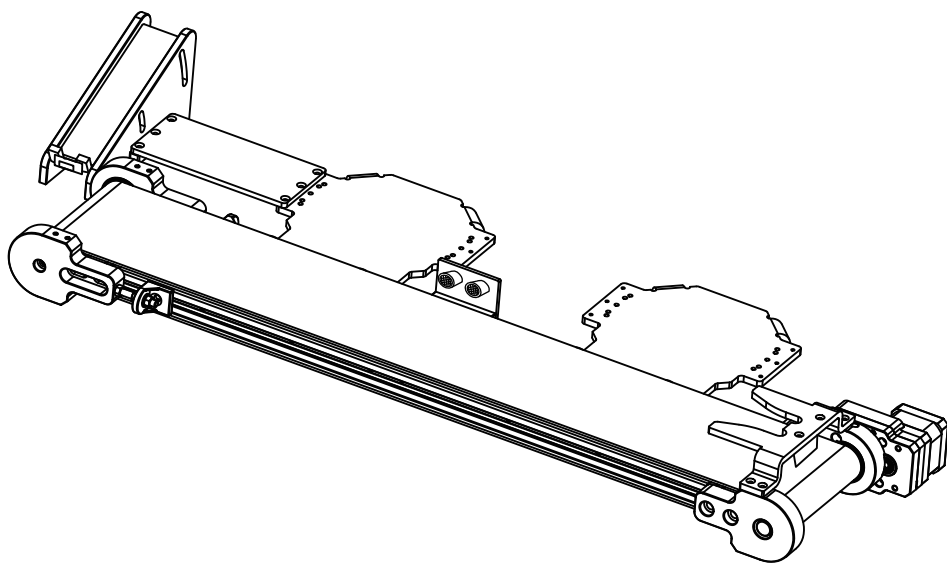


CONVEYOR BELT

User Manual



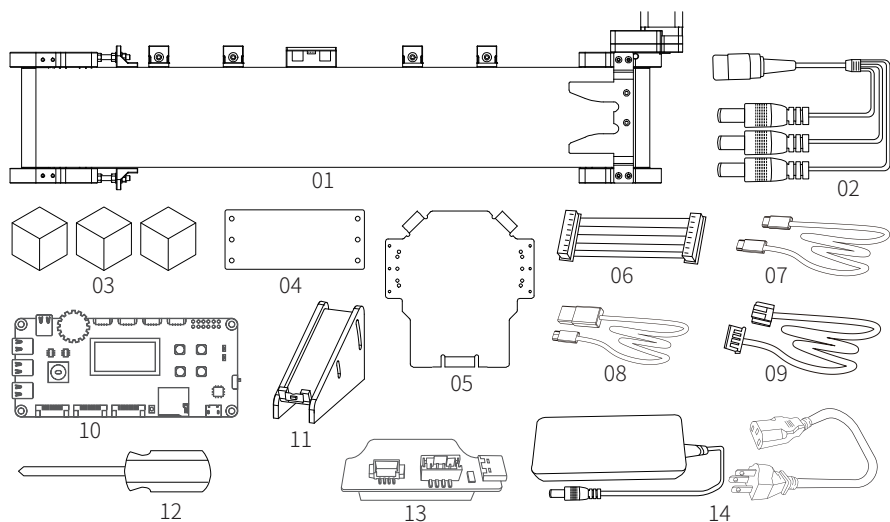
SHENZHEN UFACTORY CO.,LTD.

V 2 . 1

目录

1 Table of Contents	03
1.1Hardware	03
1.2Software	04
2System structure	04
3 Installation Tutorial	05
3.1Hardware Installation Tutorial	05
3.1.1Kit Installation Tutorial	05
3.1.2Wiring Tutorial	07
3.2Software Installation Tutorial	08
3.2.1uArm controller & uArm swift pro firmware writing tutorial	08
3.2.2 Firmware Recovery	11
3.3 uArm Conveyor Belt Example	12

1.Parts List



1.1 Hardware

01.Conveyor * 1

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

03.Target Object (Red & Green & Yellow Cube) 1

04.Connection Plate* 1

05.uArm Swift Pro stator *2

06.Slider Motor 8-Pin cable * 1

07.USB Type-C Cable * 2

08.USB Cable * 1

09.Seed Grove Sensor Cable * 3

10.Controller* 1

11.feeding Slide * 1

12.Screwdriver * 1

13.uArm 30P extendable base * 2

14.Power adapter*1

1.2 Software

01.Arduino IDE

<http://www.arduino.cc/>

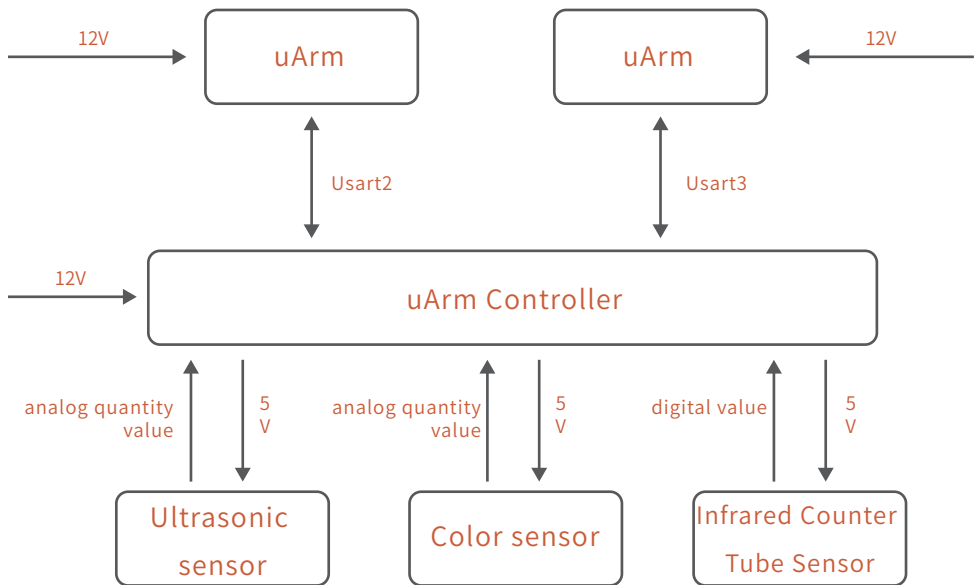
02.conveyor_belt.ino for Controller

<https://bit.ly/2YWITBZ>

03.uArmPro_V4.X.X.hex for uArm

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

2.System Structure

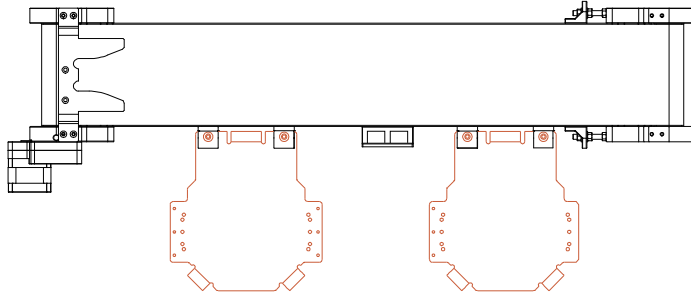


3 Installation Tutorial

3.1 Hardware installation tutorial

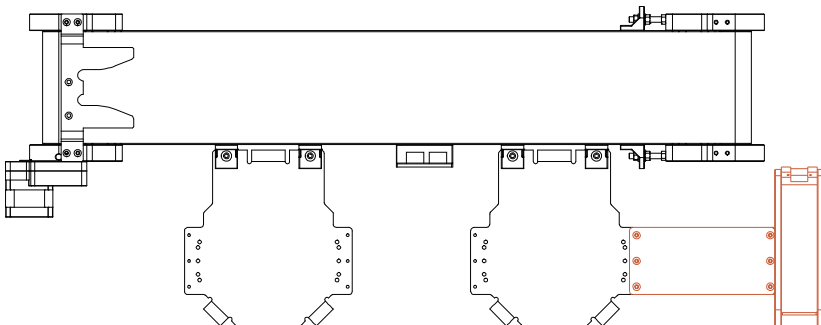
3.1.1 Kit Installation Tutorial

(1) Install the uArm stator



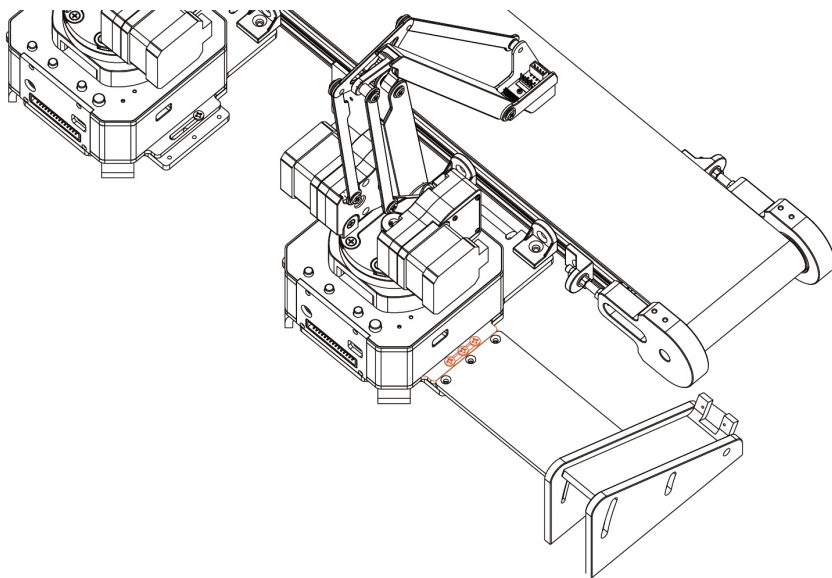
Fix the uArm stator on the conveyor belt.

(2) Install the feeding slide



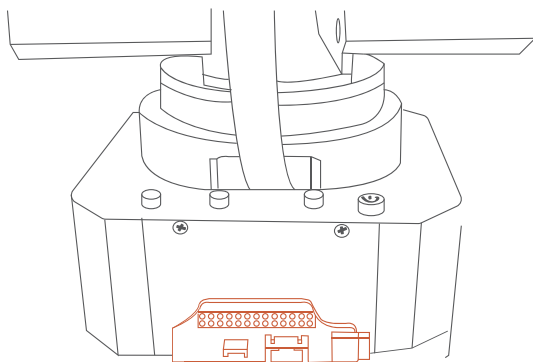
Connect the uArm Swift Pro stator to the material slide with the connection plate.

(3) Place the uArm Swift Pro



Place the Arm Swift Pro that with burnt firmware on the stator, it should stay close to the stator.

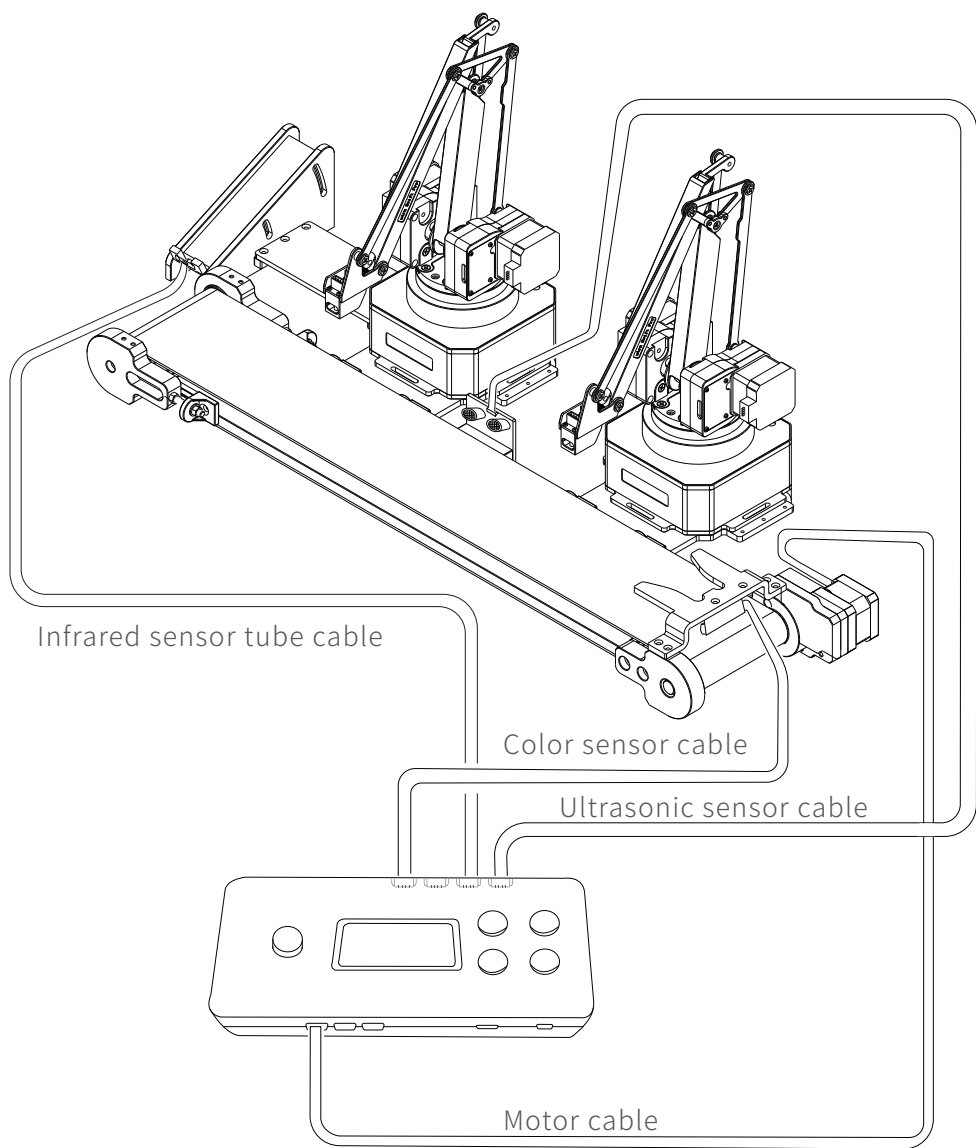
(4) Install uArm 30P extendable base.



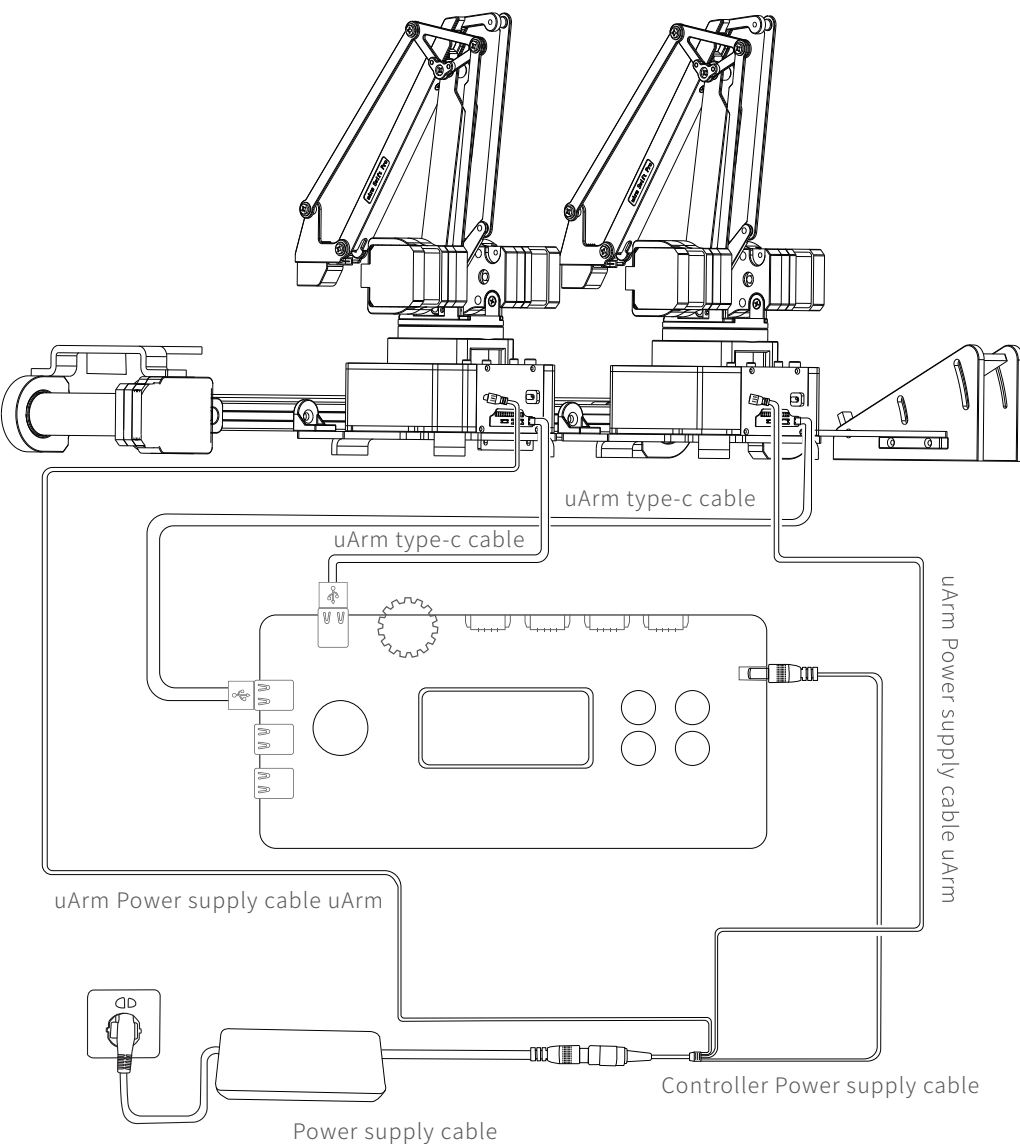
Install uArm 30P extendable base to the rear interface on uArm

3.1.2 Wiring Tutorial

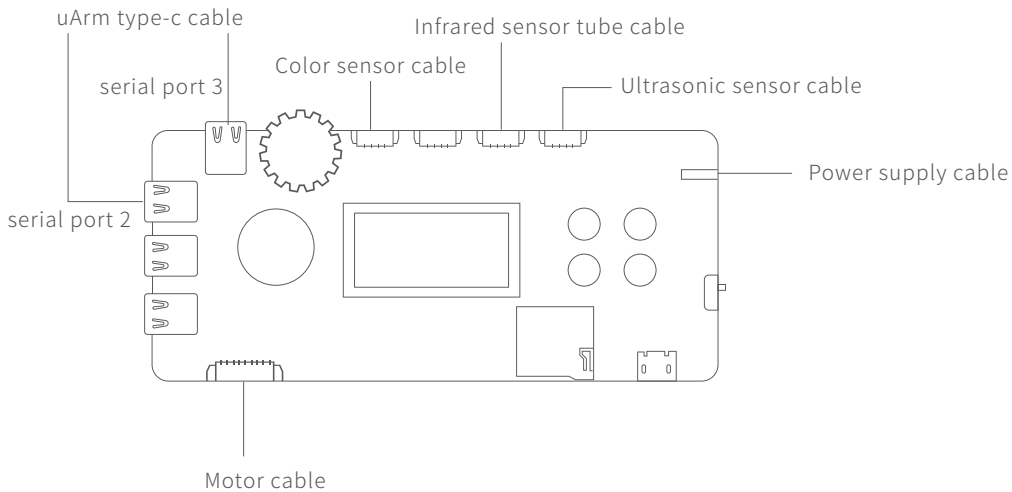
(1) Sensor cable connection



(2) Connect uArm Swift Pro Communication interface and the power supply cable



(3) uArm Controller wiring diagram



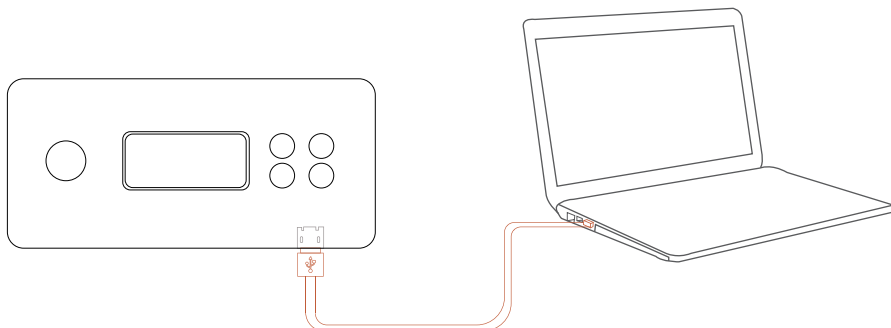
3.2 Software Installation Tutorial

3.2.1uarm controller & uArm swift pro Firmware Writing Tutorial

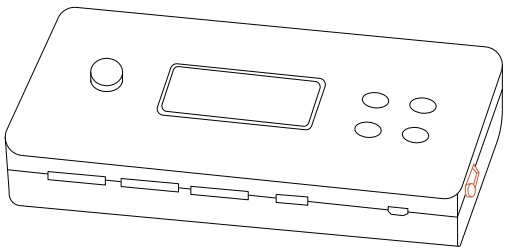
uArm Controller firmware writing

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.

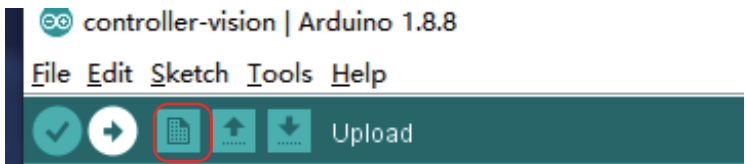
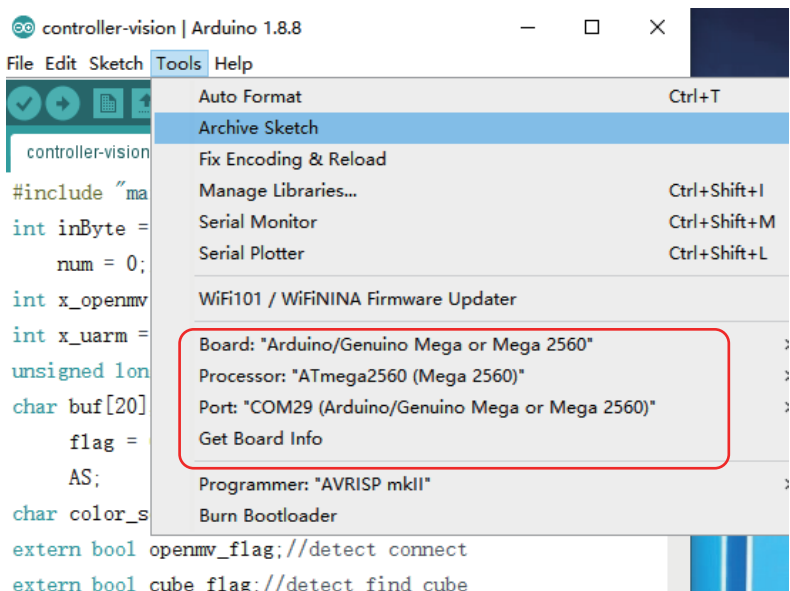
- (1) Download firmware: conveyor_belt.ino
<https://bit.ly/2YWITBZ> for Arduino Mega 2560
- (2) Connect the uArm Controller to the computer with a USB cable



(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.

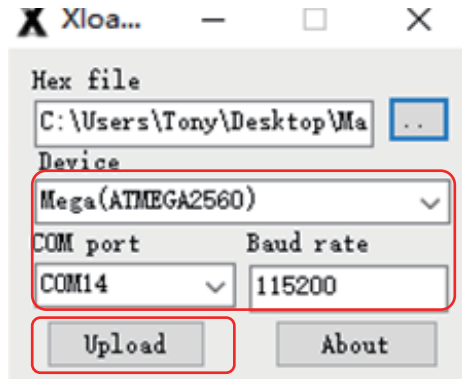


uArm Swift Pro firmware Writing tutorial

Connect uArm Swift Pro to your computer. Open XLoader (xloader.russeotto.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.

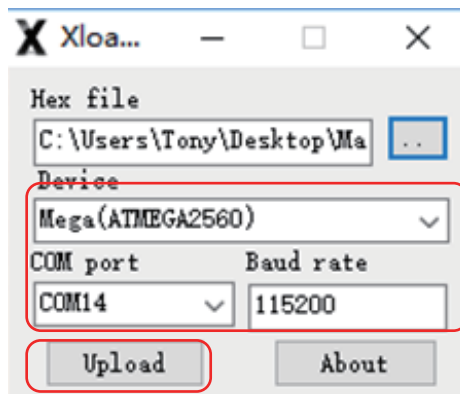


3.2.2 Firmware Recovery

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.russeotto.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.



3.4 uArm Conveyor Belt Demo

When the infrared sensor detects the cube, the uArm grabs the cube according to the predetermined path and places the cube on the conveyor belt. The conveyor belt activates and accelerates when passing by the ultrasonic sensor, and the conveyor stops when the color cube reaches the end of the conveyor belt. uArm then grabs the color cube according to the color recognized by the sensor and place it in a different position.

Video Demo:

<https://youtu.be/xjtnRsCymzc>

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 three-color values.

```
void convert_rgb_r()  
{  
    R_RED    = 178;  
    R_GREEN  = 61;  
    R_BLUE   = 56;  
}  
void convert_rgb_y()  
{  
    Y_RED    = 255;  
    Y_GREEN  = 255;  
    Y_BLUE   = 120;  
}  
void convert_rgb_g()  
{  
    G_RED    = 157;  
    G_GREEN  = 233;  
    G_BLUE   = 108;  
}
```



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