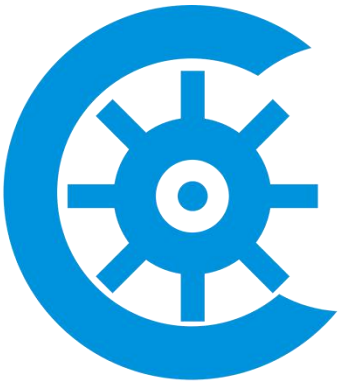
Adeept

Adeept Self-balancing Car Kit for Arduino



Adeept

www.adeept.com

Warning

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

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

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

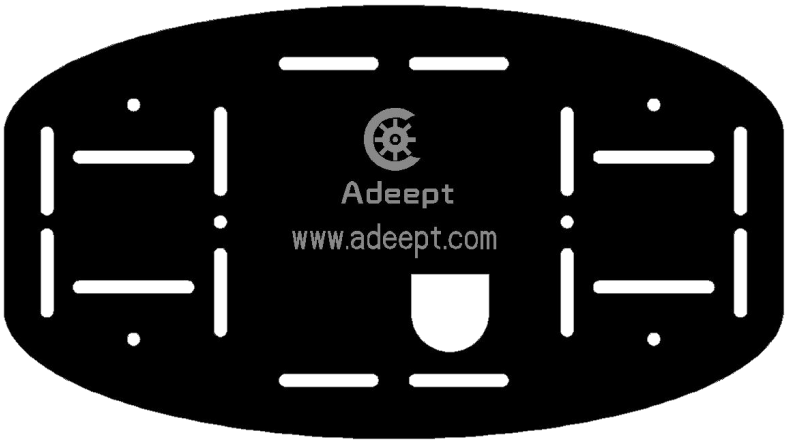
On weekdays, we usually will reply within 24 hours. Also welcome to post forums on our website.

Copyright

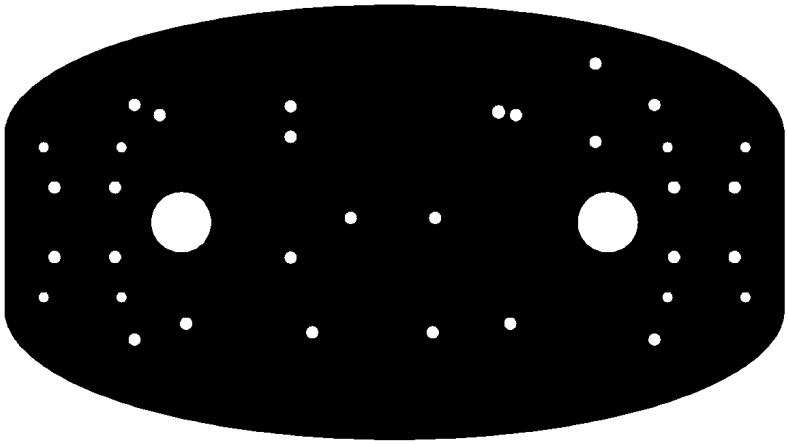
contents

Components List

Acrylic Plates



A01
1pcs



A02
1pcs

A03
1pcs

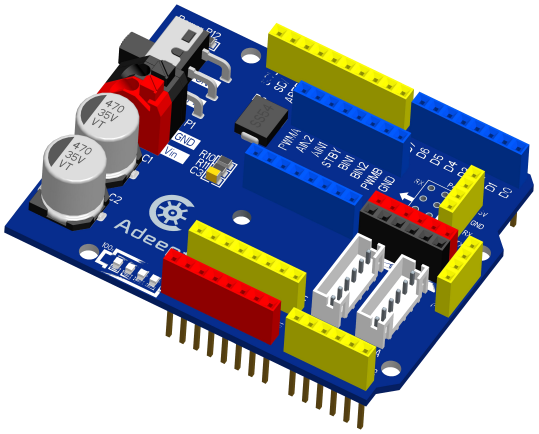
A04
6pcs

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.

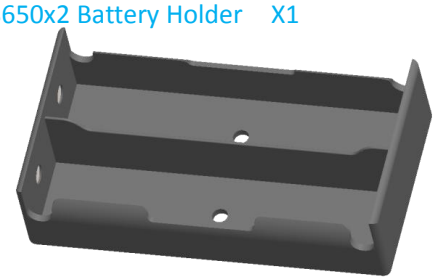
Machinery Parts

M2 Nut  X6 www.adeept.com	M3 Nut  X12 www.adeept.com	M2*14 Screw  X6 www.adeept.com	M2.5*4 Screw  X2 www.adeept.com	M3*4 Screw  X4 www.adeept.com
M3*8 Screw  X8 www.adeept.com	M3*12 Screw  X8 www.adeept.com	M3*18 Screw  X4 www.adeept.com	M3*12 Countersunk Head Screw  X2 www.adeept.com	M1.4*6 Self-tapping Screw  X4 www.adeept.com
M2.5*11 Copper Standoff  X1 www.adeept.com	M3*8 Copper Standoff  X4 www.adeept.com	M3*12 Copper Standoff  X4 www.adeept.com	M3*65 Copper Standoff  X4 www.adeept.com	

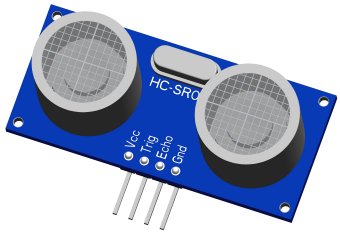
Electronic Parts

Adeept UNO R3 Board X1 	Adeept Self-Balancing Robot Shield X1 

18650x2 Battery Holder X1

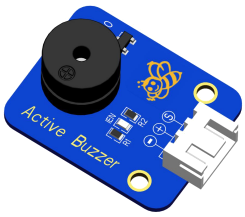


Adeept Ultrasonic Module X1

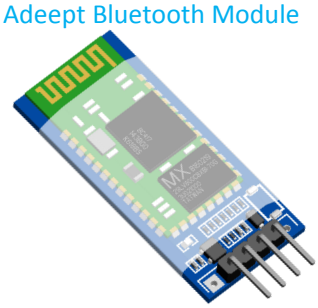


Adeept RGB LED Module X2

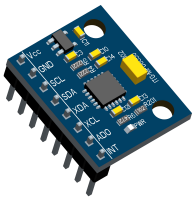
Adeept Active Buzzer Module X1



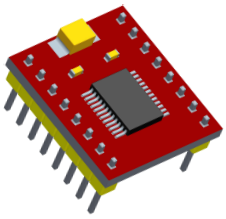
Adeept Bluetooth Module X1



Adeept MPU6050 Module X1

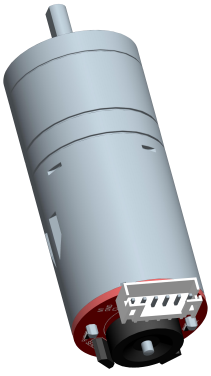


Adeept Motor Driver Module X1

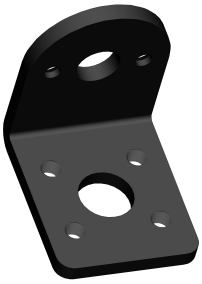


Adeept Motor Set

Tire
X 2



Adeept Motor
X 2



Bracket
X 2

Coupler
X2

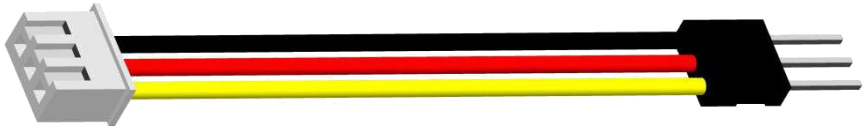
Hex Wrench
X 2

M4*8 Screw
X 2

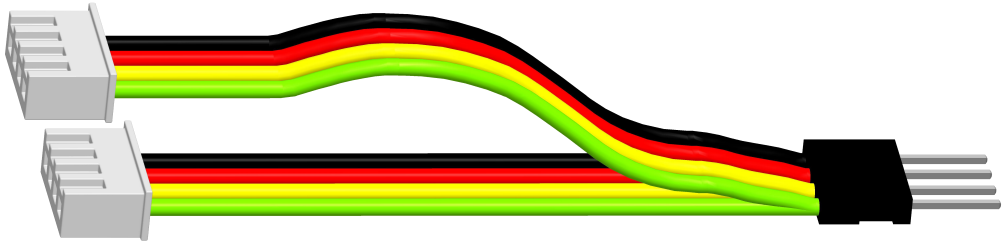
M3*6 Screw
X 4

M4*4 Locking Screw
X 4

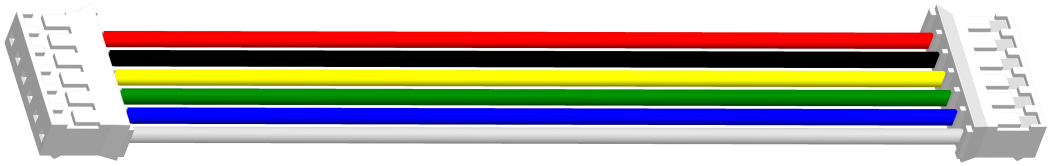
3-Pin Wires X1



4-Pin Wires X1



6-Pin Wires X2



USB Cable X1

Tools

Cross Screwdriver X1

Slotted Screwdriver X1

Cross Socket Wrench X1

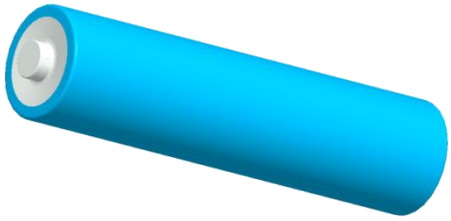
Large Cross-head Screwdriver X1

Winding Pipe X1

Ribbon X1

Self-prepared Parts

18650 Battery X2



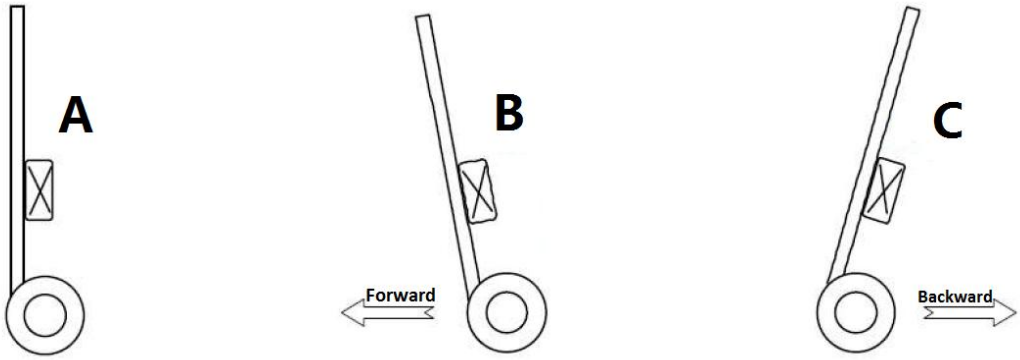
About the Adeept Self-Balancing Robot Car

Overview

This kit is a self-balancing robot learning kit for education. With all components needed for this robot car and the tutorials for assembly, you can make a self-balancing robot following the instructions step by step.

The robot is developed based on the Dynamic Stabilization principle for its movement. It uses the MPU6050 precision gyroscope sensor to tell the tilt of the car body, and the Arduino UNO R3 to calculate the PID (Proportional-Integral-Derivative) parameters based on the data sent back by the MPU6050, thus keeping a dynamic balance of the car body by controlling the motors accordingly.

Based on the figures below, in Fig. A the car stands straight and still. In Fig. B, the wheels need to accelerate towards the left to keep a balance and back to the state in Fig. A; if it keeps the tilting, it'll run forward . In Fig. C, the wheels need to spin towards the right to keep a balance and back to the state in Fig. A; if it keeps the tilting, the car will run backwards.



Understanding the principle, you can make the self-balancing robot car then.

Three working modes after the car is power on:

Mode No.1: Remote control via Bluetooth

You can control the car to go forward and backward and turn left and right by commands via Bluetooth. At the same time, you can switch between the modes and control the buzzer to beep.

Mode No.2: Obstacle avoidance by ultrasonic

Under this mode, the car can detect and bypass the obstacles in front automatically.

Mode No.3: Following

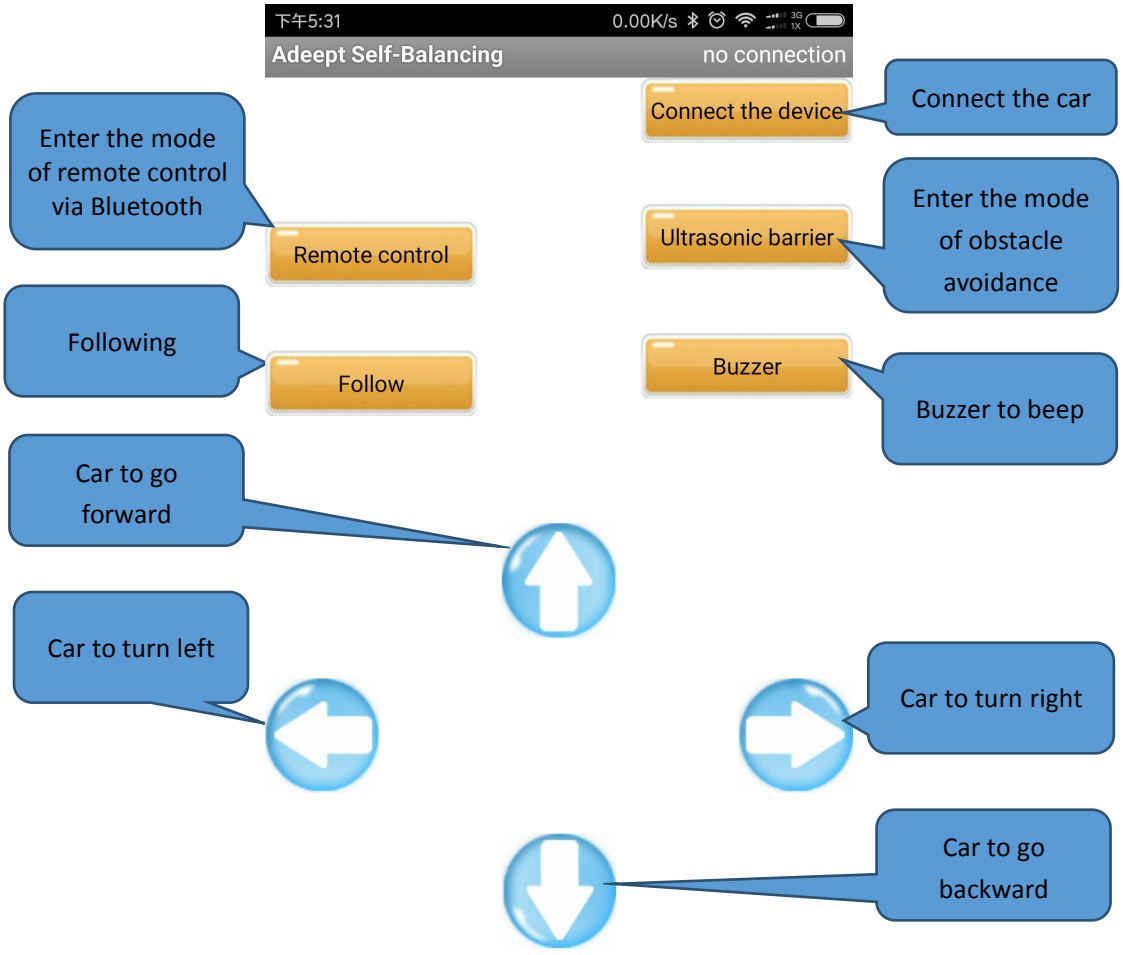
Under this mode, the car will follow the object ahead straight. When it's 30-50cm to the object, the light onside will light up and the car will follow the object to move forward; when it's 5-20cm, the light turns into green, and the car will move backward.

Commands for Remote Control via Bluetooth

Command from the Remote via Bluetooth	Car's Function to the Command
a	Car goes forward
b	Car turns right
c	Car turns left
d	Car goes backward
e	Enter the mode of remote control via Bluetooth; the car stops
f	Car starts to avoid obstacles automatically
g	Enter the mode of following
h	Turn on buzzer
i	Turn off buzzer

Android App

An application for Android mobile is provided for you to test the car:

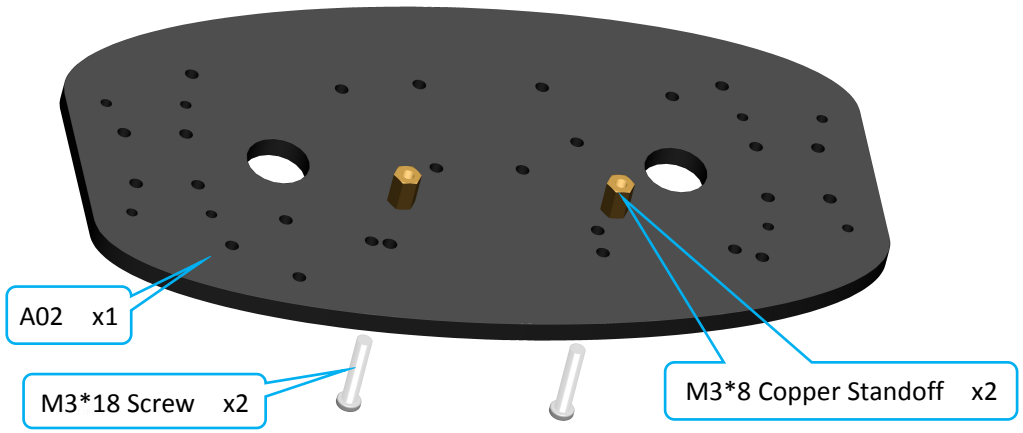


Assembly

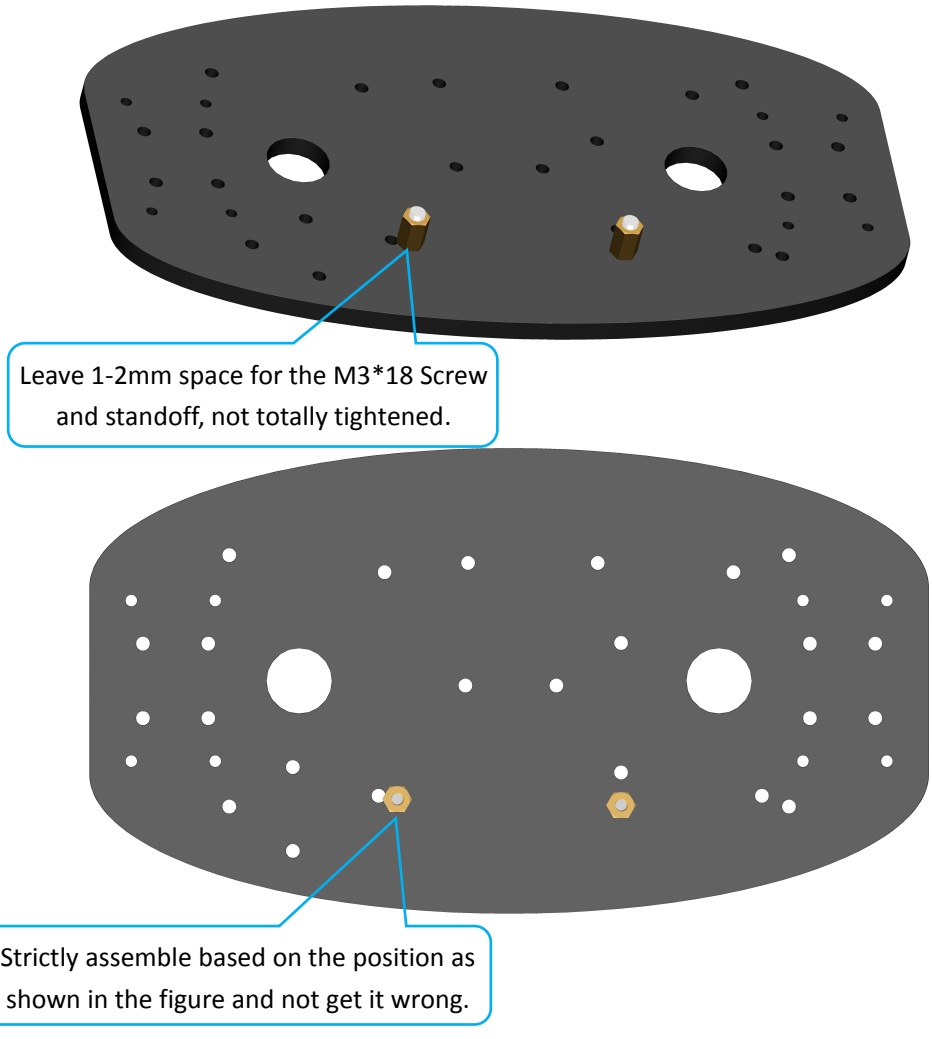
Preliminary Assembly

A. Assemble two M3*8 Copper Standoff onto the A02 plate.

Assemble the following components

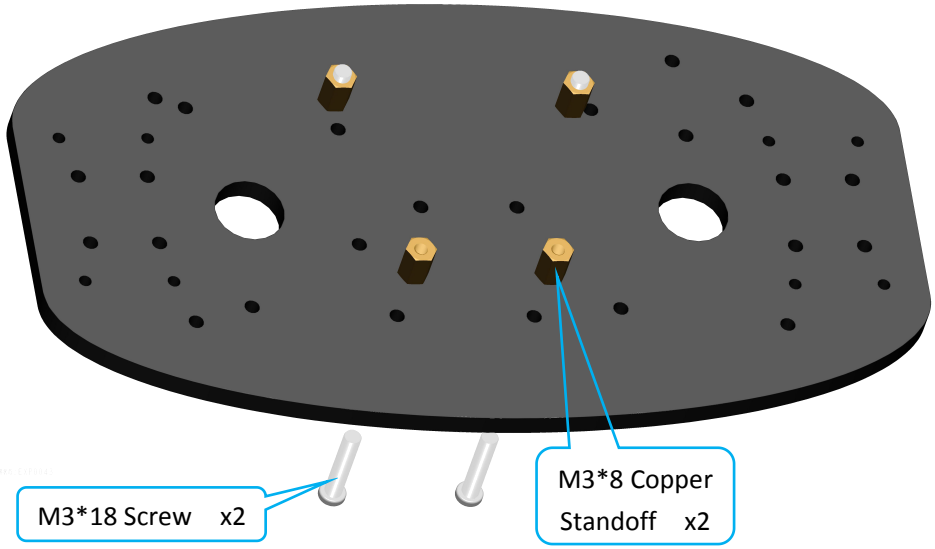


Effect diagram after assembling

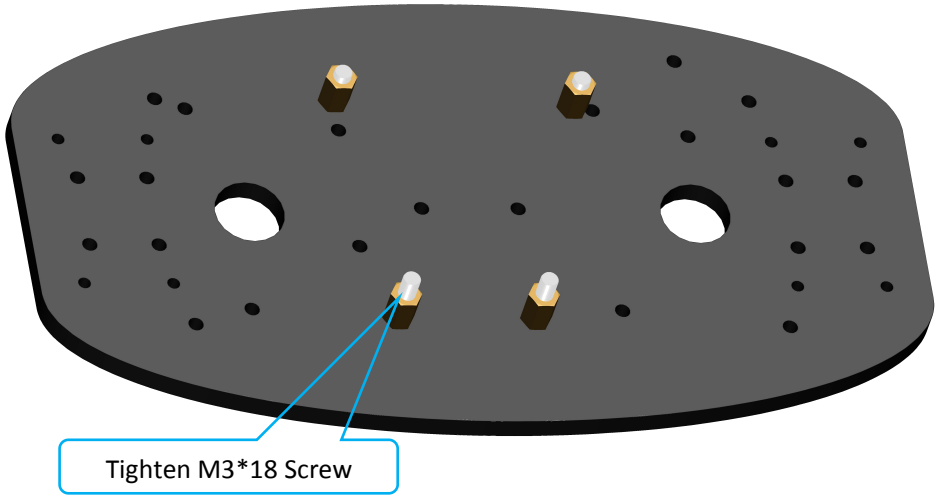


B. Assemble two M3*8 Copper Standoffs onto the A02.

Assemble the following components

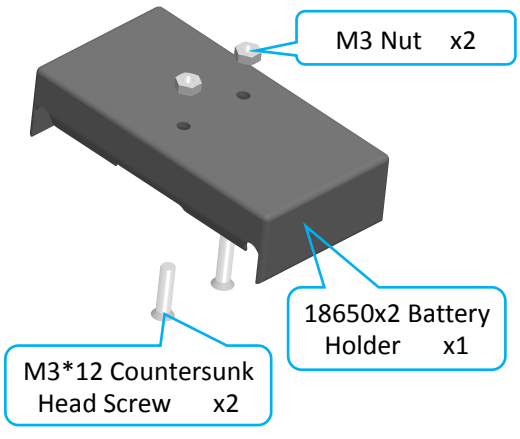


Effect diagram after assembling

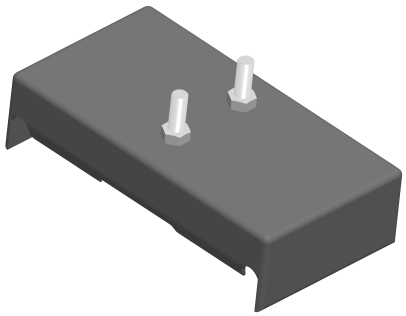


C. Assemble two M3*12 Countersunk Head Screws and the 18650x2 Battery Holder.

Assemble the following components

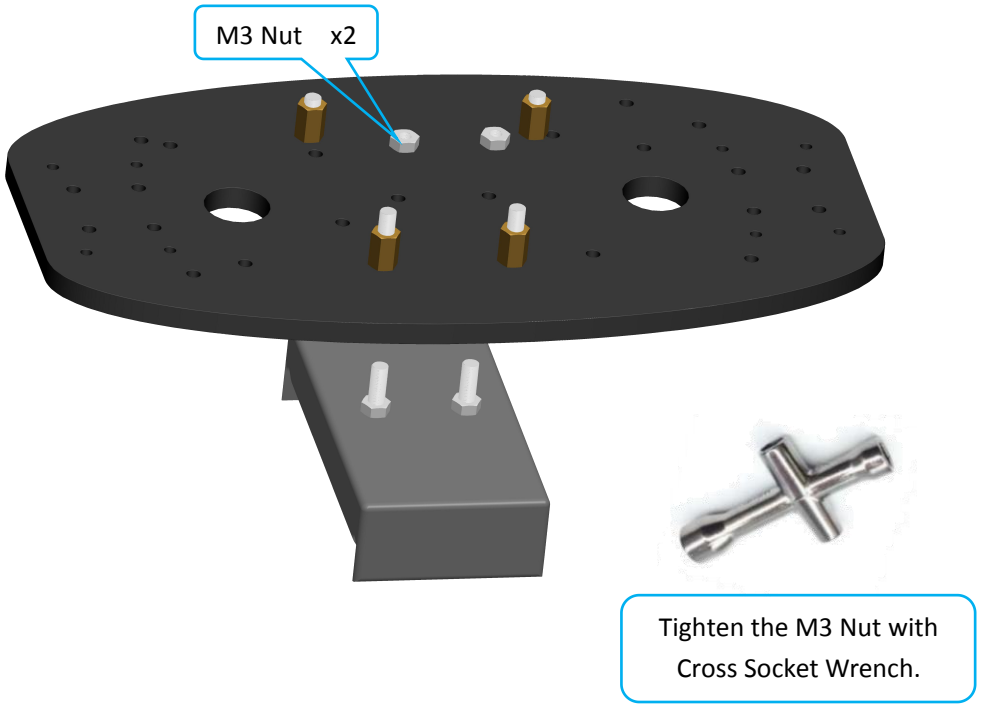


Effect diagram after assembling

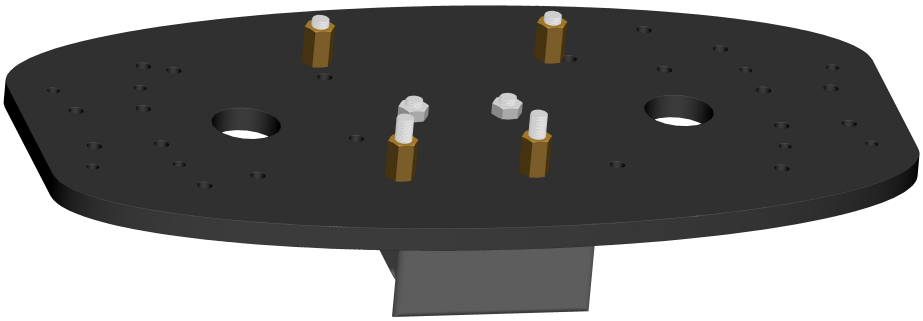


D. Fasten the 18650x2 Battery Holder onto the A02.

Assemble the following components

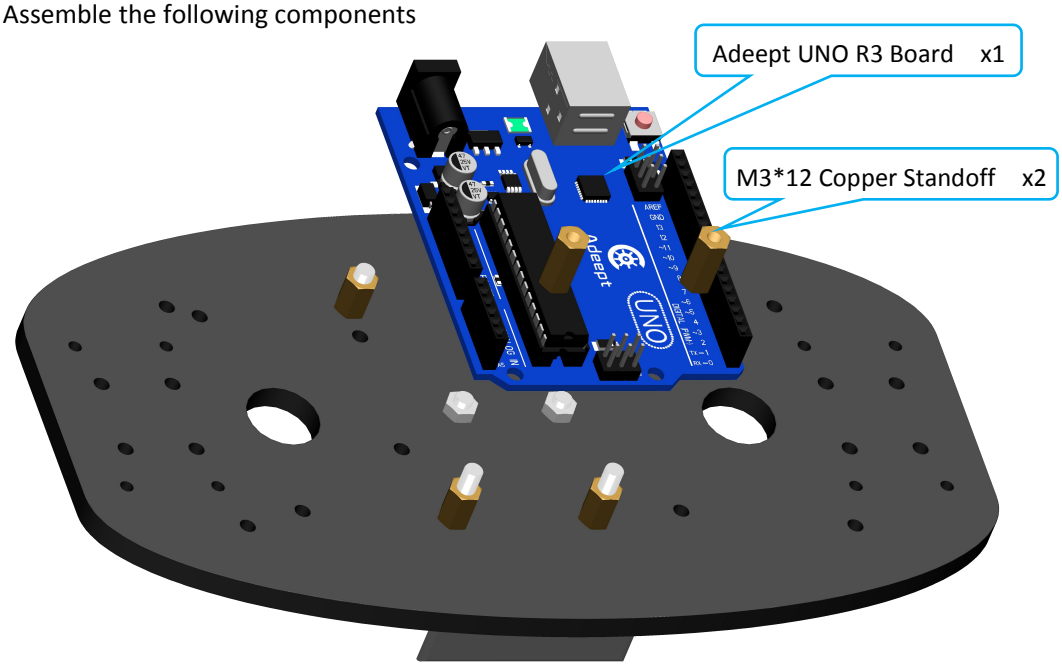


Effect diagram after assembling

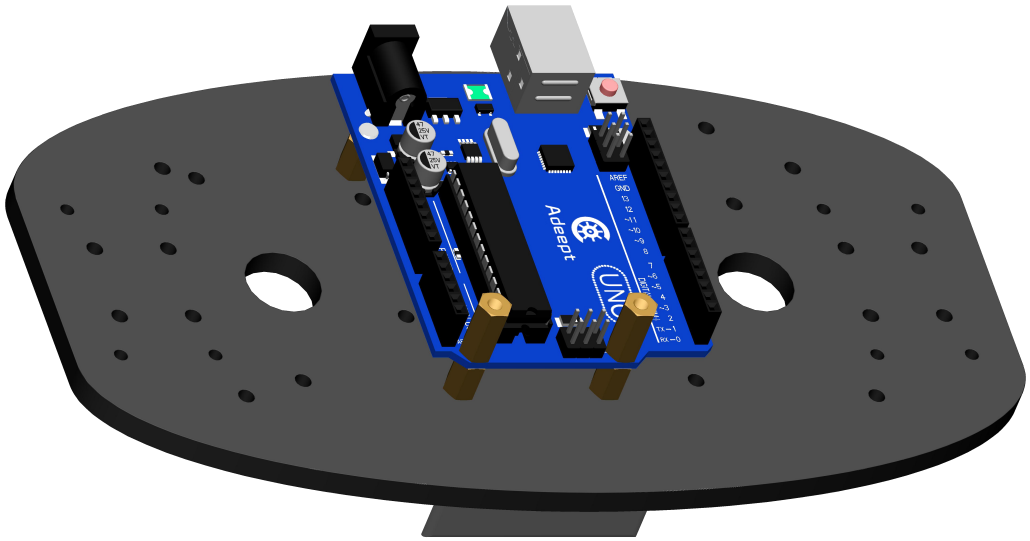


E. Fasten the Adeept UNO R3 Board.

Assemble the following components



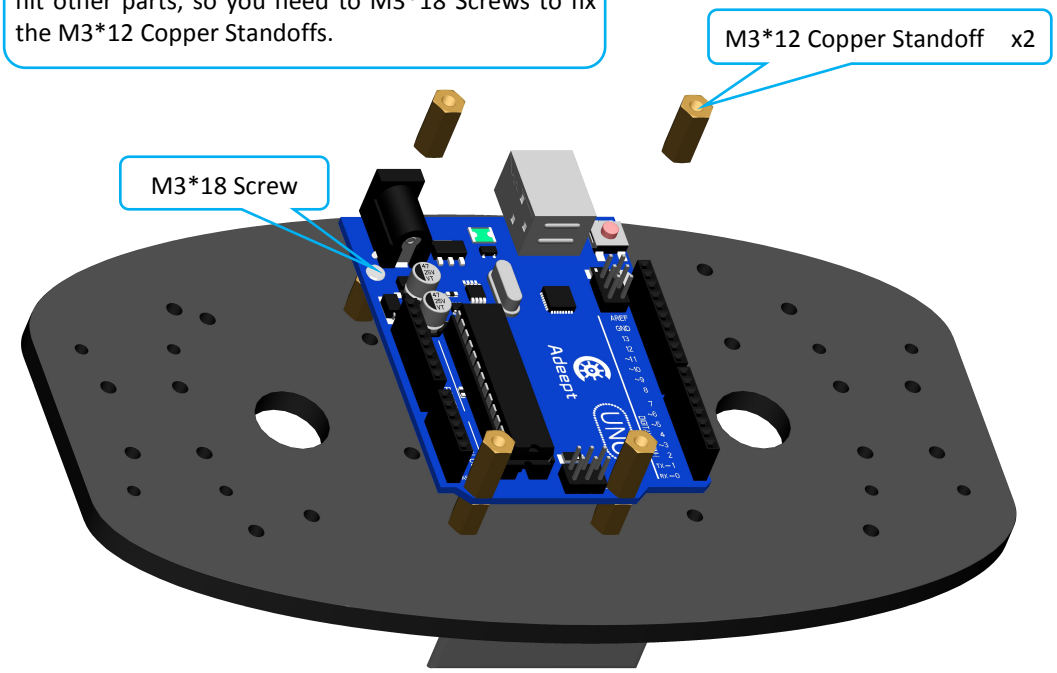
Effect diagram after assembling



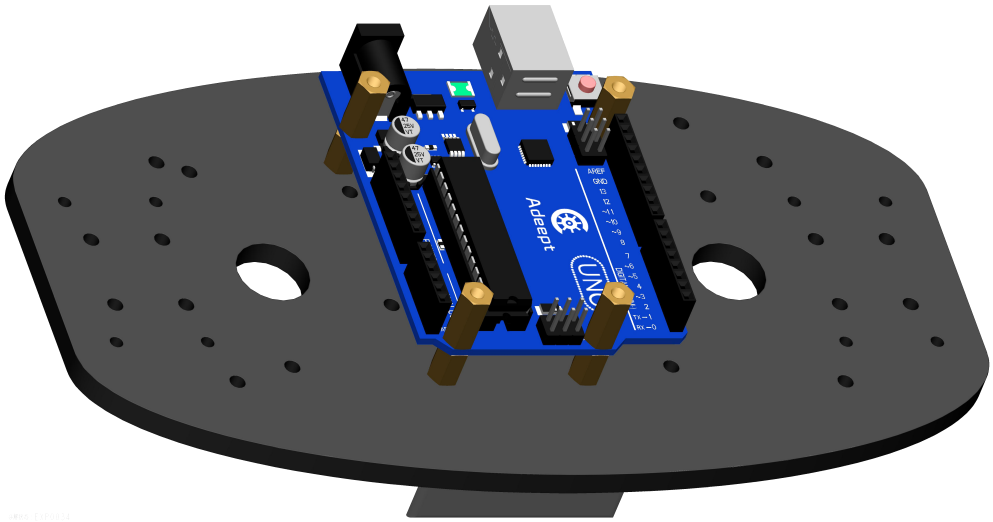
F. And fasten two M3*12 Copper Standoffs.

Assemble the following components

If you rotate the M3*12 Copper Standoffs, they may hit other parts, so you need to M3*18 Screws to fix the M3*12 Copper Standoffs.



Effect diagram after assembling

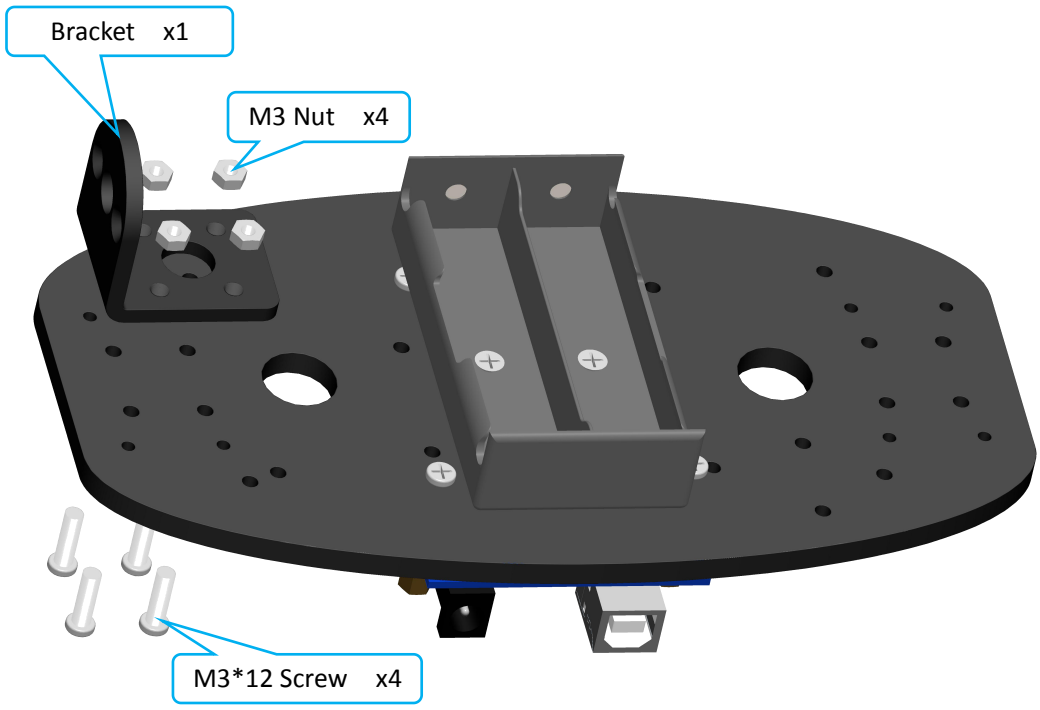


Tire and Motor Assembly

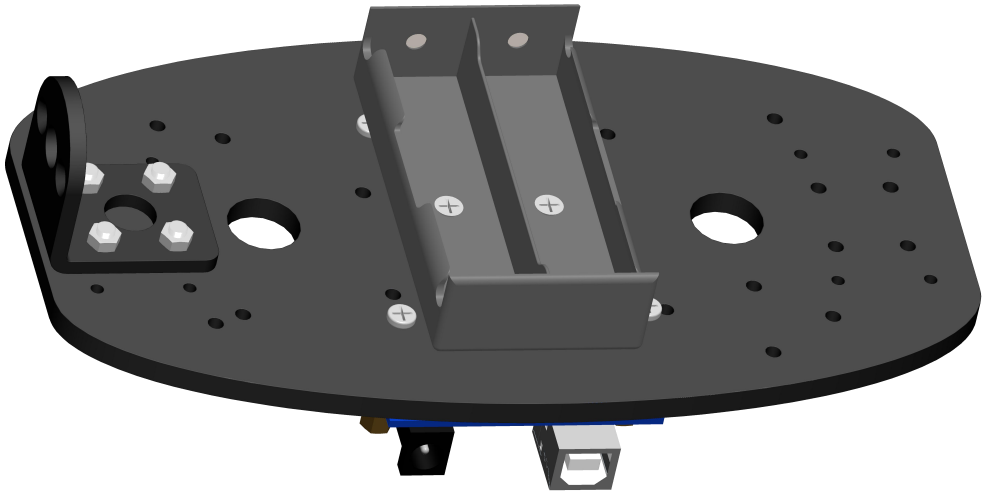
Take a motor set for assembly.

A. Fix the bracket onto the A02 plate.

Assemble the following components

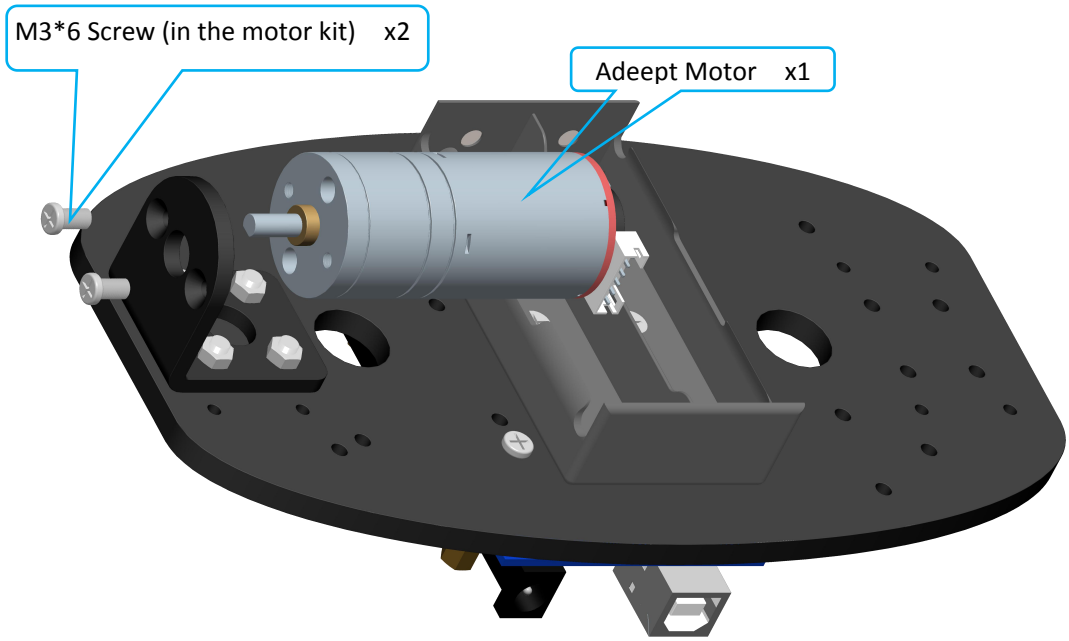


Effect diagram after assembling

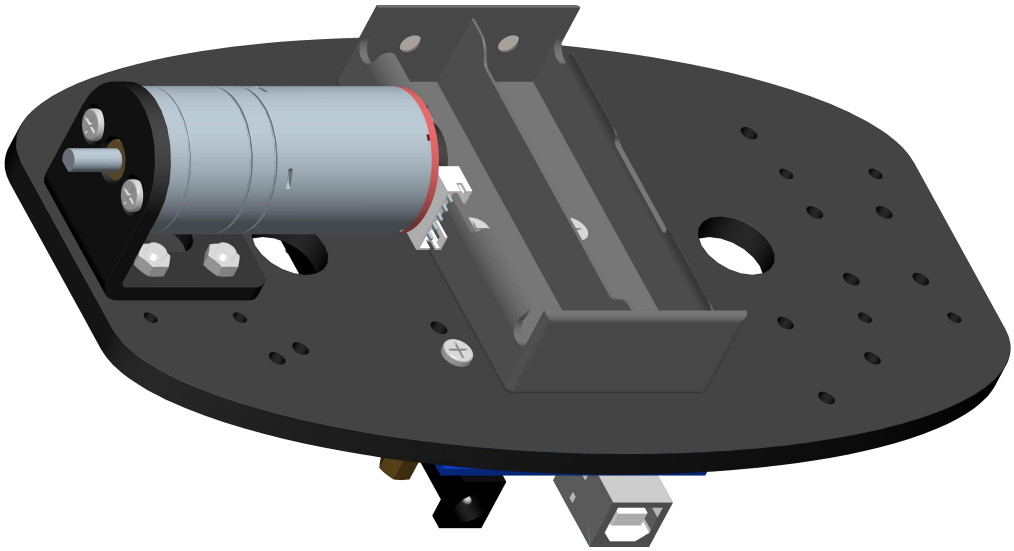


B. Install the Adeept motor to the bracket.

Assemble the following components

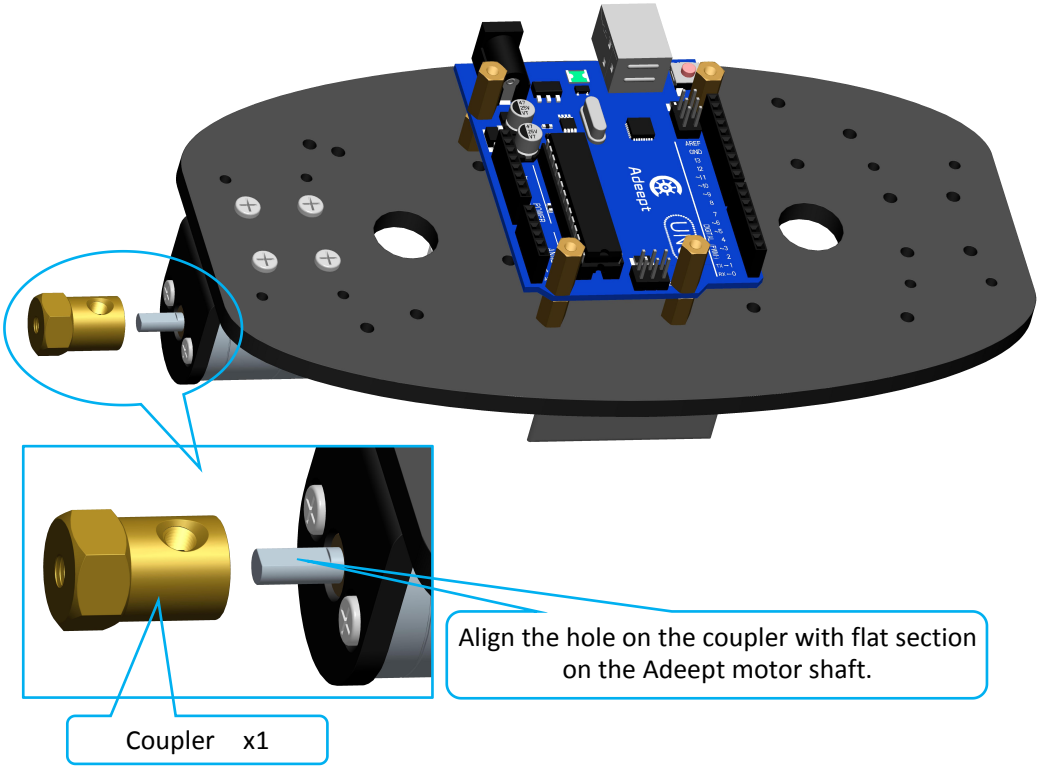


Effect diagram after assembling

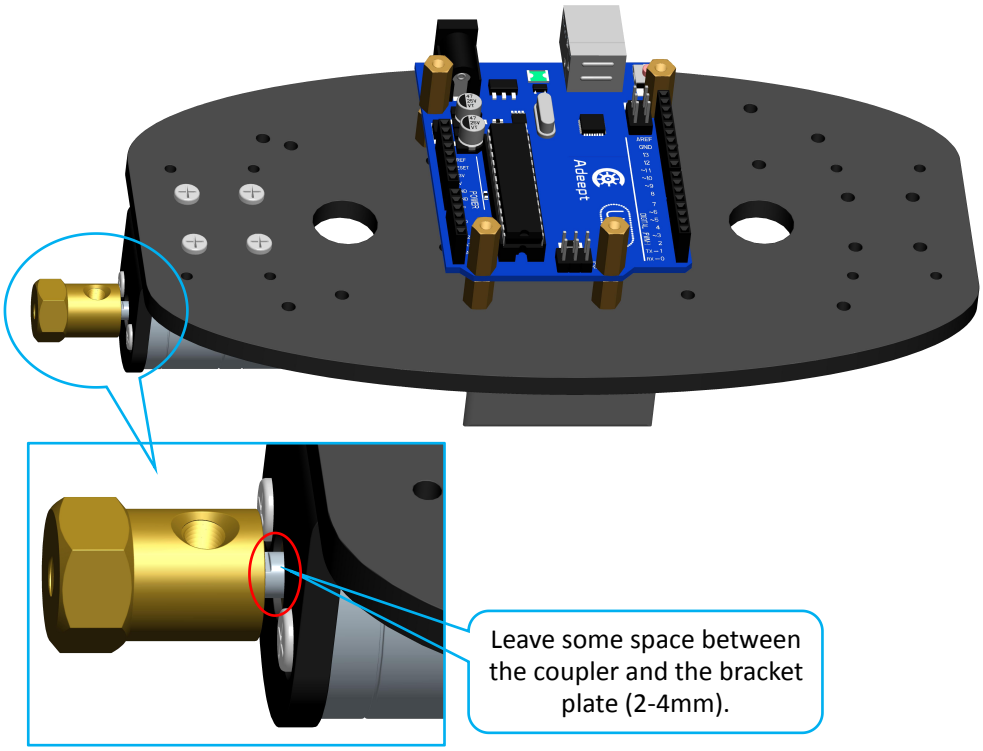


C. Insert the coupler into the Adeept motor shaft.

Assemble the following components

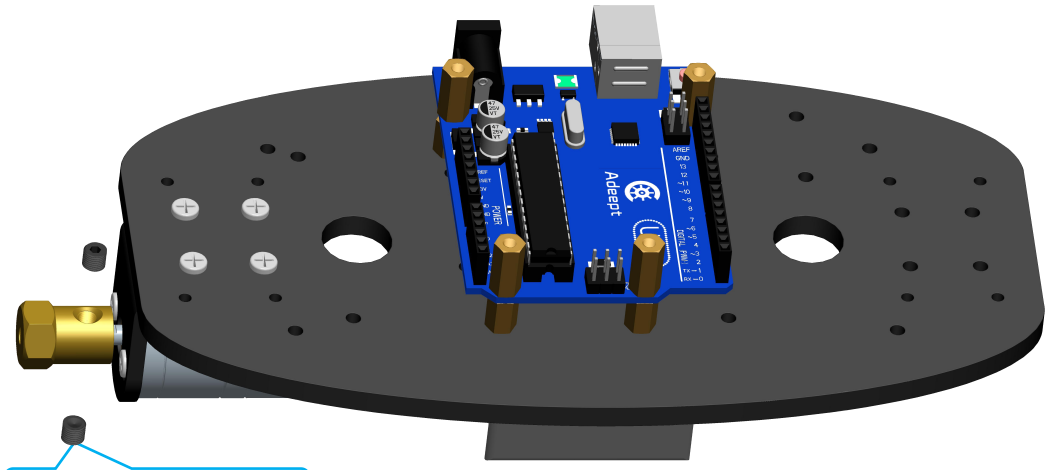


Effect diagram after assembling



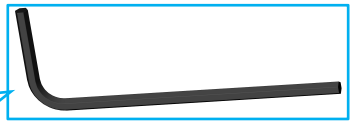
D. Fix the coupler with locking screws.

Assemble the following components

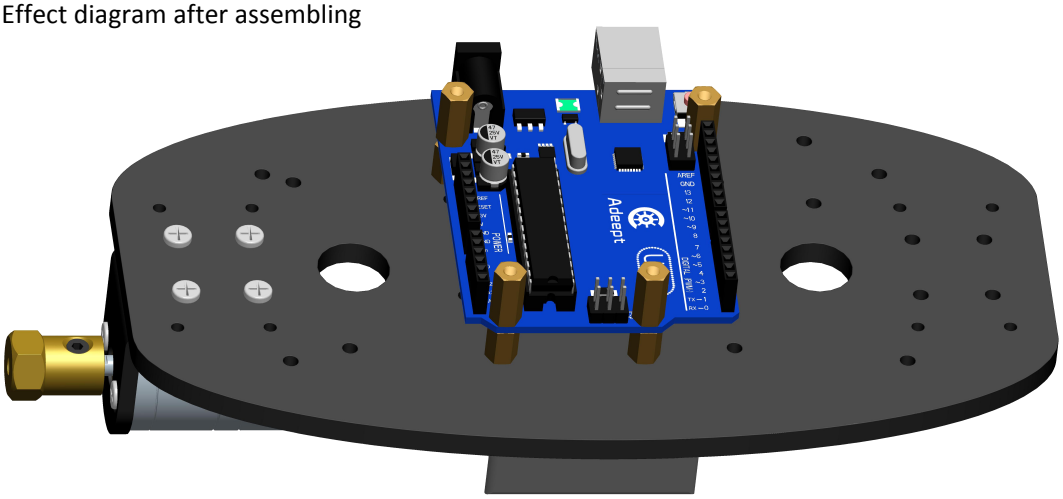


M4*4 Locking Screw x2

Screw the M4*4 locking screws tight into the coupler with the hex wrench

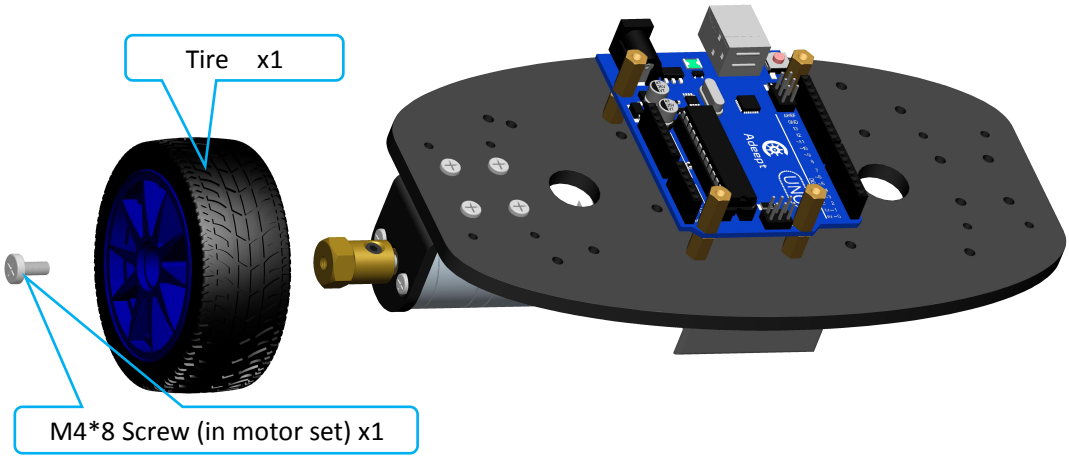


Effect diagram after assembling

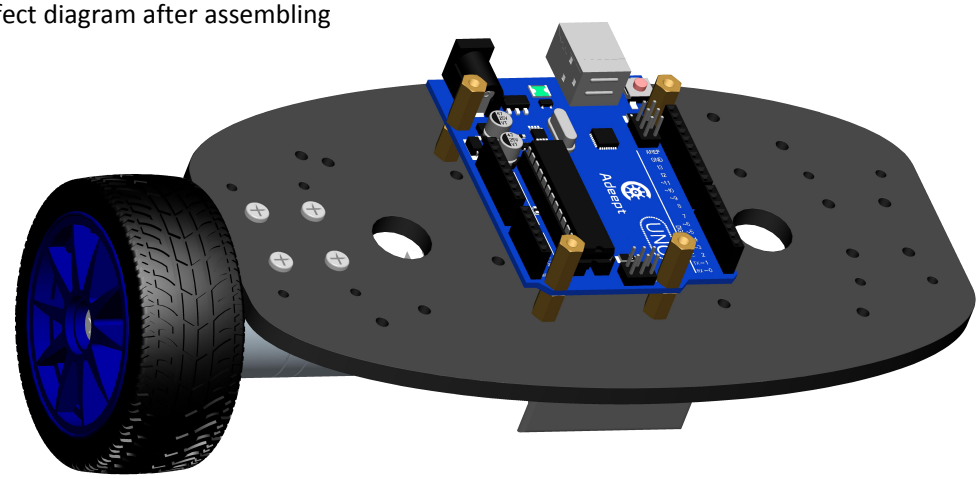


E. Fasten the tire to the coupler.

Assemble the following components

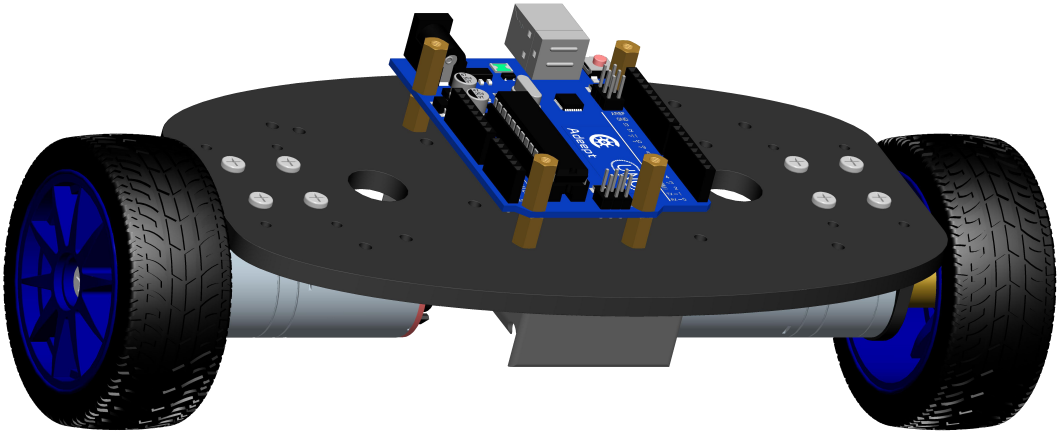


Effect diagram after assembling



Assemble the other tire similarly. They should be like the following:

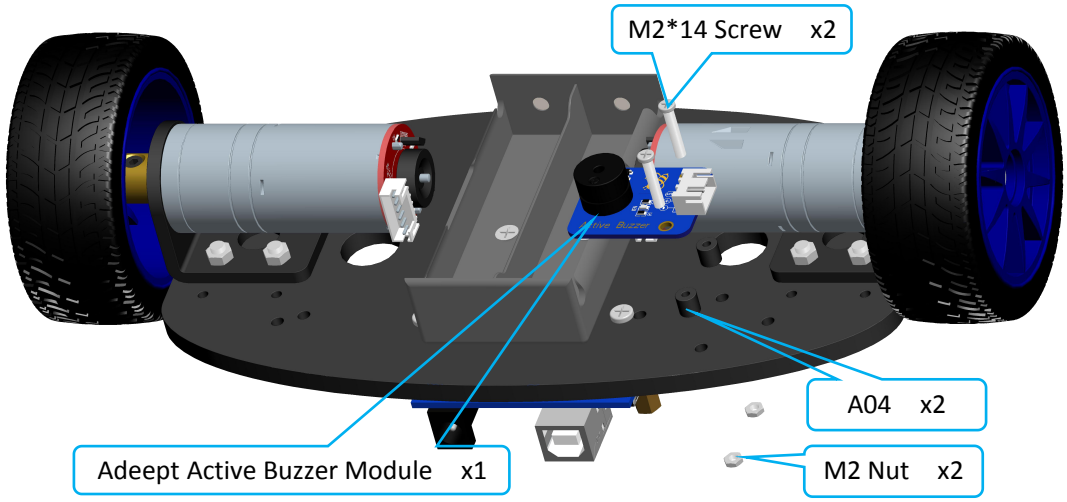
Effect diagram after assembling



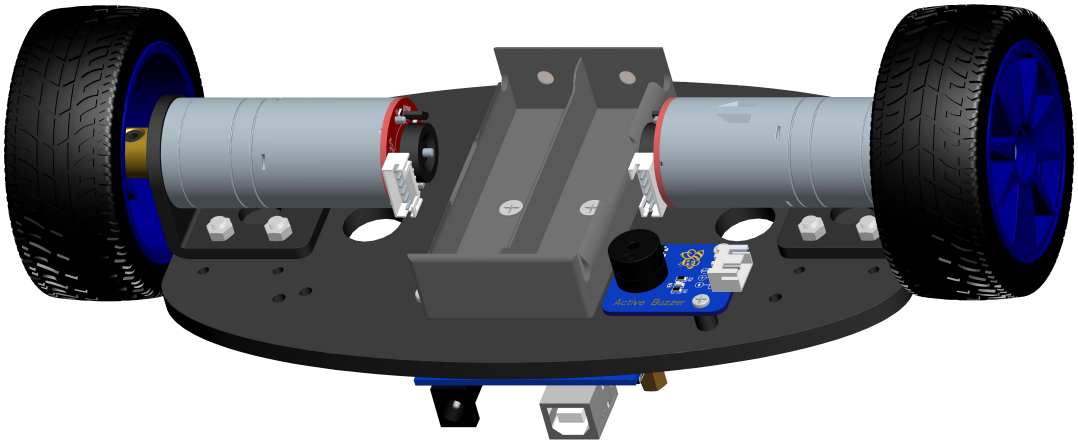
Final Assembly

A. Fasten the Adeept Active Buzzer Module beneath the A02.

Assemble the following components

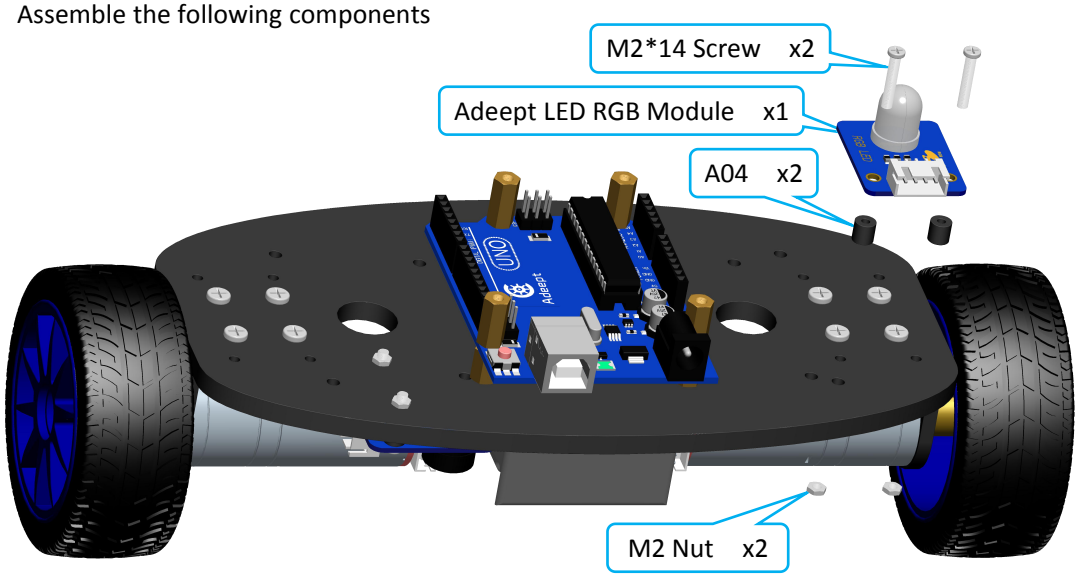


Effect diagram after assembling

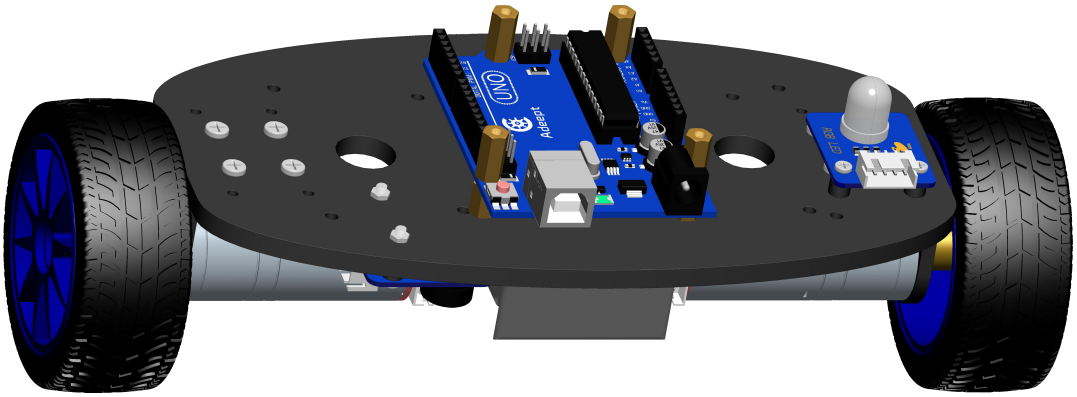


B. Mount the Adeept LED RGB Module on the A02.

Assemble the following components

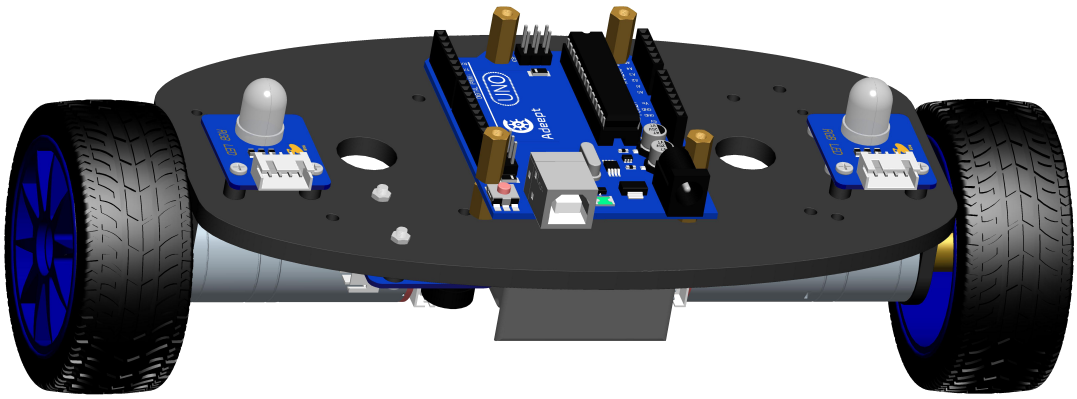


Effect diagram after assembling



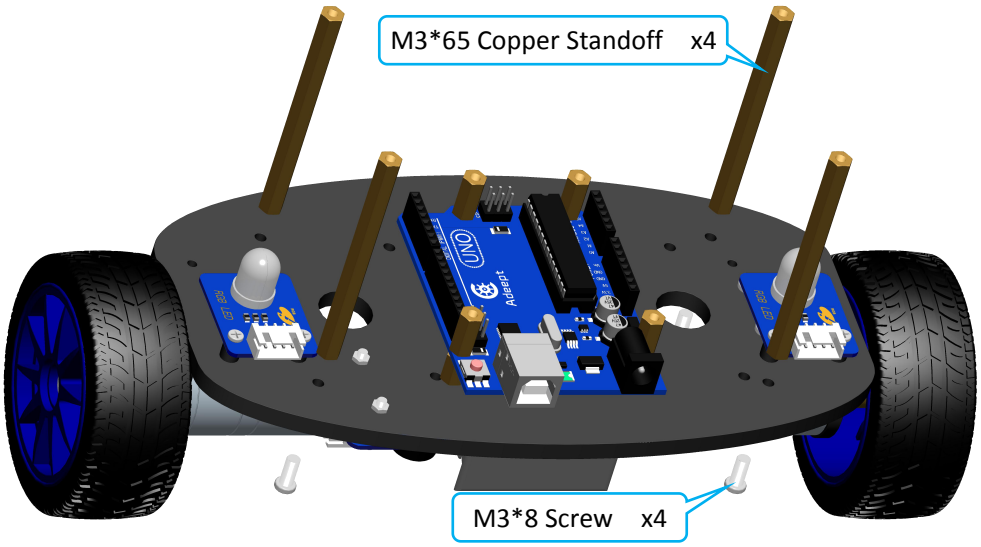
Assemble the other Adeept LED RGB Module on the other side of the A02 as below:

Effect diagram after assembling

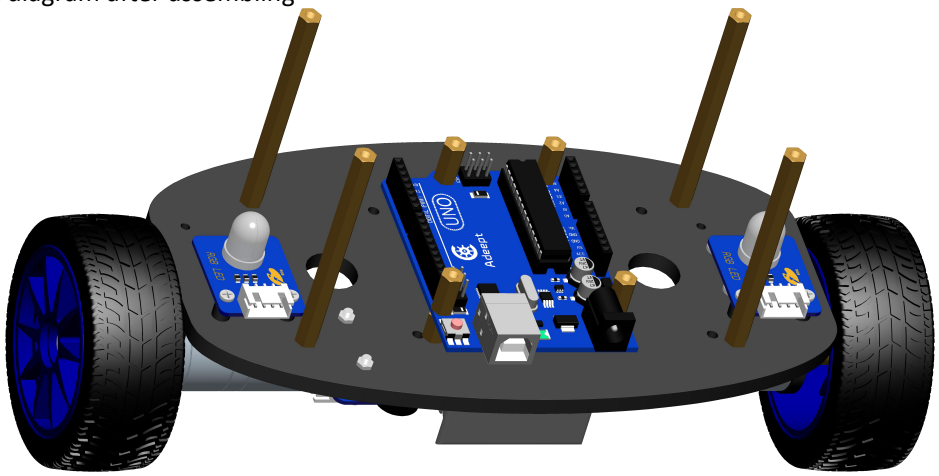


C. Mount 4 M3*65 Copper Standoffs onto the A02.

Assemble the following components

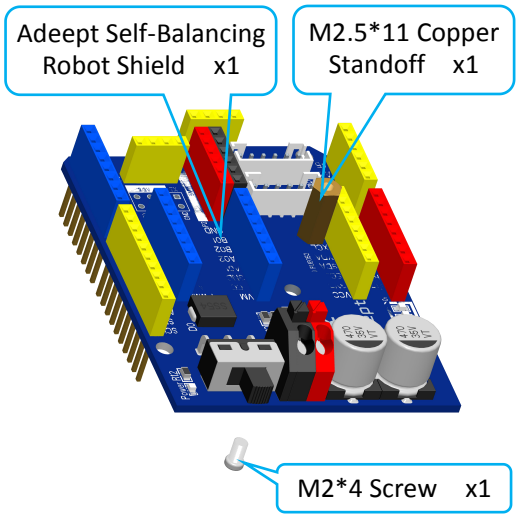


Effect diagram after assembling

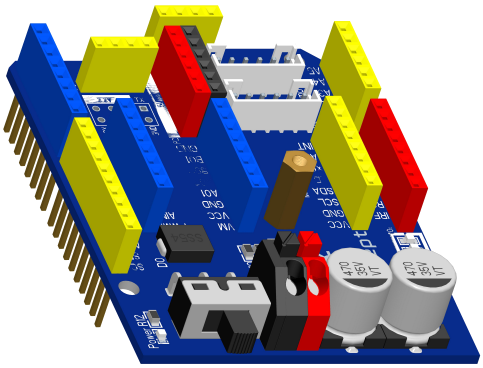


D. Fix an M2.5*11 Copper Standoff onto the Adeept Self-Balancing Robot Shield.

Assemble the following components

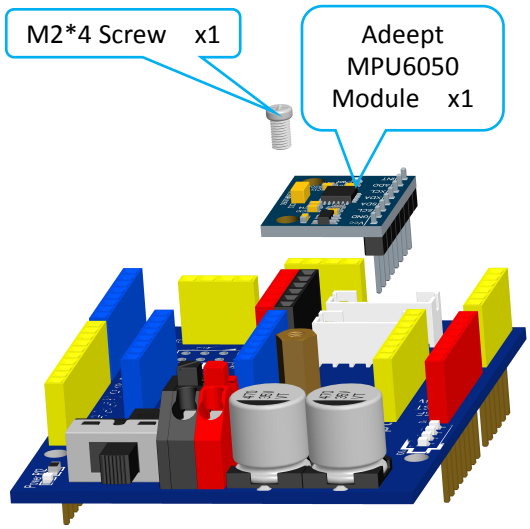


Effect diagram after assembling

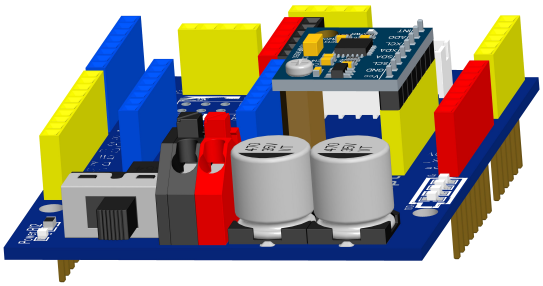


E. Fasten the Adeept MPU6050 Module to the Adeept Self-Balancing Robot Shield.

Assemble the following components

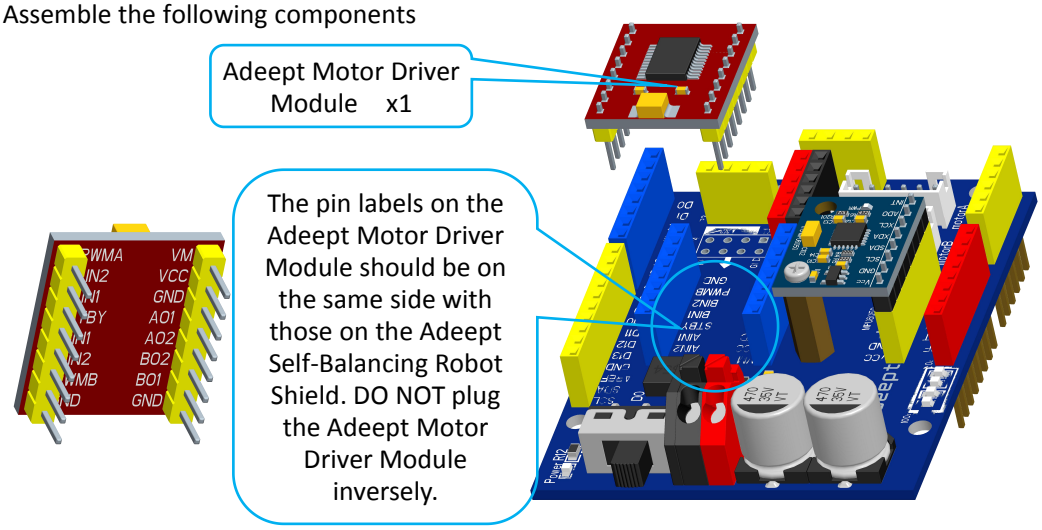


Effect diagram after assembling

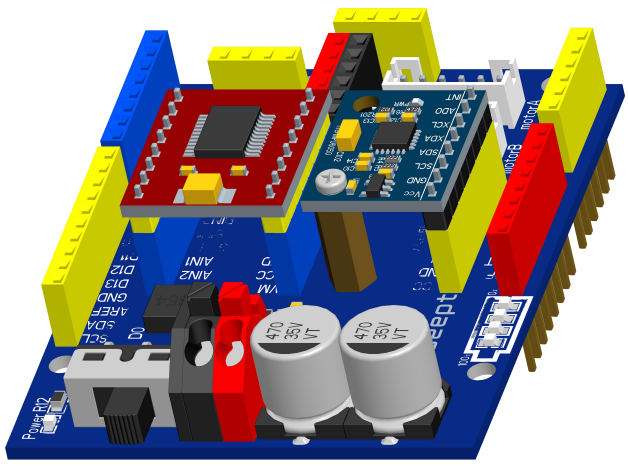


F. Plug the Adeept Motor Driver Module onto the Adeept Self-Balancing Robot Shield.

Assemble the following components

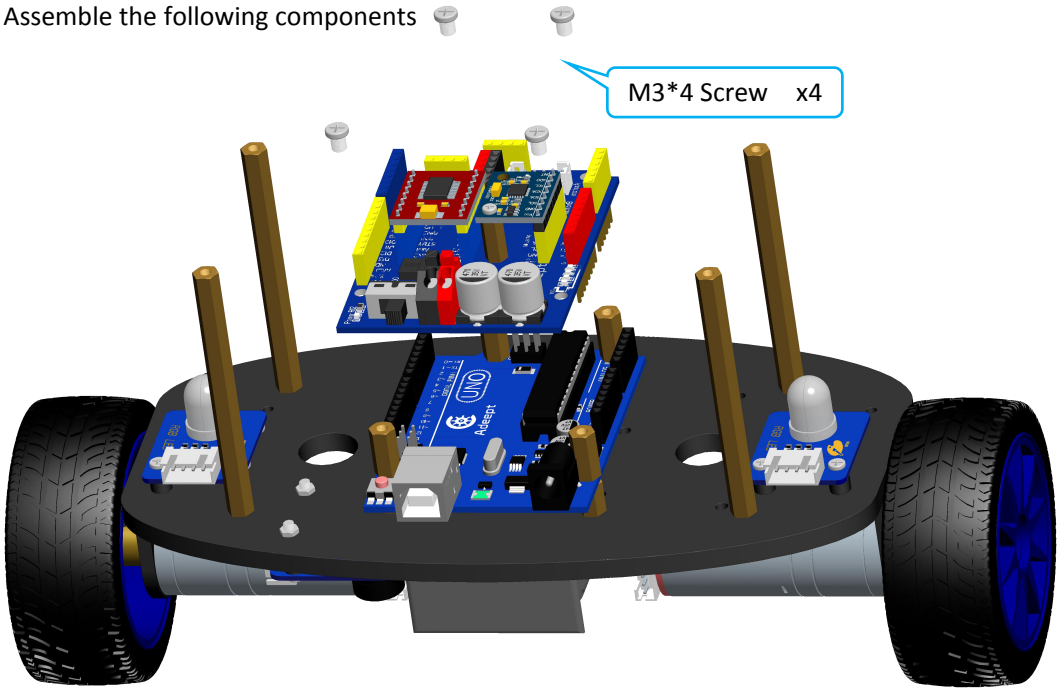


Effect diagram after assembling

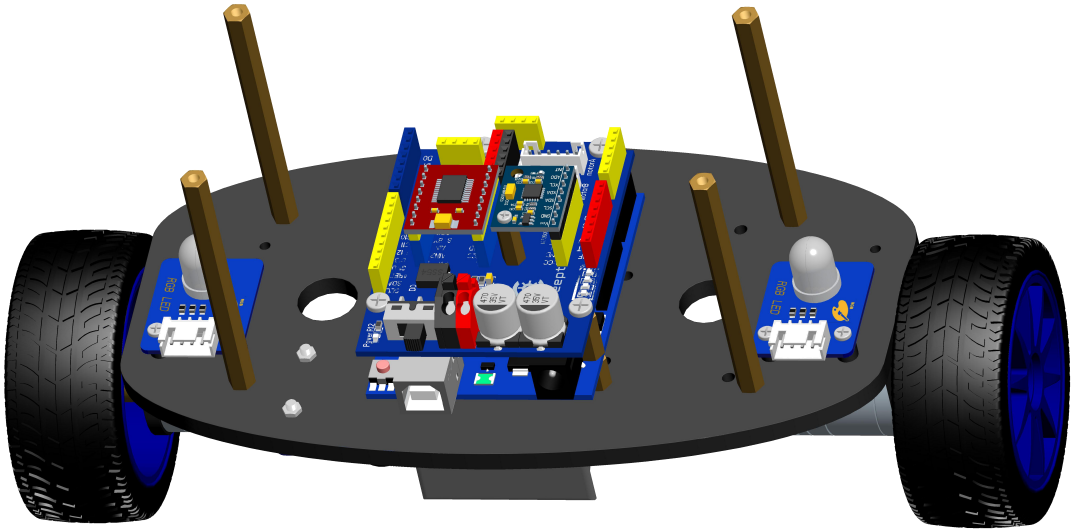


G. Assemble the Adeept Self-Balancing Robot Shield onto the Adeept UNO R3 Board.

Assemble the following components

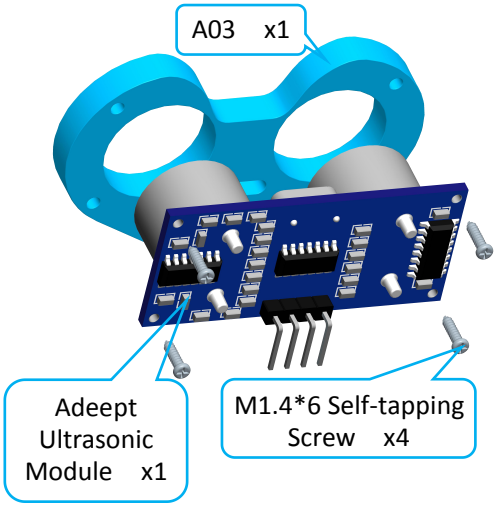


Effect diagram after assembling

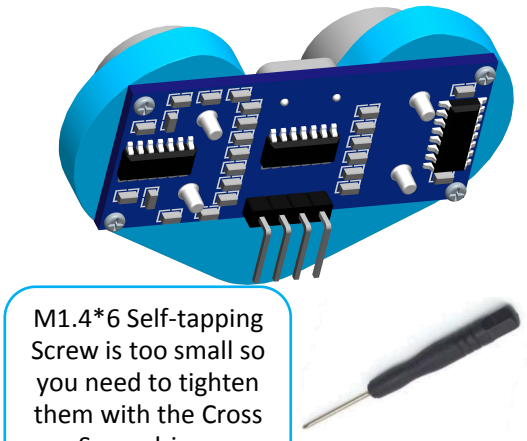


H. Fasten the Adeept Ultrasonic Module and the A03.

Assemble the following components



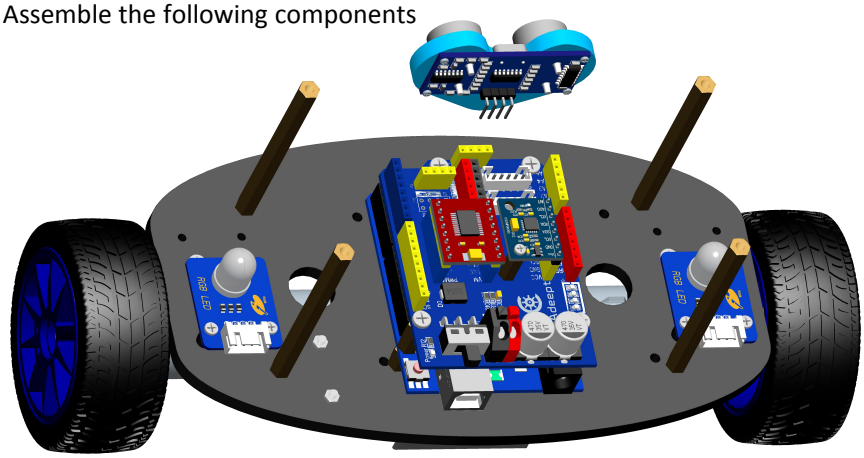
Effect diagram after assembling



M1.4*6 Self-tapping Screw is too small so you need to tighten them with the Cross Screwdriver.

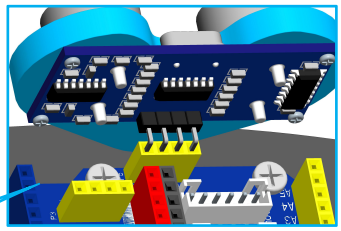
I. Then plug the Adeept Ultrasonic Module into the Adeept Self-Balancing Robot Shield.

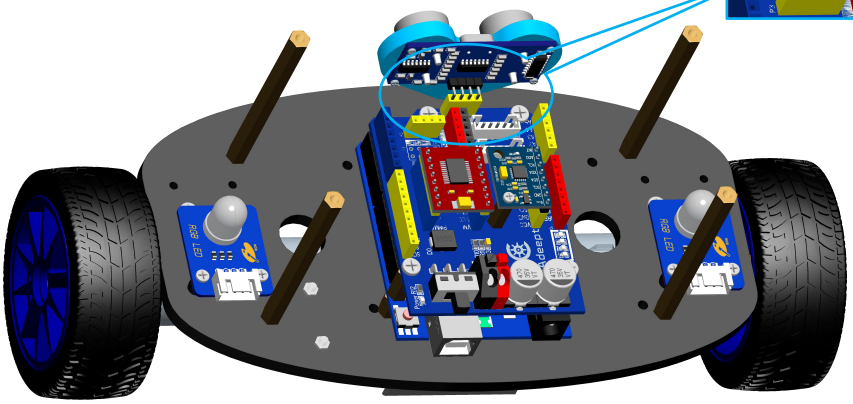
Assemble the following components



Effect diagram after assembling

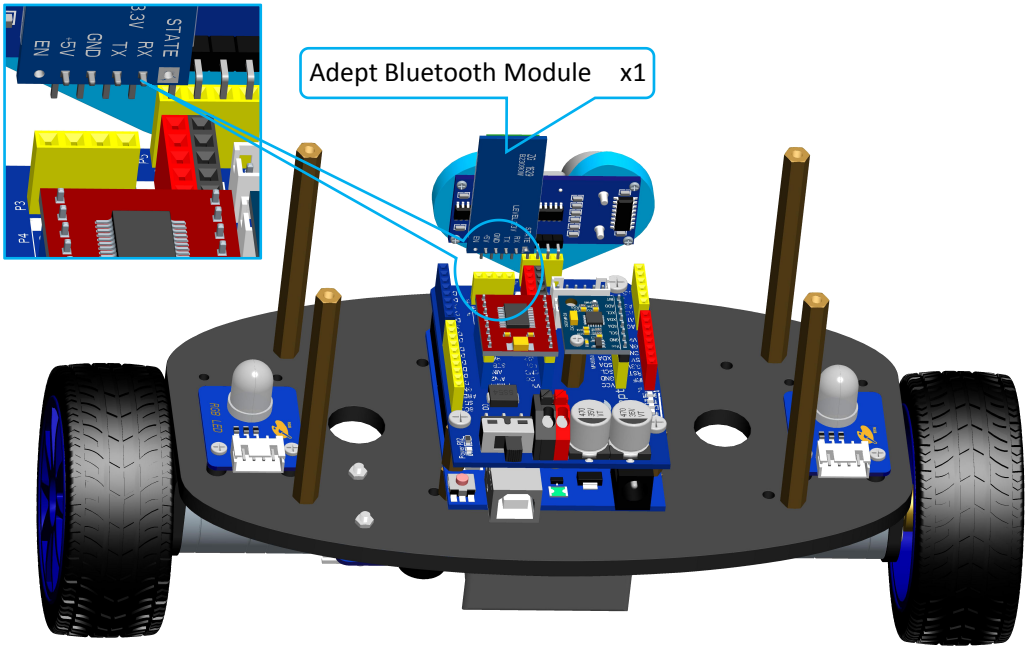
Plug the Adeept Ultrasonic Module into port P5 of the Adeept Self-Balancing Robot Shield with the two probes towards the front.



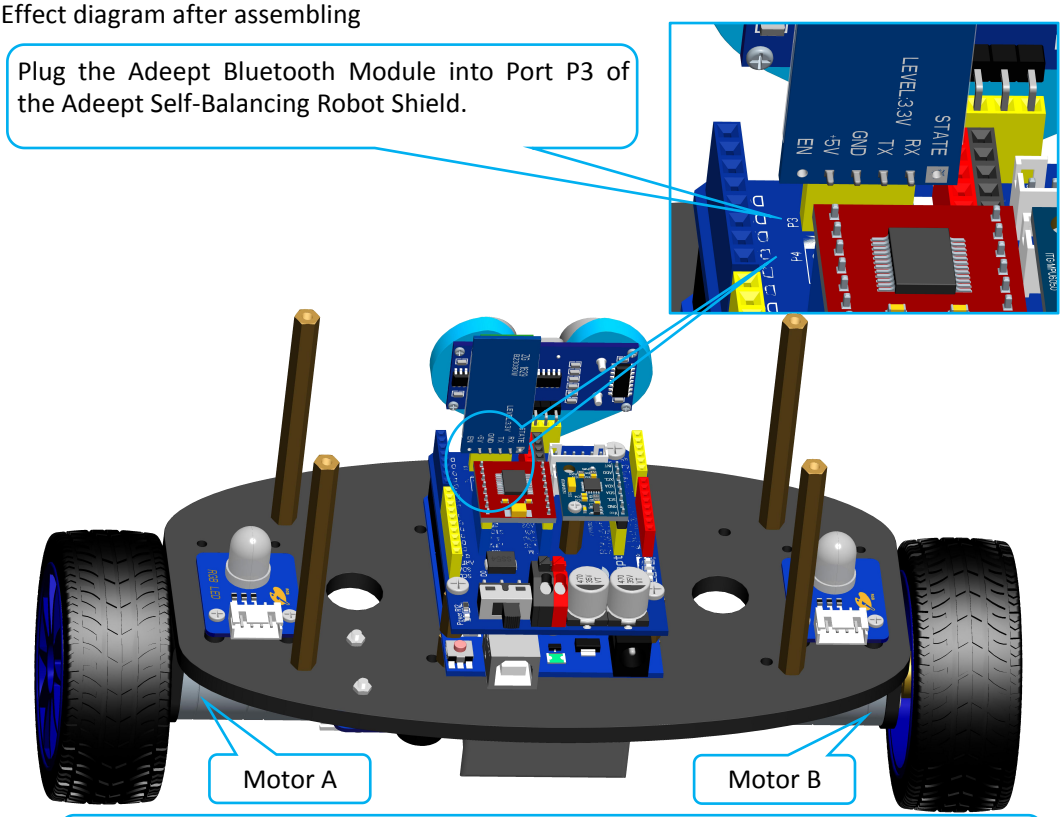


J. Plug the Adeept Bluetooth Module into the Adeept Self-Balancing Robot Shield.

Assemble the following components



Effect diagram after assembling

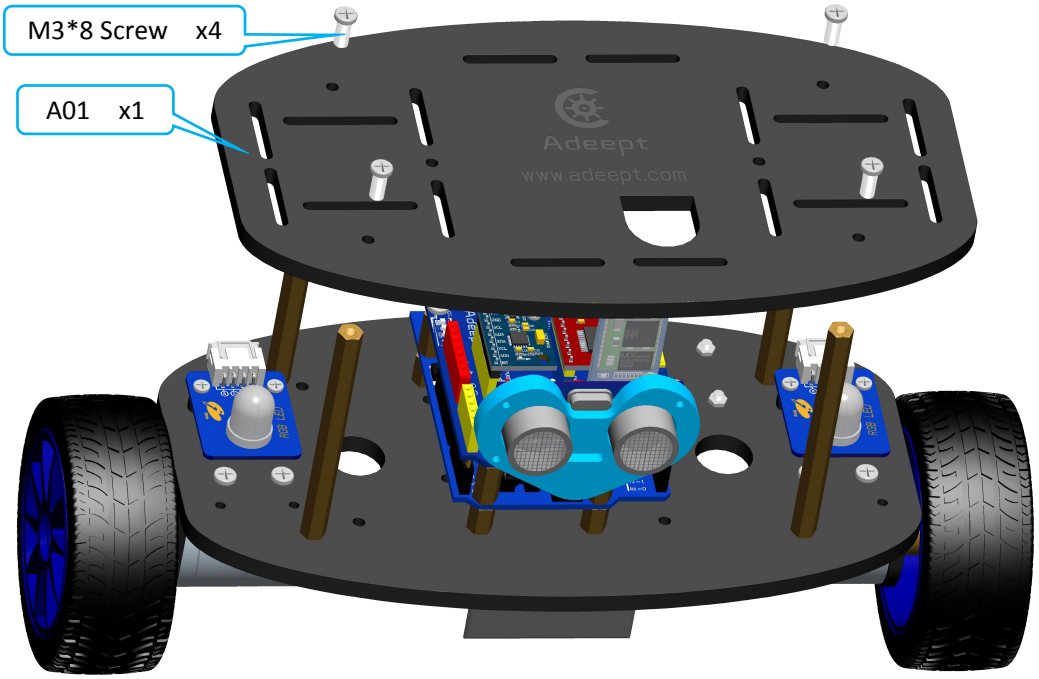


Identify the two motors based on the position by naming them A and B for the convenient wiring later.

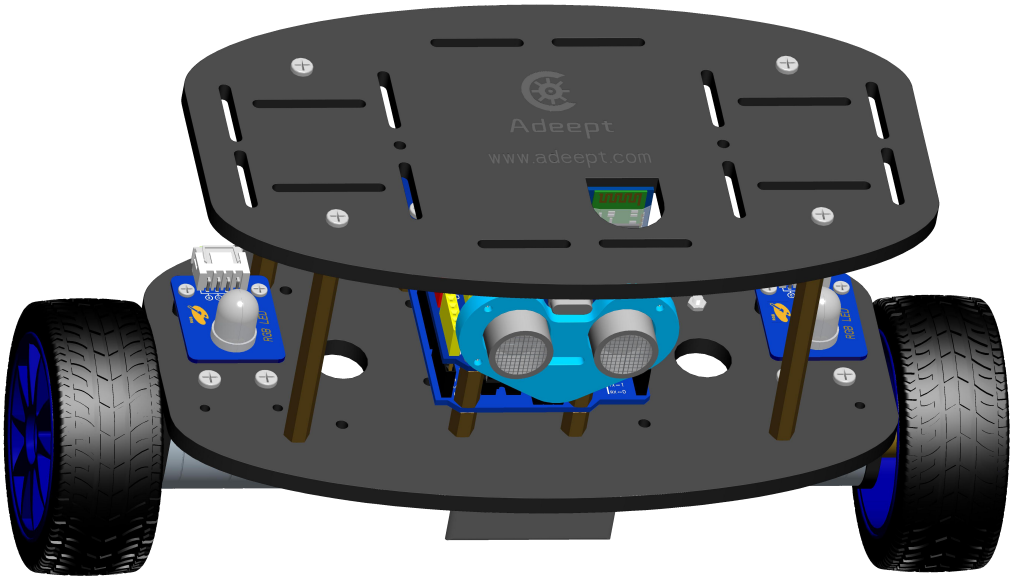
After Step J is completed, connect each module and motor to the Adeept Self-Balancing Robot Shield. with the cables provided, and then continue to assemble.

K. Fix the A01 plate.

Assemble the following components

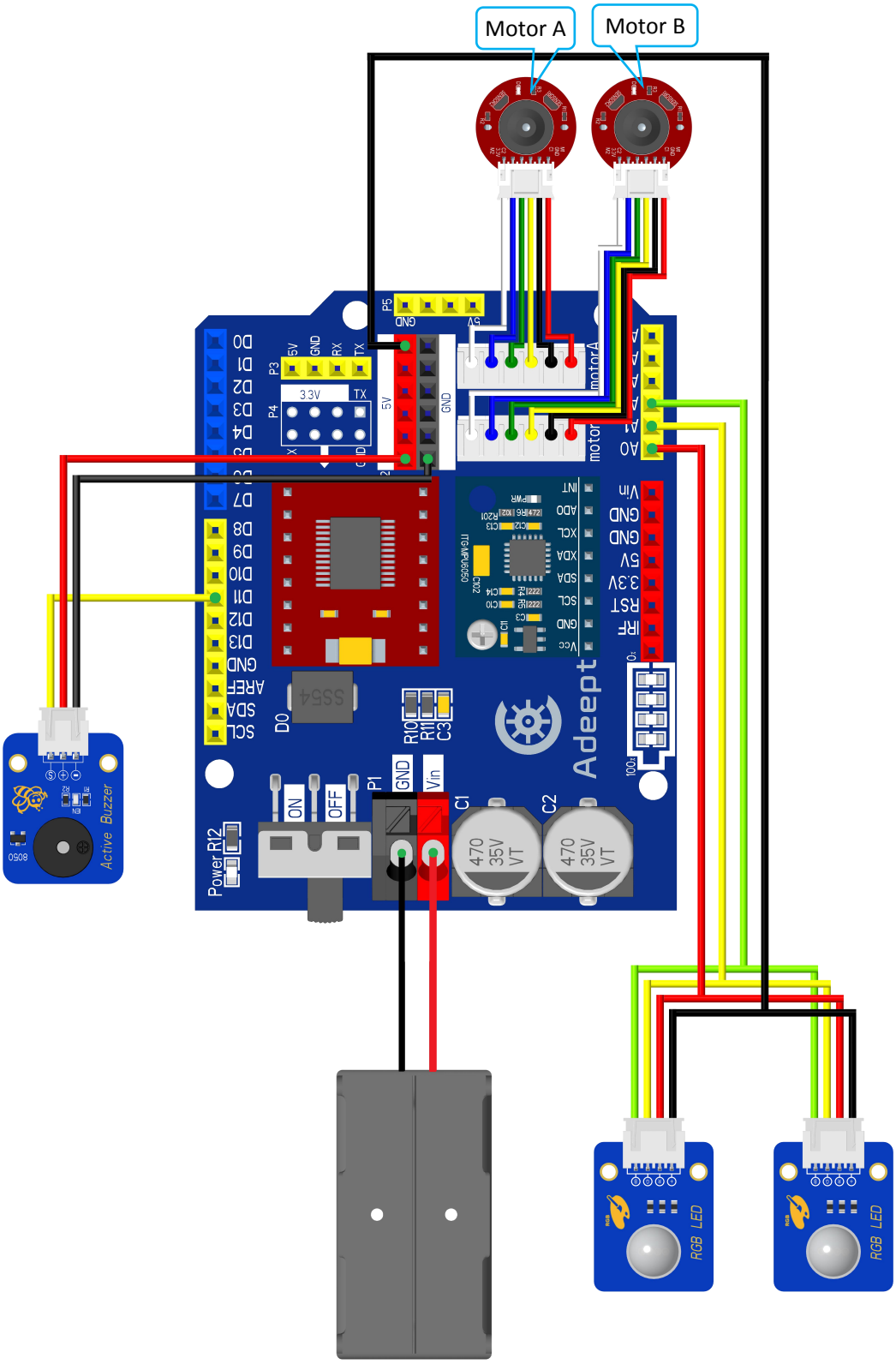


Effect diagram after assembling



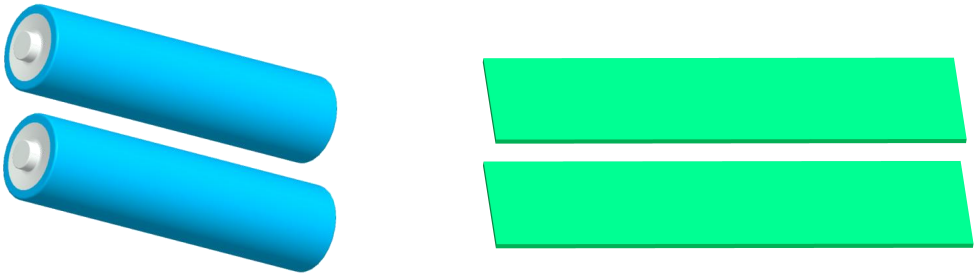
Circuit Connection

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

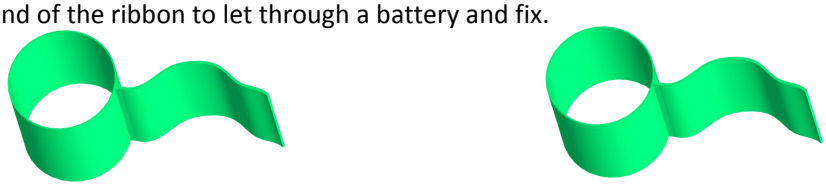


Install and Remove Batteries

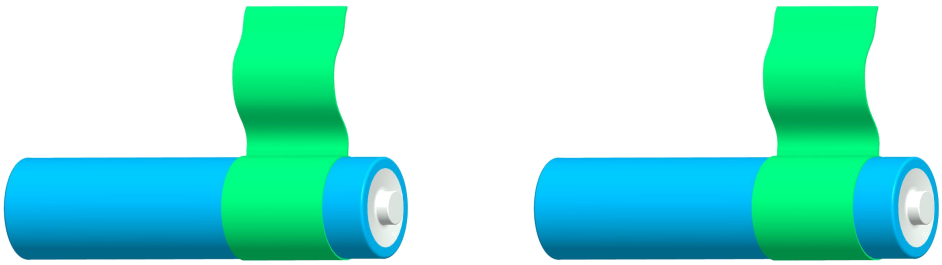
Take out 2 ribbons and 2 batteries.



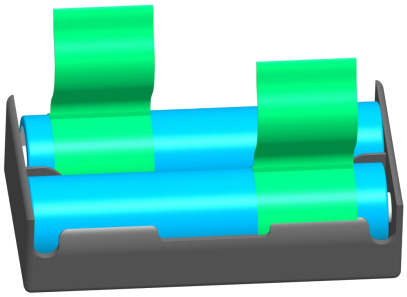
Roll one end of the ribbon to let through a battery and fix.



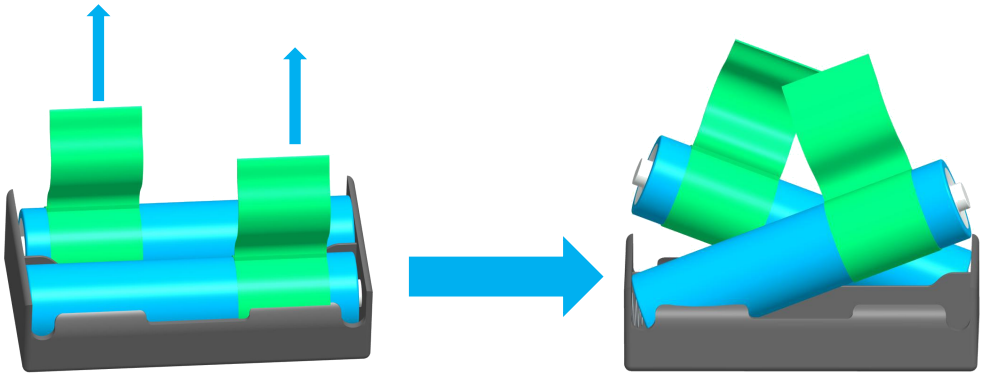
Insert the batteries into the rings - ribbon closer to the anode.



Install the batteries into the holder based on the pole.



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



Software & Hardware

What is Arduino?

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

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

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

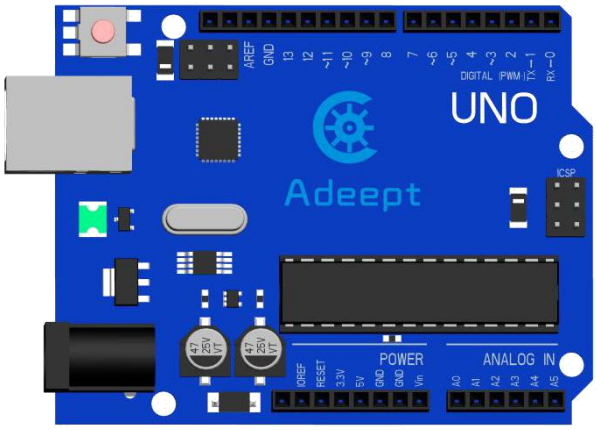
Why Arduino?

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

How Should I Use Arduino?

If you are a beginner with Arduino, Arduino learning kits on our website www.adeept.com would be a prefect step into this fantastic field!

One type of Arduino board are used in this car kit: Adeept UNO R3 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. We don't advise it.

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 5V/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.

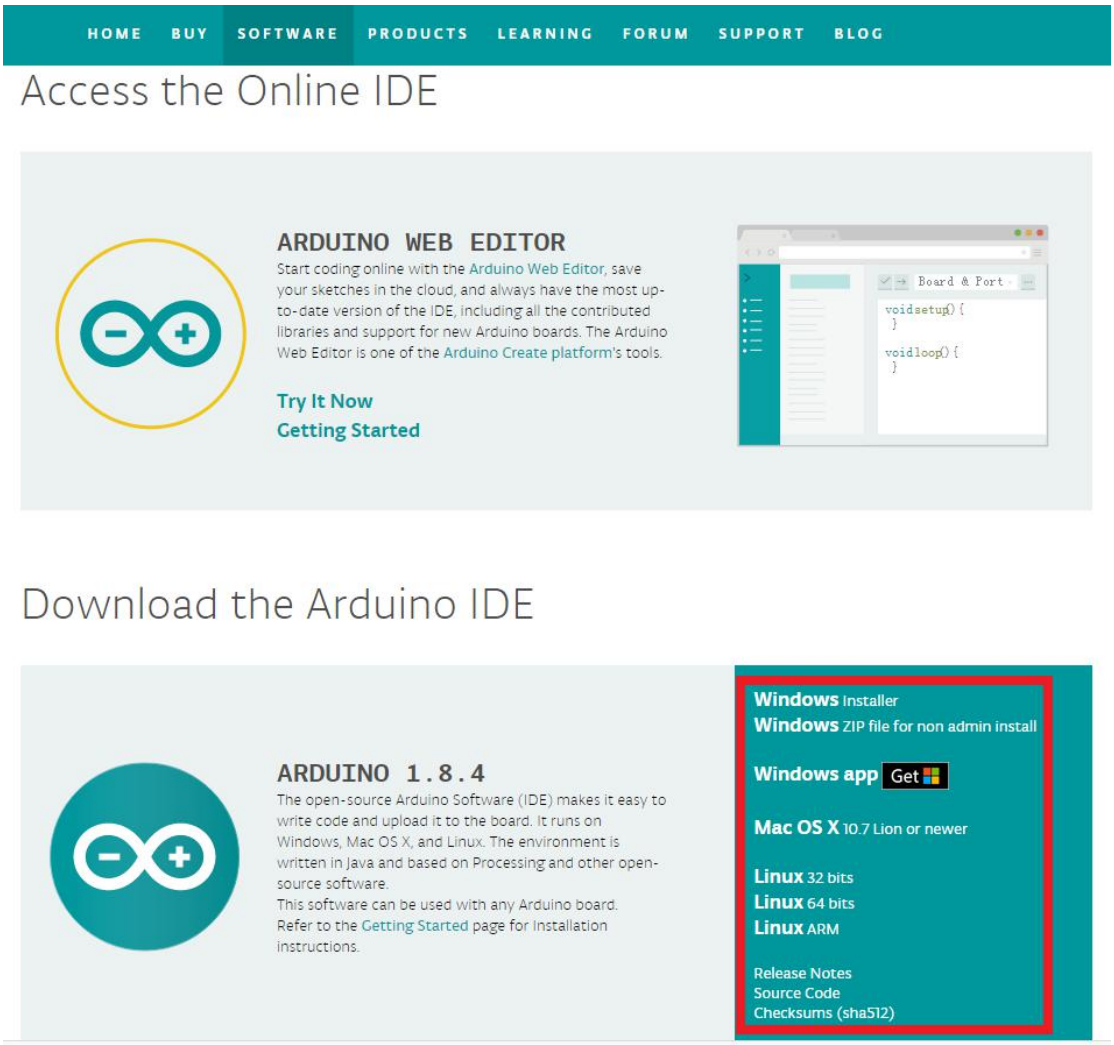
There are a couple of other pins on the board:

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)

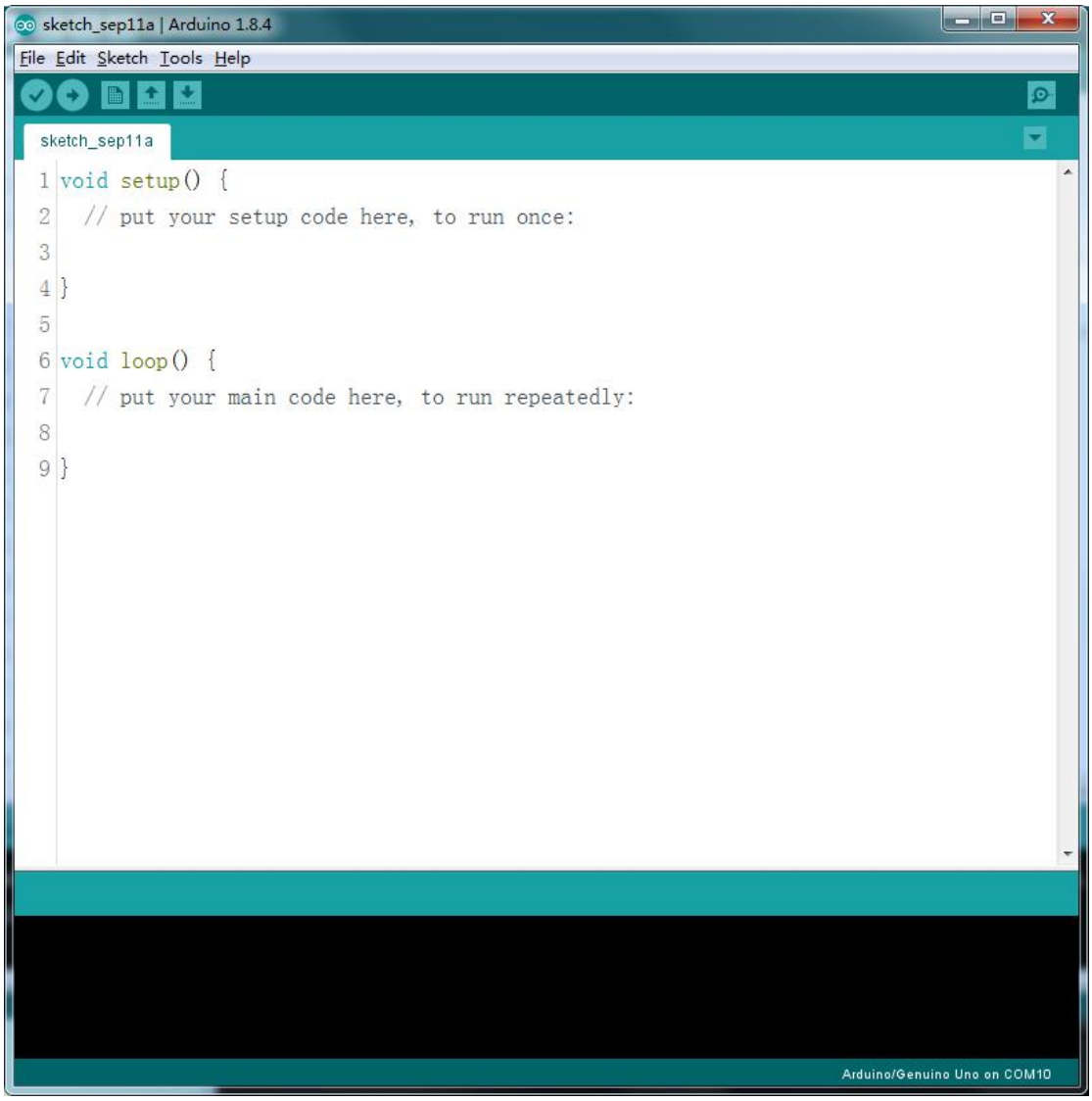
Arduino Software (IDE) is used to write and upload the code for Arduino Board. First, install Arduino software (IDE): visit https://www.arduino.cc/en/Main/Software. Download the corresponding installation program according to your operating system. If you are a Windows user, please select the “Windows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



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.

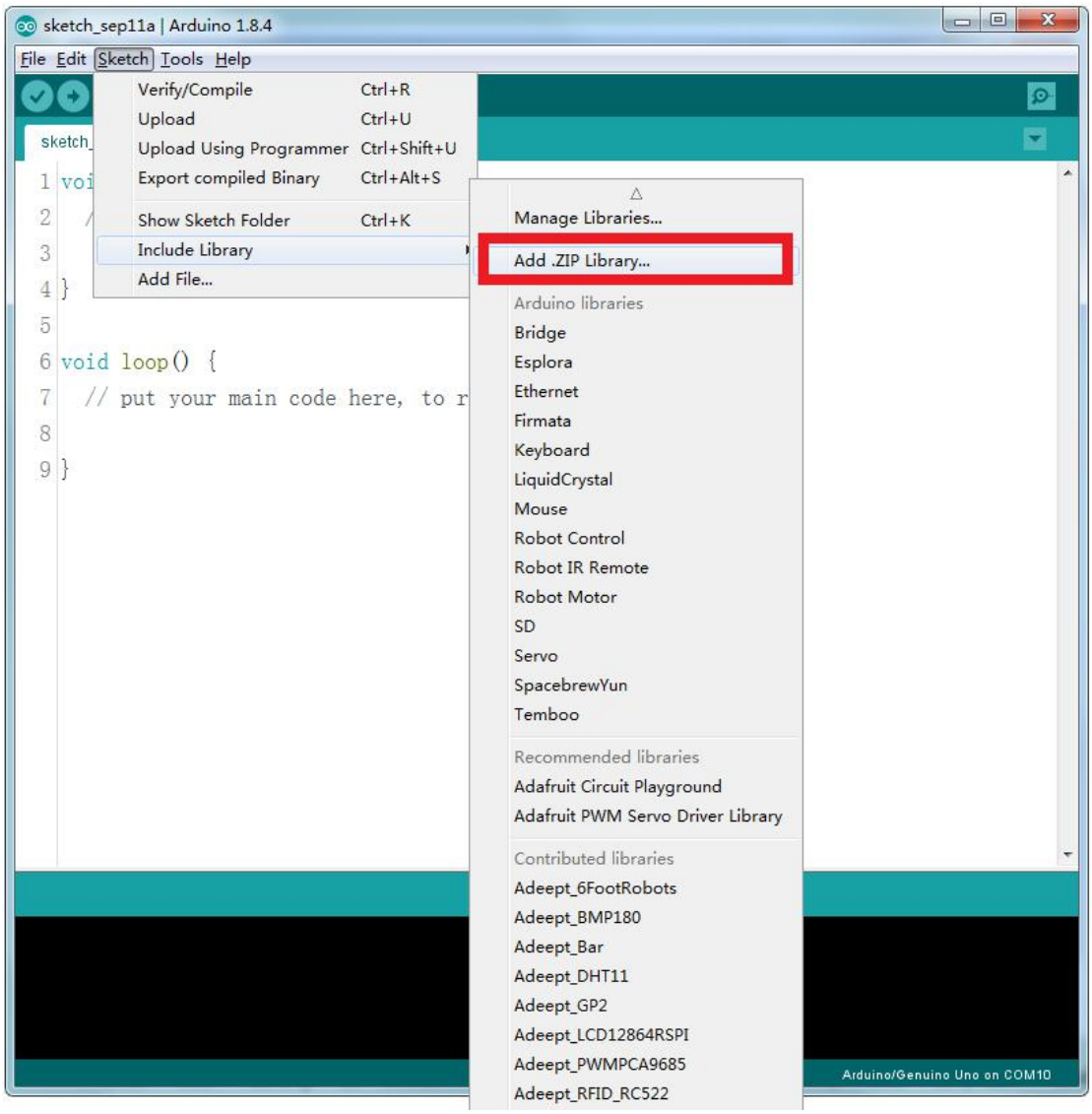
Serial Monitor: Opens the serial monitor.

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.

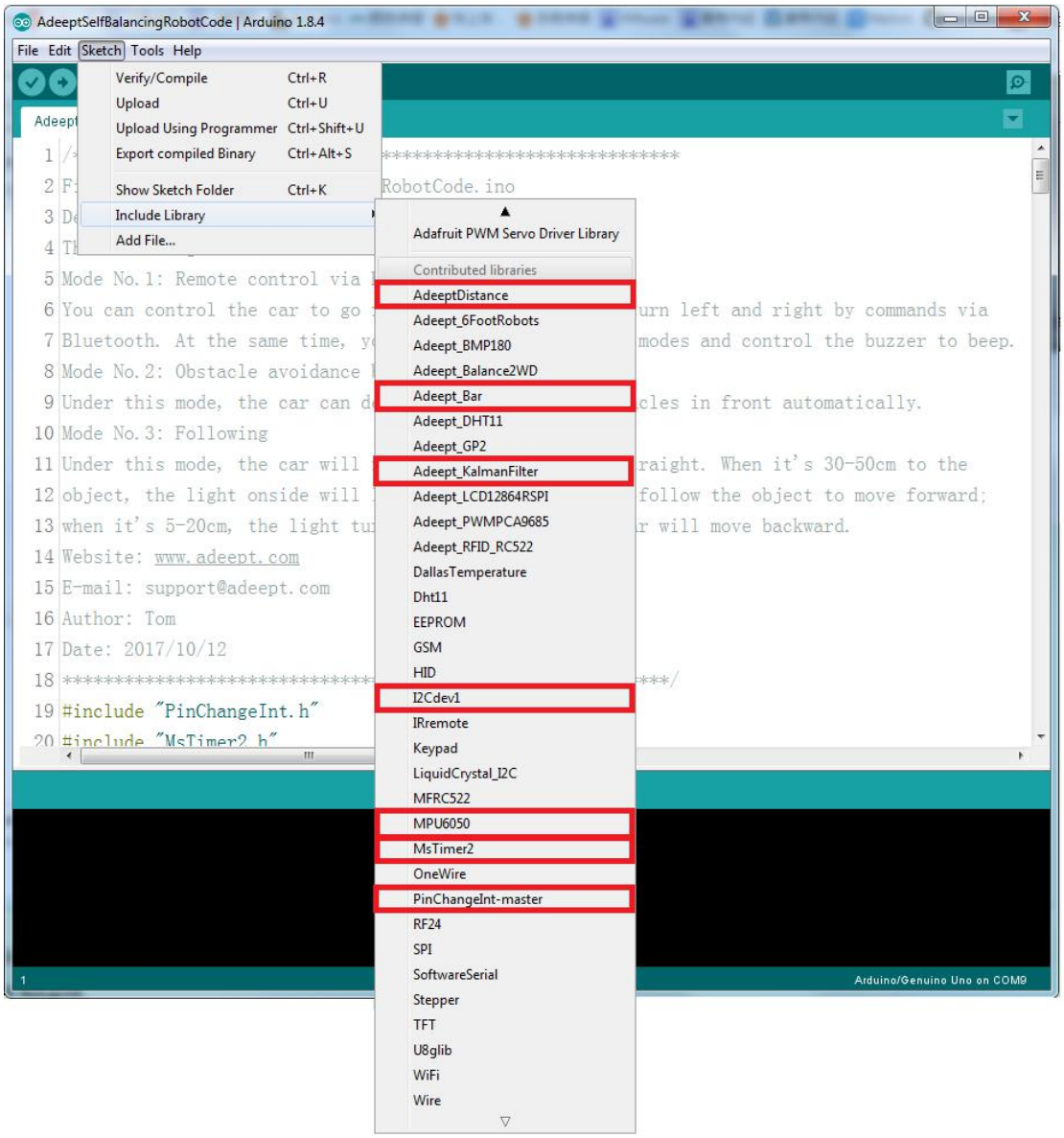
Since version 1.0, files are saved with an **.ino** file extension. Previous versions use the **.pde** extension. You may still open **.pde** named files in version 1.0 and later, and the software will automatically rename the extension to **.ino.**

Install Library

The example sketches provided use the *Adeept_Balance2WD.ZIP , Adeept_KalmanFilter.ZIP, AdeeptDistance.ZIP, I2Cdev1.ZIP, MPU6050.ZIP, MsTimer2.ZIP and PinChangeInt-master.ZIP* library, so you need to install it before compiling. Click **Add.ZIP Library** to add the *Adeept_Balance2WD.ZIP, Adeept_KalmanFilter.ZIP, AdeeptDistance.ZIP, I2Cdev1.ZIP, MPU6050.ZIP, MsTimer2.ZIP and PinChangeInt-master.ZIP* library to the *libraries* folder.



After the library is installed successfully, you can find the *Adeept_Balance2WD.ZIP, Adeept_KalmanFilter.ZIP, AdeeptDistance.ZIP, I2Cdev1.ZIP, MPU6050.ZIP, MsTimer2.ZIP and PinChangeInt-master.ZIP* library under **Sketch->Include Library**

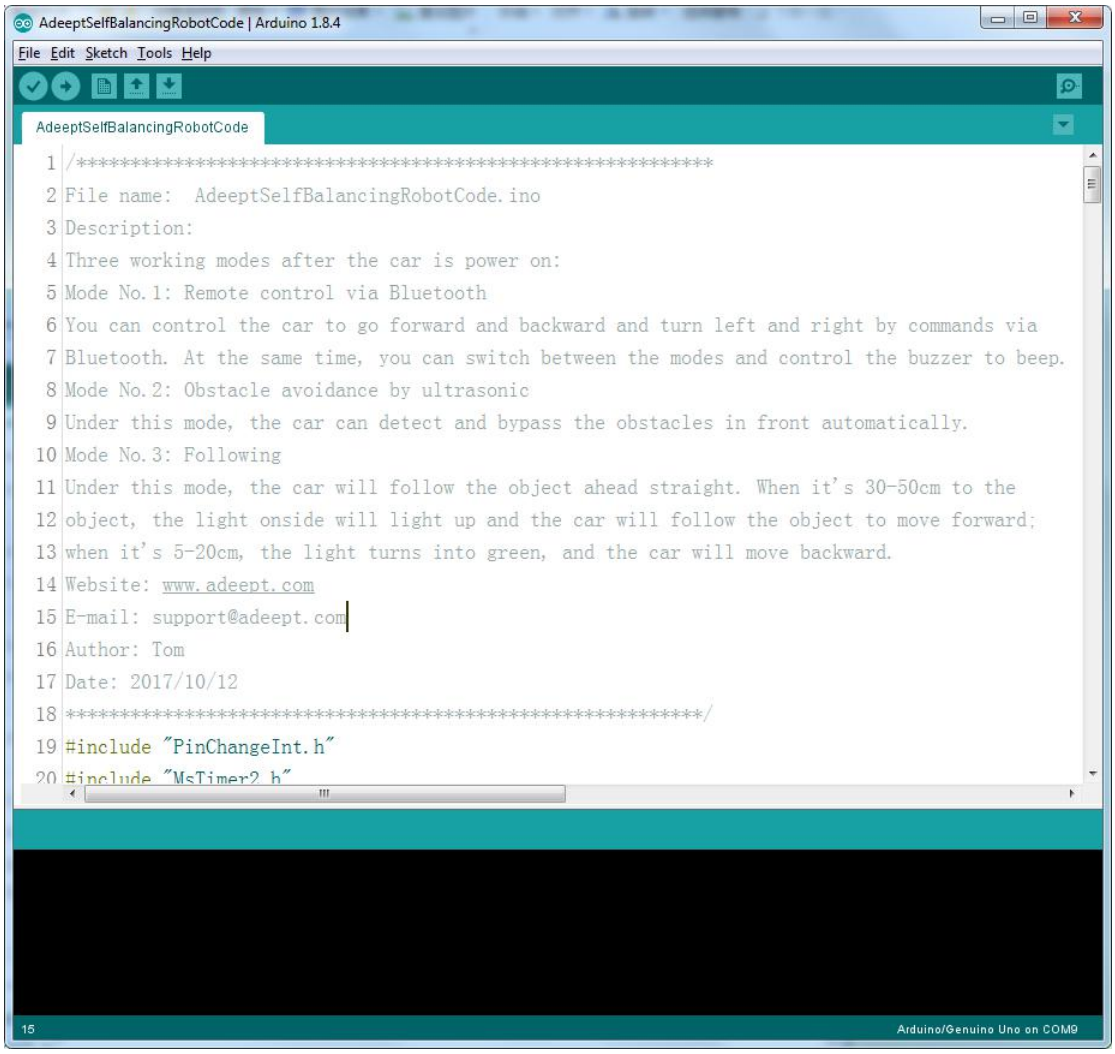


Upload Program

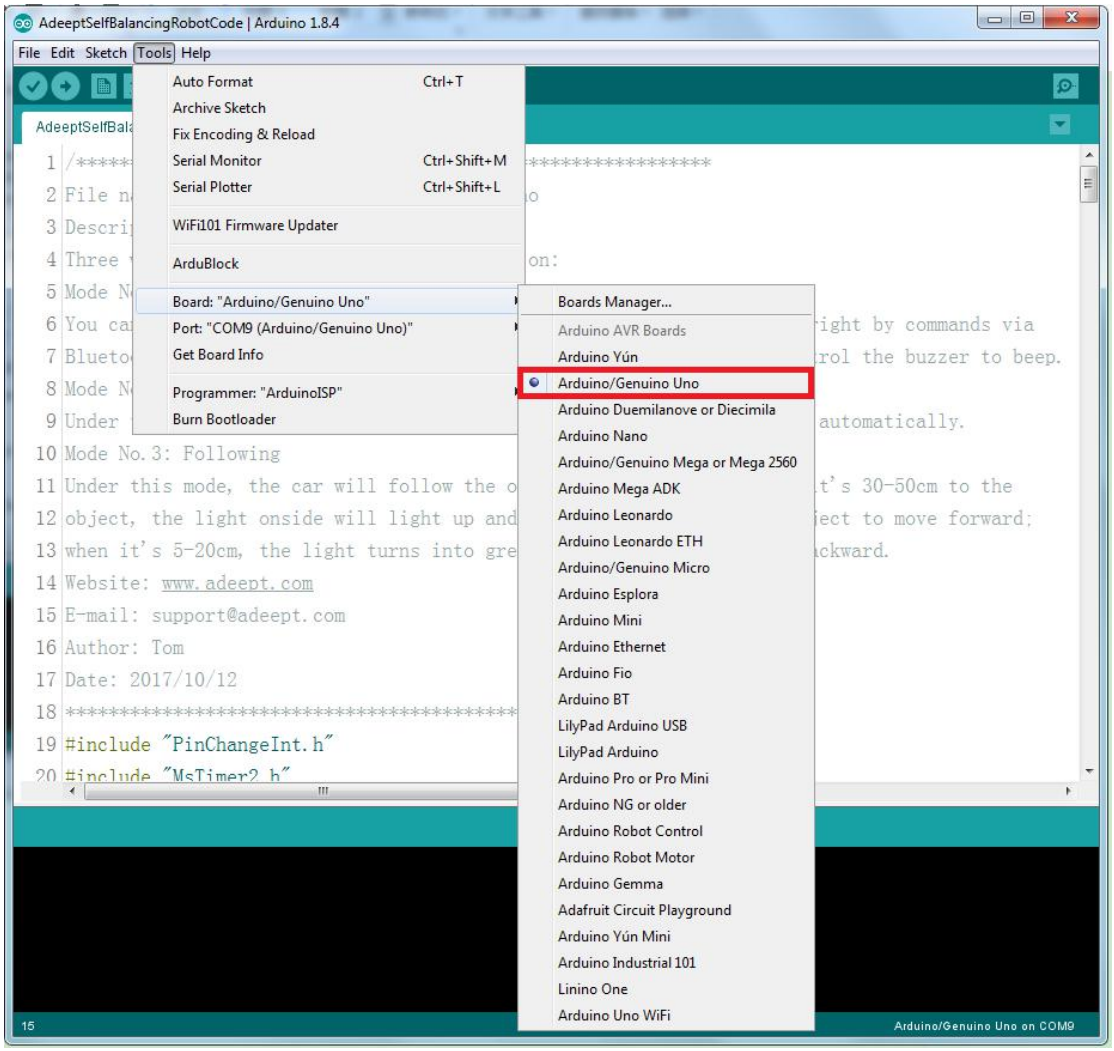
After the preparations above, next we will upload the program (example sketches provided) to the Adeept UNO R3 board.

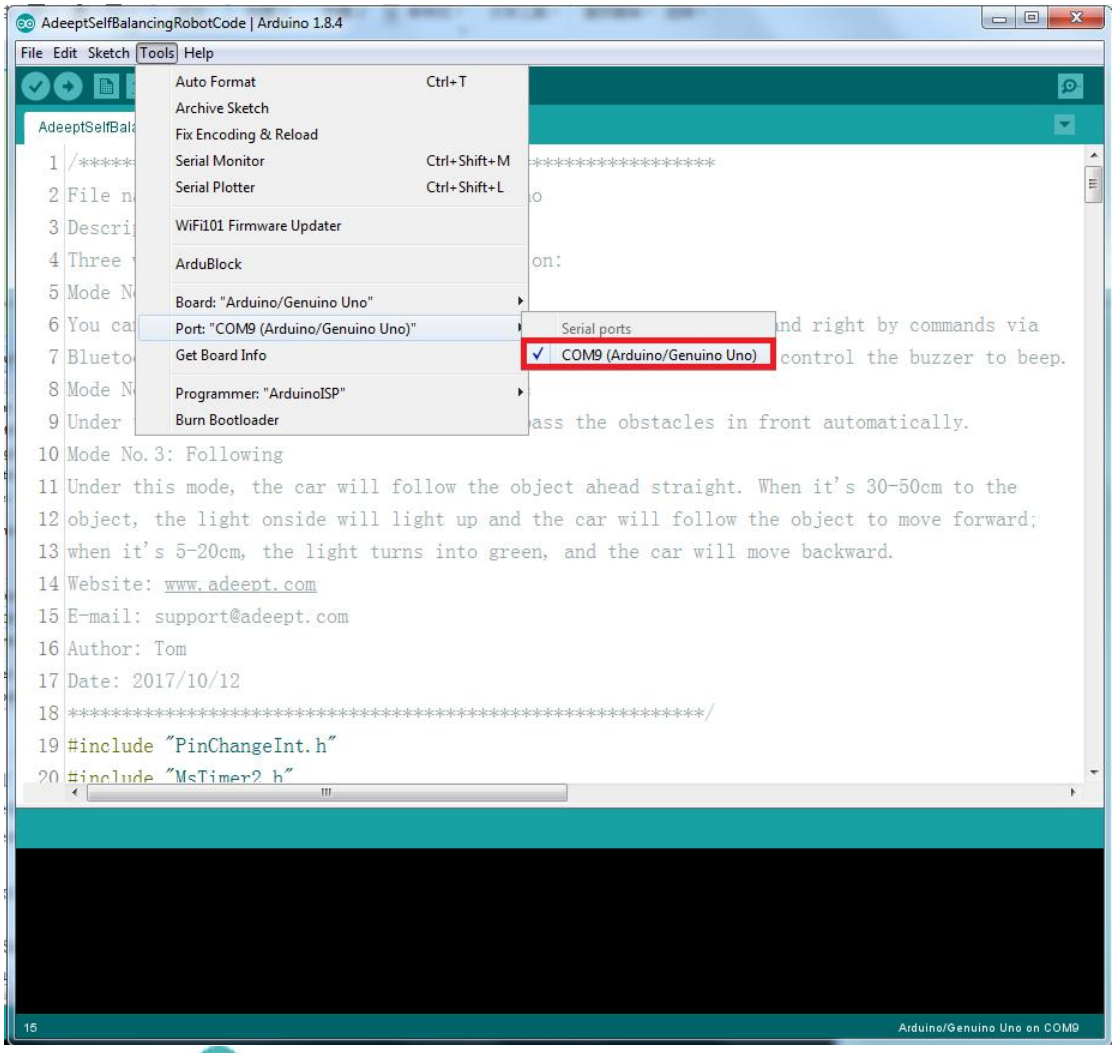
Note: Before downloading the program, please remove the Bluetooth module first, otherwise the download program will fail.

Open the program provided for the control board, the file "*AdeeptSelfBalancingRobotCode.ino*".



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


AdeeptSelfBalancingRobotCode | Arduino 1.8.4
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: "COM9 (Arduino/Genuino Uno)"
Get Board Info
Programmer: "ArduinoISP"
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
Mode No. 3: Following
Under this mode, the car will follow the o
object, the light onside will light up and
when it's 5-20cm, the light turns into gre
Website: www.adeept.com
E-mail: support@adeept.com
Author: Tom
Date: 2017/10/12
#include "PinChangeInt.h"
#include "MsTimer2.h"
Arduino/Genuino Uno on COM9



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