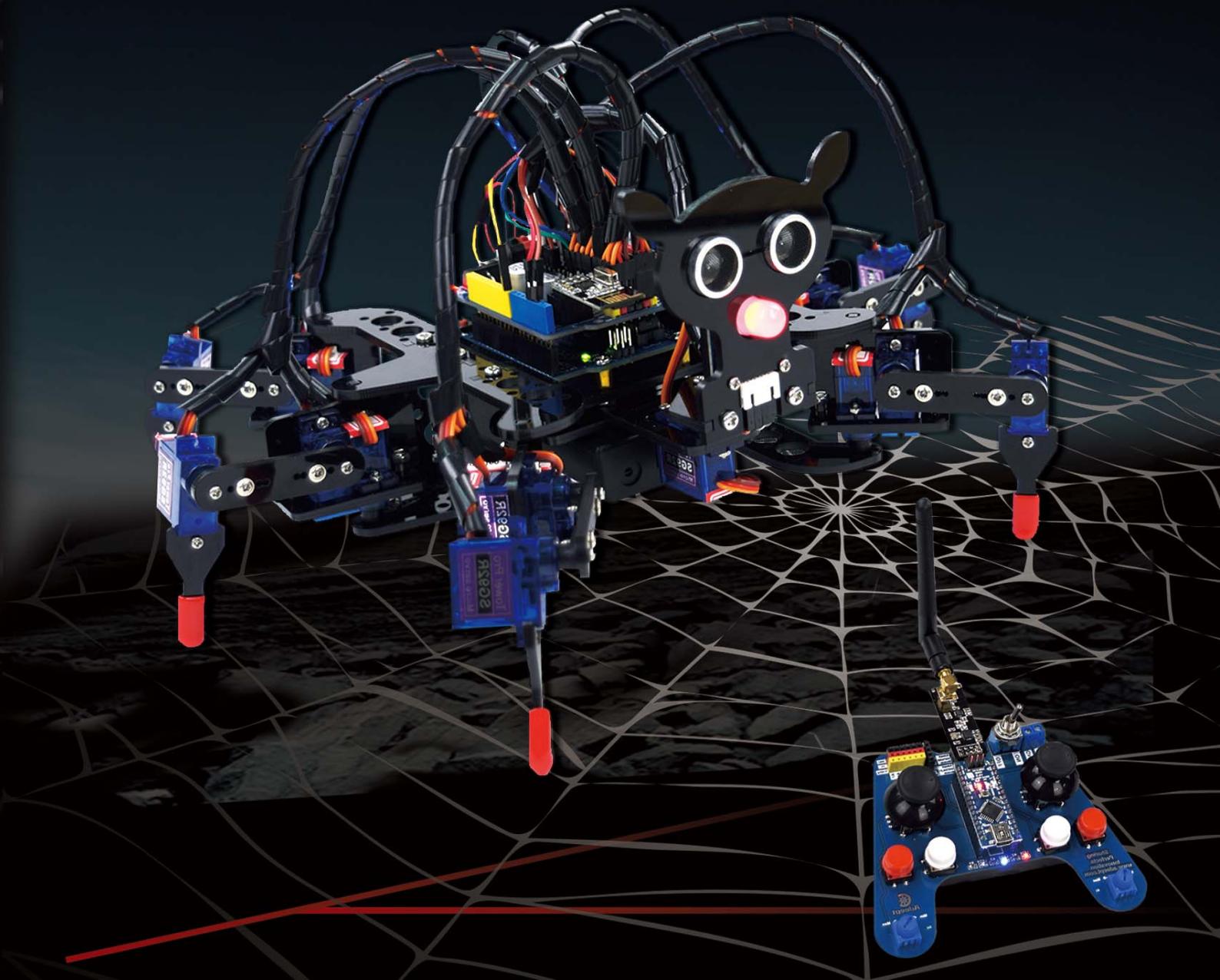
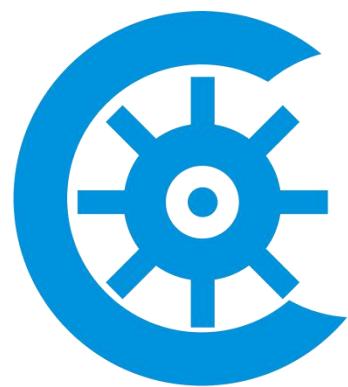




Adeept

# Adeept 6-Leg Spider Robot Kit for Arduino





# Adeept

[www.adeept.com](http://www.adeept.com)

## Warning

Please pay attention to the following issues when purchasing or using the product:

- ★ There are small components included in this kit. Swallowing mistakenly or misoperation can cause serious infection and be even fatal. When an accident occurs, please seek medical assistance immediately.
- ★ Please place the product in a safe place where an under-3-year-old cannot touch, who should not use or approach the product.
- ★ Juveniles should use the product with their parents.
- ★ Do not place the product or the components near any AC socket or other circuits, in case of potential risks of electric shock.
- ★ Do not use the product near any liquid or flame.
- ★ Do not use or store the product in an extreme environment such as extremely cold or hot and heavily humid.
- ★ Please remember to power off when the product is not in use.
- ★ Do not touch the moving or rotating part of the product.
- ★ The product may get heat at some part, which is just normal. But misoperation may cause overheat.
- ★ Misoperation may cause damage to the product. Please take care.
- ★ Do not connect the positive and negative poles of the power inversely, or the devices in the circuit may be damaged.
- ★ Please place and put the product gently. Do not smash, shock, or break it violently.

## About

Adeept is a technical service team of open source software and hardware. Dedicated to applying the Internet and the latest industrial technology in open source area, we strive to provide best hardware support and software service for general makers and electronic enthusiasts around the world. We aim to create infinite possibilities with sharing. No matter what field you are in, we can lead you into the electronic world and bring your ideas into reality.

The code and circuits of our product are open source. You can check on our website:

[www.adeept.com](http://www.adeept.com)

If you have any problems, feel free to send an email for technical support and assistance:

[support@adeept.com](mailto:support@adeept.com)

On weekdays, we usually will reply within 24 hours. Also welcome to post forums on our website.

## Copyright

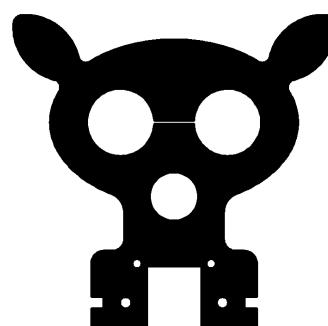
This user manual can be used for learning, DIY, refitting, etc., except for commercial purpose. The Adeept Company owns all rights of contents in the manual, including but not limited to texts, images, data, etc. Any distribution or printing should be implemented with the permission of the Company, or it will be deemed illegal.

# Contents

Components List.....	1
Acrylic Sheets.....	1
Machinery Parts.....	3
Electronic Parts.....	3
Tools.....	5
Self-prepared Parts.....	6
About Adeept 6-Leg Spider Robot.....	7
Introduction.....	7
Functions.....	8
Assembly.....	9
Preparations.....	9
Servo Adjustment .....	11
Left and Right Legs.....	18
Middle Body.....	26
Prop.....	33
Final Part.....	34
Remote Control.....	49
Circuit Connection.....	51
Install and Remove Batteries.....	53
Software & Hardware.....	54
What is Arduino?.....	54
Why Arduino?.....	54
How Should I Use Arduino?.....	55
Arduino Software (IDE).....	56
Install Library.....	60
Upload Program.....	63
Afterword.....	67

## Components List

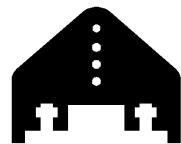
### Acrylic Sheets



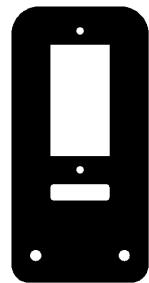
A01  
1pcs



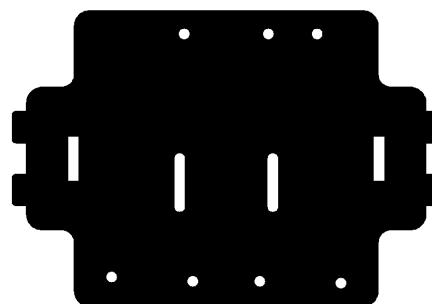
A02  
1pcs



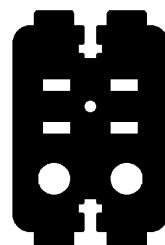
A03  
1pcs



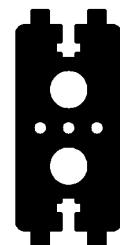
A04  
1pcs



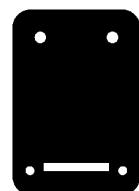
A05  
2pcs



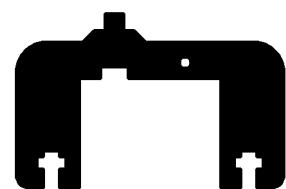
A06  
4pcs



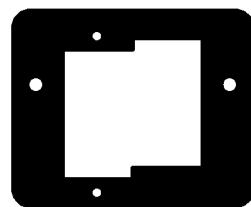
A07  
6pcs



A08  
1pcs



A09  
8pcs



A10  
8pcs



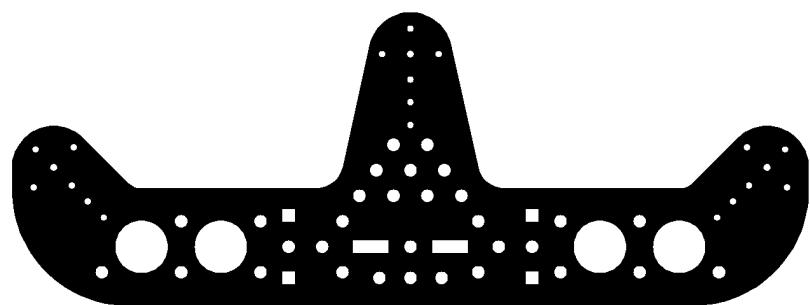
A11  
8pcs



A12  
8pcs

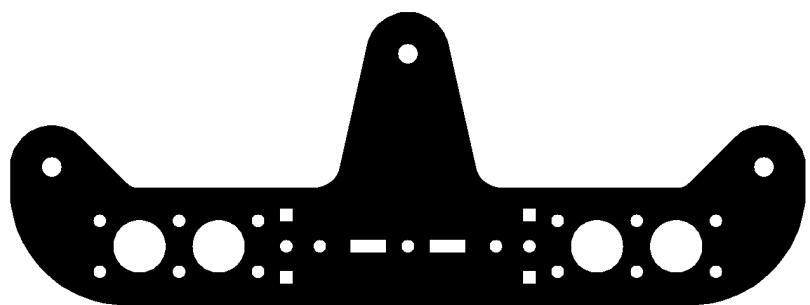


A13  
4pcs



A14

2pcs



A15

2pcs



A16

1pcs



A17

2pcs

The acrylic plates are fragile, so please be careful when assembling them in case of breaking the plates. Do not smash or shock them violently.

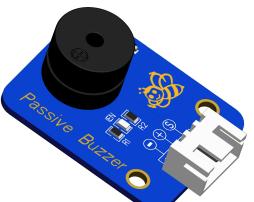
The acrylic sheet is covered with a layer of protective film. You need to remove it first.

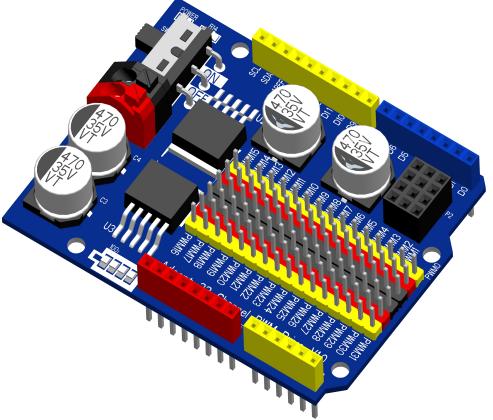
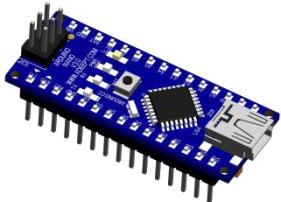
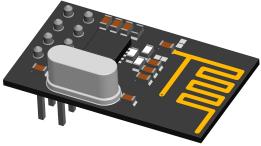
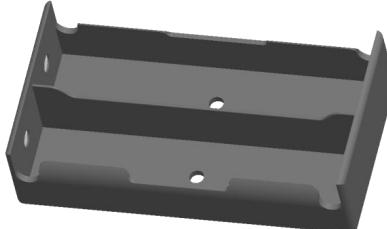
Some holes in the acrylic have residues, so you need to clean them before using it.

## Machinery Parts

M2 Nut  X32 <a href="http://www.adeept.com">www.adeept.com</a>	M3 Nut  X38 <a href="http://www.adeept.com">www.adeept.com</a>	M2*8 Screw  X47 <a href="http://www.adeept.com">www.adeept.com</a>	M2*10 Screw  X2 <a href="http://www.adeept.com">www.adeept.com</a>	M3*5 Screw  X7 <a href="http://www.adeept.com">www.adeept.com</a>
M3*8 Screw  X31 <a href="http://www.adeept.com">www.adeept.com</a>	M3*14 Screw  X1 <a href="http://www.adeept.com">www.adeept.com</a>	M3*30 Screw  X2 <a href="http://www.adeept.com">www.adeept.com</a>	M3*10 Countersunk Head Screw  X4 <a href="http://www.adeept.com">www.adeept.com</a>	M1.4*6 Self-tapping Screw  X4 <a href="http://www.adeept.com">www.adeept.com</a>
M3*6 Copper Standoff  X4 <a href="http://www.adeept.com">www.adeept.com</a>				

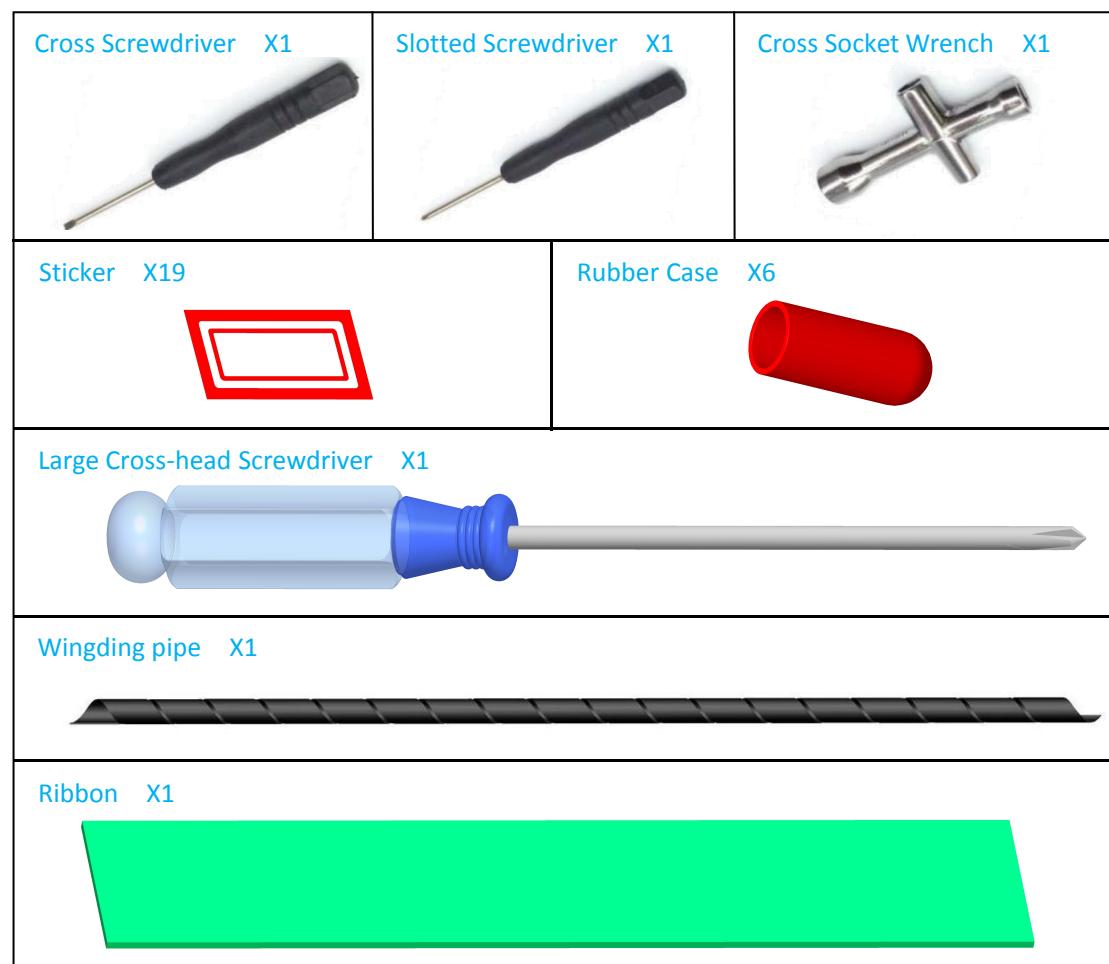
## Electronic Parts

Servo X19 	HC-SR04 Ultrasonic Module X1 
Adeept RGB LED Module X1 	Adeept Passive Buzzer Module X1 

Adeept UNO R3 X1 	Adeept 32 Channel PWM Drive X1 
Arduino Nano X1 	Adeept Remote Control Shield X1 
NRF24L01 X1 	
NRF24L01 X1 	
18650x2 Battery Holder X2 	
3-Pin Wires X1 	
Female-Female Cable X4 	

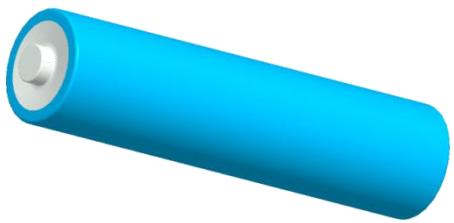


## Tools



## Self-prepared Parts

18650 Battery X4



Pencil X1



---

# About Adeept 6-Leg Spider Robot

## Introduction

Developed based on Arduino, this hexapod robot simulates the movements of 6-feet insects via programming in the Arduino IDE programming software and controlling 19 servos.

Hexapod robot, or spider robot, is one type of multi-legged robots. The design of this hexapod is derived from insects in nature, especially principles of their movement. Foot is the motion organ of insects. Some have 3 pairs of feet with each in the front, middle, and back breast; the feet are named respectively the frontfoot, midfoot and rear-foot. This hexapod robot is of the similar structure.

In this robot kit, a complete set of hardware devices are provided as well as example code for your learning. Also you can write the code for control by yourself if applicable. By programming the robot, you can control each leg and even each joint of the legs precisely, and truly feel the thrill and joy of learning and making.

The Adeept Arduino UNO R3 board is used as the core control board for this kit, and an Adeept 32 Channel PWM Drive board for control of 19 servos onside.

After assembly, you can see 2 parts, the hexapod robot and a remote control. They communicate through an NRF24L01 2.4G wireless communication module on each, which supports data transmission.

In the example code provided, the robot supports 2 working modes: remote control and automatic obstacle avoidance.

**Note:**

The robot will only move after it stands up. Press the button A and the robot will stand/squat.

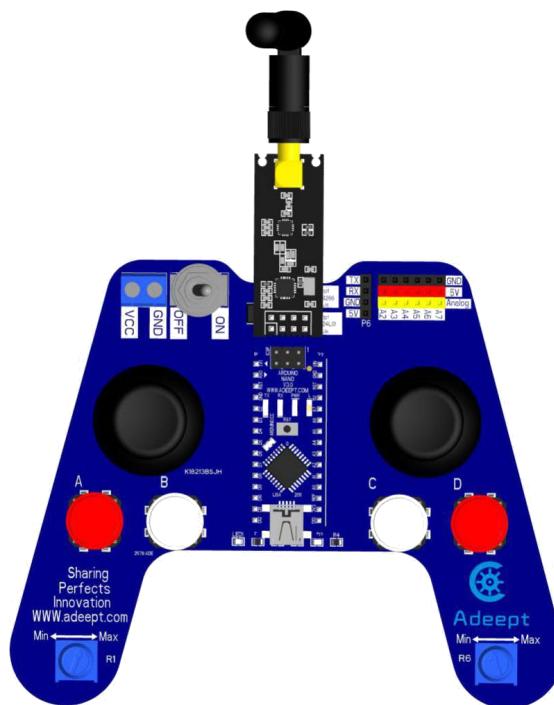
Under the remote control mode, you can make the bot stand, squat, go forward and backward, and turn left and right with the remote control.

Under the mode of automatic obstacle avoidance, it will go forward automatically, and bypass obstacles in front if any and continue to walk ahead then.

Once powered, the robot has its 6 feet stretched and the RGB LED at its head lights on red. After it connects with the remote control successfully, the LED1 on the remote will blink. Press button A on the remote. Then the robot will stand at once, and after the RGB LED on its head turns into blue, you can control it with the remote control.

## Functions

### Remote Control

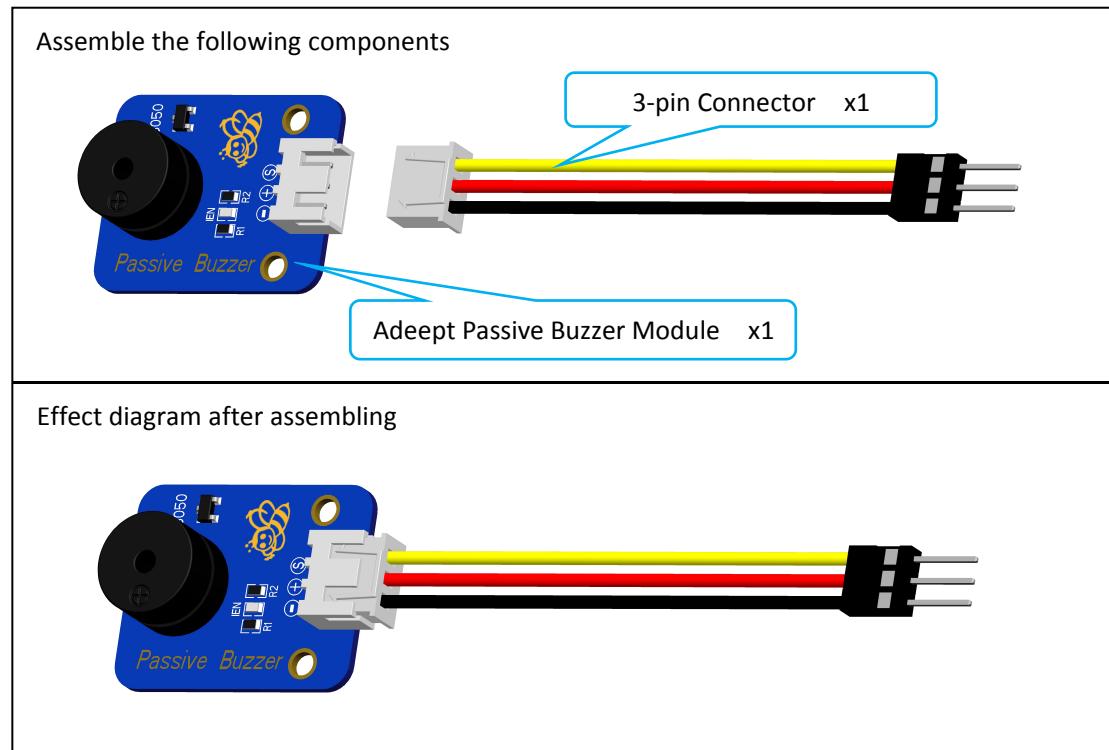


1. Push the left rocker up/forward and the robot will move forward. The farther you push each time, the faster the robot will move.
2. Pull the left rocker down/backward and the robot will move backward. The farther you pull each time, the faster the robot will move.
3. Pull the right rocker to the left and the robot will turn left. The farther you pull each time, the faster the robot will move.
4. Pull the right rocker to the right and the robot will turn right. The farther you pull each time, the faster the robot will move.
5. **Press the button A and the robot will stand/squat. After it stands up, the RGB LED will turn into blue (or green); when it squats, this LED will change to red.**  
**Note:** The robot will only move after it stands up.
6. Press button B and the robot now enters the remote control mode. At the same time the LED2 on the remote will light up when the LED3 dims. And the RGB LED on the robot's head turns back to blue.
7. Press button C to change into the mode of automatic obstacle avoidance. The LED3 lights up when LED2 goes out. The RGB LED turns into green.
8. Press button D to control the buzzer beep of the module on the robot.

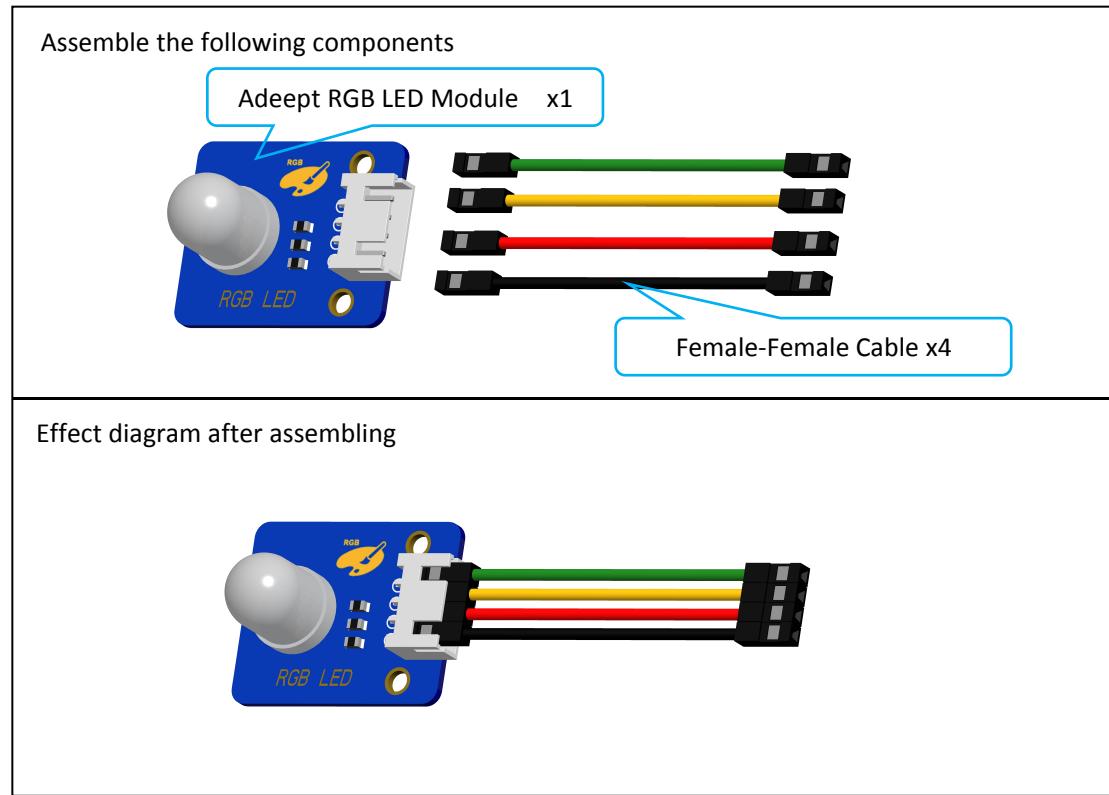
# Assembly

## Preparations

- A. Connect the Adeept Passive Buzzer Module and the 3-pin Connector.

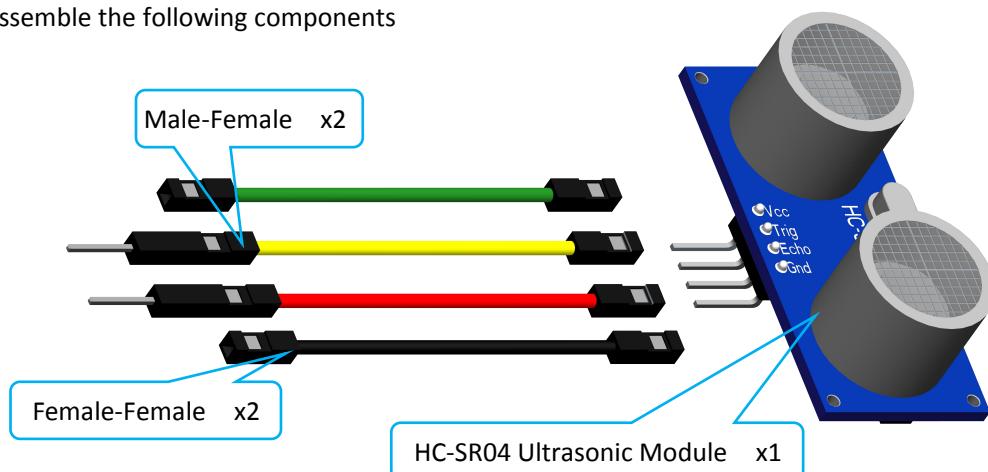


- B. Insert the Female-Female cables into the port on the Adeept RGB LED Module.

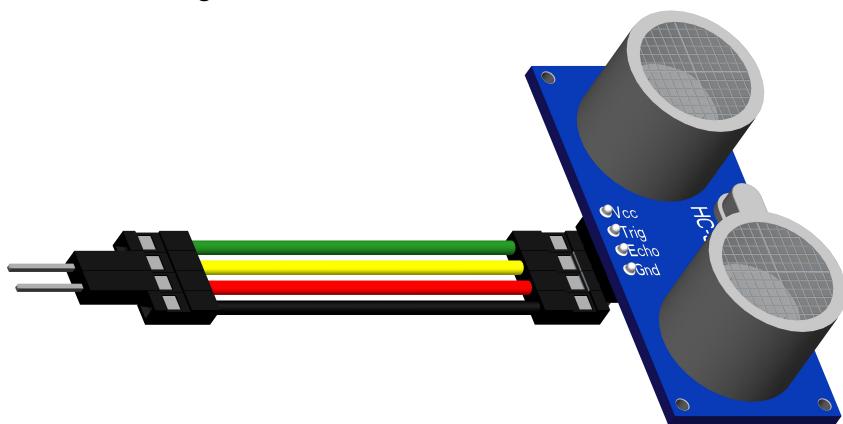


C. Insert the Female-Female and Male-Female cables into the HC-SR04 Ultrasonic Module.

Assemble the following components



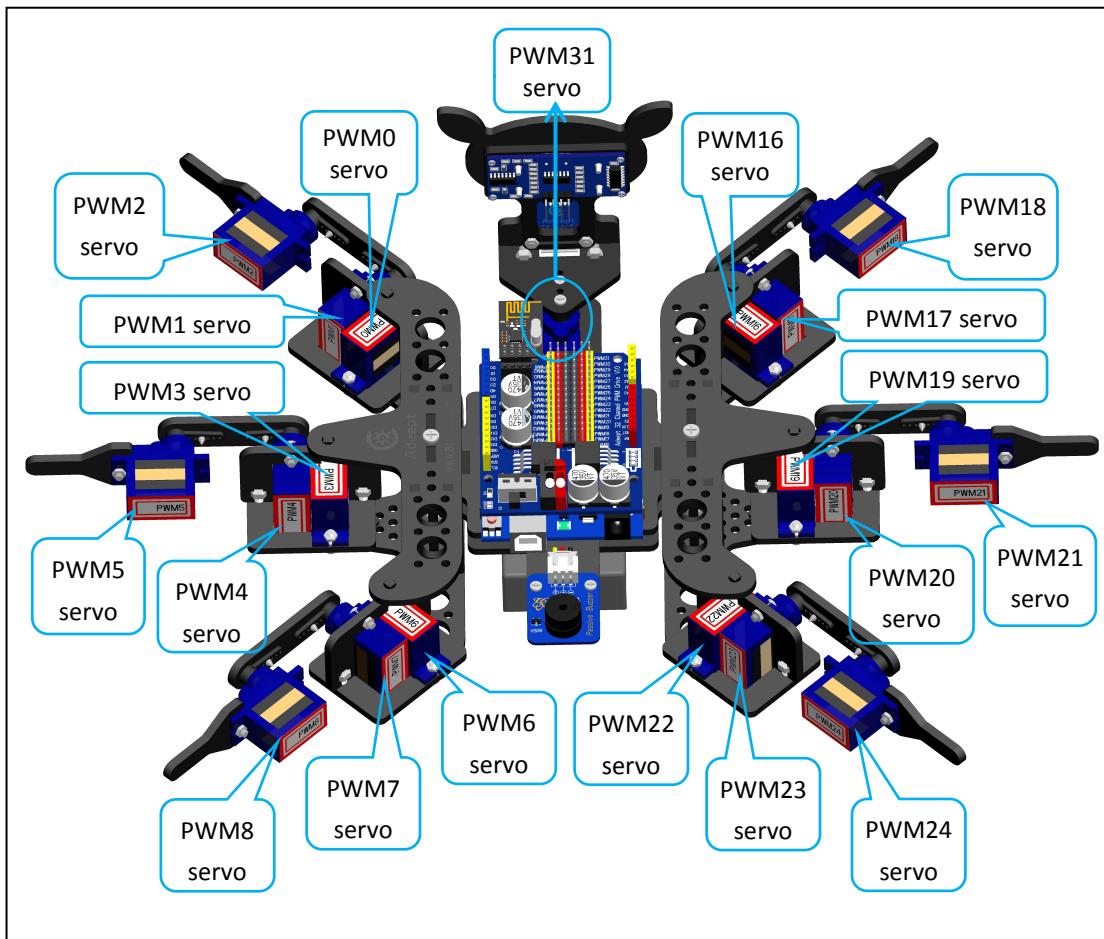
Effect diagram after assembling



## Servo Adjustment

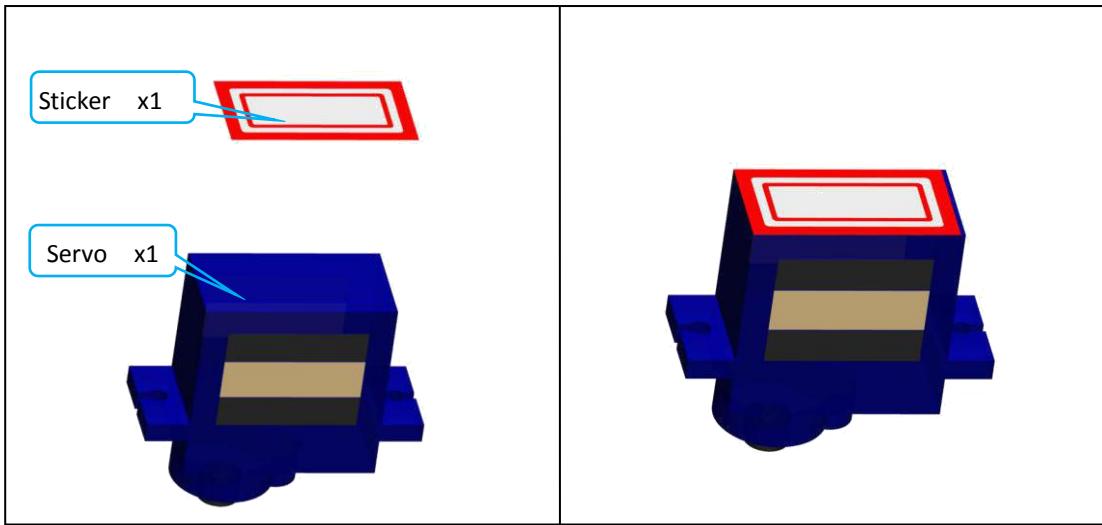
Before taking the following steps, you need to first finish reading this part "Servo Adjustment" carefully and then start to assemble. Otherwise, errors are prone to occur.

First take a look of the overall robot after assembly is done (robot power on).



Based on the figure above, you need to assemble 19 servos in total and the assembly differs for each. They all need adjustment before assembly but situations are varied. Looking at these alike gadgets, you may find it difficult to tell how each is adjusted after it's done so. Therefore, prior to adjustment, number them first and you can make it right when assembling them later.

A. Glue a sticker at the bottom of the 19 servos.

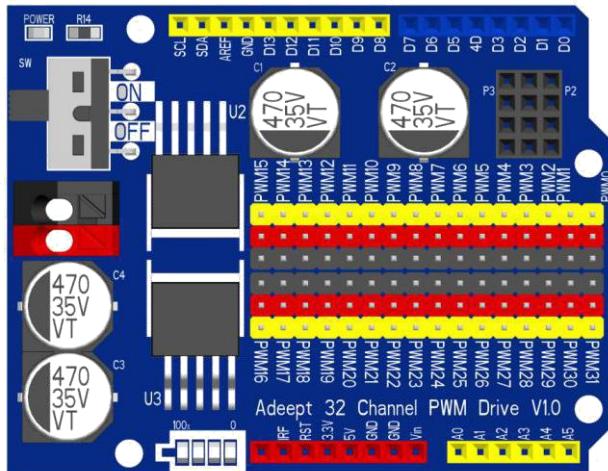


B. Number the servos by the stickers.

Since you may need to check the number of servos in the subsequent assembly and circuit connection, here number them in a set order as follows.

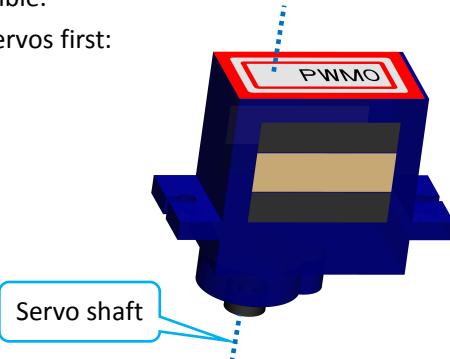
First, based on their connection to the Adeept 32 Channel PWM Drive module, use 19 numbers, i.e. PWM0-PWM8, PWM16-PWM24 and PWM31. So you can connect the right servos in the circuit easily.

**Note:** After numbering one or two servos, do not hush to write numbers on the stickers.



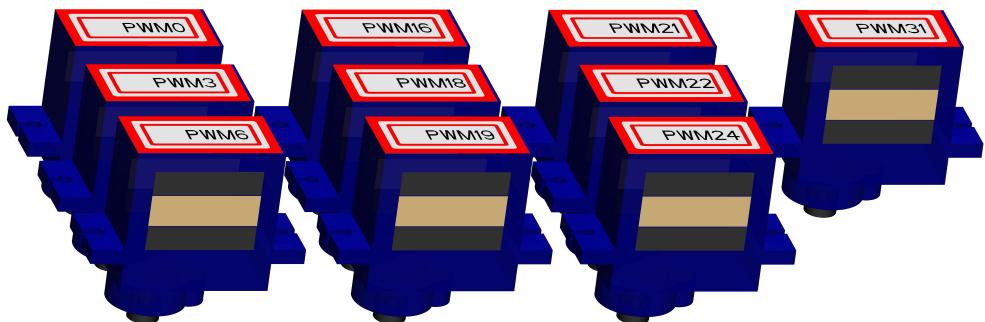
Then write the numbers on the stickers one by one. Since some or part of some may be covered later, here make some regulations for the labeling method so as to show them as much as possible.

Number 10 servos first:

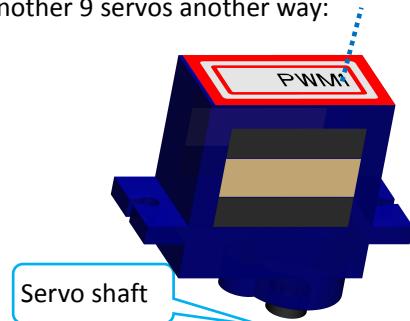


Based on the figure on the left, write "PWM0" on the right side of the sticker, and the number "0" far away from the servo shaft.

In the same way, number the rest 9 servos as PWM3, PWM6, PWM16, PWM18, PWM19, PWM21, PWM22, PWM24, and PWM31.

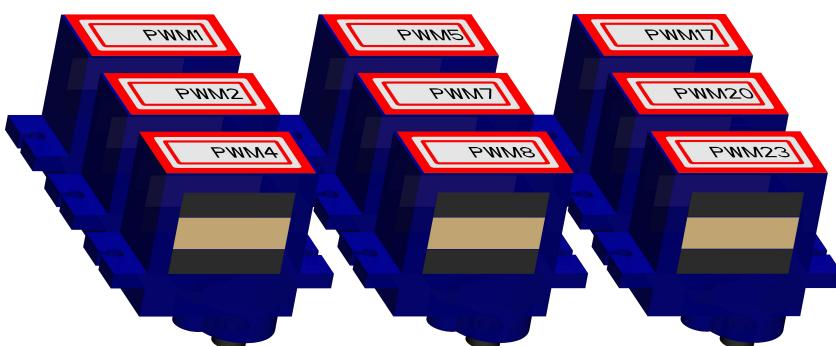


And number another 9 servos another way:



Based on the figure, write "PWM1" on the right side of the sticker, and the number "1" near the servo shaft.

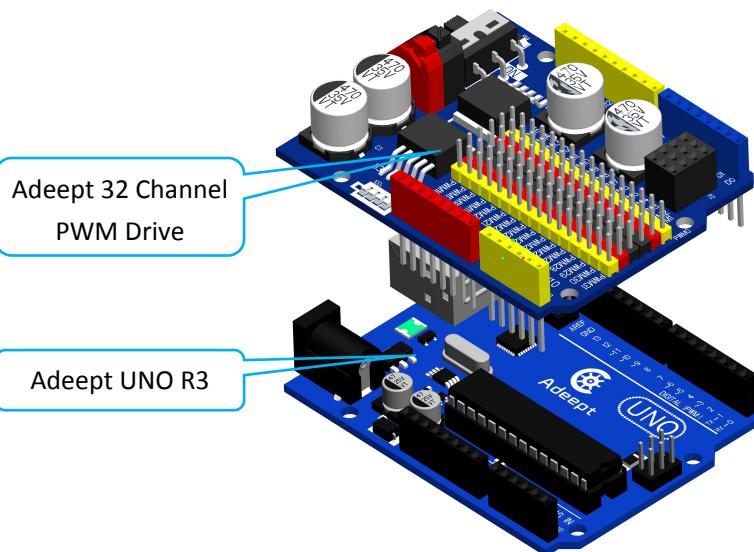
In the same way, number the rest 9 servos as PWM1, PWM2, PWM4, PWM5, PWM7, PWM8, PWM17, PWM20, and PWM23.



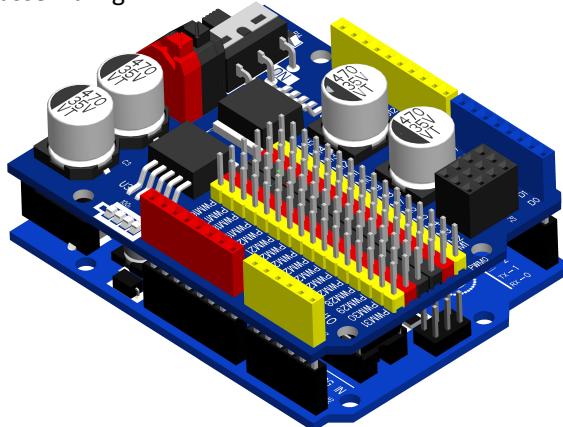
DO separately number these two groups of servos in different ways. And be careful not to erase the label during the assembly.

C. Assemble the Adeept 32 Channel PWM Drive board to the Adeept Arduino UNO R3.

Assemble the following components

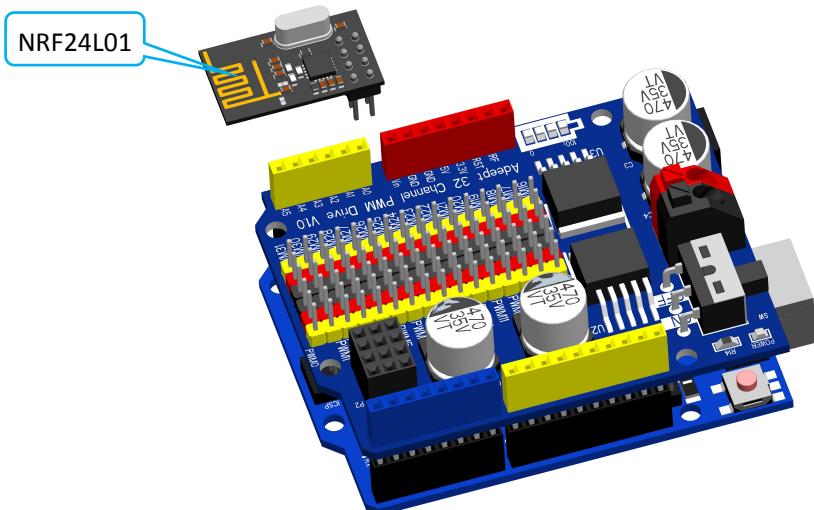


Effect diagram after assembling

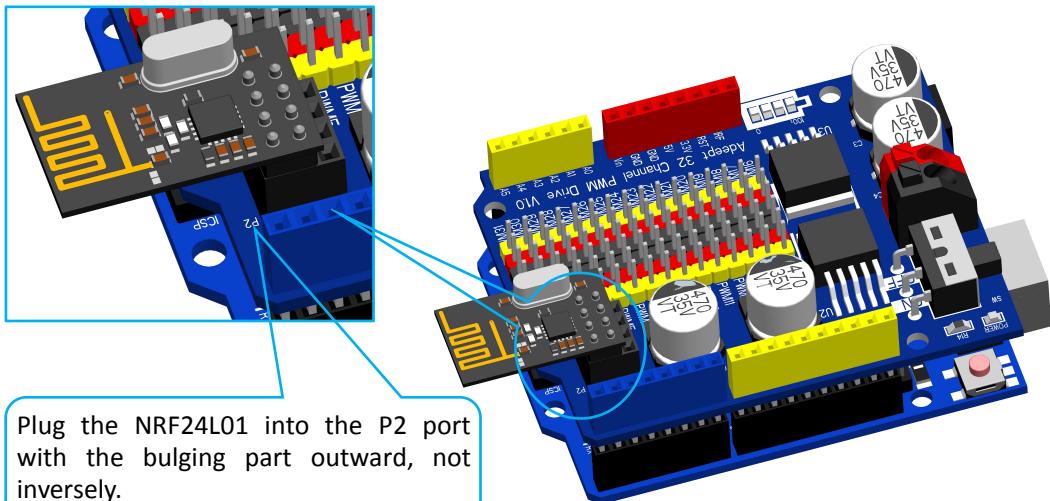


D. Plug the NRF24L01 wireless communication module to the PWM Drive board.

Assemble the following components



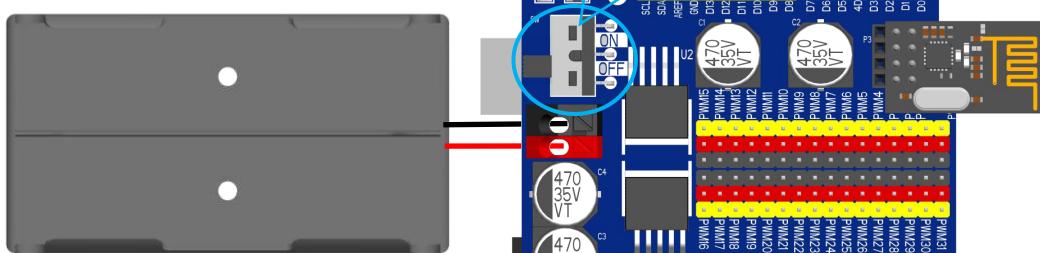
Effect diagram after assembling



#### E. Connect the 18650x2 Battery Holder and Adeept 32 Channel PWM Drive board.

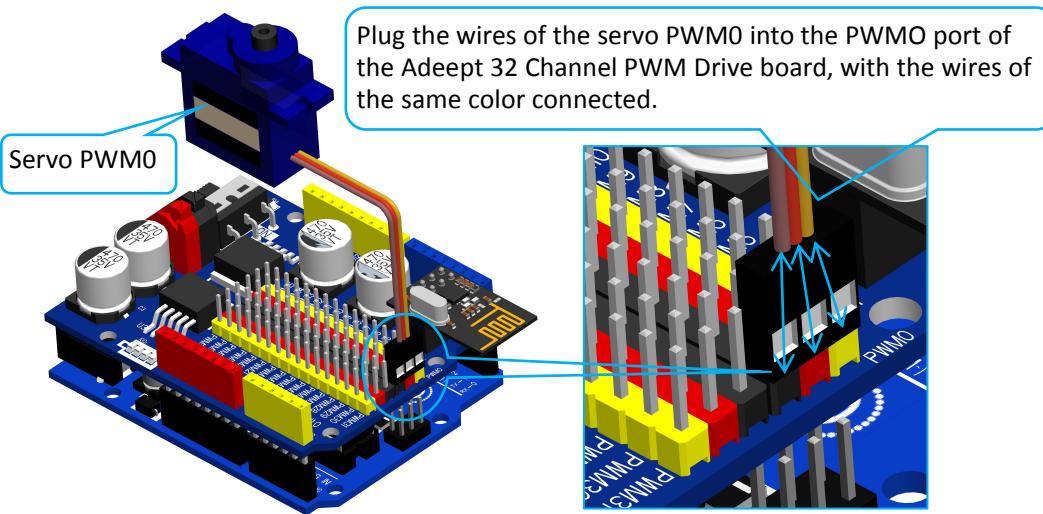
Effect diagram after assembling

After connection is done, switch off the board and install the batteries.

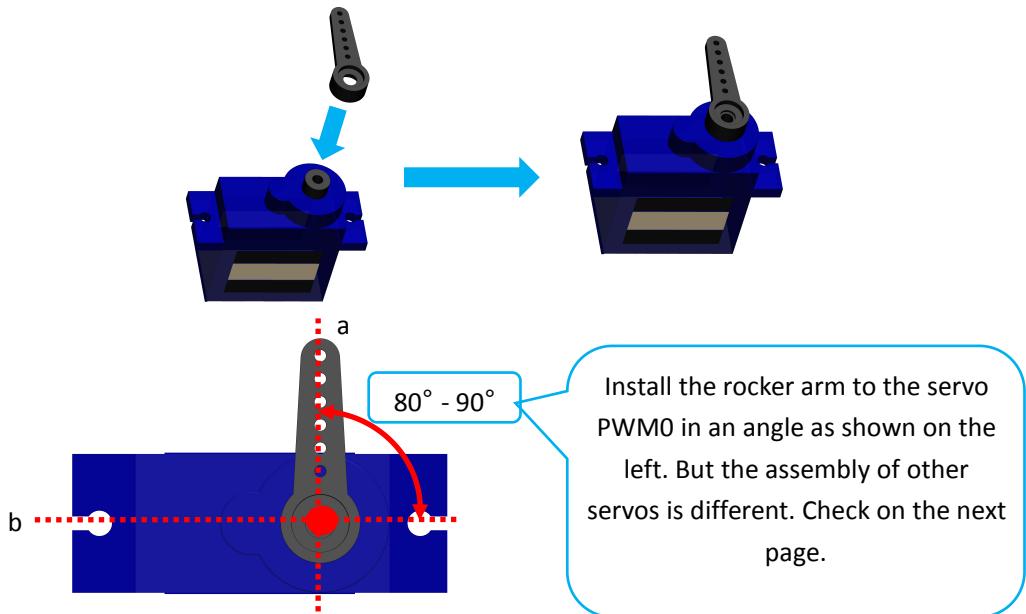


#### F. Assemble the PWM0 servo and rocker arm.

Connect the PWM0 servo and Adeept 32 Channel PWM Drive board.

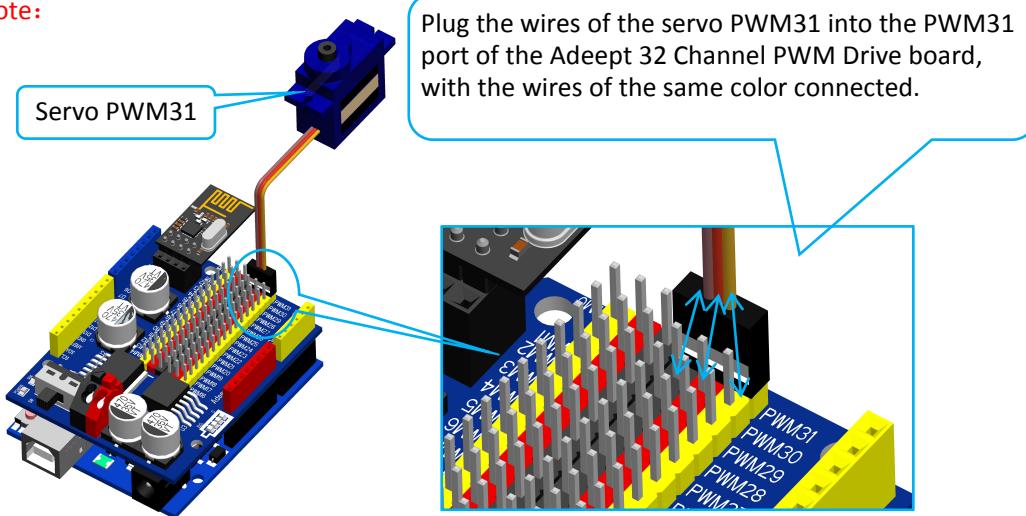


Then switch on the Adeept 32 Channel PWM Drive board. The shaft of the servo will rotate automatically to the initial position. Wait for 3-5s until the shaft stops spinning. Switch off the board and plug in the rocker arm in the angle as shown below. **Note:** During the course, if the servo won't stop rotating, it may be caused by the interference of the NRF24L01 module. The solution is to restart the power. When plugging the rocker arm, it may need a clockwise 1-10 degrees of deviation due to different gear train designs inside the servo.



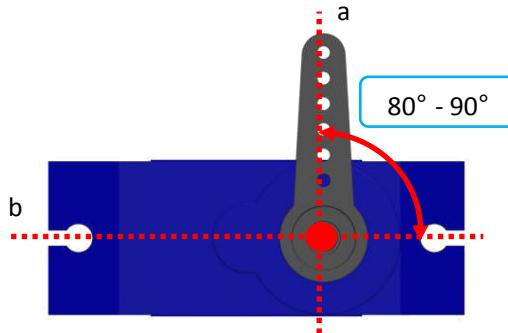
After the rocker arm is assembled, try to gently rotate it and then power on the board again. The angle between the rocker and the middle line of the PWM0 servo (namely line a and b as shown in the figure above) should be  $80^\circ - 90^\circ$ ; a small range of error is acceptable, but if it's too large, it indicates there may be something wrong when plugging in the rocker arm. Then, you need to readjust the servo. Unplug the wires of the PWM0 servo after it's done. After the rocker arm is adjusted well, before the whole robot is completely assembled, you can rotate the rocker anytime but no removing. Otherwise, you need to repeat these steps to adjust again.

**Note:**



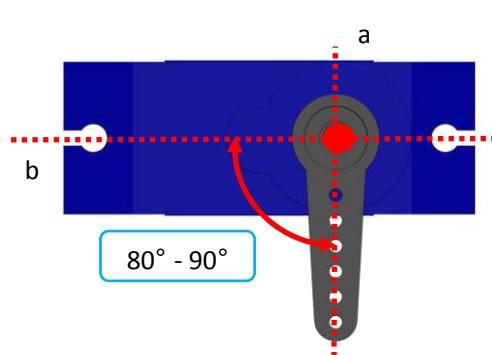
G. Assemble the rest servos and rocker arms as done in step F. Refer to the following angles:

See the following angles for these servos: PWM0, PWM3, PWM6, PWM17, PWM20, and PWM23:



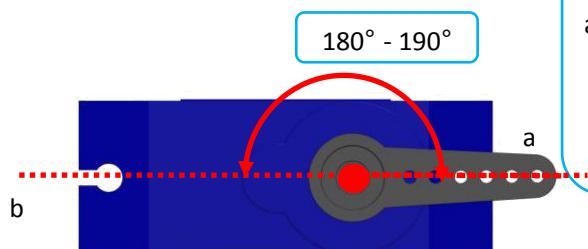
If it's not exactly 90° assembled, rotating the rocker clockwise in a small angle is also acceptable.

See the angles below for these 6 servos: PWM1, PWM4, PWM7, PWM16, PWM19, and PWM22:



If it's not exactly 90° assembled, rotating the rocker clockwise in a small angle is also acceptable.

See the angles below for these 7 servos: PWM2, PWM5, PWM8, PWM18, PWM21, PWM24, and PWM31:

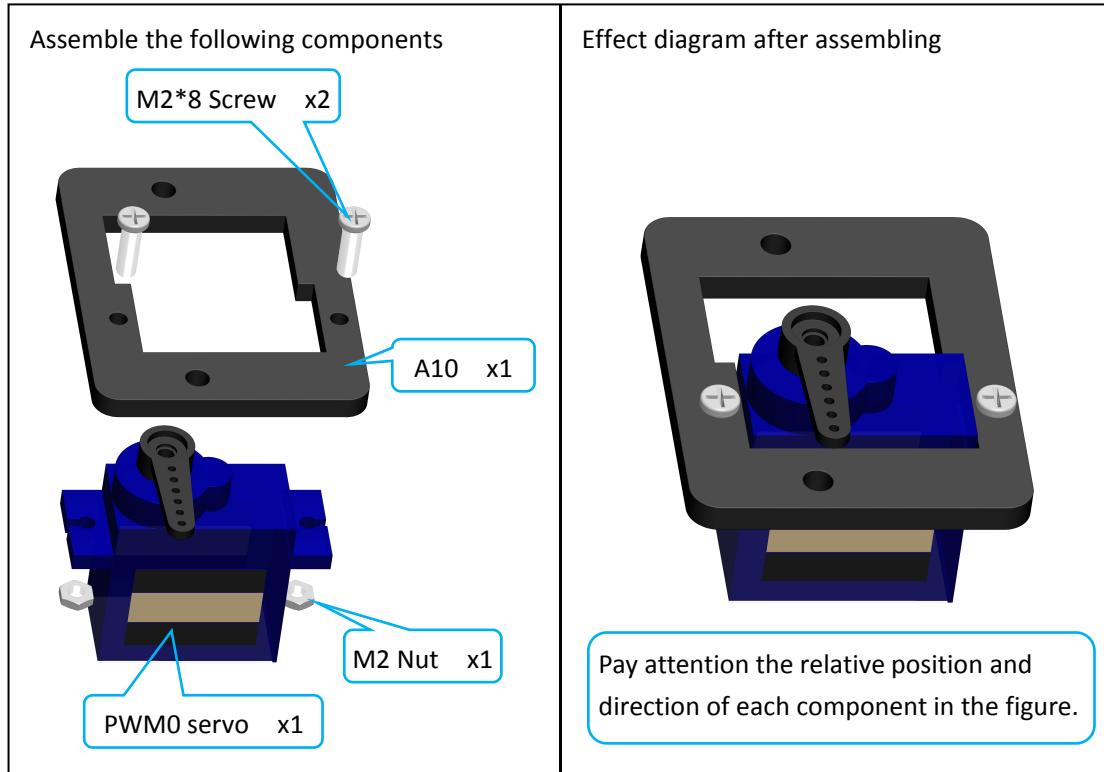


If it's not exactly 180° assembled, rotating the rocker clockwise in a small angle is also acceptable.

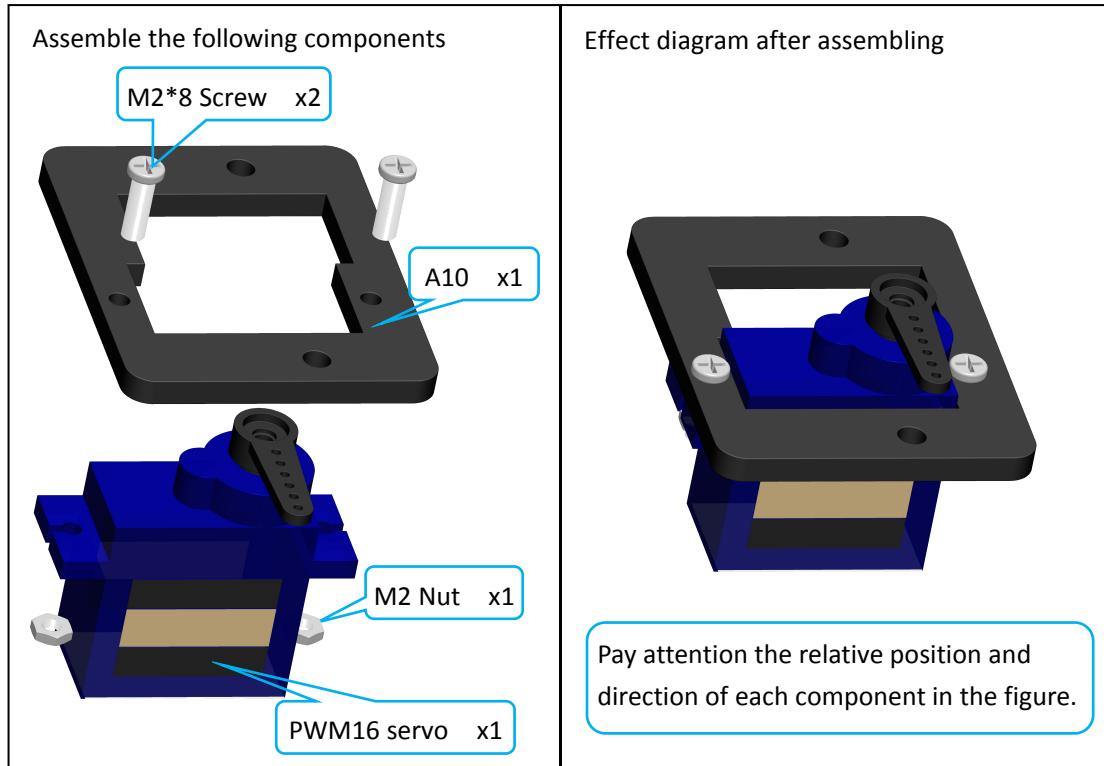
After all the 19 servos are assembled, place them aside and remove the batteries from the holder. And put Adeept 32 Channel PWM Drive, Adeept UNO R3 and the 18650x2 Battery Holder beside. In case of rocker arm falling out, you can also fix them by the M2\*4 Screw packaged with the servos and unscrew them when they are mounted later. It is very important not to keep the rocker arms well assembled on the servo.

## Left and Right Legs

- A. Take 3 pieces of A10 plates and assemble them to the PWM0, PWM3 and PWM6 servos.



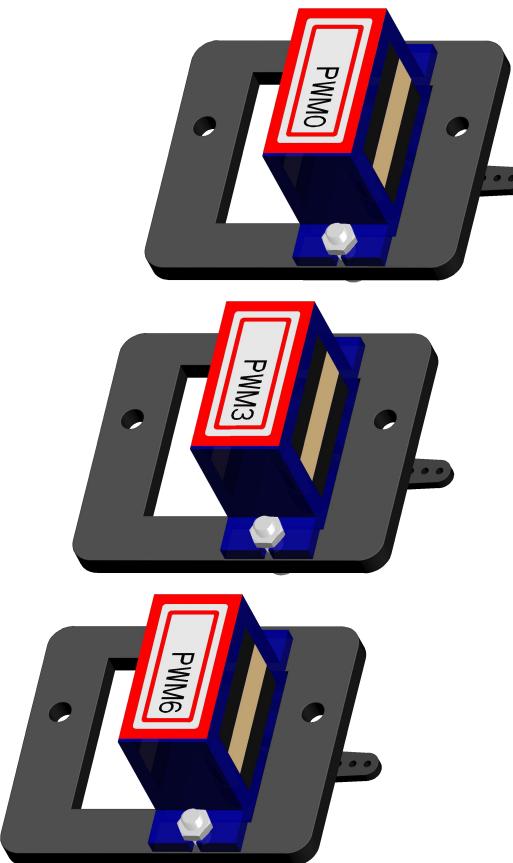
- B. Then take another 3 A10 plates and mount them to the PWM16, PWM19, and PWM22.



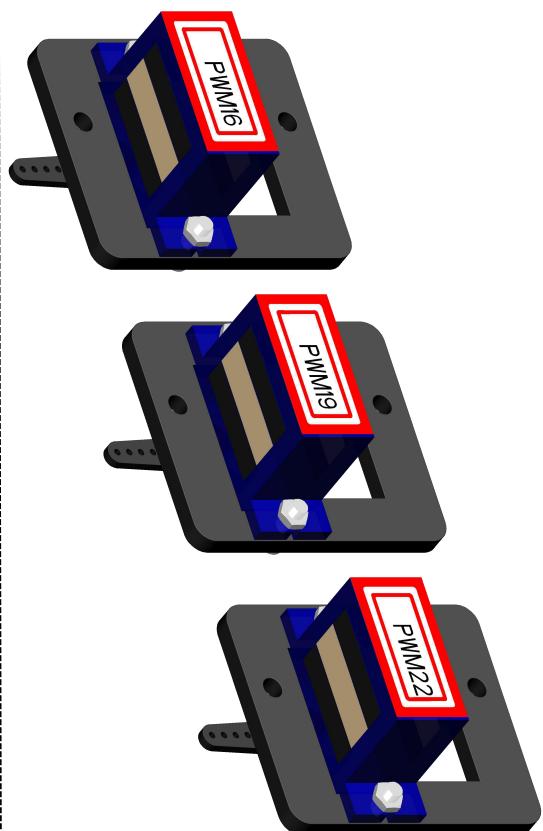
Pay attention the relative position and direction of each component in the steps A and B.

The effect diagram after the 6 servos are all assembled:

Servos assembled in step A

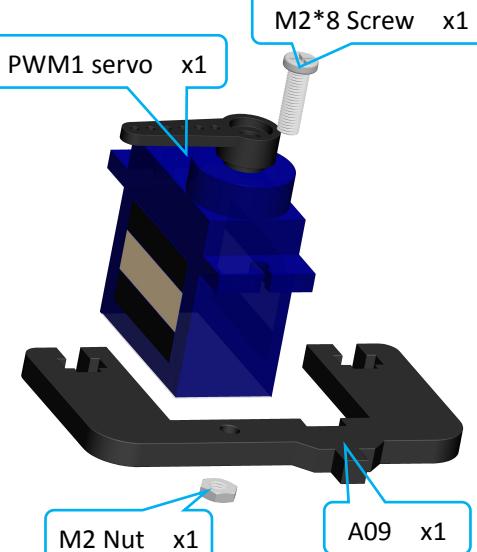


Servos assembled in step B

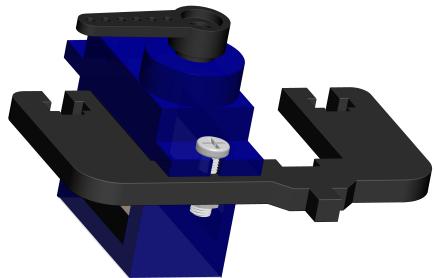


C. Take 3 pieces of A09 plates and assemble them to the PWM1, PWM4, and PWM7 servos.

Assemble the following components

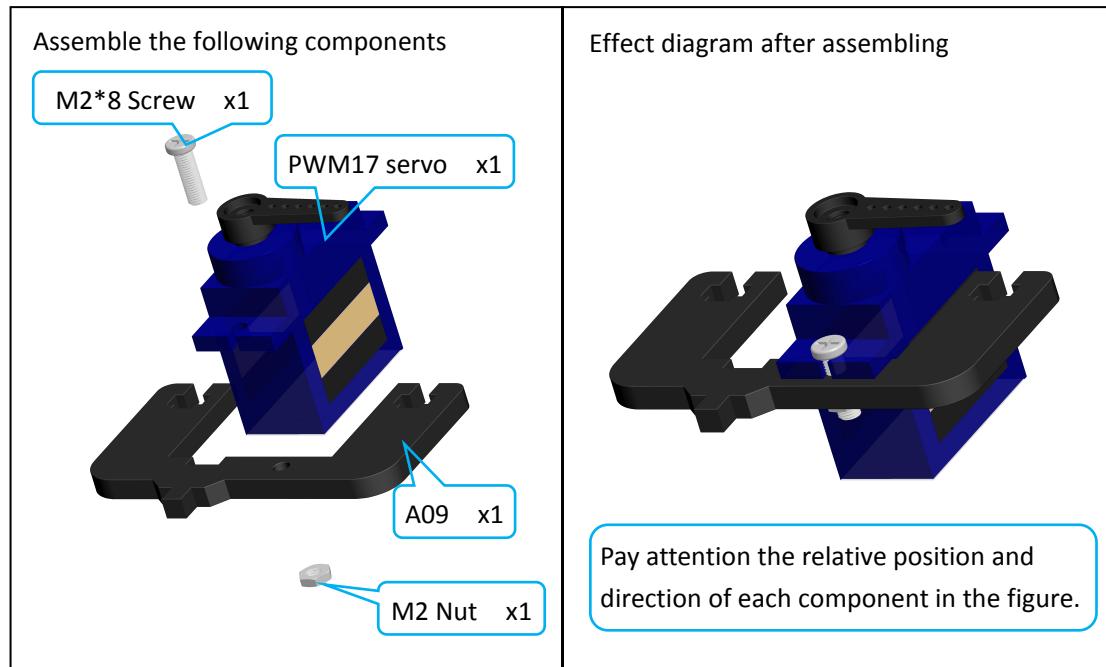


Effect diagram after assembling

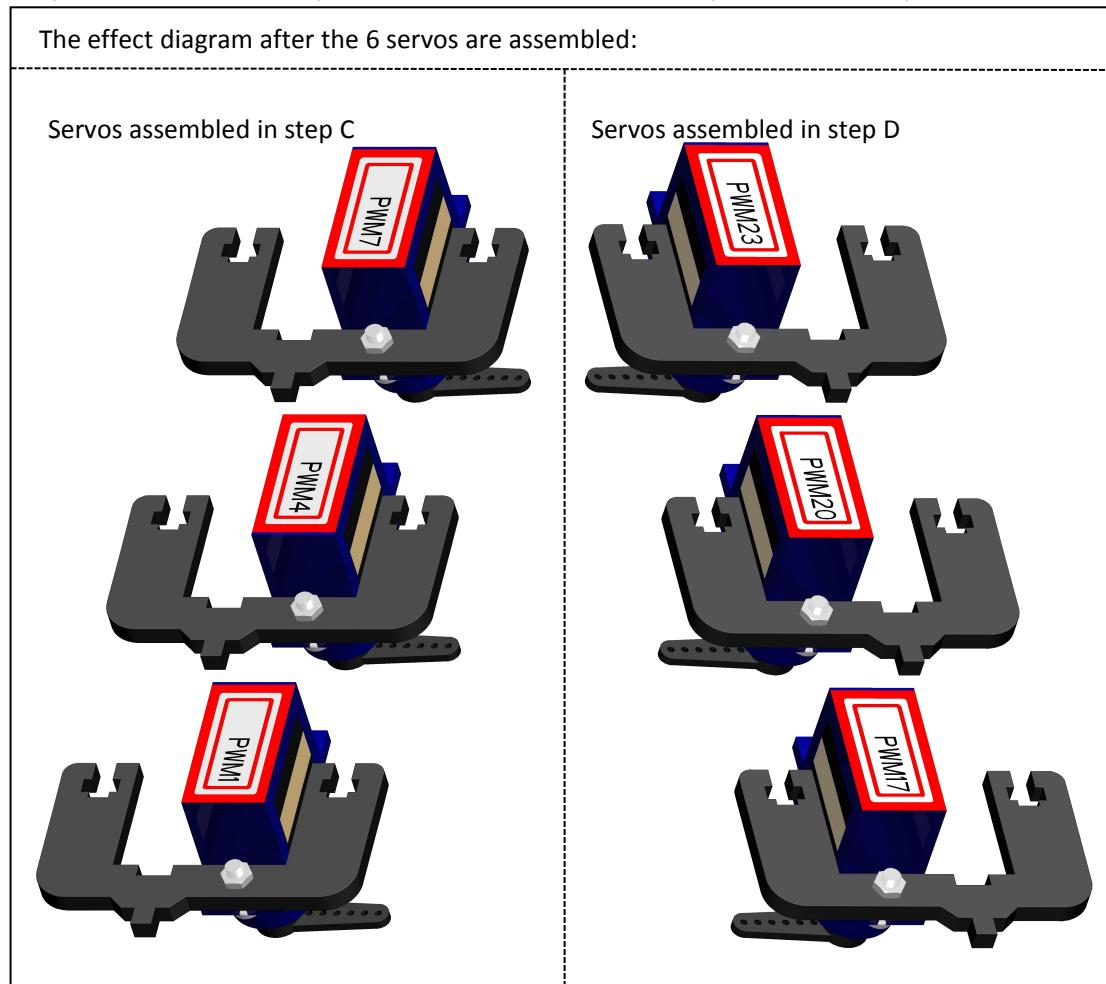


Pay attention the relative position and direction of each component in the figure.

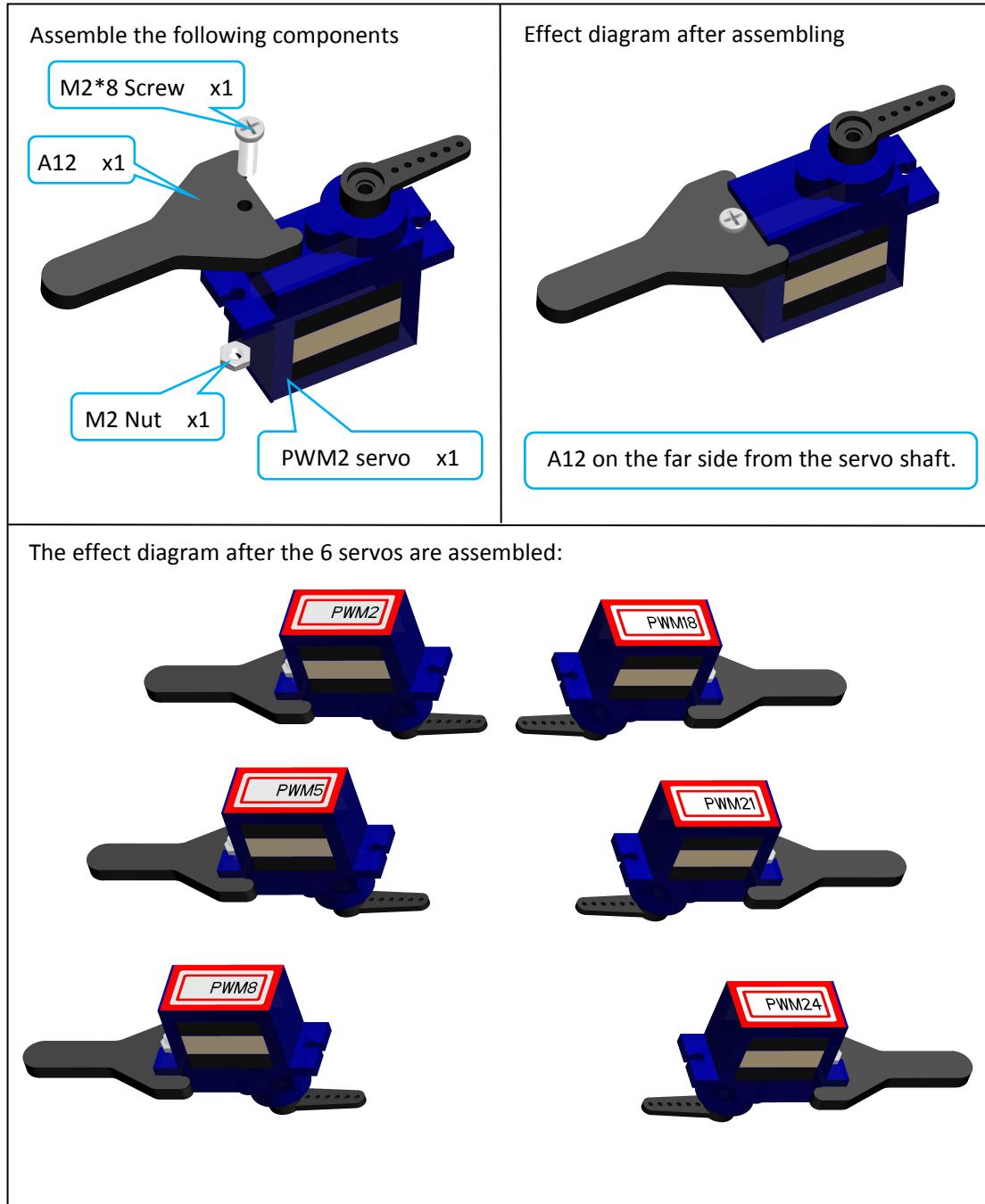
- D. Then take another 3 A09 plates and mount them to the PWM17, PWM20, and PWM23 servos.



Pay attention the relative position and direction of each component in the steps C and D.



- E. Take 6 pieces of A12 plates and assemble them to the PWM2, PWM5, PWM8, PWM18, PWM21, and PWM24 servos.



After all servos are assembled, divide the 18 servos into 6 groups (corresponding to 6 legs) for the convenient assembly later. They are grouped as:

Group #1: servo PWM0, PWM1, and PWM2;

Group #2: servo PWM3, PWM4, and PWM5;

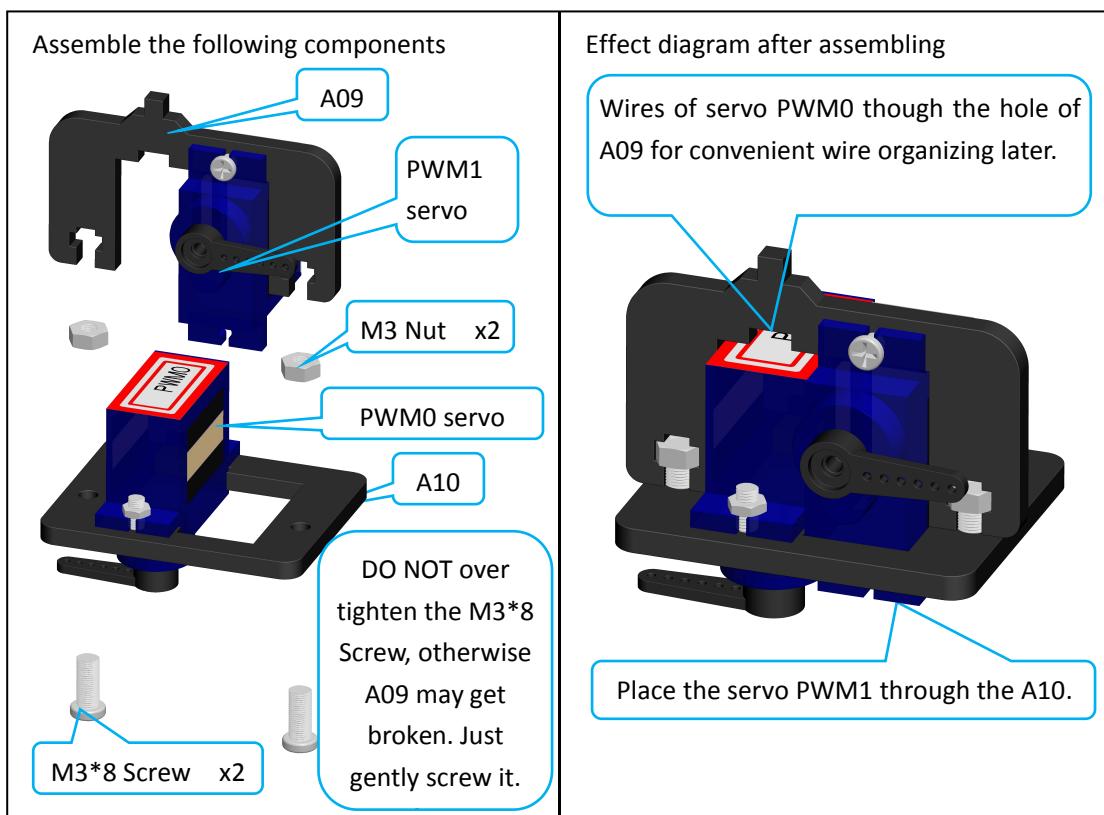
Group #3: servo PWM6, PWM7, and PWM8;

Group #4: servo PWM16, PWM17, and PWM18;

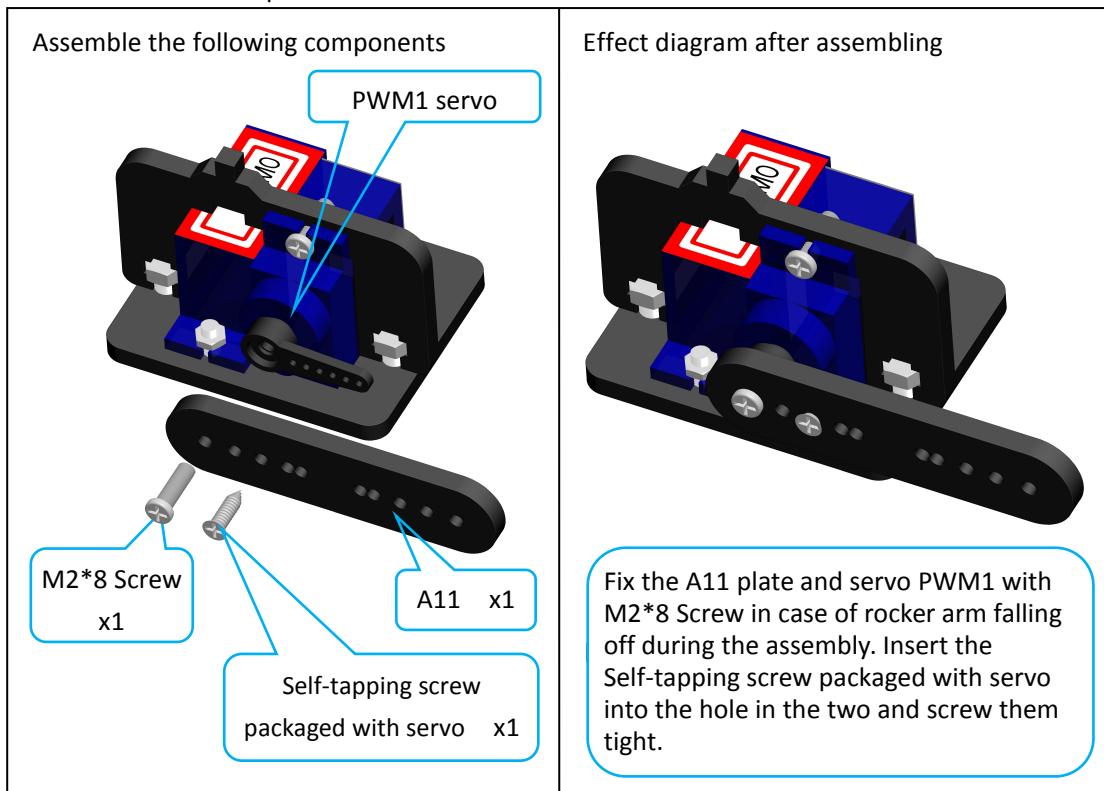
Group #5: servo PWM19, PWM20, and PWM21;

Group #6: servo PWM22, PWM23, and PWM24;

F. Combine the servo PWM0 and PWM1.

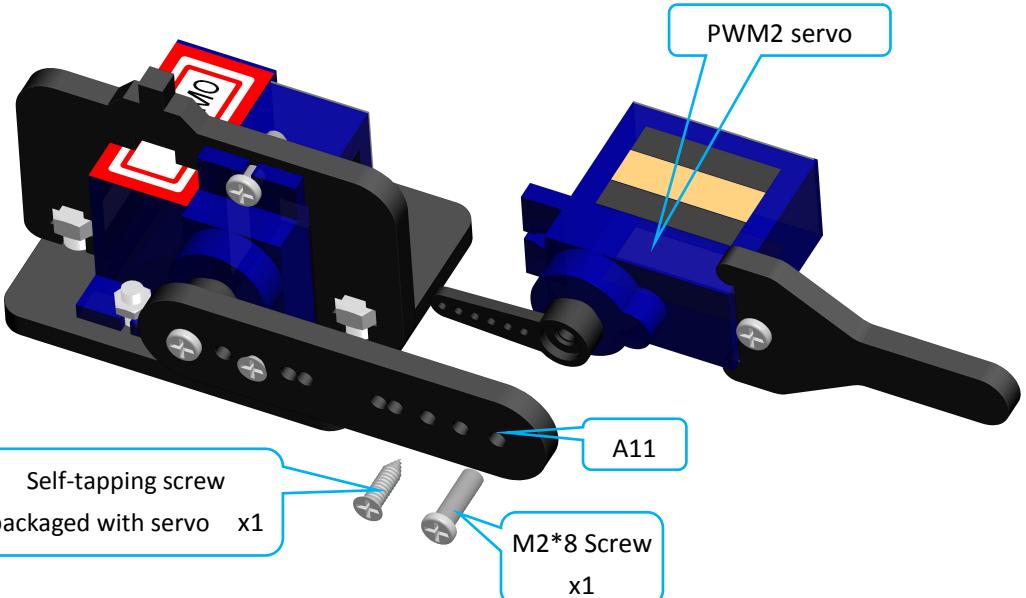


G. Assemble the A11 plate to the rocker arm of the servo PWM1.

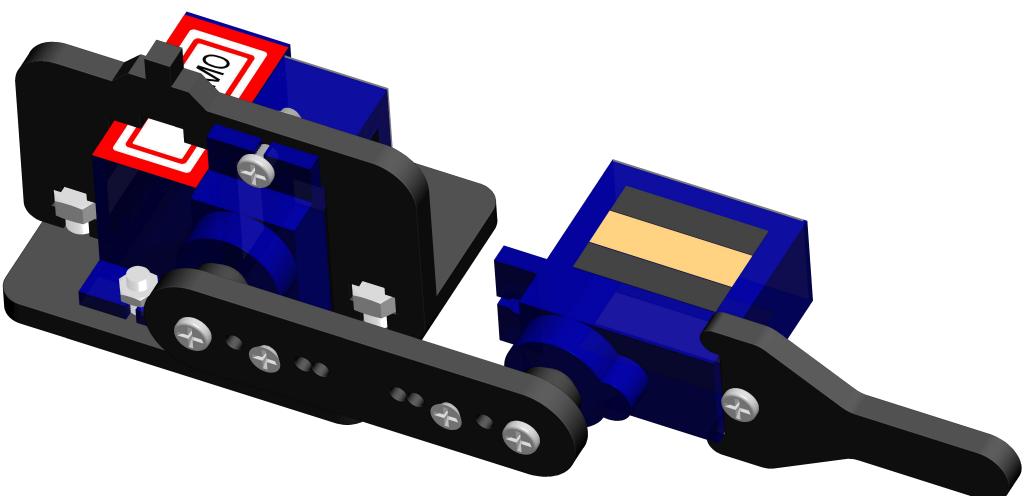


H. Then assemble the servo PWM2 on the A11 plate.

Assemble the following components



Effect diagram after assembling



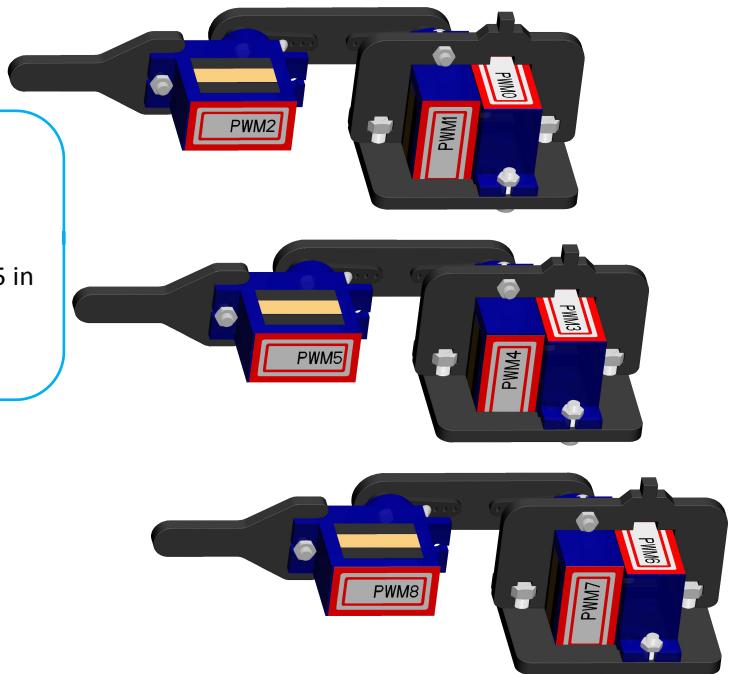
So now one leg is done assembled.

I. Assemble the rest 5 legs in the similar way.

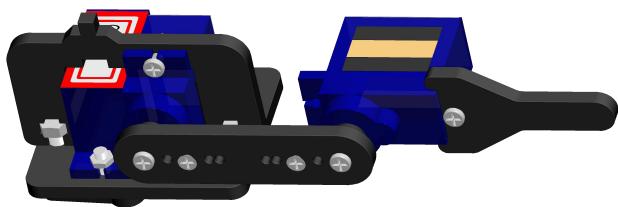
The assembly of these 6 legs is almost the same – using the same screws. Only note that the left and right legs should be symmetric, so in step F, the servo on A09 should go through A10.

Back view of the left 3 legs:

Servos of these 3 legs are:  
servo PWM0, PWM1, and  
PWM2 in one leg; servo  
PWM3, PWM4, and PWM5 in  
one; PWM6, PWM7, and  
PWM8 in one.

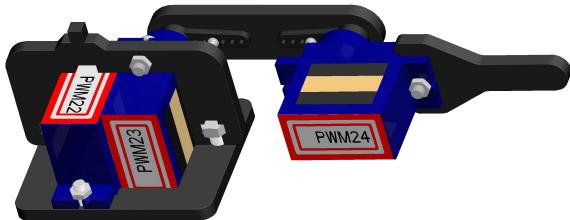
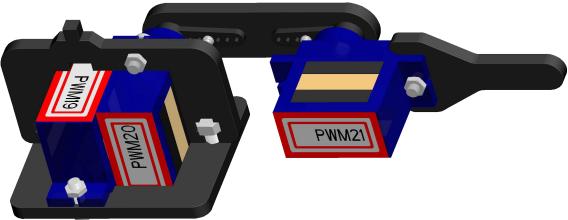
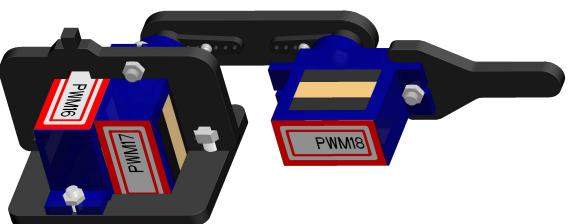


Front view of these left 3 legs:



Back view of the right 3 legs:

Servos of these 3 legs on the right are: PWM16, PWM17, and PWM18 in one leg; PWM19, PWM20, and PWM21 in one; PWM22, PWM23, and PWM24 in one.



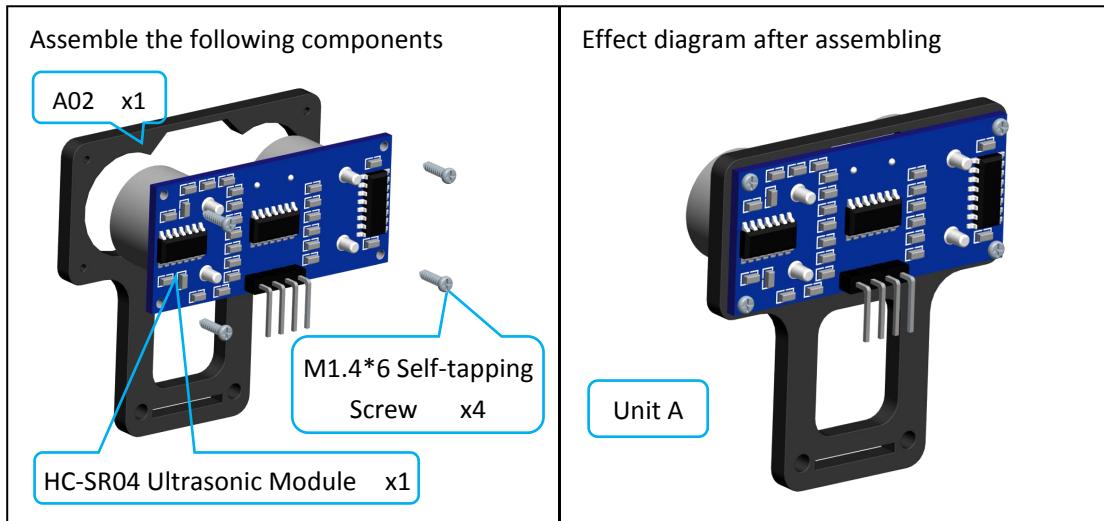
Front view of the right 3 legs:



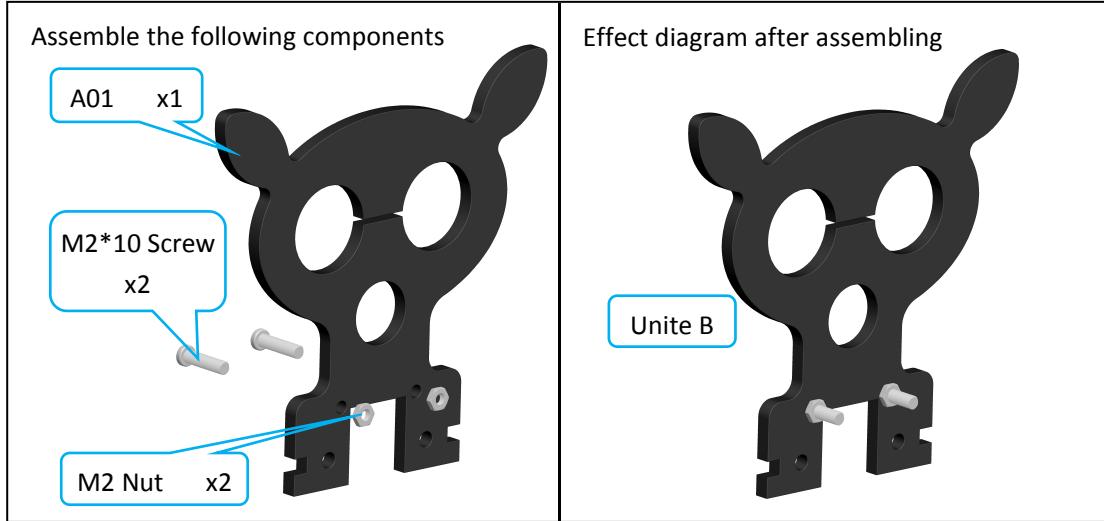
Now all 6 legs are assembled. Just place them aside and finish the middle part first.

## Middle Body

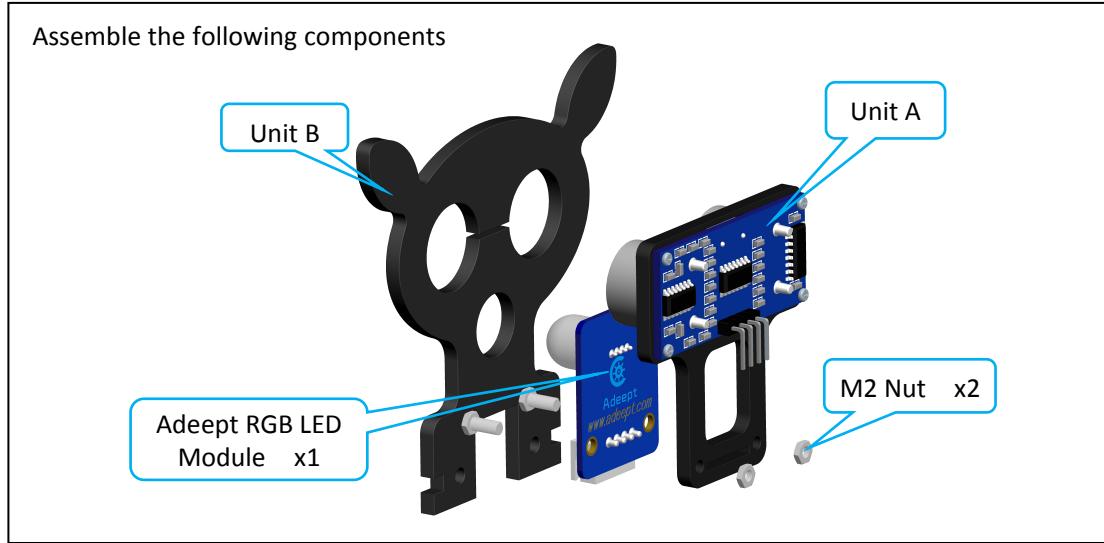
A. Assembly the ultrasonic module.



B. Fasten two M2\*10 Screw onto the plate A04.

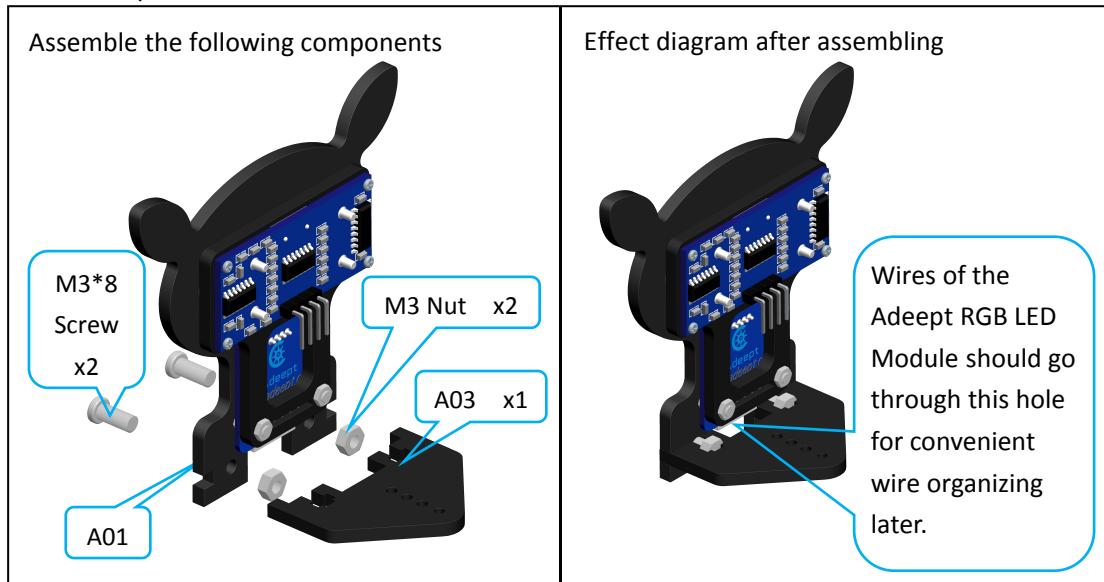


C. Assemble the two units completed in step A and B to the Adeept RGB LED Module.

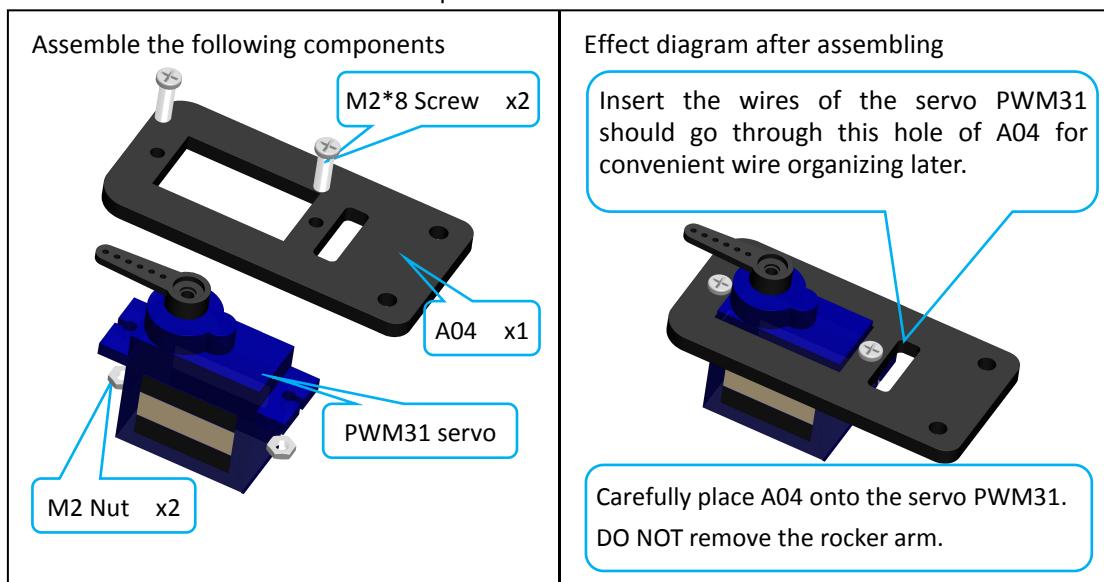




D. Fix the plate A03 onto A01.

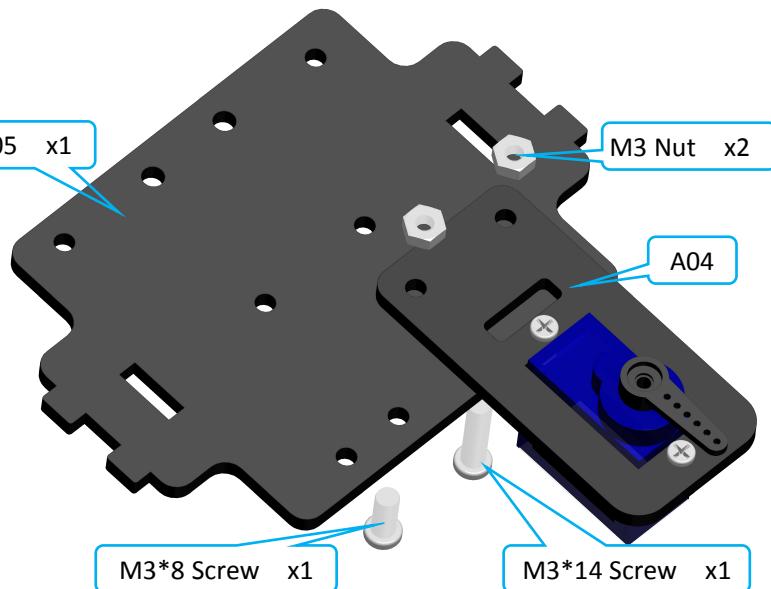


E. Fasten the servo PWM31 and the plate A04.



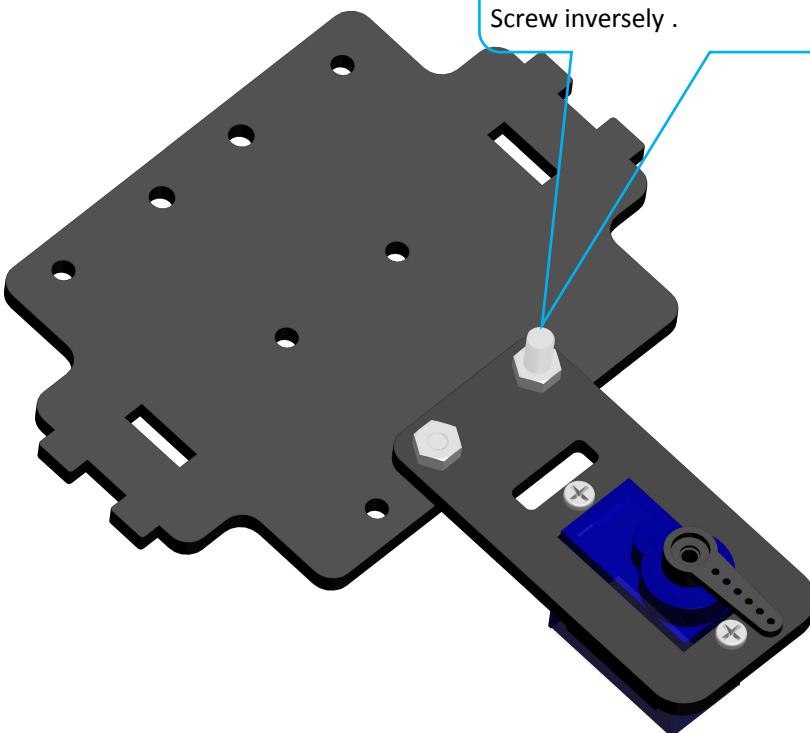
F. Assembly an A05 plate and A04.

Assemble the following components



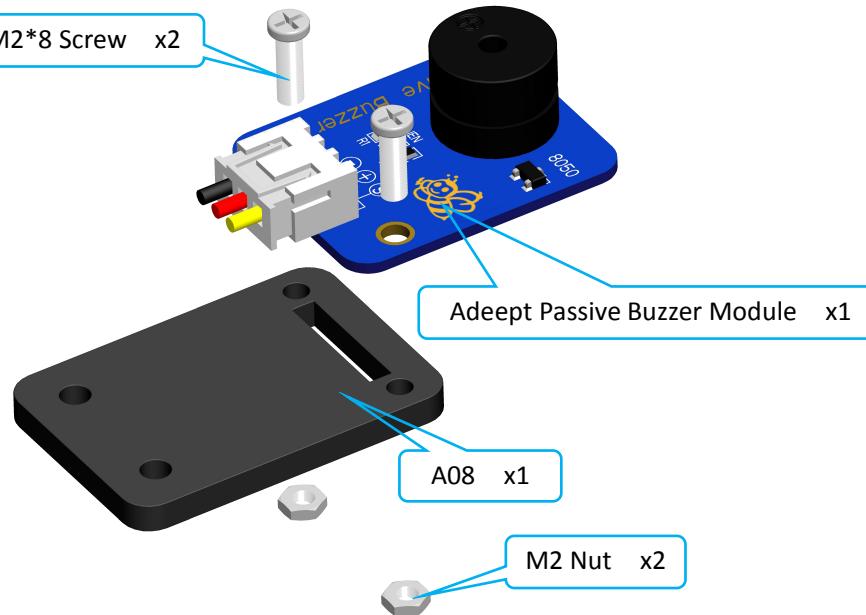
Effect diagram after assembling

Strictly follow the instructions to assemble and DO NOT fix M3\*8 Screw and M3\*14 Screw inversely .

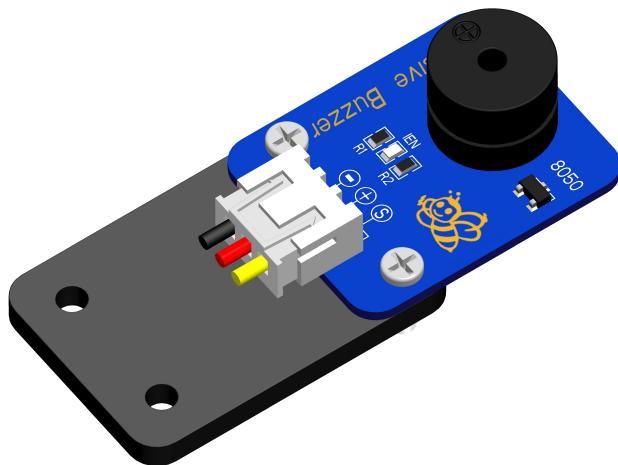


G. Fasten the Adeept Passive Buzzer Module and the A08 plate.

Assemble the following components

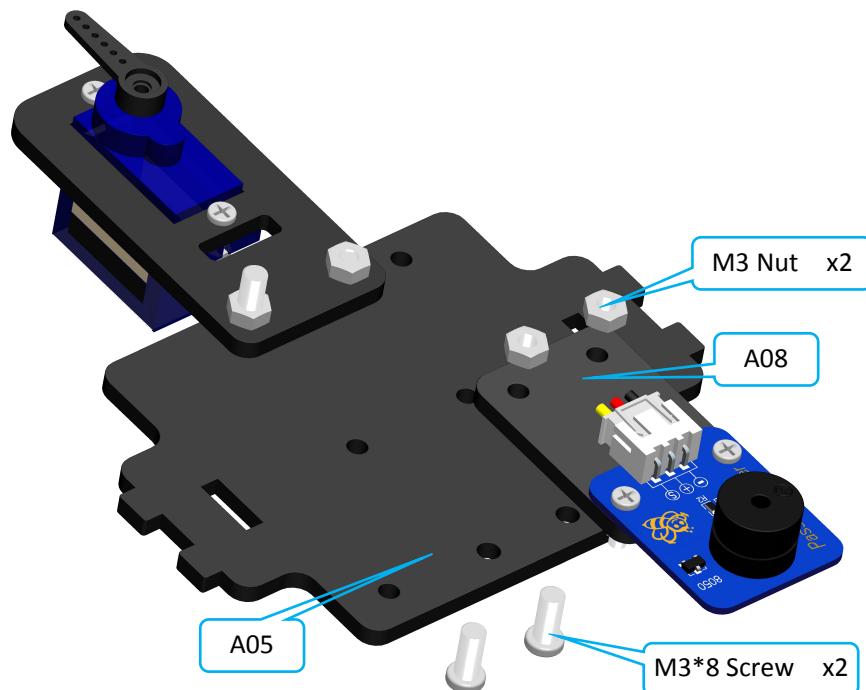


Effect diagram after assembling

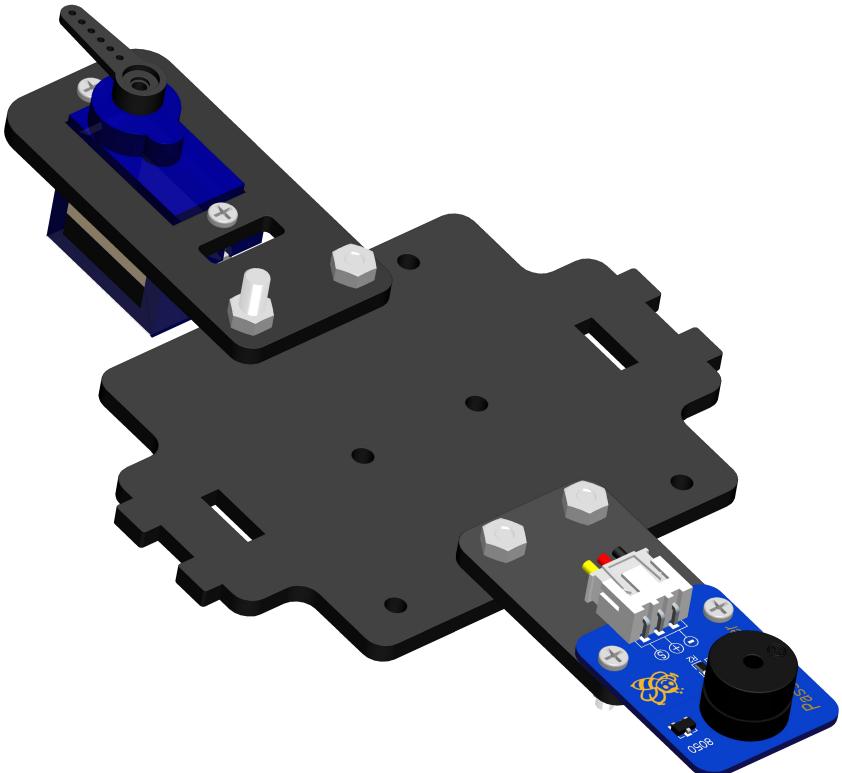


H. Then fasten and fix the A08 and A05 plates.

Assemble the following components



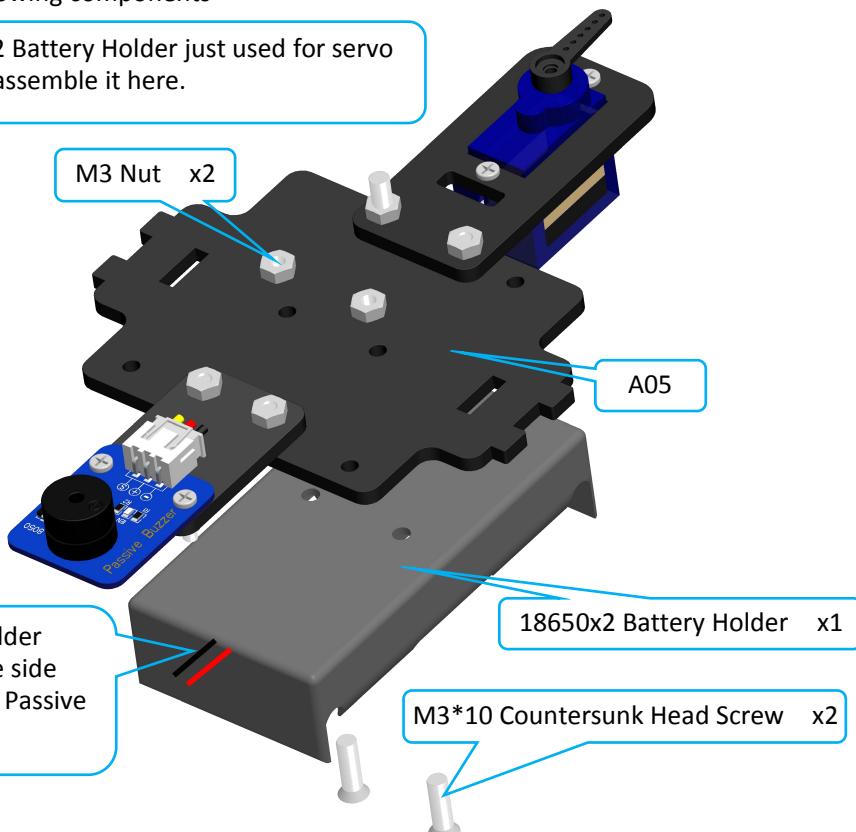
Effect diagram after assembling



I. Assemble the 18650x2 Battery Holder onto the A05 plate.

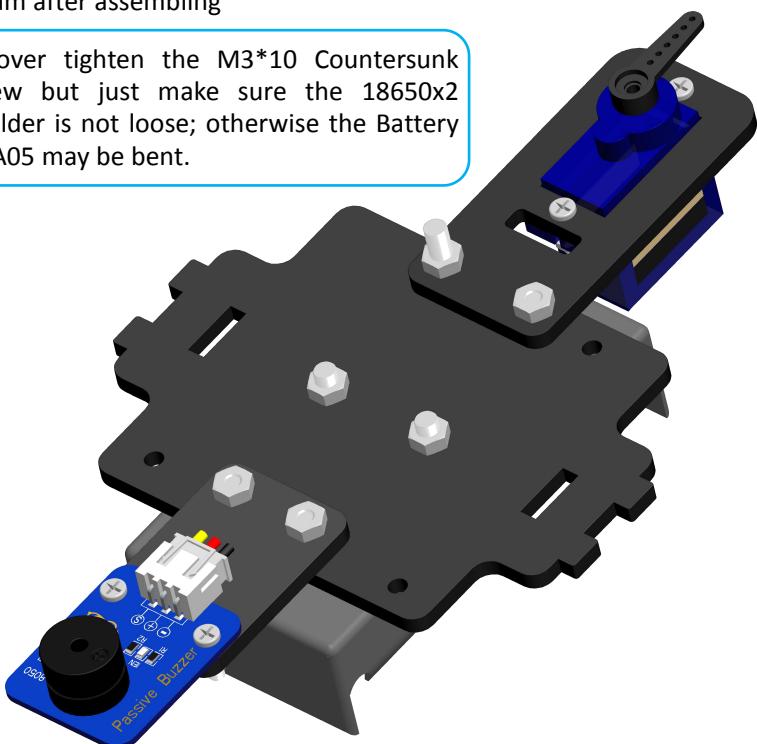
Assemble the following components

Take the 18650x2 Battery Holder just used for servo adjustment and assemble it here.



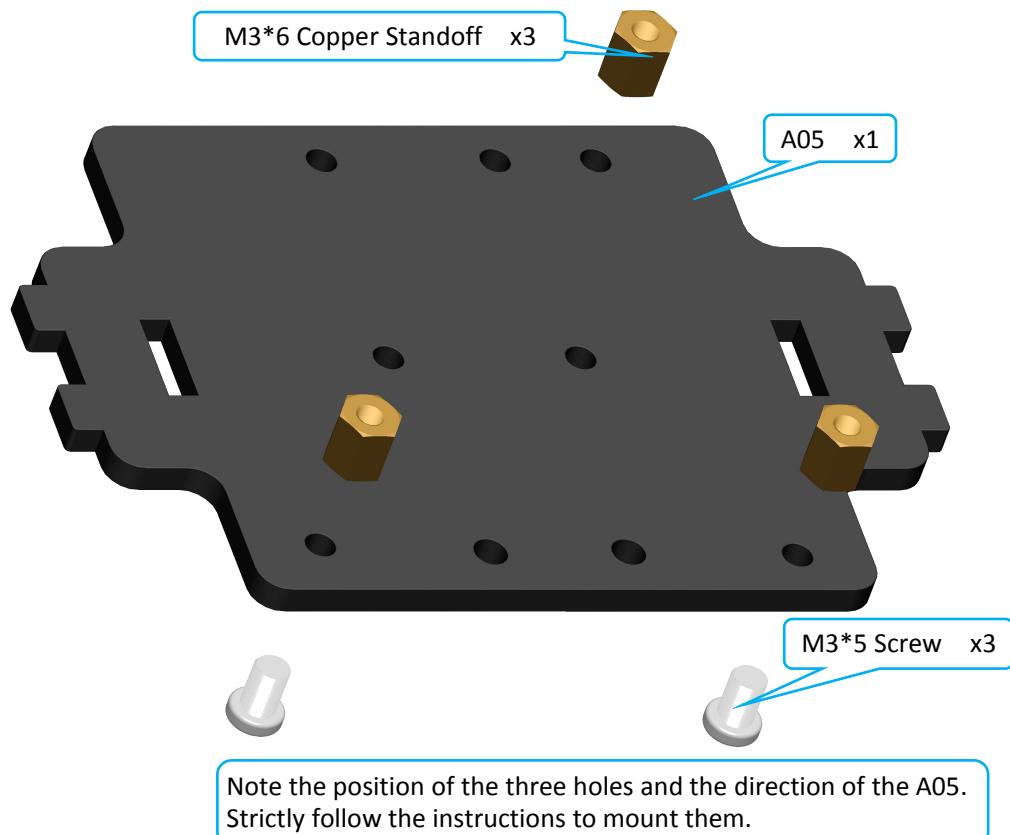
Effect diagram after assembling

DO NOT over tighten the M3\*10 Countersunk Head Screw but just make sure the 18650x2 Battery Holder is not loose; otherwise the Battery Holder or A05 may be bent.

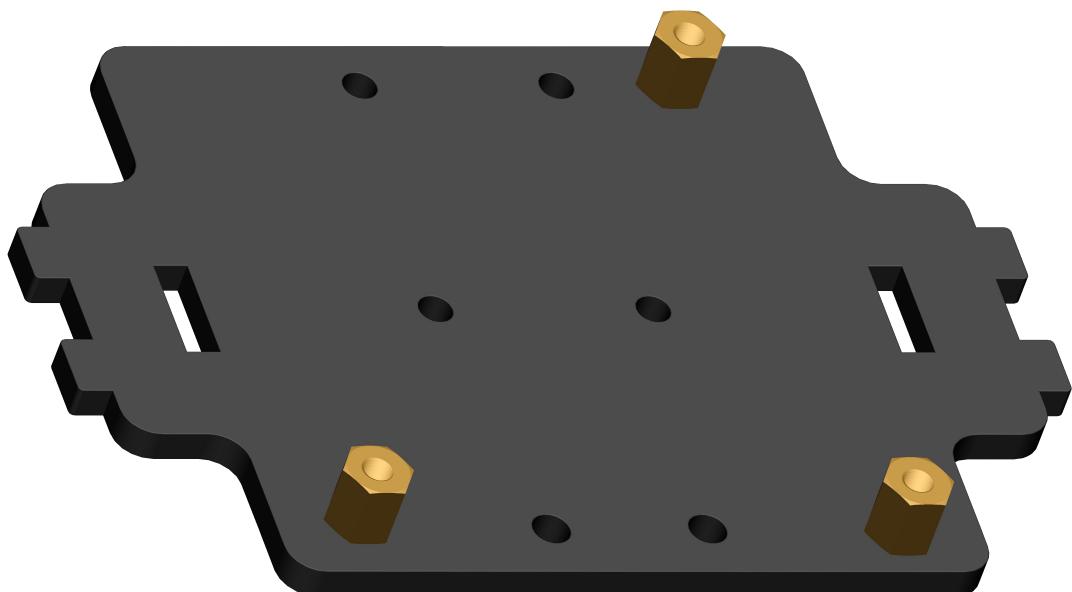


J. Take an A05 plate and fix 3 M3\*6 Copper Standoffs onside.

Assemble the following components



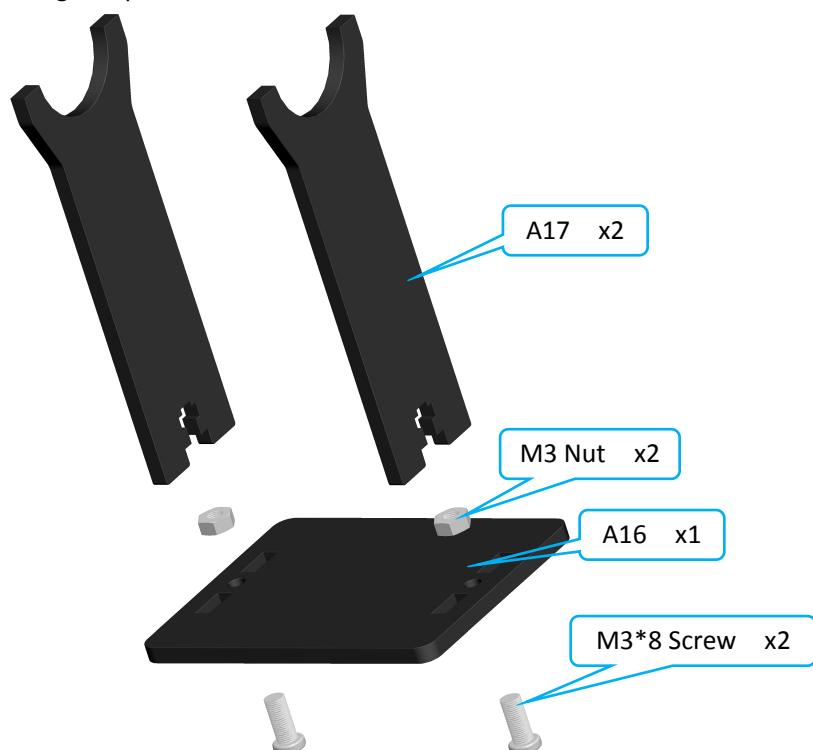
Effect diagram after assembling



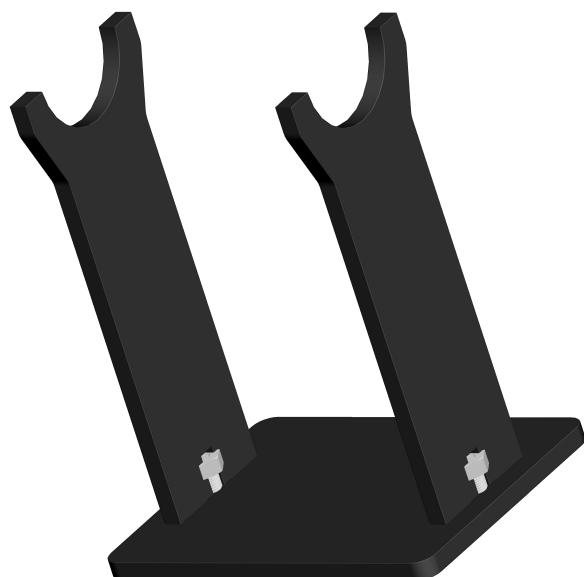
## Prop

### A. Fasten an A17 plate and an A16.

Assemble the following components



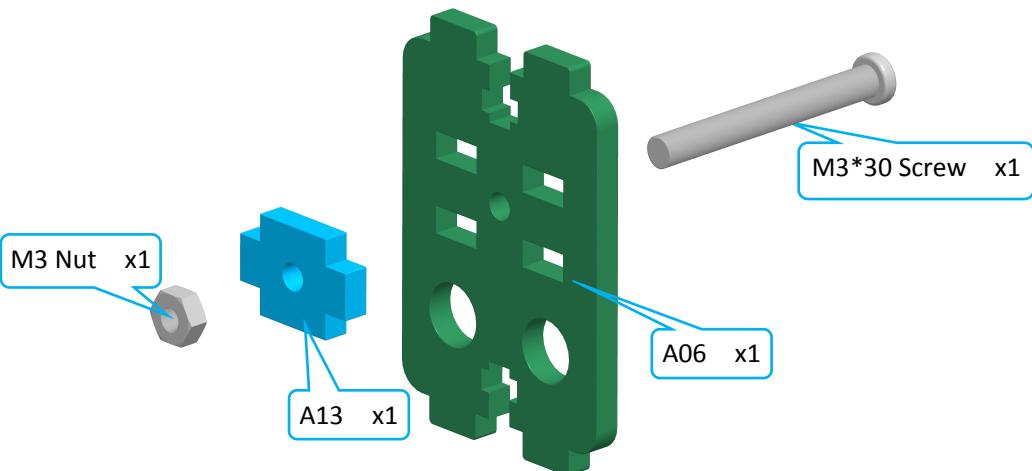
Effect diagram after assembling



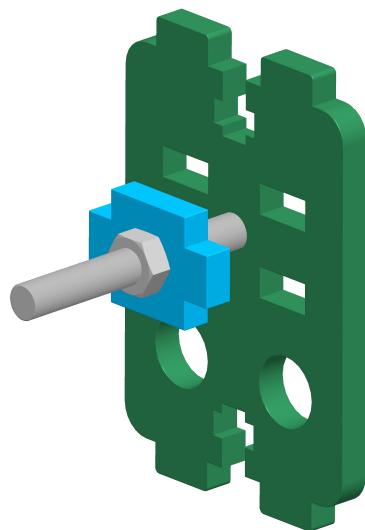
## Final Part

- A. Insert the M3\*30 Screw through the A06 plate, A13 and then the nut (2 groups).

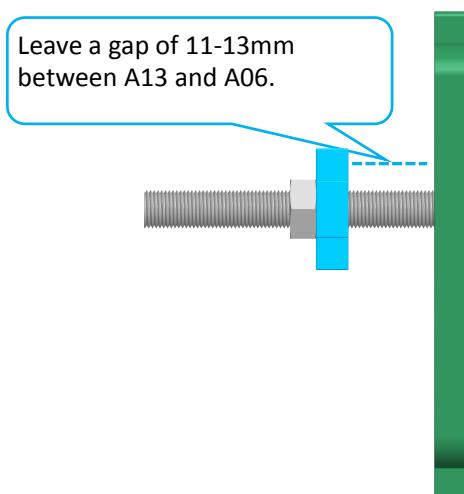
Assemble the following components



Effect diagram after assembling

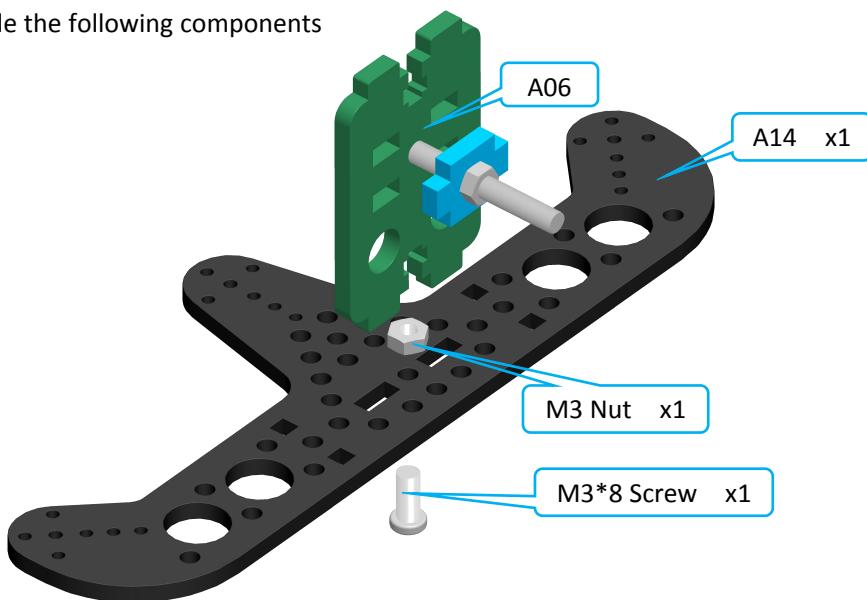


View from the side:

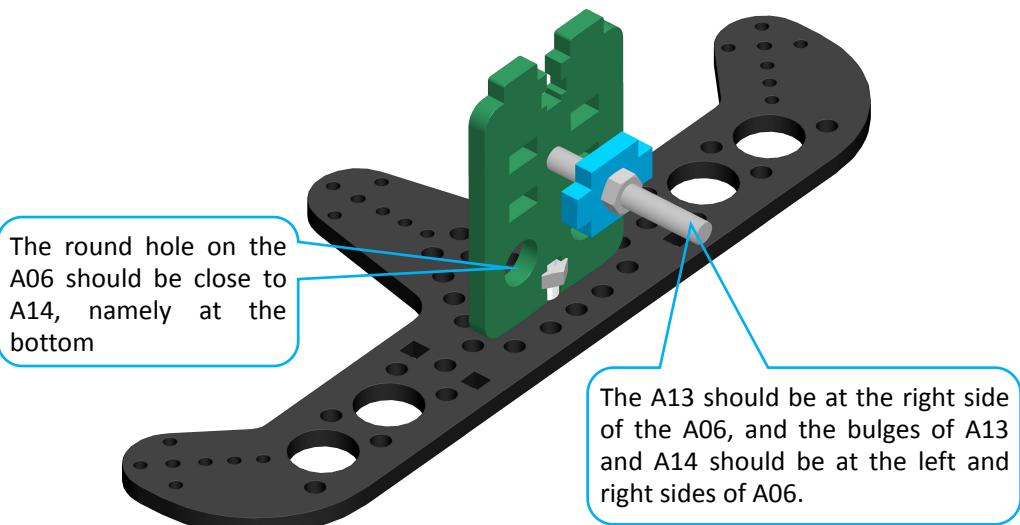


B. Assemble the A06 and A14 plates (two groups).

Assemble the following components

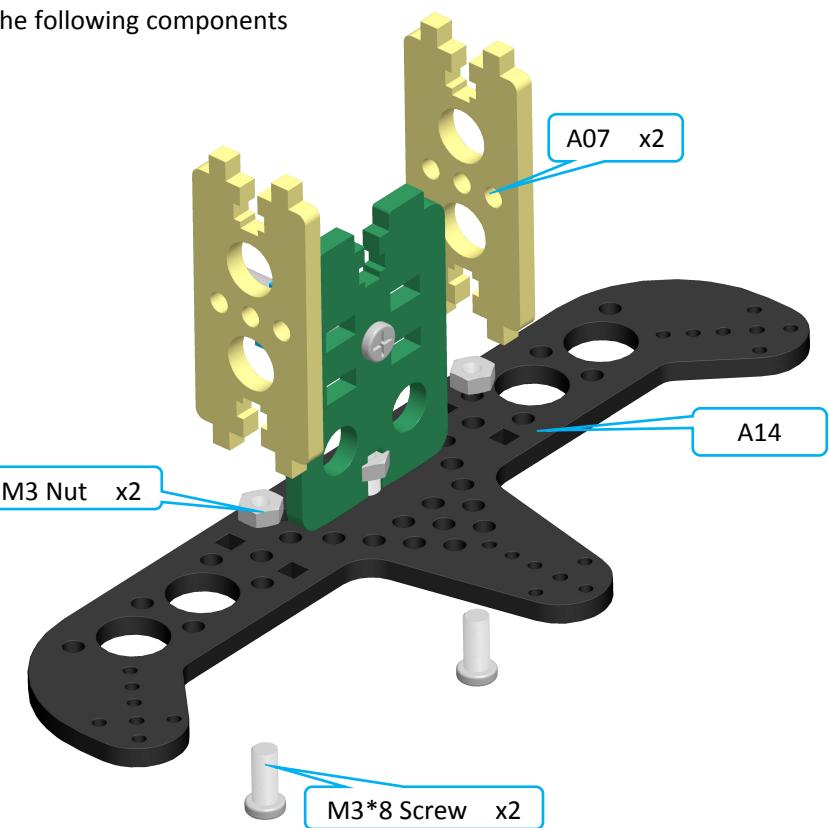


Effect diagram after assembling

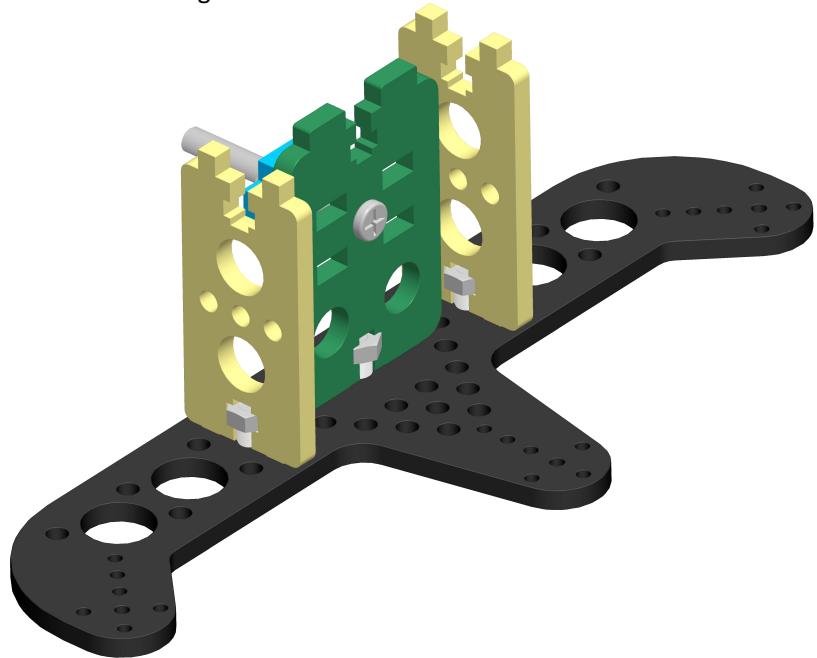


C. Assemble 2 A07 plates onto the A14 (2 groups).

Assemble the following components

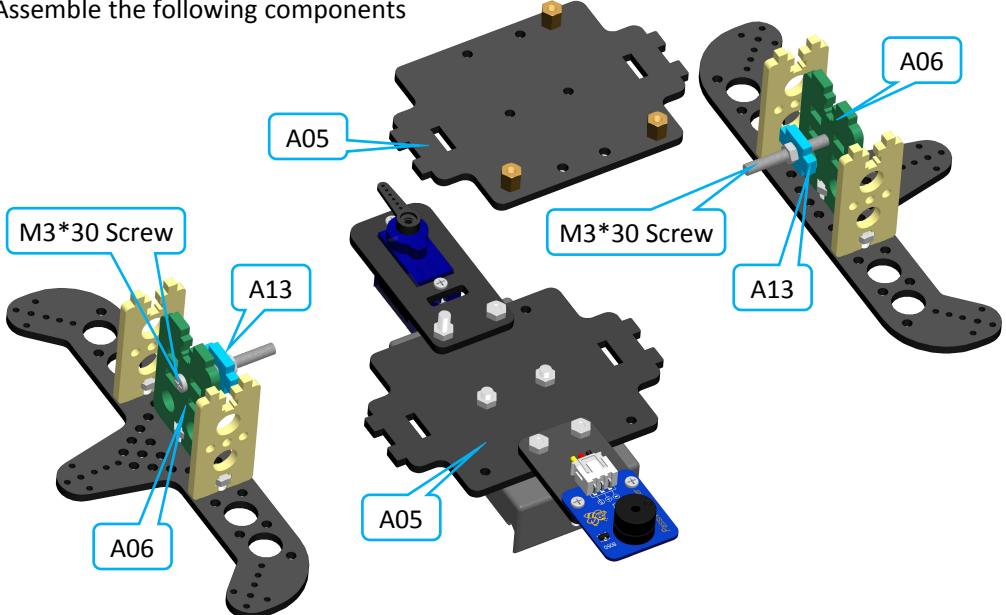


Effect diagram after assembling



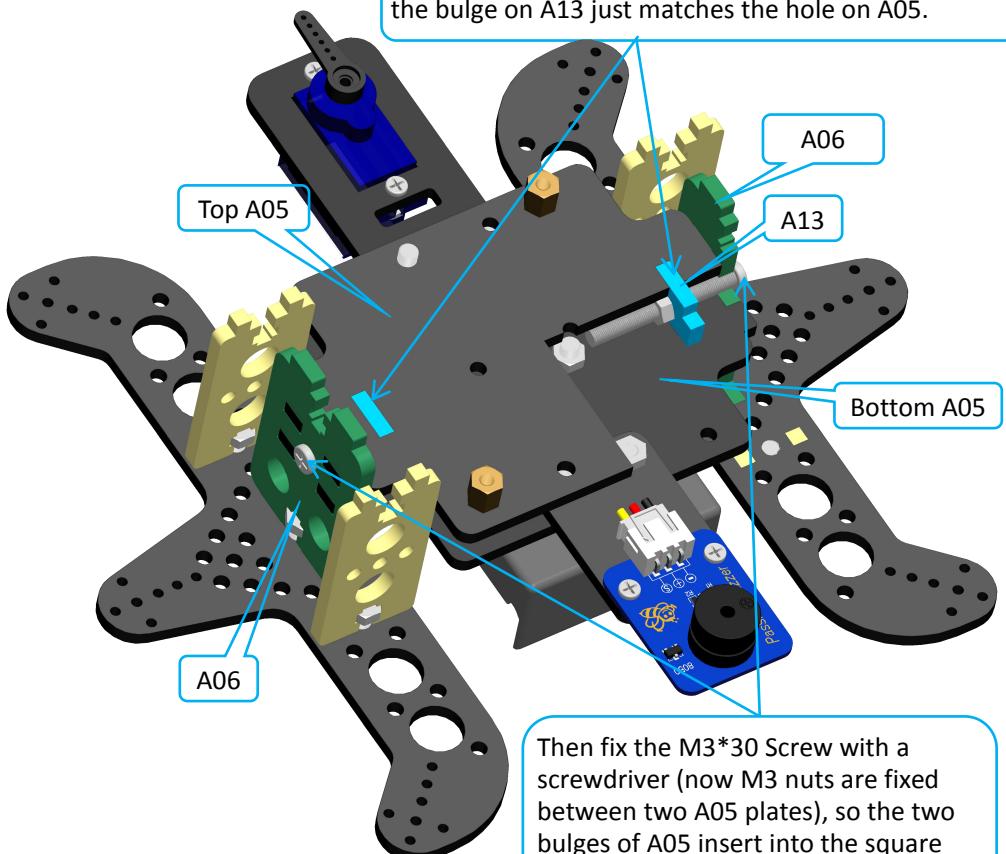
D. Combine the middle body parts and those finished in the previous steps:

Assemble the following components

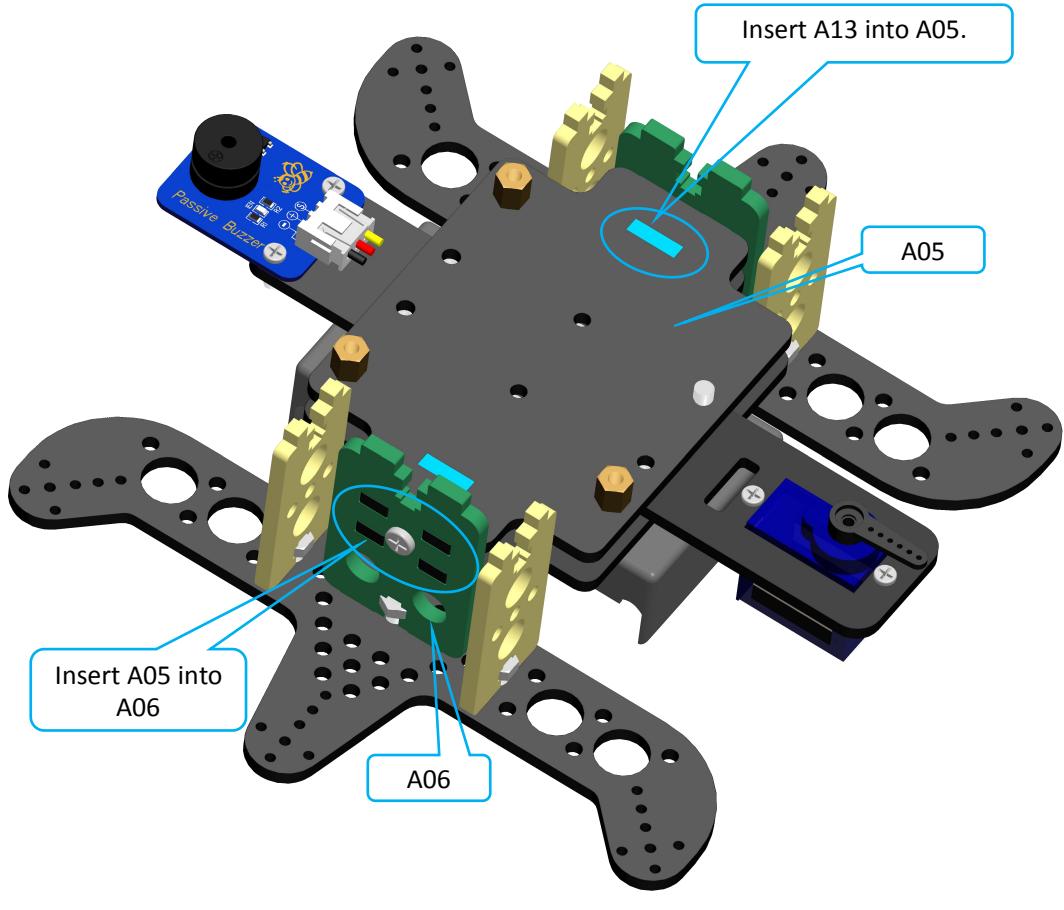


See the following figure for the section of the top A05, right A06, and A07:

Fix A13 with the square holes on the two A05 plates, and the bulge on A13 just matches the hole on A05.

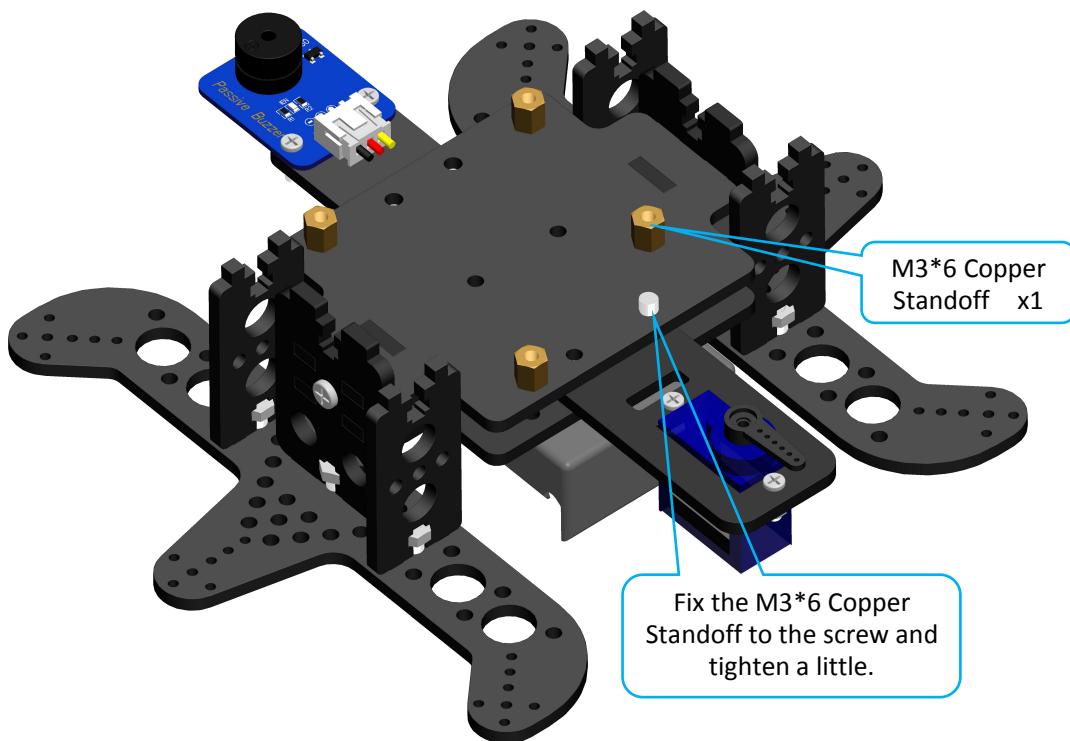


Effect diagram after assembling

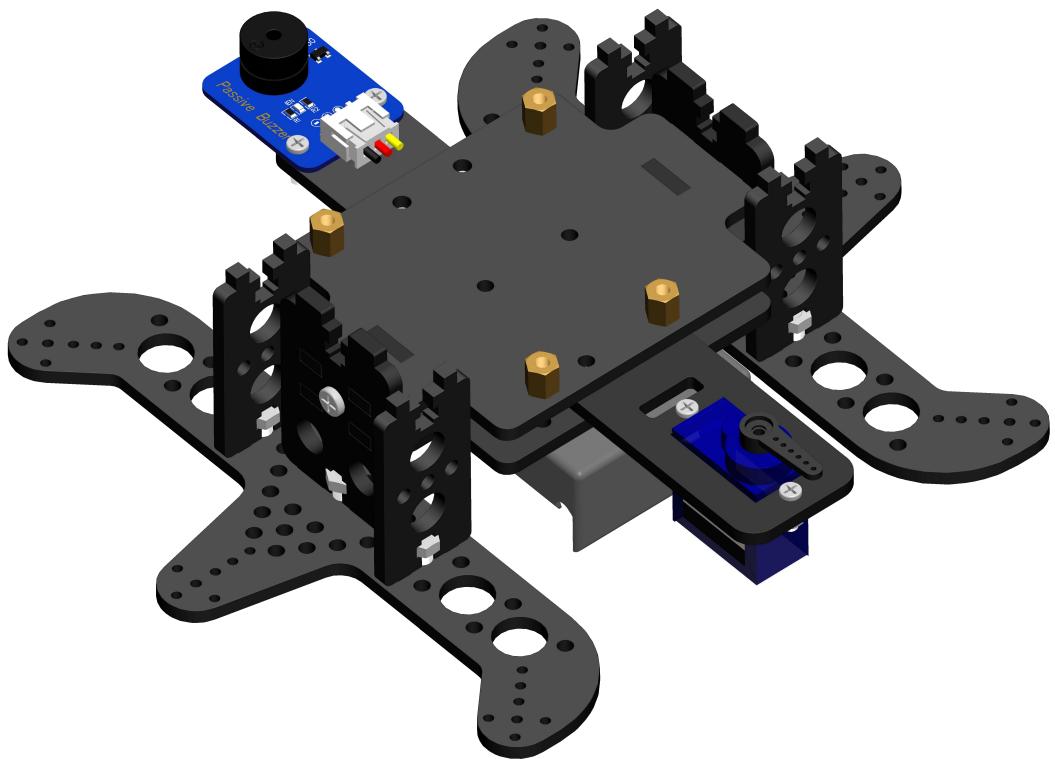


E. Fix an M3\*6 Copper Standoff.

Assemble the following components

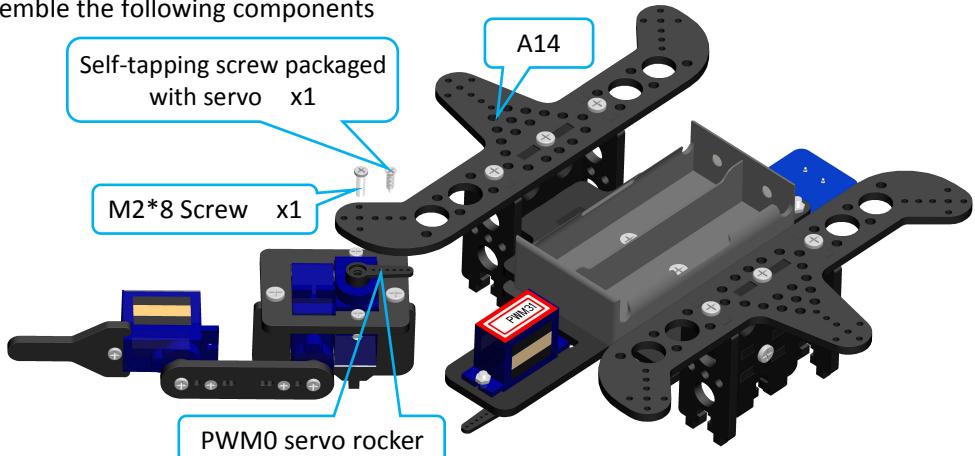


Effect diagram after assembling



F. Fix the rocker arm of the servo PWM0 to the A14 plate.

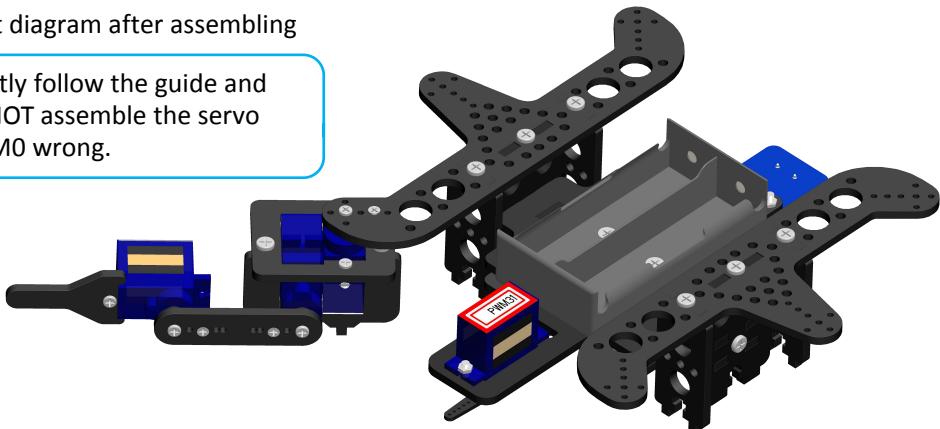
Assemble the following components



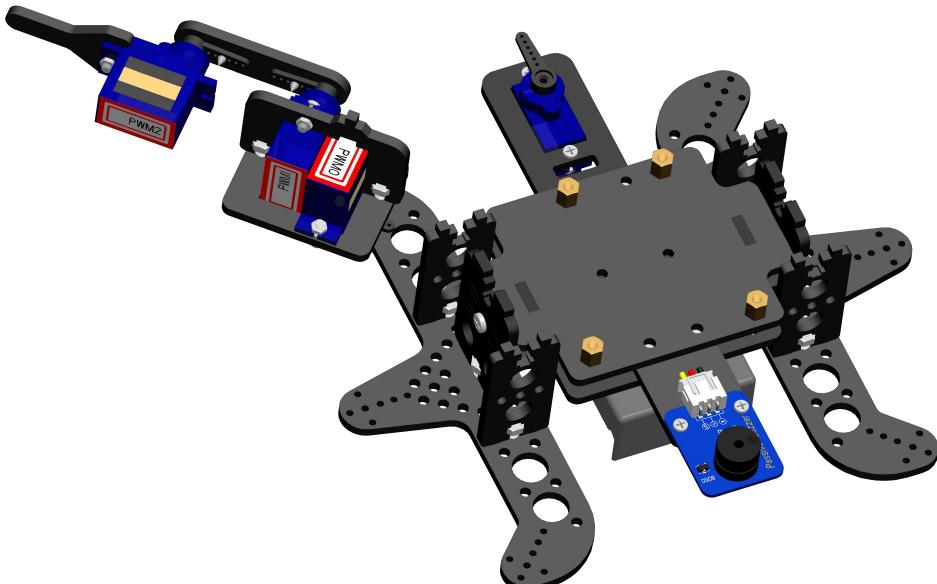
Fasten the servo and plate with an M2\*8 Screw in case of the rocker falling out, then align the holes on the plate and the rocker arm, insert the Self-tapping screw packaged with servo into the plate and rocker. For convenient operation, you can place the middle body on a stand.

Effect diagram after assembling

Strictly follow the guide and  
DONOT assemble the servo  
PWM0 wrong.

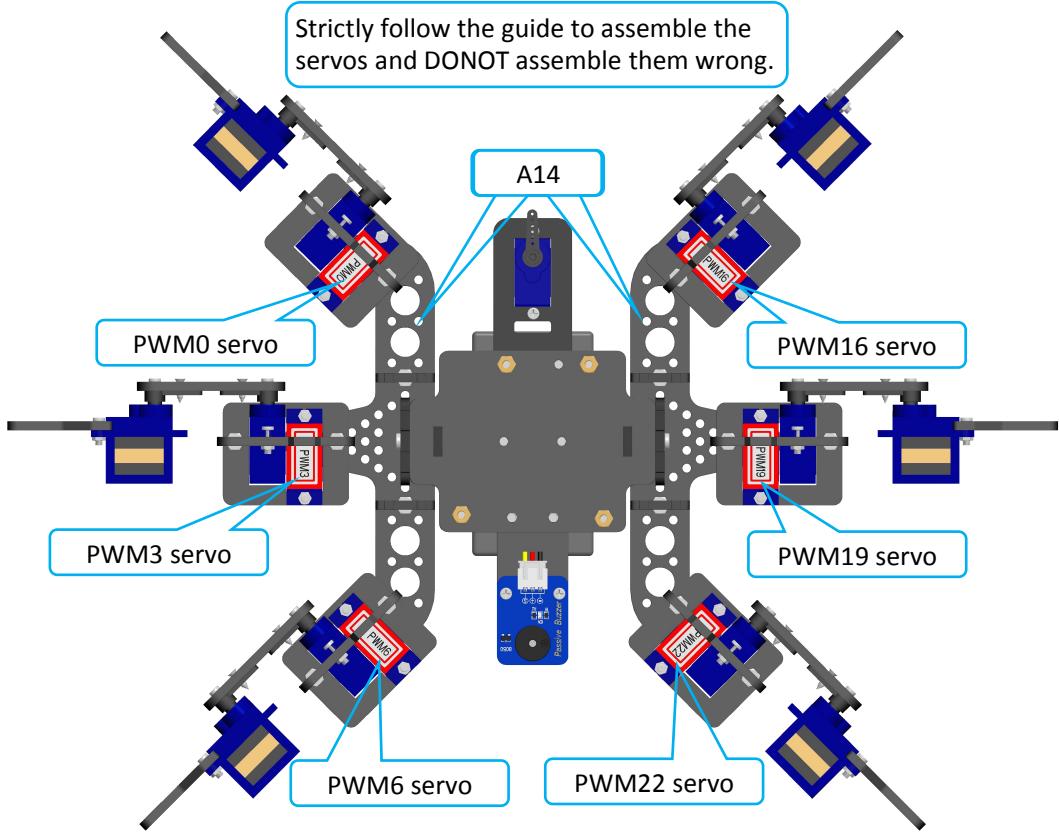


Front view:

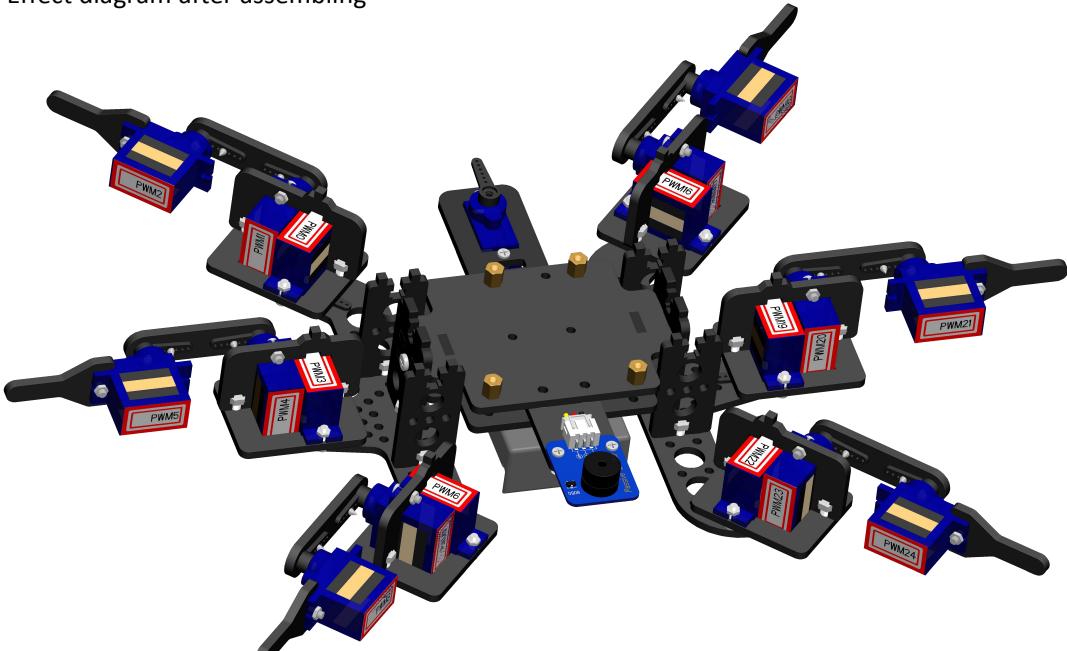


G. Install the rest 5 legs in the similar way onto the A14 plate.

Top view:

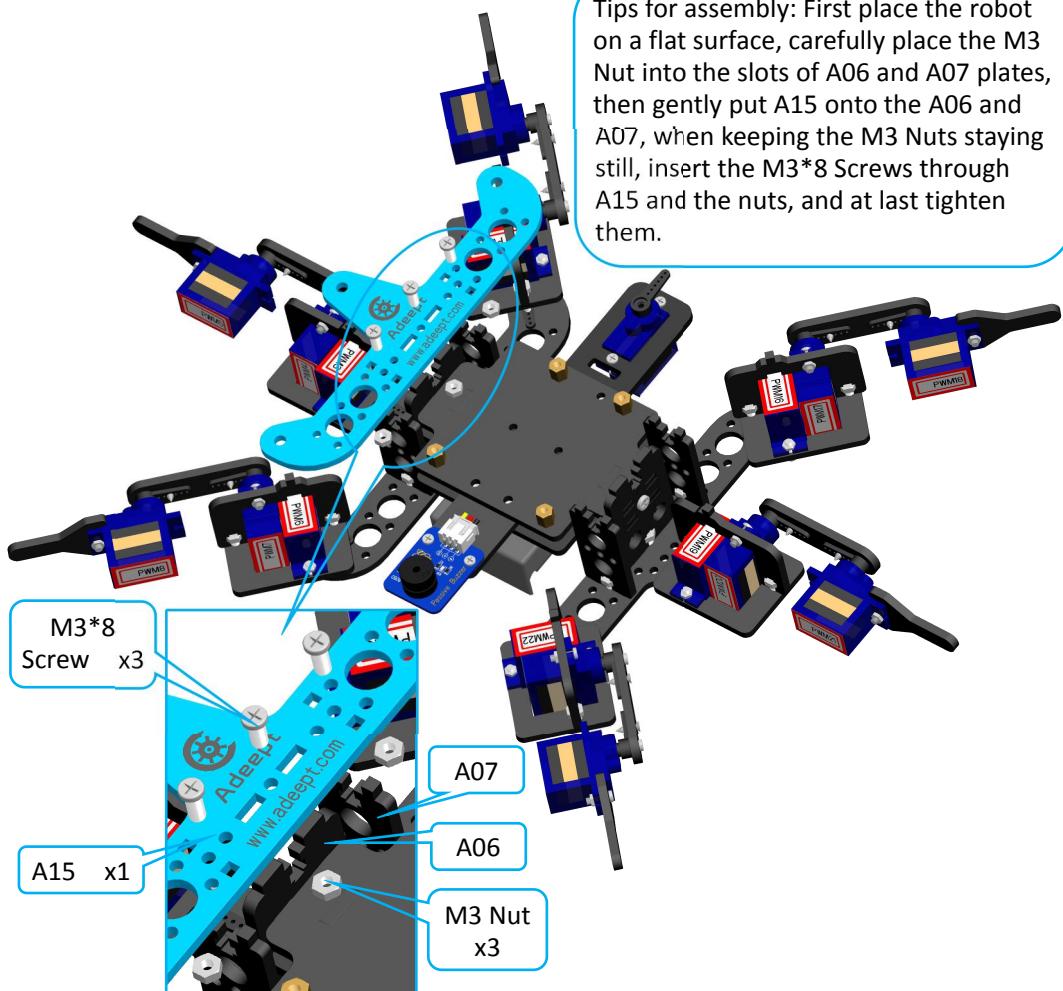


Effect diagram after assembling

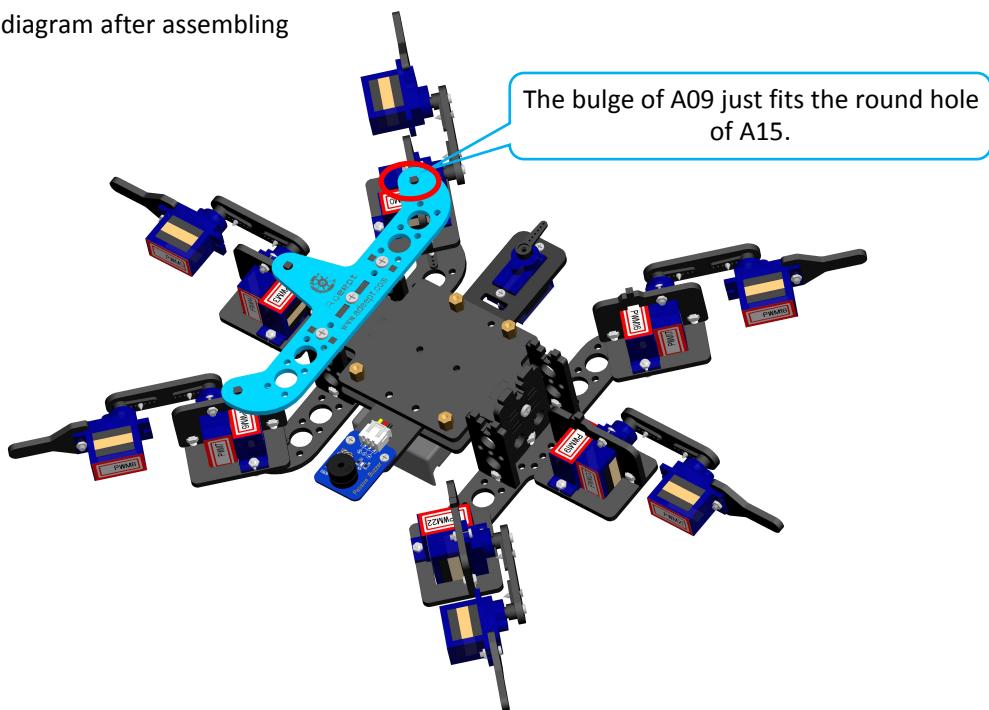


H. Fasten the A15 plate to A06 and A07.

Assemble the following components

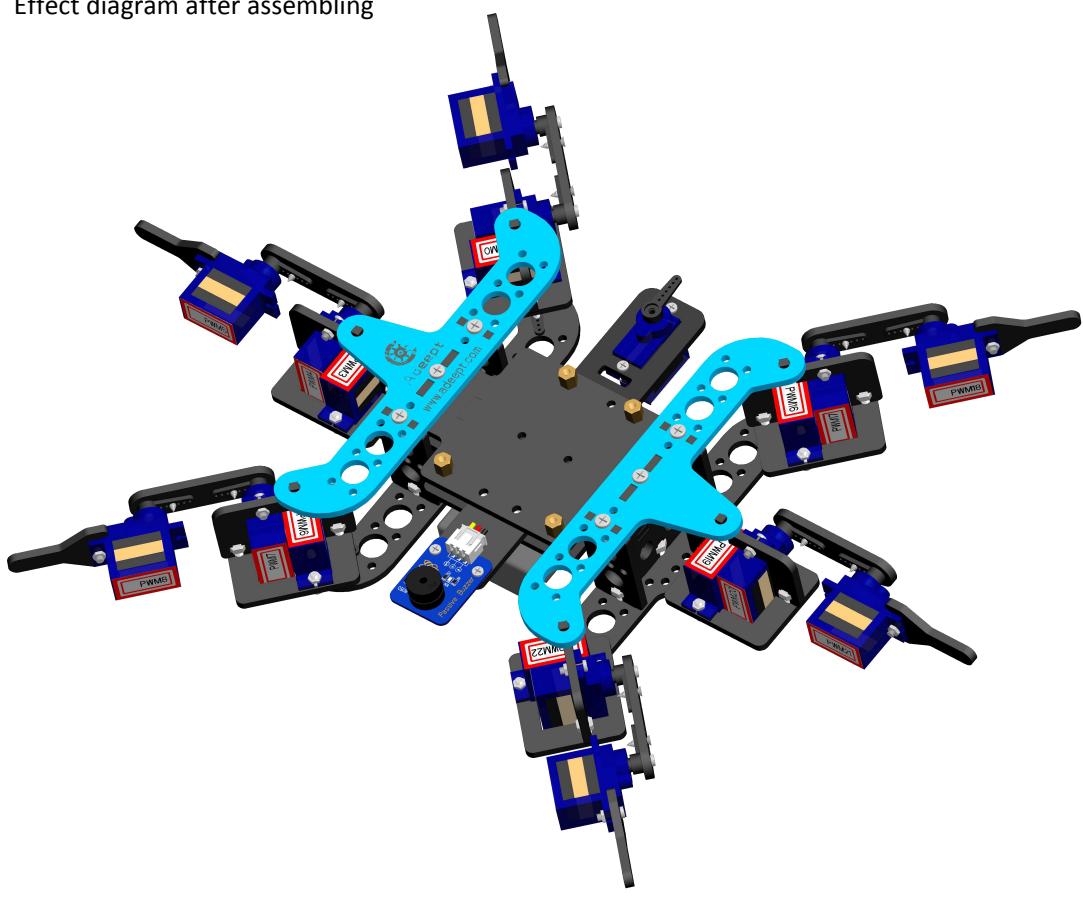


Effect diagram after assembling



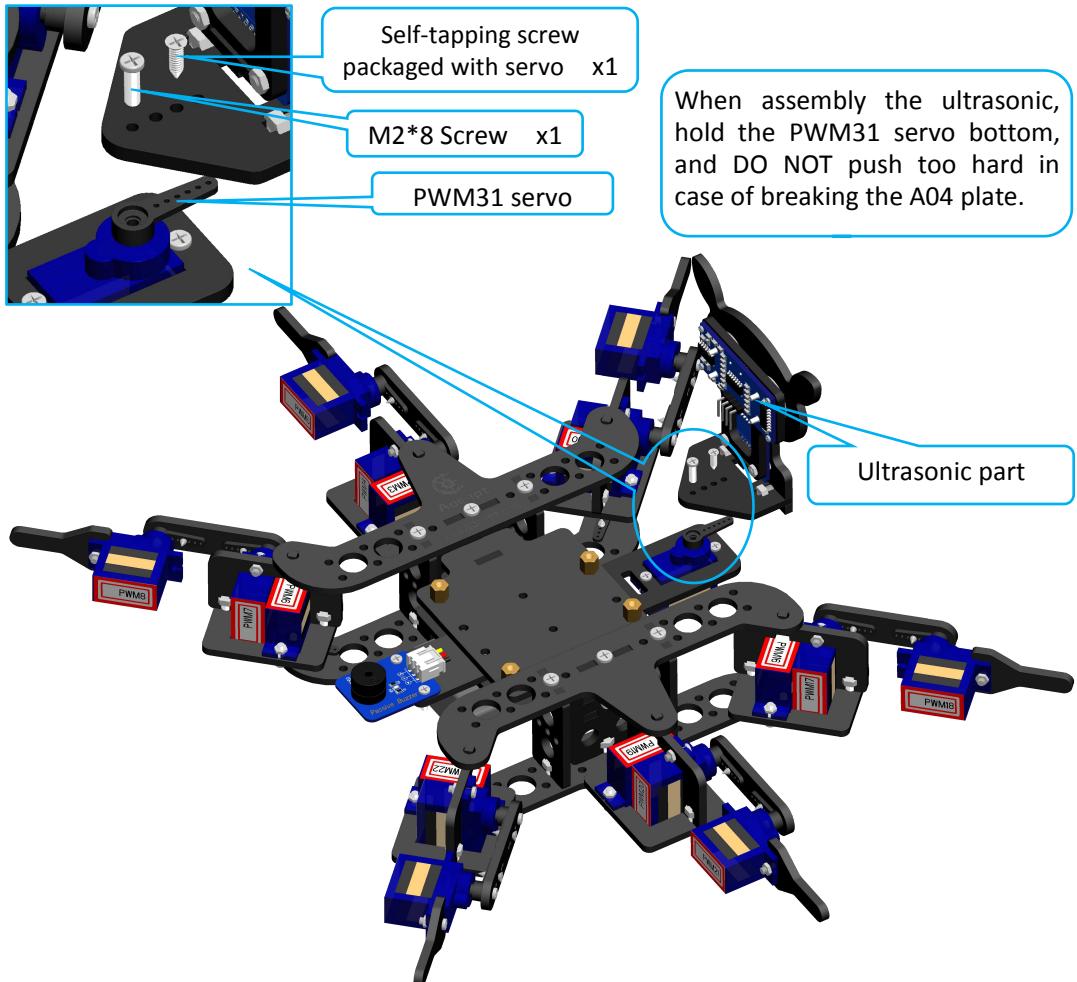
Fasten the other A15 plate in the same way.

Effect diagram after assembling

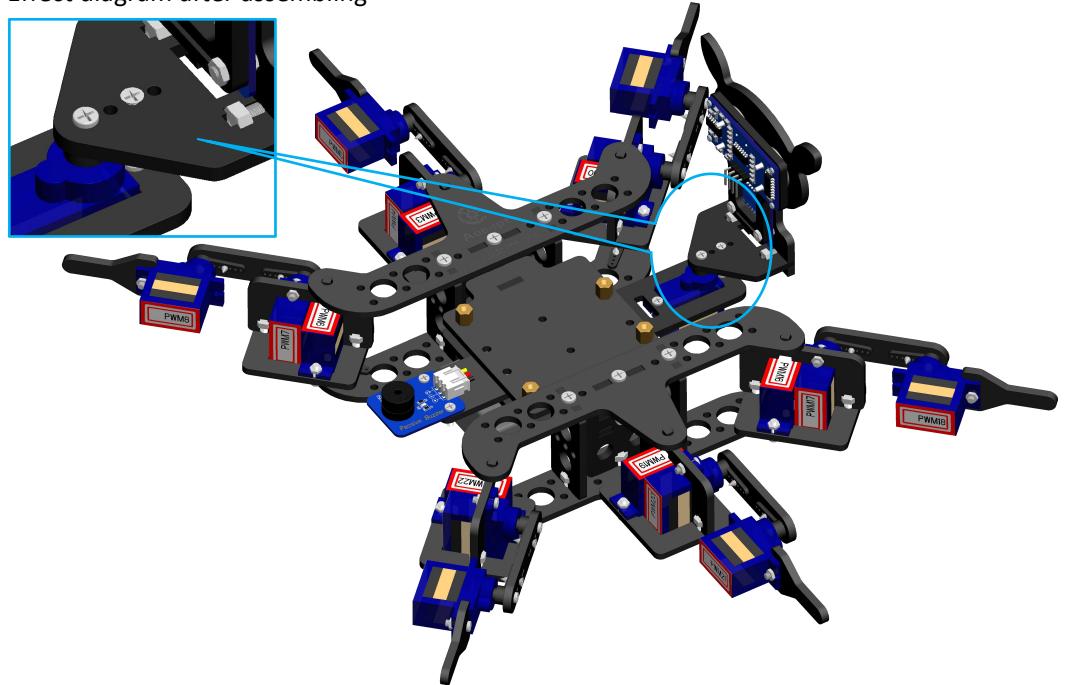


I. Fasten the ultrasonic part and the PWM31 servo.

Assemble the following components

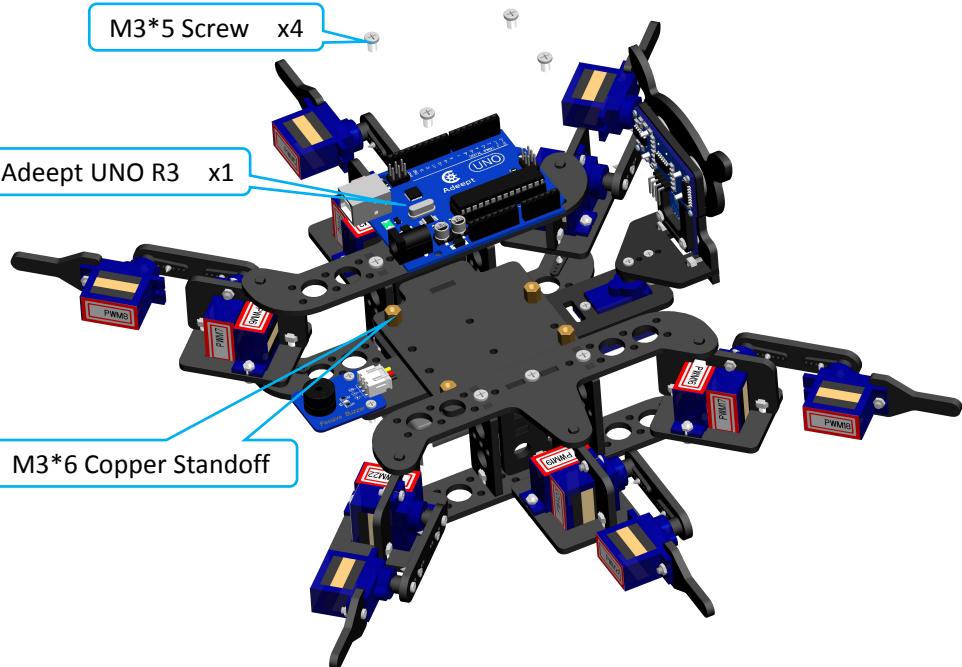


Effect diagram after assembling



J. Fix the Adeept Arduino UNO R3 board to the M3\*6 Copper Standoffs.

Assemble the following components

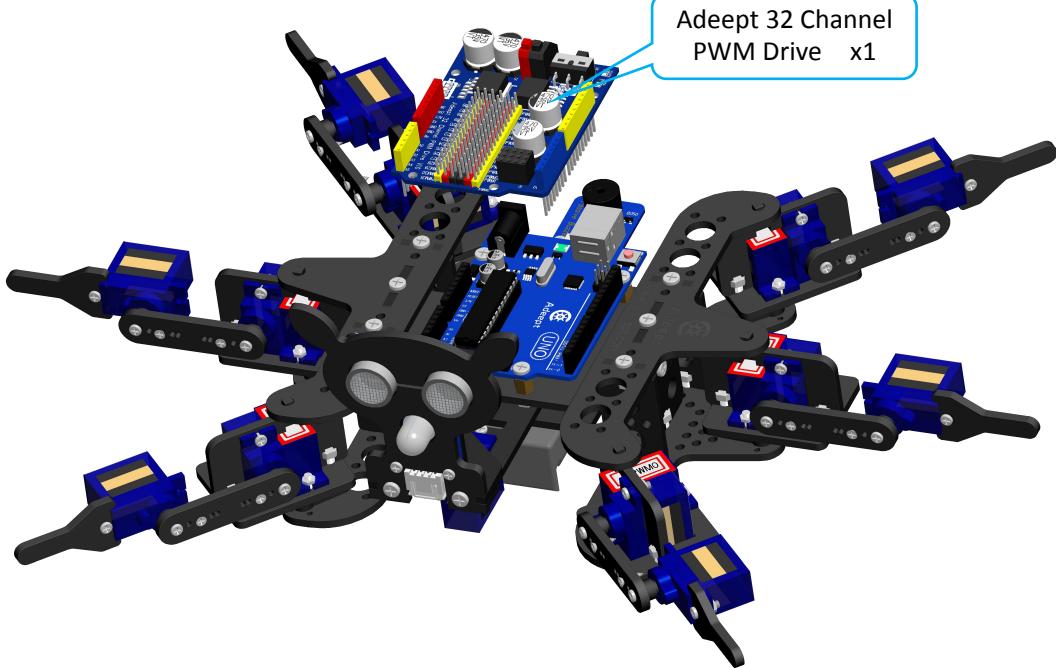


Effect diagram after assembling

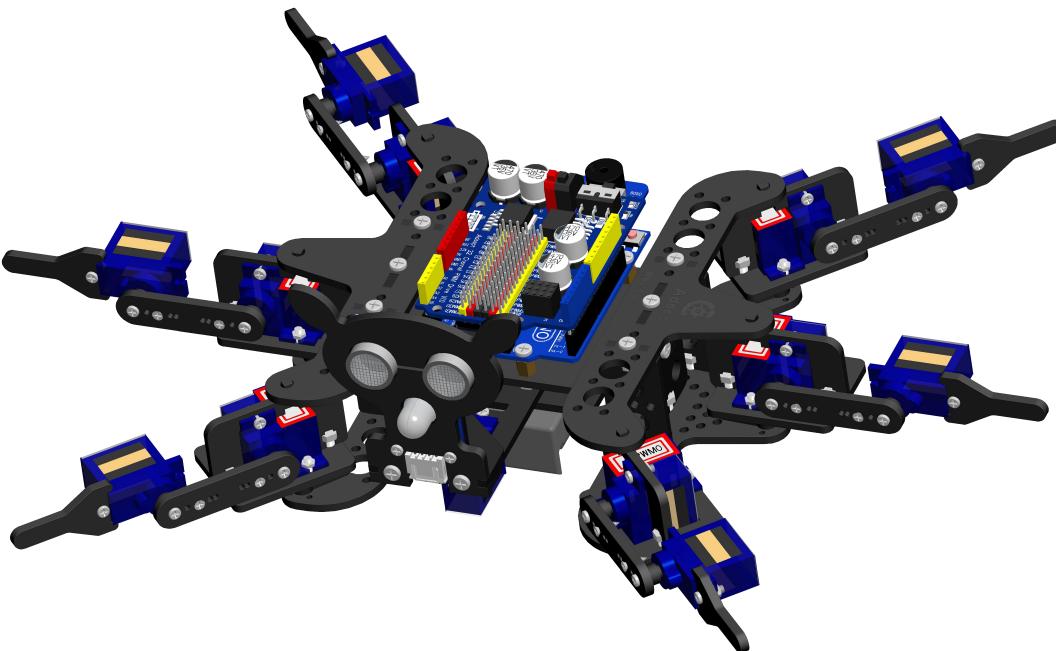


K. Plug the Adeept 32 Channel PWM Drive board into the Adeept UNO R3.

Assemble the following components

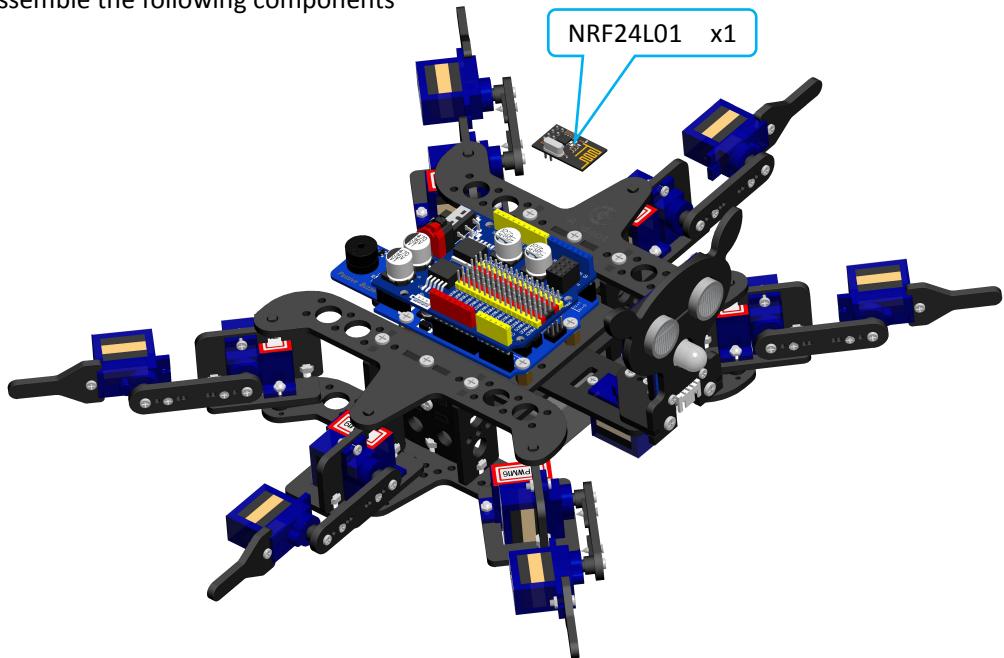


Effect diagram after assembling

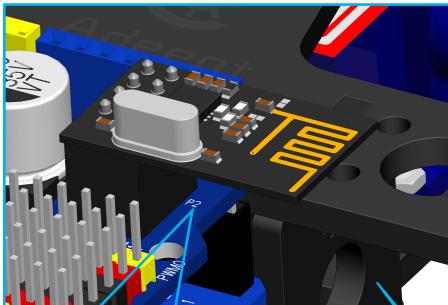


- L. Plug the NRF24L01 into the Adeept 32 Channel PWM Drive board.

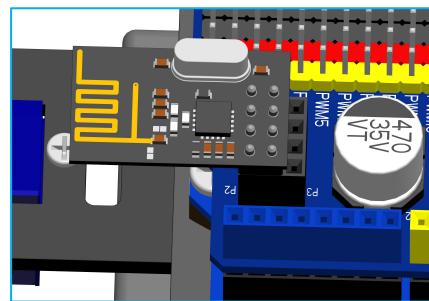
Assemble the following components



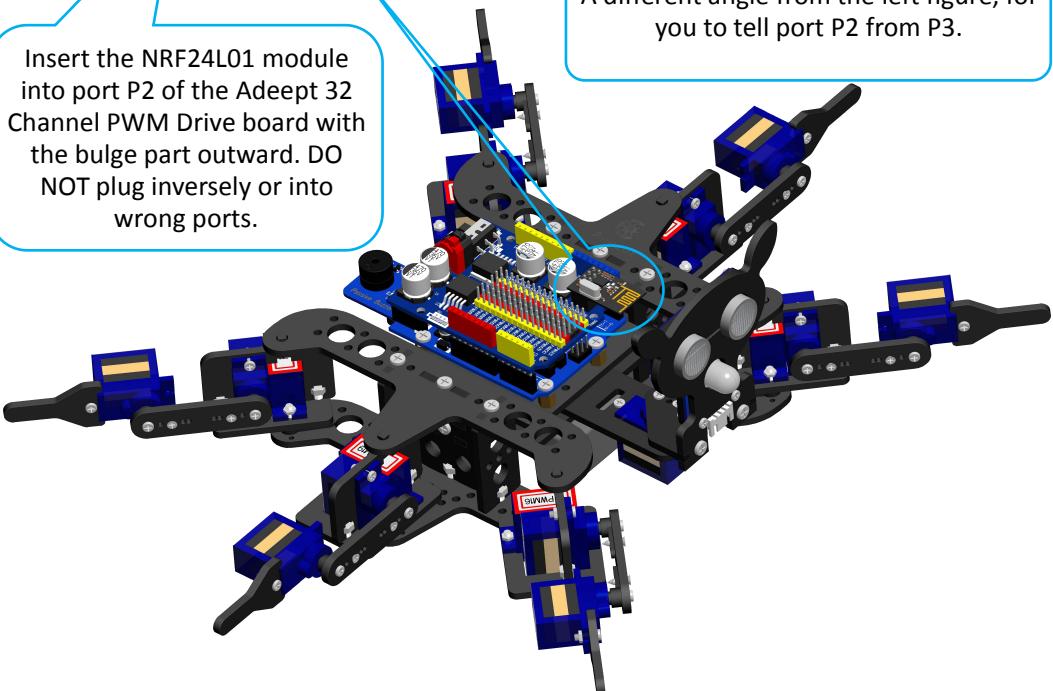
Effect diagram after assembling



Insert the NRF24L01 module into port P2 of the Adeept 32 Channel PWM Drive board with the bulge part outward. DO NOT plug inversely or into wrong ports.

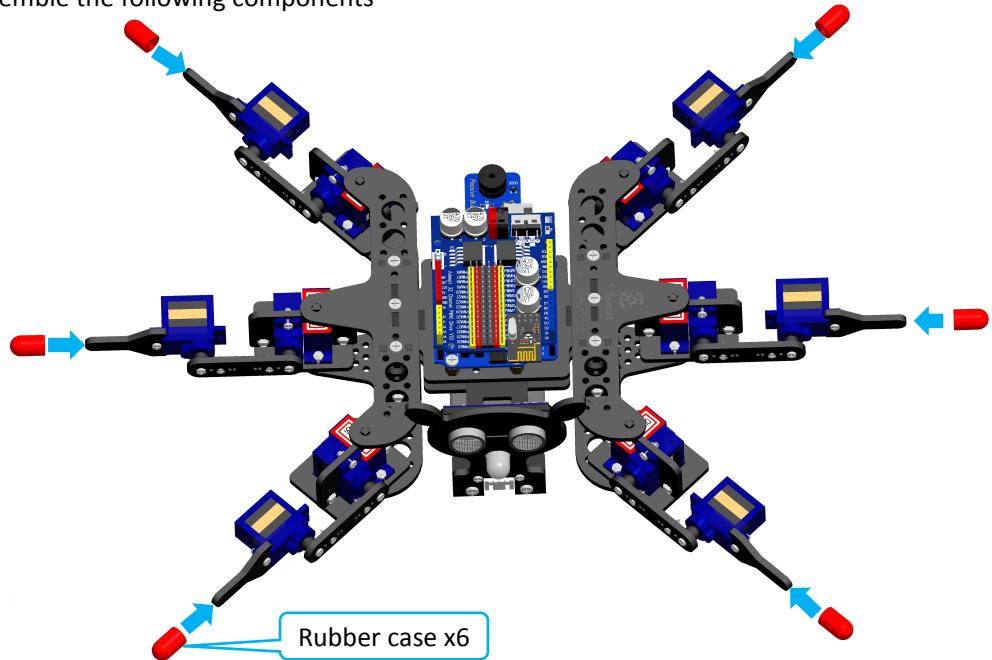


A different angle from the left figure, for you to tell port P2 from P3.

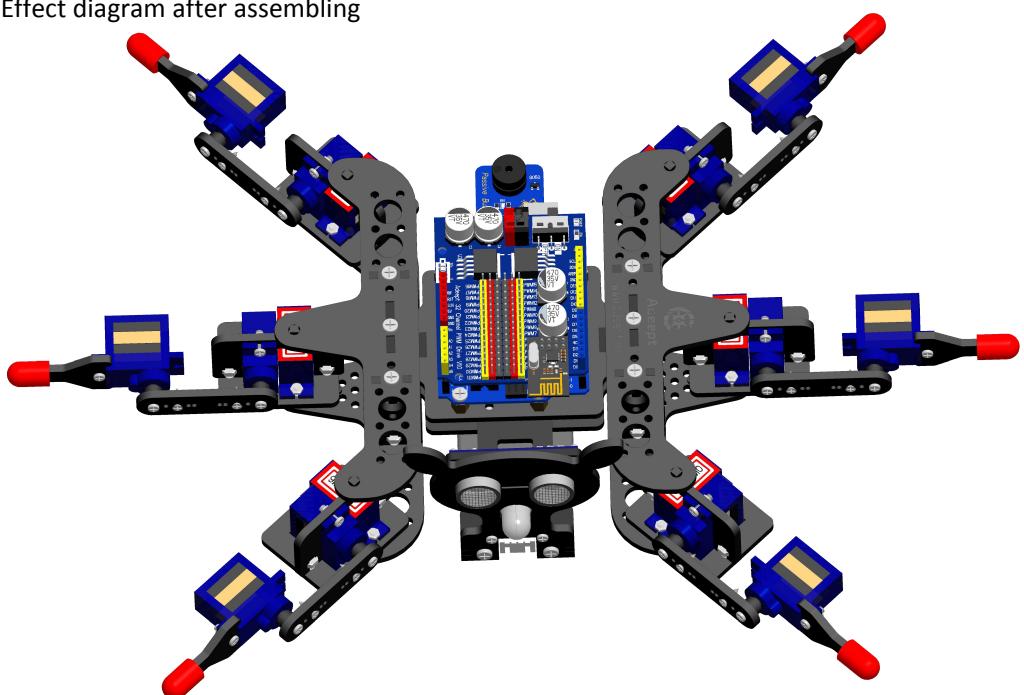


M. Wrap the pointed part of A12 plates with the 6 rubber cases.

Assemble the following components



Effect diagram after assembling

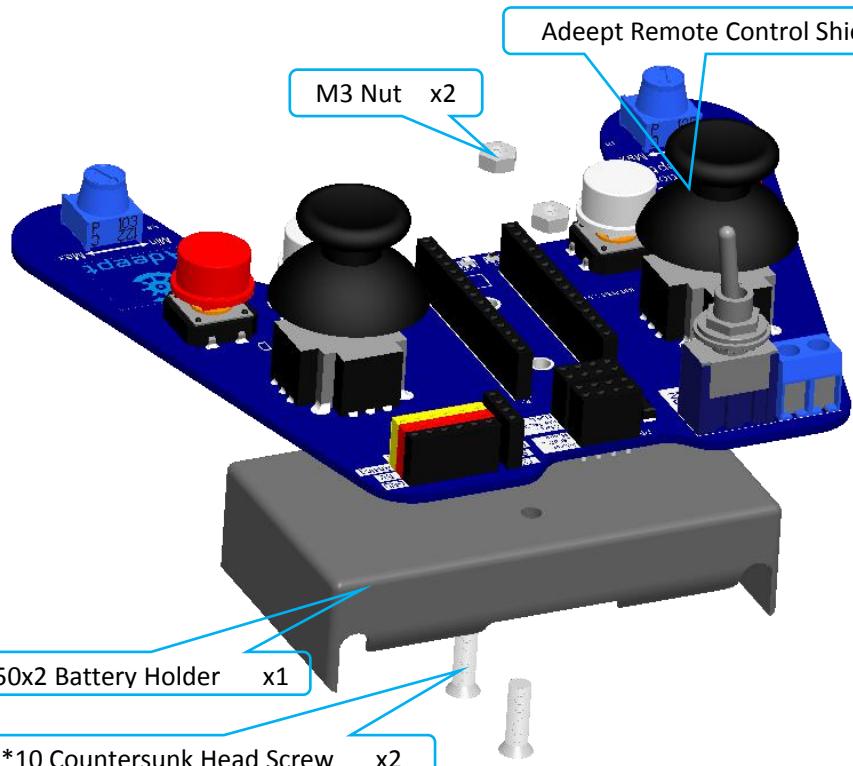


Function of the rubber case: Anti-slide, so the robot won't slide on frictionless surfaces.

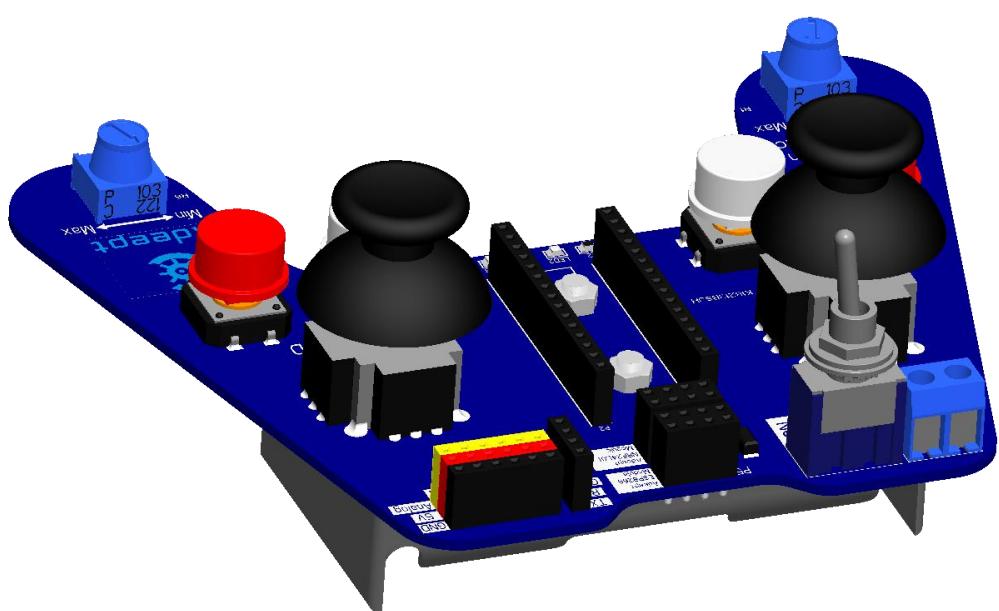
## Remote Control

A. Fix the 18650x2 Battery Holder and Adeept Remote Control Shield.

Assemble the following components

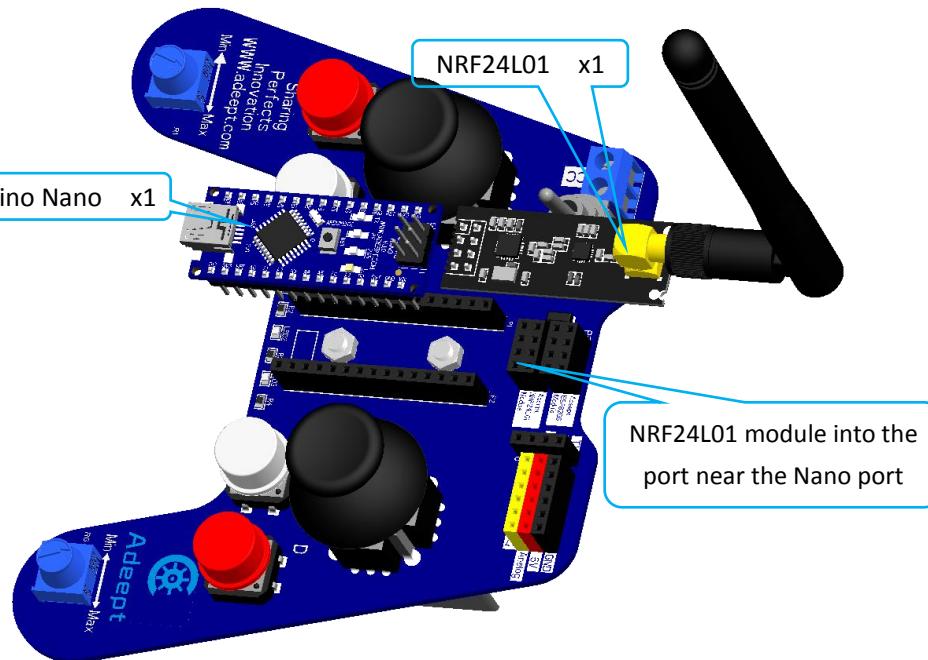


Effect diagram after assembling

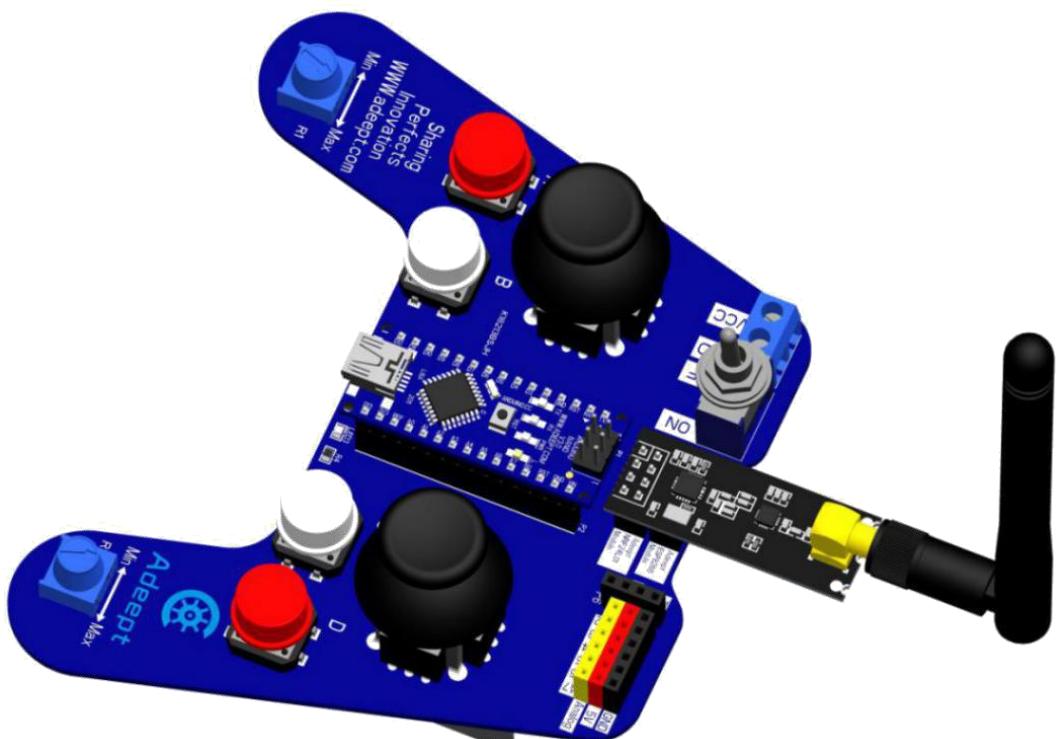


B. Plug the NRF24L01 Module and Arduino Nano board onto the Adeept Remote Control Shield.

Assemble the following components

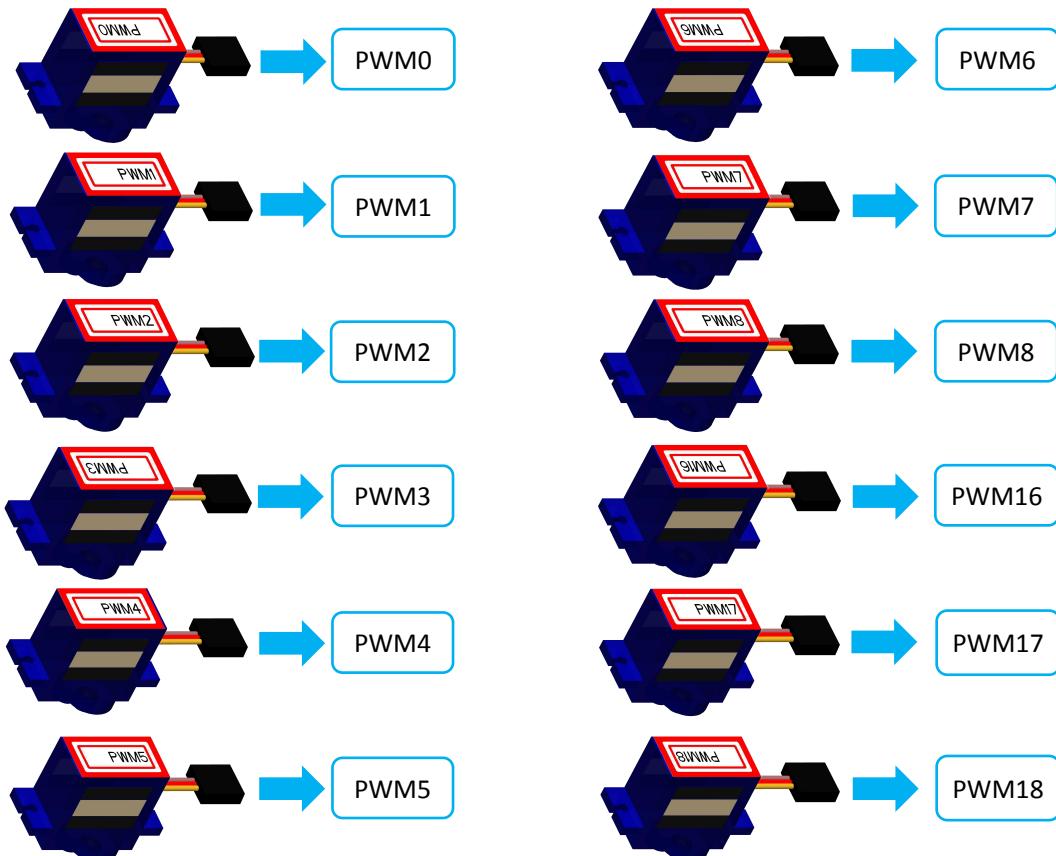
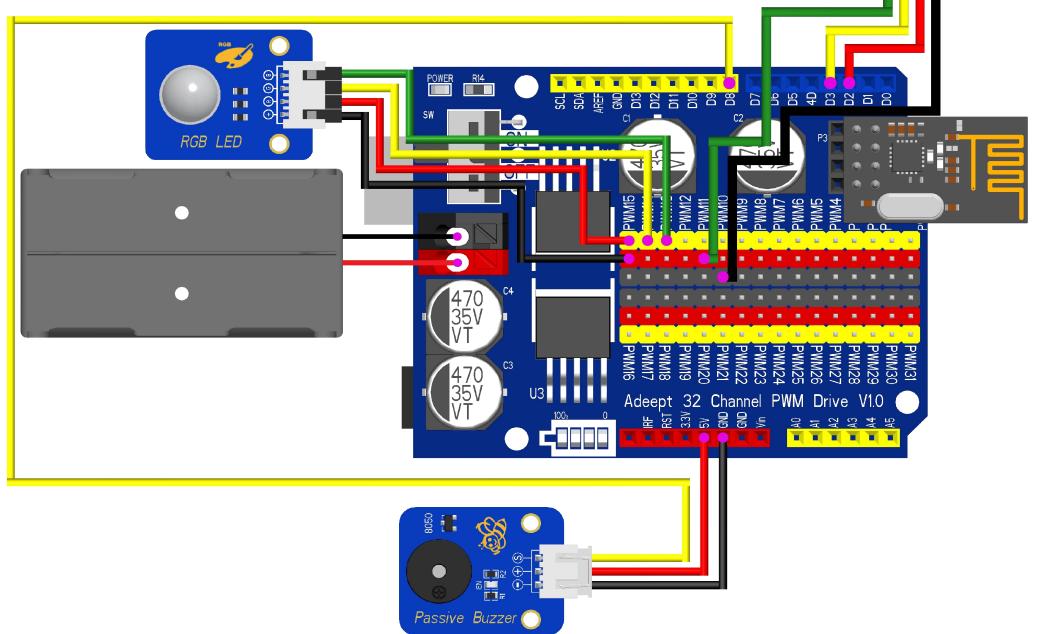


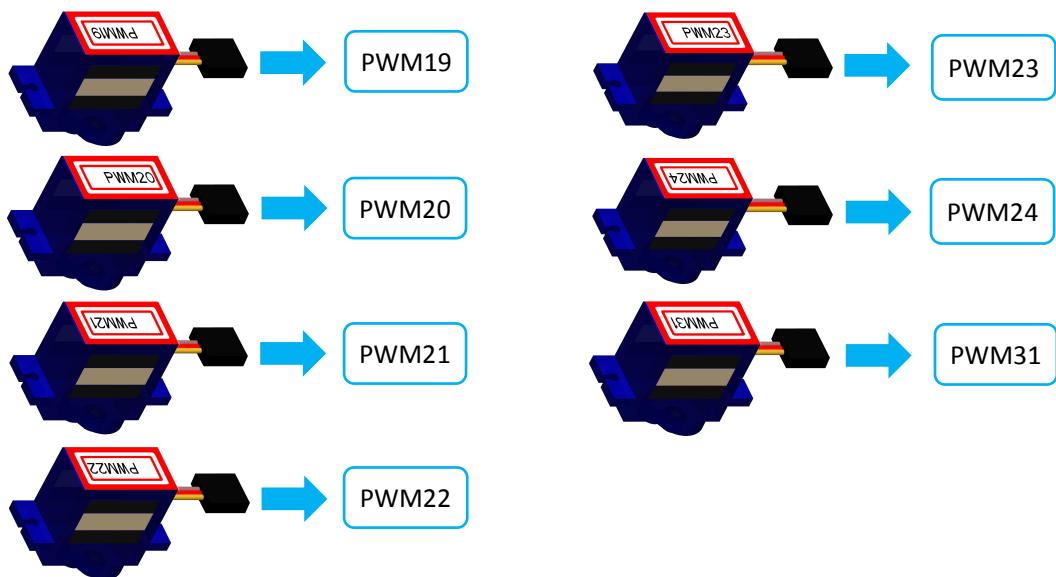
Effect diagram after assembling



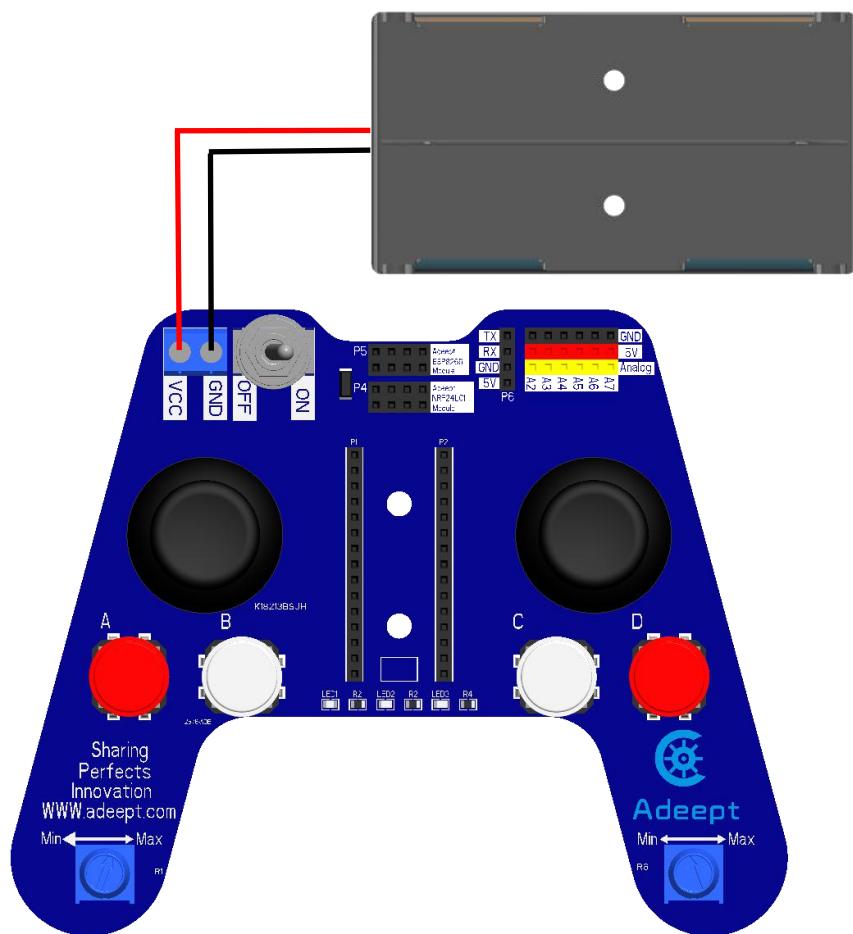
## Circuit Connection

Connect components based on the figure  
Pay attention to match the wire and port and not connect inversely.



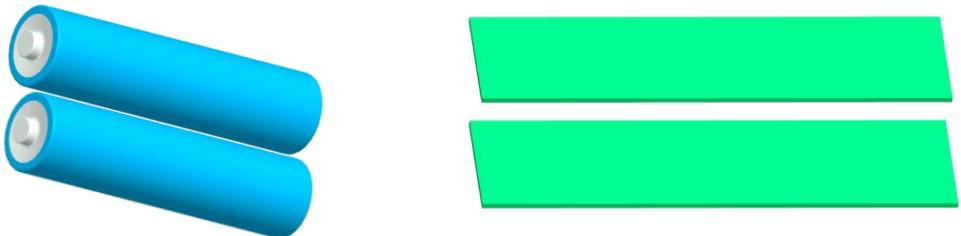


Circuit Connection of the Remote Control:

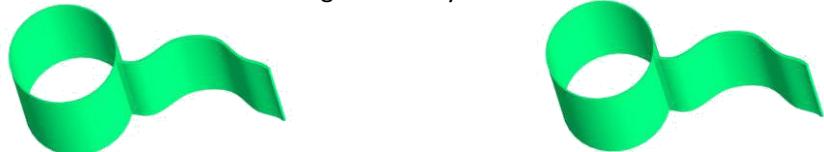


## Install and Remove Batteries

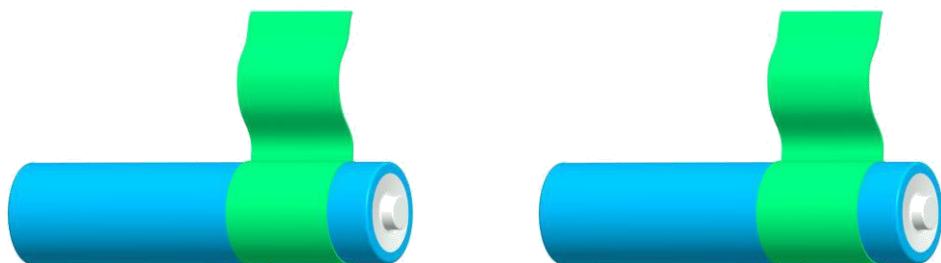
Take out 2 ribbons and 2 batteries.



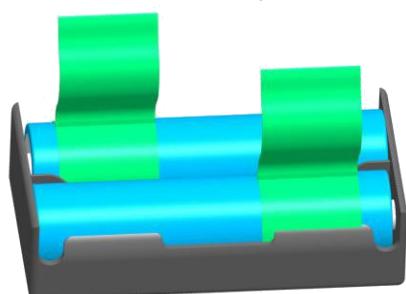
Roll one end of the ribbon to let through a battery and fix.



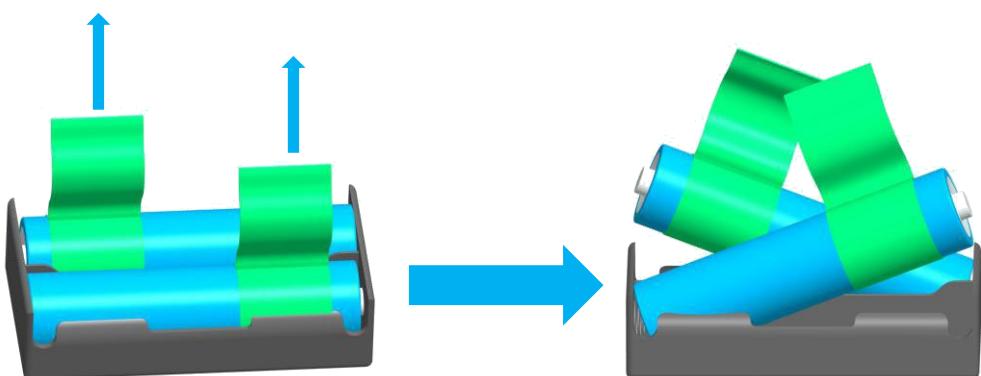
Insert the batteries into the rings-ribbon closer to the anode.



Install the batteries into the holder based on the pole.



To remove the batteries, just pull the ribbon and take them out.



# Software & Hardware

## What is Arduino?

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

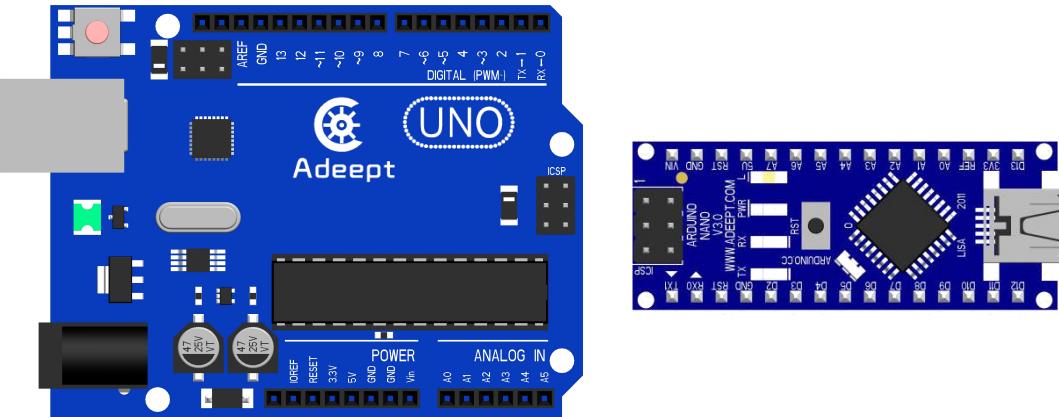
## Why Arduino?

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community.

## How Should I Use Arduino?

If you are a beginner with Arduino, Arduino learning kits on our website [www.adeept.com](http://www.adeept.com) would be a perfect step into this fantastic field!

Two types of Arduino board are used in this car kit: Adeept UNO R3 board and Arduino Nano board.



### Power

The Arduino/Genuino Uno board can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the GND and Vin pin headers of the POWER connector.

The board can operate on an external supply from 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may become unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

**Vin.** The input voltage to the Arduino/Genuino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

**5V.** This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board, which is not recommended.

**3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

**GND.** Ground pins.

**IOREF.** This pin on the Arduino/Genuino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V.

## Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50k ohm. A maximum of 40mA is the value that must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller.

In addition, some pins have specialized functions:

Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.

External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.

PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.

SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.

LED: 13. There is a built-in LED driven by digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

TWI: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the analogReference() function.

There are a couple of other pins on the board:

AREF. Reference voltage for the analog inputs. Used with analogReference().

Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

## Arduino Software (IDE)

Arduino Software (IDE) is used to write and upload the code for Arduino Board. First, install Arduino software (IDE): visit <https://www.arduino.cc/en/Main/Software>. Download the corresponding installation program according to your operating system. If you are a Windows user, please select the “Windows Installer” to download and install the driver correctly.

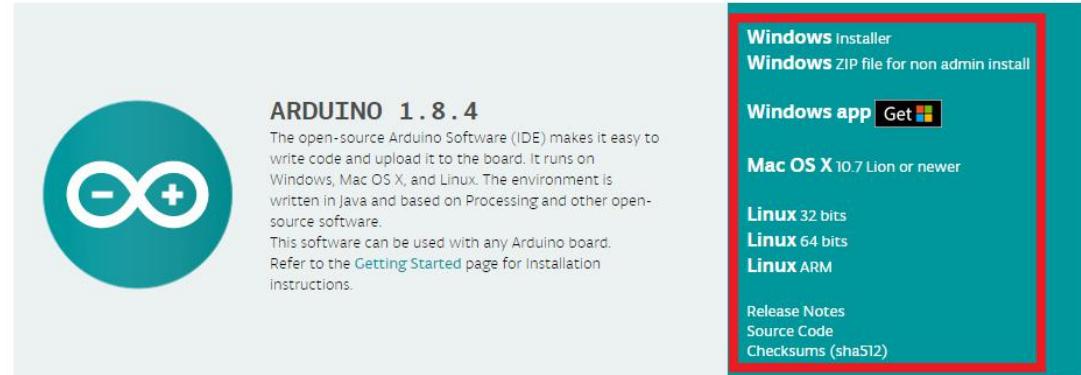
## Access the Online IDE



The screenshot shows the Arduino Web Editor interface. On the left is a yellow circle containing a teal infinity symbol with a minus sign on the left and a plus sign on the right. To its right is the title "ARDUINO WEB EDITOR". Below it is a brief description: "Start coding online with the [Arduino Web Editor](#), save your sketches in the cloud, and always have the most up-to-date version of the IDE, including all the contributed libraries and support for new Arduino boards. The Arduino Web Editor is one of the [Arduino Create platform's tools](#)". Below the description are two buttons: "Try It Now" and "Getting Started". On the right side of the interface is a code editor window titled "Board & Port". It contains the following code:

```
void setup() {  
}  
  
void loop() {  
}
```

## Download the Arduino IDE



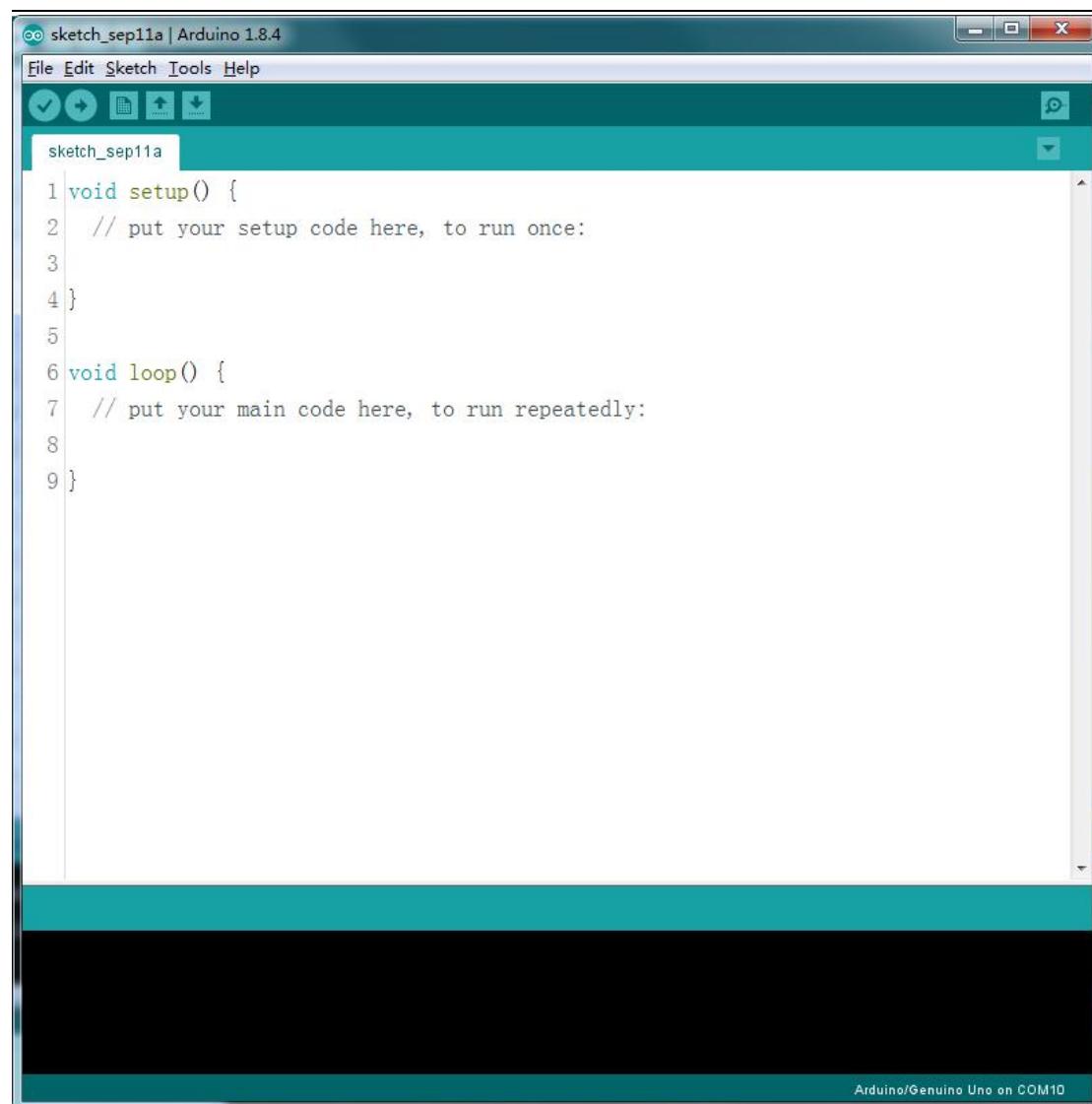
The screenshot shows the Arduino download page. On the left is a teal circle containing a white infinity symbol with a minus sign on the left and a plus sign on the right. To its right is the title "ARDUINO 1.8.4". Below it is a description: "The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board. Refer to the [Getting Started](#) page for installation instructions." To the right of the description is a large red-bordered box containing download links and options: "Windows Installer", "Windows ZIP file for non admin install", "Windows app Get", "Mac OS X 10.7 Lion or newer", "Linux 32 bits", "Linux 64 bits", "Linux ARM", "Release Notes", "Source Code", and "Checksums (sha512)".

After the download completes, run the installer. For Windows users, there may pop up an installation dialog box of the driver during the installation. Please agree the installation when it appears.

After installation is completed, an Arduino software shortcut will be generated on the desktop. Run the IDE.



The interface of Arduino software is as follows:



The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them.

 **Verify** : Checks your code for errors when compiling it.

 **Upload** : Compiles your code and uploads it to the configured board.

Before uploading your sketch, you need to select the correct items from the **Tools > Board** and **Tools > Port** menus. The boards are described below. On the Mac OS X, the serial port is probably something like **/dev/tty.usbmodem241** (for an Uno or Mega2560 or Leonardo) or **/dev/tty.usbserial-1B1** (for a Duemilanove or earlier USB board), or **/dev/tty.USA19QW1b1P1.1** (for a serial board connected with a Keyspan USB-to-Serial adapter). On Windows, it's probably **COM1** or **COM2** (for a serial board) or **COM4, COM5, COM7**, or higher (for a USB board) - to find

---

out, you look for USB serial device in the ports section of the Windows **Device Manager**. On Linux, it should be **/dev/ttyACMx** , **/dev/ttyUSBx** or similar.

Once you've selected the correct serial port and board, press the upload button in the toolbar or select the **Upload** item from the **Sketch** menu. Current Arduino boards will reset automatically and begin the upload. With older boards (pre-Diecimila) that lack auto-reset, you'll need to press the reset button on the board just before starting the upload. On most boards, you'll see the RX and TX LEDs blink as the sketch is uploaded. The Arduino Software (IDE) will display a message when the upload is completed, or show an error.

When you upload a sketch, you're using the Arduino bootloader, a small program that has been loaded on to the microcontroller on your board. It allows you to upload code without using any additional hardware. The bootloader is active for a few seconds when the board resets; then it starts whichever sketch was most recently uploaded to the microcontroller. The bootloader will blink the on-board (pin 13) LED when it starts (i.e. when the board resets).

**Note:** If you are using an external programmer with your board, you can hold down the "shift" key on your computer when using this icon. The text will change to "Upload using Programmer"



**New:** Creates a new sketch.



**Open:** Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window overwriting its content.

**Note:** Due to a bug in Java, this menu doesn't scroll; if you need to open a sketch late in the list, use the File | Sketchbook menu instead.



**Save:** Saves your sketch.



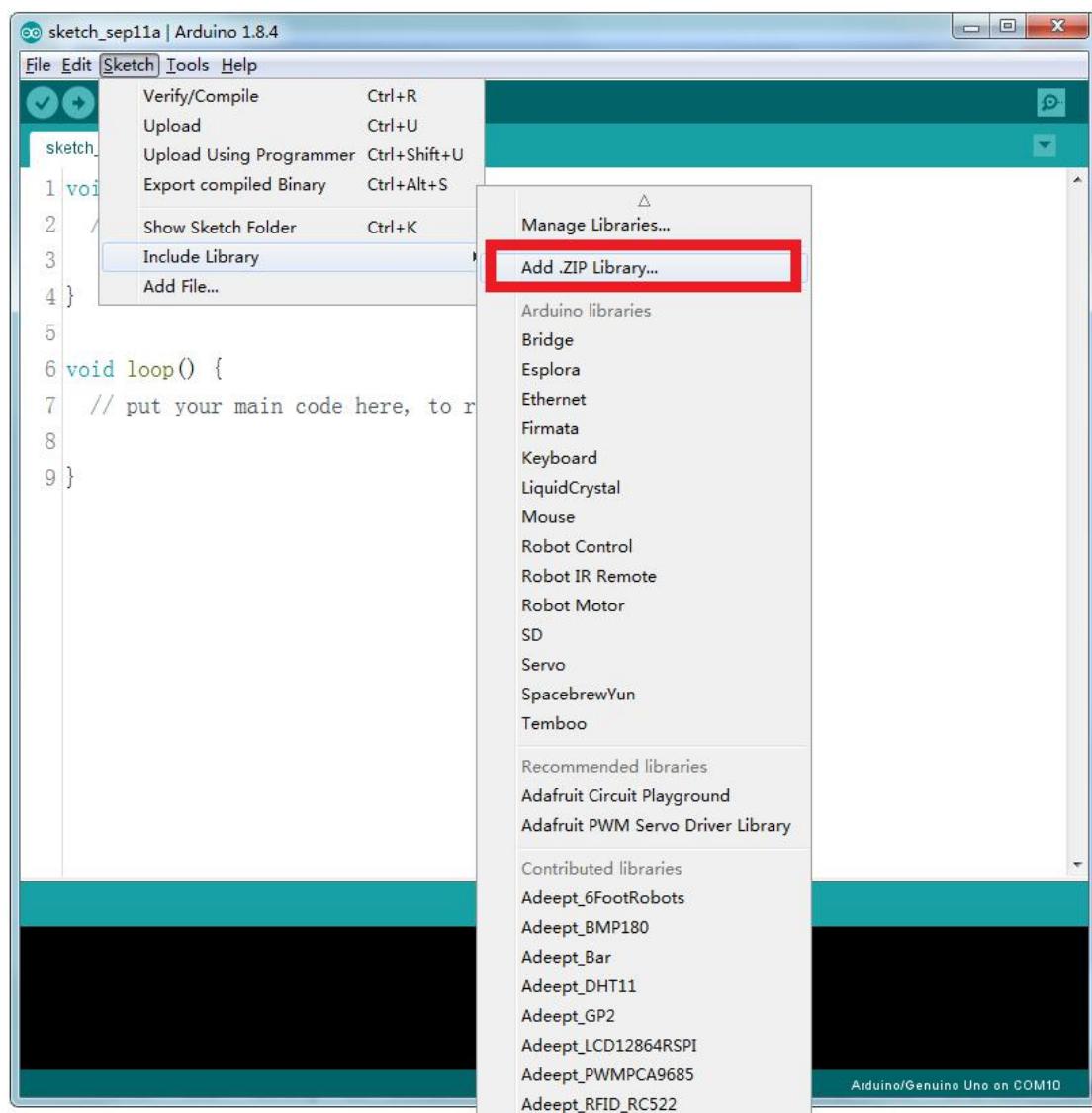
**Serial Monitor:** Opens the serial monitor.

Additional commands are found within the five menus: **File**, **Edit**, **Sketch**, **Tools**, and **Help**. The menus are context sensitive, which means only those items relevant to the work currently being carried out are available.

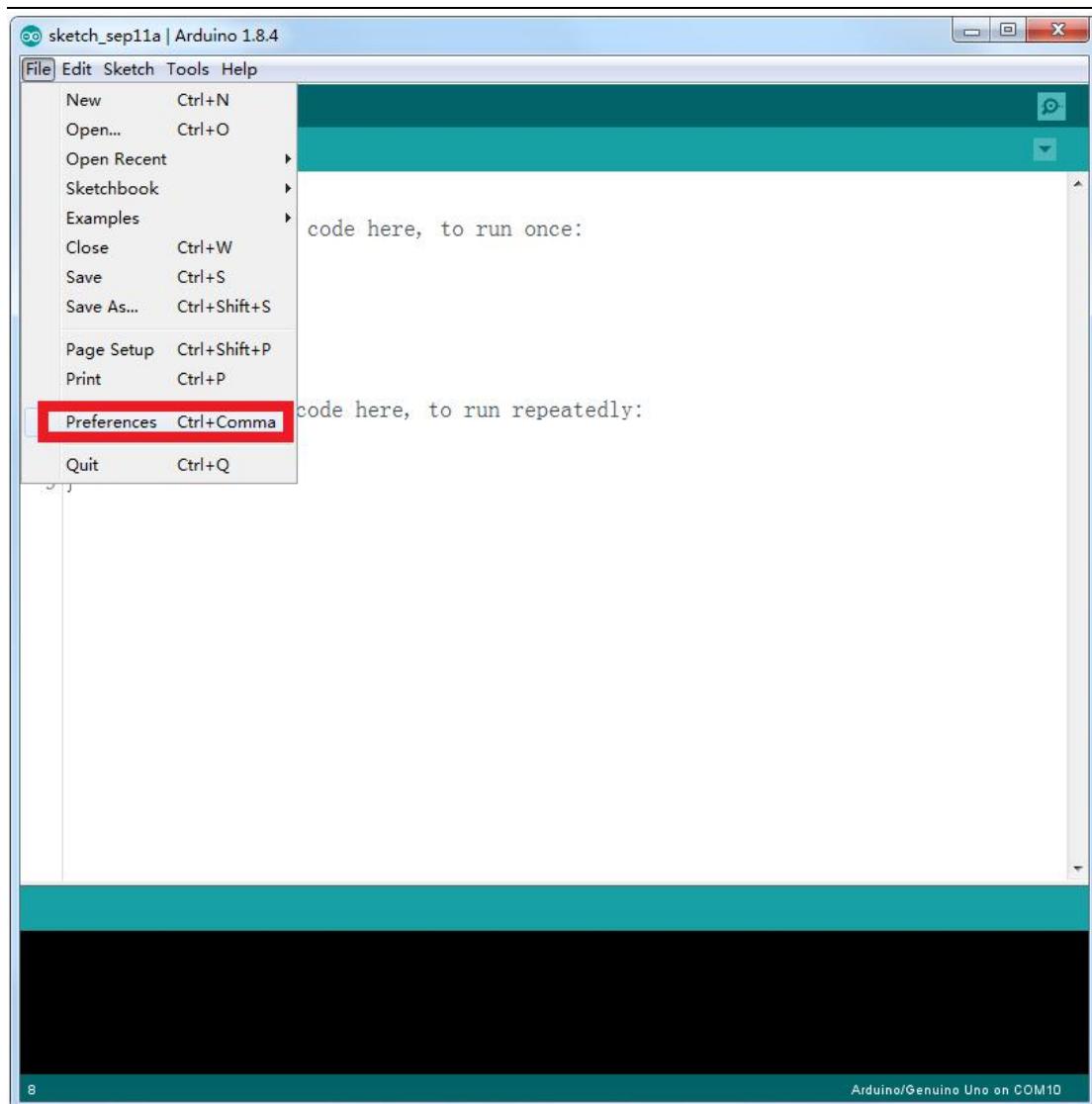
Since version 1.0, files are saved with an **.ino** file extension. Previous versions use the **.pde** extension. You may still open **.pde** named files in version 1.0 and later, and the software will automatically rename the extension to **.ino**.

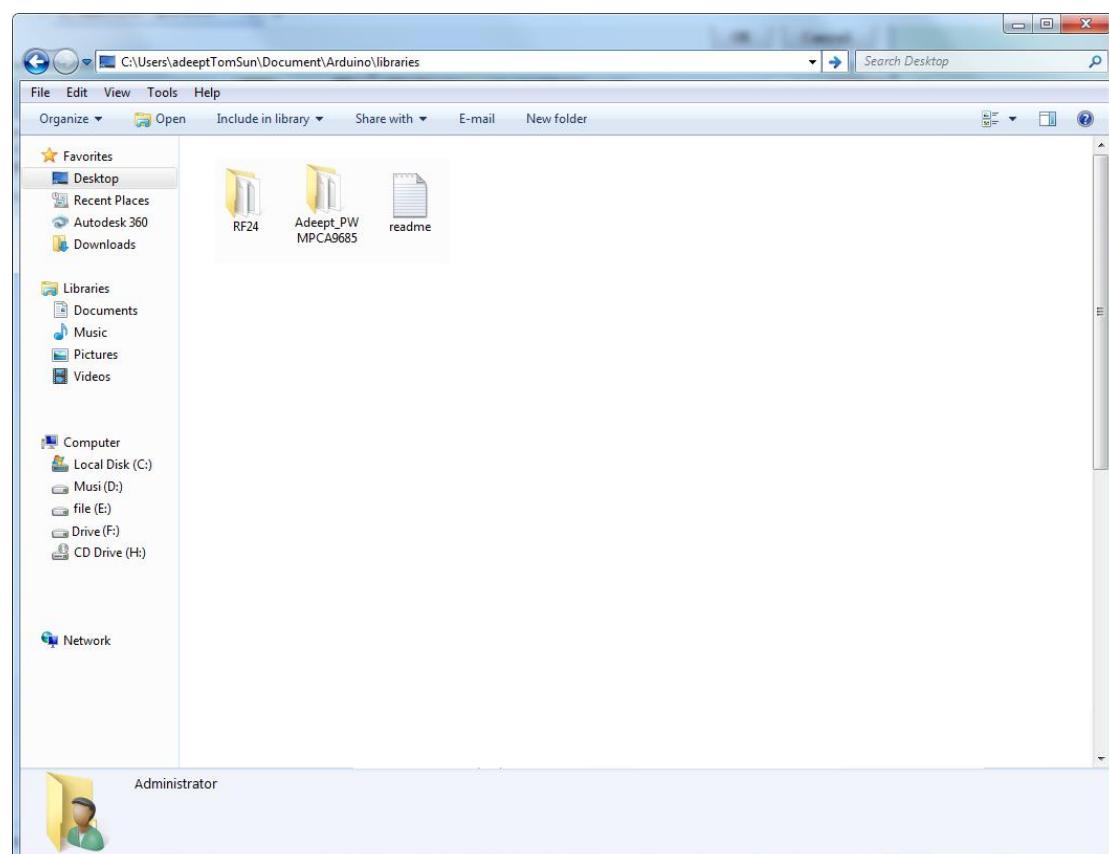
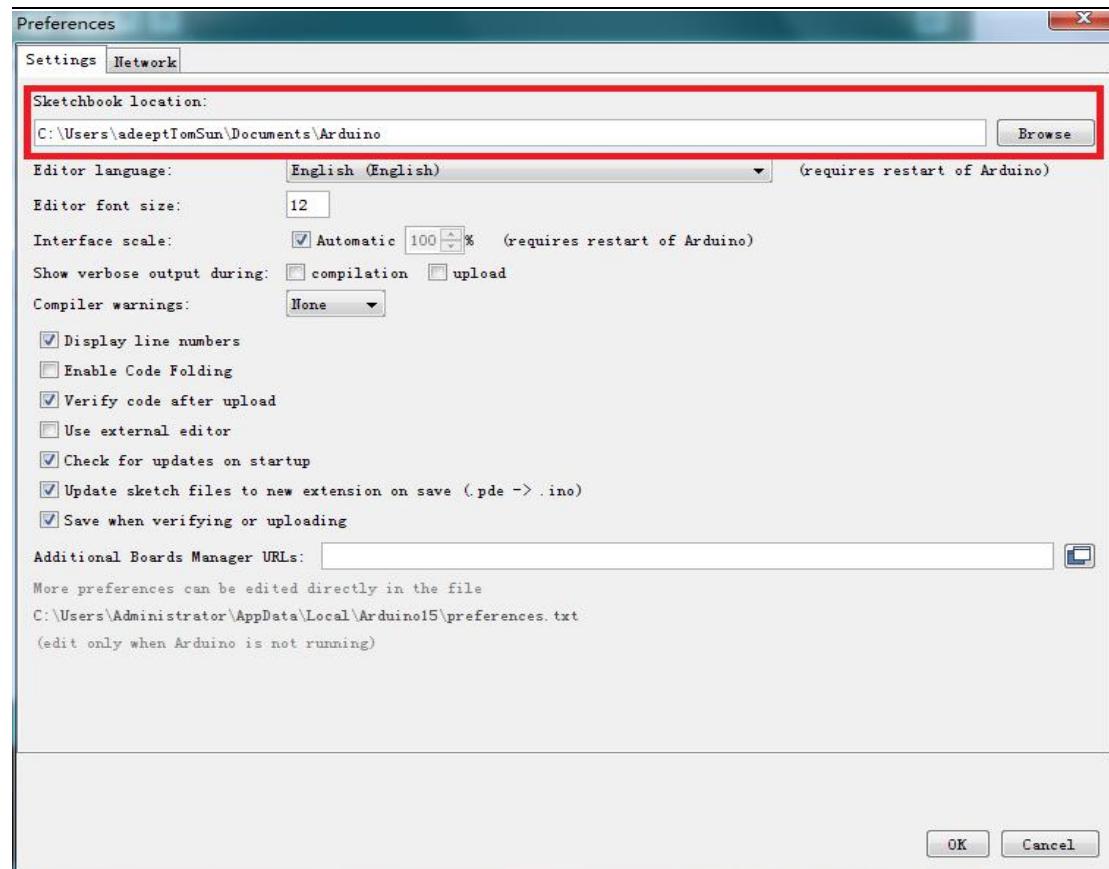
## Install Library

The example sketches provided use the *RF24.ZIP* and *Adeept\_PWMPCA9685.ZIP* library, so you need to install it before compiling. Click **Add.ZIP Library** to add the *RF24.ZIP* and *Adeept\_PWMPCA9685.ZIP* to the *libraries* folder.



After the library is installed successfully, you can find the *RF24.ZIP* and *Adeept\_PWMPCA9685.ZIP* under **Sketchbook location:** on the window popped up by clicking Preferences.

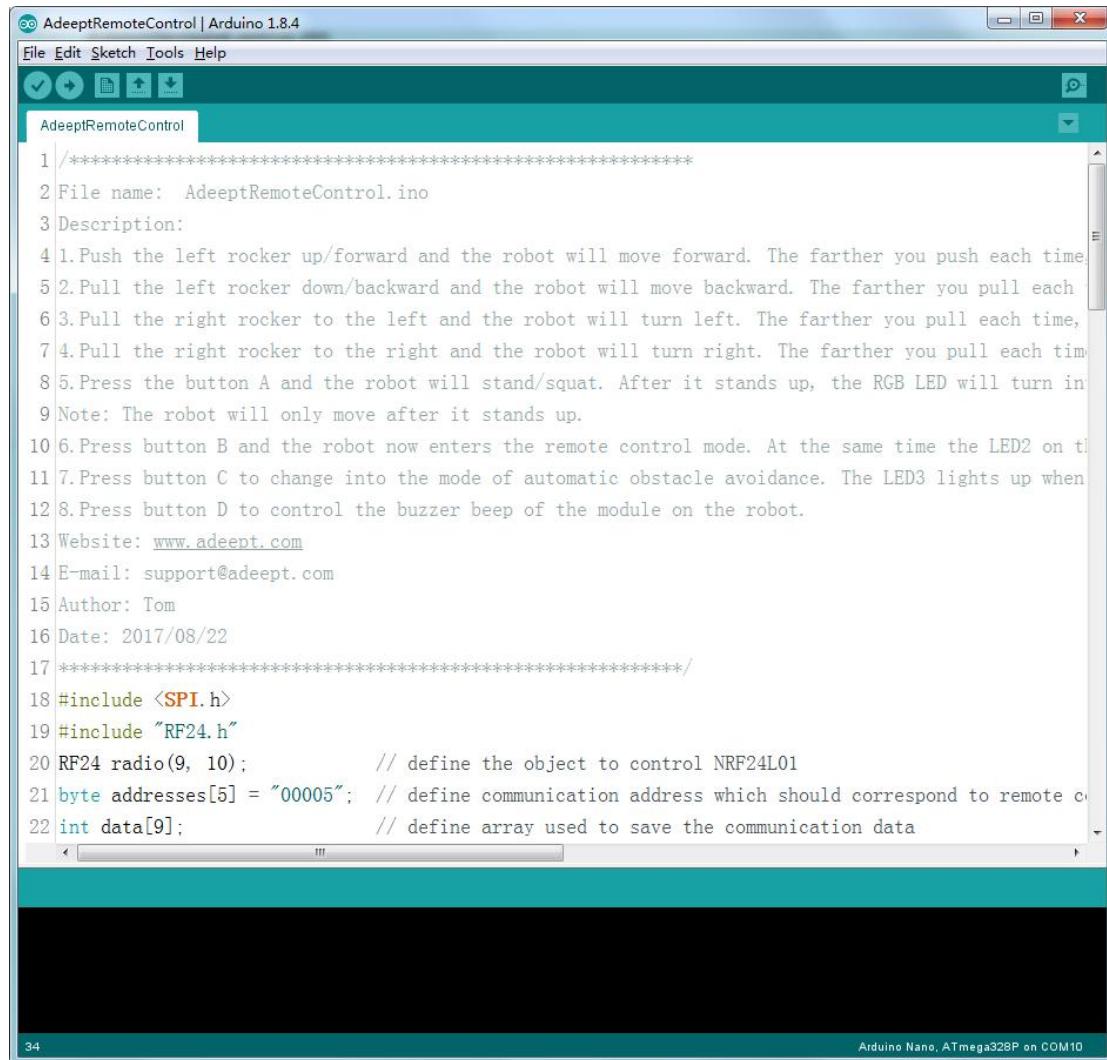




## Upload Program

After the preparations above, next we will upload the program (example sketches provided) to the Arduino Nano and Adeept UNO R3 boards. The Robots kit comprises of two parts: the remote control based on Arduino Nano and the Robots controller on Adeept UNO R3.

First, upload the sketch to Arduino Nano. Open the file for the remote control, *AdeeptRemoteControl.ino*.



The screenshot shows the Arduino IDE interface with the title bar "AdeeptRemoteControl | Arduino 1.8.4". The menu bar includes File, Edit, Sketch, Tools, and Help. The toolbar has icons for new, open, save, and upload. The main code editor window contains the following code:

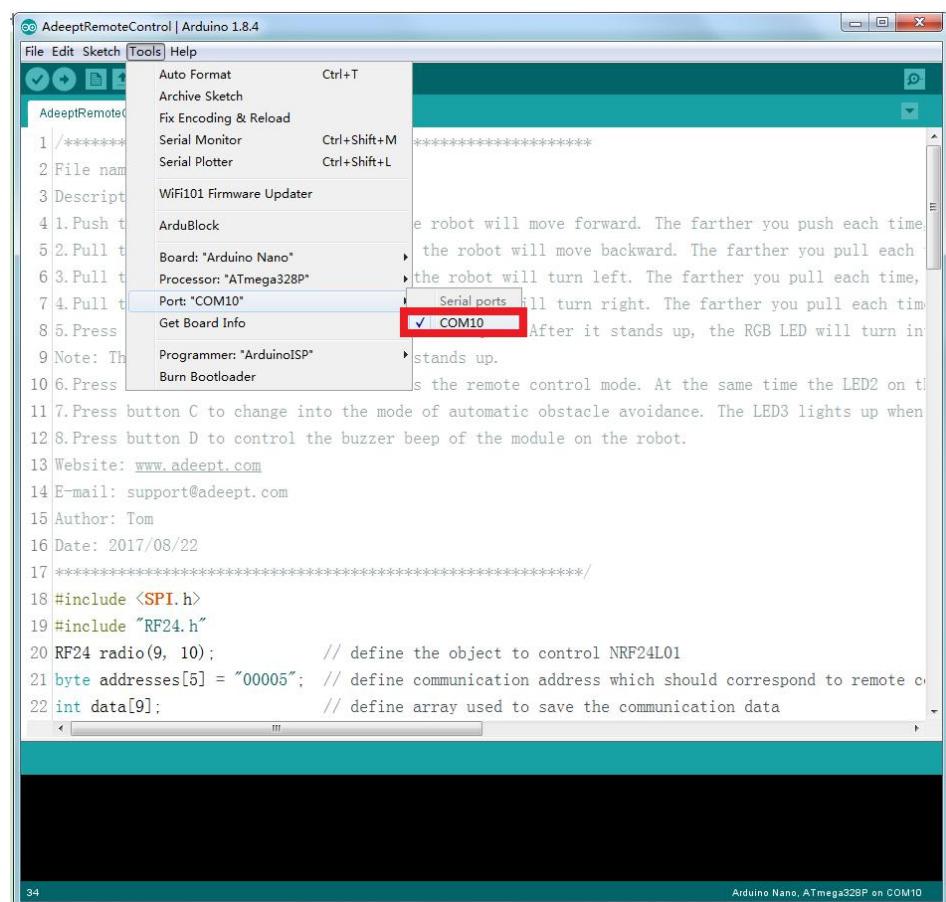
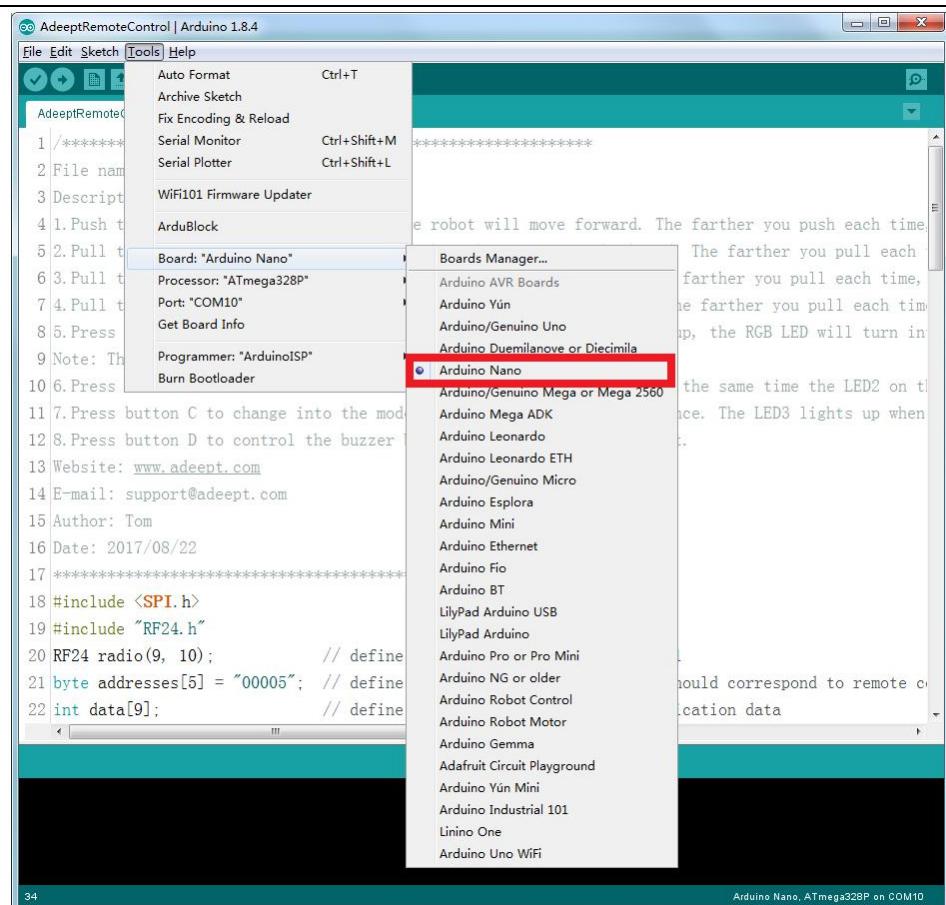
```

1 //*****
2 File name: AdeeptRemoteControl.ino
3 Description:
4 1. Push the left rocker up/forward and the robot will move forward. The farther you push each time,
5 2. Pull the left rocker down/backward and the robot will move backward. The farther you pull each
6 3. Pull the right rocker to the left and the robot will turn left. The farther you pull each time,
7 4. Pull the right rocker to the right and the robot will turn right. The farther you pull each time
8 5. Press the button A and the robot will stand/squat. After it stands up, the RGB LED will turn in
9 Note: The robot will only move after it stands up.
10 6. Press button B and the robot now enters the remote control mode. At the same time the LED2 on t
11 7. Press button C to change into the mode of automatic obstacle avoidance. The LED3 lights up when
12 8. Press button D to control the buzzer beep of the module on the robot.
13 Website: www.adeept.com
14 E-mail: support@adeept.com
15 Author: Tom
16 Date: 2017/08/22
17 *****/
18 #include <SPI.h>
19 #include "RF24.h"
20 RF24 radio(9, 10);           // define the object to control NRF24L01
21 byte addresses[5] = "00005"; // define communication address which should correspond to remote co
22 int data[9];                // define array used to save the communication data

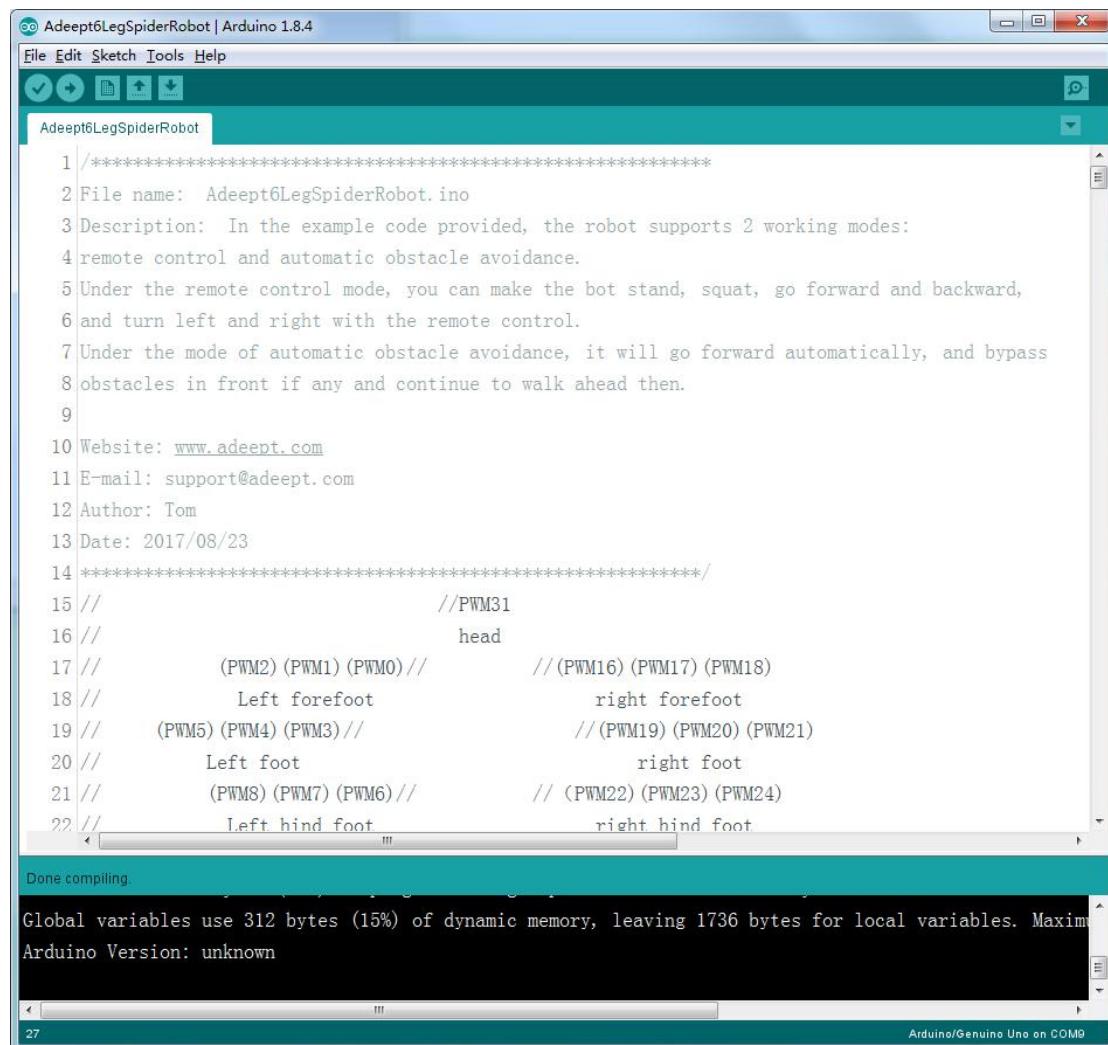
```

The status bar at the bottom shows "34" on the left and "Arduino Nano, ATmega328P on COM10" on the right.

Connect the Arduino Nano to the computer. Select **Tool -> Board: "Arduino Nano"-> Arduino Nano**, and **Port ->COM10**. COMx is the port number assigned to the Arduino Nano and can be COM1, COM2, COM3...So it depends.



Next, click the upload button  . After the program is uploaded to the Nano successfully, upload another sketch to the Adeept UNO R3. Open the program provided for the control board, the file “*Adeept6LegSpiderRobot.ino*”.



```

1 //*****
2 File name: Adeept6LegSpiderRobot.ino
3 Description: In the example code provided, the robot supports 2 working modes:
4 remote control and automatic obstacle avoidance.
5 Under the remote control mode, you can make the bot stand, squat, go forward and backward,
6 and turn left and right with the remote control.
7 Under the mode of automatic obstacle avoidance, it will go forward automatically, and bypass
8 obstacles in front if any and continue to walk ahead then.
9
10 Website: www.adeept.com
11 E-mail: support@adeept.com
12 Author: Tom
13 Date: 2017/08/23
14 *****/
15 // PWM31
16 // head
17 // (PWM2) (PWM1) (PWM0) // // (PWM16) (PWM17) (PWM18)
18 // Left forefoot right forefoot
19 // (PWM5) (PWM4) (PWM3) // // (PWM19) (PWM20) (PWM21)
20 // Left foot right foot
21 // (PWM8) (PWM7) (PWM6) // // (PWM22) (PWM23) (PWM24)
22 // Left hind foot right hind foot

```

Done compiling.

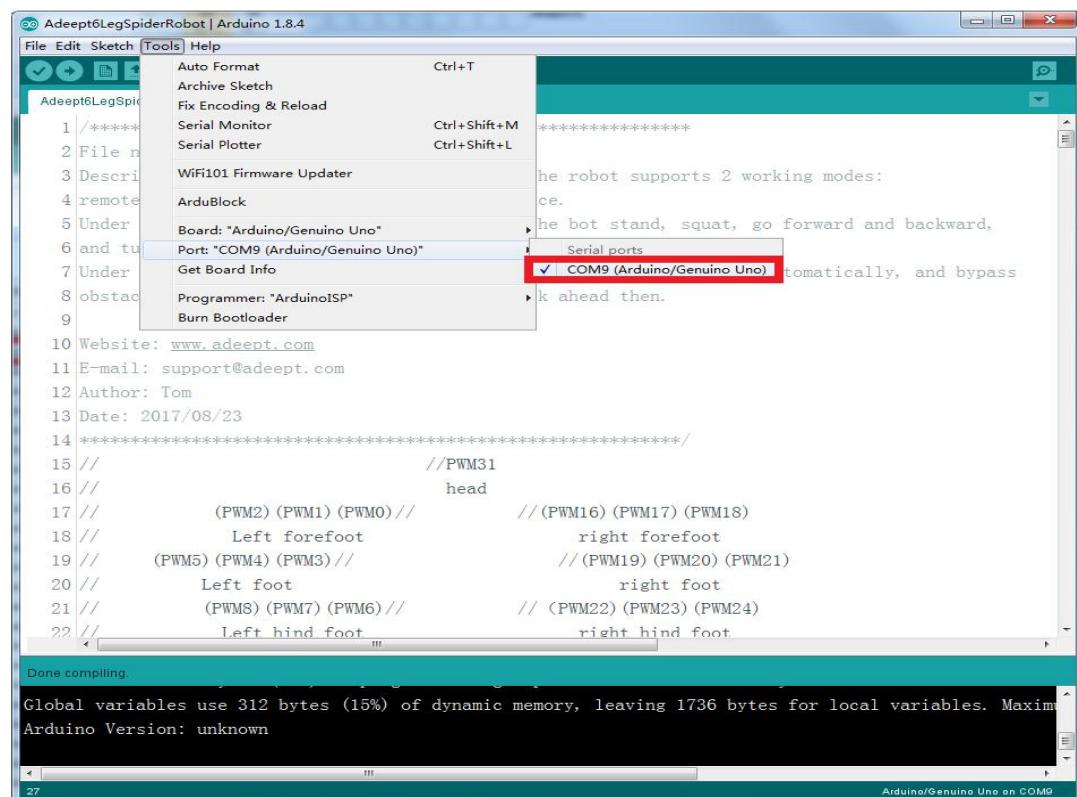
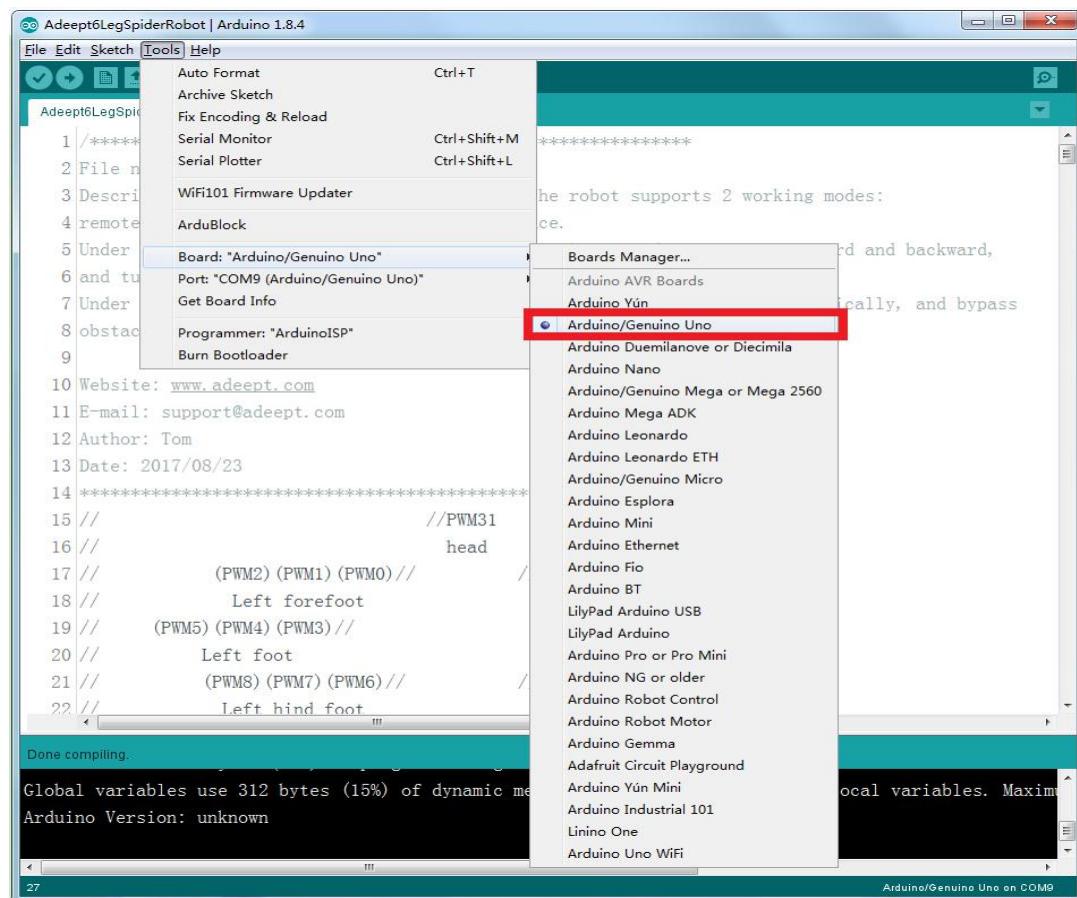
Global variables use 312 bytes (15%) of dynamic memory, leaving 1736 bytes for local variables. Maximum allowed is 2048.

Arduino Version: unknown

27

Arduino/Genuino Uno on COM9

Connect the Arduino UNO R3 board to the PC. Select **Tool -> Board “Arduino/Genuino Uno”**, and **Port -> COM9**. Also here is COM9, assigned to the Uno, but it can be COM1, COM2, COM3...



Click the button  to upload the sketch to the board.

## Afterword

Thanks for purchasing our product and reading the manual! If you spot any errors or have any ideas or questions for the product and this guide, welcome to contact us! We will correct them if any as quickly as possible.

After completing all projects in the guide, you should have some knowledge of the book and Arduino, thus you can try to change the robot into other projects by adding more Adeept modules or changing the code for extended functions.

For more information about Arduino, Raspberry Pi, smart car robot, or robotics, etc., please follow our website [www.adeept.com](http://www.adeept.com). We will introduce more cost-effective, innovative and intriguing products!

Thanks again for choose Adeept product!



Adeept

Sharing Perfects Innovation

E-mail:[support@adeept.com](mailto:support@adeept.com)

website:[www.adeept.com](http://www.adeept.com)