Free your innovation

Freenove is an open-source electronics platform.
www.freenove.com

Warning

When you purchase or use this product, please note the following:

- This product contains small parts. Swallowing or improper operation them can cause serious infections and death. Seek immediate medical attention when the accident happened.
- Do not allow children under 3 years old to play with or near this product. Please place this product in where children under 3 years of age cannot reach.
- Do not allow children lack of ability of safe to use this product alone without parental care.
- Never use this product and its parts near any AC electrical outlet or other circuits to avoid the potential risk of electric shock.
- Never use this product near any liquid and fire.
- Keep conductive materials away from this product.
- Never store or use this product in any extreme environments such as extreme hot or cold, high humidity and etc.
- Remember to turn off circuits when not in use this product or when left.
- Do not touch any moving and rotating parts of this product while they are operating.
- Some parts of this product may become warm to touch when used in certain circuit designs. This is normal. Improper operation may cause excessively overheating.
- Using this product not in accordance with the specification may cause damage to the product.

About

Freenove is an open-source electronics platform. Freenove is committed to helping customer quickly realize the creative idea and product prototypes, making it easy to get started for those enthusiasts of programing and electronics and launching innovative open source products. Our services include:

- Electronic components and modules
- Learning kits for Arduino
- Learning kits for Raspberry Pi
- Learning kits for Technology
- Robot kits
- Auxiliary tools for creations

Our code and circuit are open source. You can obtain the details and the latest information through visiting the following web sites:
http://www.freenove.com
https://github.com/freenove

Your comments and suggestions are warmly welcomed, please send them to the following email address:
support@freenove.com

References

You can download the sketches and references used in this product in the following websites:
http://www.freenove.com
https://github.com/freenove
If you have any difficulties, you can send email to technical support for help.

The references for this product is named Freenove Three-wheeled Smart Car Kit for Arduino, which includes the following folders and files:

- Libraries Libraries for Arduino sketches
- Sketches Sketches for Arduino and Processing projects
- Readme.txt Instructions
- Tutorial.pdf Tutorial

Support

Freenove provides free and quick technical support, including but not limited to:

- Quality problems of products
- Problems in using products
- Questions for learning and technology
- Opinions and suggestions
- Ideas and thoughts

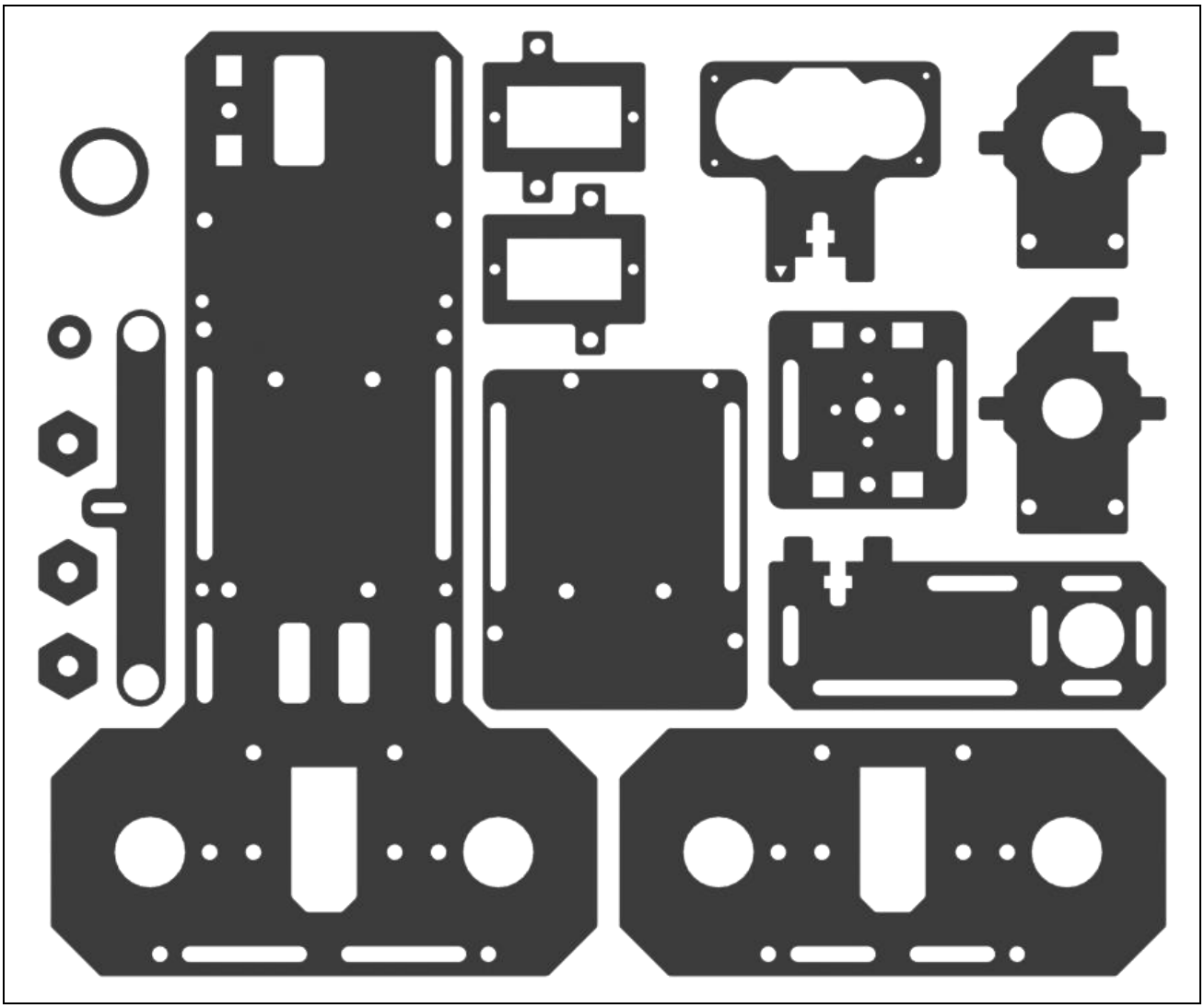
Please send email to:
support@freenove.com
On working day, we usually reply to you within 24 hours.

Copyright

Contents

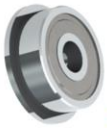
List

Acrylic Sheet

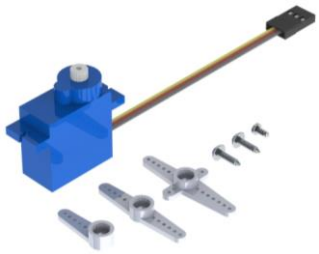


The surface of the acrylic sheet is covered with a layer of protective film, you need to remove it first.
Some holes in the acrylic sheet may have residues, so you need to clean them before using.

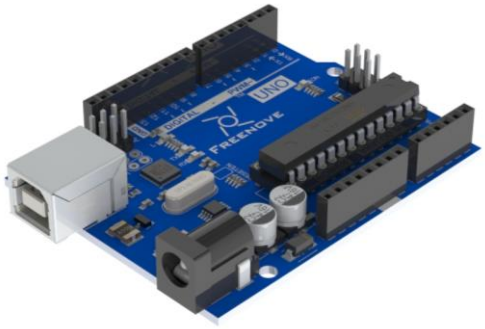
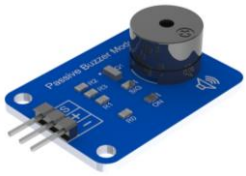
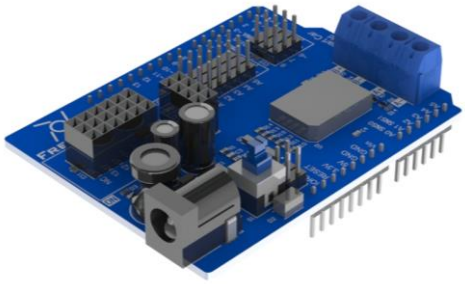
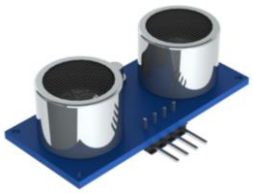
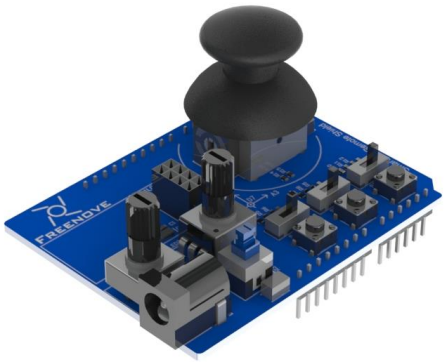
Machinery Parts

M4 Washer  x4 Freenove	M2 Nut  x6 Freenove	M3 Nut  x12 Freenove	M4 Nut  x2 Freenove	M2*10 Screw  x6 Freenove	M3*4 Screw  x12 Freenove
M3*8 Screw  x30 Freenove	M3*12 Screw  x4 Freenove	M3*30 Screw  x6 Freenove	M4*40 Screw  x2 Freenove	M3*10 Countersunk Head Screw  x6 Freenove	M1.4*6 Self-tapping Screw  x4 Freenove
F624ZZ Bearing  x2 Freenove	F687ZZ Bearing  x4 Freenove	M3*6 Copper Standoff  x14 Freenove	M3*12 Copper Standoff  x6 Freenove	M3*30 Copper Standoff  x6 Freenove	M4 Spring Washer  x4 Freenove

Transmission Parts

Servo x2 	Driven wheel x1 
DC speed reduction motor x2 	Driving wheel x2 

Electronic Parts

Freenove UNO x2 	Freenove Passive Buzzer Module x1 	Freenove RGB LED Module x1 
Freenove Smart Car Shield for Arduino UNO x1 	NRF24L01 Module x2 	HC-SR04 Ultrasonic Module x1 
Freenove Smart Car Remote Shield for Arduino UNO x1 	18650x2 Battery Holder x2 	
USB Cable x1 		
Jumper Wire M/M x10 		

Tools

Cross screwdriver x1	Slotted screwdriver x1	Socket x1
		

Self-prepared Parts

At least two 18650 lithium batteries (pointed head) without protection board.
Four batteries are recommended.

Preface

Welcome to the Freenove smart car kit! No matter whether you are a senior maker, or novice with little professional technology, this tutorial will help you to make something really cool: a smart car with remote control, automatic driving and exploration mode.

This kit is based on the popular open source electronics platform Arduino.You can share and exchange your experience and design ideas with enthusiasts all over the world. All components in this kit are individually packaged, including all the electronic components, modules and machanical components needed in these projects. Meanwhile, there are detailed assembly and commissioning instruction in this tutorial. To ensure that your smart car can be smoothly assembled and run, you can seek free and quick technical support at any time, if you have any problems.

The contents of this tutorial can ensure novice with no professional technology to accomplish the target. If you are interested into the Arduino, and want to learn how to program and build the circuit, please visit our website or contact with us to buy Freenove Starter Kit for Arduino which is specially prepared for the beginners.

Arduino

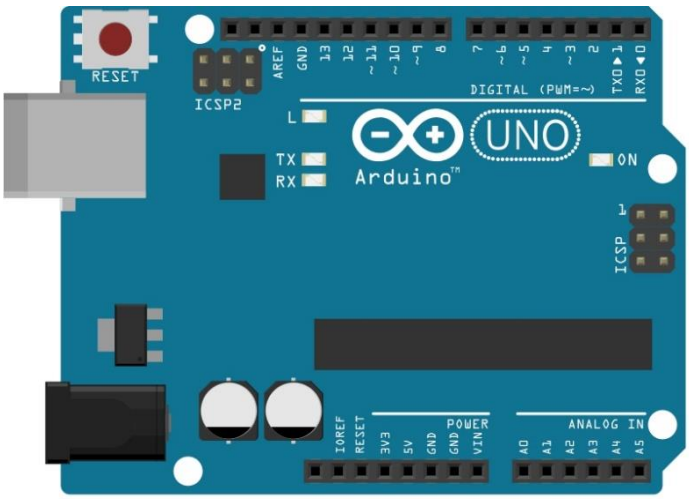
Arduino is an open-source electronics platform based on easy-to-use hardware and software. It's intended for anyone making interactive projects. Usually, an Arduino project consists of circuit and code.

Arduino Board

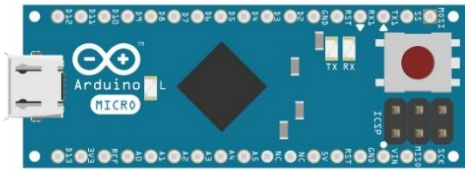
Arduino Board is a circuit board, with integrates micro controller, input, output interface and etc. Arduino Board can use the sensor to sense the environment and receive user's operation to control LED, motor rotation, etc. We just need to assembly circuit and write the code.

Currently, Arduino Board has several models, and the code between boards of different types is universal (some boards may not be completely compatible because of the differences in hardware). Popular boards include:

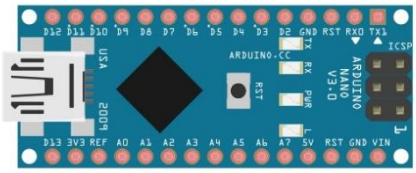
UNO



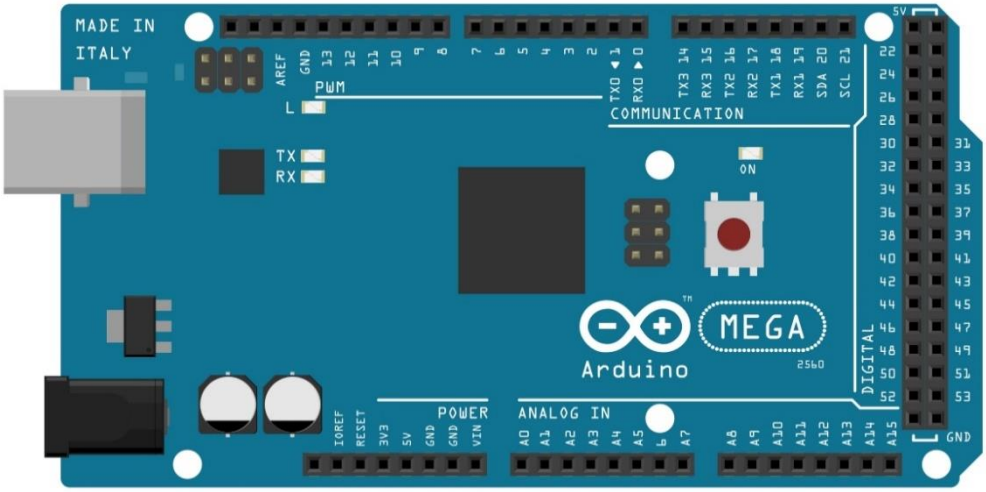
MICRO



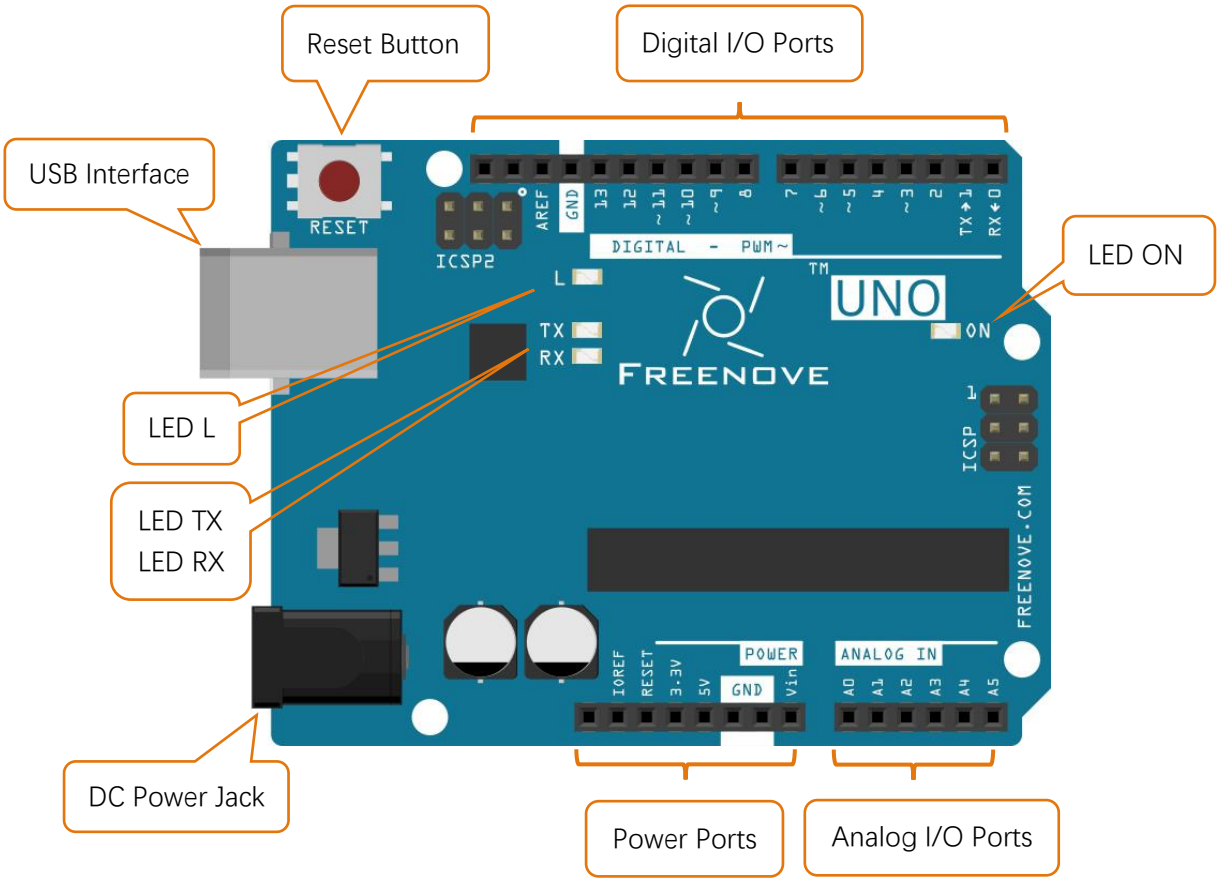
NANO



MEGA



The board used in this tutorial is Freenove UNO, and it is fully compatible to Arduino UNO. Diagram of Freenove UNO board is shown below:



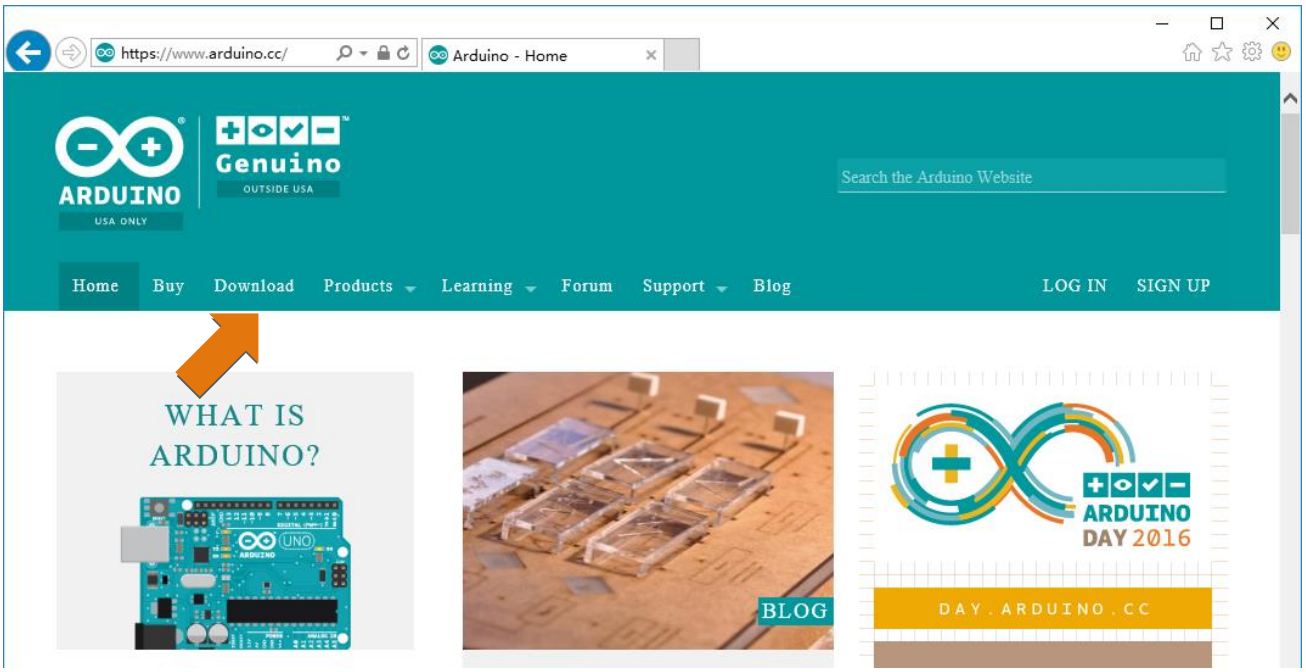
- Digital I/O Ports is used to connect to other components or modules, to receive an input signal, or to send a control signal. Usually, we name it by adding a "D" in front of the number, such as D13.
- USB Interface is used to provide power, upload code or communicate with PC.
- LED L is connected to digital I/O port 13 (D13).
- LED TX, RX is used to indicate the state of the serial communication.
- DC Power Jack is connected DC power to provide power for the board.
- Power Ports can provide power for electronic components and modules.
- Analog I/O Ports can be used to measure analog signals.
- LED ON is used to indicate the power state.

Freenove UNO is the most suitable board to complete this kit. You can also choose to use Arduino UNO (or other board compatible to it).

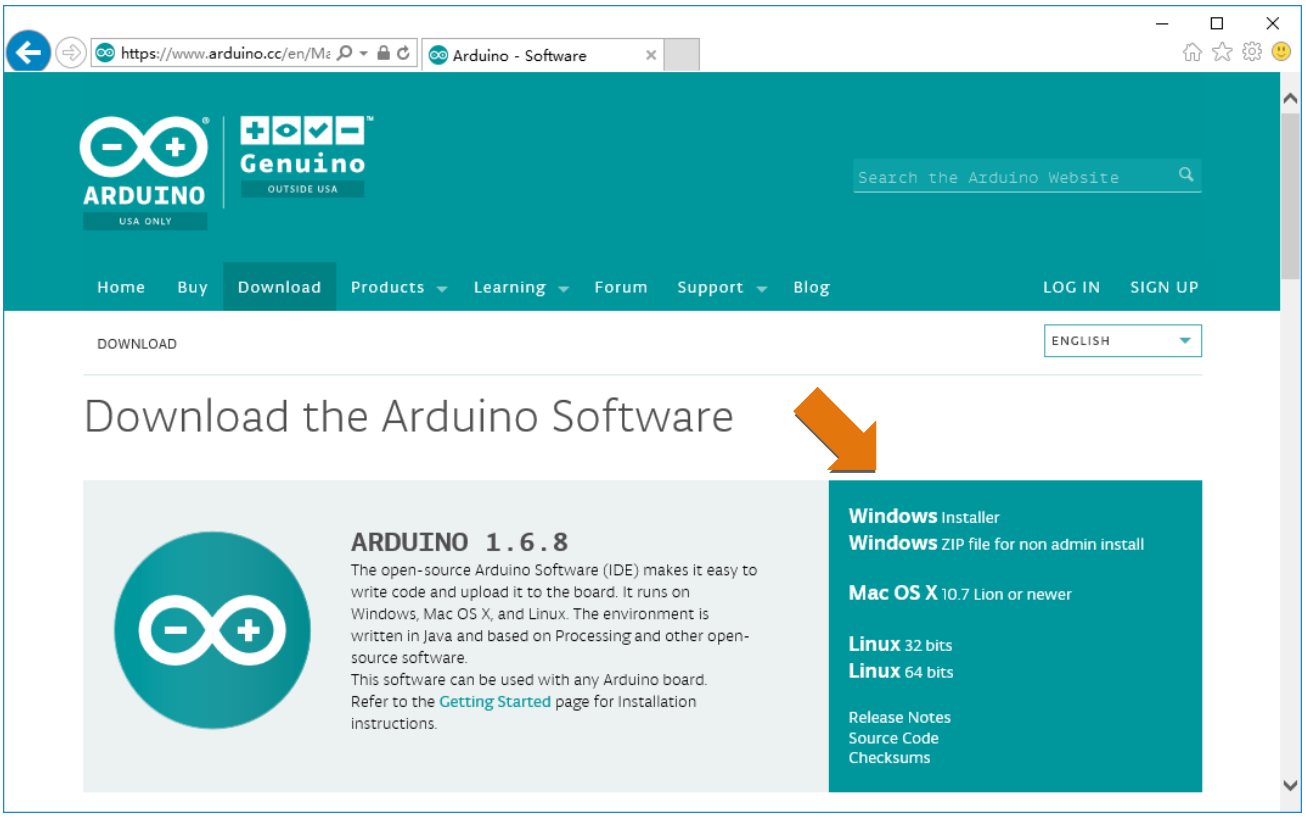
Arduino Software

Arduino Software (IDE) is used to write and upload the code for Arduino Board.

First, install Arduino Software (IDE): visit https://www.arduino.cc, click "Download" to enter the download page.

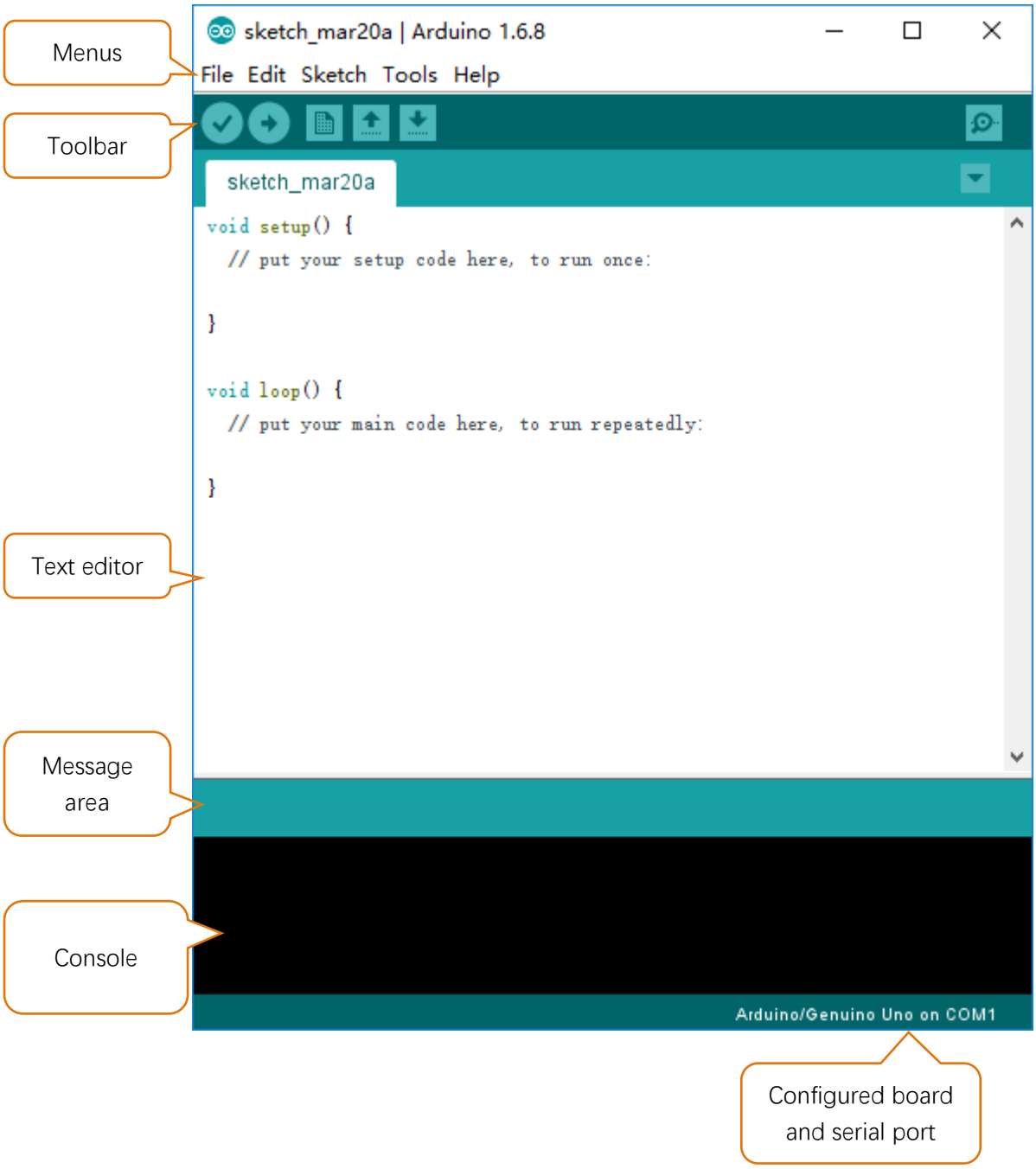


Download corresponding installation program according to your operating system. If you are a Windows user, please select the "Windows Installer" to download and install the driver correctly.



After the download completes, run the installer. For Windows users, there may pop up a installation dialog box of driver during the installation process. When it is popped up, please allow the installation.
After installation is complete, an Arduino Software shortcut will be generated in the desktop. Run the Arduino Software.

The interface of Arduino Software is as follows:



Programs written using Arduino Software (IDE) are called **sketches**. These sketches are written in the text editor and are saved with the file extension **.ino**. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom righthand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

Verify
Checks your code for errors compiling it.

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

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.

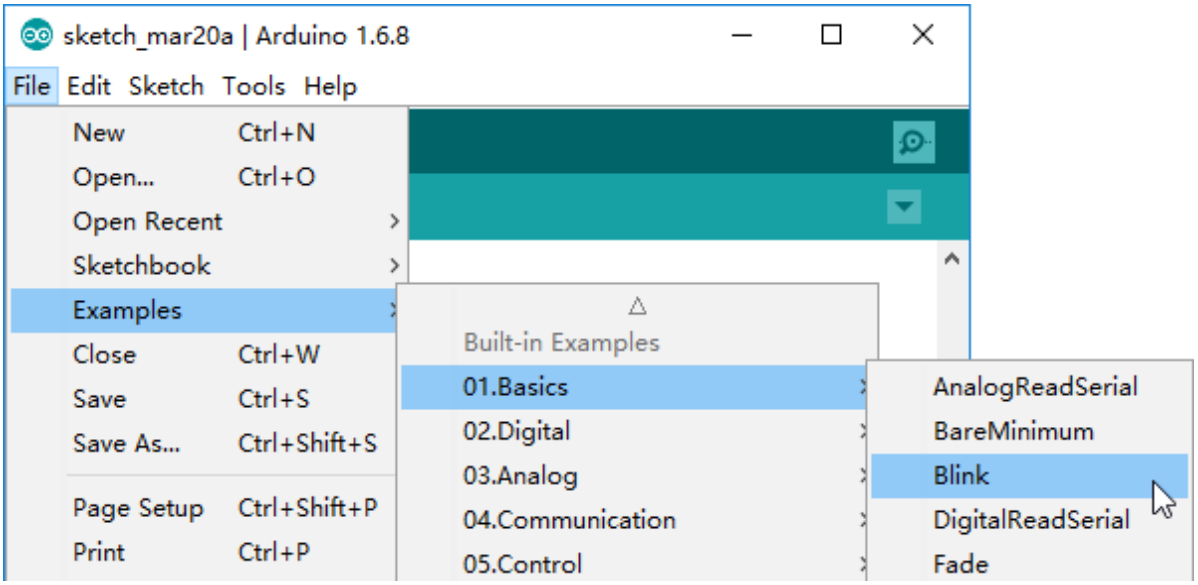
Save
Saves your sketch.

Serial Monitor
Opens the serial monitor.

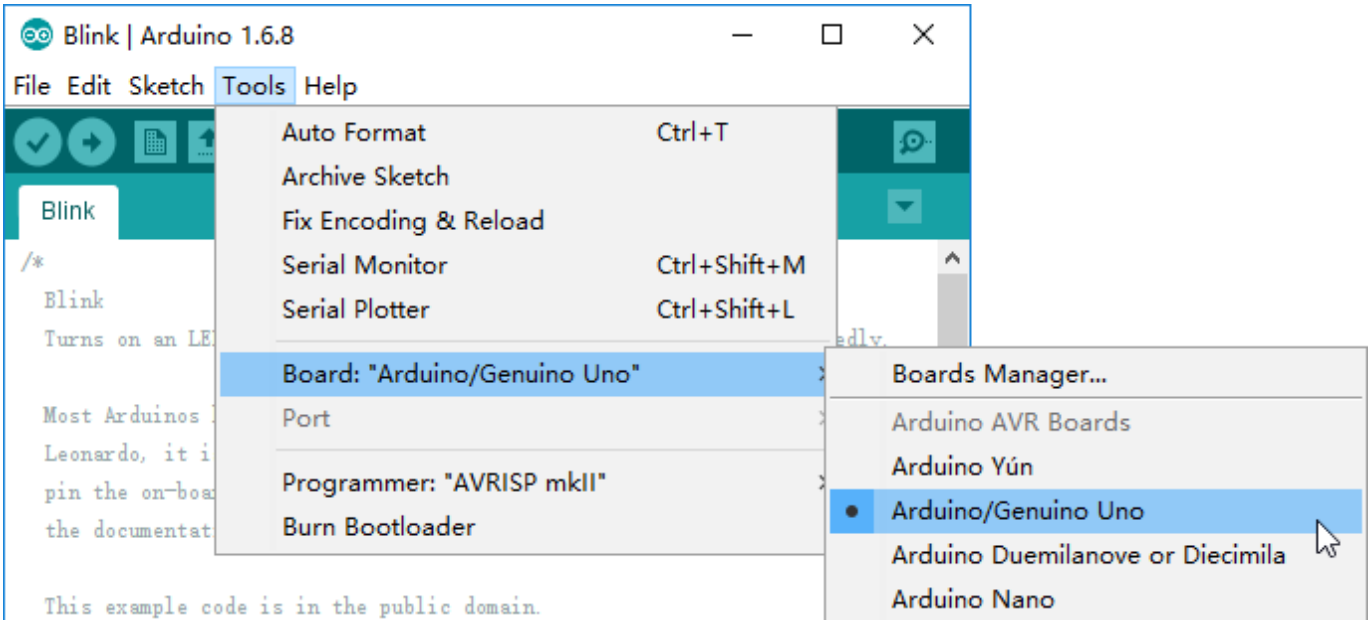
Additional commands are found within the five menus: File, Edit, Sketch, Tools, Help. The menus are context sensitive, which means only those items relevant to the work currently being carried out are available.

First Use

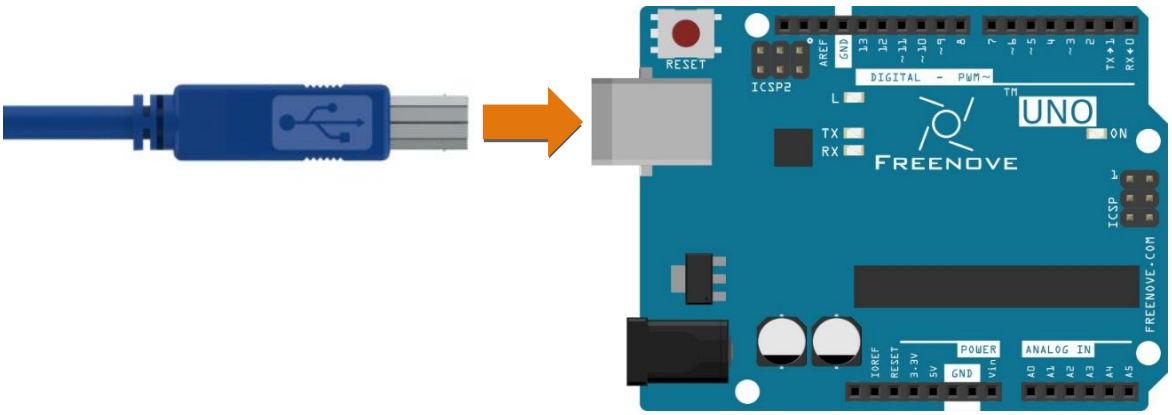
Open the exmple sketch "Blink" with Arduino Software.



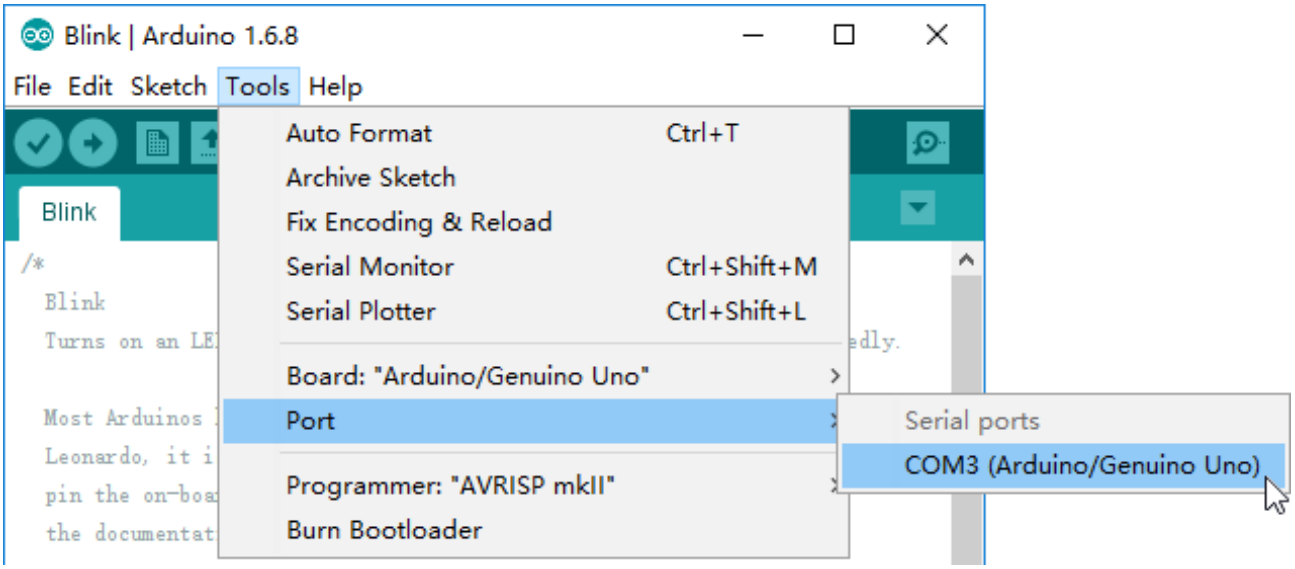
Select board "Arduino/Genuino Uno".



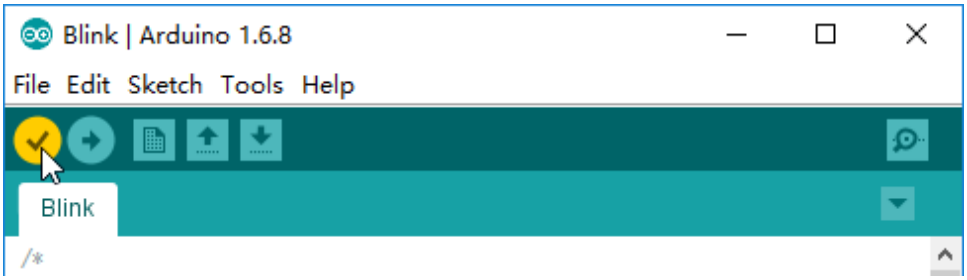
Connect Freenove UNO to computer with USB cable.



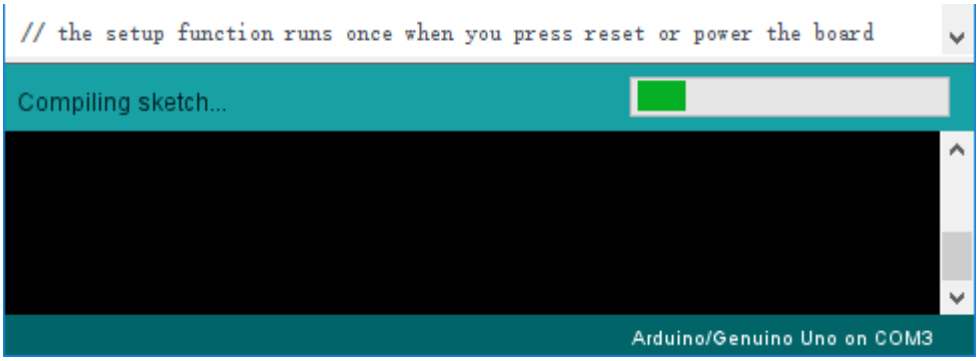
Select the serial port. Your serial number may be different from the following figure. If it is not detected immediately, please wait for a while, then click "Tools" to check again.



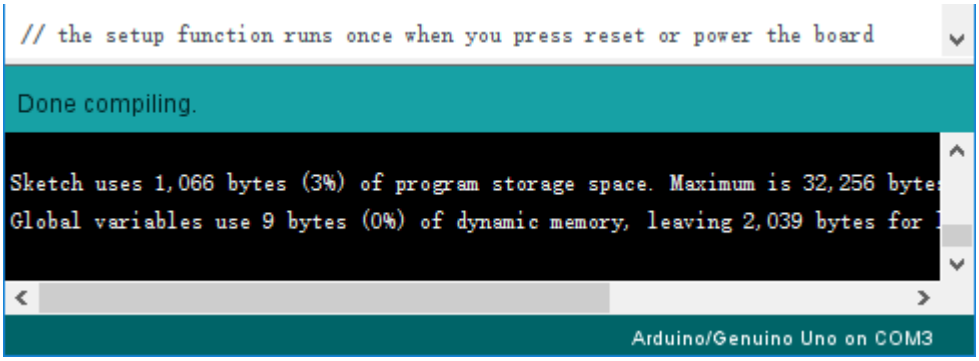
Click "Verify" button.



The following figure shows the code is being compiled.



Wait a moment for the compiling to be completed. Figure below shows the code size and percentage of space occupation.



Usually, when we write code, if it has a syntax error, the interface will prompt the error message. Then the compiling can't be completed.

Click "Upload" button.

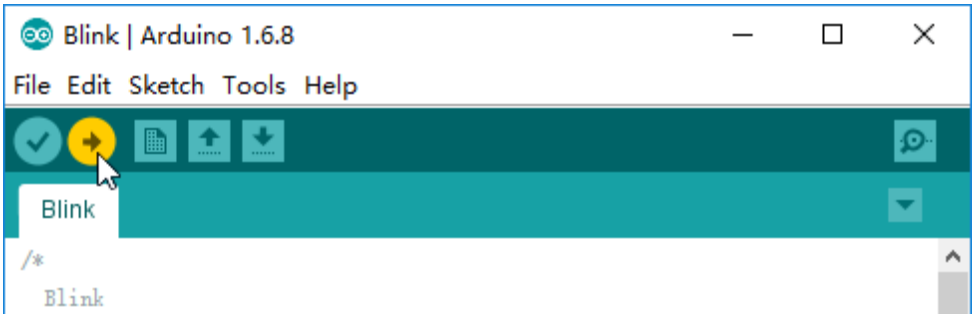
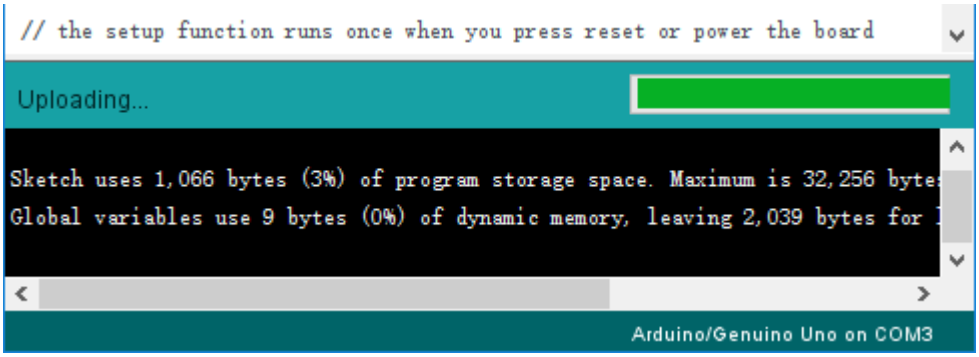
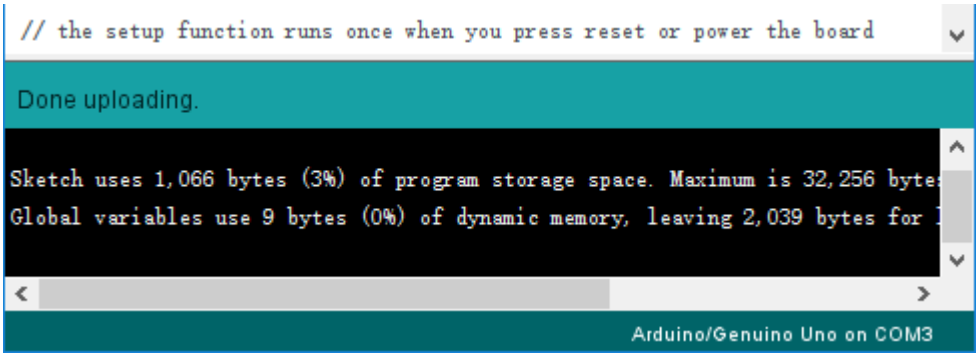


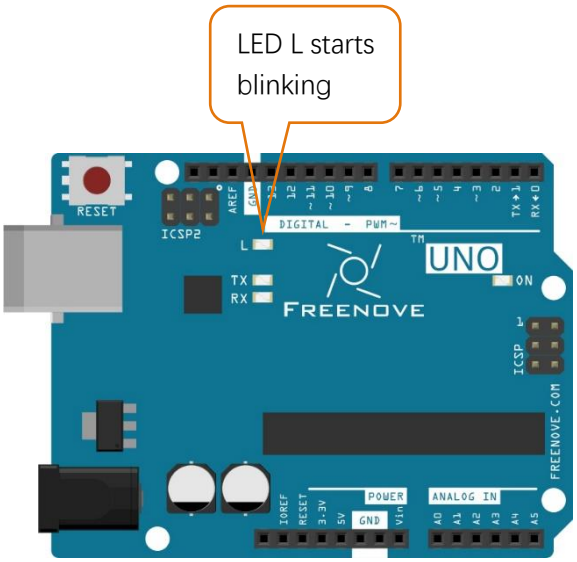
Figure below shows code are uploading.



Wait a moment, then the uploading is completed.



After that, we will see the LED marked with "L" on Freenove UNO starts blinking. It indicates that the code is running now!



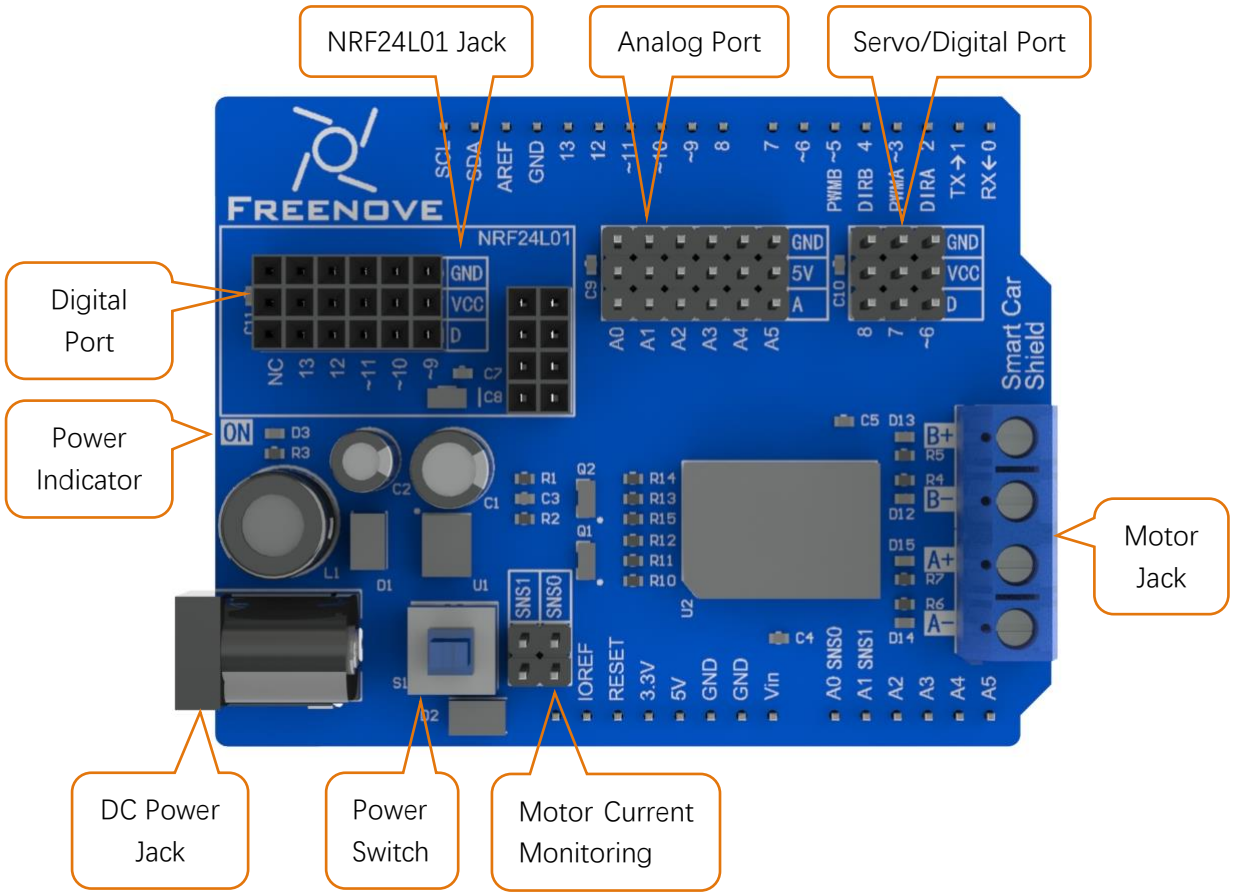
So far, we have completed the first use. I believe you have felt the joy of it. Next, we will use Arduino to control the smart car.

1 Freenove Expansion Shield

Generally, a project that is completed by Arduino consists of code and circuit, and it can be simplified by special expansion shield. We can insert the expansion shield directly onto Arduino board, or insert the Arduino board onto the expansion shield.

Freenove Smart Car Shield for Arduino UNO

Freenove Smart Car Shield for Arduino UNO is a special expansion shield designed to control the smart cars. It applies to Arduino UNO or other compatible control board.

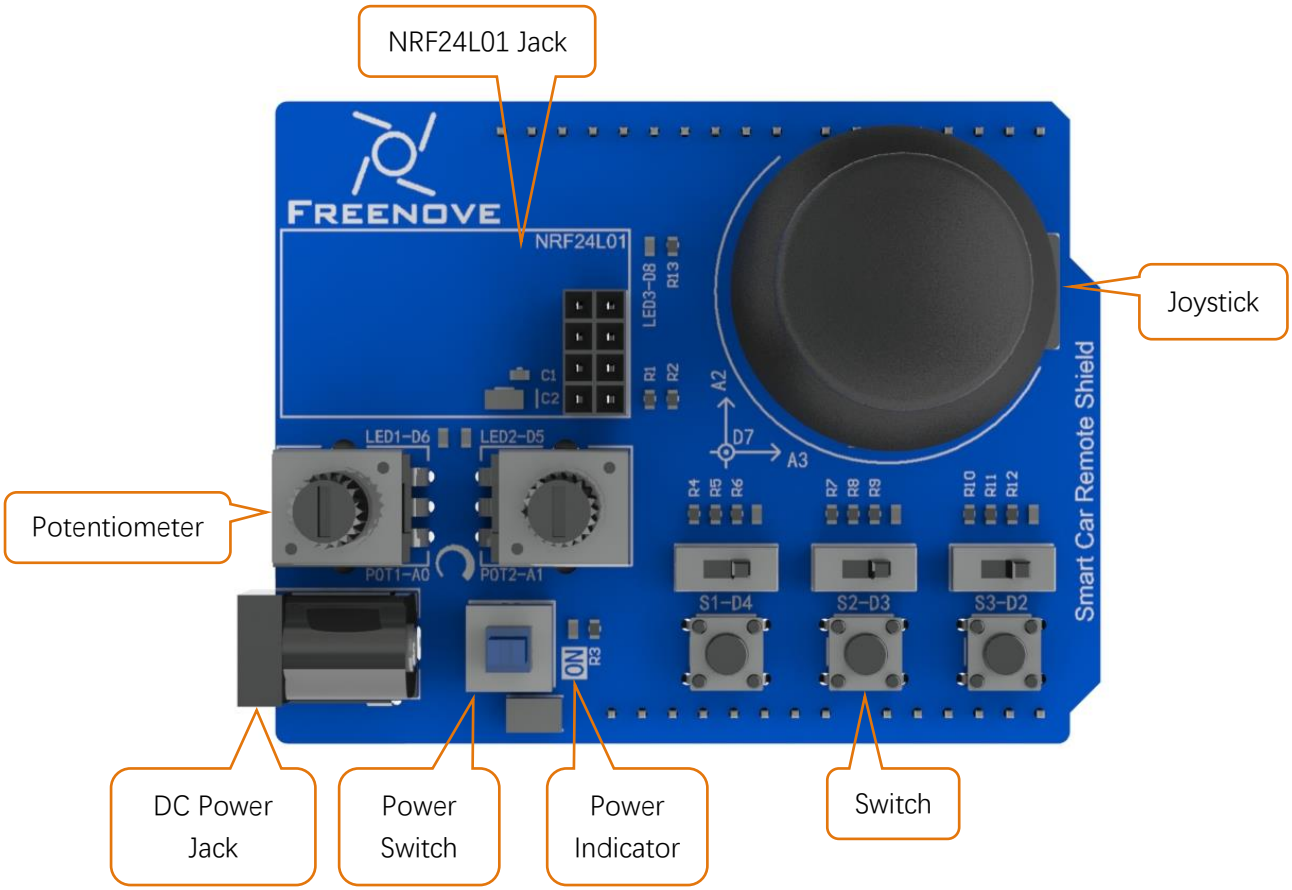


The main functions of the expansion shield are as follows:

- Power: Expansion shield uses 7~12V DC power, and can output 5V 3A power, which is marked with "VCC". Meanwhile, provide power supply for UNO board, and lead 5V power on UNO board to the expansion shield and mark it with "5V".
- Power Switch: A switch on the expansion shield to control the on/off of power, and has 1 LED indicator for power state.
- Motor Control: Control steering and speed of two motors. Output 4A current at maximum, and each motor outputs 3A current at maximum. Four LEDs on the motor interface indicates the state of each channel.
- Servo Jack: Make it very convenient to connect to servo to control the pan-tilt and steering of smart car.
- NFR24L01 Jack: Make it easy to connect to NFR24L01 module, and control the smart car wirelessly.
- Digital Port: Lead unoccupied digital port out. NRF24L01 jack uses digital port 9~13, so these interfaces can't be used when use the NRF24L01 module.
- Analog Port: Lead unoccupied analog port out. Analog interface uses power from UNO board to reduce the interference. Monitoring for motor current uses analog port A1, A0, therefore, these part interfaces can't be used when use this function.

Freenove Smart Car Remote Shield for Arduino UNO

Freenove Smart Car Remote Shield for Arduino UNO is a special expansion shield used for remote control of smart car. It applies to Arduino UNO or other compatible control board.



The main functions of the expansion shield are as follows:

- Power: Expansion shield uses 7~12V DC power, and provide power for UNO board. The expansion shield can use the power from UNO board, so the expansion board can work normally when UNO board is powered.
- Power Switch: A switch on the expansion shield to control the on/off of power, and has 1 LED indicator for power state. It won't work if expansion shield gets powered from UNO.
- NFR24L01 Jack: Make it easy to connect to NFR24L01 module, and control the smart car wirelessly.
- Joystick: Integrated with 2D analog, and can be pressed in a vertical direction. With this joystick, you can control the smart car to run at different speed and direction.
- Switch: Include tact switch and toggle switch. When you need to trigger the short-time function, you can use a tact switch, for long-time function, you can use the toggle switch. Beside each switch, there is a LED indicator for switch state. These switches can control other electronic module of smart car or set custom functions.
- Potentiometer: Beside each potentiometer there is a LED indicator for the potentiometer state. These potentiometers can control other electronic module of smart car or set custom functions.

2 Remote Control Mode

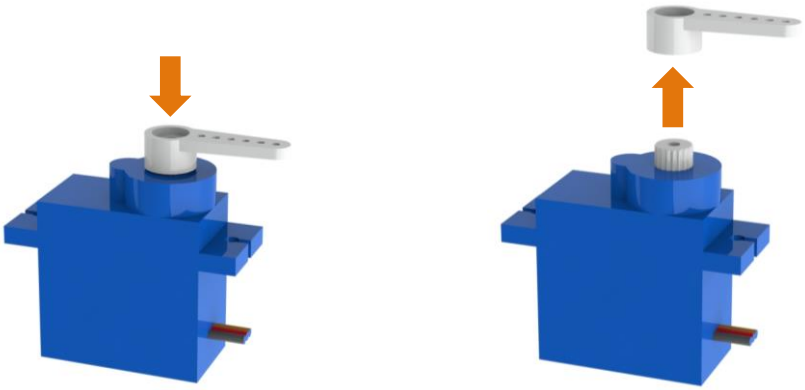
First, let's use the remote control to control the smart car.

Step 2.1 Adjust the Servo

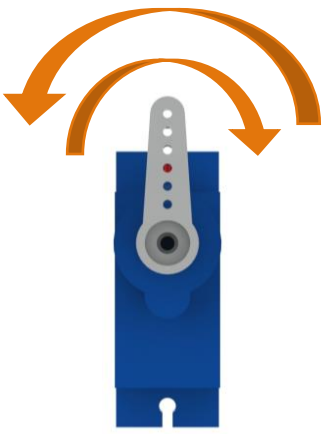
First, get to know about the servo.

The servo can be connected with the rocker arm, and can drive other parts to move through the rocker arm. There are 3 kinds of rocker arm, and 3 screws for the servo. The smaller one screw is used to fix the rocker arm onto it.

Install or remove the rocker arm.

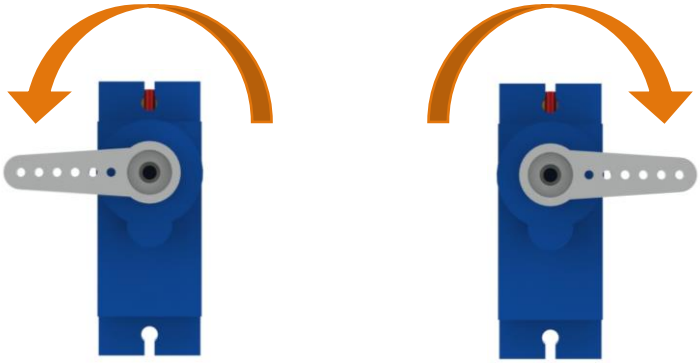


Turn the rocker arm to make it rotate in the range from 0 to 180 degrees.

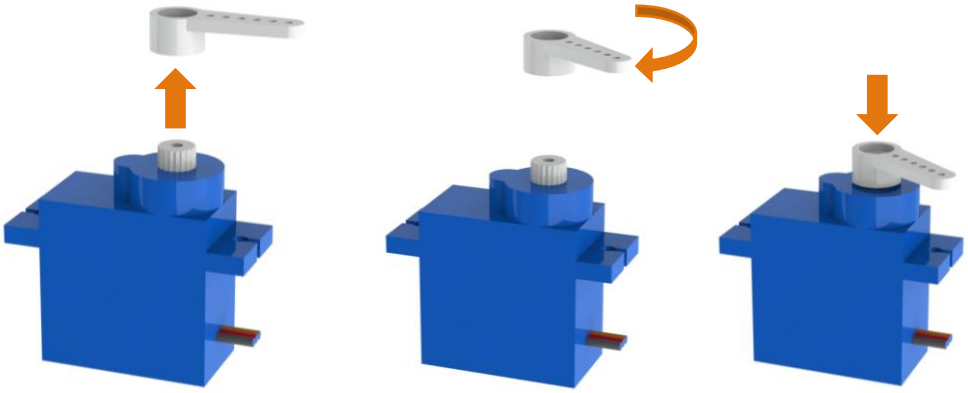


Now adjust the servo. The purpose we do this is to rotate the shaft of servor to the middle position, and ensure that the servo can drive the corresponding parts to move in the desired range.

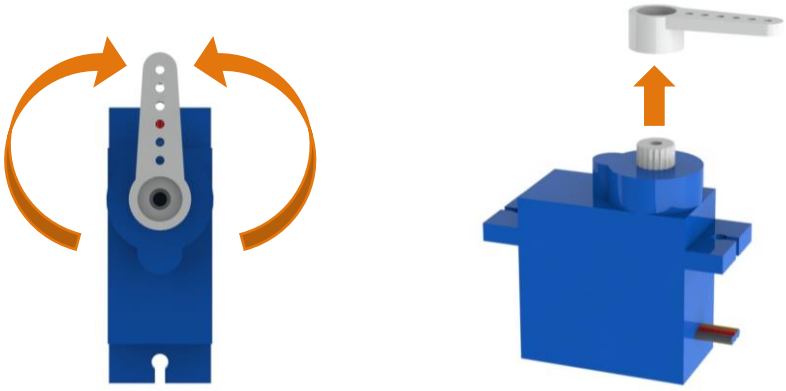
We need to adjust the servo, so that the rocker arm can basically rotate the same angle to the right or left.



If the rocker arm can't rotate around the same angle, remove the rocker arm and reinstall it from other angle. Turn the rocker arm to both sides and see whether it meets the requirement. Repeat this operation until the rocker arm can basically rotate the same angle to the right or left.



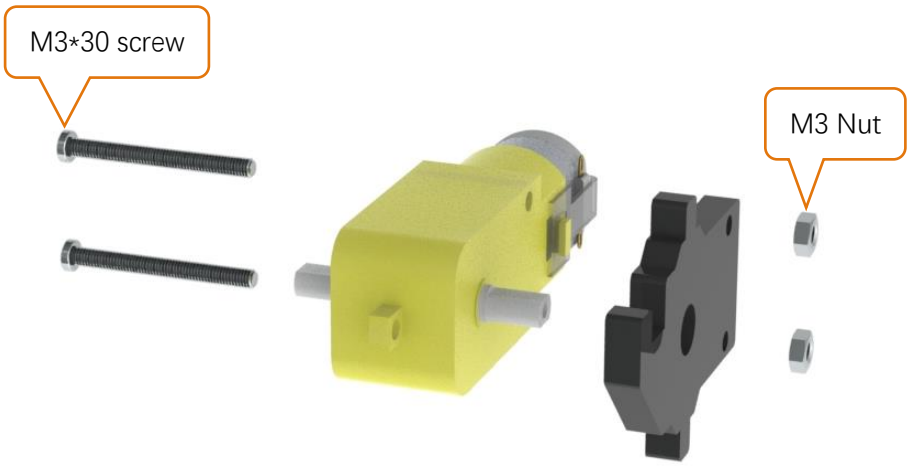
After the adjustment is finished, turn the rocker arm to the middle position, and take it off.



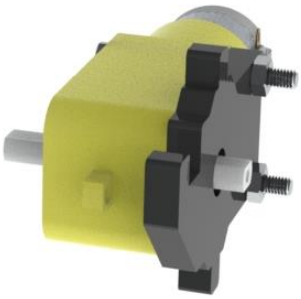
Make sure that all servos have been adjusted, and no unexpected rotation of the shaft happen in the following installation process. If it happens , you will need to adjust it once again before installing.

Step 2.2 Assemble the Smart Car

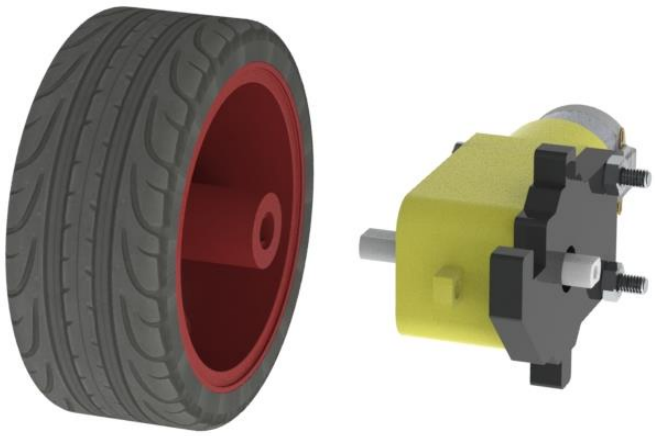
Assemble the following components



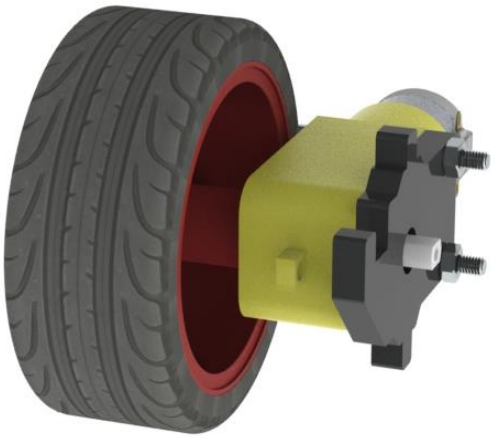
Effect diagram after assembling



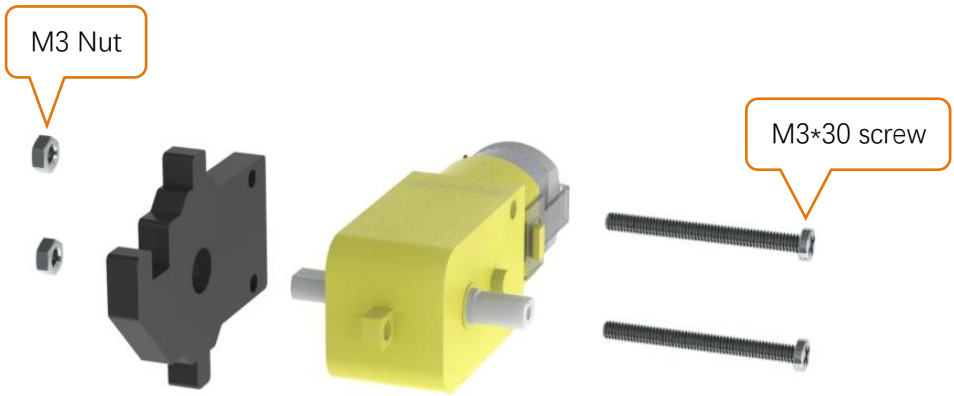
Install the wheel(Left)



Effect diagram after assembling



Assemble the following components

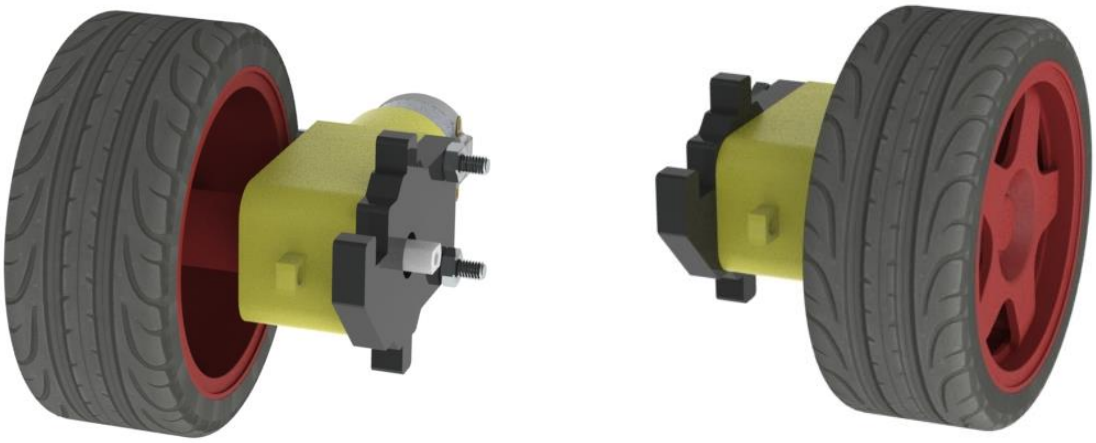


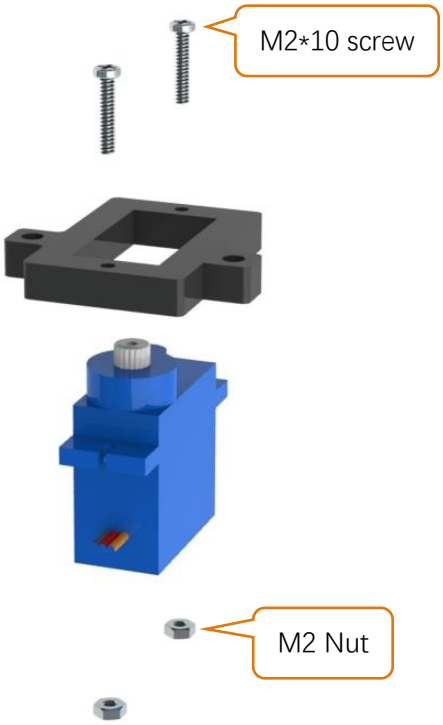
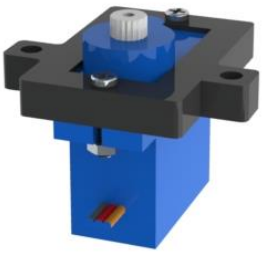
Effect diagram after assembling

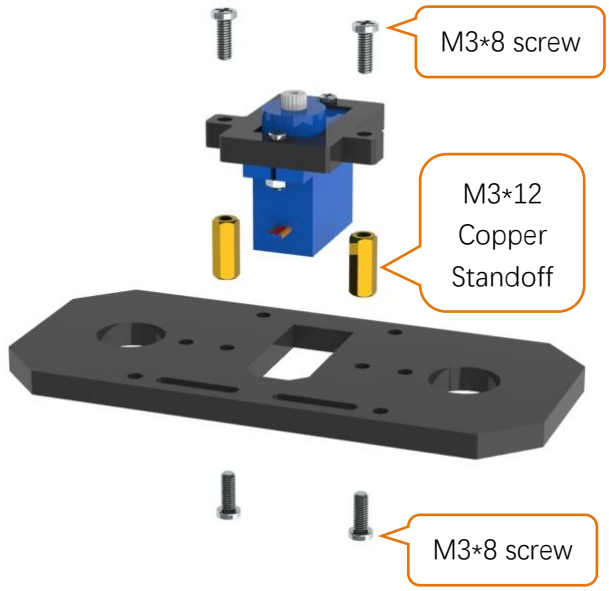
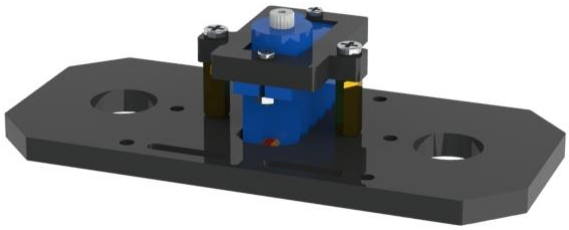
Install the wheel(Right)

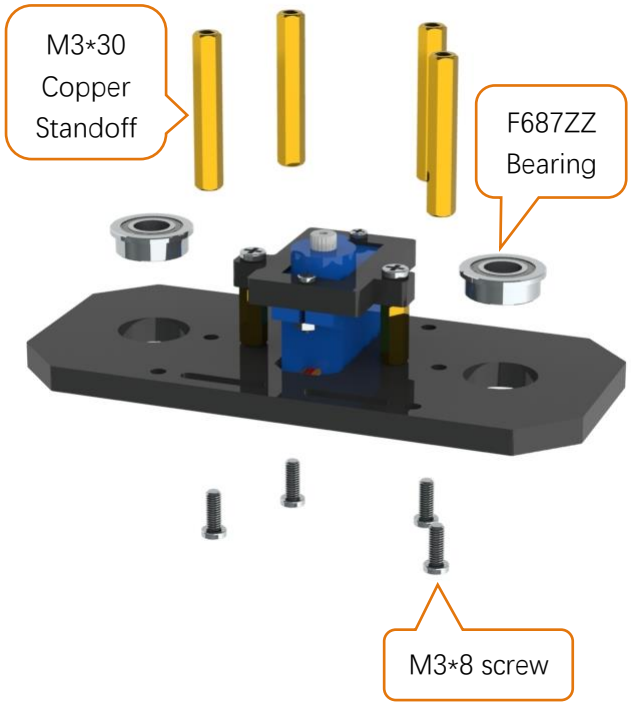
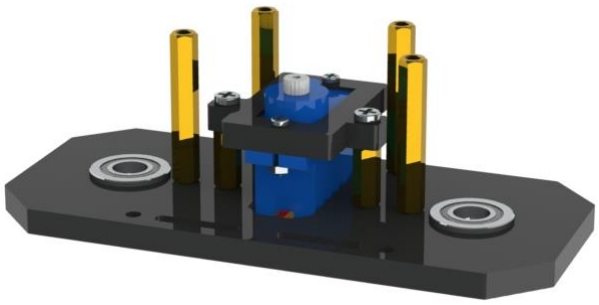
Effect diagram after assembling

After finishing the previous installing, you'll get two pats as follows:

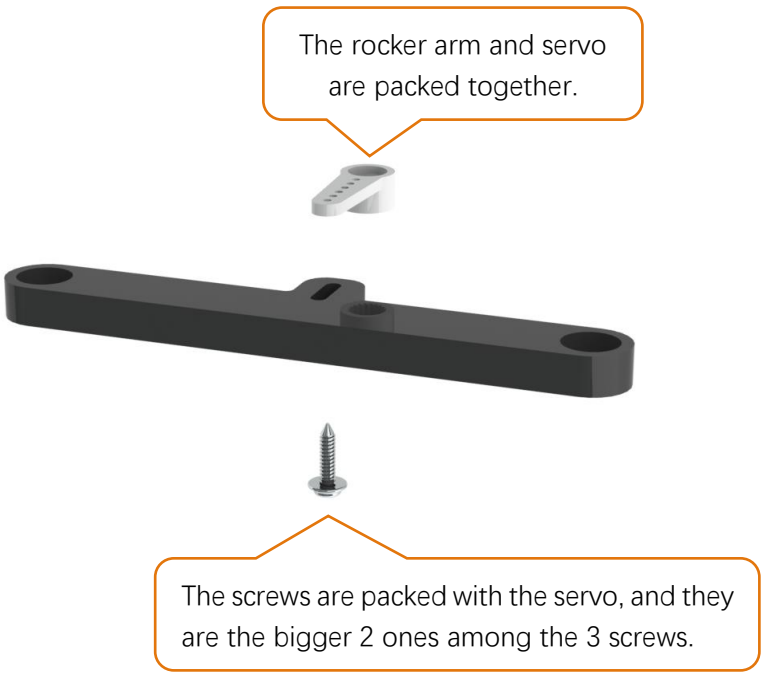


Assemble the following components	Effect diagram after assembling
	

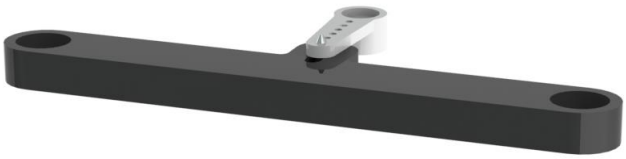
Assemble the following components	Effect diagram after assembling
	

Assemble the following components	Effect diagram after assembling
	

Assemble the following components



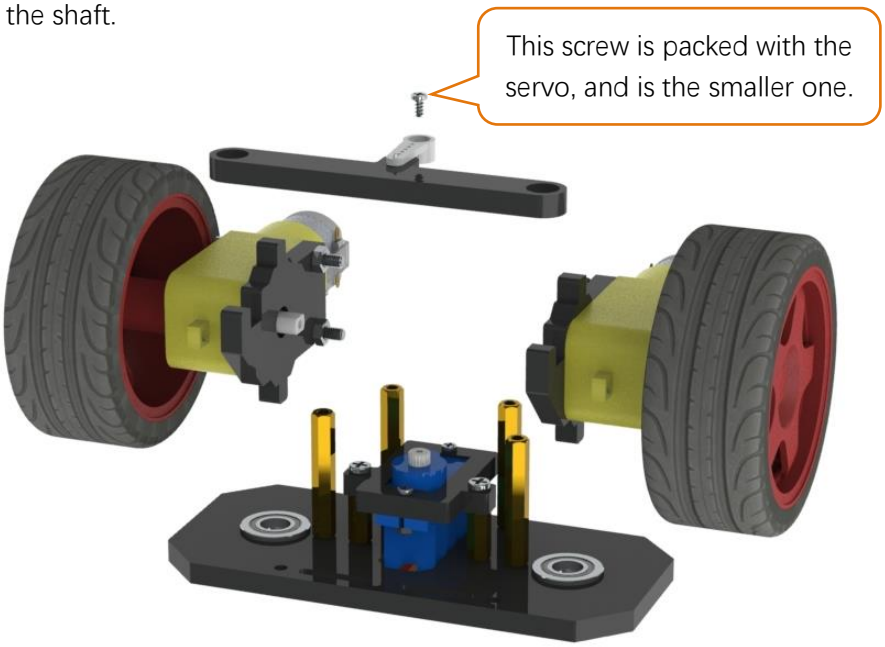
Effect diagram after assembling



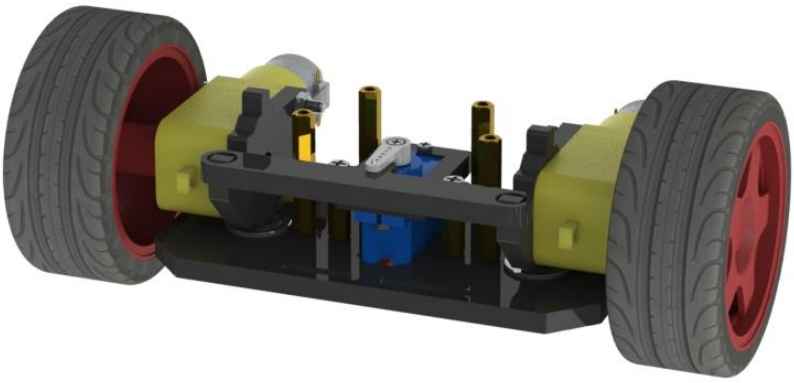
Pay attention not to tighten the screw, otherwise the acrylic plate can't move freely.

Assemble the following components

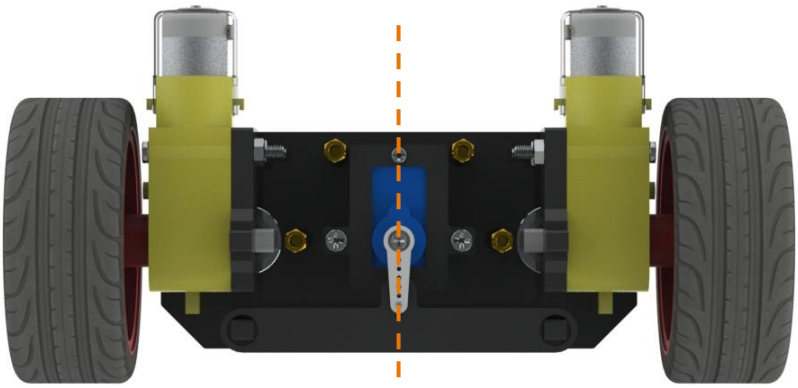
The rocker arm should be installed in the middle position between its rotation range. A little deviation is acceptable. If it is not installed in the middle position, you should remove the rocker arm and install it again instead of turning the shaft.



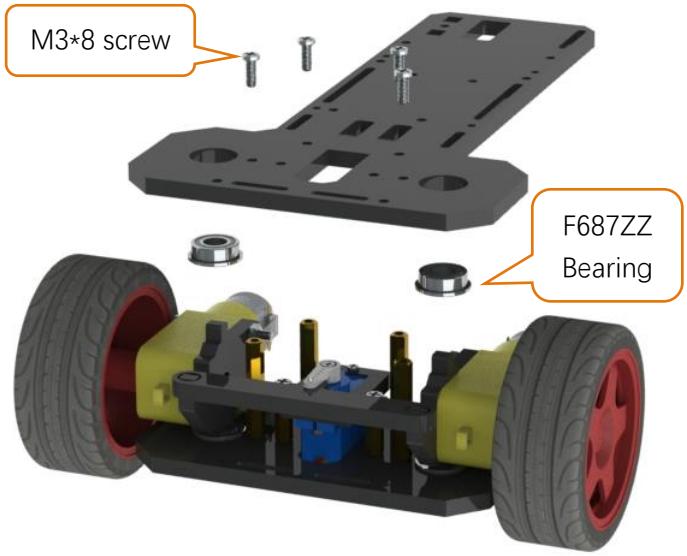
Effect diagram after assembling



When we look the front axle from vertical direction, the rocker arm should be installed in the middle position between its rotation range. If you turn the shaft by accident, you need adjust the servo according to Step 2.1 and install it again.

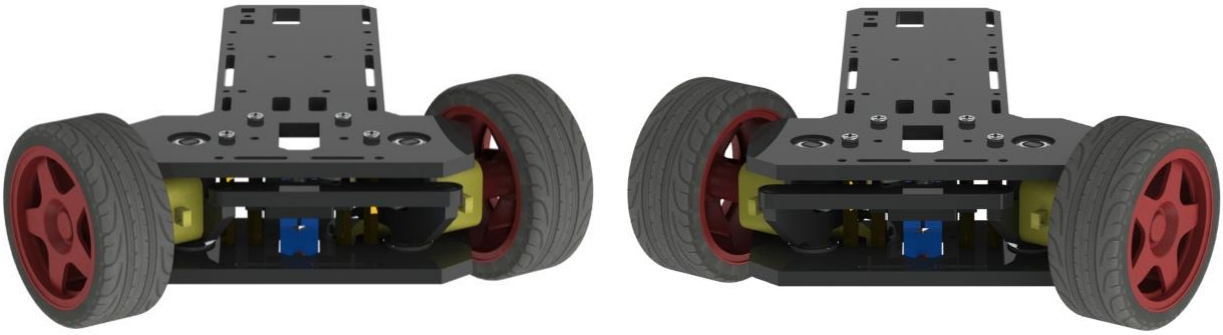


Assemble the following components

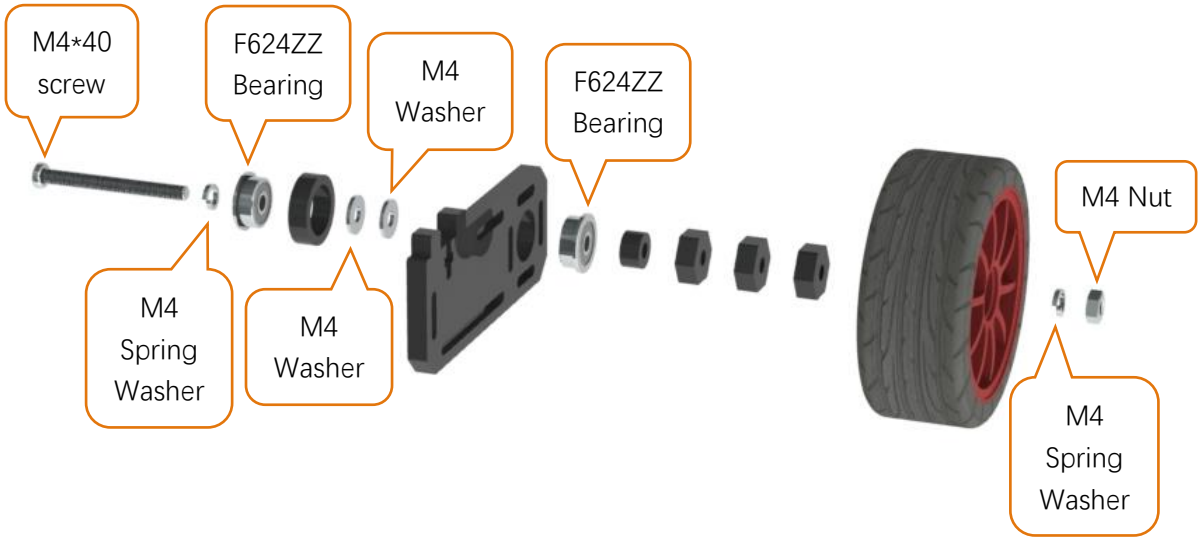


Effect diagram after assembling

The two wheels can turn left and right

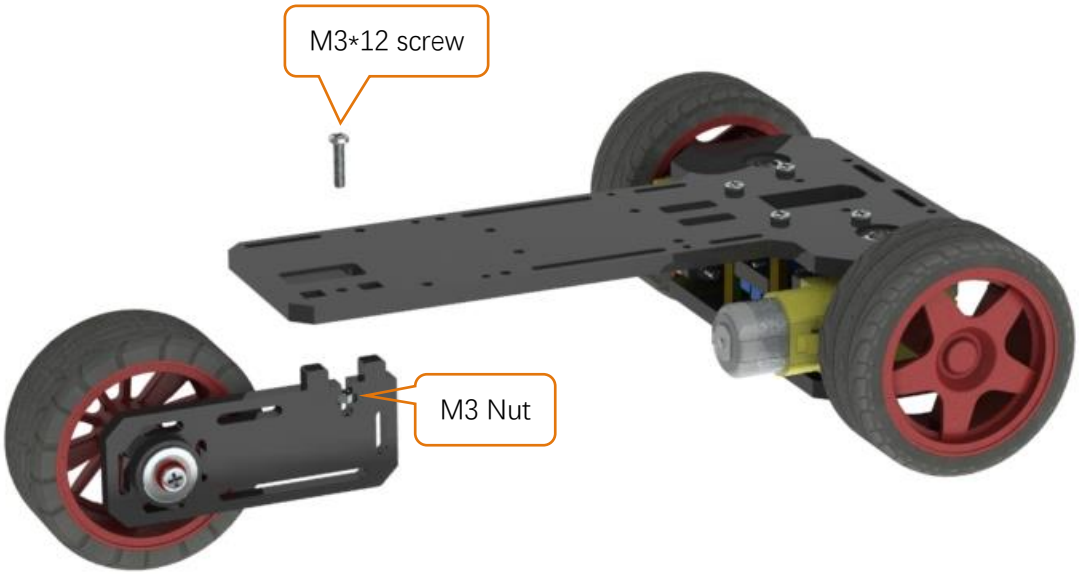


Assemble the following components

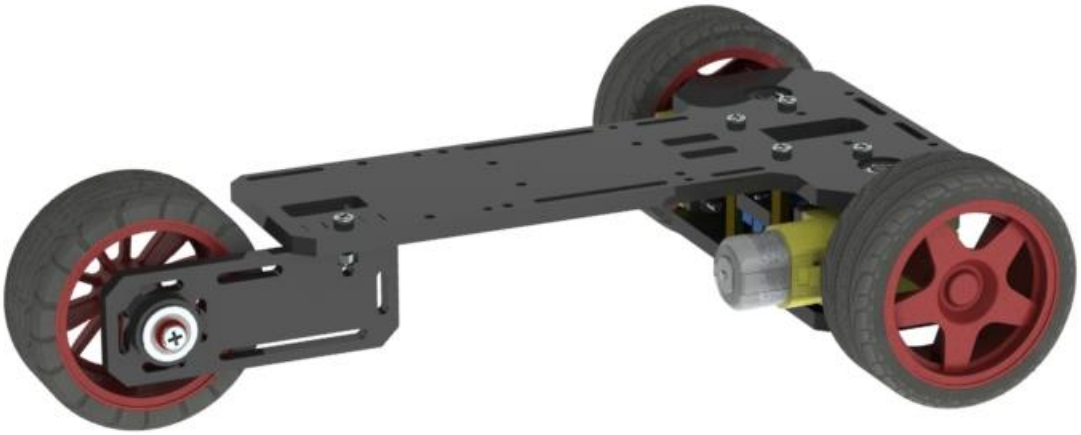


Effect diagram after assembling

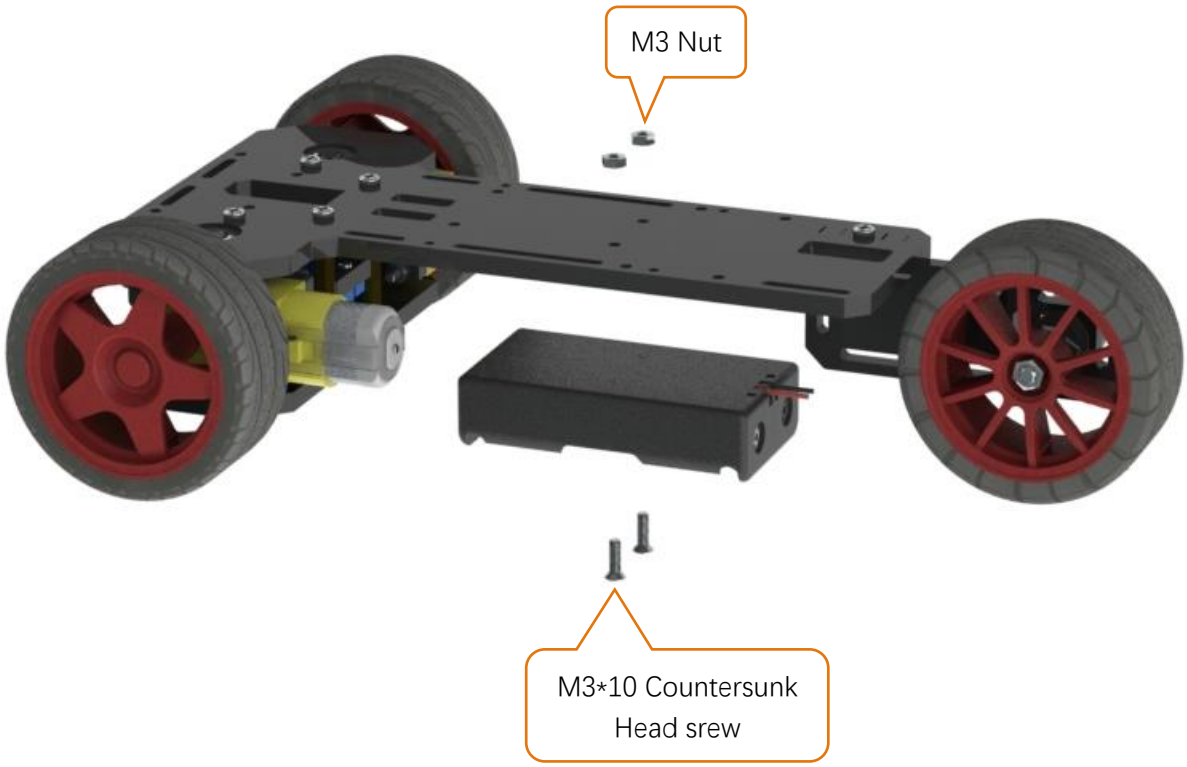
Assemble the following components



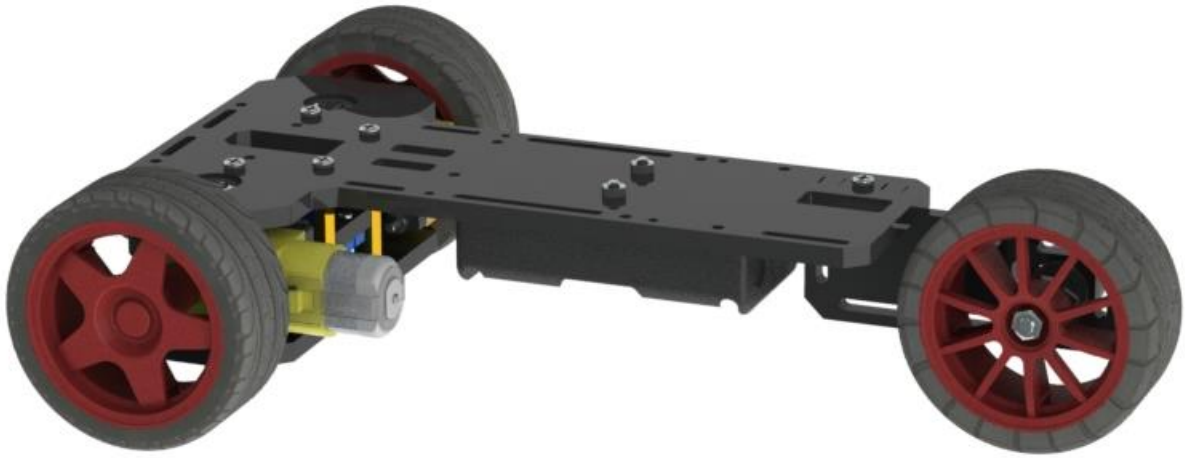
Effect diagram after assembling



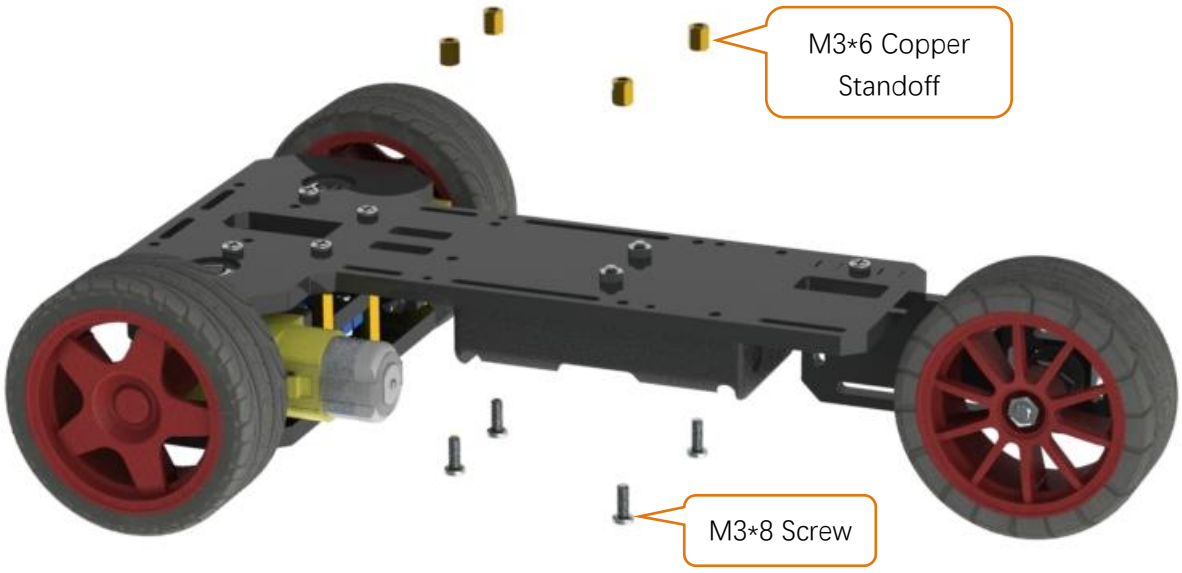
Assemble the following components



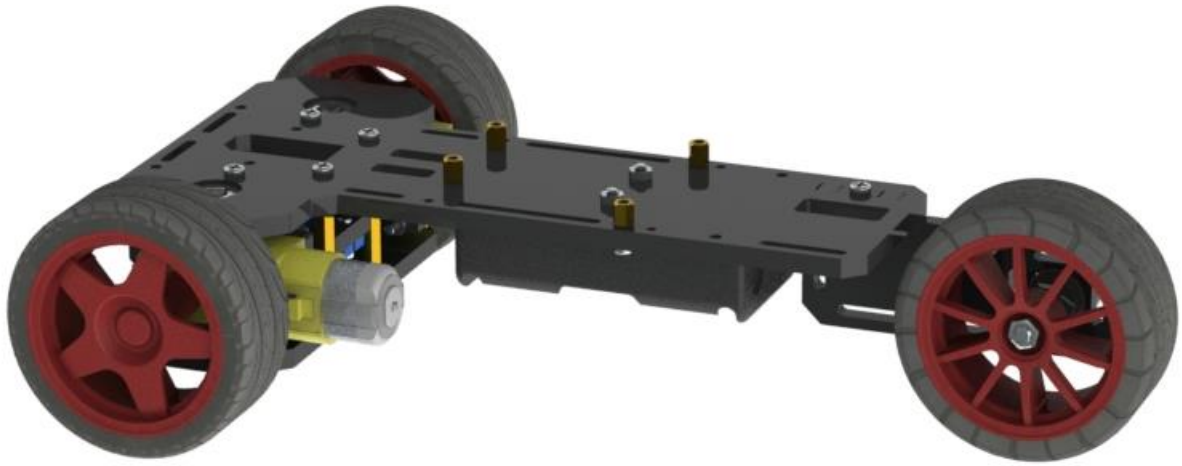
Effect diagram after assembling



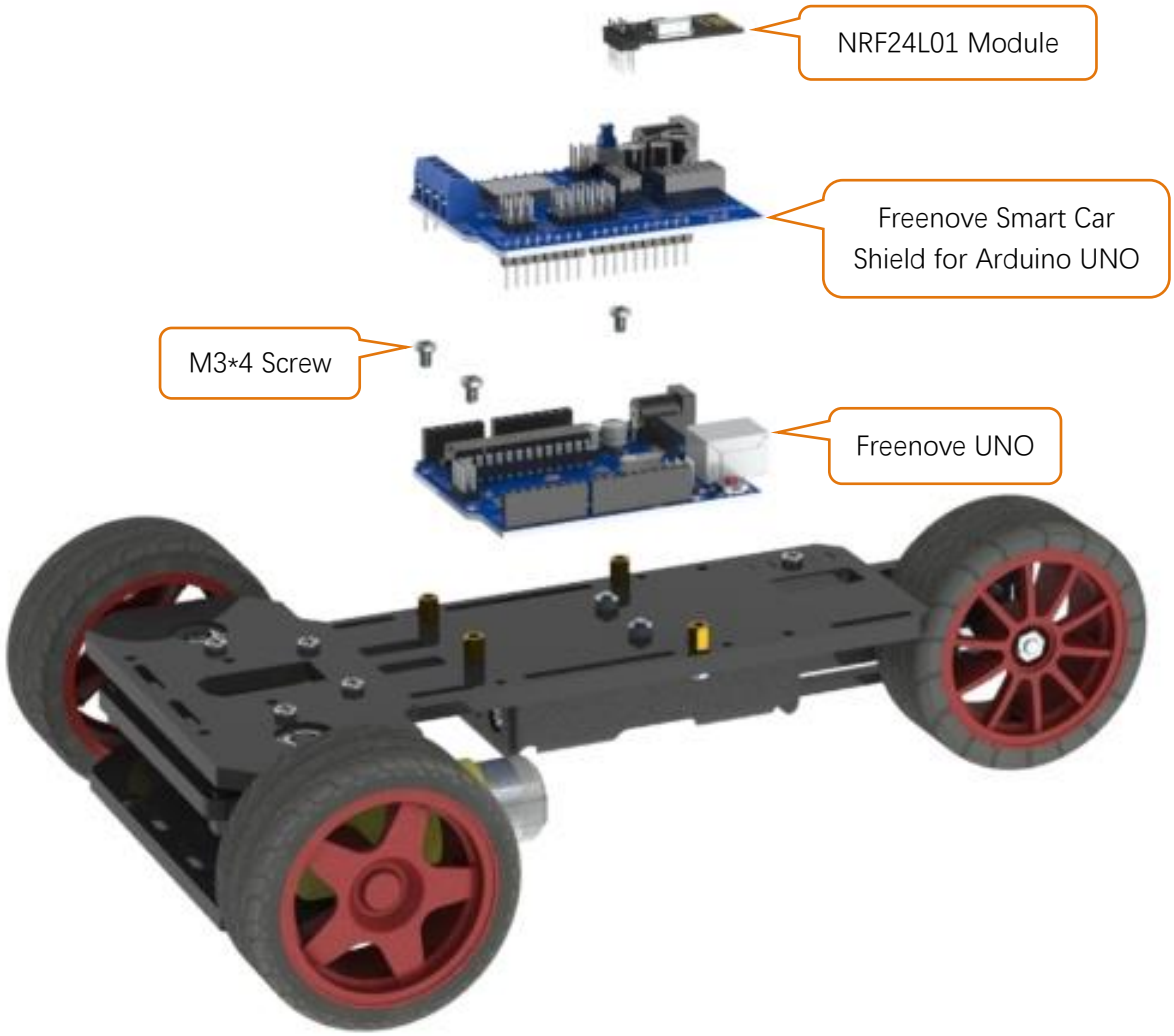
Assemble the following components



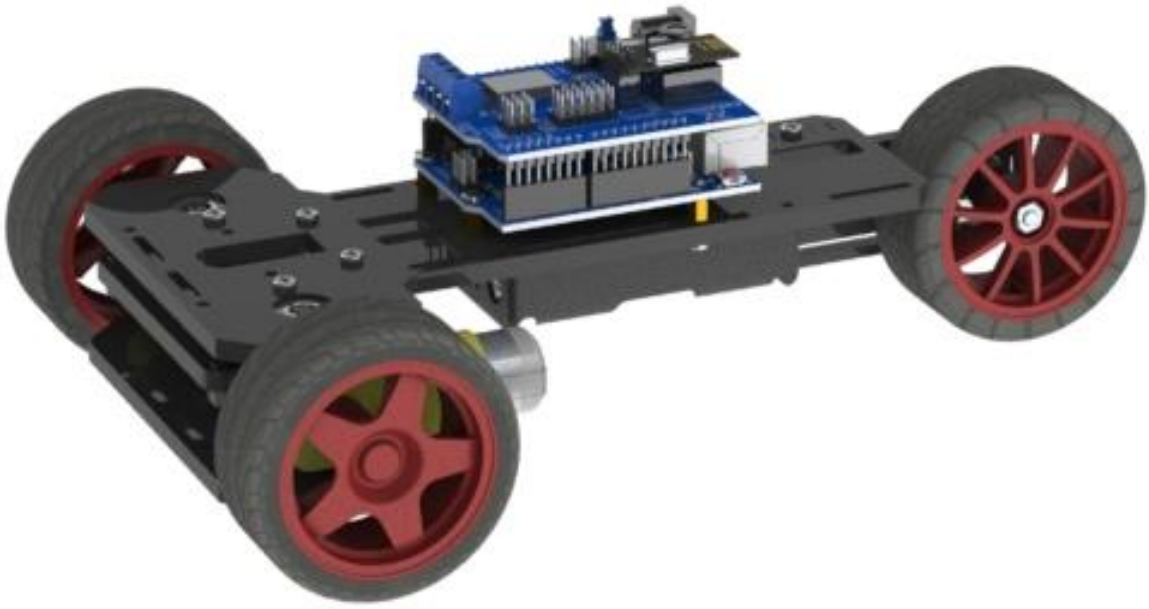
Effect diagram after assembling



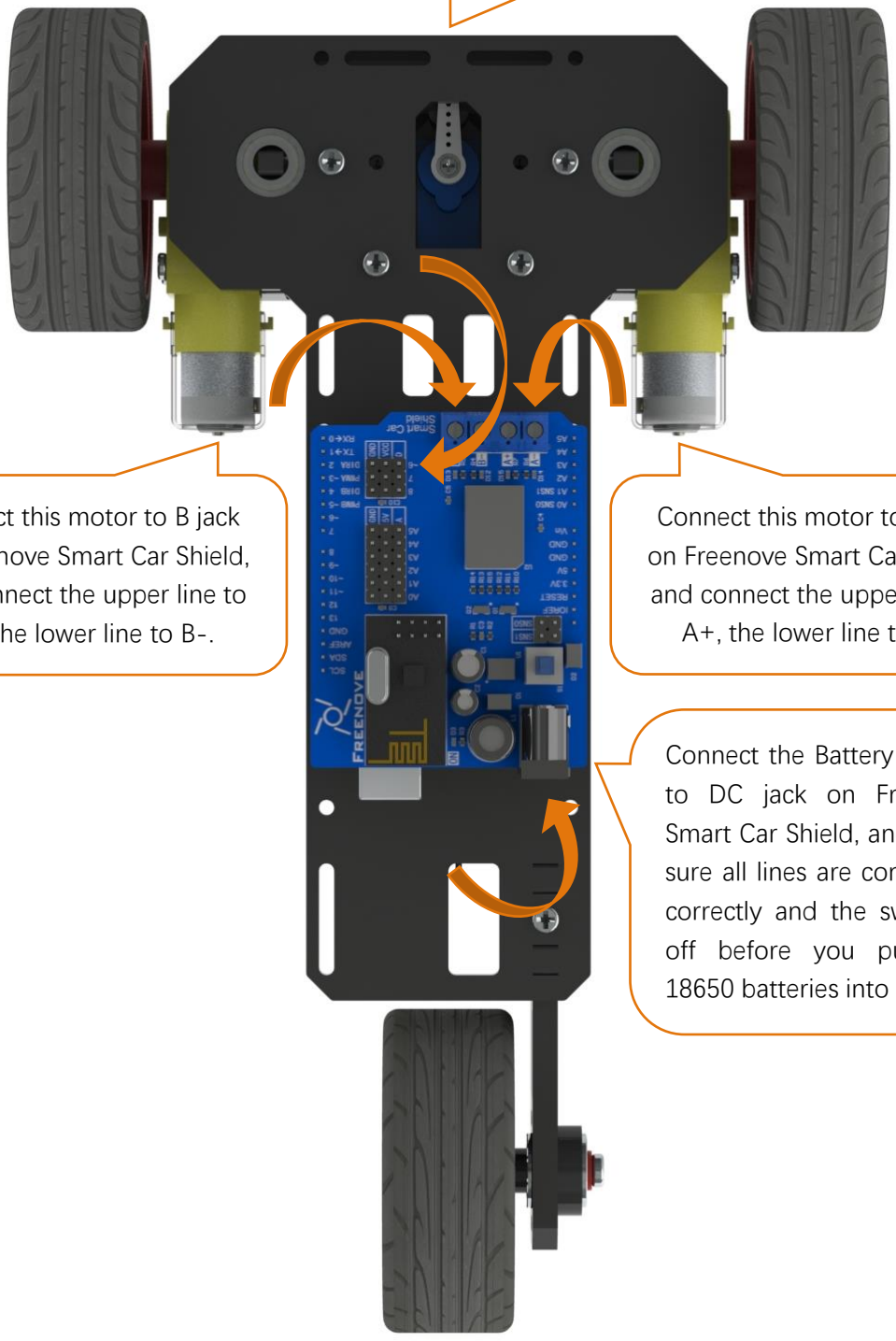
Assemble the following components



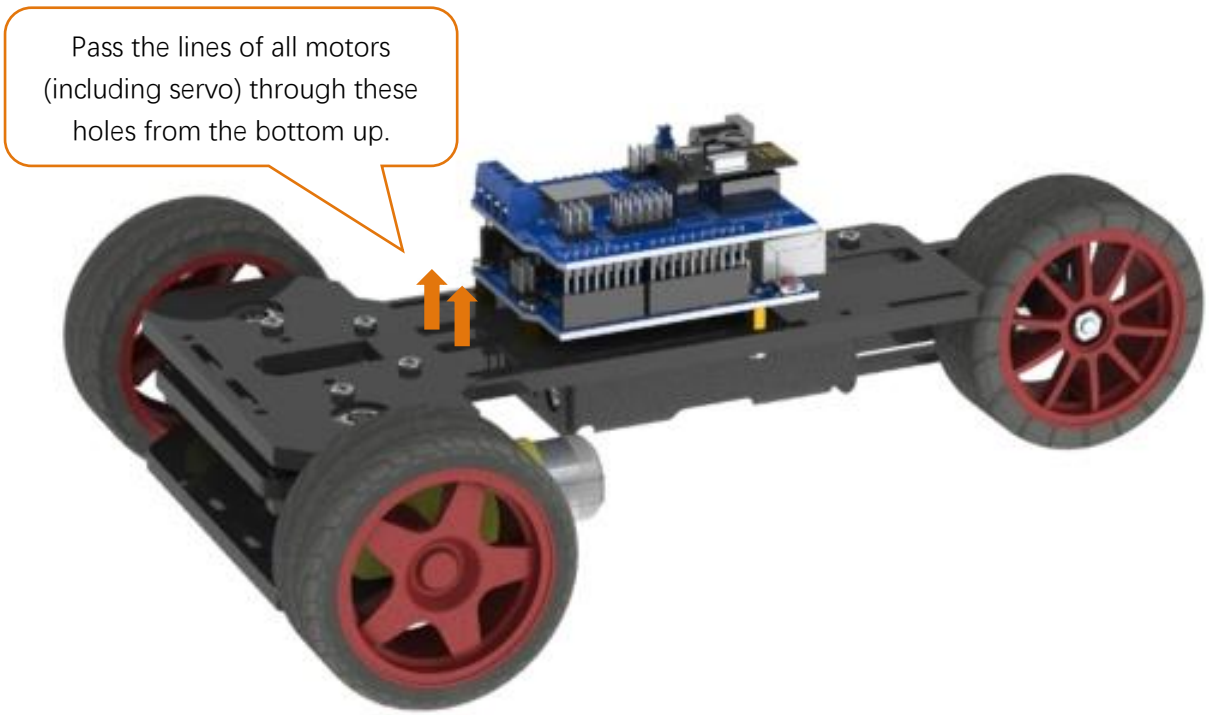
Effect diagram after assembling



Build the line connection



Please pay attention to the following line-connecting

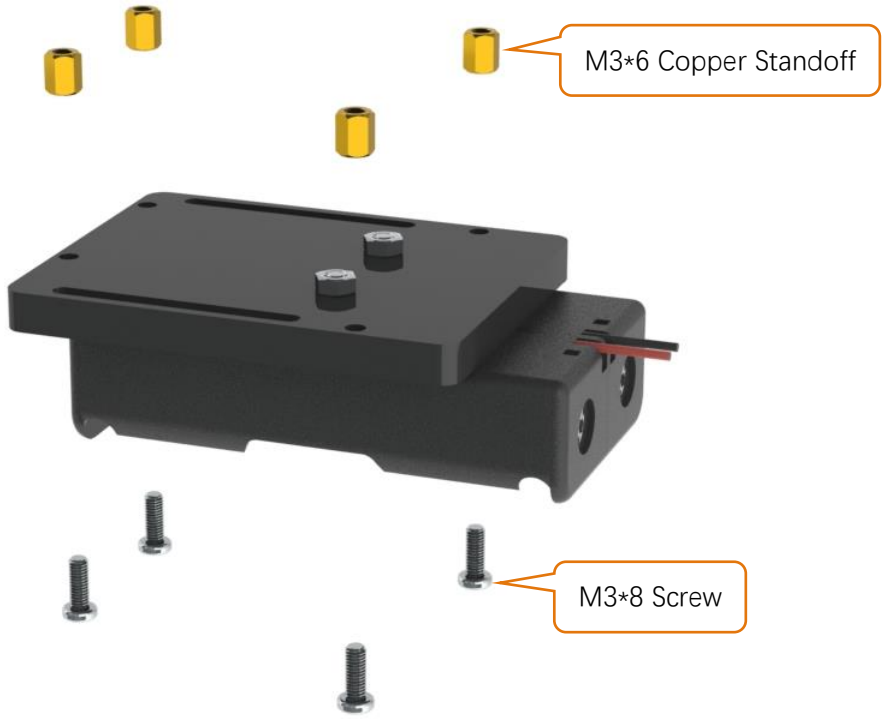


Step 2.3 Assemble the Romote Control

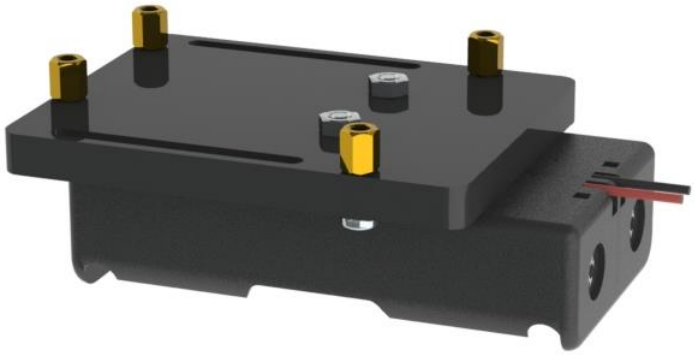
Assemble the following components

Effect diagram after assembling

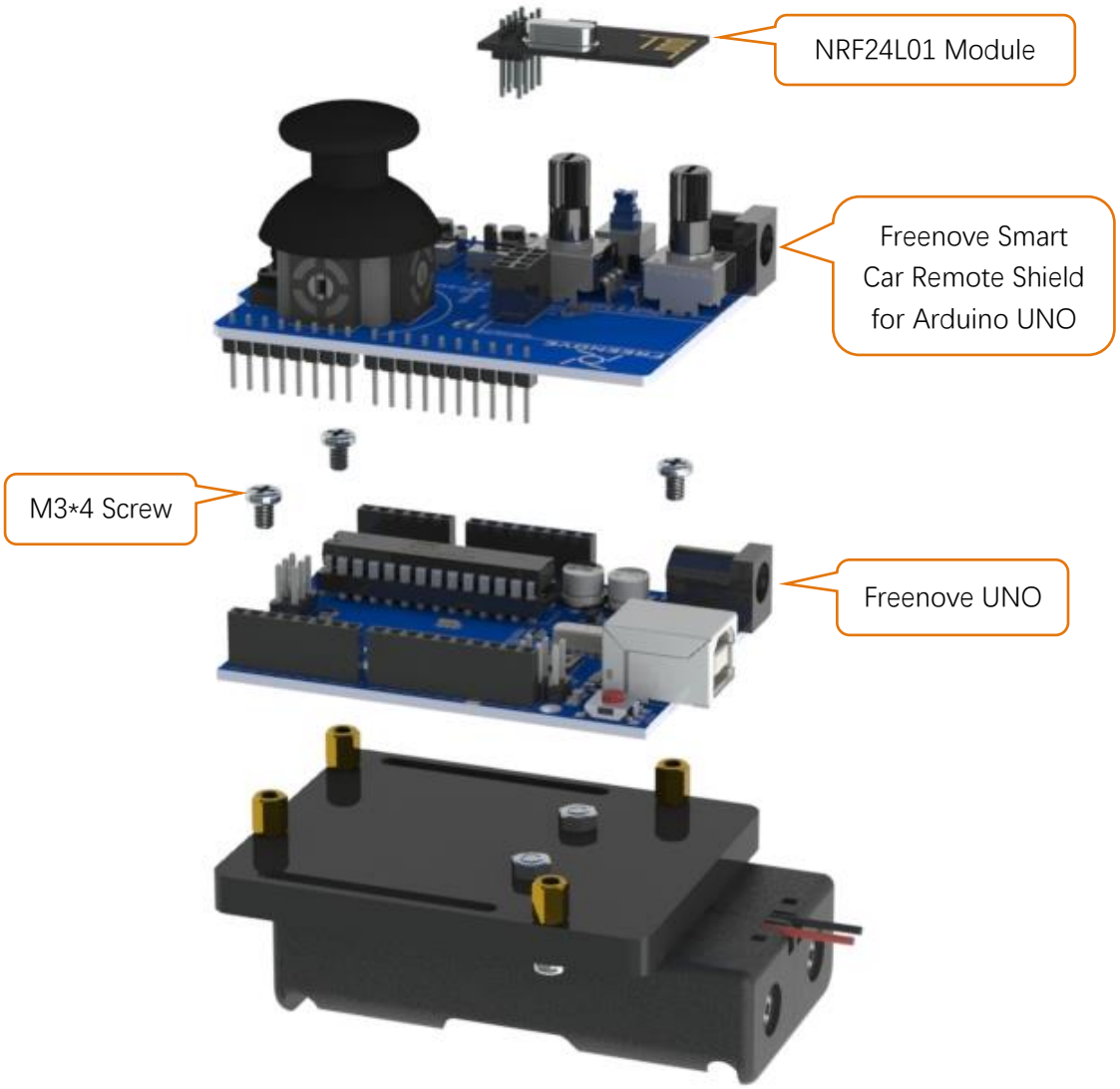
Assemble the following components



Effect diagram after assembling

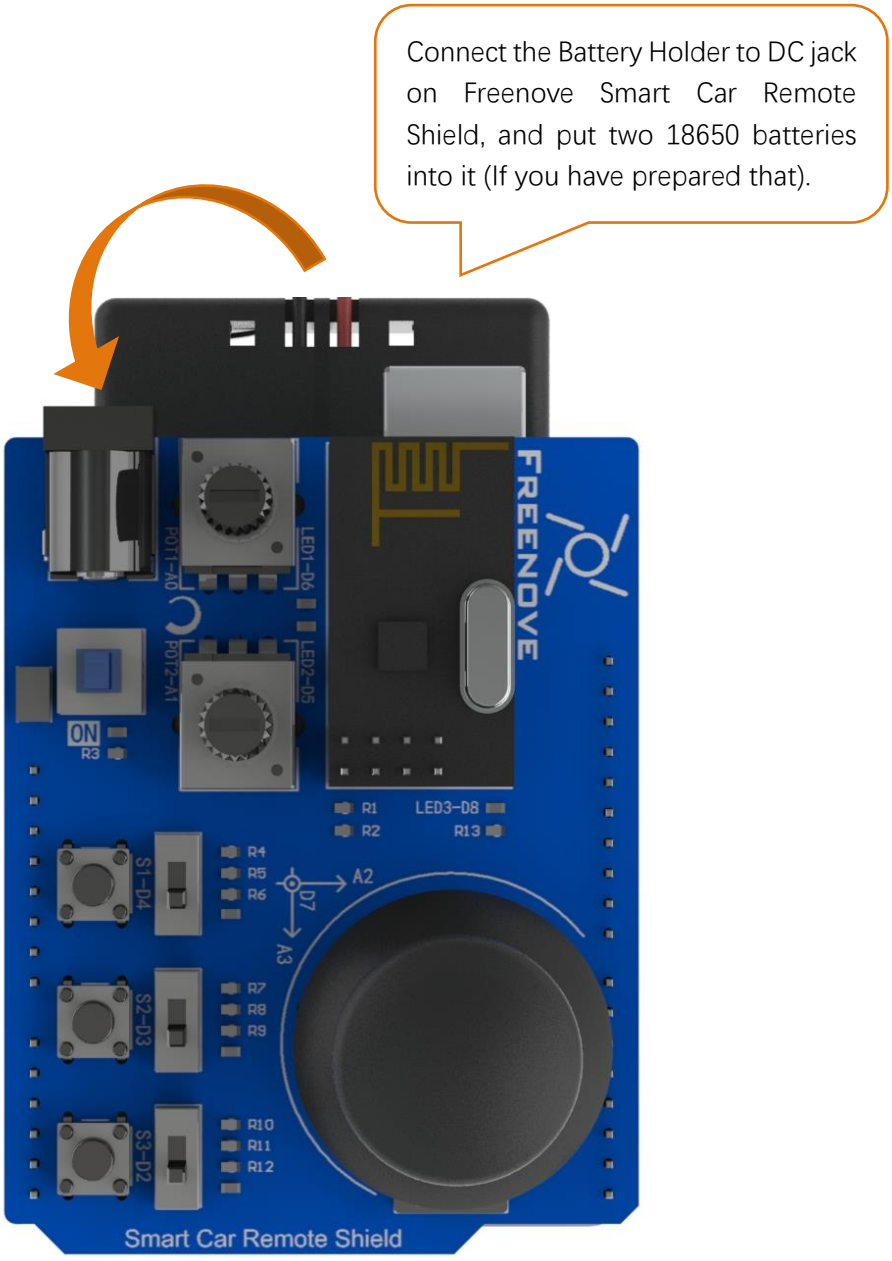


Assemble the following components



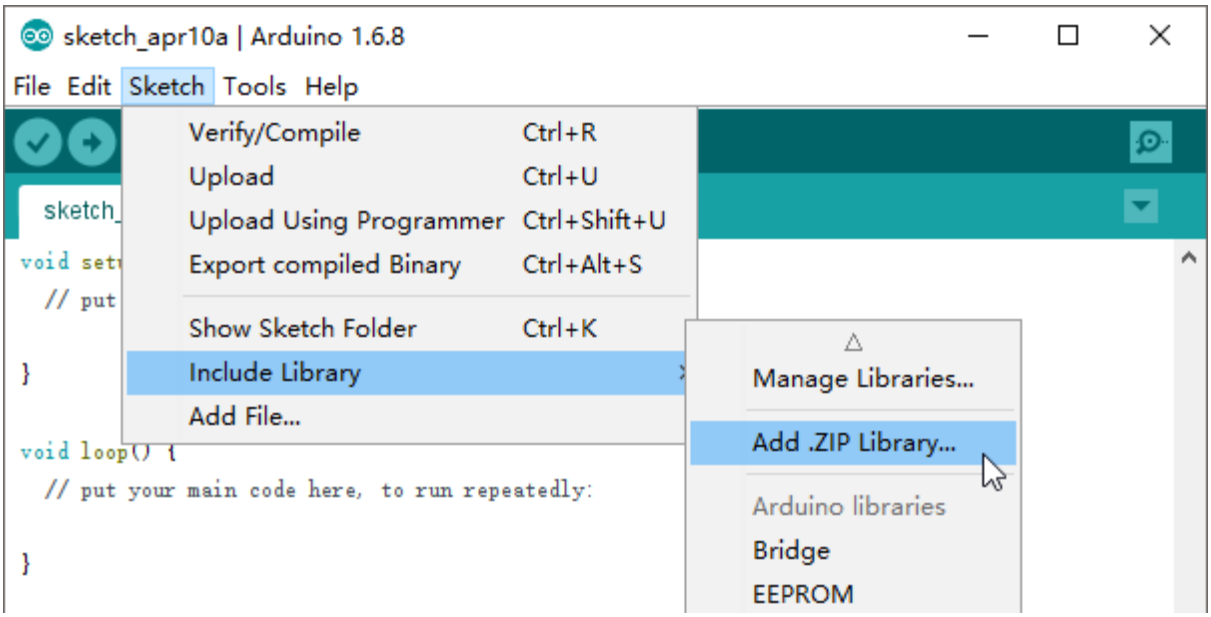
Effect diagram after assembling

Build the line connection

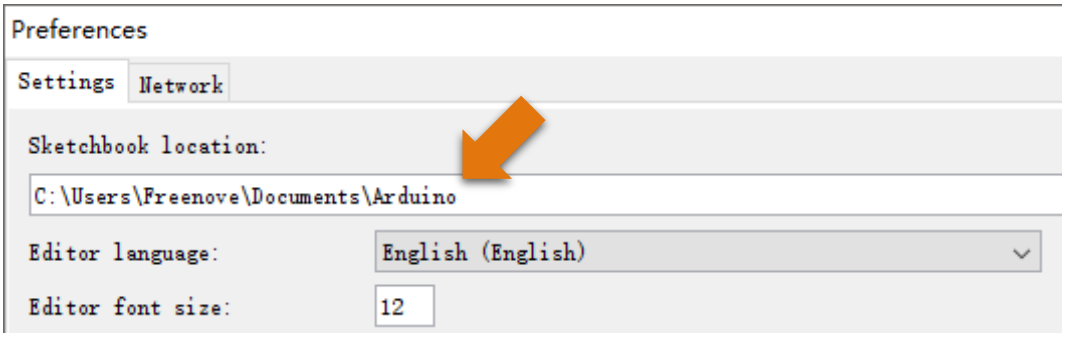


Step 2.4 Upload the Sketch

Before uploading the sketch, we need to import the library file we need. These library files are contributed by other open source developers, which is not previously installed in Arduino Software. Click Add.ZIP Library, and then add the RF24.zip under Libraries folder.



When these libraries are added, you can locate them in the libraries under Sketchbook location in the File-Preference window. You can view the source code of these library files to understand their specific usage.



Then, connect the UNO of smart car to the computer and upload the sketch:

Sketches\Remote_control\Remote_control_Car\Remote_control_Car.ino

After it is finished, pull out the USB cable from the UNO. Then, put smart car on the ground, and turn on the power switch.

Connect the UNO of remote control to the computer and upload the sketch:

Sketches\Remote_control\Remote_control_Remote\Remote_control_Remote.ino

After it is finished, if the remote control is fitted with batteries, pull out the USB cable from the UNO, and turn on the power switch. If not, keep the connection of UNO and computer.

Now, you can enjoy the fun of controlling your smart car with remote control. The function of this mode includes:

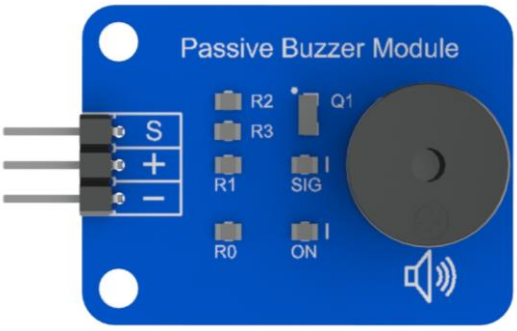
- Control the movement of the smart car. When you push the joystick with different direction and strength, your smart car will move with different steering degree and speed.
- Adjust the front wheels. The potentiometer POT1 on the remote control can be used to adjust the corresponding relationship between the joystick and steering degree. When you release the joystick, the wheel of smart car may not be in the middle position, so it can be adjusted through the potentiometer.
- Adjust the motor. The potentiometer POT2 on the remote control is used to adjust the corresponding relationship between the joystick and the motor of smart car. When you release the joystick, the smart car may be still marching, or the LED beside motor interface is still on, so it can be adjusted through this potentiometer.

If your smart car fails to march as expected, please try to eliminate the malfunction in the following ways:

- The power indicator LED does not light up. Make sure that the battery has been charged, and installed correctly.
- Can't control the smart car. Please check whether the sketches are uploaded correctedly, and NRF24L01 module is properly installed. Meanwhile, make sure that the smart car power is open, and remote control power is also open or use USB cable for power supply. When the wireless communication is normal, the LED3 on the remote control will light up.
- Smart car can not steer. Please check whether the servo line is correctly connected.
- Smart car marches in a deviated direction. Turn the potentiometer POT1 to make it right.
- The wheels are not in the middle position when you release the joystick. Adjust the potentiometer POT1.
- The front wheels can not rotate. Please check whether the motor line is correctly connected.
- The front wheel rotates in a wrong direction. Just change the connection of motor line.
- Smart car keeps marching when you release the joystick. Turn this potentiometer POT2 to adjust.
- LED beside the motor interface is still light up when you release the joystick. Turn the potentiometer POT2 to adjust.
- Other questions. Please contact us for more technical help!

3 Expand Remote Control Mode

So far, we have been able to control the smart car with the remote control. Now we will install some interesting accessories for the smart car.
The following is a buzzer module and RGB LED module.

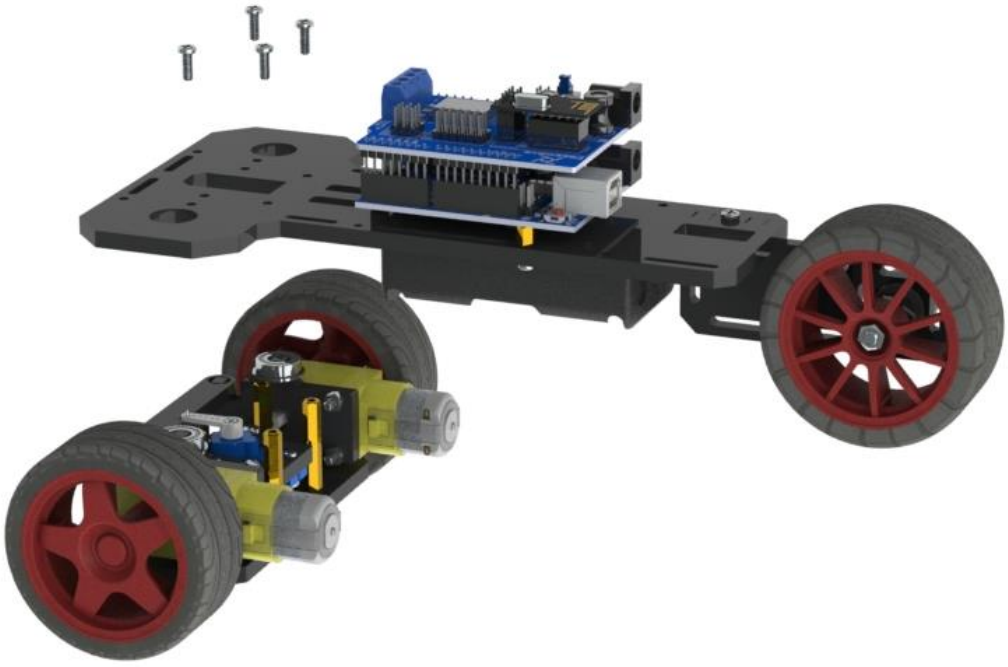


Buzzer module is used to make a sound, and RGB LED module can creat different colors of light. In this section, we will use the buzzer module to work as the smart car horn, and RGB LED module as the warning light.
The fixing hole of the Freenove electronic module is unified, so these modules can be installed on the smart car. You can visit our website or contact us to purchase other electronic modules for further creation.

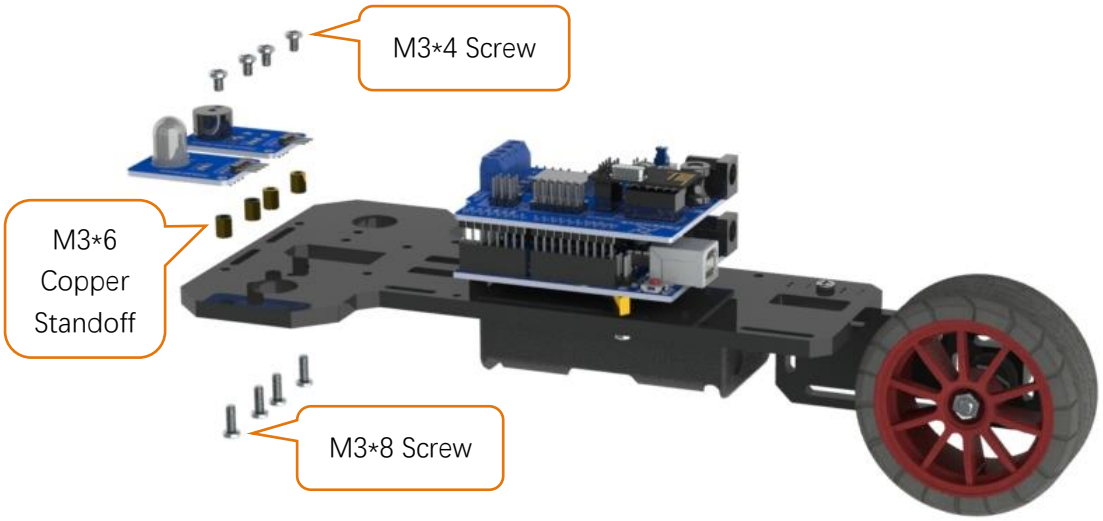
Step 3.1 Assemble the Modules

Please turn off the power switch before assembling the modules!

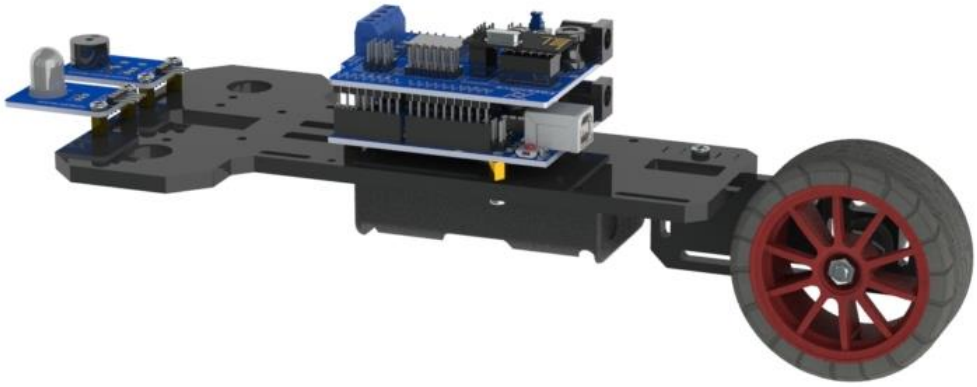
In order to install the buzzer and RGB LED module in front of the smart car, you need to remove some of the components first.



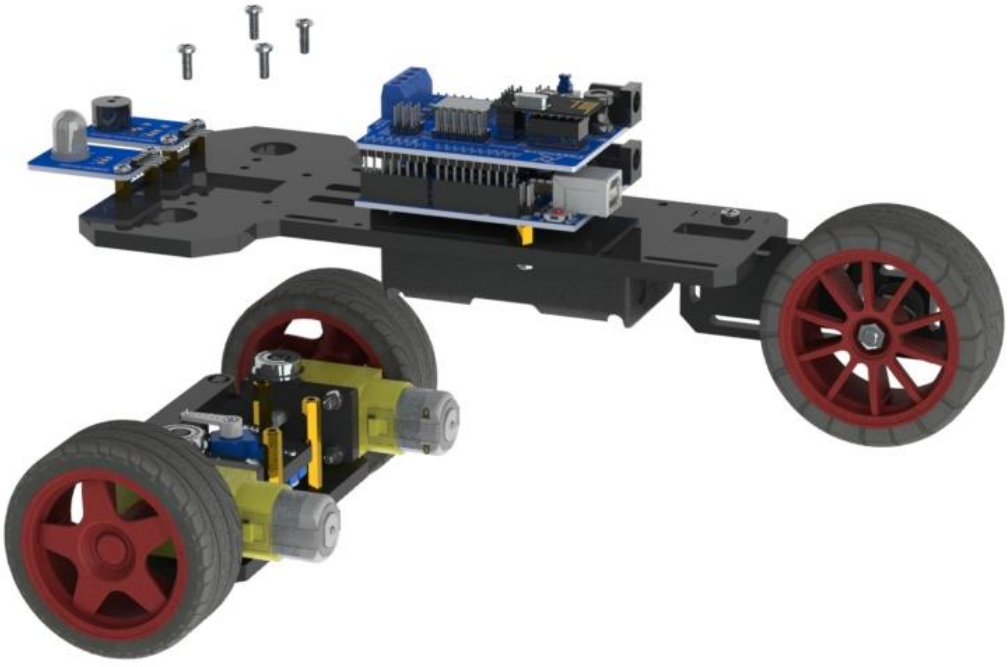
Then fix the following electronic modules on the smart car



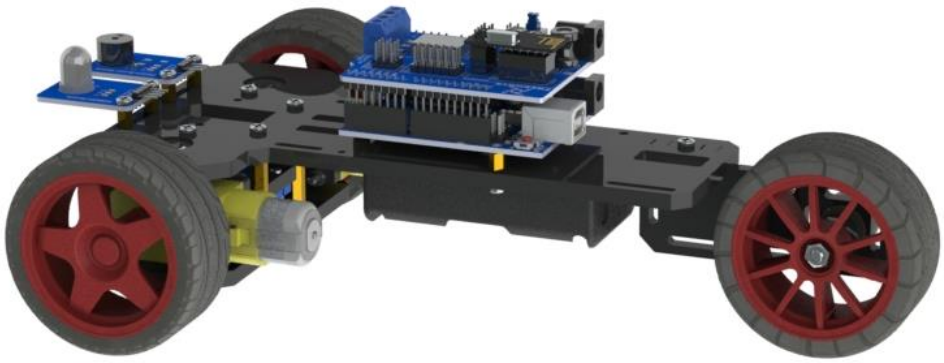
Effect diagram after the modules are fixed.



Then reassemble the removed components together.



Effect diagram after assembling



Step 3.2 Connect the Modules

Connect these two modules to the expansion shield with jumper. Here is the instruction:

Module	Module Interface	Expansion Shield Interface
Buzzer	S	D3
	+	VCC
	-	GND
RGB LED	R	A3
	G	A4
	B	A5
	+	5V

After the connection is confirmed correct, put smart car on the ground, and turn on the power switch.

As the sketches that previously uploaded has already contained the function of controlling these two modules, so no need to upload the sketches again.

Now, you can start controlling these modules with the remote control. The function includes:

- Control the buzzer. You can press the joystick vertically to make the buzzer beep.
- Control RGB LED light. You can use the tact switch and toggle switch S1, S2 and S3 to make the RGB LED generate colorful light.

If these modules fail to work as expected, please try to eliminate the malfunction in the following ways:

- The buzzer can't beep. Check that whether the connection is correct, and the LED ON on the module should light up under normal condition.
- RGB LED can't light up. Please check whether the connection is correct.
- The RGB LED can't light off. Make sure that the toggle switch S1, S2 and S3 are closed, and the LED beside them should light off.
- Other questions. Please contact us for more technical help!

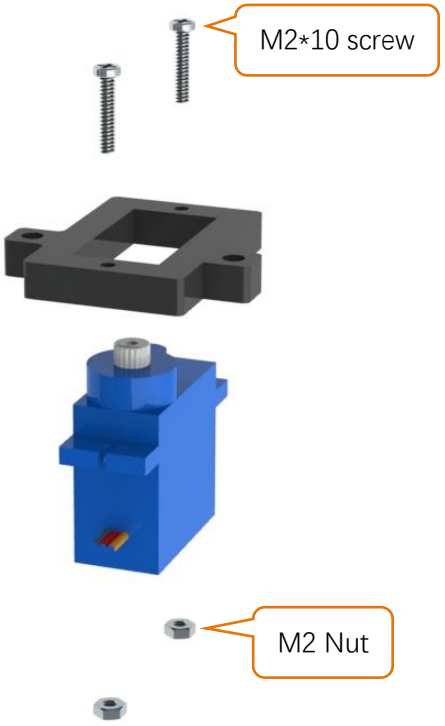
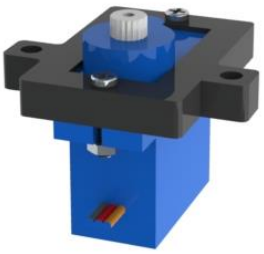
4 Automatic Driving Mode

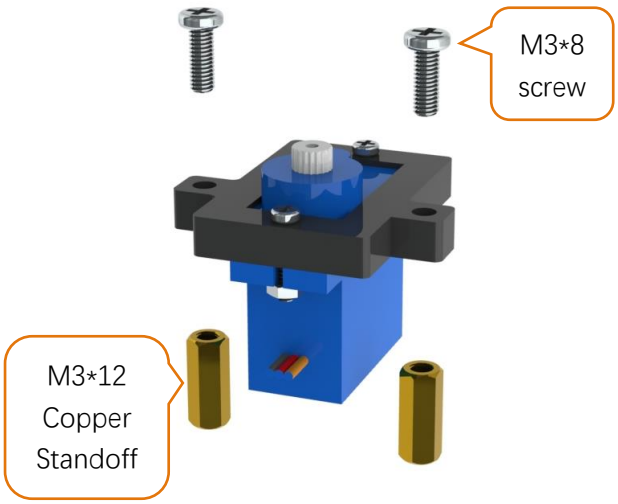
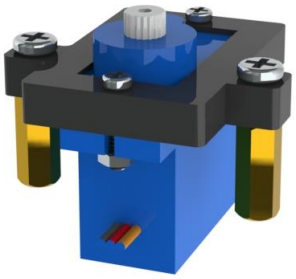
From previous project, we have been able to control the smart car with a remote control, and install some interesting accessories for the smart car. Now we're trying to make your smart car drive automatically with the help of ultrasonic sensor.
The following is the ultrasonic ranging module.

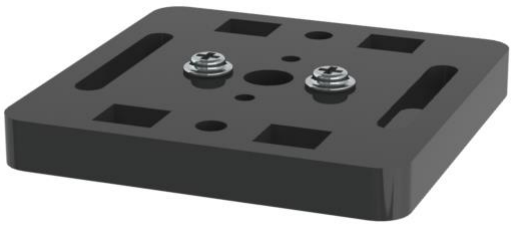
When ultrasonic ranging module works, it will send out ultrasonic wave. Once it encounters an obstacle, the ultrasonic wave will reflect back. Then this module can calculate the distance from the obstacle according to the reflected time.

Step 4.1 Assemble the Pan-tilt

First, assemble the pan-tilt.

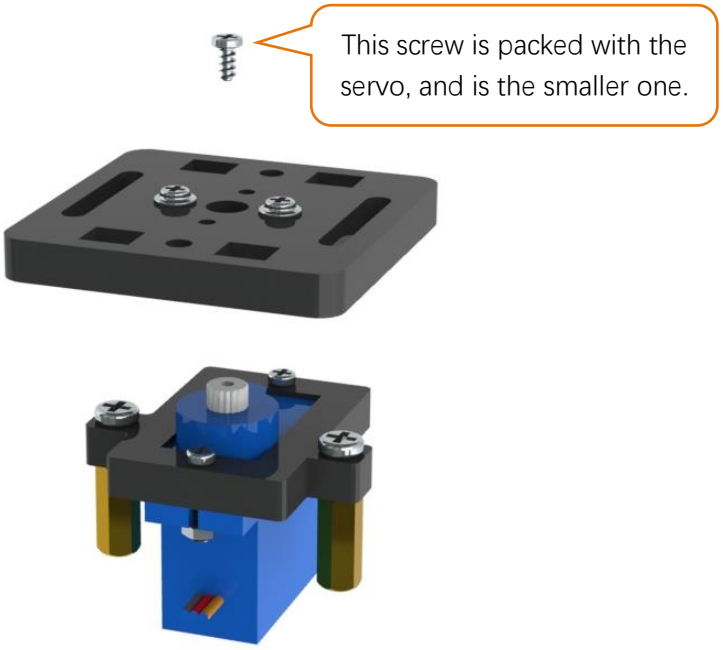
Assemble the following components	Effect diagram after assembling
	

Assemble the following components	Effect diagram after assembling
	

Assemble the following components	Effect diagram after assembling
	

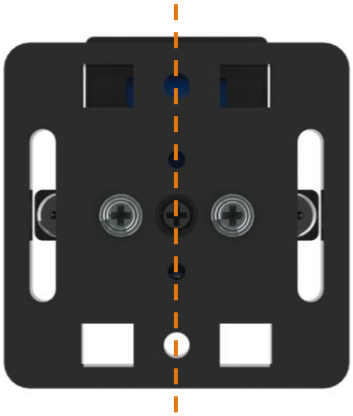
Assemble the following components

The rocker arm should be installed in the middle position between its rotation range. A little deviation is acceptable. If it is not installed in the middle position, you should remove the rocker arm and install it again instead of turning the shaft.

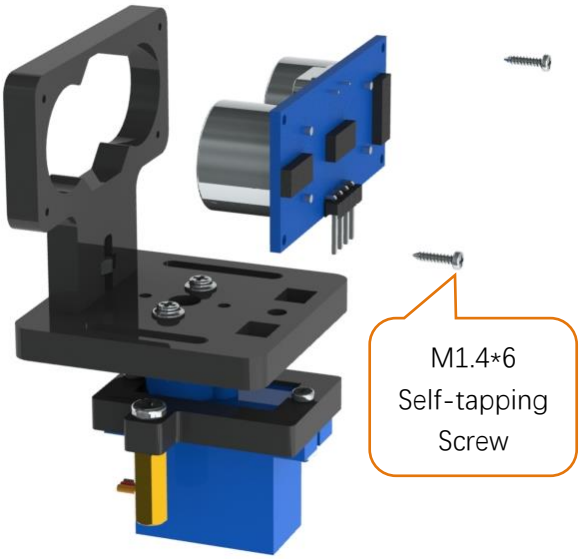


Effect diagram after assembling

When we look the front axle from vertical direction, the rocker arm should be installed in the middle position between its rotation range. If you turn the shaft by accident, you need adjust the servo and install it again.



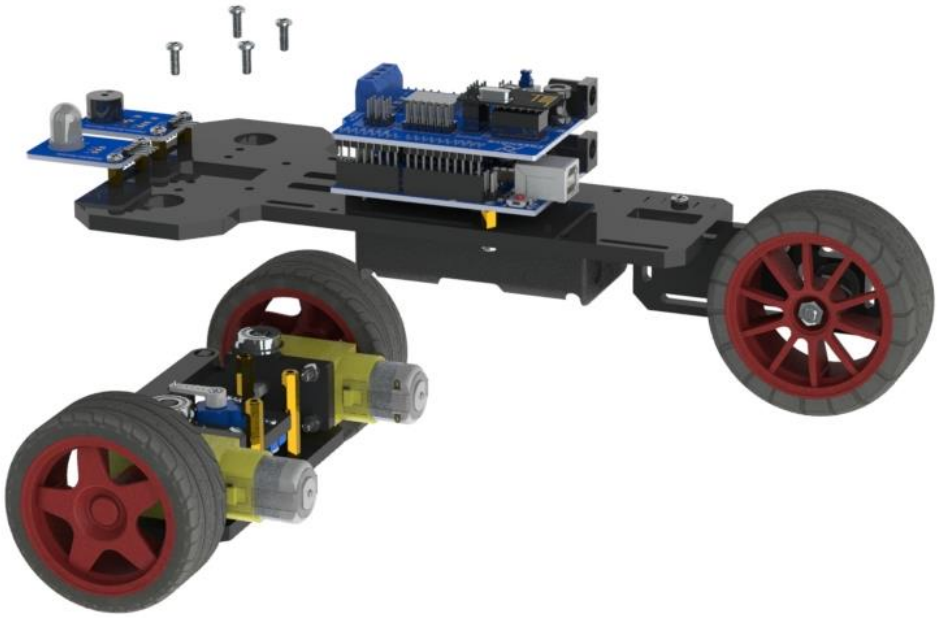
Assemble the following components	Effect diagram after assembling
	

Assemble the following components. Please put the jumper connected to the ultrasonic module first before assembling, otherwise it could be difficult to connect jumper after assembling.	Effect diagram after assembling If the ultrasonic module only has two holes, please assemble the screws to the other two holes on the acrylic plate.
	

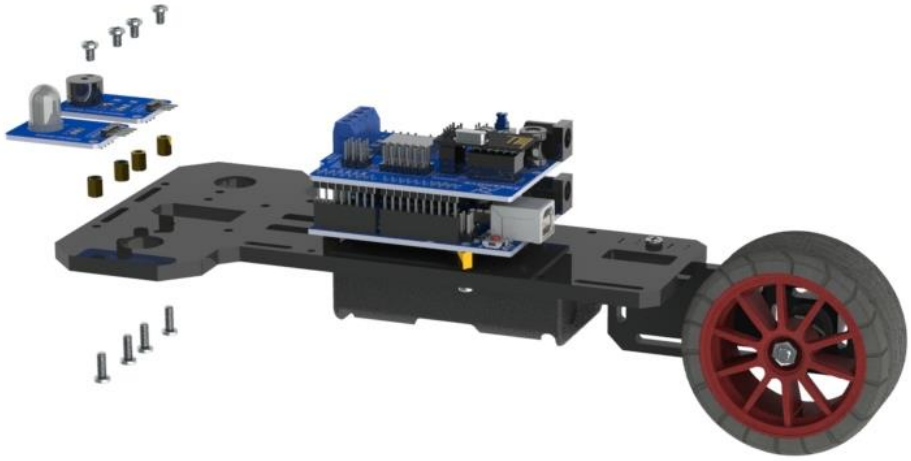
Step 4.2 Install the Pan-tilt

Please turn off the power switch before assembling these modules.

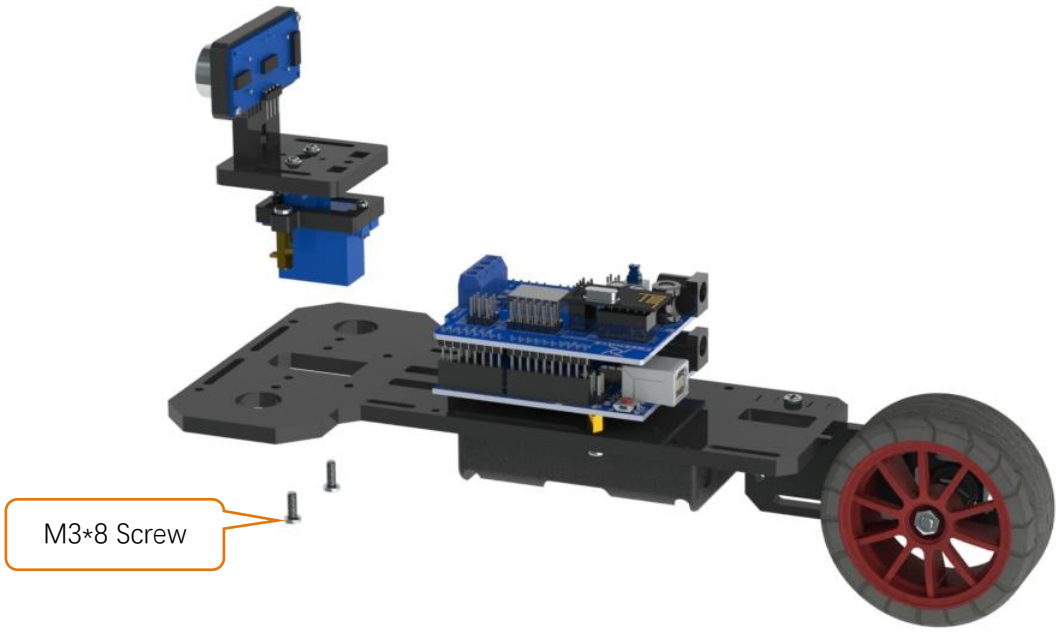
To remove the electronic modules installed in front of the smart car, we need to remove some components first.



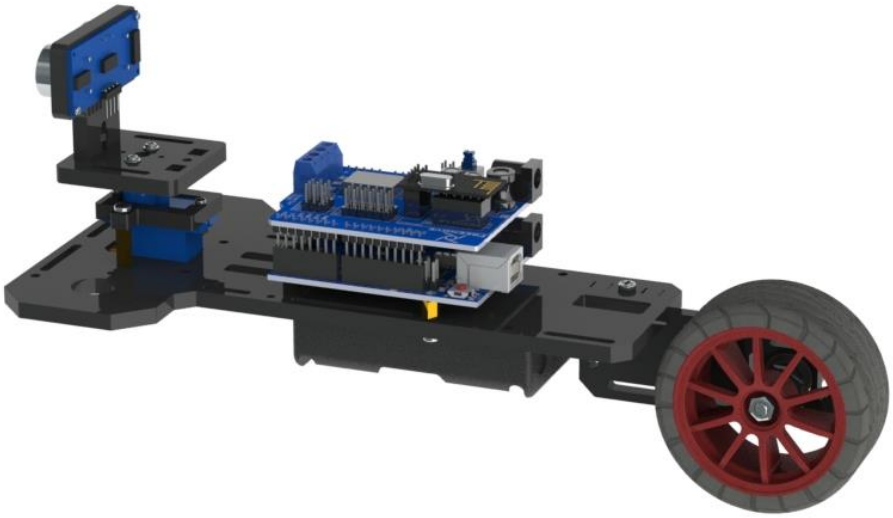
Remove the following electronic modules



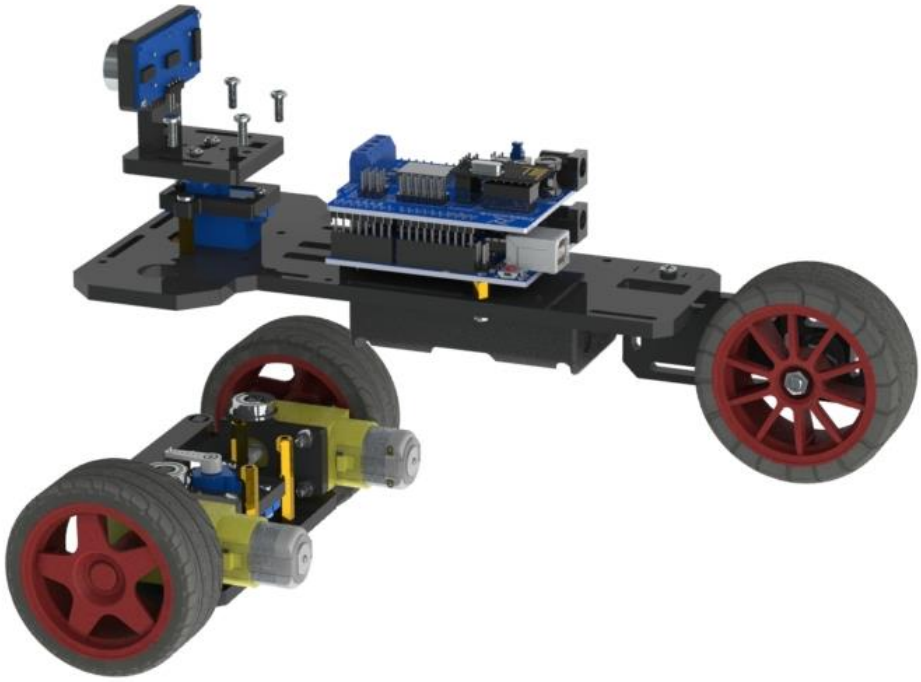
Assemble the pan-tilt



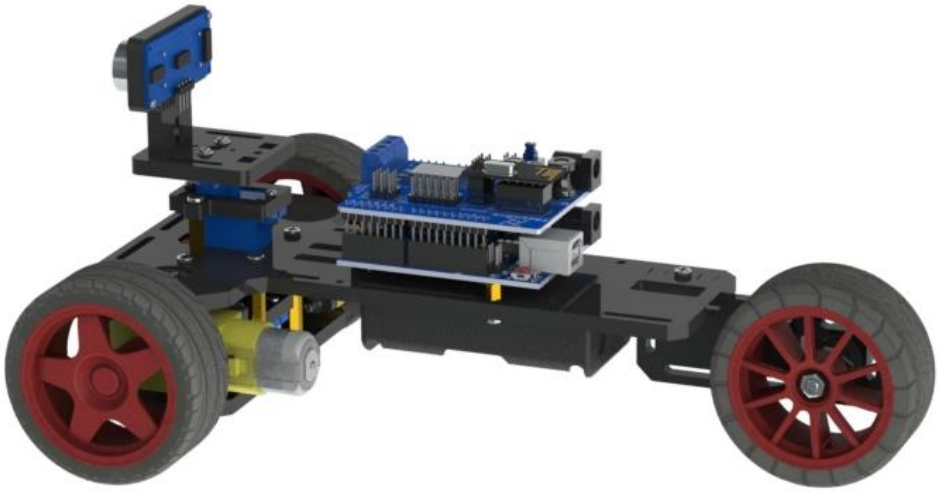
Effect diagram after assembling



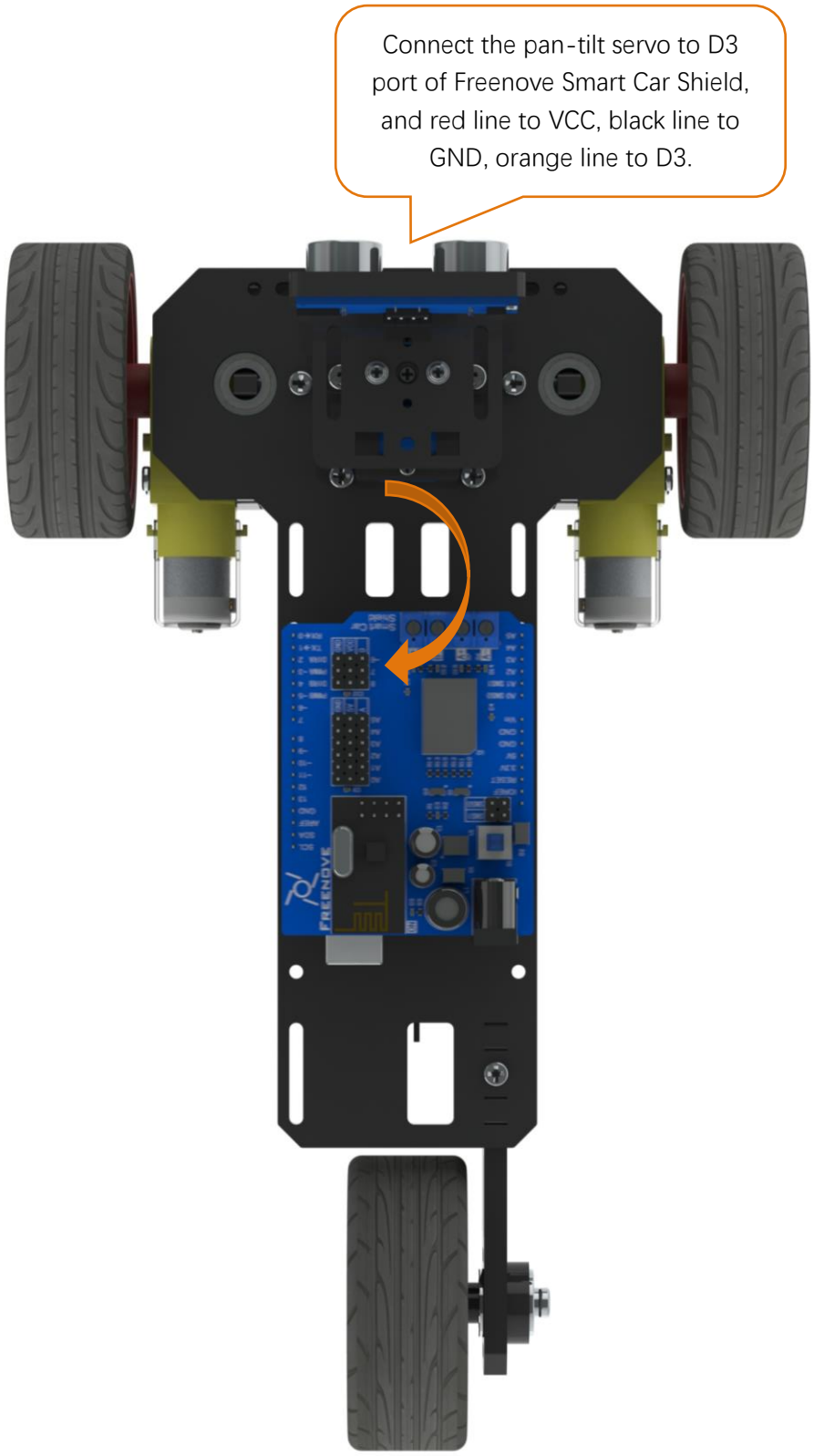
Assemble the removed components



Effect diagram after assembling



Build the line connection



Step 4.3 Connect the Modules

Connect the ultrasonic ranging module to expansion shield with the jumper. The connection table is as follows:

Module	Module Interface	Expansion shield Interface
Ultrasonic	Vcc	5V
	Trig	A4
	Echo	A5
	Gnd	GND

Check again to make sure there is no wrong connection!

Step 4.4 Upload the Sketch

Connect the UNO of smart car to the computer and upload the sketch:

Sketches\Automatic_driving\Automatic_driving_Car\Automatic_driving_Car.ino

After it is finished, pull out the USB cable from the UNO. Then, put smart car on the ground, and turn on the power switch.

Now, you can see that your smart car starts marching automatically. The function of this mode includes:

- Rotate the pan-tilt to detect obstacles.
- Change the marching direction when it encounters obstacles.

If your smart car fails to march as expected, please try to eliminate the malfunction in the following ways:

- Pan-tilt can't rotate. Check whether the connection of servo lines is correct.
- Smart car can't avoid obstacles. Check whether the connection of ultrasonic module lines is correct.
- Steering servo deviates its expectation. Modify the value of the variable "dirServoOffset" in the sketch, then upload it again.
- Pan-tilt servo deviates its expectation. Modify the value of the variable "plaServoOffset" in the sketch, then upload it again.
- Moving speed of smart car is not as expected. Modify the value of the variable "motorSpeed" in the sketch, then upload it again.
- Other questions. Please contact us for more technical help!

5 Exploration Mode

From previous project, we have archieved the goal of controlling your smart car through the remote control and automatic driving mode. Now we will try the exploration mode, a mode that interacts with Processing.

Processing

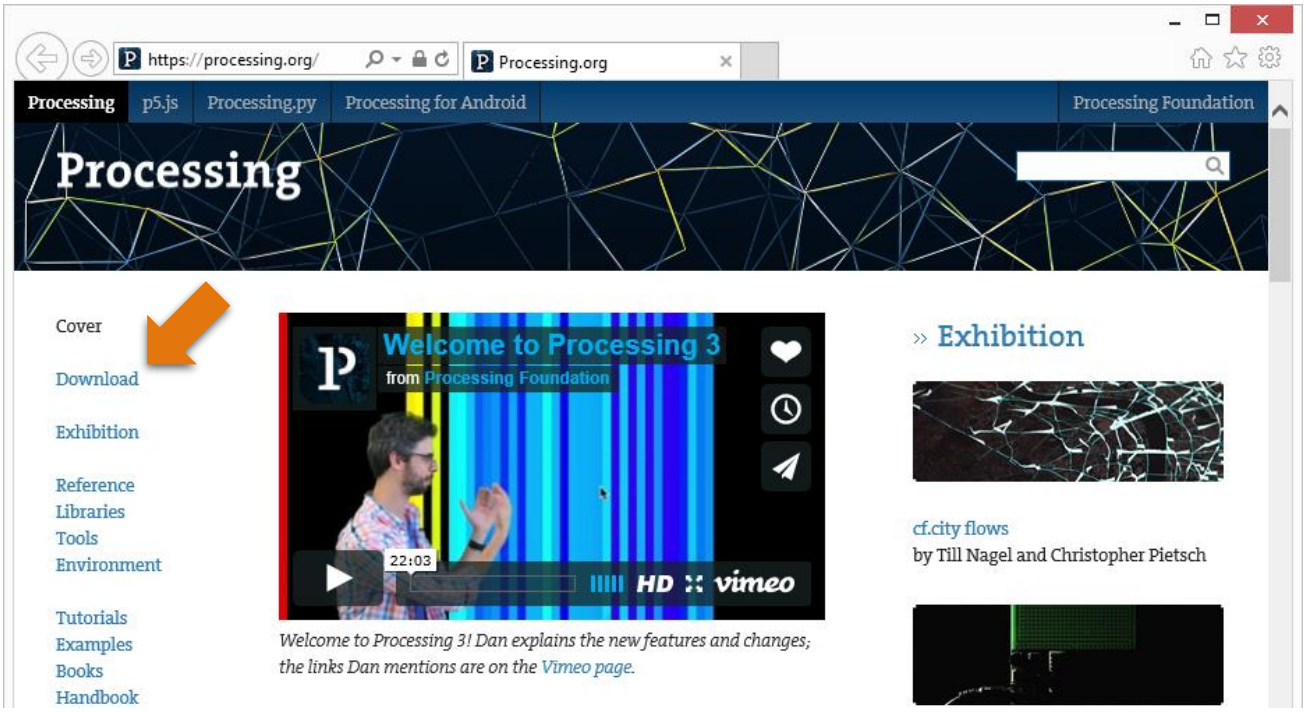
Processing software is used to write programs that run on computer. Processing software is free and open source, and runs on Mac, Windows, and GNU / Linux platforms, which is the same as Arduino software. In fact, Arduino software is based on Processing software. At present, they still have similar interface.

Programs written with Processing are also called sketches, and use Java language in default. Java language and C++ language has many similarities, so if you are familiar with C++ language, it will help you understand and write simple Processing sketches quickly.

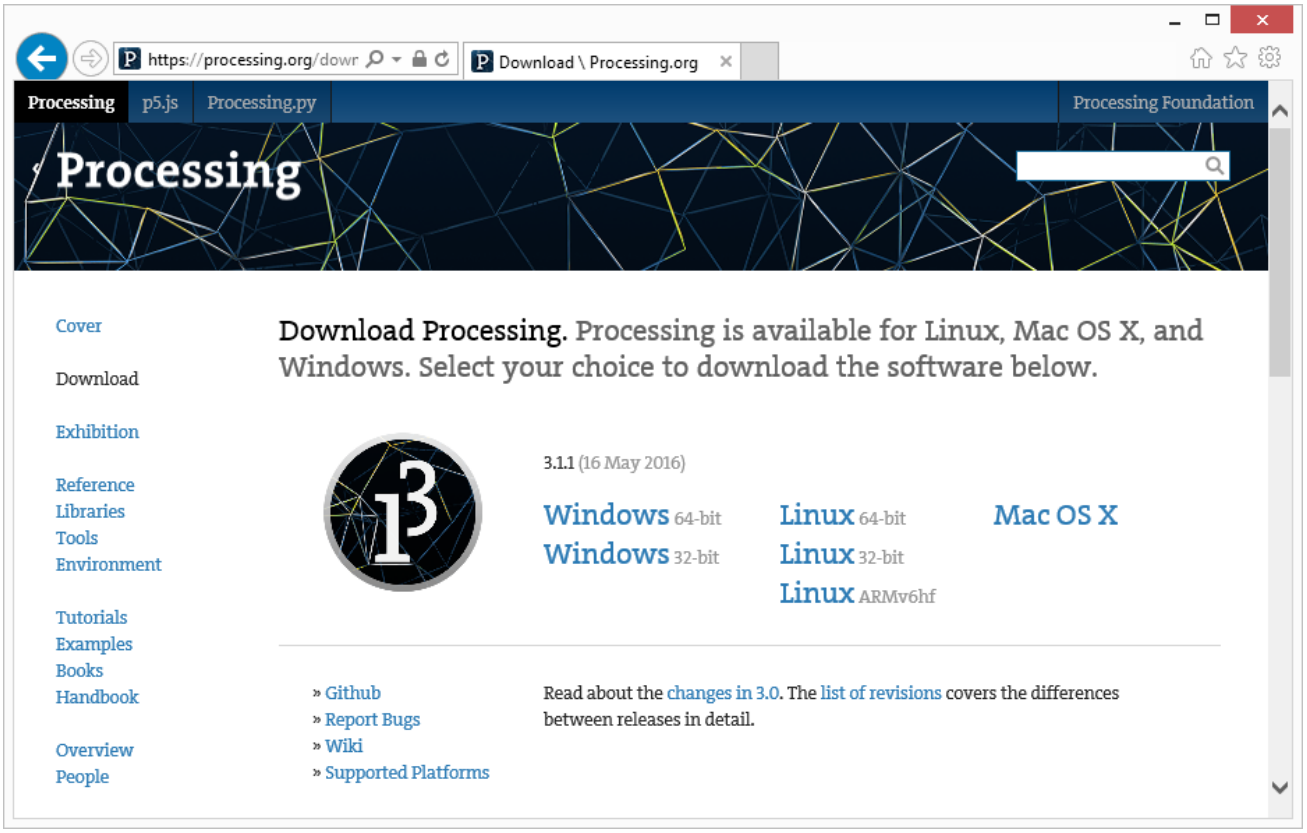
Now, we will introduce you how to install and use Processing software, and guide you to complete exploration mode of our smart car.

Processing Software

Processing software / Processing Development Environment (PDE) makes it easy to write Processing programs. First, install Processing software: visit https://processing.org/, click "Download" to enter the download page.



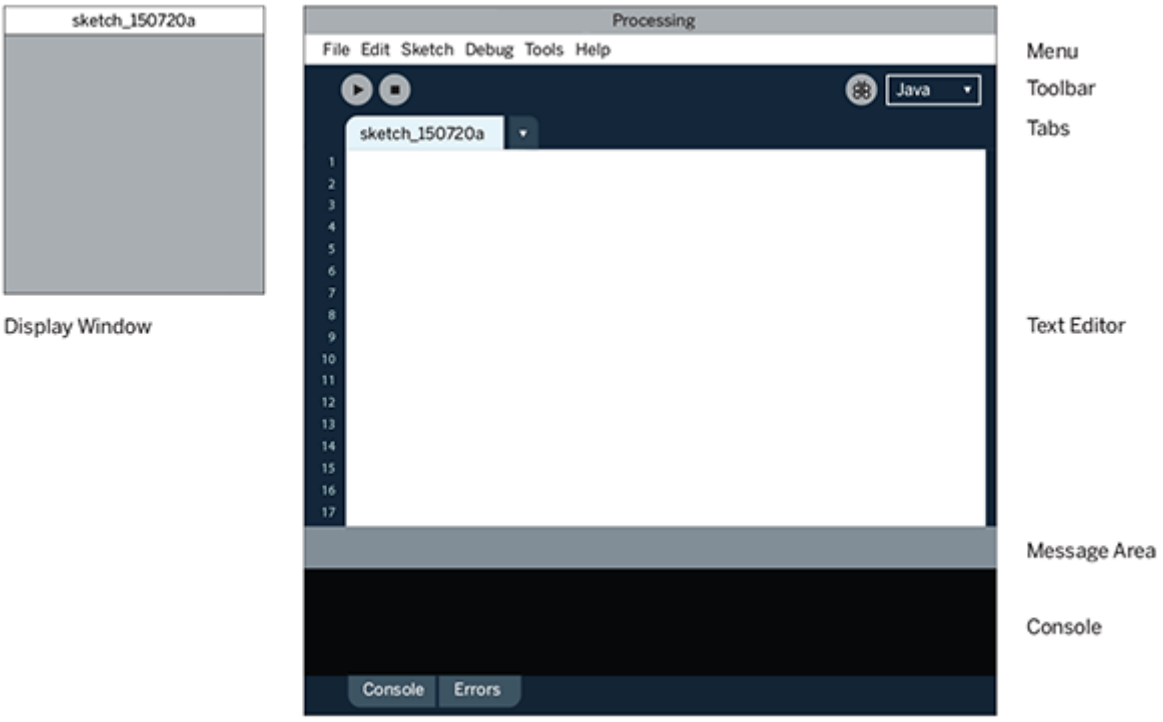
Select the Mac, Windows, or Linux version, depending on what machine you have.



Installation on each machine is straightforward:

- On Windows, you'll have a .zip file. Double-click it, and drag the folder inside to a location on your hard disk. It could be Program Files or simply the desktop, but the important thing is for the processing folder to be pulled out of that .zip file. Then double-click processing.exe to start.
- The Mac OS X version is also a .zip file. Double-click it and drag the Processing icon to the Applications folder. If you're using someone else's machine and can't modify the Applications folder, just drag the application to the desktop. Then double-click the Processing icon to start.
- The Linux version is a .tar.gz file, which should be familiar to most Linux users. Download the file to your home directory, then open a terminal window, and type:
  tar xvfz processing-xxxx.tgz
  (Replace xxxx with the rest of the file's name, which is the version number.) This will create a folder named processing-2.0 or something similar. Then change to that directory:
  cd processing-xxxx
  and run it:
  ./processing

With any luck, the main Processing window will now be visible. Everyone's setup is different, so if the program didn't start, or you're otherwise stuck, visit the troubleshooting page for possible solutions.



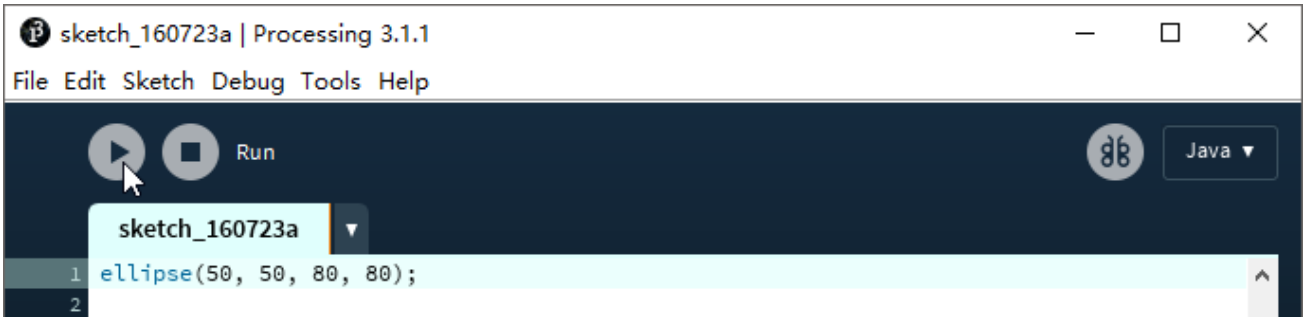
You're now running the Processing Development Environment (or PDE). There's not much to it; the large area is the Text Editor, and there's a row of buttons across the top; this is the toolbar. Below the editor is the Message Area, and below that is the Console. The Message Area is used for one line messages, and the Console is used for more technical details.

First Use

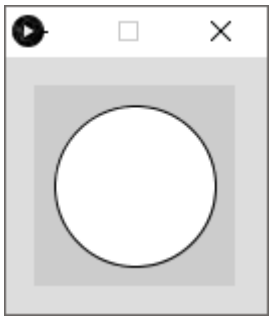
In the editor, type the following:

```
1  ellipse(50, 50, 80, 80);
```

This line of code means "draw an ellipse, with the center 50 pixels over from the left and 50 pixels down from the top, with a width and height of 80 pixels." Click the Run button (the triangle button in the Toolbar).

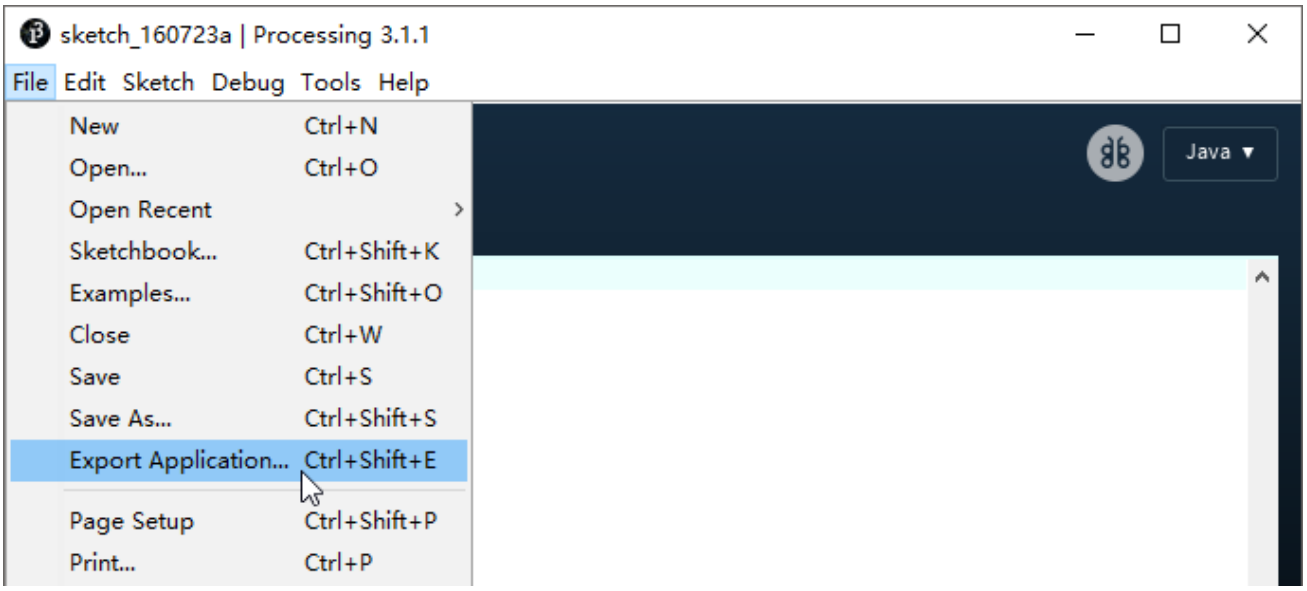


If you've typed everything correctly, you'll see a circle on your screen.



If you didn't type it correctly, the Message Area will turn red and complain about an error. If this happens, make sure that you've copied the example code exactly: the numbers should be contained within parentheses and have commas between each of them, and the line should end with a semicolon.

You can export this sketch to an application to run it directly without opening the Processing.
To export the sketch to the application, you must save it first.



So far, we have completed the first use. I believe you have felt the joy of it.

Interact with Processing

Connect the UNO of smart car to the computer and upload the sketch:

Sketches\Exploration\Exploration_Car\Exploration_Car.ino

After it is finished, pull out the USB cable from the UNO. Then, put smart car on the ground, and turn on the power switch.

Connect the UNO of remote control to the computer and upload the sketch:

Sketches\Exploration\Exploration_Remote\Exploration_Remote.ino

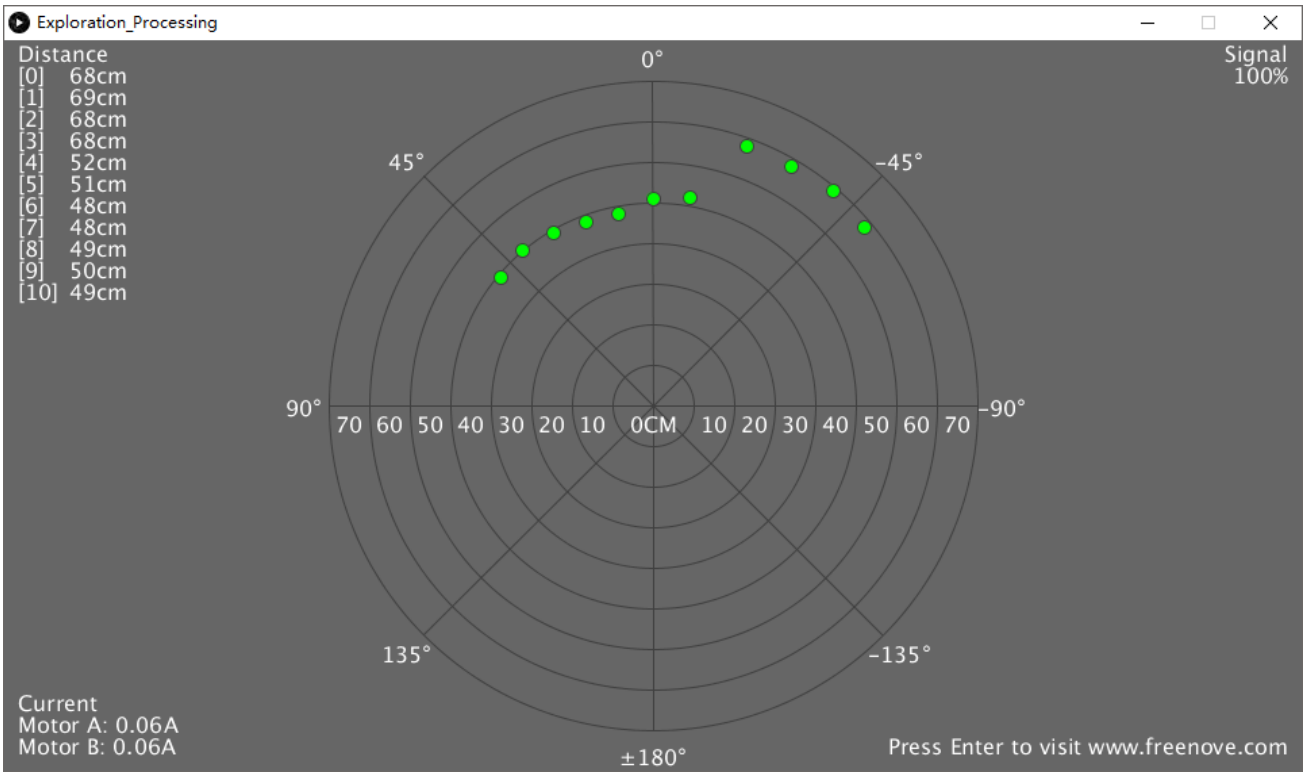
After it is finished, keep the connection of UNO and computer.

Use Processing to open the sketch:

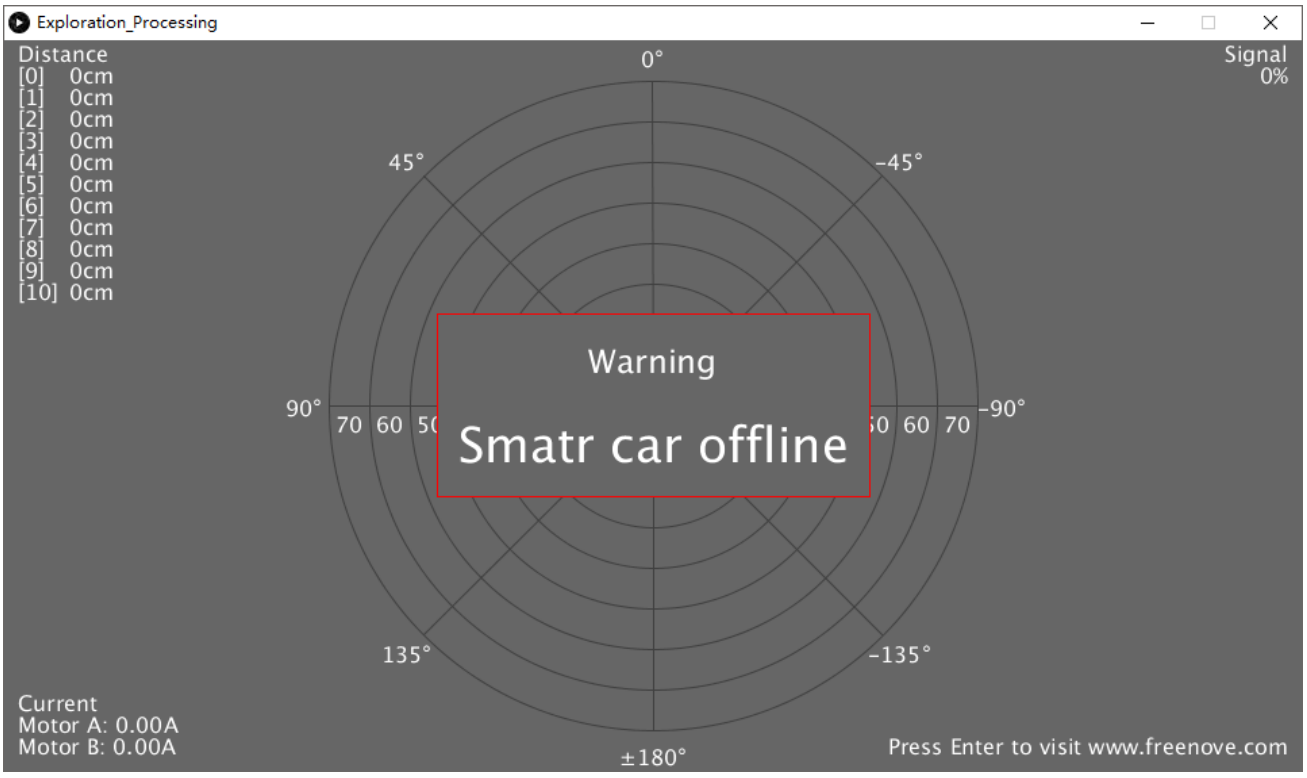
Sketches\Exploration\Exploration_Processing\Exploration_Processing.pde

Click Run. Then, the following window will pop up and its connection to remote control will be started.

If the remote control has not been connected to computer, please connect it to computer. If the connection succeeds, the following will be shown:



If the wireless communication fails, the following will be shown:



Now, you can see that your smart car starts rotating the pan-tilt to detect obstacles, and the Processing display window shows relevant information. The function of this mode includes:

- Display relevant information. Green points on the Processing display window represent the detected obstacles, you can get their direction and distance from the car. The quality of wireless communication and the current of motors will also be displayed.
- Control the smart car to avoid obstacles. Use the joystick on remote control to control the movement of the smart car.
- You can also adjust the front wheels and motors through potentiometer POT1 and POT2 on the remote control.

If your smart car fails to march as expected, please try to eliminate the malfunction in the following ways:

- Pan-tilt can't rotate. Check whether the connection of servo lines is correct.
- Can't show obstacles. Check whether the connection of ultrasonic module lines is correct. If obstacles are far away, they will not be detected and shown.
- Can't show current of motors or the valuesare random. Check whether the jumpers are installed on SNSA and SNSB header pins on Smart Car Shield.
- Smart car offline. Please check whether the sketches are uploaded correctedly, and NRF24L01 module is properly installed. Meanwhile, make sure that the smart car power is open, when the wireless communication is normal, the LED3 on the remote control will light up. If smart car is far away from remote control, the wireless communication may fail.
- Pan-tilt servo deviates its expectation. Modify the value of the variable "plaServoOffset" in the sketch, then upload it again.
- Moving speed of smart car is not as expected. Modify the value of the variable "motorSpeed" in the sketch, then upload it again.
- Other questions. Please contact us for more technical help!

What's next?

Thanks for your reading.

This tutorial is all over here. If you find any mistakes, missions or you have other ideas and questions about contents of this tutorial or the kit and ect, please feel free to contact us, and we will check and correct it as soon as possible.

After completing the projects of this tutorial, you can try to remodify this smart car, including purchasing and installing other electronic modules of Freenove, or improving the code to achieve different functions you want.

If you want to learn more about Arduino, Raspberry Pi, smart cars, robots and orther interesting products in science and technology, please continue to focus on our website. We will continue to launch cost-effective, innovative and exciting products.

Thank you again for choosing Freenove products.