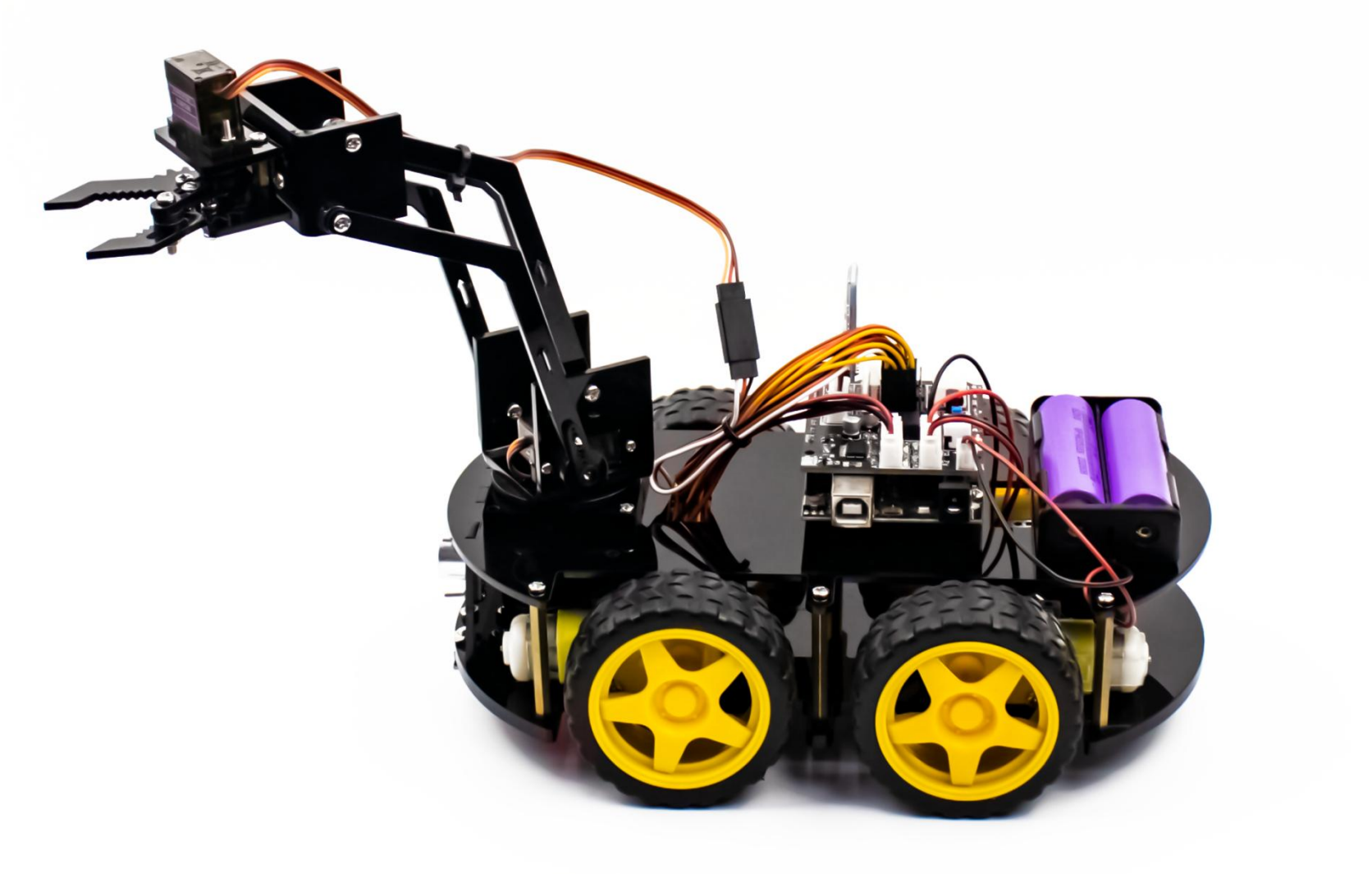
4WD Robot Arm Smart Car Kit



Content

LAFVIN

Company Profile

Established in 2011, LAFVIN is a manufacturer and trader specialized in research,development and production of Mega2560,UNO,Nano boards,and all kinds of accessories or sensors use for Arduino,raspberry.We also complete starter kits designed for interested lovers of any levels to learn Arduino or Raspberry.We are located in Shenzhen,China.All of our products comply with international quality standards and are greatly appreciated in a variety of different markets throughout the world.

Customer Service

We are cooperating with a lot of companies from different countries.Also help them to purchase electronic component products in china,and became the biggest supplier of them. We look forward to build cooperate with more companies in future.
By the way,We also look forward to hearing from you and any of your critical comment or suggestions.Pls email us by lafvin_service@163.com if you have any questions or suggestions.As a continuous and fast growing company. We keep striving our best to offer you excellent products and quality service.

Our Store

Aliexpress store: https://www.aliexpress.com/store/1942043 Brand in Amazon:LAFVIN

Product Catalog

https://drive.google.com/drive/folders/0BwvEeRN9dKllblZING00TkhYbGs?usp=sharing

Tutorial

This tutorial include codes,libraries,lessons and installation guide video. It is designed for beginners. It will teach every users how to assembly the robot car and use Arduino UNO controller board, sensors,servo and Bluetooth module. Provide 3D dynamic installation tutorial to quickly build your robot car. The power supply uses two 18650 lithium batteries with long-lasting battery life.The newly created APP control software, the 7 major functions of the robot car kit are comprehensively controlled by the APP, and the function mode can be switched freely.

Package List

R3 x1

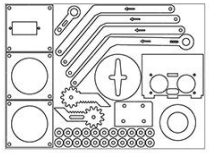

Expanding Board x1

Remote x1

Ultrasound x1

MG90S Servo x3


Bluetooth x1


Acrylic A1 x1


Line Tracking Module x3


Motor x4


Tire x4

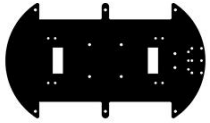

Acrylic A 2 x1


Acrylic A 3 x1


Acrylic A 4 x1


USB Cable x1


Battery Case x1


Bottom Acrylic Layer x1


Upper Acrylic Layer x1


Adhesive Tape x1


AXK Plain Bearing x1


Cylinder Screwdriver x1


Tie x3


Screw bag


Screwdriver x1

CD Tutorail x1


4pin XH2.54 to F Dupont Wire x1


3pin Servo Extension Cable x1

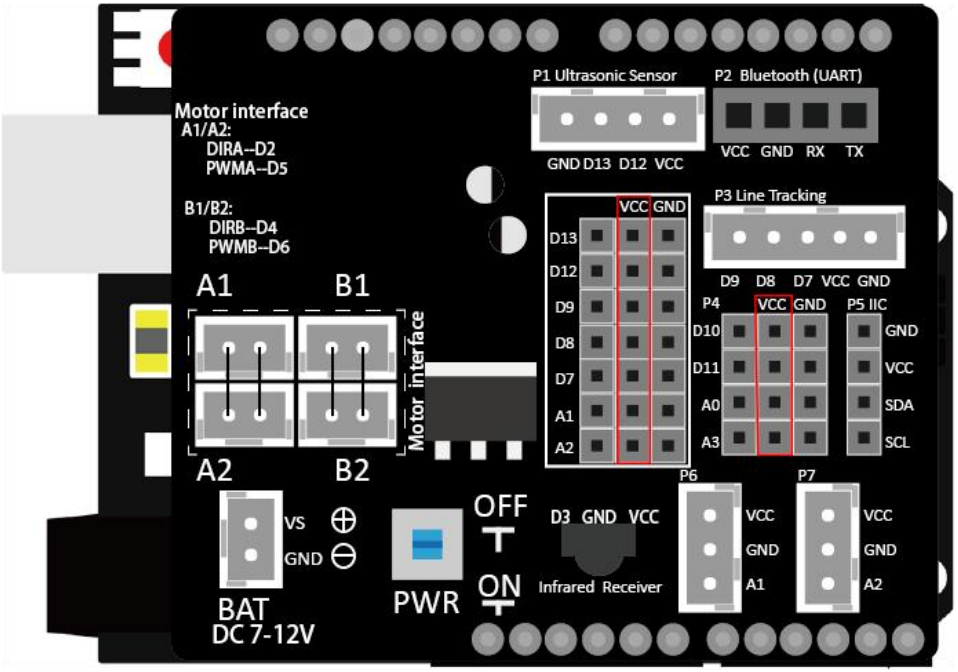

3pin F-F Dupont Wire x3

LAFVIN 4WD Robot Arm Smart Car Kit Introduction

4WD Robot Arm Smart Car Kit is mainly manufactured using the Arduino UNO R3 main control and the TB6612 Motor drive expansion board, two reduction motors, Acrylic frame chassis and some sensors.

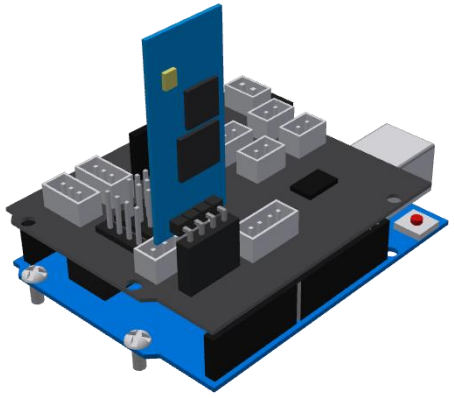
LAFVIN 4WD Robot Arm Smart Car Kit have the following advantages:

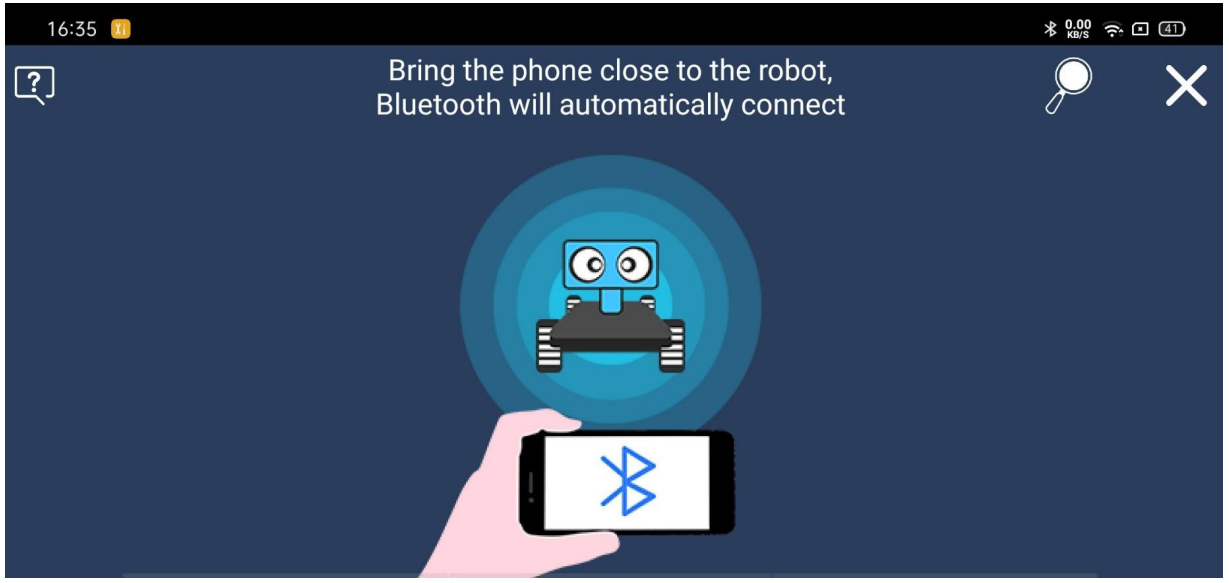
1)Arduino motor drive shield integrates the driver chip, eliminating the traditional complex wiring and installation space.The shield uses standard interface wiring, simple and easy to understand, fast installation wiring.



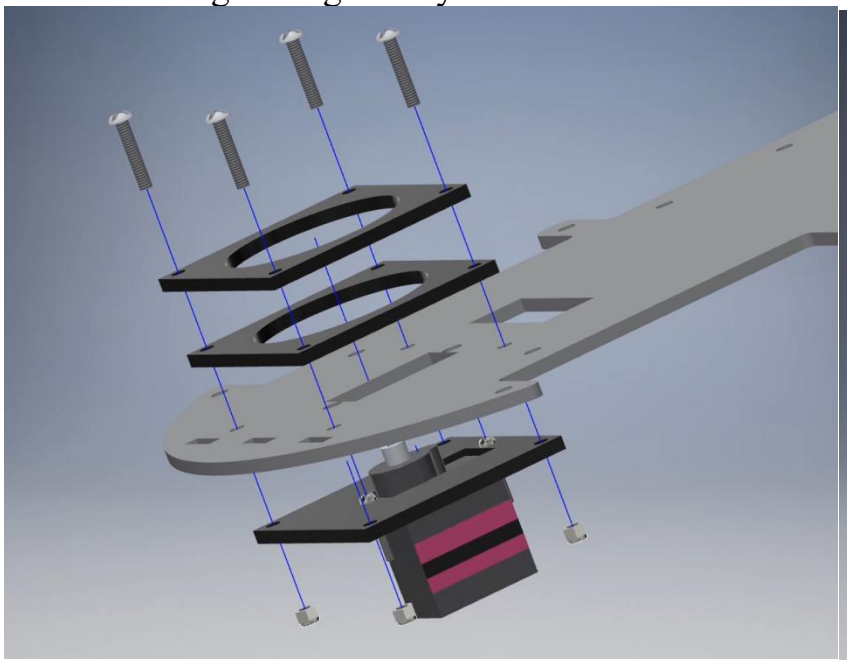
2) The newly created APP control software, the 7 major functions of the robot car kit are comprehensively controlled by the APP, and the function mode can be switched freely.Click to 7-in-1 Multi-Function Robot Arm Smart Car.

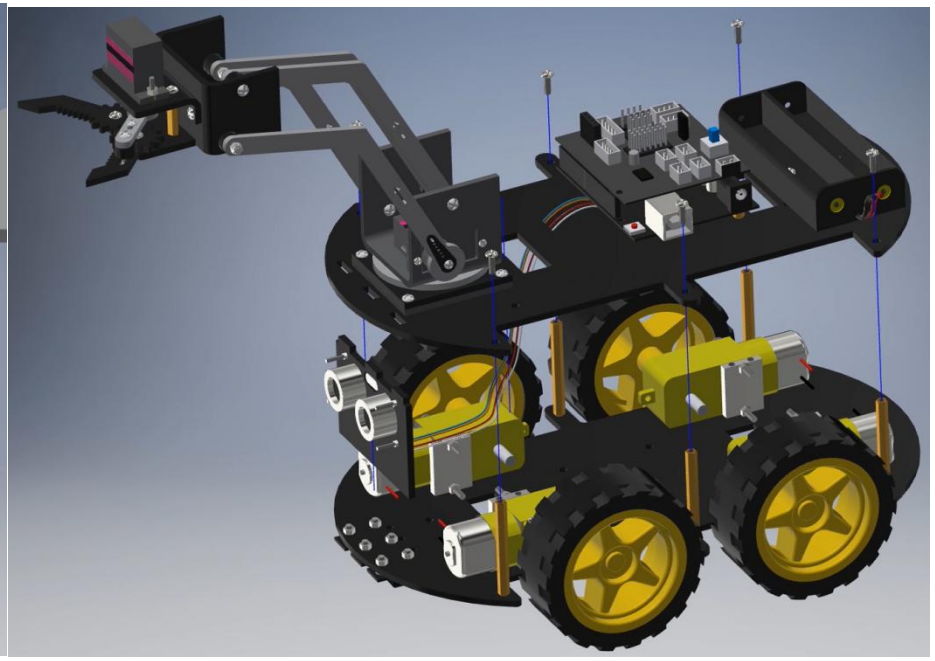
3) The newly designed Bluetooth module is connected to the circuit, and the upload program does not require manual disconnection of the Bluetooth module. Help beginner learners to successfully avoid the trap of uploading program failure. Connect the power to the Bluetooth module, then open the app remote control application, and bring the phone close to the Bluetooth module and it will automatically connect. No longer need the cumbersome connection process before.





4)Provide 3D dynamic installation tutorial to quickly build your robot car. The power supply uses two 18650 lithium batteries with long-lasting battery life





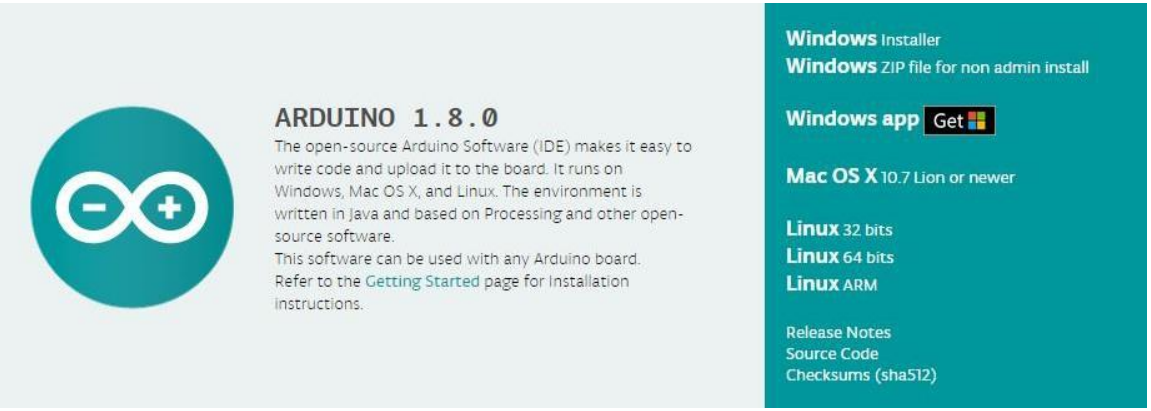
Lesson 1 Getting Started with Arduino IDE

How to Install Arduino IDE

Introduction

The Arduino Integrated Development Environment (IDE) is the software side of the Arduino platform.In this Project, you will learn how to setup your computer to use Arduino and how to set about the Projects that follow.The Arduino software that you will use to program your Arduino is available for Windows, Mac and Linux. The installation process is different for all three platforms and unfortunately there is a certain amount of manual work to install the software.

STEP 1: Go to [https://www.arduino.cc/en/Main/Software and find below page.](https://www.arduino.cc/en/Main/Software)



The version available at this website is usually the latest version, and the actual version may be newer than the version in the picture.

STEP2：Download the development software that is compatible with the operating system of your computer.
Take Windows as an example here.

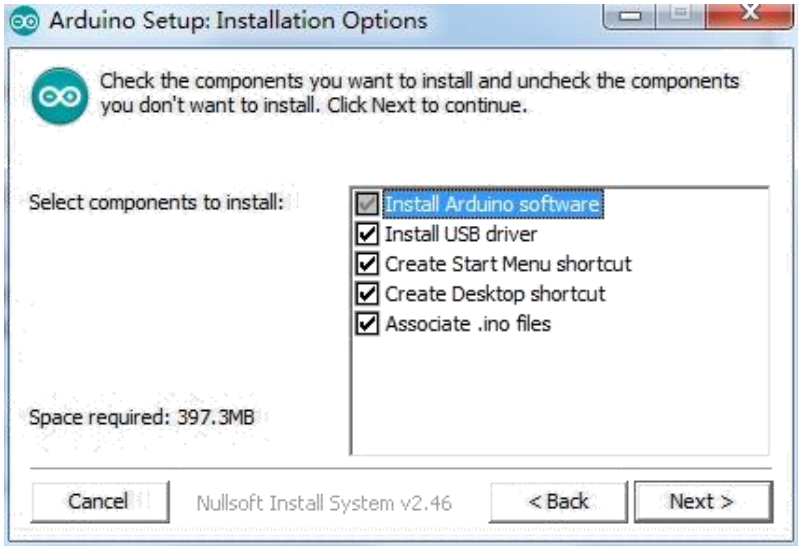
Click Windows Installer.

Click JUST DOWNLOAD.
Also version 1.8.0 is available in the material we provided, and the versions of our materials are the latest versions when this course was made.

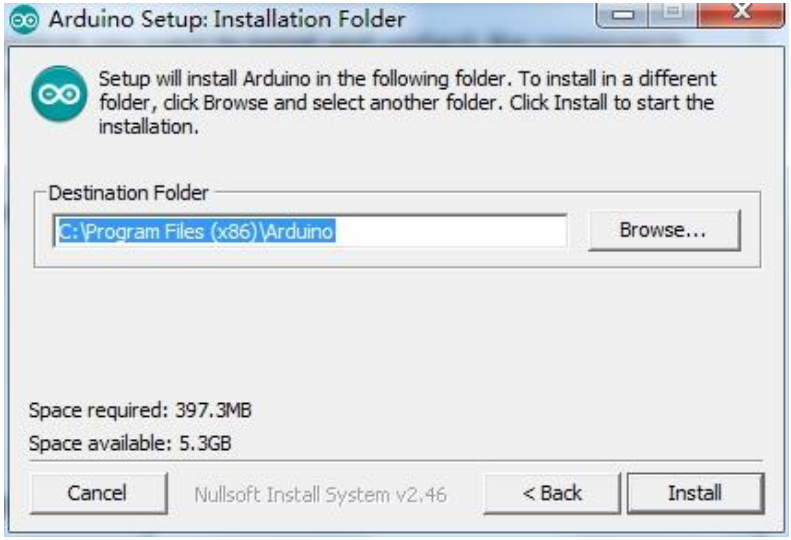
Installing Arduino (Windows)

Install Arduino with the exe. Installation package.

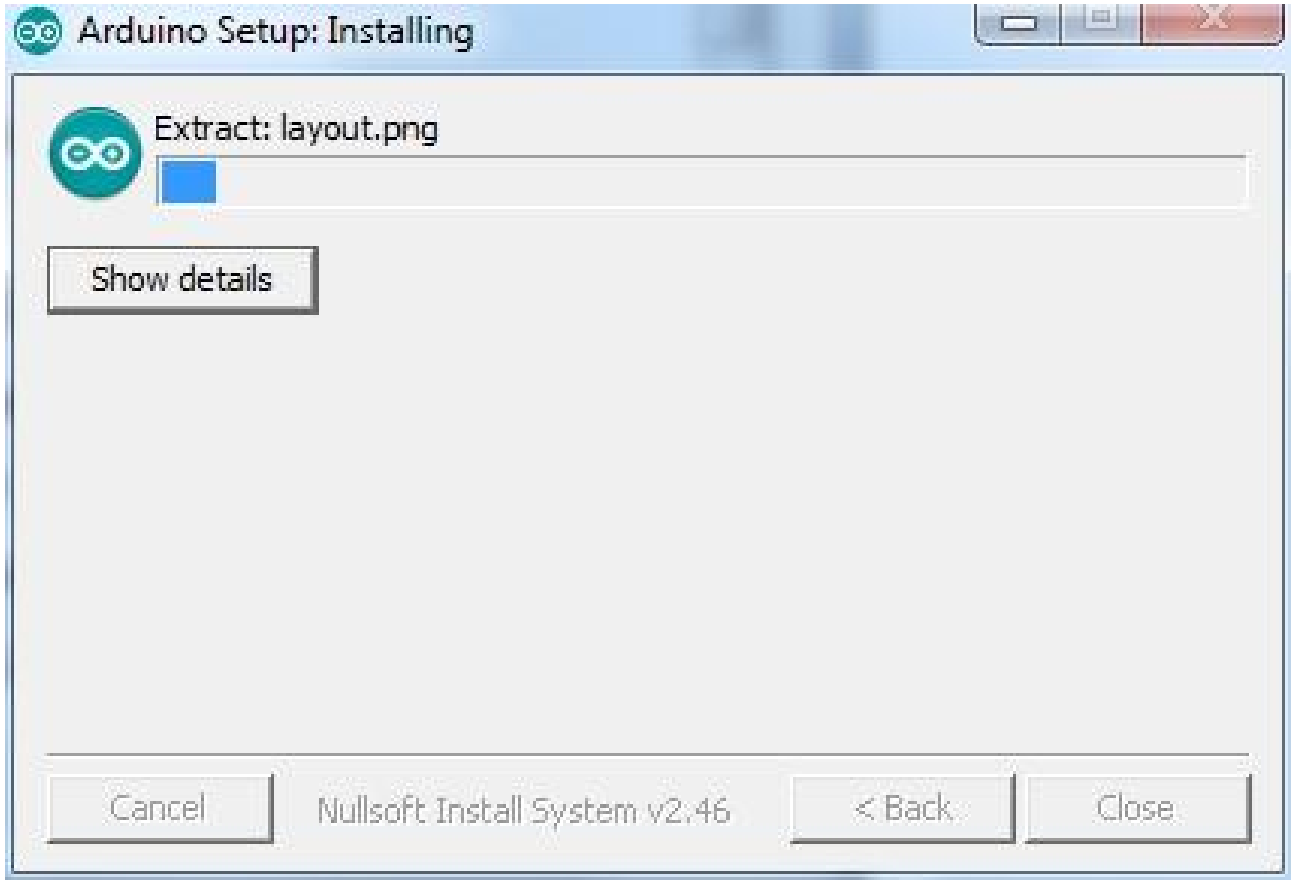
Click I Agree to see the following interface


Arduino Setup: Installation Options
Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.
Select components to install:
Install Arduino software
Install USB driver
Create Start Menu shortcut
Create Desktop shortcut
Associate .ino files
Space required: 397.3MB
Cancel
Nullsoft Install System v2.46
< Back
Next >

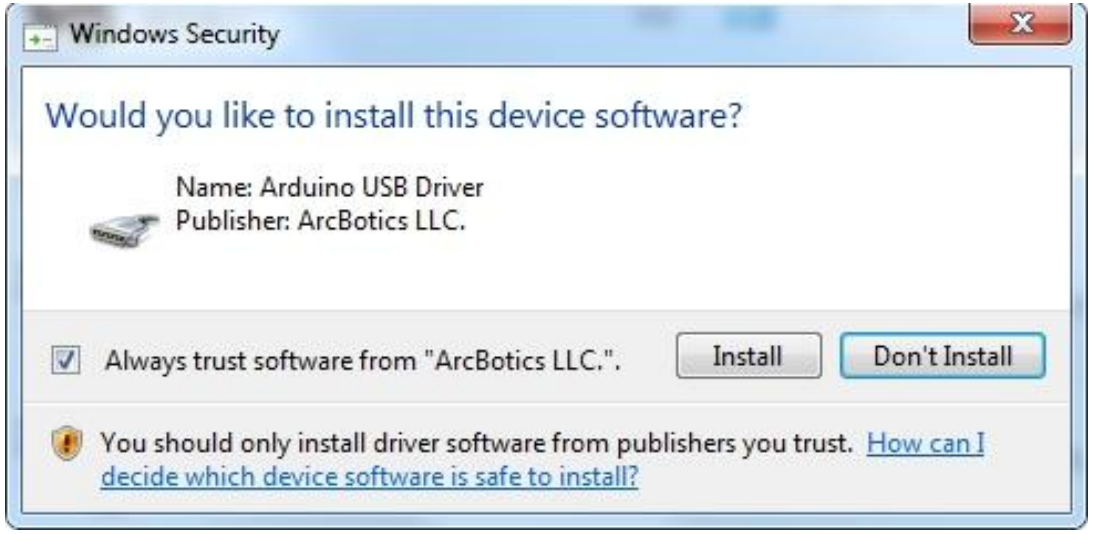
Click Next


Arduino Setup: Installation Folder
Setup will install Arduino in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.
Destination Folder
C:\Program Files (x86)\Arduino
Browse...
Space required: 397.3MB
Space available: 5.3GB
Cancel
Nullsoft Install System v2.46
< Back
Install

You can press Browse… to choose an installation path or directly type in the directory you want.
Click Install to initiate installation

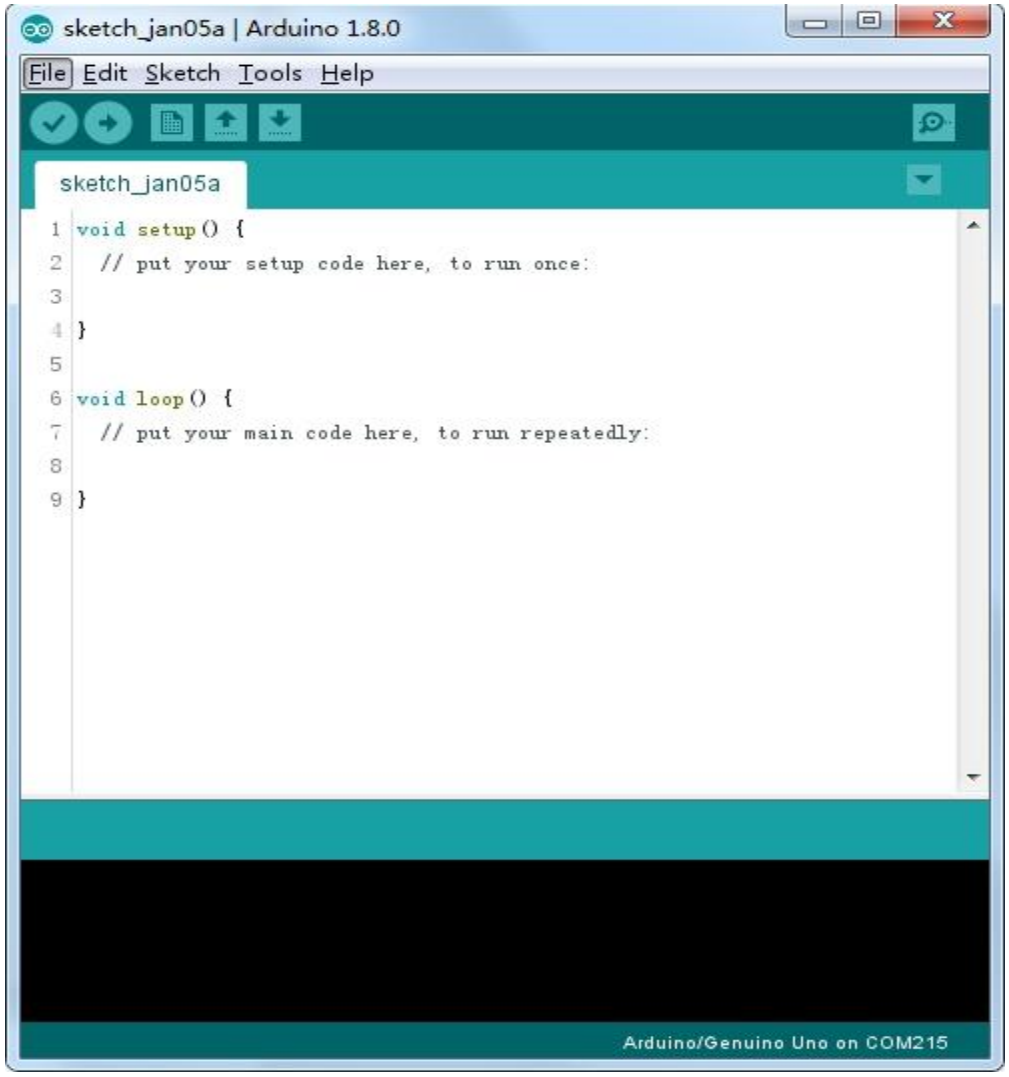


Wait for the installing process, if appear the interface of Window Security, just continue to click Install to finish the installation.



Next, the following icon appears on the desktop

Double-click to enter the desired development environment

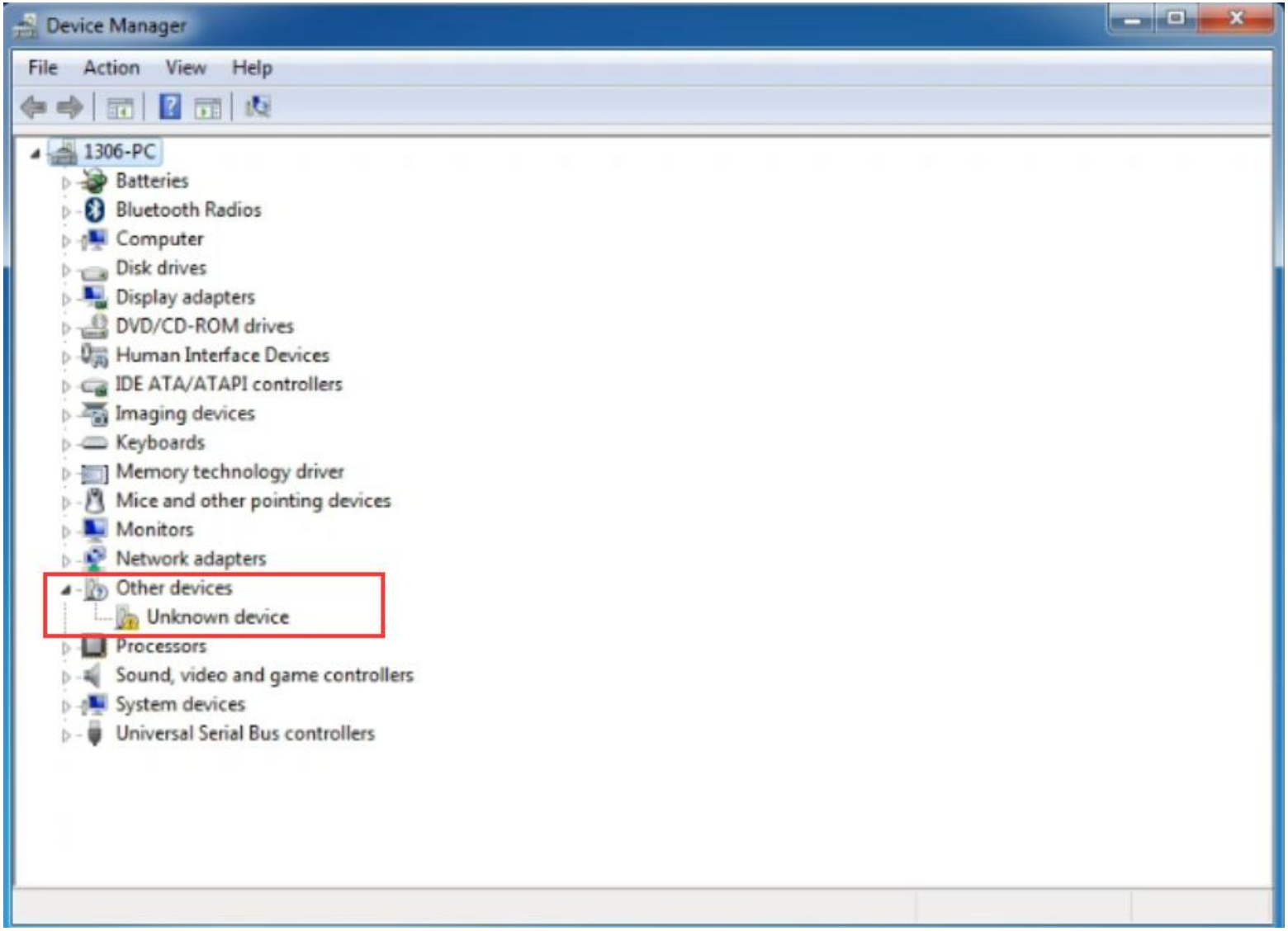


How to Install Arduino Driver

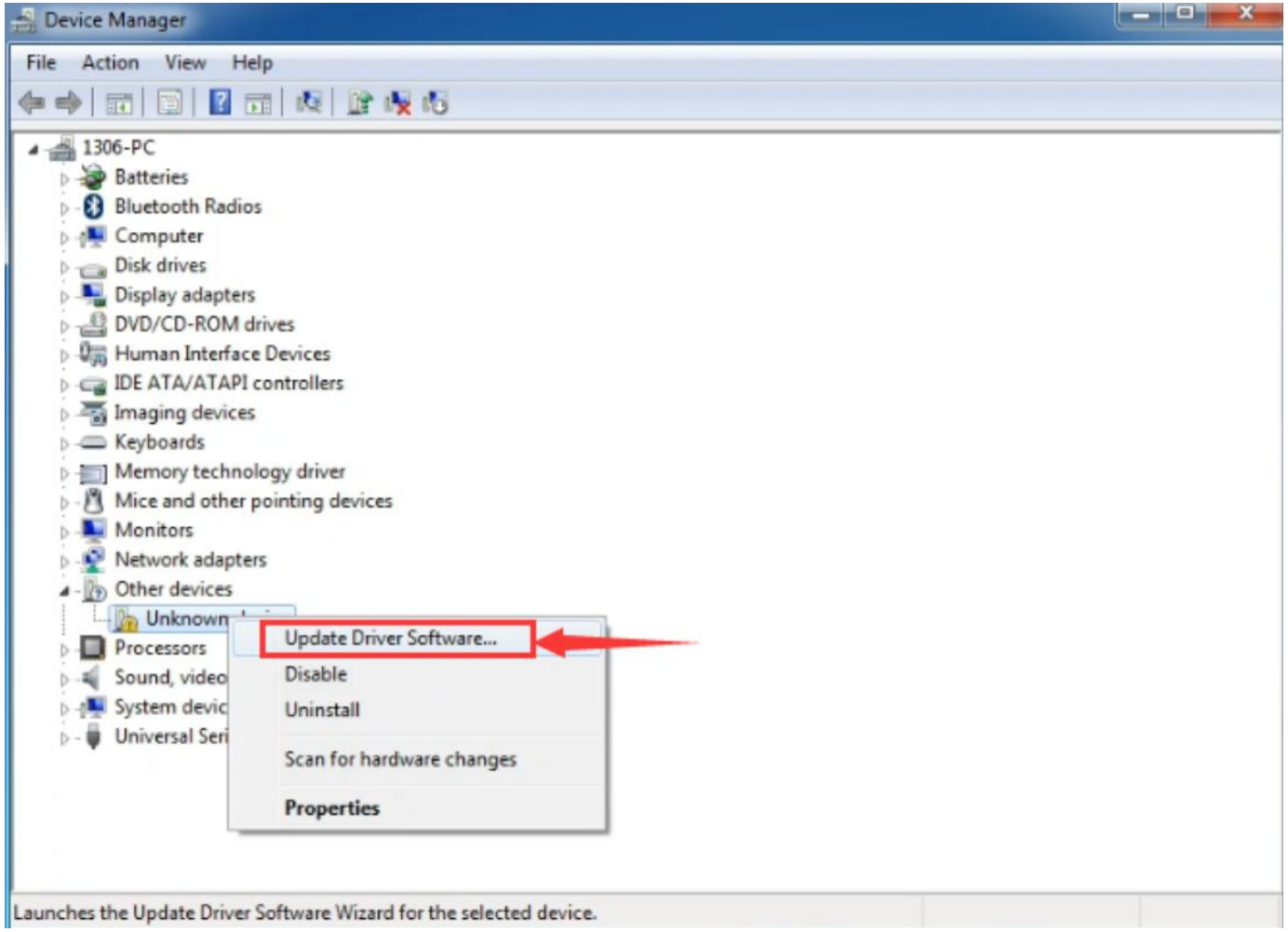
Next, we will introduce the driver installation of UNO R3 development board. The driver installation may have slight differences in different computer systems. So in the following let's move on to the driver installation in the Window system.

The Arduino folder contains both the Arduino program itself and the drivers that allow the Arduino to be connected to your computer by a USB cable. Before we launch the Arduino software, you are going to install the USB drivers.

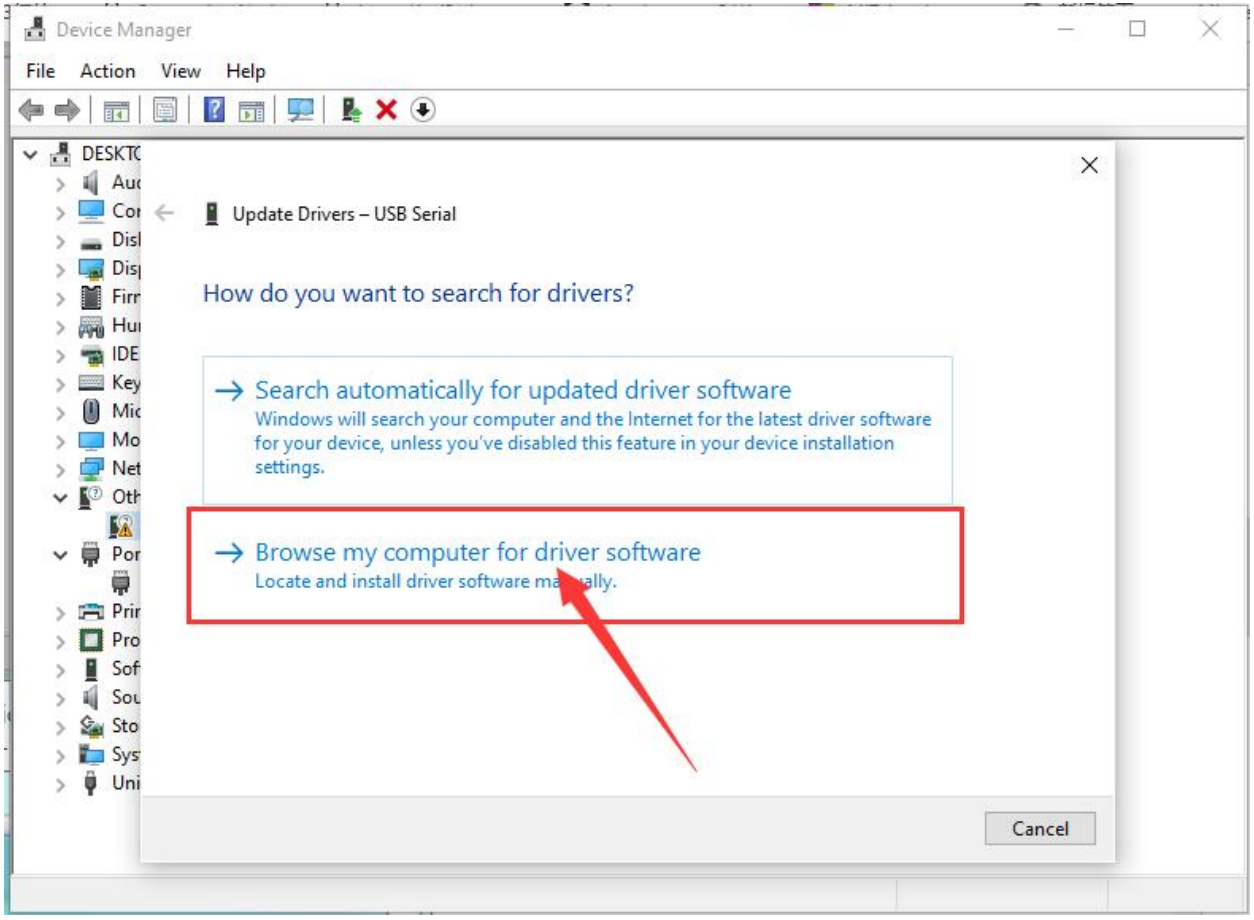
When you connect UNO board to your computer at the first time, right click the icon of your
"Computer" —>for "Properties"—> click the "Device manager" ,
under "Other Devices"or"USB-Serial", you should see an icon for "Unknown device" with a little yellow warning triangle next to it. This is your Arduino.Or you can search for "device" in your computer, or you can open the device manager of your computer.


Device Manager
File Action View Help
1306-PC
Batteries
Bluetooth Radios
Computer
Disk drives
Display adapters
DVD/CD-ROM drives
Human Interface Devices
IDE ATA/ATAPI controllers
Imaging devices
Keyboards
Memory technology driver
Mice and other pointing devices
Monitors
Network adapters
Other devices
Unknown device
Processors
Sound, video and game controllers
System devices
Universal Serial Bus controllers

Then right-click on the device and select the top menu option (Update Driver Software...) shown as the figure below.

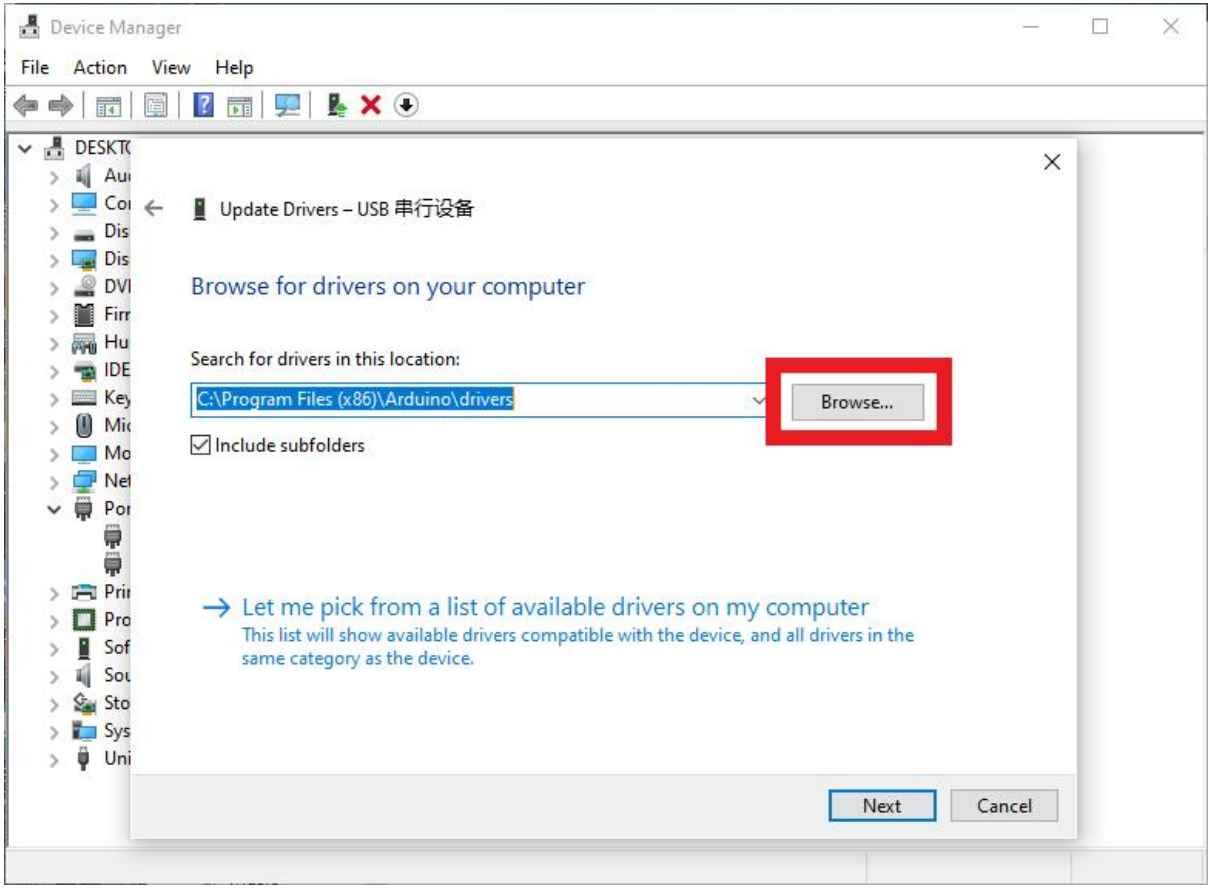


Then it will be prompted to either “Search Automatically for updated driver software” or “Browse my computer for driver software”. Shown as below. In this page, select “Browse my computer for driver software”.



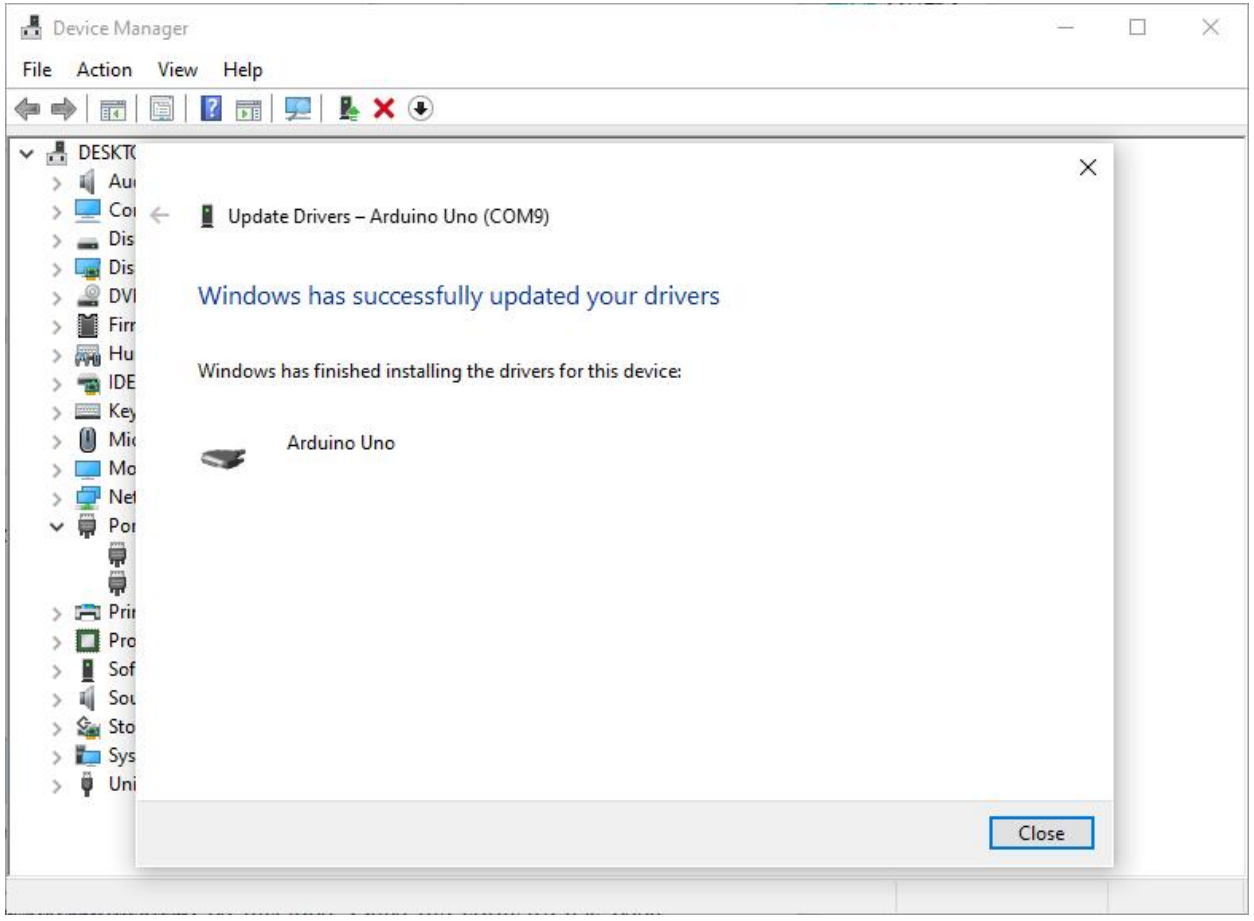
Right-click on the device and select the top menu option (Update Driver Software...).

You will then be prompted to either ‘Search Automatically for updated driver software’ or ‘Browse my computer for driver software’. Select the option to browse and navigate to the :**C\Program Files(x86)\Arduino\drivers.**(Note: Here is the path you choose to install arduino IDE. The path chosen in the installation tutorial in the previous section is that, so the path I chose is**C\Program Files(x86)\Arduino\drivers**)

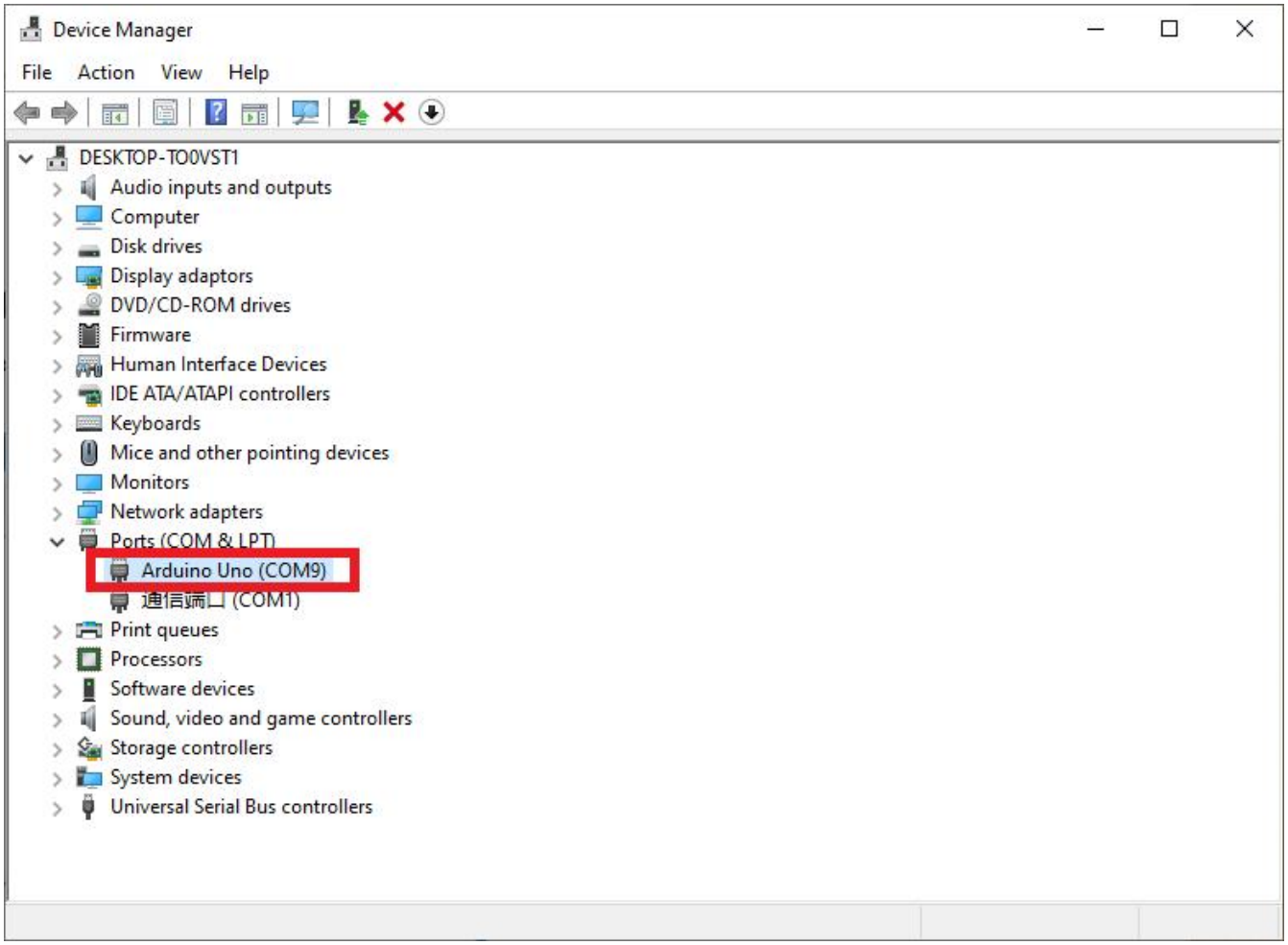


Click “Next” and you may get a security warning, if so, allow the software to be installed.

Once the software has been installed, you will get a confirmation message. Installation completed,click “Close”.



Up to now, the driver is installed well. Then you can right click “Computer”—>“Properties”—>“Device manager” , you should see the device as the figure shown below.



How to Add Libraries

Installing Additional Arduino Libraries

Once you are comfortable with the Arduino software and using the built-in functions, you may want to extend the ability of your Arduino with additional libraries.

What are Libraries?

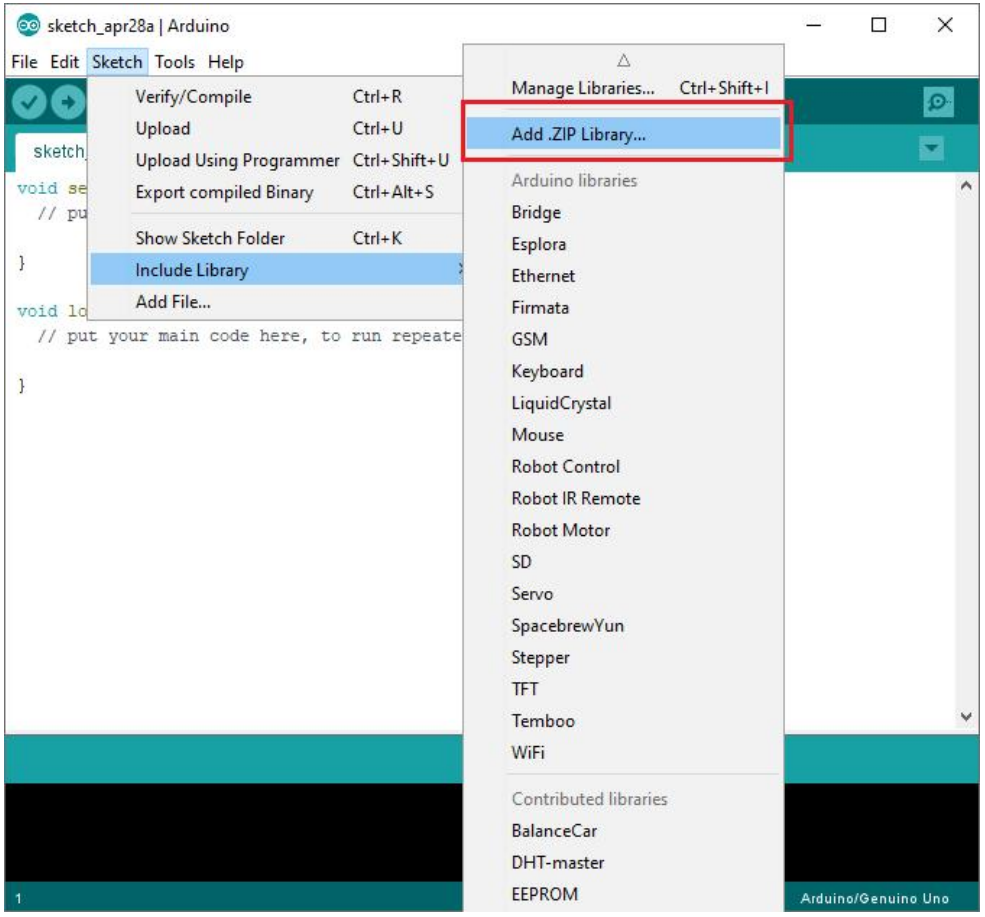
Libraries are a collection of code that makes it easy for you to connect to a sensor, display, module, etc. For example, the built-in LiquidCrystal library makes it easy to talk to character LCD displays. There are hundreds of additional libraries available on the Internet for download. The built-in libraries and some of these additional libraries are listed in the reference. To use the additional libraries, you will need to install them.

How to Install a Library

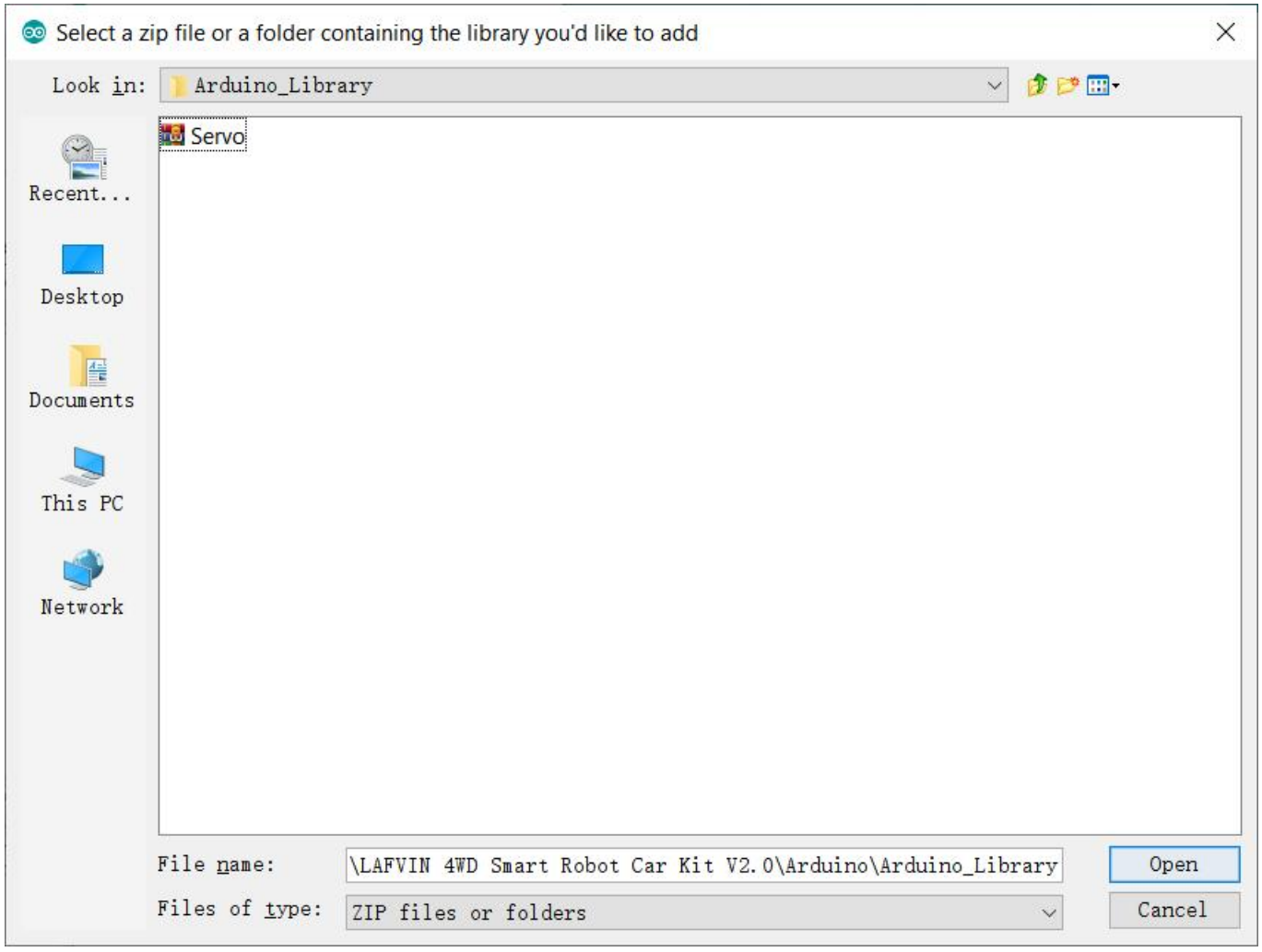
Importing a .zip Library

Libraries are often distributed as a ZIP file or folder. The name of the folder is the name of the library. Inside the folder will be a .cpp file, a .h file and often a keywords.txt file, examples folder, and other files required by the library. you can install 3rd party libraries in the IDE. Do not unzip the downloaded library, leave it as is.

In the Arduino IDE, navigate to Sketch > Include Library. At the top of the drop down list, select the option to "Add .ZIP Library".



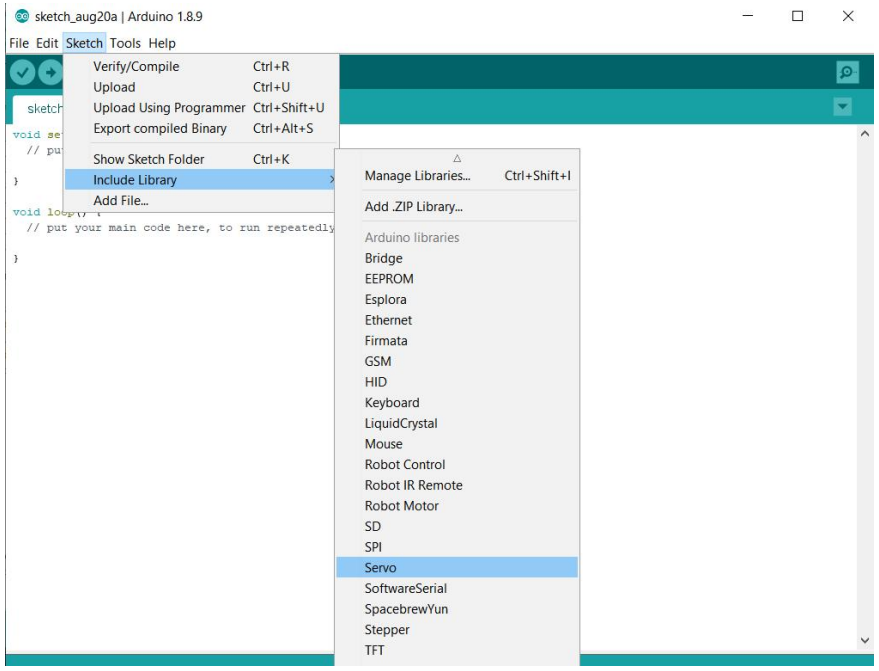
You will be prompted to select the library you would like to add. Navigate to the .zip file's location and open it.



(Note: This is just to demonstrate how to add a zip library file, whether you need to add a library file depends on your actual program needs)

Return to the Sketch > Import Library menu. You should now see the library at the bottom of the drop-down menu. It is ready to be used in your sketch. The zip file will have been expanded in the libraries folder in your Arduino sketches directory.

NB: the Library will be available to use in sketches, but examples for the library will not be exposed in the File > Examples until after the IDE has restarted.



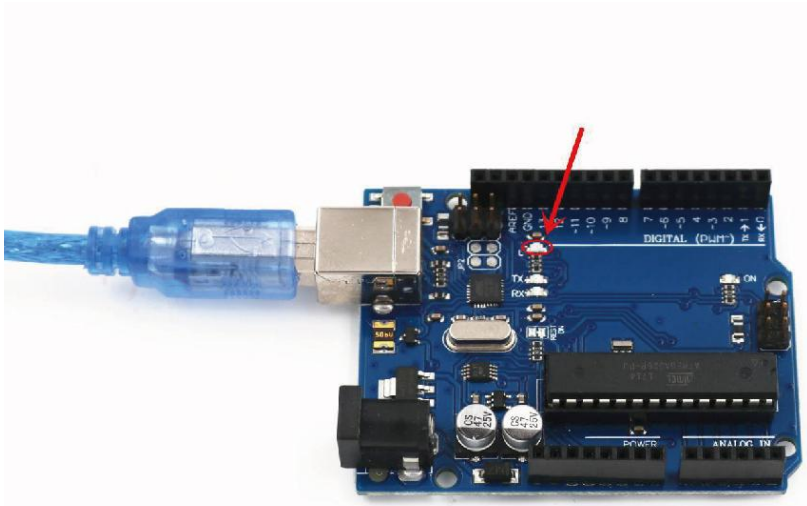
Those two are the most common approaches. MAC and Linux systems can be handled likewise. The manual installation to be introduced below as an alternative may be seldom used and users with no needs may skip it.

Blink Test(Test your first program)

you will learn how to program your UNO controller board to blink the Arduino’s built-in LED, and how to download programs by basic steps.

The UNO board has rows of connectors along both sides that are used to connect to several electronic devices and plug-in 'shields' that extends its capability.

It also has a single LED that you can control from your sketches. This LED is built onto the UNO board and is often referred to as the 'L' LED as this is how it is labeled on the board.



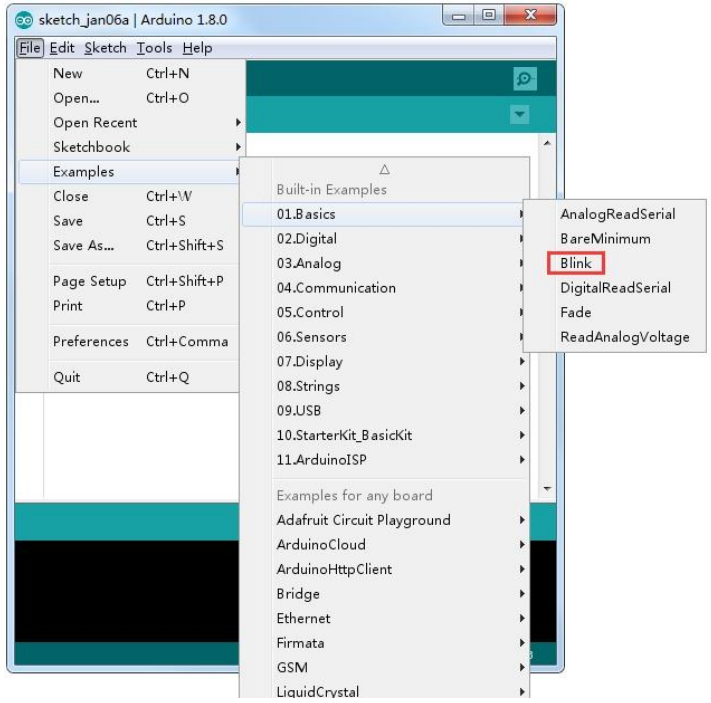
You may find that your UNO board's 'L' LED already blinks when you connect it to a USB plug. This is because the boards are generally shipped with the 'Blink' sketch pre-installed.

In this Project, we will reprogram the UNO board with our own Blink sketch and then change the rate at which it blinks.

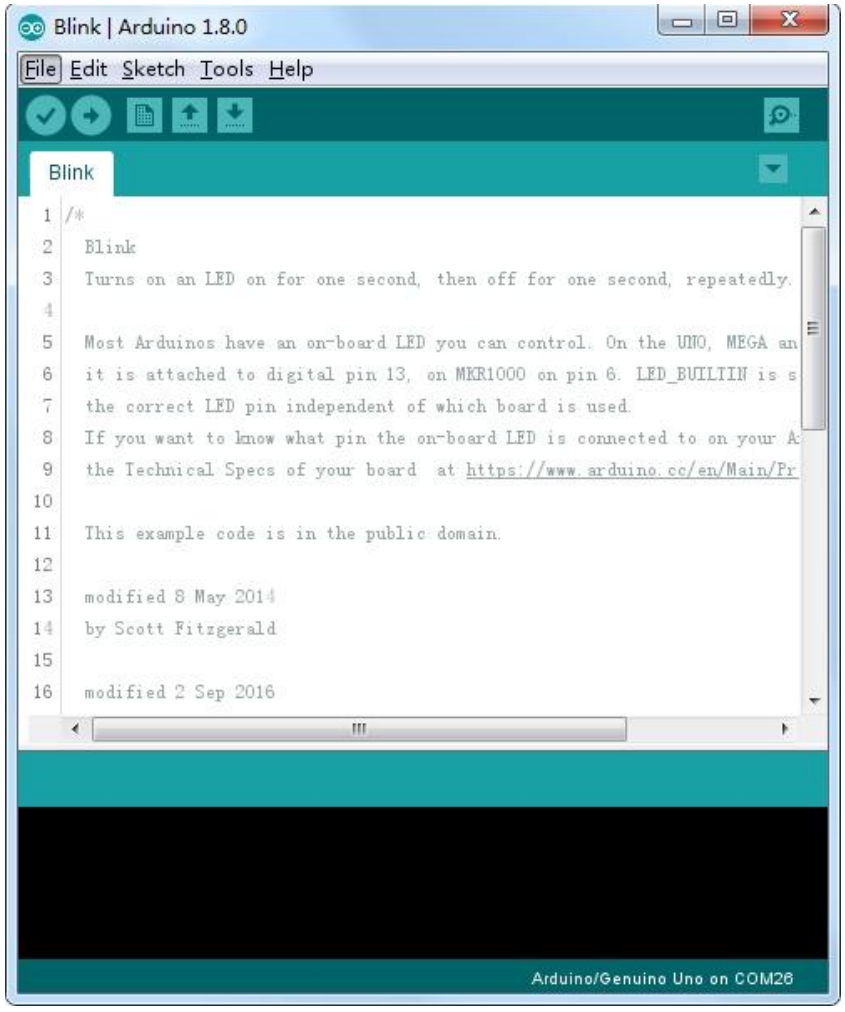
In the previous chapter-How to install Arduino IDE, you set up your Arduino IDE and made sure that you could find the right serial port for it to connect to your UNO board. The time has now come to put that connection to the test and program your UNO board.

The Arduino IDE includes a large collection of example sketches that you can load up and use. This includes an example sketch for making the 'L' LED blink.

Load the 'Blink' sketch that you will find in the IDE's menu system under File > Examples > 01.Basics



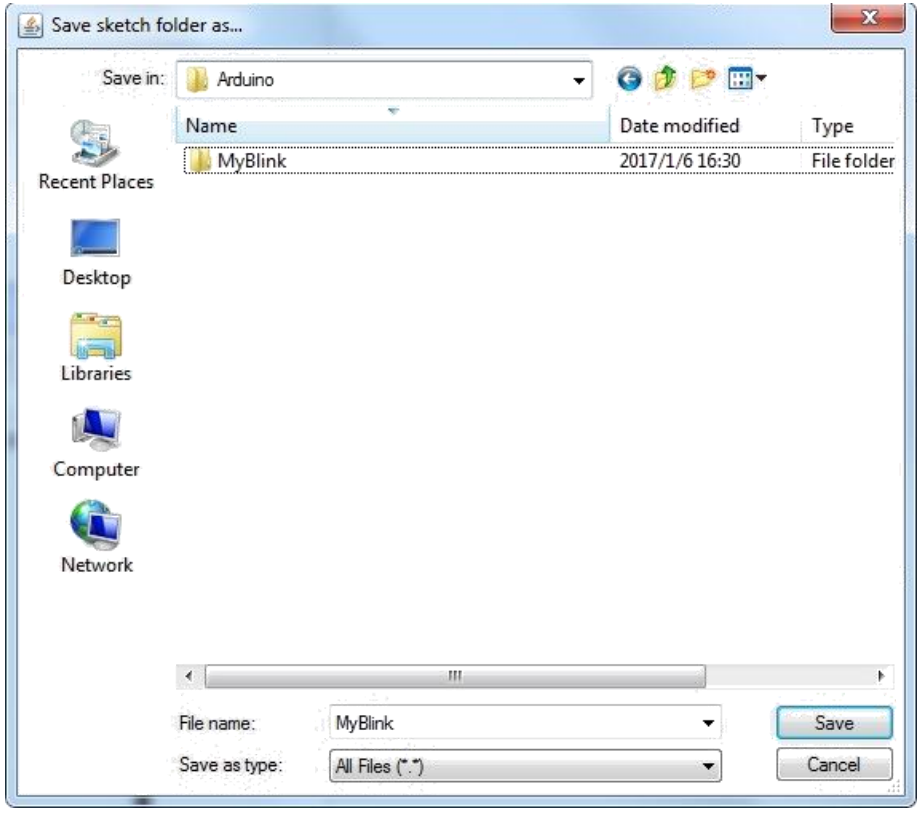
When the sketch window opens, enlarge it so that you can see the entire sketch in the window.



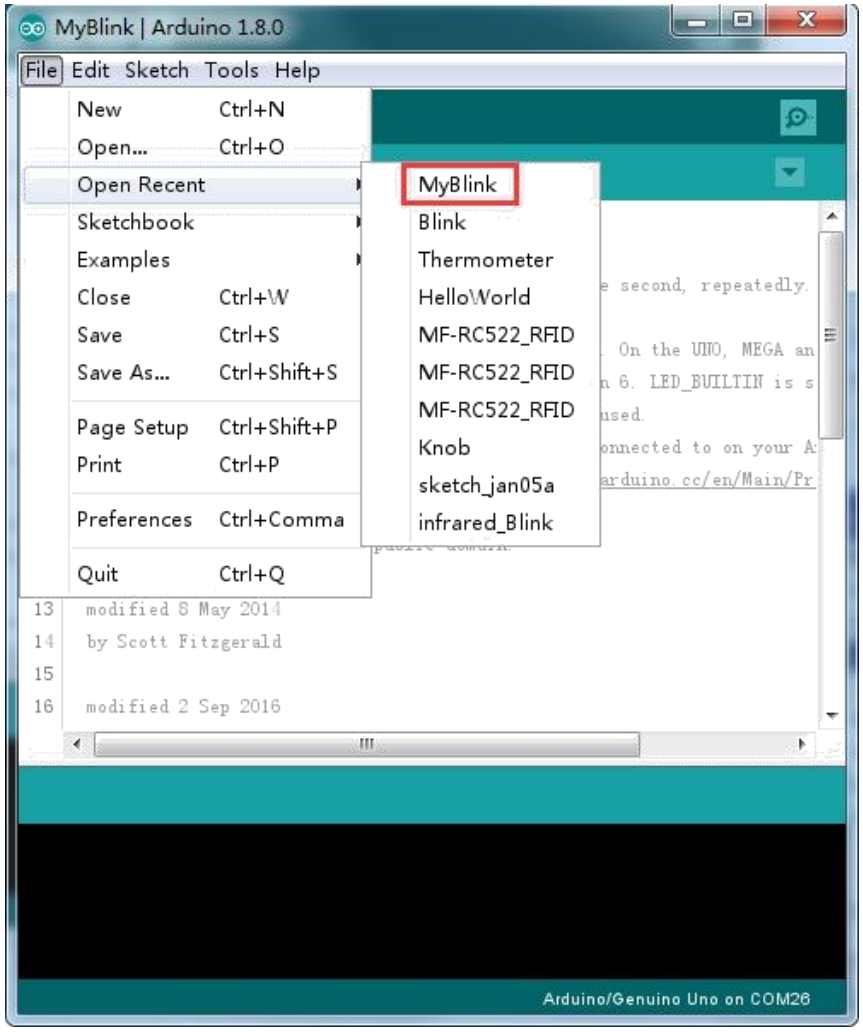
The example sketches included with the Arduino IDE are 'read-only'. That is, you can upload them to an UNO R3 board, but if you change them, you cannot save them as the same file.

Since we are going to change this sketch, the first thing you need to do is save your own copy.

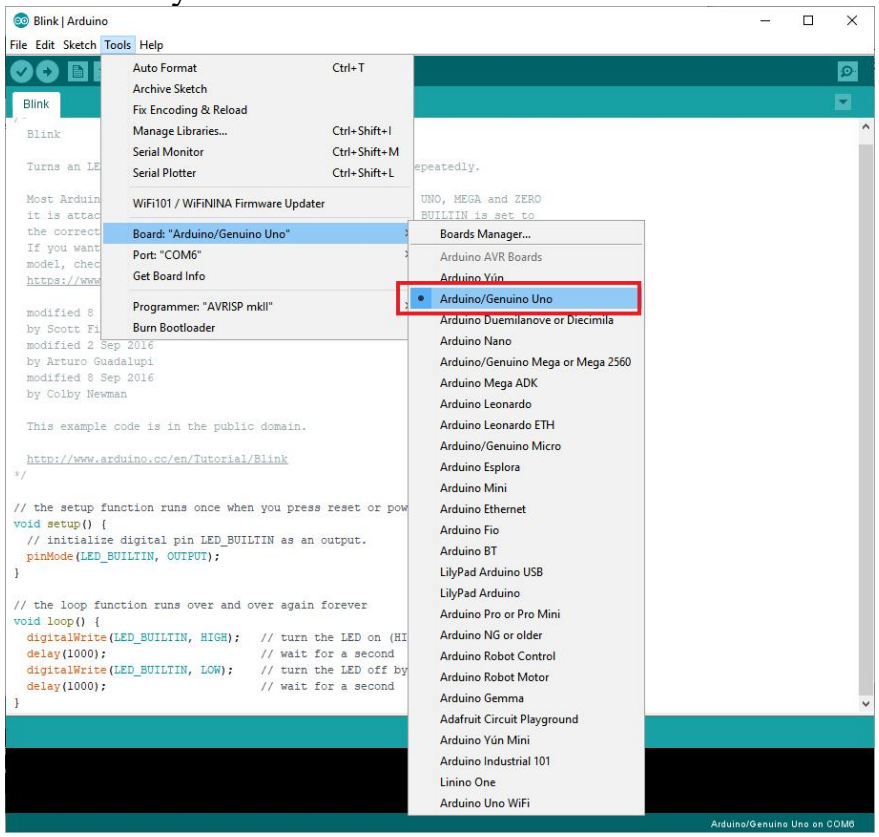
From the File menu on the Arduino IDE, select 'Save As..' and then save the sketch with the name 'MyBlink'.

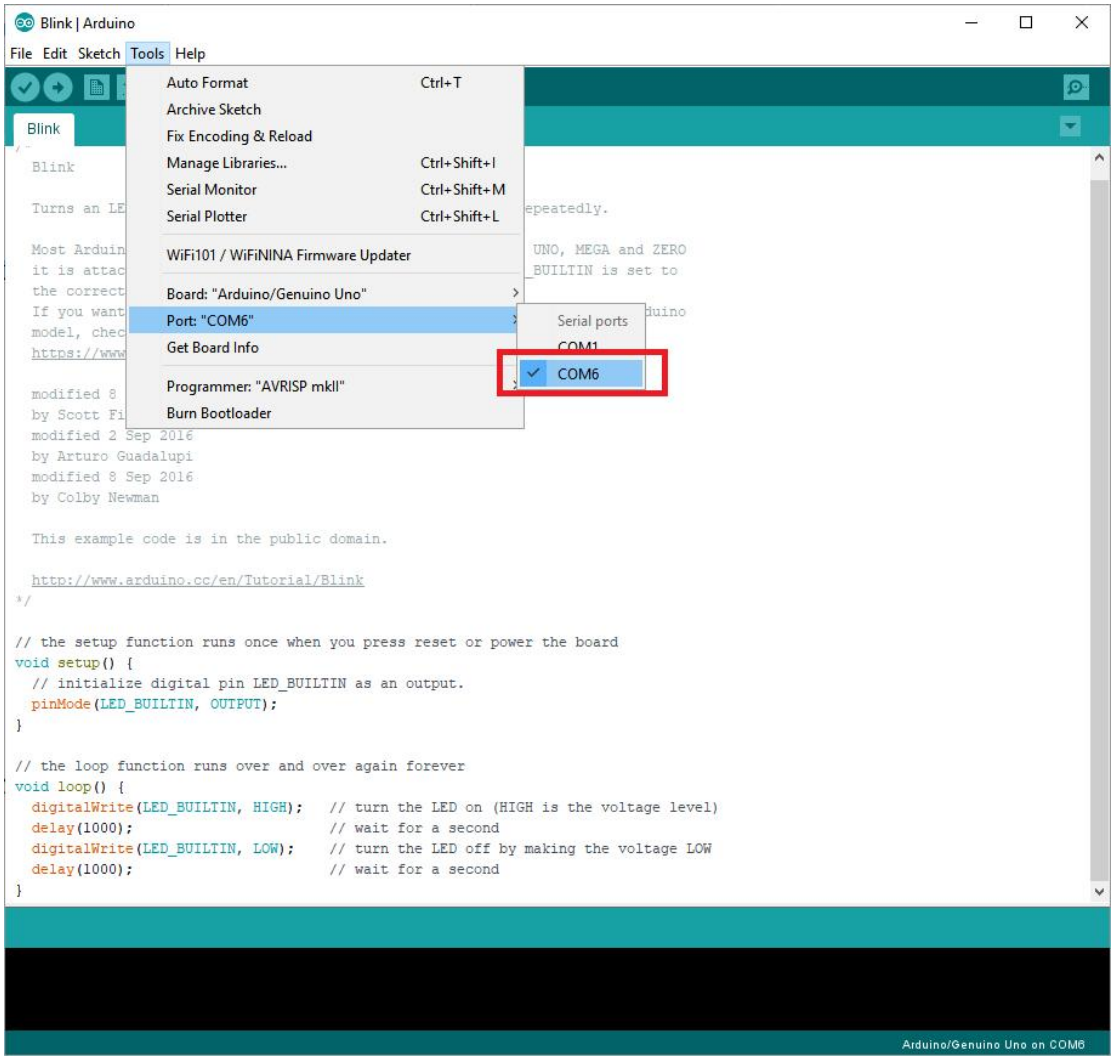


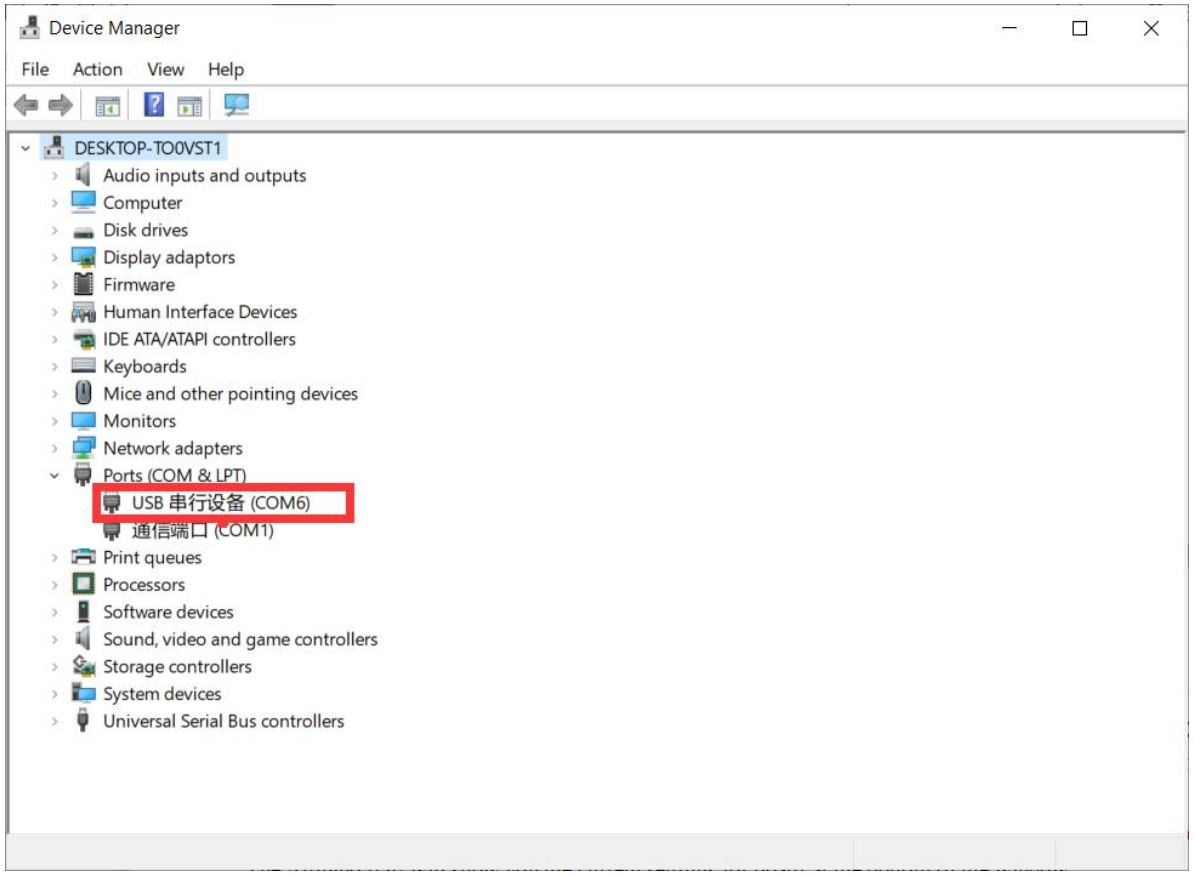
You have saved your copy of 'Blink' in your sketchbook. This means that if you ever want to find it again, you can just open it using the File > Sketchbook menu option.



Attach your Arduino board to your computer with the USB cable and check that the 'Board Type' and 'Serial Port' are set correctly.



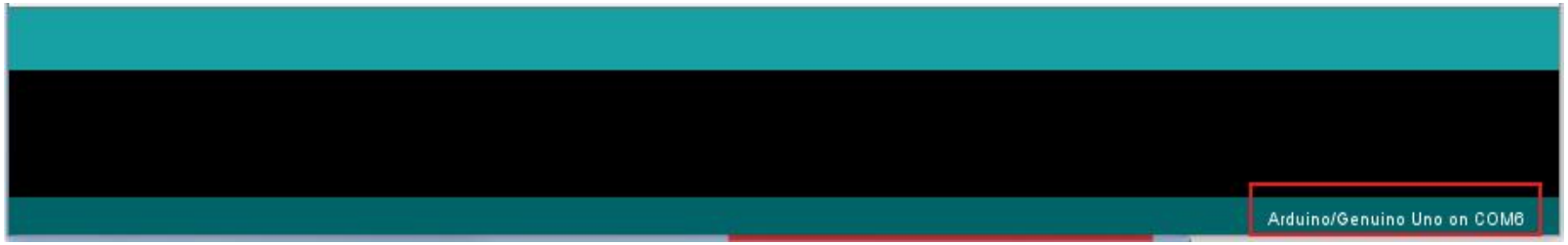

Blink | Arduino
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino/Genuino Uno"
Port: "COM6"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Serial ports
COM1
COM6
Blink
Arduino/Genuino Uno on COM6



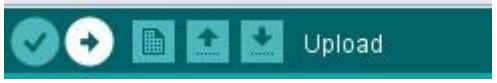
Note: The Board Type and Serial Port here are not necessarily the same as shown in picture. If you are using UNO, then you will have to choose Arduino UNO as the Board Type, other choices can be made in the same manner. And

the Serial Port displayed for everyone is different, despite COM 6 chosen here, it could be COM3 or COM4 on your computer. A right COM port is supposed to be COMX (arduino XXX), which is by the certification criteria.

The Arduino IDE will show you the current settings for board at the bottom of the window.



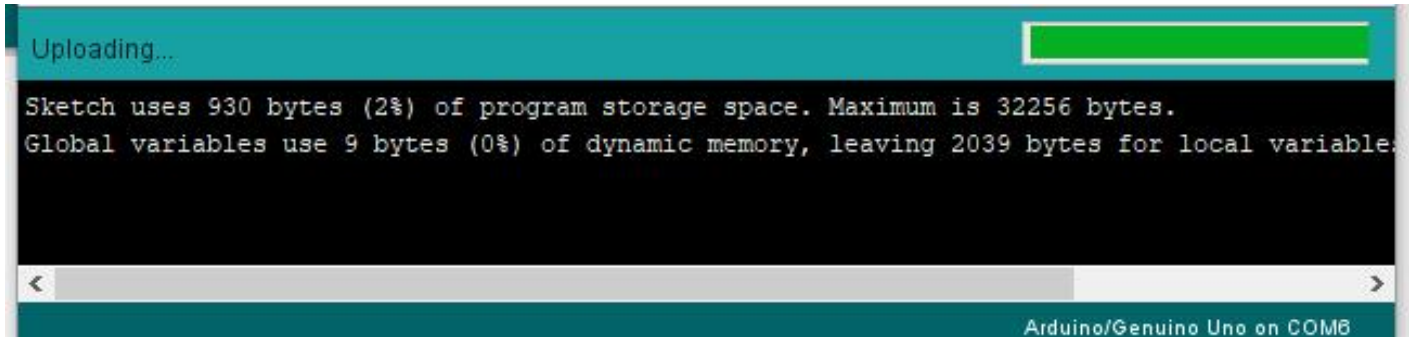
Click on the 'Upload' button. The second button from the left on the toolbar.



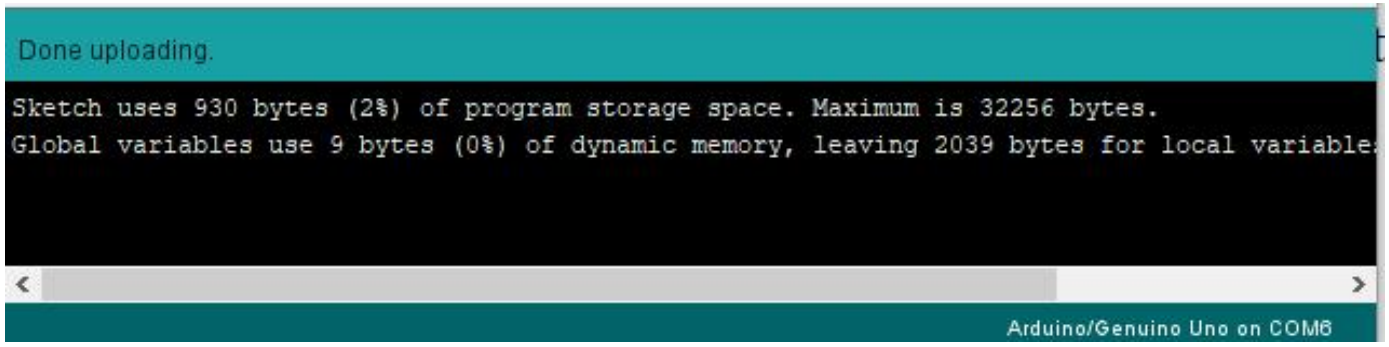
If you watch the status area of the IDE, you will see a progress bar and a series of messages. At first, it will say 'Compiling Sketch...'. This converts the sketch into a format suitable for uploading to the board.

]

Next, the status will change to 'Uploading'. At this point, the LEDs on the Arduino should start to flicker as the sketch is transferred.



Finally, the staus will change to 'Done'.



The other message tells us that the sketch is using 928 bytes of the 32,256 bytes available.After the 'Compiling Sketch..' stage you could get the following error message:

It can mean that your board is not connected at all, or the drivers have not been installed (if necessary) or that the wrong serial port is selected.

If you encounter this, go back to Project 0 and check your installation.

Once the upload has completed, the board should restart and start blinking.

Note that a huge part of this sketch is composed of comments. These are not actual program instructions; rather, they just explain how the program works. They are there for your benefit.

Everything between /* and */ at the top of the sketch is a block comment; it explains what the sketch is for.
Single line comments start with // and everything up until the end of that line is considered a comment.

The first line of code is: int led = 13;

As the comment above it explains, this is giving a name to the pin that the LED is attached to. This is 13 on most Arduinos,

including the UNO and Leonardo.

Next, we have the 'setup' function. Again, as the comment says, this is executed when the reset button is pressed. It is also executed whenever the board resets for any reason, such as power first being applied to it, or after a sketch has been uploaded.

```
void setup() {
// initialize the digital pin as an output. pinMode(led, OUTPUT);
}
```

Every Arduino sketch must have a 'setup' function, and the place where you might want to add instructions of your own is between the { and the }.

In this case, there is just one command there, which, as the comment states tells the Arduino board that we are going to use the LED pin as an output.

It is also mandatory for a sketch to have a 'loop' function. Unlike the 'setup' function that only runs once, after a reset, the 'loop' function will, after it has finished running its commands, immediately start again.

```
void loop()
{ digitalWrite(led, HIGH); delay(1000);
digitalWrite(led, LOW); delay(1000);
}
```

Inside the loop function, the commands first of all turn the LED pin on (HIGH), then 'delay' for 1000 milliseconds (1 second), then turn the LED pin off and pause for another second.

You are now going to make your LED blink faster. As you might have guessed, the key to this lies in changing the parameter in () for the 'delay' command.

// turn the LED off (LOW is the voltage level) // wait for a second
// turn the LED on (HIGH is the voltage level) // wait for a second

```
30 // the loop function runs over and over again forever
31 void loop() {
32   digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the volt
33   delay(500)                         // wait for a second
34   digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the vo
35   delay(500)                         // wait for a second
36 }
```

This delay period is in milliseconds, so if you want the LED to blink twice as fast, change the value from 1000 to 500. This would then pause for half a second each delay rather than a whole second.

Upload the sketch again and you should see the LED start to blink more quick.

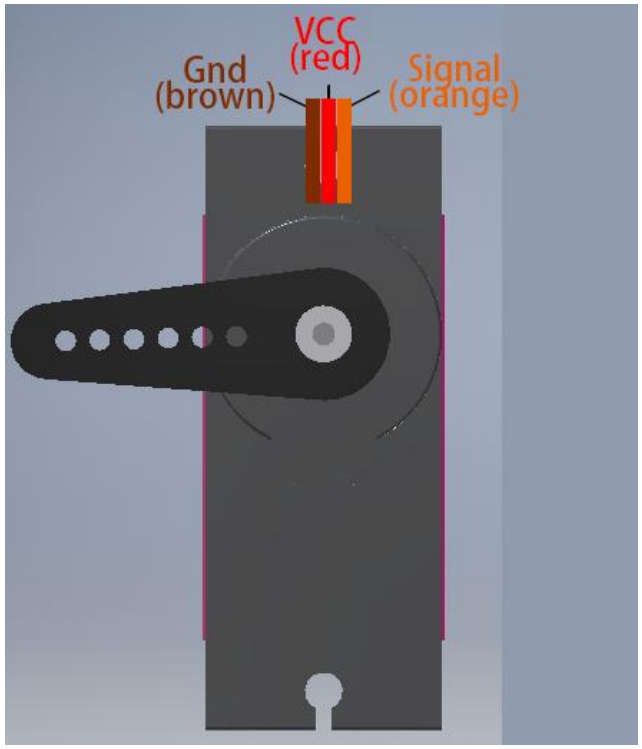
Lesson 2 Servo Control and Installation Angle Calibration

Overview

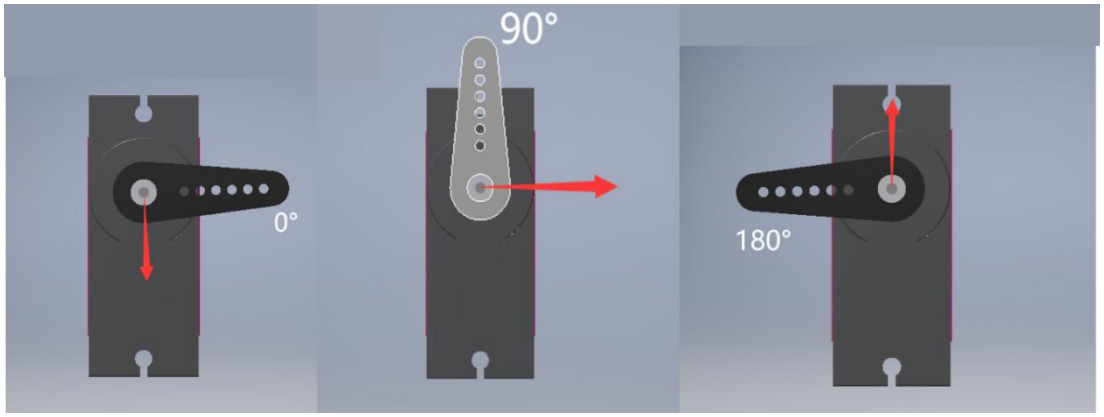
In this lesson, we learn how to control the rotation of the servo.Each joint of the robot arm is controlled by a servo.

What is Servo

Servo motor is a position control rotary actuator. It mainly consists of a housing, circuit board, core-less motor, gear and position sensor. Its working principle is that the servo receives the signal sent by MCUs or receivers and produces a reference signal with a period of 20ms and width of 1.5ms, then compares the acquired DC bias voltage to the voltage of the potentiometer and obtain the voltage difference output.When the motor speed is constant, the potentiometer is driven to rotate through the cascade reduction gear, which leads that the voltage difference is 0, and the motor stops rotating. Generally, the angle range of servo rotation is 0° --180 °.The rotation angle of servo motor is controlled by regulating the duty cycle of PWM (Pulse-Width Modulation) signal. The standard cycle of PWM signal is 20ms (50Hz). Theoretically, the width is distributed between 1ms-2ms, but in fact, it's between 0.5ms-2.5ms. The width corresponds the rotation angle from 0° to 180°. But note that for motors of different brands, the same signal may have different rotation angle.



In general, servo has three lines in brown, red and orange. The brown wire is grounded, the red one is a positive pole line and the orange one is a signal line.



The corresponding servo angles are shown below:

High level time	Servo angle
0.5ms	0 degree
1ms	45 degree
1.5ms	90 degree
2ms	135 degree
2.5ms	180 degree

Specification：

Working voltage: DC 4.8V ~ 6V

Operating angle range: about 180 ° (at 500 → 2500 μsec)

Pulse width range: 500 → 2500 μsec

No-load speed: 0.12 ± 0.01 sec / 60 (DC 4.8V) 0.1 ± 0.01 sec / 60 (DC 6V)

No-load current: 200 ± 20mA (DC 4.8V) 220 ± 20mA (DC 6V)

Stopping torque: 1.3 ± 0.01kg · cm (DC 4.8V) 1.5 ± 0.1kg · cm (DC 6V)

Stop current: ≦ 850mA (DC 4.8V) ≦ 1000mA (DC 6V)

Standby current: 3 ± 1mA (DC 4.8V) 4 ± 1mA (DC 6V)

Note:

If you have installed the robotic arm and found that the servo generates heat during use, it means that the initial angle of the servo is not calibrated.

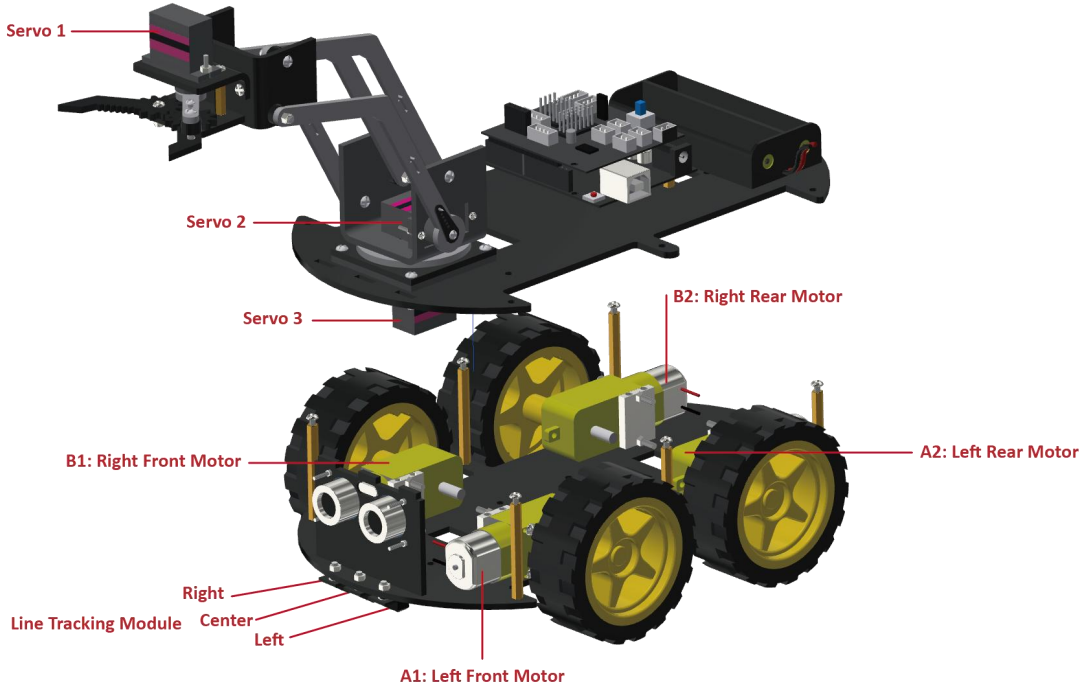
Before installing the servo, you need to set the initial position of the three servo motors. Only after the angle is set, you can install it according to the vertical or parallel direction indicated by the video.

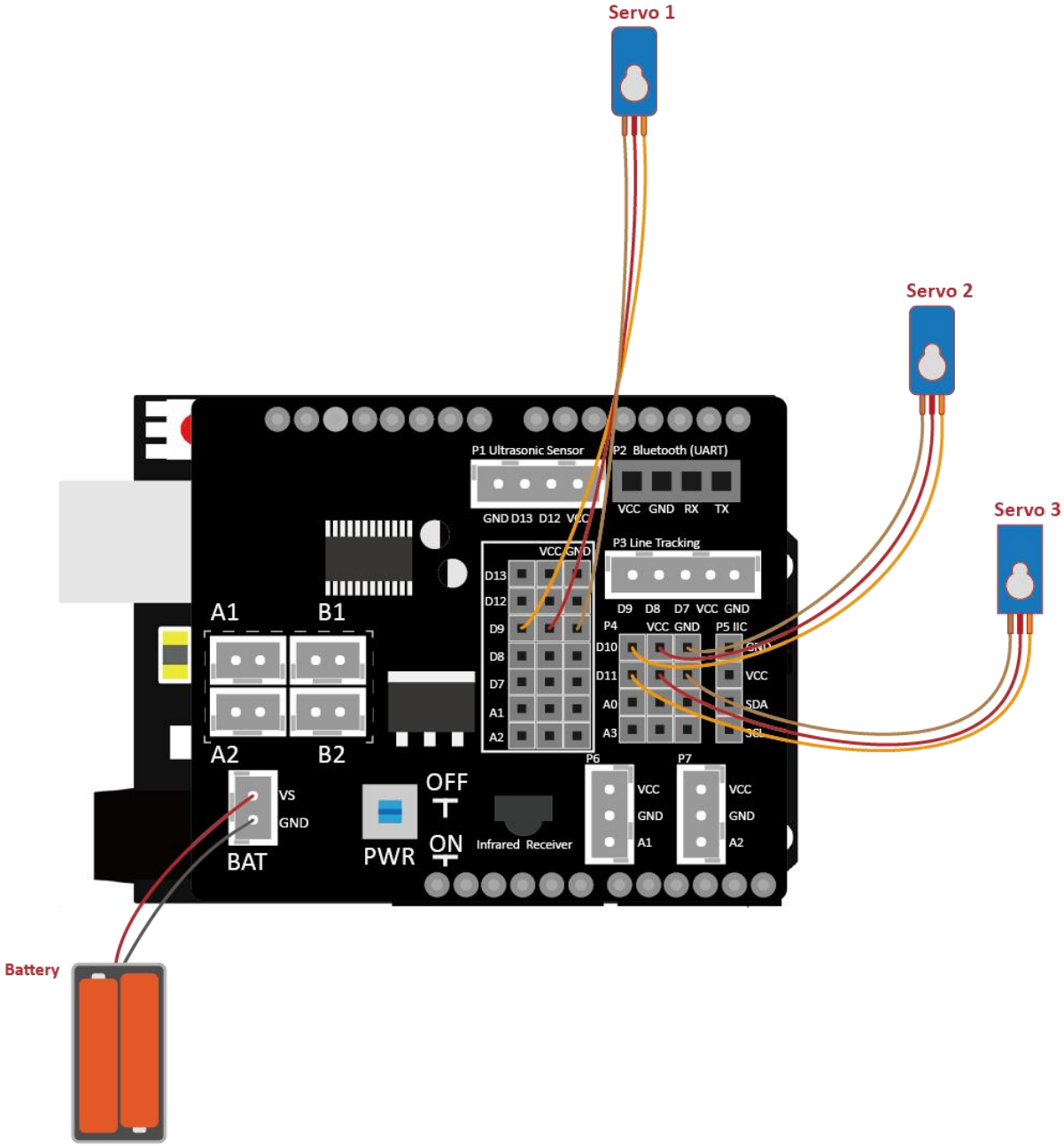
The angle calibration method of the servo motor is as follows:

Claw-->Servo1-->D9--> 90°

Arm-->Servo2-->D10-->135°

Base-->Servo3-->D11-->90°




Servo 1
Servo 2
Servo 3
P1 Ultrasonic Sensor
GND D13 D12 VCC
P2 Bluetooth (UART)
VCC GND RX TX
P3 Line Tracking
D9 D8 D7 VCC GND
P4
VCC GND
P5 IIC
D10
D11
A0
A3
GND
VCC
SDA
SCL
VCC GND
D13
D12
D9
D8
D7
A1
A2
A1
B1
A2
B2
VS
GND
BAT
PWR
OFF
ON
Infrared Receiver
P6
VCC
GND
A1
P7
VCC
GND
A2
Battery

The following code is to initialize the angle value of servo, causing the convenience of installation and adjustment.Open this source program in **Arduino_Code>Lesson_2>Servo_Installation_Angle_Initialization.ino.**

```
#include <Servo.h>
int claw_degrees;
int arm_degrees;
int base_degrees;
Servo myservo1;//servo of claw
Servo myservo2;//servo of arm
Servo myservo3;//servo of base


void setup()
{
```

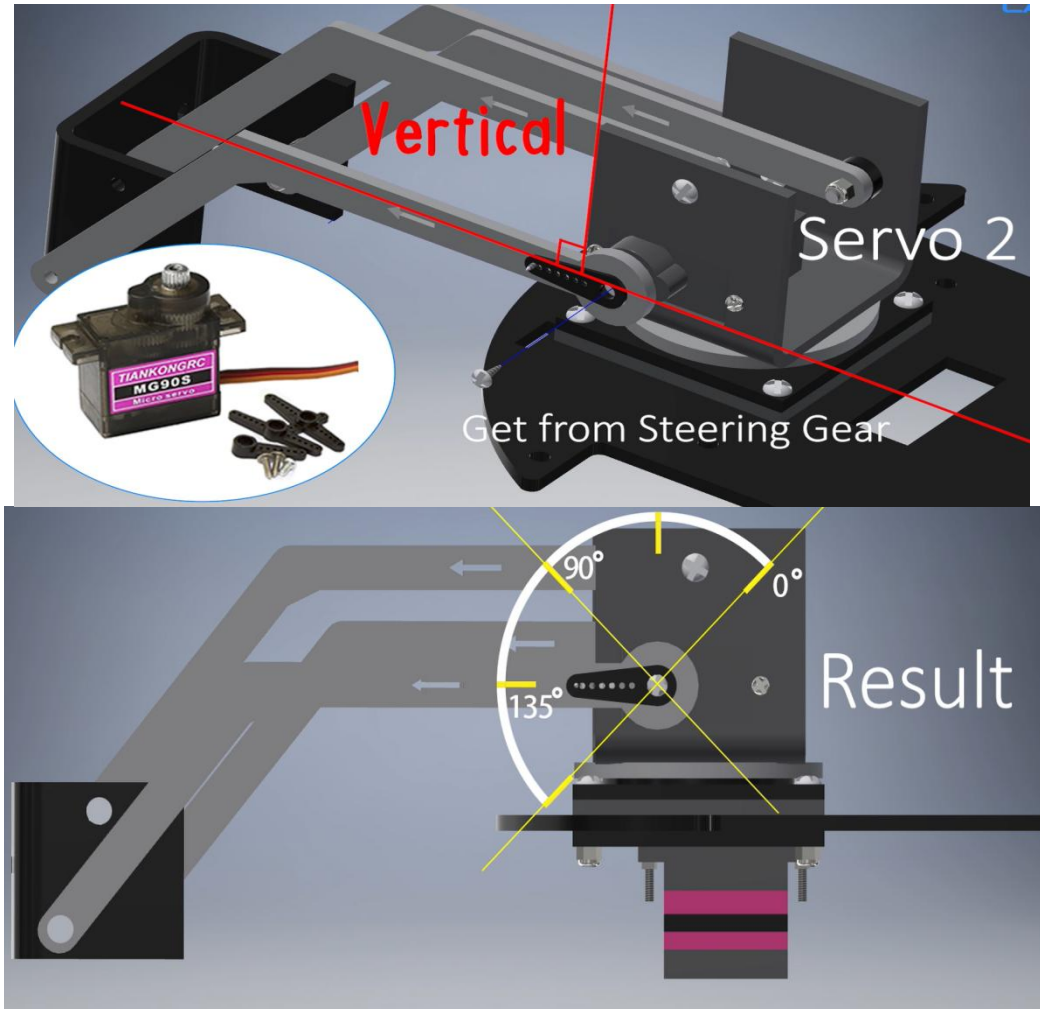
```
    claw_degrees = 90;//initialize the angle value of claw servo 1
    arm_degrees = 135;//initialize the angle value of arm servo 2
    base_degrees = 90;//initialize the angle value of base servo 3
    myservo1.attach(9);//claw Servo 1 is connected to D9
    myservo2.attach(10);//arm Servo 2 is connected to D10
    myservo3.attach(11);//base Servo 3 is connected to D11
}
void loop()
{
    myservo1.write(claw_degrees);//make claw servo 1 rotate to 90°
    delay(200);
    myservo2.write(arm_degrees);//make arm servo 2 rotate to 135°
    delay(200);
```

```
    myservo3.write(base_degrees);//make base servo 3 rotate to 90°
    delay(200);
}
```

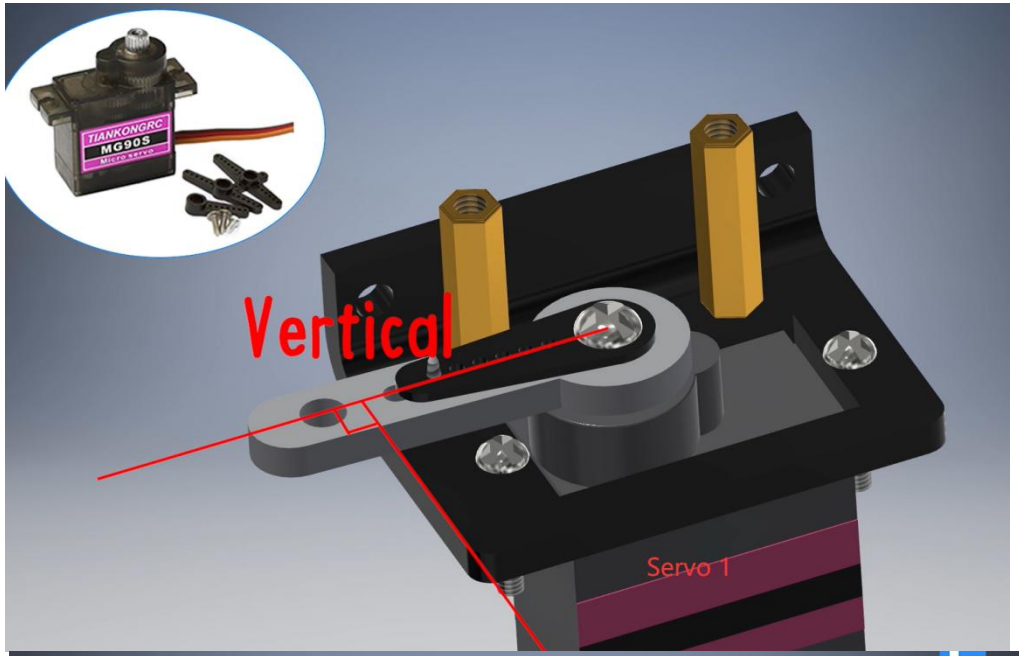
Upload this code to Arduino UNO, turn on the power.Servo 1 rotates to 90°, servo 2 rotates to 135°, and servo 3 rotates to 90°.Insert the plastic servo swing arm into the gear of the servo according to the parallel and vertical directions marked in the video.

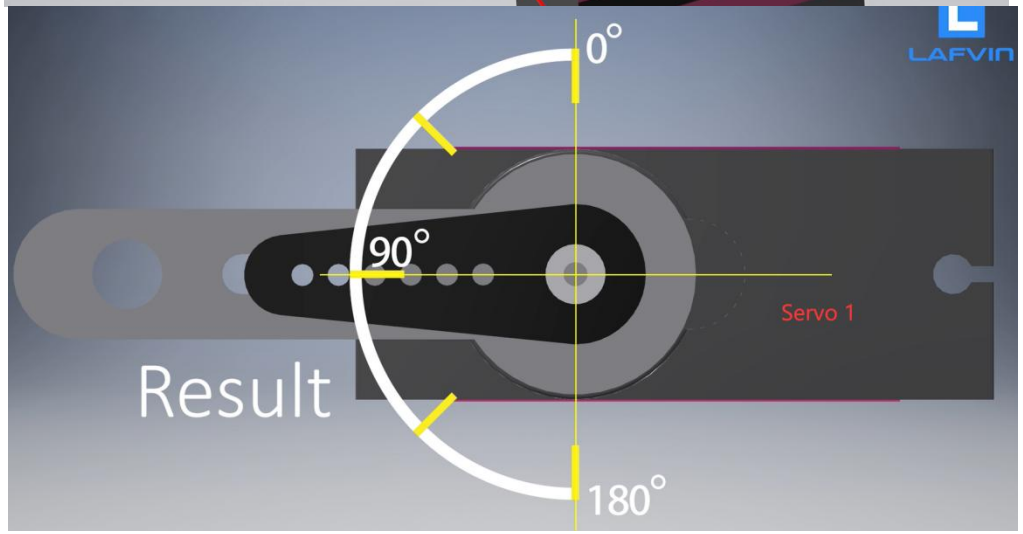
Base-->Servo3-->D11-->90°

Arm-->Servo2-->D10-->135°



Claw-->Servo1-->D9--> 90°





Lesson 3 Motor Speed and Direction Control

Overview

In this lesson we will learn how to control the direction and speed control of the robot car.

Motor Driver

The expansion board has integrated the motor drive chip.Current motor drive device, which has a large current MOSFET-H bridge junction Structure, dual-channel circuit output, can drive 2 motors at the same time.It outputs continuous drive power up to 1 A per channel Current, starting peak current up to 2A/3A (continuous pulse/single pulse (Punch); 4 motor control modes: forward/reverse/brake/stop End;

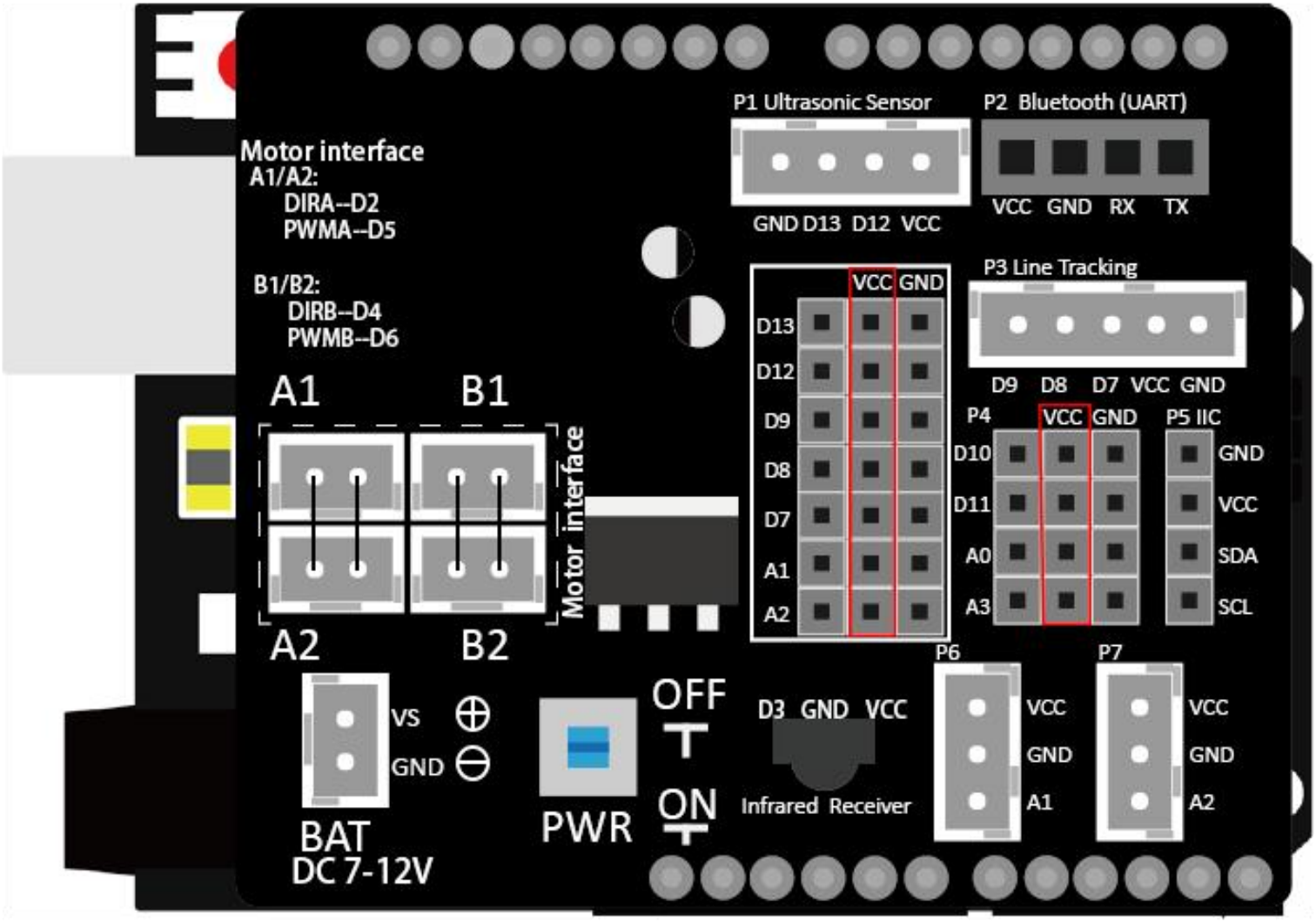
Specifications:

Recommended motor voltage (VMOT): 7.4– 13.5 V

• Logic voltage (VCC): 2.7 – 5.5 V

• Output current maximum: 3 A per channel

• Output current continuous: 1 A per channel (can be paralleled to deliver 2 A continuous)


P1 Ultrasonic Sensor
GND D13 D12 VCC
P2 Bluetooth (UART)
VCC GND RX TX
Motor interface
A1/A2:
DIRA--D2
PWMA--D5
B1/B2:
DIRB--D4
PWMB--D6
A1
B1
A2
B2
Motor interface
VCC GND
D13
D12
D9
D8
D7
A1
A2
P3 Line Tracking
D9 D8 D7 VCC GND
P4
VCC GND
D10
D11
A0
A3
P5 IIC
GND
VCC
SDA
SCL
VS
GND
BAT
DC 7-12V
PWR
OFF
ON
D3 GND VCC
Infrared Receiver
P6
VCC
GND
A1
P7
VCC
GND
A2

A1 A2 B1 B2 of the motor Arduino shield is the interface of the motor.

The motors connected to the A1 interface and the A2 interface have the same speed and the same direction.

The motors connected to the B1 and B2 ports have the same speed and the same direction.

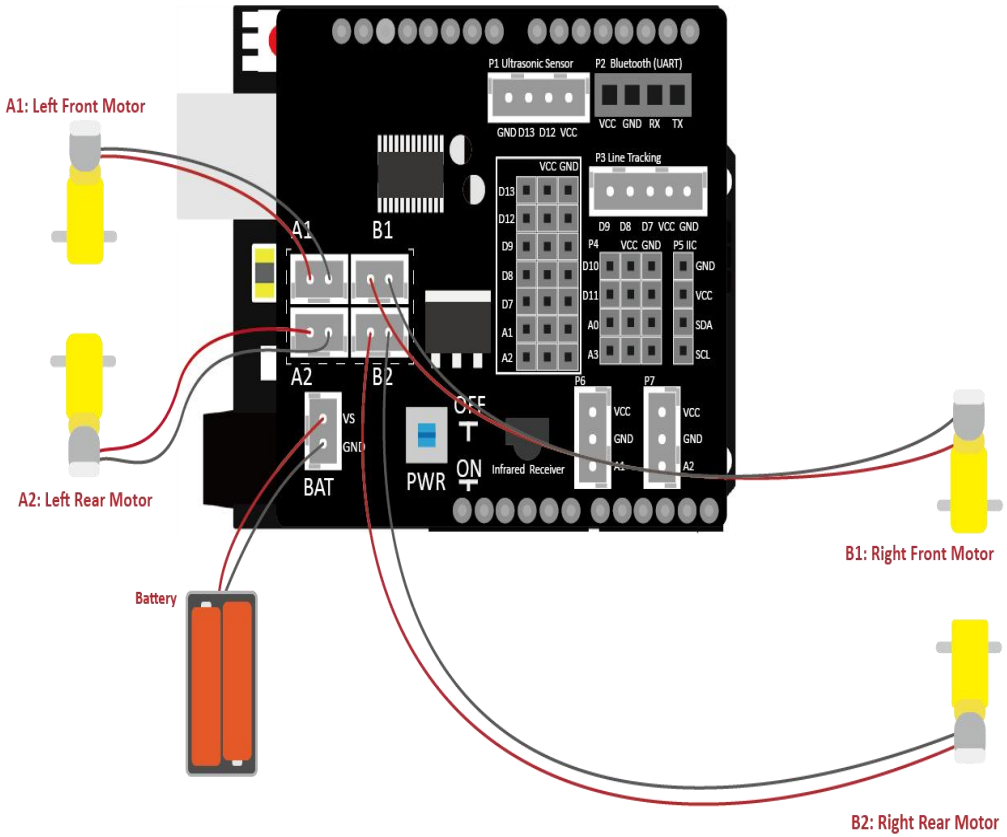
The D2 digital I/O port controls the direction of the motor of the port A, and the D5 digital I/O port outputs the PWM signal to control the speed of the motor of the port A.

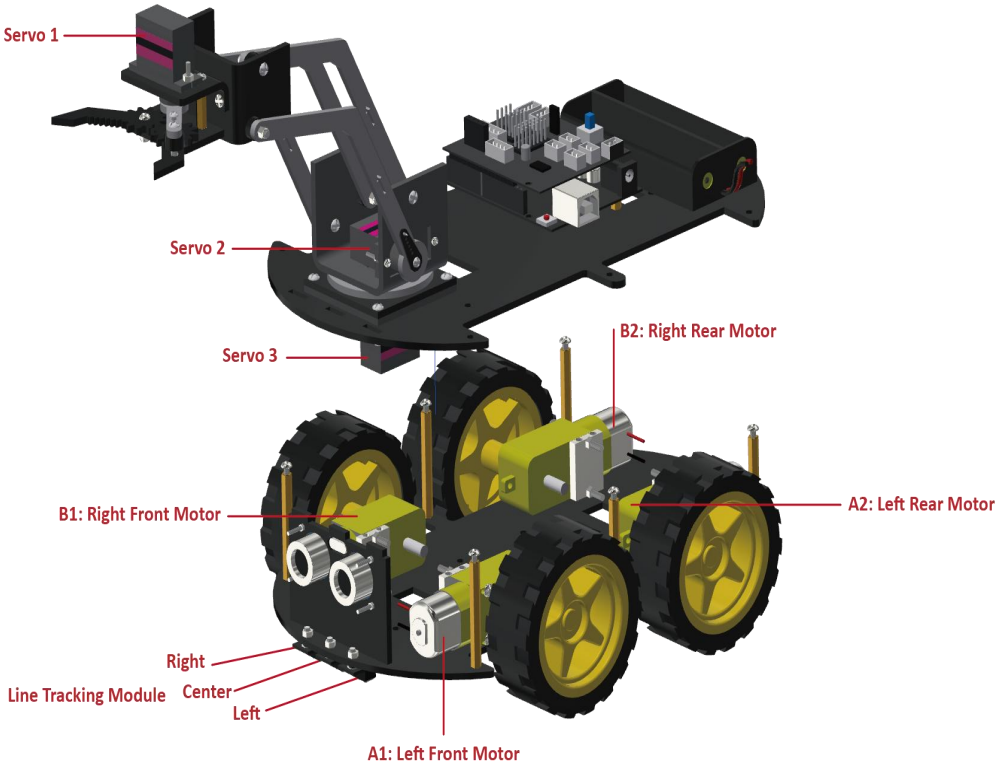
The D4 digital I/O port controls the direction of the motor of the interface B, and the D6 digital I/O port outputs the PWM signal to control the speed of the motor of the interface B.

The PWM value is in the range of 0-255. The greater the value, the faster the motors turn.When PWM=0, it means stop, when PWM=255 it means maximum speed.

4WD Robot	D2	D5(PWM)	D4	D6(PWM)
Forward	HIGH	0-255	LOW	0-255
Backward	LOW	0-255	HIGH	0-255
Rotate to left	LOW	0-255	LOW	0-255
Rotate to right	HIGH	0-255	HIGH	0-255
Stop	/	0	/	0

How to connect the circuit





Let's program

Test 1--Motor Speed and Direction Control

In experimental test 1, we will write code to control the robot car to move forward, backward, rotate to left, rotate to right, stop.

Arduino Code

if you want to refer to the program we provide.Open this source program in Arduino_Code>Lesson_3>**Test_1_Motor_Speed_and_Direction_Control.ino**.

Programming Thinking

```
/*Define a sub-function, the function of the sub-function is to control the robot car to move forward, convenient to call in other functions or repeated loops*/
void Move_Forward(int speed) //Define the forward function of the input speed
{
  digitalWrite(2,HIGH); //D2 digital I/O port controls the direction of the motor of interface A
```

```
    analogWrite(5,speed); //D5 digital I/O port outputs PWM signal to control the speed of the motor of port A.
    digitalWrite(4,LOW);//D4 digital I/O port controls the direction of the motor of interface B
    analogWrite(6,speed);//D6 digital I/O port outputs PWM signal to control the speed of interface B motor.
}
```

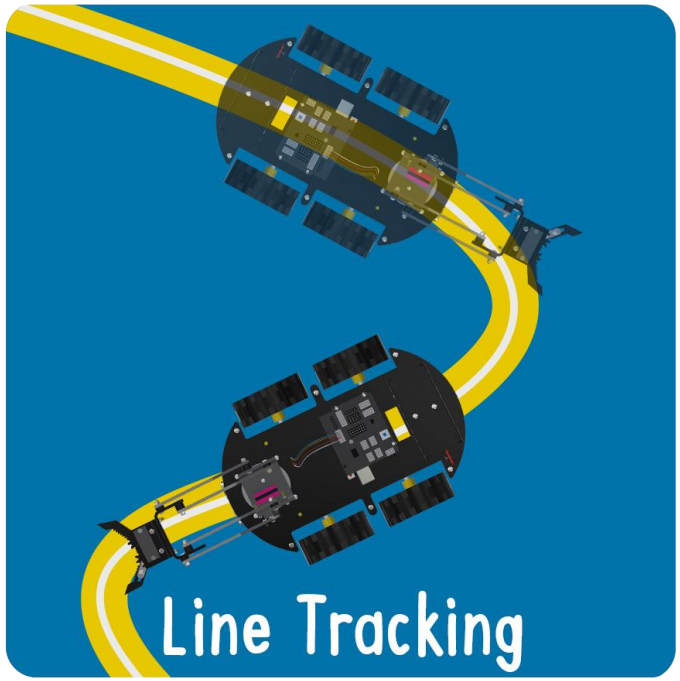
What will you see

Hook up by connection diagram, upload code and power on, smart car goes forward and back for 2s, turns left and right for 2s, and stops for 2s alternately.

Lesson 4 Line Tracking Smart Car

Overview

In this lesson, we will complete the test of two experimental codes. In experimental test 1, we learned how to use the infrared line-following sensor, and observed the results returned by the sensor to distinguish black and white objects. In the experimental test 2, we learned to combine the infrared line-following sensor with the motor to control the robot car to complete the line-following function.



What is line tracking sensor

The tracking sensor is actually an infrared sensor. The component used here is the TCRT5000 infrared tube. Its working principle is to use the different reflectivity of infrared light to the color, then convert the strength of the reflected signal into a current signal. During the process of detection, black is active at HIGH level, but white is active at LOW level. The detection height is 0-3 cm.

By rotating the adjustable potentiometer on the sensor, it can adjust the detection sensitivity of the sensor.

How to use the line tracking sensor

Operating Voltage: 3.3-5V (DC)

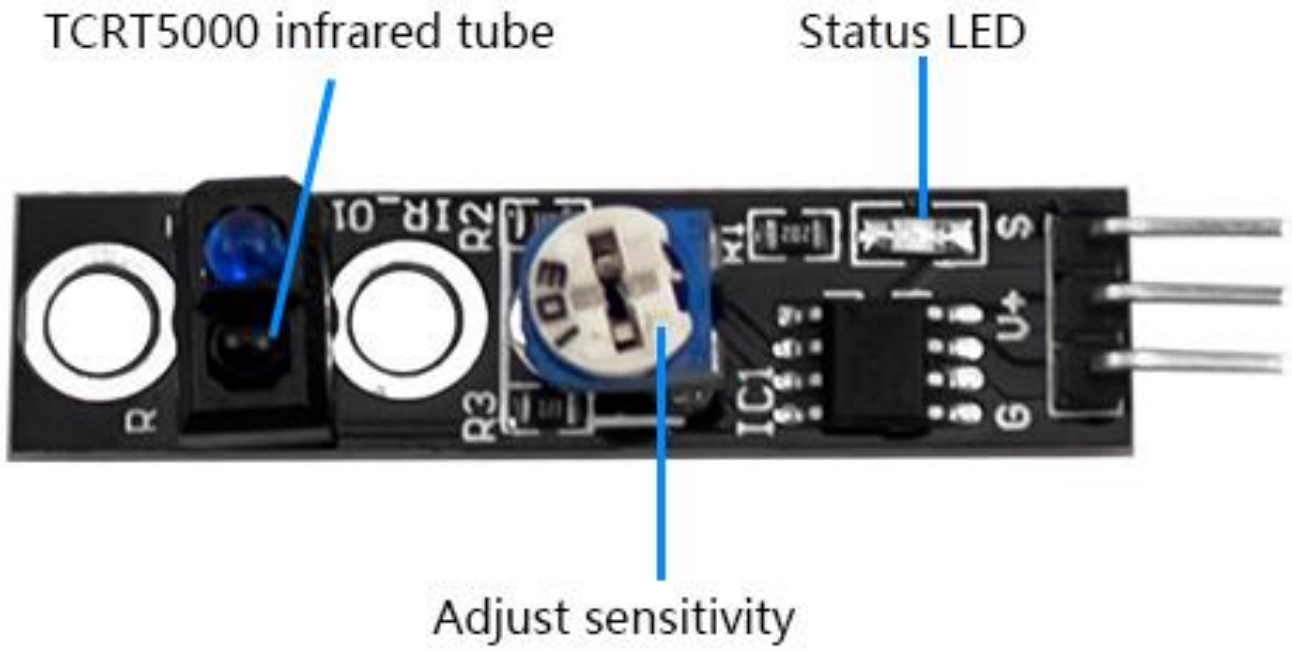
Interface: G(GND) V+(VCC) S(Signal)

Output Signal: Digital signal

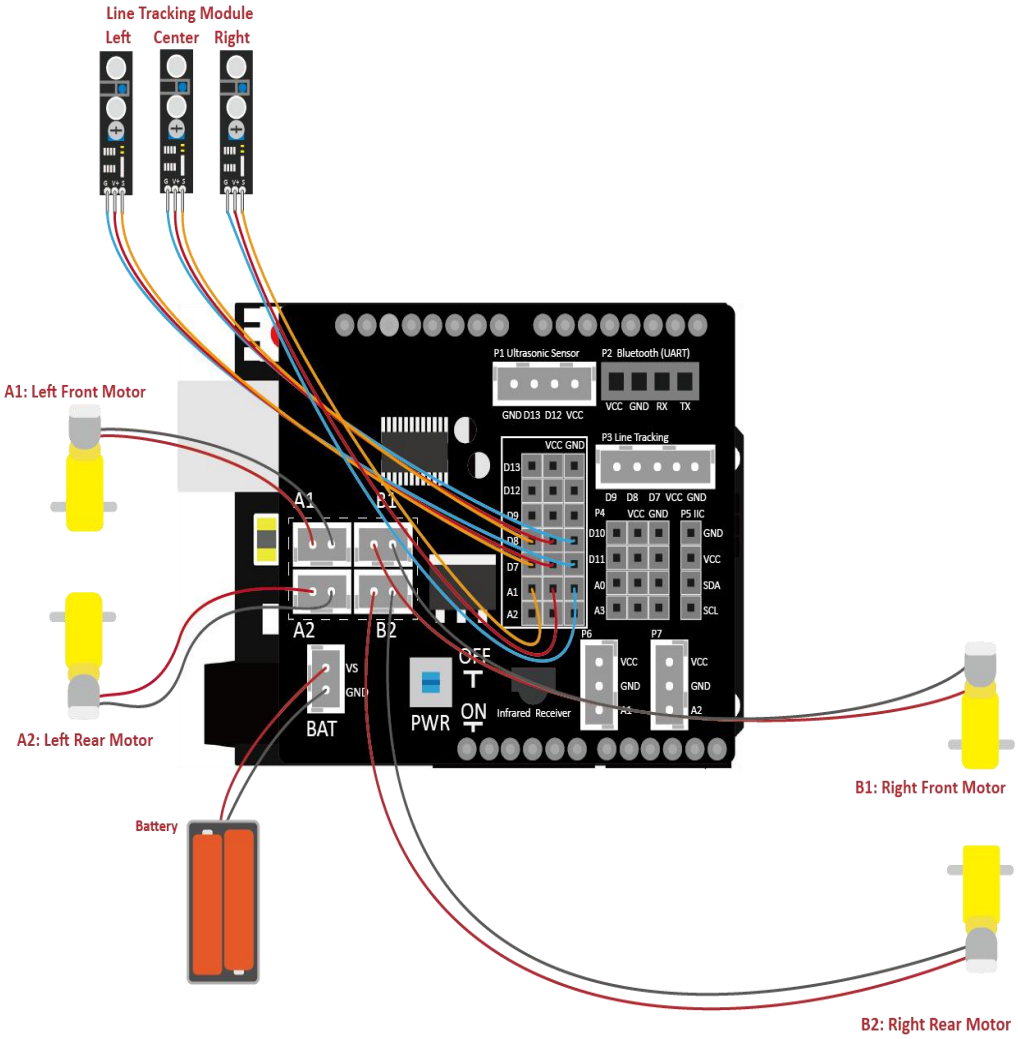
Detection Height: 0-3 cm

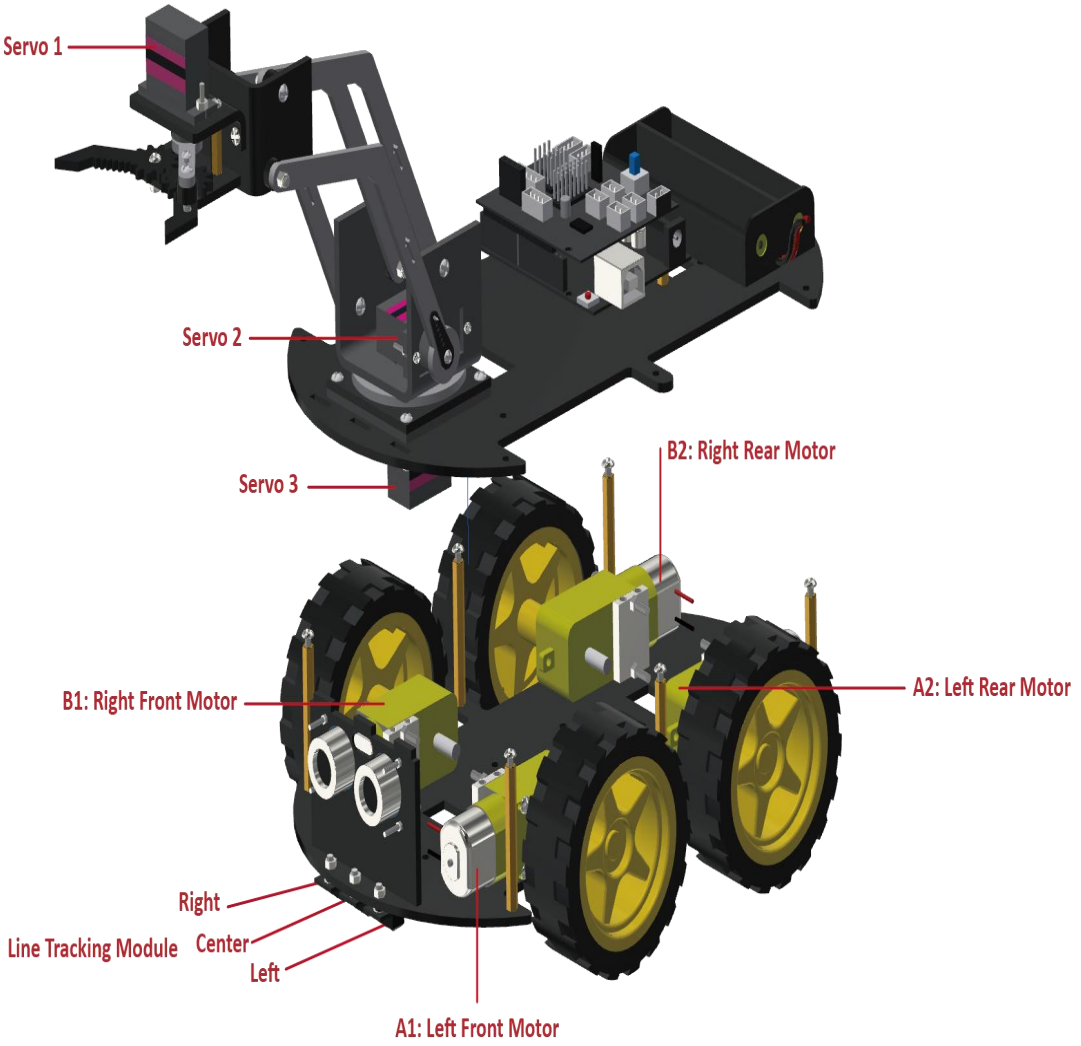
Special note: before testing, turn the potentiometer on the sensor to adjust the detection sensitivity. When adjust the LED at

the threshold between ON and OFF, the sensitivity is the best.



How to connect the circuit





Let's program

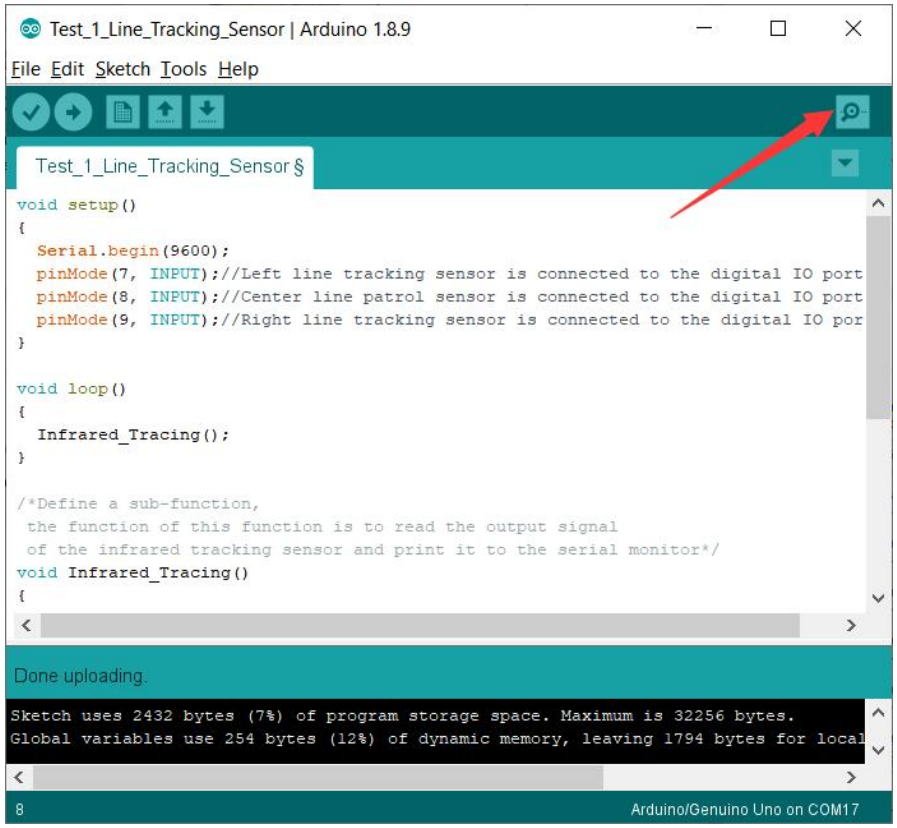
Test 1--Line Tracking Sensor

The main purpose of the test experiment is to read the return signal of the line tracking sensor and print it to the serial port monitor.When detects white paper, sensor’s signal pin outputs LOW (display 0),and status LED is on;When detects black, sensor’s signal pin outputs HIGH (display 1),and status LED is off.

Arduino Code

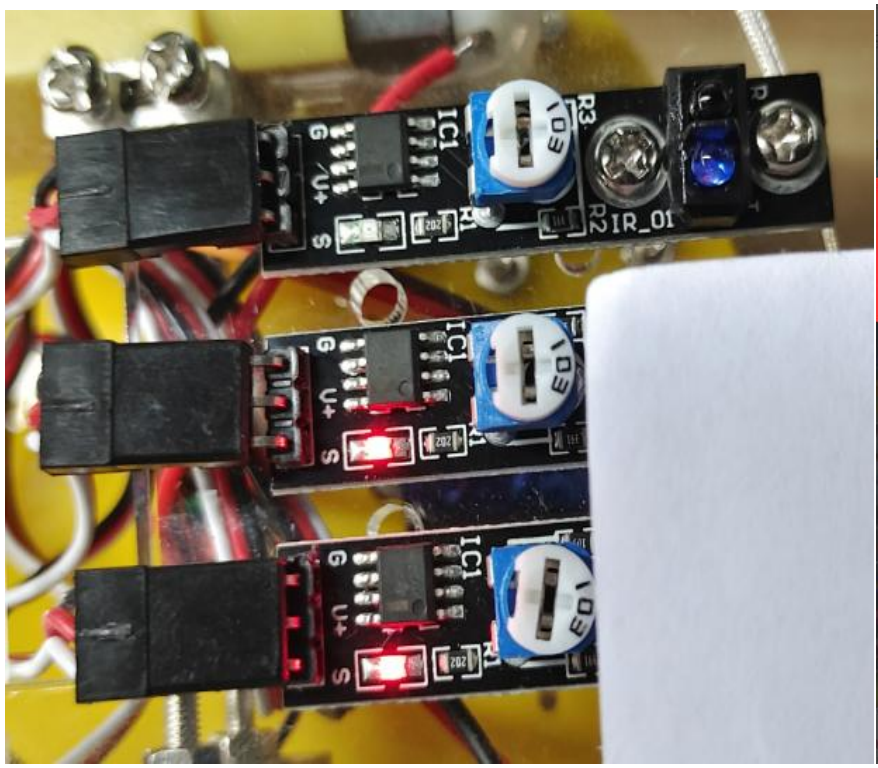
if you want to refer to the program we provide.Open this source program in Arduino_Code>Lesson_4>**Test_1_Line_Tracking_Sensor.ino**.

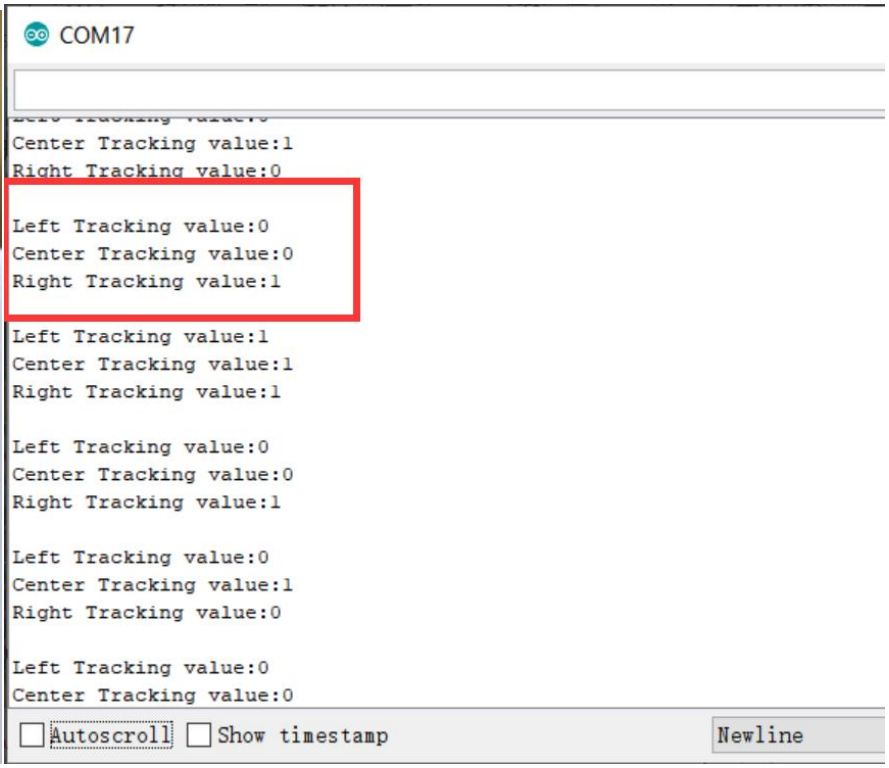
After uploading the code, click the button in the upper right corner to open the serial monitor to view the measured distance



Then you can see the data as blow:

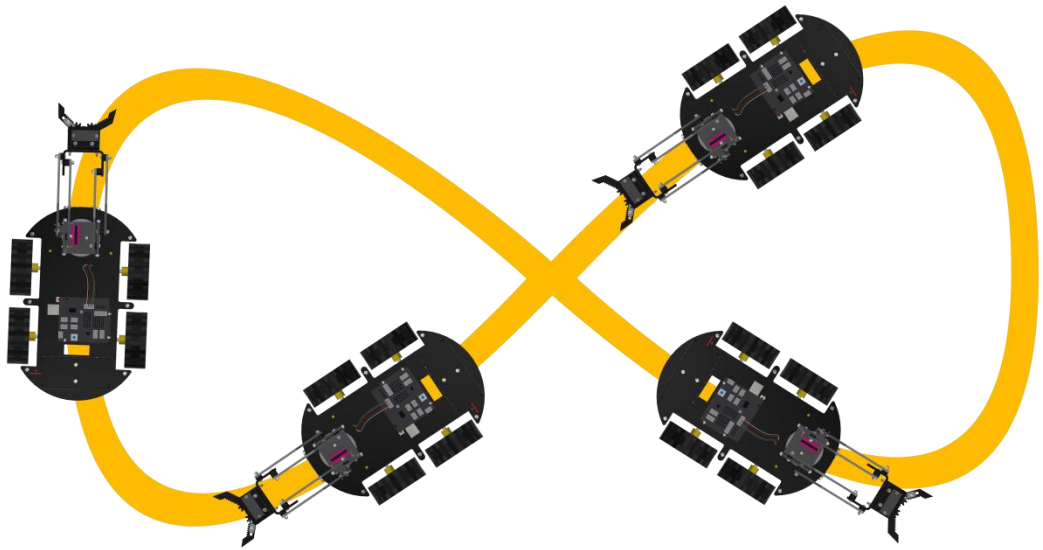
When detects white paper, sensor’s signal pin outputs LOW (display 0),and status LED is on;When detects black, sensor’s signal pin outputs HIGH (display 1),and status LED is off.


IC1
R3
IR_01


COM17
Center Tracking value:1
Right Tracking value:0
Left Tracking value:0
Center Tracking value:0
Right Tracking value:1
Left Tracking value:1
Center Tracking value:1
Right Tracking value:1
Left Tracking value:0
Center Tracking value:0
Right Tracking value:1
Left Tracking value:0
Center Tracking value:1
Right Tracking value:0
Left Tracking value:0
Center Tracking value:0
Autoscroll
Show timestamp
Newline

Test 2--Line Tracking Smart Car

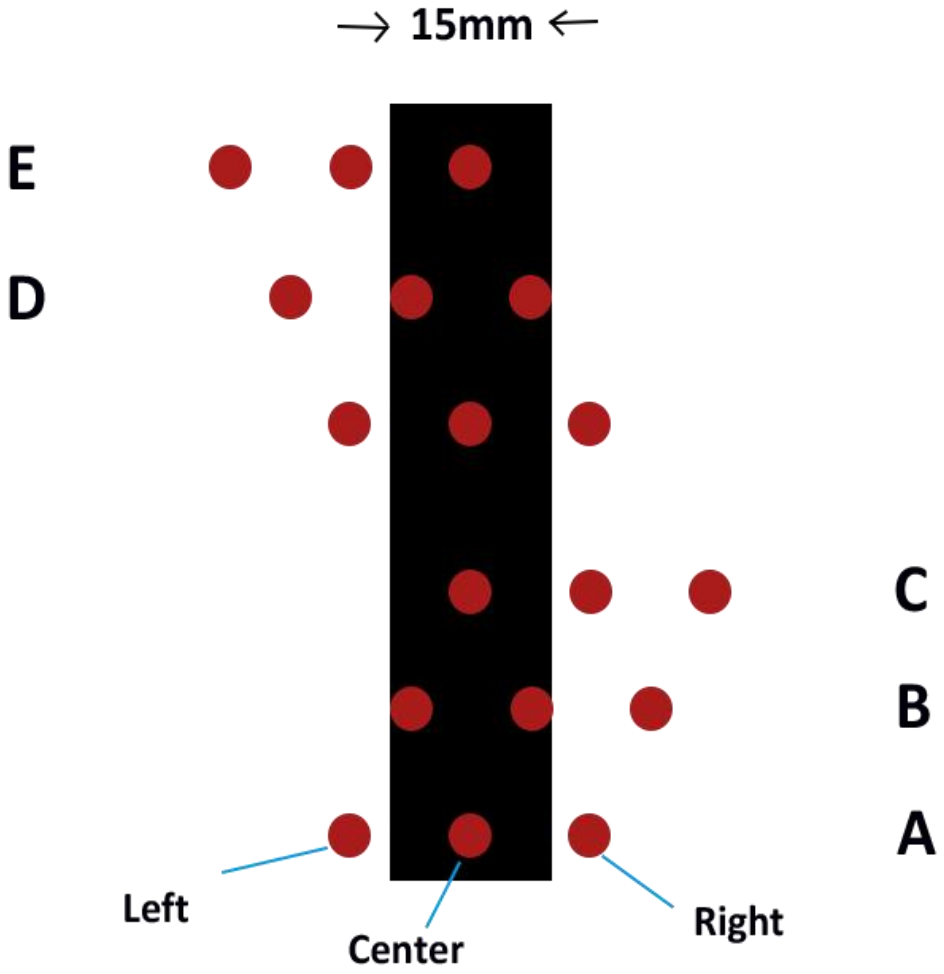
In the experimental test 2, we learned to combine the infrared line-following sensor with the motor to control the robot car to complete the line-following function.



Programming Thinking

The line inspection part of the robot car includes three infrared line tracking sensors, namely the left line tracking sensor, the

middle line tracking sensor, and the right line tracking sensor. A roll of black electrical tape is included in the kit parts. The width of the tape is 15mm, you can use it to plan the trajectory of the car. When the robot completes the line-following function, the following situations may occur.



Arduino Code

if you want to refer to the program we provide.Open this source program in Arduino_Code>Lesson_4>**Test_2_Line_Tracking_Smart_Car.ino**.

Program Analysis

When in the **A state**, only the middle line-following sensor detects the black line, and the robot car moves straight at a speed of 105.

```
if (Left_Tra_Value != Black && (Center_Tra_Value == Black && Right_Tra_Value != Black))
  {
    Move_Forward(105);
  }
```

When in the **B state**, the left line-following sensor and the center line-following sensor detect the black line, and the robot car rotates to the left at a speed of 95.

```
else if (Left_Tra_Value == Black && (Center_Tra_Value == Black && Right_Tra_Value != Black))
  {
    Rotate_Left(95);
  }
```

When in the **D state**, the right line-following sensor and the center line-following sensor detect the black line, and the robot car rotates to the right at a speed of 95.

```
 else if (Left_Tra_Value != Black && (Center_Tra_Value == Black && Right_Tra_Value == Black))
  {
     Rotate_Right(95);
  }
```

When in the **C state**, the left line-following sensor detects the black line, and the robot car rotates to the left at a speed of 120.

```
else if (Left_Tra_Value == Black && (Center_Tra_Value != Black && Right_Tra_Value != Black))
```

```
{
Rotate_Left(120);
}
```

When in the **E state**, the right line-following sensor detects the black line, and the robot car rotates to the right at a speed of 120%.

```
else if (Left_Tra_Value != Black && (Center_Tra_Value != Black && Right_Tra_Value == Black))
{
  Rotate_Right(120);
}
```

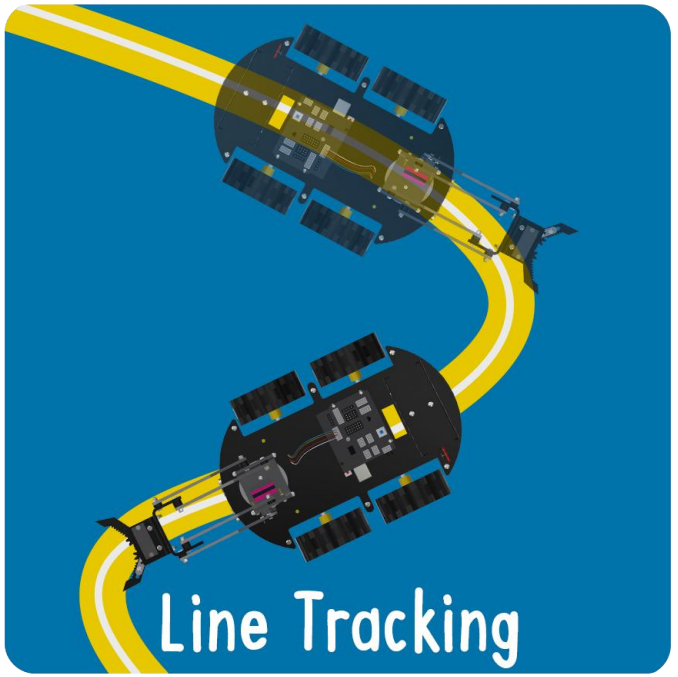
When the three line-following sensors detect the black line at the same time, the robot car stops.

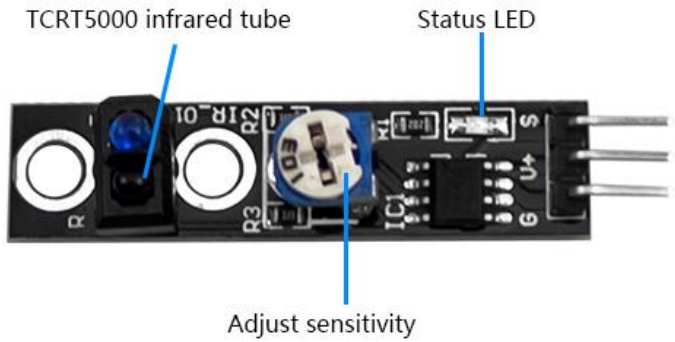
```
else if (Left_Tra_Value == Black && (Center_Tra_Value == Black && Right_Tra_Value == Black))
    {
    Stop();
    }
```

What will you see

Upload the test code to UNO R3 control board, turn the POWER switch ON.Then the smart car will move along the black line.

Note:The floors of different materials in the home have different degrees of light reflection. You can adjust the potentiometer on the line-following sensor to change the response sensitivity. This can make the car follow the black line more smoothly.





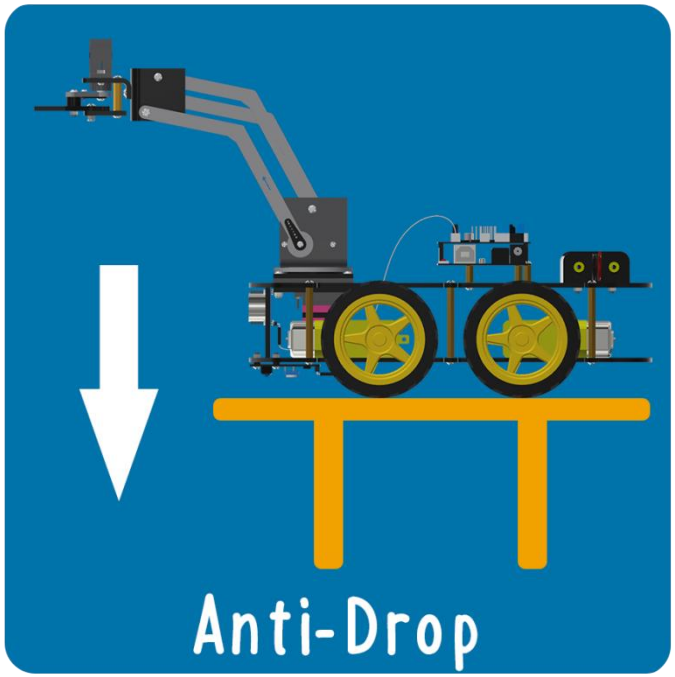
Lesson 5 Anti-fall Robot Car

If you test the infrared line patrol sensor more deeply, you will find that the detection range is between 0-3cm.

When the sensor detects a non-black object, it outputs a signal "0" and the indicator light on the sensor lights up;

when the sensor does not detect an object, it outputs a signal "1" and the indicator light on the sensor turns off.

If the robot car is placed on the desktop to move, the infrared line-following sensor can help the robot car to determine whether it has reached the edge of the desktop. If it is, the robot car will back up and turn to avoid falling off the table.



What is line tracking sensor

The tracking sensor is actually an infrared sensor. The component used here is the TCRT5000 infrared tube. Its working principle is to use the different reflectivity of infrared light to the color, then convert the strength of the reflected signal into a current signal. During the process of detection, black is active at HIGH level, but white is active at LOW level. The detection height is 0-3 cm.

By rotating the adjustable potentiometer on the sensor, it can adjust the detection sensitivity of the sensor.

How to use the line tracking sensor

Operating Voltage: 3.3-5V (DC)

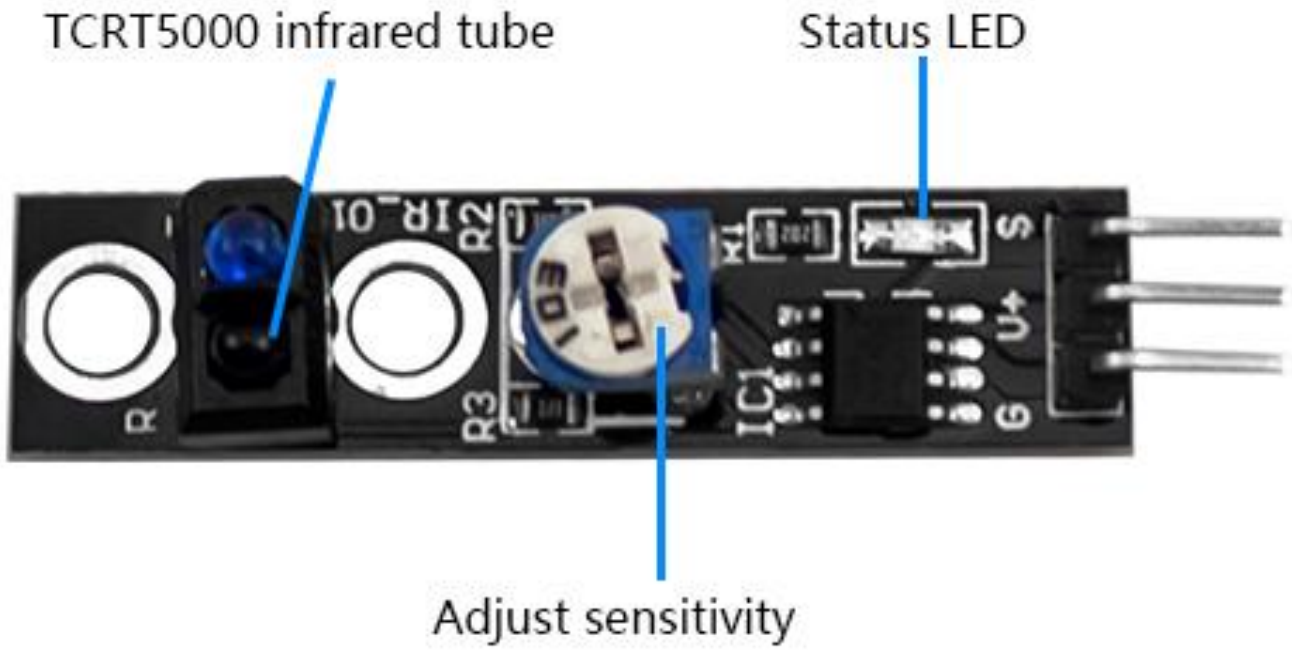
Interface: G(GND) V+(VCC) S(Signal)

Output Signal: Digital signal

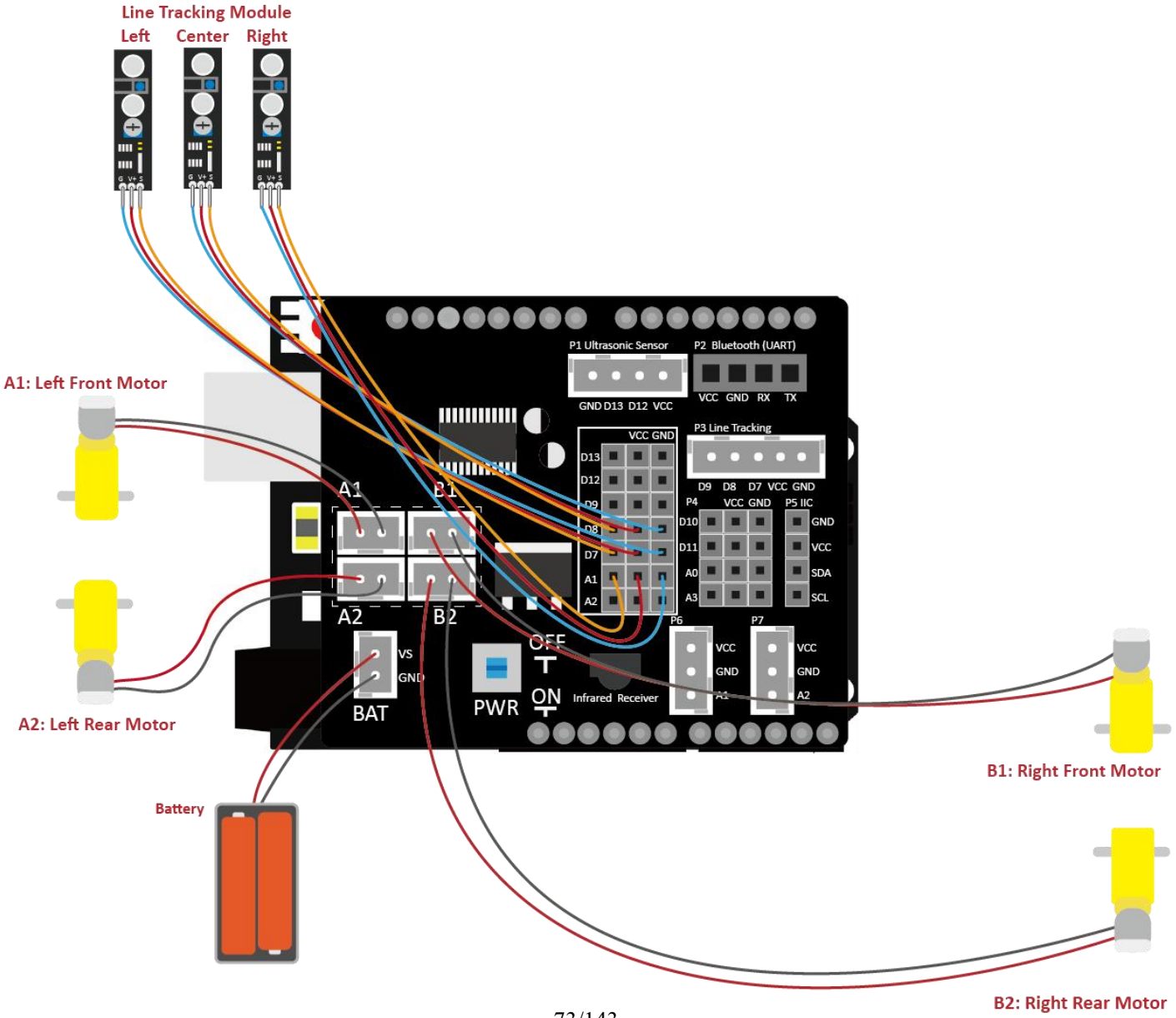
Detection Height: 0-3 cm

Special note: before testing, turn the potentiometer on the sensor to adjust the detection sensitivity. When adjust the LED at

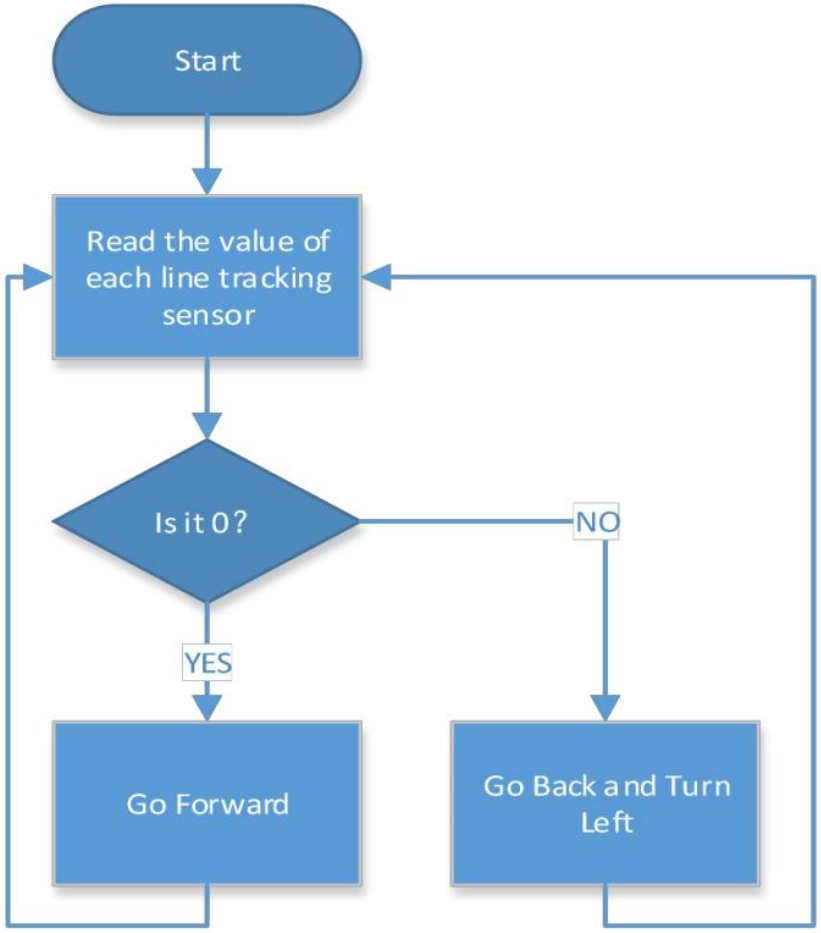
the threshold between ON and OFF, the sensitivity is the best.



How to connect the circuit



Programming Thinking

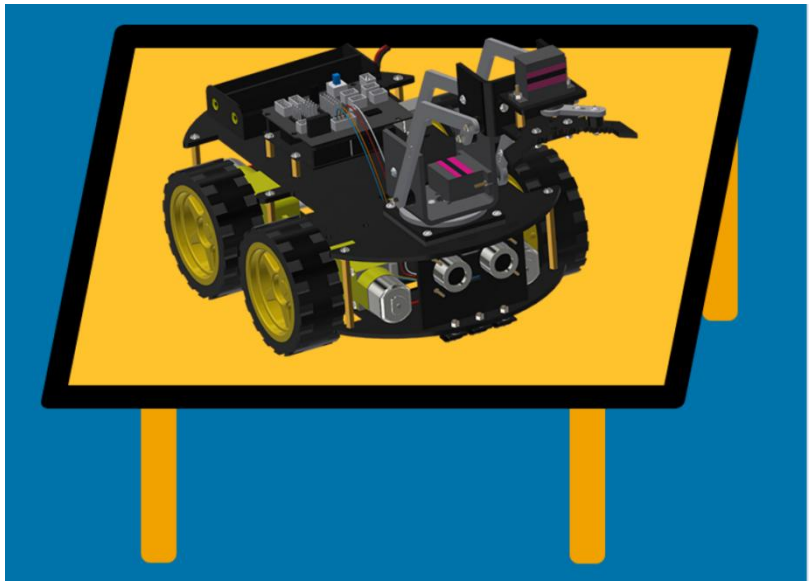


Arduino Code

if you want to refer to the program we provide.Open this source program in Arduino_Code>Lesson_5>**Test_1_Anti_Drop_Robot_Car.ino**.

What will you see

Upload the test code to UNO R3 control board.Use black tape to draw a fixed area on the table, and the robot car will move in the fixed area.The smart car will go back and turn left once it is close to edge of desk.



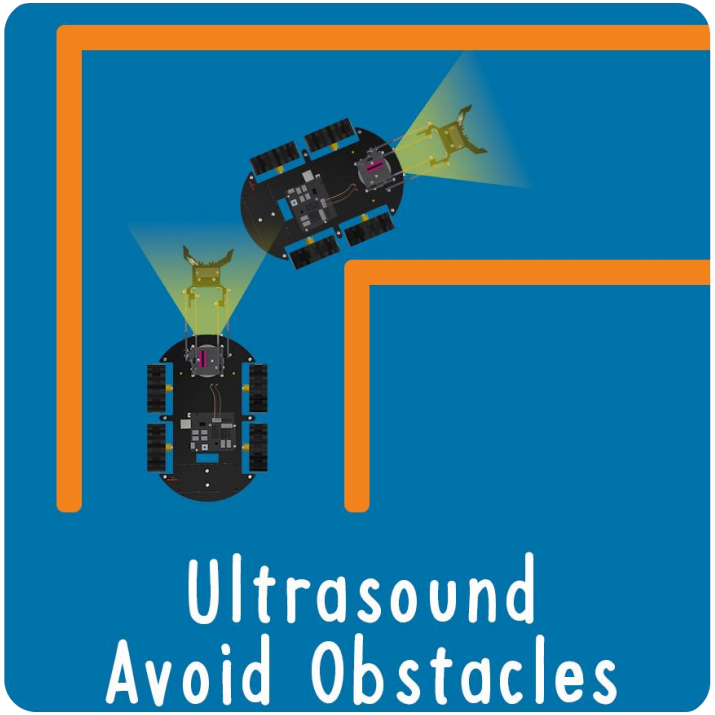
Lesson 6 Ultrasonic Obstacle Avoidance Robot Car

Overview

In this lesson, we will complete the test of 2 experimental codes.

In the second experimental test 1, we will learn to use the ultrasonic module to measure distance.

In the experimental test 2, The ultrasonic sensor is used to measure the distance of the obstacle directly in front, and help the robot car to avoid the obstacle.

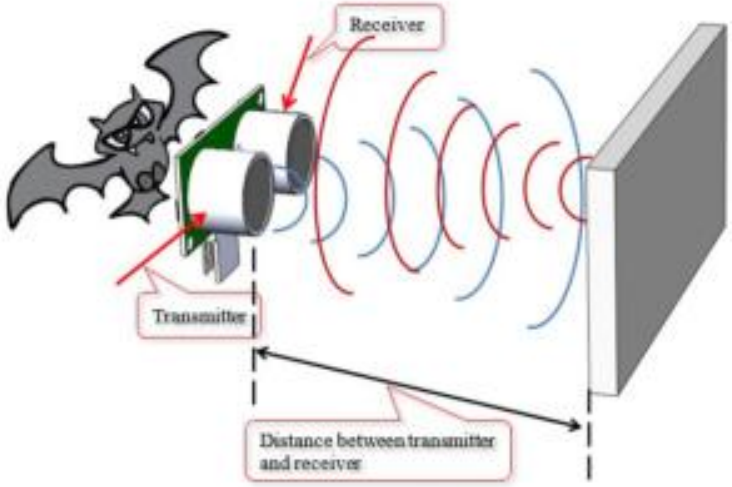


What is an ultrasonic sensor

Review the ultrasonic sensor from the previous lesson. It works like a bat's eye. Determine the distance of obstacles in front after receiving and receiving high-frequency sound waves.

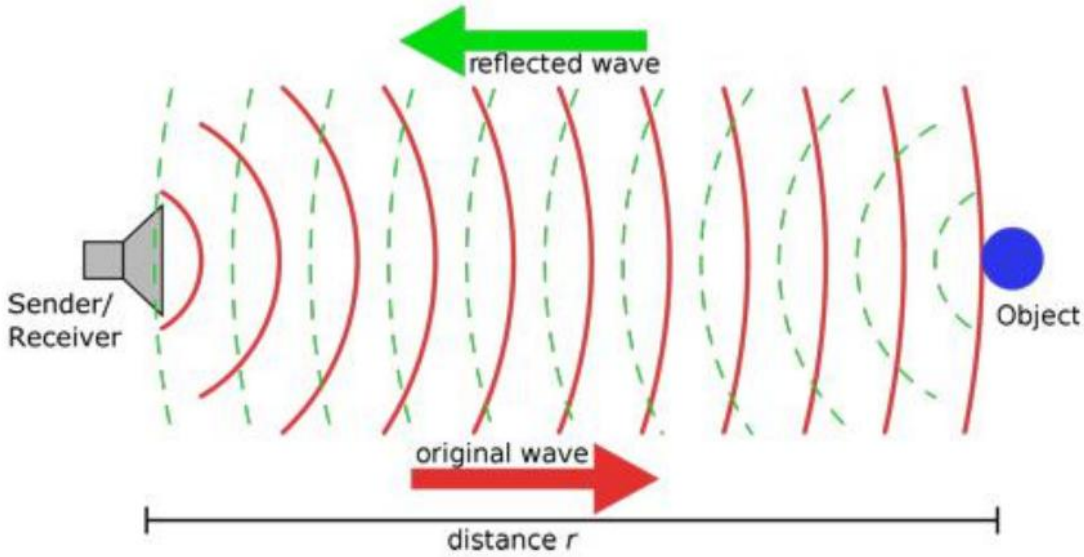
As the following picture shown, it is our ultrasonic module. It has two something like eyes.

One is transmitting end, the other is receiving end.



The HC-SR04 ultrasonic sensor uses sonar to determine distance to an object like what bats do. It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package. It comes complete with ultrasonic transmitter and receiver modules.The HC-SR04 or the ultrasonic sensor is being used in a wide range of electronics projects for creating obstacle detection and distance measuring application as well as various other applications. Here we have brought the simple method to measure the distance with arduino and ultrasonic sensor

and how to use ultrasonic sensor with arduino.



Specification

Power Supply :+5V DC

Quiescent Current : <2mA

Working Current: 15mA

Effectual Angle: <15° 92

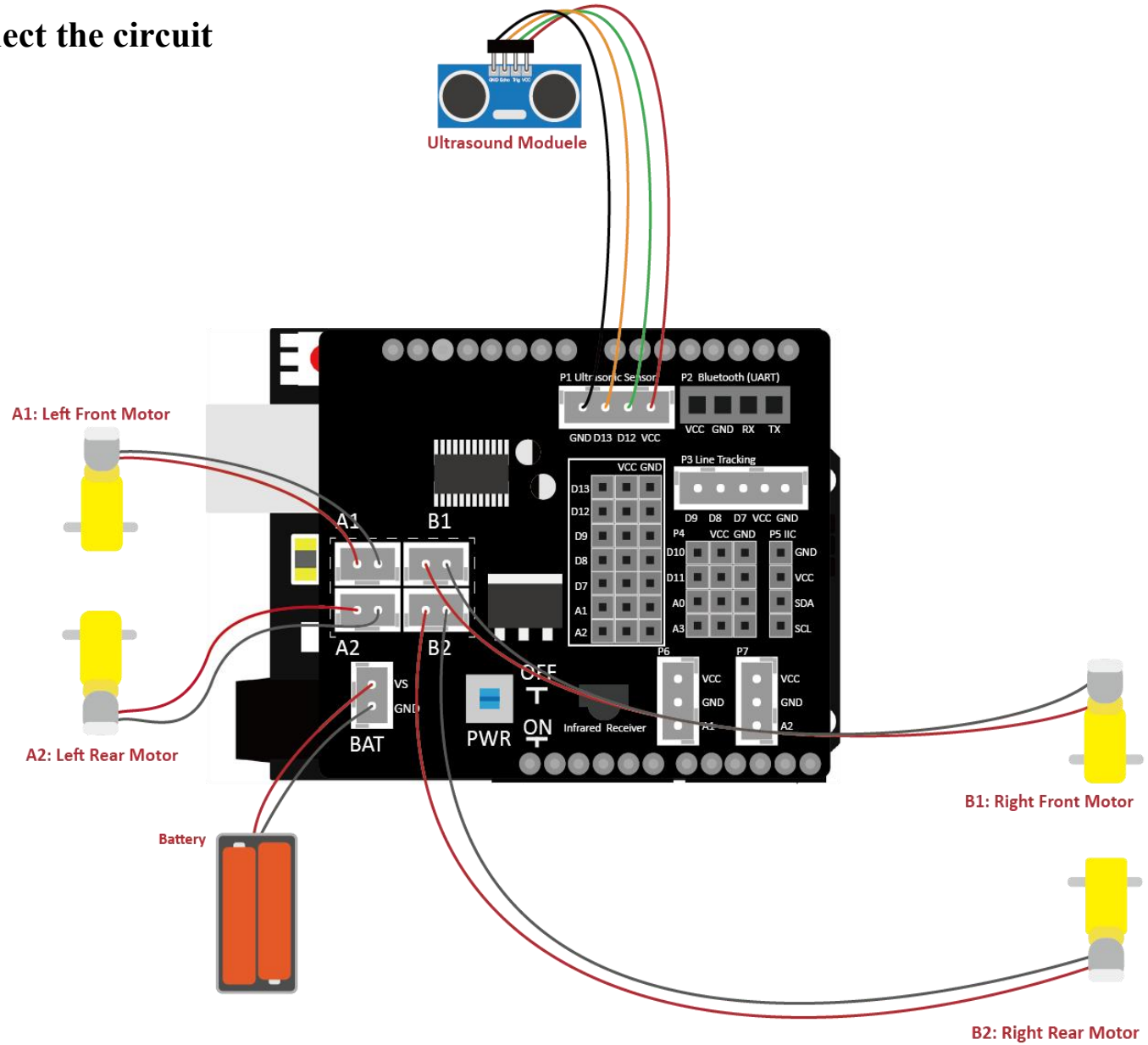
Ranging Distance : 2cm – 400 cm

Resolution : 0.3 cm

Measuring Angle: 30 degree

Trigger Input Pulse width: 10uS

How to connect the circuit



Let's program

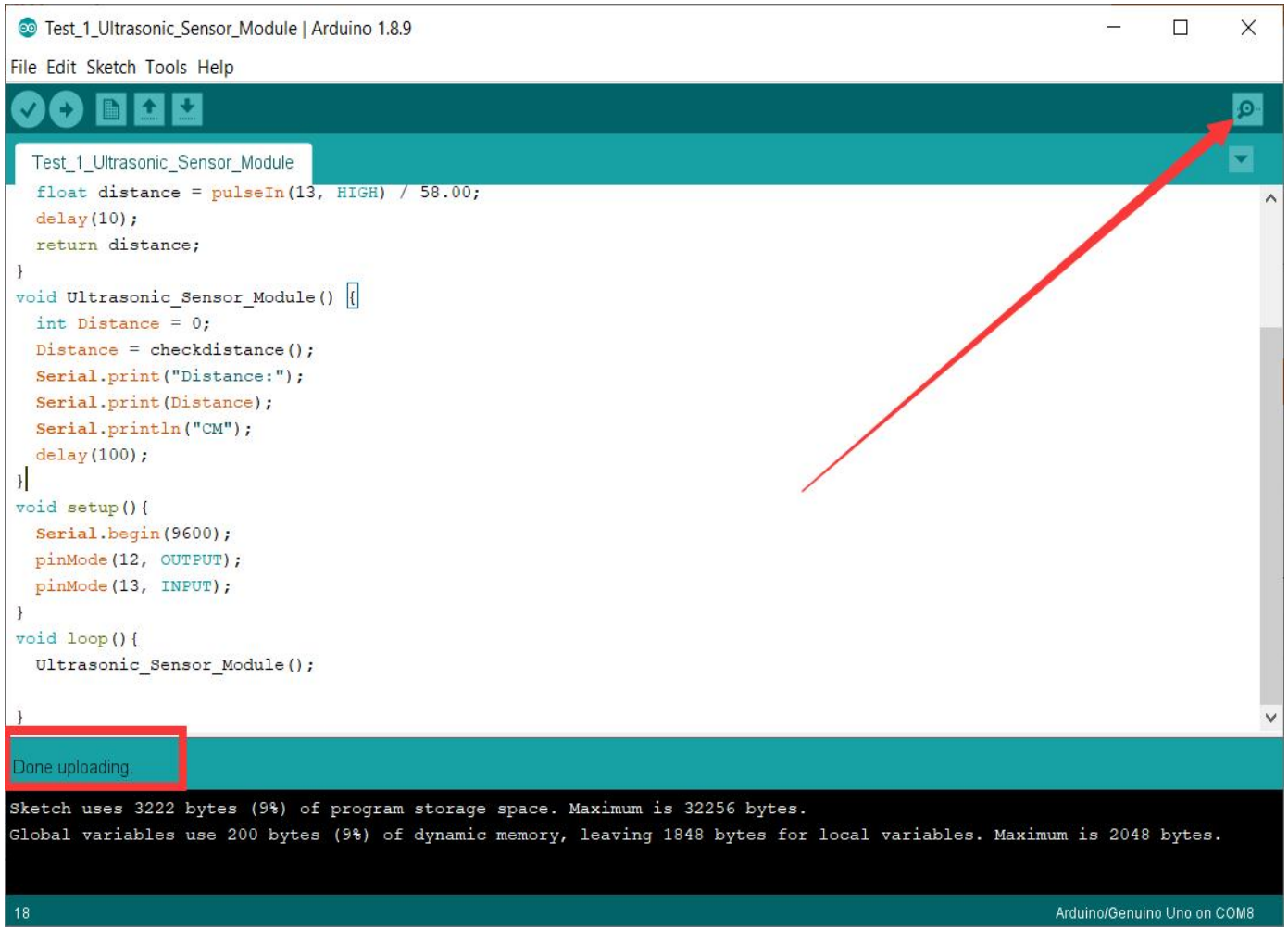
Test 1--Ultrasonic Sensor Module

In Experimental Test 1, we will learn how to control the ultrasonic sensor, and display the distance measured by the ultrasonic sensor on the serial monitor.

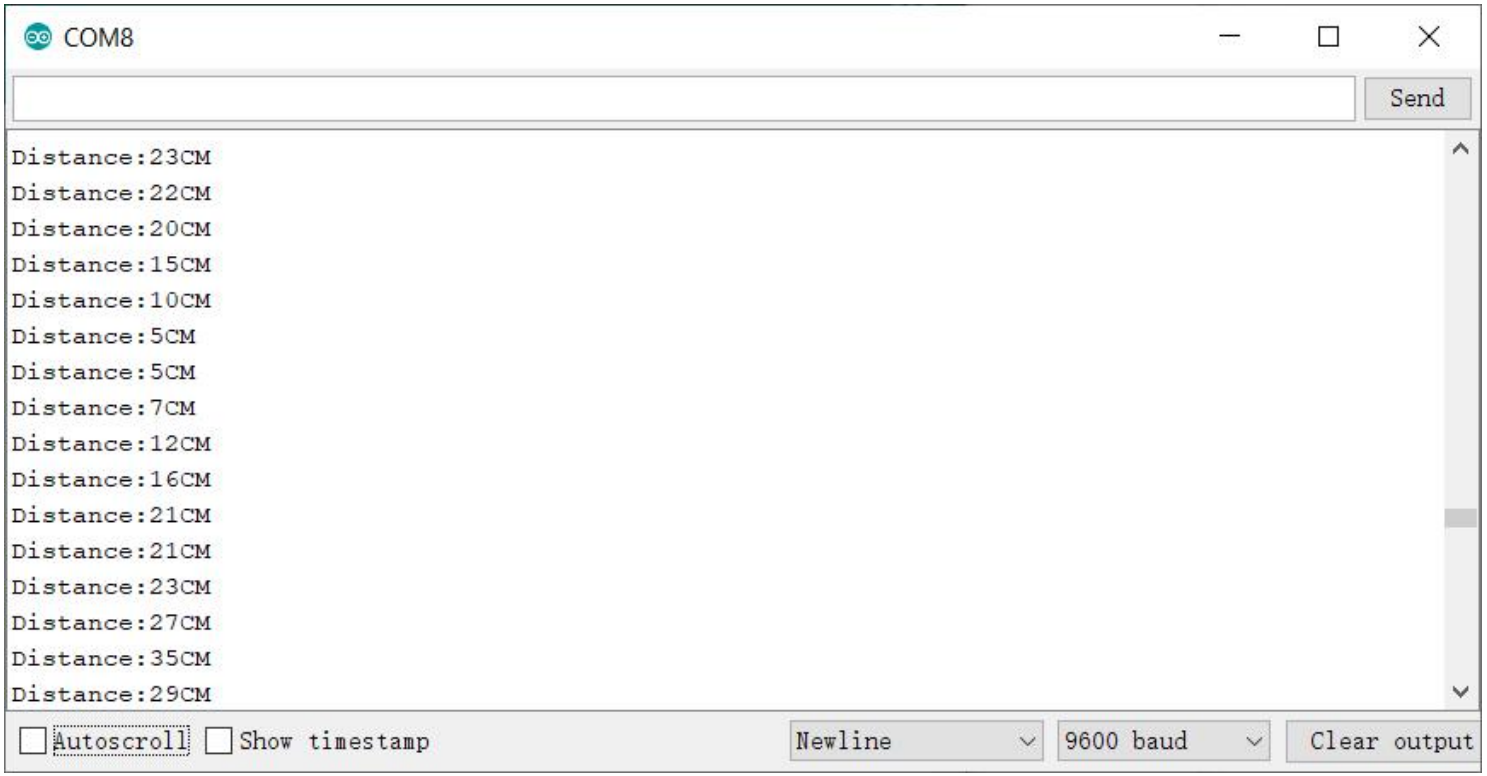
Arduino Code

if you want to refer to the program we provide.open Arduino IDE software and Open this source program in Arduino_Code>Lesson_6>**Test_1_Ultrasonic_Sensor_Module.ino**

After uploading the code, click the button in the upper right corner to open the serial monitor to view the measured distance


Test_1_Ultrasonic_Sensor_Module | Arduino 1.8.9
File Edit Sketch Tools Help
Test_1_Ultrasonic_Sensor_Module
float distance = pulseIn(13, HIGH) / 58.00;
delay(10);
return distance;
}
void Ultrasonic_Sensor_Module() {
int Distance = 0;
Distance = checkdistance();
Serial.print("Distance:");
Serial.print(Distance);
Serial.println("CM");
delay(100);
}
void setup(){
Serial.begin(9600);
pinMode(12, OUTPUT);
pinMode(13, INPUT);
}
void loop(){
Ultrasonic_Sensor_Module();
}
Done uploading.
Sketch uses 3222 bytes (9%) of program storage space. Maximum is 32256 bytes.
Global variables use 200 bytes (9%) of dynamic memory, leaving 1848 bytes for local variables. Maximum is 2048 bytes.
18
Arduino/Genuino Uno on COM8

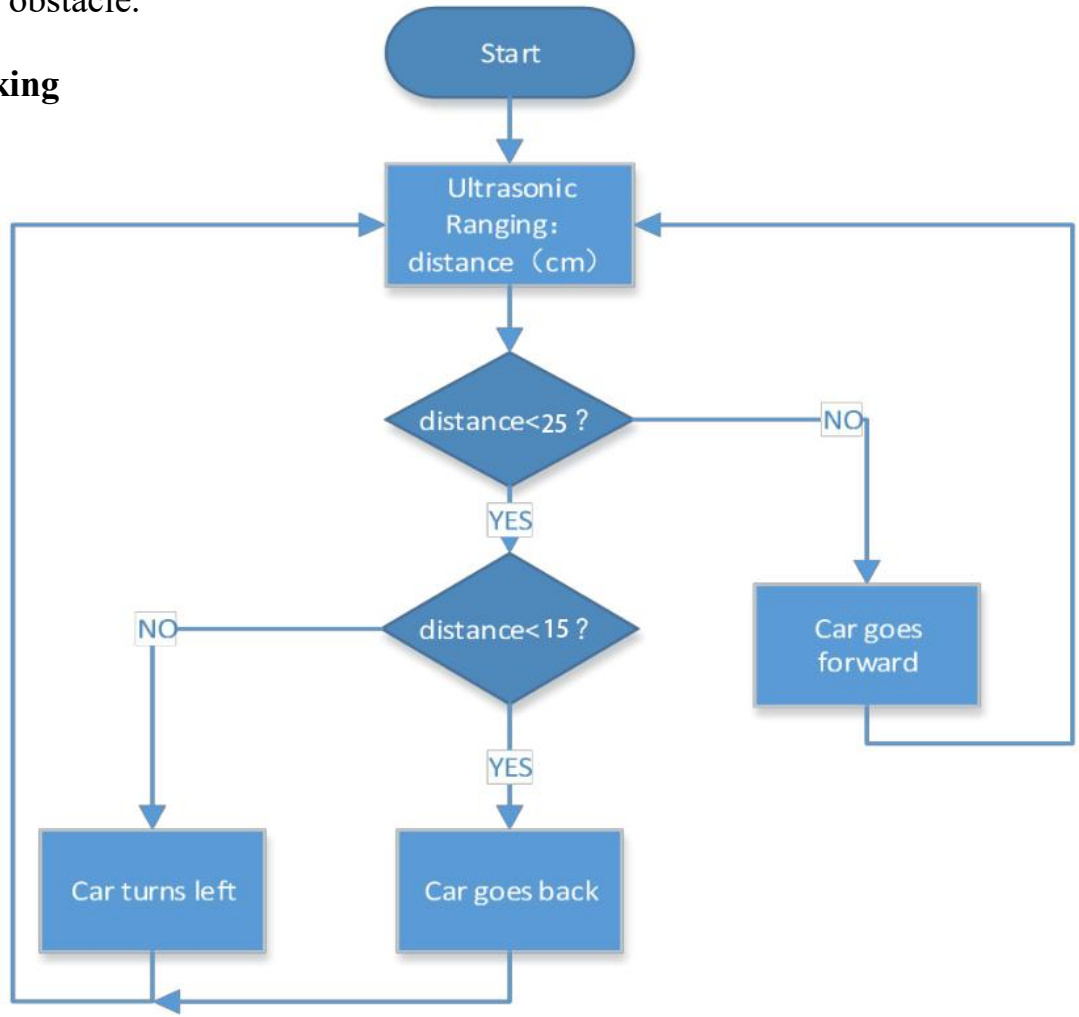
Then you can see the data as blow:



Test 2--Ultrasonic_Obstacle_Avoidance_Robot_Car

In the experimental test 2, the ultrasonic sensor is used to measure the distance of the obstacle directly in front, and help the robot car to avoid the obstacle.

Programming Thinking

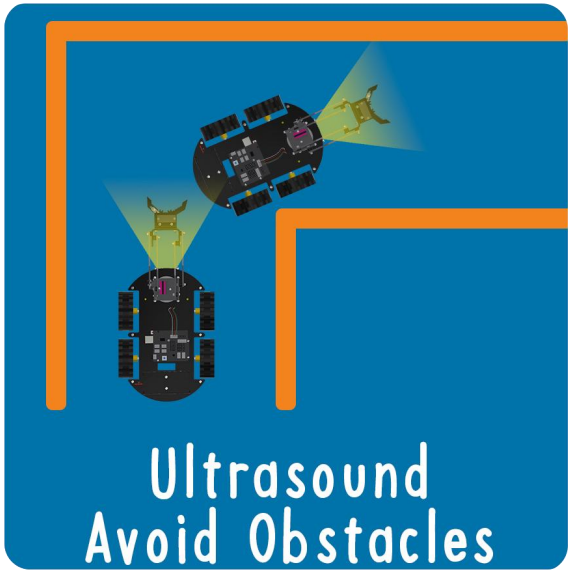


Arduino Code

if you want to refer to the program we provide.Open this source program in Arduino_Code>Lesson_6>**Test_2_Ultrasonic_Infrared_Obstacle_Avoidance_Robot_Car.ino.**

What will you see

Upload the code to UNO R3 control board,, and turn the POWER switch ON.The smart car will avoid the obstacle detected.



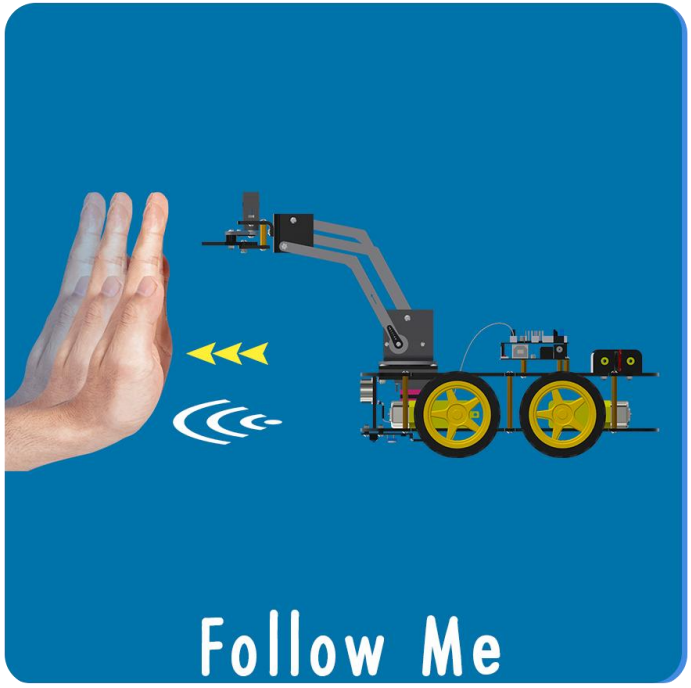
Lesson 7 Ultrasonic Follow Robot Car

Overview

In the last course, we learned how to use the ultrasonic module and infrared obstacle avoidance module. Similarly, use the ultrasonic module to help the robot car follow the moving objects ahead.

When the guided object is directly in front of the robot car, the ultrasonic sensor can detect the distance between the robot car and the guide object.When the distance is greater than 20mm, control the robot car to approach the guide;

When the distance is less than 15mm, control the robot car to move backward and keep a proper distance, 15mm<d<20mm.

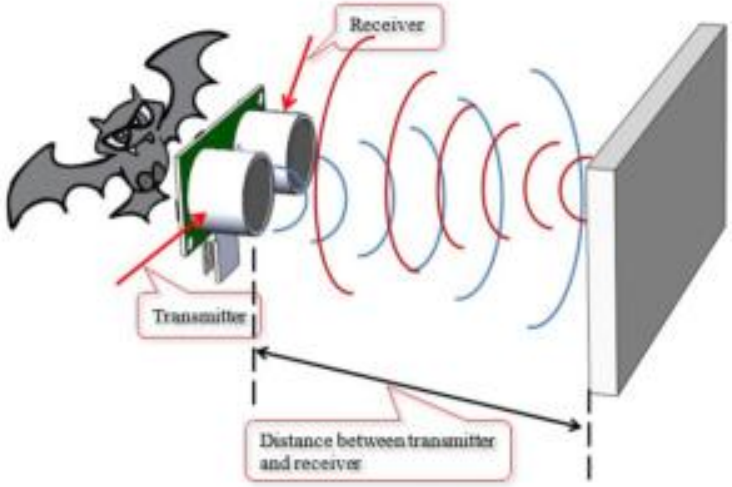


What is an ultrasonic sensor

Review the ultrasonic sensor from the previous lesson. It works like a bat's eye. Determine the distance of obstacles in front after receiving and receiving high-frequency sound waves.

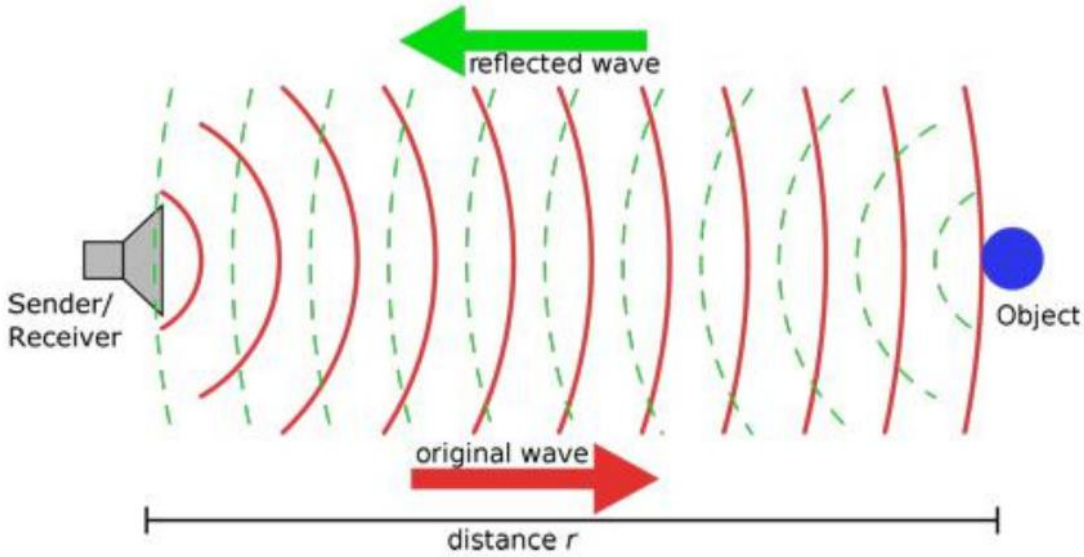
As the following picture shown, it is our ultrasonic module. It has two something like eyes.

One is transmitting end, the other is receiving end.

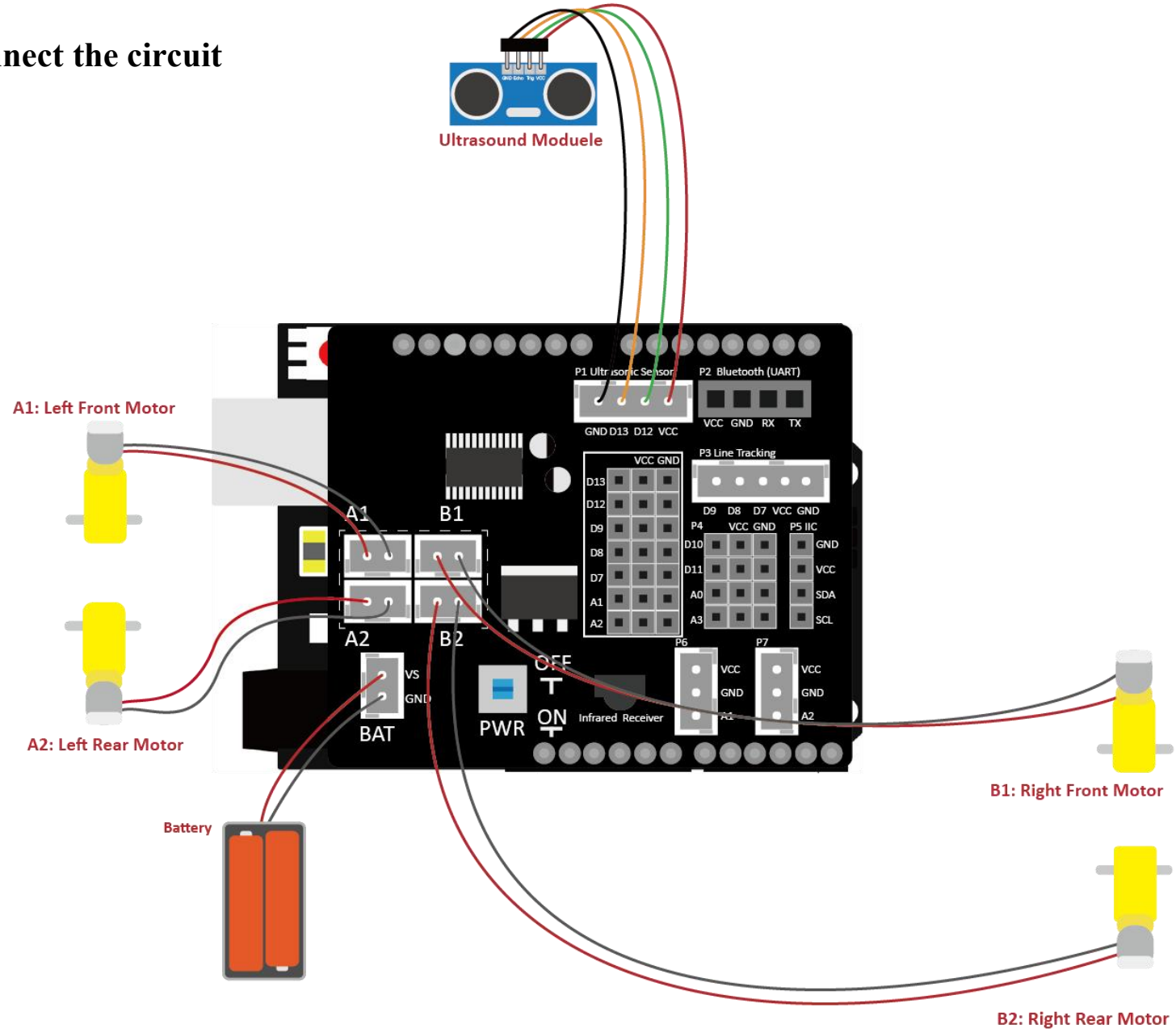


The HC-SR04 ultrasonic sensor uses sonar to determine distance to an object like what bats do. It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package. It comes complete with ultrasonic transmitter and receiver modules.The HC-SR04 or the ultrasonic sensor is being used in a wide range of electronics projects for creating obstacle detection and distance measuring application as well as various other applications. Here we have brought the simple method to measure the distance with arduino and ultrasonic sensor

and how to use ultrasonic sensor with arduino.



How to connect the circuit

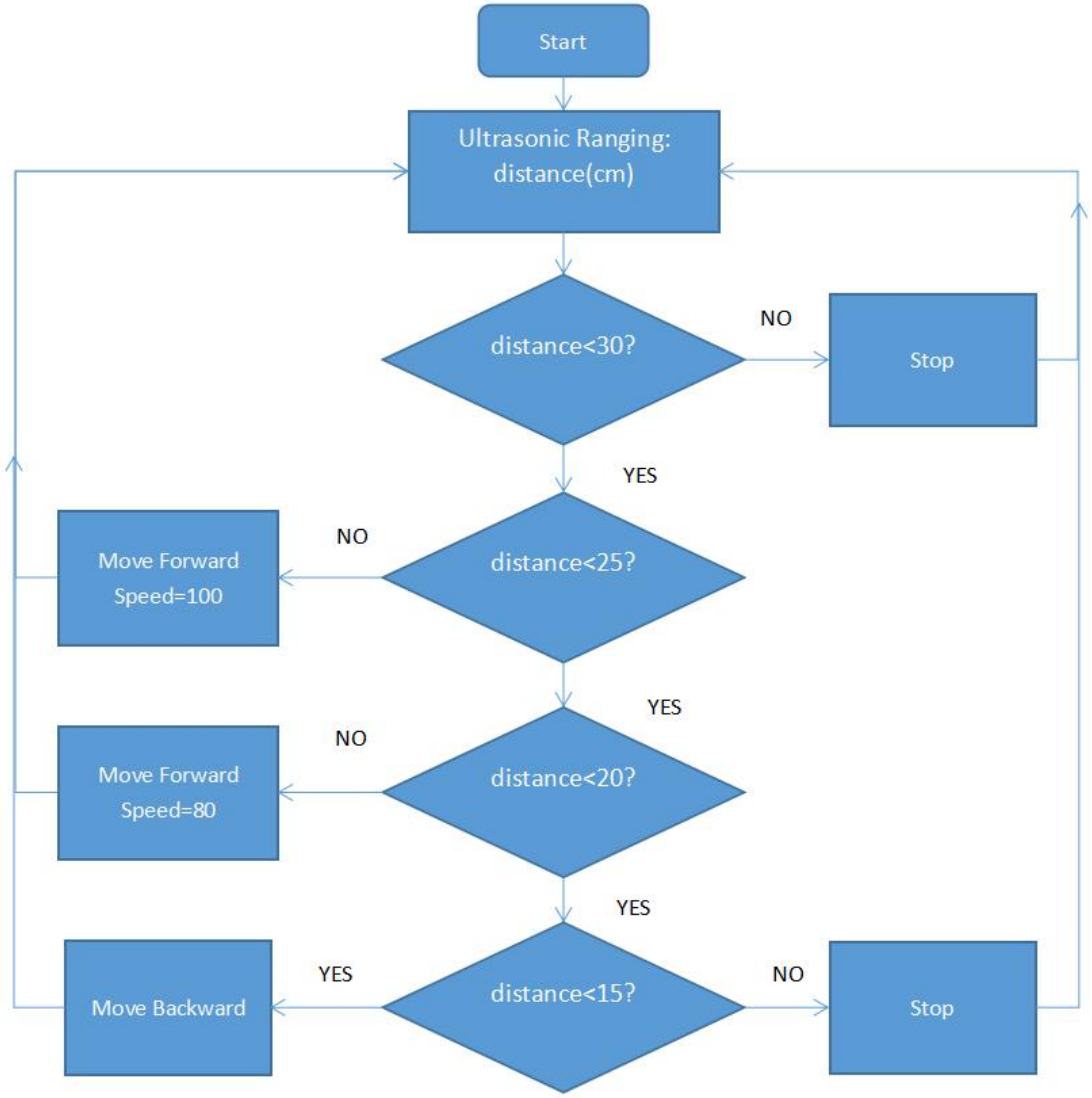


Let's program

Test 1--Ultrasonic Follow Robot Car

When the guided object is directly in front of the robot car, the ultrasonic sensor can detect the distance between the robot car and the guide object.When the distance is greater than 20mm, control the robot car to approach the guide;

When the distance is less than 15mm, control the robot car to move backward and keep a proper distance, 15mm<d<20mm.

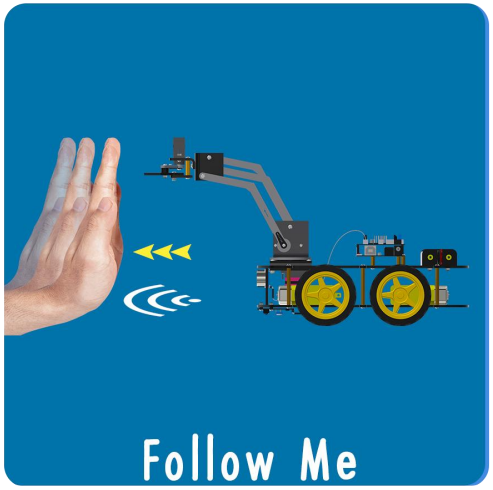

Start
Ultrasonic Ranging: distance(cm)
distance<30?
NO
Stop
YES
distance<25?
NO
Move Forward Speed=100
YES
distance<20?
NO
Move Forward Speed=80
YES
distance<15?
YES
Move Backward
NO
Stop

Arduino Code

if you want to refer to the program we provide.Open this source program in Arduino_Code>Lesson_7>**Test_1_Ultrasonic_Follow_Robot_Car.ino.**

What will you see

Upload the code to UNO R3 control board,, and turn the POWER switch ON.

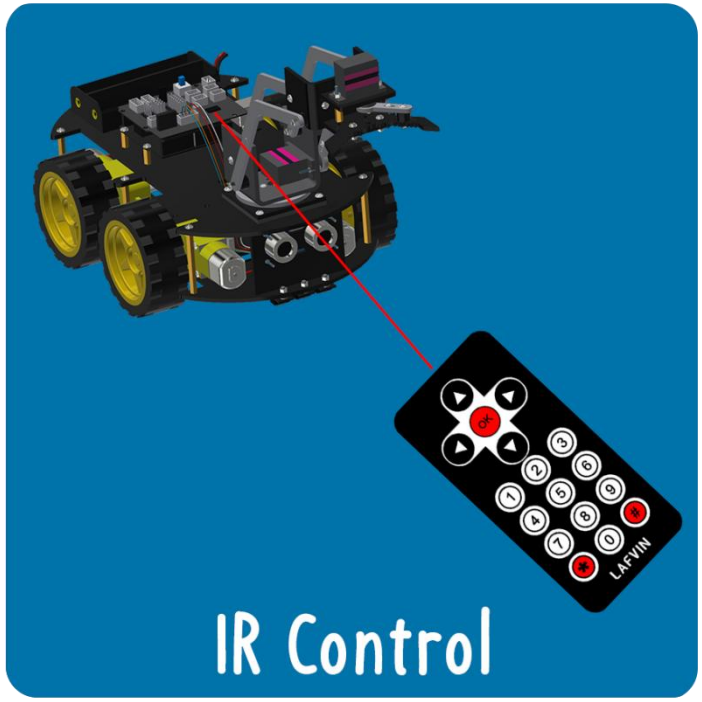


The smart car will follow the obstacle to move along the straight line, but it is unable to make a turn.

Lesson 8 Infrared Remote Control Robot Car

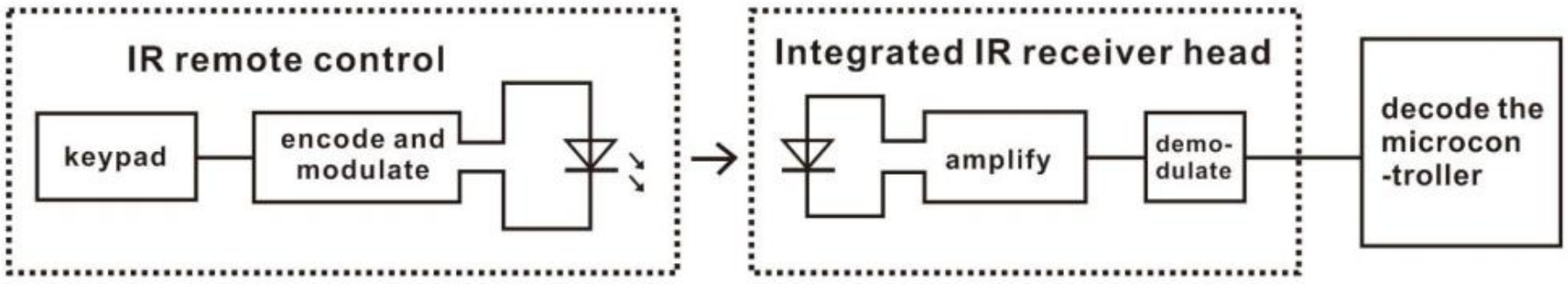
Overview

In this lesson, we will learn the infrared remote control, and use the infrared remote control to control the robot car go forward, backward, rotate left ,rotate right.At the same time, the infrared remote control can control the robotic arm.



What is an infrared remote control

There is no doubt that infrared remote control is commonly seen in our daily life. It's hard to imagine our world without it.An infrared remote control can be used to control a wide range of home appliances such as television, audio, video recorders and satellite signal receivers.Well, in the following let’s get a better understanding of the infrared remote control.Infrared remote control is composed of infrared transmitting and infrared receiving systems.That is, consist of an infrared remote control, an infrared receiver module and a microcontroller that can decode.You can refer to the figure below.

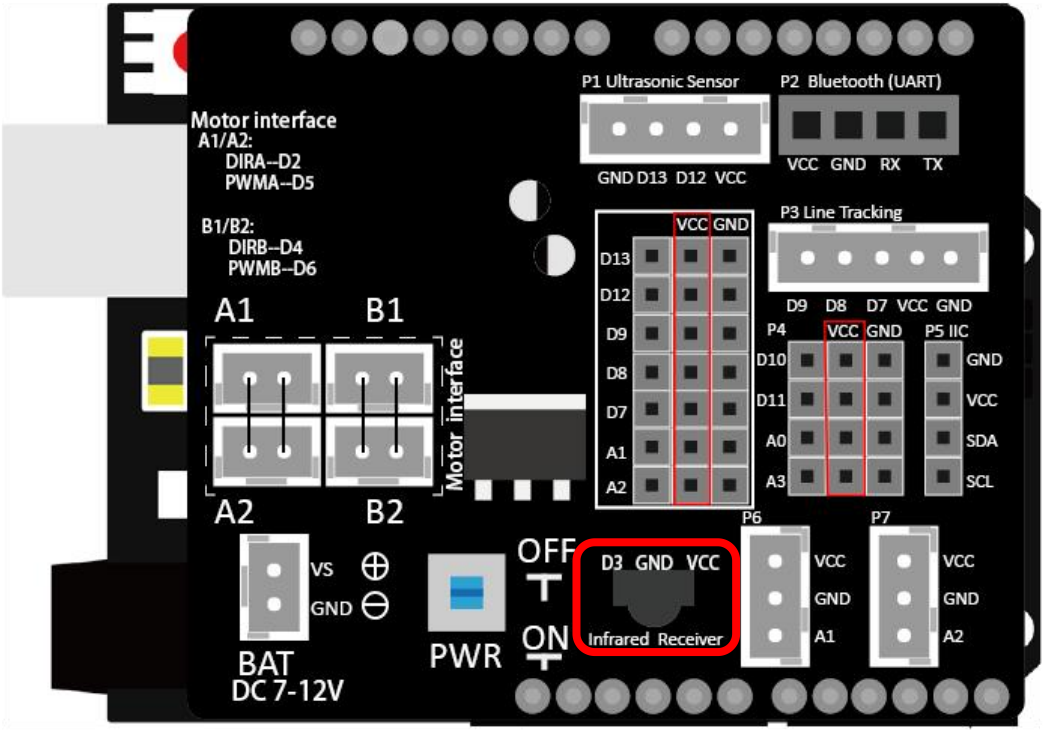


The 38K infrared carrier signal transmitted by an infrared remote controller is encoded by an encoding chip inside the remote controller. It is composed of a pilot code, user code, data code, and data inversion code.The time interval between pulses is used to distinguish whether it is a signal 0 or 1. (when the ratio of high level to low level is about 1:1, considered as signal 0.)

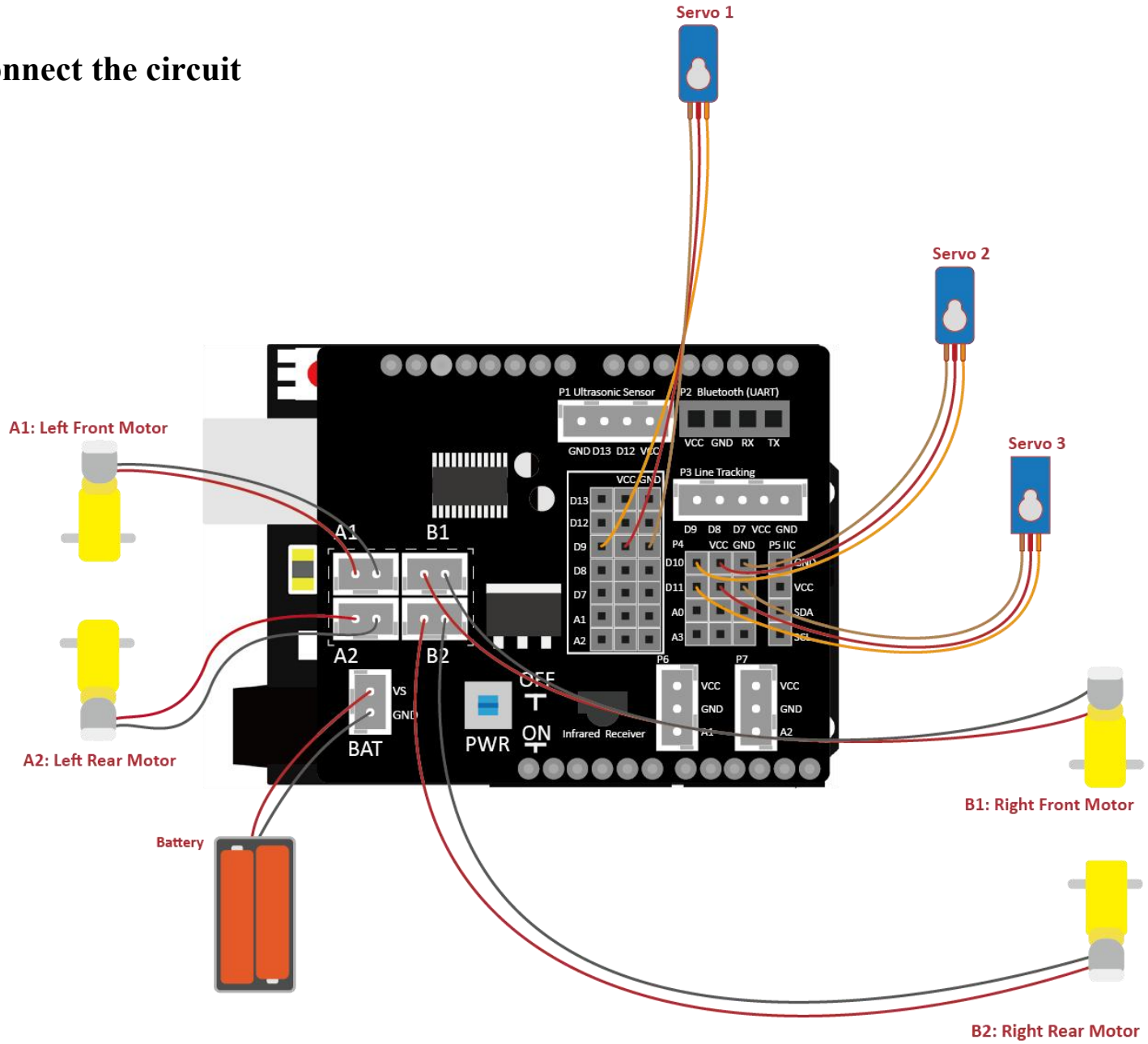
And the encoding is just well composed of signal 0 and 1.The user code of the same button on remote controller is unchanged. Using difference data distinguish the key pressed on the remote control. When press down a button on the remote control, it will send out an infrared carrier signal. And when infrared receiver receives that signal, its program will decode the carrier signal, and through different data codes, thus can judge which key is pressed.The microcontroller is decoded by an received signal 0 or 1 to determine which key is pressed by the remote control.

What is an infrared receiver

The robot shield comes with infrared receiver module. It is mainly composed of an infrared receiving head. This device integrates with reception, amplification and demodulation.Its inter IC has been demodulated, outputting Digital signal. Suitable for IR remote control and infrared data transmission.The data interface of the infrared receiver is connected to the D3 digital IO port.



How to connect the circuit



Let's program

Test 1--Infrared Remote Control Robot Arm Smart Car

In experimental test 1, learn to receive infrared remote control signals, and distinguish the corresponding remote control key value, and finally realize infrared remote control robot car.

Programming Thinking

Introduce the infrared remote control library file and the key value library file, otherwise the compilation will report an error.

```
#include "IR_remote.h"
#include "keymap.h"
```

This program block is to initialize the infrared receiver, and the port number is filled with "3".

The data interface of the infrared receiver is connected to the D3 digital IO port.

```
IRremote ir(3);
```

This sentence judges whether to press the set button, if the "UP" button is pressed, control the car to move forward.If you want to set other buttons, you only need to modify the button value" IR_KEYCODE_UP"

```
if (ir.getIrKey(ir.getCode(),1) == IR_KEYCODE_UP)
{
    Move_Forward(100);
    delay(300);
    Stop();
}
```

The key value of the remote control:

Key					
Key value	IR_KEYCODE_UP	IR_KEYCODE_DOWN	IR_KEYCODE_LEFT	IR_KEYCODE_RIGHT	IR_KEYCODE_OK
Key					
Key value	IR_KEYCODE_1	IR_KEYCODE_2	IR_KEYCODE_3	IR_KEYCODE_4	IR_KEYCODE_5
Key					
Key value	IR_KEYCODE_6	IR_KEYCODE_7	IR_KEYCODE_8	IR_KEYCODE_9	IR_KEYCODE_STAR
Key					
Key value	IR_KEYCODE_0	IR_KEYCODE_POUND			

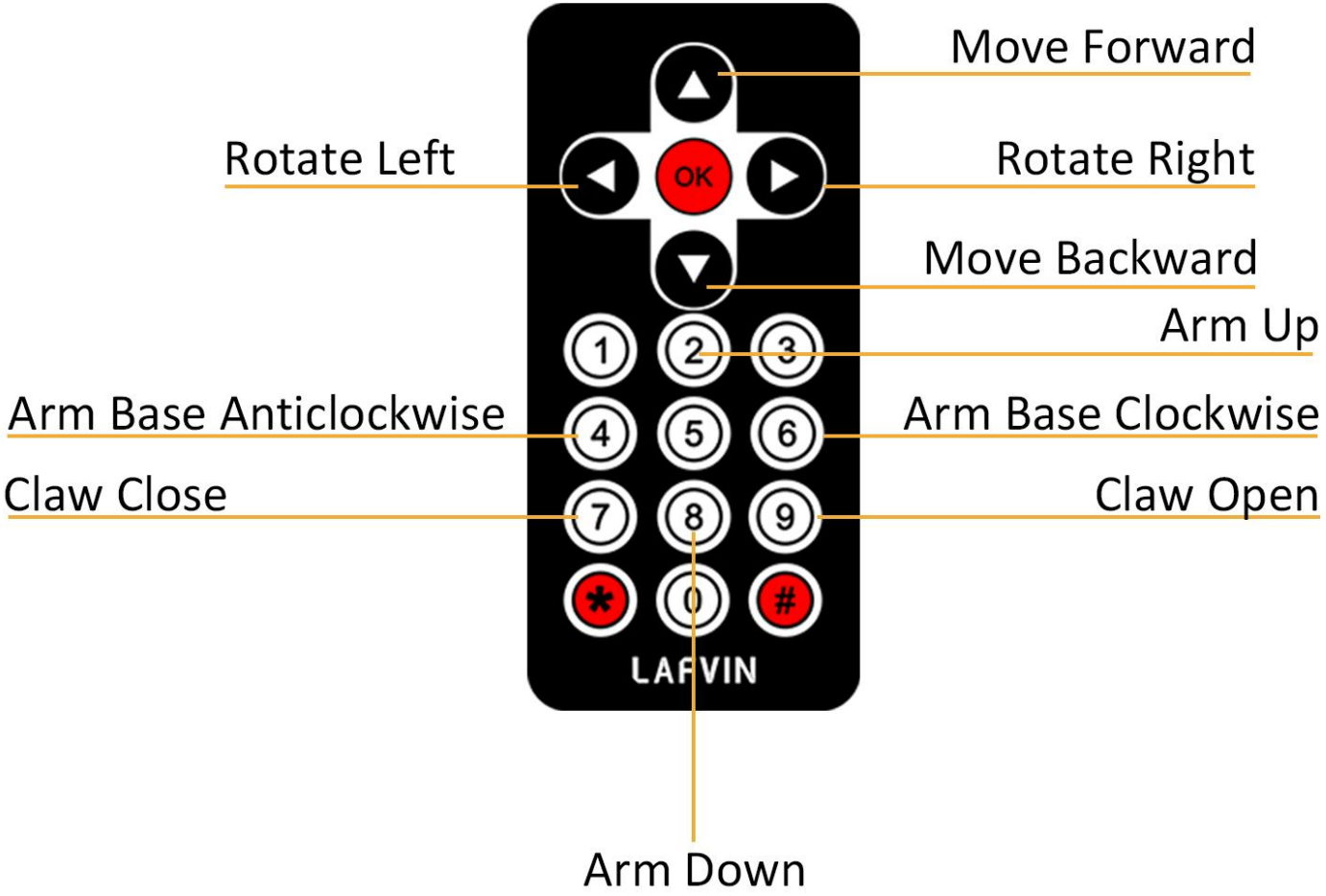
Arduino Code

if you want to refer to the program we provide.Open this source program in Arduino_Code>Lesson_8>**Test_1_Infrared_Remote_Control_Robot_Arm_Smart_Car.ino.**

What will you see

Upload the code to UNO R3 control board,, and turn the POWER switch ON. Use the remote control to control the direction of the robot car and the servo of the robotic arm.

Note: You need to press and hold the infrared remote control button to keep the motor moving. The motor will stop moving when the button is released.


Move Forward
Rotate Left
OK
Rotate Right
Move Backward
Arm Up
1
2
3
Arm Base Anticlockwise
4
5
6
Arm Base Clockwise
Claw Close
7
8
9
Claw Open
*
0
#
LAFVIN
Arm Down

Lesson 9 7-in-1 Multi-Function Robot Arm Smart Car

Overview

If you want to experience all the functions of the robot car directly, you can skip the previous lessons and directly refer to the steps in this lesson.In this lesson, we mainly study how to use the APP to control the robot smart car.

In Test 1, we learn the communication between the Bluetooth module and the mobile phone.

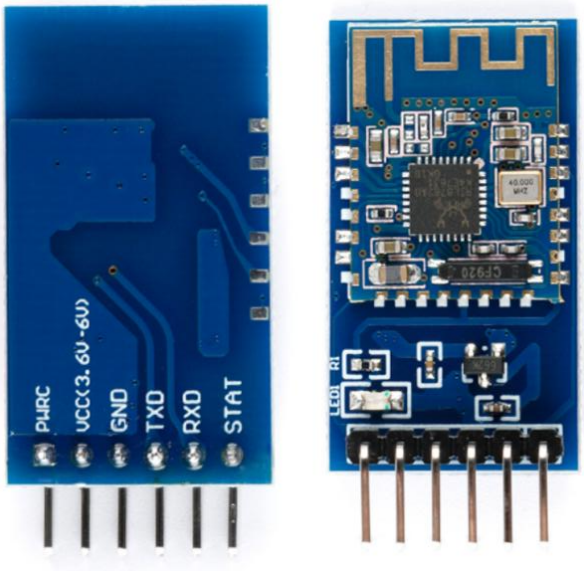
In Test 2, we can freely switch the various functional modes through the mobile phone app, such as Line Tracking,Ultrasonic Infrared Avoid Obstacles, Infrared Remote, Light Seeking,Follow Me,Remote Control, Gravity Sensor Remote Control.


Line Tracking
Gravity sensor
Follow Me
Auto Carry
Ultrasound
Avoid Obstacles
Anti-Drop
Developed by LAFVIN
only support android
4WD Robot Arm Smart Car
Actions
Speed

What is a Bluetooth module

Bluetooth Module JDY-16

The JDY-16 transparent transmission module is based on the Bluetooth-compatible 4.2 protocol standard, the working frequency band is 2.4GHZ range, the modulation method is GFSK, the maximum transmission power is 0db, the maximum transmission distance is 60 meters, and it adopts imported original chip design and supports users to modify the device through AT commands Name, service UUID, transmit power, pairing password and other instructions are convenient and flexible to use. The JDY-16 Bluetooth-compatible module can realize the data transmission between the module and the mobile phone or between the module and the module. The communication mode of UART or IIC can be selected through IO, and the Bluetooth-compatible can be quickly used for product application through simple configuration. Make BLE's application in products faster and more convenient.



Product Parameters:

Model: JDY-16

Working frequency: 2.4G

Transmitting power: 0db (maximum)

Communication interface: UART or IIC

Working voltage: 1.8V-3.6V

Working temperature: -40℃-80℃

Antenna: Built-in PCB antenna

Receiving sensitivity: -97dbm

Transmission distance: 60 meters

Module size: 19.6mm * 14.94 *2.6

Bluetooth-compatible version: BLE 4.2 (compatible with BLE4.0, BLE4.1)

Transparent transmission rate: 115200 bps/s

Wake-up state current: 4mA (with broadcast)

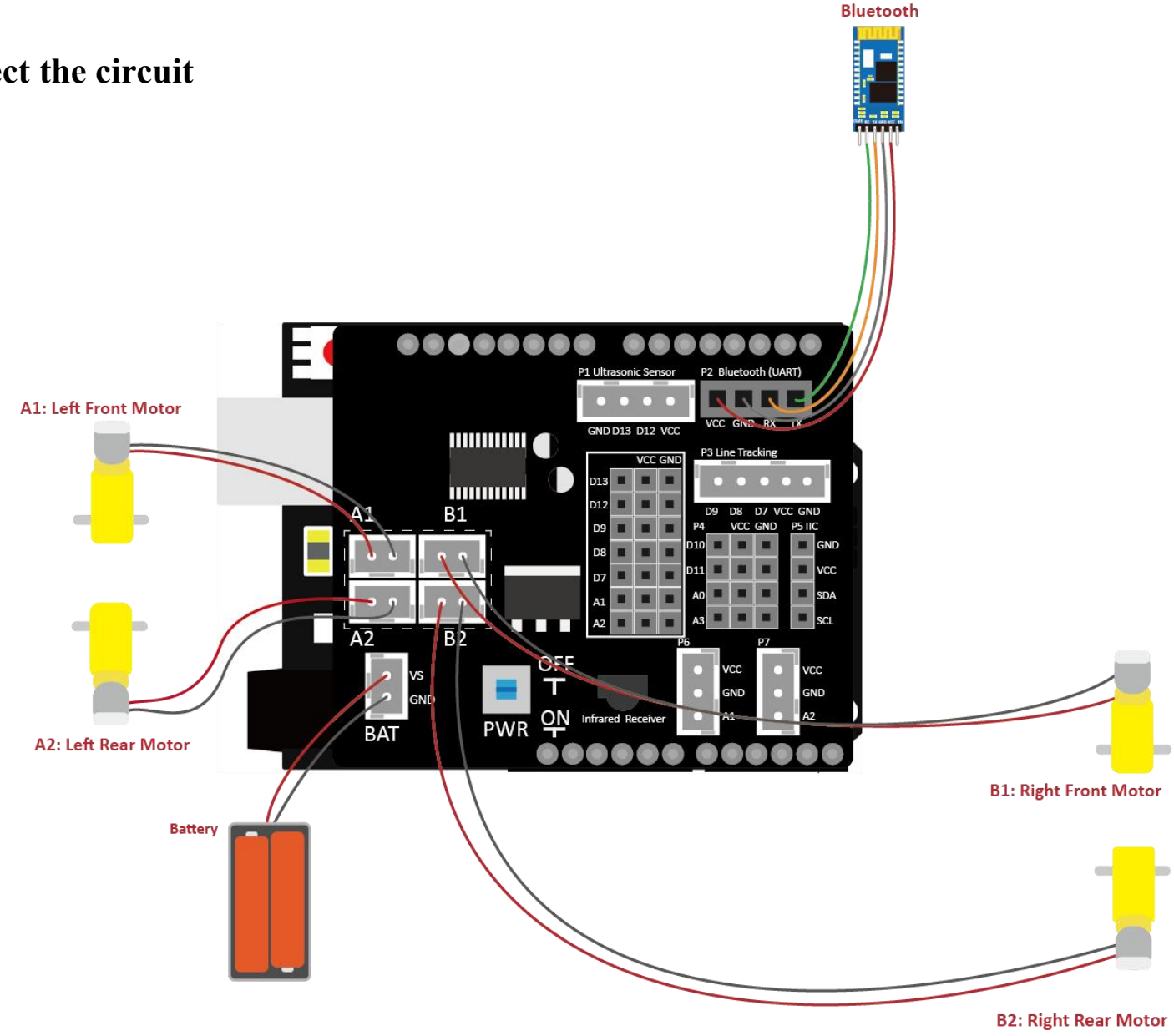
Light sleep state current: <300uA (with broadcast)

Deep sleep current: 1.8uA (no broadcast)

Command parameter saving: parameter configuration data is saved after power-off

STM welding temperature: <300℃

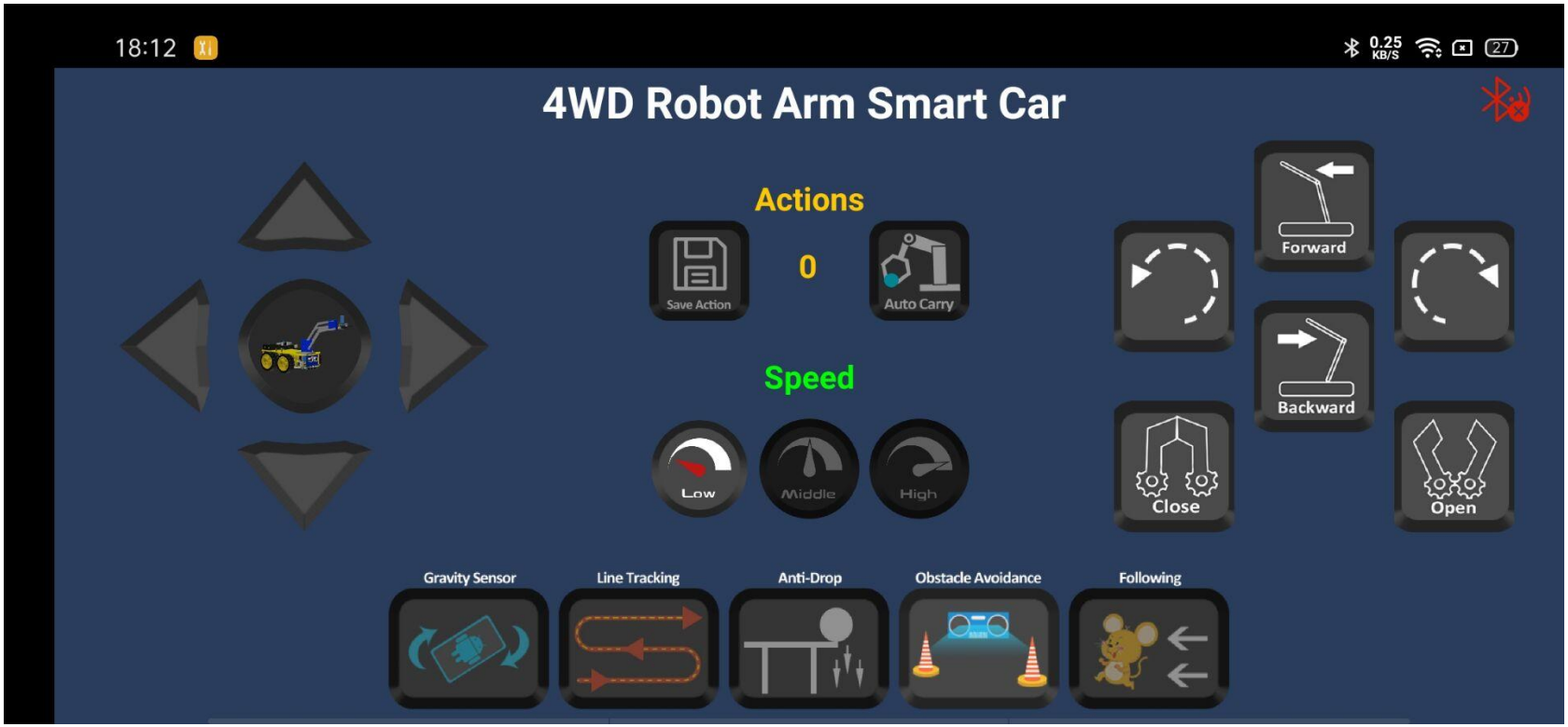
How to connect the circuit



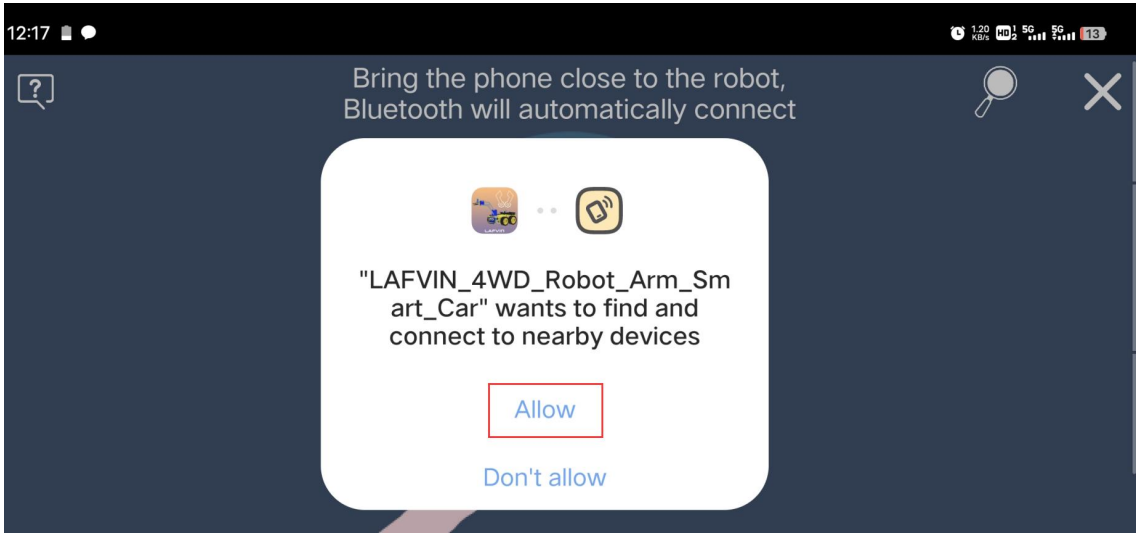
How to connect APP with Bluetooth module

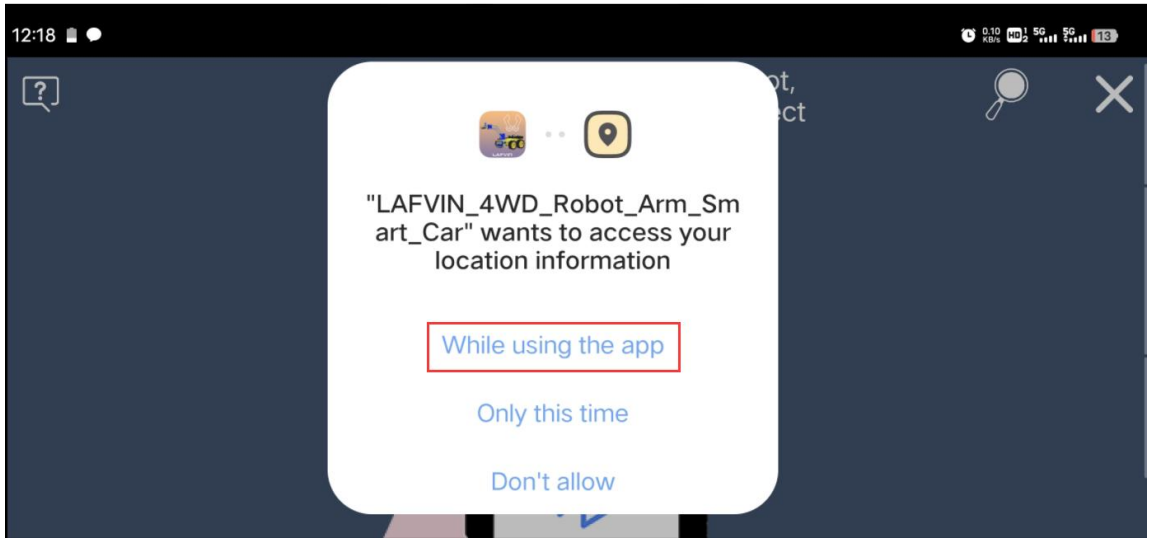
Firstly, copy the “LAFVIN_4WD_Robot_Arm_Smart_Car_V2_1.apk” file from the APP folder to your mobile phone and install it into an application software.The built-in Bluetooth module of the mobile phone needs to support Bluetooth 4.0 or higher.

Open the APP, you will see the following control interface.

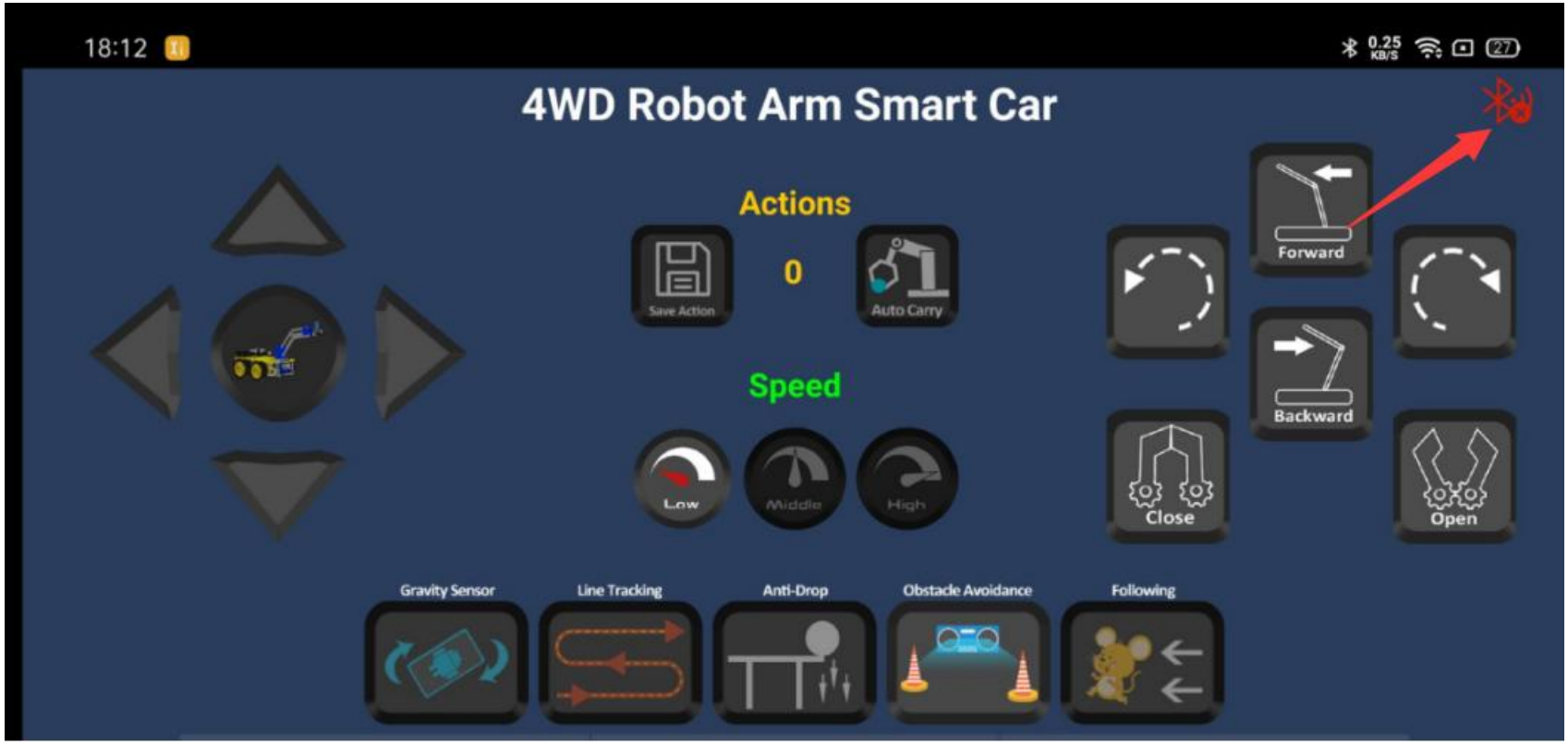


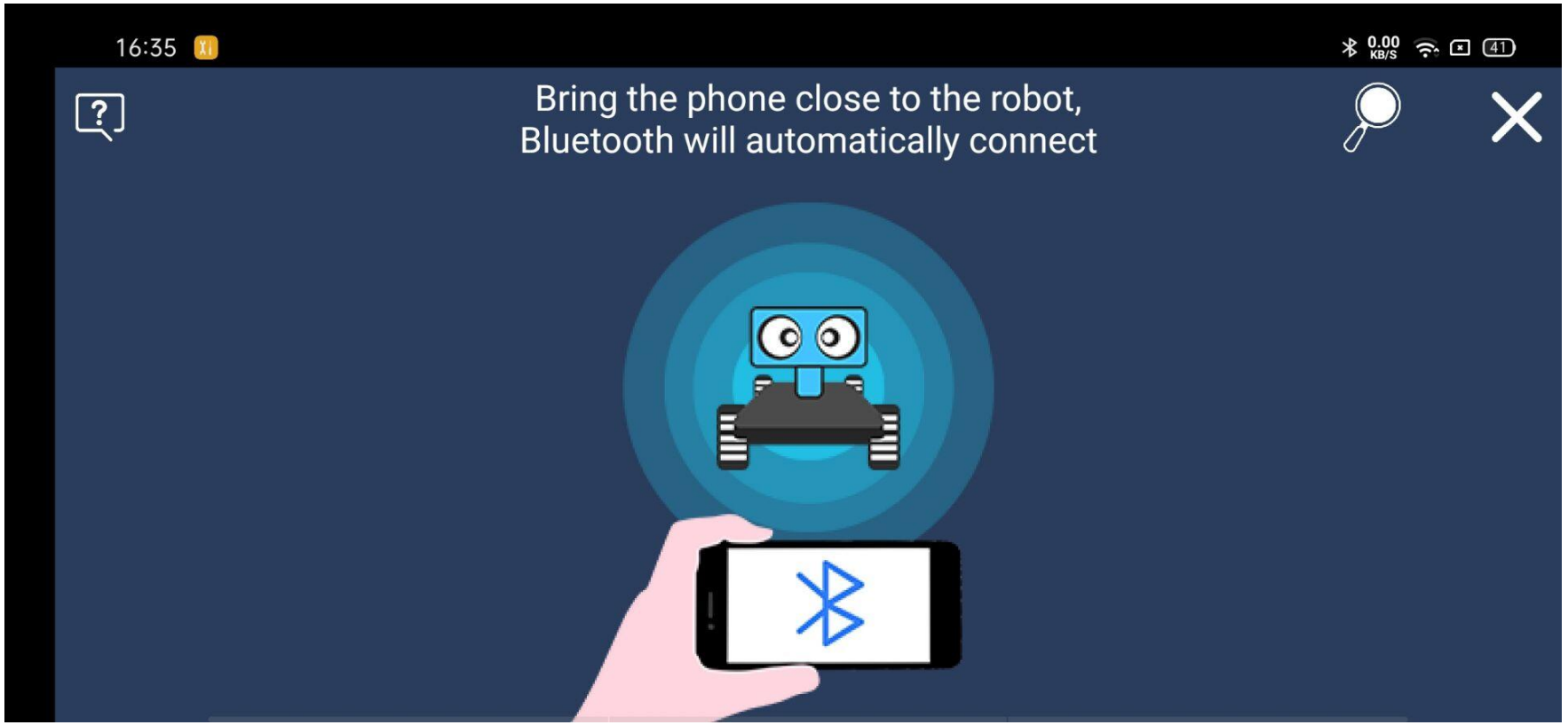
Important:Allows access to location permissions and permission to connect to nearby devices, otherwise it may not be possible to search for nearby Bluetooth devices.





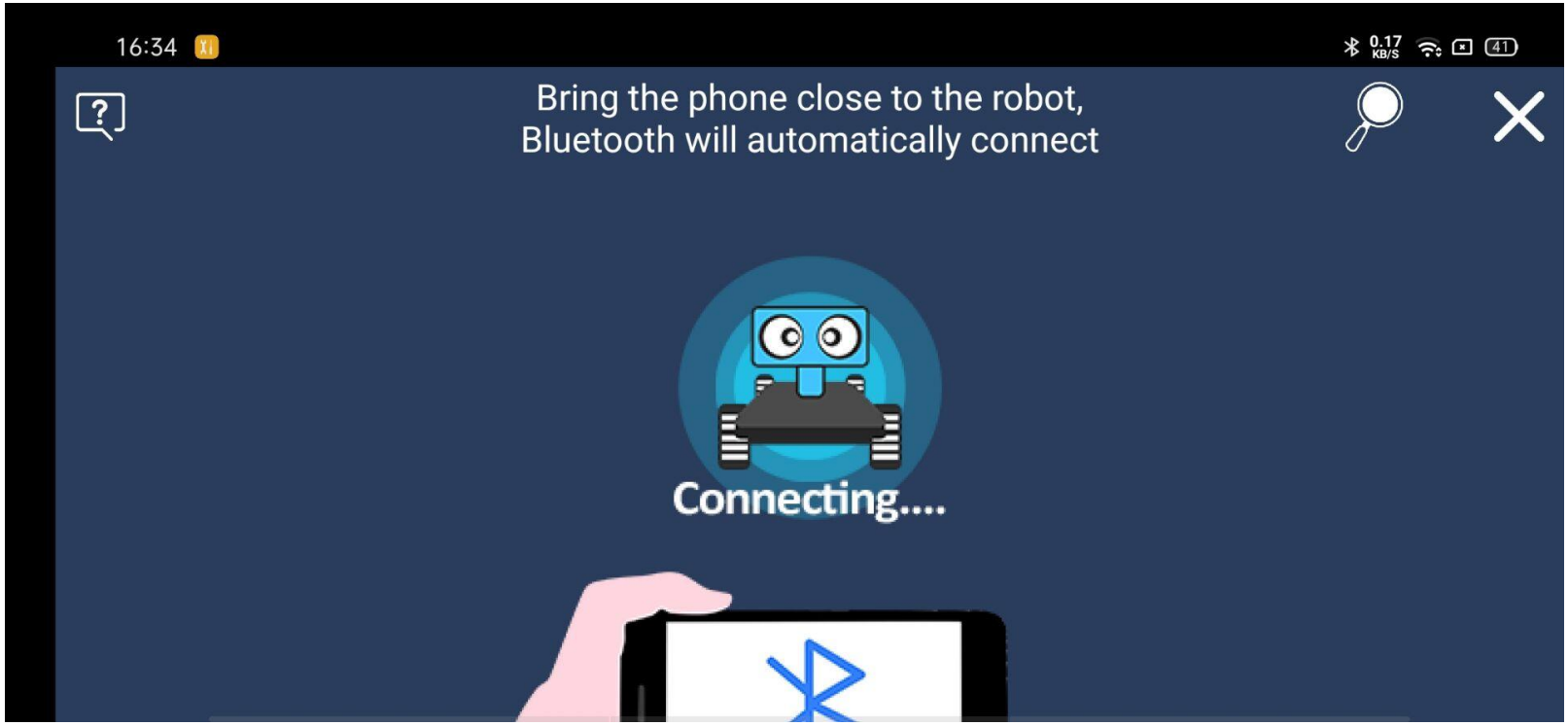
Install the JDY-16 Bluetooth module to the arduino uno Motor Driver Shied, turn on the power switch, and the indicator light of the Bluetooth module starts to flash once in 1.5s, which means it is in a connectable state.Click the Bluetooth icon  to enter the Bluetooth automatic pairing page.




16:35
0.00 KB/S
41
Bring the phone close to the robot,
Bluetooth will automatically connect

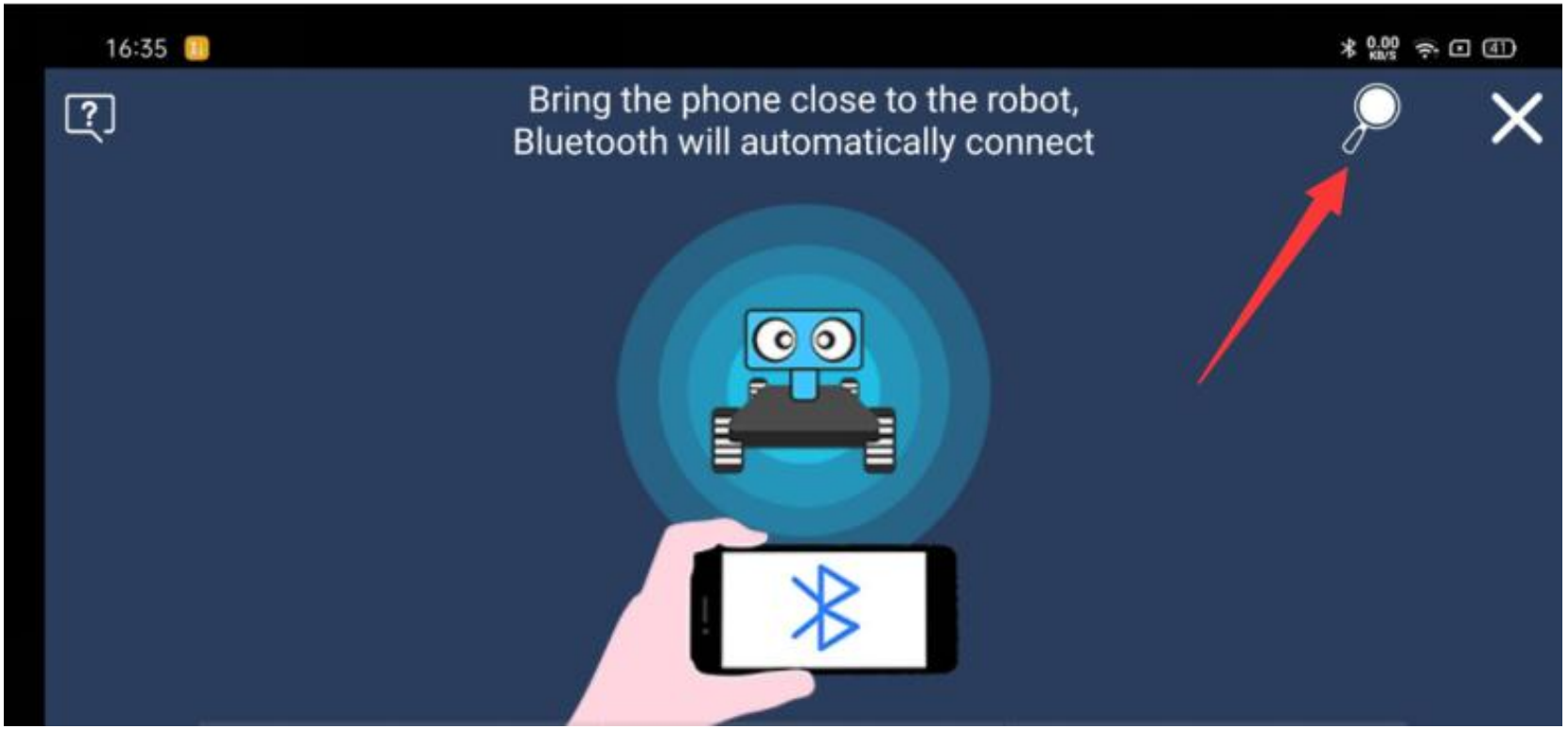
Bring the phone close to the robot,wait for 5 seconds, Bluetooth will automatically connect.

Note:if there are multiple devices at the same time, the Bluetooth of the mobile phone will automatically connect to the robot car that is closest to it. Therefore, in order to ensure a faster and more accurate connection, it is recommended to move the mobile phone close to the robot car you want to connect to.

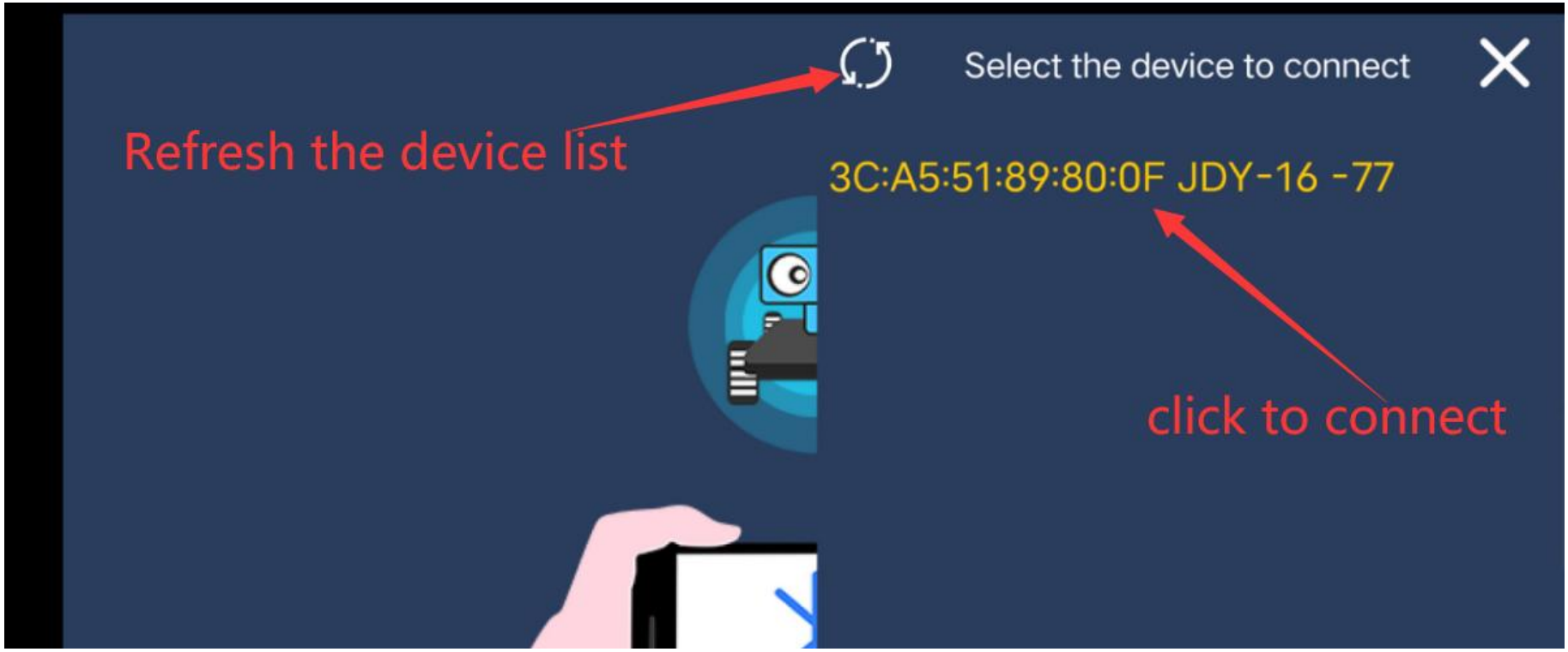


When Bluetooth is successfully connected, it will automatically return to the initial function control interface.

If the automatic connection fails to connect, click the search icon in the upper right corner to open the Bluetooth device list and try to connect manually.



You can try to refresh the list of Bluetooth devices and click the device name to manually connect.



lf you cannot connect automatically and there is no Bluetooth in the Bluetooth list, pull out the bluetooth module from arduino shield, then plug it back in. The bluetooth module LED indicator flashes once in 1.5s, waiting for the connection status,and try again.

Let's program

Test 1--Bluetooth Module Receives Information

In Experimental Test 1, we learned how to receive the information sent by the mobile phone app to the Bluetooth module JDY-16.

Arduino Code

if you want to refer to the program we provide.Open this source program in Arduino_Code>Lesson_9>**Test_1_Bluetooth_Module_Receives_Information.ino.**

Programming Thinking

```
String BLE_value;

void setup(){

  Serial.begin(9600);
```

```
    BLE_value = "";
}
```

Set the baud rate for serial communication to 9600. The communication speed between the serial port of the Arduino UNO main control board and the serial port of Bluetooth needs to be the same. The communication baud rate of the JDY-16 Bluetooth module is 9600 by default.

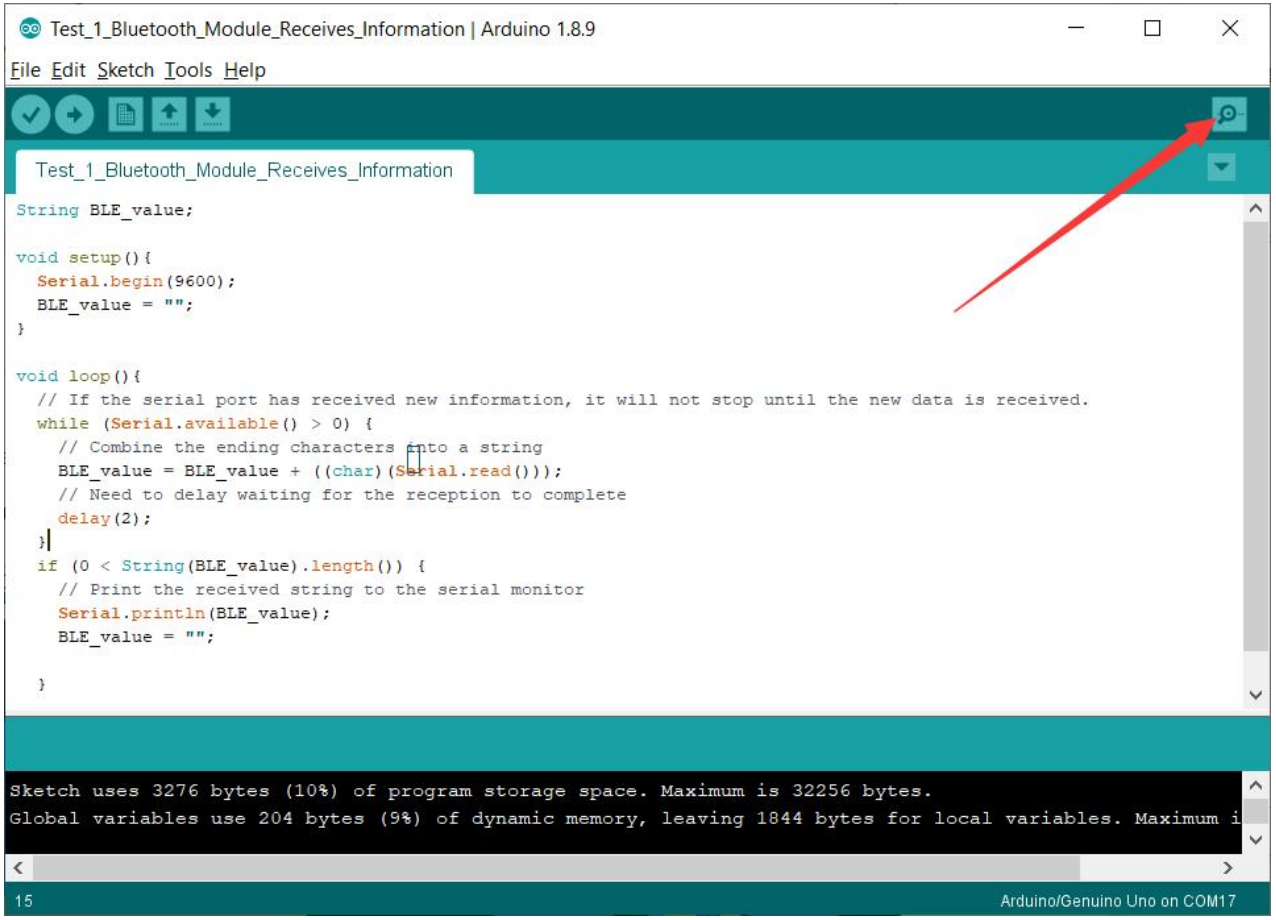
```
void loop(){
    // If the serial port has received new information, it will not stop until the new data is received.
    while (Serial.available() > 0) {
        // Combine the ending characters into a string
        BLE_value = BLE_value + ((char)(Serial.read()));
        // Need to delay waiting for the reception to complete
delay(2);
    }
```

If the serial port has received new information, it will not stop until the new data is received.Combine the ending characters into a string.Need to delay waiting for the reception to complete.

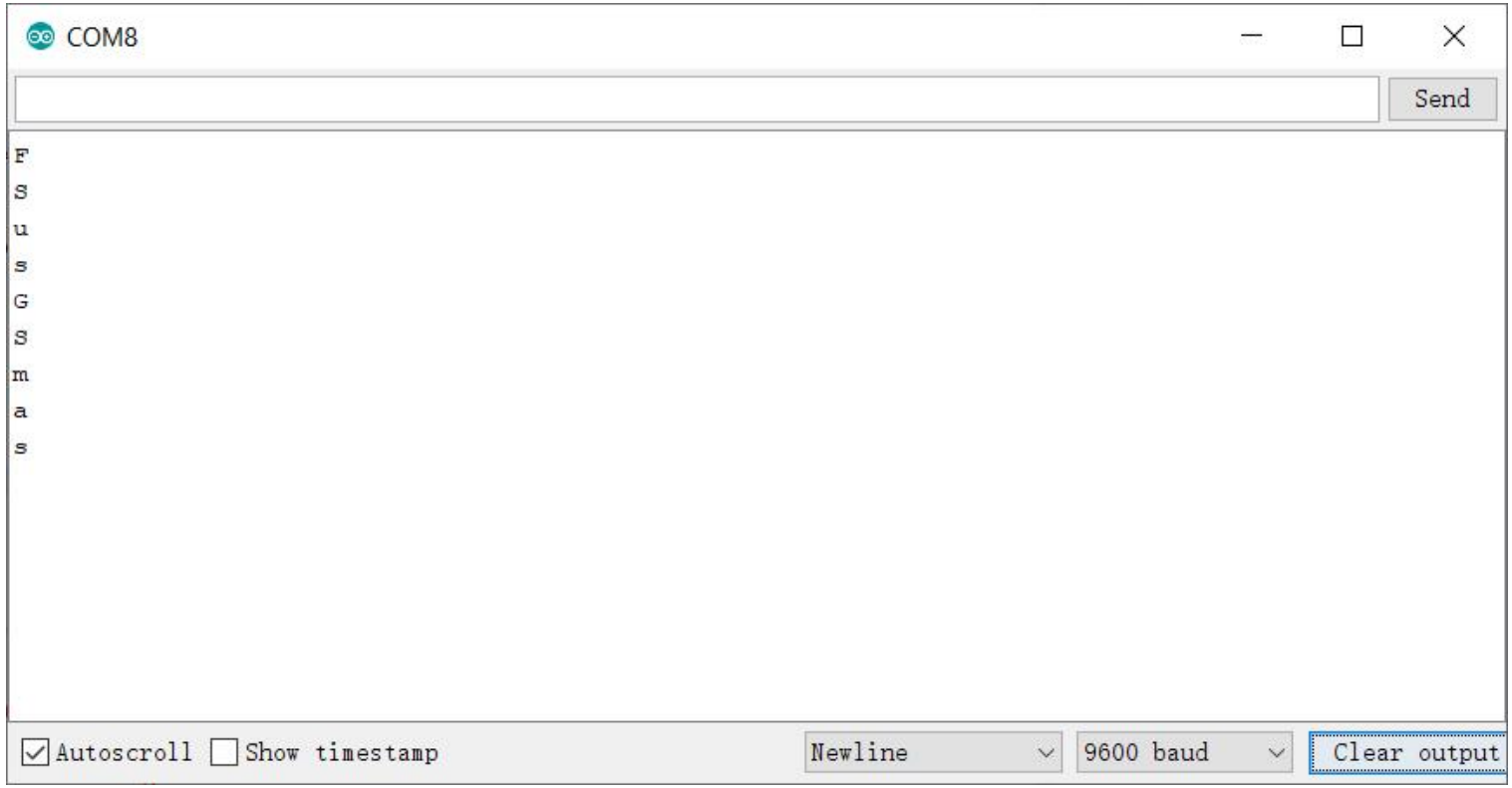
```
 if (0 < String(BLE_val).length() && 2 >= String(BLE_val).length()) {//If a new character string is received
      // Print the received string to the serial monitor
      Serial.println(BLE_value);
      BLE_value = "";//clear the last data without preparing for the next reception.
   }
}
```

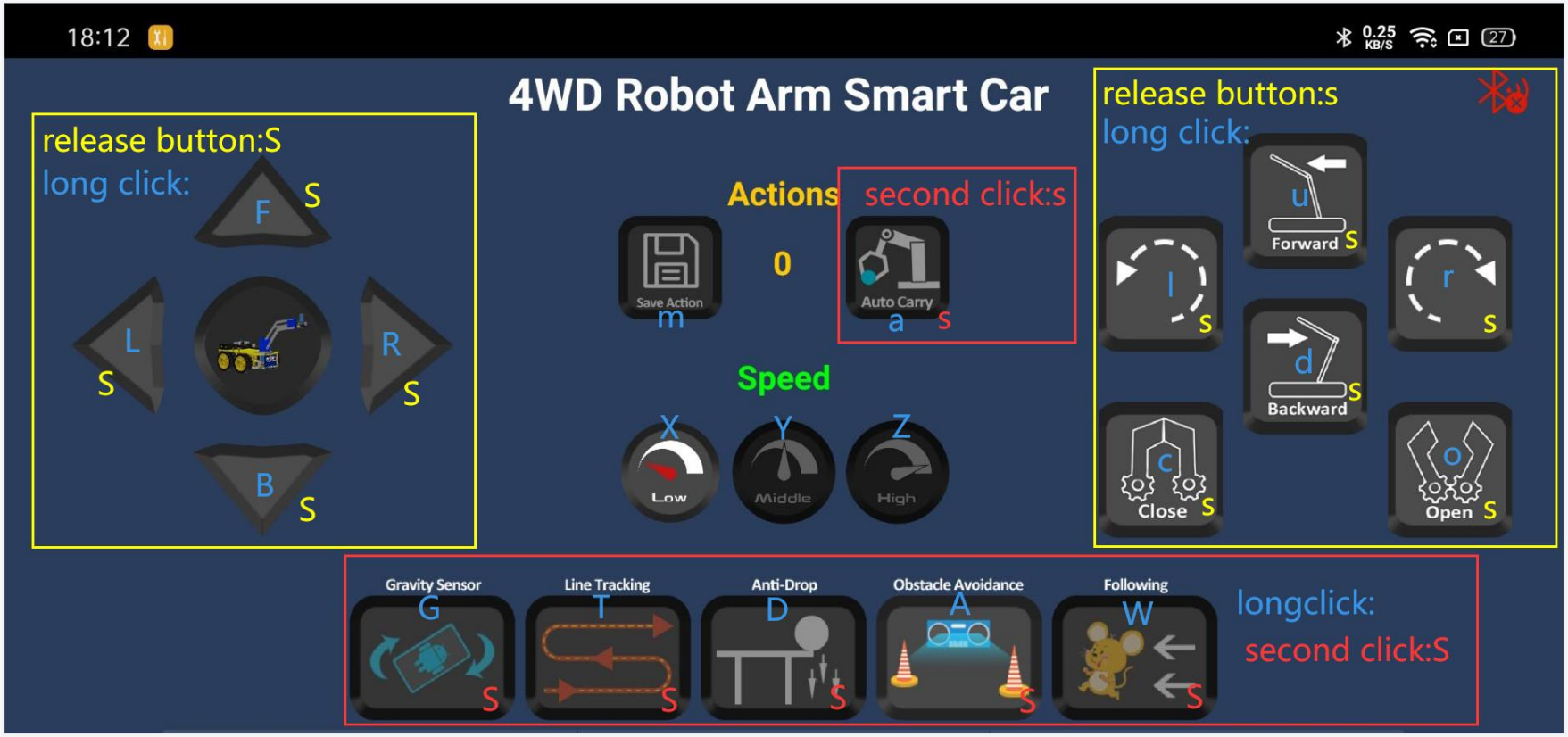
If a new character string is received, print the received character string to the serial monitor, and clear the last data without preparing for the next reception.

After completing the program upload, open the serial monitor to view the information received by the Bluetooth module.

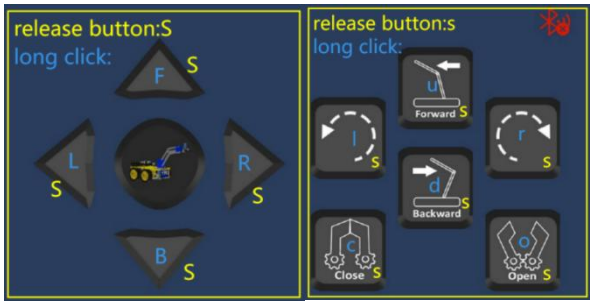


Then you can see the data as blow:




18:12
0.25 KB/S
27
4WD Robot Arm Smart Car
release button:S
long click:
F S
L S
R S
B S
Actions
second click:s
Save Action m
0
Auto Carry a s
Speed
X Low
Y Middle
Z High
release button:s
long click:
u Forward s
l s
r s
d Backward s
c Close s
o Open s
Gravity Sensor G S
Line Tracking T S
Anti-Drop D S
Obstacle Avoidance A S
Following W S
longclick:
second click:S

The button in the yellow box needs to be pressed and the command is the character marked in blue, and the command when the button is released is the character marked in yellow.



For example, when the button 

of the mobile phone APP is pressed, the command ‘F’ will be sent to the Bluetooth module of the robot car, and when the button 

is released, the command ‘S’ will be sent to the Bluetooth module of the robot car.

When the button 

of the mobile phone APP is pressed, the command ‘u’ will be sent to the Bluetooth module of the robot car, and when the button 

is released, the command ‘s’ will be sent to the Bluetooth module of the robot car.

The buttons in the red box have the highest authority. When the buttons in the red box are pressed, other buttons will be locked and cannot be used. You need to click the button again to unlock other buttons. The command sent when the button in the red box is clicked for the first time is the characters marked in blue, and the command sent when the button in the red box is clicked for the second time is still the characters marked in red.

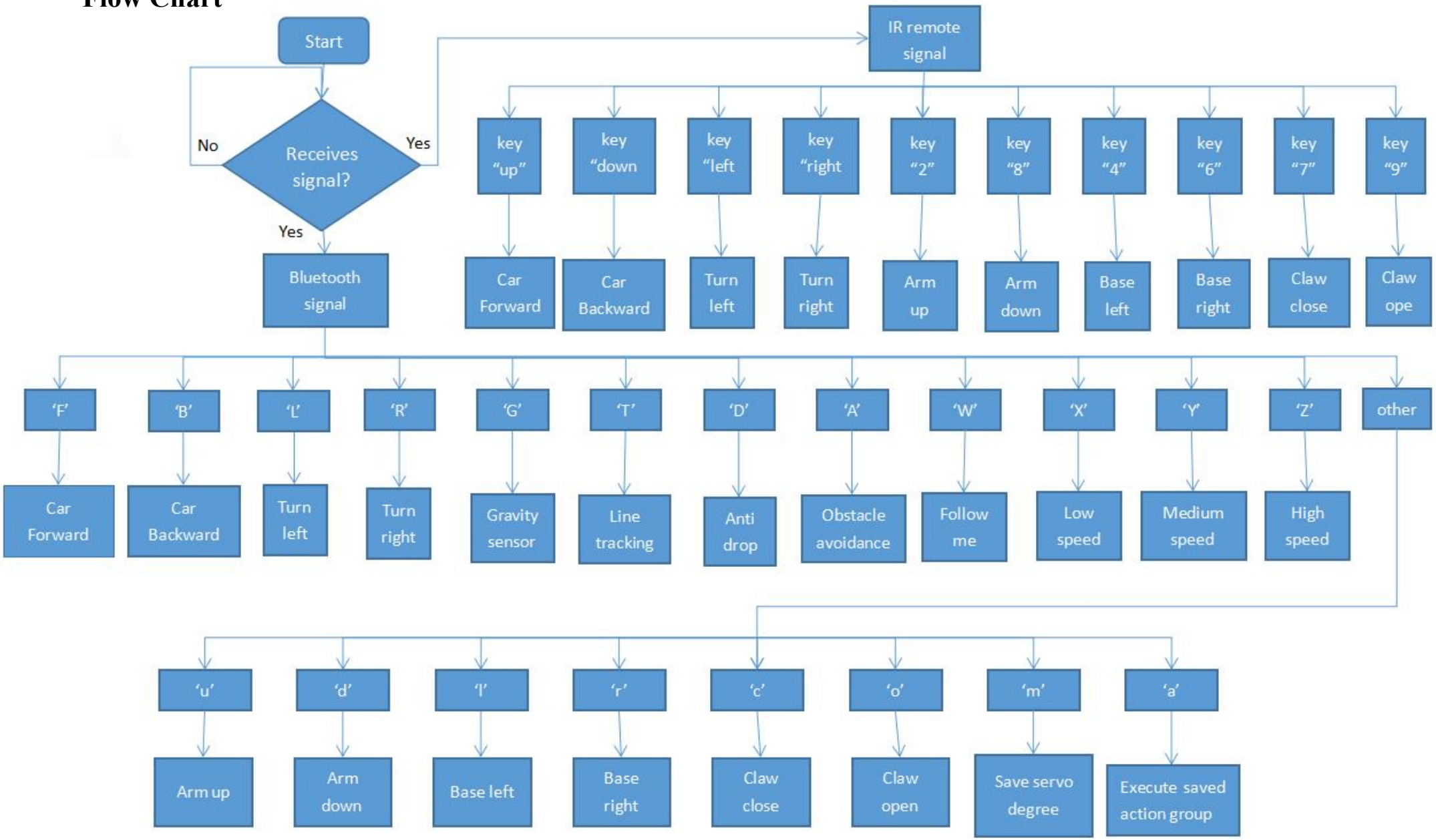
For example, when the button  of the mobile phone APP is clicked, the command ‘G’ will be sent to the Bluetooth module of the robot car. When the button  is clicked for the second time, the command ‘S’ will be sent to the Bluetooth module of the robot car.

For example, when the button  of the mobile phone APP is clicked, the command ‘a’ will be sent to the Bluetooth module of the robot car. When the button  is clicked for the second time, the command ‘s’ will be sent to the Bluetooth module of the robot car.

7-in-1 Multi-Function Robot Arm Smart Car

In this experimental test, All functions are combined into the same program, you can use the app to freely switch between different functions. At the same time, the infrared remote control can also control the direction of the robot car and the degree of the robot arm servo.

Flow Chart

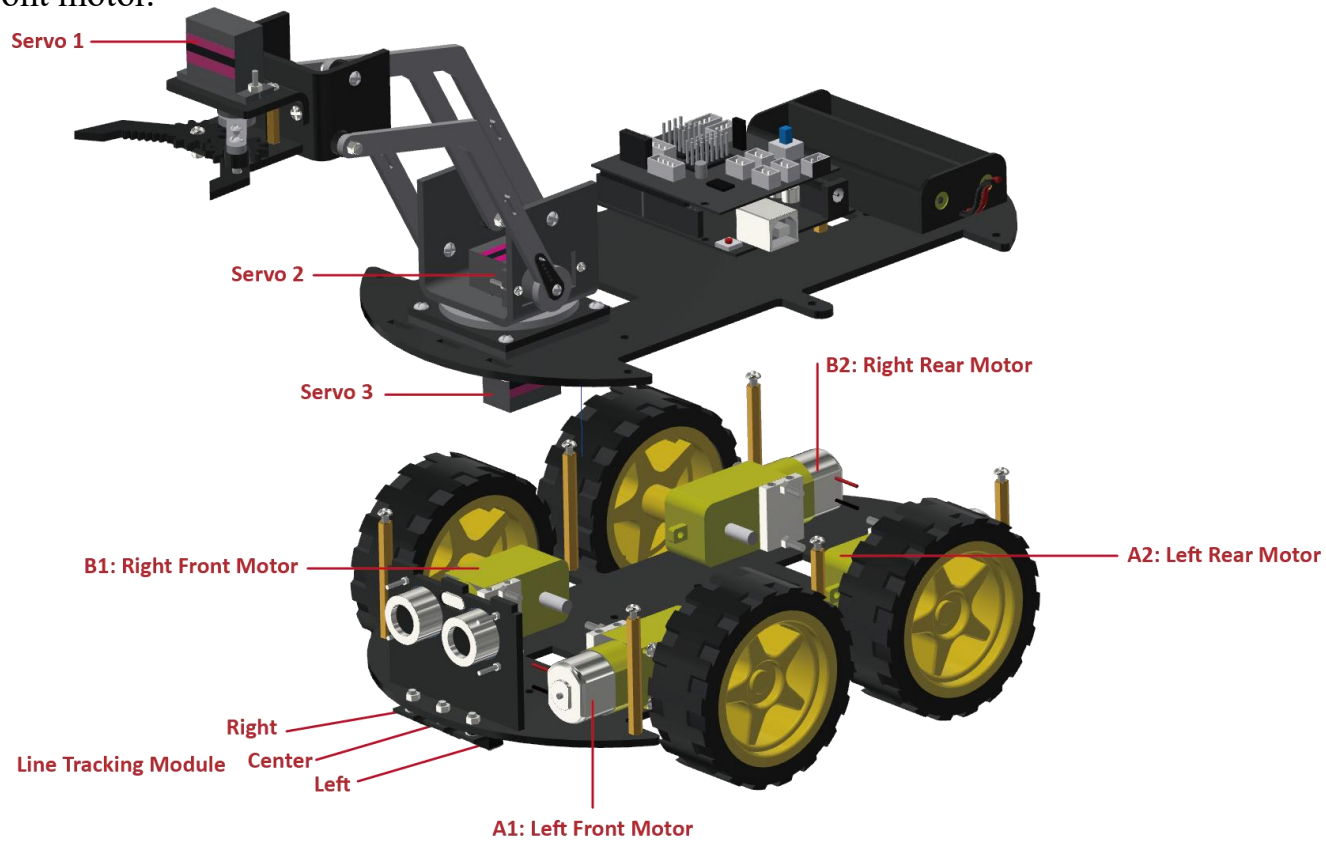

Start
Receives signal?
No
Yes
Yes
Bluetooth signal
IR remote signal
key "up"
key "down
key "left
key "right
key "2"
key "8"
key "4"
key "6"
key "7"
key "9"
Car Forward
Car Backward
Turn left
Turn right
Arm up
Arm down
Base left
Base right
Claw close
Claw ope
'F'
'B'
'L'
'R'
'G'
'T'
'D'
'A'
'W'
'X'
'Y'
'Z'
other
Car Forward
Car Backward
Turn left
Turn right
Gravity sensor
Line tracking
Anti drop
Obstacle avoidance
Follow me
Low speed
Medium speed
High speed
'u'
'd'
'l'
'r'
'c'
'o'
'm'
'a'
Arm up
Arm down
Base left
Base right
Claw close
Claw open
Save servo degree
Execute saved action group

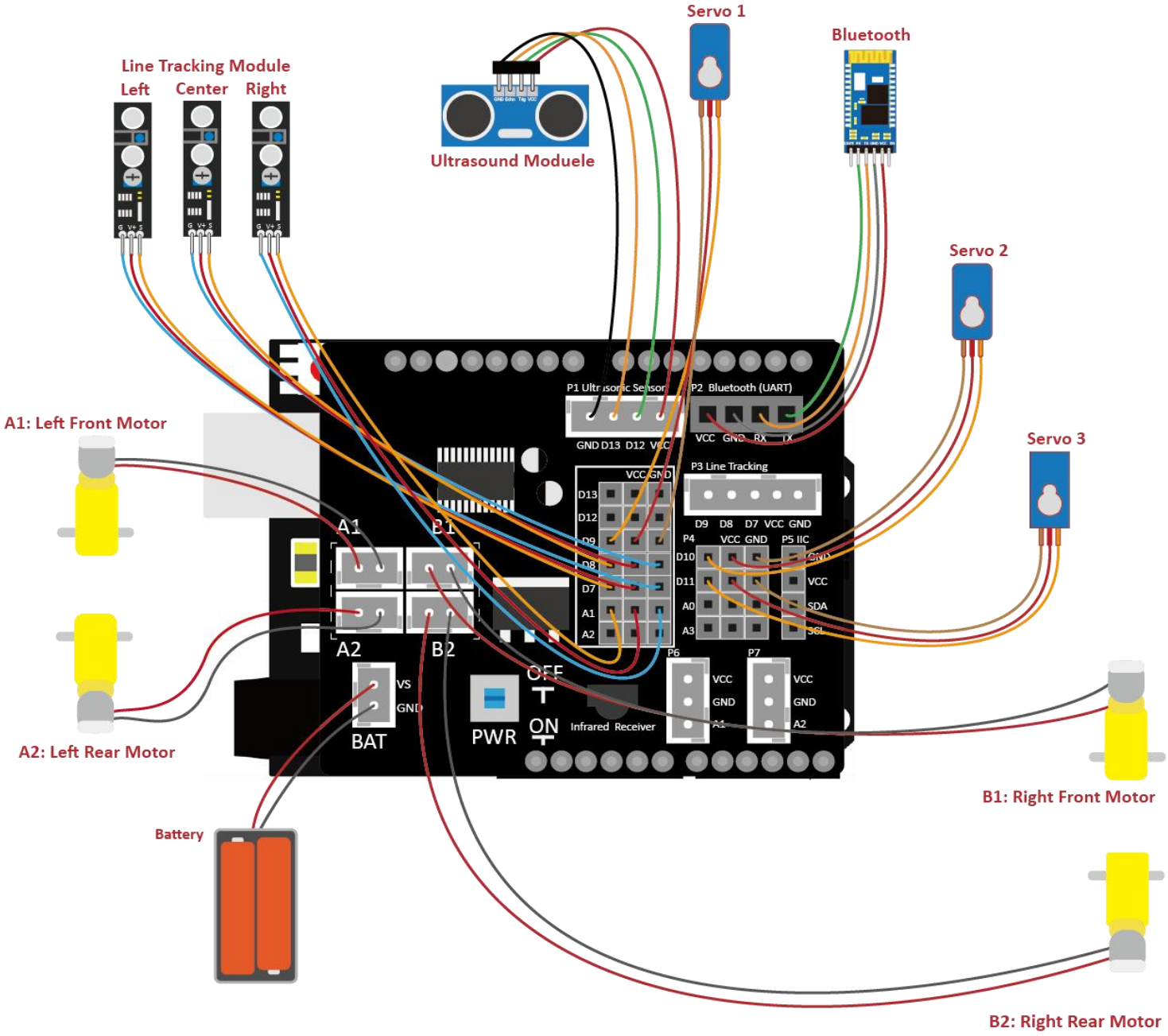
How to use the 7-in-1 function

Step 1:Circuit wiring

Connect the wiring according to the wiring diagram, paying attention to the positive and negative poles of the power supply.

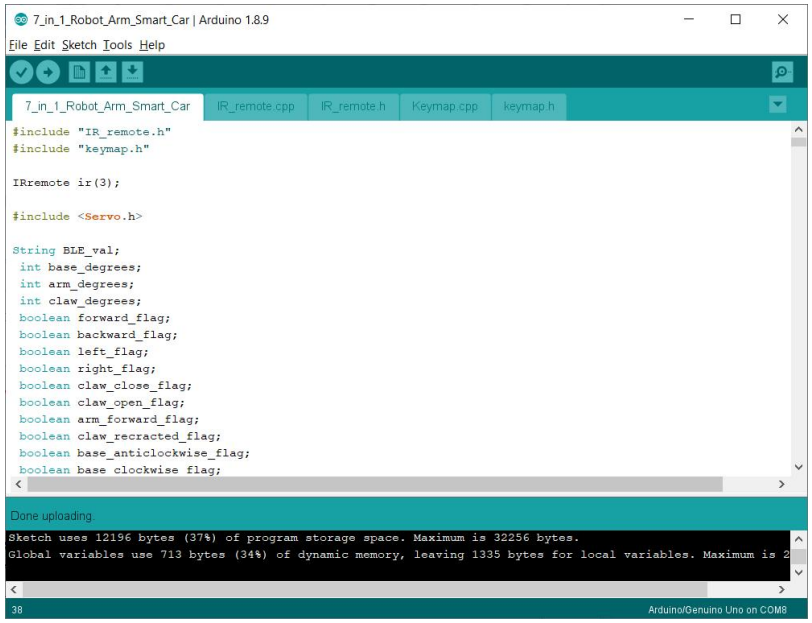
Correctly distinguish the installation position of the sensor module, such as left, center, right line tracking module. A1: Left front motor B1: Right front motor.




Servo 1
Bluetooth
Line Tracking Module
Left
Center
Right
Ultrasound Moduele
Servo 2
Servo 3
A1: Left Front Motor
P1 Ultrasonic Sensor
P2 Bluetooth (UART)
GND D13 D12 VCC
VCC GND RX TX
P3 Line Tracking
A1
B1
A2
B2
D9 D8 D7 VCC GND
P4 VCC GND
P5 IIC
D10
D11
A0
A3
SDA
SCL
P6
P7
VCC
GND
A1
A2
OFF
ON
PWR
BAT
VS
GND
Infrared Receiver
A2: Left Rear Motor
B1: Right Front Motor
Battery
B2: Right Rear Motor

Step 2:Upload code

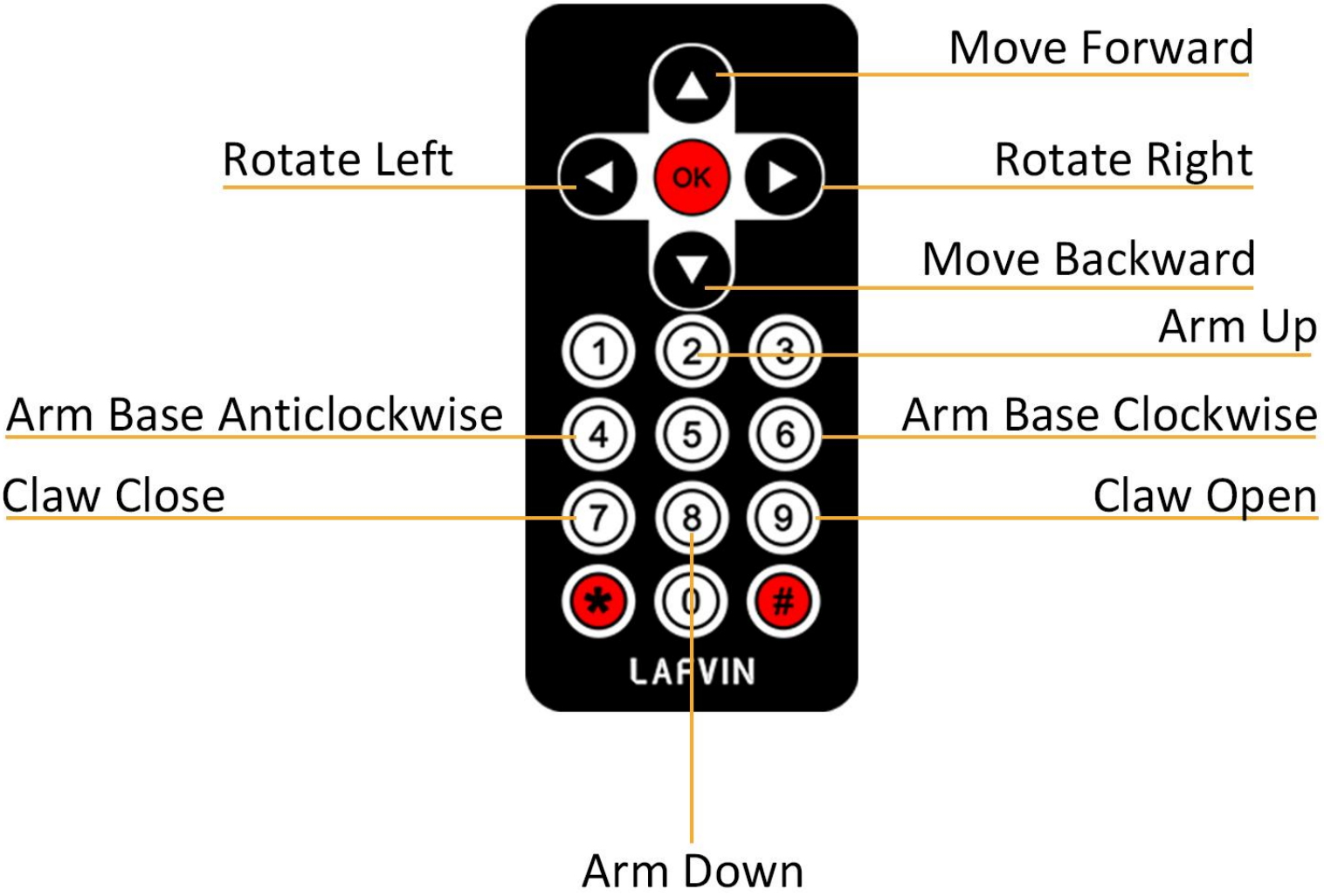
Open this source program in Arduino_Code>Lesson_9>**7_in_1_Multi_Function_Robot_Arm_Smart_Car.ino.**Upload the program to arduino uno. If you report an error during uploading the program, you can refer to Lesson 1 Getting Started with Arduino IDE.



Before you can run this, make sure that you have installed the <Servo> library or re-install it, if necessary. Otherwise, your code won't work.For details about loading the library file, see Lesson about how to add libraries .

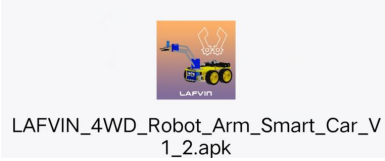
Step 3:Infrared remote control

After successfully uploading the code, you can directly use the infrared remote control to control the robot smart car

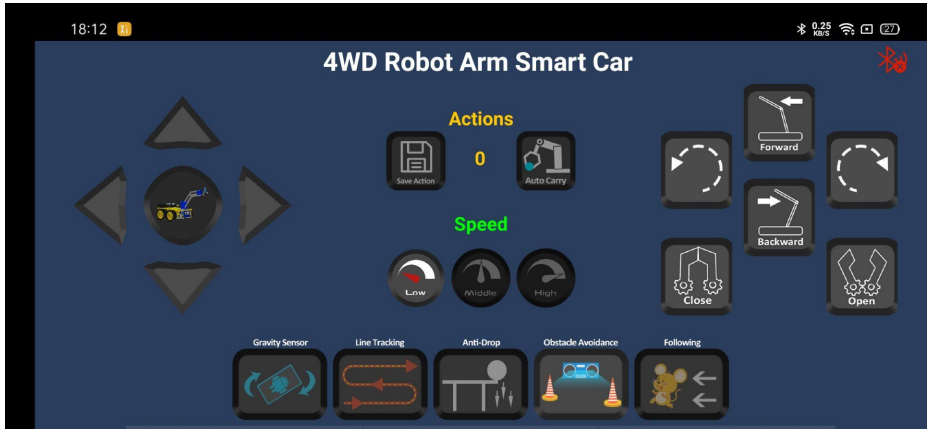


Step 4:Install the app

Firstly, copy the “LAFVIN_4WD_Robot_Arm_Smart_Car_V1_2.apk” file from the APP folder on CD to your mobile phone and install it into an application software.The built-in Bluetooth module of the mobile phone needs to support Bluetooth 4.0 or higher.

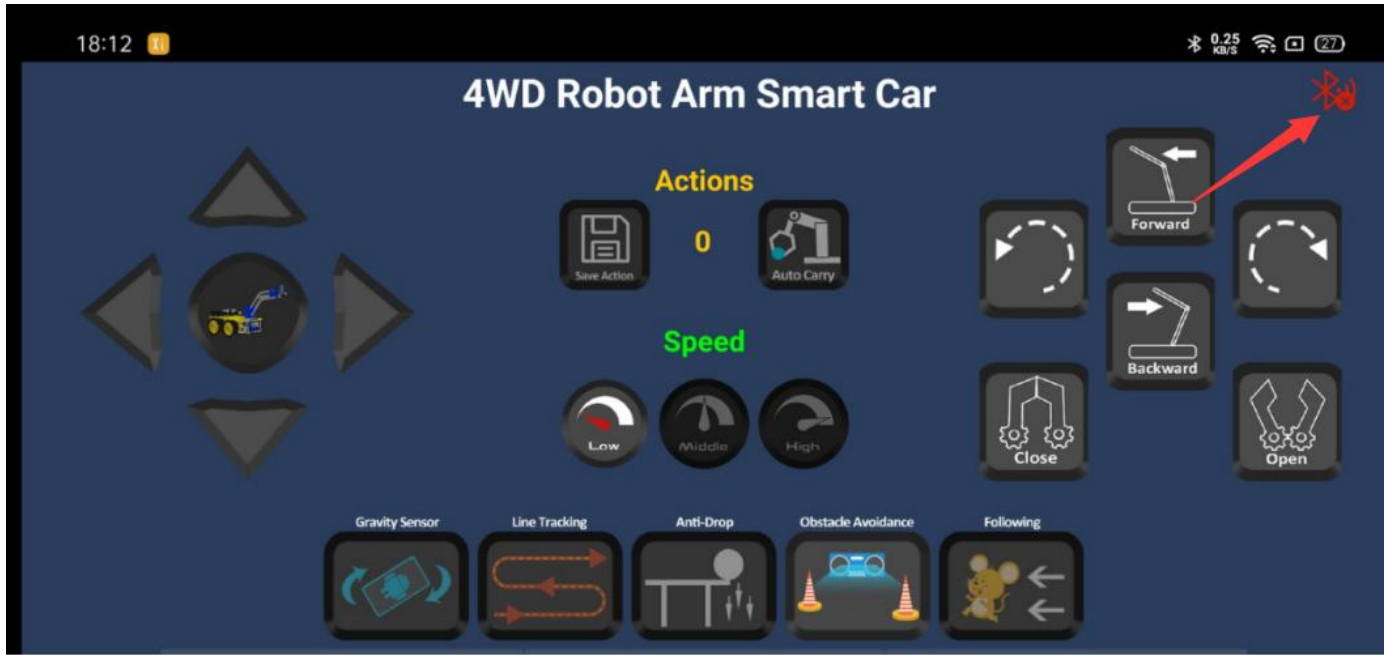


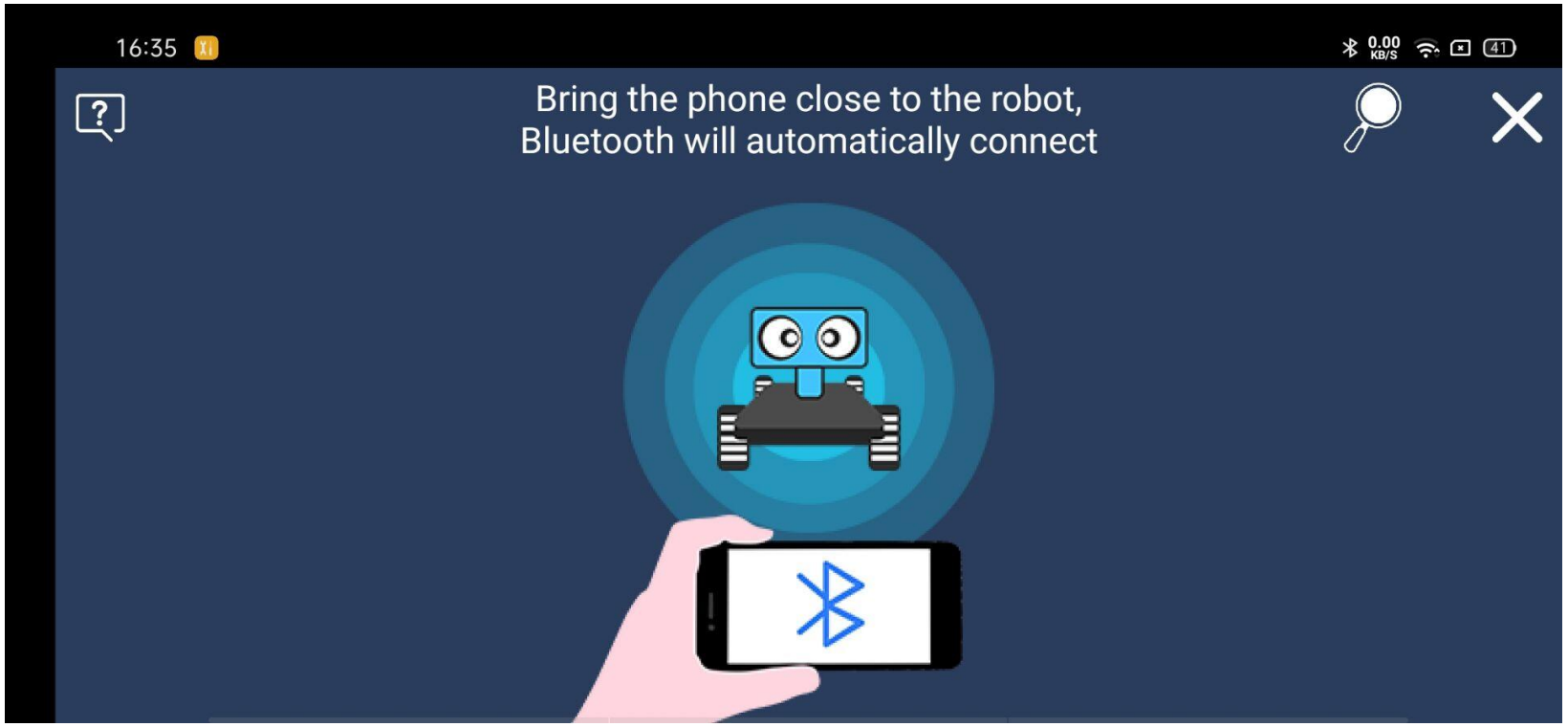
Open the APP, you will see the following control interface.



Step 5:Bluetooth connection

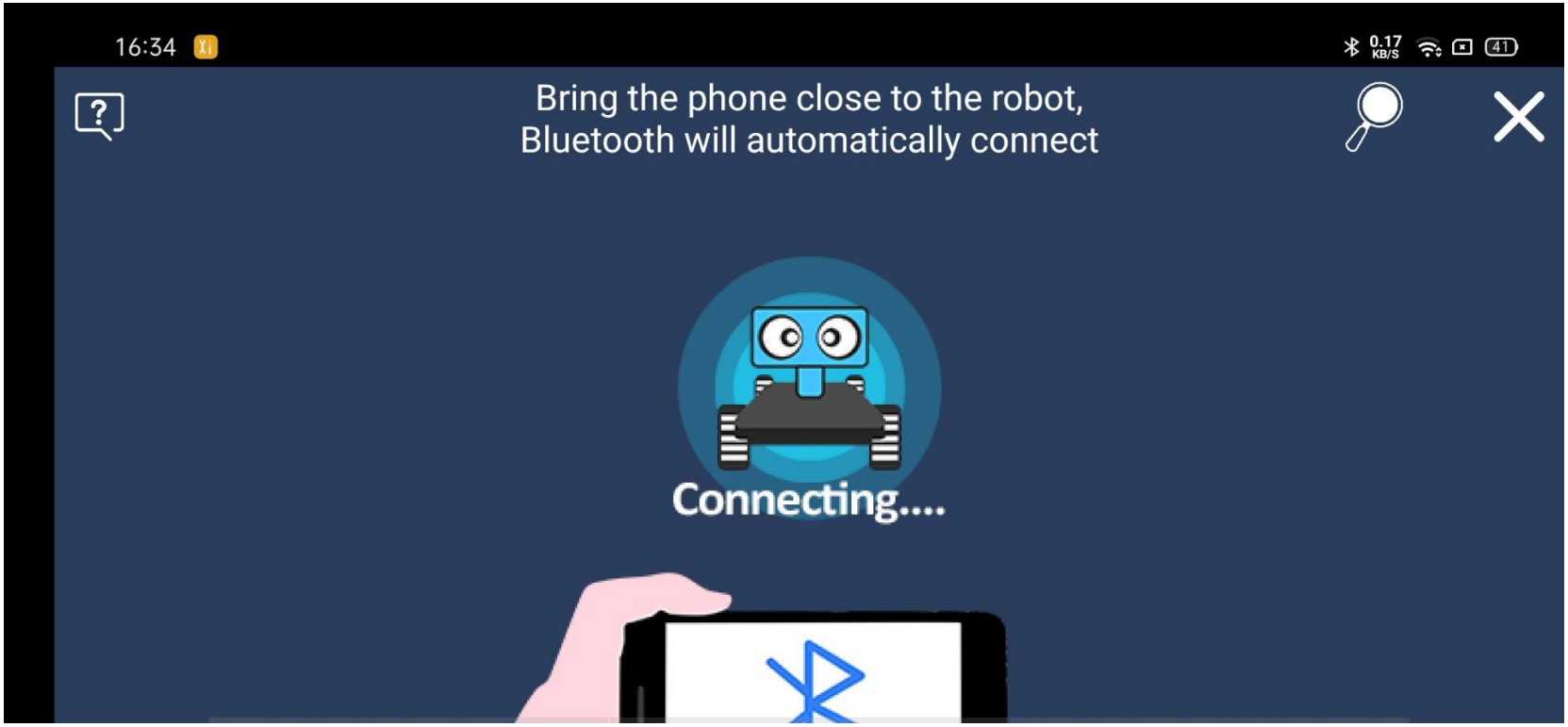
Install the JDY-16 Bluetooth module to the Arduino uno Motor Driver Shied, turn on the power switch, and the indicator light of the Bluetooth module starts to flash once in 1.5s, which means it is in a connectable state.Click the Bluetooth icon  to enter the Bluetooth automatic pairing page.





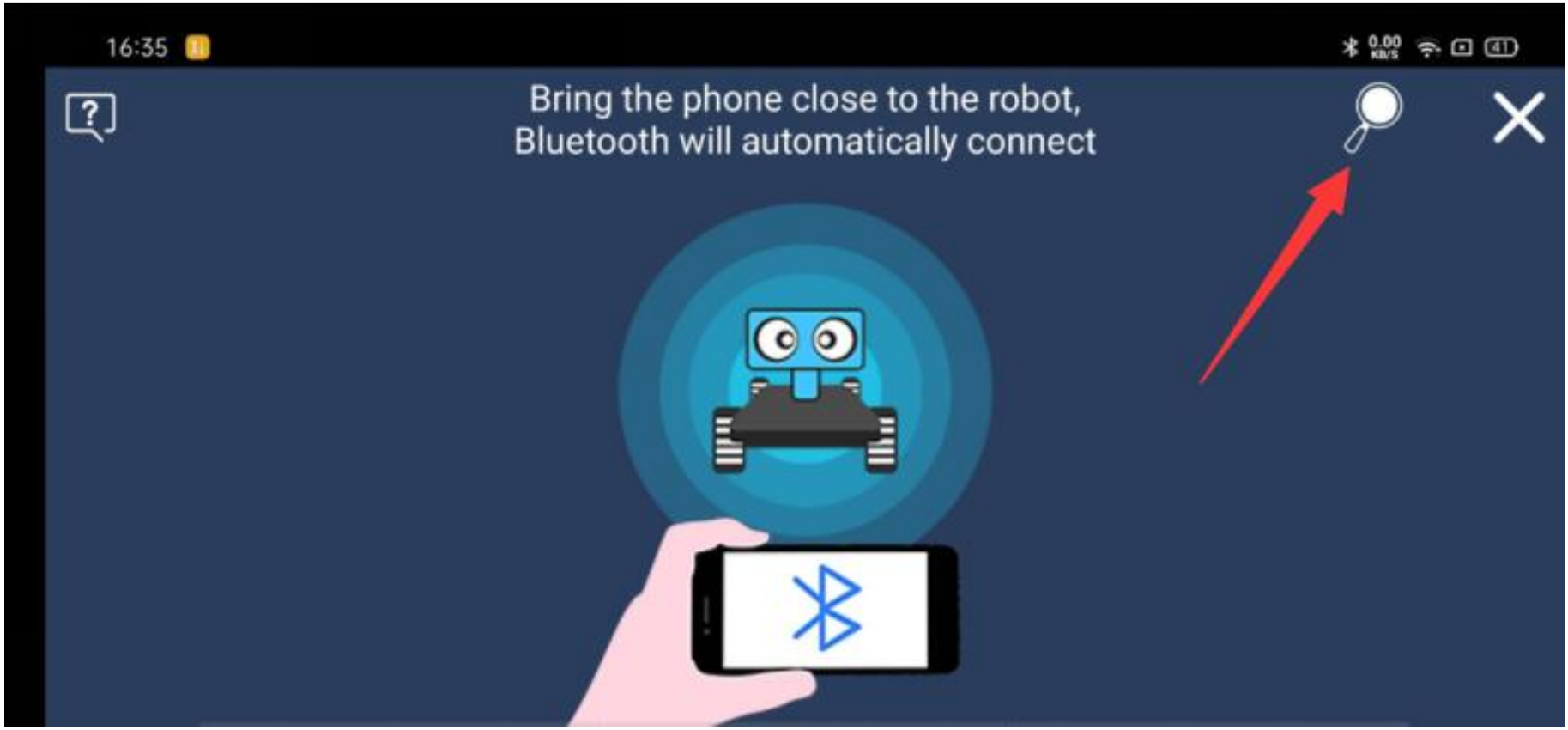
Note:if there are multiple devices at the same time, the Bluetooth of the mobile phone will automatically connect to the robot car that is closest to it. Therefore, in order to ensure a faster and more accurate connection, it is recommended to move the mobile phone close to the robot car you want to connect to.

Bring the phone close to the robot,wait for 5 seconds, Bluetooth will automatically connect.

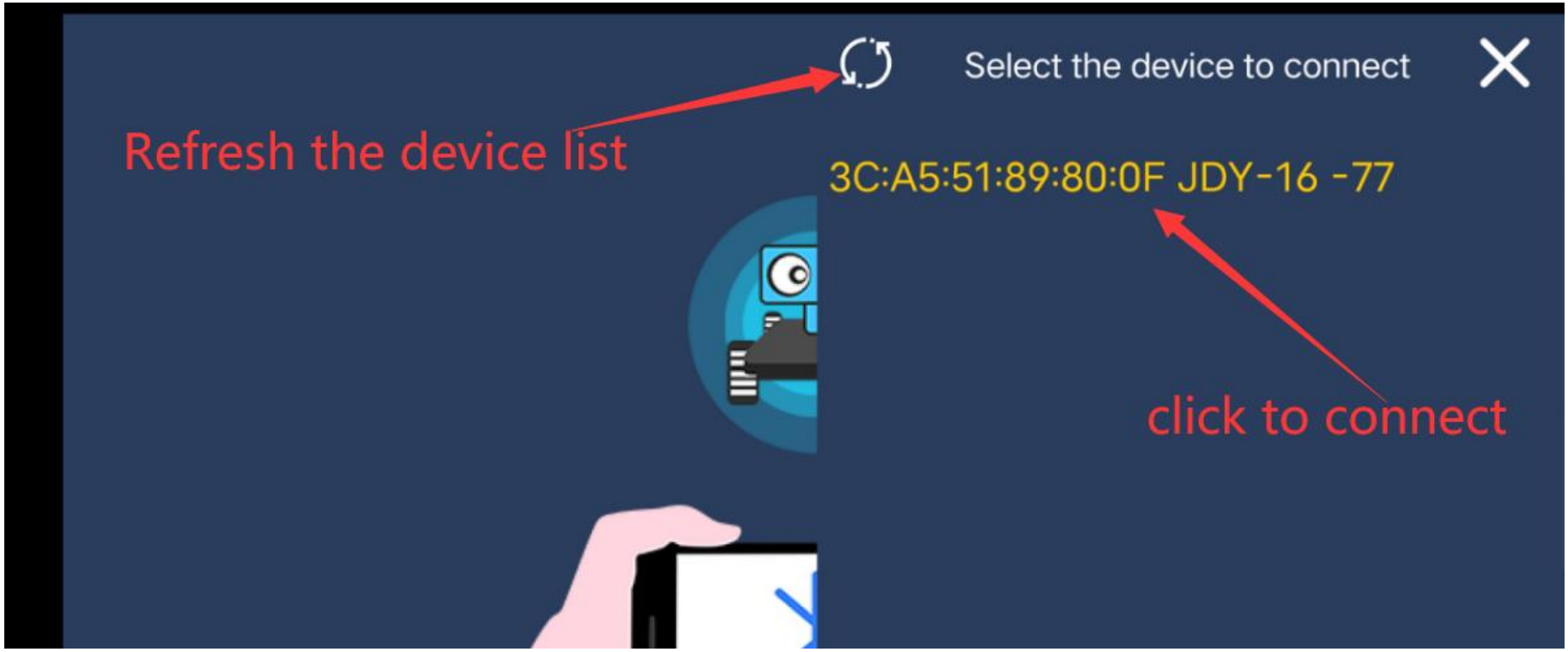


When Bluetooth is successfully connected, it will automatically return to the initial function control interface.

If the automatic connection fails to connect, click the search icon in the upper right corner to open the Bluetooth device list and try to connect manually.



You can try to refresh the list of Bluetooth devices and click the device name to manually connect.

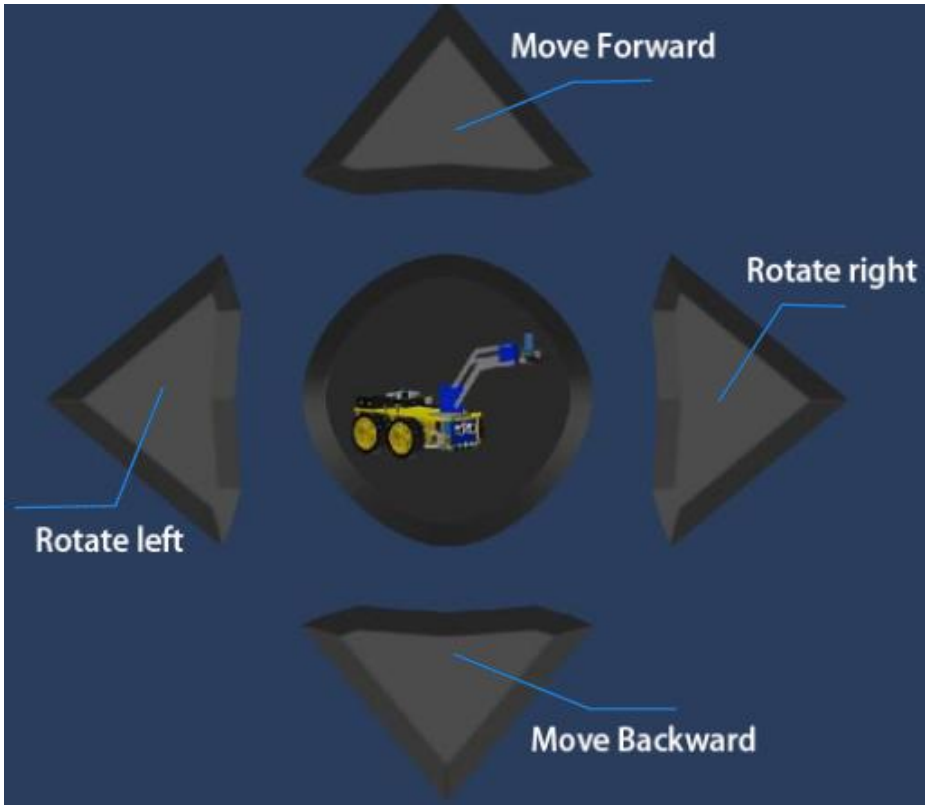


lf you cannot connect automatically and there is no Bluetooth in the Bluetooth list, pull out the bluetooth module from arduino shield, then plug it back in. The bluetooth module LED indicator flashes once in 1.5s, waiting for the connection status,and try again.

Step 6:App control interface

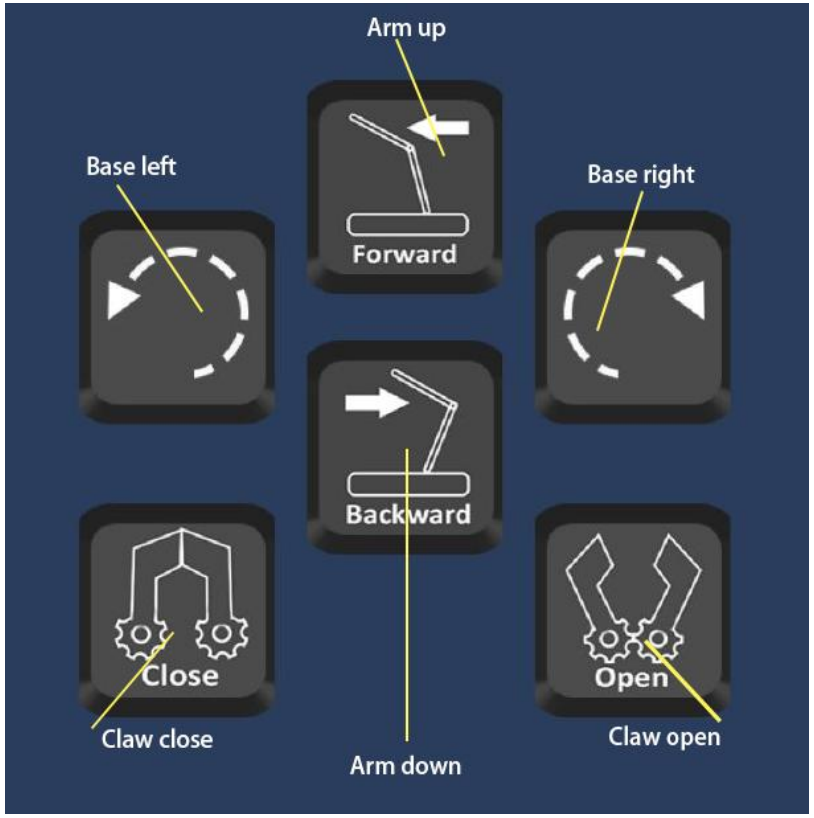
Robot car direction button:

Press to trigger, release to stop



Robot arm control buttons:

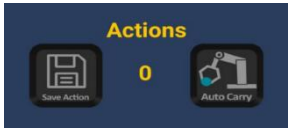
Press to trigger, release to stop



Multi-function mode button:

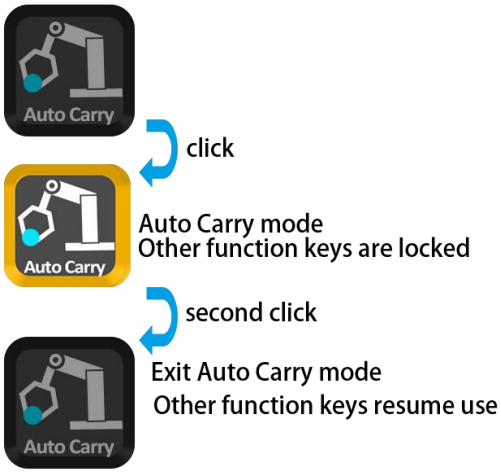
Click the mode button to activate the function. Other buttons are locked and cannot be used. You need to click again to exit the function mode. The other buttons are unlocked and resume use.

Action memory button:



 :Click to save the current action of the robot arm

:Shows the number of saved actions, the maximum number of saved actions is 20.



 :Click to execute all the saved actions. When the button is clicked, it will enter the auto carry mode. Other buttons of other apps are locked and cannot be used. You need to click again to exit the automatic execution mode, and the other buttons will be unlocked to resume use.

Robot car speed control button:

Click the button to switch speed