric sensor on the Linear Motor can not be disassembled.

The product is intended for installation on a robot or other automated machinery and equipment.

Info

Always comply with local and/or national laws, regulations and directives on automation safety and general machine safety.

The unit may be used only within the range of its technical data. Any other use of the product is deemed improper and unintended use.

UFACTORY will not be liable for any damages resulting from any improper or unintended use.

2. Installation

The following subsections will guide you through the installation and general setup of Linear Motor.

(1) The Scope of Delivery section

(2) The Mechanical Installation section

Warning

Before installing:

Read and understand the safety instructions related to the Linear Motor.

Verify your package according to the Scope of delivery and your order info.

Have the required parts, equipment and tools listed in the requirements readily available.

Installing:

Satisfy the environmental conditions.

Please do not disassemble the photoelectric sensor on the Linear Motor.

Do not operate the Linear Motor, or even turn on the power supply, before it is firmly anchored and the danger zone is cleared.

2.1. Scope of Delivery

A Linear Motor Kit generally includes these items:

UFactory Linear Motor *1 (including Power cable for the Linear Motor*1, Power cable for the Robotic Arm*1)

Power cable for the AC Control Box Pro*1

Communication cable for the Robotic Arm*1

AC Control Box Pro*1

Ethernet Cable*1

Head hexagon socket screws M6*20 (28)

Head hexagon socket screws M5*12 (5)

5MM L type wrench*1

4MM L type wrench*1

Debugging tool*1(USB to 485 cable)

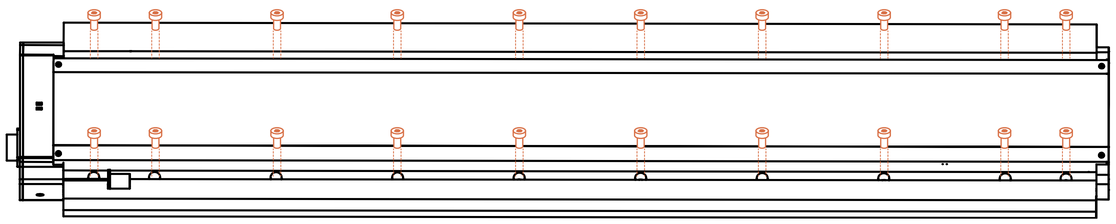
2.2. Mechanical Installation

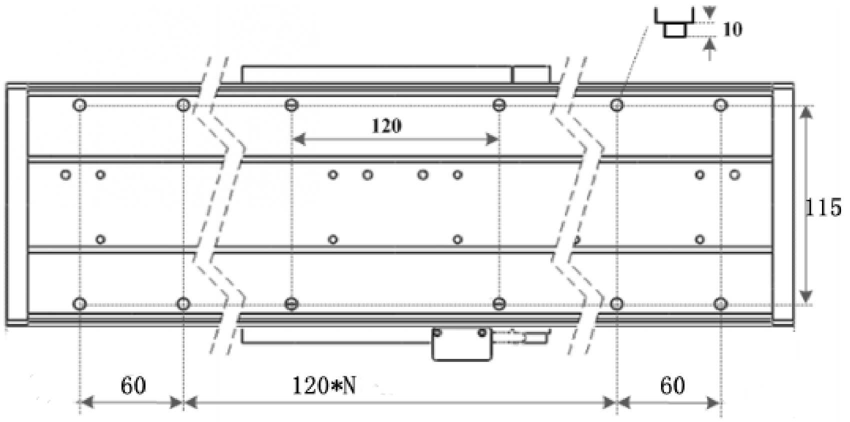
The Linear Motor is directly connected to the AC Control Box Pro via a cable, which is used for 48V DC power supply and Modbus RTU communication over RS-485.

Linear Motor installation steps (as shown below):

1. Move the Linear Motor and robotic arm to a safe position. Avoid touching other equipment or obstacles; Please install the Linear Motor horizontally only, not vertically.

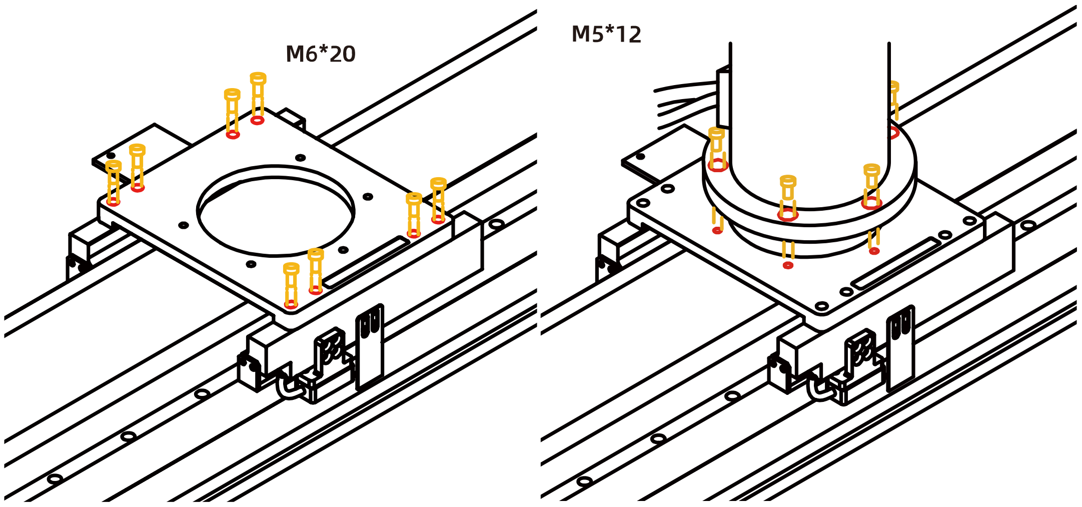
There are 20 ϕ6.2 screw holes on the linear motor which is designed for fixing the linear motor on the table or base. There are also 28 M6*20 screws in the package.





Dimension of screws holes in the linear motor (unit: mm)

2. Fix the base plate on the Linear Motor with 8 M6*20 screws.



3. Fix the robotic arm on the base plate with 5 M5*12 screws.

4. Cable connection:

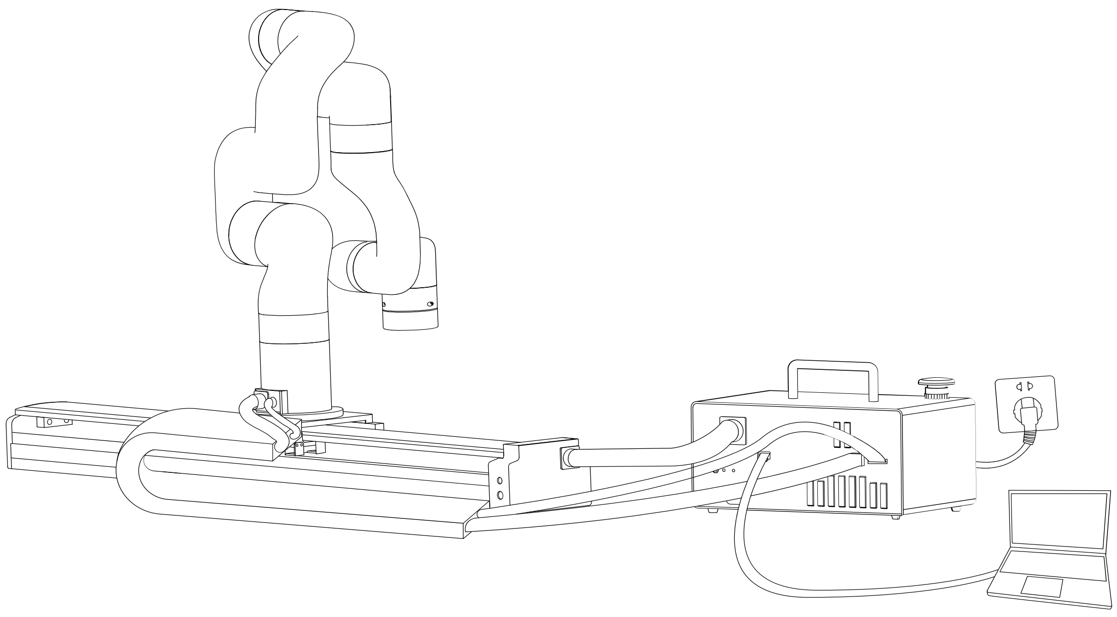
1）Plug the Linear Motor Power Supply Cable, LAN cable into control box.

2）Plug the connector of the Robotic Arm Power Supply Cable and the Robotic Arm Signal Cable into the interface of the Robotic Arm. The connector is a foolproof design. Please do not unplug and plug it violently.

3）Plug the Robotic Arm Power Supply Cable and the Robotic Arm Signal Cable into the interface of control box. The connector is a foolproof design.

5. Turn on the power switch of the control box and release the emergency stop button.

6. Enter into 'xArmStudio-Settings-Tools-Linear Motor', click 'Initialize' button to return to zero position and finish initialization.



Note:

1. When wiring the Linear Motor connection cable, be sure to power off the Robotic Arm, the emergency stop button is pressed down and the power indicator of the robotic arm is off, so as to avoid robotic arm failure caused by hot plugging;
2. The Linear Motor has no brake design, please installed horizontally only.

3. Control

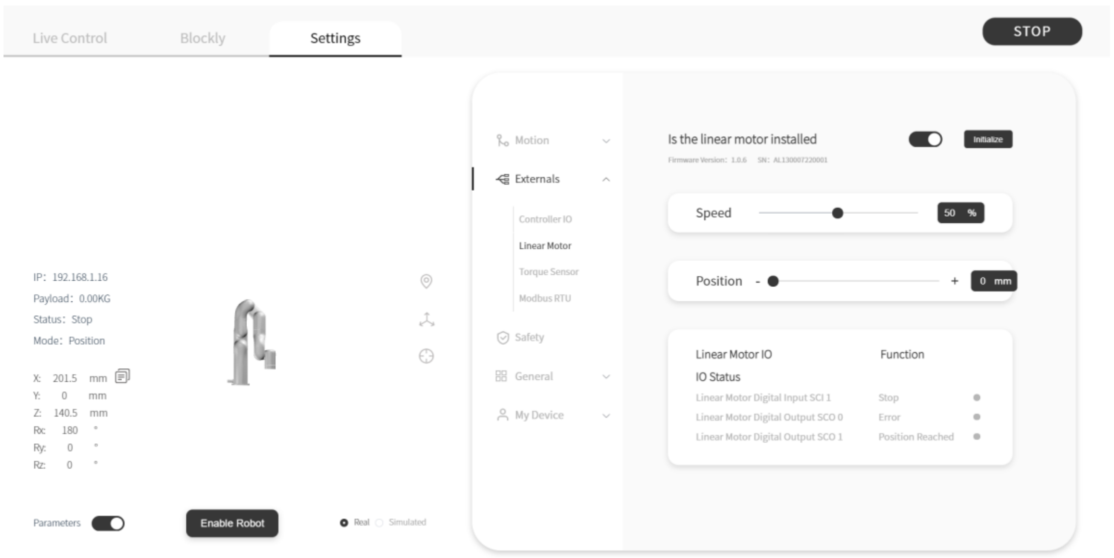
3.1. Control Linear Motor through xArm Studio

1. Set up Linear Motor

Enter [Settings]-[Tools]-[Linear Motor]

- Turn on [Is Linear Motor installed].

- Click [Initialize] button.



IP: 192.168.1.16
Payload: 0.00KG
Status: Stop
Mode: Position

X: 201.5 mm
Y: 0 mm
Z: 140.5 mm
Rx: 180 °
Ry: 0 °
Rz: 0 °

Parameters
Real
Simulated

Is the linear motor installed
Firmware Version: 1.0.6 SN: AL130007220001

Speed 50 %

Position - + 0 mm

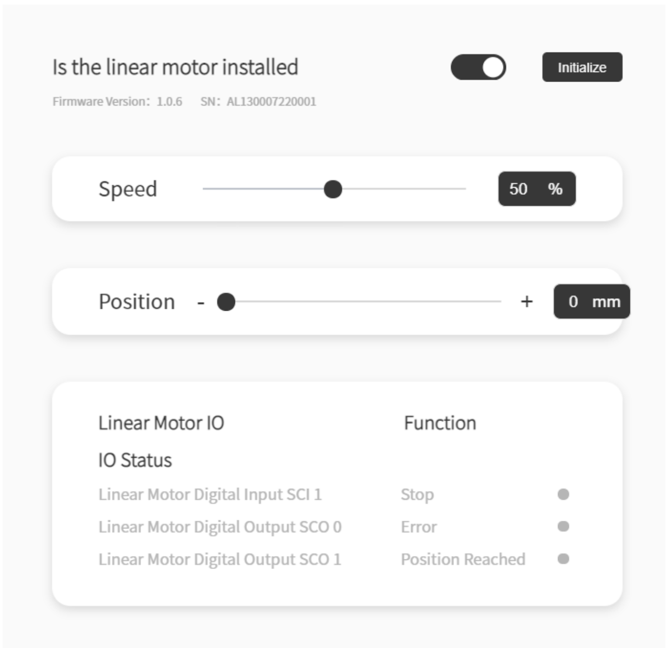
Linear Motor IO	Function
IO Status	
Linear Motor Digital Input SCI 1	Stop
Linear Motor Digital Output SCO 0	Error
Linear Motor Digital Output SCO 1	Position Reached

1. [Is Linear Motor installed] button should be turn on.

2. Click [Initialize] button to enable the Linear Motor and return to zero position. After the initialization is completed, the Linear Motor will be ready to move.

2.Control Linear Motor

1) In Linear Motor interface, the speed and position of the Linear Motor can be adjusted through the progress bar, +/- keys, and input box.



Is the linear motor installed

Initialize

Firmware Version: 1.0.6 SN: AL130007220001

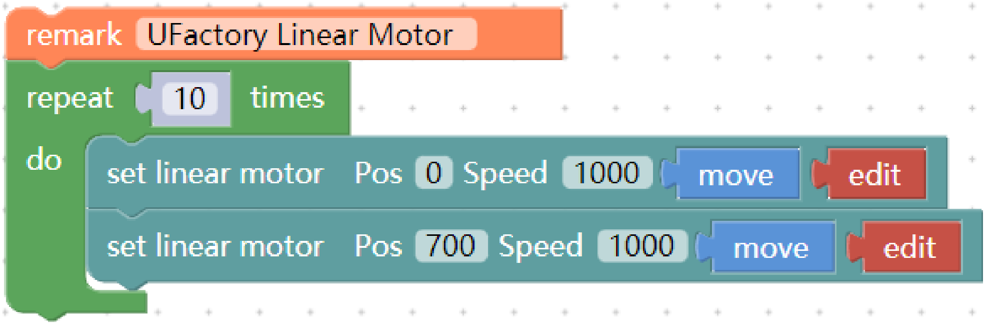
Speed 50 %

Position - + 0 mm

Linear Motor IO	Function	
IO Status		
Linear Motor Digital Input SCI 1	Stop	●
Linear Motor Digital Output SCO 0	Error	●
Linear Motor Digital Output SCO 1	Position Reached	●

2) Control the Linear Motor through Blockly

UF_Linear_Motor.Blockly



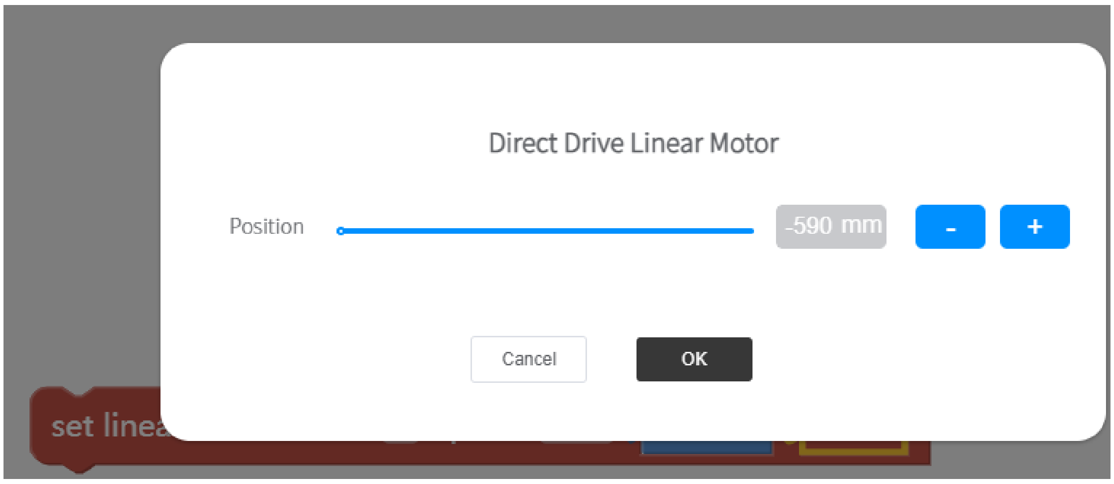
```
remark UFactory Linear Motor
repeat 10 times
do
    set linear motor Pos 0 Speed 1000 move edit
    set linear motor Pos 700 Speed 1000 move edit
```

The role of this program: execute this program to control the Linear Motor to reciprocate 10 times at the highest speed(1000mm/s) from the zero position to the farthest position.

Note:

1) Before moving the linear motor for the first time after power on, it is a must to go back to zero position and do initialization first.

2) Click 'Edit' button, can quickly modify the position of Linear Motor.



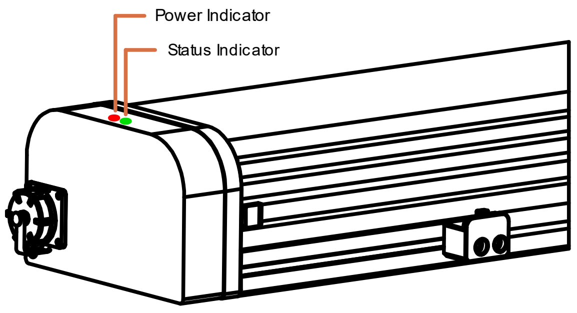
3.2. Control Linear Motor through Python-SDK

For details on controlling Linear Motor with python-SDK, please refer to the link below:

https://github.com/xArm-Developer/xArm-Python-SDK/blob/master/example/wrapper/common/9000-set_linear_track.py

3.3. Indicator

Linear Motor has two indicators, which are at the end plate of Linear Motor.



1. Power indicator: it will show red light when power on.

2. Status indicator: a steady green light indicates normal status; a flashing green light indicates there is an error.

3.4. IO Control

Linear Motor has three IOs, one Input and two Outputs.

SCI1: Emergency stop of Linear Motor, low level effective.

SCO0: Output high level, indicating there is an error of Linear Motor.

SCO1: Output high level, indicating the position has been reached.

4. Linear Motor Alarm Code

Software Error Handling:

1. If there is a software error, please refer to Error Handling Method.

2. If it does not work, please re-power on the Linear Motor. Press down the Emergency stop button on the AC Control Box Pro, release it after 5 seconds, and click xArmStudio 'initialize' button to enable and initialize the Linear Motor.

If the problem remains unsolved after power on/off multiple times, please contact UFACTORY team for support.

Software Error Code	Error Handling

T9	Linear Motor Current Detection Error Please restart the Controller. If multiple reboots are not working, please contact technical support.
T11	Linear Motor Current Overlimit Please click “Clear Error” clear the Linear Motor error.
T12	Linear Motor Speed Overlimit Please click “Clear Error” clear the Linear Motor error.
T13	Linear Motor Large Motor Position Deviation Please check if the movement of the Linear Motor is blocked, if not, please click “Clear Error” clear the Linear Motor error.
T14	Linear Motor Position Command Overlimit Please click “Clear Error” clear the Linear Motor error.
T15	Linear Motor EEPROM Read and Write Error Please click “Clear Error” clear the Linear Motor error.
T20	Linear Motor Driver IC Hardware Error Please click “Clear Error” clear the Linear Motor error.
T21	Linear Motor Driver IC Initialization Error Please click “Clear Error” clear the Linear Motor error.
T25	Linear Motor Command Over Software Limit Please check if the Linear Motor command is set beyond the software limit.
T26	Linear Motor Feedback Position Software Limit Please contact technical support.
T33	Linear Motor Drive Overloaded Please contact technical support.
T34	Linear Motor Motor Overload Please contact technical support.
T36	Linear Motor Driver Type Error Please click “Clear Error” clear the Linear Motor error.
For alarm codes that are not listed in the above table: Please click “Clear Error” clear the Linear Motor error.If it reports the same error repeatedly, please contact technical support.	

Appendix:

xArm-Python-SDK alarm processing method:

When designing the Linear Motor path with the Python library, if the errors appear, you need to manually clear the errors. After clearing the error, re-enable the Linear Motor.

1.error clearing: clean_linear_track_error()

2. initialize and enable Linear Motor if necessary:

set_linera_track_enable()

set_linear_track_back_origin()

5. Linear Motor Technical Specifications

UFactory Linear Motor	
Continue	62N
Peak force	160N
Maximum speed	1000mm/s
Travel distance	AL0700:0-700mm AL1000:0-1000mm

	AL1500:0-1500mm
Maximum load mass	200kg
Communication Mode	RS-485
Communication Protocol	Modbus RTU
Programmable Gripping Specification	Position, Speed
Feedback	Position
Dimensions(L*W*H)	AL0700: 1092.6*213*114.6mm AL1000: 1382*214*115mm AL1500: 1884.6*213*114.6mm
Weight	AL0700:20kg； AL1000:24kg AL1500:30.4kg
AC Control Box Pro Dimensions(L*W*H)	262*185*176mm
AC Control Box Pro Weight	4.44Kg

6. After-sales Service

1. After-sales policy:

For the detailed after-sales policy of the product, see the official website:

https://store-ufactory-cc.myshopify.com/pages/warranty-returns

1. The general process of after-sales service is:

(1) Contact UFACTORY technical support (support@ufactory.cc) to confirm whether the product needs to repair and which part should be sent back to UFACTORY.

(2) After the bill of lading on UPS/DHL, we will send the invoice and label to you by mail. You need to make an appointment with the local UPS/DHL and then send the product to us.

(3) UFACTORY will check the product warranty status according to the after-sales policy.

(4) Generally, the process takes around 1-2 weeks except for shipment.

Note:

1. Please keep the original packaging materials of the product. When you need to send the product back to get repaired, please pack the

product with the original box to protect the product during the transportation.

Appendix

Appendix1 - Use Modbus-RTU communication Protocol to Control Linear Motor

1.1 Modbus RTU Communication Format

The Linear Motor defaults to the standard Modbus RTU protocol at a default baudrate is **2Mbps** and the slave ID is **0x01**. The currently supported function codes are: 0x03 / 0x06/ 0x10.

Commonly used address for Linear Motor are: 0x0100, 0x0700, 0x0303, 0x0A0A, 0x0404, 0x004F.

If need to store EEPROM, perform an '|' operation with the communication address and 0x1000.

For example, write servo operation mode to EEPROM, the communication address(0x0A0B) should be changed to 0x1A0B.

Note: Linear Motor needs to be initialized after each powered on.

1.2 Read Linear Motor Register

Read Register			
Request			
Modbus RTU Data	Slave ID (Linear Motor)	1 Byte	0x01
	Function Code	1 Byte	0x03
	Register Starting Address	2 Bytes	**Address**
	Quantity of Register	2 Bytes	**N***
	Modbus CRC16	2 Bytes	**CRC***
Response			
Modbus RTU Data	Slave ID	1 Byte	0x01
	Function Code	1 Byte	0x03
	Byte Count	1 Byte	**N*x2**
	Registers Value	**N***x2 Bytes	**Value**
	Modbus CRC16	2 Bytes	**CRC***

N* = Quantity of Registers

Address = Register Starting Address

CRC* = Cyclic Redundancy Check

	Resgister Starting Address	Registers Value	
Get Linear Motor status Register	0x0000	2 Bytes	**End status:** 0x0000 **Motion status:** 0x0001 **Stop status:** 0x0002
Get Linear Motor position Register	0x0702	4bytes	0xFFFFFFFB-0x00000320
Get Linear Motor Error Register	0x000F	2 Bytes	**An error occurs:** all other return values indicate an error(except 0)

			No error occurred: 0x0000

1.3 Write Linear Motor Register

Write Register			
Request			
Modbus RTU Data	Slave ID (Linear Motor)	1 Byte	0x01
	Function Code	1 Byte	0x10
	Register Starting Address	2 Bytes	**Address**
	Quantity of Register	2 Bytes	**N***
	Byte Count	1 Byte	**N*x2**
	Registers Value	**N***x2 Bytes	**Value**
	Modbus CRC16	2 Bytes	**CRC***
Response			
Modbus RTU Data	Slave ID	1 Byte	0x01
	Function Code	1 Byte	0x10
	Register Starting Address	2 Bytes	**Address**
	Quantity of Registers	2 Bytes	**N***
	Modbus CRC16	2 Bytes	**CRC***

N* = Quantity of Registers

Address = Register Starting Address

CRC* = Cyclic Redundancy Check

1.4 Modbus RTU Example

1. Enable Linear Motor

enable Linear Motor			
Request			
Modbus RTU Data	Slave ID (Linear Motor)	1 Byte	0x01
	Function Code	1 Byte	0x10
	Register Starting Address	2 Bytes	0x01,0x00

	Quantity of Registers	2 Bytes	0x00,0x01
	Byte Count	1 Byte	0x02
	Registers Value	2 Bytes	0x00,0x01
	Modbus CRC16	2 Bytes	0x1D,0x00
Response			
Modbus RTU Data	Slave ID	1 Byte	0x01
	Function Code	1 Byte	0x10
	Register Starting Address	2 Bytes	0x01,0x00
	Quantity of Registers	2 Bytes	0x00,0x01
	Modbus CRC16	2 Bytes	0x00,0xAC

2. Set Linear Motor position

Set Linear Motor position			
Request			
Modbus RTU Data	Slave ID (Linear Motor)	1 Byte	0x01
	Function Code	1 Byte	0x10
	Register Starting Address	2 Bytes	0x07,0x00
	Quantity of Registers	2 Bytes	0x00,0x02
	Byte Count	1 Byte	0x04
	Registers Value	4Bytes	0x00,0x1E,0x84,0x80
	Modbus CRC16	2 Bytes	0x7B,0x62
Response			
Modbus RTU Data	Slave ID	1 Byte	0x01
	Function Code	1 Byte	0x10
	Register Starting Address	2 Bytes	0x07,0x00
	Quantity of Registers	2 Bytes	0x00,0x02
	Modbus CRC16	2 Bytes	0x40,0x25

3. Set Linear Motor Speed

Set Linear Motor Speed			
Request			
Modbus RTU Data	Slave ID (Linear Motor)	1 Byte	0x01
	Function Code	1 Byte	0x10
	Register Starting Address	2 Bytes	0x03,0x03
	Quantity of Registers	2 Bytes	0x00,0x01
	Byte Count	1 Byte	0x02

	Registers	2 Bytes	0x17,0x70
	Modbus CRC16	2 Bytes	0xFD,0xFA
Response			
Modbus RTU Data	Slave ID	1 Byte	0x01
	Function Code	1 Byte	0x10
	Register Starting Address	2 Bytes	0x03,0x03
	Quantity of Registers	2 Bytes	0x00,0x01
	Modbus CRC16	2 Bytes	0xF1,0x14

4. Set Linear Motor to Zero position

Set Linear Motor tozero position			
Request			
Modbus RTU Data	Slave ID (Linear Motor)	1 Byte	0x01
	Function Code	1 Byte	0x06
	Register Starting Address	2 Bytes	0x0A,0x0A
	Quantity of Registers	2 Bytes	0x00,0x01
	Modbus CRC16	2 Bytes	0xFD,0xFA
Response			
Modbus RTU Data	Slave ID	1 Byte	0x01
	Function Code	1 Byte	0x10
	Register Starting Address	2 Bytes	0x0A,0x0A
	Quantity of Registers	2 Bytes	0x00,0x01
	Modbus CRC16	2 Bytes	0xFD,0xFA

5. Set Linear Motor speed to Zero position

Set Linear Motor speed to zero position			
Request			
Modbus RTU Data	Slave ID (Linear Motor)	1 Byte	0x01
	Function Code	1 Byte	0x10
	Register Starting Address	2 Bytes	0x04,0x04
	Quantity of Registers	2 Bytes	0x00,0x01
	Byte Count	1 Byte	0x02
	Registers	2 Bytes	0x0B,0xB8
	Modbus CRC16	2 Bytes	0xFD,0xFA
Response			
Modbus RTU Data	Slave ID	1 Byte	0x01
	Function Code	1 Byte	0x10

	Register Starting Address	2 Bytes	0x04,0x04
	Quantity of Registers	2 Bytes	0x00,0x01
	Modbus CRC16	2 Bytes	0xF1,0x14

6. Get if Linear Motor is back to zero

Get if Linear Motor is back to zero			
Request			
Modbus RTU Data	Slave ID (Linear Motor)	1 Byte	0x01
	Function Code	1 Byte	0x03
	Register Starting Address	2 Bytes	0x0A,0x25
	Quantity of Registers	2 Bytes	0x00,0x01
	Modbus CRC16	2 Bytes	0xB5,0xDD
Response			
Modbus RTU Data	Slave ID	1 Byte	0x01
	Function Code	1 Byte	0x03
	Byte Count	1 Byte	0x02
	Registers Value	2 Bytes	0x00,0x01
	Modbus CRC16	2 Bytes	0x79,0x84

7. Monitor the distance between the photoelectric sensor and the fisrt Z phase

Monitor the distance			
Request			
Modbus RTU Data	Slave ID (Linear Motor)	1 Byte	0x01
	Function Code	1 Byte	0x03
	Register Starting Address	2 Bytes	0x0A,0x28
	Quantity of Registers	2 Bytes	0x00,0x01
	Modbus CRC16	2 Bytes	0xE4,0x1D
Response			
Modbus RTU Data	Slave ID	1 Byte	0x01
	Function Code	1 Byte	0x03
	Byte Count	1 Byte	0x02
	Registers Value	2 Bytes	0x06,0xF2
	Modbus CRC16	2 Bytes	0x3A,0x61

8. Locating end range

Locating end range			
Request			
Modbus RTU Data	Slave ID (Linear Motor)	1 Byte	0x01
	Function Code	1 Byte	0x10
	Register Starting Address	2 Bytes	0x0A,0x0B
	Quantity of Registers	2 Bytes	0x00,0x01
	Byte Count	1 Byte	0x02
	Registers Value(1000)	2 Bytes	0x03,0xE8
	Modbus CRC16	2 Bytes	0x0C,0xC0
Response			
Modbus RTU Data	Slave ID	1 Byte	0x01
	Function Code	1 Byte	0x10
	Register Starting Address	2 Bytes	0x0A,0x0B
	Quantity of Registers	2 Bytes	0x00,0x01
	Modbus CRC16	2 Bytes	0x73,0xD3

9. Soft emergency stop

trigger soft emergency stop			
Request			
Modbus RTU Data	Slave ID (Linear Motor)	1 Byte	0x01
	Function Code	1 Byte	0x10
	Register Starting Address	2 Bytes	0x0A,0x0B
	Quantity of Registers	2 Bytes	0x00,0x01
	Byte Count	1 Byte	0x02
	Registers Value(1000)	2 Bytes	0x00,0x01
	Modbus CRC16	2 Bytes	0xCC,0xBE
Response			
Modbus RTU Data	Slave ID	1 Byte	0x01
	Function Code	1 Byte	0x10
	Register Starting Address	2 Bytes	0x0A,0x0E
	Quantity of Registers	2 Bytes	0x00,0x01
	Modbus CRC16	2 Bytes	0x63,0xD2

10. get Linear Motor SN

get Linear Motor SN			
Request			
Modbus RTU Data	Slave ID (Linear Motor)	1 Byte	0x01

	Function Code	1 Byte	0x03
	Register Starting Address	2 Bytes	0x0B,0x10
	Quantity of Registers	2 Bytes	0x00,0x02
	Modbus CRC16	2 Bytes	0xB5,0xDD
Response			
Modbus RTU Data	Slave ID	1 Byte	0x01
	Function Code	1 Byte	0x03
	Byte Count	1 Byte	0x04
	Registers Value	2 Bytes	**
	Modbus CRC16	2 Bytes	**

11. get input SCI status

get input SCI status			
Request			
Modbus RTU Data	Slave ID (Linear Motor)	1 Byte	0x01
	Function Code	1 Byte	0x03
	Register Starting Address	2 Bytes	0x0A,0x26
	Quantity of Registers	2 Bytes	0x00,0x01
	Modbus CRC16	2 Bytes	0x86,0x17
Response			
Modbus RTU Data	Slave ID	1 Byte	0x01
	Function Code	1 Byte	0x03
	Byte Count	1 Byte	0x02
	Registers Value	2 Bytes	**
	Modbus CRC16	2 Bytes	**

12. get output SCO status

get output SCO status			
Request			
Modbus RTU Data	Slave ID (Linear Motor)	1 Byte	0x01
	Function Code	1 Byte	0x03
	Register Starting Address	2 Bytes	0x0A,0x27
	Quantity of Registers	2 Bytes	0x00,0x01
	Modbus CRC16	2 Bytes	0xD7,0xD7
Response			
Modbus RTU Data	Slave ID	1 Byte	0x01
	Function Code	1 Byte	0x03

	Byte Count	1 Byte	0x02
	Registers Value	2 Bytes	**
	Modbus CRC16	2 Bytes	**

13. get status area

address start from 0x0A20

get status area			
Request			
Modbus RTU Data	Slave ID (Linear Motor)	1 Byte	0x01
	Function Code	1 Byte	0x03
	Register Starting Address	2 Bytes	0x0A,0x20
	Quantity of Registers	2 Bytes	0x00,0x08
	Modbus CRC16	2 Bytes	0x46,0x1E
Response			
Modbus RTU Data	Slave ID	1 Byte	0x01
	Function Code	1 Byte	0x03
	Byte Count	1 Byte	0x10
	Registers Value	16 Bytes	**
	Modbus CRC16	2 Bytes	**

Registers Value - 16 Bytes: 00 00 0F A0 00 00 00 00 00 01 00 01 00 02 00 02

1-4 bytes: 00 00 0F A0 , current position(unit: number of pulses)

5-6 bytes: 00 00, get Linear Motor status(same as address 0x0000)

7-8 bytes: 00 00, error code

9-10 bytes: 00 01, enable status

11-12 bytes: 00 01, if get back to zero

12-14 bytes: 00 02, SCI status

15-16 bytes: 00 02, SCO status

14. Clear Linear Motor Error

enable Linear Motor			
Request			
Modbus RTU Data	Slave ID (Linear Motor)	1 Byte	0x01

	Function Code	1 Byte	0x10
	Register Starting Address	2 Bytes	0x01,0x09
	Quantity of Registers	2 Bytes	0x00,0x01
	Byte Count	1 Byte	0x02
	Registers Value	2 Bytes	0x00,0x01
	Modbus CRC16	2 Bytes	0x57,0x0C
Response			
Modbus RTU Data	Slave ID	1 Byte	0x01
	Function Code	1 Byte	0x10
	Register Starting Address	2 Bytes	0x01,0x09
	Quantity of Registers	2 Bytes	0x00,0x01
	Modbus CRC16	2 Bytes	0xD0,0x37