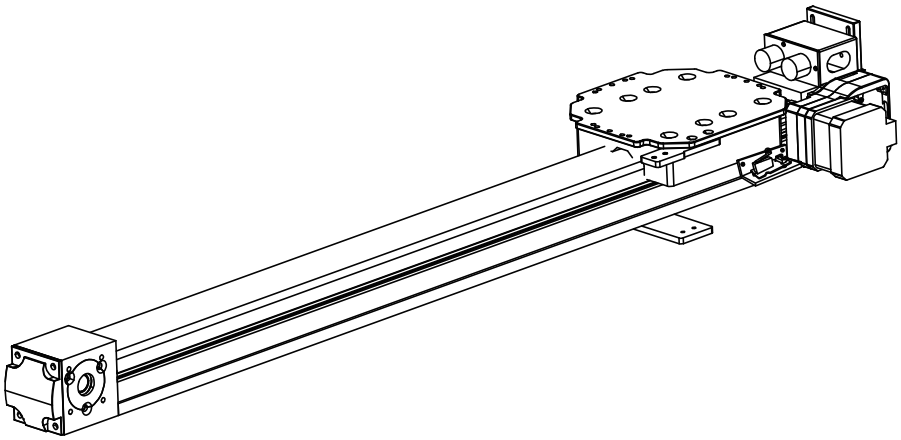
UFACTORY

SLIDER

User Manual

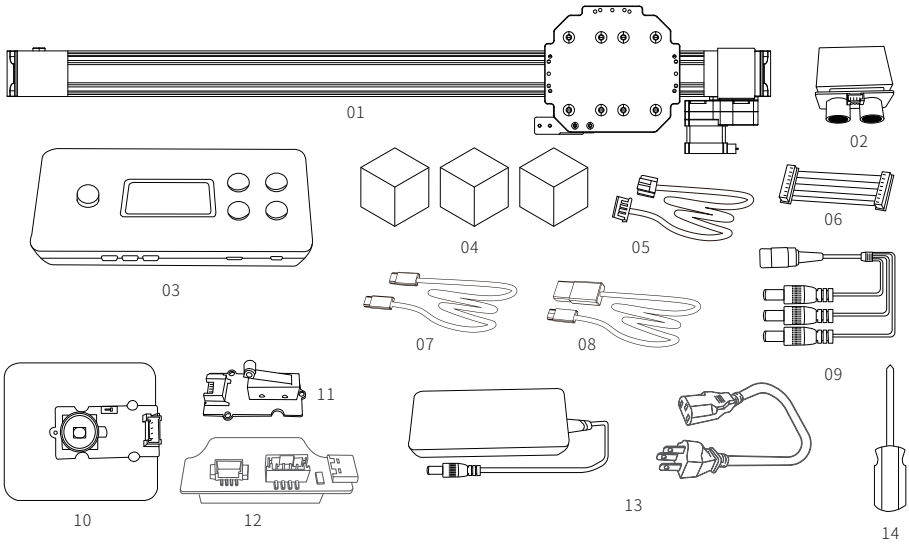


SHENZHEN UFACTORY CO.,LTD.

V2.1

Contents

1 Parts List



1.1 Hardware

01.Slider * 1

02.Ultrasonic sensor * 1

03.uArm Controller * 1

04.Target Object (Red Cube, Green Cube, Yellow Cube) * 1

05.Seeed Grove Sensor Cable* 3

06.Slider Motor 8-Pin cable * 1

07.USB Type C Cable * 1

08.USB Cable* 1

09.DC Power Supply 3-in-1 cable * 1

10.Color sensor * 1

11.Limit Switch *1

12.uArm 30P extendable base *1

13.Power adapter * 1

14.Screwdriver * 1

1.2 Software

1.Arduino IDE

http://www.arduino.cc

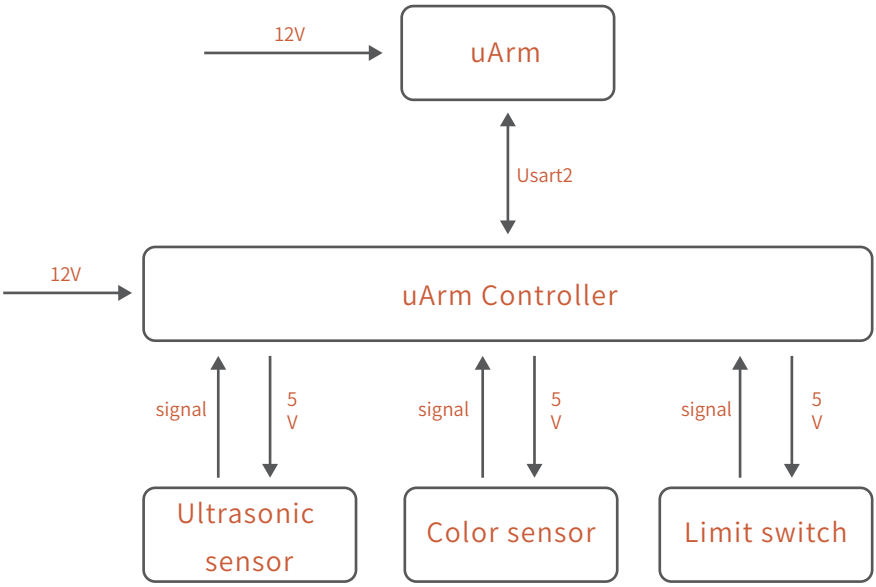
2.Slider.ino for Controller

https://bit.ly/2I8tvvP

3.uArmPro_V4.X.X.hex for uArm

https://github.com/uArm-Developer/SwiftProForArduino/tree/Version_V4.0/hex

2 System Structure



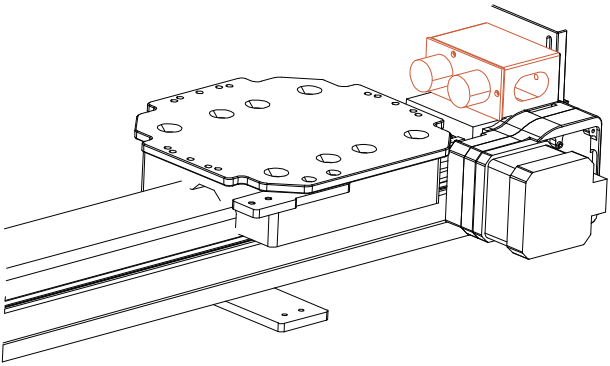
3 Installation Tutorial

3.1 Hardware installation tutorial

3.1.1 Suite Installation Tutorial

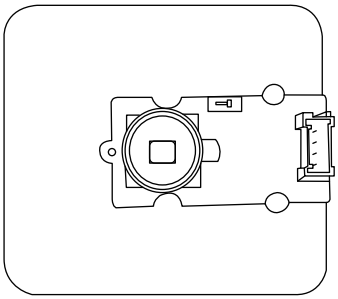
(1) Install the ultrasonic module

The mounting position for the Ultrasonic sensor



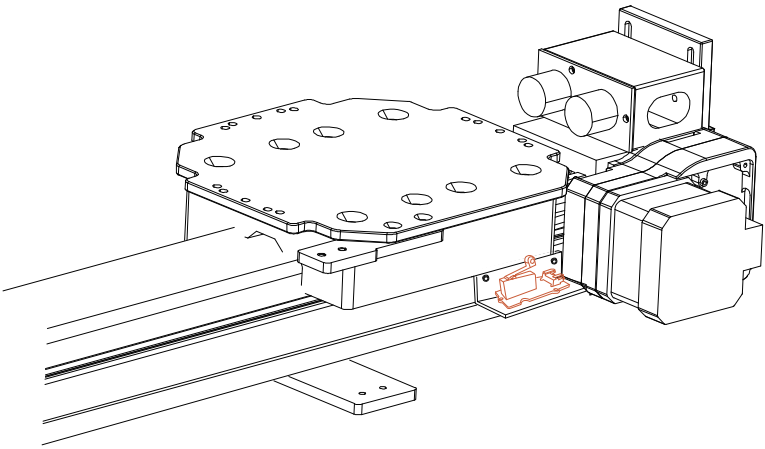
Put the ultrasonic sensor to the corresponding position of the slide rail.

(2) Color sensor



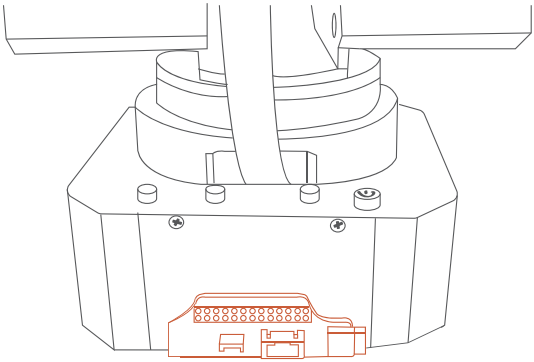
After locking in the position of the color sensor, users may run the system once to observe the gripping position of uArm, and then place the color sensor according to the grabbing position of the uArm.

(3)Limit switch



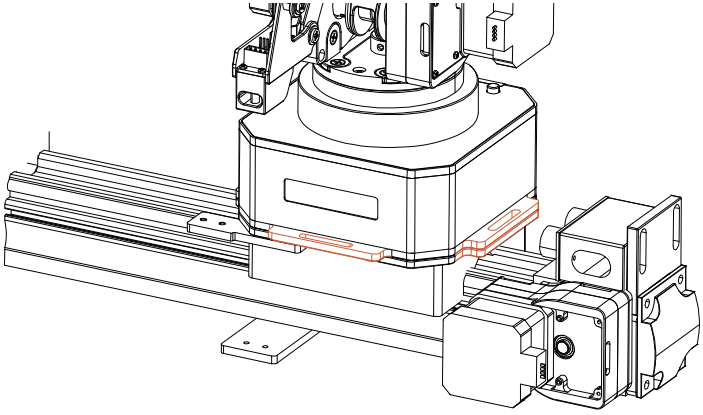
The main function of the limit switch module is to establish the uArm starting point and assist the uArm to reset.

(5) Install uArm 30P extendable base.



Install uArm 30P extendable base to the rear interface on uArm

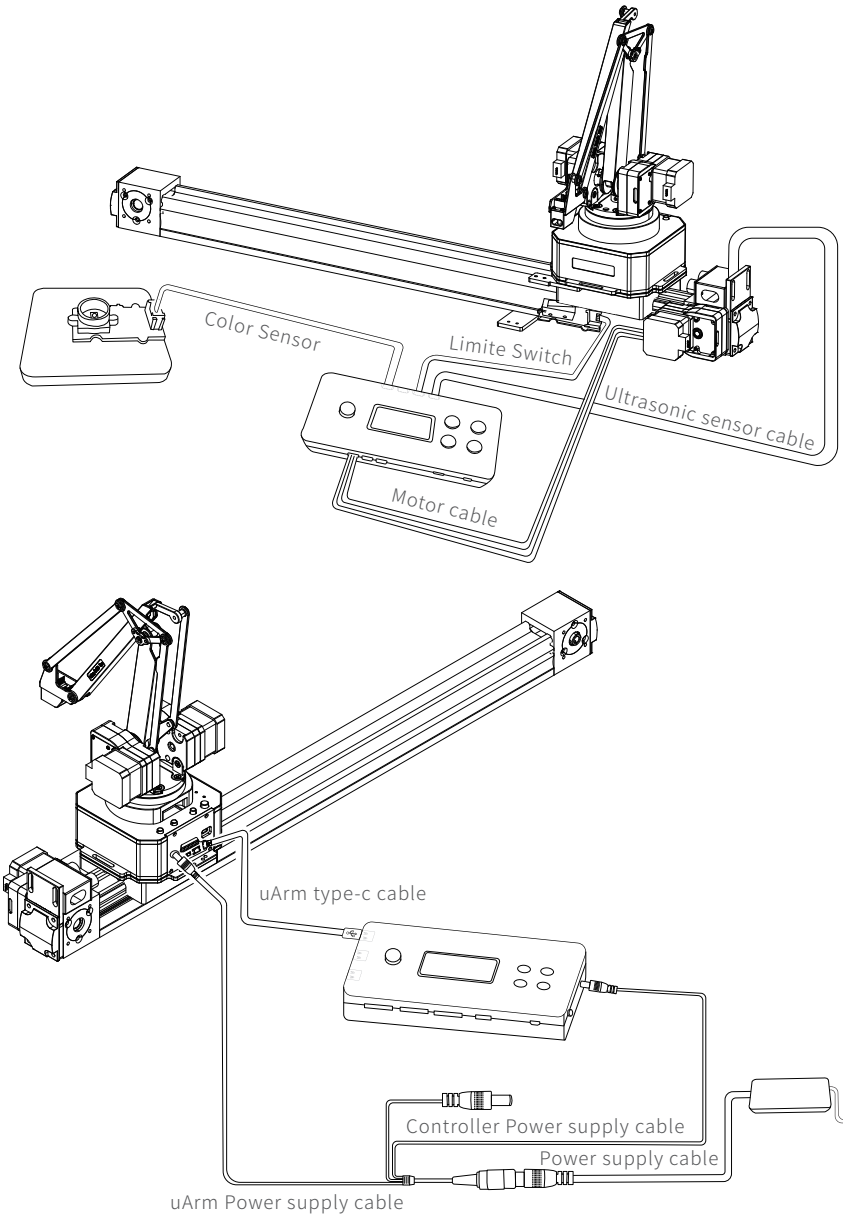
(6)Fix the uArm



Put uArm on the fixing plate of the slider.

3.1.2 Wiring Tutorial

Connecting cables of each module.



3.2 Software Installation Tutorial

3.2.1uArm controller & uArm Swift Pro Firmware Writing Tutorial

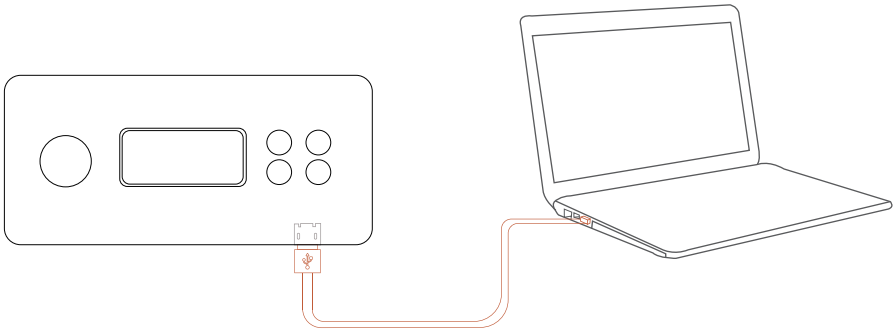
uArm Controller Firmware Writing

The firmware of uArm Controller has been written before leaving the factory. If you need to re-write the firmware, please refer to the following steps.

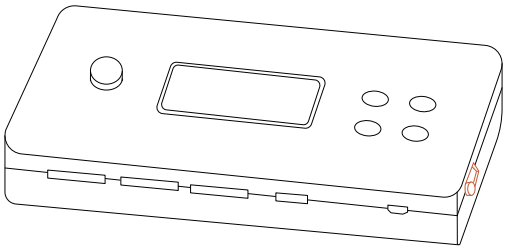
(1) Download firmware: Slider.ino for Arduino Mega 2560

[https://bit.ly/2I8tvvP]

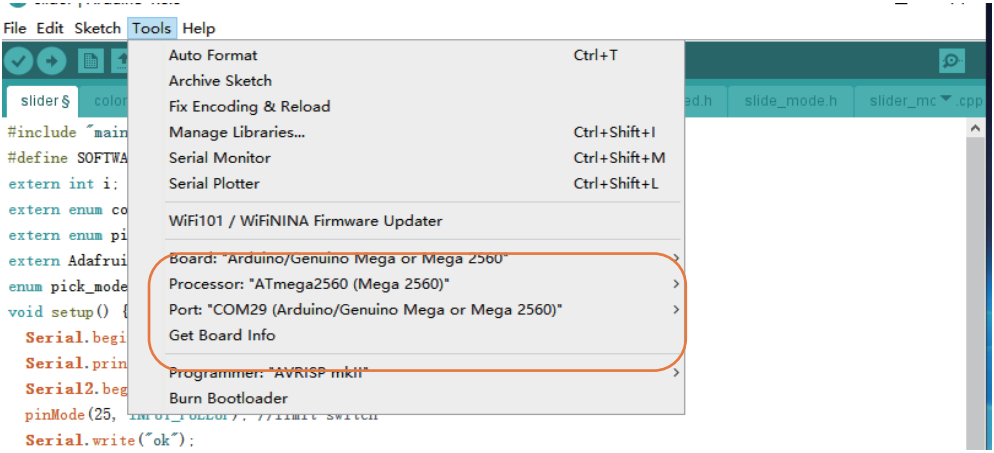
(2) Connect the uArm Controller to the computer with a USB cable

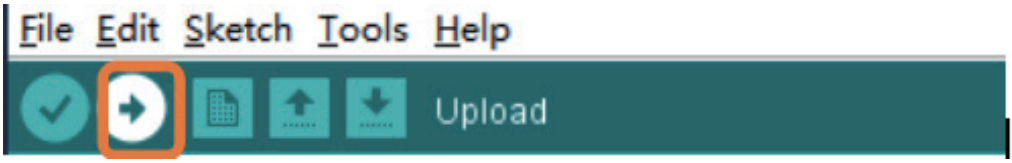


(3) Turn on the switch



(4) Open the firmware in the Arduino IDE and set the parameters as shown below to send the firmware to the uArm Controller.

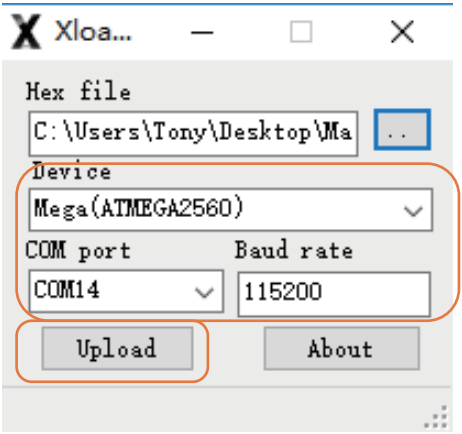




uArm Swift Pro firmware Writing tutorial

Connect uArm Swift Pro to your computer. Open XLoader (xloader.russemotto.com/) and load uArmPro_V4.X.X.hex (download link：https://github.com/uArm-Developer/SwiftProForArduino/tree/Version_V4.0/hex)

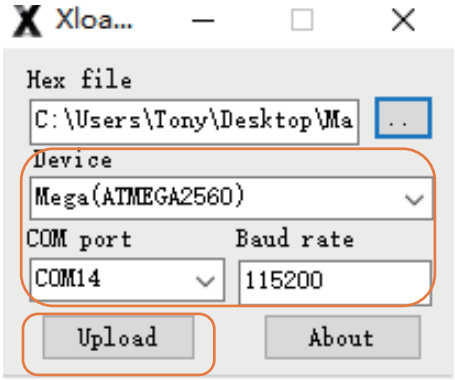
Click the "Upload" button to upload the code to uArm Swift Pro.



3.2.2 Firmware Recovery

A dedicated firmware has been flashed for uArm Swift Pro, such as firmware is not able to control the robotic arm with uArm Studio. To control the robotic arm with uArm Studio, please restore the firmware as follows:

Connect uArm Swift Pro to your computer, open XLoader (xloader.russemotto.com/), and load SWIFTPRO3.2.0.hex (http://download.ufactory.cc/firmware/SWIFTPRO3.2.0.hex?attname=). Click the "Upload" button to upload the code to uArm Swift Pro.



3.3 uArm Slider Demo

After power-on, the uArm will perform the resetting, if the uArm is within the limit switch, the uArm will move to the right first and then reset to prevent the starting point from deviating. When the color sensor recognizes the cube, uArm grabs it and places it in a different location depending on the color of the cube.

Video Demo：

Note: If the cube is placed on the color sensor, and uArm is not able to grab, it is likely due to the value of the color sensor recognition is inconsistent with the preset color cube value, which makes it impossible to follow the default procedure. You can view the real-time values by uncommenting the following code and connecting the uArm Controller to your computer.

```
void get_color()
{
  r = tcs.read16(TCS34725_RDATAL) / 255;
  g = tcs.read16(TCS34725_GDATAL) / 255;
  b = tcs.read16(TCS34725_BDATAL) / 255;
  r = constrain(r, 0, 255);
  b = constrain(b, 0, 255);
  g = constrain(g, 0, 255);
//  Serial.print("r:"); Serial.println(r);
//  Serial.print("b:"); Serial.println(b);
//  Serial.print("g:");  Serial.println(g);
```

Taking a yellow cube as an example, r, b, and g are the three-color values of the yellow cube read.

```
r:255
b:96
g:255
Yellow
r:255
b:96
g:255
Yellow
r:255
b:96
g:255
Yellow
```

According to the real-time value of the color cube, the following values are changed, R corresponds to the red cube, Y corresponds to the yellow square, and G corresponds to the green cube.
RED, GREEN and BLUE are the three-color values.

```
void slider_rgb_r()
{
  R_RED     =  140;
  R_GREEN   =   45;
  R_BLUE    =   37;
}
void slider_rgb_y()
{
  Y_RED     =  255;
  Y_GREEN   =  255;
  Y_BLUE    =  100;
}
void slider_rgb_g()
{
  G_RED     =  151;
  G_GREEN   =  218;
  G_BLUE    =   89;
}
```

Red Cube r,g,b value

Yellow Cube r,g,b value

Green Cube r,g,b value

Ufactoty2013

UFACTORY_UF

UFACTORY

Website: www.ufactory.cc | Email:info@ufactory.cc
Address:2F, Building M-6, Ma Que Ling Industrial Zone,
Nanshan District, Shenzhen, Guangdong, P.R. China