Preface

About SunFounder

SunFounder is a technology company focused on Raspberry Pi and Arduino open source community development. Committed to the promotion of open source culture, we strive to bring the fun of electronics making to people all around the world and enable everyone to be a maker. Our products include learning kits, development boards, robots, sensor modules and development tools. In addition to high quality products, SunFounder also offers video tutorials to help you build your own project. If you have interest in open source or making something cool, welcome to join us! Visit www.sunfounder.com for more!

About Crawling Quadruped Robot Kit V2.0

This learning kit is based on the popular open source electronics platform Arduino. It is different from most popular kits on the market. With this kit, you cannot only learn how to use Arduino, servo and wireless module but also DIY a cool crawling quadruped robot with wireless remote control. We provide a very detailed manual (code explanation is provided for high-level learning) and technical support for free. And we are ready to answer your questions at any time. If you want to DIY your own robot, this kit is a good start.

Apart from the current functions the robot supports, you may also explore much more possibilities on it! Just think big and try to bring it into reality. If you find it hard to implement, welcome to share your thoughts with our engineers!

Notes

- There are some 3D models showed in this manual, whose color and shape may be different from real objects.
- Please follow instructions in the manual in case of damage to the components.
- These components may be fragile, so please do not attempt any operations that may hurt them.
- We provide free technical support, but man-made damages to parts are excluded.

Free Support

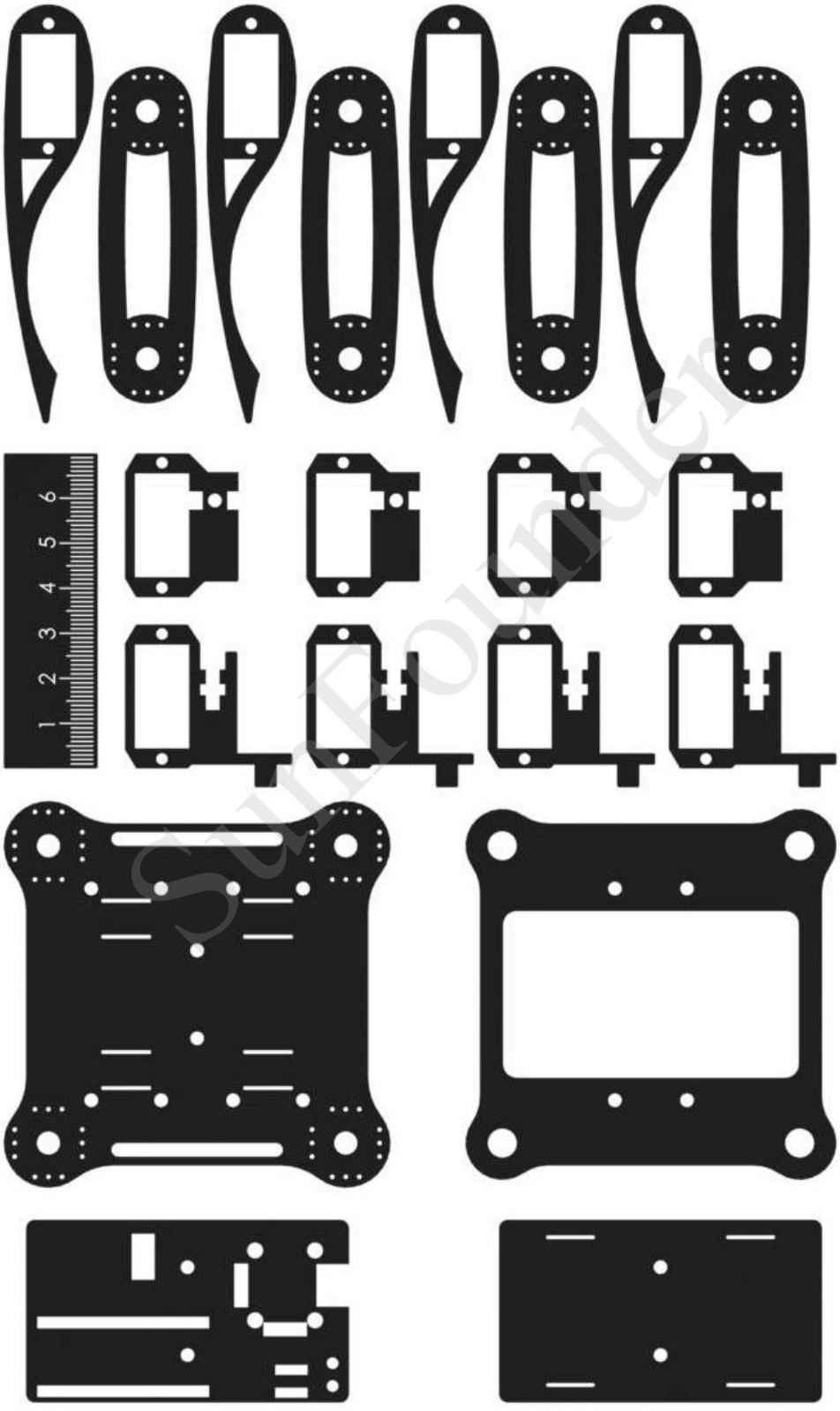
If you have any **TECHNICAL questions**, add a topic under **FORUM** section on our website and we'll reply as soon as possible.

For **NON-TECH questions** like order and shipment issues, please **send an email to service@sunfounder.com**. You're also welcomed to share your projects on FORUM.

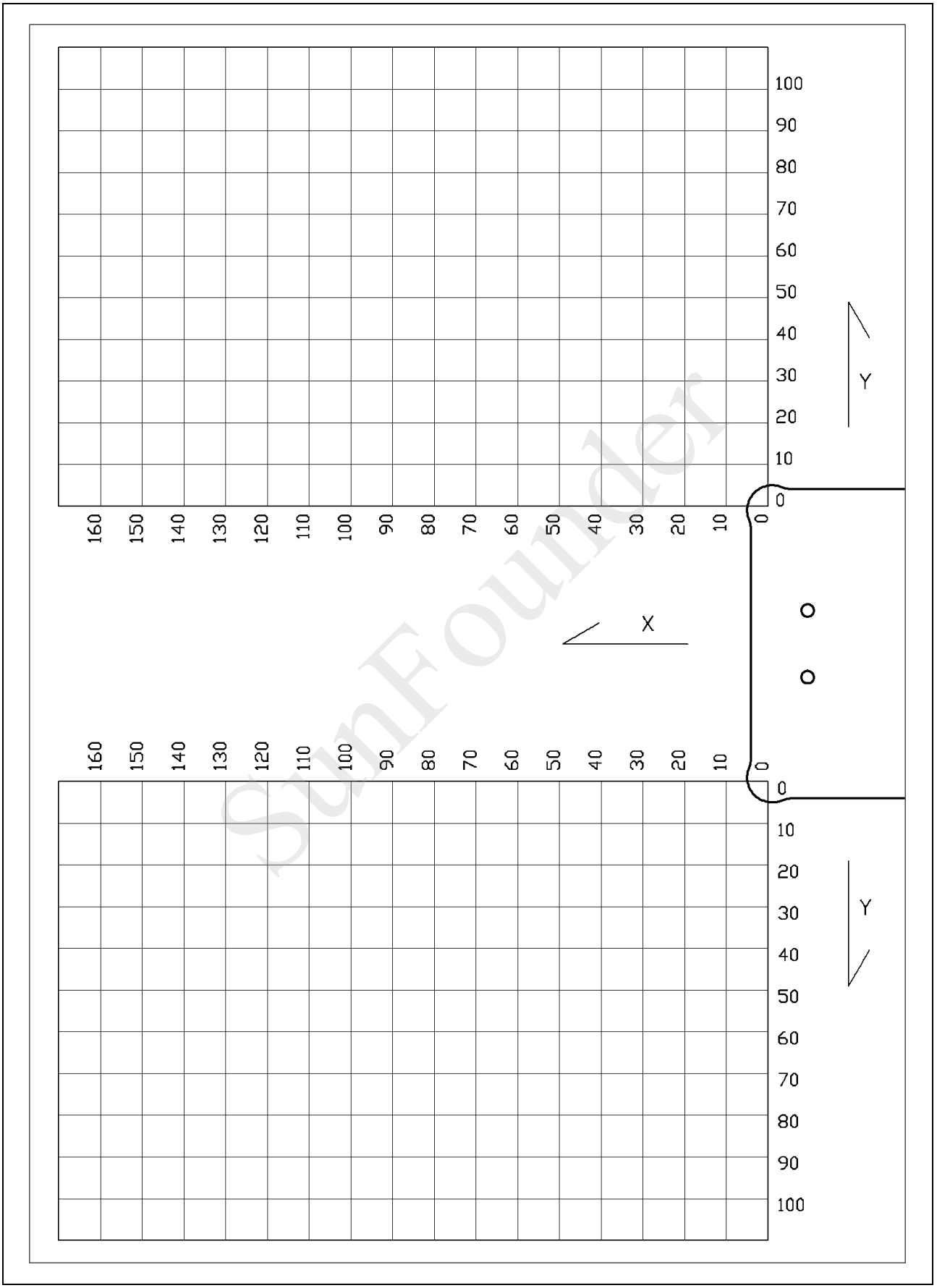
Contents

Components List

Acrylic Plates



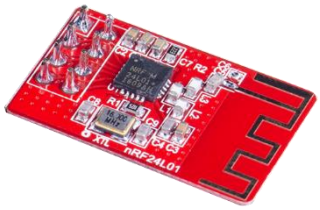
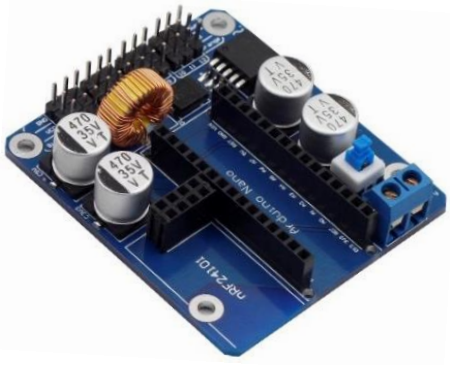
Calibration Chart

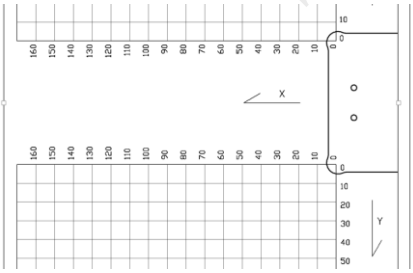


Threaded Fasteners

Accessory	Name	Quantity
	M1.4*6 Self-tapping Screw	60
	M2*10 Screw	26
	M2 Nut	26
	M3*12 Screw	6
	M3*8 Screw	14
	M3*6 Screw	6
	M3*18 Countersunk Screw	4
	M3*12 Countersunk Screw	4
	M3*36 Copper Standoff	6
	M3*8 Copper Standoff	6
	M3 Nut	20

Electronic Accessories

Accessory	Name	Quantity
	nRF24l01 Module	2
	SunFounder Nano Board	2
	SunFounder Mobile Robot Remote Controller	1
	SunFounder Servo Control Board	1

	2x18650 Battery Holder	2
	SunFounder Micro servo	12
	USB Data Cable	1
	Wire Harness Tube (80cm)	1
	Ribbon (50cm) (Divided into four equal parts)	1
	Calibration Chart	1

Tools

Accessory	Name	Quantity

	Screwdriver	1

Self-provided Components

The following components are not included in this kit.

Accessory	Name	Quantity
	18650 Rechargeable Li-ion Battery (3.7V)	4

Note:

1. You are recommend to use 18650 batteries **without** a protective board. Otherwise, the car may be cut power and stop running because of the overcurrent protection of the protective board.
2. For unprotected batteries, please purchase those with the anode bulged out (as shown below), so that it can ensure the well connection with the battery holder.

3. If you only have batteries with the protected board, it's ok. To use it, you can remove the board on the cathode. But you should operate with care to avoid any hurt; Since the metal shell is connected to the cathode for most batteries, please don't use screw driver or other metal objects to remove, or the battery may get shorted and be burnt if the shell is connected to the positive pole accidentally.

1. Introduction

The SunFounder Crawling Quadruped Robot kit is a great learning tool for Arduino and robotics enthusiasts. With the knowledge in mechanic structure and electronic design, you can take it as a functional stepping stone into the amazing coding world!

This interesting "creature" is a four-leg mobile robot, and each leg has three joints driven by a servo. It is powered by two 18650 rechargeable Li-ion batteries, compatible with Arduino Nano V4.0 board and uses the SunFounder Nano board as control. In addition, a SunFounder Servo Control Board integrates battery, servo, SunFounder Nano board, and nRF24l01 wireless module together. This kit is equipped with a SunFounder Mobile Robot Remote Controller so that you can observe and control the robot remotely. Or you may control it by your computer (mouse or keyboard) after installing a sketch provided on your PC.

2. Getting Started

Note:

Before starting your own project, you must download the file ***Quadruped Crawling Robot Kit V2.0 for Arduino***.*zip* on our official website by visiting **LEARN** -> **Get Tutorials** -> **Quadruped Crawling Robot Kit V2.0 for Arduino** and unzip it.

Arduino

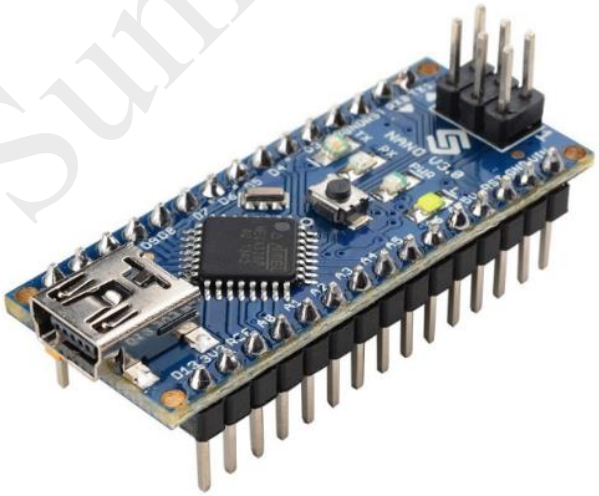
Description

Arduino is an open source platform that applies simple software and hardware. You can get it in a short even when you know little of it. It provides an integrated development environment (IDE) for code editing and compiling, compatible with multiple control boards. So you can just download the Arduino IDE, upload the sketches (i.e. the code files) to the board, and then you can see experimental phenomena. For more information, refer to http://www.arduino.cc.

Arduino Board – SunFounder Compatible

Arduino senses the environment by receiving inputs from many sensors, and affects its surroundings by controlling lights, motors, and other actuators.

In this kit, SunFounder Nano board is used.



Install Arduino IDE

The code in this kit is written based on Arduino, so you need to install the IDE first. Skip it if you have done this.

Now go to the arduino.cc website and click **DOWNLOAD**. On the page, check the software

list on the right side under **Download the Arduino Software**.

Note: the IDE version shown in the figure is Arduino 1.8.1, which may not be the latest version now. Please go to the Arduino page to get the latest IDE.

Find the one that suits your operation system and click to download. There are two versions of Arduino for Windows: Installer or ZIP file. You're recommended to download the former. Just download the package, and run the executable file to start installation. It will download the driver needed to run Arduino IDE. After downloading, follow the prompts to install. For the details of installing steps, you can refer to the guide on **Learning**->**Getting Started with Arduino**, scroll down and see **Install the Arduino Software**.

After installing, you will see Arduino icon on your desk and double click to open it.

Install the Driver

If the driver is not installed, the Nano board will not be able to be recognized by your computer. Therefore, before using it, please install appropriate driver.

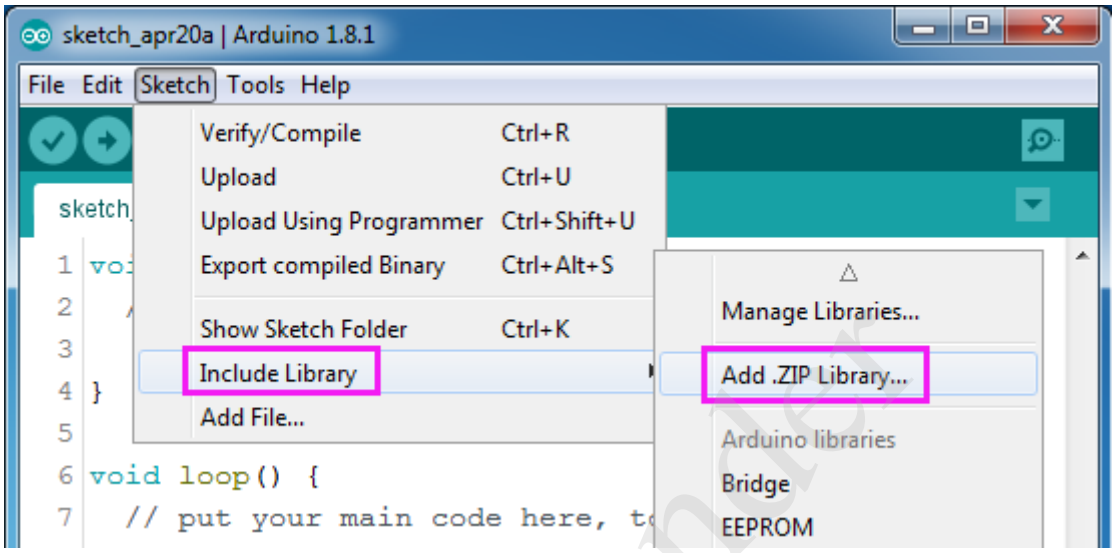
For Windows users, run ***PL2303_Prolific_DriverInstaller_v1160***

For Mac users, refer to the folder ***PL2303_MacOSX_1_6_1_20160309***

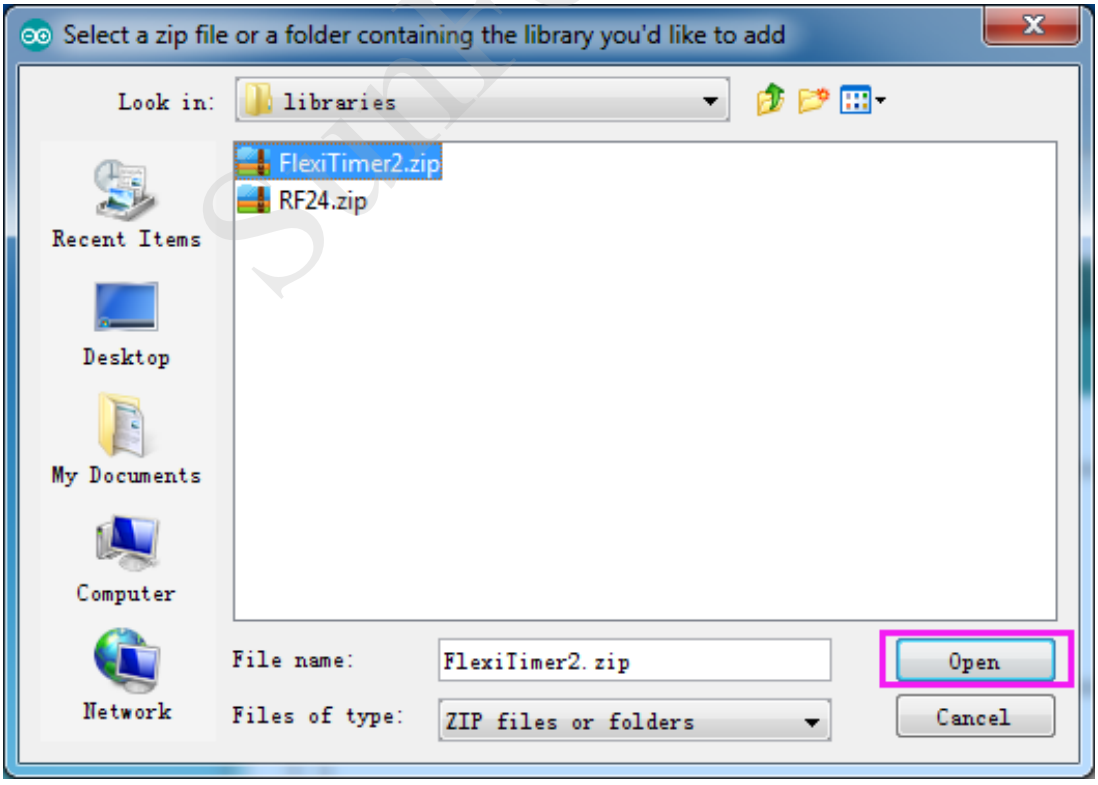
Add Libraries

1. Libraries are a collection of code that makes it easy for you to connect to a sensor, display, module, etc. In this kit, you will need to add two libraries to the Arduino libraries folder: *FlexiTimer2 and RF24*.

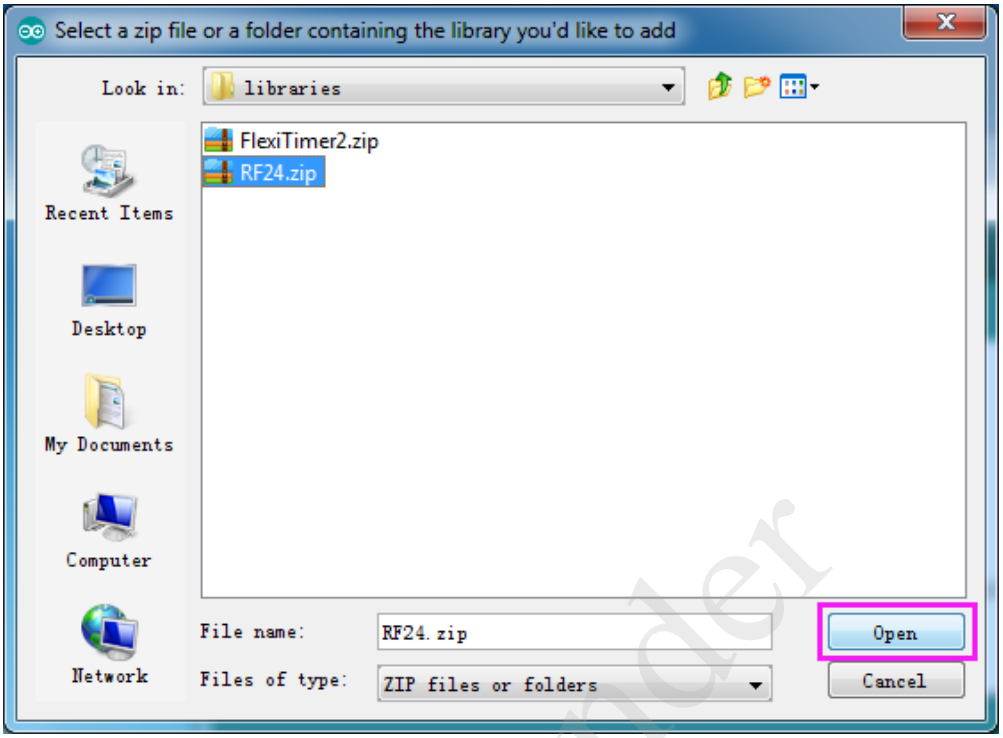
1) Select **Sketch** -> **Import Library** -> **Add Library**.



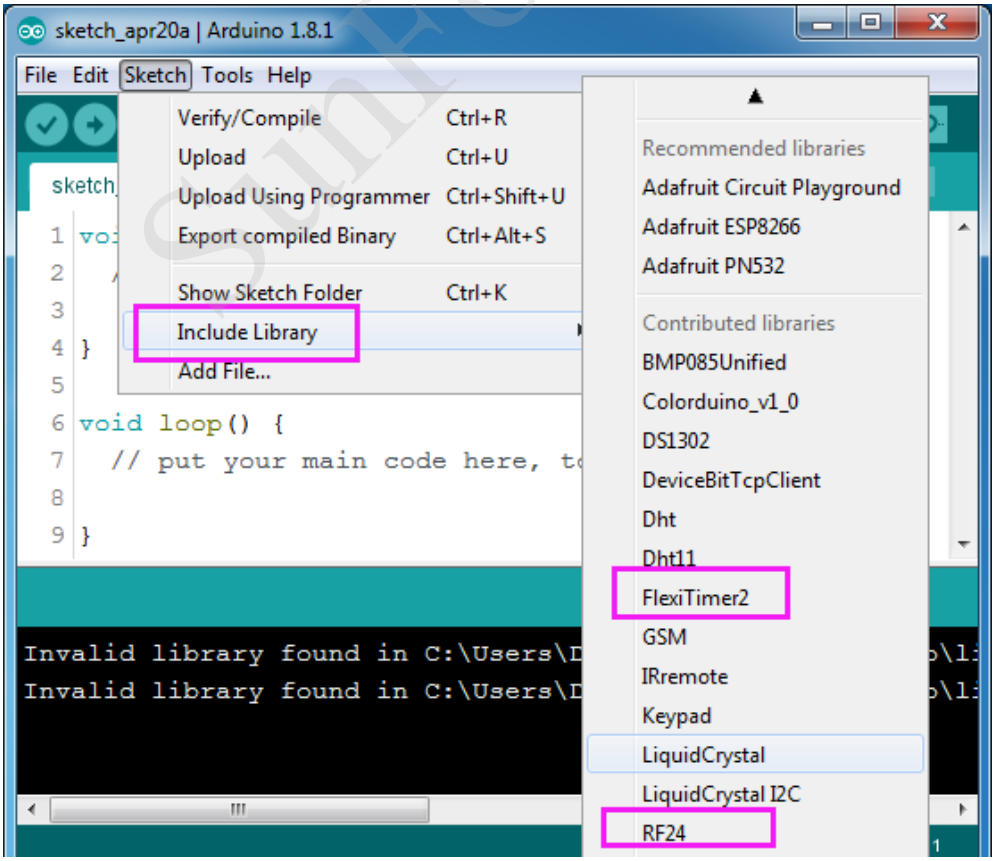
2) Find the *FlexiTimer2.zip* library under the Quadruped Crawling Robot Kit V2.0 for Arduino\libraries. Click **Open**.



3) Import the *RF24.zip* library from the *libraries* folder in the same way.



4) Here you should see the library added to your *libraries*. Click **Sketch**-> **Include Library** and the libraries just imported now appears on the list.

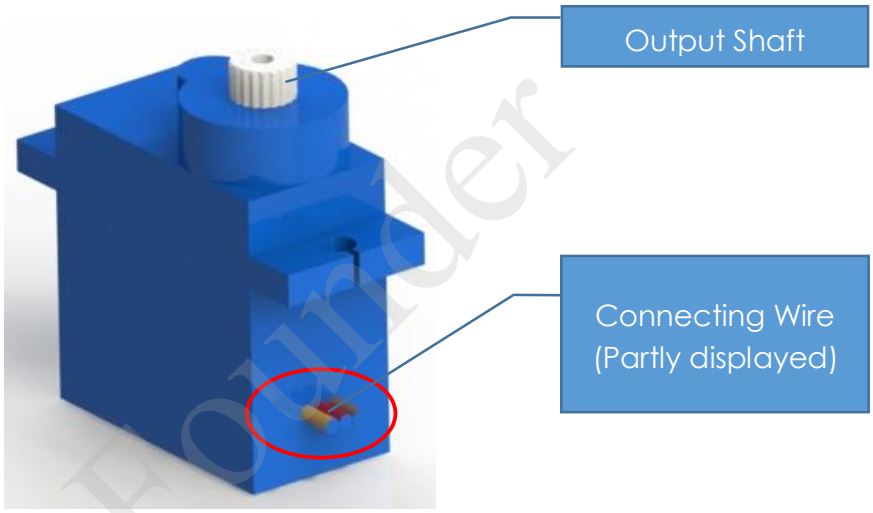


Servo

Description

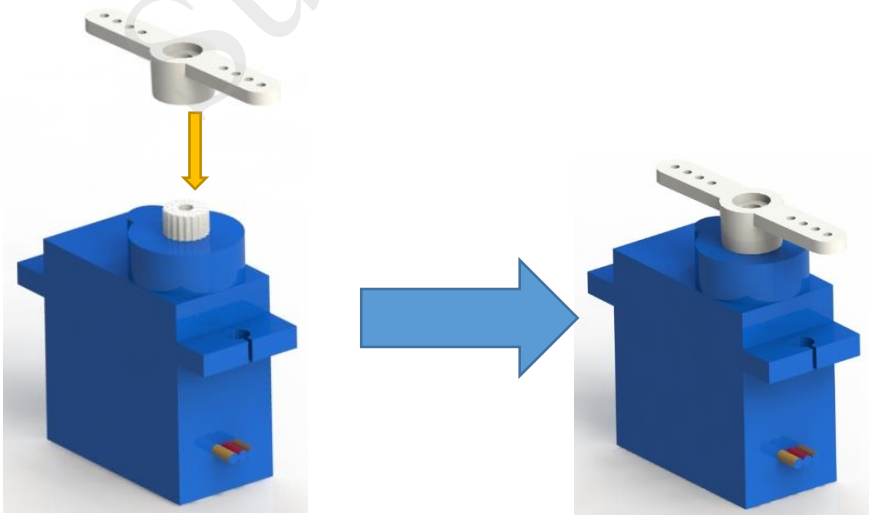
Servo is a set of automatic control system composed of DC motors, reduction gear set, sensors and control circuits. The output shaft can be rotated to a certain angle by sending signals. The servo can only rotate in a certain range, for example, 180°. It cannot rotate any circles like the DC motor. The servo enables you to easily rotate an object in a certain angle, so it is widely used in model planes and robot joints.

In this kit, twelve 9 micro servos are used to drive the joints of the robot.



Servo Test

1) Find the rocker arm packaged with the servo, and mount it onto the servo.

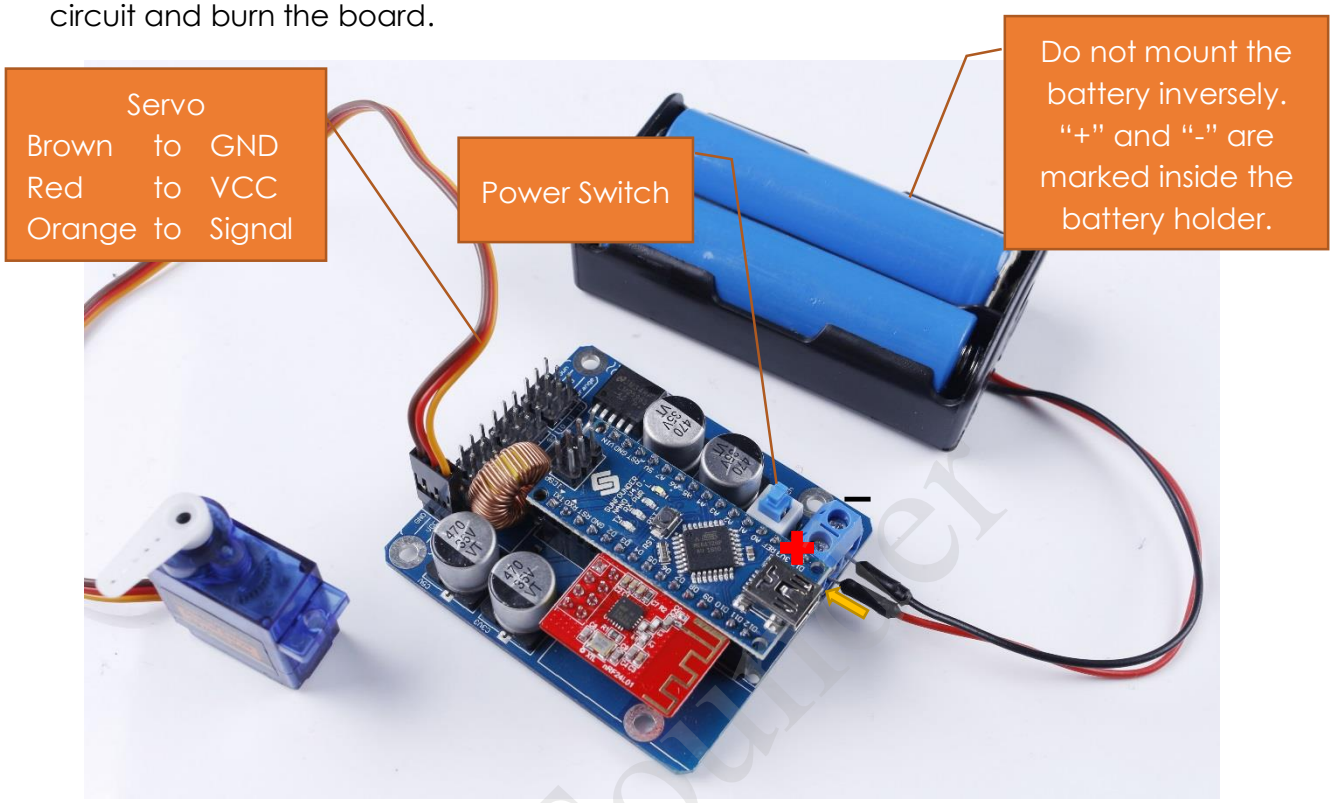


2) Connect the battery holder with the servo control board and the servo, then insert two batteries into the battery holder correctly.

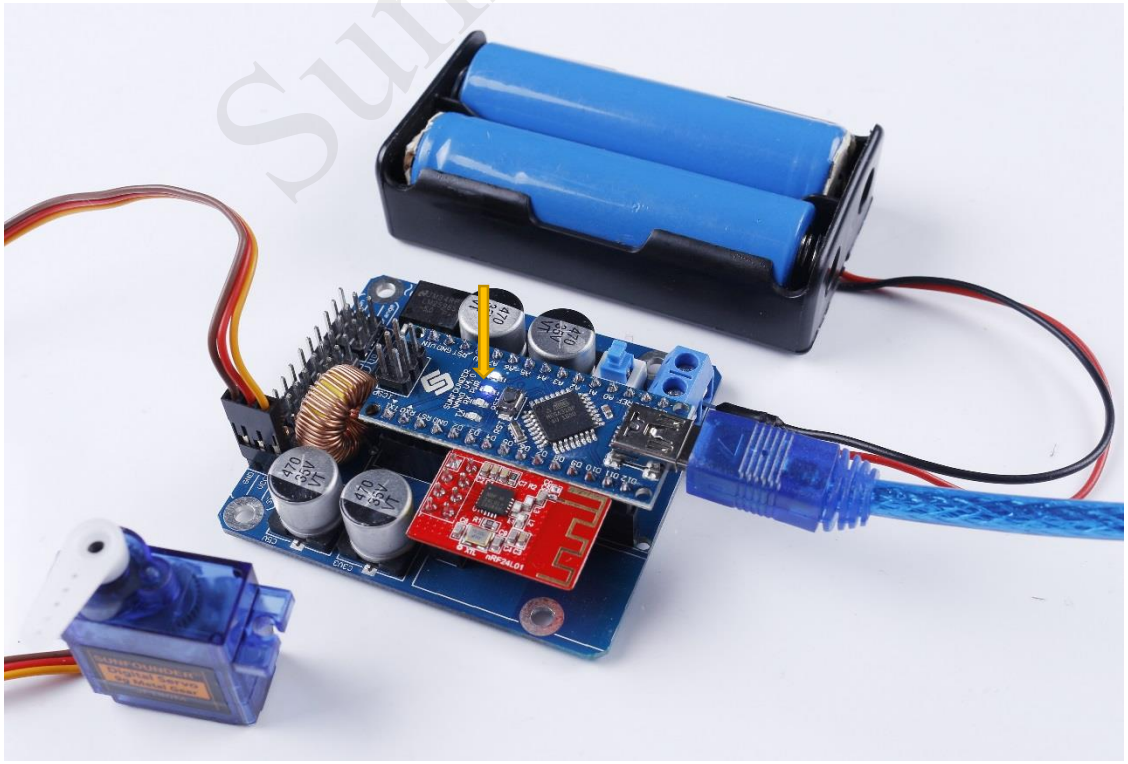
   Plug the servo to the pins marked with 1 on the SunFounder Servo Control Board. The

name of the pins and the color of the wires are marked on two sides of all the pins.

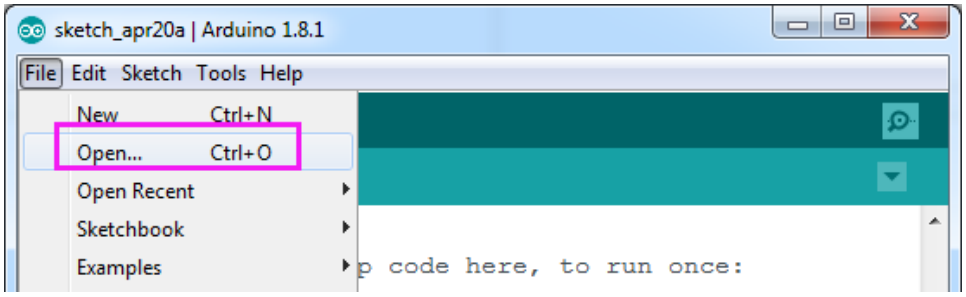
Connect to the red wire of the battery holder to "+" of the power switch, the black wire to the "-". Pay attention not to connect them inversely; otherwise it may cause a short circuit and burn the board.



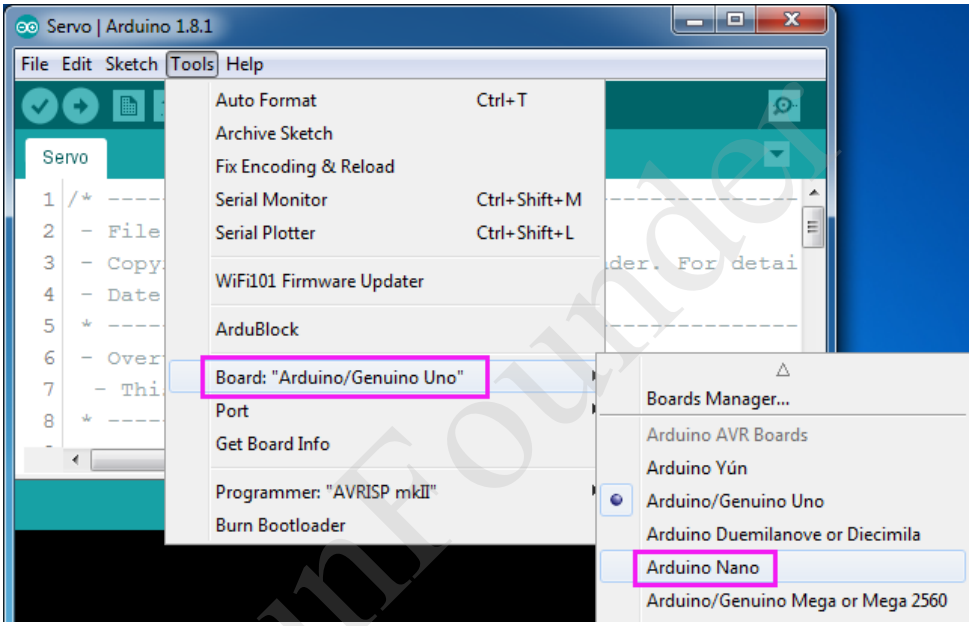
3) If everything looks good, connect the SunFounder Servo Control Board to your PC with a USB cable, then the blue LED on SunFounder Nano board will light up.



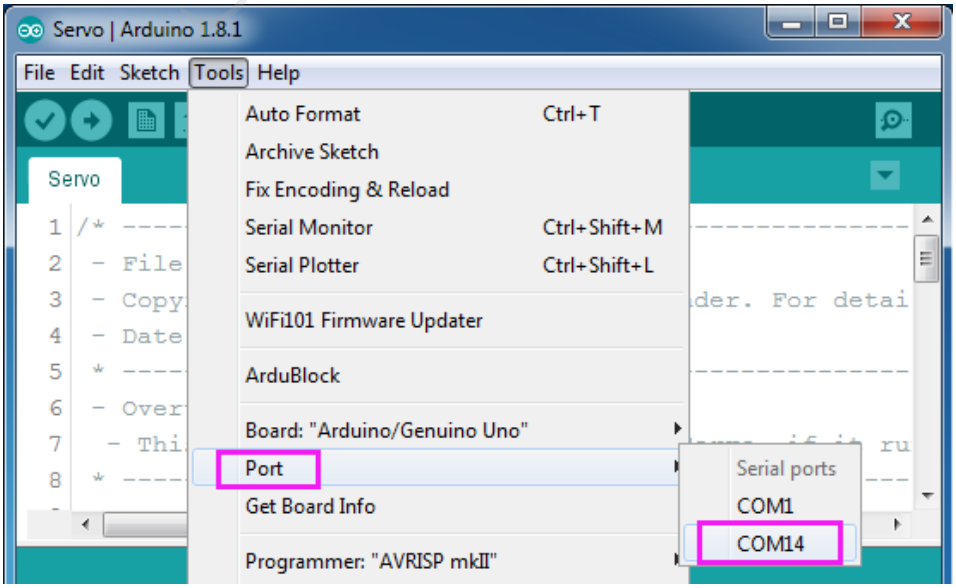
4) Open the Arduino Software (IDE), Select **File**->**Open**. On the pop-up window, go to the Quadruped Crawling Robot Kit V2.0 for Arduino\code\1.Servo\Servo directory and find *Servo.ino*. Click it to open.



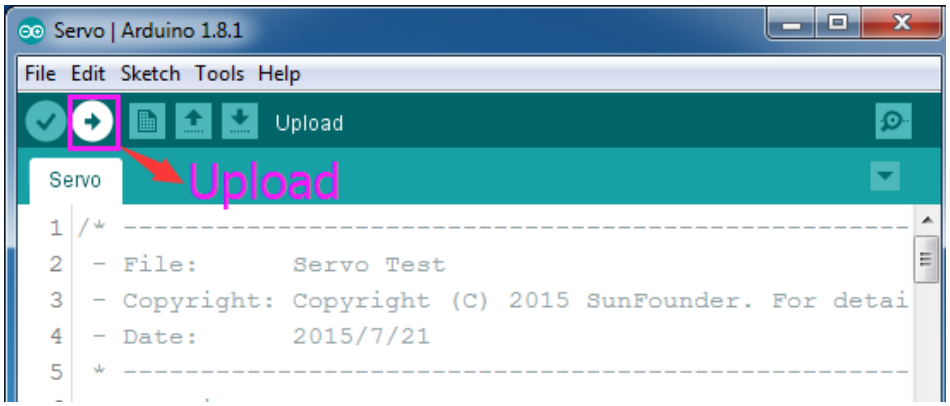
5) Select a board. Here we select **Arduino Nano**.



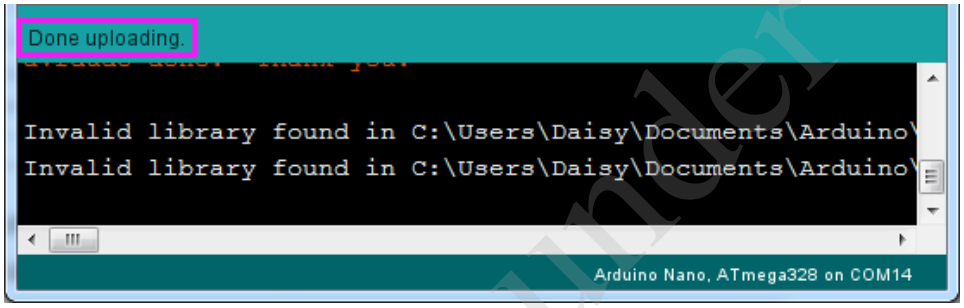
6) Go to **Tools** -> **Serial Port** to select the port. Your serial port may be different from what's shown here (right-click **My Computer** on desktop, click **Properties**>**Device Manager**>**Ports**).



7) Click the following button to upload.



8) Wait for a moment until the following information appears at the bottom of the window, which indicates it is uploaded successfully.



9) Remove the USB cable, switch on the power. Now, the rocker arm will sway continuously. So the servo test is done. Test all the 12 servos in the same way if necessary. If you find some jitters at 0 degree of a few servos, it's fine. The point here is to check whether the servo can work.

nRF24l01

Description

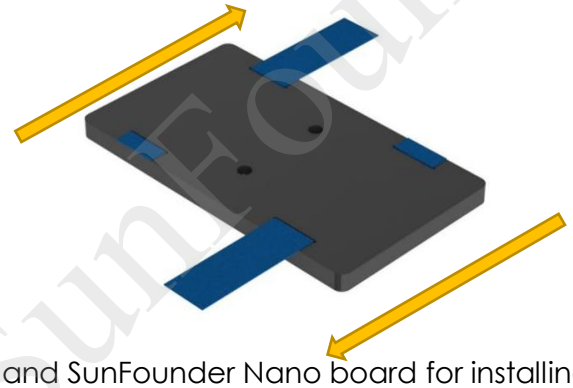
The nRF24L01 is a single-chip wireless transceiver chip, which is manufactured by NORDIC and works in the 2.4 GHz ~ 2.5 GHz ISM frequency band. The wireless transceiver includes frequency generator, enhanced ShockBurst mode controller, power amplifier, crystal oscillator, modulator and demodulator.

nRF24l01 Test

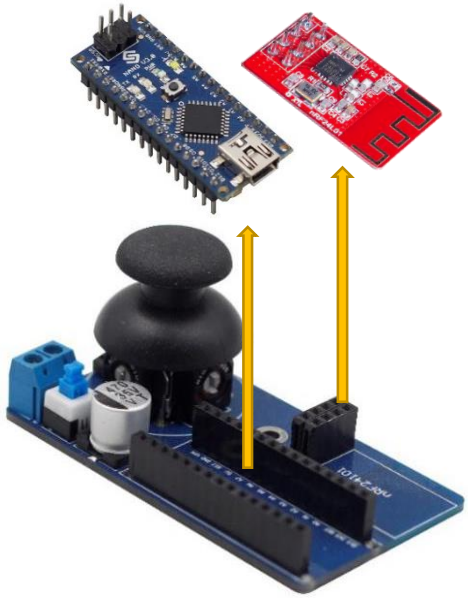
1. Cross the ribbon through the following acrylic plate.

   The ribbon enables you to easily remove the battery. Also you can skip this step.

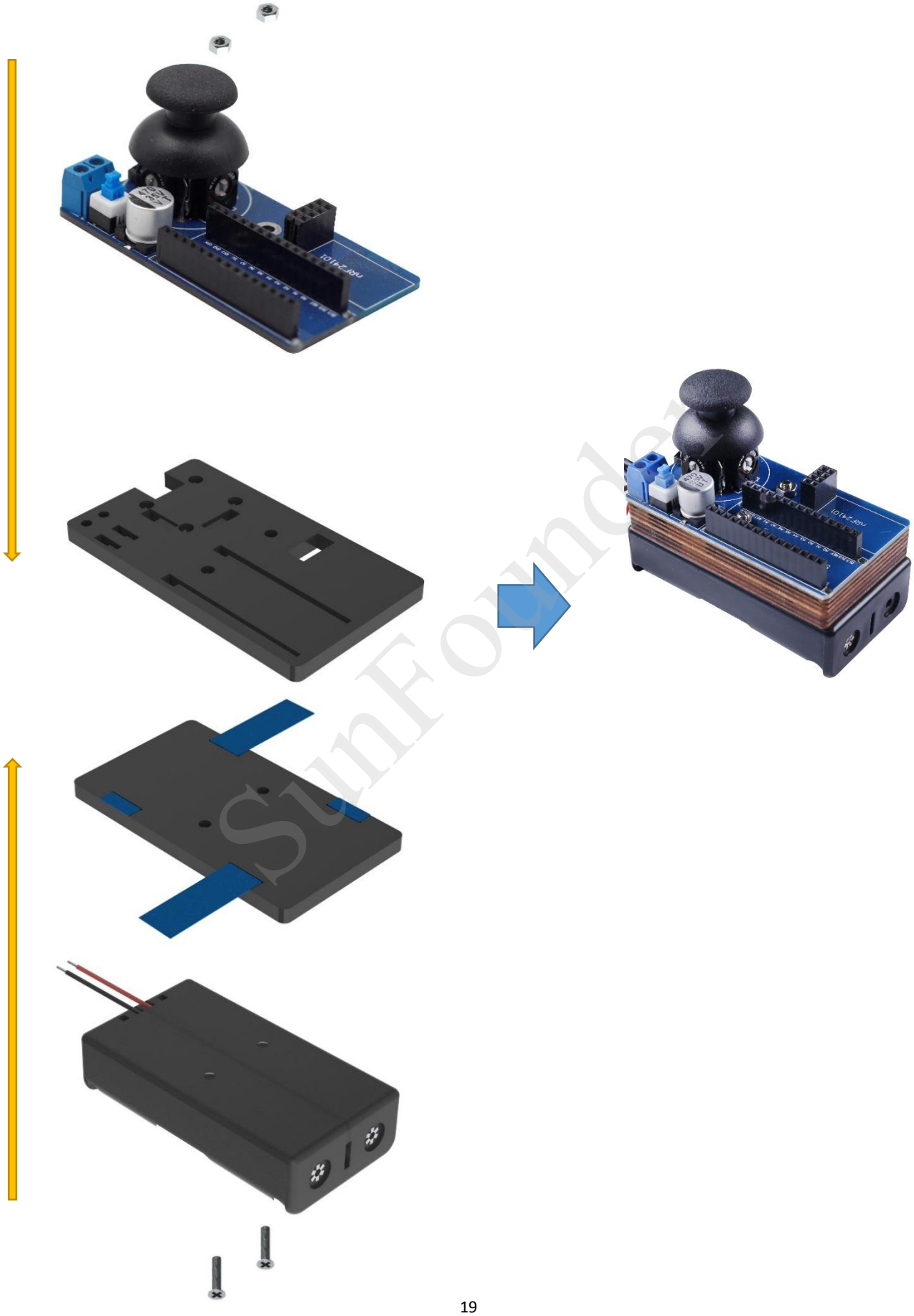
   Please note that one side of the ribbon is long and the other is short. The direction is shown as below.



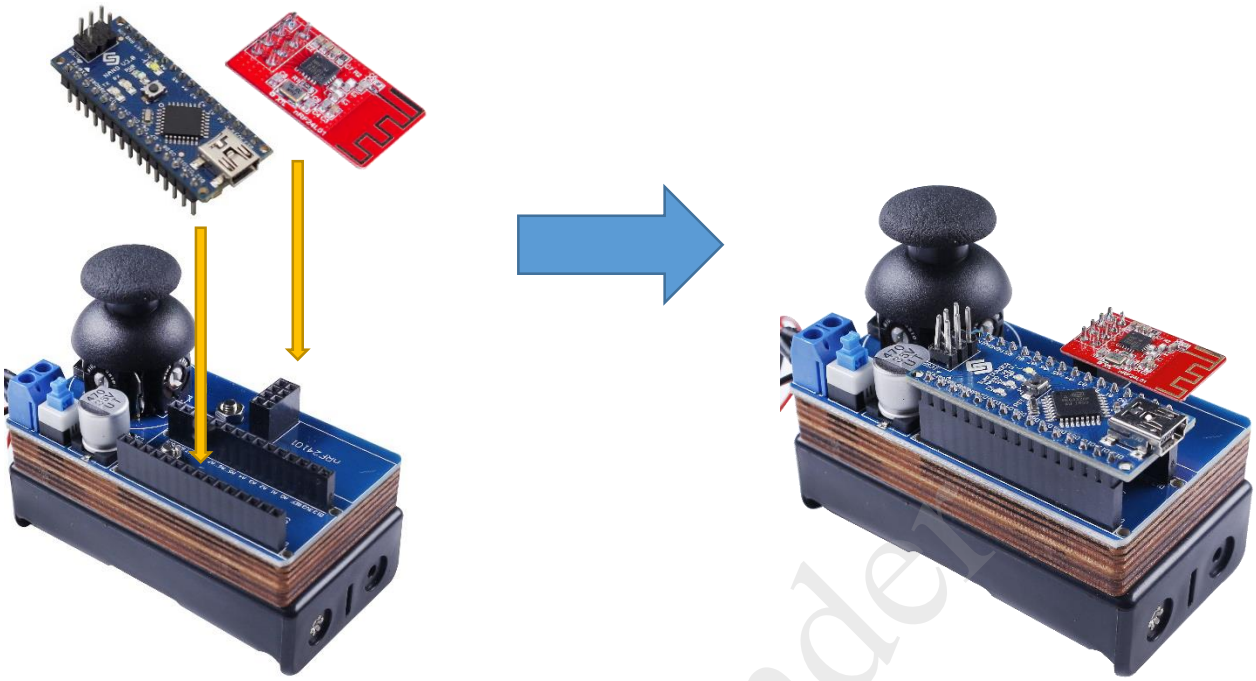
2. Remove the nRF24L01 and SunFounder Nano board for installing the battery holder later.



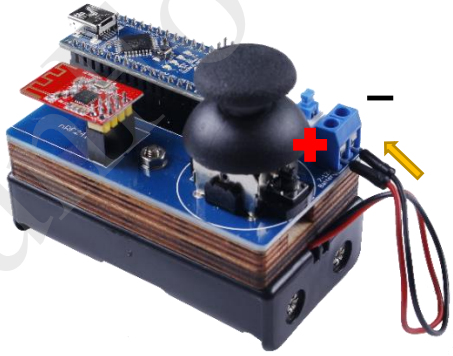
3. Connect the following components with **two M3*18 countersunk screws** and **two M3 nuts**.



4. Plug SunFounder Nano board and nRF24l01 module into the SunFounder Mobile Robot Remote Controller board.

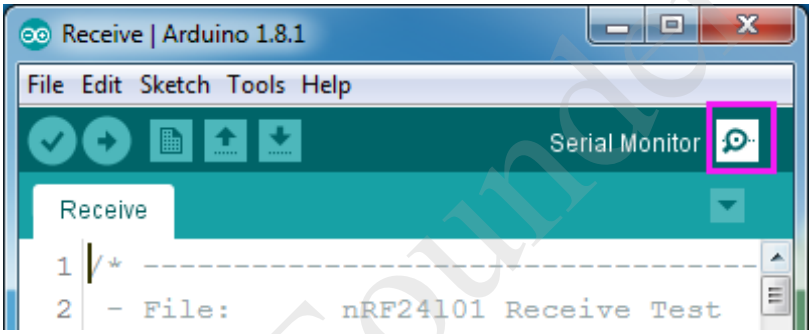


5. Connect the remote controller with the battery holder. Make sure the power source is wired correctly and the power is kept off.

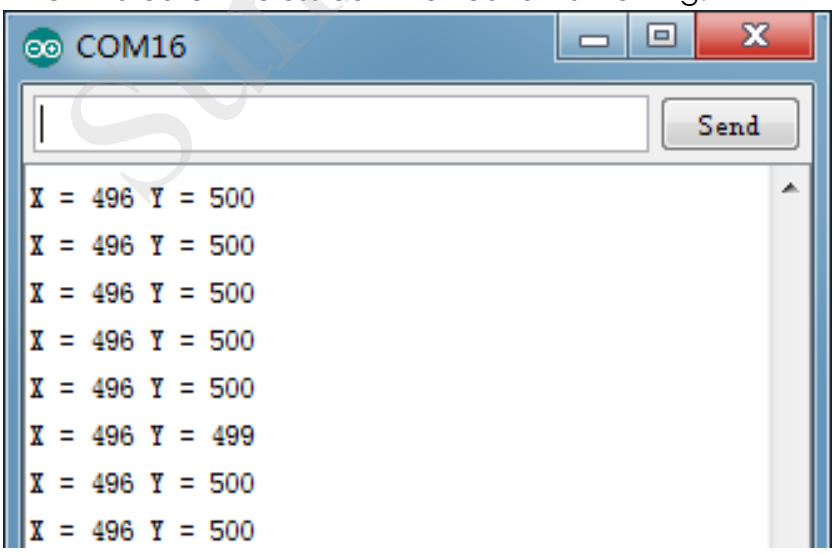


6. Mount two 18650 batteries (please pay attention not to mount it inversely).
   Check to see whether the battery holder and the battery are wired correctly.

7. Connect the **SunFounder Mobile Robot Remote Controller** to PC with a USB cable. If everything looks good, switch it on. Then the blue LED on SunFounder Nano board will light up.
8. Upload the ***Transmit.ino*** under **Quadruped Crawling Robot Kit V2.0 for Arduino***code**2.nrf24l01**Transmit* directory for the SunFounder Mobile Robot Remote Controller according to the method in the previous section. Then Remove the USB cable after upload, and keep the power on.
9. Connect the **SunFounder Servo Control Board** to your PC with a USB cable.
10. Upload the *Receive.ino* under the Quadruped Crawling Robot Kit V2.0 for Arduino*code**2.nrf24l01**Receive* directory for the SunFounder Servo Control Board according to the method in the previous section:
    Do not remove the USB cable after upload. Open **Serial Monitor**.



Now, you should see information received by the nFR24l01 module displayed on the Serial Monitor window. Turn the rocker arm. The value displayed on the **Serial Monitor** window will change, which indicate wireless communication is working.

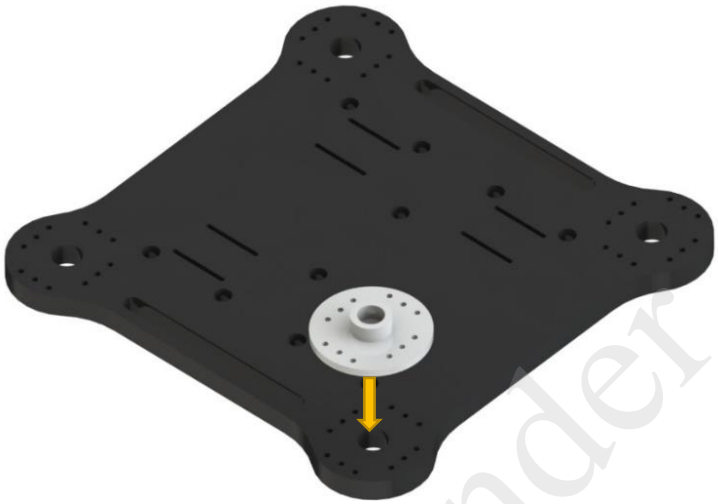


Remove the USB cable and turn off the power switch.

3. Assembly

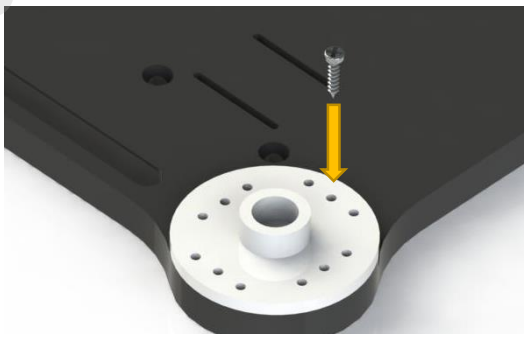
Upper Plate + Rocker Arm

1. Assemble the rocker arm (packaged with the servo) of the servo to the following acrylic plate.



2. Fasten the rocker arm to the acrylic plate.

3. Connect the two components with an **M1.4*6 self-tapping screw**.

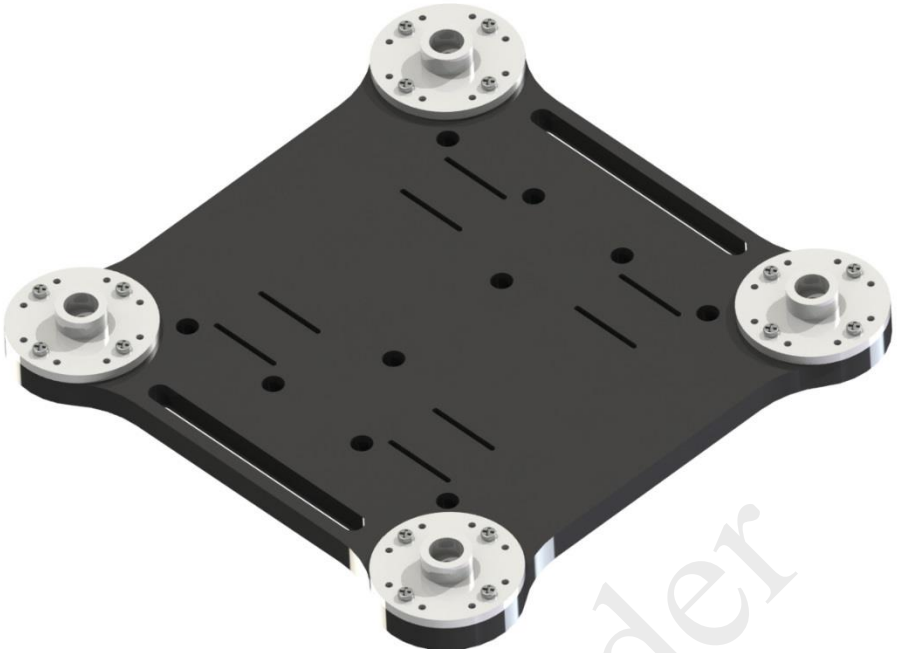


4. Tighten the screw.

   Do not over tighten it! Or the screw may be broken!

5. Tighten the other 7 screws.

6. Assemble the other 3 rocker arms.

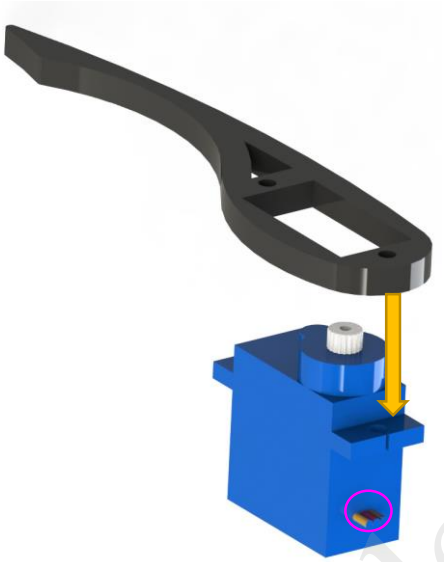


Thigh + Rocker Arm

In the same way, connect the rocker arms and the thighs with **M1.4*6 self-tapping screws**.

Crus + Servo

1. Assemble the servo to the crus. Pay attention to the direction of servo wires.



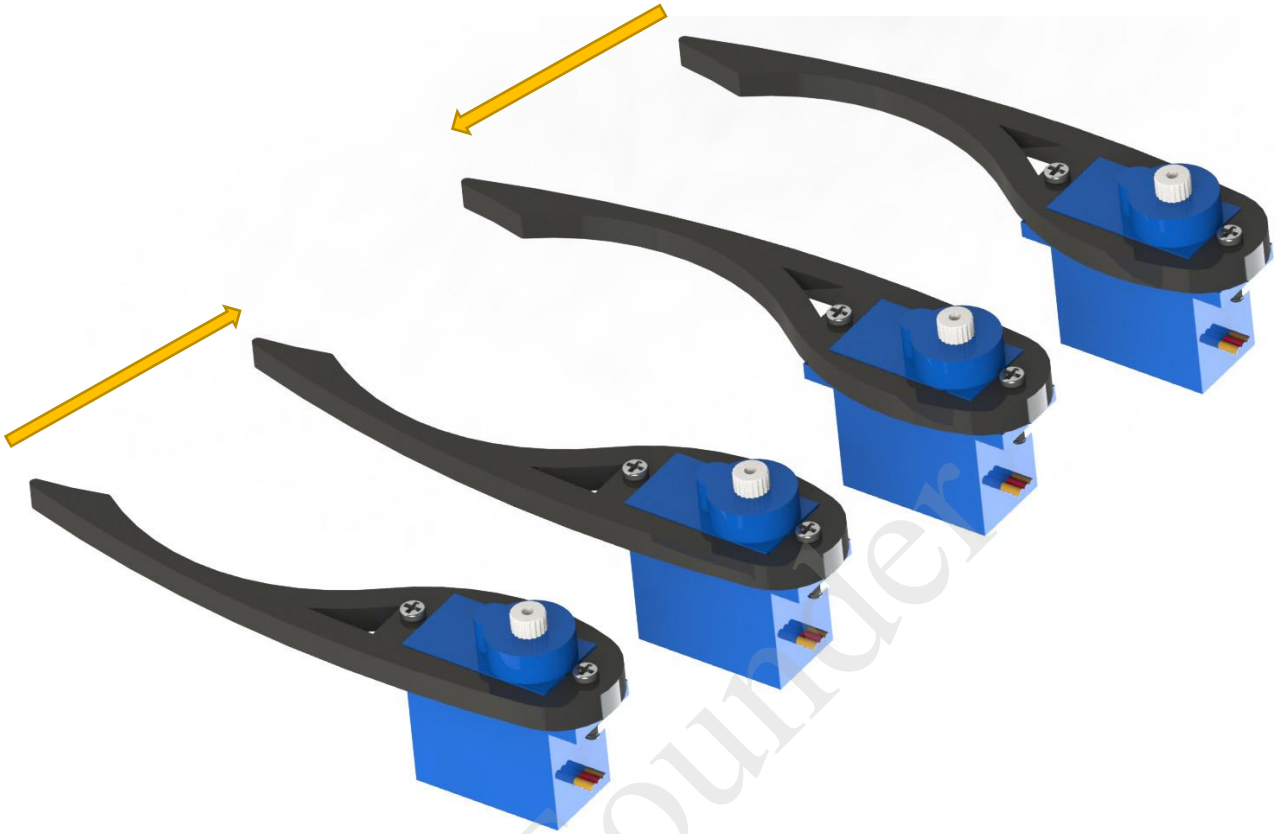
2. Fasten the servo to the crus.

3. Connect these two components with an **M2*10 screw** and an **M2 nut**.

4. Tighten the screw.

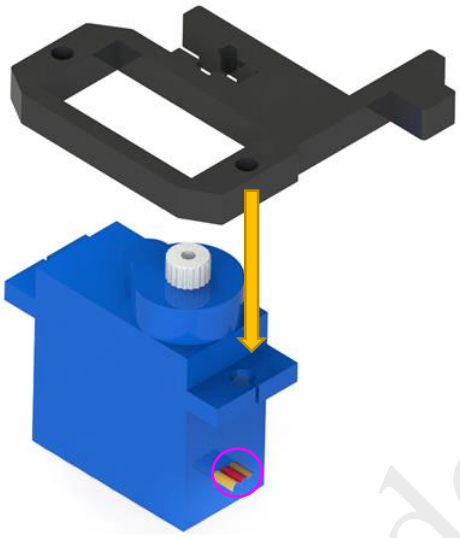
5. Tighten the other screw and nut.

6. Assemble the other 3 servos.
   Please pay attention to the direction. They should be **two opposite pairs**, as shown below:

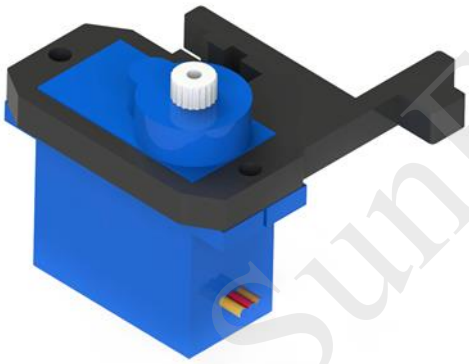


Thigh Joint 1 + Servo

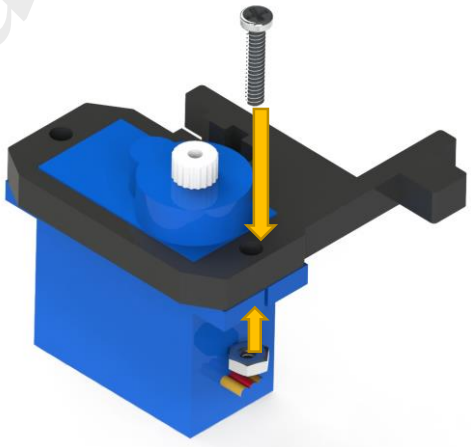
1. Assemble the servo to the following thigh joint 1 (the one next to the thigh).



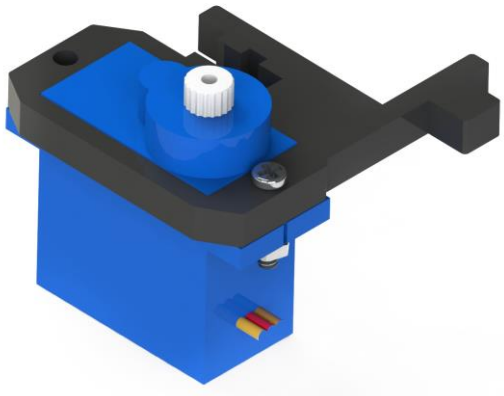
2. Fasten the servo to the thigh joint 1.



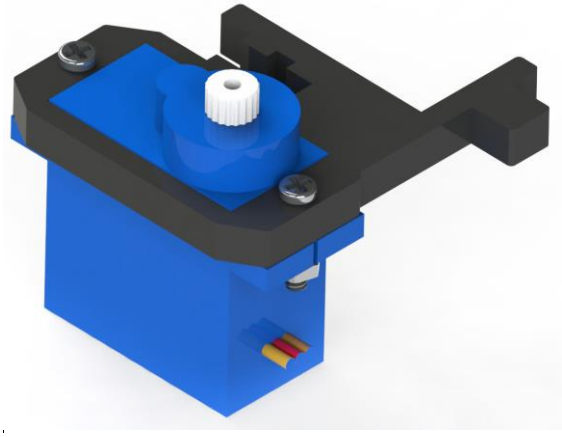
3. Connect the two components with an **M2*10 screw** and an **M2 nut**.



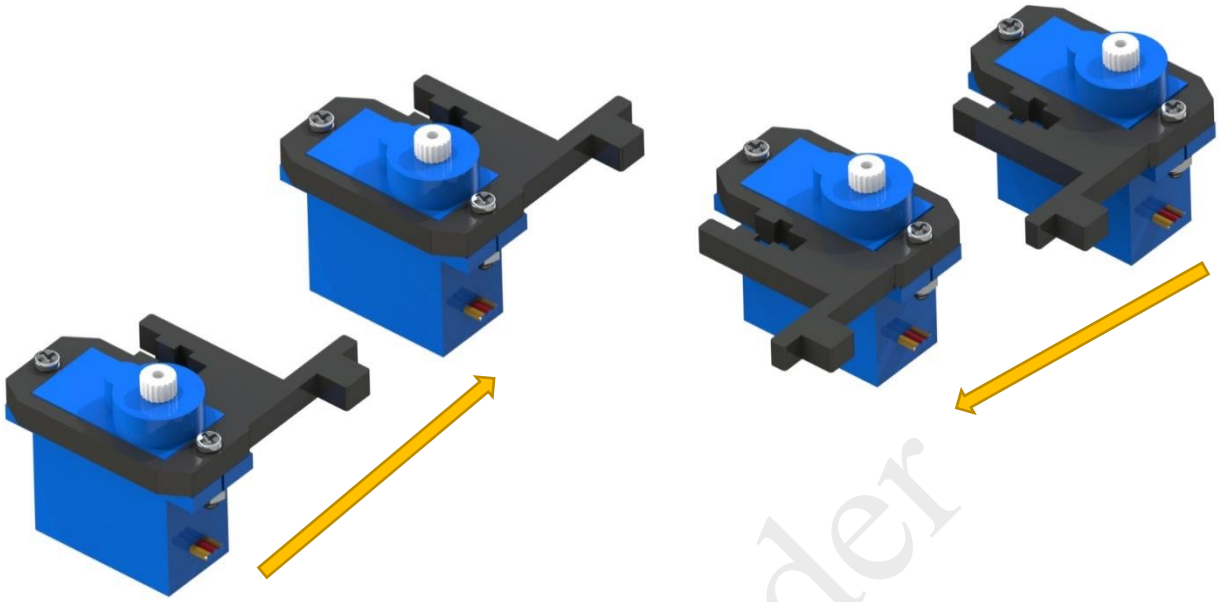
4. Tighten the screw.



5. Tighten the other screw and nut.

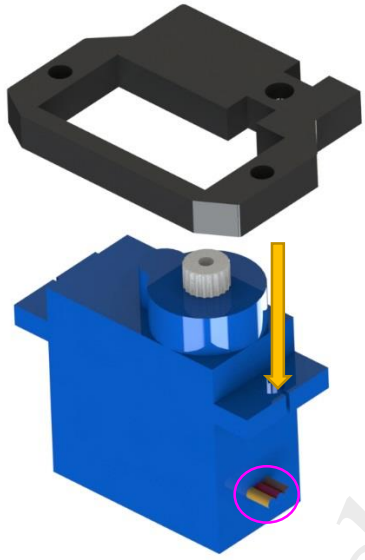


6. Assemble the other 3 joints.
   Please pay attention to the direction. They should be two opposite pairs, as shown below:



Thigh Joint 2 + Servo

1. Assemble the servo to the following thigh joint 2 (the one close to the upper body plate).

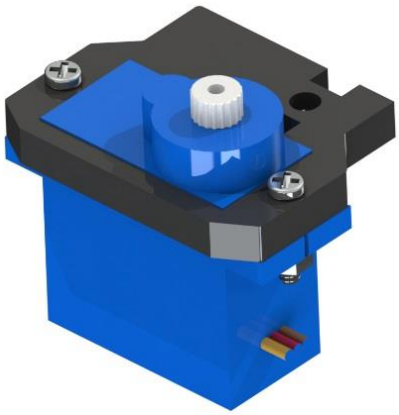


2. Fasten the servo to the thigh joint.

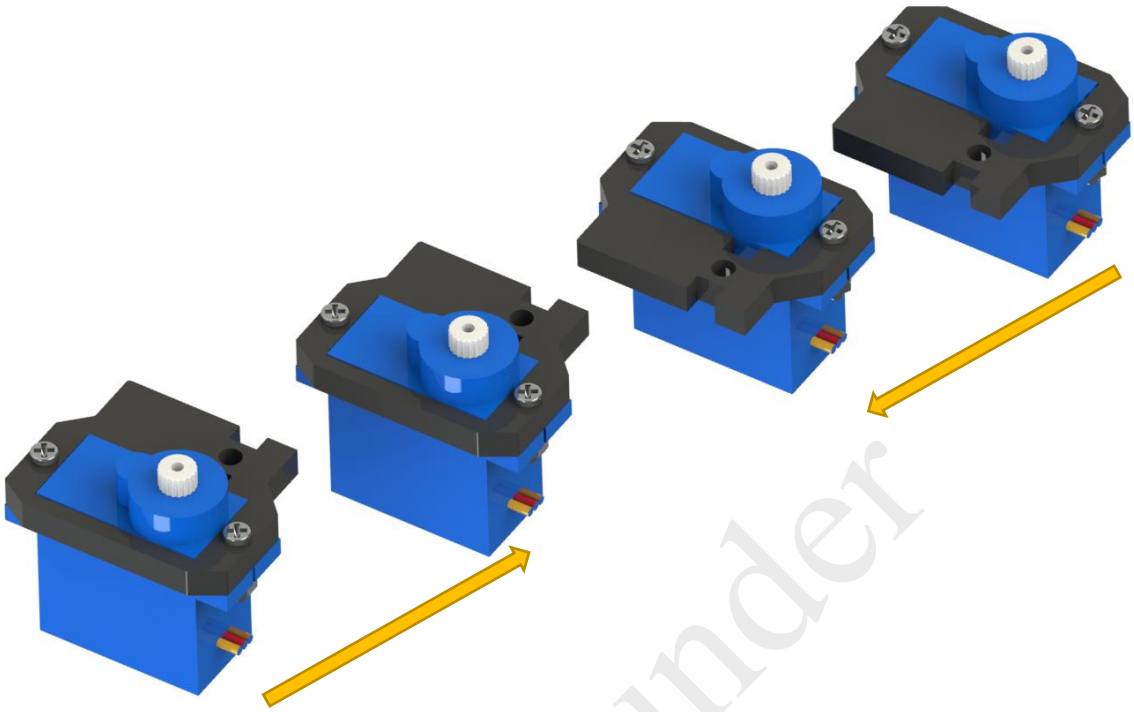
3. Connect the two components with an **M2*10 screw** and an **M2 nut.**

4. Tighten the screw.

5. Tighten the other screw and nut.

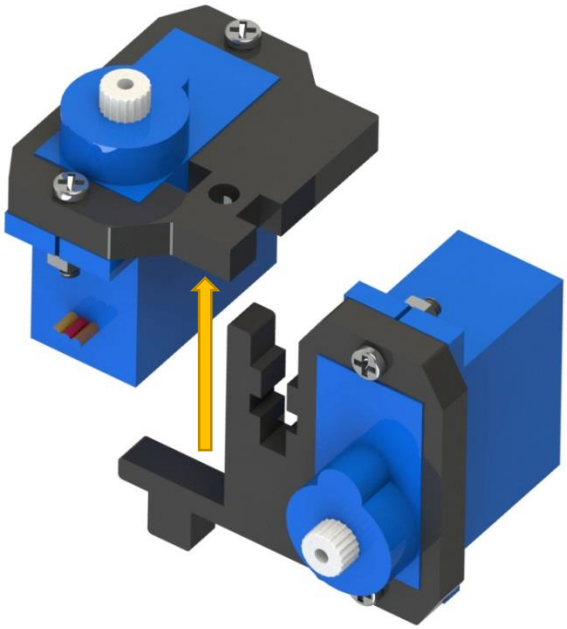


6. Assemble the other 3 servos.
   Please pay attention to the direction. They should be two opposite pairs, as shown below:

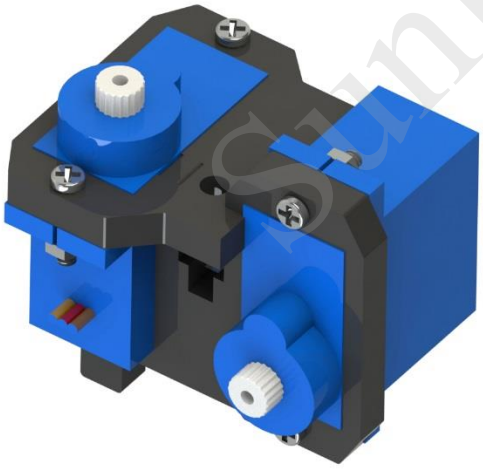


Thigh Joint 1 + Thigh Joint 2

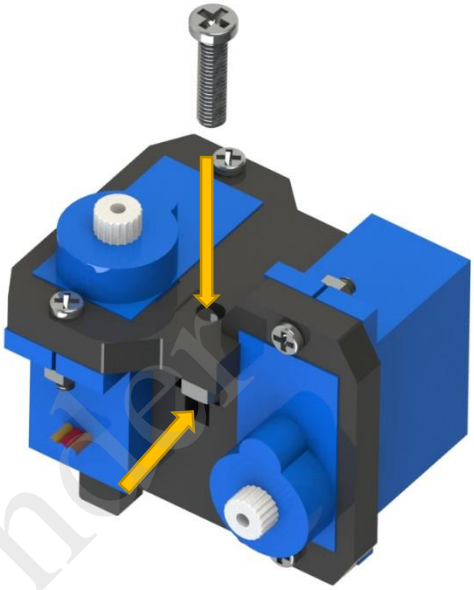
1. Assemble the following two joints.



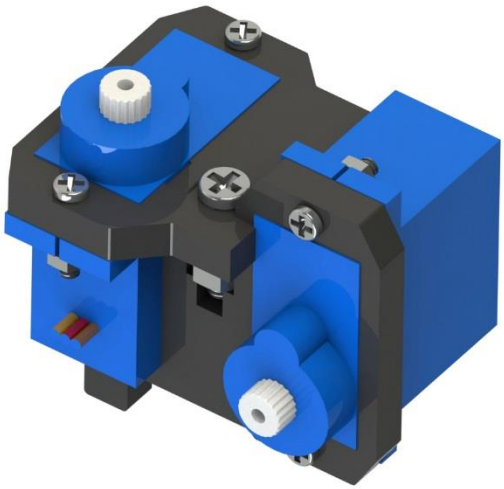
2. Stick one into the other tightly.



3. Fasten with an **M3*12 screw** and **M3 nut** (for ease, put the nut into the hole first, then the screw into the other from top).

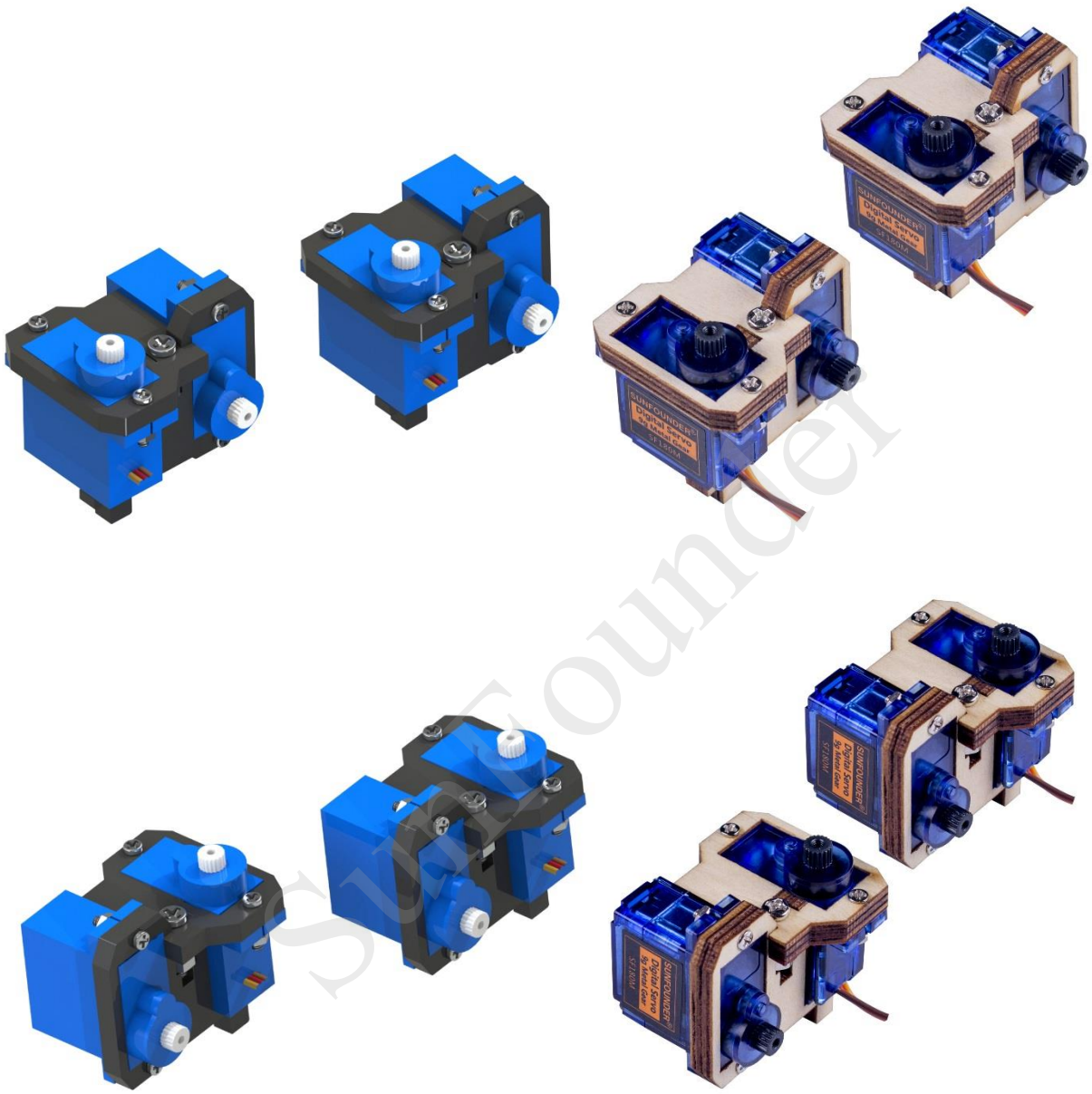


4. Tighten the screw.



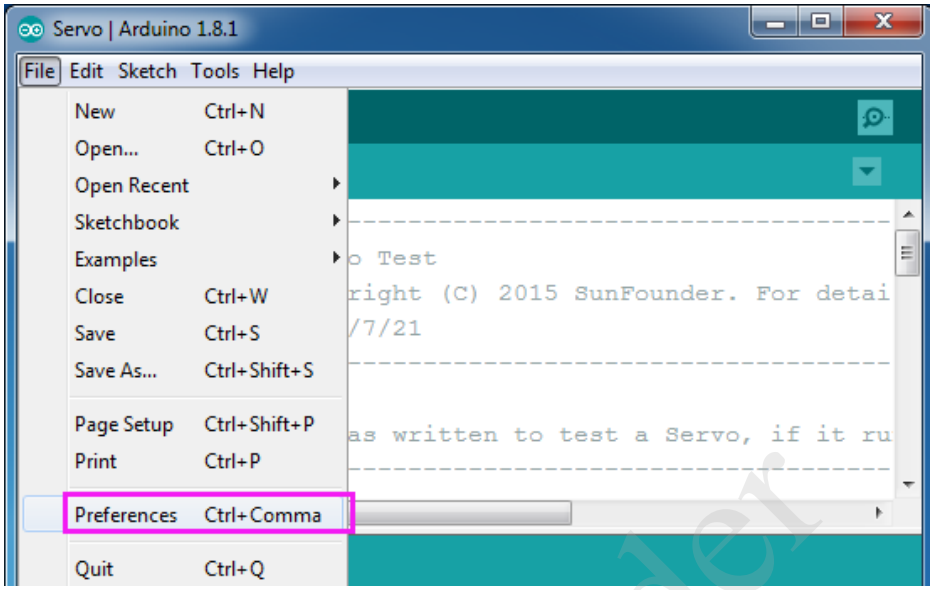
5. Assemble the other 3 pairs similarly.

Please pay attention to the direction. The two combined pairs should be of opposite directions, as shown below:

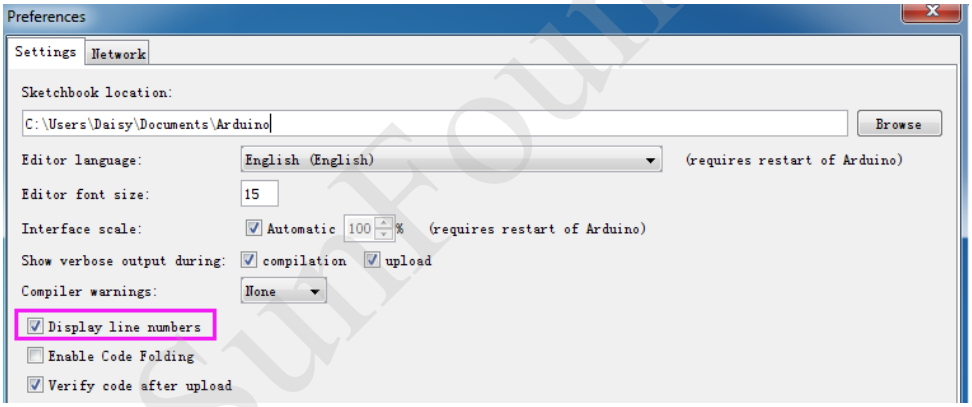


Uploading Assembly Program

Open Arduino IDE, and then select **File** -> **Preferences**.



On the pop-up window, tick the checkbox **Display line numbers**.

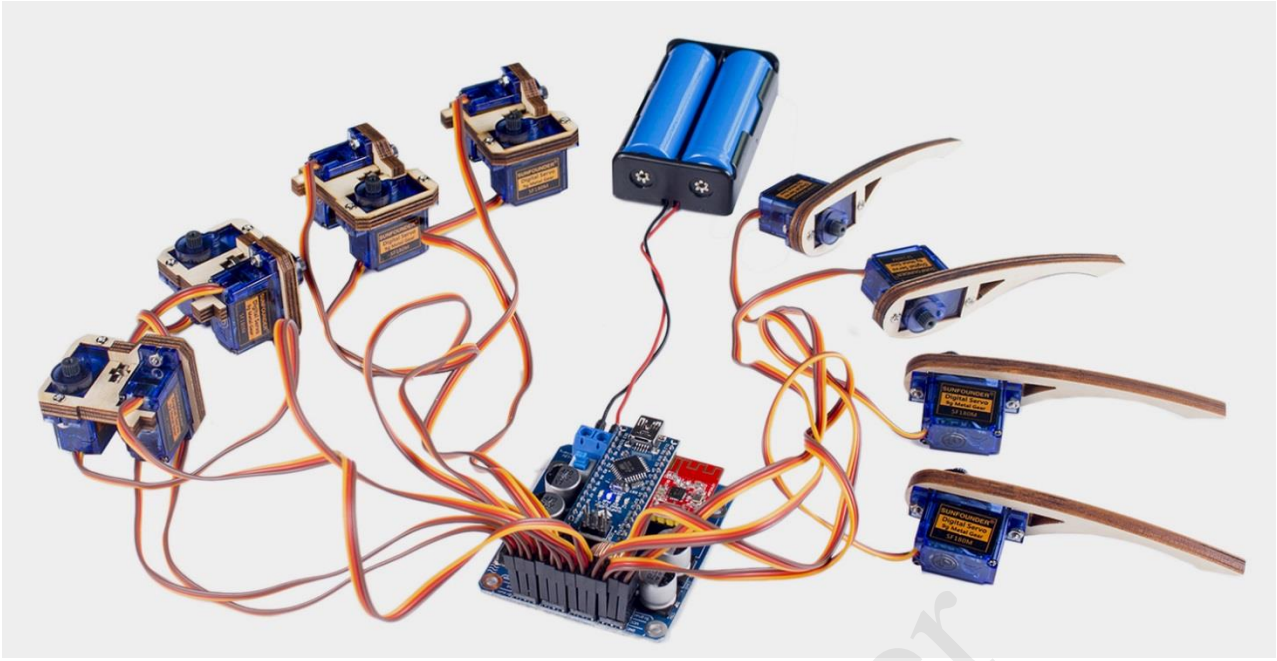


Go to *Crawler.ino* under Quadruped Crawling Robot Kit V2.0 for Arduino*code**3.Crawl**Crawler* directory.

Modify code Lines 32-34 as follows (uncomment the line 32 #define INSTALL), compile and upload the sketch to the SunFounder Servo Control Board, and then remove the USB cable.

```
31  /* Installation and Adjustment ----------------------------------------
32  #define INSTALL //uncomment only this to install the robot
33  //#define ADJUST        //uncomment only this to adjust the servos
34  //#define VERIFY        //uncomment only this to verify the adjustment
```

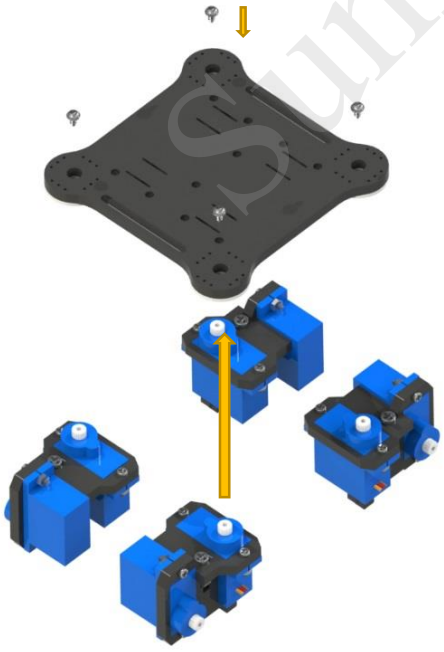
! ! Connect all the 12 servos to the SunFounder Servo Control Board, and then turn on the power switch. Now all the servos will rotate and then stay in a certain position. Keep the power ON and the servos CONNECTED.



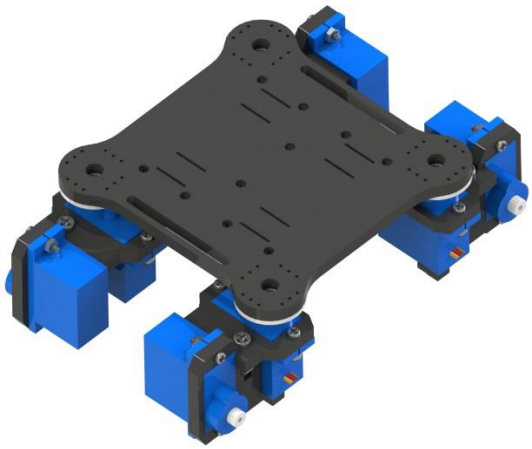
Upper Plate + Thigh Joint

Note: Through the assembly below, please keep the power ON and the servos CONNECTED

1. Connect the servo rocker arm and the servo with servo screws.
   (Packaged with the servo, the servo screw is the shortest one among the three.)
   The directions of all components must be exactly the same with the following picture:

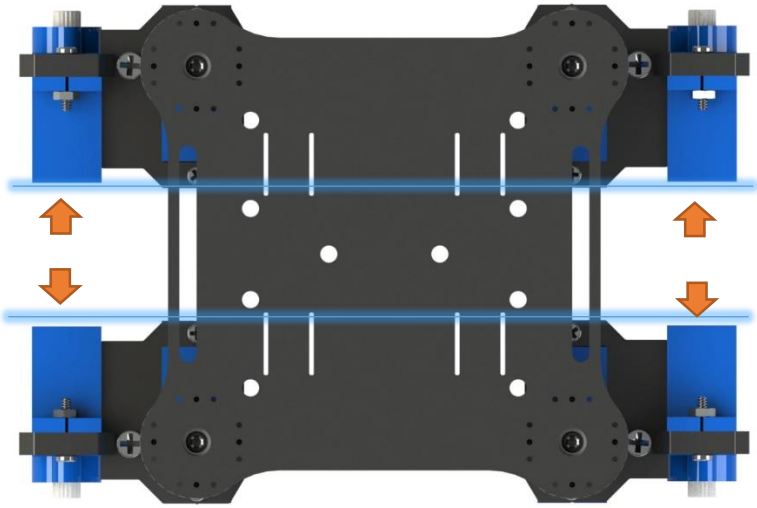


2. Tighten the screws. Pay attention that in the whole course of the subsequent assembly, the shaft of the unconnected servos should be outward.



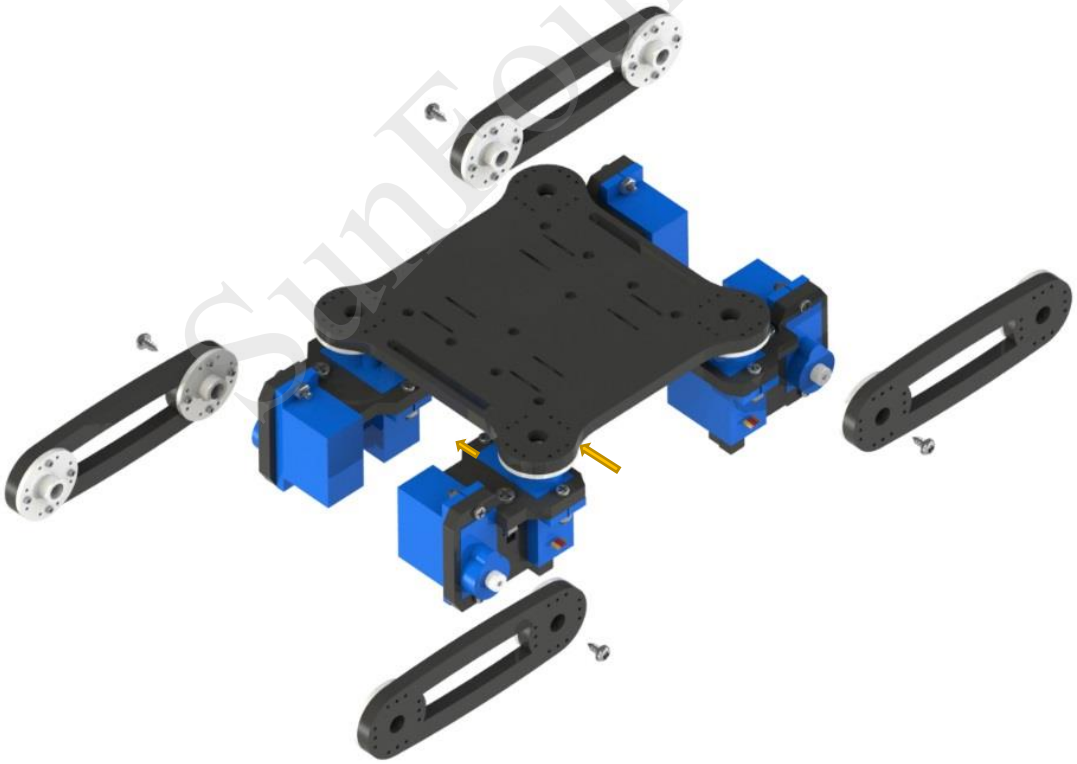
3. After connection, the top view is as follows. Please pay attention to the direction of these components. Try your best to make it close to the following picture. But if there is

some slight deviations, that's OK.

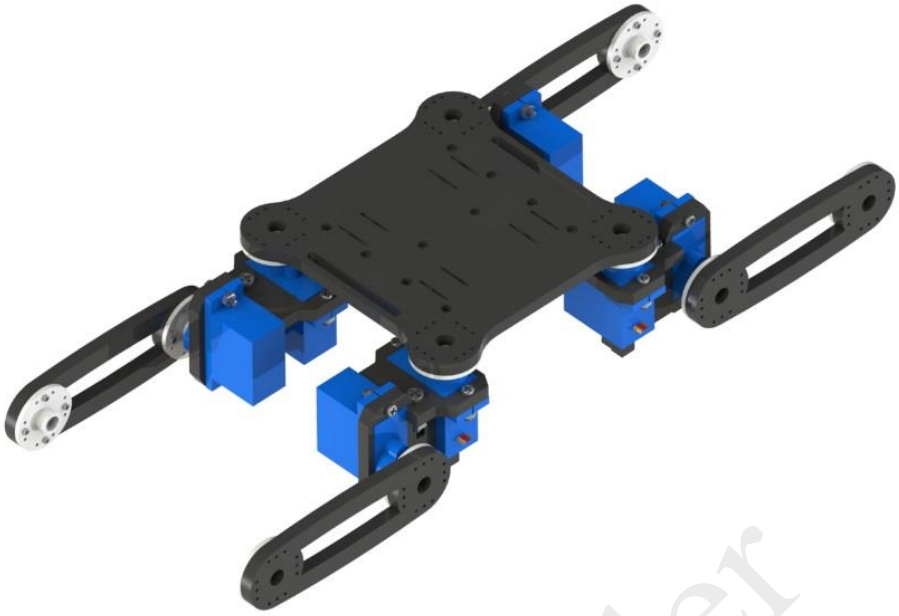


Thigh Joint + Thigh

1. Connect the servo rocker arm of thighs and the servo of the joints with servo screws. (Packaged with the servo, the servo screws are within the smaller two of five screws.) The directions of all components must be exactly the same with the following picture:

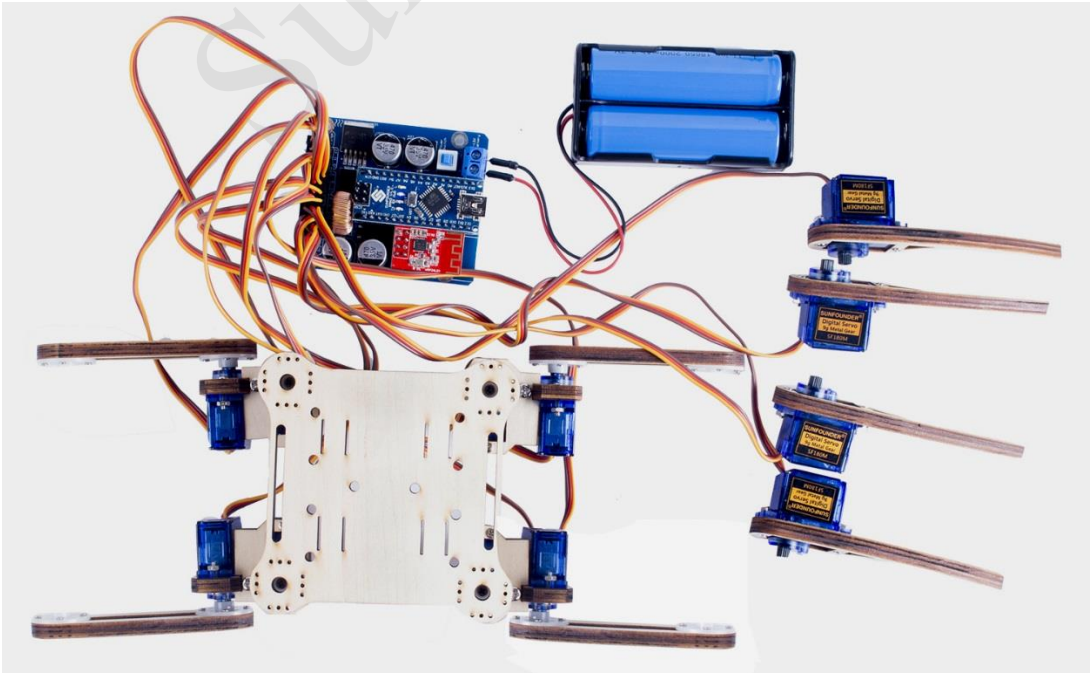


2. Tighten the screws.



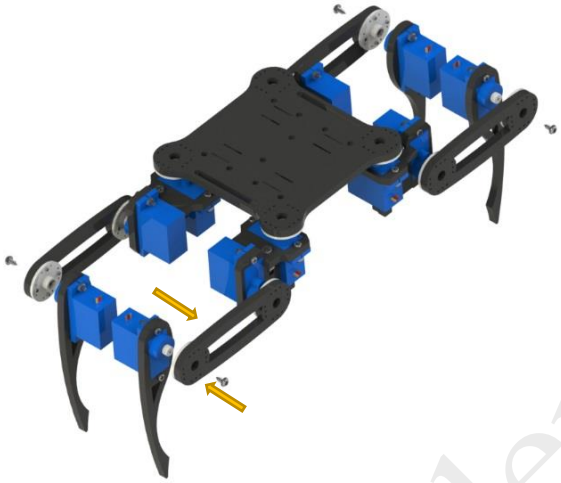
3. After connection, the front view is as follows. Please pay attention to the direction of these components, it should be horizontal.

Try your best to make it close to the following picture, but if there is some deviation, that's OK.



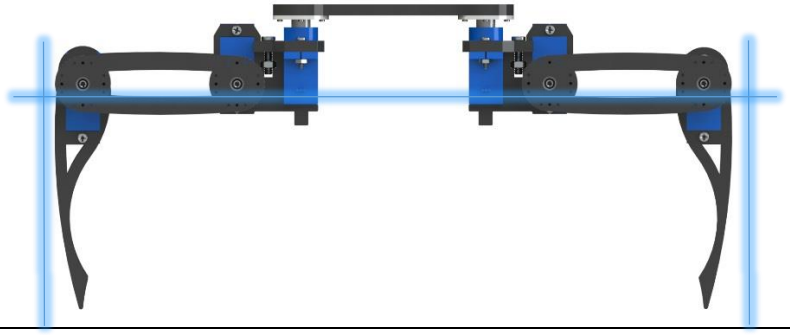
Thigh + Crus

1. Connect the servo rocker arm of thighs and the servo of crura with servo screws. (Packaged with the servo, the servo screws are within the smaller two of five screws.) The directions of all components must be exactly the same with the following picture:



2. Tighten the screws.

3. After connection, the front view is as follows. Please pay attention to the direction of these components, it should be vertical. Try your best to make it close to the following picture, but if there is some deviation, that's OK.



Switch off the SunFounder Servo Control Board. Remove the batteries first, then the battery holder, and finally all the servo wires. **NOW** you can rotate any joints of the robot.

Upper Plate + Copper Standoffs

Fasten the **M3*36 copper standoffs**, **M3*8 copper standoffs** and **M3*8 screws** into the following plate.

It is shown as follows after connection.

Battery Holder

Cross the ribbon through the plate, and connect the following components with the **M3*12 countersunk screws** and **M3 nuts**.

It is shown as follows after connection.

SunFounder Servo Control Board

Connect the SunFounder Servo Control Board to the following plate with **M3*6 screws**.

It is shown as follows after connection.

Lower Plate

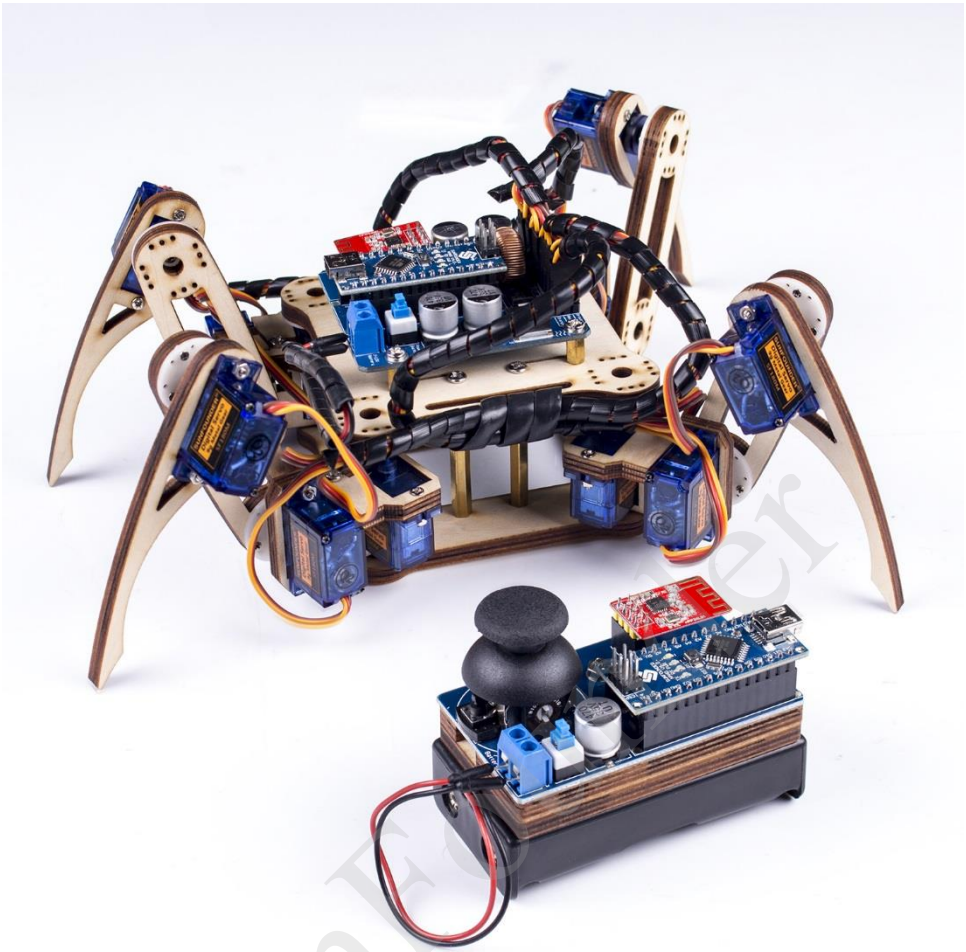
1. Keep the power off and connect the battery holder. Pay attention not to connect the power inversely.

2. Mount the batteries to the lower plate. Fix the lower plate with **M3*8 Screws**.

Servo Wiring

Wire the servos to the SunFounder Servo Control Board according to the following servo number. The numbers are corresponding to the pin number on the control board

Organize the wires of the servo with wire harness tube. Now the crawling quadruped robot has been fully assembled.



4. Calibrating

During the installation, errors may happen to mechanical connection. So you need to calibrate the robot to make sure the accuracy.

Uploading Code for Calibration

Open *Crawler.ino* under Quadruped Crawling Robot Kit V2.0 for Arduino*code**3.Crawl**Crawler* directory

Modify Lines 32-34 as follows (uncomment the Line 33 #define ADJUST), compile the code, upload it to the SunFounder Servo Control Board, and then remove the USB cable.

```
31  /* Installation and Adjustment ---------------------------------------
32  //#define INSTALL       //uncomment only this to install the robot
33  #define ADJUST  //uncomment only this to adjust the servos
34  //#define VERIFY        //uncomment only this to verify the adjustment
```

Switch on SunFounder Servo Control Board. At this point, the crawling robot will keep the calibration poses.

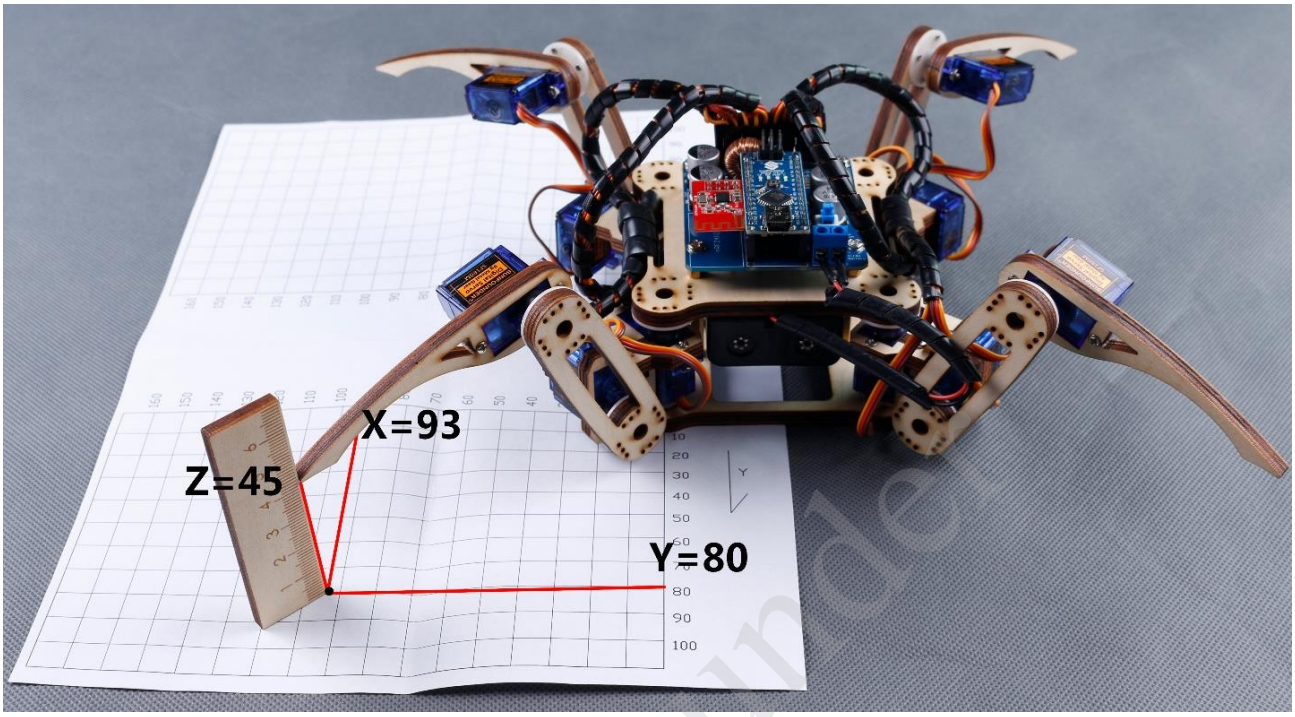
Calibration

Measure the 3D (x, y, z) coordinates of the toe tip of the four legs, put them in the array *real_site[4][3]* at Line 36 to calibrate the error during the installation.

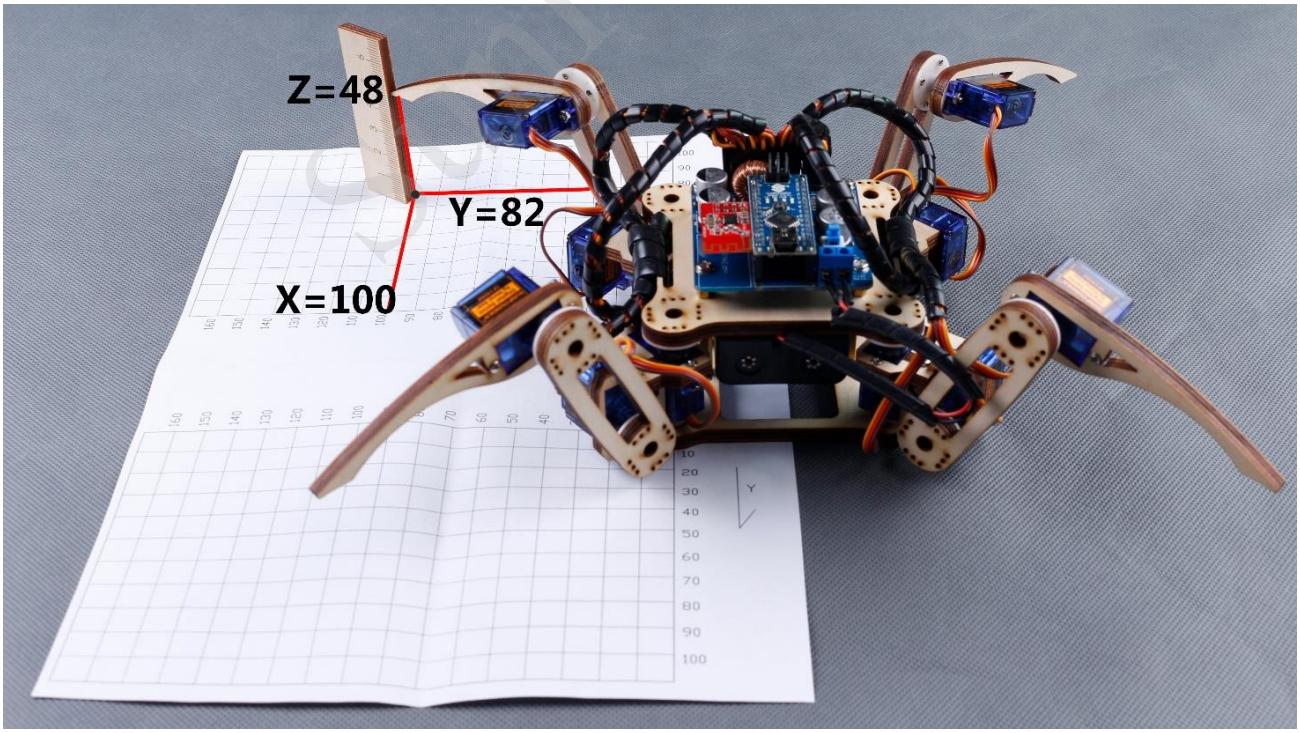
```
34  //#define VERIFY        //uncomment only this to verify the adjustment
35  const float adjust_site[3] = { 100,  80,  42 };
36  const float real_site[4][3] = { { 100,  80,  42 },  { 100,  80,  42 },
37                                  { 100,  80,  42 },  { 100,  80,  42 } };
```

a. Place the robot on the Calibration Chart as shown in the following picture. Measure the coordinates (x, y, z) of leg 1 with a ruler. Here it is (93, 80, 45).

   Please pay attention to the direction of the robot and refer to the orange arrow below.

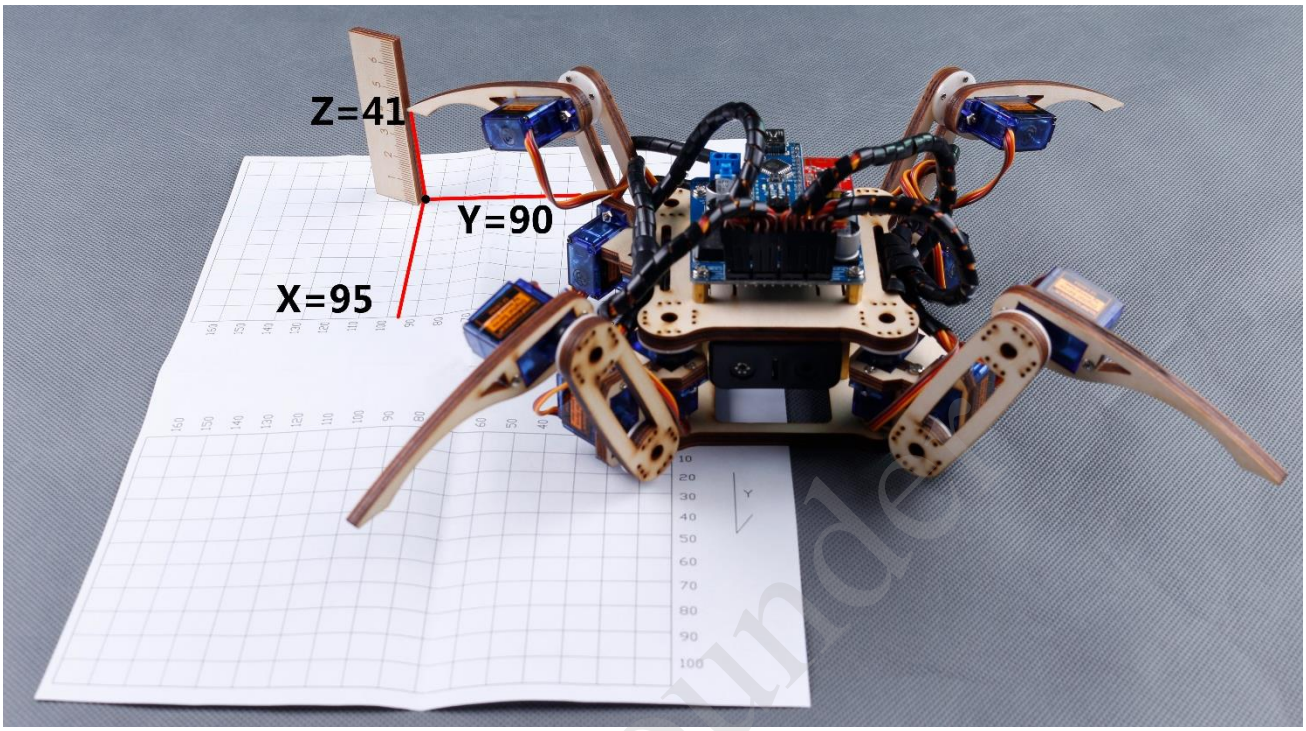


b. Measure the coordinate (x, y, z) of leg 2. Here it is (100, 82, 48).

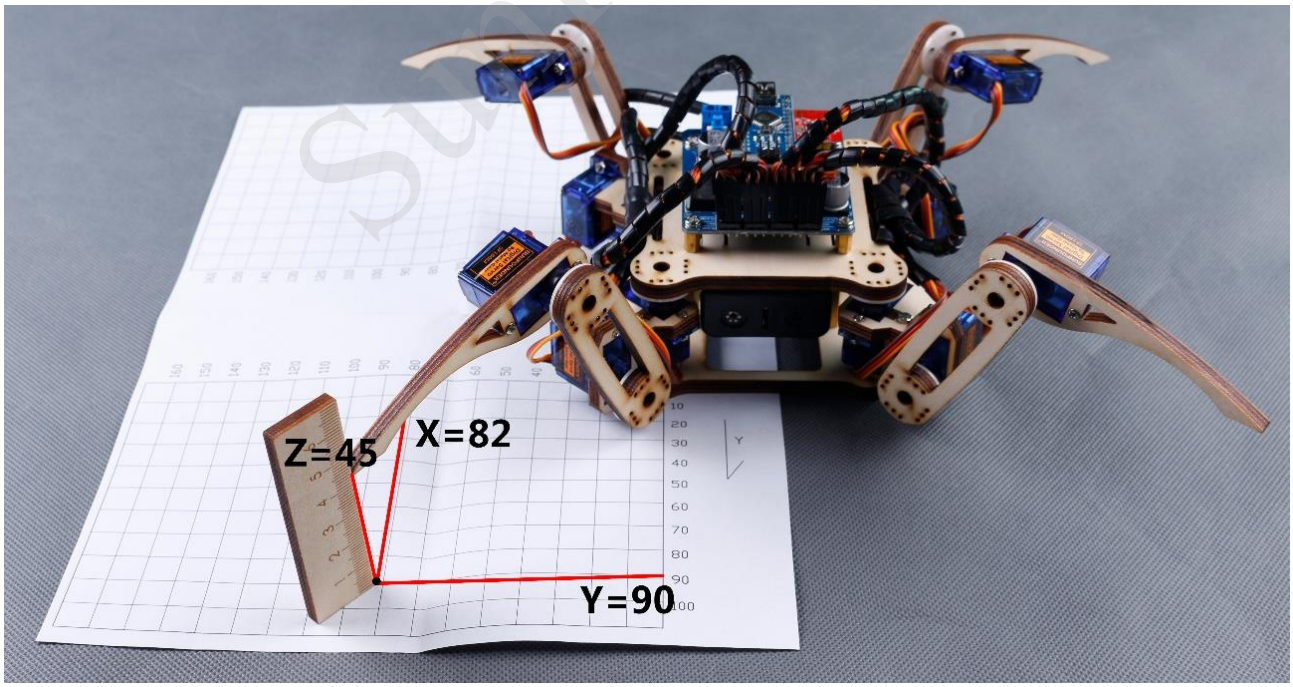


c. Rotate the robot as shown in the following figure, and measure the coordinates (x, y, z) of leg 3. Here it is (95, 90,41).

Please pay attention to the direction of the robot and refer to the orange arrow below.



d. Measure the coordinate (x, y, z) of leg 4. Here it is (82, 90, 45).



So, the coordinates filled in the array are (93, 80, 45), (100, 82, 48), (95, 90, 41), and (82, 90,

45), as shown below:

```
31 /* Installation and Adjustment ---------------------------------
32 //#define INSTALL //uncomment only this to install the robot
33 #define ADJUST  //uncomment only this to adjust the servos
34 //#define VERIFY  //uncomment only this to verify the adjustment
35 const float adjust_site[3] = { 100, 80, 42 };
36 const float real_site[4][3] = { { 93, 80, 45 }, { 100, 82, 48 },
37                                 { 95, 90, 41 }, { 82, 90, 45 }};
```

Fill in the values according to your actual measurements to calibrate the robot correctly.

Uploading Code for Verification

Modify Lines 32-34 as follows (uncomment the Line 34 #define VERIFY), compile and upload the sketch to the SunFounder Servo Control Board, and then remove the USB cable.

```
31 /* Installation and Adjustment ---------------------------------
32 //#define INSTALL //uncomment only this to install the robot
33 //#define ADJUST  //uncomment only this to adjust the servos
34 #define VERIFY  //uncomment only this to verify the adjustment
35 const float adjust_site[3] = { 100, 80, 42 };
36 const float real_site[4][3] = { { 93, 80, 45 }, { 100, 82, 48 },
37                                 { 95, 90, 41 }, { 82, 90, 45 }};
```

Switch on SunFounder Servo Control Board. At this point, the crawling robot will keep the verification poses.

Verifying

Place the crawling robot on the calibration chart, and then measure the 3D coordinates (x, y, z) with a ruler. If it is close to the calibration coordinates (100, 80, 42) set in *adjust_site[3]* and the error of each axis is less than 10mm, it means calibration is successful. Otherwise, recalibrate it until it succeeds.

5. Movement

Crawling

Uploading Code for SunFounder Mobile Robot Remote Controller

Open *Remoter.ino* under the Quadruped Crawling Robot Kit V2.0 for Arduino*code**3.Crawl**Remoter* directory.

Compile the code, upload it to SunFounder Mobile Robot Remote Controller, and then remove the USB cable. Switch on the remote controller.

Uploading Code for SunFounder Servo Control Board

Open *Crawler.ino* under the Quadruped Crawling Robot Kit V2.0 for Arduino*code**3.Crawl**Crawler* directory.

Modify Lines 32-34 as follows (uncomment the Lines 32-34), compile and upload the sketch to the SunFounder Servo Control Board, and then remove the USB cable.

```
31  /* Installation and Adjustment ----------------------------------------
32  //#define INSTALL       //uncomment only this to install the robot
33  //#define ADJUST        //uncomment only this to adjust the servos
34  //#define VERIFY        //uncomment only this to verify the adjustment
```

Switch on SunFounder Servo Control Board. Now you can control the robot crawling with the remote controller.

Dancing

Uploading Code for SunFounder Mobile Robot Remote Controller

The SunFounder Mobile Robot Remote Controller uses the above code, and you don't need to upload it again.

Uploading Code for SunFounder Servo Control Board

Open *Dance.ino* under the Quadruped Crawling Robot Kit V2.0 for Arduino*code**4.Dance**Dance* directory.

Modify Lines 36-37 as follows:

```
/* Installation and Adjustment ---------------------------------
//#define INSTALL //uncomment only this to install the robot
//#define ADJUST  //uncomment only this to adjust the servos
//#define VERIFY  //uncomment only this to verify the adjustment
const float adjust_site[3] = { 100, 80, 42 };
const float real_site[4][3] = { { 93, 80, 45 }, { 100, 82, 48 },
                                { 95, 90, 41 }, { 82, 90, 45 }};
```

Compile the code, upload the code to SunFounder Servo Control Board, and then remove the USB cable.

Switch on SunFounder Servo Control Board. The robot will dance for a while.

When the robot stops, switch on SunFounder Mobile Robot Remote Controller. Now, you can use it to make the robot dancing.

6. Code Explanation

Overview

For the quadruped, you can not only learn things about robotics and electrics, but also the code that animates the robot. In this section, the core code of the sketch, manipulator model of each leg, and proof of the model as well as the corresponding code for the proof will be presented in detail. When you've thoroughly understood these, you can write your own code for the robot! For example, you may write a sketch to make the robot swing the legs when walking, or sway a bit, walk in a bigger pace, dance more steps, etc. Sound amazing? Let's get started!

Core Code

This chapter focuses on how to transform the coordinates of the end of each leg into the rotational angle of each servo. First check the functions void cartesian_to_polar (volatile float &alpha, volatile float &beta, volatile float &gamma, volatile float x, volatile float y, and volatile float z). These are the core of the code for the quadruped robot, which is to transform the coordinates of the legs into the servo rotational angles.

Parameters: alpha, beta, gamma, the address that stores the output angle.

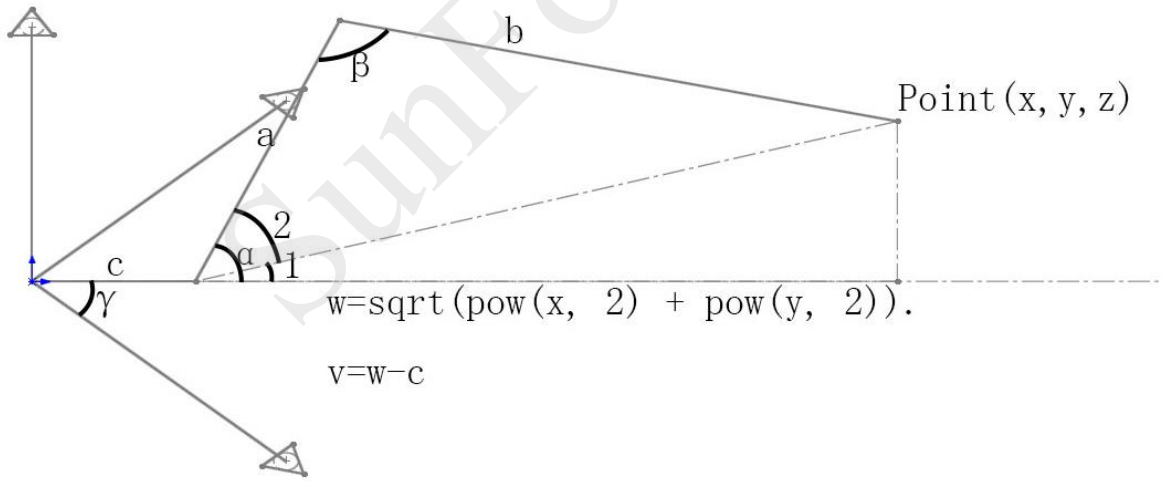
Parameters: x, y, z, the coordinates of the position of the leg end.

The source code of cartesian_to_polar:

```
/*
 - trans site from cartesian to polar
```

```
 - mathematical model 2/2
 * ------------------------------------------------------------------------*/
void cartesian_to_polar(volatile float &alpha, volatile float &beta, volatile
float &gamma, volatile float x, volatile float y, volatile float z)
{
    //calculate w-z degree
    float v, w;
    w = (x >= 0 ? 1 : -1)*(sqrt(pow(x, 2) + pow(y, 2)));
    v = w - length_c;
    alpha = atan2(z, v) + acos((pow(length_a, 2) - pow(length_b, 2) + pow(v, 2)
+ pow(z, 2)) / 2 / length_a / sqrt(pow(v, 2) + pow(z, 2)));
    beta = acos((pow(length_a, 2) + pow(length_b, 2) - pow(v, 2) - pow(z, 2)) /
2 / length_a / length_b);
    //calculate x-y-z degree
    gamma = (w >= 0) ? atan2(y, x) : atan2(-y, -x);
    //trans degree pi->180
    alpha = alpha / pi * 180;
    beta = beta / pi * 180;
    gamma = gamma / pi * 180;
}
```

First build a 3D model for a certain leg. The coordinate direction should be consistent with that on the calibration chart, as shown below:



Here we'll only analyze the first quadrant of the leg end: given the end position Point (x,y,z) and segment a, b, c (the length of each segment of the leg), to calculate the rotational angle of the servo α, β, γ. Within, $\pi/2 \le \alpha \le \pi/2$, $0 \le \beta \le \pi$, $-\pi/2 \le \gamma \le \pi/2$. In this way, transform these into a basic mathematic model. The proof of the model:

w= $\sqrt{x^2 + y^2}$

V=w-c

With the law of cosines, cos a = $\frac{b^2+c^2-a^2}{2*b*c}$, the result of $\angle 2$ can be calculated.

$\angle 2 = \arccos \frac{a^2+(z^2+v^2)-b^2}{2*a*\sqrt{z+v^2}.}$

$\therefore \angle \alpha = \angle 1 + \angle 2 = \arctan(z/v) + \arccos \frac{a^2+(z^2+v^2)-b^2}{2*a*\sqrt{z+v^2}.}$

The program should be:

```
alpha = atan2(z, v) + acos((pow(length_a, 2) - pow(length_b, 2) + pow(v, 2) +
pow(z, 2)) / 2 / length_a / sqrt(pow(v, 2) + pow(z, 2)));
```

Similarly, $\angle \beta = \arccos \frac{a^2+b^2-(z^2+v^2)}{2*a*b}$.

The program should be:

```
beta = acos((pow(length_a, 2) + pow(length_b, 2) - pow(v, 2) - pow(z, 2)) / 2 /
length_a / length_b);
```

Similarly, $\angle \gamma = \arctan(y/x)$.

The program should be (here only analyze the case for the leg end in the first quadrant):

```
gamma = (w >= 0) ? atan2(y, x) : atan2(-y, -x);
```

Hereto all the transformation from coordinates of the leg end into the servo rotational angle is done.

Each leg has its own coordinate system, which is calculated independently.

Servo_Service Function

After the function cartesian_to_polar is done in the sketch, immediately call the function void polar_to_servo(int leg, float alpha, float beta, float gamma) to adjust the servo rotational angle to the set angle. These two functions will be called one by one in the 50HZ service function void servo_service(void). It is a critical function and you need to pay much attention here.

Streamline Programming

After you've understood the core code and the working sequence, review the code:

```
/* Installation and Adjustment ------------------------------------------------
*/
#define INSTALL //uncomment only this to install the robot
//#define ADJUST    //uncomment only this to adjust the servos
//#define VERIFY    //uncomment only this to verify the adjustment
```

Activate the INSTALL command line and then add a *for()* loop in setup.

```
void setup()
{
#ifdef INSTALL
    //initialize all servos
    for (int i = 0; i < 4; i++)
    {
        for (int j = 0; j < 3; j++)
```

```
        {
            servo[i][j].attach(servo_pin[i][j]);
            delay(100);
        }
    }
    while (1);
```

Here set the shaft of the each servo in the center position so as to minimize the error during the installation. After servos are installed, run the calibration program to check whether all the servo are in the center position. Activate ADJUST line and start the calibration:

```
/* Installation and Adjustment ---------------------------------------------
*/
//#define INSTALL   //uncomment only this to install the robot
#define ADJUST  //uncomment only this to adjust the servos
//#define VERIFY    //uncomment only this to verify the adjustment
```

The program still waits in the loop in setup. Set a set of calibration coordinates manually. Then obtain the real coordinates via the calibration chart provided in the kit and a ruler (also an acrylic one included), and then modify the default real coordinates in the sketch.

```
const float real_site[4][3] = { { 115, 68, 42 }, { 105, 66, 60 },
{ 92, 70, 56 }, { 92, 70, 56 } };
```

Activate VERIFY and store the coordinates just obtained. Calculate the error and add it every time the servo rotates, so the accuracy of each segment moving can be ensured.

When all the calibration above mentioned is done, comment the three lines under Installation and Adjustment. After initialization, enter the loop. Here the servo service program runs in the frequency of 50Hz.

During this period, the main function waits for the remote control commands, so the robot moves accordingly under different command, while the service function is executed all the time, constantly determines whether there is a new target position, and drives the servo to rotate to the position by the functions cartesian_to_polar and polar_to_servo. Thus, when you push the joystick of the remote control, the corresponding command sent can be executed.

After all the explanation, you may hopefully be able to solve the problem encountered in coding and gain a lot from the kit now. Then try to make your own projects by modifying the code!

7. Afterword

So, you have the SunFounder crawling quadruped robot totally completed! You can start your journey now and have lots of fun with the bot. Show it to your friends and share with them your joy!

You may have encountered with assembly or code issues during the course. Though we have improved this manual and avoided some mistakes, it can still be unsatisfactory. We're sorry for the bothering and welcome to provide your feedbacks!

If you have any suggestions or ideas, post forums on our website www.sunfounder.com!

Copyright Notice