Adeept

Adeept Omni-Directional Wheel Smart Car Kit for Arduino

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-6-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 or shock it.
- The plastic balls can't be squeezed or crashed in case of damage.

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

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

support@adeept.com

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

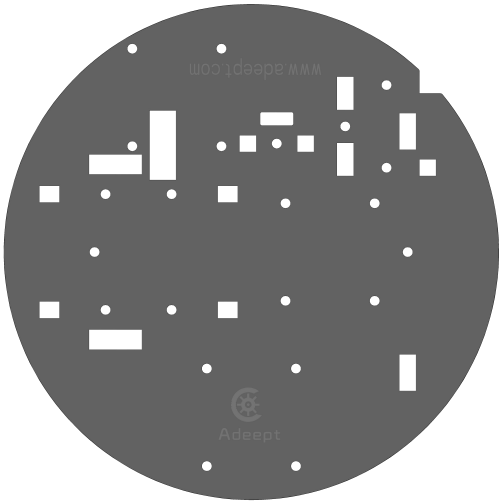
Copyright

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.

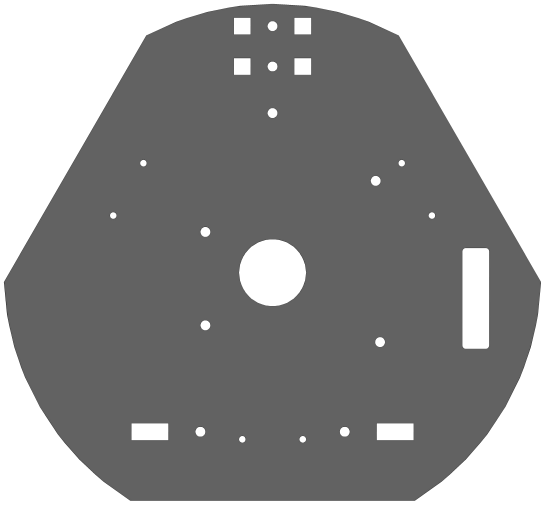
contents

Components List

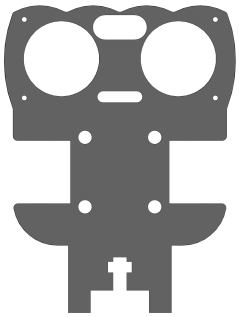
Acrylic Plates



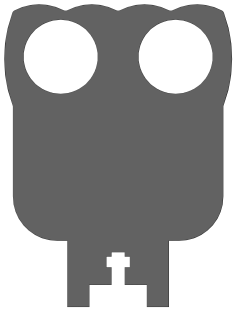
A01
1pcs



A02
1pcs

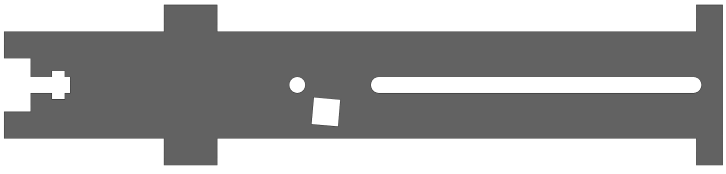


A03
1pcs



A04
1pcs

A05
1pcs



A06
1pcs

A07
1pcs

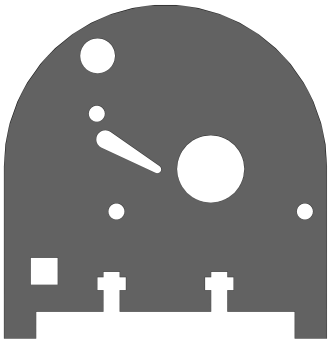
A08
1pcs

A09
1pcs

A10
1pcs

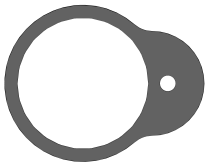
A11
1pcs

A12
1pcs



A13
2pcs

A14
1pcs



A15
1pcs

A16
1pcs

A17
2pcs

A18
1pcs

A19
1pcs

The acrylic plates are fragile, so please be careful when assembling them in case of breaking.
The acrylic plate is covered with a layer of protective film. You need to remove it first.
Some holes in the acrylic may have residues, so you need to clean them before the use.

Machinery Parts

M2 Nut  X2 www.adeept.com	M3 Nut  X27 www.adeept.com	M2*10 Screw  X2 www.adeept.com	M3*4 Screw  X6 www.adeept.com	M3*8 Screw  X10 www.adeept.com
M3*12 Screw  X19 www.adeept.com	M3*35 Screw  X2 www.adeept.com	M4*14 Screw  X1 www.adeept.com	M1.4*6 Self-tapping Screw  X4 www.adeept.com	M3*10 Countersunk Head Screw  X4 www.adeept.com
M3*3 Locking Screw  X3 www.adeept.com	M3*6 Copper Standoff  X4 www.adeept.com	M3*20 Copper Standoff  X2 www.adeept.com	M3*30 Copper Standoff  X1 www.adeept.com	M3*65 Copper Standoff  X1 www.adeept.com

S8D3 Couple Set

S8D3 Coupel
X 3

M4*18 Hexagon socket head cap screws x3

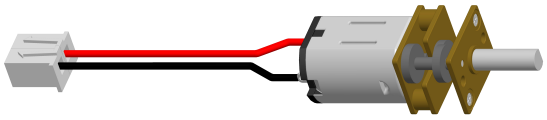
S12D5 Coupler Set

S12D5 Coupler x1

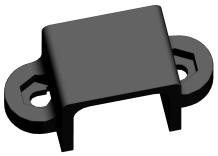
M4*4 Locking Screw x2

Transmission Parts

GA12-N20 Gear Motor Set



GA12-N20 Gear Motor x3

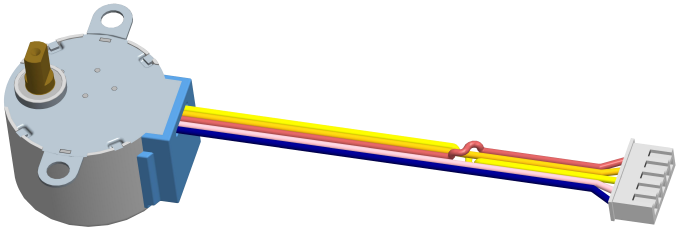


Motor Seat
X 3

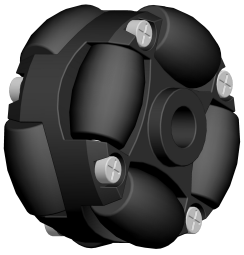
M2 Nut-B
X 6

M2*10 Screw-B
X 6

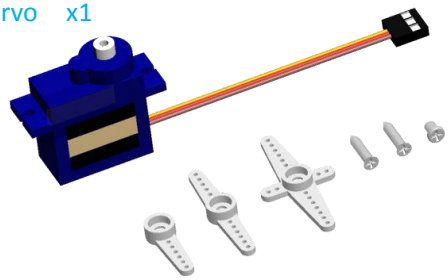
28BYJ-48 Gear Motor x1



Omnidirectional Wheel x3

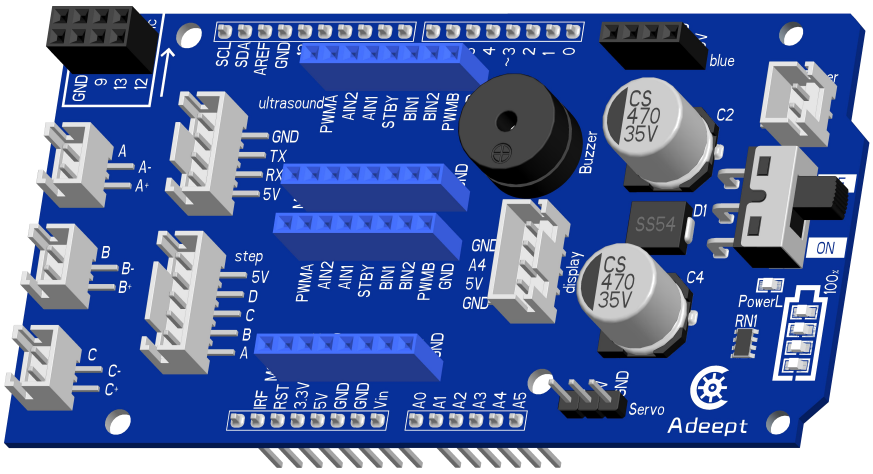


Servo x1

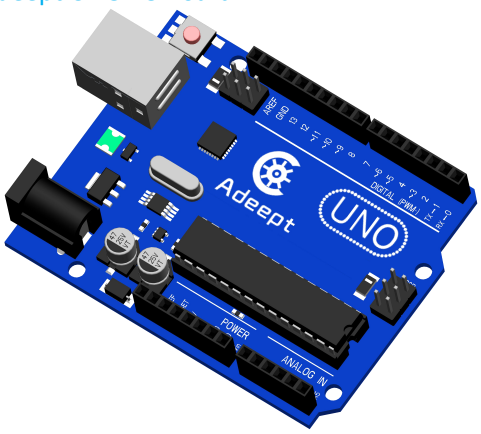


Electronic Parts

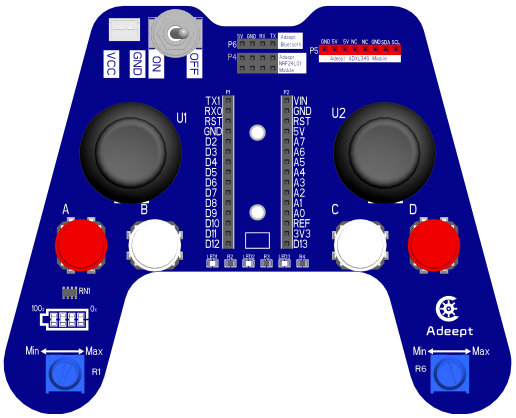
Adeept omnidirectional wheel drive board X1



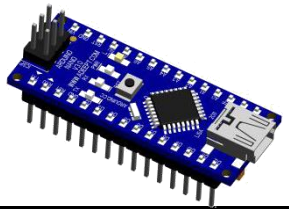
Adeept UNO R3 Board X1



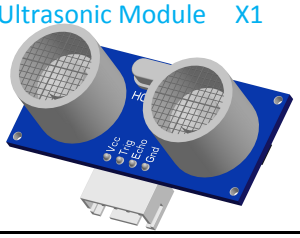
Adeept Remote Control Shield X1



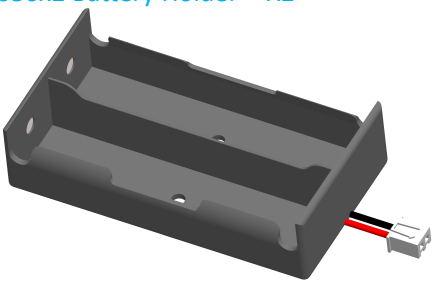
Adeept Nano Board X1



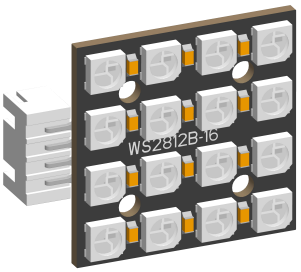
Adeept Ultrasonic Module X1



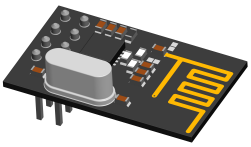
18650x2 Battery Holder X2



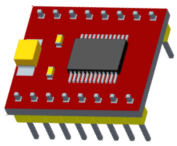
4x4 LED Dot Matrix X1



NRF24L01 X2



Adeept Motor Driver Module X2



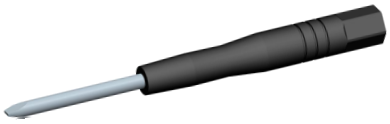
Mini USB Cable X1

USB Cable X1

4-Pin Wires X2

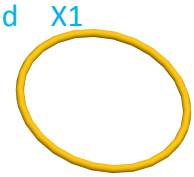
Tools

Slotted Screwdriver x1

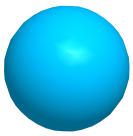


Cross Screwdriver X1

Rubber Band X1



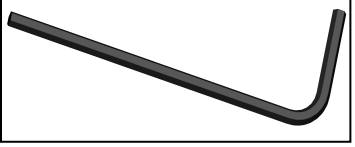
Ball X3



Hex Wrench-1.5mm X1

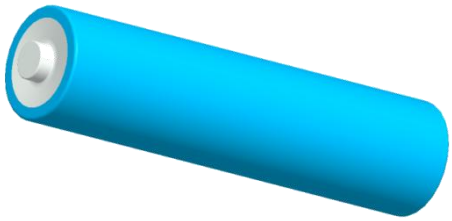
Hex Wrench-2mm X1

Hex Wrench-3mm X1



Self-prepared Parts

18650 Battery X4



About Omnidirectional wheel Smart Car Kit

Overview

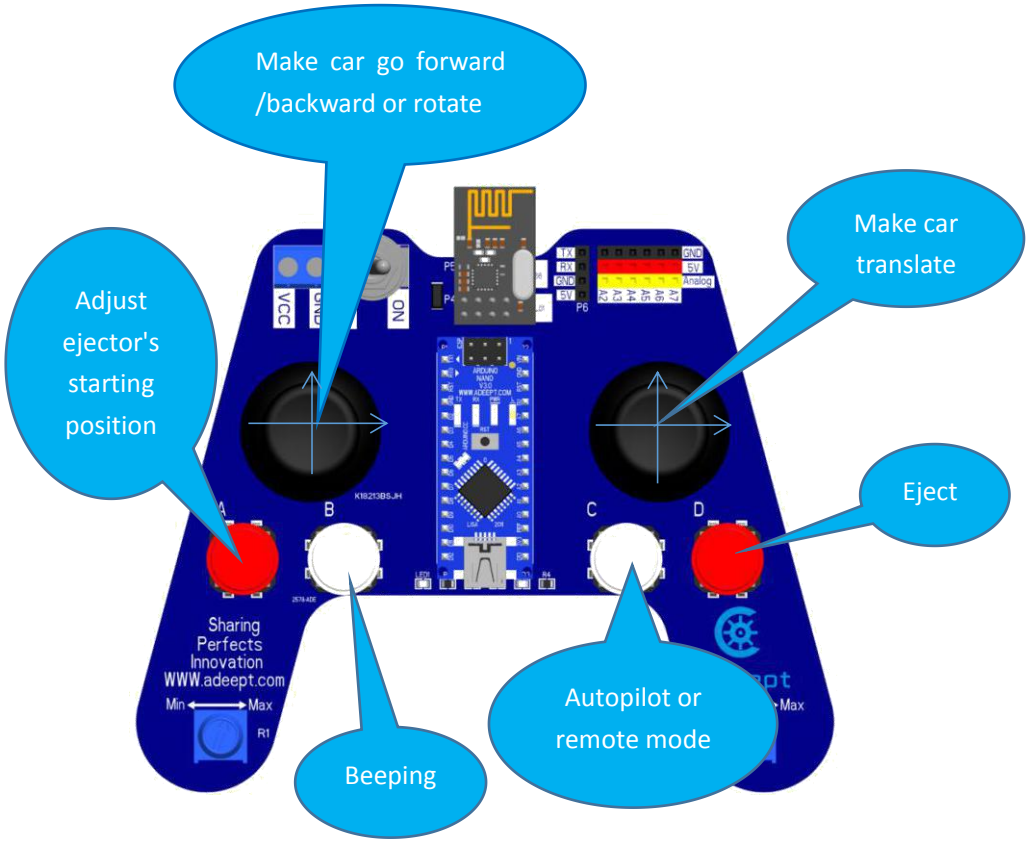
In our daily life, wheels are usually seen in cars and trains. They can only move forward and turn to some directions, not turning in all directions or turning around. But the omnidirectional wheel in this kit can do. Three omnidirectional wheels are the driving ones in this smart robot car. They are installed radially symmetrically; each forms a 120-degree angle with another, and the roller is perpendicular with each wheel. The three wheels are no different in size and quality and driven by motors of the same power. You may control the stick on the remote control to make the car translate, go forward and backward, turn left and right, and whirl and so on. You can also make the car beep by pressing the button on the remote, make it drive automatically or use it as an ejector, etc.

Functions of the omnidirectional car:

1. You may push the stick U1 left and right to control the car to rotate counterclockwise and clockwise. Then the colorful LEDs will display a pattern of left and right arrow. Push the stick forward or pull backward to control the car to go ahead or reverse, and the lights will show an up or down arrow.
2. Move the stick U2 in a certain direction, the robot car will translate in that direction and the lights will show a translation pattern.
3. Press button A to adjust the original position of the ejector on the car, which lies beneath the projector plate (the contact surface of the two). Note: You may adjust the position by button A for many times together with button D. Though the ball may be struck by the servo and cannot slide during the process, don't worry and just take it out or push it in gently. If you feel there is resistance doing so, please switch off the car before pulling the rocker arm of the servo to avoid damage to the servo.
4. Press button B and you may control the buzzer to beep with the remote. As long as the button is pressed down, the buzzer will beep; it still works under the autopilot mode.
5. Press button C and you can change the mode of the car to autopilot and remote. Under the remote mode, the LED2 will light up accordingly and LED3 goes out, reminding you that the car now is in the remote mode; when it's under the autopilot mode (you cannot control it via the stick as it's set), the LED3 will be on and LED2 off. So you know the car is now under the autopilot mode and the lights will show a pattern of the letter "O".
6. Press button D and then you can control the projector part to shoot a ball via the remote (it doesn't work if the ejector is not adjusted to the corresponding starting position).
7. Use the remote to switch the work modes of the car.
8. On the remote, the LED2 is on and LED3 is off by default.

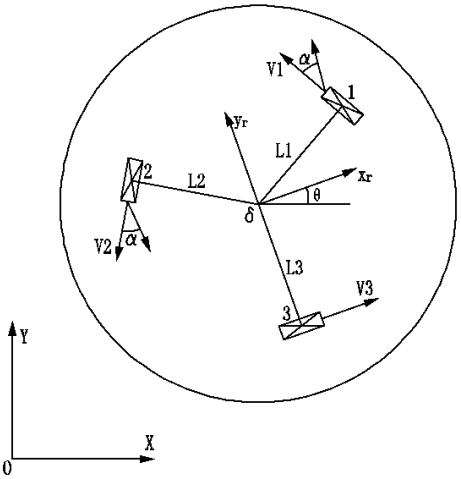
Functions of the car:

1. Once the car is powered on, it automatically enters the mode of remote. Switch the car's work mode (autopilot or remote) by the stick based on the LED2 and LED3 on the remote.
2. After the car is energized, the colorful lights will show the pattern of letter "X", and it is the same when it stops moving.
3. The car can rotate, go forward and backward, and translate in a certain angle.
4. Under the autopilot mode, the car will move forward automatically, rotate counterclockwise if it encounters an obstacle ahead, and continue to advance once it bypasses the obstacle.



Algorithm principle of the omnidirectional car

Analysis of the car's movement:



Explanation: α is the angle between the axis yr and the wheel; L1, L2, and L3 are the distance between each wheel's center to the base center; v1, v2, and v3 are the linear speed of the center of wheel number i (i=1,2,3). Based on the relationship of resolution and synthesis of velocity in plane motion, this formula is created:

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} -Sin(\alpha+\theta) & Cos(\alpha+\theta) & L \\ -Sin(\alpha+\theta) & -Cos(\alpha+\theta) & L \\ Cos\theta & Sin\theta & L \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \theta \end{bmatrix}$$

In the formula, take counterclockwise as a positive direction; L_1, L_2, and L_3 are distance between the center of each wheel and the surface the car is running on at some point; v_x and v_y are the velocity of the car relative to that of its center. For easier calculation, we take an approximate average value of L_1, L_2, and L_3, which is L. Based on the actual structure, we can see $\alpha=30^\circ$. Meanwhile, the instantaneous center of the wheel is the contact point between the wheel and the surface, so $v_i=r\omega_i$ (i=1,2,3). To simplify the calculation, when creating the relative coordinates, take $\theta=0^\circ$, and put $\alpha=30^\circ$ and $\theta=0^\circ$ into the formula 1, so we can get the final kinematics model of the car, as shown by formula 2:

$$\Rightarrow\begin{cases} v_1=-\frac{1}{2}v_x+\frac{\sqrt{3}}{2}v_y+L\theta \\ v_2=-\frac{1}{2}v_x-\frac{\sqrt{3}}{2}v_y+L\theta \\ v_3=v_x+L\theta \end{cases}$$

And $v_i=r\omega_i, i=(1,2,3)$

$$\Rightarrow\begin{cases} \omega_1=\frac{1}{\mathrm{r}}\left(-\frac{1}{2}v_x+\frac{\sqrt{3}}{2}v_y+L\theta\right) \\ \omega_2=\frac{1}{\mathrm{r}}\left(-\frac{1}{2}v_x-\frac{\sqrt{3}}{2}v_y+L\theta\right) \\ \omega_3=\frac{1}{\mathrm{r}}\left(v_x+L\theta\right) \end{cases}$$

Then we can get the model for the car in the x and y directions:

①for going forward and backward, $v_x=0, v_y=a, \theta=0.$

$$\Rightarrow\begin{cases} \omega_1=\frac{1}{\mathrm{r}}\left(\frac{\sqrt{3}}{2}v_y\right) \\ \omega_2=\frac{1}{\mathrm{r}}\left(-\frac{\sqrt{3}}{2}v_y\right) \\ \omega_3=0 \end{cases}$$

②for translation, $v_x = 0, v_y = 0, \theta = a.$

$$\Rightarrow \begin{cases} \omega_1 = \frac{1}{\mathrm{r}} L\theta \\ \omega_2 = \frac{1}{\mathrm{r}} L\theta \\ \omega_3 = \frac{1}{\mathrm{r}} L\theta \end{cases}$$

③for going left and right, $v_x = a, v_y = 0, \theta = 0.$

$$\Rightarrow \begin{cases} \omega_1 = \frac{1}{\mathrm{r}} \left(-\frac{1}{2} v_x \right) \\ \omega_2 = \frac{1}{\mathrm{r}} \left(-\frac{1}{2} v_x \right) \\ \omega_3 = \frac{1}{\mathrm{r}} (v_x) \end{cases}$$

After the proof, the low level code for the control is as follows:

```
#define VX_VALUE            (0.5f)   // macro definition
#define VY_VALUE            (sqrt(3)/2.f)
#define L_value             (20*0.01f)
void Speed_Moto_Control(float vx,float vy,float vz) //Wheel speed algorithm function
{
  motor_one   = (-VX_VALUE*vx + VY_VALUE*vy + L_value*vz);//Calculate the speed of
motor 1
  motor_two   = (-VX_VALUE*vx - VY_VALUE*vy + L_value*vz);//Calculate the speed of
motor 2
  motor_there = (vx + L_value*vz);//Calculate the speed of motor 3
}
```

This function is to determine the direction of the three motors based on whether the value returned by the algorithm is positive or negative.

Notes:

1. When both the remote and the car are power on, they will connect immediately. The flashing of LED1 on the remote indicates a successful communication. Otherwise, the connection fails and you need to debug it by checking the circuit to see whether the Adeept omnidirectional wheel drive board and UNO R3 Board are well connected. If the pins are not all connected well, please replug them.
2. Possible reasons for disconnection include the followings: not well-connected communication module on the remote and the car, insufficient power (all power indicators are off), abnormal program running, etc.

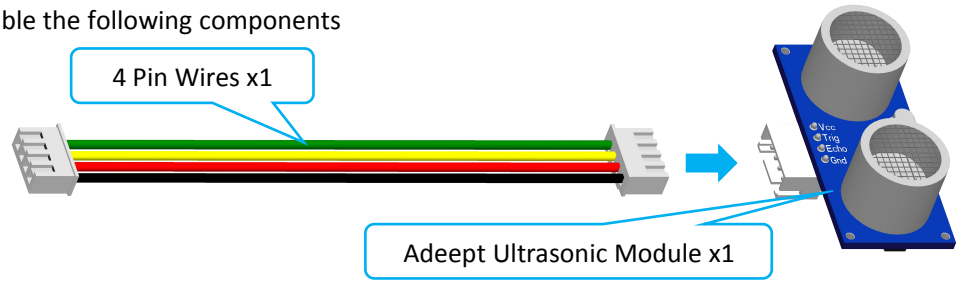
3. When the ejector program is run, the LED1 will go out and the communication will stop. Now the other keys and stick will not work based on setting. After the ejector finishes the task, LED1 will light up again, which means the communication resumes, and you may carry out other tasks.
4. After the car enters the autopilot mode, the stick will not work based on setting but the buttons do.
5. The work mode of the car corresponds to the lights on the remote. When LED2 lights up, it means the car enters the remote mode; when LED3 is on, the car will enter the mode of autopilot.

Assembly

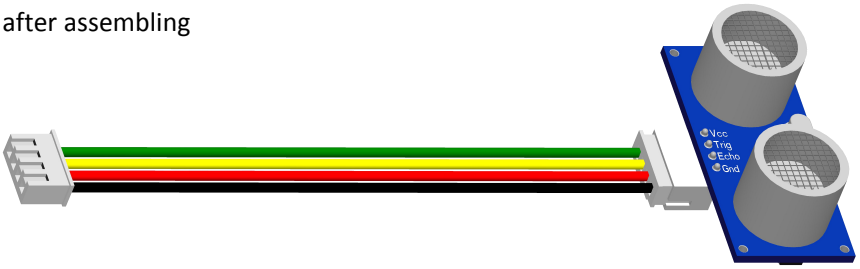
Preparations before Assembly

A. Connect all modules with the correct wires.

Assemble the following components

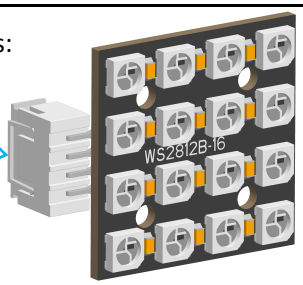


Effect diagram after assembling



Effect diagram of other modules:

For convenient demonstration, hereafter only the connecting end, instead of the entire wire, will be shown in the figure .



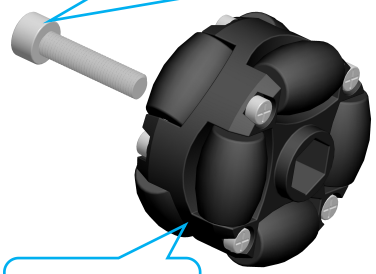
4-Pin Wires + 4x4 LED Dot Matrix

Assemble Bottom

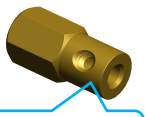
A. Assemble the omnidirectional wheel and the S8D3 Couple (3 sets).

Assemble the following components

M4*18 Hexagon socket head cap screw (in the S8D3 Couple set) x1

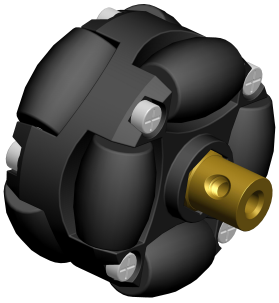


Omnidirectional wheel x1



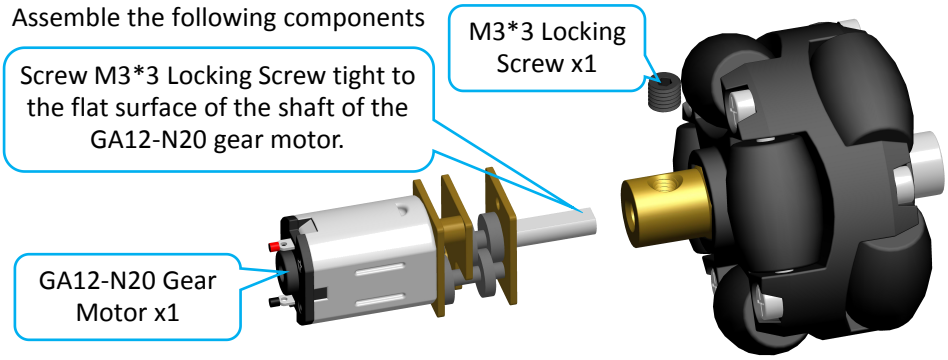
S8D3 Couple x1

Effect diagram after assembling

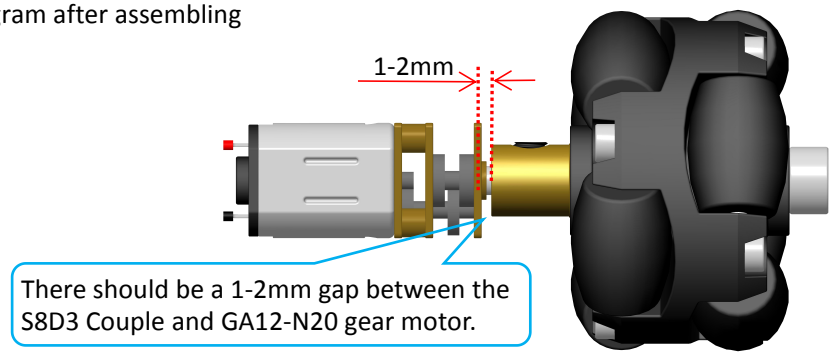


B. Fasten the S8D3 Couple and the GA12-N20 gear motor (3 sets).

Assemble the following components

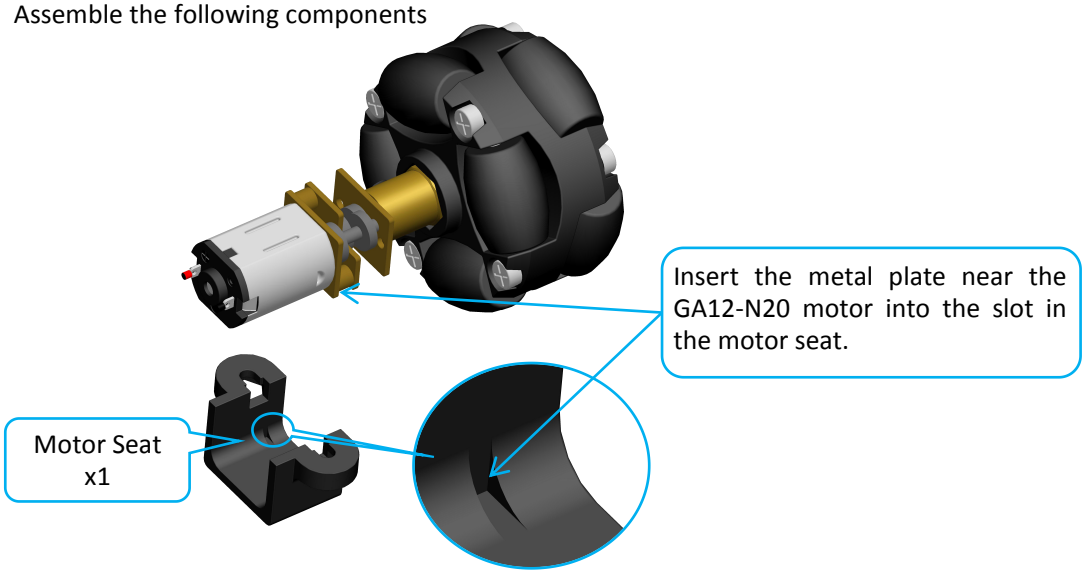


Effect diagram after assembling

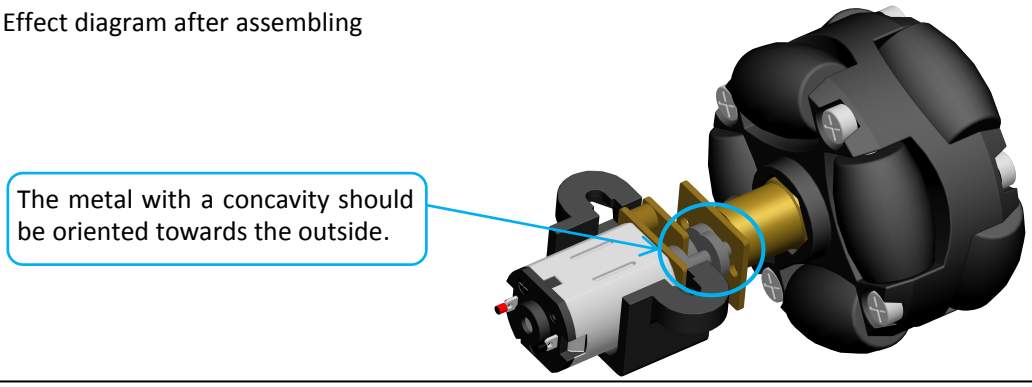


C. Fix the GA12-N20 with the motor seat.

Assemble the following components

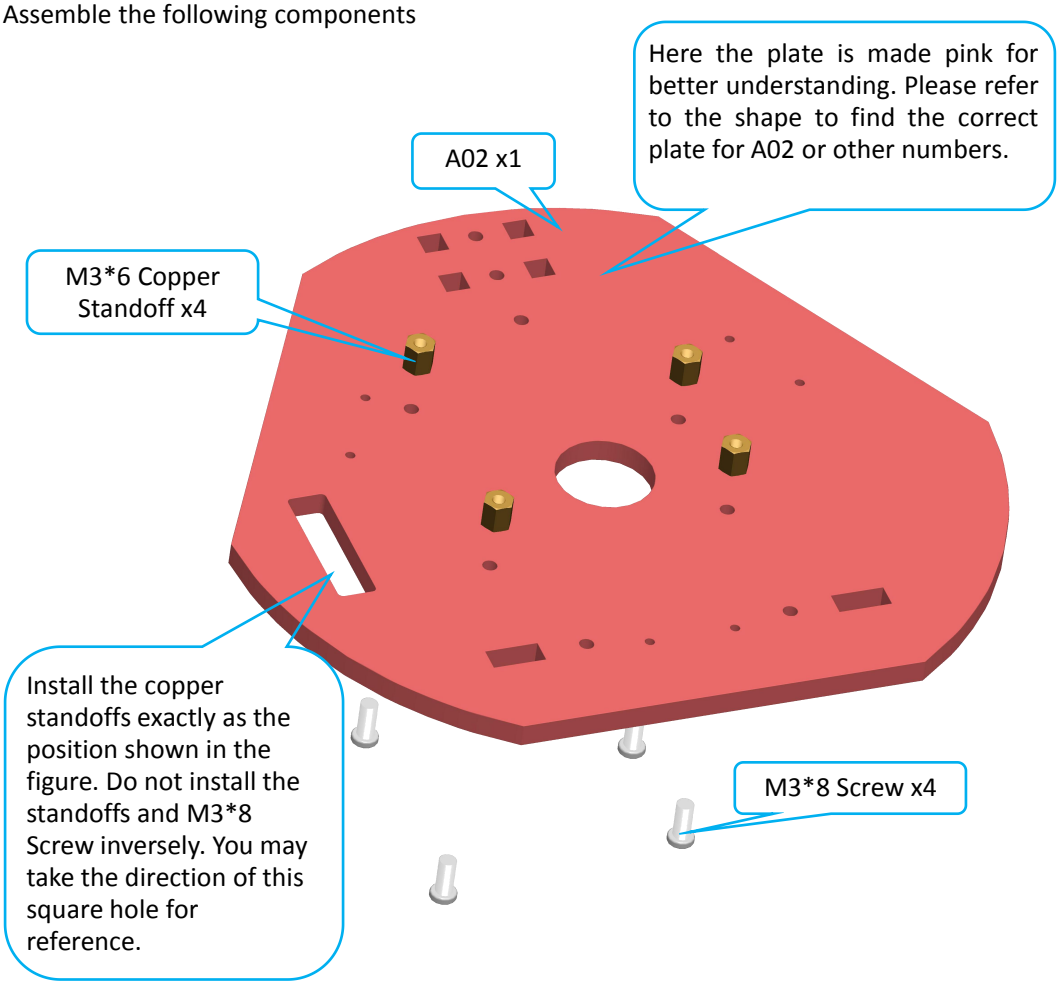


Effect diagram after assembling

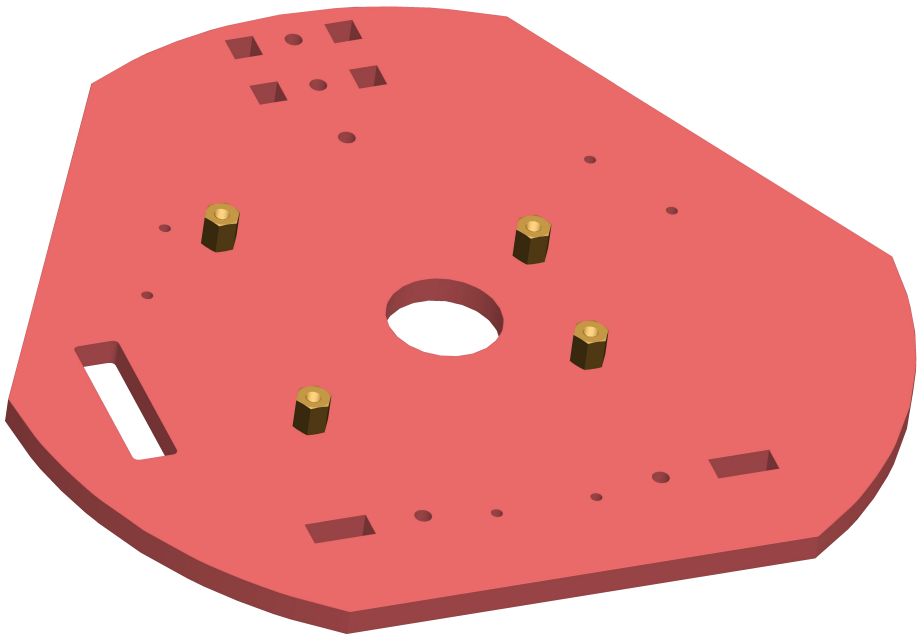


D. Install four M3*6 Copper Standoff onto the A02 plate.

Assemble the following components

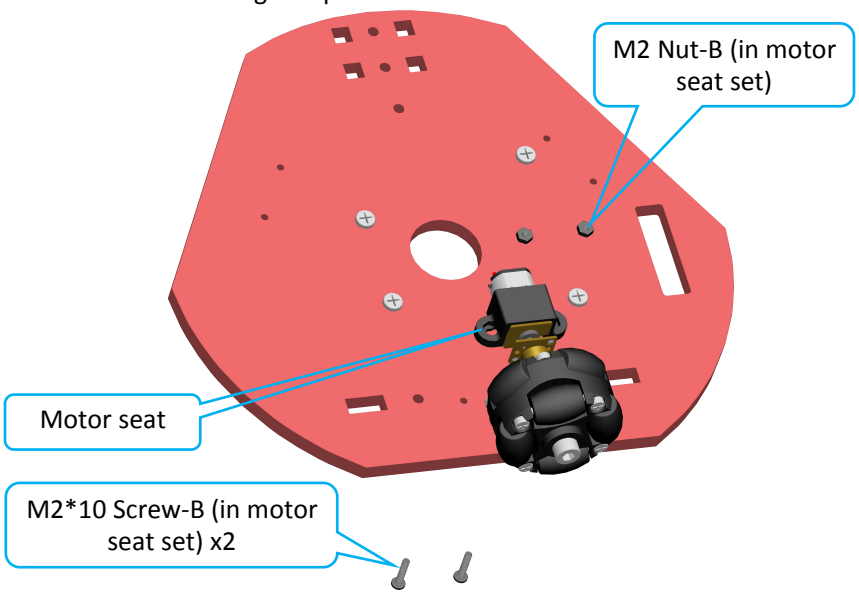


Effect diagram after assembling

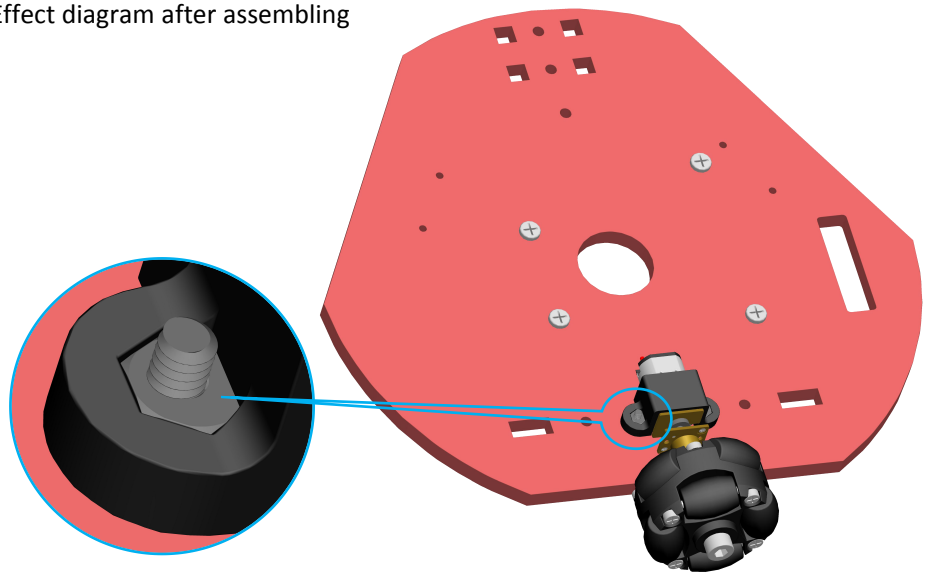


E. Fix the GA12-N20 gear motors onto A02.

Assemble the following components

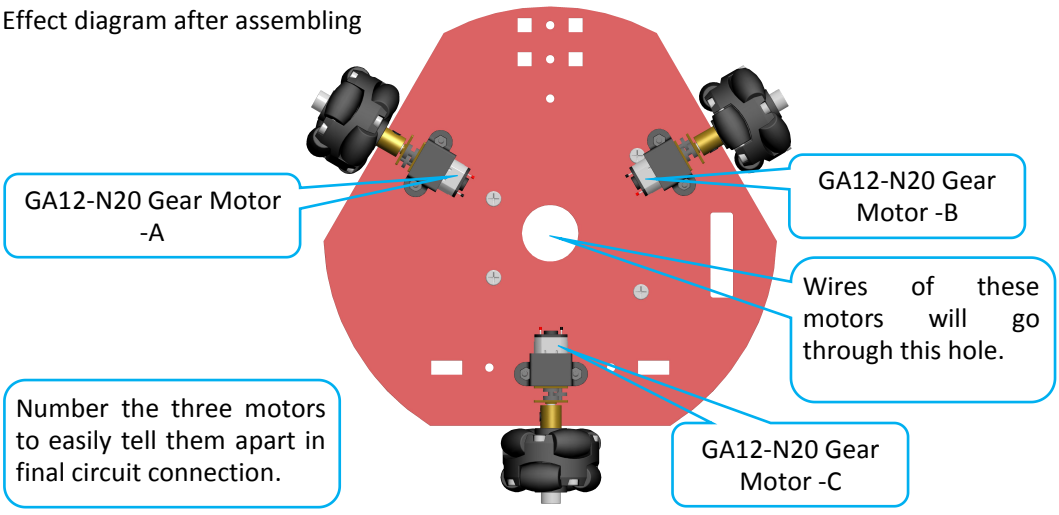


Effect diagram after assembling

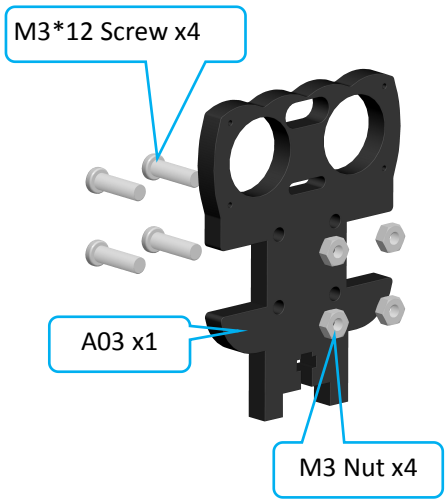
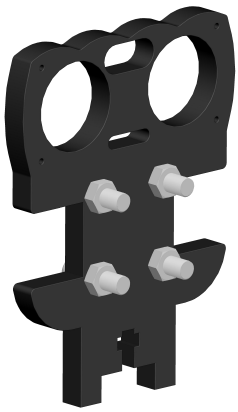


Install the other 2 sets of GA12-N20 motors to A02 in the same way.

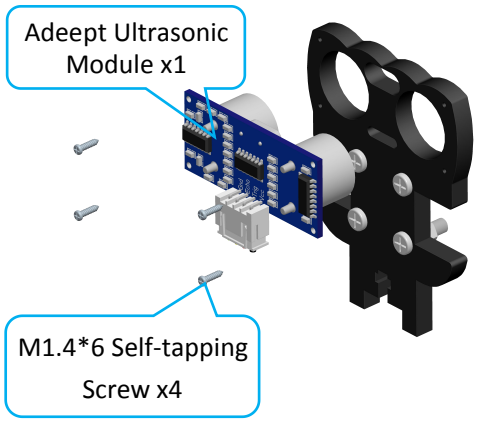
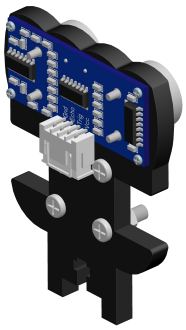
Effect diagram after assembling



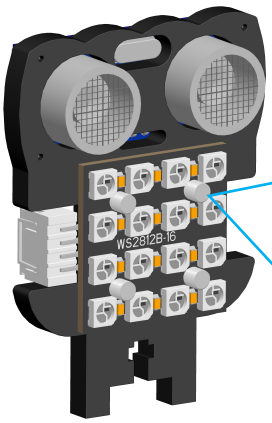
F. Fix 4 M3*12 Screws to A03.

Assemble the following components	Effect diagram after assembling
	

G. Fasten the Adeept Ultrasonic Module and the A03 plate.

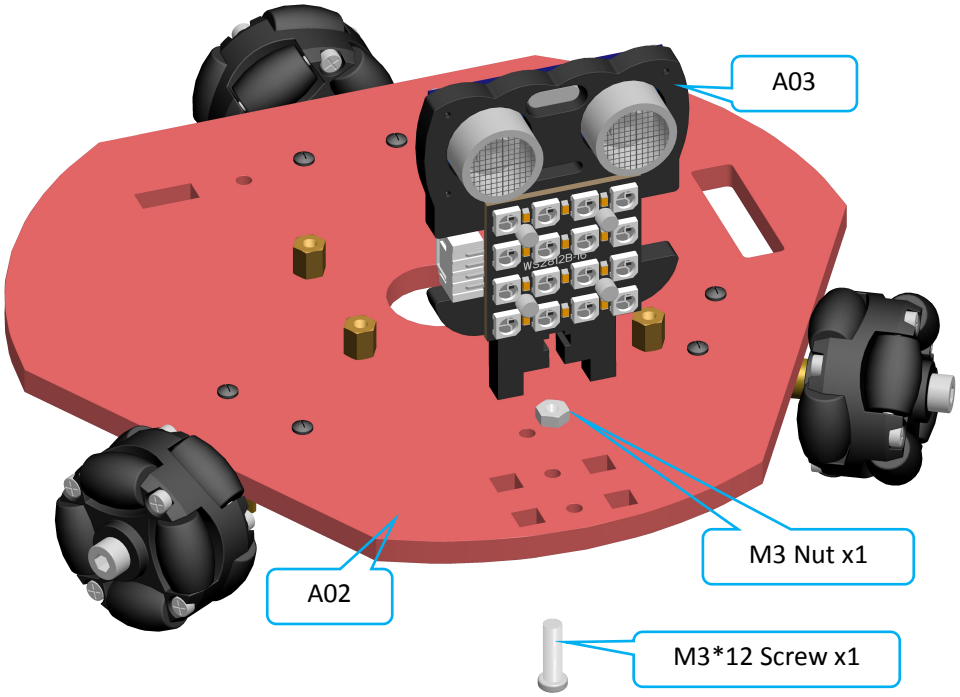
Assemble the following components	Effect diagram after assembling
	

H. Insert the M3x12 Screws into holes on the 4x4 LED Dot Matrix.

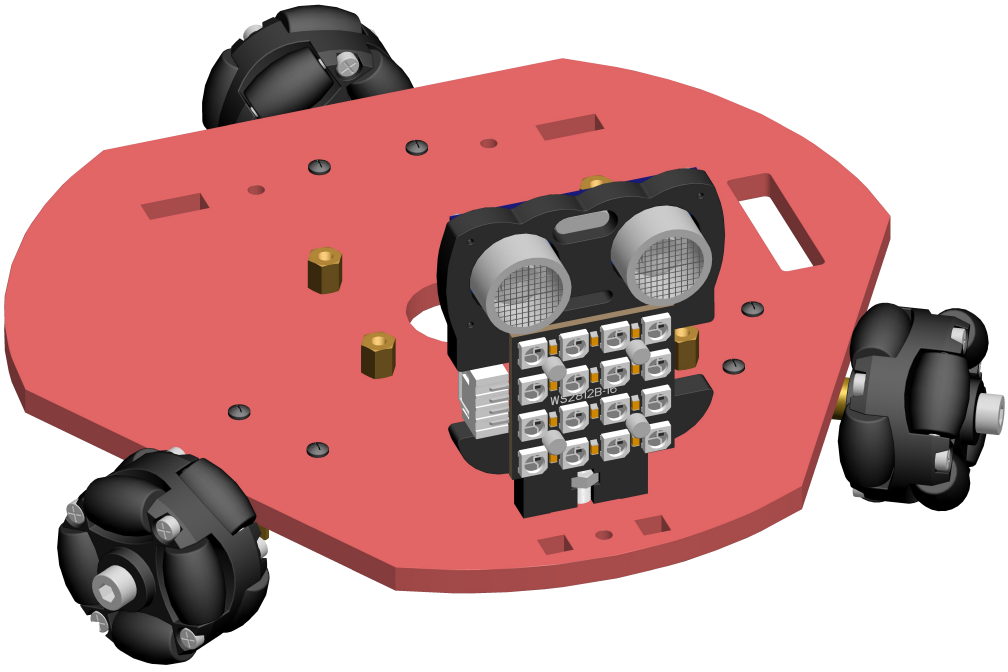
Assemble the following components	Effect diagram after assembling
	 If you cannot insert the screws into the matrix easily, loose the M3x12 Screw a little, and try to plug the 4x4 dot matrix again, and tighten the screws back.

I. Fix the A03 onto the A02 plate.

Assemble the following components

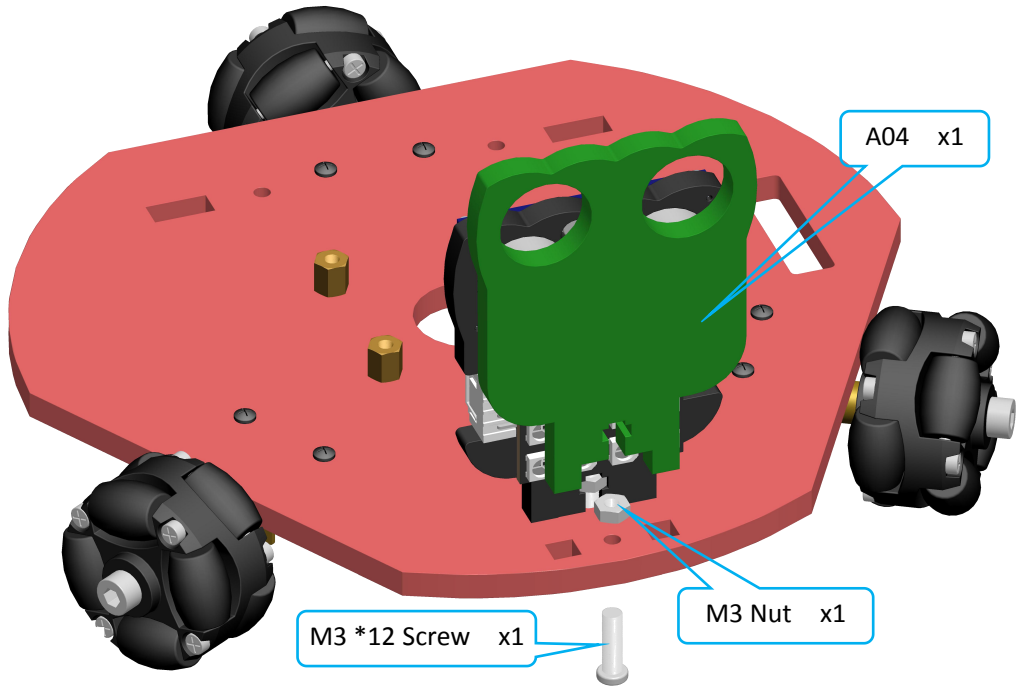


Effect diagram after assembling

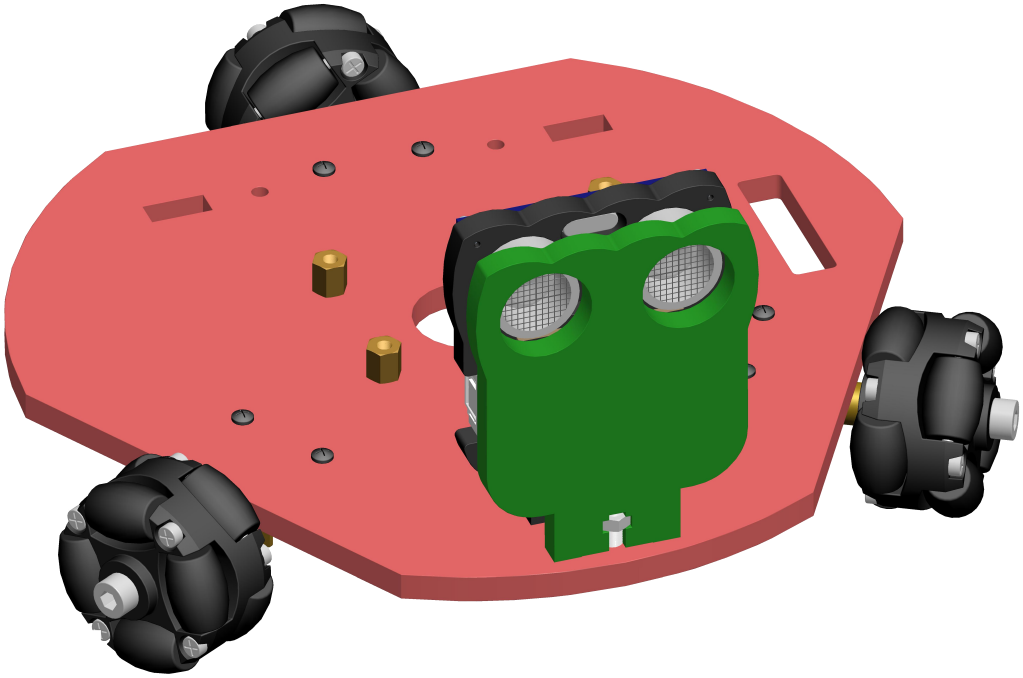


J. Install A04 onto A02.

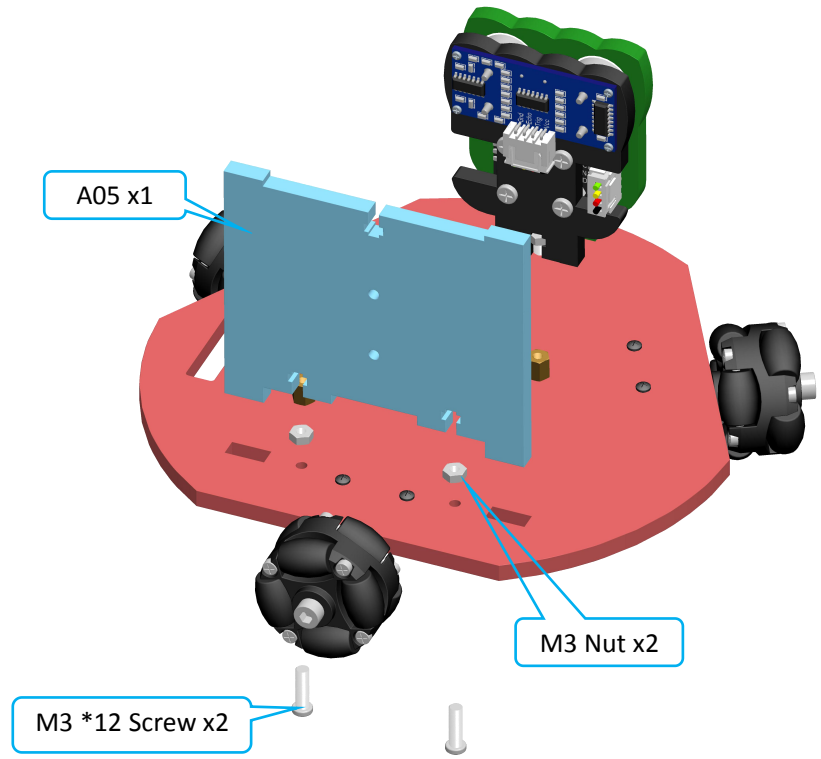
Assemble the following components



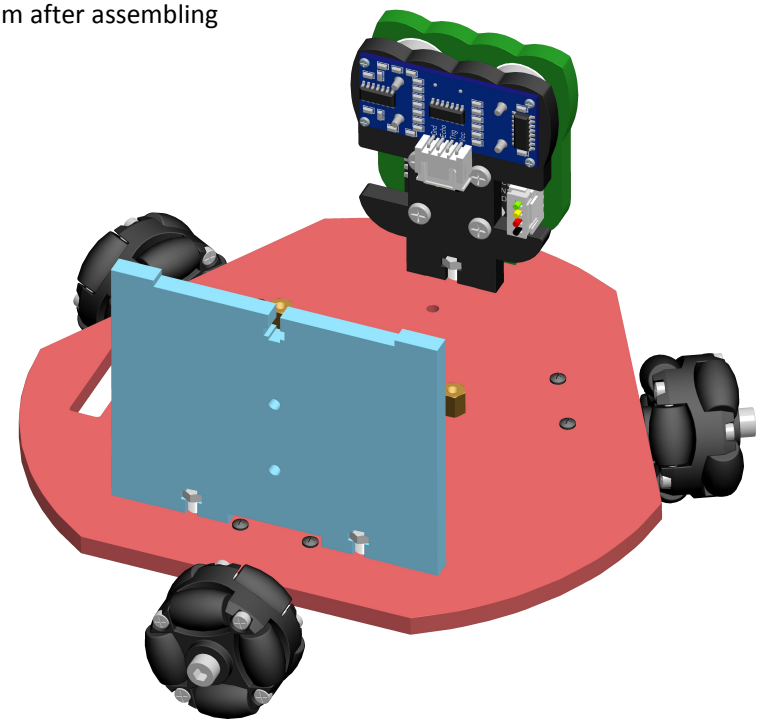
Effect diagram after assembling



K. Fasten A05 and A02.

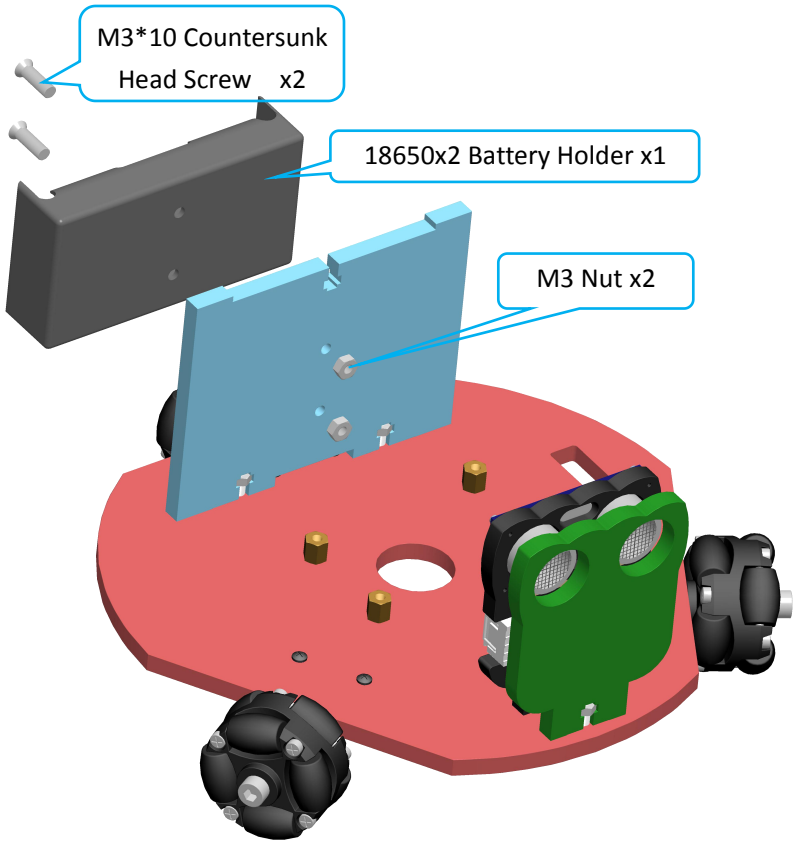


Effect diagram after assembling

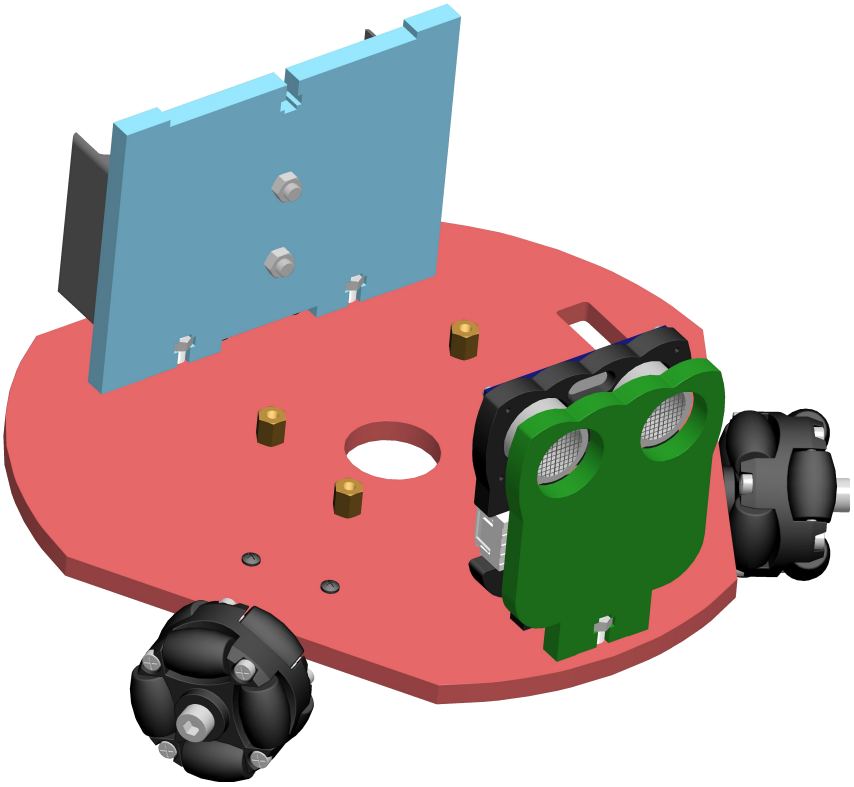


L. Install the 18650x2 Battery Holder onto the A05 plate.

Assemble the following components

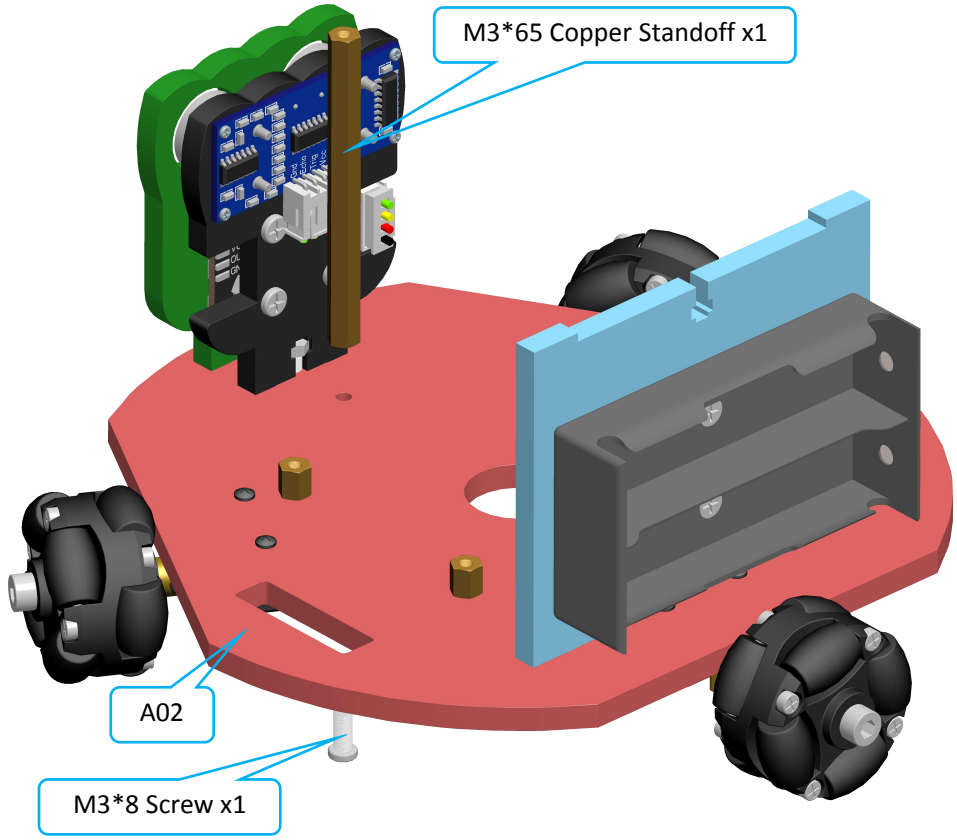


Effect diagram after assembling

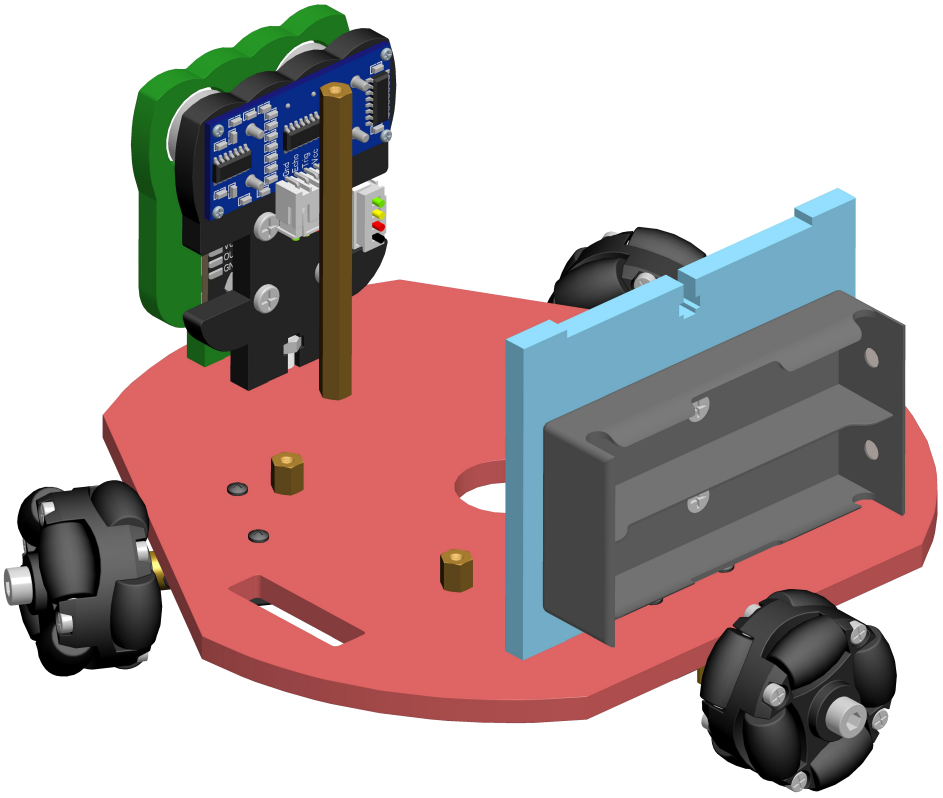


M. Fix M3*65 Copper Standoff onto A02.

Assemble the following components

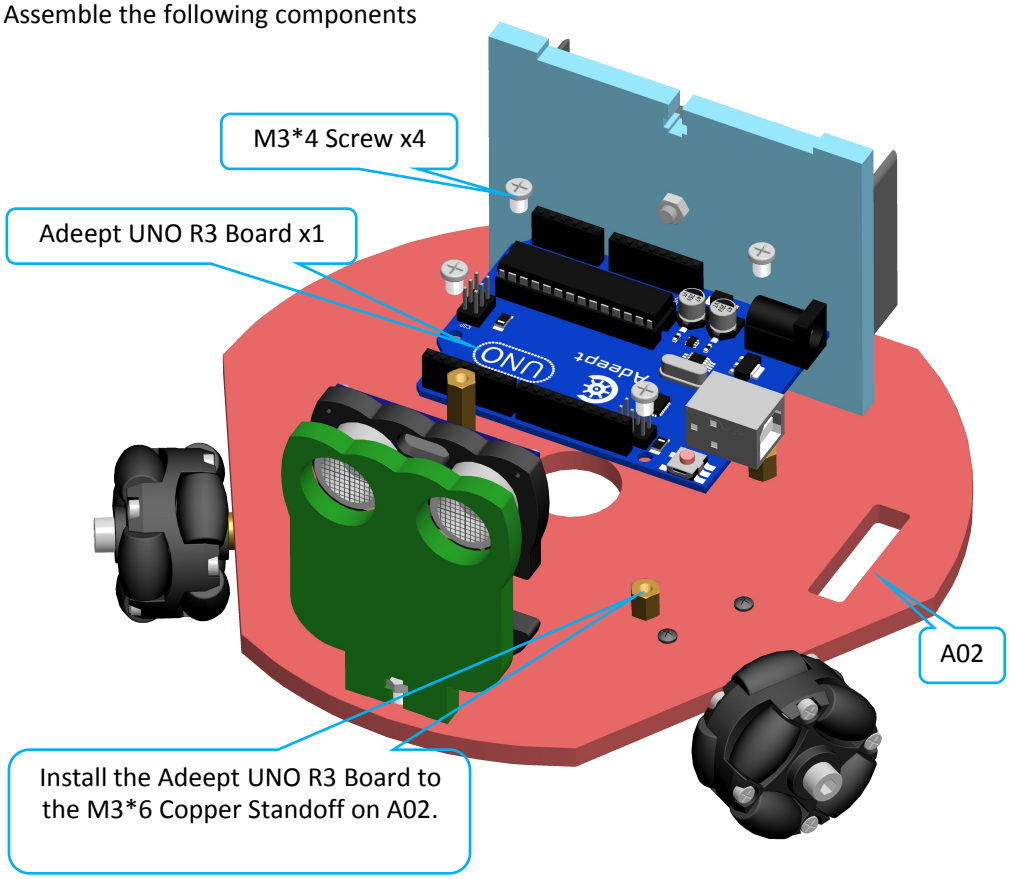


Effect diagram after assembling

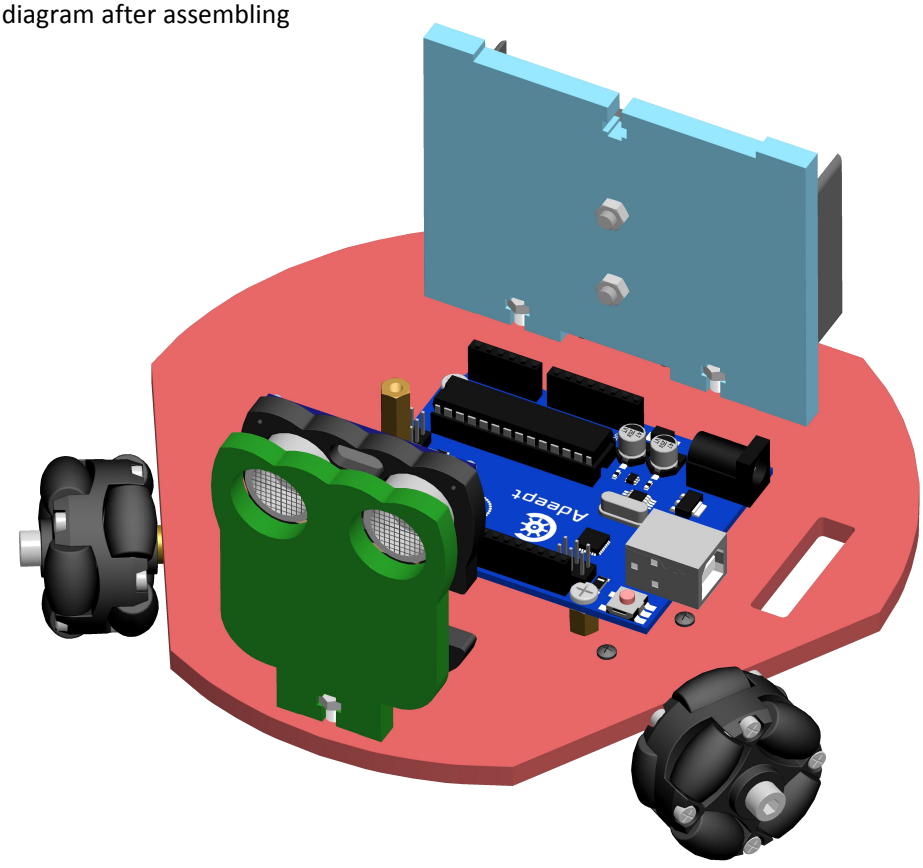


N. Fasten the Adeept UNO R3 Board onto A02.

Assemble the following components

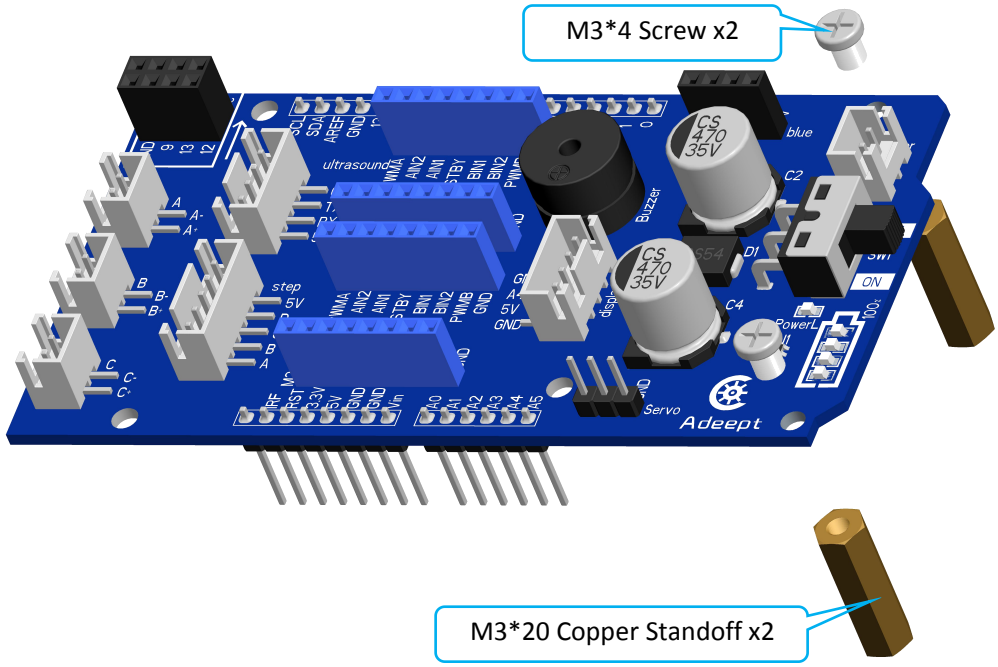


Effect diagram after assembling

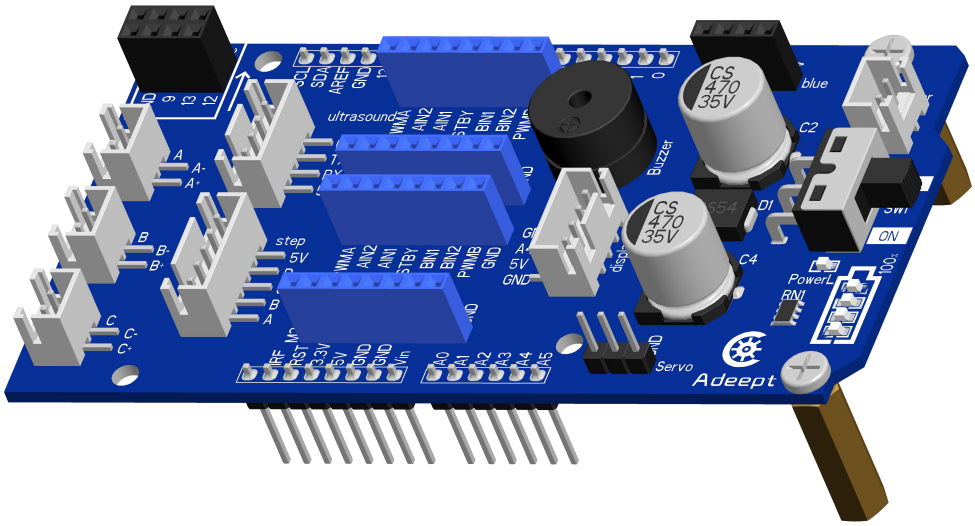


O. Install 2 M3*20 Copper Standoffs onto the Adeept omnidirectional wheel drive board.

Assemble the following components

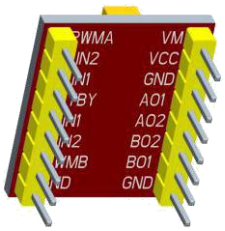


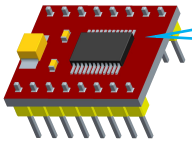
Effect diagram after assembling



P. Insert the Adeept Motor Driver Module into the Adeept omnidirectional wheel drive board.

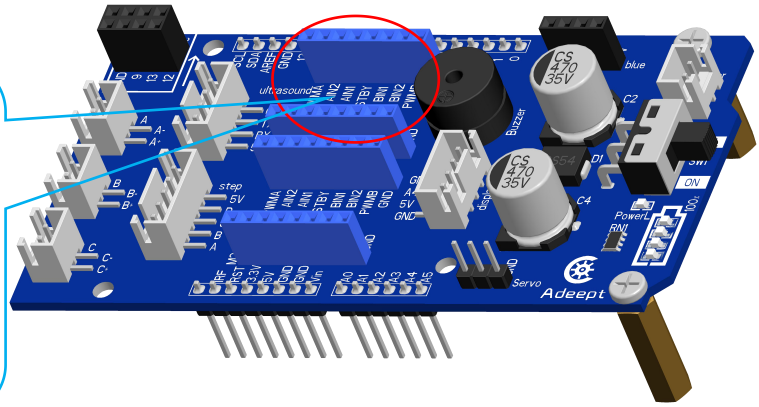
Assemble the following components



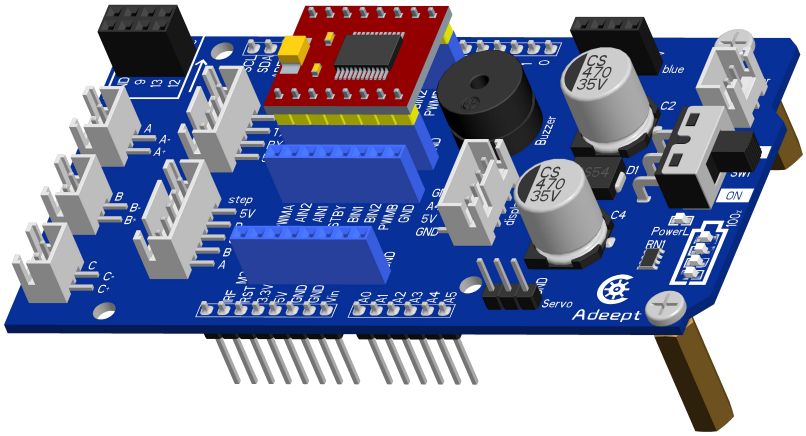


Adeept Motor Driver Module x1

The pin labels on the Adeept Motor Driver Module should be on the same side with those on the Adeept omnidirectional wheel drive board. DO NOT plug in the Adeept Motor Driver Module inversely.

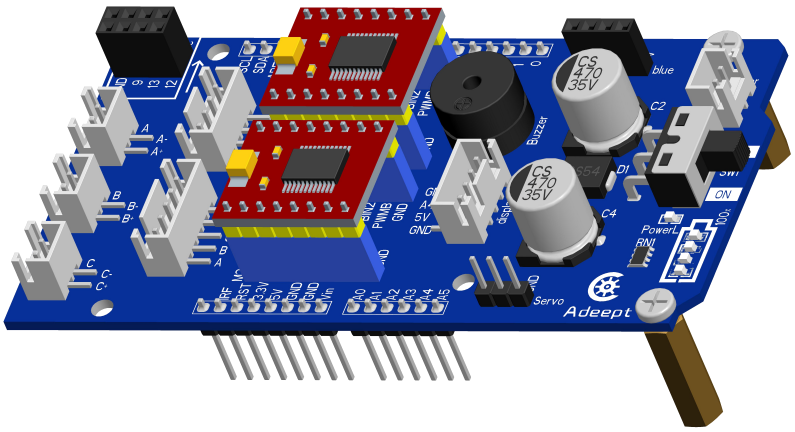


Effect diagram after assembling



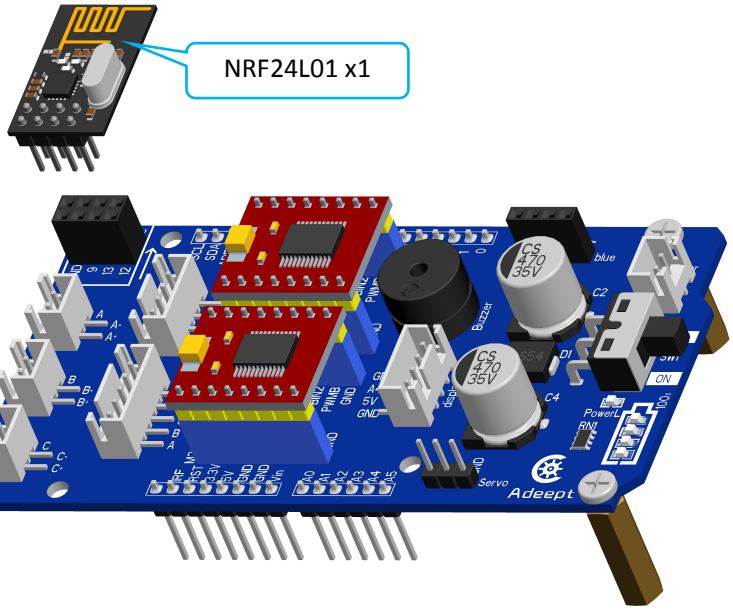
Install the other Adeept Motor Driver Module onto the Adeept omnidirectional wheel drive board in the same way.

Effect diagram after assembling

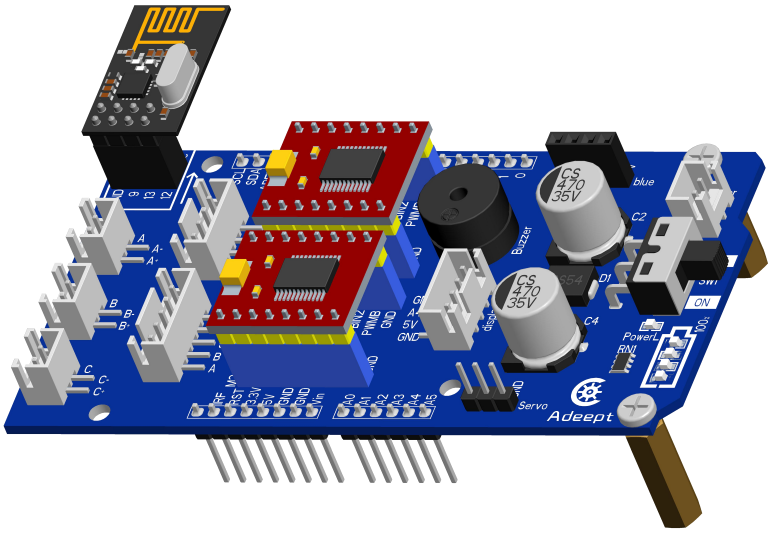


Q. Insert the NRF24L01 module into the Adeept omnidirectional wheel drive board.

Assemble the following components

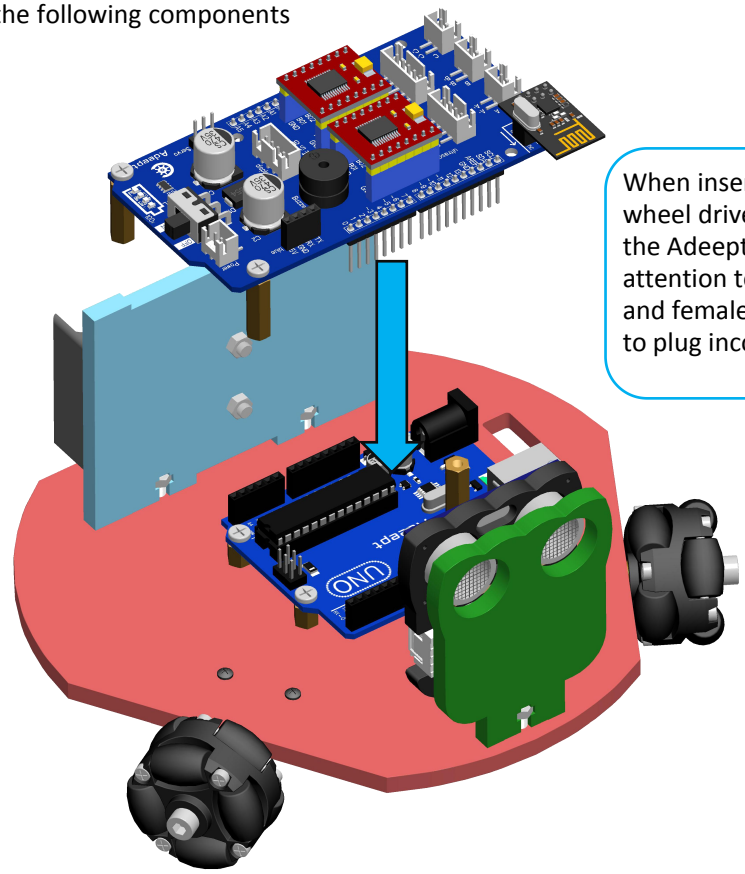


Effect diagram after assembling



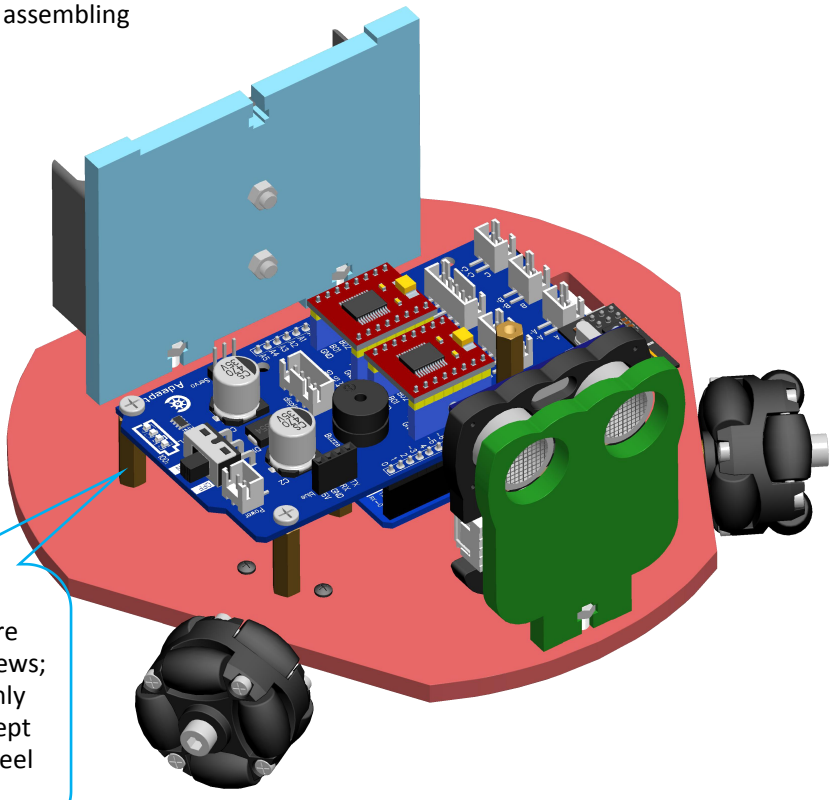
R. Then insert Adeept omnidirectional wheel drive board into Adeept UNO R3 Board.

Assemble the following components



When inserting the wheel drive board into the Adeept UNO R3, pay attention to align male and female pins and not to plug incorrectly.

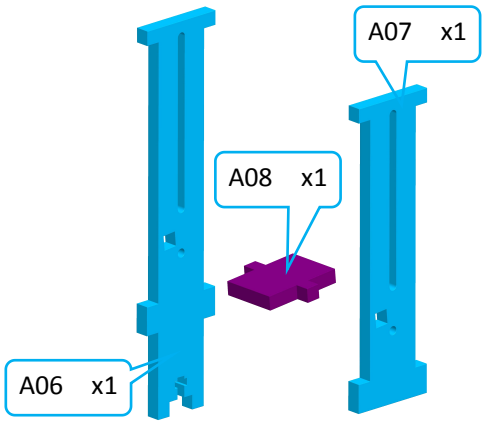
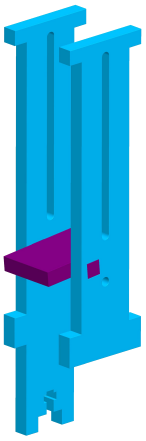
Effect diagram after assembling



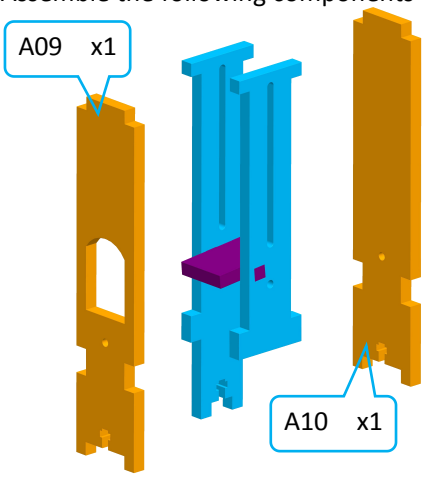
The M3*20 Copper Standoff and A02 are not fastened by screws; the standoffs are only to prop up the Adeept omnidirectional wheel drive board.

Assemble Top

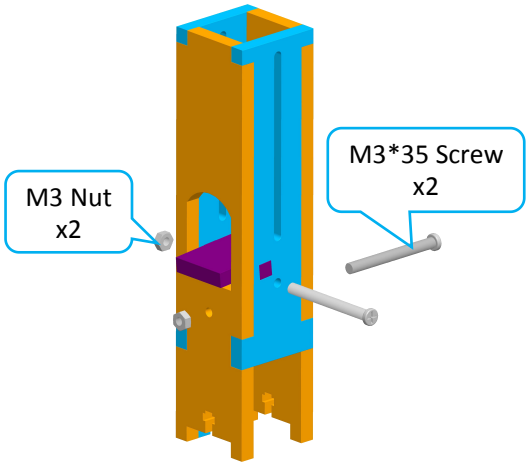
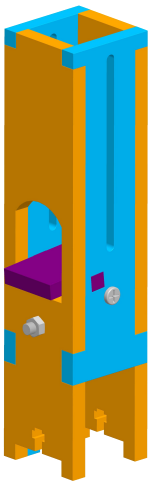
A. Fasten the plates A06, A07, and A08.

Assemble the following components	Effect diagram after assembling
	 The plates are now loose; please hold them gently to keep them fastened.

B. Then assemble with A09 and A10.

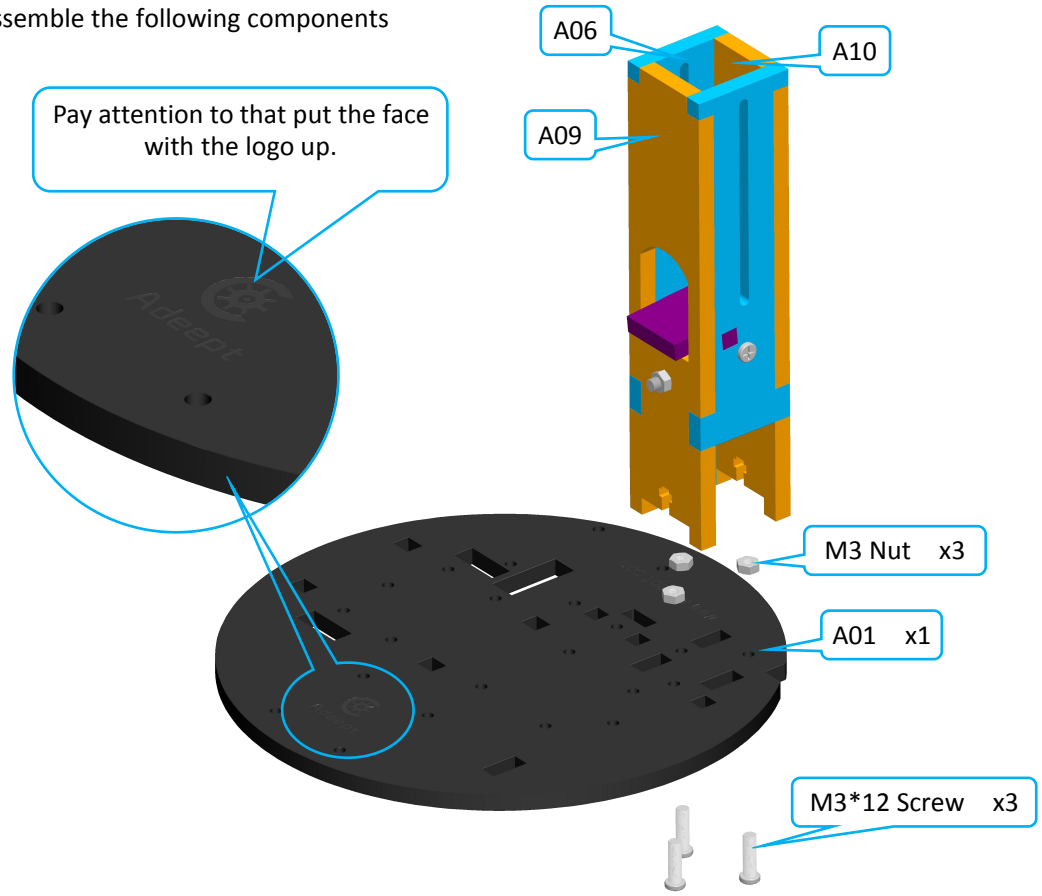
Assemble the following components	Effect diagram after assembling
	 The plates are now loose; please hold them gently to keep them fastened.

C. Solidify these parts with M3*35 Screws.

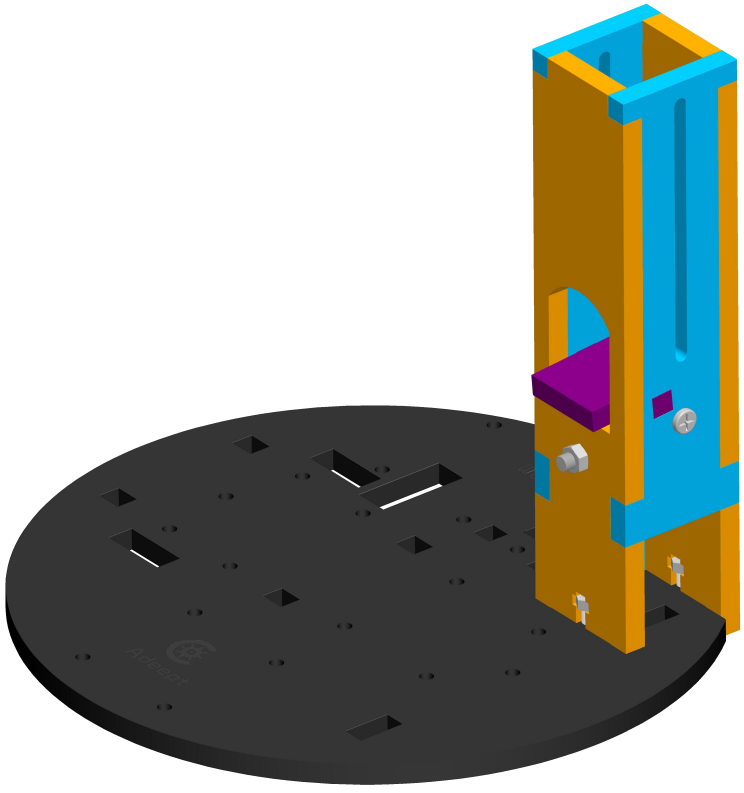
Assemble the following components	Effect diagram after assembling
	Insert the M3*35 Screws through A06 and A17, and A09 and A10, before into the nuts. Tighten the screws and nuts, so the part is fixed. 

D. Fasten the A06, A09, and A10 part onto the A01 plate.

Assemble the following components

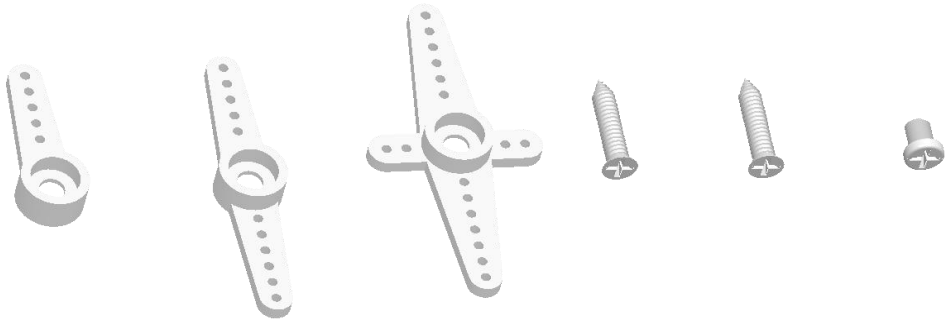


Effect diagram after assembling

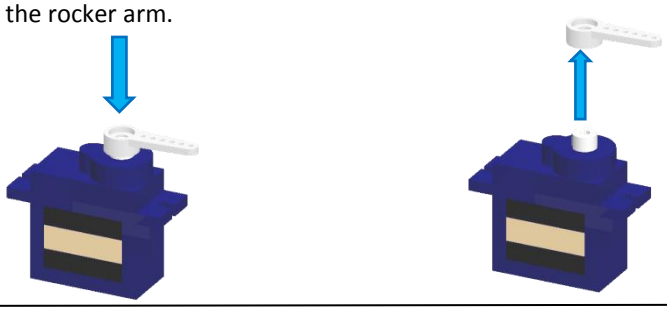


E. Calibrate the servos.

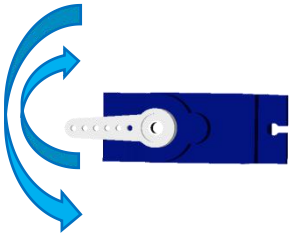
First, learn the structure. The servo can connect the rocker arm and spin to drive components bound with the arm. There are 3 types of rocker arms and 3 screws in the package. The smallest one is to fix the arm onto the servo.



Mount and remove the rocker arm.

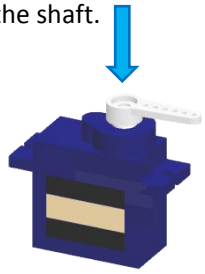


Rotate the rocker arm between 0 and 180 degrees.

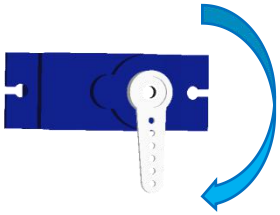


Now adjust the servo - leave the servo shaft at the initial position.

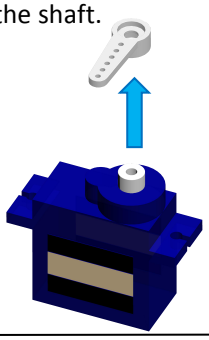
Take out a rocker arm and install it to the shaft.



Rotate the arm counterclockwise until it meets the end.



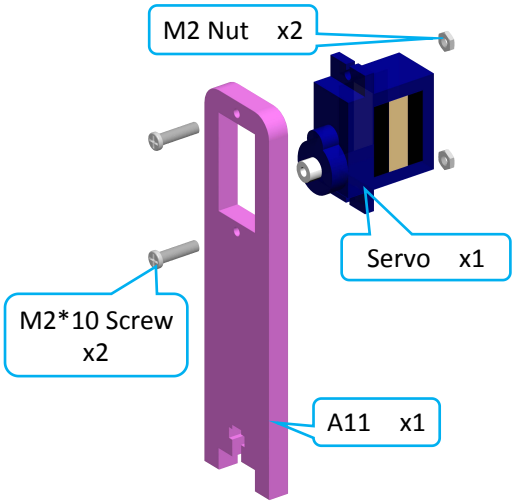
Remove the arm vertically. Do not spin the shaft.



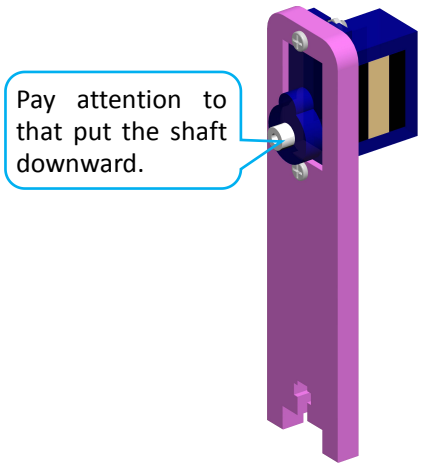
Note: Before install the arm back, do not spin the servo shaft; otherwise, repeat the step and spin the shaft back to the initial position.

F. Install the servo into A11.

Assemble the following components

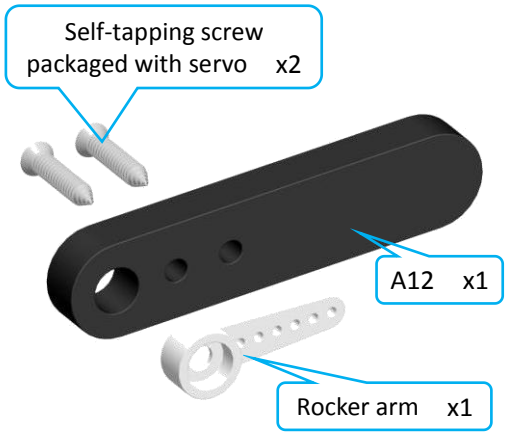


Effect diagram after assembling

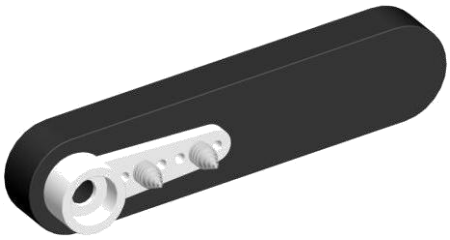


G. Install the rocker onto A12.

Assemble the following components

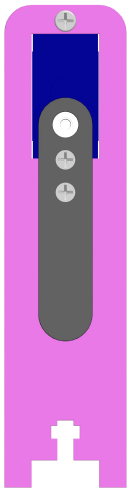


Effect diagram after assembling



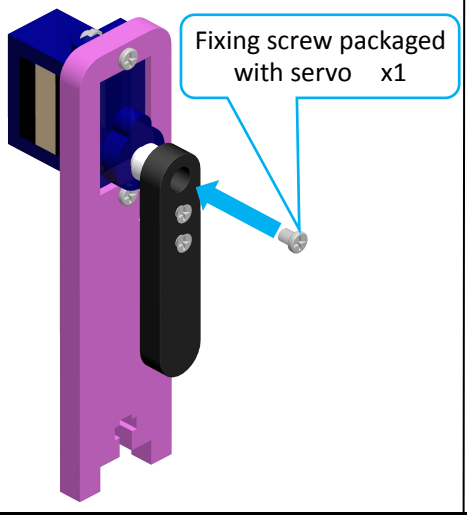
H. Fasten the servo and rocker arm.

Insert the shaft into the hole on the arm (downward vertically)

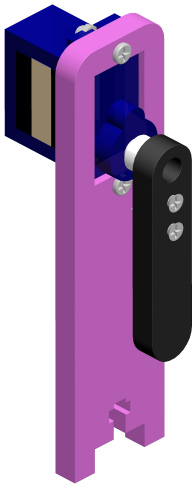


Fix the arm with the smaller screw in the servo package in case of movement.

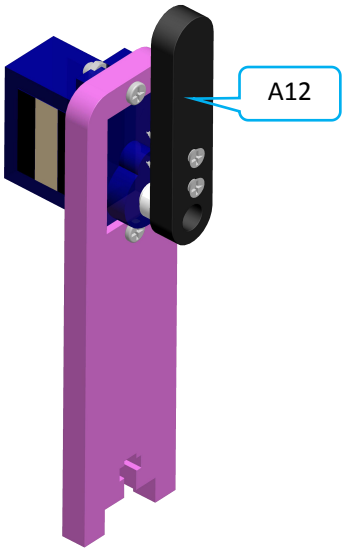
Assemble the following components



Effect diagram after assembling

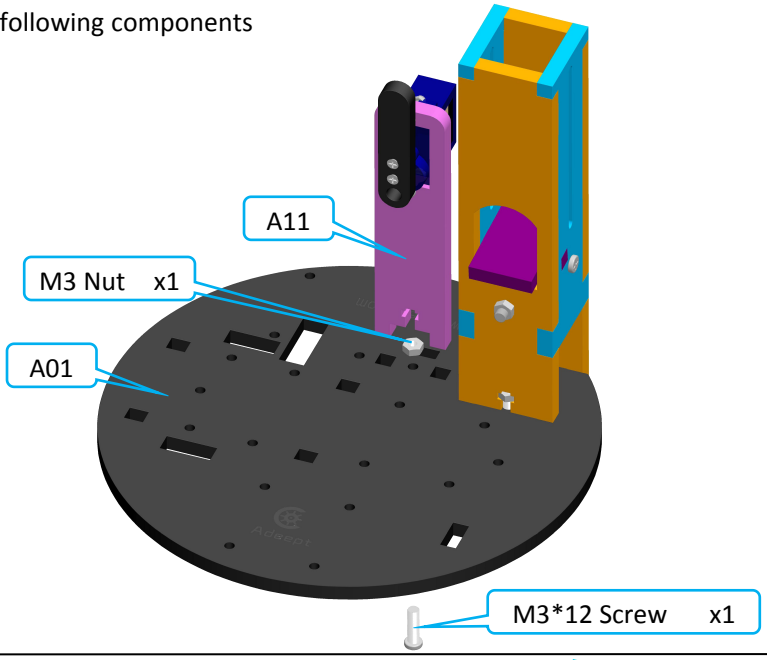


After assembly, spin A12 counterclockwise to upward vertically.

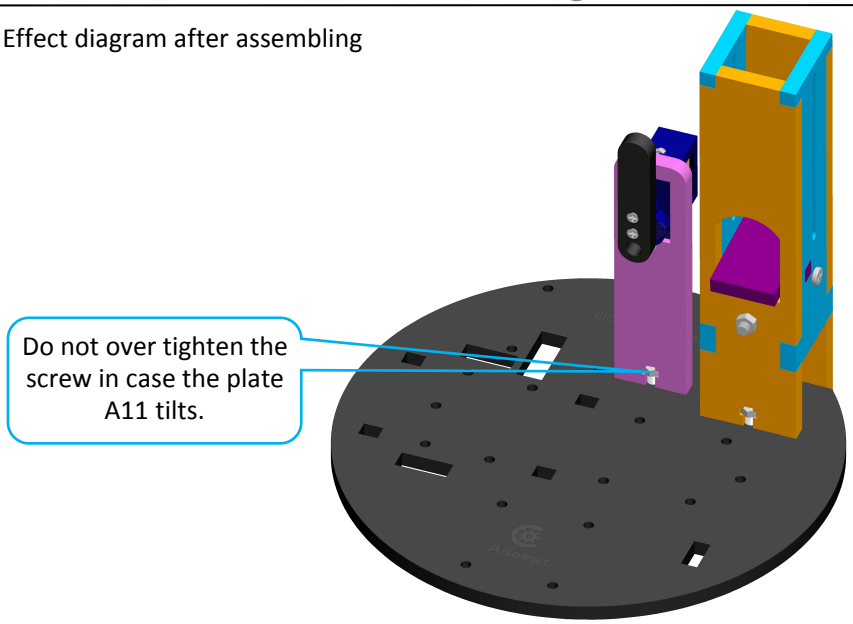


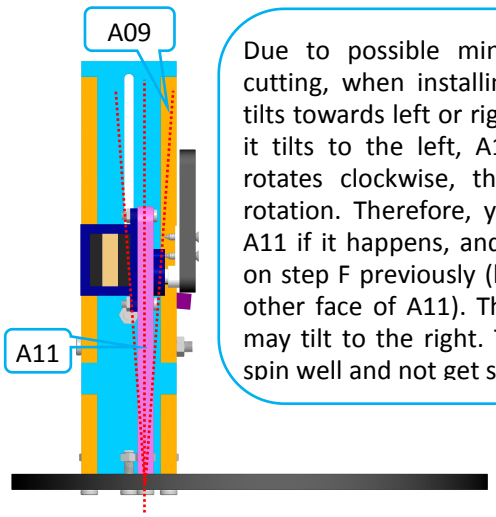
I. Fasten the plates A11 and A01.

Assemble the following components

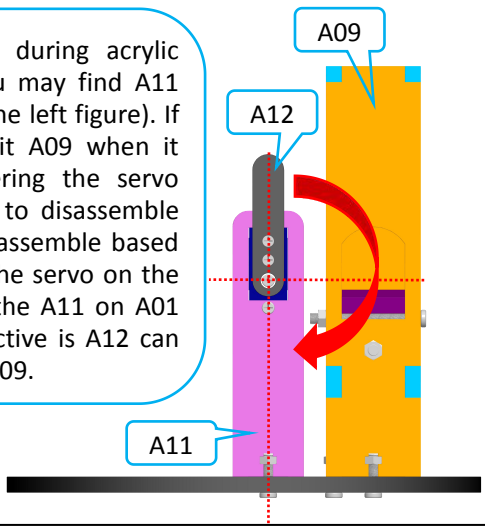


Effect diagram after assembling



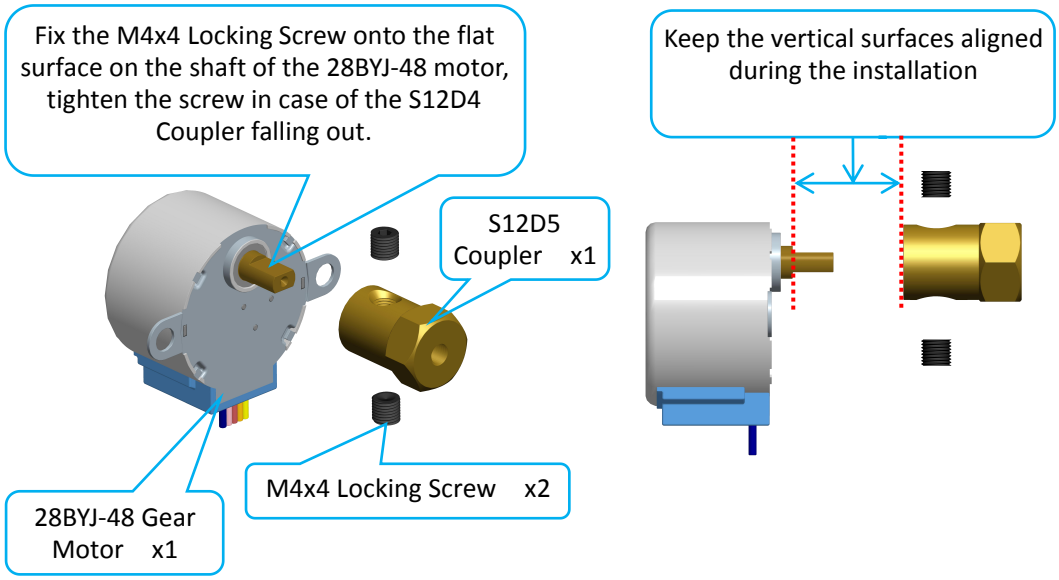


Due to possible minor error during acrylic cutting, when installing it, you may find A11 tilts towards left or right (see the left figure). If it tilts to the left, A12 will hit A09 when it rotates clockwise, thus hindering the servo rotation. Therefore, you need to disassemble A11 if it happens, and then reassemble based on step F previously (but put the servo on the other face of A11). This time the A11 on A01 may tilt to the right. The objective is A12 can spin well and not get stuck by A09.

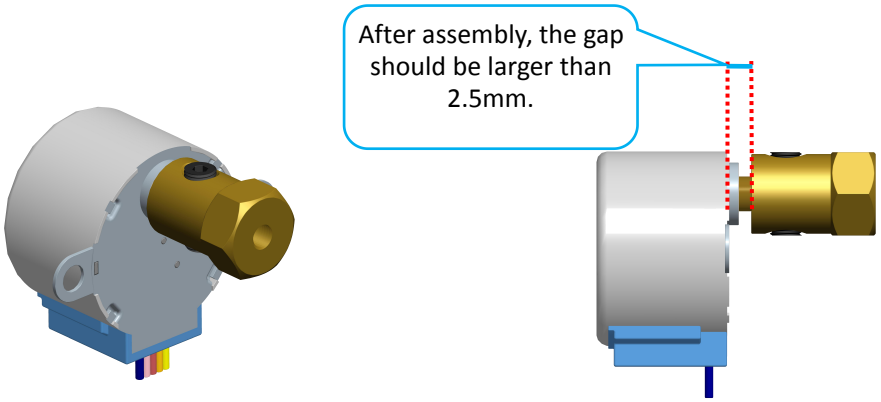


J. Fasten the S12D5 Coupler and 28BYJ-48 Gear Motor.

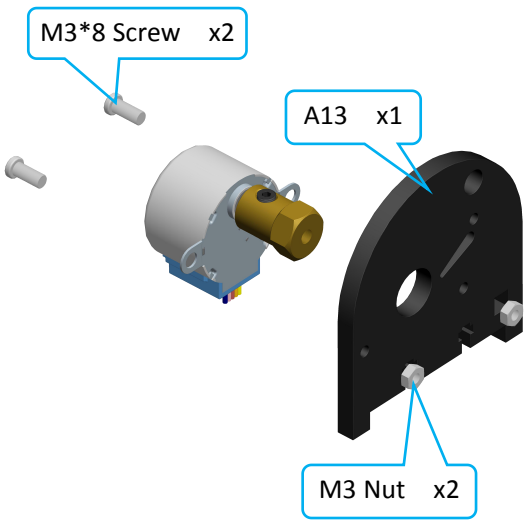
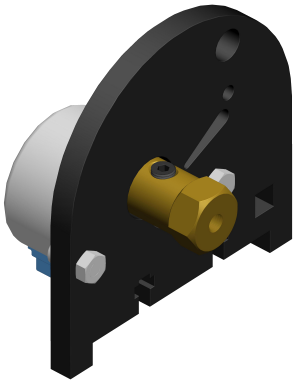
Assemble the following components



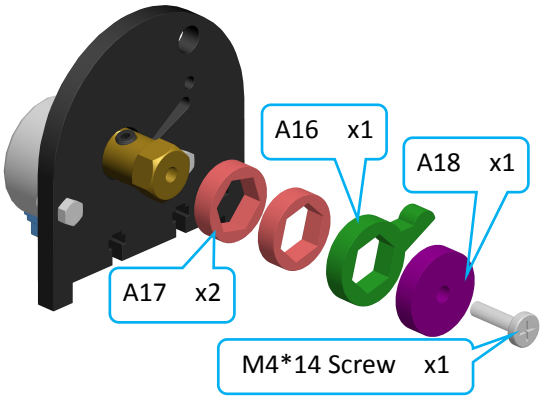
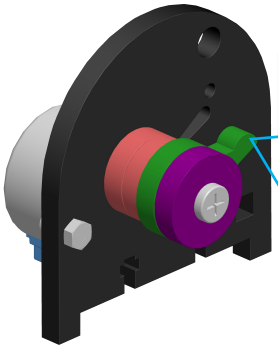
Effect diagram after assembling



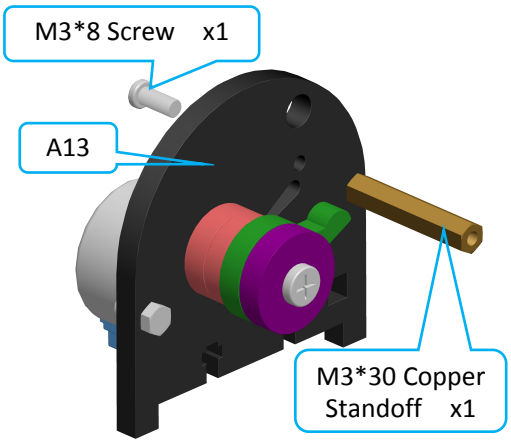
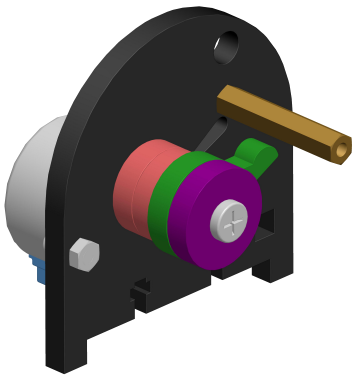
K. Install the 28BYJ-48 Gear Motor into the plate A13.

Assemble the following components	Effect diagram after assembling
	 Pay attention to the relative position of 28BYJ-48 motor and plate A13 and do not assemble them wrong. The wiring part of the motor should be put downward.

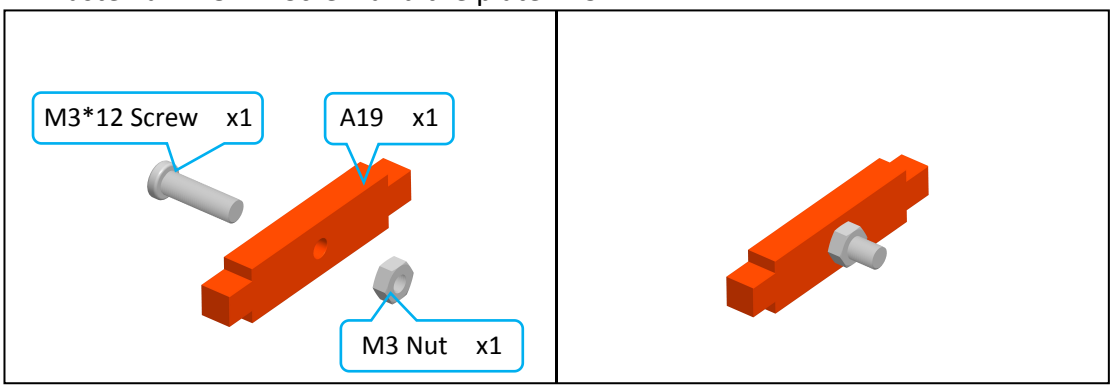
L. Wrap the S12D5 Coupler by A16.

Assemble the following components	Effect diagram after assembling
	 Pay attention to the direction of A16 and do not install the other way.

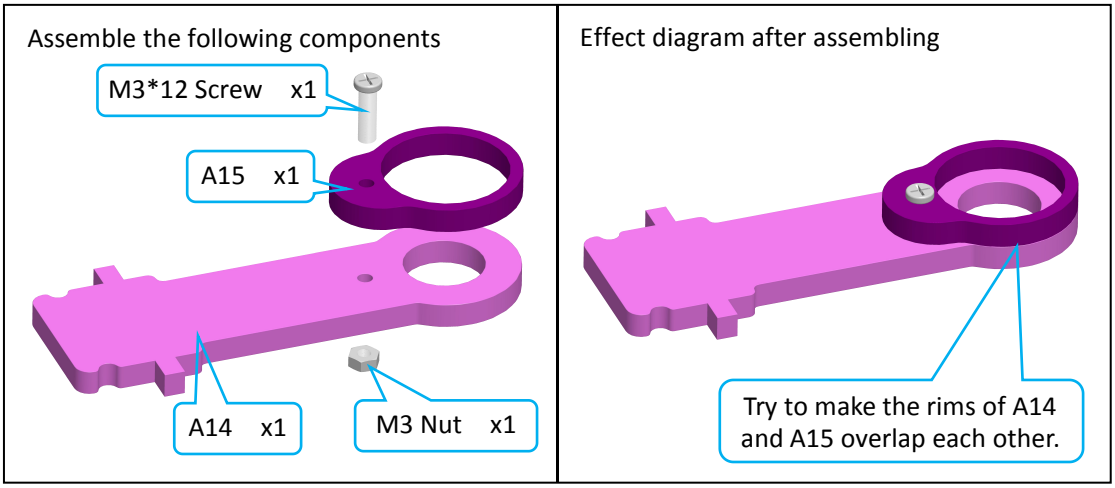
M. Fasten the M3*30 Copper Standoff onto the plate A13.

Assemble the following components	Effect diagram after assembling
	

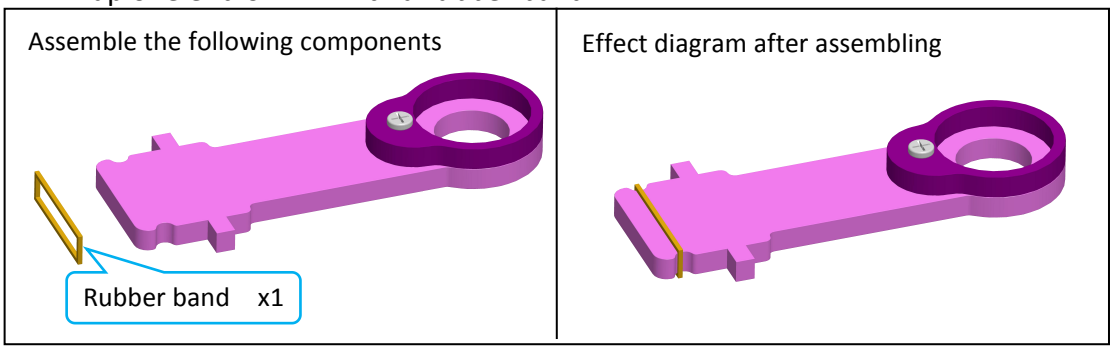
N. Fasten an M3*12 Screw and the plate A19.



O. Fasten A14 and A15.

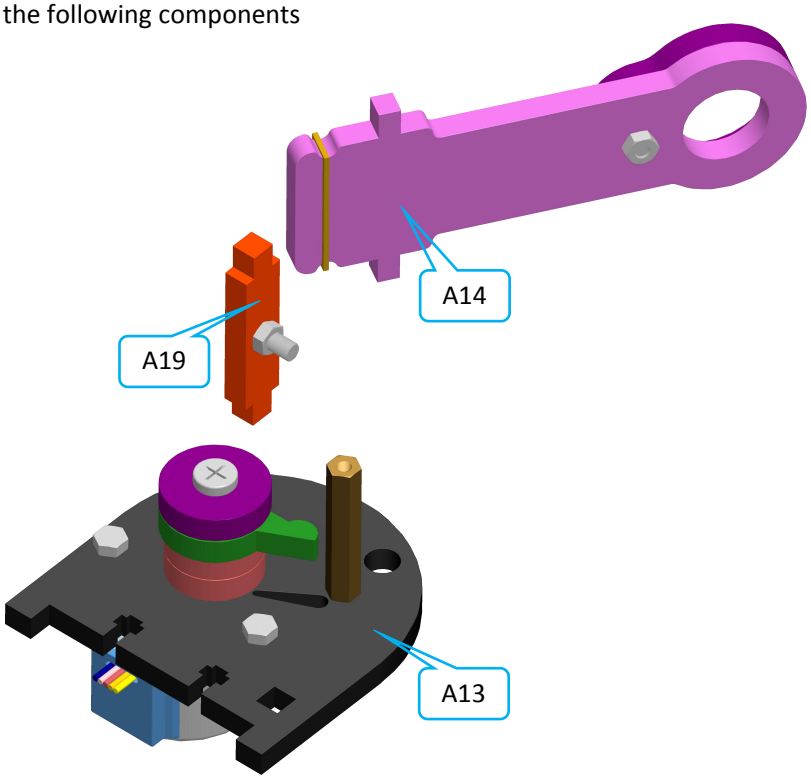


P. Wrap one end of A14 with a rubber band.

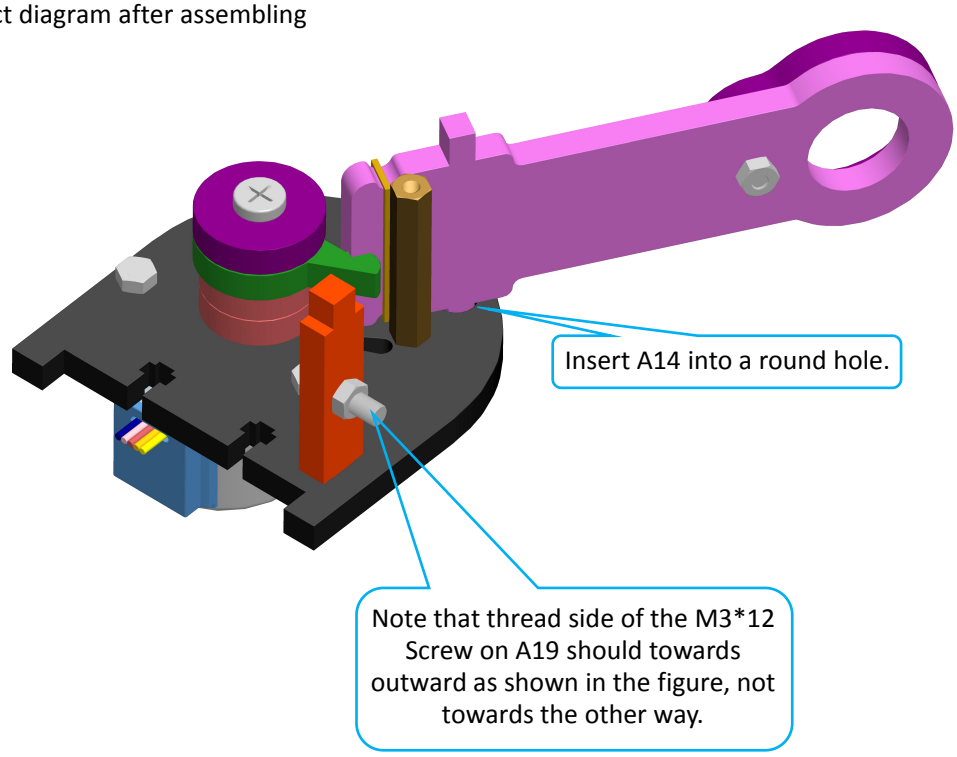


Q. Insert the bulging of A14 and A19 into the hole on A13.

Assemble the following components

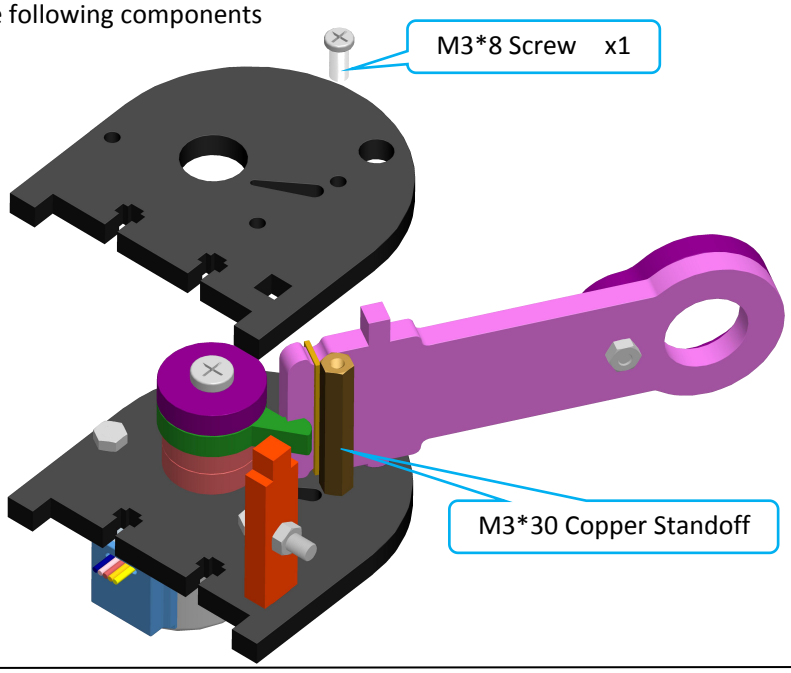


Effect diagram after assembling

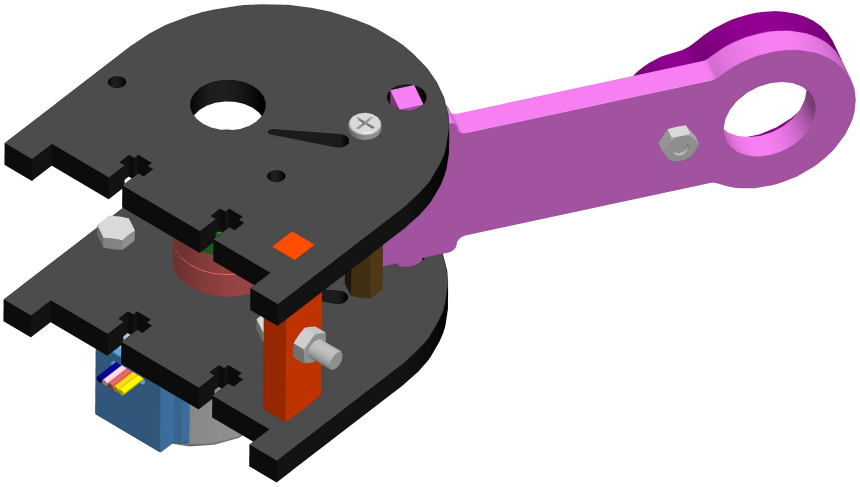


R. Take the other A13 and fasten it with an M3*30 Copper Standoff.

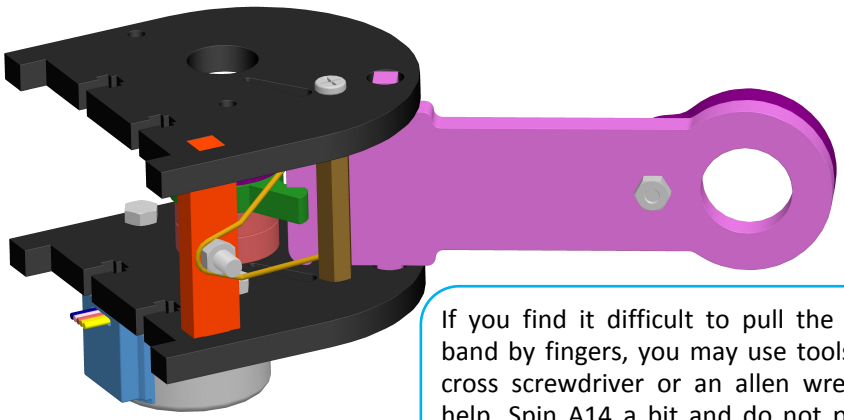
Assemble the following components



Effect diagram after assembling



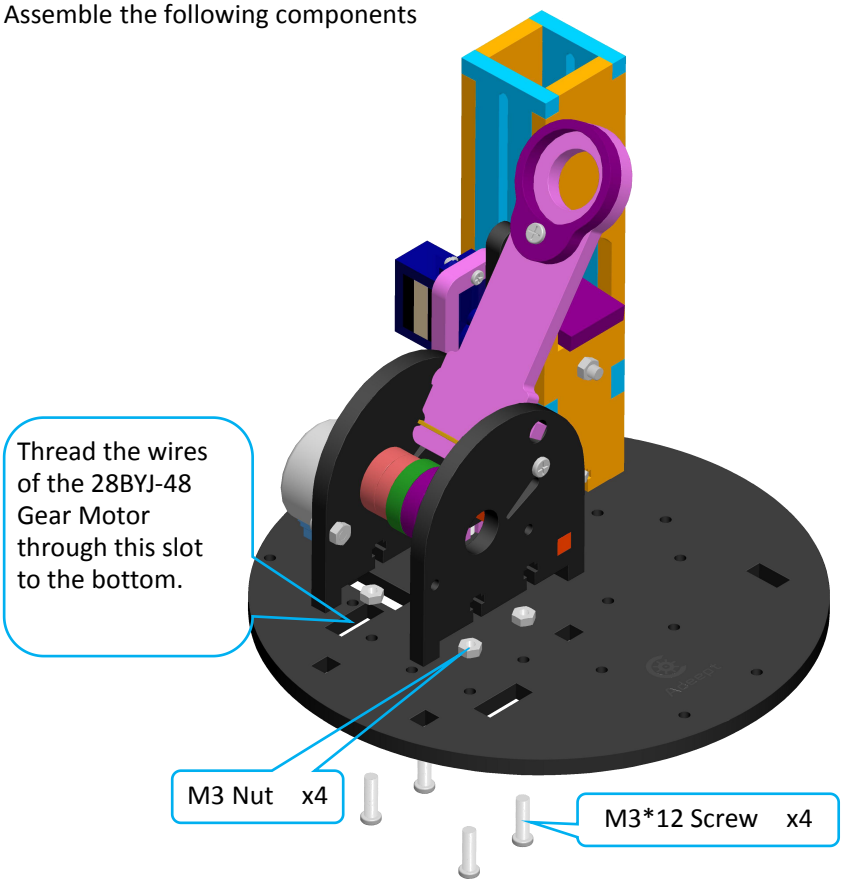
Pull the rubber band on A14 to wrap the M3*12 Screw on A19.



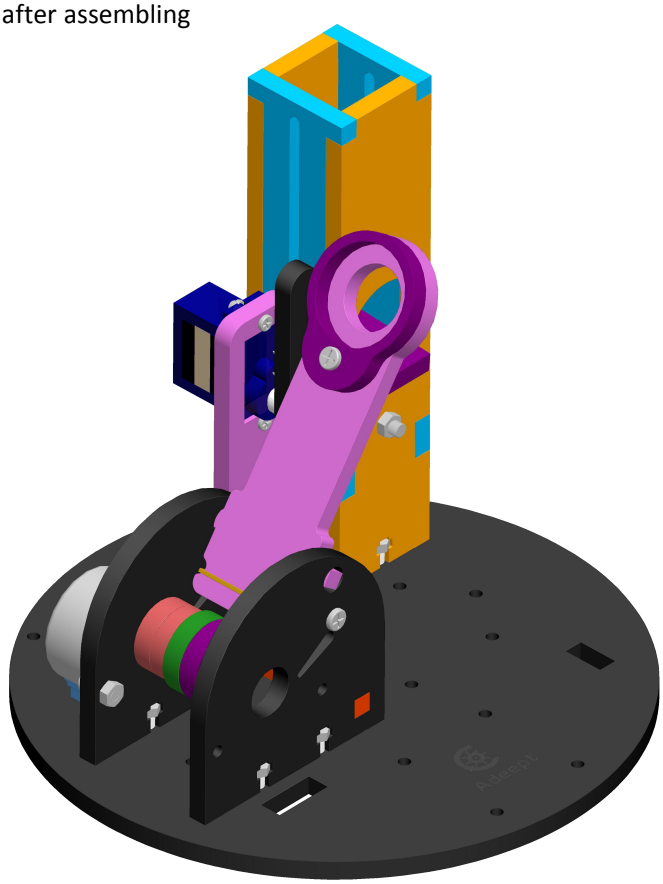
If you find it difficult to pull the rubber band by fingers, you may use tools like a cross screwdriver or an allen wrench to help. Spin A14 a bit and do not pull the rubber too hard, otherwise the 28BYJ-48 Gear Motor cannot pull A14.

S. Fasten 2 A13 plates onto A01.

Assemble the following components

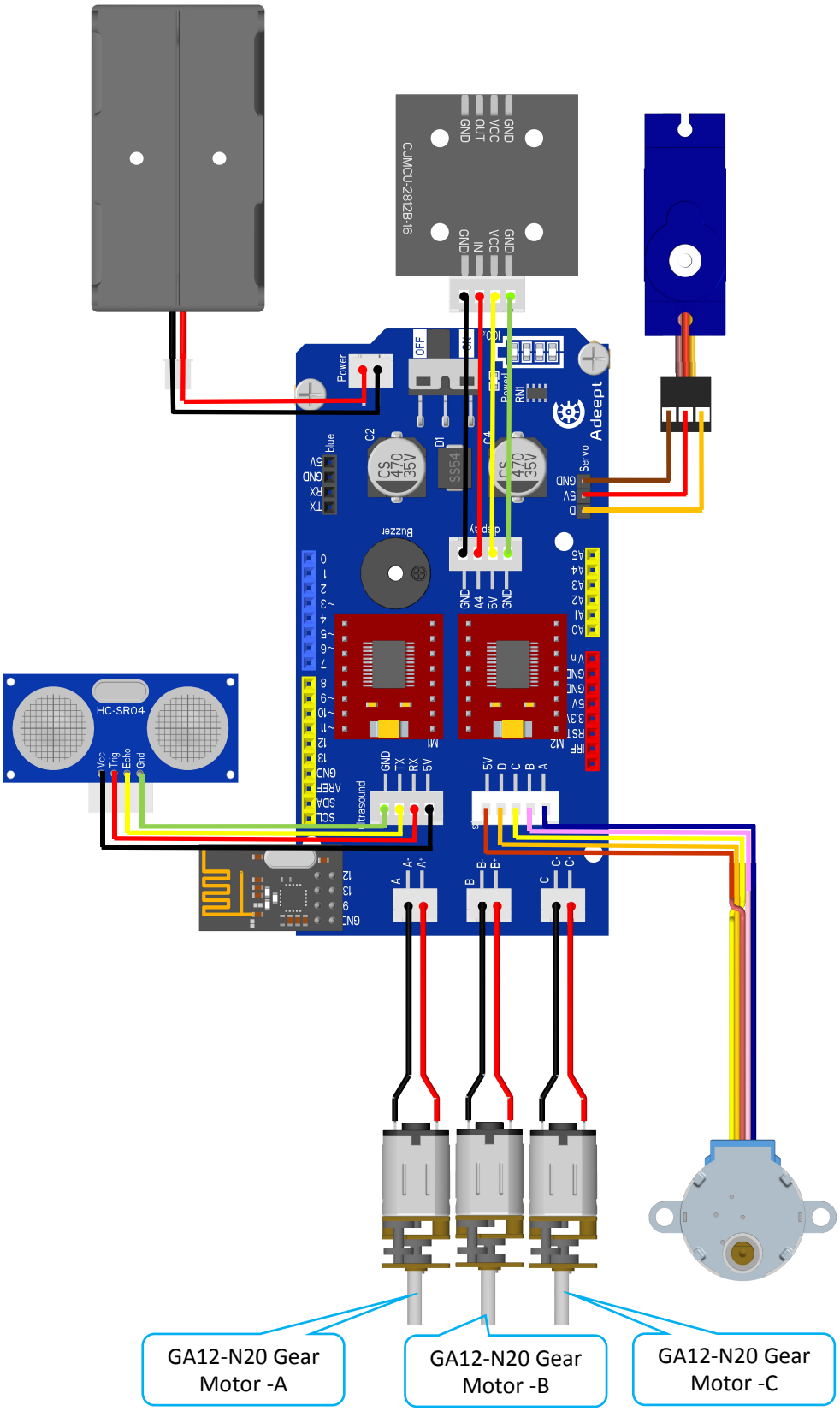


Effect diagram after assembling



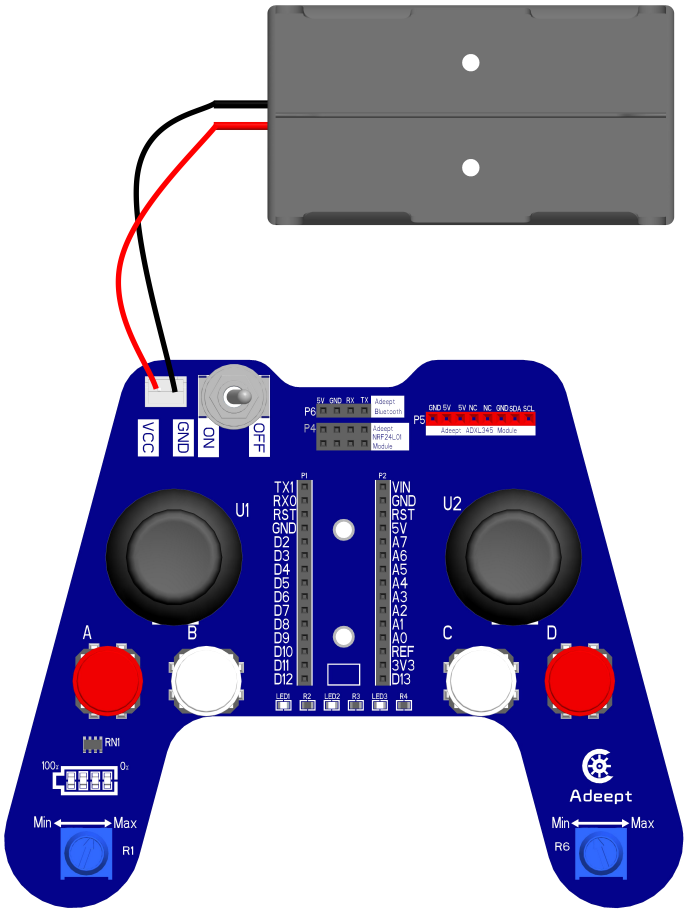
Circuit Connection

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



Connect the circuit of the car as shown above.

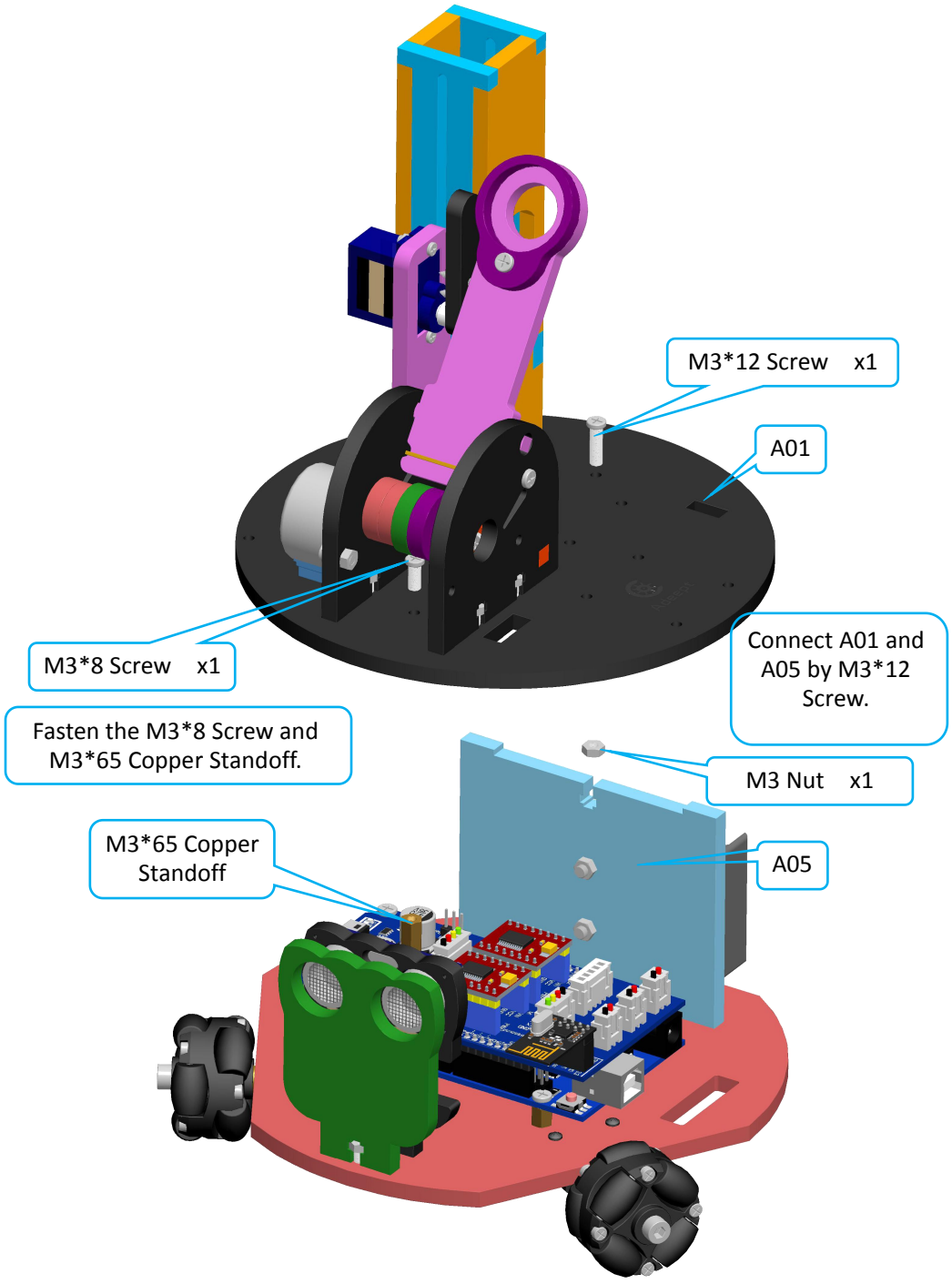
Circuit Connection of the Remote Control：



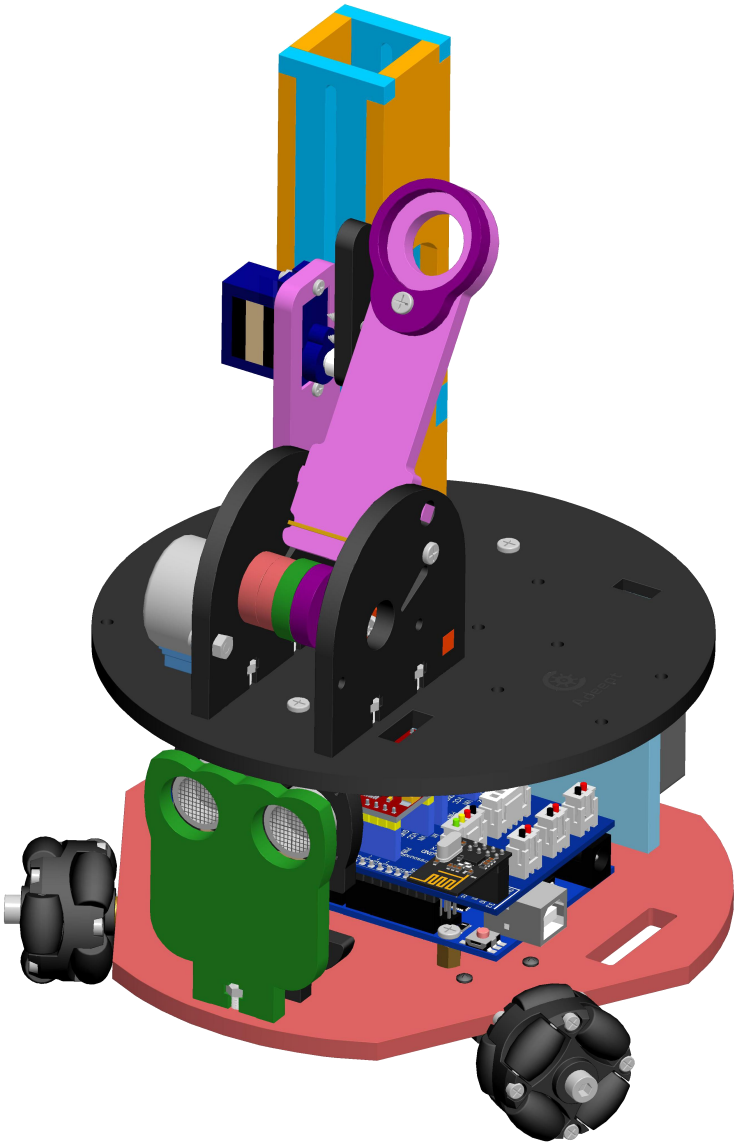
Assemble Bottom and Top

A. Assemble the bottom and top parts.

Assemble the following components



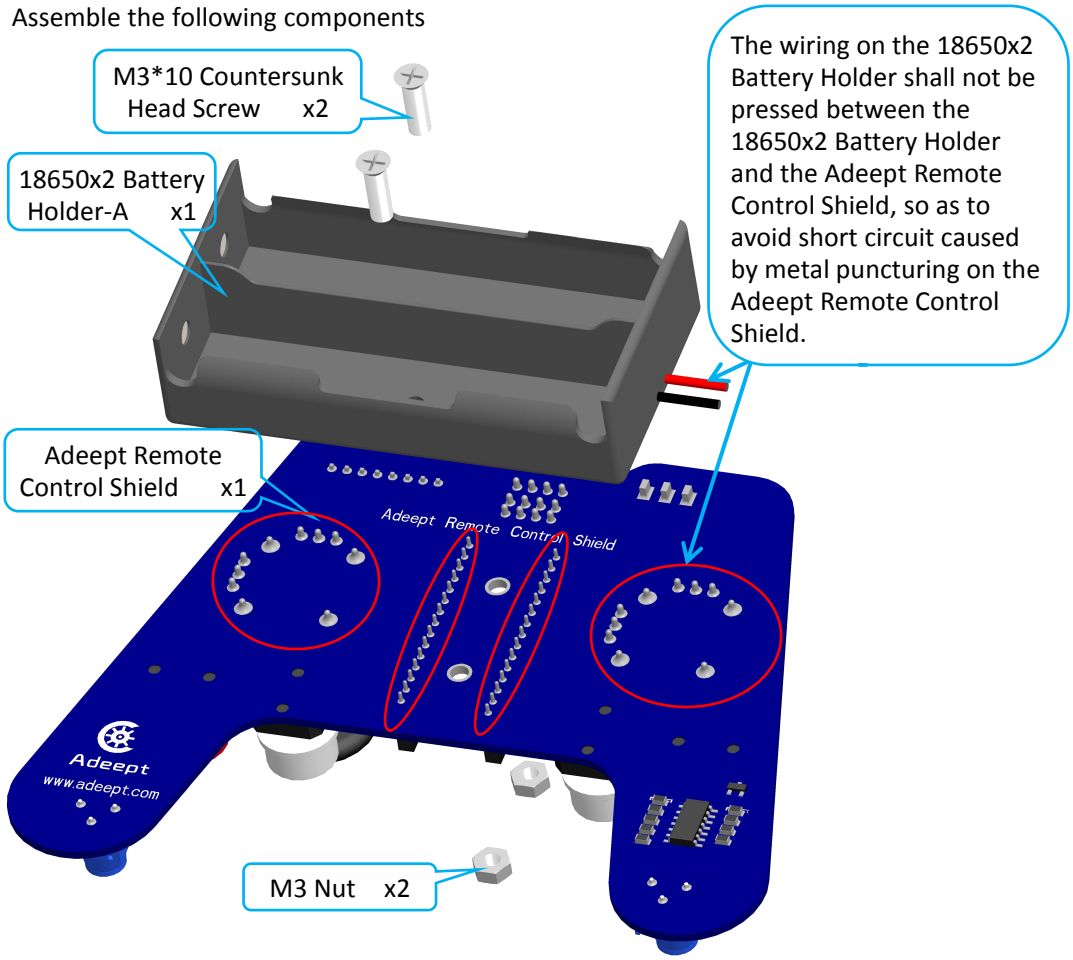
Effect diagram after assembling



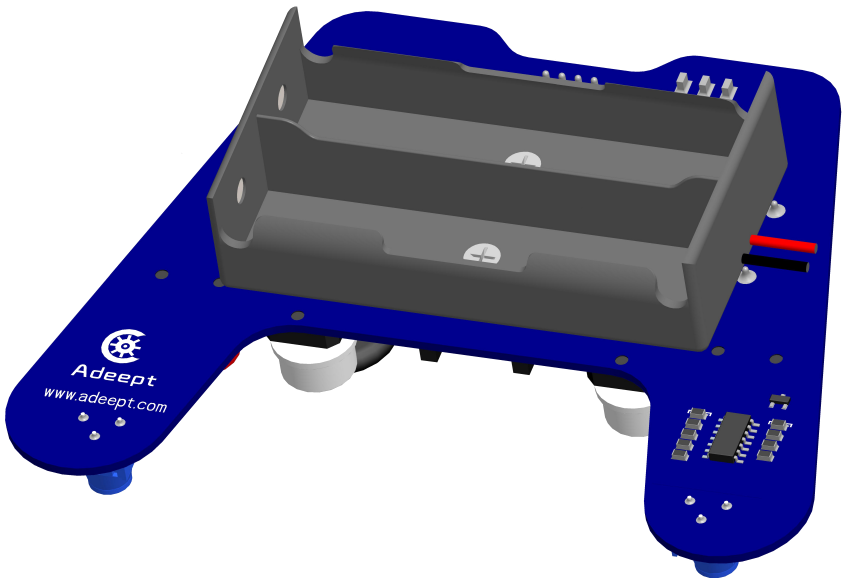
Remote Control

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

Assemble the following components

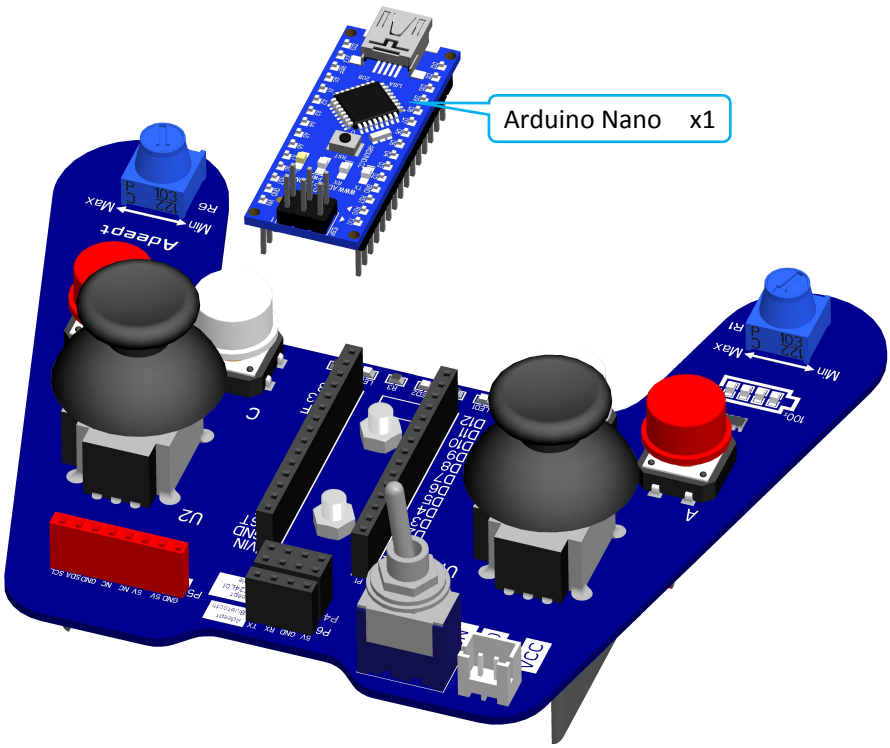


Effect diagram after assembling

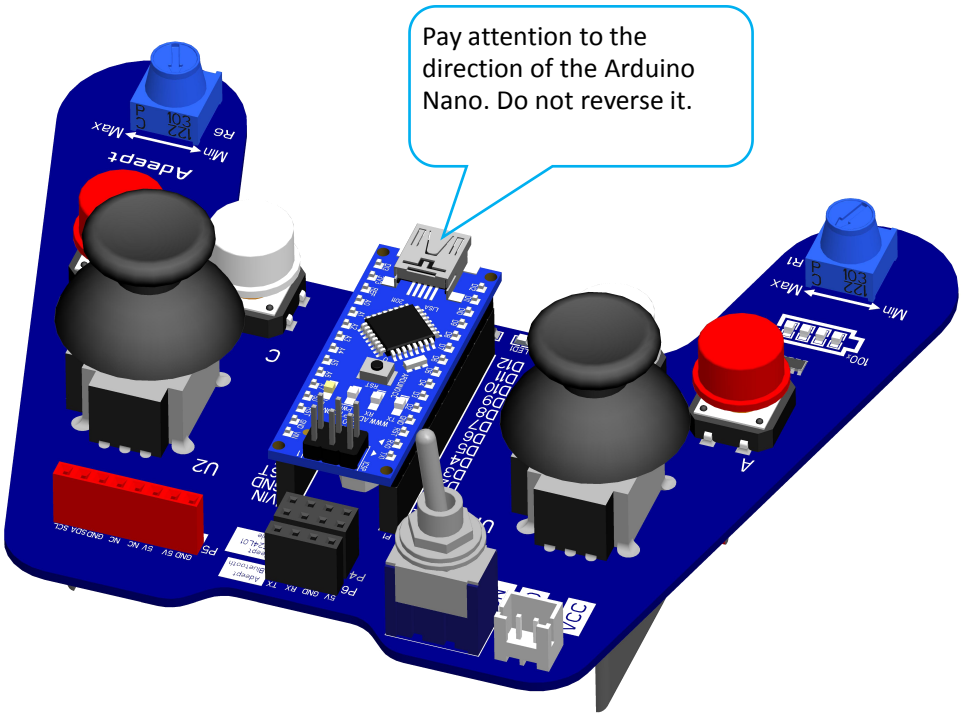


B. Plug Arduino Nano in Adeept Remote Control Shield.

Assemble the following components

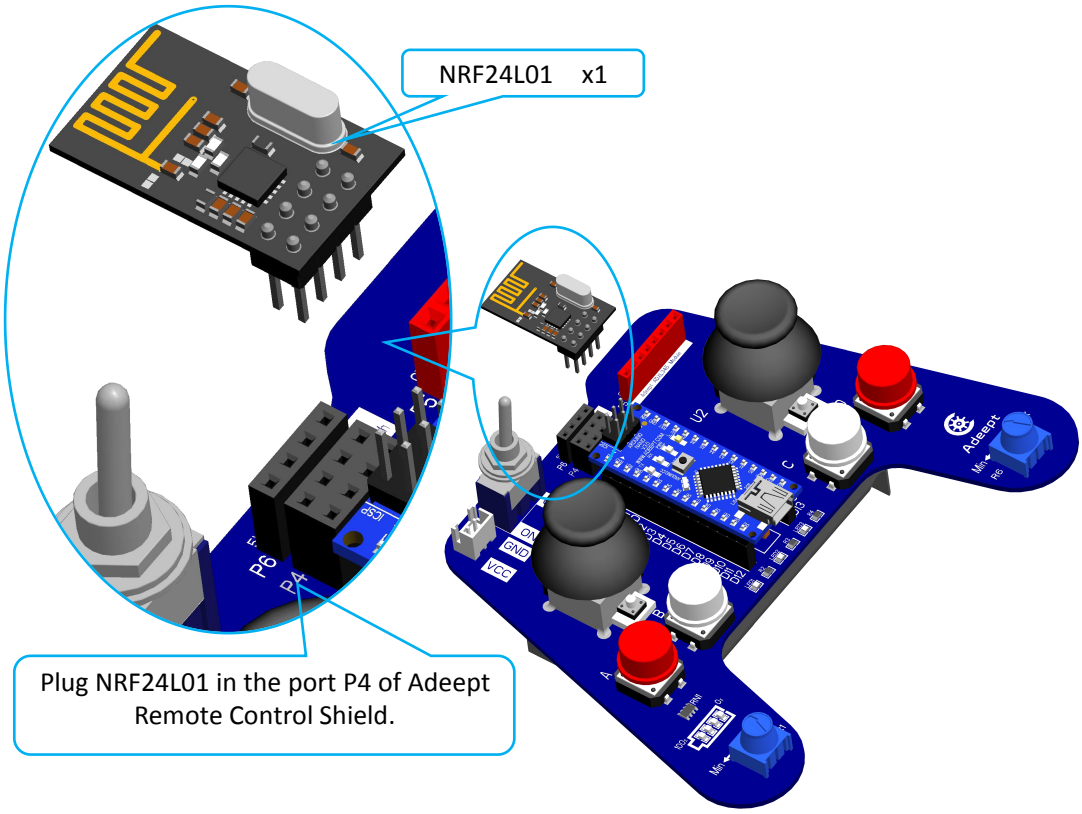


Effect diagram after assembling

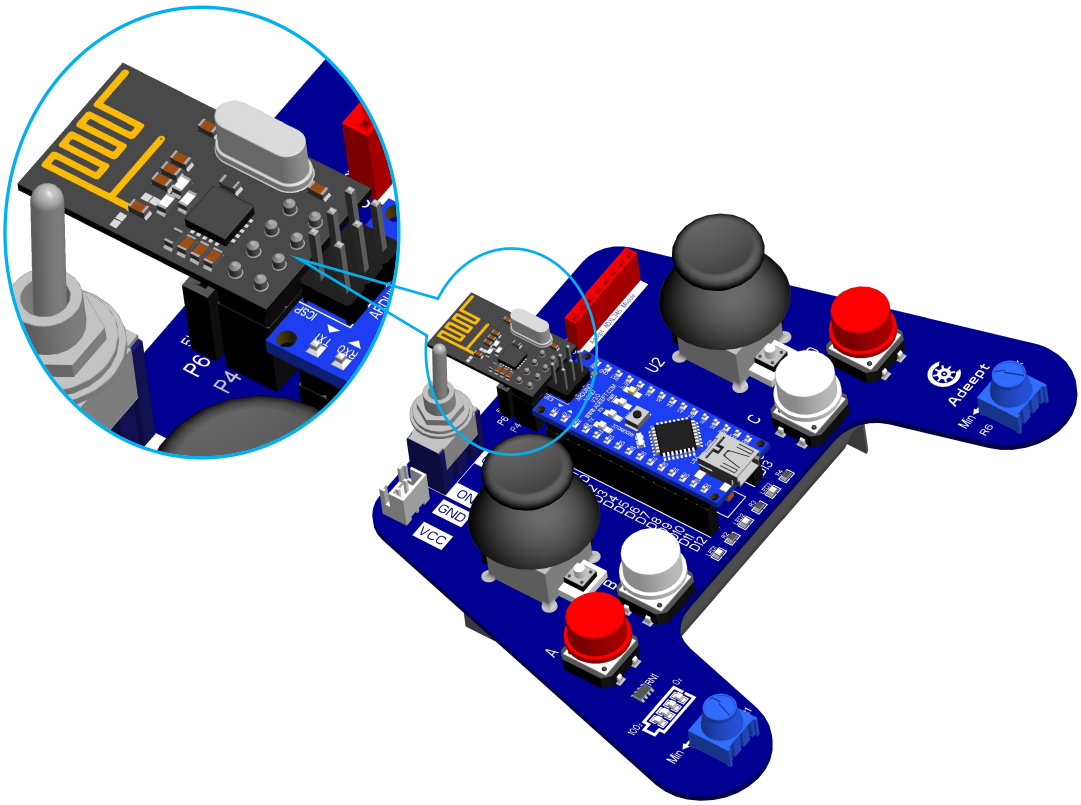


C. Plug NRF24L01 in Adeept Remote Control Shield.

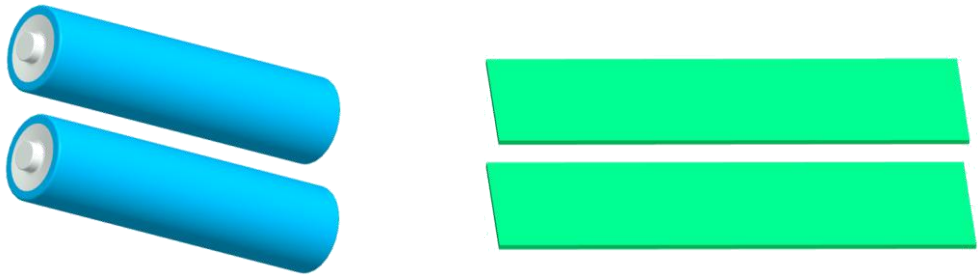
Assemble the following components



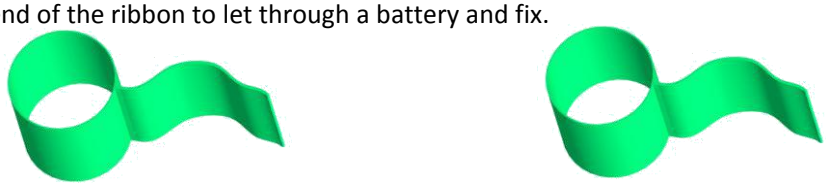
Effect diagram after assembling



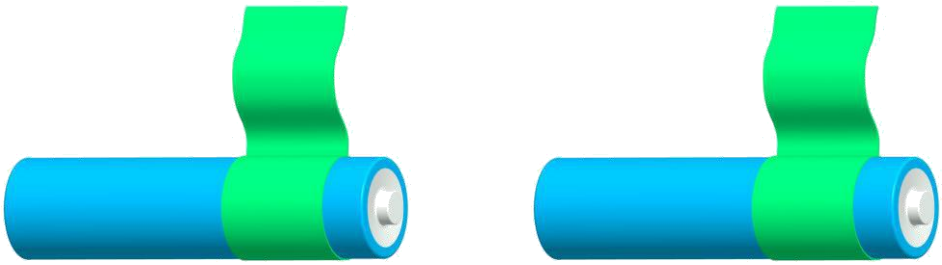
Install and Remove Batteries



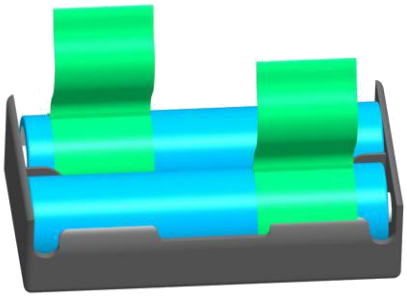
Roll one end of the ribbon to let through a battery and fix.



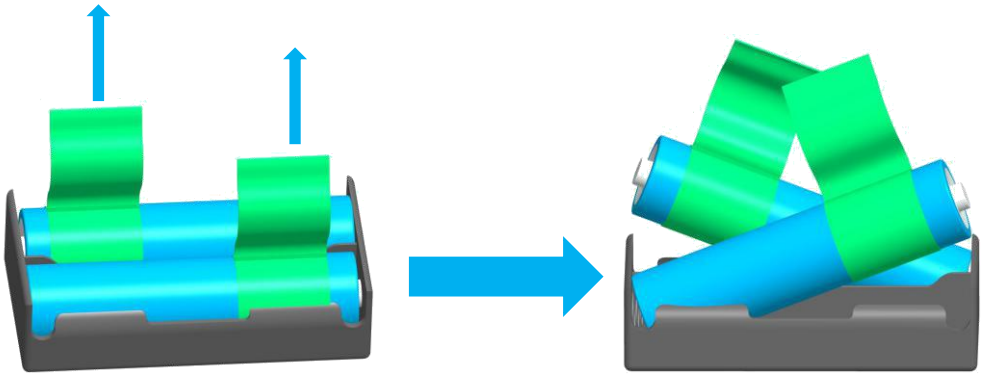
Insert the batteries into the rings - ribbon closer to the anode.

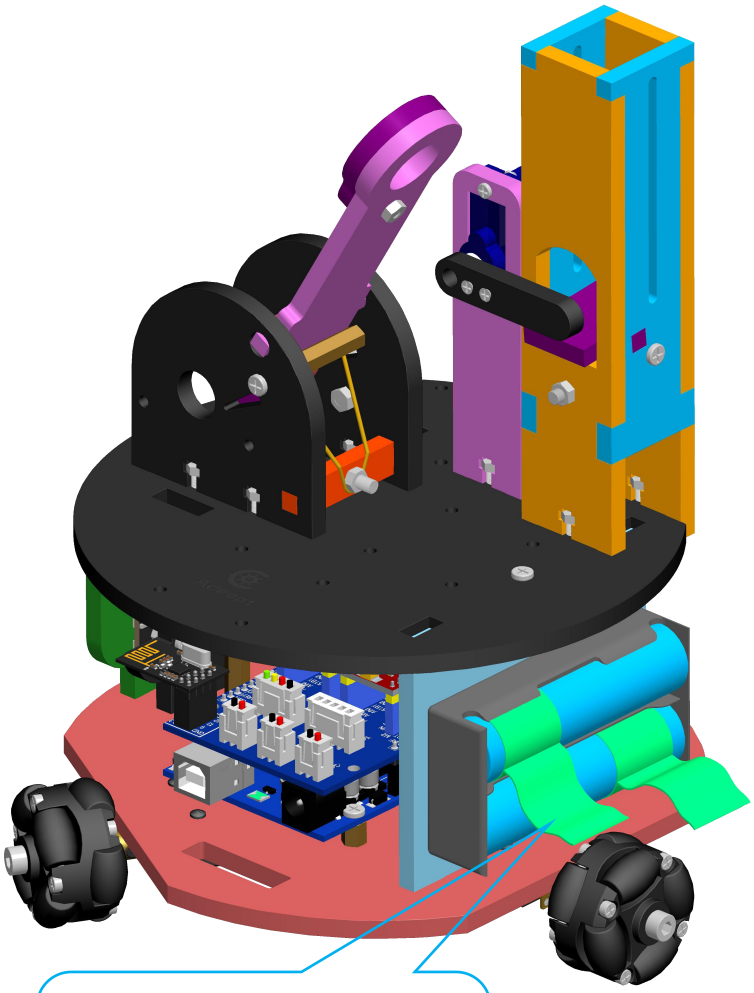


Install the batteries into the holder based on the pole.



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

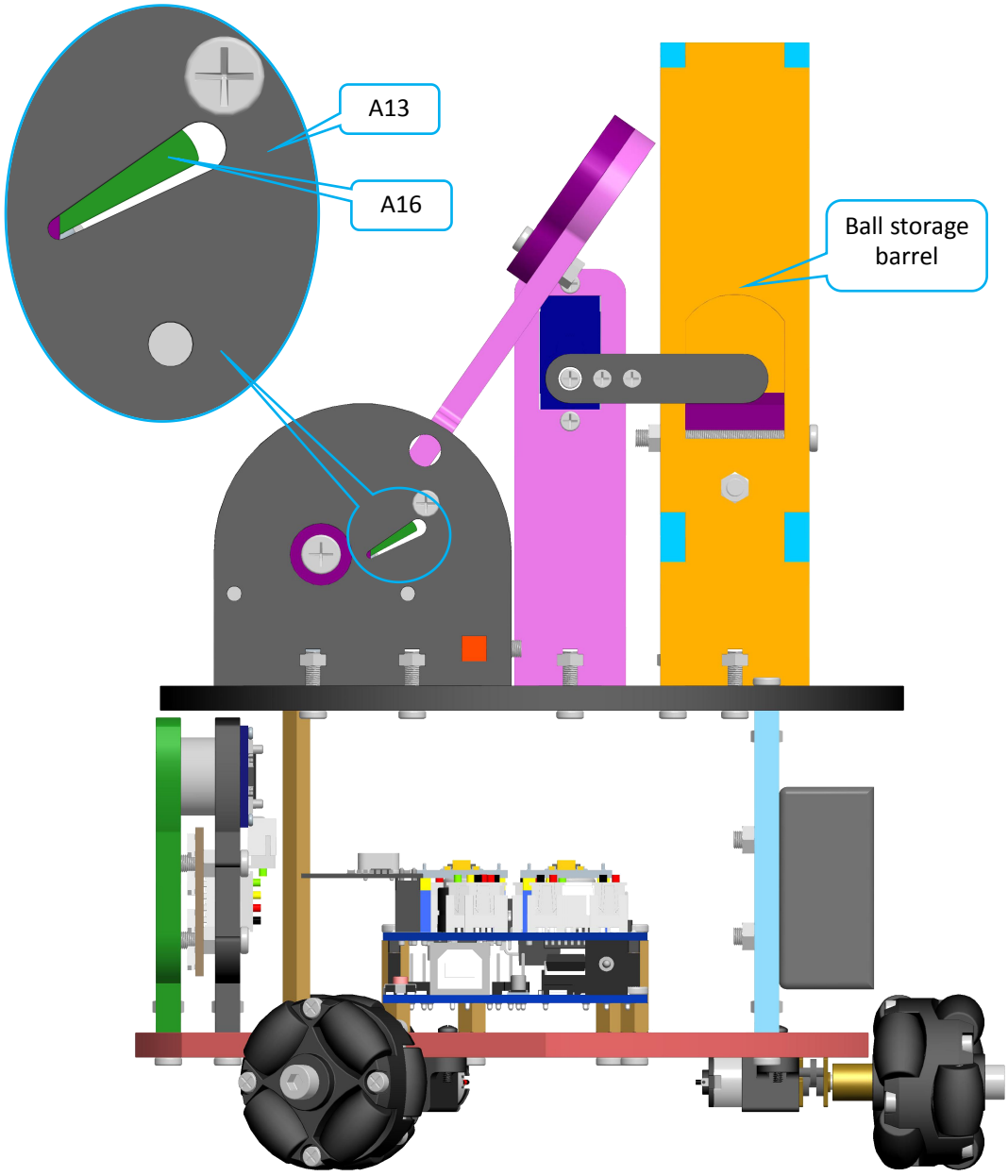



Note: Do not leave an overlong ribbon; otherwise it may get winded on the wheel shaft, causing battery to be pulled out or motor burnt.

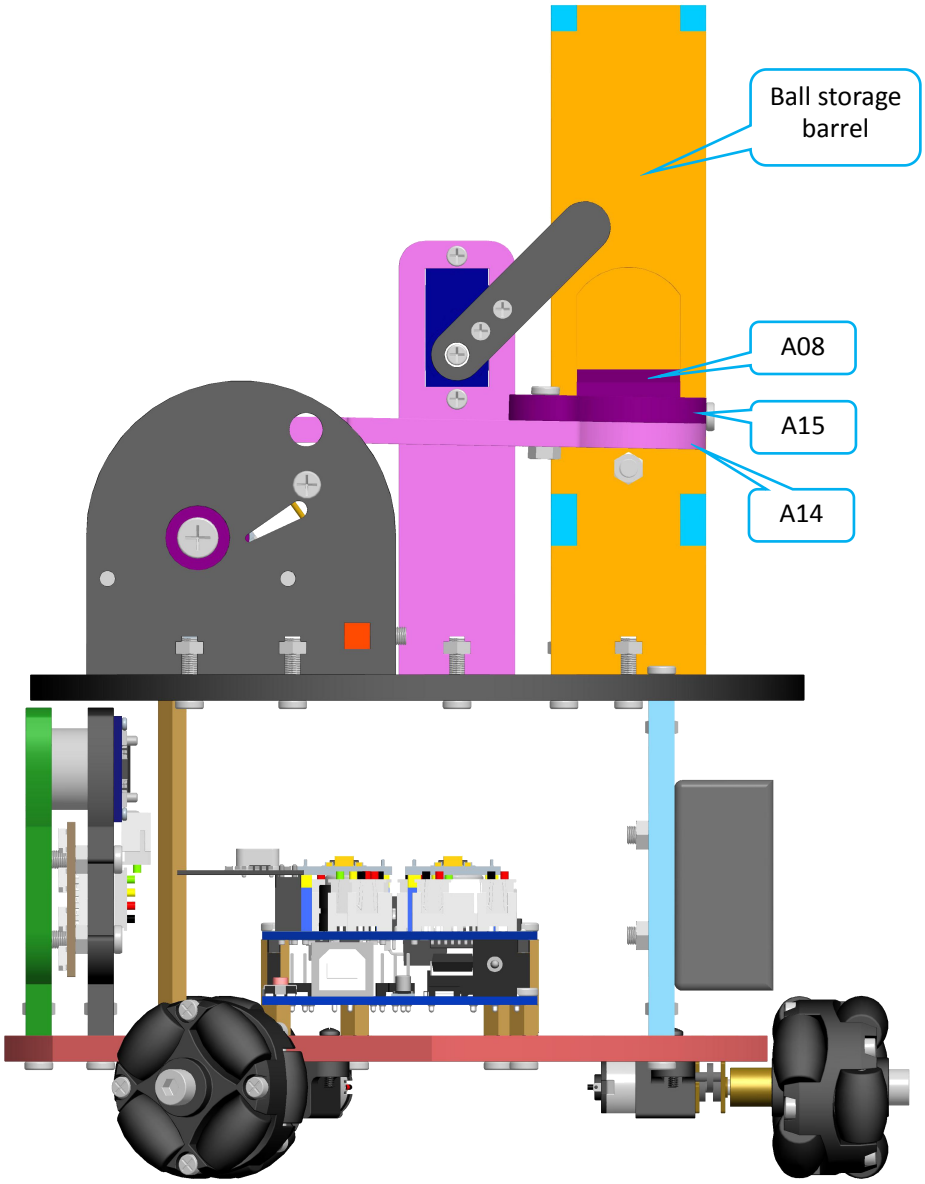
Adjuh 28BYJ-48 Gear Motor

A. Connect the car to power.

Power the car, control A16 to spin to the position as shown below by the remote, i.e. the edge of A16 to be on the hole marked on A13 in the figure.



B. Execute an ejection process.



As shown in the figure, there'll be a pause before the A14 is about to raise. The servo spins counterclockwise (to place the ball). Now A15 should be lower than A08; otherwise the ball will be stuck and cannot roll into the round hole on A15. If you find A15 is higher than A08 now, after an ejection, control the 28BYJ-48 Gear Motor to make A14 spin clockwise a bit via the remote. Repeat the ejection process. If the A14 is pulled back (its position in the previous step) by the rubber band before the rocker arm is raised, you need to adjust the motor with the remote, to make it spin 355° counterclockwise. During the adjustment, if you cannot see the position of A14 clearly, put 3 balls into the barrel and check whether they can be ejected successfully. After the adjustment, you may put the balls in and control the car wireless.
Note: 1-3 balls packaged can be put into the barrel one time. The number and weight of the balls will influence the speed of the ball at the bottom to roll out. If you use other balls, you need to change the program of putting in a ball.

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.

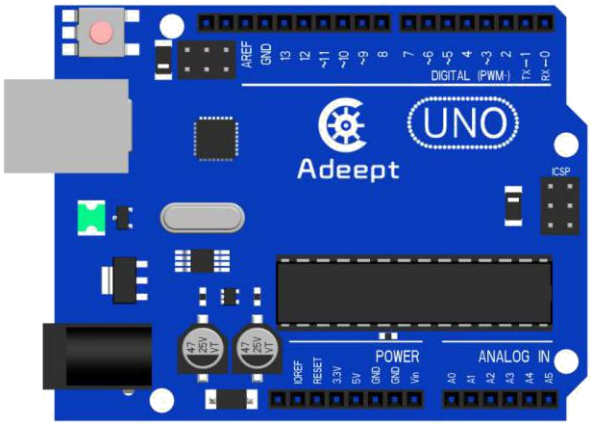
Why Arduino?

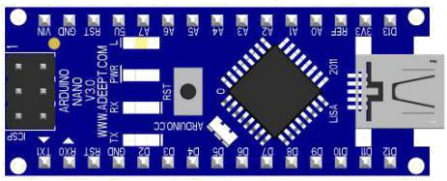
Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost 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 would be a prefect 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 is it 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:

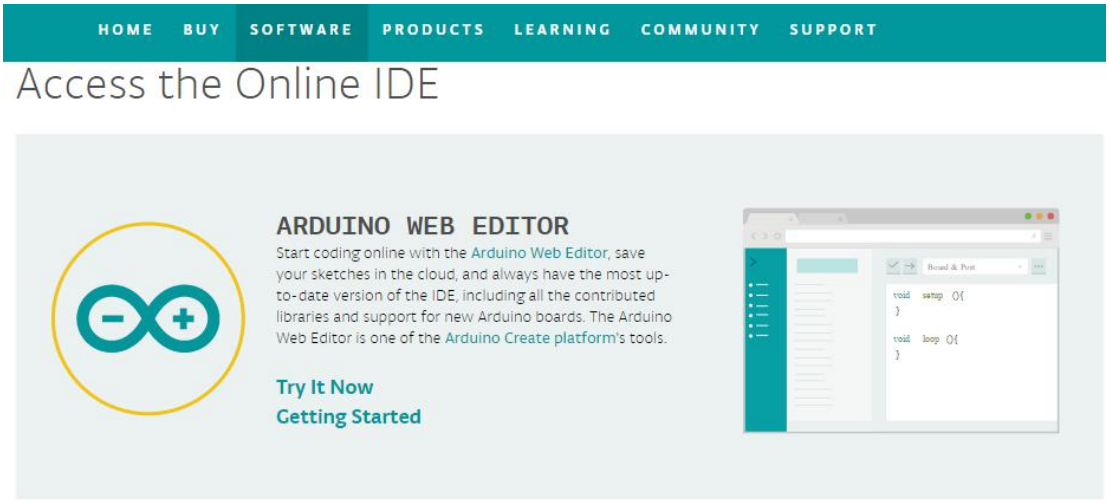
AREF. Reference voltage for the analog inputs. Used with analogReference().

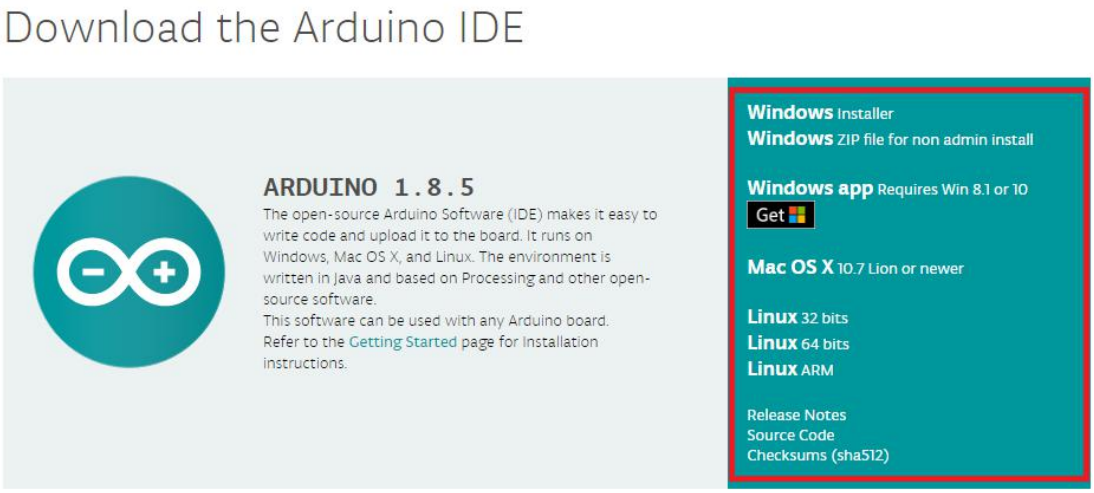
Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Arduino Software (IDE)

For more detailed information about Arduino IDE, please refer to the following
https://youtu.be/BsTDVB8B240

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 Intaller” to download and install the driver correctly.

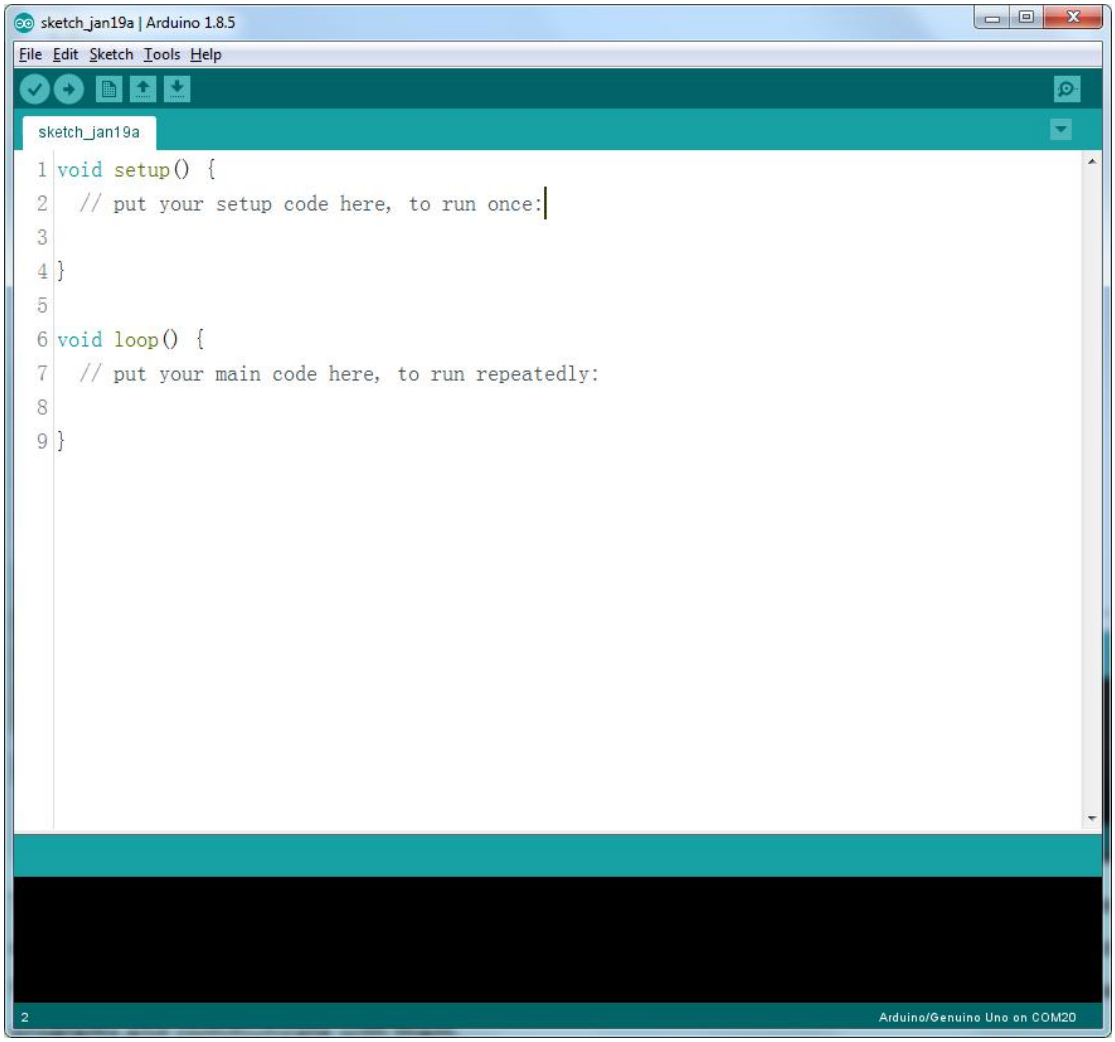




After the download completes, run the installer. For Windows users, there may pop up an installation dialog box of the driver during the installation. Please agree the installation when it appears.

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.

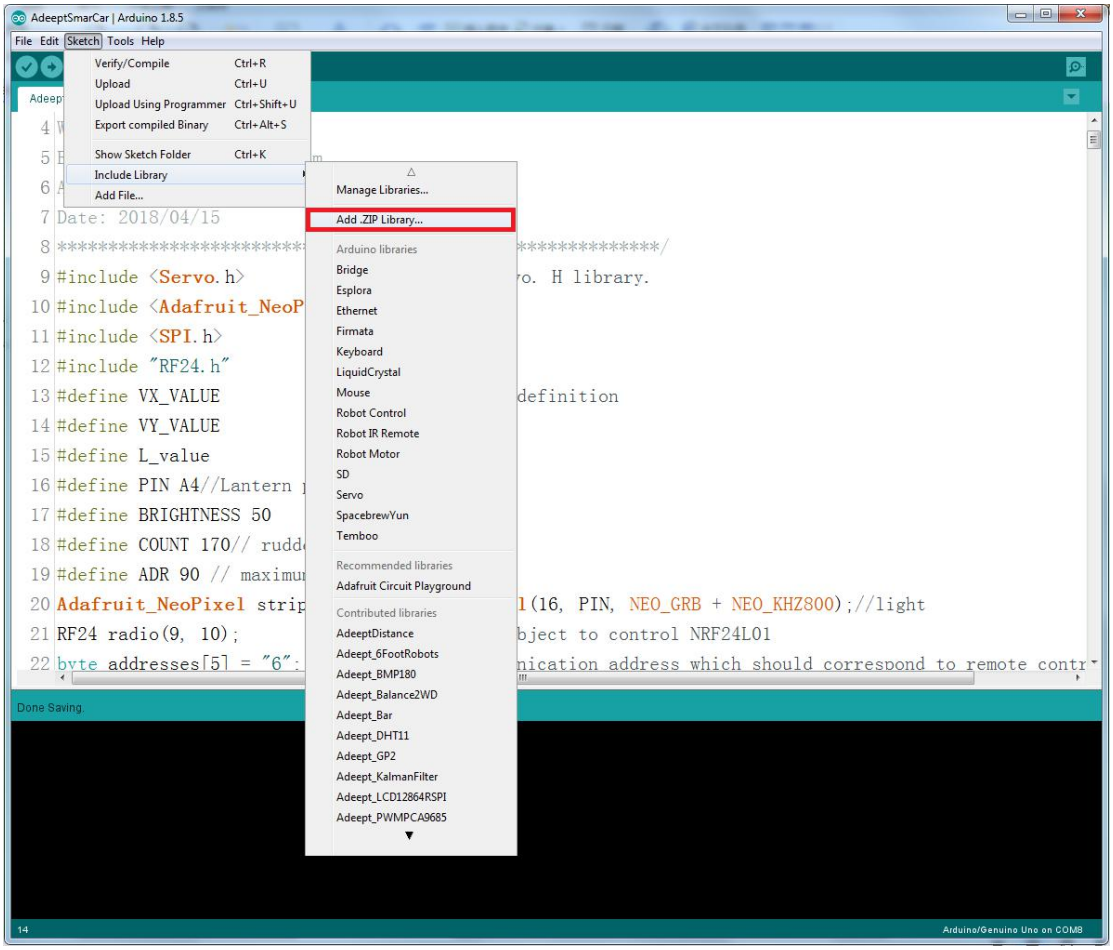

Save: Saves your sketch.


Serial Monitor: Opens the serial monitor.

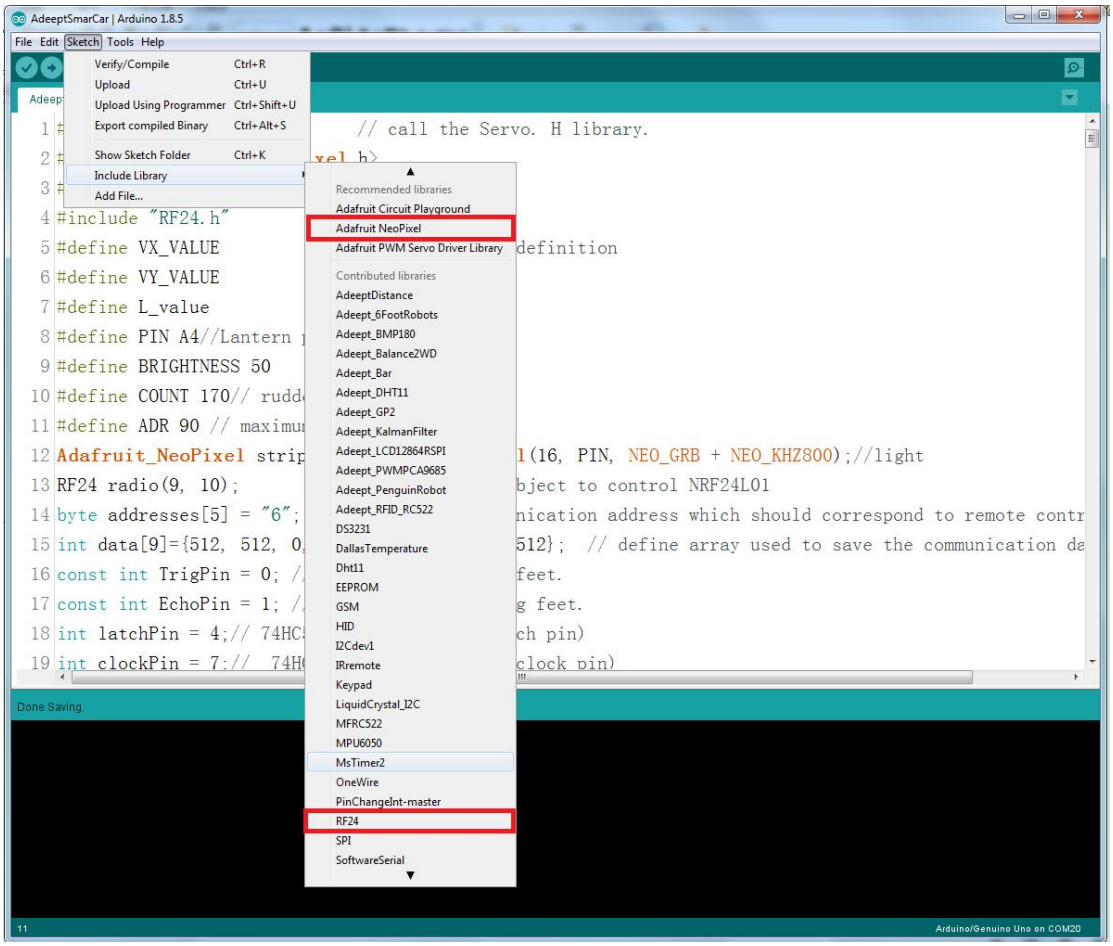
Additional commands are found within the five menus: **File**, **Edit**, **Sketch**, **Tools**, and **Help**. The menus are context sensitive, which means only those items relevant to the work currently being carried out are available.
Since version 1.0, files are saved with an **.ino** file extension. 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 Adafruit_NeoPixel.ZIP* library, so you need to install it before compiling. Click **Add.ZIP Library** to add the *RF24.ZIP* and *Adafruit_NeoPixel.ZIP* to the *libraries* folder.



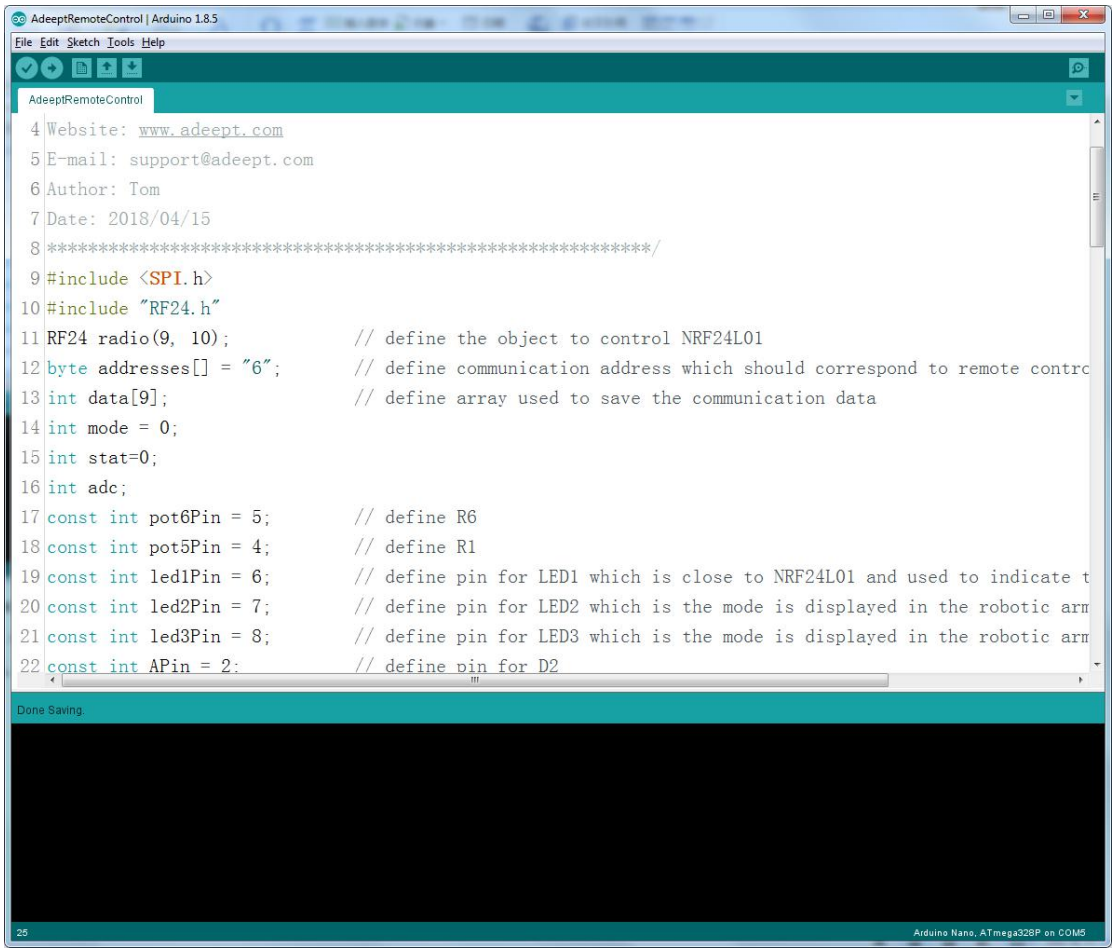
After the library is installed successfully, you can find the RF24.ZIP and *Adafruit_NeoPixel.ZIP*


AdeeptSmarCar | Arduino 1.8.5
File Edit Sketch Tools Help
Verify/Compile Ctrl+R
Upload Ctrl+U
Upload Using Programmer Ctrl+Shift+U
Export compiled Binary Ctrl+Alt+S
Show Sketch Folder Ctrl+K
Include Library
Add File...
Recommended libraries
Adafruit Circuit Playground
Adafruit NeoPixel
Adafruit PWM Servo Driver Library
Contributed libraries
AdeeptDistance
Adeept_6FootRobots
Adeept_BMP180
Adeept_Balance2WD
Adeept_Bar
Adeept_DHT11
Adeept_GP2
Adeept_KalmanFilter
Adeept_LCD12864RSPI
Adeept_PWMPCA9685
Adeept_PenguinRobot
Adeept_RFID_RC522
DS3231
DallasTemperature
Dht11
EEPROM
GSM
HID
I2Cdev1
IRremote
Keypad
LiquidCrystal_I2C
MFRC522
MPU6050
MsTimer2
OneWire
PinChangeInt-master
RF24
SPI
SoftwareSerial
// call the Servo. H library.
#include "RF24.h"
#define VX_VALUE
#define VY_VALUE
#define L_value
#define PIN A4//Lantern
#define BRIGHTNESS 50
#define COUNT 170// rudd
#define ADR 90 // maximu
Adafruit_NeoPixel strip
RF24 radio(9, 10);
byte addresses[5] = "6";
int data[9]={512, 512, 0
const int TrigPin = 0; //
const int EchoPin = 1; //
int latchPin = 4;// 74HC
int clockPin = 7;// 74H
definition
l(16, PIN, NEO_GRB + NEO_KHZ800);//light
bject to control NRF24L01
nication address which should correspond to remote contr
512}; // define array used to save the communication da
feet.
g feet.
ch pin)
clock pin)
Done Saving.
11
Arduino/Genuino Uno on COM20

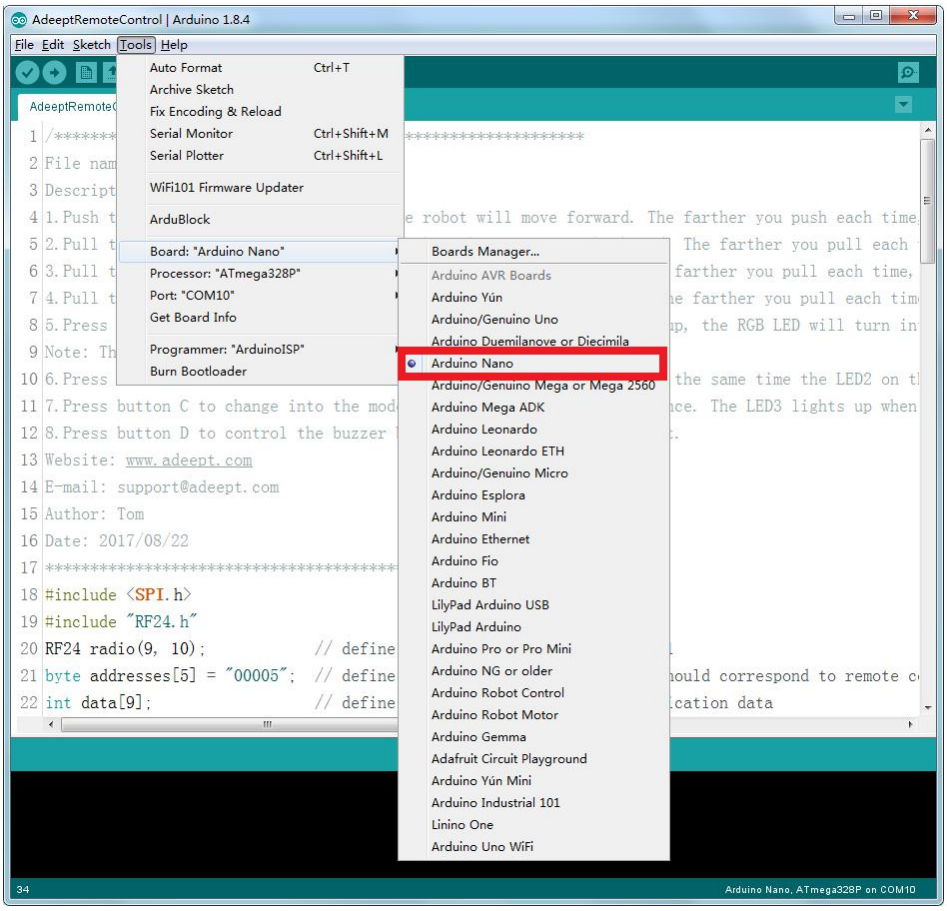
Upload Program

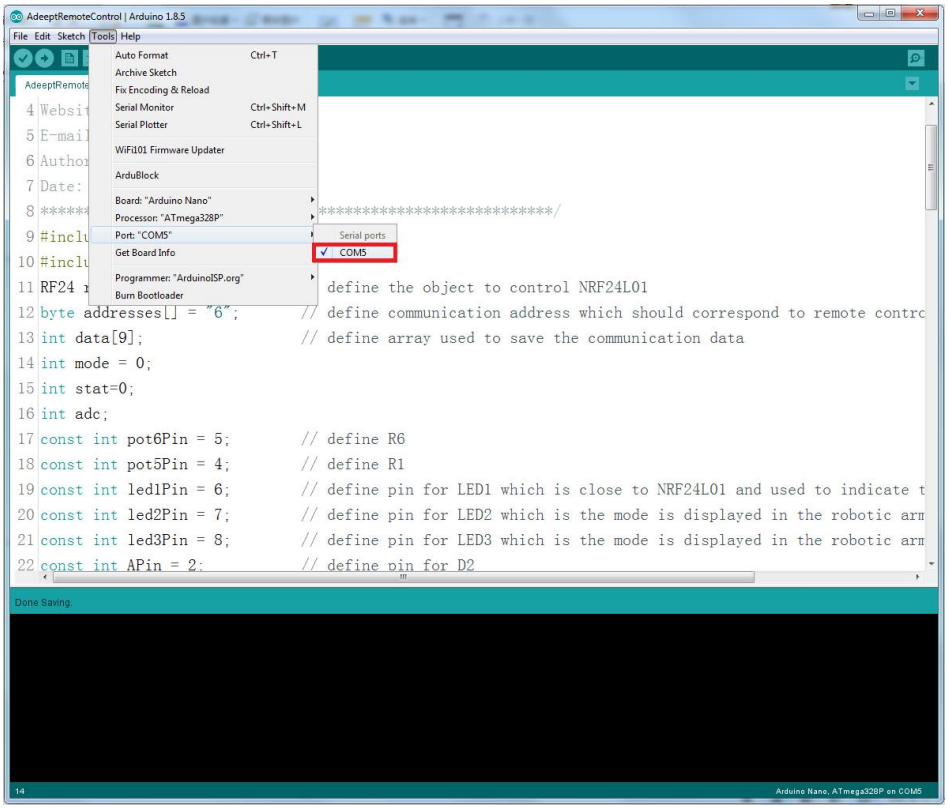
After the preparations above, next we will upload the program (example sketches provided) to the Arduino Nano and Adeept UNO R3 boards. The Car kit comprises of two parts: the remote control based on Arduino Nano and the car controller on Adeept UNO R3.

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

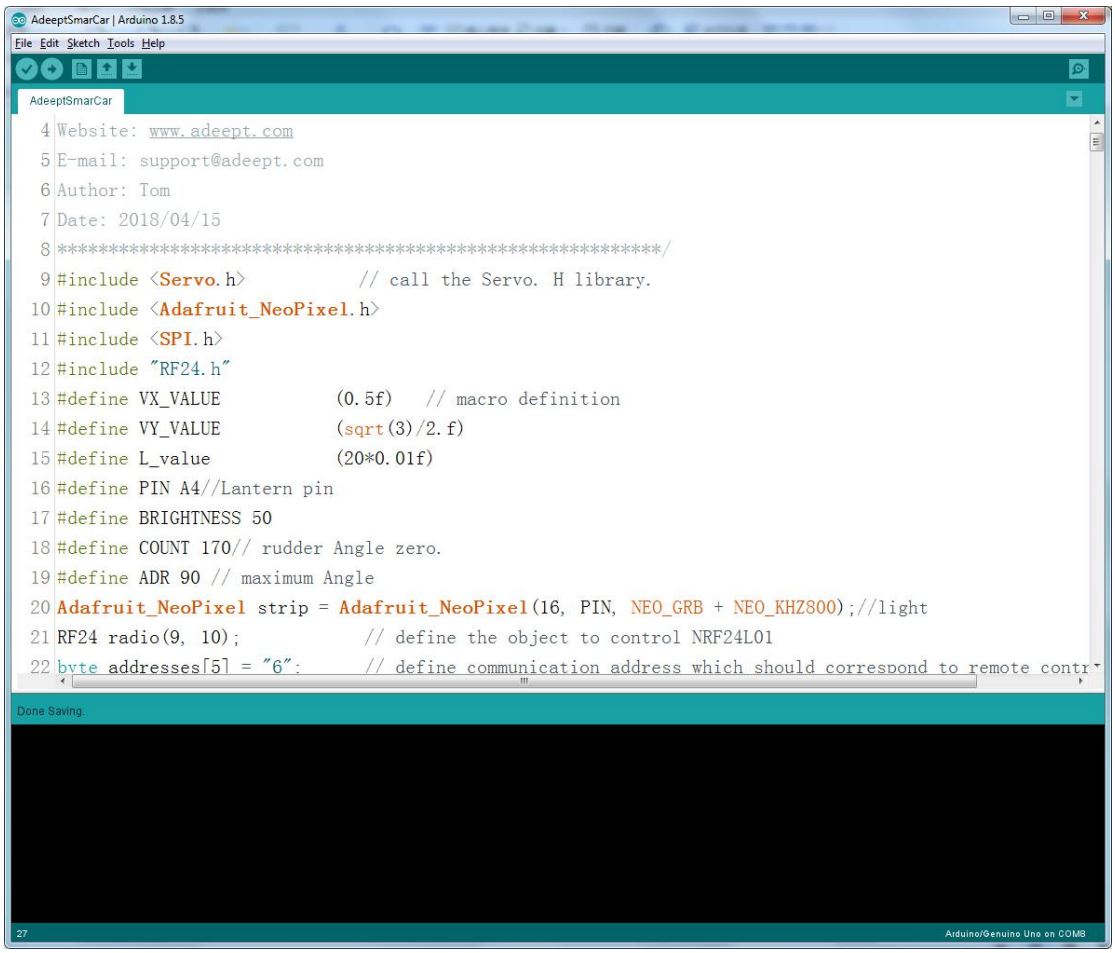


Connect the Arduino Nano to the computer. Select **Tool** -> **Board**: "**Arduino Nano**"-> **Arduino Nano**, and **Port** ->**COM5.** COMx is the port number assigned to the Arduino Nano and can be COM1, COM2, COM3...So it depends.

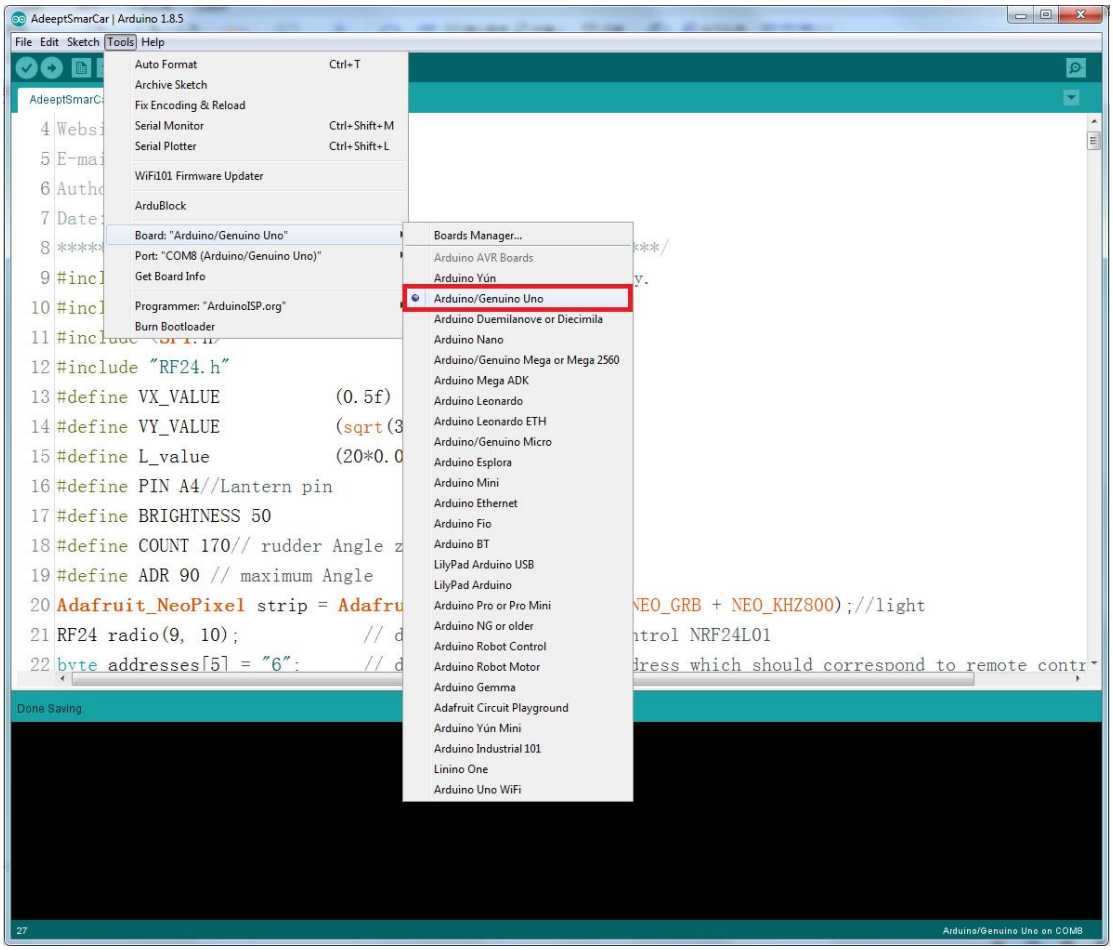


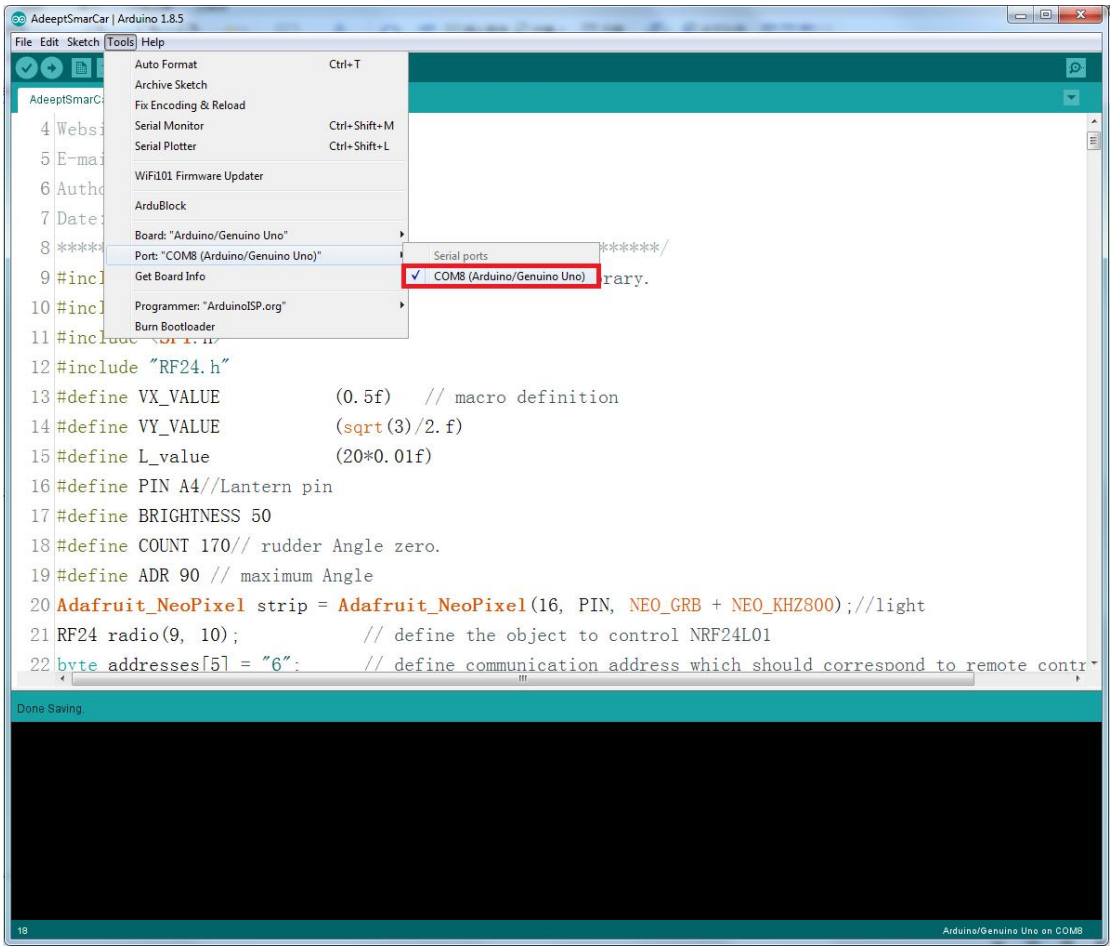


Next, click the upload button  . After the program is uploaded to the Nano successfully, upload another sketch to the Adeept UNO R3. Open the program provided for the control board, the file "AdeeptSmarCar.*ino*".



Connect the Arduino UNO R3 board to the PC. Select **Tool** -> **Board "Arduino/Genuino Uno"**, and **Port** - > **COM8**. Also here is COM8, assigned to the Uno, but it can be COM1, COM2, COM3...


AdeeptSmarCar | Arduino 1.8.5
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 Firmware Updater
ArduBlock
Board: "Arduino/Genuino Uno"
Port: "COM8 (Arduino/Genuino Uno)"
Get Board Info
Programmer: "ArduinoISP.org"
Burn Bootloader
Boards Manager...
Arduino AVR Boards
Arduino Yún
Arduino/Genuino Uno
Arduino Duemilanove or Diecimila
Arduino Nano
Arduino/Genuino Mega or Mega 2560
Arduino Mega ADK
Arduino Leonardo
Arduino Leonardo ETH
Arduino/Genuino Micro
Arduino Esplora
Arduino Mini
Arduino Ethernet
Arduino Fio
Arduino BT
LilyPad Arduino USB
LilyPad Arduino
Arduino Pro or Pro Mini
Arduino NG or older
Arduino Robot Control
Arduino Robot Motor
Arduino Gemma
Adafruit Circuit Playground
Arduino Yún Mini
Arduino Industrial 101
Linino One
Arduino Uno WiFi
#include "RF24.h"
#define VX_VALUE (0.5f)
#define PIN A4//Lantern pin
#define BRIGHTNESS 50
#define ADR 90 // maximum Angle
Adafruit_NeoPixel strip = Adafru
RF24 radio(9, 10);
byte addresses[5] = "6";
NEO_GRB + NEO_KHZ800);//light
ntrol NRF24L01
dress which should correspond to remote contr
Done Saving.
27
Arduino/Genuino Uno on COM8



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 robotics 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. 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
website:www.adeept.com