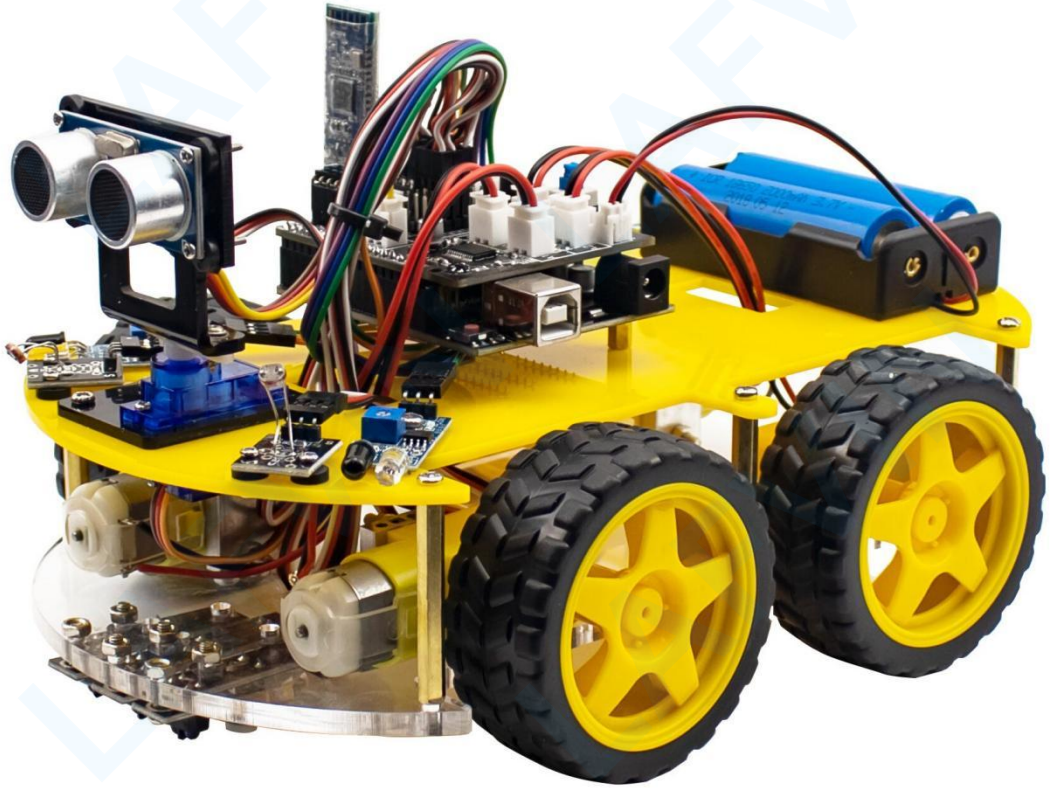
4WD Smart Robot Car Kit V2.2



Content

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. Simultaneously supports Mixly graphical programming and Arduino IDE code programming, which is convenient for beginners to learn.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 6 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

SG90 Servo x1

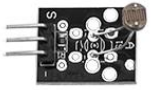
Bluetooth x1

IR Avoid Module x2

Line Tracking Module x3

Motor x4

Tire x4



Photoresistive sensor x2

Servo Holder x1

Ultarsound Holder x1

USB Cable x1

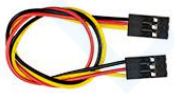
Battery Case x1

Bottom Acrylic Layer x1

Upper Acrylic Layer x1

Adhesive Tape x1

4pin XH2.54 to F Dupont x1
3pin XH2.54 to F Dupont x2



3pin F-F Dupont*5

Tie x3

Screw bag

Screwdriver x1

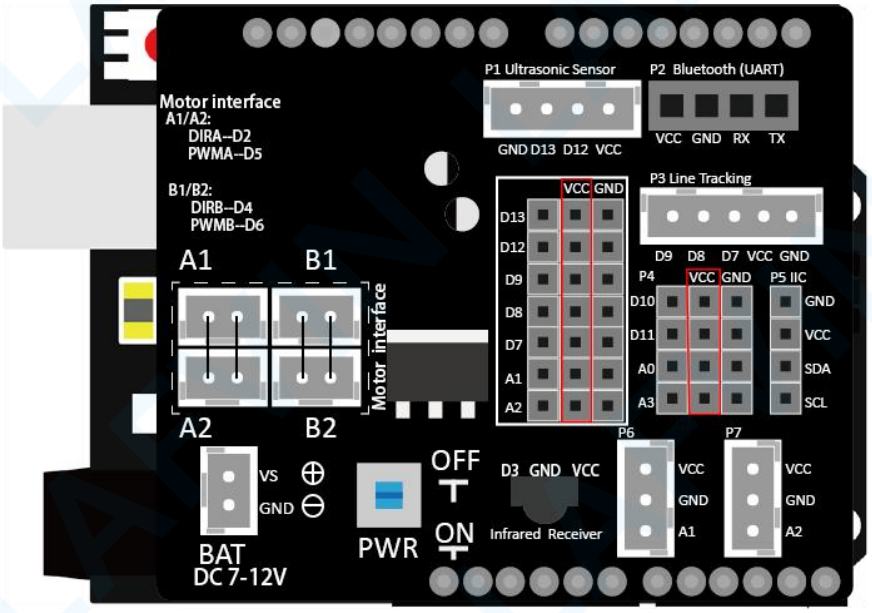
CD Tutorail x1

LAFVIN Smart Robot Car Kit V2 Introduction

Smart Robot Car Kit V2.0 is mainly manufactured using the Arduino UNO R3 main control and the Motor drive expansion board, two reduction motors, Acrylic frame chassis and some sensors.

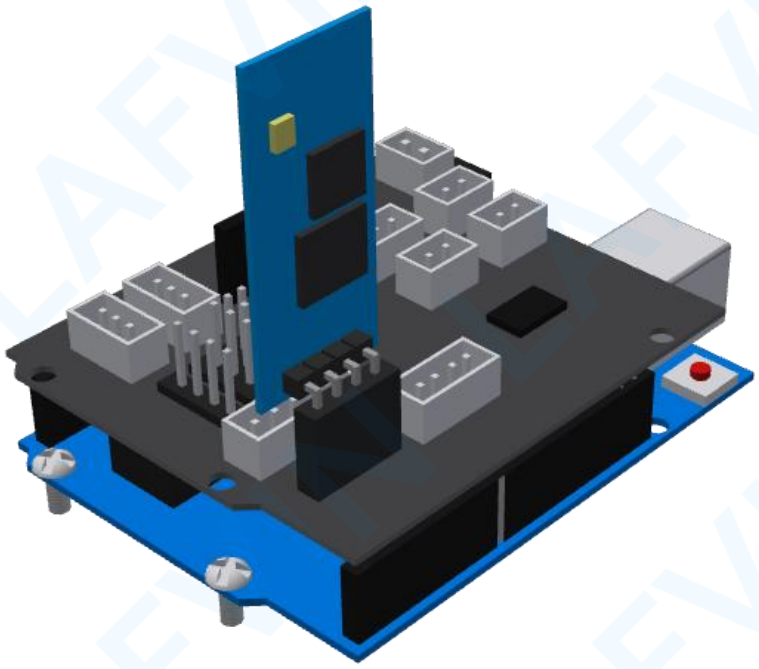
LAFVIN Smart Robot Car Kit V2 have the following advantages:

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.

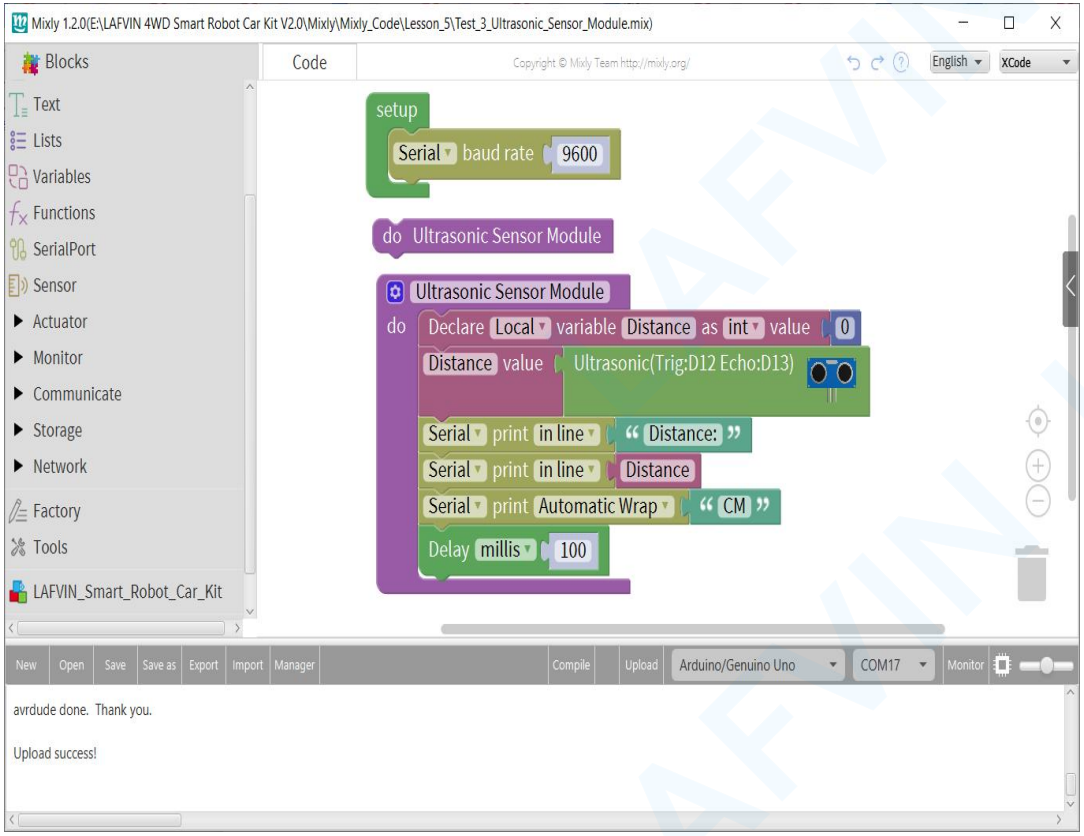


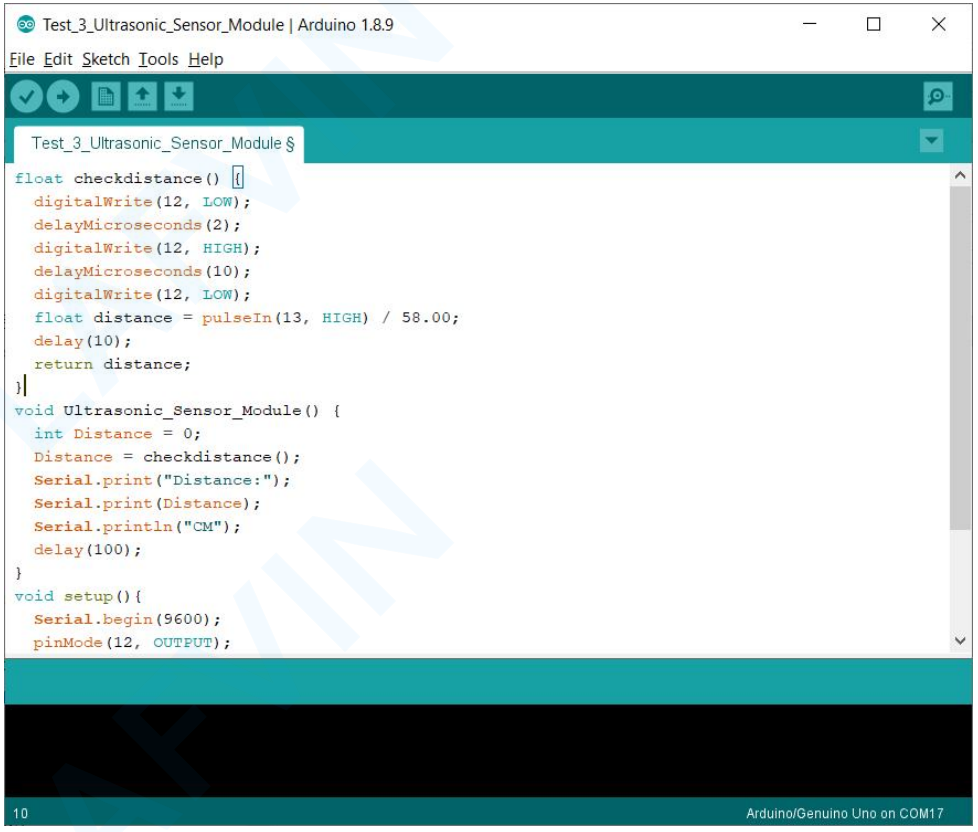
2) The newly created APP control software, the 6 major functions of the robot car kit are comprehensively controlled by the APP, and the function mode can be switched freely.**6 in 1 Multi-purpose Bluetooth Robot Car**

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.

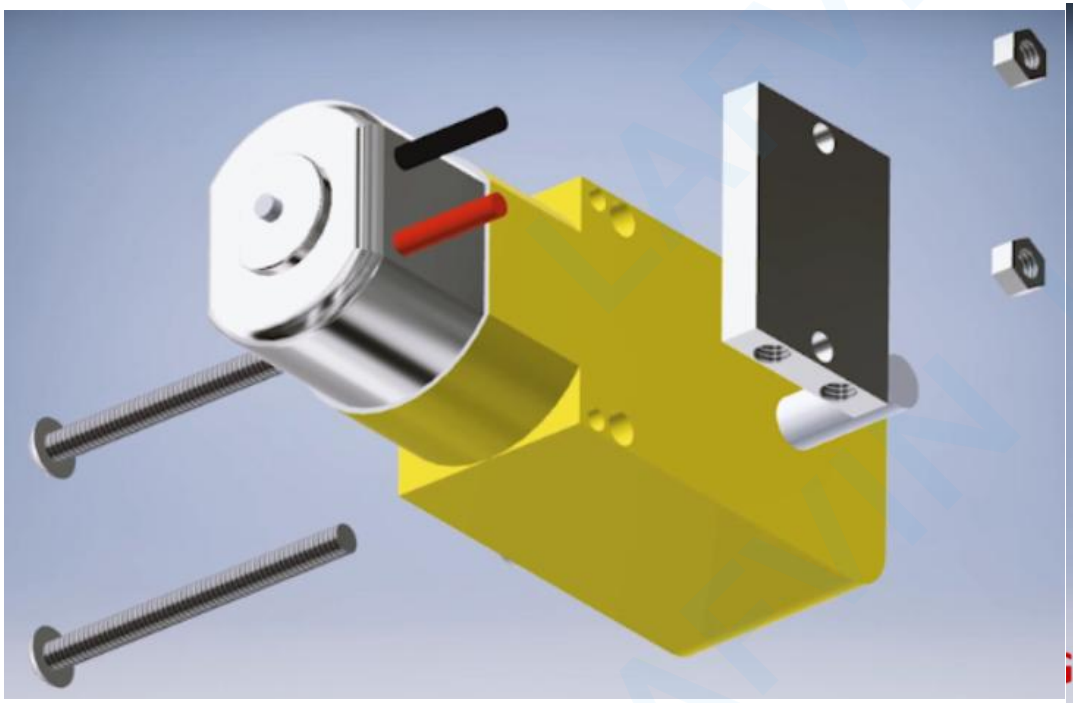


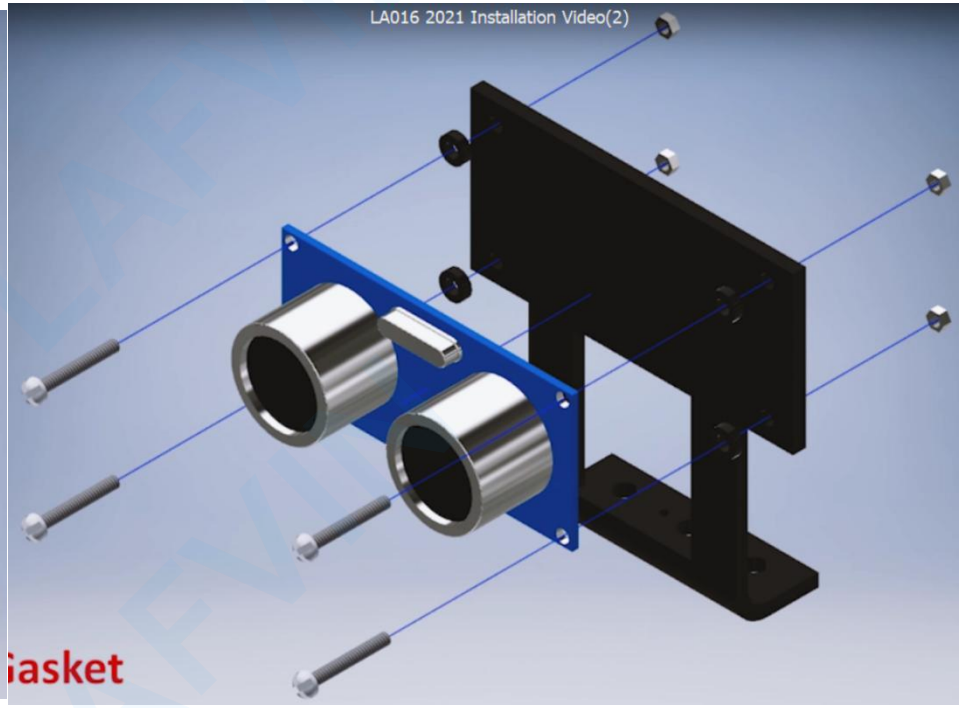
4)Simultaneously supports mixly graphical programming and arduino IDE code programming, which is convenient for beginners to learn.





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





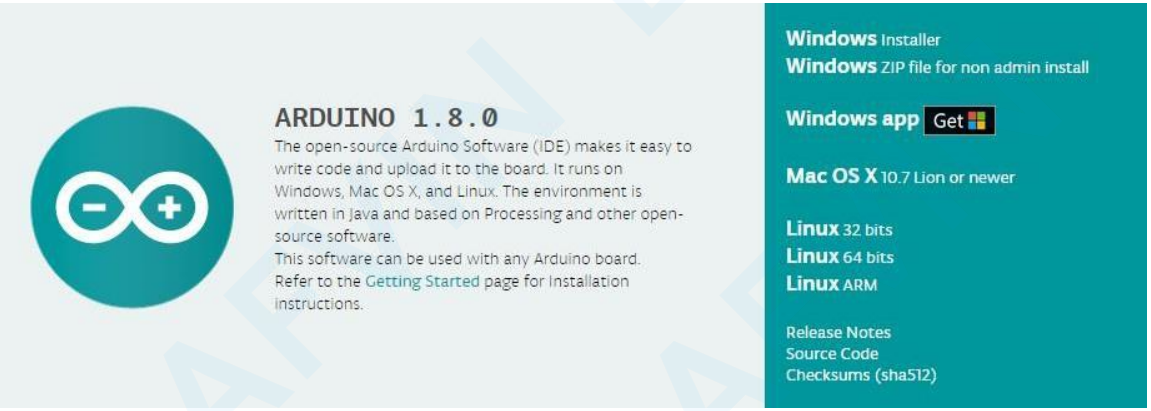
Lesson 1 Getting Started with Arduino IDE

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)

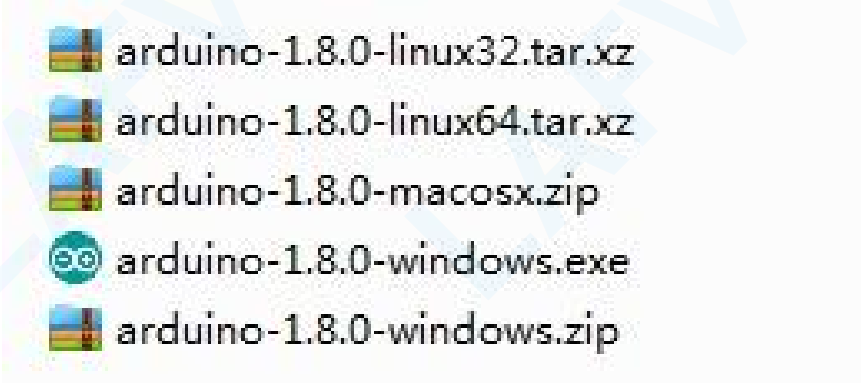


The version available at this website is usually the latest version, and the actual version may be newer than the version in the picture.

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

Click Windows Installer.

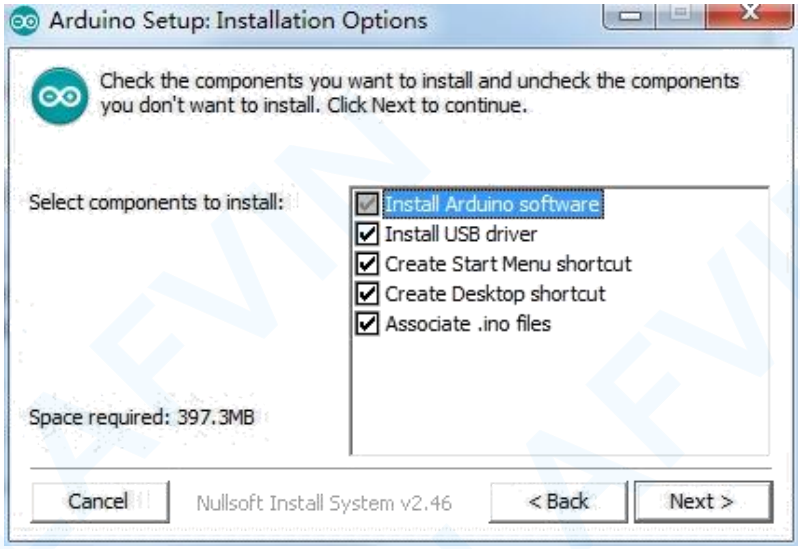
Click JUST DOWNLOAD.
Also version 1.8.0 is available in the material we provided, and the versions of our materials are the latest versions when this course was made.



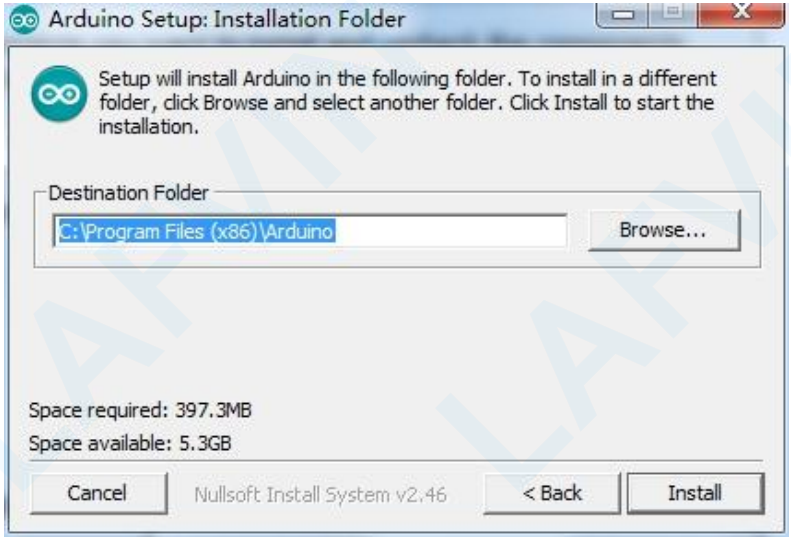
Installing Arduino (Windows)

Install Arduino with the exe. Installation package.

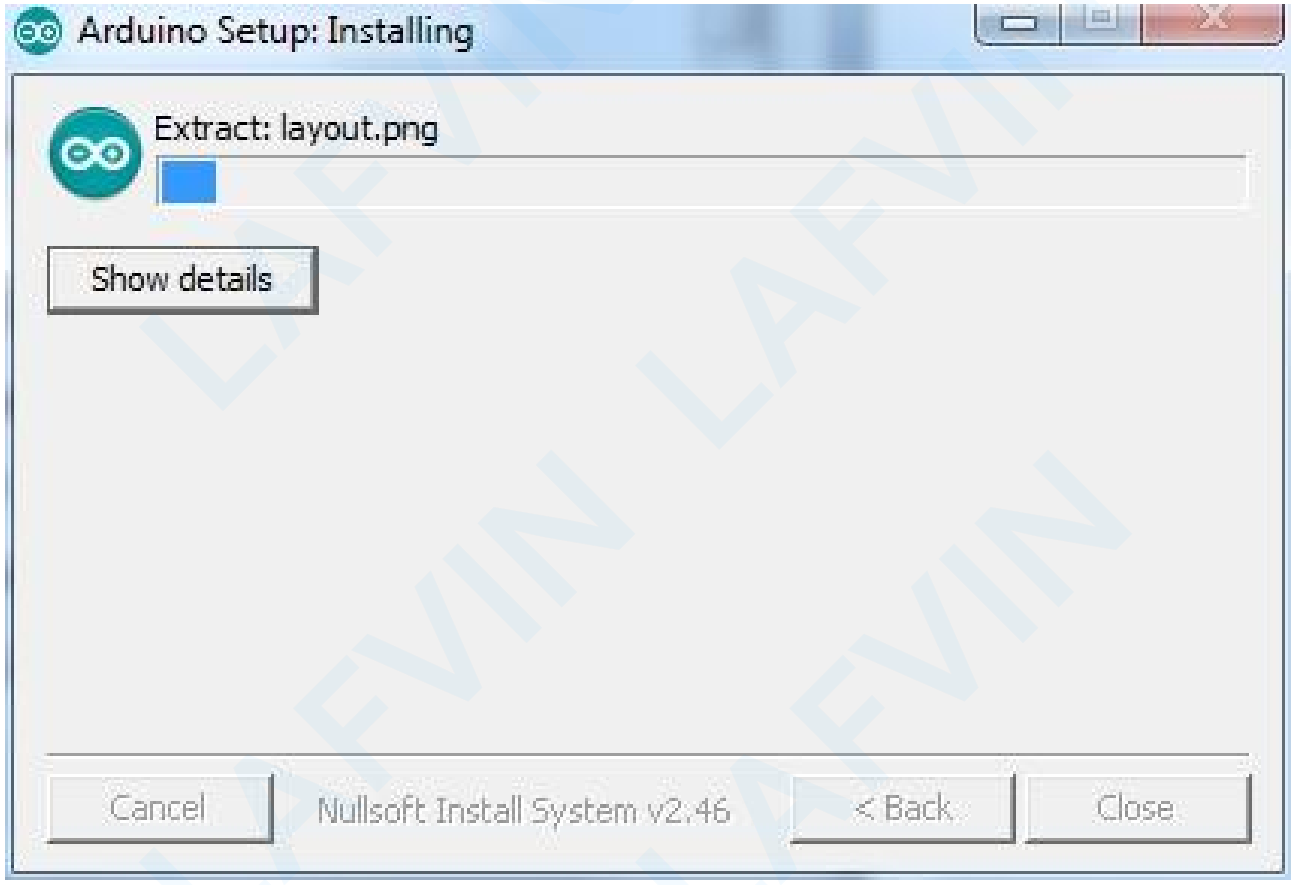
Click I Agree to see the following interface


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 >

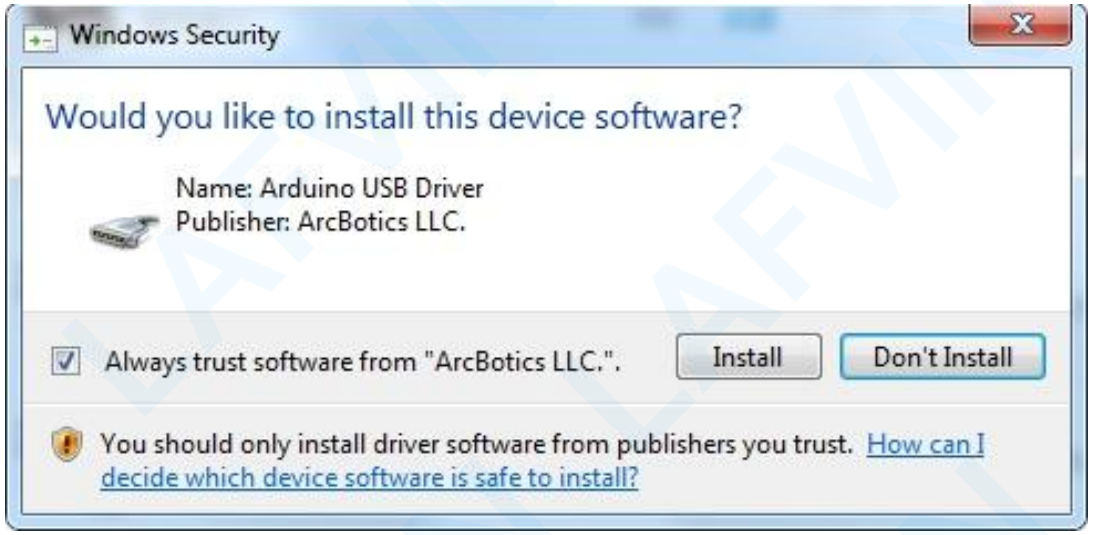
Click Next


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

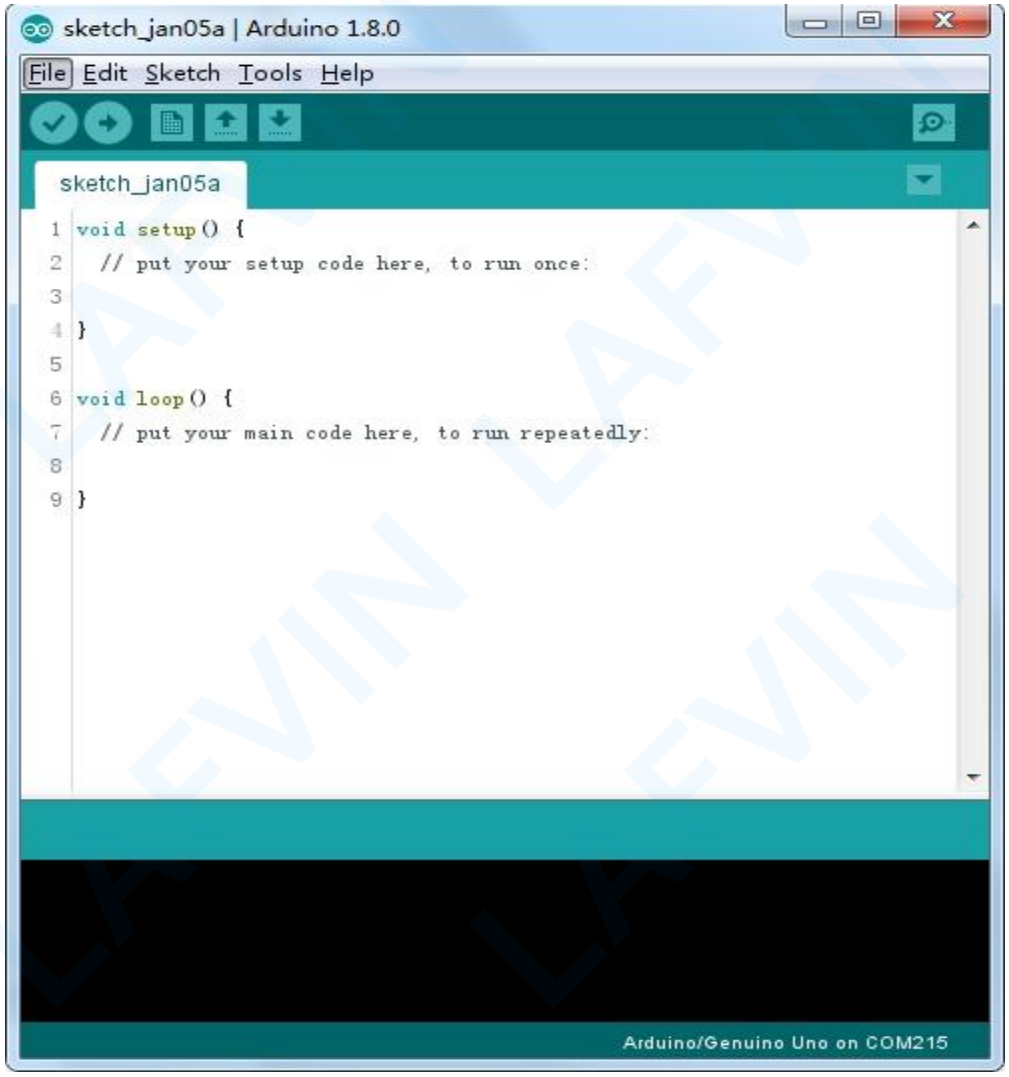


Wait for the installing process, if appear the interface of Window Security, just continue to click Install to finish the installation.



Next, the following icon appears on the desktop

Double-click to enter the desired development environment

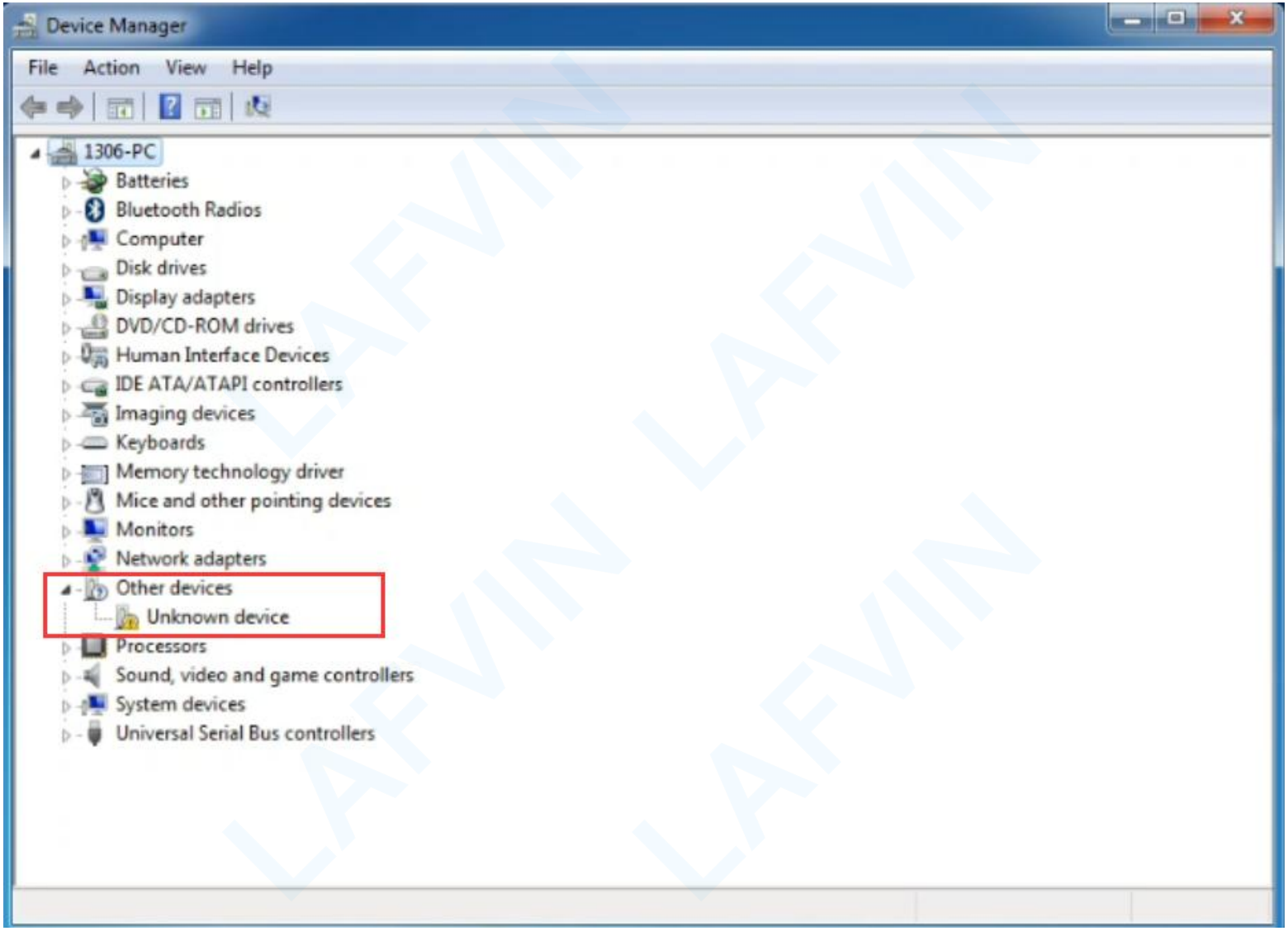


How to Install Arduino Driver

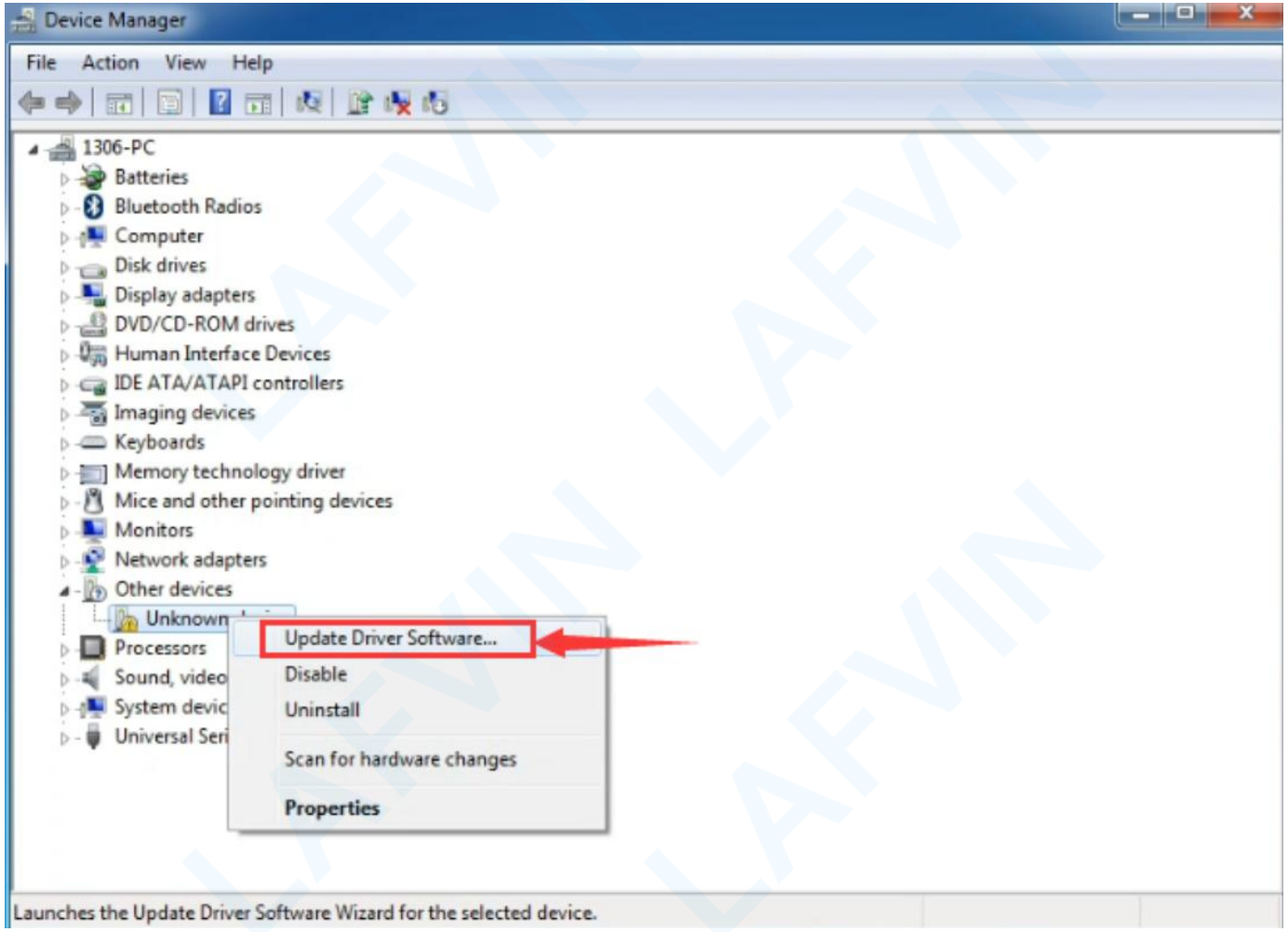
Next, we will introduce the driver installation of UNO R3 development board. The driver installation may have slight differences in different computer systems. So in the following let's move on to the driver installation in the Window system.

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.

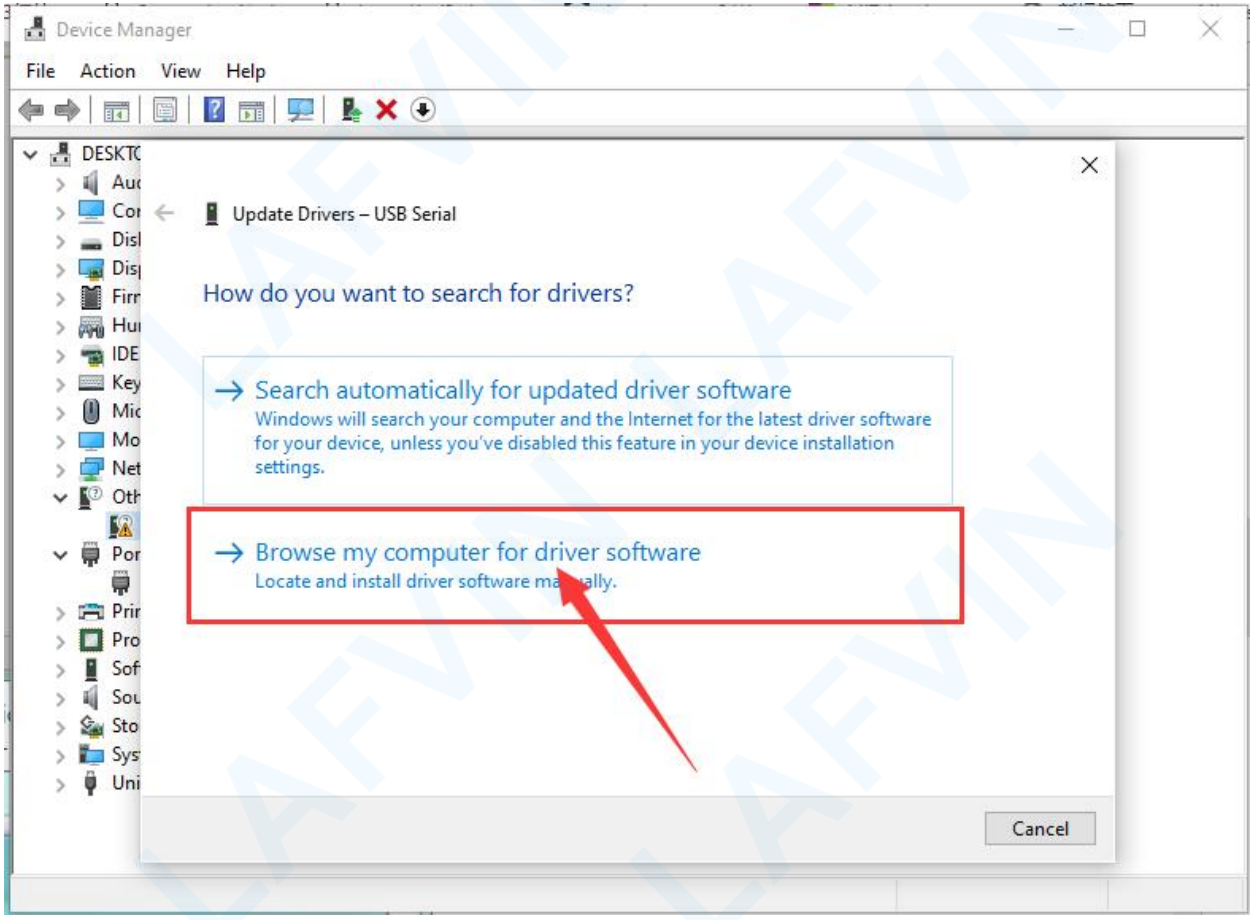
When you connect UNO board to your computer at the first time, right click the icon of your
"Computer" —>for "Properties"—> click the "Device manager" ,
under "Other Devices"or"USB-Serial", you should see an icon for "Unknown device" with a little yellow warning triangle next to it. This is your Arduino.Or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

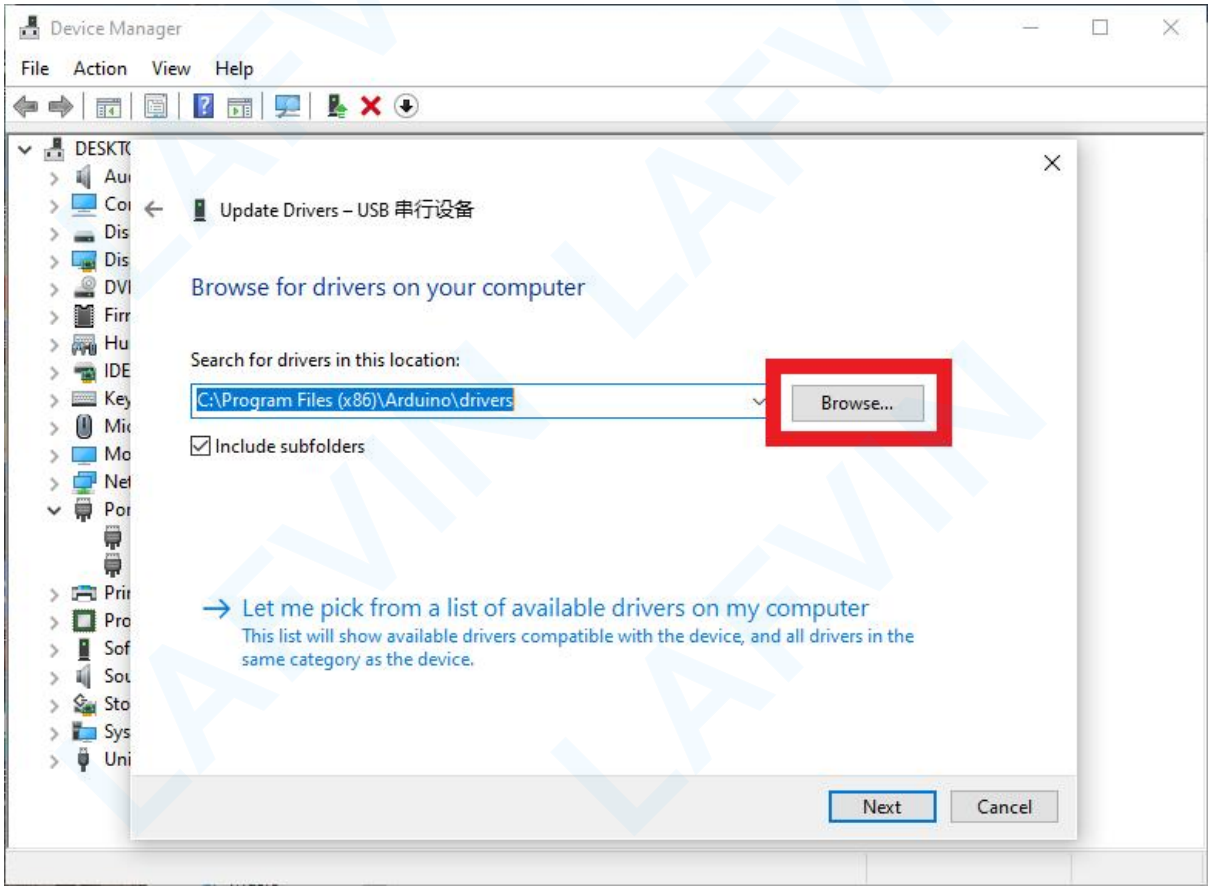


Then it will be prompted to either “Search Automatically for updated driver software” or “Browse my computer for driver software”. Shown as below. In this page, select “Browse my computer for driver software”.



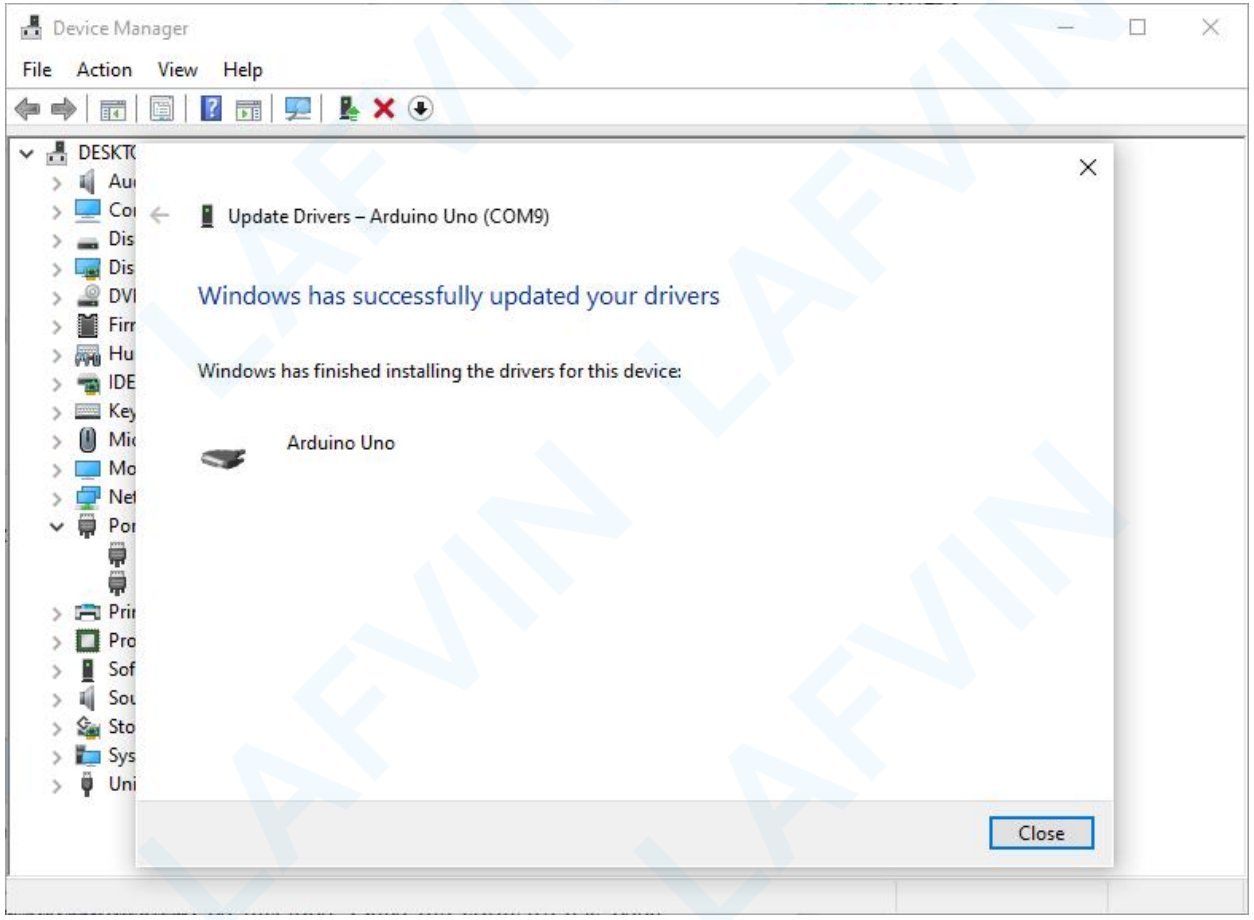
Right-click on the device and select the top menu option (Update Driver Software...).

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**)

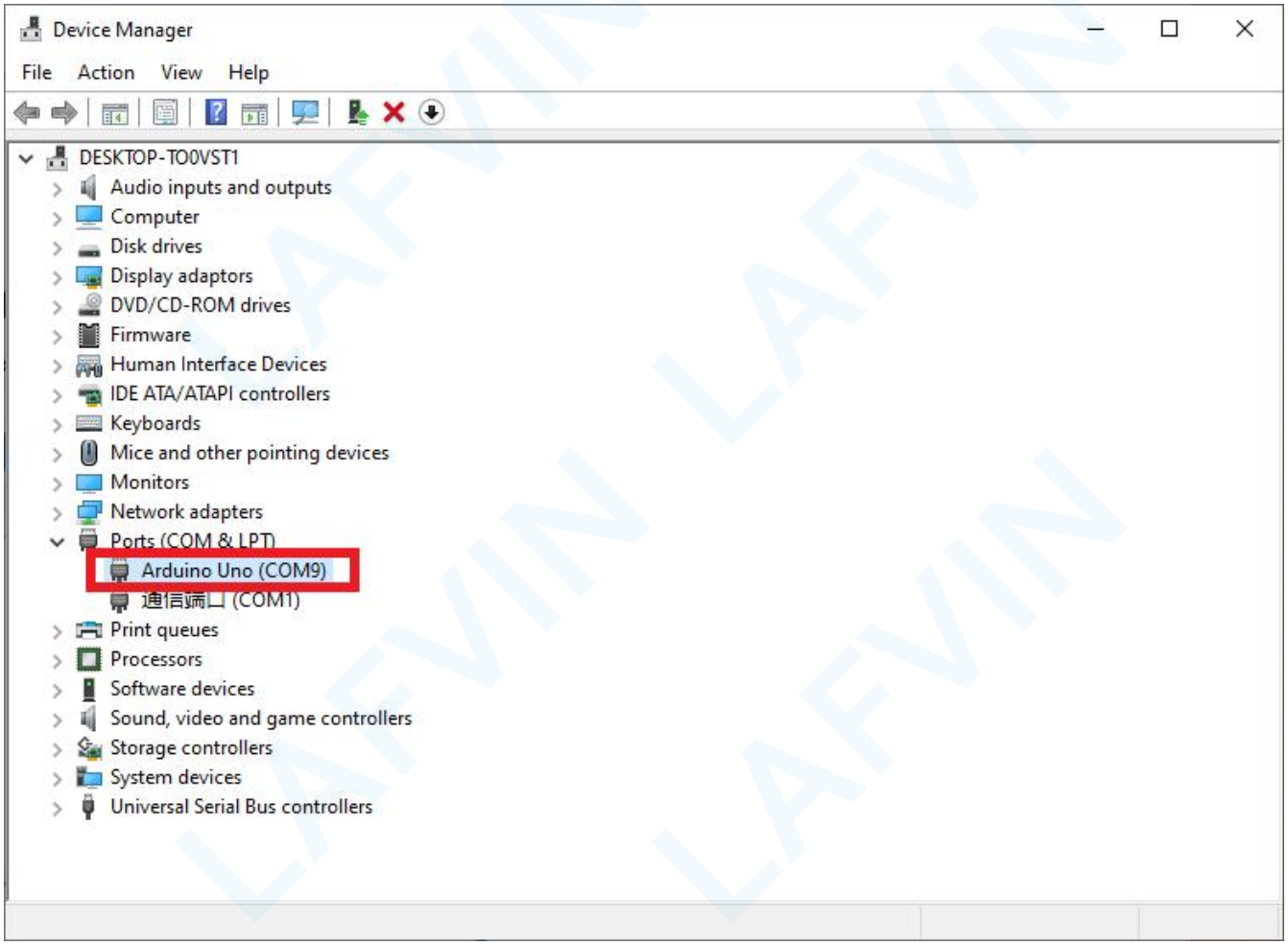


Click “Next” and you may get a security warning, if so, allow the software to be installed.

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?

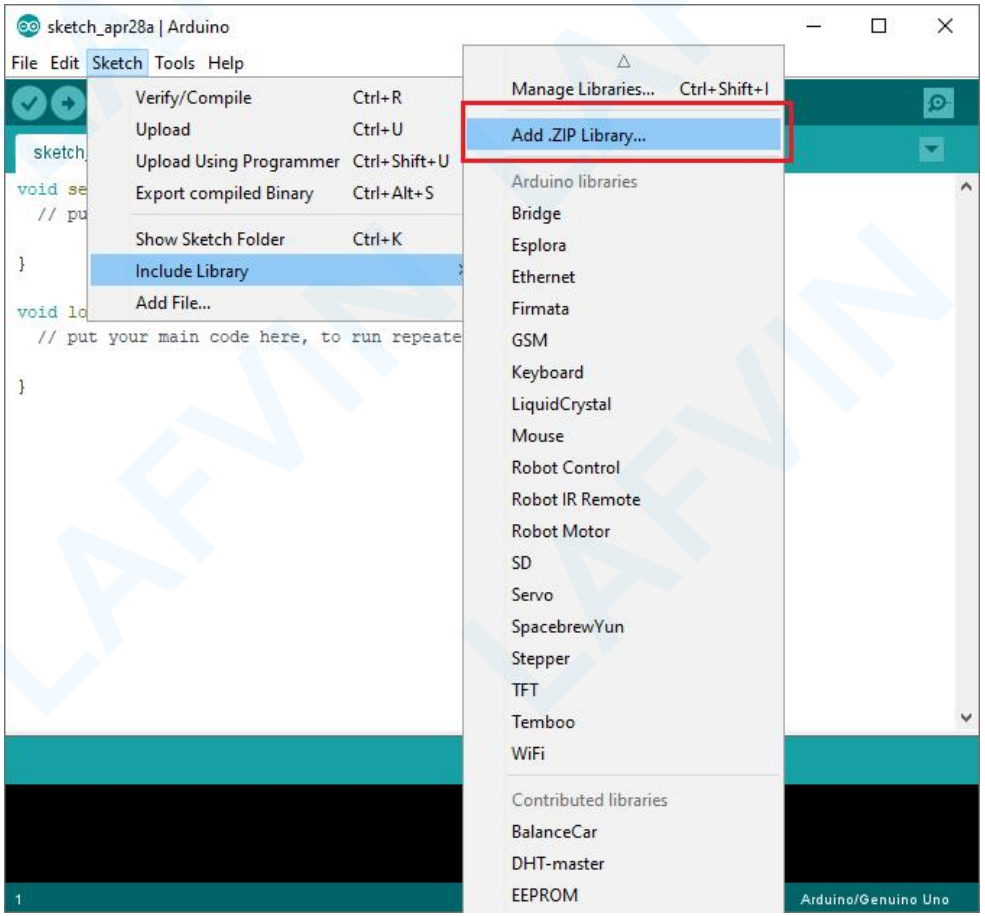
Libraries are a collection of code that makes it easy for you to connect to a sensor, display, module, etc. For example, the built-in LiquidCrystal library makes it easy to talk to character LCD displays. There are hundreds of additional libraries available on the Internet for download. The built-in 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

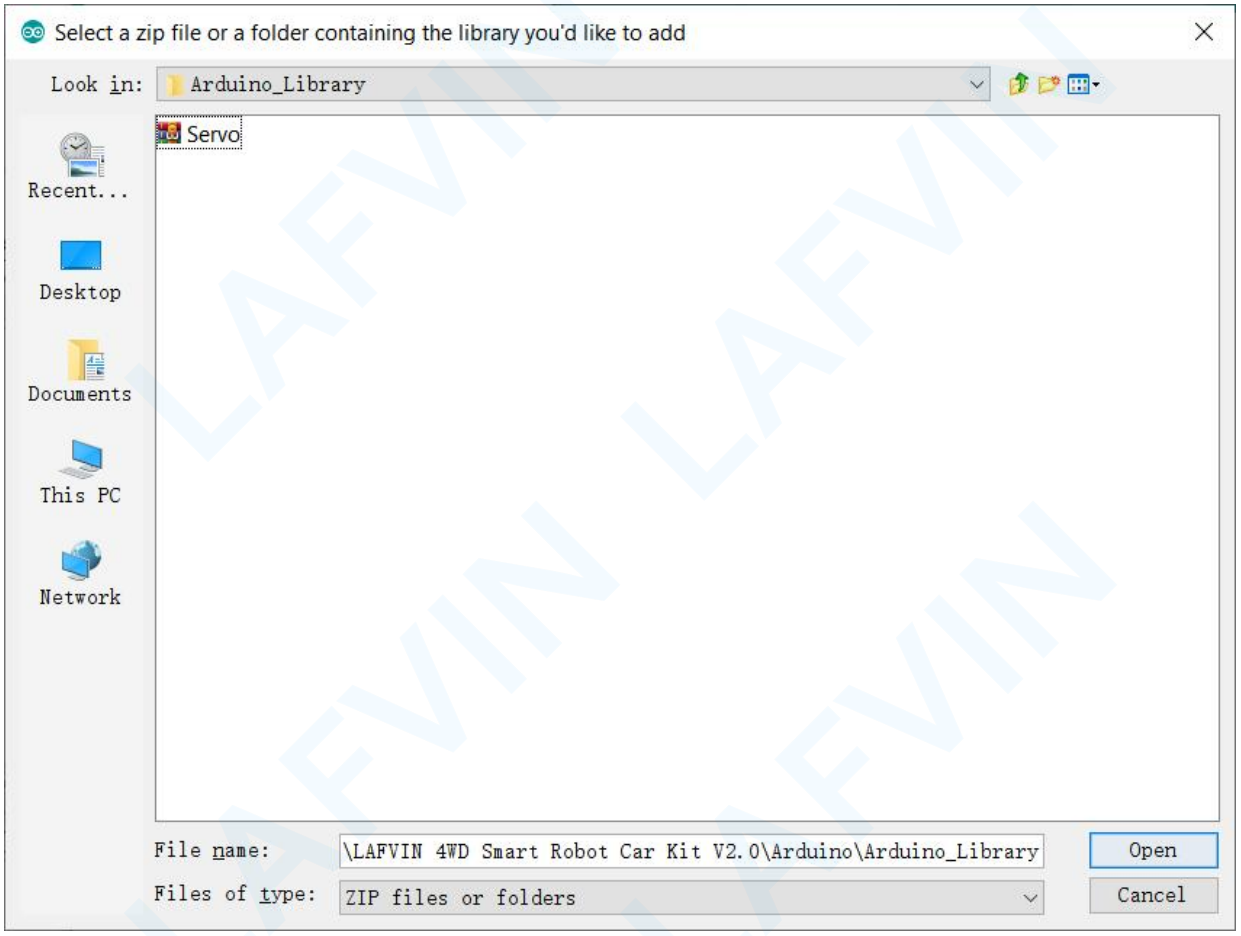
Importing a .zip Library

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".



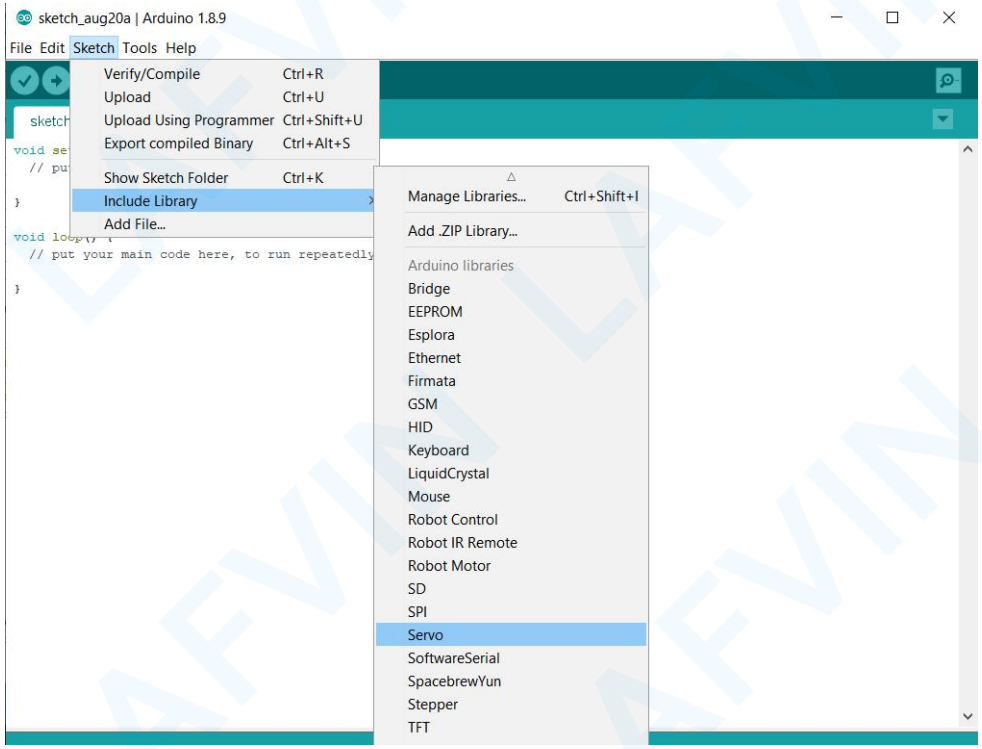
You will be prompted to select the library you would like to add. Navigate to the .zip file's location and open it.



(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.



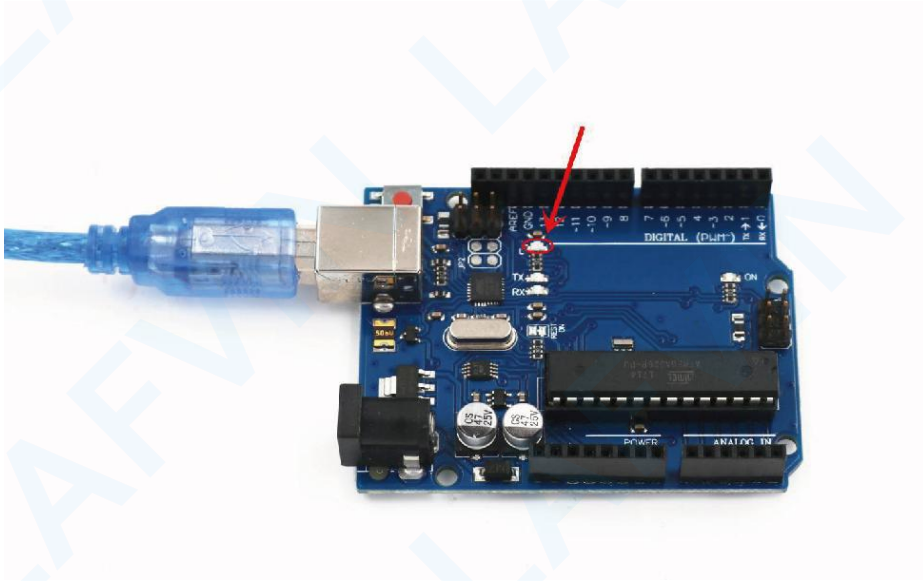
Those two are the most common approaches. MAC and Linux systems can be handled likewise. The manual installation to be introduced below as an alternative may be seldom used and users with no needs may skip it.

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.



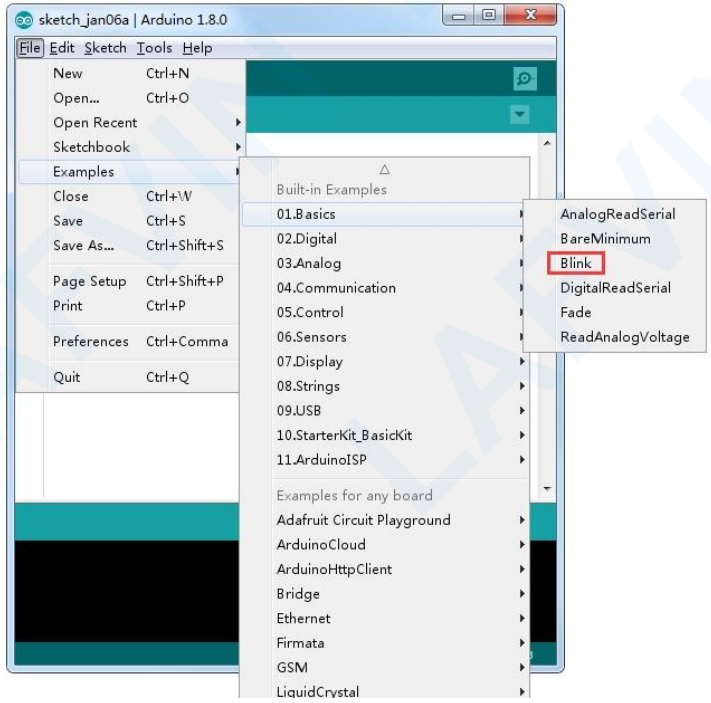
You may find that your UNO board's 'L' LED already blinks when you connect it to a USB plug. This is because the boards are generally shipped with the 'Blink' sketch pre-installed.

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

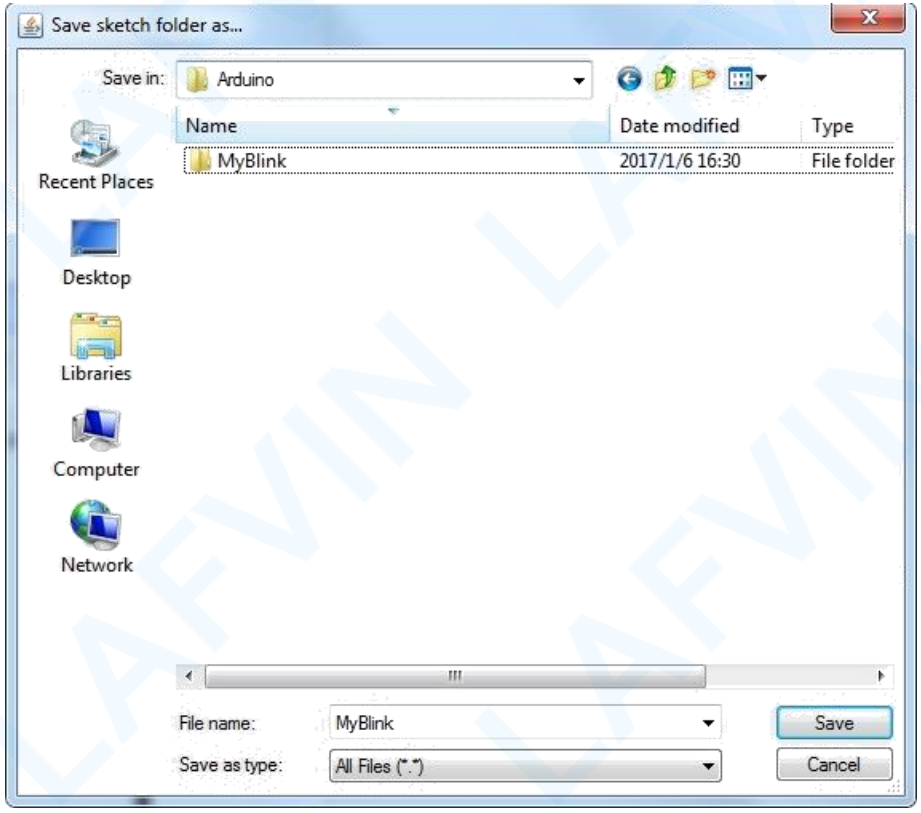


When the sketch window opens, enlarge it so that you can see the entire sketch in the window.

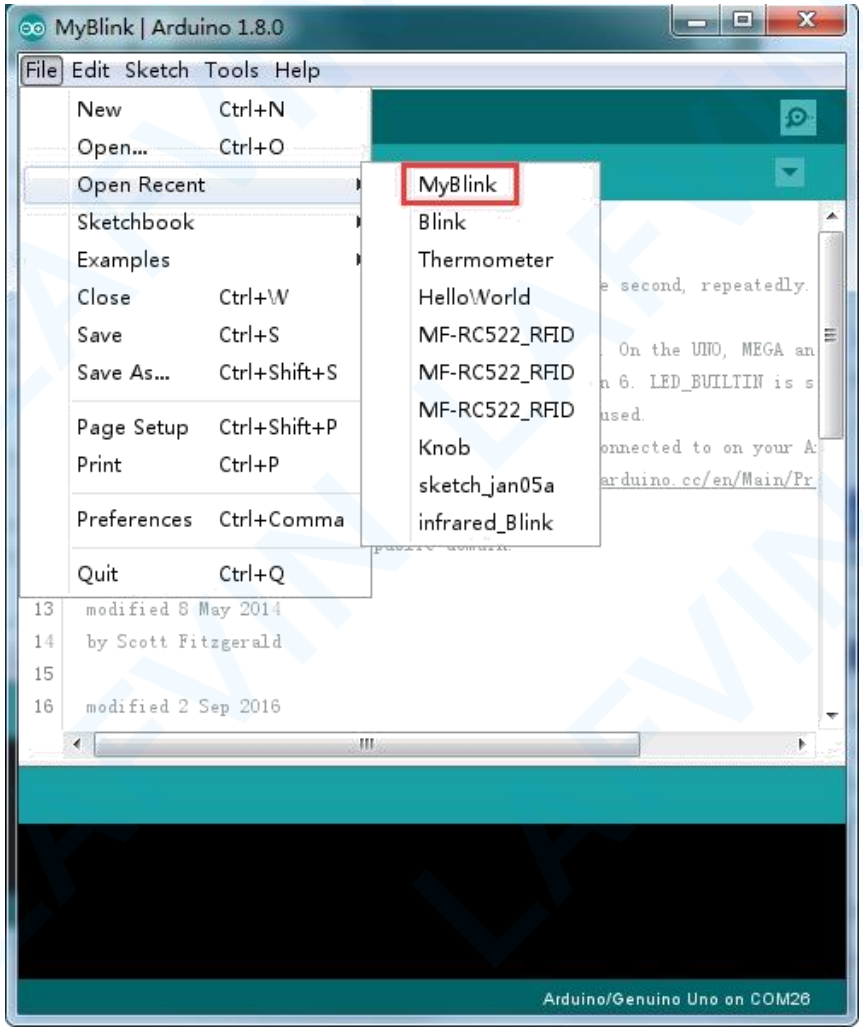
The example sketches included with the Arduino IDE are 'read-only'. That is, you can upload them to an UNO R3 board, but if you change them, you cannot save them as the same file.

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

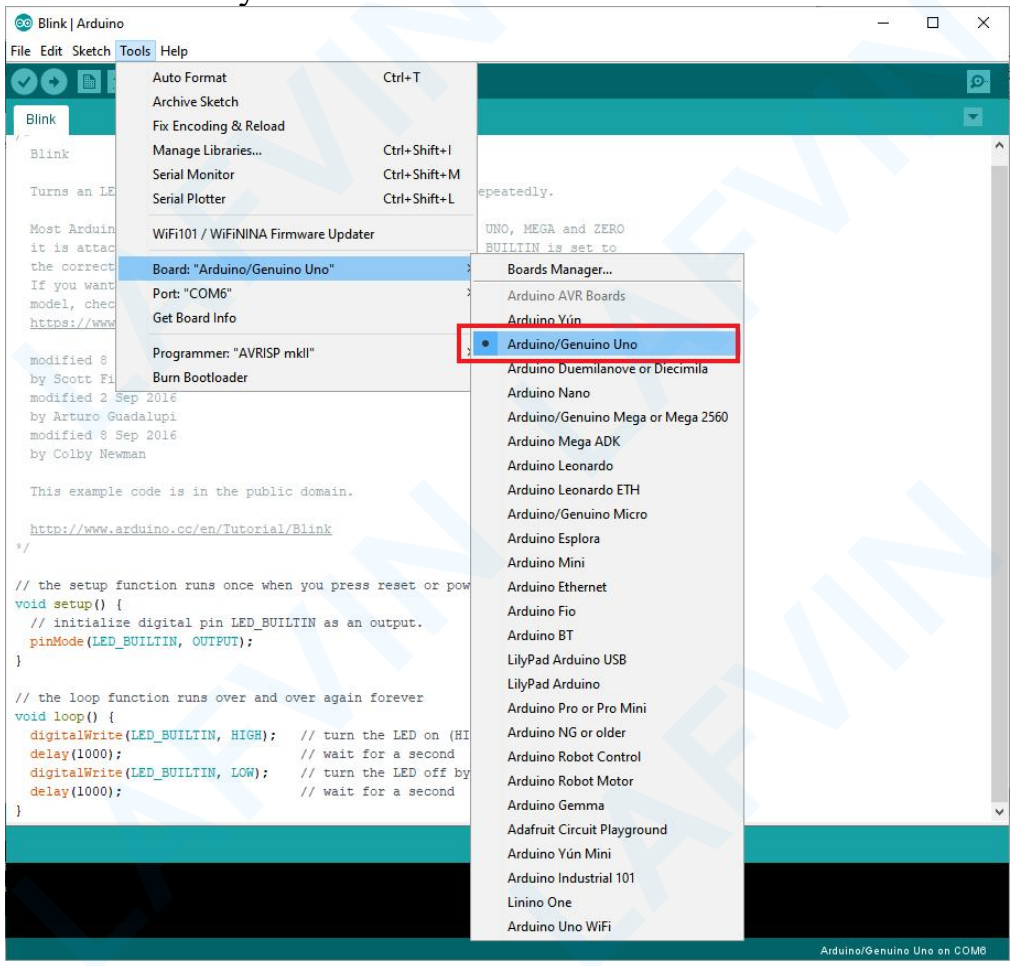
From the File menu on the Arduino IDE, select 'Save As..' and then save the sketch with the name 'MyBlink'.

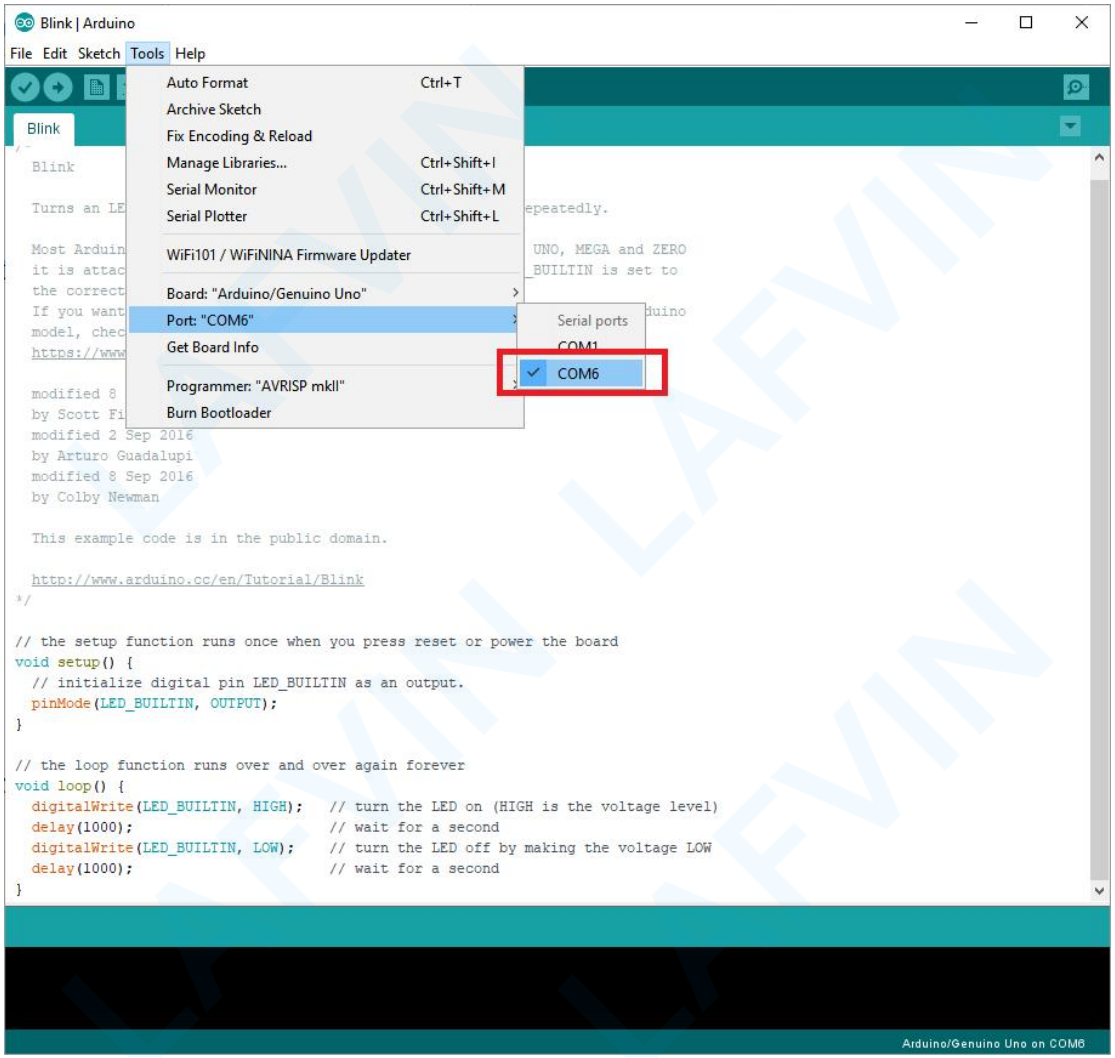


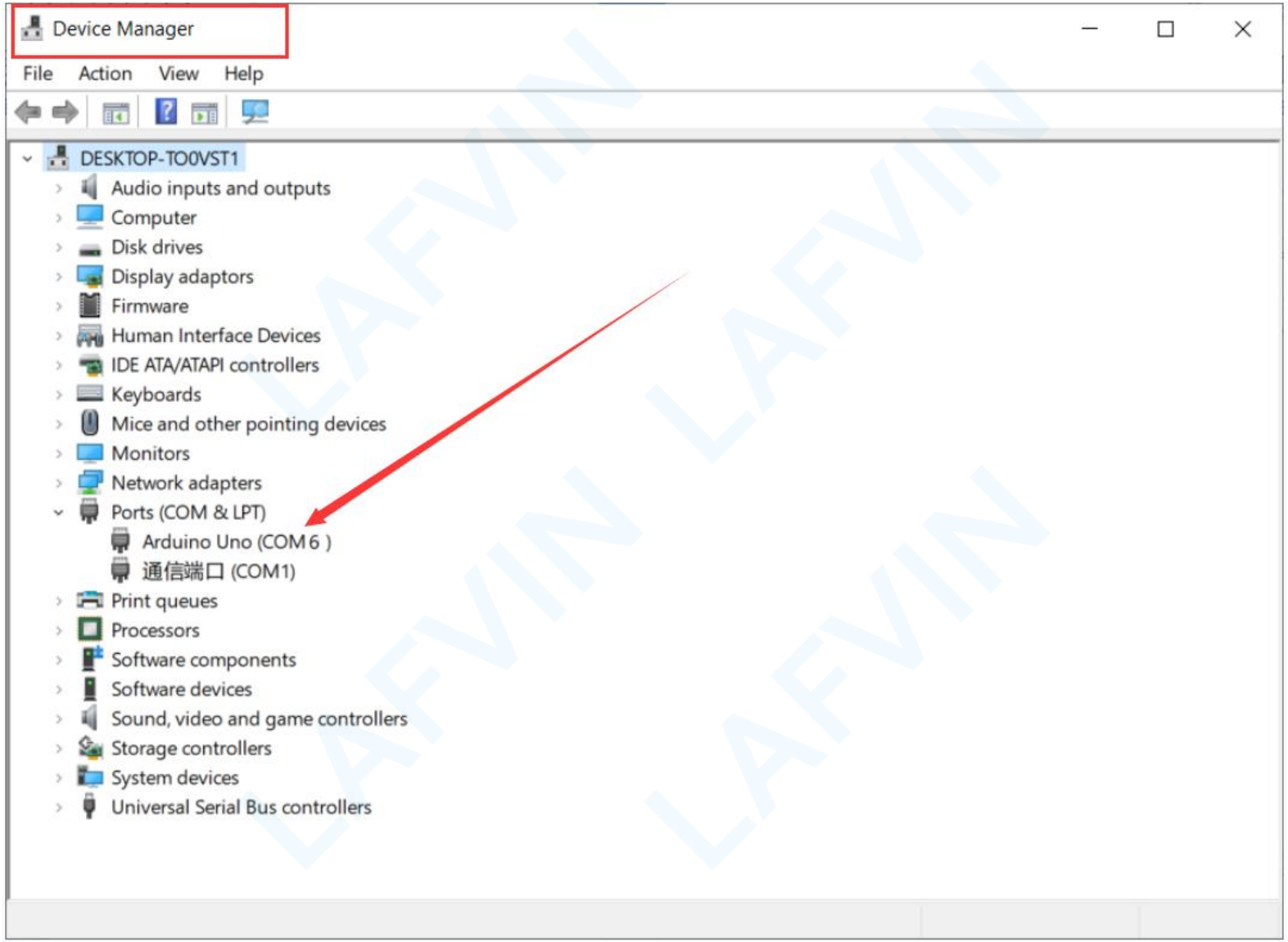
You have saved your copy of 'Blink' in your sketchbook. This means that if you ever want to find it again, you can just open it using the File > Sketchbook menu option.



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
Blink
Turns an LE
epeatedly.
Most Arduin
UNO, MEGA and ZERO
it is attac
_BUILTIN is set to
the correct
If you want
duino
model, chec
https://www
modified 8
by Scott Fi
modified 2 Sep 2016
by Arturo Guadalupi
modified 8 Sep 2016
by Colby Newman
This example code is in the public domain.
http://www.arduino.cc/en/Tutorial/Blink
*/
// the setup function runs once when you press reset or power the board
void setup() {
// initialize digital pin LED_BUILTIN as an output.
pinMode(LED_BUILTIN, OUTPUT);
}
// the loop function runs over and over again forever
void loop() {
digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
delay(1000); // wait for a second
digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
delay(1000); // wait for a second
}
Arduino/Genuino Uno on COM6


Device Manager
File Action View Help
DESKTOP-TO0VST1
Audio inputs and outputs
Computer
Disk drives
Display adaptors
Firmware
Human Interface Devices
IDE ATA/ATAPI controllers
Keyboards
Mice and other pointing devices
Monitors
Network adapters
Ports (COM & LPT)
Arduino Uno (COM6)
通信端口 (COM1)
Print queues
Processors
Software components
Software devices
Sound, video and game controllers
Storage controllers
System devices
Universal Serial Bus controllers

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.

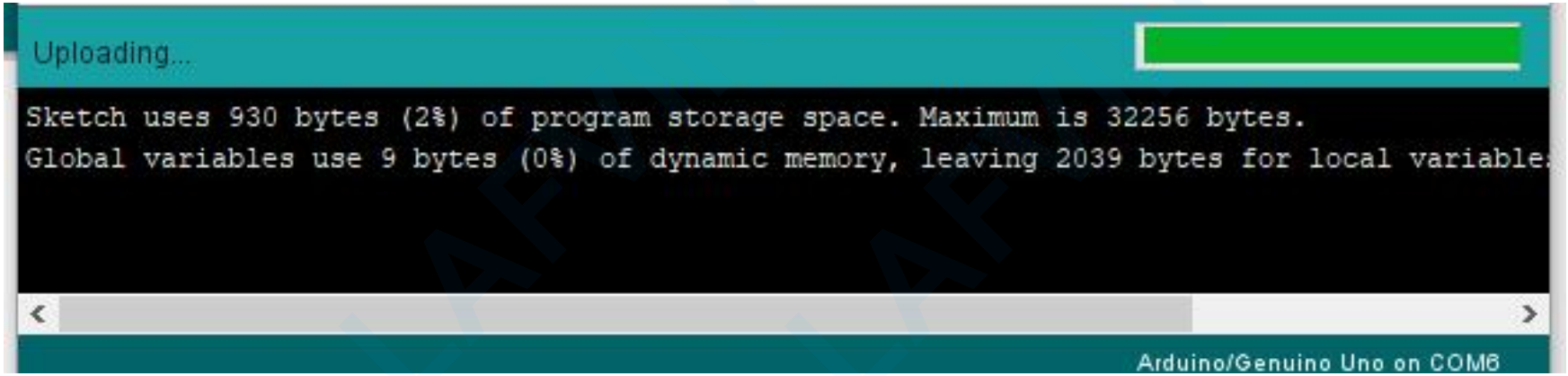
The Arduino IDE will show you the current settings for board at the bottom of the window.

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

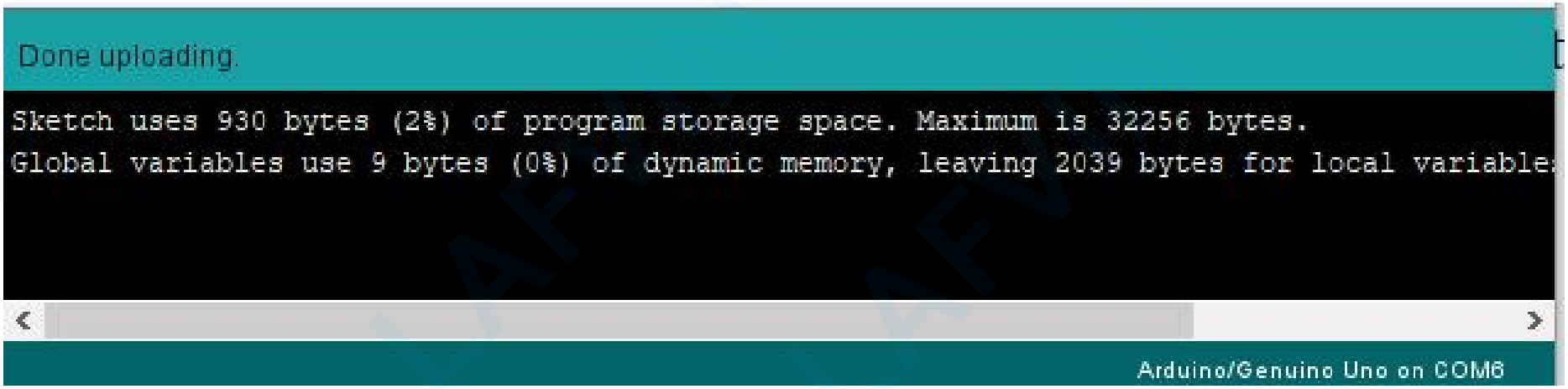
If you watch the status area of the IDE, you will see a progress bar and a series of messages. At first, it will say 'Compiling Sketch...'. This converts the sketch into a format suitable for uploading to the board.

]

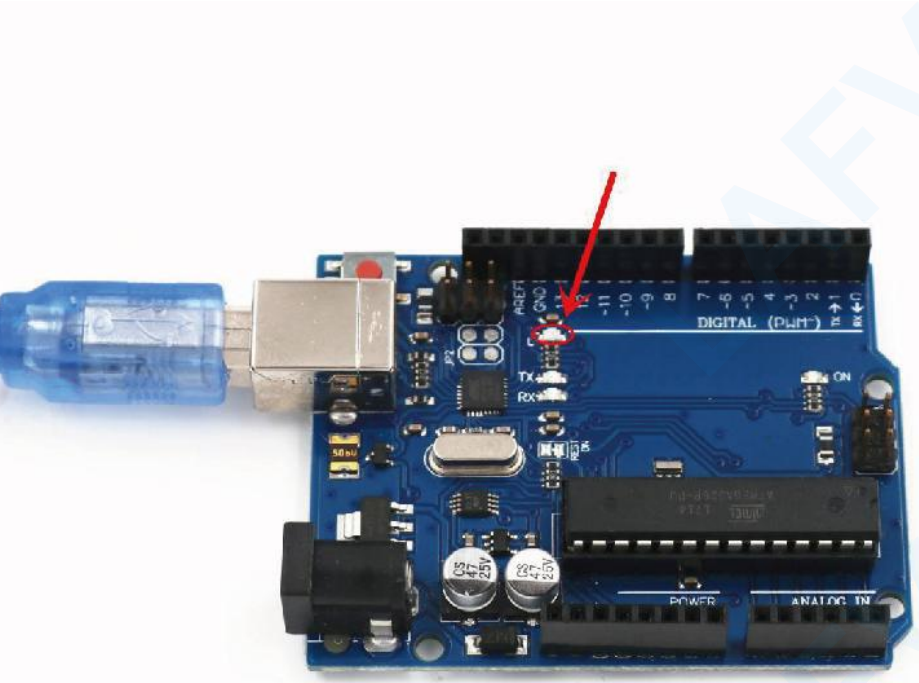
Next, the status will change to 'Uploading'. At this point, the LEDs on the Arduino should start to flicker as the sketch is transferred.



Finally, the status will change to 'Done uploading'.



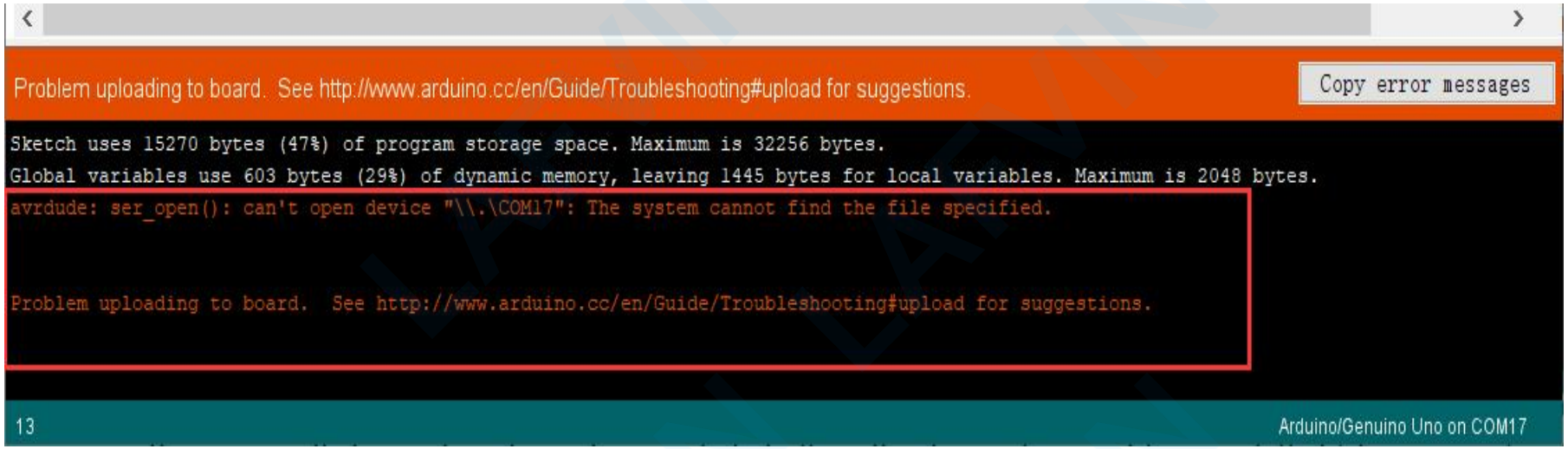
After uploading the code, you will see that the "L" LED on the Arduino UNO motherboard will light up for 1 second, and then turn off for 1 second, and the process will continue to loop.



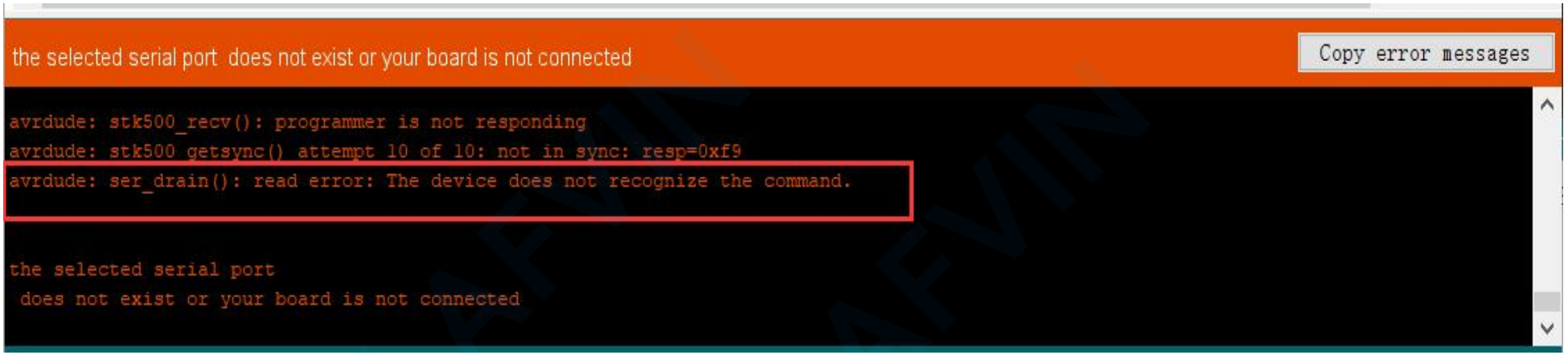

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```

If you are missing some steps, Arduino IDE may report some errors. You can refer to the following solutions.

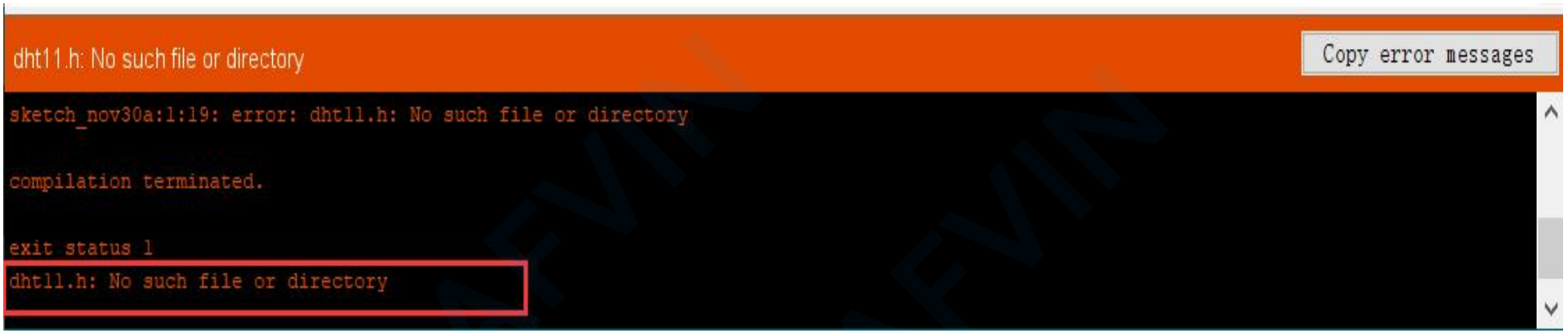


(1) It means that your board is not connected at all, or the drivers have not been installed (if necessary) or that the wrong serial port is selected.How to Install Arduino Driver



(2)It means that the communication serial port for uploading code is occupied. If you connect the Bluetooth module when uploading the code, you need to disconnect the Bluetooth module from the Arduino UNO expansion board, and then reinstall the Bluetooth module after uploading.

Another possibility is that the USB interface of the Arduino UNO is insufficiently powered, and you need to turn on the external power switch of the expansion board.



(3)If you need to use library files in some project programs, you may encounter this error.

It means that some library files are used in the code, but you did not add the corresponding library files to the arduino IDE before uploading the code.How to Add Libraries

Lesson 2 Getting Started with Mixly

Introduction of Mixly Software

Mixly is a free open-source graphical Arduino programming software, based on Google's Blockly graphical programming framework, and developed by Mixly Team@ BNU. It is a free open-source graphical programming tool for creative electronic development; a complete support ecosystem for creative e-education; a stage for maker educators to realize their dreams.Although there is an Ardublock graphical programming software launched by Arduino official, Ardublock is not perfect enough, and many common functions cannot be realized.

Design Concept:

(1) Usability

Mixly is designed to be completely green. Currently. It can run on Windows xp and above.

(2) Simplicity

Mixly uses the Blockly graphical programming engine to replace complex text manipulation with graphical building blocks, providing a good foundation for beginners to get started quickly.

1 Use the different color icons to represent different types of functional blocks, very convenient for users to classify.

2 Provide default options in the composite function block to effectively reduce the number of user drags.

3 Integrate all the features of the software in the same interface.

4 Provide the reference tutorial and code examples.

(3) Functionality

It has versatile functions. Mixly can almost implement all the functions that Arduino IDE has. Support all official development boards of arduino.

(4) Continuity

The goal of the graphical programming system is definitely not to replace the original text programming method,but to better understand the programming principles and program thinking through graphical programming, and lay the foundation for future text programming. It is also the design philosophy for Mixly. More continuous content has been added to the design of the software to protect the user's learning outcomes. To be specific, it includes the introduction of variable types, the consistency of text programming as much as possible in the design of the module, and the support of

both graphical and text programming.

(5) Ecological

The most important design concept of Mixly is its ecological feature, which can distinguish it from other Arduino graphical programming.In order to achieve sustainable development, Mixly is designed to allow manufacturers to develop their own unique modules.But users require JavaScript programming foundation to make this part of the module).It also allows users directly use Mixly's graphical programming function to generate common modules (such as LED digital display, buzzer broadcast, etc. Users are able to make this part of the module only using Mixly). Both of the two kinds of modules mentioned above can be imported into the Mixly system through the "Import" function, thereby realizing the user's own value in the popularity of Mixly software.

User Groups:

From the above design concept, it can be seen that Mixly is suitable for primary and secondary school students to cultivate programming thinking. It is also available for quick programming when creating a work.it is good for those lovely friends who don't want to learn text programming, but want to do some small works with intelligent control.

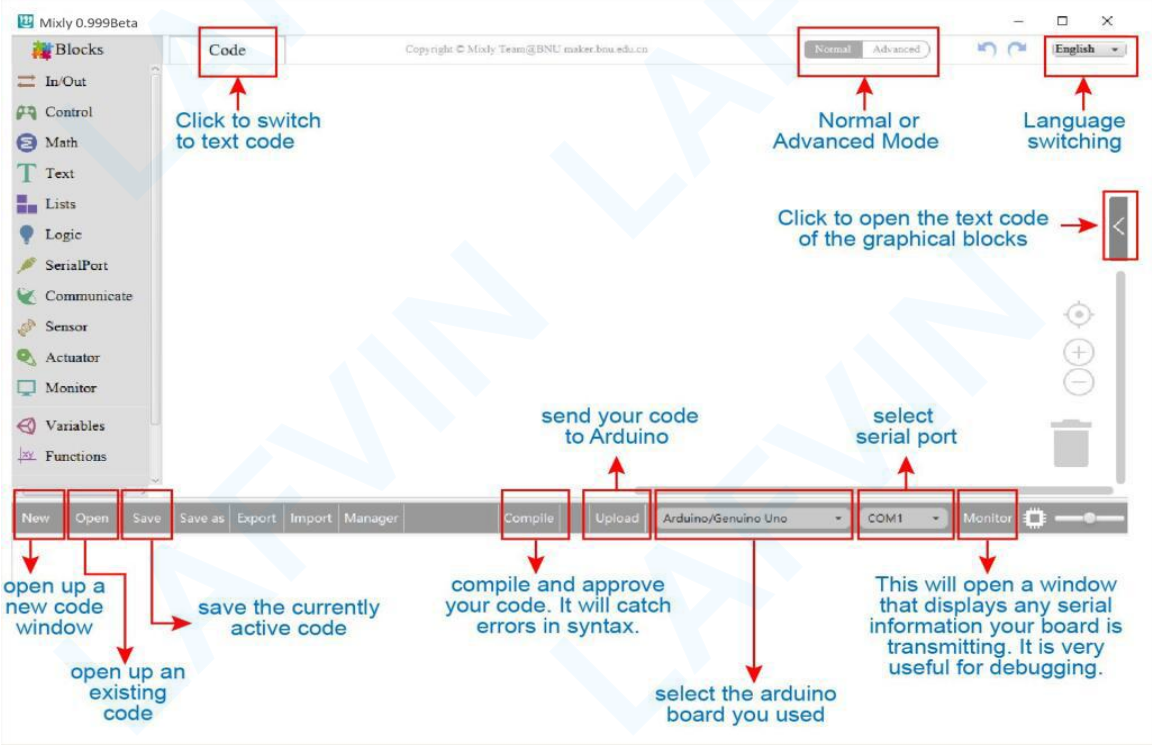
Mixly Blocks Functions:

System Functions

Look at the main interface of Mixly, it includes five parts, that is, Blocks selection, code edit, text code (hidden), system function and message prompt area. Shown below.

Some common functions:

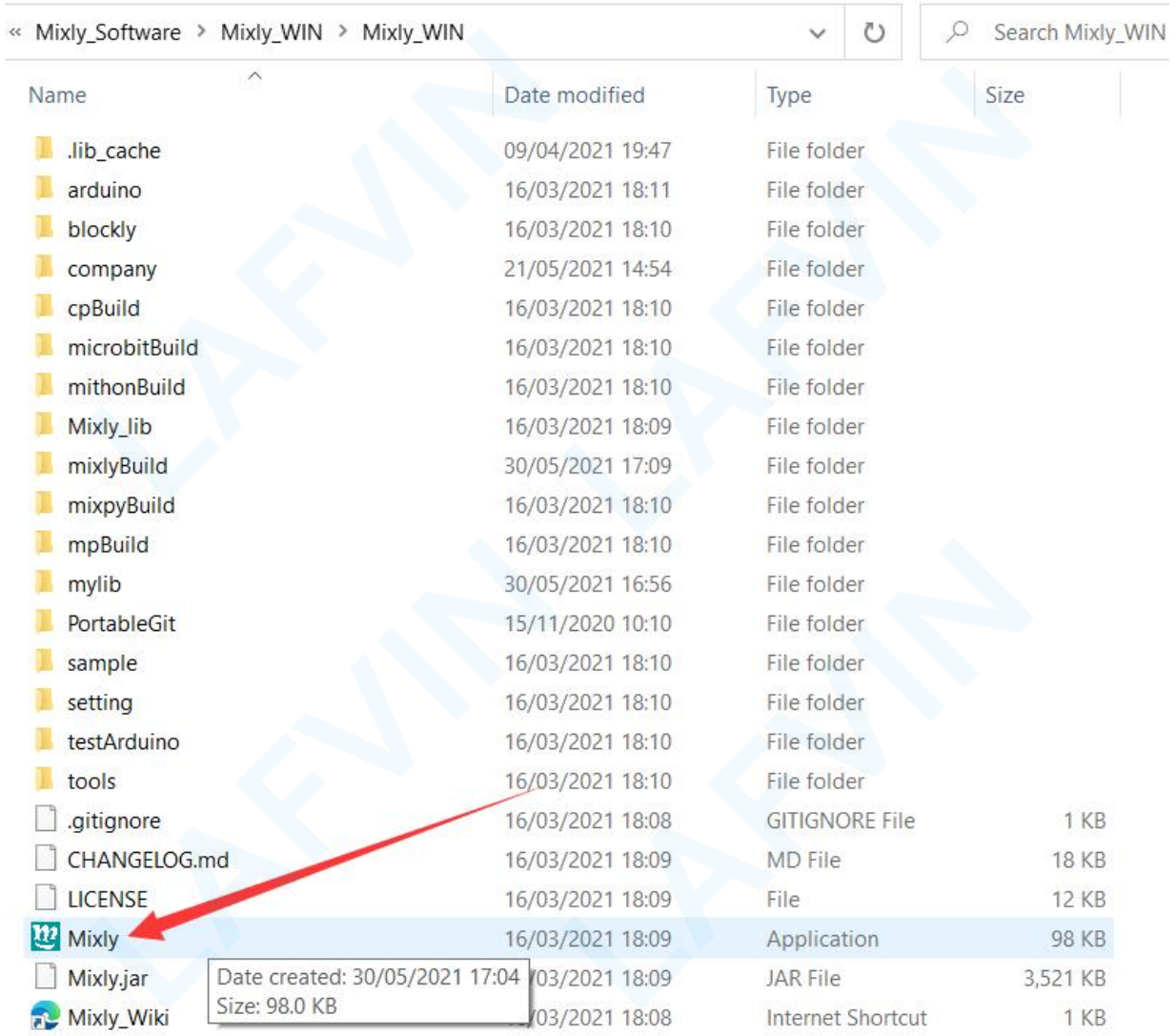
Through this interface, you can complete the code compile、upload、save and manage. It support four remove methods: drag it left out code window, or drag to Recycle Bin, delete key, or right-click to delete block. It supports four languages: English、Español (Spanish)、中文简体(Chinese Simplified)、中文繁体(Chinese Traditional).



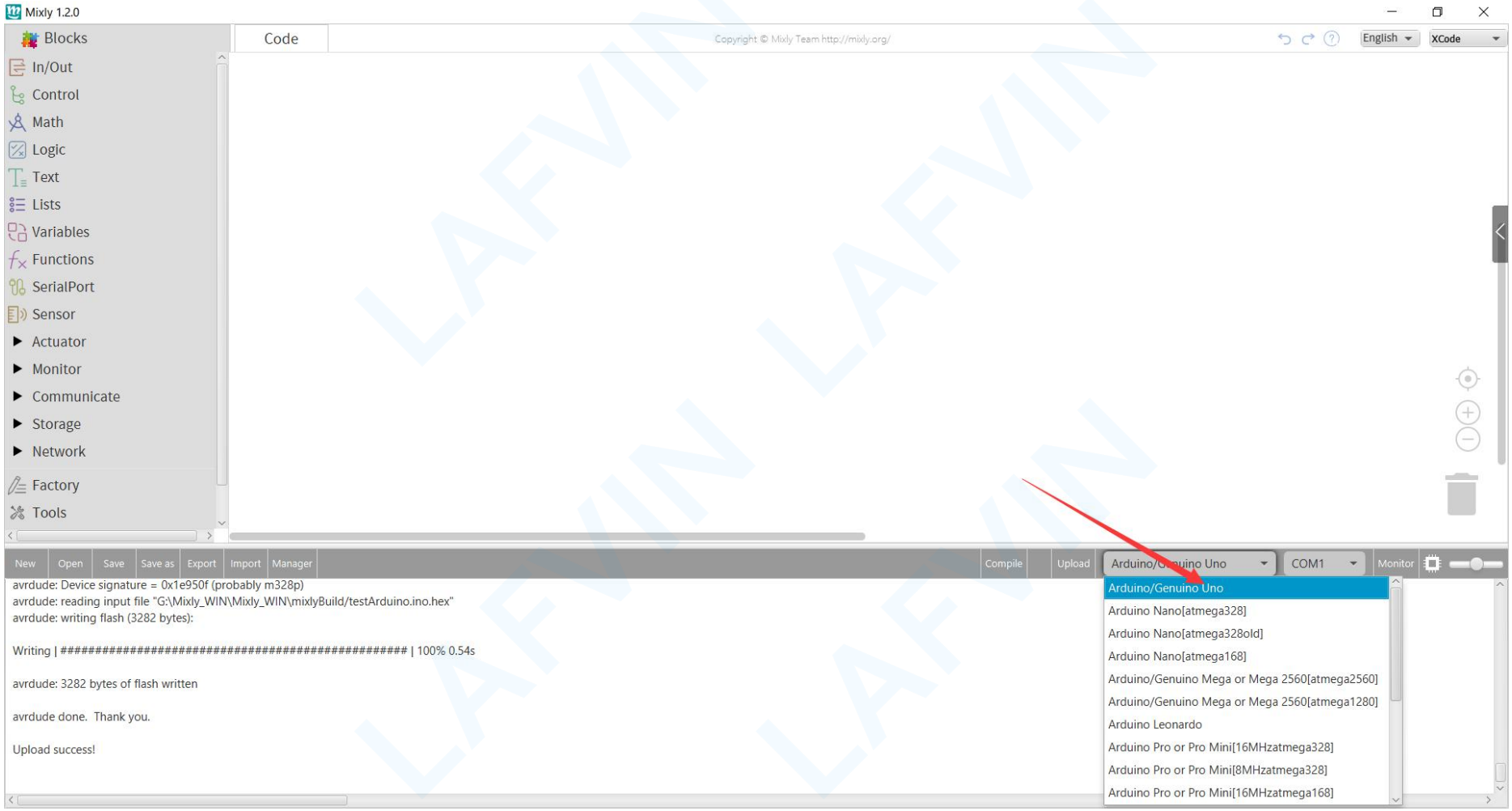
How to Install Mixly Software

First decompress the mixly programming software file

After unzipping you will get the following files(Note: Do not include special characters such as Chinese, spaces, brackets, etc. in the unzipped directory.The name of the uncompressed path cannot have a double space bar. If the named name needs a separator, you can use an underscore to replace the double space bar, for example,you should name the folder Mixly_Sofewave instead of Mixly Sofewave)

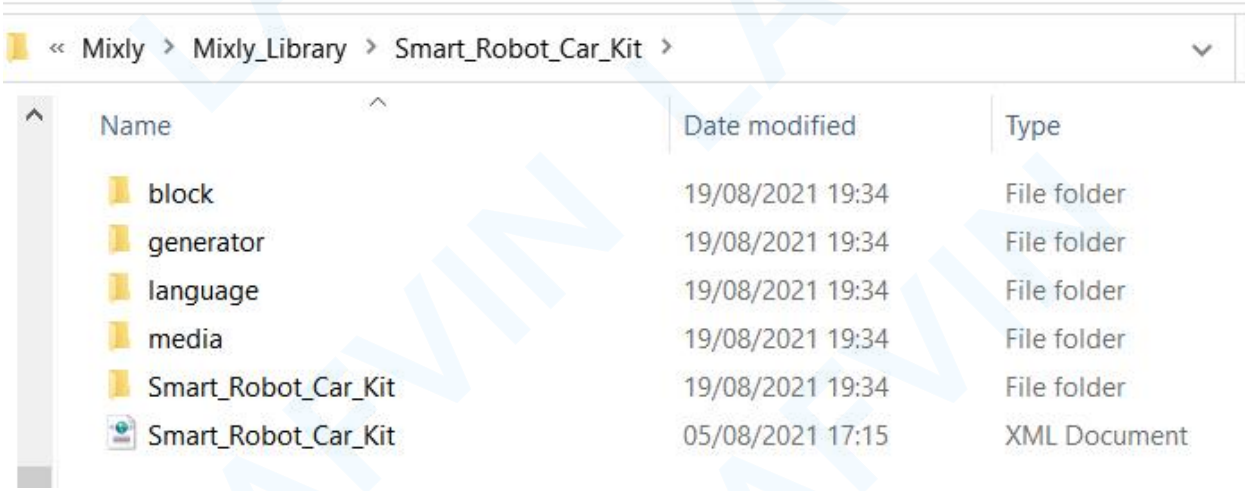

Mixly_Software › Mixly_WIN › Mixly_WIN
Search Mixly_WIN
Name
Date modified
Type
Size
.lib_cache 09/04/2021 19:47 File folder
arduino 16/03/2021 18:11 File folder
blockly 16/03/2021 18:10 File folder
company 21/05/2021 14:54 File folder
cpBuild 16/03/2021 18:10 File folder
microbitBuild 16/03/2021 18:10 File folder
mithonBuild 16/03/2021 18:10 File folder
Mixly_lib 16/03/2021 18:09 File folder
mixlyBuild 30/05/2021 17:09 File folder
mixpyBuild 16/03/2021 18:10 File folder
mpBuild 16/03/2021 18:10 File folder
mylib 30/05/2021 16:56 File folder
PortableGit 15/11/2020 10:10 File folder
sample 16/03/2021 18:10 File folder
setting 16/03/2021 18:10 File folder
testArduino 16/03/2021 18:10 File folder
tools 16/03/2021 18:10 File folder
.gitignore 16/03/2021 18:08 GITIGNORE File 1 KB
CHANGELOG.md 16/03/2021 18:09 MD File 18 KB
LICENSE 16/03/2021 18:09 File 12 KB
Mixly 16/03/2021 18:09 Application 98 KB
Mixly.jar 03/2021 18:09 JAR File 3,521 KB
Mixly_Wiki 03/2021 18:08 Internet Shortcut 1 KB
Date created: 30/05/2021 17:04
Size: 98.0 KB

Double-click the mixly application file to open the software. And select the main control board as **Arduino/Genuino Uno**



How to Add Mixly Libraries

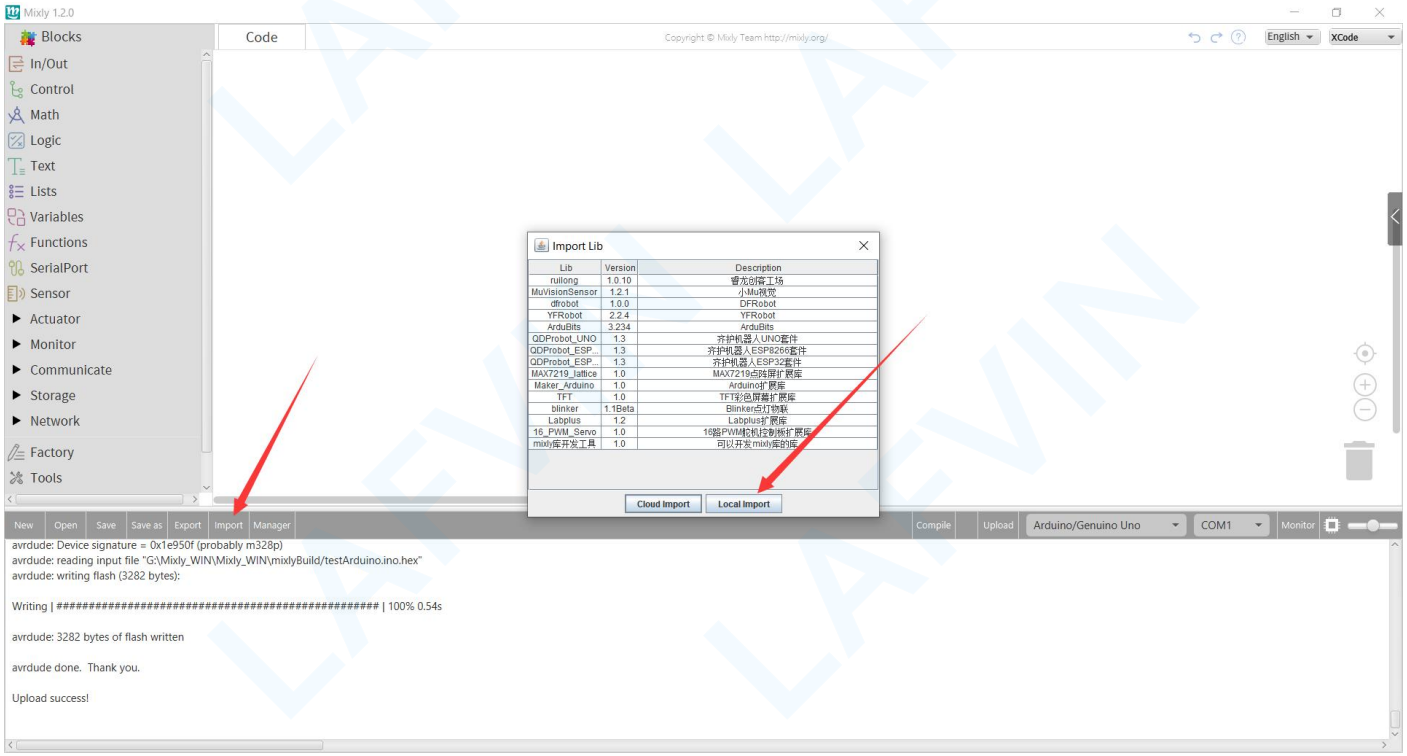
The mixly library file is an integrated graphical code customized for a certain control board. For example, we have written a mixly library file for smart home learning kit. You can directly load it into mixly software and use it directly, which brings convenience to your programming process.The general library file contains the following files, the last .xml file is the readable library file of the mixly software

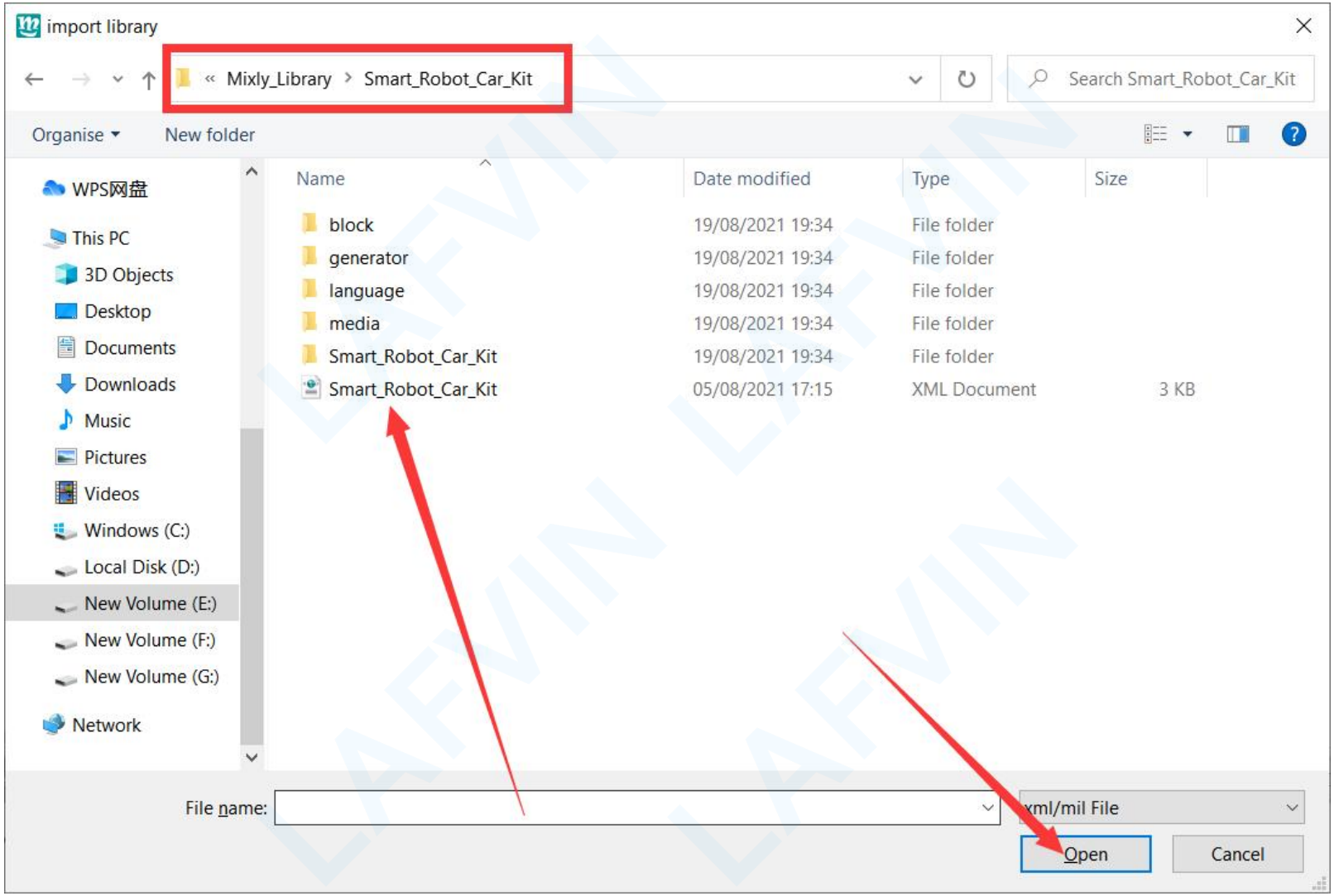


(Note:After completing the installation of the mixly software, the library file <LAFVIN Smart Robot Car Kit> already contains the automatic addition to the programming software, you can use it directly without adding and adding repeatedly.)

If you need to accidentally delete the library file, or you want to add other functional library files, you can refer to the following steps to add library files.

After opening the mixly software, select "**import**" and click **local import** .then select the path where the "**Smart Robot Car Kit**" folder is located,select the file --"**Smart Robot Car Kit.xml**", and click "open"

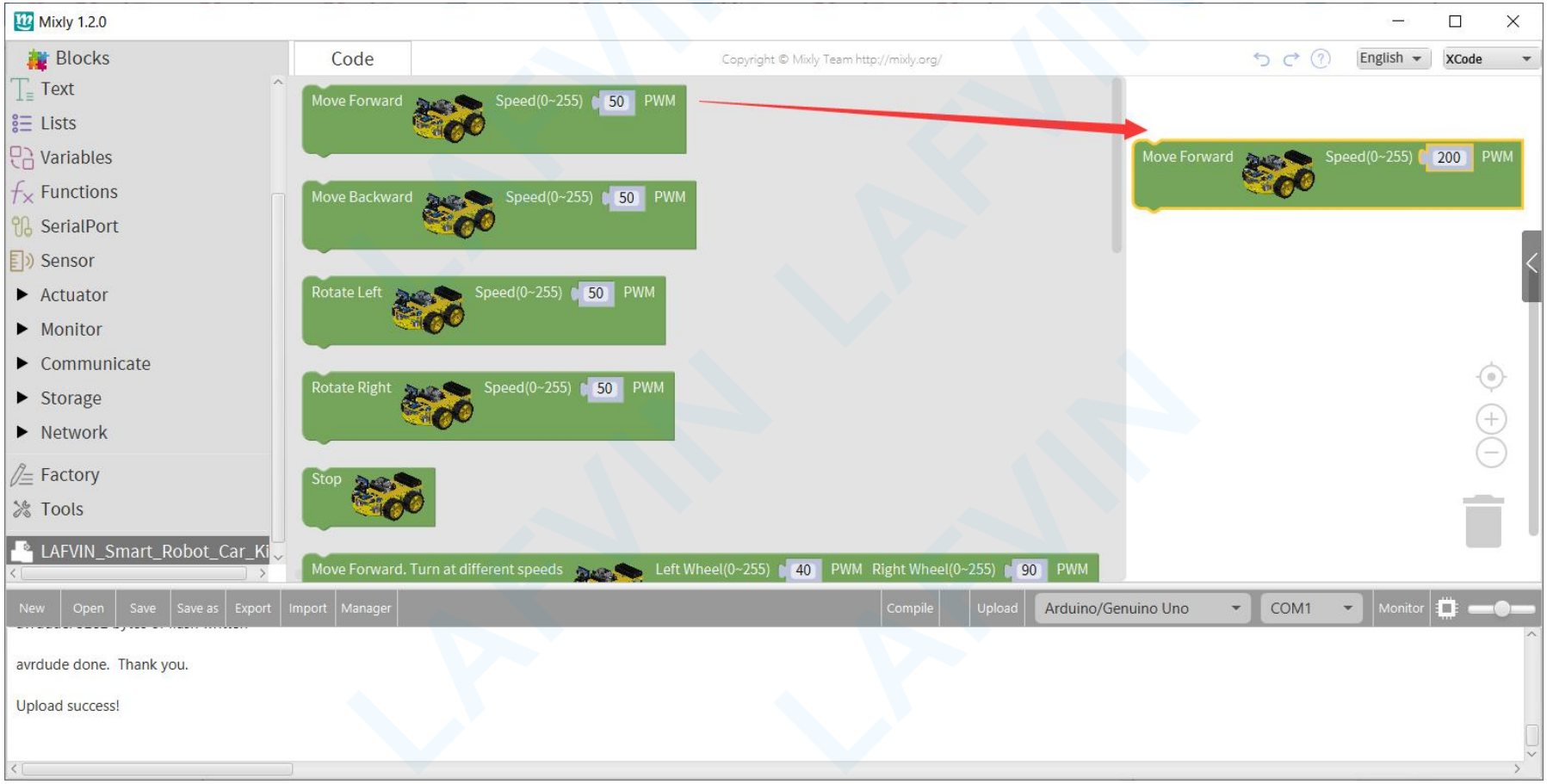



import library
« Mixly_Library › Smart_Robot_Car_Kit
Search Smart_Robot_Car_Kit
Organise
New folder
WPS网盘
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Windows (C:)
Local Disk (D:)
New Volume (E:)
New Volume (F:)
New Volume (G:)
Network
Name
Date modified
Type
Size
block 19/08/2021 19:34 File folder
generator 19/08/2021 19:34 File folder
language 19/08/2021 19:34 File folder
media 19/08/2021 19:34 File folder
Smart_Robot_Car_Kit 19/08/2021 19:34 File folder
Smart_Robot_Car_Kit 05/08/2021 17:15 XML Document 3 KB
File name:
xml/mil File
Open
Cancel

The figure below shows that the ”Smart_Robot_Tank”library file has been successfully imported into the mixly software.

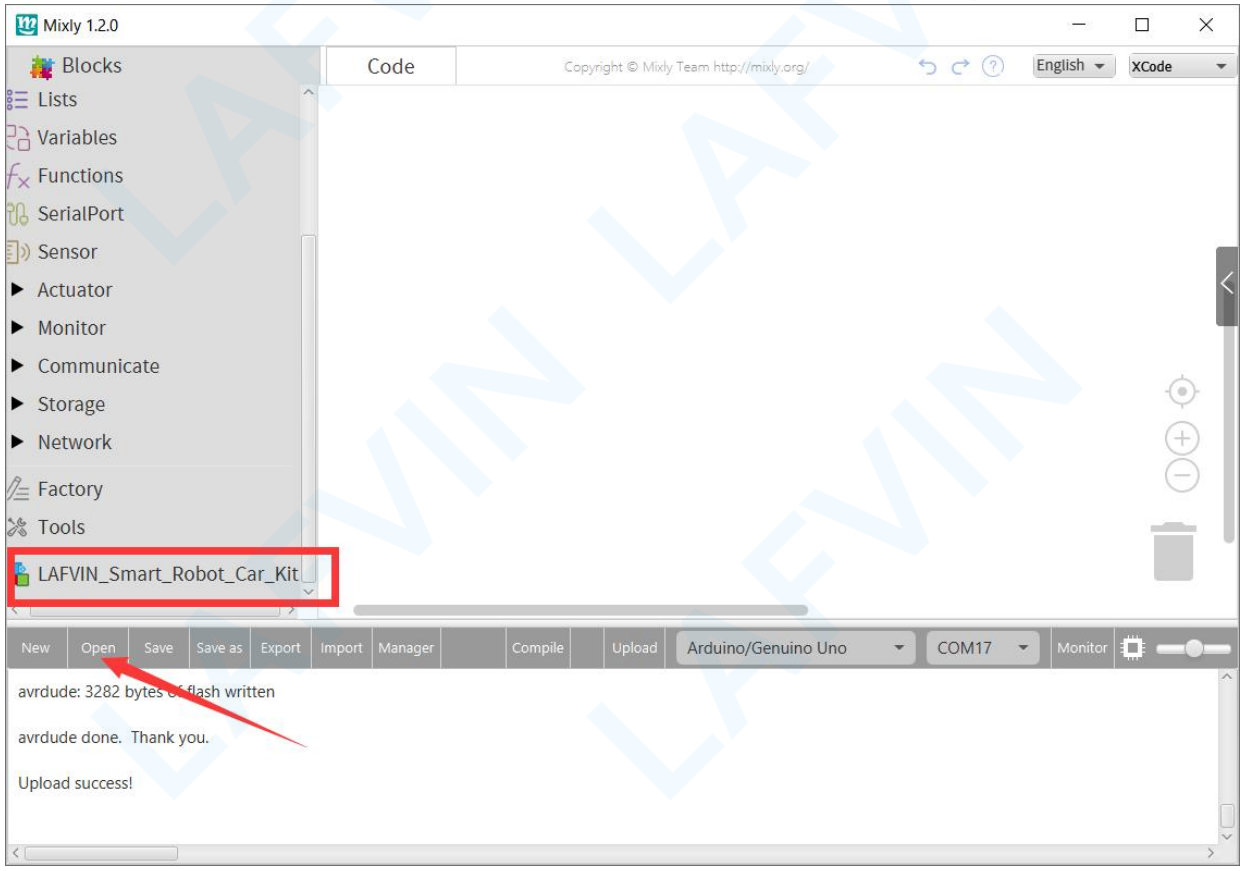
Next we write a program on the software and upload it to smart robot car.

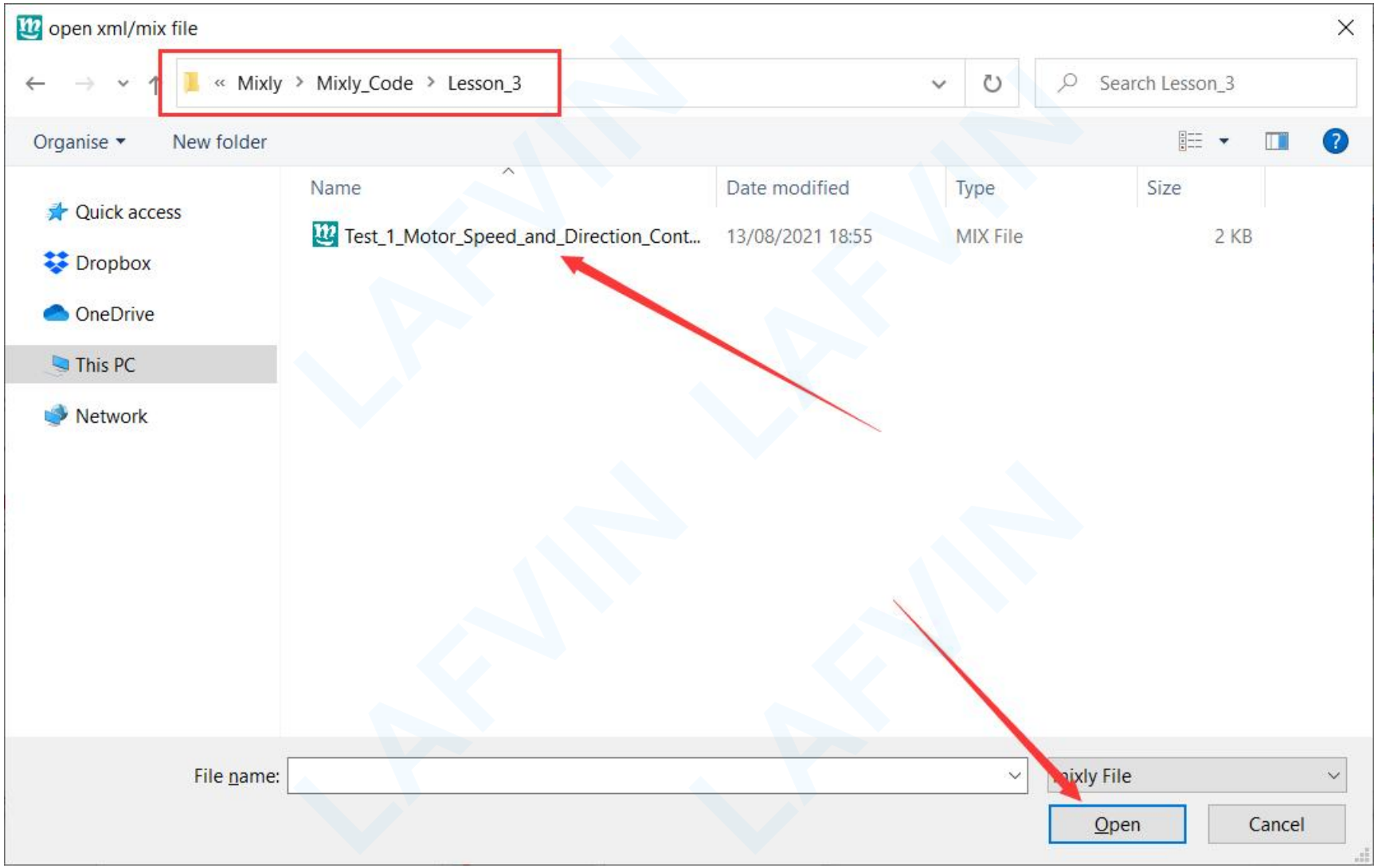
Drag the program block named "Move forward" to the programming area,and modify the PWM variable to 200.



How to Upload the Reference Program

Click "Open" on the mixly software and select the reference program.Take the reference program from Lesson 3 as an example,Open this source program in Mixly_Code>Lesson_3>”**Test_1_Motor_Speed_and_Direction_Control.mix”.**

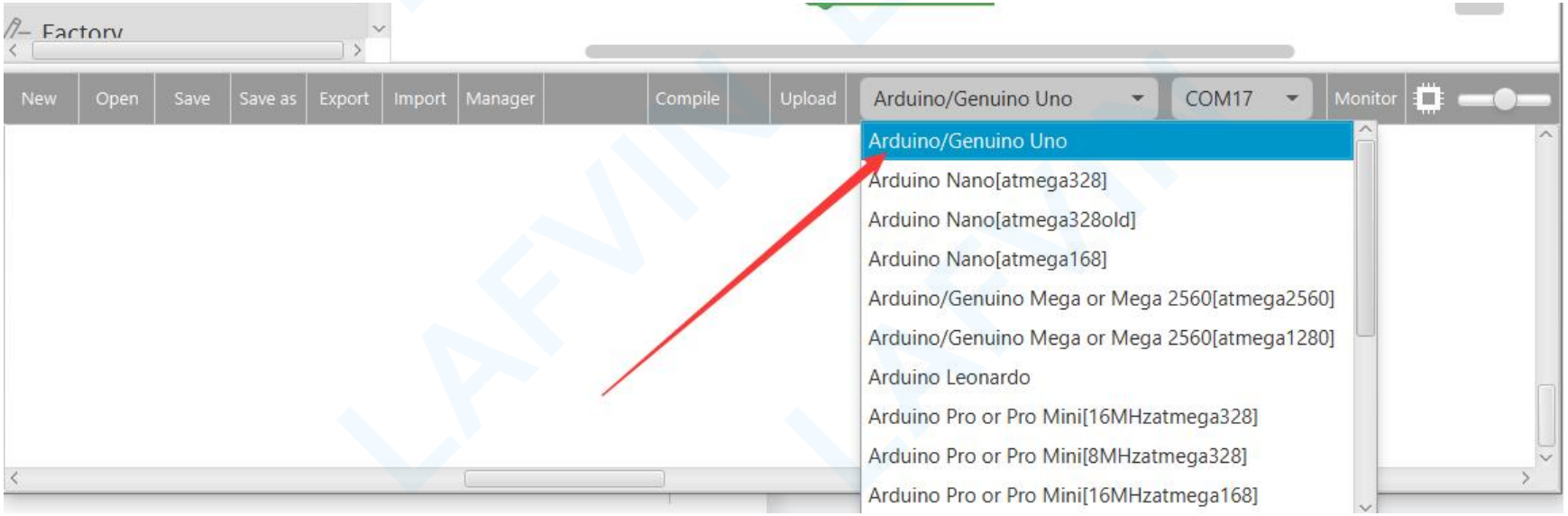



open xml/mix file
Mixly
Mixly_Code
Lesson_3
Search Lesson_3
Organise
New folder
Quick access
Dropbox
OneDrive
This PC
Network
Name
Date modified
Type
Size
Test_1_Motor_Speed_and_Direction_Cont...
13/08/2021 18:55
MIX File
2 KB
File name:
Open
Cancel

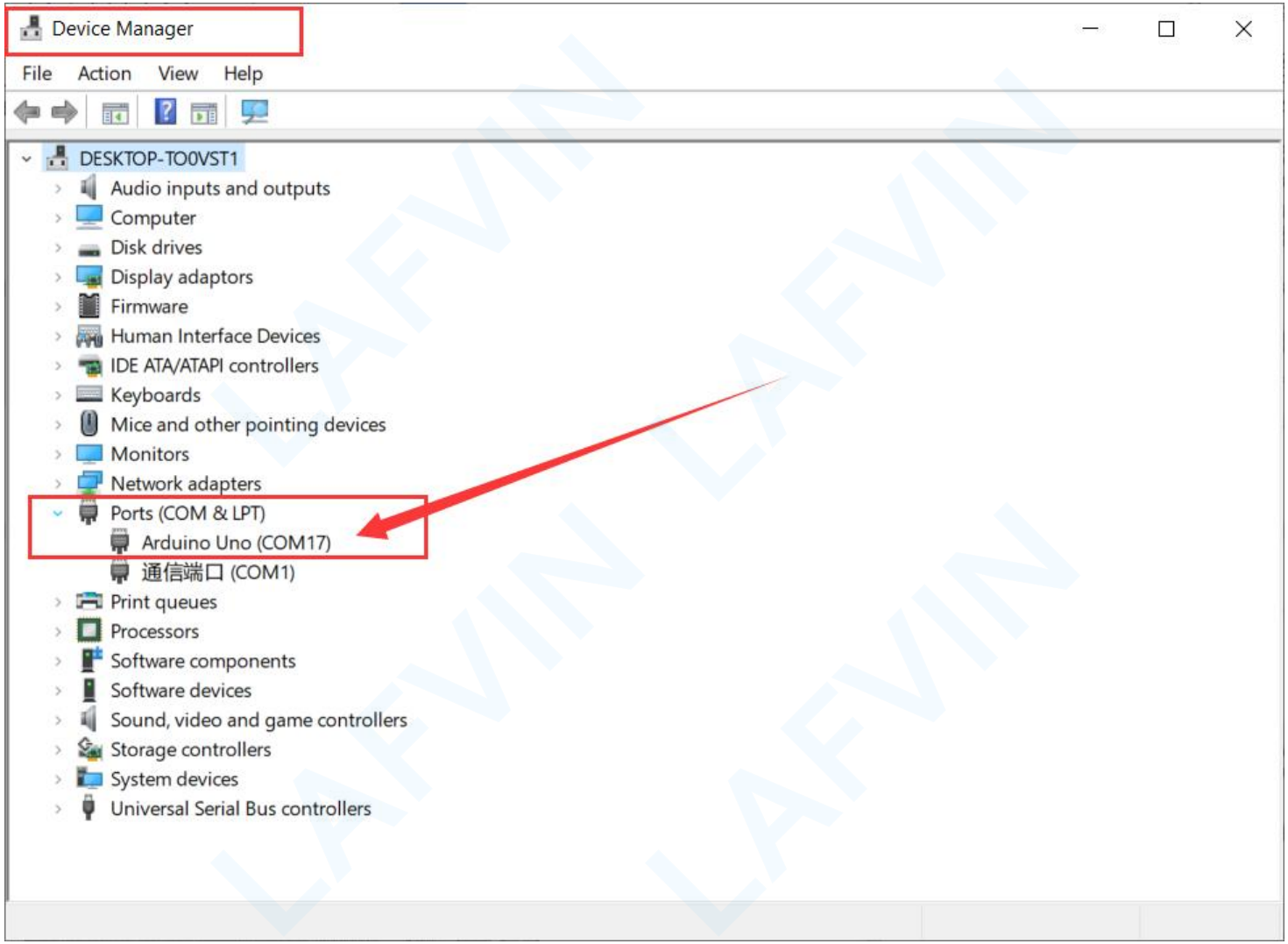

Mixly 1.2.0(E:\LAFVIN 4WD Smart Robot Car Kit V2.0\Mixly\Mixly_Code\Lesson_3\Test_1_Motor_Speed_and_Direction_Control.mix)
Blocks
Code
Copyright © Mixly Team http://mixly.org/
English
Chrome
Functions
SerialPort
Sensor
Actuator
Monitor
Communicate
Storage
Network
Factory
Tools
LAFVIN_Smart_Robot_Car_Kit
Move Forward
Speed(0~255)
50
PWM
Delay
millis
2000
Move Backward
Rotate Left
Rotate Right
New
Open
Save
Save as
Export
Import
Manager
Compile
Upload
Arduino/Genuino Uno
COM17
Monitor

Connect the Arduino UNO development board to the computer with a USB data cable, and turn on the power switch. Select the development board type as "Arduino/Genuino Uno" on the software and you will see a new connection serial port "COM17" appears.

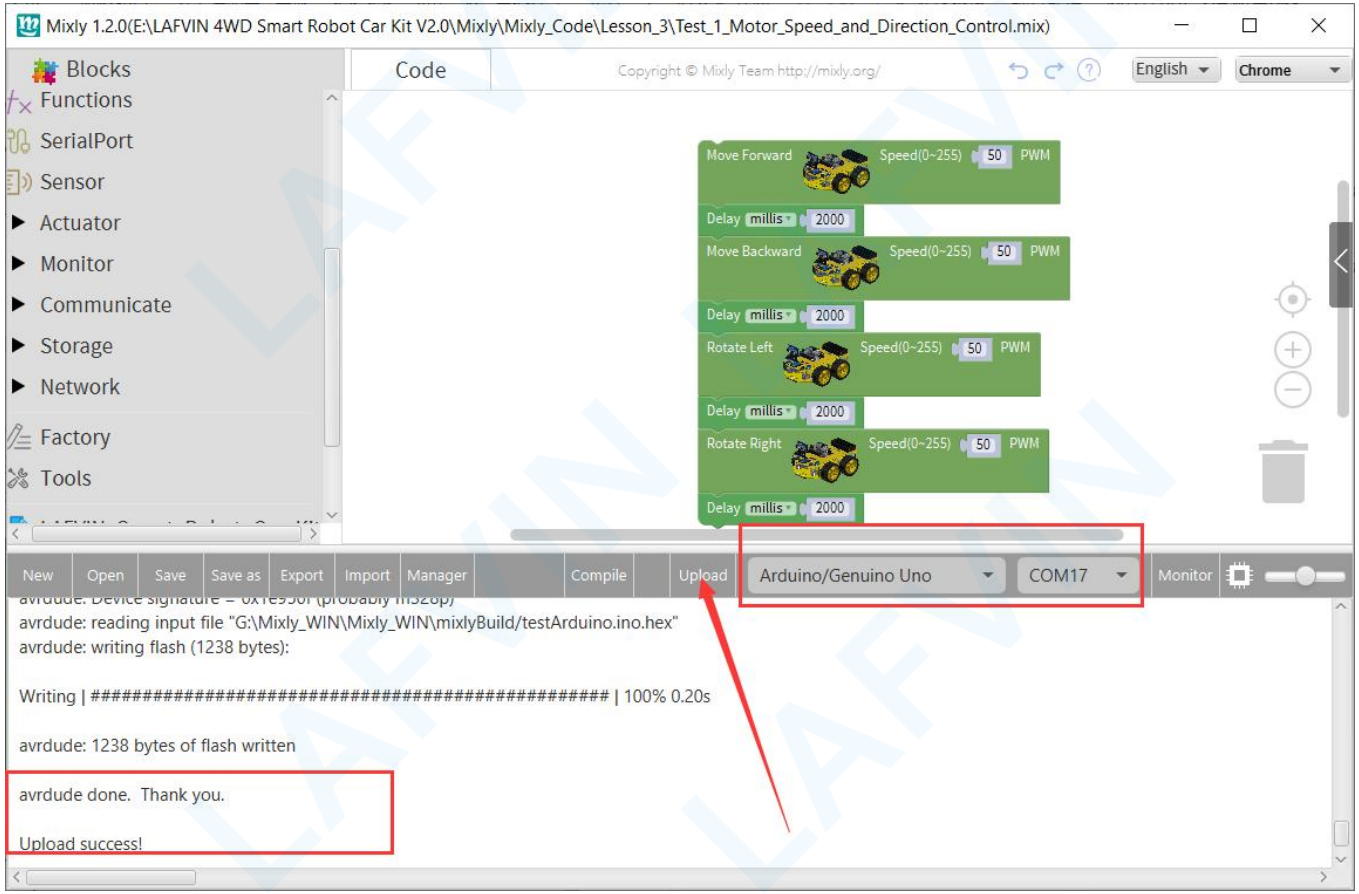
Tip: The serial number of your arduino uno on your PC is other. Every computer is different, check the correct serial number in the device manager of your computer.If no available COM interface is found, you need to install the arduino driver. How to Install Arduino Driver



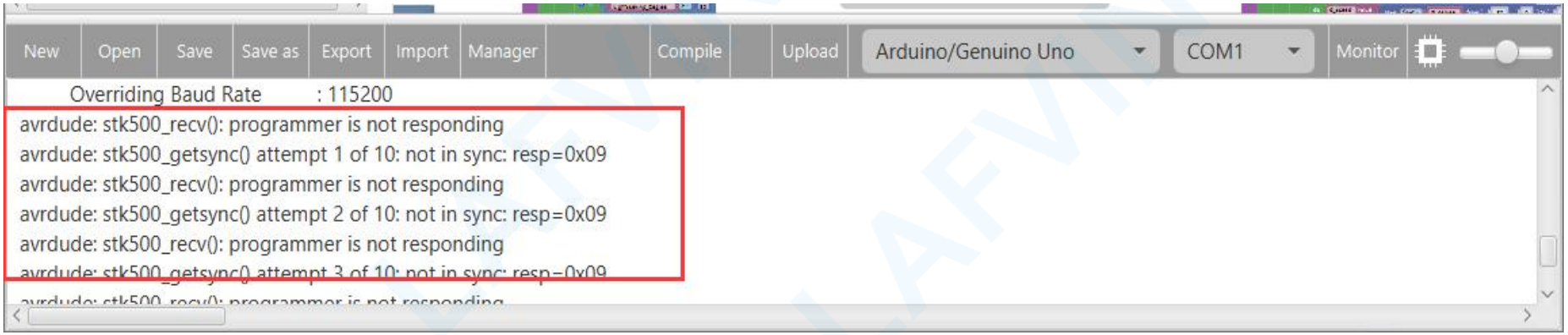

Factory
New
Open
Save
Save as
Export
Import
Manager
Compile
Upload
Arduino/Genuino Uno
COM17
Monitor
COM1
COM17
select Arduino UNO COM


Device Manager
File
Action
View
Help
DESKTOP-TO0VST1
Audio inputs and outputs
Computer
Disk drives
Display adaptors
Firmware
Human Interface Devices
IDE ATA/ATAPI controllers
Keyboards
Mice and other pointing devices
Monitors
Network adapters
Ports (COM & LPT)
Arduino Uno (COM17)
通信端口 (COM1)
Print queues
Processors
Software components
Software devices
Sound, video and game controllers
Storage controllers
System devices
Universal Serial Bus controllers

Click Upload and wait for the upload to complete.After the program upload is completed, smart car goes forward and back for 2s, turns left and right for 2s, and stops for 2s alternately.



In the process of uploading the code, if you report an error, there may be the following reasons:



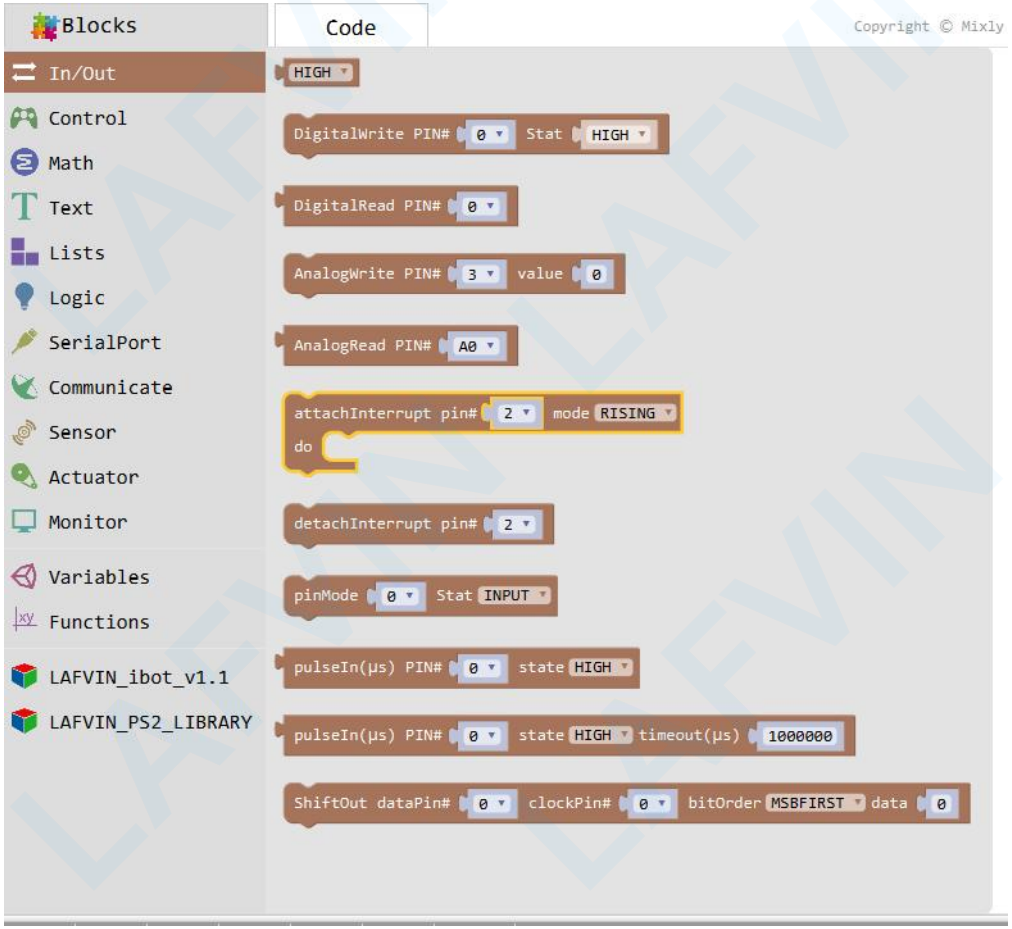
(1)The USB data cable interface is not firmly connected, or the drivers have not been installed (if necessary) or that the wrong serial port is selected.How to Install Arduino Driver

(2)It means that the communication serial port for uploading code is occupied. If you connect the Bluetooth module when uploading the code, you need to disconnect the Bluetooth module from the Arduino UNO expansion board, and then reinstall the Bluetooth module after uploading.

(3)Another possibility is that the USB interface of the Arduino UNO is insufficiently powered, and you need to turn on the external power switch of the expansion board.

The use of each statement block is described below

In/Out Block:

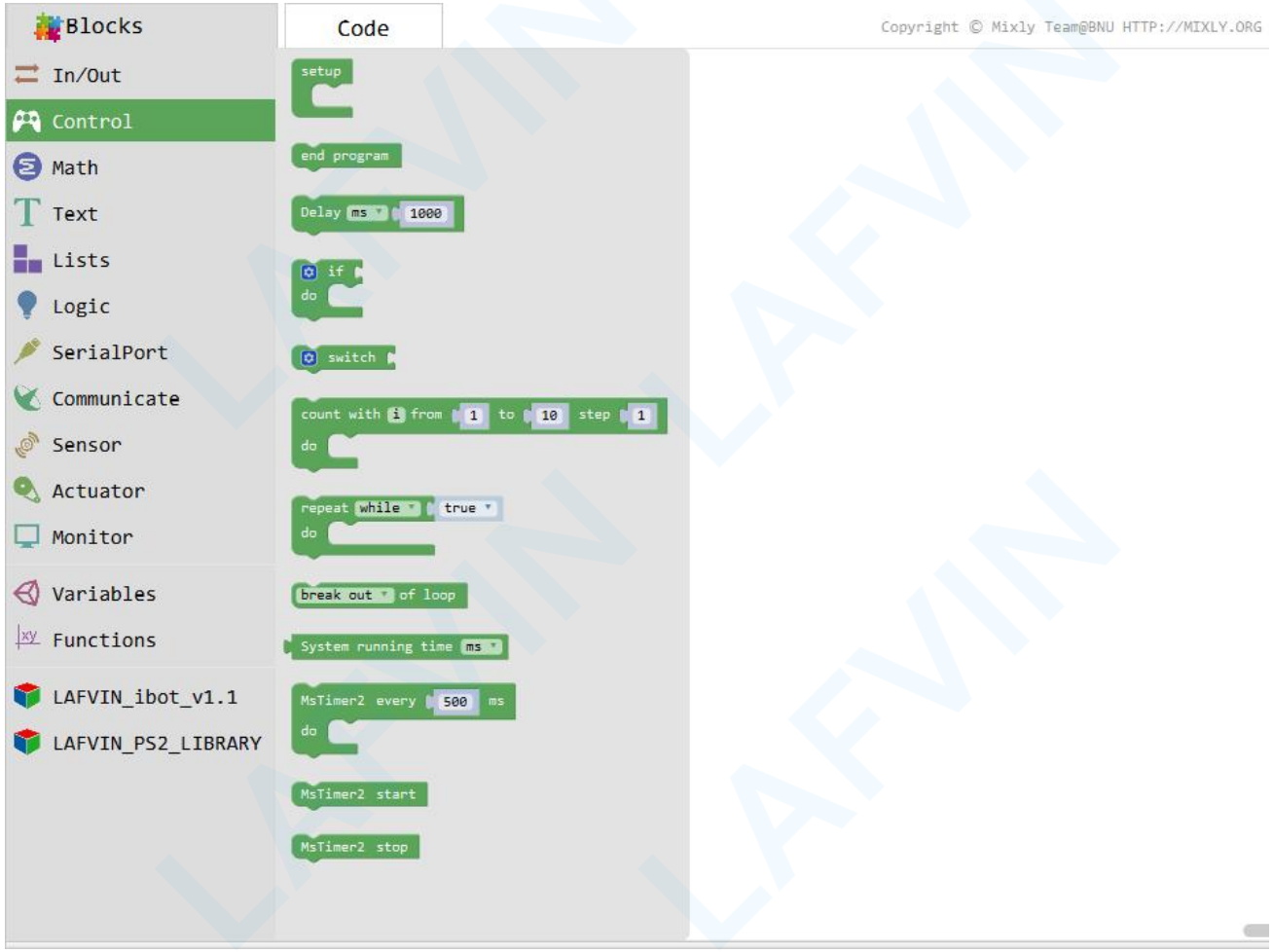


NO.	BLOCK ICON	DEFINITION
1		Returns HIGH or LOW voltage
2		Write digital value to a specific Port. Digital Output: set the HIGH or LOW output for IO pins
3		Returns a digital value of a specific Port. Digital IO Read Pin, generally used to read the HIGH or LOW level detected by Digital sensor

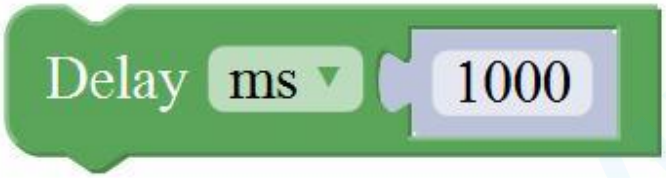
4		Write analog value between 2 and 255 to a specific Port. Analog Output: set the Analog value output by Analog IO pins (0~255).
5		Returns value between 0 and 1023 of a specific Port. Analog IO Read Pin, generally used to read the Analog value detected by Analog sensor.
6		External Interrupts function, with three trigger interrupt modes RISING, FALLING, CHANGE.
7		Detachs interrupt to a specific Port. Turn off the given interrupt function.

8		Set the IO pins as Output or Input state
9		Read the continuous time of HIGH or LOW pulse from IO pins (generally used for ultrasonic ranging)

Control Block:



NO.	BLOCK ICON	DEFINITION
1		Initialization (run only once)
2		End the program, means the program will stop running when use this block.
3		Delay function, click to select **ms** or **us** (pause the program for the amount of time (in milliseconds)

		specified as parameter. There are 1000 milliseconds in a second.)
4		**if_do** function (first evaluate a value be true or false, if a value is true, then do some statement. You can click the blue gear icon to select the **else if** block or **else** block.)

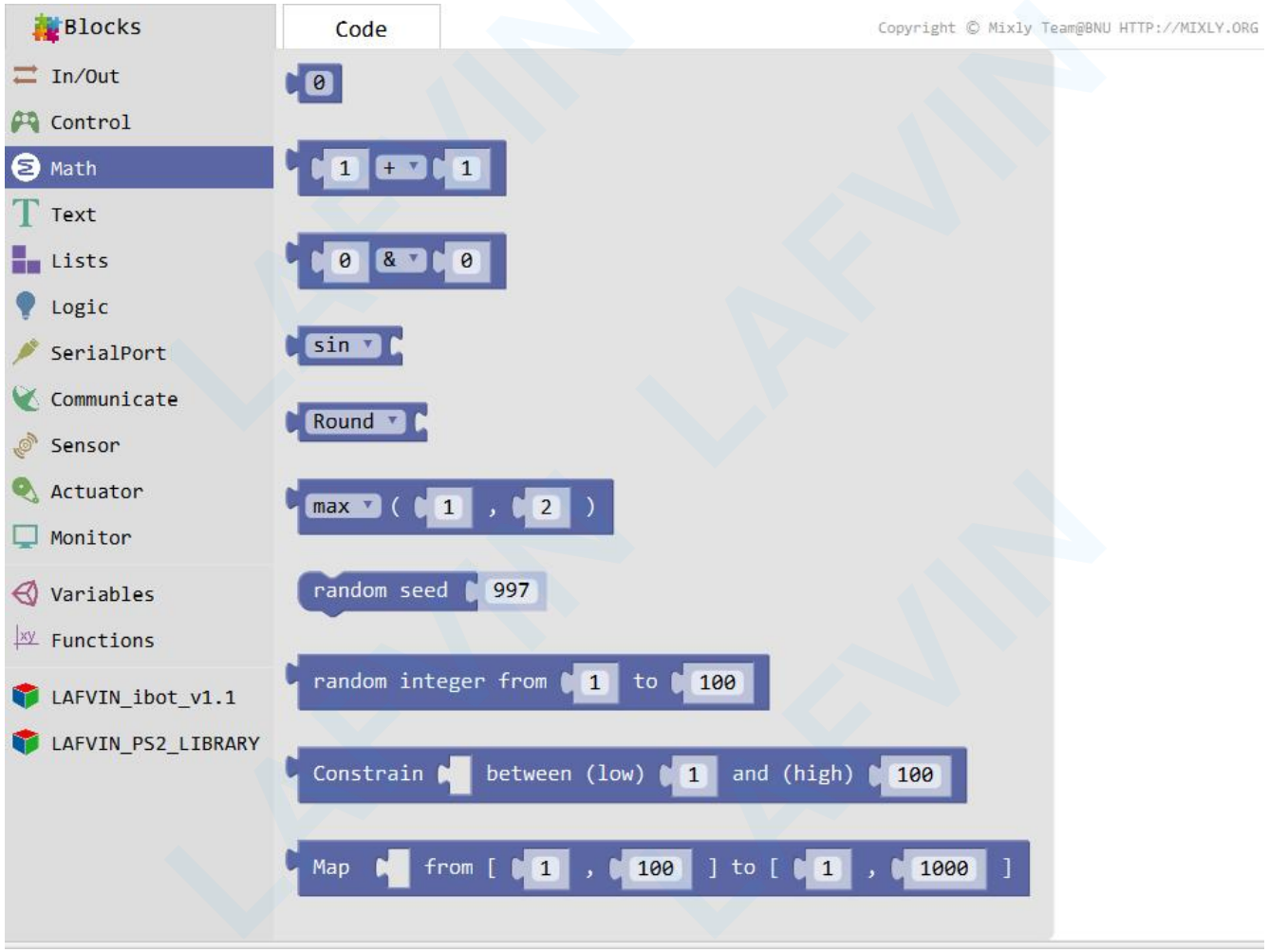
5		**switch** function. You can click the blue gear icon to select the **case** block or **default** block. (used to evaluate several programs then execute the corresponding function matched with program.)
6		Equal to **for** statement.

7		A **while** loop statement.
8		**break** function, used to exit from the containing loop.

9		**millis()** function, returns the system running time since the program started. (The unit can be **ms** (milliseconds) or **μs**（ m icrosecond)).
10		Timer interrupt function, that is, set a trigger interrupt for the amount of time (in milliseconds) specified as parameter.

11		Timer interrupt start block
12		Timer interrupt stop block

Math Block:

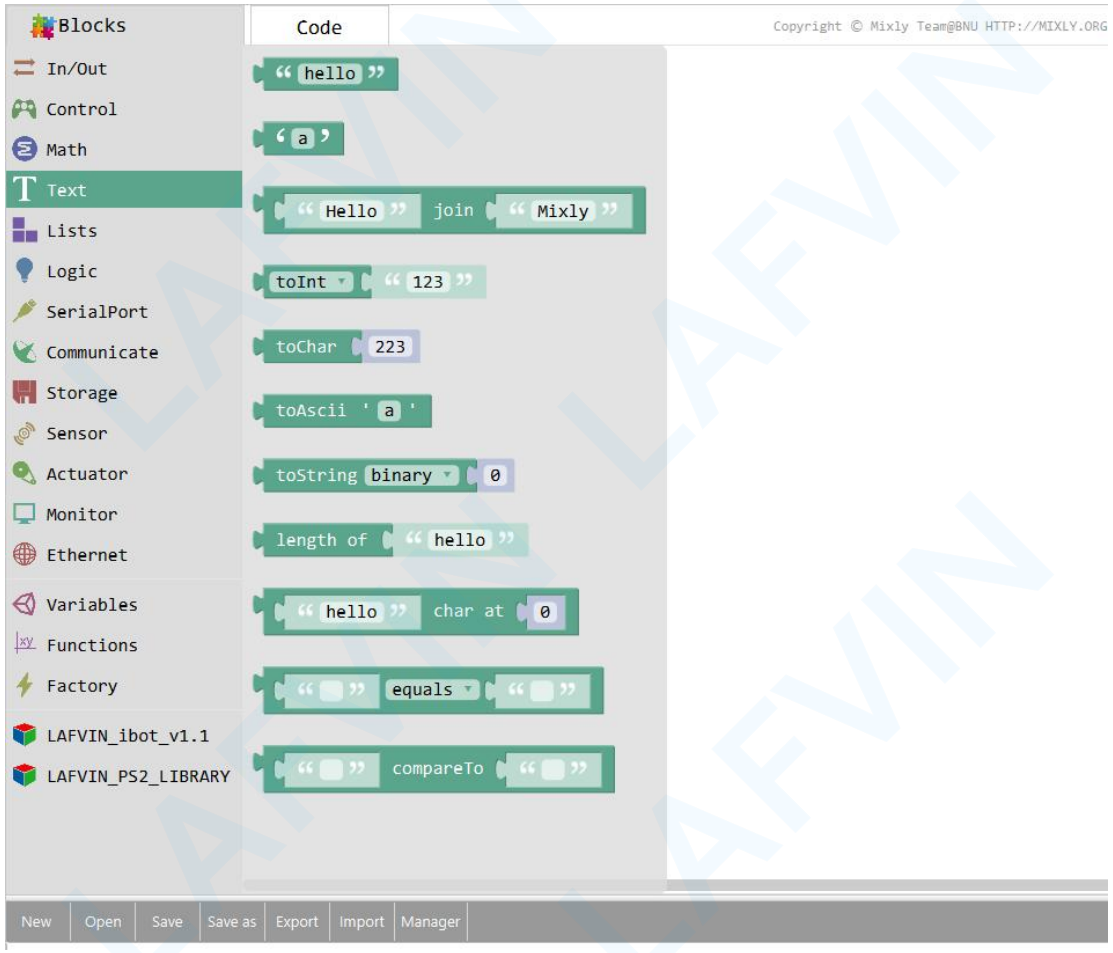

Blocks
Code
Copyright © Mixly Team@BNU HTTP://MIXLY.ORG
In/Out
Control
Math
Text
Lists
Logic
SerialPort
Communicate
Sensor
Actuator
Monitor
Variables
Functions
LAFVIN_ibot_v1.1
LAFVIN_PS2_LIBRARY
0
1 + 1
0 & 0
sin
Round
max (1 , 2)
random seed 997
random integer from 1 to 100
Constrain between (low) 1 and (high) 100
Map from [1 , 100] to [1 , 1000]

NO.	BLOCK ICON	DEFINITION
1		A number
2		Click to select the Arithmetic Operators: + (addition) ; − (subtraction) ; x (Multiplication); ÷ (division); % (remainder); ^ (bitwise xor)
3		Click to select the & (bitwise end); I (bitwise or); << (bitshift left); >> (bitshift right)

4		Click to select the **sin; cos; tan; asin; acos; atan; ln; log10; e^; 10^; ++ (increment) ; --(decrement)**
5		Click to select the **Round; Ceil; Floor; abs; sq; sqrt** **Round:** Returns the integer part a number using around. **Ceil:** Returns the integer part a number using ceil. **Floor:** Returns the integer part a number using floor. **abs:** Return the absolute value of a number. **sq:**

		Return the square of a number. **sqrt:** Return the square root of a number.
6		If select the max, returns the larger number; if select the min, returns the smaller number.

Text Block:

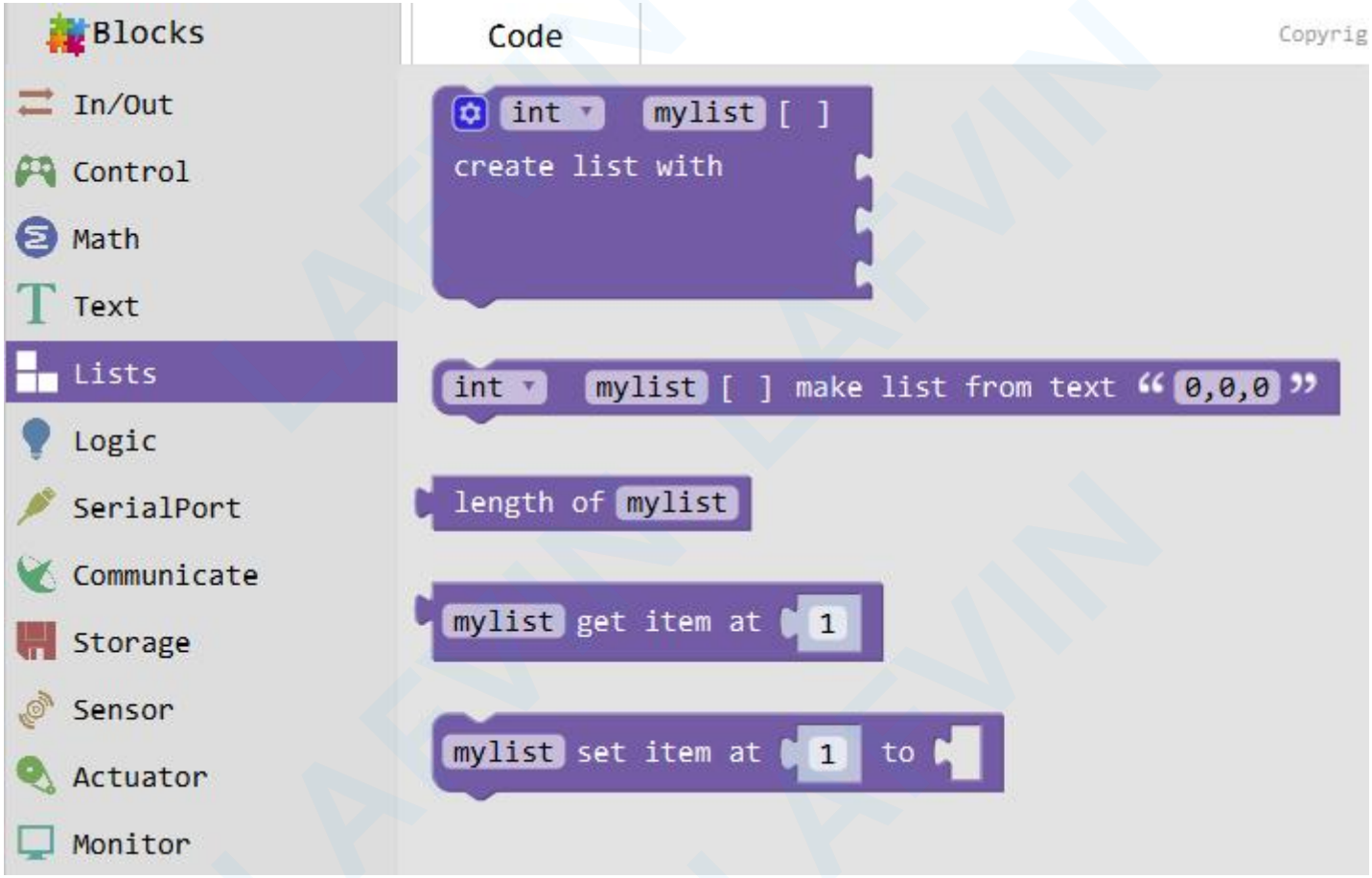

Blocks
Code
Copyright © Mixly Team@BNU HTTP://MIXLY.ORG
In/Out
Control
Math
Text
Lists
Logic
SerialPort
Communicate
Storage
Sensor
Actuator
Monitor
Ethernet
Variables
Functions
Factory
LAFVIN_ibot_v1.1
LAFVIN_PS2_LIBRARY
“ hello ”
‘ a ’
“ Hello ” join “ Mixly ”
toInt “ 123 ”
toChar 223
toAscii ' a '
toString binary 0
length of “ hello ”
“ hello ” char at 0
“ ” equals “ ”
“ ” compareTo “ ”
New
Open
Save
Save as
Export
Import
Manager

NO.	BLOCK ICON	DEFINITION
1		character string: a letter, word, or line of text.
2		A character
3		Creates a piece of text by joining together two piece of text. (Here Hello join Mixly equals HelloMixly)
4		Converts a string into an integer or an float.

5		Returns the char corresponding to an ASCII code (Decimal number 97 corresponding to a)
6		Returns the ASCII code corresponding to a char.
7		Converts a number into a string.
8		Calculates the length of a string
9		Output the char of a string (the char at 0 of hello is h)

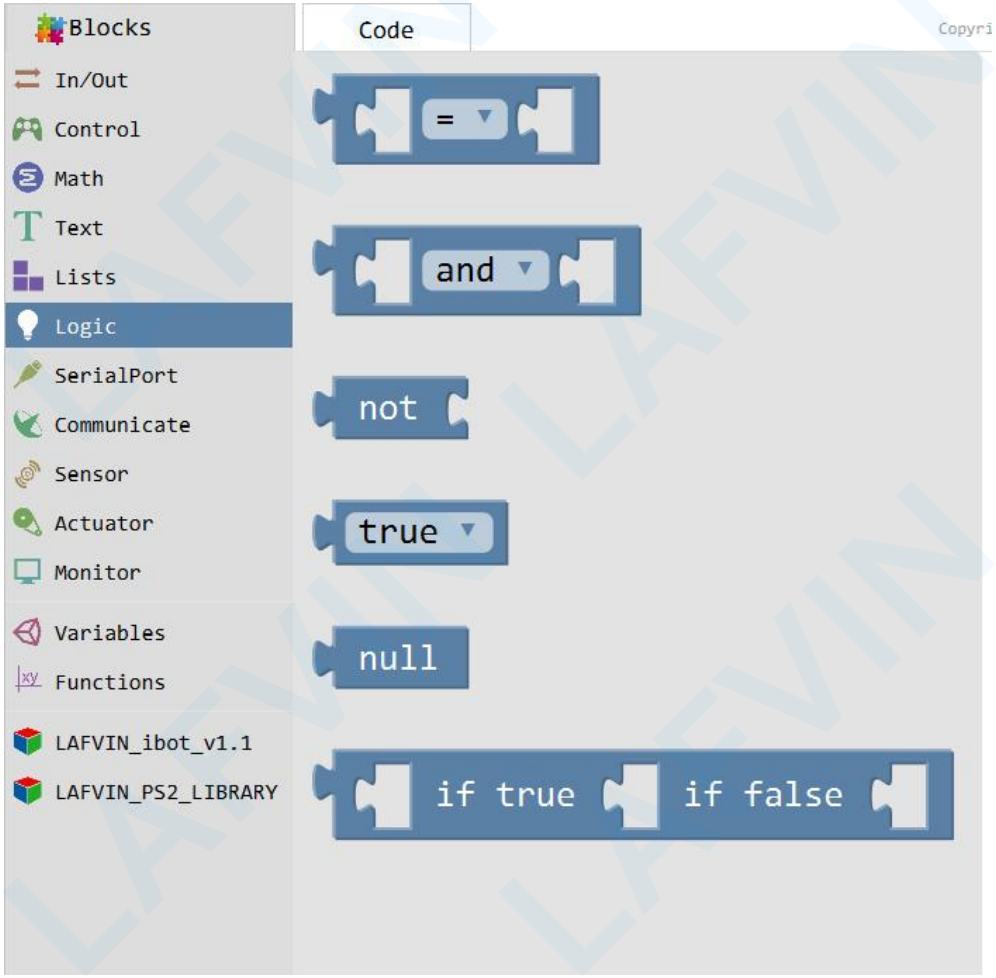
		
10		The first string equals or startsWith or endsWith the second string, returns 1, otherwise returns 0. (if equals, both strings are abc, returns 1.)
11		Returns a decimal value of the first string subtracts the second string.

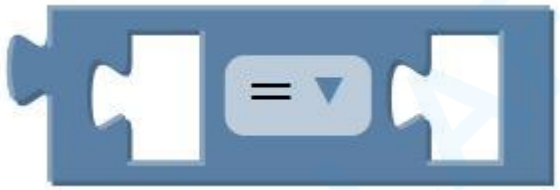
List Block:

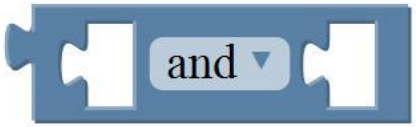

Blocks
Code
Copyrig
In/Out
Control
Math
Text
Lists
Logic
SerialPort
Communicate
Storage
Sensor
Actuator
Monitor
int mylist [ ]
create list with
int mylist [ ] make list from text “ 0,0,0 ”
length of mylist
mylist get item at 1
mylist set item at 1 to

NO.	BLOCK ICON	DEFINITION
1		Create a list with any number of items
2		Creats a list from a text. (int mylist [ ]={0,0,0};)
3		Returns the length of a list
4		Returns the value of at the specified position in a list.
5		Sets the value of at the specified position in a list. Set the first item in mylist to another item.

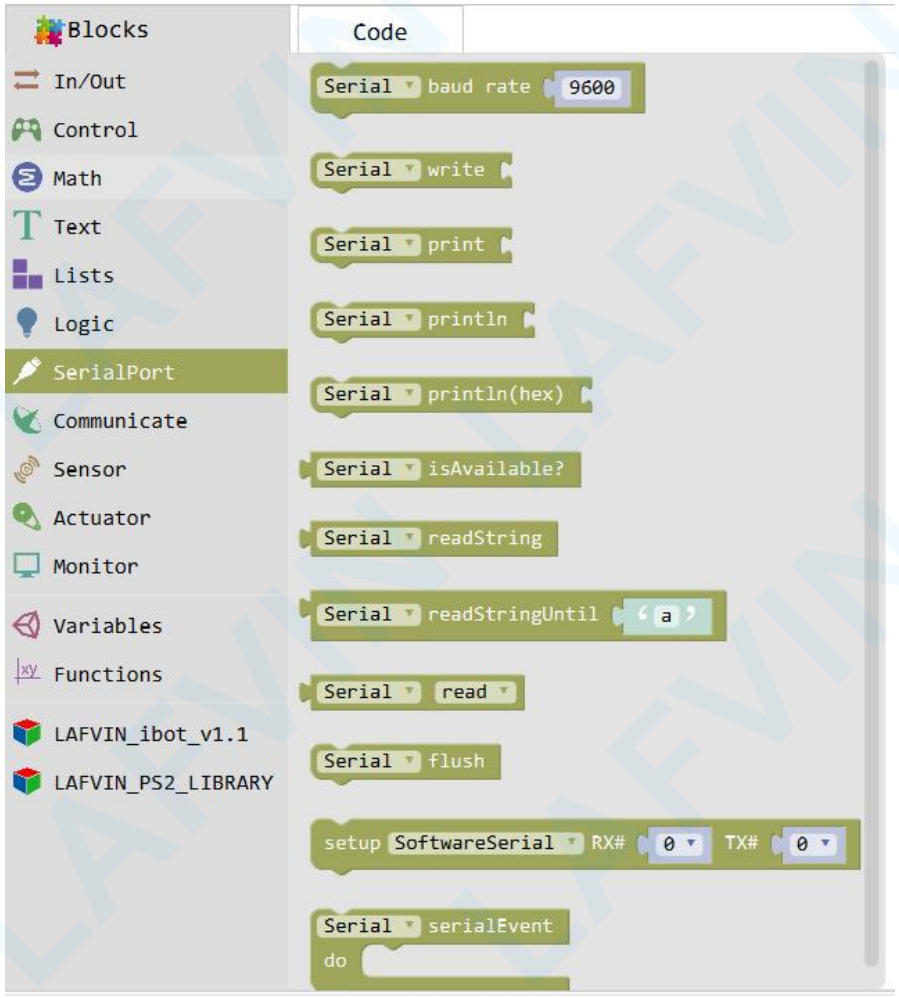
Logic Block:


Blocks
In/Out
Control
Math
Text
Lists
Logic
SerialPort
Communicate
Sensor
Actuator
Monitor
Variables
Functions
LAFVIN_ibot_v1.1
LAFVIN_PS2_LIBRARY
Code
=
and
not
true
null
if true
if false

NO.	BLOCK ICON	DEFINITION
1		**logic comparision** =: Return true if both inputs equal each other. ≠: Return true if both inputs are not equal to each other. <: Return true if the first input is smaller than the second input. ≤: Return true if the first input is smaller than or equal to the second input. >: Return true if the first input is greater than the second input. ≥: Return true if the first input is greater than or equal to the second input.

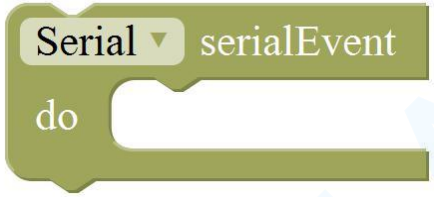
2		**and:** Return true if both inputs are true; **or:** Return true if at least one of the inputs is true
3		Returns true if the input is false. Returns false if the input is true.
4		Returns either true or false.
5		Returns null
6		If the first number is true, the second number is returned, otherwise the third number.

SerialPort Block:

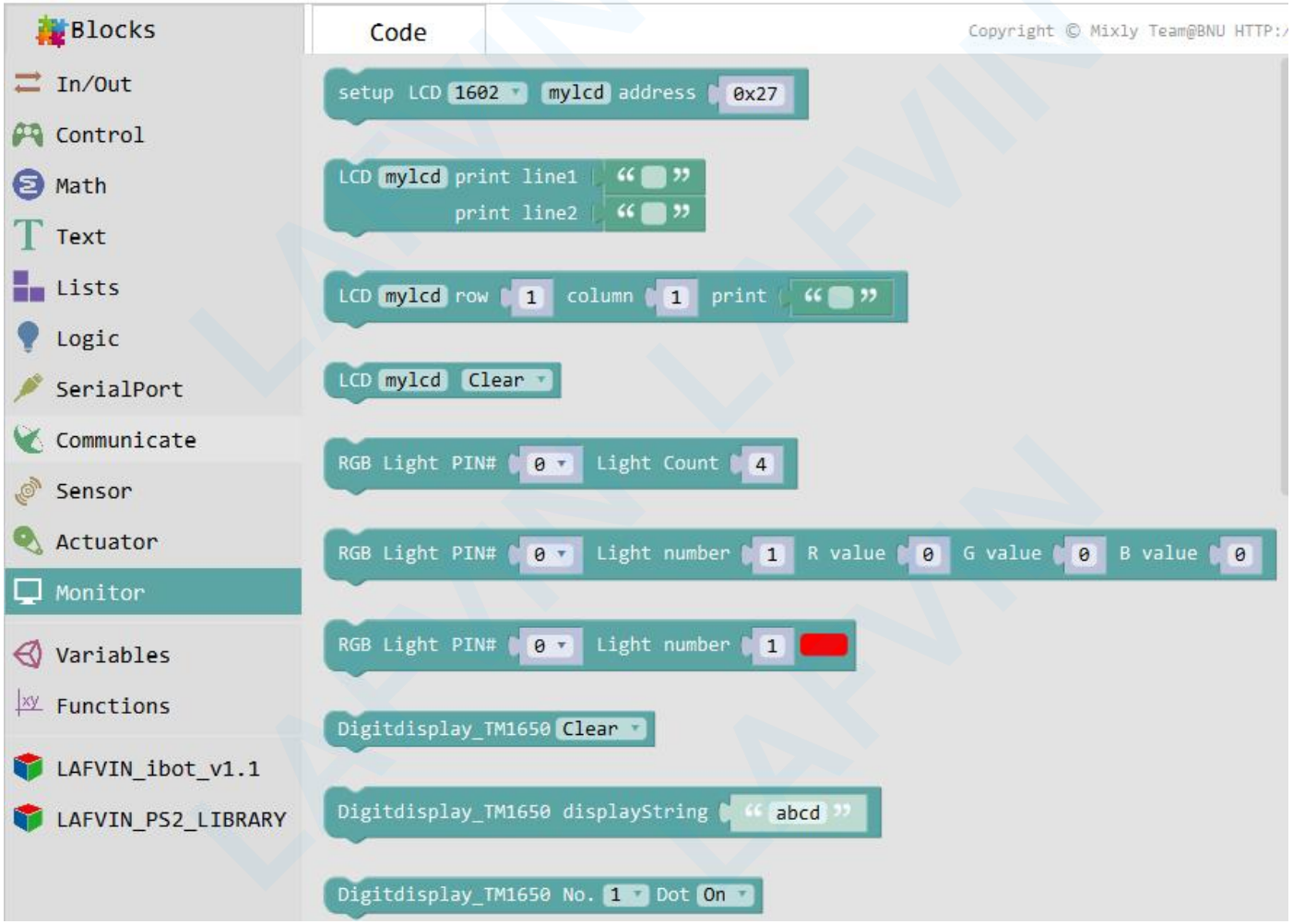


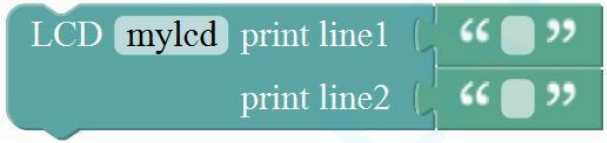
NO.	BLOCK ICON	DEFINITION
1		Set the serial buad rate to 9600
2		Write the specified number, text or other value.
3		Print the specified number, text or other value on monitor.
4		Print the specified number, text or other value on newline of monitor.

5		Print the specified number in hexademical format on newline of monitor.
6		If the serial port is available, it returns true, otherwise returns false. (generally used in Bluetooth communication)
7		Returns a string in serial port
8		A string read from serial port to a string variable, pause until read the specified character.

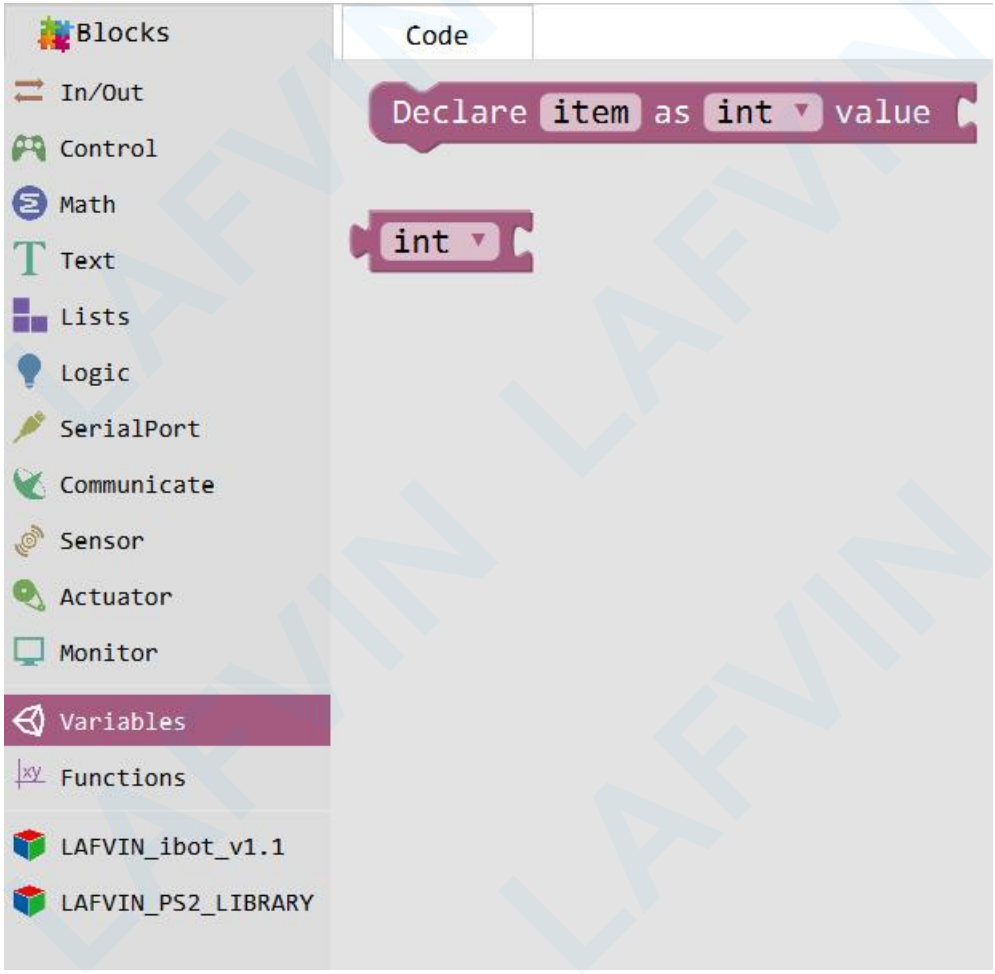
9		Read the serial data by byte (generally used to read the value sent from Bluetooth) (delete the data has been read)
10		Wait for the output data completed
11		Set the software serial port (call this function if need to use several serial ports)
12		Event function trigger by serial port data, that is, serial port is ready to call this function. (equal to an interrupt function)

Monitor Block:



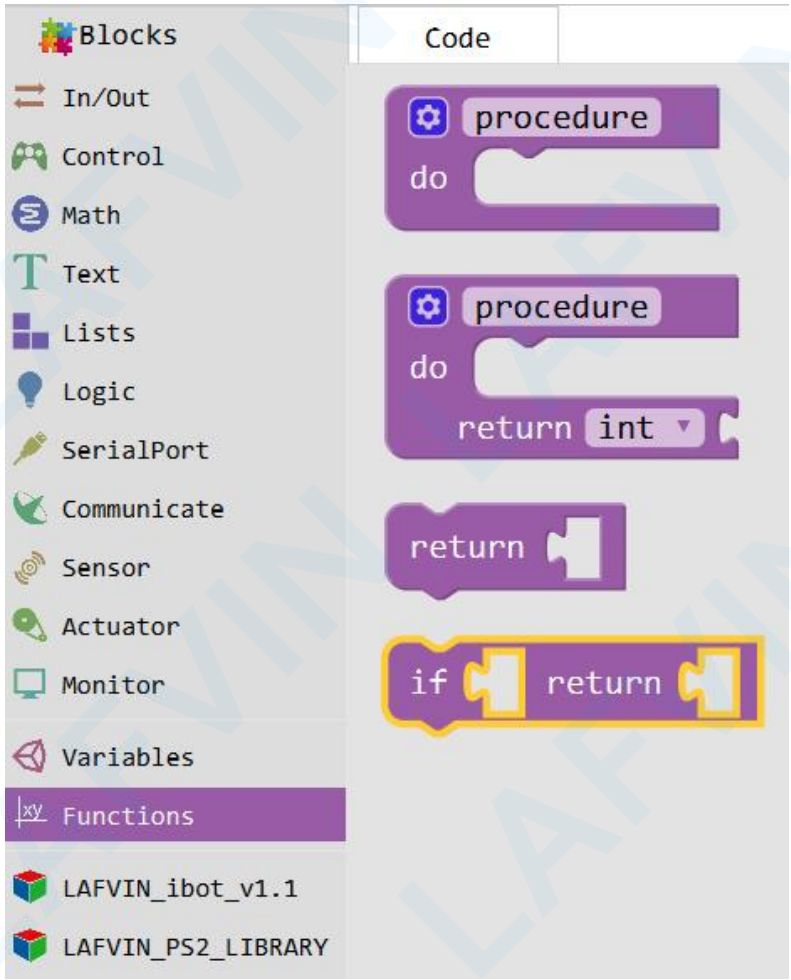
NO.	BLOCK ICON	DEFINITION
1		Set the IIC LCD1602 address
2		Input the value on LCD line 1 and line 2 from left to right.
3		Set the row and column of LCD to print the char
4		Clear the LCD screen

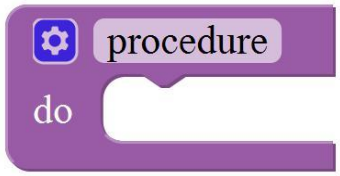
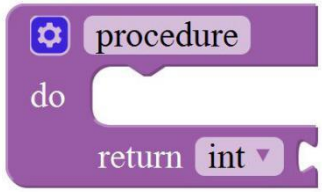
Variables Block



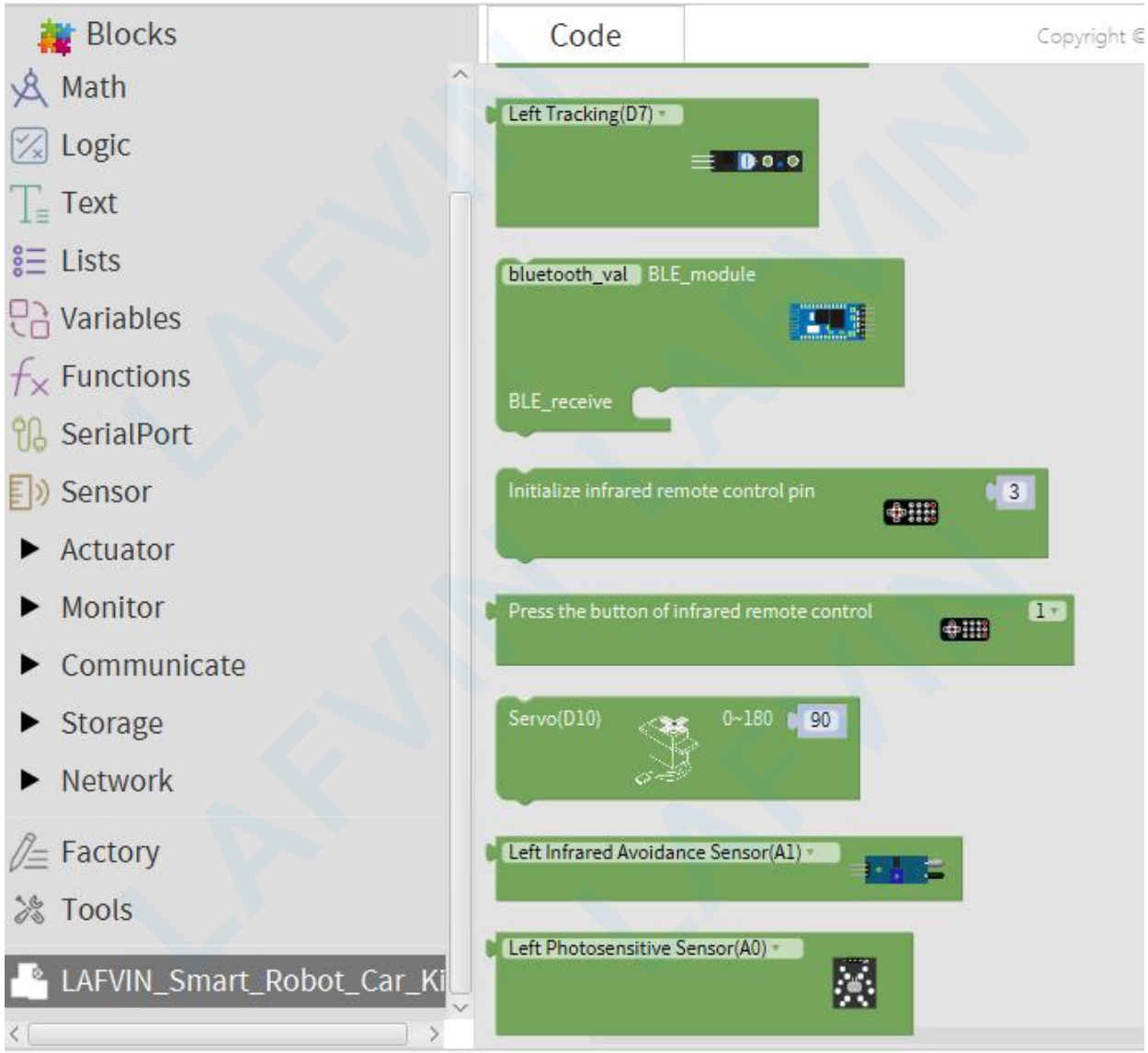
NO.	BLOCK ICON	DEFINITION
1		Define an integer variable whose name is item
2		Mandatory type conversion of constants or variables

Functions Block:



NO.	BLOCK ICON	DEFINITION
1		Creates a function with no output. Click the blue icon to set the procedure parameter. (no return value)
2		Creates a function with an output. Click the blue icon to set the procedure parameter. (with return value and can set the data types)
3		If a value is true, then return a second value.

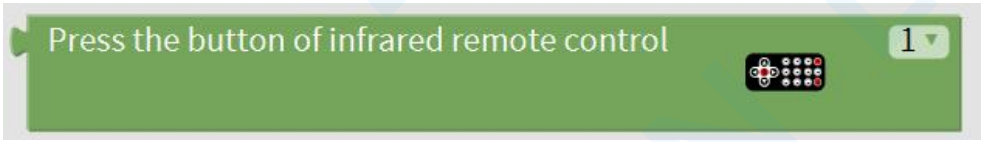
Smart_Robot_Tank Block:


Blocks
Math
Logic
Text
Lists
Variables
Functions
SerialPort
Sensor
Actuator
Monitor
Communicate
Storage
Network
Factory
Tools
LAFVIN_Smart_Robot_Car_Ki
Code
Copyright ©
Left Tracking(D7)
bluetooth_val BLE_module
BLE_receive
Initialize infrared remote control pin
3
Press the button of infrared remote control
1
Servo(D10)
0~180
90
Left Infrared Avoidance Sensor(A1)
Left Photosensitive Sensor(A0)

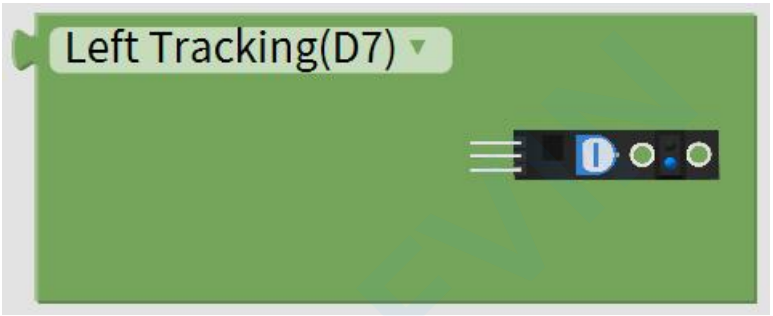
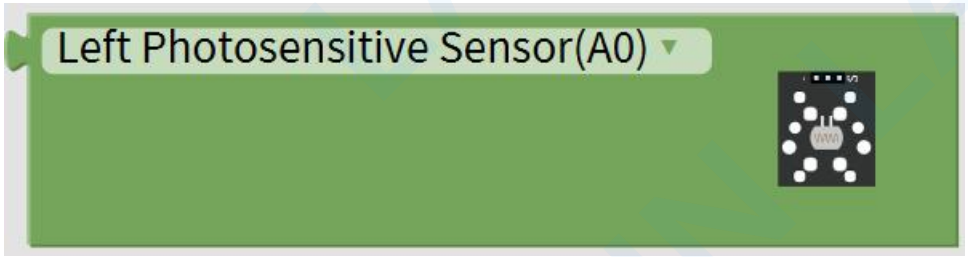
NO.	BLOCK ICON	DEFINITION
1		Control the robot car to move forward, and the digital PWM can be filled to control the speed. The range of the PWM value is 0~255. When the PWM equals 255, the tank moves at the maximum speed, and the PWM equals 0 means stop.
2		Control the robot car to move backward, and the digital PWM can be filled to control the speed. The range of the PWM value is 0~255. When the PWM equals 255, the tank moves at the maximum speed, and the PWM equals 0 means stop.

3		Control the robot car to rotate counterclockwise (rotate to the left), the digital PWM can be filled to control the speed, the range of PWM value is 0~255, when the PWM is equal to 255, the tank moves at the maximum speed, and the PWM equals 0 means stop
4		The left wheel and the right wheel move forward at the same time, and the speed of each wheel can be set separately. Since the forward speed of the two wheels is different, the turning can be realized.
		Control the tank to rotate clockwise (rotate

5		to the right), the digital PWM can be filled to control the speed, the range of PWM value is 0~255, when the PWM is equal to 255, the tank moves at the maximum speed, and the PWM equals 0 means stop
6		The left wheel and the right wheel move backward at the same time, and the speed of each wheel can be set separately. Since the two wheels have different backward speeds, turning can be realized.
7		Control tank stop

8		Control the rotation angle of the servo motor, the angle range of rotation is 0~180 degrees
9		This program block is initialized and defines the IO data interface connected to the infrared receiver
		When the key command of the remote control is received and the value set by the program block is the same, the program block outputs logic ture.
10		Use the ultrasonic module to measure the distance, the program block returns the measured distance value, the unit is cm

11		The function of this program block is to receive the detection result of the infrared obstacle avoidance sensor and save the received result to the variable left_infrared_avoidance_sensor or right_infrared_avoidance_sensor
12		This program block is used to receive the information sent by the Bluetooth module, and store the received information in the variable bluetooth_val

13		This program block obtains the output result of the infrared line-following sensor. When the infrared sensor detects black, it outputs a high level "1", and when no black is detected, it outputs a low level "0".
14		This program block obtains the output result of the photosensitive sensor. The photosensitive sensor can detect the light intensity in the environment and convert it into a range of 0-1024. 0 represents the maximum light intensity, and 1024 represents the weakest light intensity.

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;

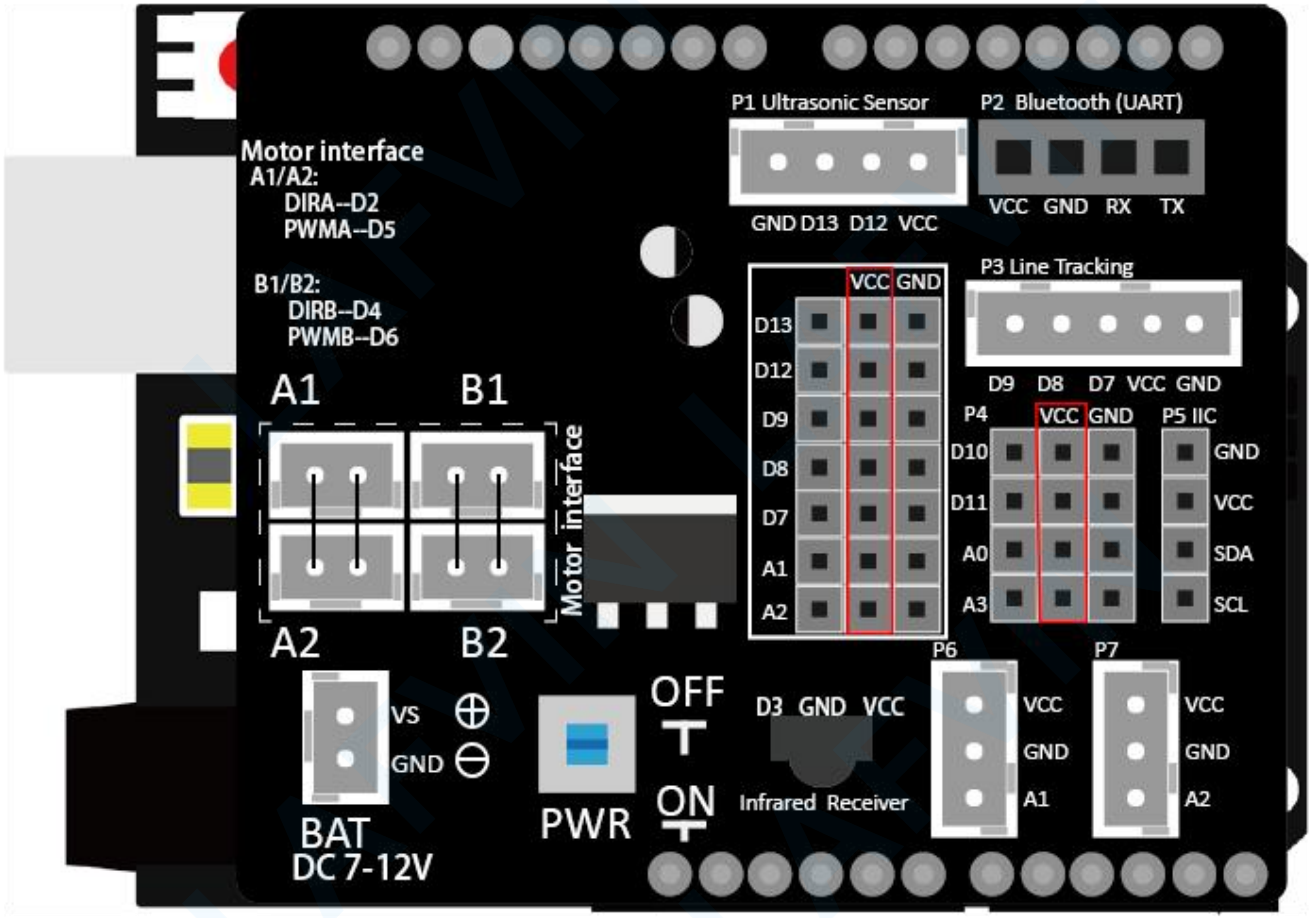
Specifications:

Recommended motor voltage (VMOT): 7.4– 13.5 V

• Logic voltage (VCC): 2.7 – 5.5 V

• Output current maximum: 3Aper channel

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


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

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.

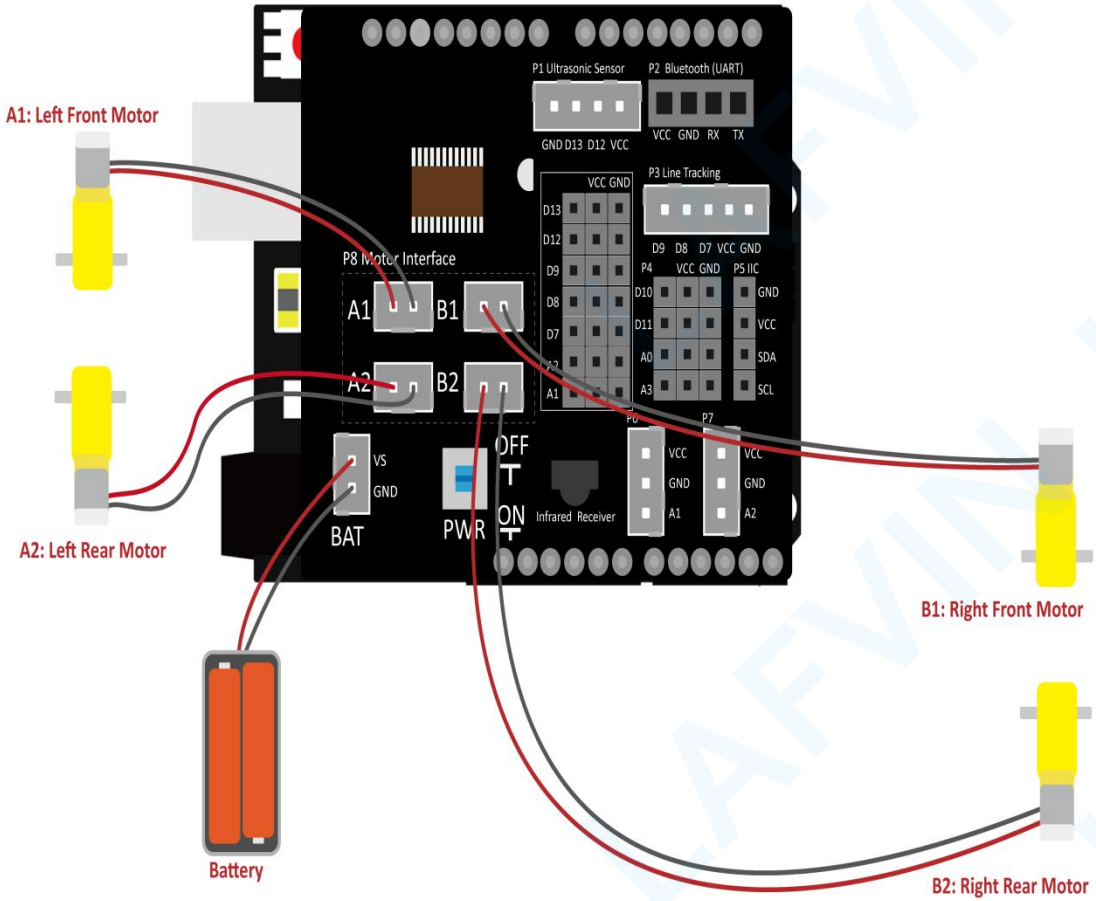
The D2 digital I/O port controls the direction of the motor of the port A, and the D5 digital I/O port outputs the PWM signal to control the speed of the motor of the port A.

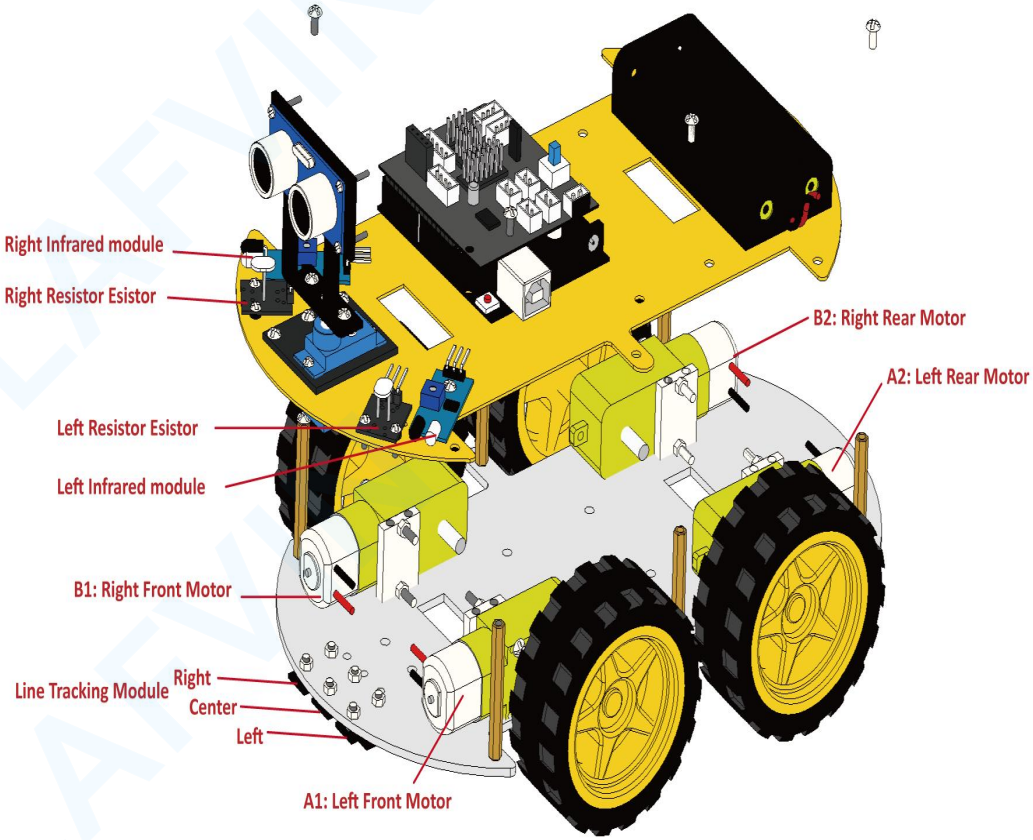
The D4 digital I/O port controls the direction of the motor of the interface B, and the D6 digital I/O port outputs the PWM signal to control the speed of the motor of the interface B.

The PWM value is in the range of 0-255. The greater the value, the faster the motors turn.

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





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

Mixly Code

Wire it up well as the above diagram. Okay, let's move on to write the test code.Open mixly software.Click "New" to add new project, then start your programming .

if you want to refer to the program we provide.open the reference code for this lesson "**Test_1_Motor_Speed_and_Direction_Control.mix**" in the reference files we provided.

Programming Thinking


```
 9   digitalWrite(2,HIGH);
10   analogWrite(5,50);
11   digitalWrite(4,LOW);
12   analogWrite(6,50);
```


```
digitalWrite(2,LOW);
analogWrite(5,50);
digitalWrite(4,HIGH);
analogWrite(6,50);
```

This block controls the direction of the robot car to forward or backward, 255 represents the maximum speed, and the range that can be modified is 0-255. When the speed is set to 0, it means stop.


```
digitalWrite(2,LOW);
analogWrite(5,90);
digitalWrite(4,LOW);
analogWrite(6,90);
```


```
digitalWrite(2,HIGH);
analogWrite(5,50);
digitalWrite(4,HIGH);
analogWrite(6,50);
```

The "Rotate Left" program block controls the robot car to make a circle to the left. Speed 90 PWM means that the car moves at a speed of 90. The minimum speed 0 PWM means stop, and the maximum speed ratio is 255 PWM.

Arduino Code

if you want to refer to the program we provide.Open the reference code for this lesson "Test_1_Motor_Speed_and_Direction_Control.ino" in the reference file we provided.

Programming Thinking

```
/*Move Forward*/
  digitalWrite(2,HIGH); //D2 digital I/O port controls the direction of the motor of interface A
  analogWrite(5,50); //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(6,50);//D6 digital I/O port outputs PWM signal to control the speed of interface B motor.
  delay(2000);//delay 2s
```

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)

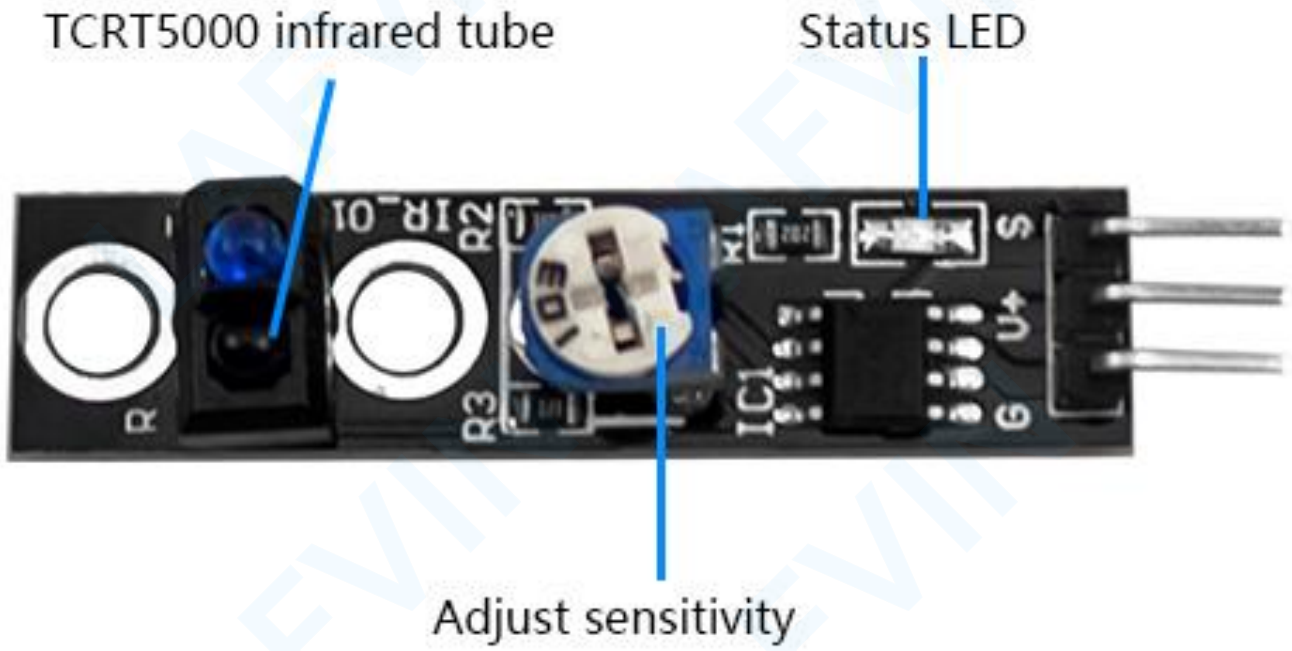
Interface: G(GND) V+(VCC) S(Signal)

Output Signal: Digital signal

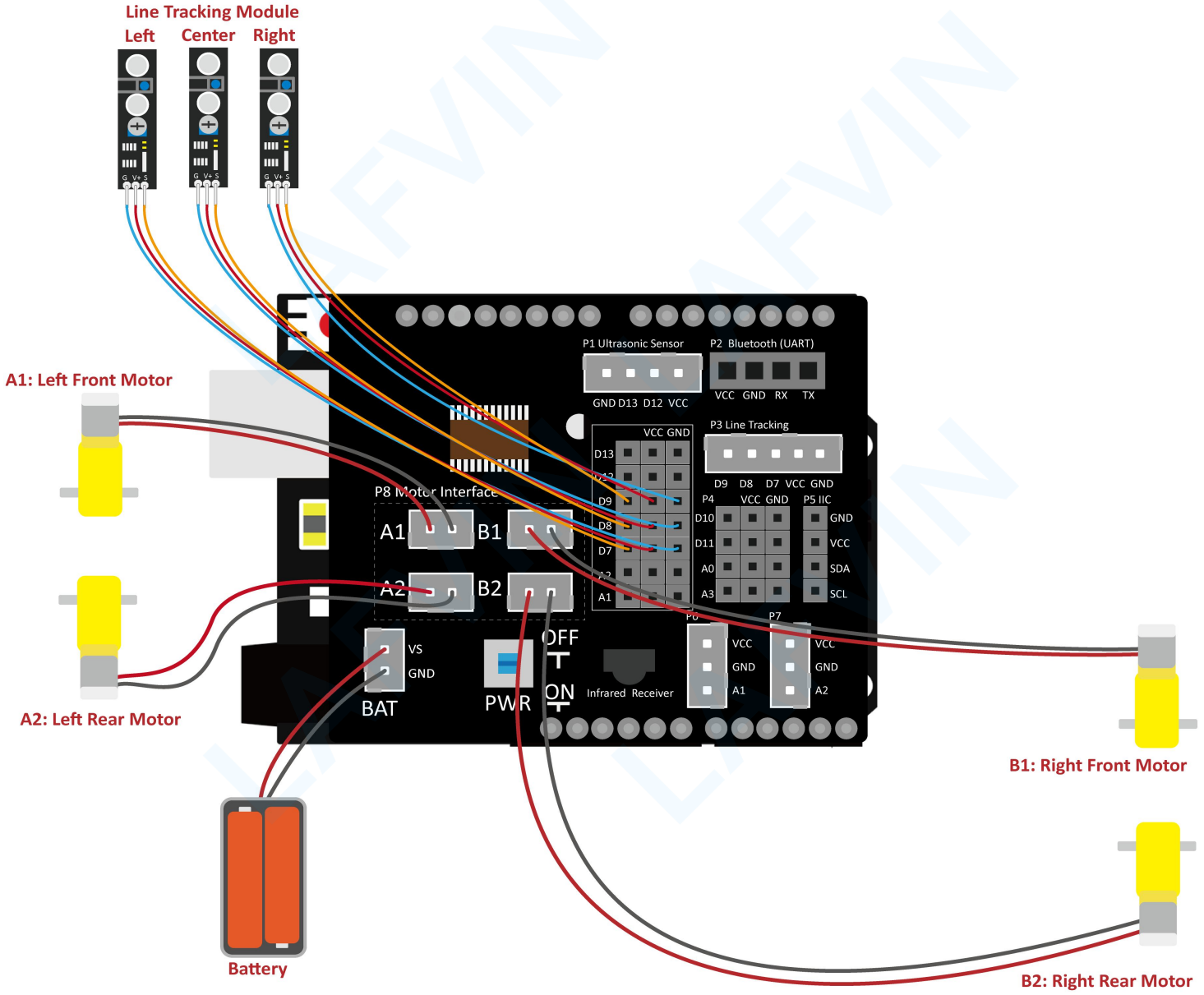
Detection Height: 0-3 cm

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

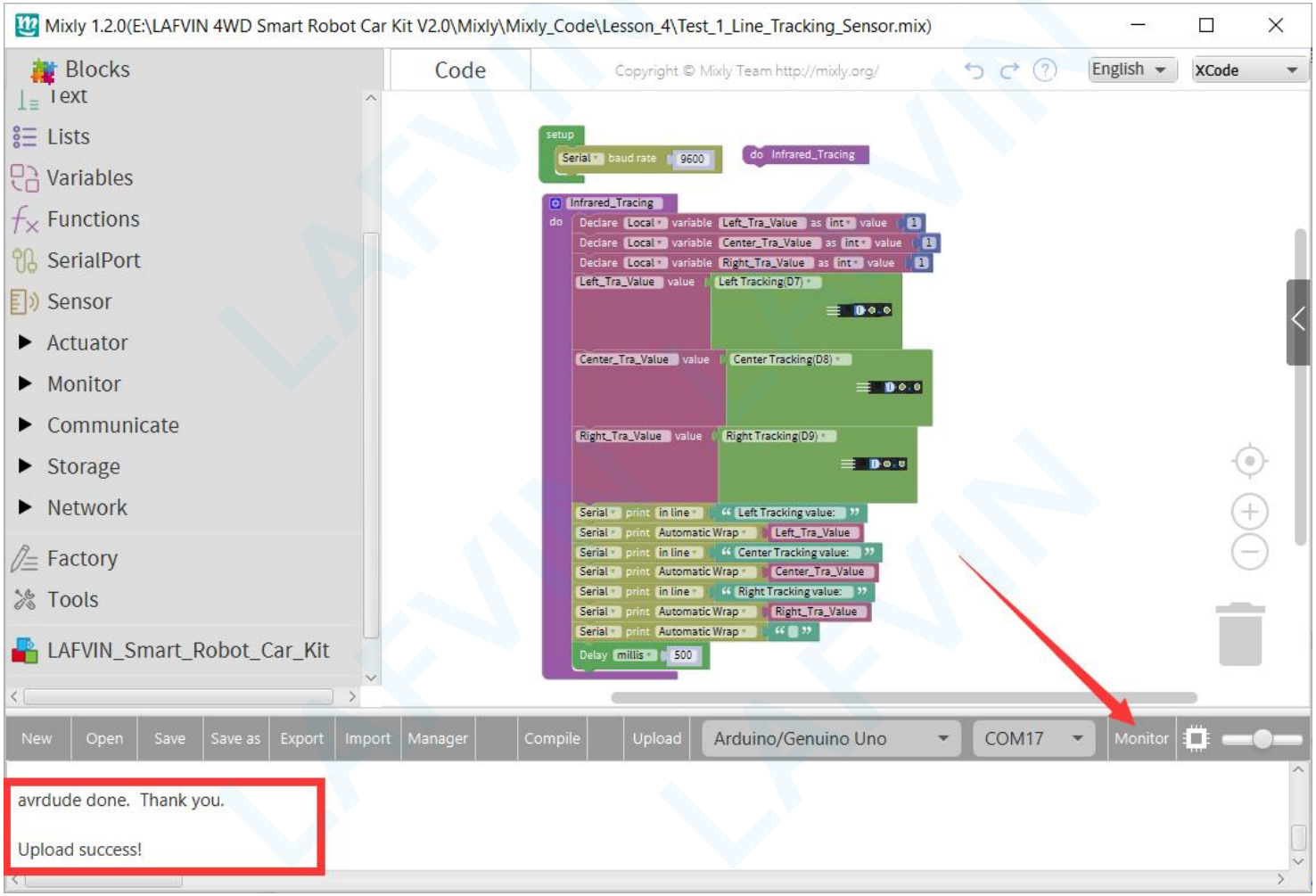
The main purpose of the test experiment is to read the return signal of the line tracking sensor and print it to the serial port monitor.When detects white paper, sensor’s signal pin outputs LOW (display 0),and status LED is on;When detects black, sensor’s signal pin outputs HIGH (display 1),and status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

Mixly Code

Wire it up well as the above diagram. Okay, let’s move on to write the test code.Open Mixly software.Click “New” to add new project, then start your programming .

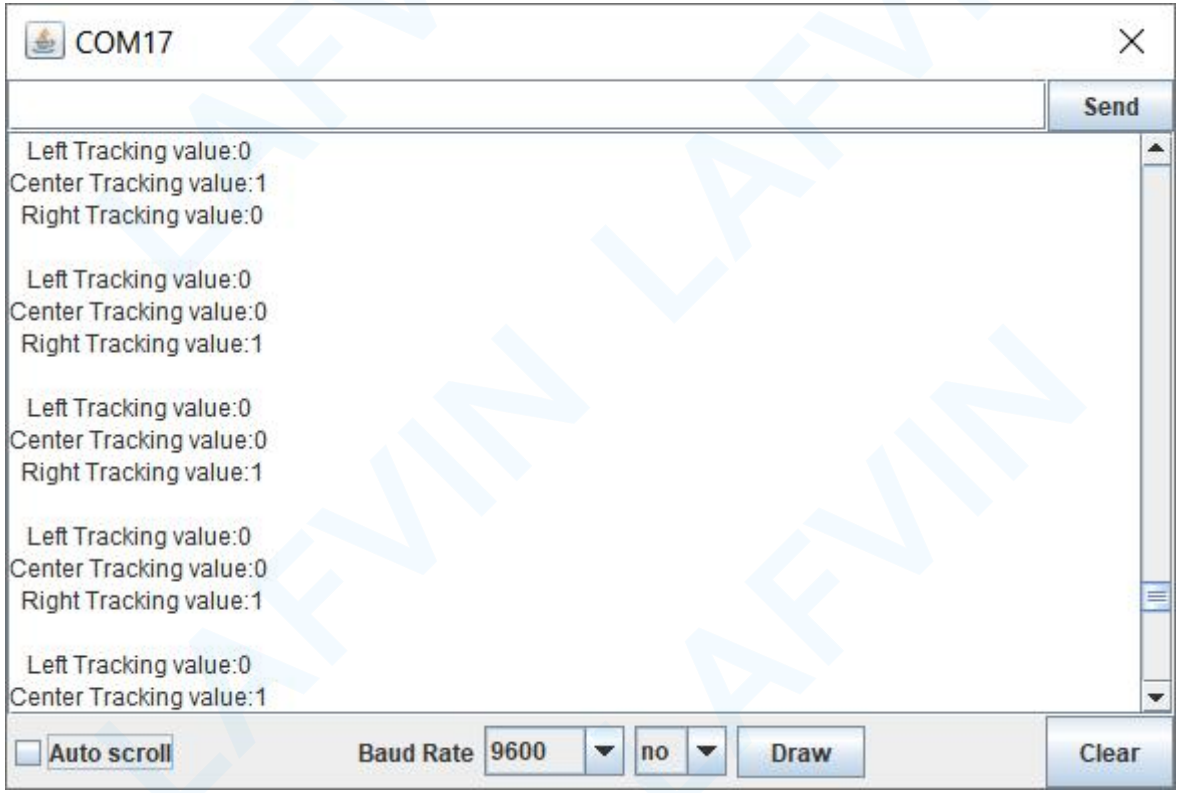
if you want to refer to the program we provide.open the reference code for this lesson "**Test_1_Line_Tracking_Sensor.mix**" in the reference file we provided on CD.

After uploading the program, open the serial monitor to view the signal returned by the infrared obstacle avoidance sensor.



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.



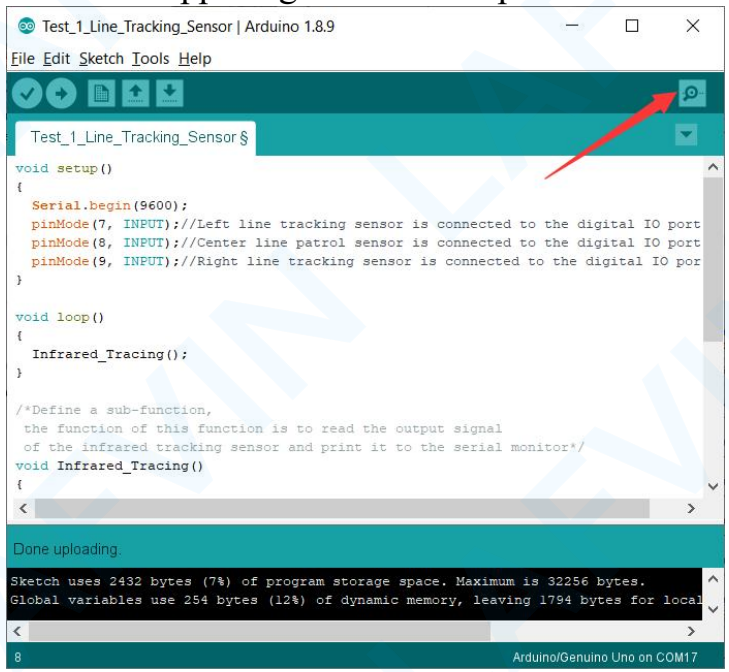
Important note: Before you upload the program, you need to close the serial monitor, otherwise the serial monitor will

occupy the communication interface of the uploaded program.

Arduino Code

if you want to refer to the program we provide.open the reference code for this lesson "**Test_1_Line_Tracking_Sensor.ino**" in the reference materials we provided.

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



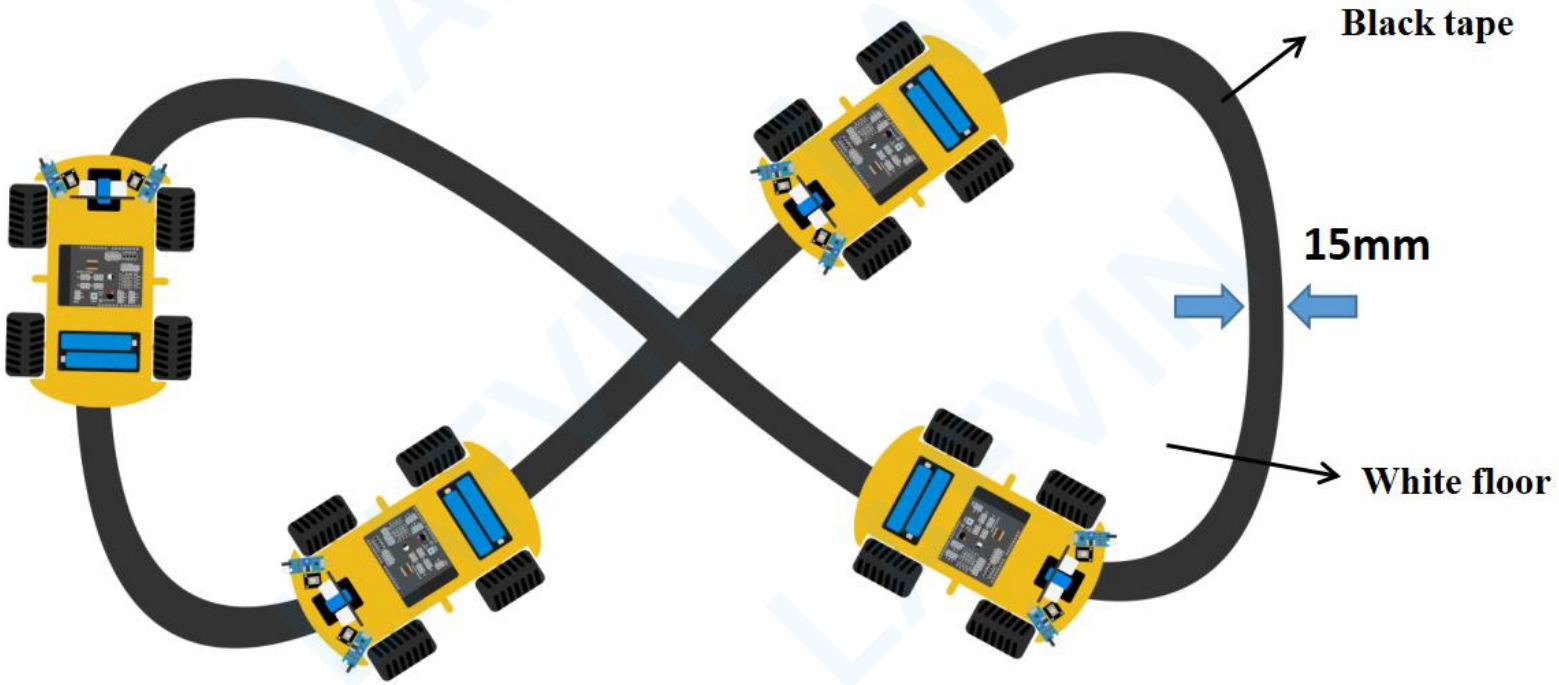
Then you can see the data as blow:

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.

Important note: Before you upload the program, you need to close the serial monitor, otherwise the serial monitor will occupy the communication interface of the uploaded program.

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.



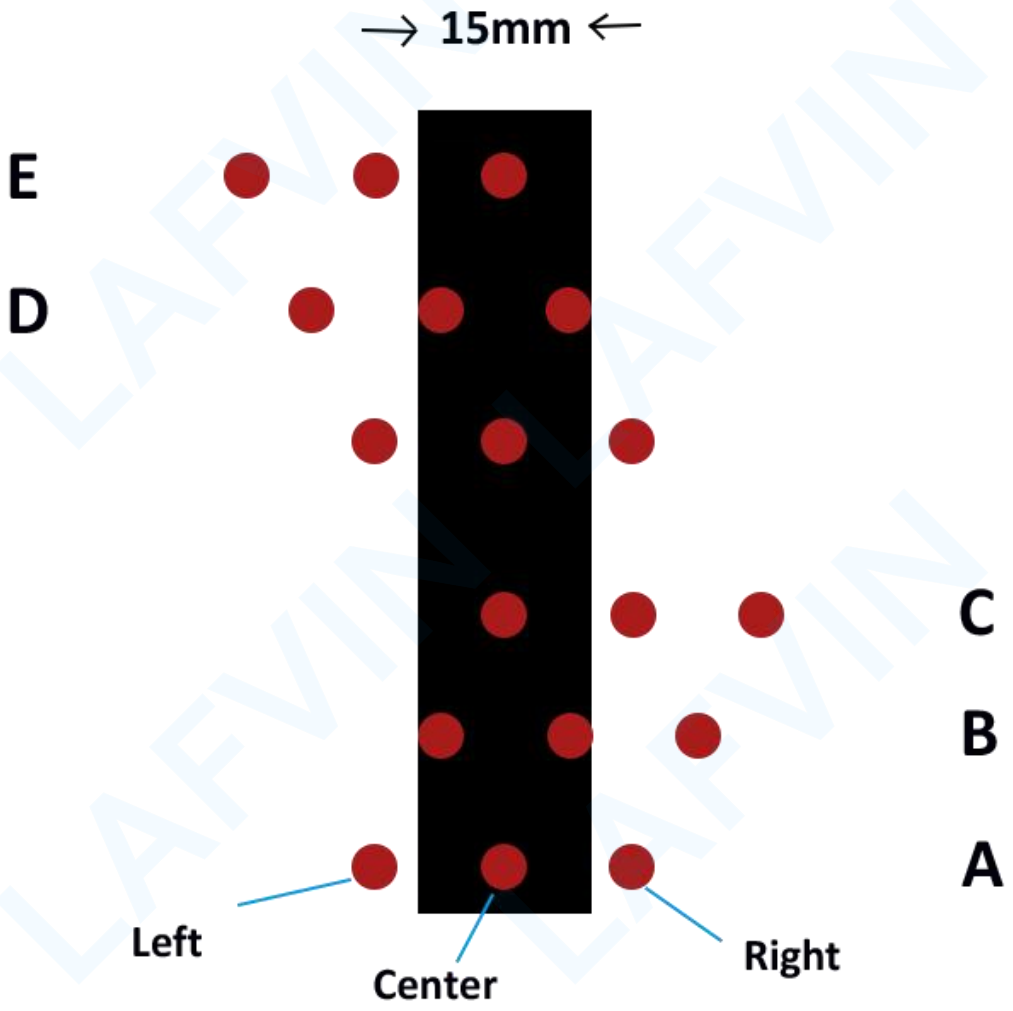
Mixly Code

Think about the code logic.Open Mixly software.Click “New” to add new project, then start your programming .

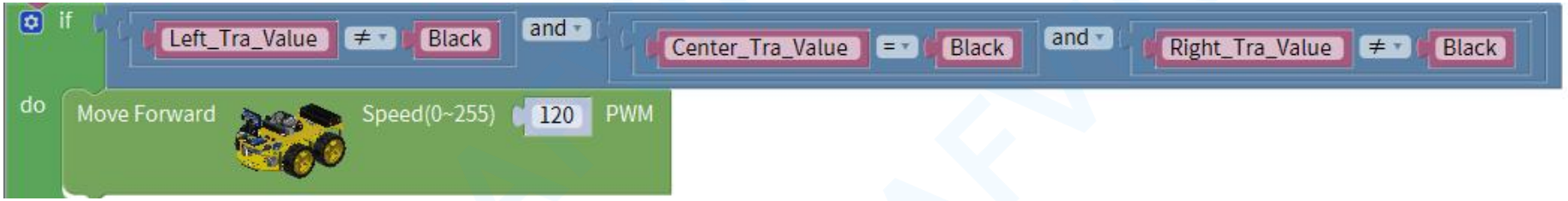
if you want to refer to the program we provide.Open the reference code for this lesson "**Test_2_Line_Tracking_Smart_Car.mix**" in the reference file we provided on CD.

Programming Thinking

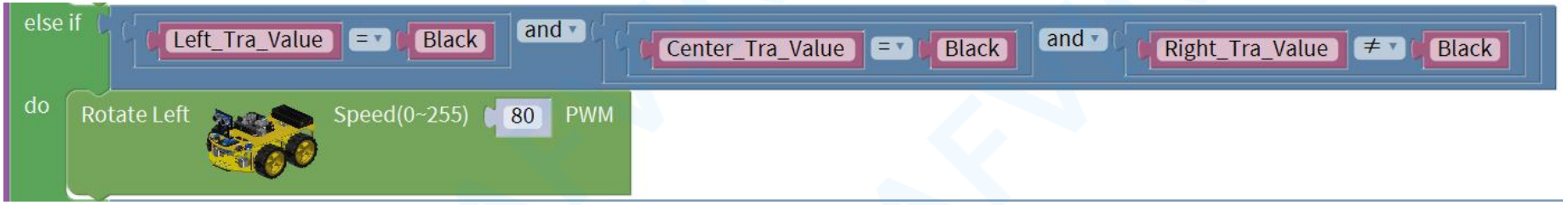
The line inspection part of the robot car includes three infrared line tracking sensors, namely the left line tracking sensor, the middle line tracking sensor, and the right line tracking sensor. A roll of black electrical tape is included in the kit parts. The width of the tape is 15mm, you can use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


15mm
E
D
C
B
A
Left
Center
Right

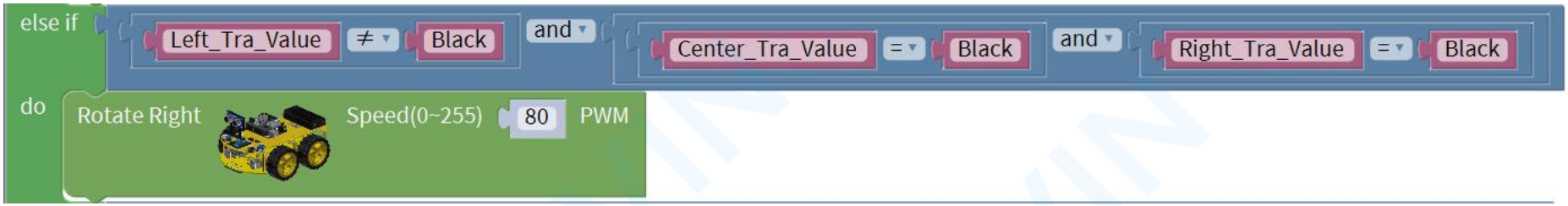
When in the A state, only the middle line-following sensor detects the black line, and the robot car moves straight at a speed of 120.



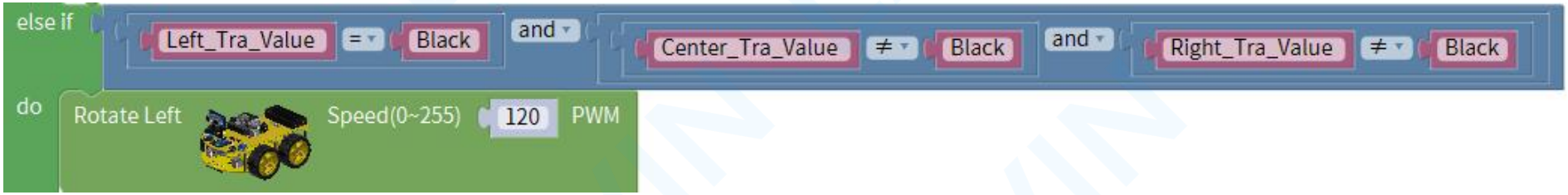
When in the B state, the left line-following sensor and the center line-following sensor detect the black line, and the robot car rotates to the left at a speed of 80%.



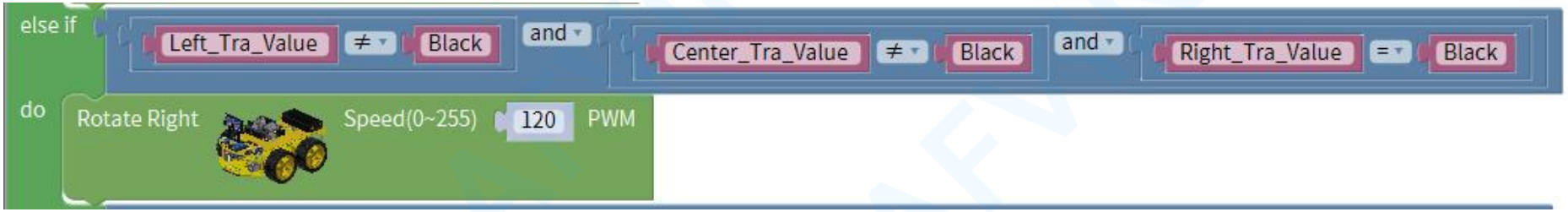
When in the D state, the right line-following sensor and the center line-following sensor detect the black line, and the robot car rotates to the right at a speed of 80%.



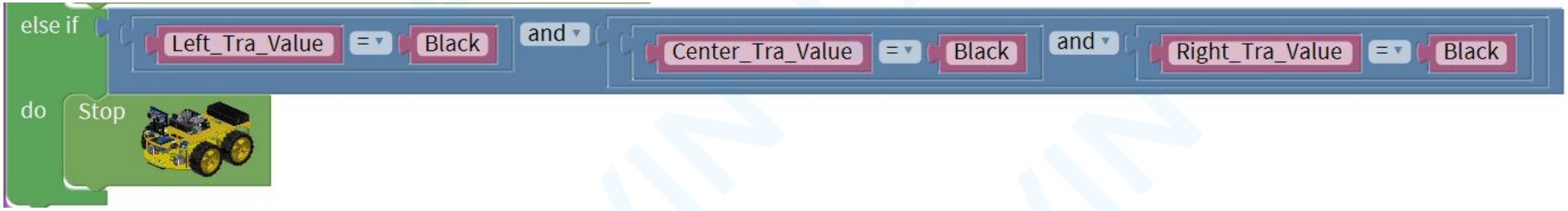
When in the C state, the left line-following sensor detects the black line, and the robot car rotates to the left at a speed of 120%.



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%.



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



Arduino Code

if you want to refer to the program we provide.open the reference code for this lesson "**Test_2_Line_Tracking_Smart_Car.ino**" in the reference materials we provided.

Programming Thinking

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 120.

```
if (Left_Tra_Value != Black && (Center_Tra_Value == Black && Right_Tra_Value != Black))
  {
    digitalWrite(2,HIGH);
    analogWrite(5,120);
    digitalWrite(4,LOW);
    analogWrite(6,120);
```

```
}
```

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 80%.

```
else if (Left_Tra_Value == Black && (Center_Tra_Value == Black && Right_Tra_Value != Black))
  {
    digitalWrite(2,LOW);
    analogWrite(5,80);
    digitalWrite(4,LOW);
    analogWrite(6,80);
  }
```

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 80%.

```
else if (Left_Tra_Value != Black && (Center_Tra_Value == Black && Right_Tra_Value == Black))
  {
     digitalWrite(2,HIGH);
     analogWrite(5,80);
     digitalWrite(4,HIGH);
     analogWrite(6,80);
  }
```

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))
  {
     digitalWrite(2,LOW);
```

```
    analogWrite(5,120);
    digitalWrite(4,LOW);
    analogWrite(6,120);
  }
```

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%.

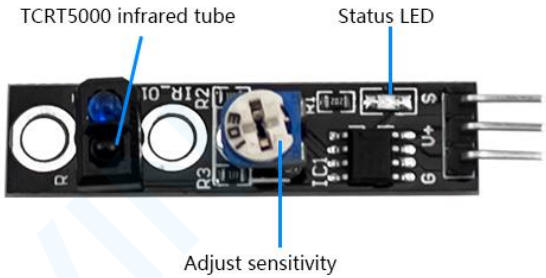
```
else if (Left_Tra_Value != Black && (Center_Tra_Value != Black && Right_Tra_Value == Black))
  {
    digitalWrite(2,HIGH);
```

```
    analogWrite(5,120);
    digitalWrite(4,HIGH);
    analogWrite(6,120);
  }
```

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

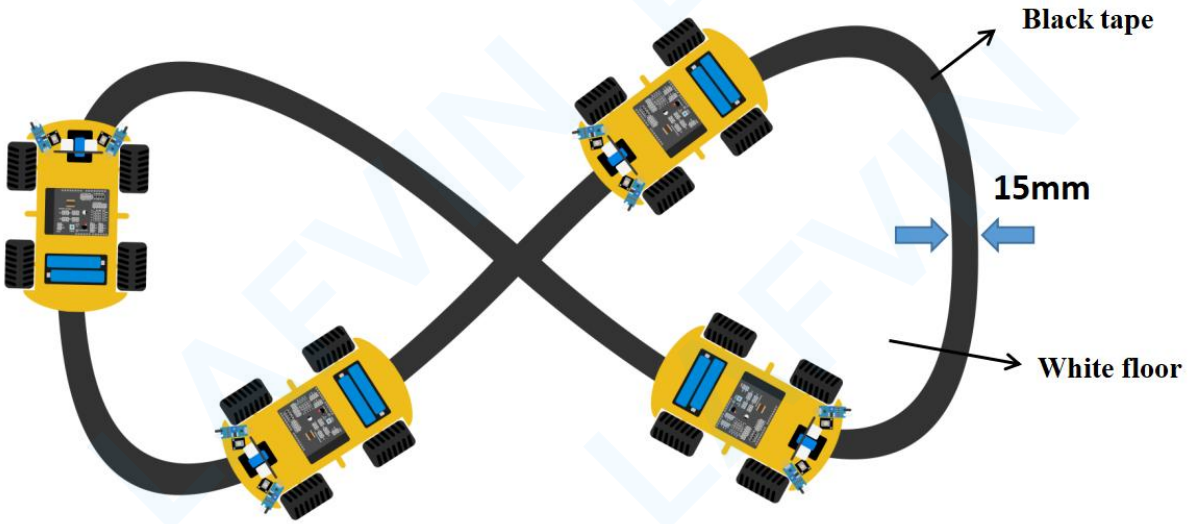
```
else if (Left_Tra_Value == Black && (Center_Tra_Value == Black && Right_Tra_Value == Black))
  {
    digitalWrite(2,LOW);
    analogWrite(5,0);
    digitalWrite(4,HIGH);
    analogWrite(6,0);
  }
```

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 Ultrasonic Infrared Obstacle Avoidance Robot Car

Overview

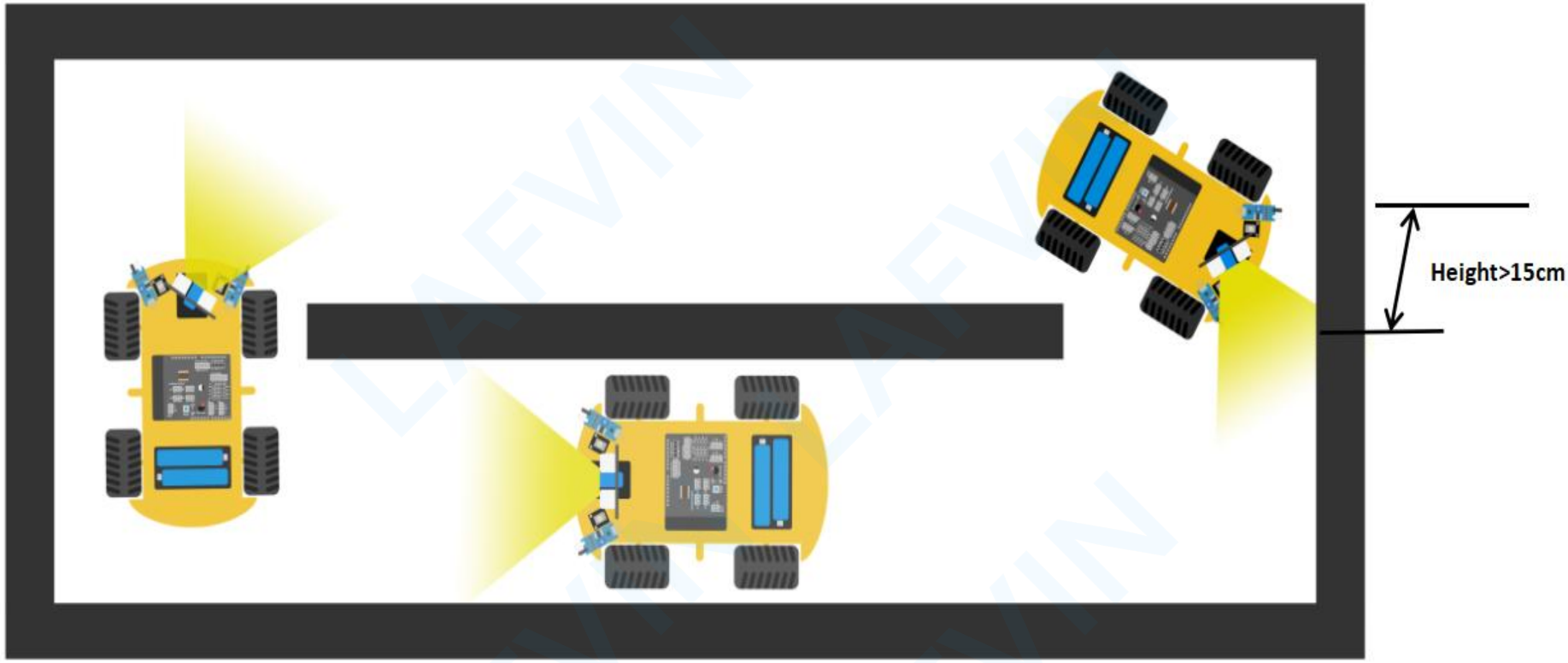
In this lesson, we will complete the test of 4 experimental codes.

In experimental test 1, we separately learn how to use infrared obstacle- avoidance sensors to detect obstacles, and observe the results returned by the sensors to determine whether obstacles are detected.

In the second experimental test 2, we will learn to use the ultrasonic module to measure distance.

In the experimental test 3, we will learn how to control the servo motor to rotate to any angle.

In the experimental test 4, the infrared obstacle avoidance sensor, the ultrasonic module and the servo motor were assembled on the robot car at the same time, and the data of these sensors were used at the same time to assist the robot car to complete the obstacle avoidance function more accurately.


Height>15cm

What is infrared obstacle avoidance sensor

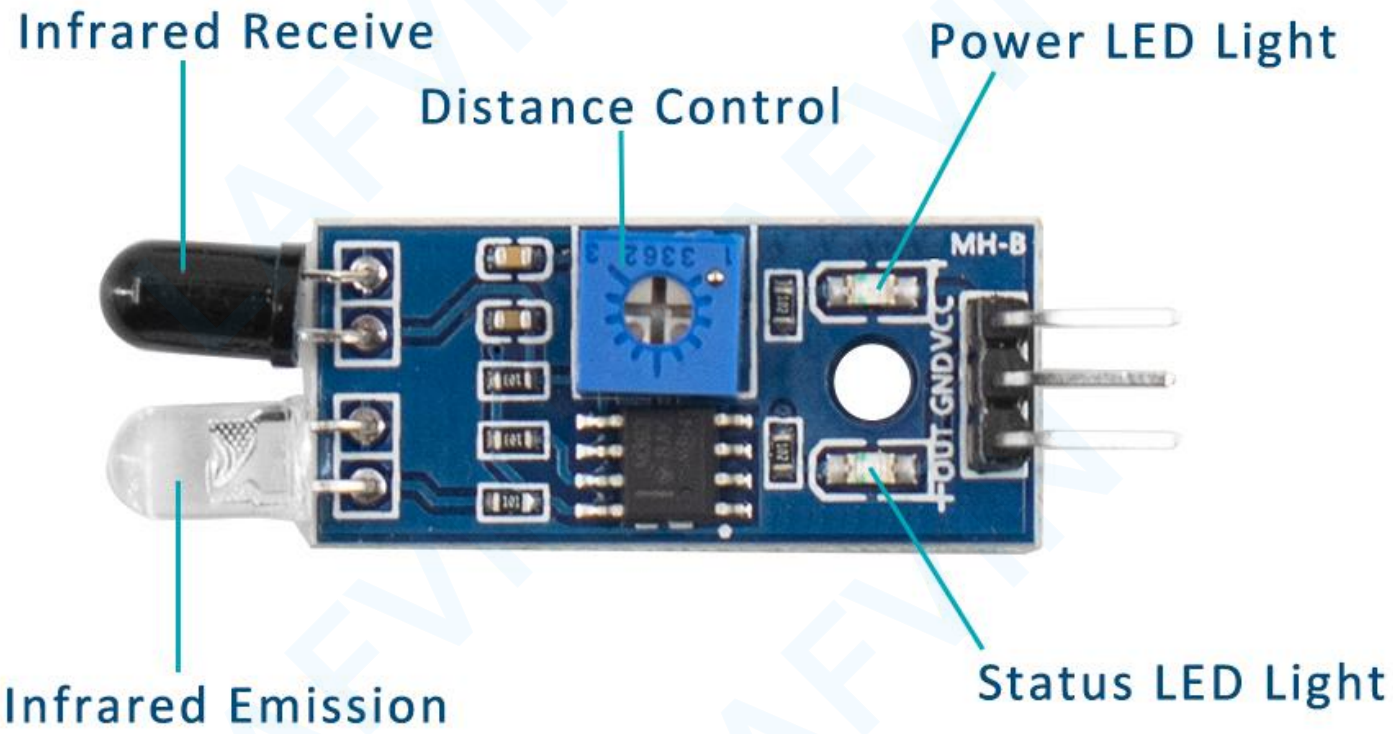
The infrared obstacle detector sensor has a pair of infrared transmitting and receiving tubes.The transmitter emits an infrared rays of a certain frequency. When the detection direction encounters an obstacle (reflecting surface), the infrared rays are reflected back, and receiving tube will receive it.At this time, the indicator (green LED) lights up. After processed by the circuit, the signal output terminal will output Digital signal.You can rotate the potentiometer on the shield to adjust the detection distance. It is better to adjust the potentiometer to make the green LED in a state between on and off. The detection distance is the best,almost 10cm.

How to use the infrared obstacl e avoidance sensor

we read the signal level of obstacle detector sensor to judge whether detect obstacles or not.

When detects an obstacle, sensor’ s signal pin outputs LOW (display 0); otherwise, output HIGH (display 1).

Show the result on the serial monitor, and control the external LED module turn ON/OFF.

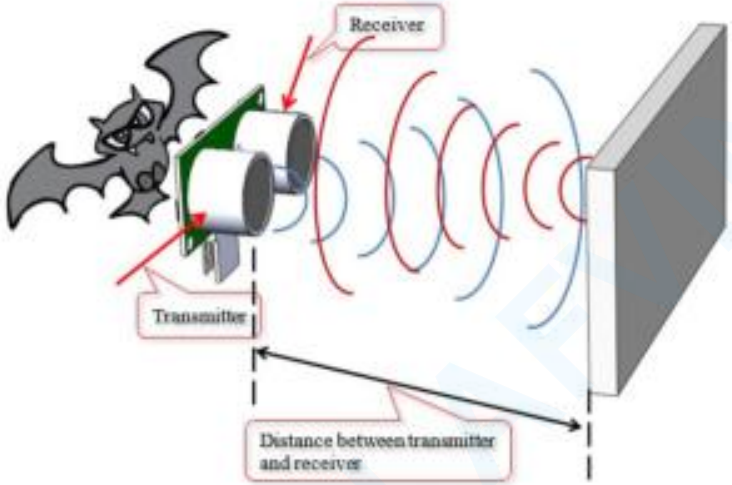

Infrared Receive
Distance Control
Power LED Light
MH-B
OUT GND VCC
Infrared Emission
Status LED Light

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.

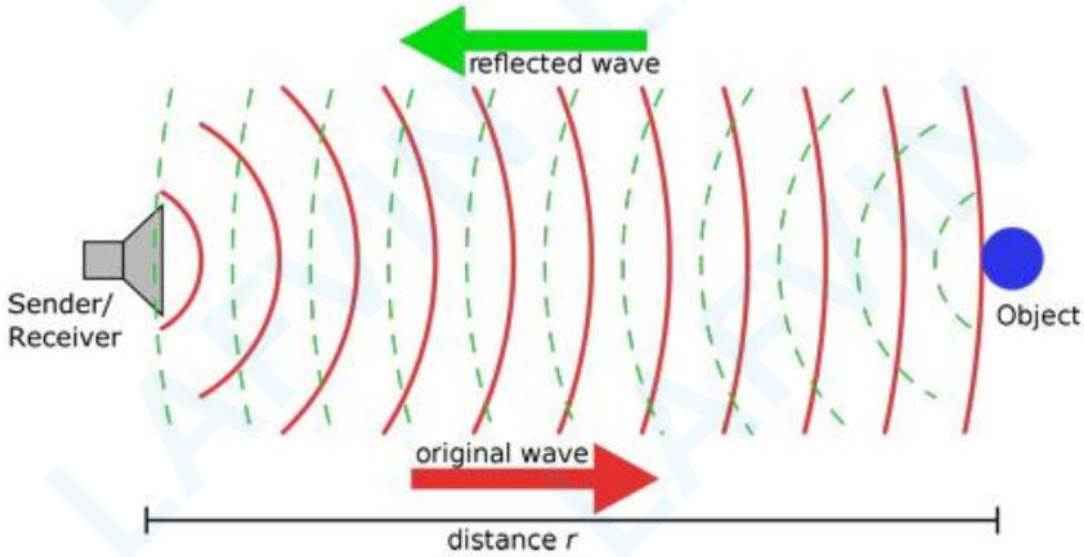
As the following picture shown, it is our ultrasonic module. It has two something like eyes.

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

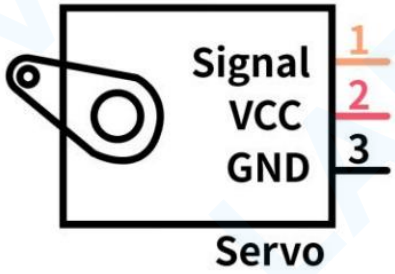
Measuring Angle: 30 degree

Trigger Input Pulse width: 10uS

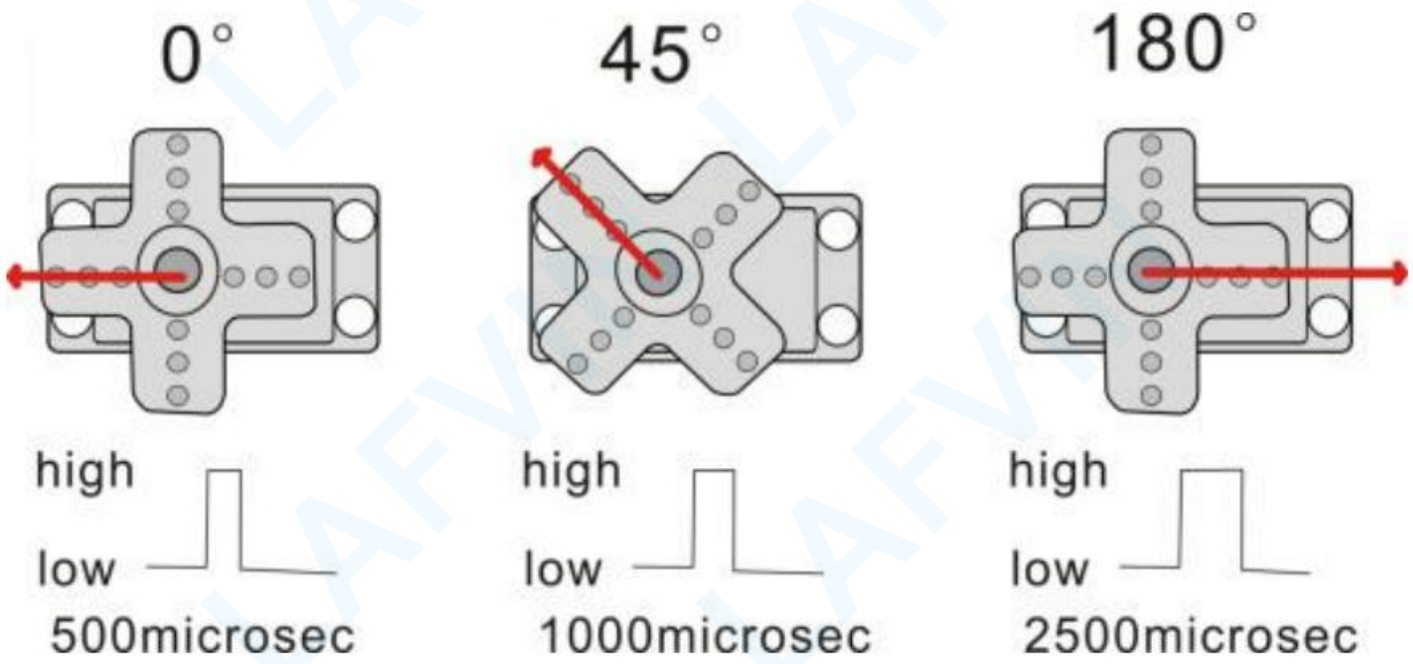
What is a servo motor

Description：

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:

Specification

Power Supply :+5V DC

Quiescent Current : <2mA

Working Current: 15mA

Effectual Angle: <15° 92

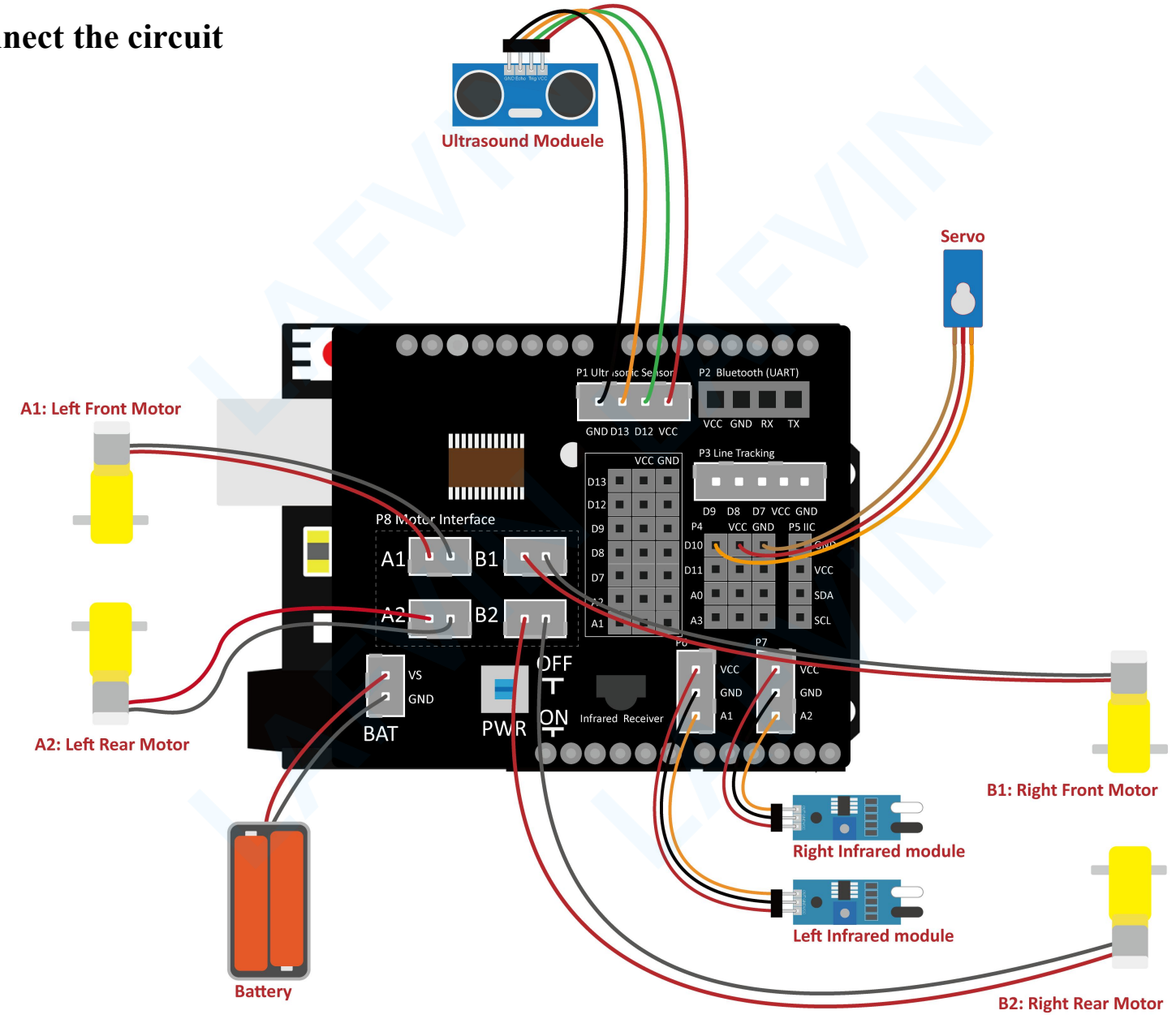
Ranging Distance : 2cm – 400 cm

Resolution : 0.3 cm

Measuring Angle: 30 degree

Trigger Input Pulse width: 10uS

How to connect the circuit



Let's program

Test 1--infrared obstacle avoidance sensor

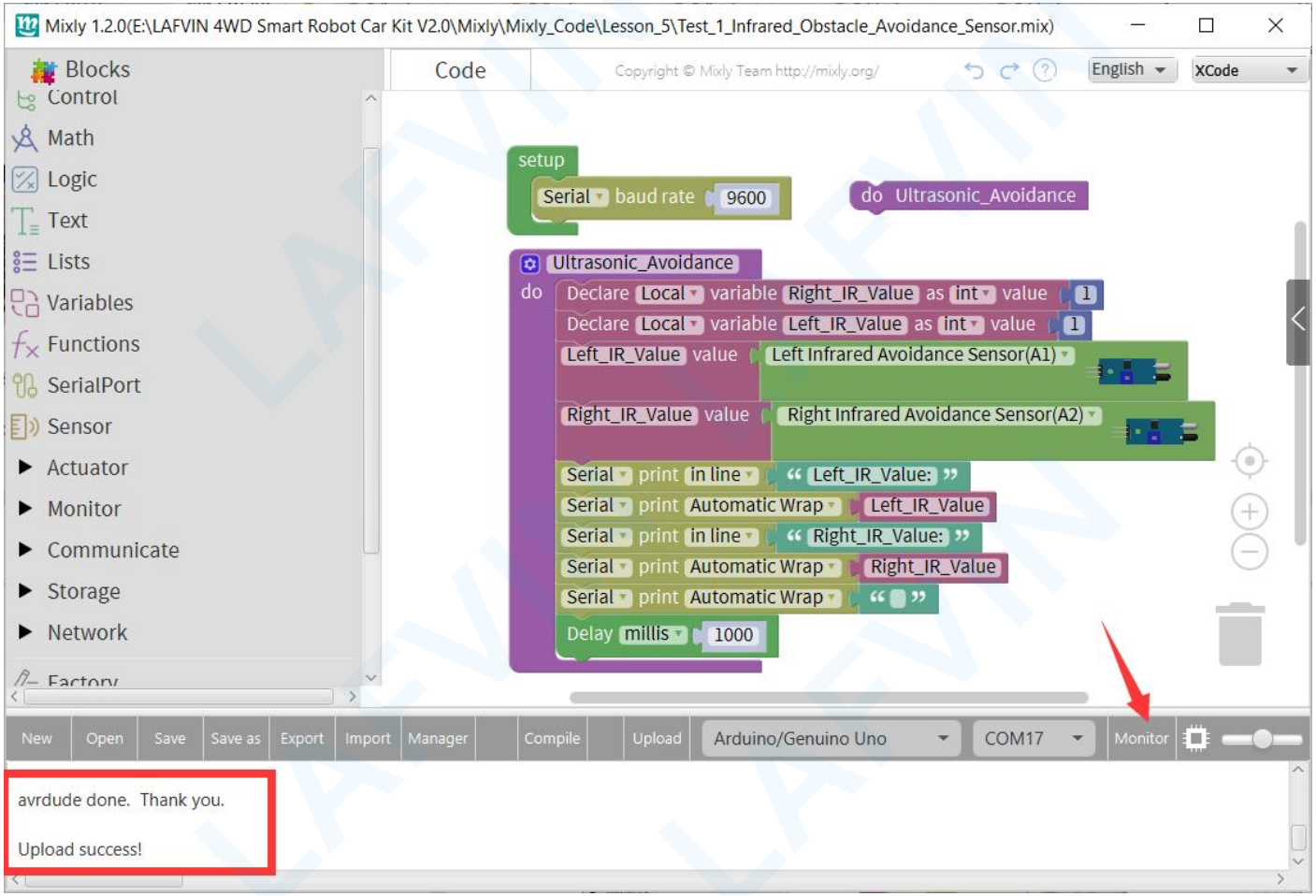
The main purpose of the test experiment is to read the return signal of the infrared obstacle avoidance sensor and print it to the serial port monitor.When detects an obstacle, sensor’s signal pin outputs LOW (display 0); otherwise, output HIGH (display 1).

Mixly Code

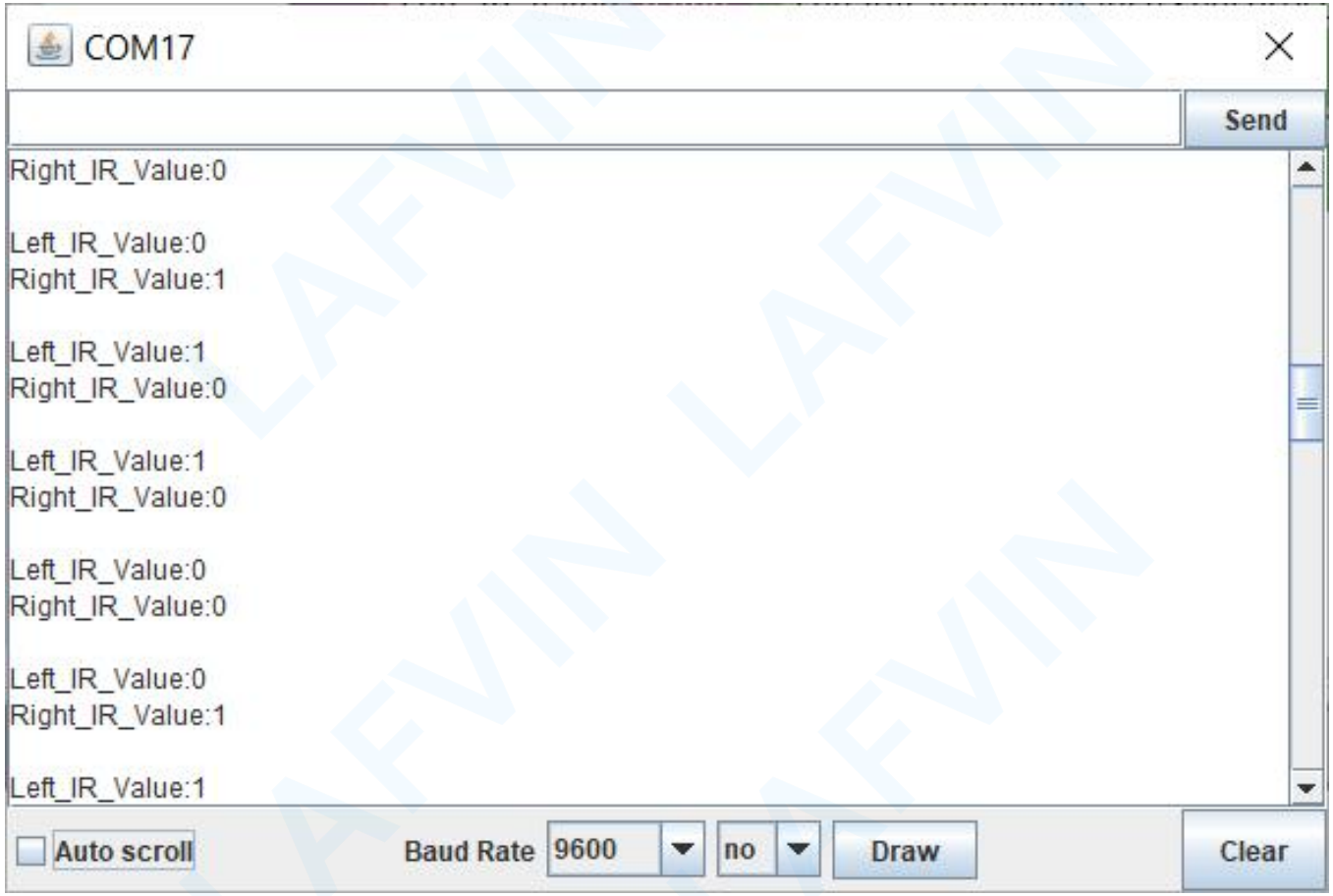
Wire it up well as the above diagram. Okay, let’s move on to write the test code.Open Mixly software.Click “New” to add new project, then start your programming .

if you want to refer to the program we provide.open the reference code for this lesson "**Test_1_Infrared_Obstacle_Avoidance_Sensor.mix**" in the reference materials we provided.

After uploading the program, open the serial monitor to view the signal returned by the infrared obstacle avoidance sensor.



Then you can see the data as blow:
When there is an obstacle in the front, IR_value=0, and the indicator light on the sensor lights up.



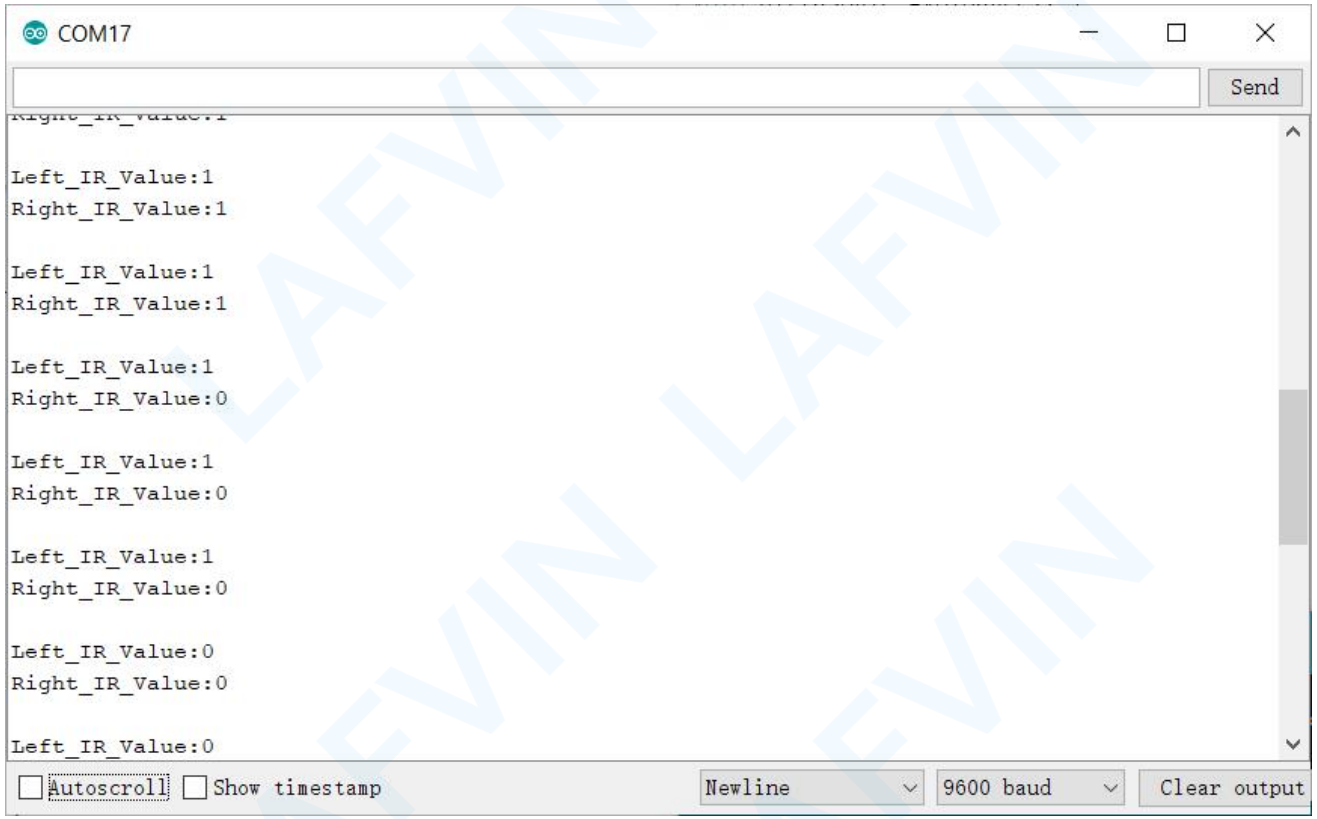
Important note: Before you upload the program, you need to close the serial monitor, otherwise the serial monitor will occupy the communication interface of the uploaded program.

Arduino Code

if you want to refer to the program we provide.open the reference code for this lesson "**Test_1_Infrared_Obstacle_Avoidance_Sensor.ino**" in the reference materials we provided.

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 there is an obstacle in the front, IR_value=0, and the indicator light on the sensor lights up.



Note: Before uploading the code, you need to turn off the serial monitor of the Arduino IDE and the monitor of the Mixly software, otherwise the code upload will fail because the serial port is usually occupied.

Test 2--Servo Control

In this experimental test , we learn how to control the servo motor and control the servo motor to rotate to the specified angle.

Mixly Code

Wire it up well as the above diagram. Okay, let's move on to write the test code.Open mixly software.Click "New" to add new project, then start your programming .

if you want to refer to the program we provide.open the reference code for this lesson "**Test_2_Servo_Control.mix**" in the reference files we provided CD.

Programming Thinking

This code block is used to control the motor rotation angle,the range is 0-180.This program block can only control the servo motor connected to the D10 digital IO interface

Arduino Code

if you want to refer to the program we provide.open Arduino IDE software and open the reference code for this lesson "**Test_2_Servo_Control.ino**" in the reference files we provided on CD.

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 .

Code Explanation

Arduino comes with #include <Servo.h> (servo function and statement）

The following are some common statements of the servo function:

1. attach（interface）——Set servo interface, port 9 and 10 are available

2. write（angle）——The statement to set rotation angle of servo, the angle range is from 0° to 180°

3. read（）——used to read angle of servo, read the command value of "write()"

4. attached（）——Judge if the parameter of servo is sent to its interface

Note: The above written format is"servo variable name, specific statement （）", for instance: myservo.attach(10)

After uploading the code, The servo motor rotates to 15° and waits for 2 seconds and then rotates to 165°, and after 2 seconds, it rotates to 90°, and repeats the above process.

Test 3--Ultrasonic Sensor Module

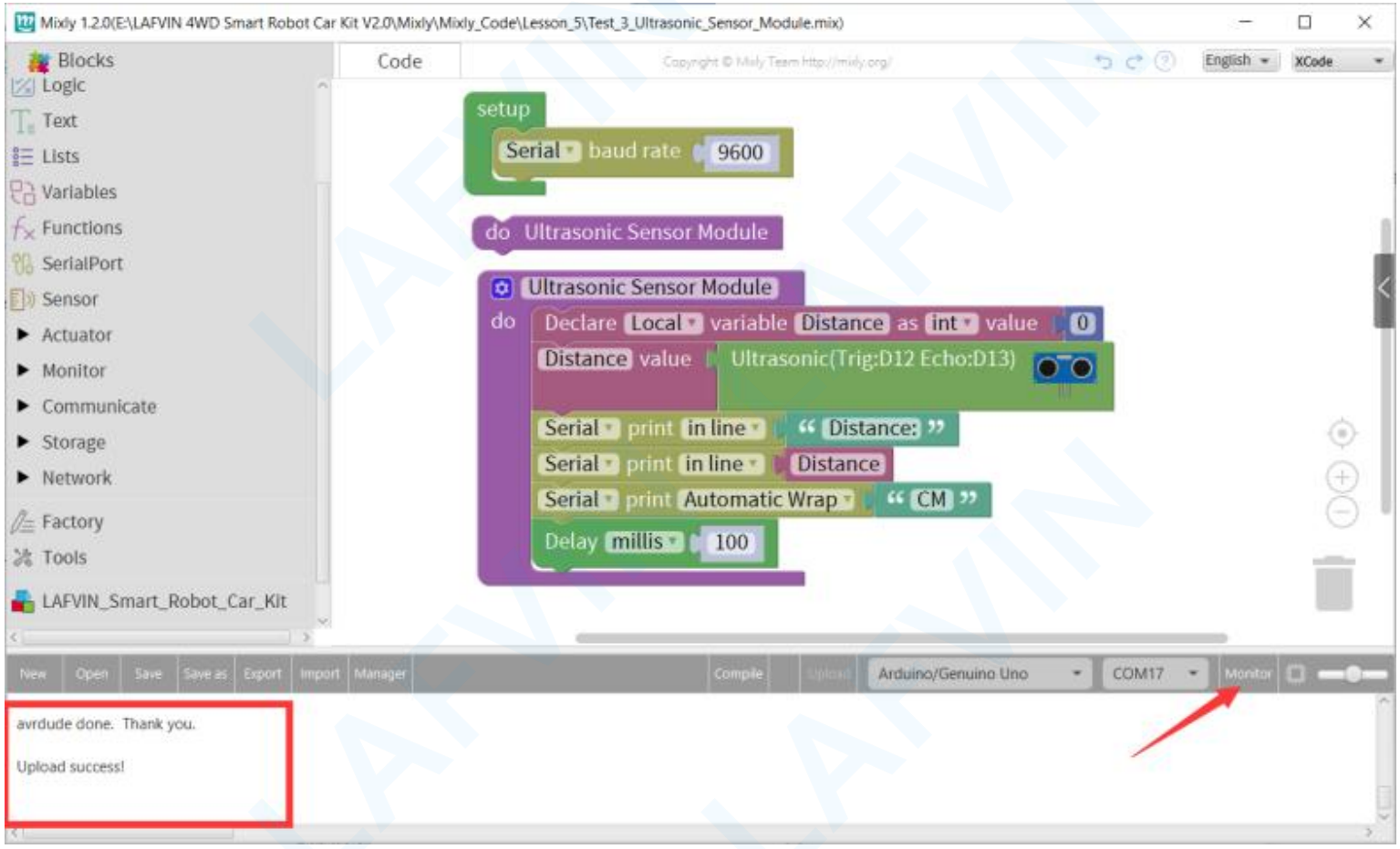
In Experimental Test 3, we will learn how to control the ultrasonic sensor, and display the distance measured by the ultrasonic sensor on the serial monitor.

Mixly Code

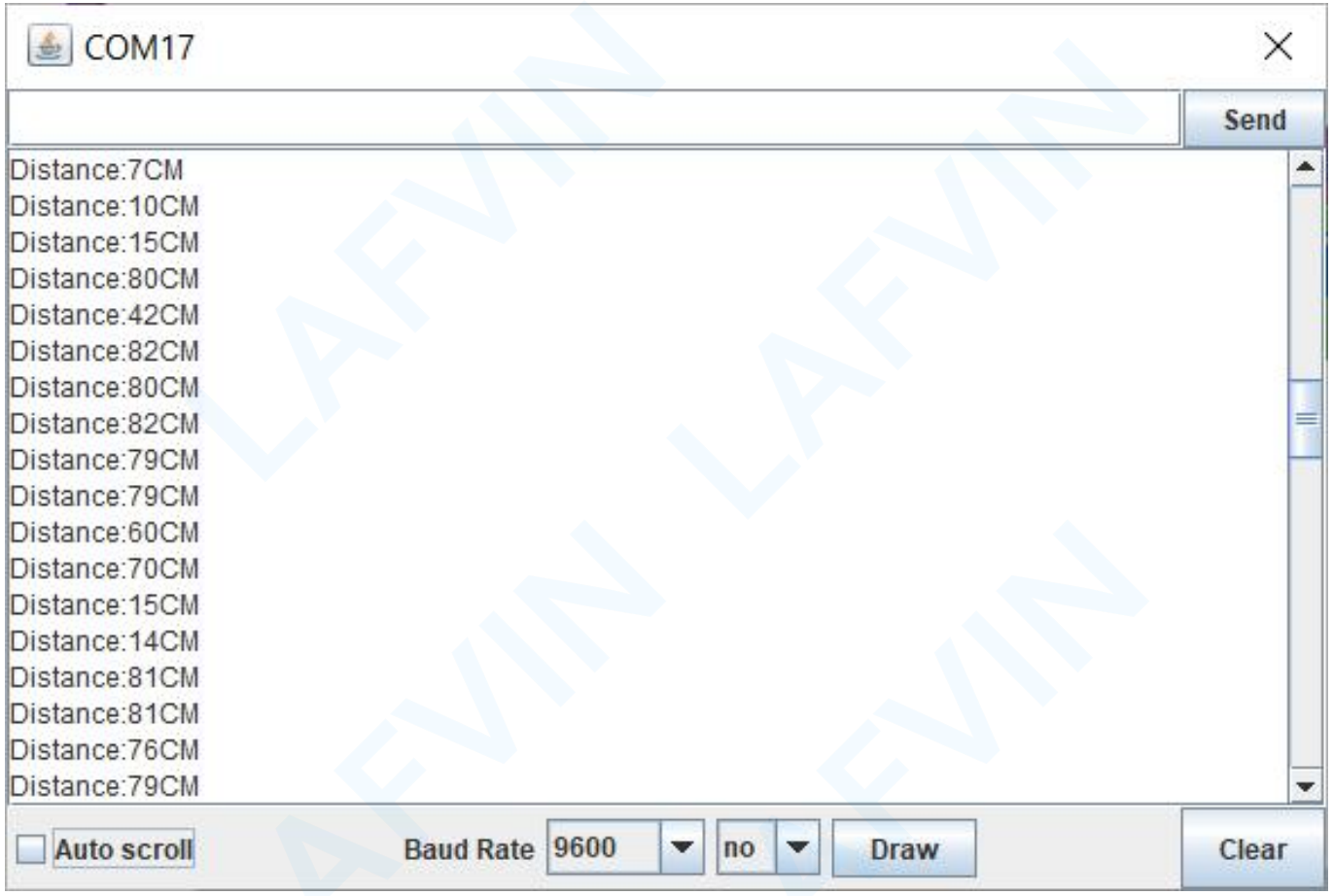
Wire it up well as the above diagram. Okay, let's move on to write the test code.Open mixly software.Click "New" to add new project, then start your programming .

if you want to refer to the program we provide.open the reference code for this lesson "**Test_3_Ultrasonic_Sensor_Module.mix**" in the reference files we provided CD.

After uploading the program, open the serial monitor to view the distance measured by the ultrasonic sensor.



Then you can see the data as blow:

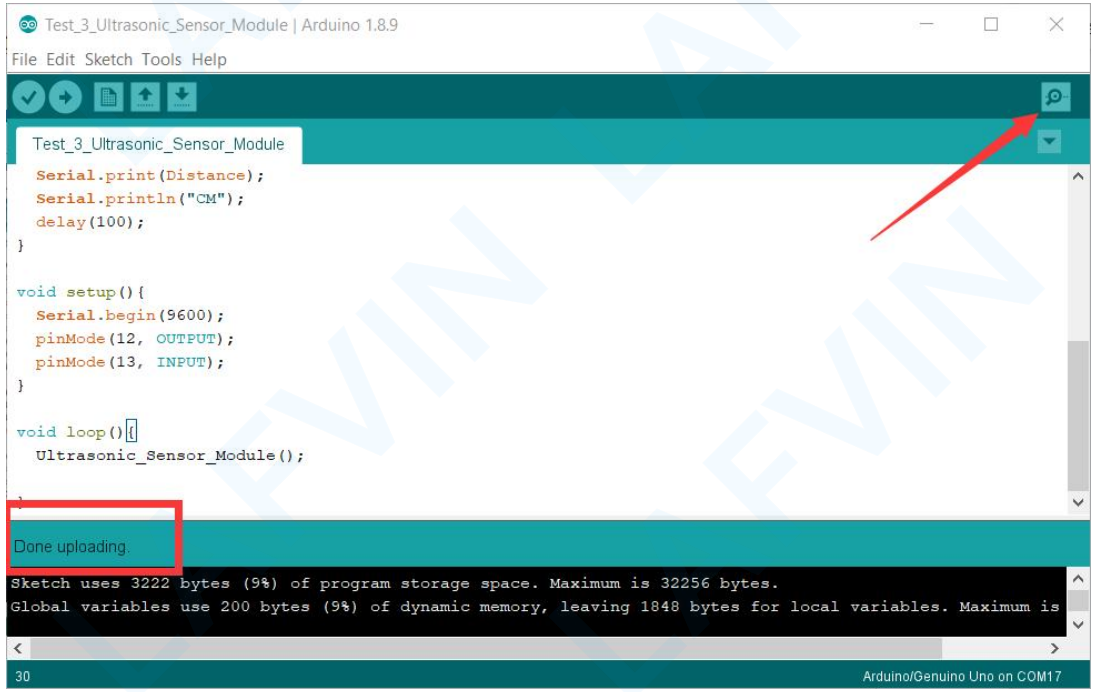


Note:that you need to close both the Arduino IDE serial monitor and the serial monitor of the Mixly software before uploading the code, otherwise the code will fail to upload because the communication serial port is occupied.

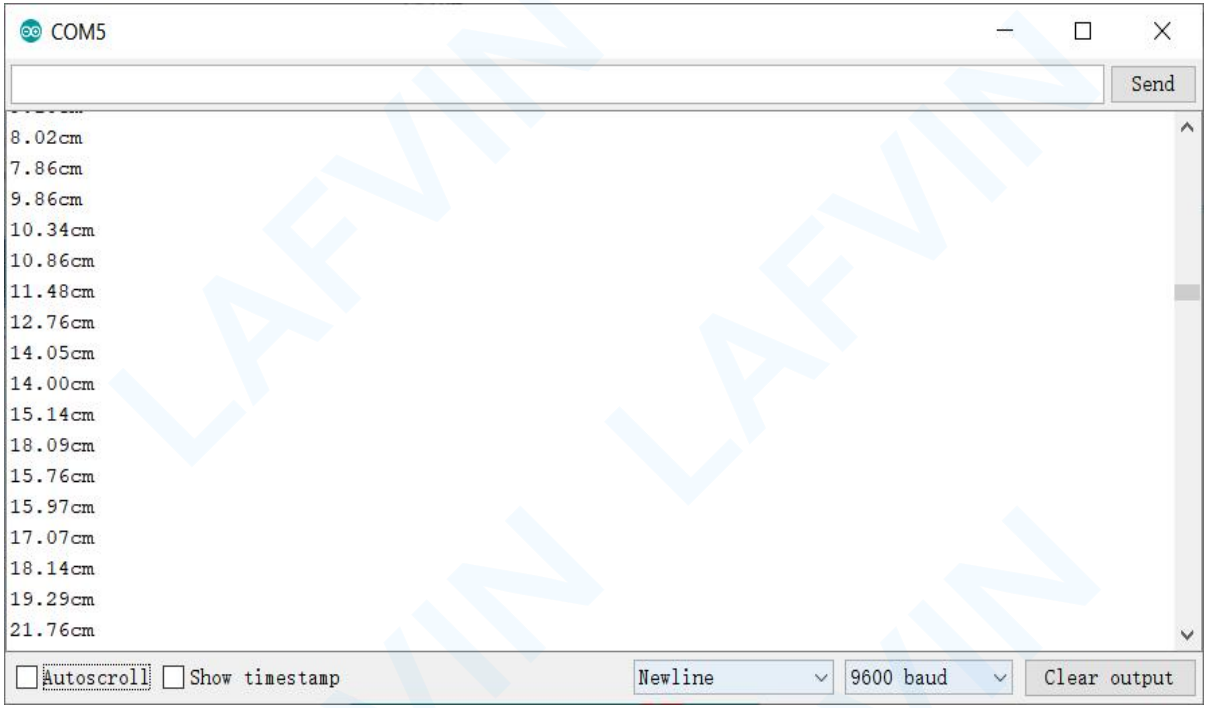
Arduino Code

if you want to refer to the program we provide.open Arduino IDE software and open the reference code for this lesson "**Test_3_Ultrasonic_Sensor_Module.ino**" in the reference materials we provided.

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:



Note: Before uploading the code, you need to turn off the serial monitor of the Arduino IDE and the monitor of the Mixly software, otherwise the code upload will fail because the serial port is usually occupied.

Test 4--Ultrasonic_Infrared_Obstacle_Avoidance_Robot_Car

In the experimental test 4, the infrared obstacle avoidance sensor, the ultrasonic module and the servo motor were assembled on the robot car at the same time, and the data of these sensors were used at the same time to assist the robot car to complete the obstacle avoidance function more accurately.

Mixly Code

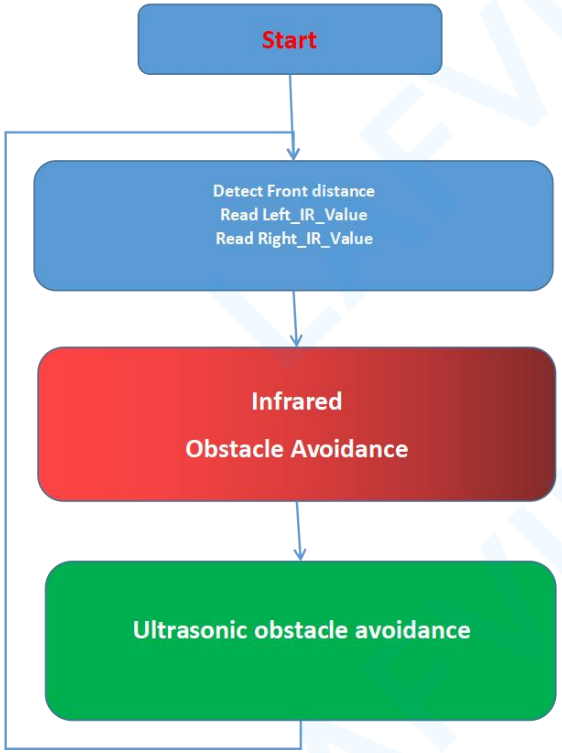
Think about the code logic.Open mixly software.Click “New” to add new project, then start your programming .

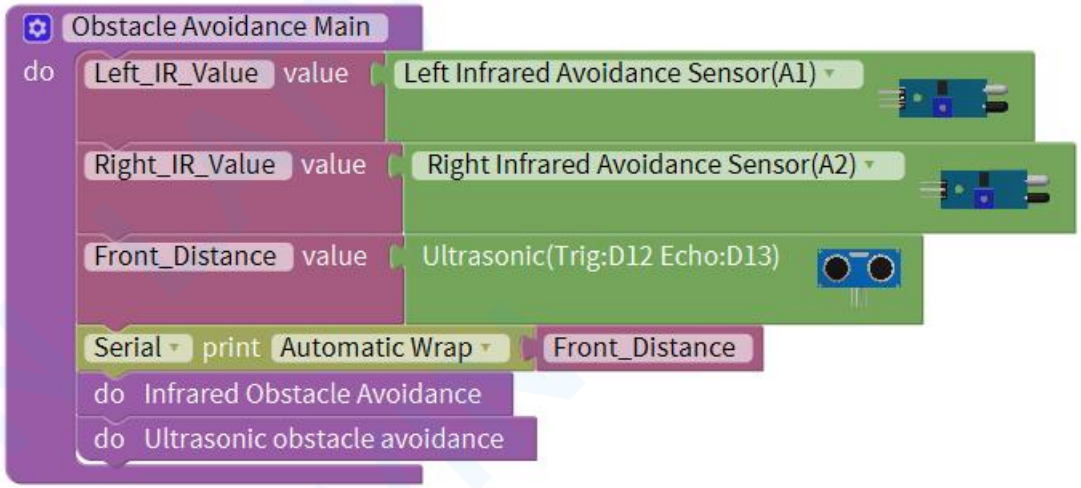
if you want to refer to the program we provide.open the reference code for this lesson "**Test_4_Ultrasonic_Infrared_Obstacle_Avoidance_Robot_Car.mix**" in the reference files we provided CD.

Programming Thinking

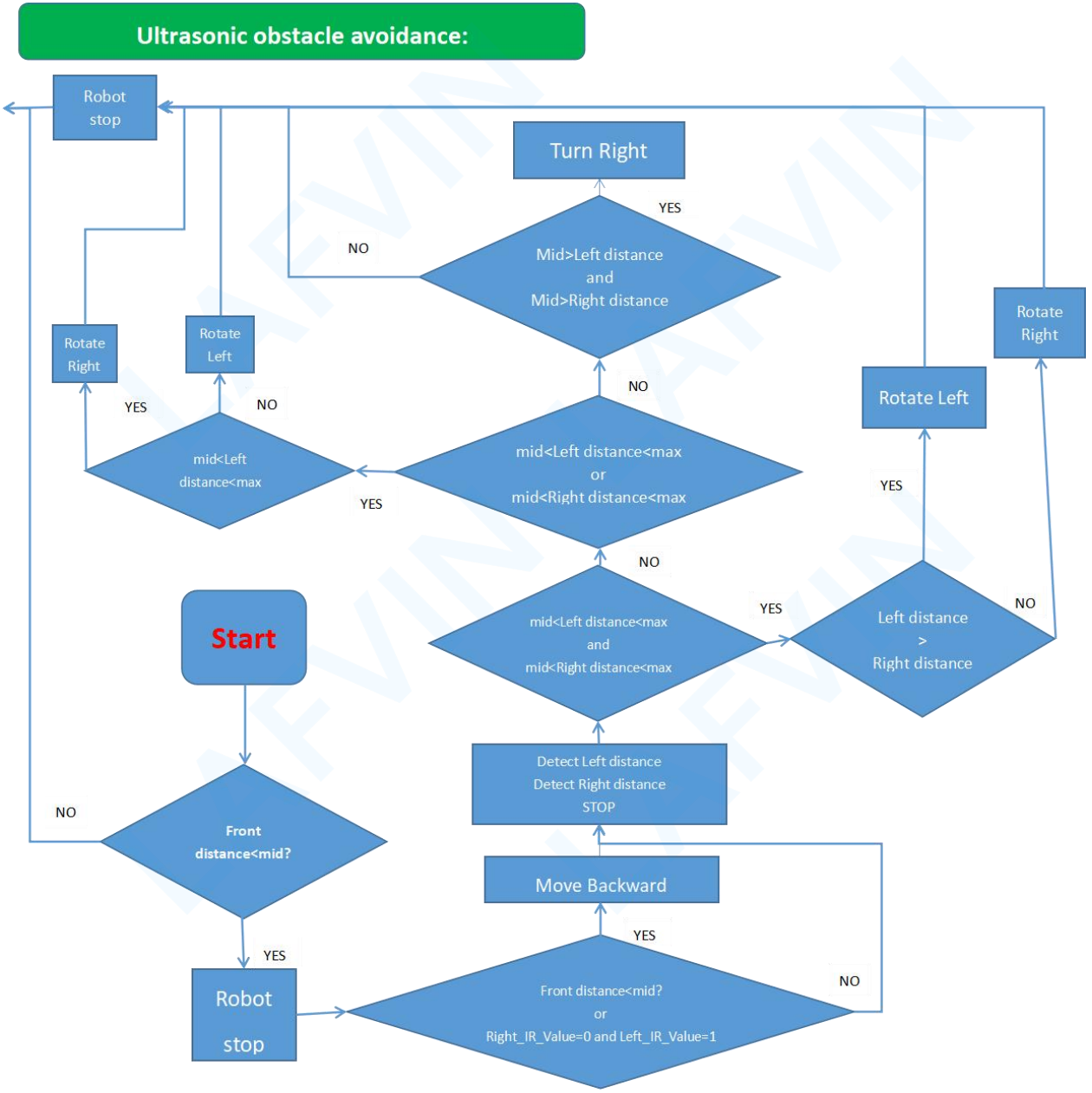
The following is the program execution flow chart of the robot car to complete the obstacle avoidance function

Main:


Start
Detect Front distance
Read Left_IR_Value
Read Right_IR_Value
Infrared
Obstacle Avoidance
Ultrasonic obstacle avoidance

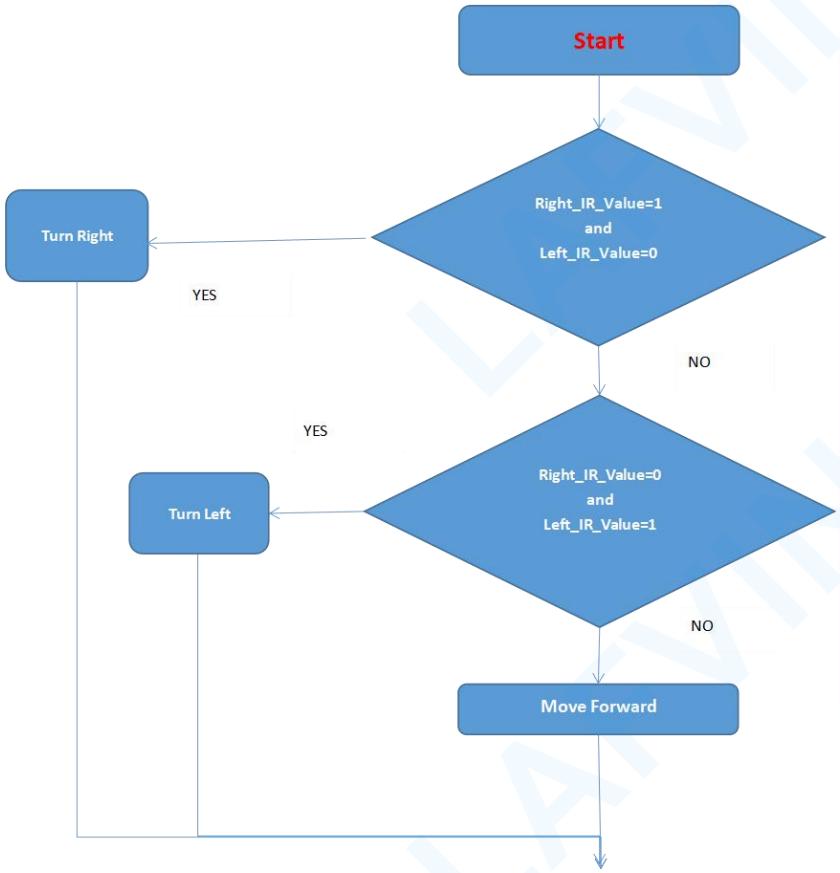

Obstacle Avoidance Main
do
Left_IR_Value value
Left Infrared Avoidance Sensor(A1)
Right_IR_Value value
Right Infrared Avoidance Sensor(A2)
Front_Distance value
Ultrasonic(Trig:D12 Echo:D13)
Serial print Automatic Wrap
Front_Distance
do Infrared Obstacle Avoidance
do Ultrasonic obstacle avoidance

Ultrasonic obstacle avoidance:




Ultrasonic obstacle avoidance
do
if
Front_Distance
D_mid
do
Stop
if
Front_Distance
D_mix
or
Left_IR_Value
0
and
Right_IR_Value
0
do
Move Backward
Speed(0~255)
4.5
22.5
PWM
Delay
millis
300
Stop
do
Detect Left and Right distance
if
D_mix
Left_Distance
and
Left_Distance
D_max
and
D_mix
Right_Distance
and
Right_Distance
D_max
do
if
Right_Distance
Left_Distance
do
Rotate Right
Speed(0~255)
9
22.5
PWM
Delay
millis
300
else
Rotate Left
Speed(0~255)
9
22.5
PWM
Delay
millis
300
else if
D_mix
Left_Distance
and
Left_Distance
D_max
or
D_mix
Right_Distance
and
Right_Distance
D_max
do
if
D_mix
Left_Distance
and
Left_Distance
D_max
do
Rotate Left
Speed(0~255)
7
22.5
PWM
Delay
millis
310
else if
D_mix
Right_Distance
and
Right_Distance
D_max
do
Rotate Right
Speed(0~255)
7
22.5
PWM
Delay
millis
310
else if
Right_Distance
D_mix
and
Left_Distance
D_mix
do
Move Forward. Turn at different speeds
Left Wheel(0~255)
0
PWM
Right Wheel(0~255)
9
22.5
PWM
Delay
millis
510
Stop
Stop

Infrared Obstacle Avoidance:



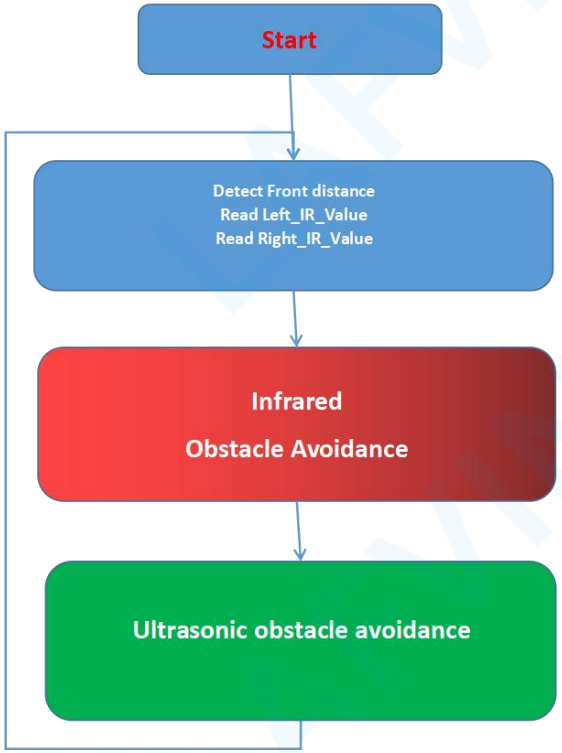
Arduino Code

if you want to refer to the program we provide.open the reference code for this lesson "**Test_4_Ultrasonic_Infrared_Obstacle_Avoidance_Robot_Car.ino**" in the reference materials we provided.

Programming Thinking

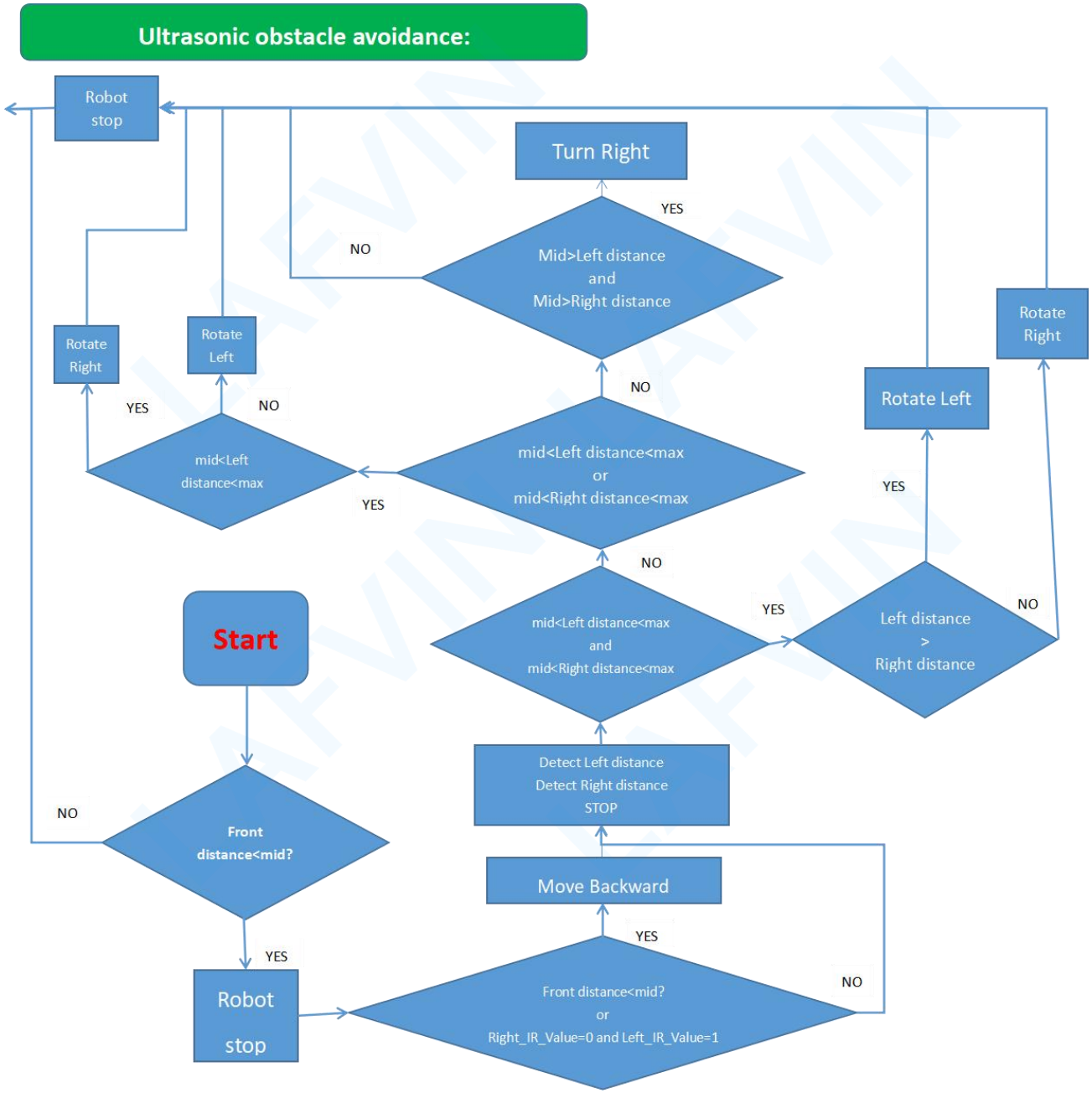
The following is the program execution flow chart of the robot car to complete the obstacle avoidance function

Main:




```
void Obstacle_Avoidance_Main() {
  Left_IR_Value = digitalRead(A1);
  Right_IR_Value = digitalRead(A2);
  Front_Distance = checkdistance();
  Serial.println(Front_Distance);
  Infrared_Obstacle_Avoidance();
  Ultrasonic_obstacle_avoidance();
}
```

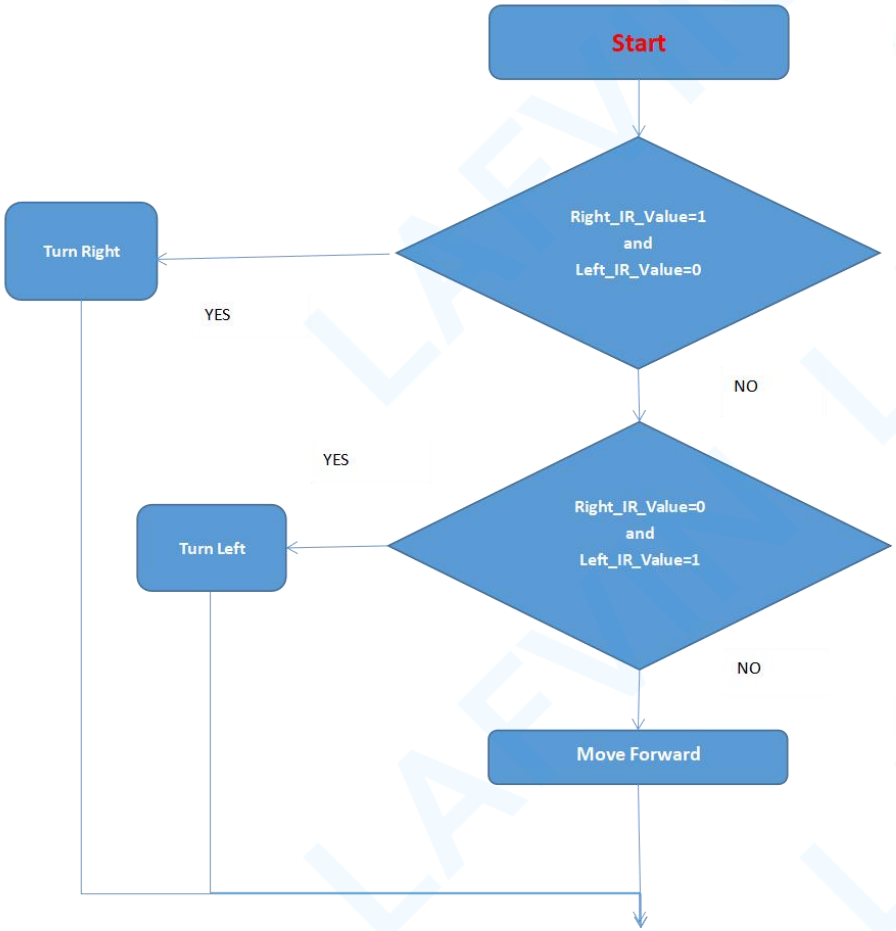
Ultrasonic obstacle avoidance:



```
void Ultrasonic_obstacle_avoidance() {
  if (Front_Distance <= D_mid) {
    digitalWrite(2,LOW);
    analogWrite(5,0);
    digitalWrite(4,HIGH);
    analogWrite(6,0);
    if (Front_Distance <= D_mix || Left_IR_Value == 0 && Right_IR_Value == 0) {
      digitalWrite(2,LOW);
      analogWrite(5,(4.5 * 22.5));
      digitalWrite(4,HIGH);
      analogWrite(6,(4.5 * 22.5));
      delay(300);
      digitalWrite(2,LOW);
      analogWrite(5,0);
      digitalWrite(4,HIGH);
      analogWrite(6,0);
    }
    Detect_Left_and_Right__distance();
    if ((D_mix < Left_Distance && Left_Distance < D_max) && (D_mix < Right_Distance && Right_Distance < D_max)) {
      if (Right_Distance > Left_Distance) {
        digitalWrite(2,HIGH);
        analogWrite(5,(9 * 22.5));
        digitalWrite(4,HIGH);
        analogWrite(6,(9 * 22.5));
        delay(300);
      } else {
        digitalWrite(2,LOW);
        analogWrite(5,(9 * 22.5));
        digitalWrite(4,LOW);
        analogWrite(6,(9 * 22.5));
        delay(300);
      }
    } else if (D_mix < Left_Distance && Left_Distance < D_max || D_mix < Right_Distance && Right_Distance < D_max) {
```

```
  } else if (D_mix < Left_Distance && Left_Distance < D_max || D_mix < Right_Distance && Right_Distance < D_max) {
    if (D_mix < Left_Distance && Left_Distance < D_max) {
      digitalWrite(2,LOW);
      analogWrite(5,(7 * 22.5));
      digitalWrite(4,LOW);
      analogWrite(6,(7 * 22.5));
      delay(310);
    } else if (D_mix < Right_Distance && Right_Distance < D_max) {
      digitalWrite(2,HIGH);
      analogWrite(5,(7 * 22.5));
      digitalWrite(4,HIGH);
      analogWrite(6,(7 * 22.5));
      delay(310);
    }
  } else if (Right_Distance < D_mix && Left_Distance < D_mix) {
    digitalWrite(2,HIGH);
    analogWrite(5,0);
    digitalWrite(4,LOW);
    analogWrite(6,(9 * 22.5));
    delay(510);
    digitalWrite(2,LOW);
    analogWrite(5,0);
    digitalWrite(4,HIGH);
    analogWrite(6,0);
  }
  digitalWrite(2,LOW);
  analogWrite(5,0);
  digitalWrite(4,HIGH);
  analogWrite(6,0);
 }
}
```

Infrared Obstacle Avoidance:



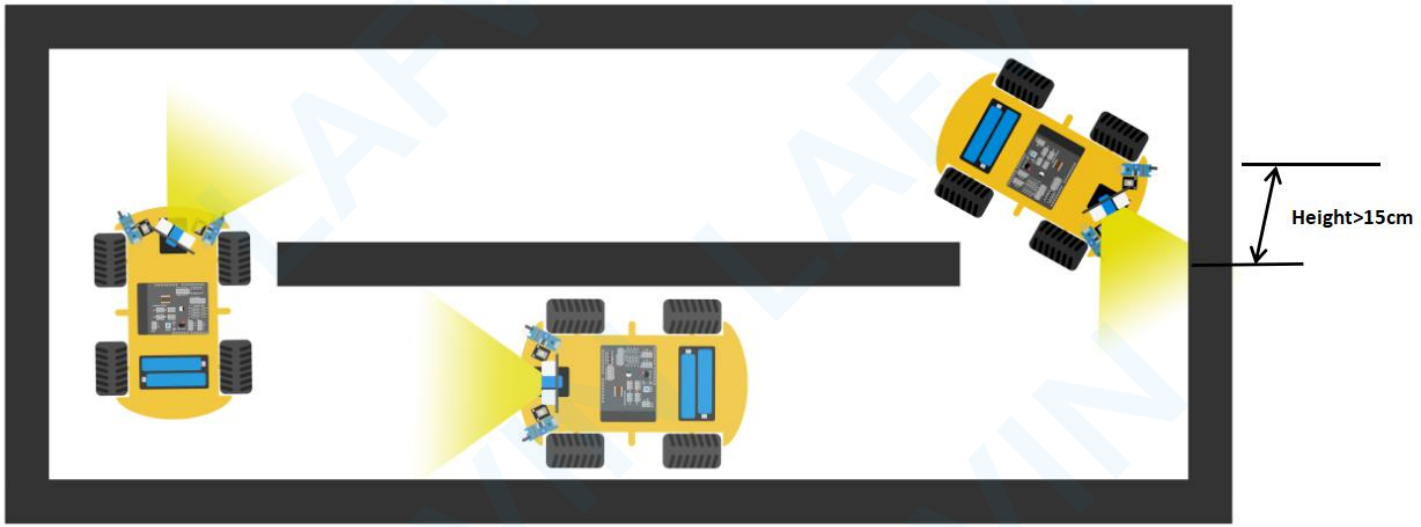

```
void Infrared_Obstacle_Avoidance() {
  if (Left_IR_Value == 0 && Right_IR_Value == 1) {
    digitalWrite(2,HIGH);
    analogWrite(5,255);
    digitalWrite(4,LOW);
    analogWrite(6,12);

  } else if (Left_IR_Value == 1 && Right_IR_Value == 0) {
    digitalWrite(2,HIGH);
    analogWrite(5,12);
    digitalWrite(4,LOW);
    analogWrite(6,255);
  } else {
    digitalWrite(2,HIGH);
    analogWrite(5,(4 * 22.5));
    digitalWrite(4,LOW);
    analogWrite(6,(4 * 22.5));

  }
}
```

What will you see

Upload the code to UNO R3 control board,, and turn the POWER switch ON.Note: It is recommended that the height of the obstacle is greater than 15cm.which means that the height of the obstacle is greater than the height of the ultrasonic sensor.



Because of receiving the detection signals of ultrasonic and infrared obstacle avoidance modules at the same time, the robot car can complete the obstacle avoidance function more accurately. When obstacles are on the left and right sides of the robot car, they can be discovered by the infrared obstacle avoidance module in time and turn to avoid. When the obstacle is directly in front, the ultrasonic will be able to measure the distance between the robot car and the obstacle.

Lesson 6 Ultrasonic Follow Robot Car

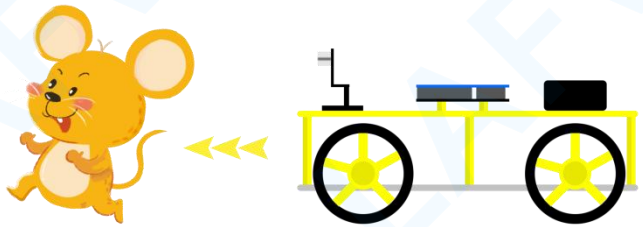
Overview

In the last course, I learned how to use the ultrasonic module and infrared obstacle avoidance module. Similarly, combining these two modules can control the robot car to complete the following function.

When the guiding object is on the left and right sides of the robot car, the infrared obstacle avoidance module can detect and judge the direction of the guiding object, and control the robot car to turn to the guide object.

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 10mm, control the robot car to approach the guide;

When the distance is less than 5mm, control the robot car to move backward and keep a proper distance, 5mm<D<10mm.



What is infrared obstacle avoidance sensor

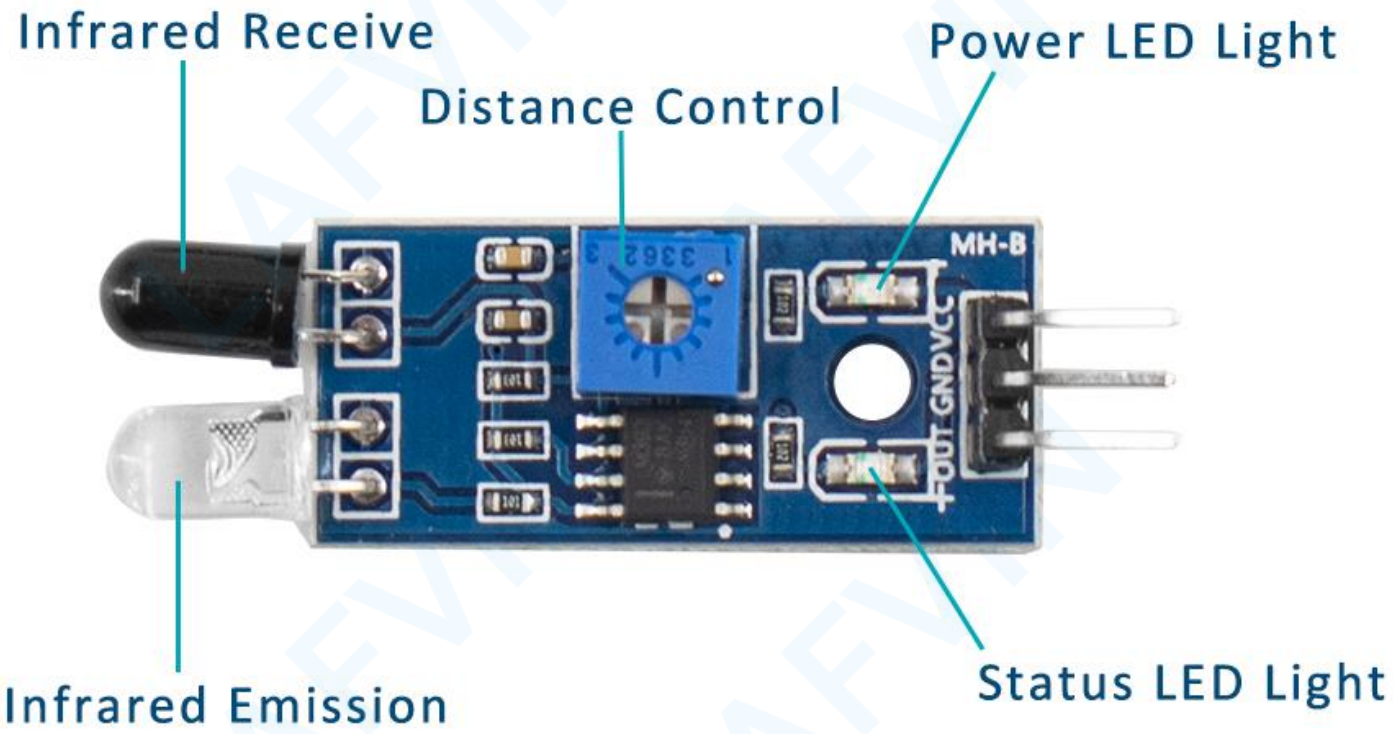
The infrared obstacle detector sensor has a pair of infrared transmitting and receiving tubes.The transmitter emits an infrared rays of a certain frequency. When the detection direction encounters an obstacle (reflecting surface), the infrared rays are reflected back, and receiving tube will receive it.At this time, the indicator (green LED) lights up. After processed by the circuit, the signal output terminal will output Digital signal.You can rotate the potentiometer on the shield to adjust the detection distance. It is better to adjust the potentiometer to make the green LED in a state between on and off. The detection distance is the best,almost 10cm.

How to use the infrared obstacle avoidance sensor

we read the signal level of obstacle detector sensor to judge whether detect obstacles or not.

When detects an obstacle, sensor’ s signal pin outputs LOW (display 0); otherwise, output HIGH (display 1).

Show the result on the serial monitor, and control the external LED module turn ON/OFF.

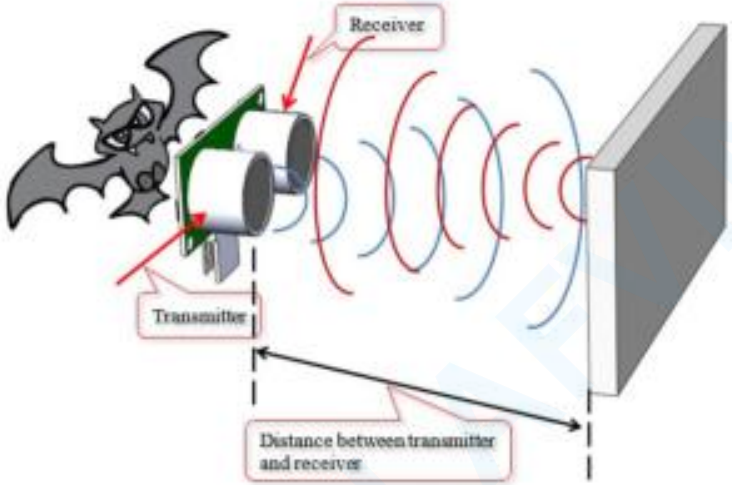

Infrared Receive
Distance Control
Power LED Light
MH-B
OUT GND VCC
Infrared Emission
Status LED Light

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.

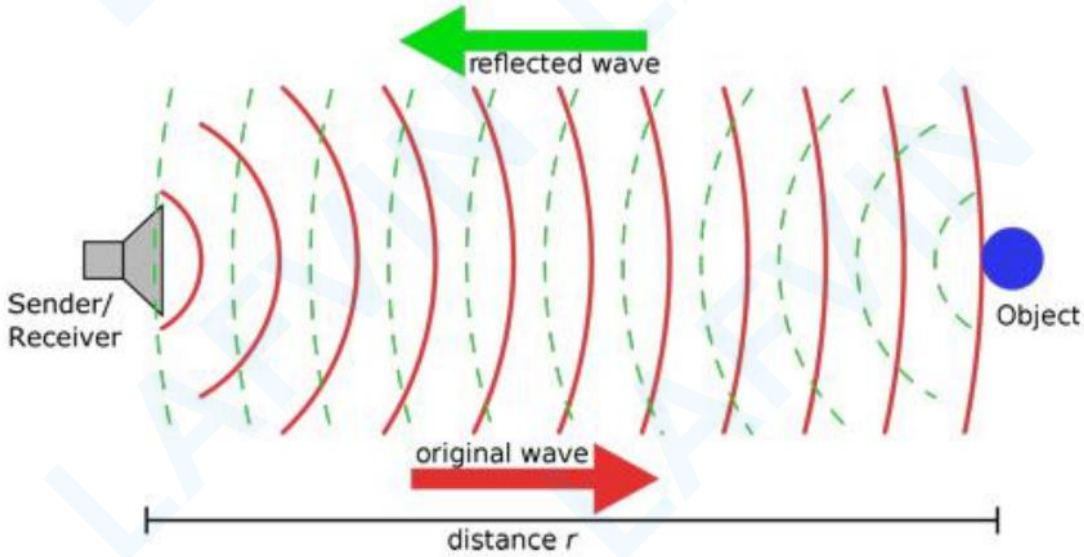
As the following picture shown, it is our ultrasonic module. It has two something like eyes.

One is transmitting end, the other is receiving end.

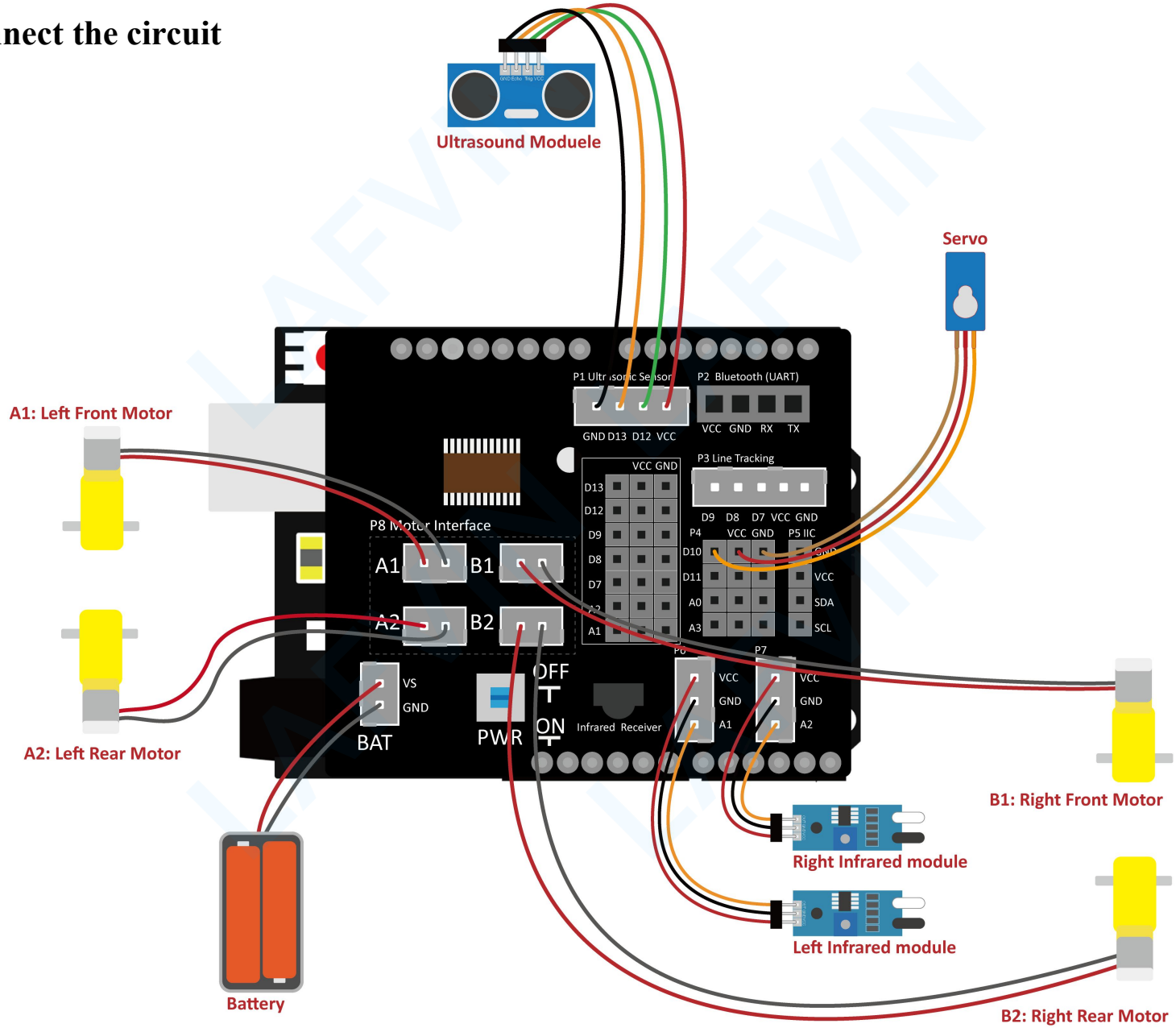


The HC-SR04 ultrasonic sensor uses sonar to determine distance to an object like what bats do. It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package. It comes complete with ultrasonic transmitter and receiver modules.The HC-SR04 or the ultrasonic 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 guiding object is on the left and right sides of the robot car, the infrared obstacle avoidance module can detect and judge the direction of the guiding object, and control the robot car to turn to the guide object.

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 10mm, control the robot car to approach the guide;

When the distance is less than 5mm, control the robot car to move backward and keep a proper distance, 5mm<d<10mm.

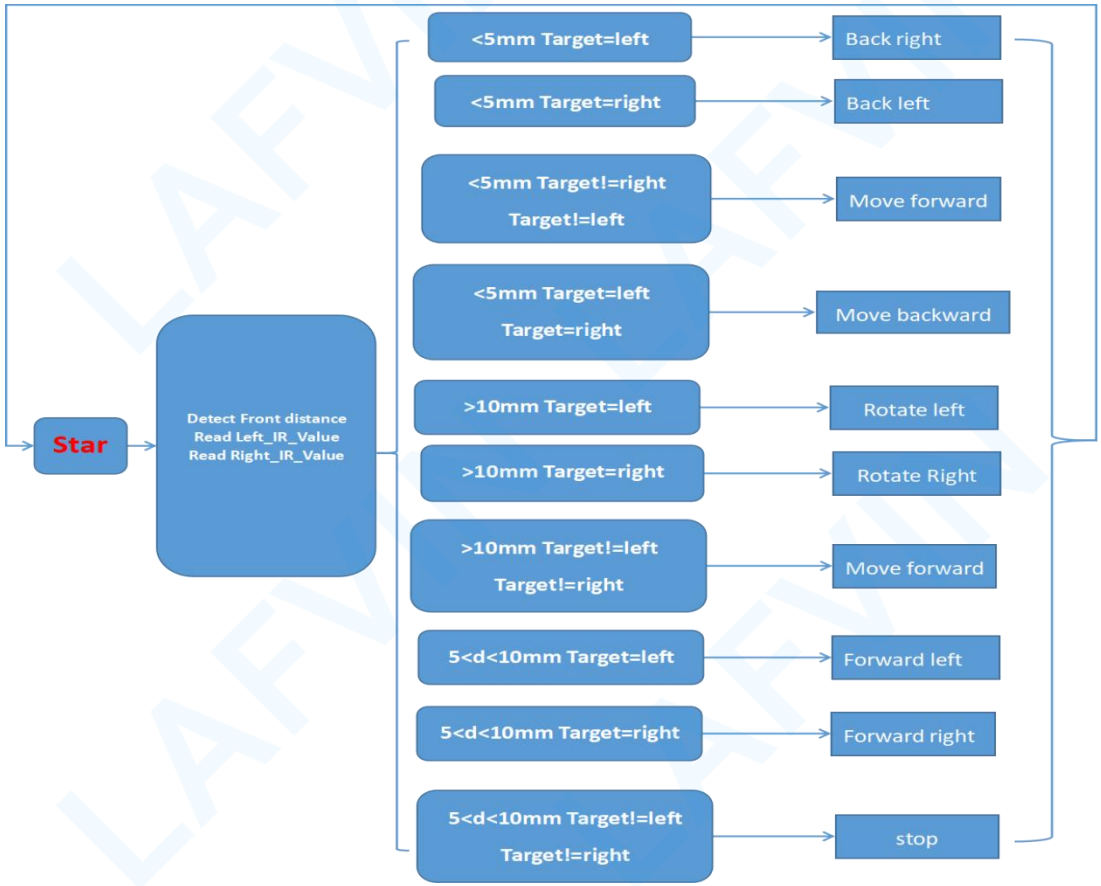
Mixly Code

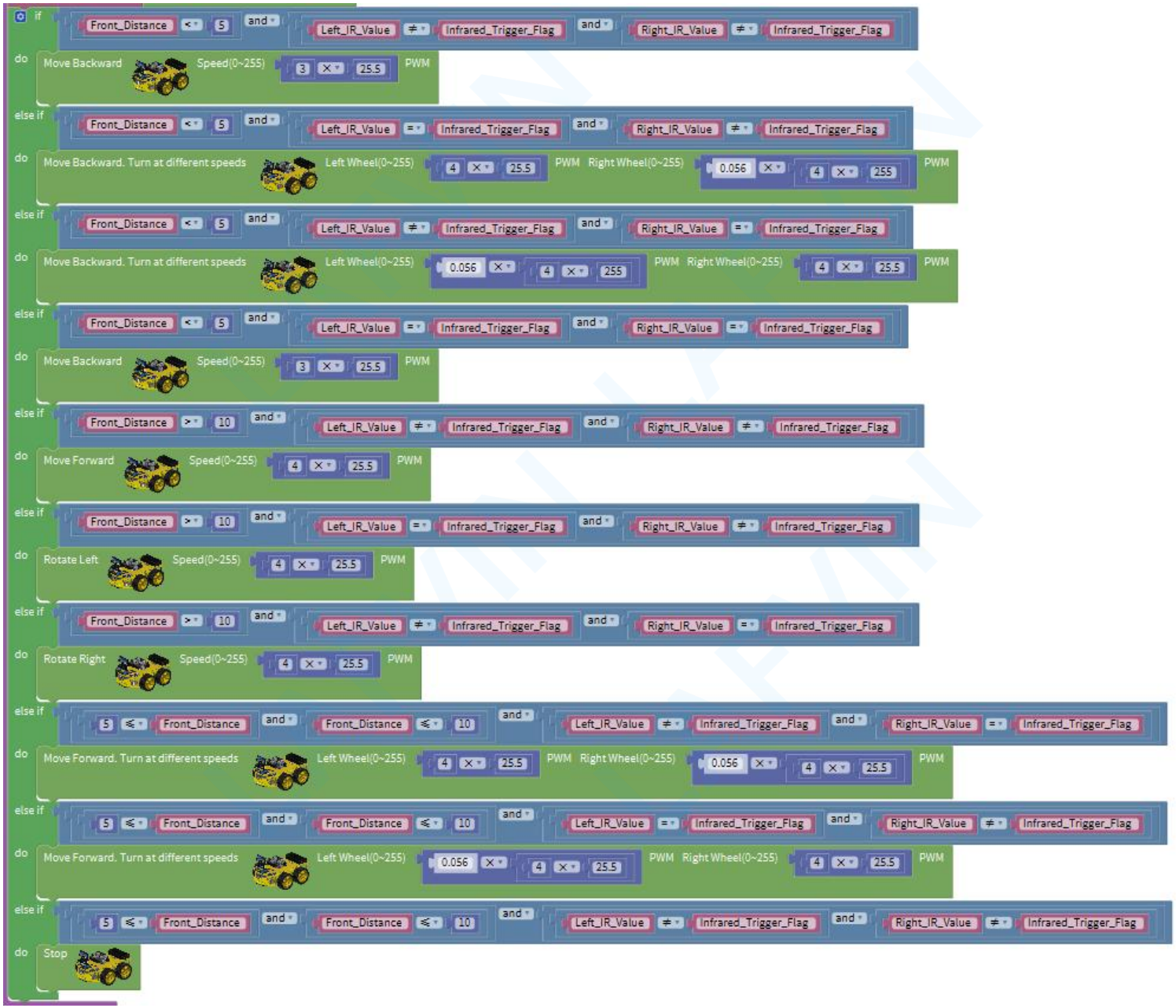
Think about the code logic.Open mixly software.Click “New” to add new project, then start your programming .

if you want to refer to the program we provide.open the reference code for this lesson "**Test_1_Ultrasonic_Follow_Robot_Car.mix**" in the reference materials we provided.

Programming Thinking

The following is the program execution flow chart of the robot car to complete the ultrasonic follow function



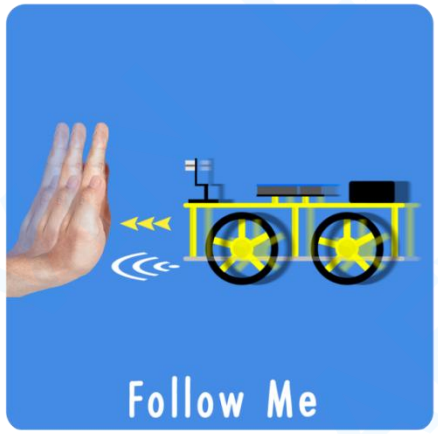

if
Front_Distance < 5 and Left_IR_Value ≠ Infrared_Trigger_Flag and Right_IR_Value ≠ Infrared_Trigger_Flag
do
Move Backward Speed(0~255) 3 × 25.5 PWM
else if
Front_Distance < 5 and Left_IR_Value = Infrared_Trigger_Flag and Right_IR_Value ≠ Infrared_Trigger_Flag
do
Move Backward. Turn at different speeds Left Wheel(0~255) 4 × 25.5 PWM Right Wheel(0~255) 0.056 × 4 × 255 PWM
else if
Front_Distance < 5 and Left_IR_Value ≠ Infrared_Trigger_Flag and Right_IR_Value = Infrared_Trigger_Flag
do
Move Backward. Turn at different speeds Left Wheel(0~255) 0.056 × 4 × 255 PWM Right Wheel(0~255) 4 × 25.5 PWM
else if
Front_Distance < 5 and Left_IR_Value = Infrared_Trigger_Flag and Right_IR_Value = Infrared_Trigger_Flag
do
Move Backward Speed(0~255) 3 × 25.5 PWM
else if
Front_Distance > 10 and Left_IR_Value ≠ Infrared_Trigger_Flag and Right_IR_Value ≠ Infrared_Trigger_Flag
do
Move Forward Speed(0~255) 4 × 25.5 PWM
else if
Front_Distance > 10 and Left_IR_Value = Infrared_Trigger_Flag and Right_IR_Value ≠ Infrared_Trigger_Flag
do
Rotate Left Speed(0~255) 4 × 25.5 PWM
else if
Front_Distance > 10 and Left_IR_Value ≠ Infrared_Trigger_Flag and Right_IR_Value = Infrared_Trigger_Flag
do
Rotate Right Speed(0~255) 4 × 25.5 PWM
else if
5 ≮ Front_Distance and Front_Distance ≮ 10 and Left_IR_Value ≠ Infrared_Trigger_Flag and Right_IR_Value = Infrared_Trigger_Flag
do
Move Forward. Turn at different speeds Left Wheel(0~255) 4 × 25.5 PWM Right Wheel(0~255) 0.056 × 4 × 25.5 PWM
else if
5 ≮ Front_Distance and Front_Distance ≮ 10 and Left_IR_Value = Infrared_Trigger_Flag and Right_IR_Value ≠ Infrared_Trigger_Flag
do
Move Forward. Turn at different speeds Left Wheel(0~255) 0.056 × 4 × 25.5 PWM Right Wheel(0~255) 4 × 25.5 PWM
else if
5 ≮ Front_Distance and Front_Distance ≮ 10 and Left_IR_Value ≠ Infrared_Trigger_Flag and Right_IR_Value ≠ Infrared_Trigger_Flag
do
Stop

Arduino Code

if you want to refer to the program we provide.open the reference code for this lesson "**Test_1_Ultrasonic_Follow_Robot_Car.ino**" in the reference materials we provided.

What will you see

Upload the code to UNO R3 control board,, and turn the POWER switch ON.Note: It is recommended to use a rectangular object box to guide the robot car. Length>10cm.Width>15cm.

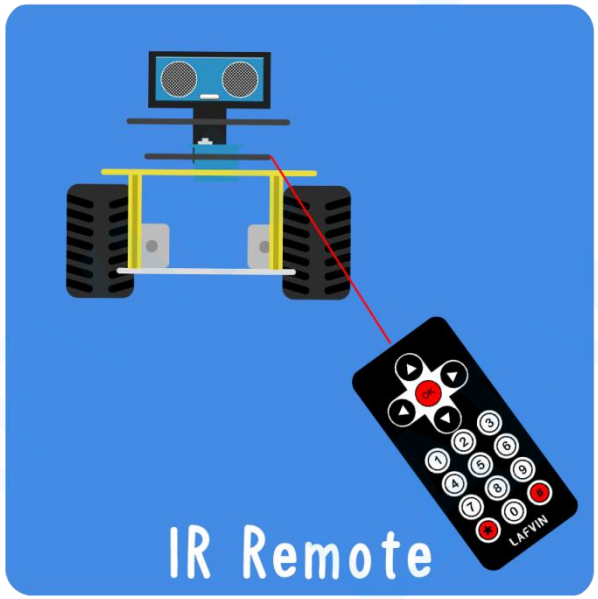


The robot car detects the distance of the object in front through the ultrasonic sensor. When the front object moves, the robot car will follow the movement, keeping the distance between the tank and the object in front between 5cm~8cm.

Lesson 7 Infrared Remote Control Robot Car

Overview

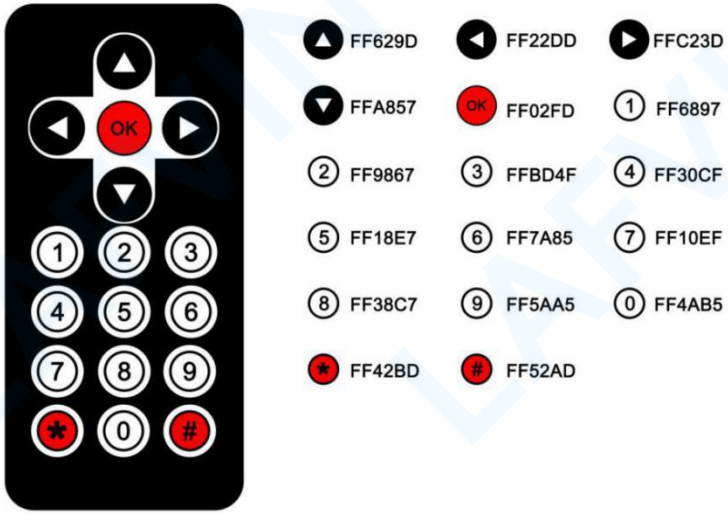
In this lesson, we will learn the infrared remote control, and use the infrared remote control to control the robot car go forward, backward, turn left, turn right,rotate left rotate right.Note: Due to air transportation, the remote control does not contain batteries, you need to prepare a button battery CR2025 before use.

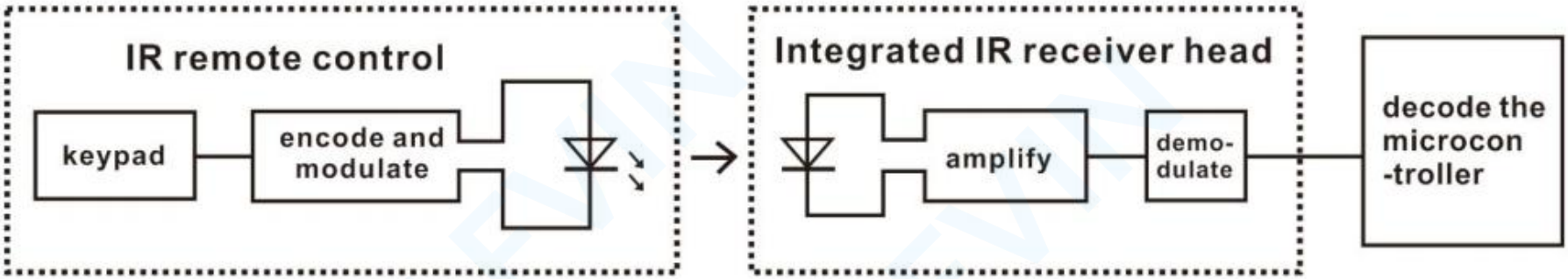


What is an infrared remote control

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.

Remote control key value



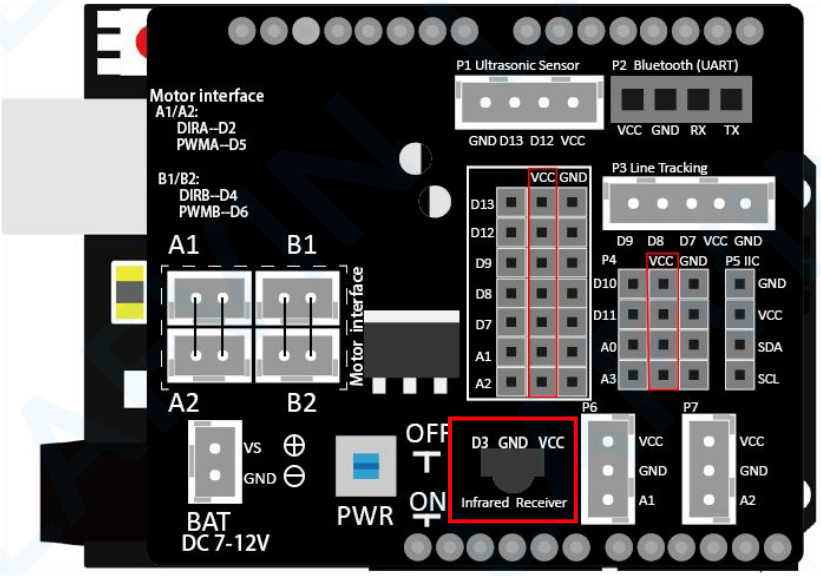


The 38K infrared carrier signal transmitted by an infrared remote controller is encoded by an encoding chip inside the remote controller. It is composed of a pilot code, user code, data code, and data inversion code.The time interval between pulses is used to distinguish whether it is a signal 0 or 1.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



Let's program

Test 1--Infrared Remote Control Robot 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.

Mixly Code

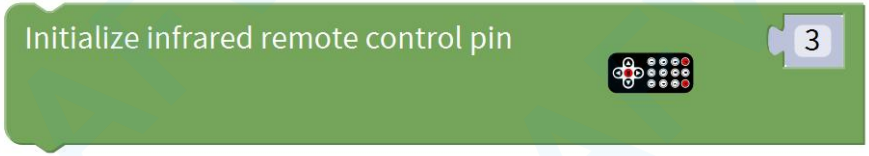
Think about the code logic.Well, figure out the logic,you can have a try to write out the code logic of infrared obstacle avoidance.Open mixly software.Click “New” to add new project, then start your programming .

if you want to refer to the program we provide.open the reference code for this lesson "**Test_1_Infrared_Remote_Control_Robot_Car.mix**" in the reference materials we provided.

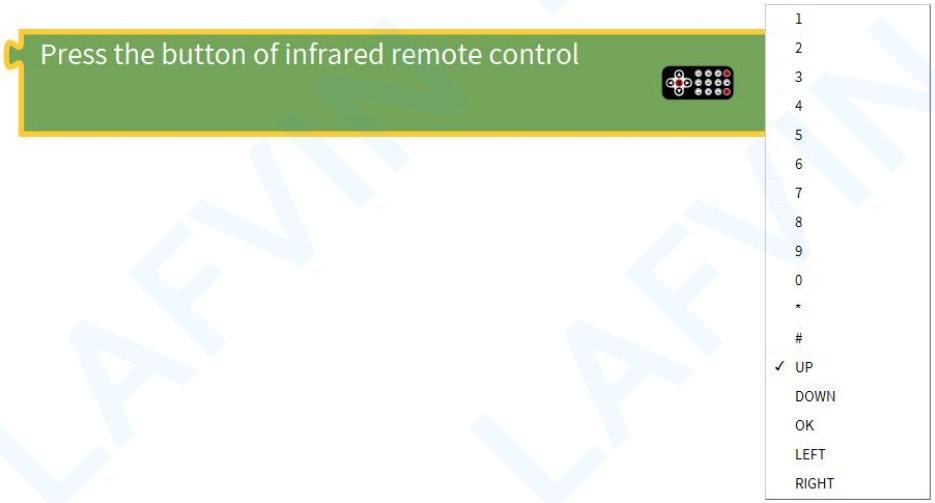
Programming Thinking

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.



Select the button you need to set. When the specified button is pressed, the block outputs true logic.



Arduino Code

if you want to refer to the program we provide.open the reference code for this lesson "**Test_1_Infrared_Remote_Control_Robot_Car.ino**" in the reference materials we provided.

What will you see

Upload the code to UNO R3 control board,and turn the POWER switch ON. Use the infrared remote control to control the tank's direction of advance.Note: Due to air transportation, the remote control does not contain batteries, you need to prepare a button battery CR2025 before use.

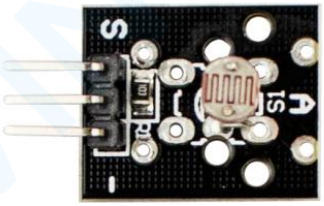
Key on IR remote control							
Robot status	Go forward	Go backward	Turn left	Turn right	Rotate to left	Rotate to right	Stop

Lesson 8 Light Seeking Robot Car

Overview

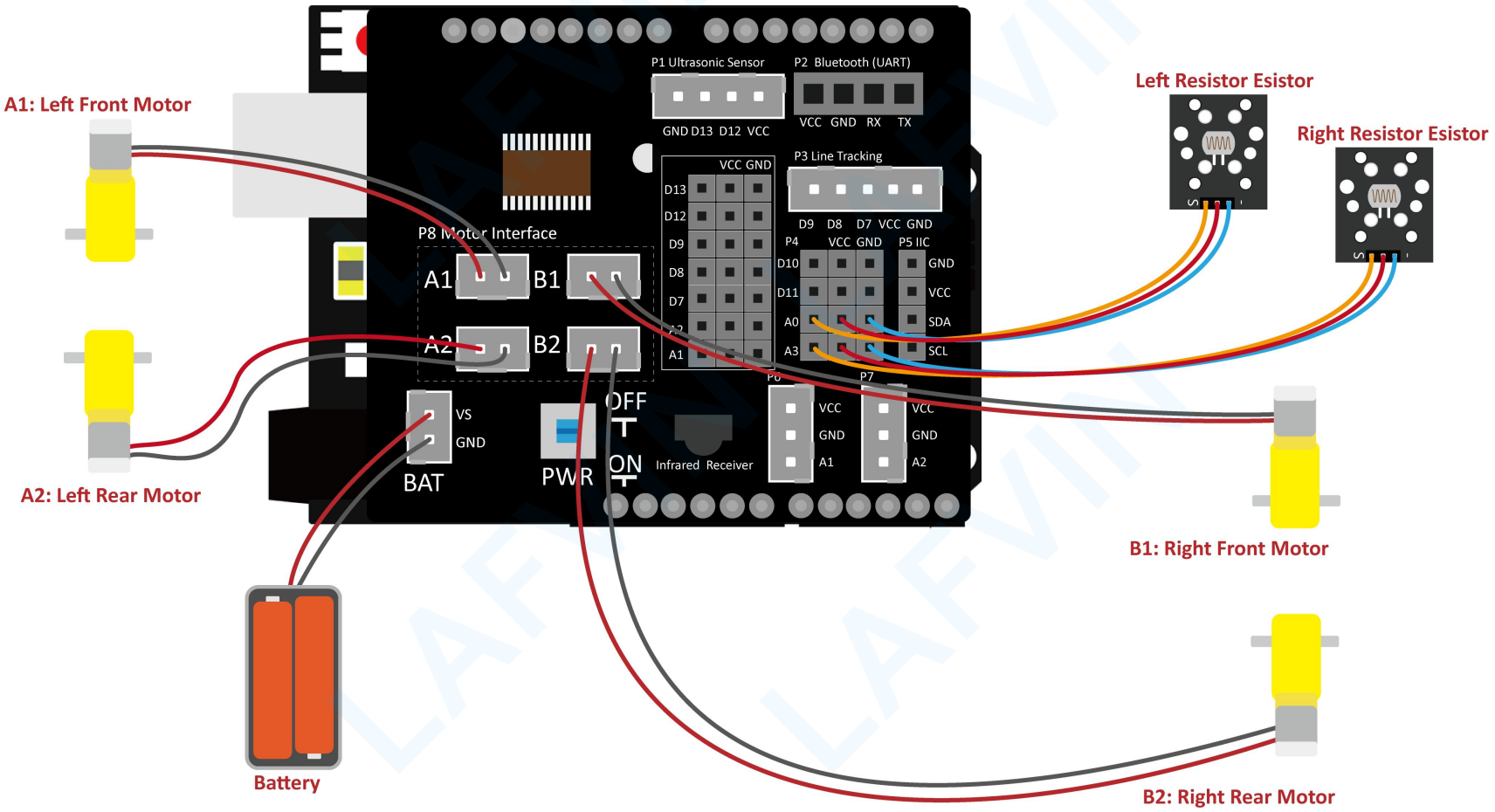
In this lesson, we mainly study two experimental tests.In the first test experiment, we learn the photoelectric sensor and use it to obtain the light intensity in the environment; in the second test experiment, the photoelectric sensor and motor control need to be combined, so that when the car senses strong light, Will move in the direction of the light.

What is photoelectric sensor



The photoelectric sensor (photoresistor) is a resistor semiconductor made by the photoelectric effect. It is very sensitive to ambient light, so its resistance value changes with different light intensities. We use its functions to design circuits and generate photoresistor sensor modules. The signal end of the module is connected to the microcontroller. When the light intensity increases, the resistance decreases, and the voltage of the signal output port of the module decreases, that is, the voltage detected by the analog port of the microcontroller will decrease. Otherwise, when the light intensity decreases, the resistance increases, and the voltage of the signal output port of the module increases, that is, the voltage detected by the analog port of the microcontroller will increase. Therefore, we can use the photoresistor sensor module to read the corresponding analog value and sense the light intensity in the environment. It is usually used in light measurement, control and conversion, and light control circuits.

How to connect the circuit



Let's program

Test 1--Light Seeking Signal

On the module , When the photoresistor is illuminated by strong light , Its resistance value drops rapidly , the current passed increases, the resistance of the photoresistor rises rapidly in a dark environment, the current passed through is reduced, The main control board determines whether there is a light source.

Mixly Code

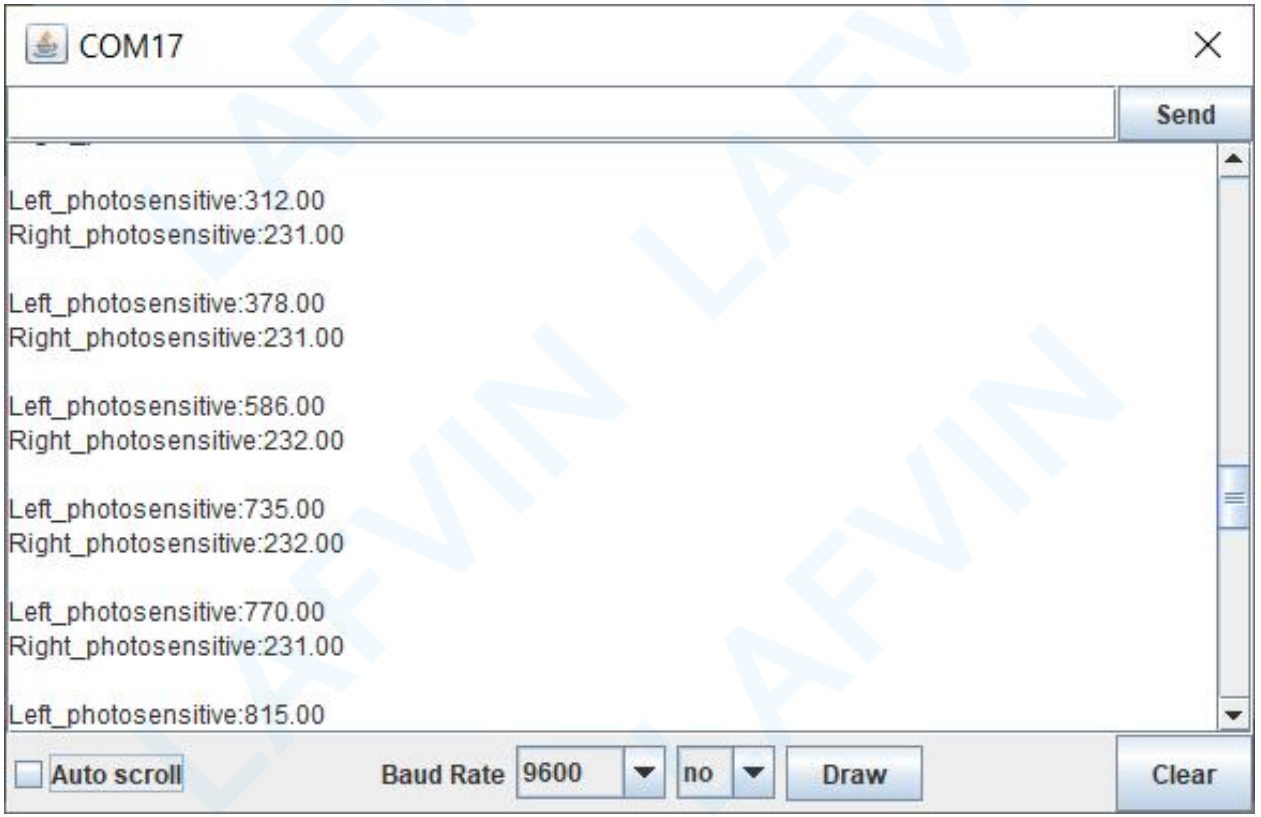
Think about the code logic.Well, figure out the logic,you can have a try to write out the code logic of infrared obstacle avoidance.Open mixly software.Click “New” to add new project, then start your programming .

if you want to refer to the program we provide.open the reference code for this lesson "**Test_1_Light_Seeking_Signal.mix**" in the reference files we provided on CD.

Arduino Code

if you want to refer to the program we provide.open the reference code for this lesson "**Test_1_Light_Seeking_Signal.ino**" in the reference files we provided on CD.

After uploading the program, Use a cover to block the light near the photoresistor, and then observe that the output value of the light-seeking signal on the serial monitor is relatively large, and then remove the obstruction. It is found that the output value of the light-seeking signal on the serial monitor becomes smaller. You can also use the flashlight of the mobile phone to illuminate the photoresistor to observe the change of the output value of the light-seeking signal on the serial monitor.



Test 2--Light Seeking Robot Car

the photoelectric sensor and the motor control need to be combined, so that when the car senses strong light, it will move in the direction of the light.

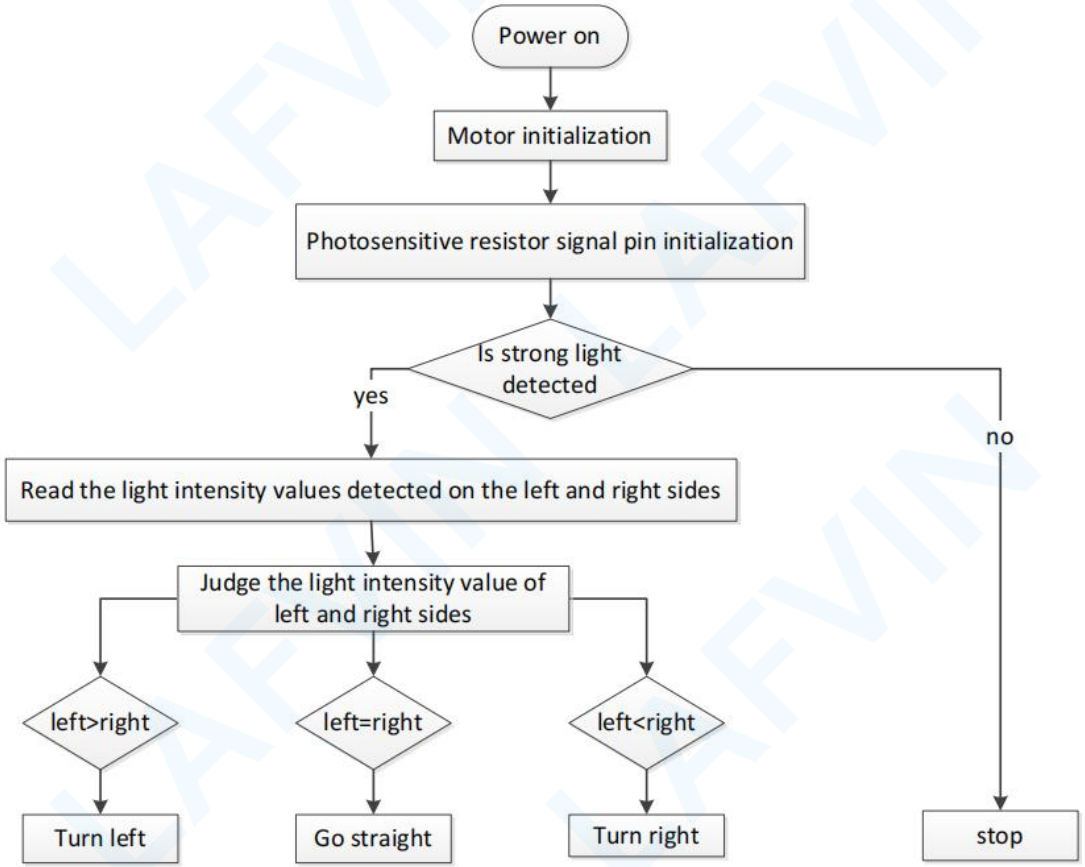
Mixly Code

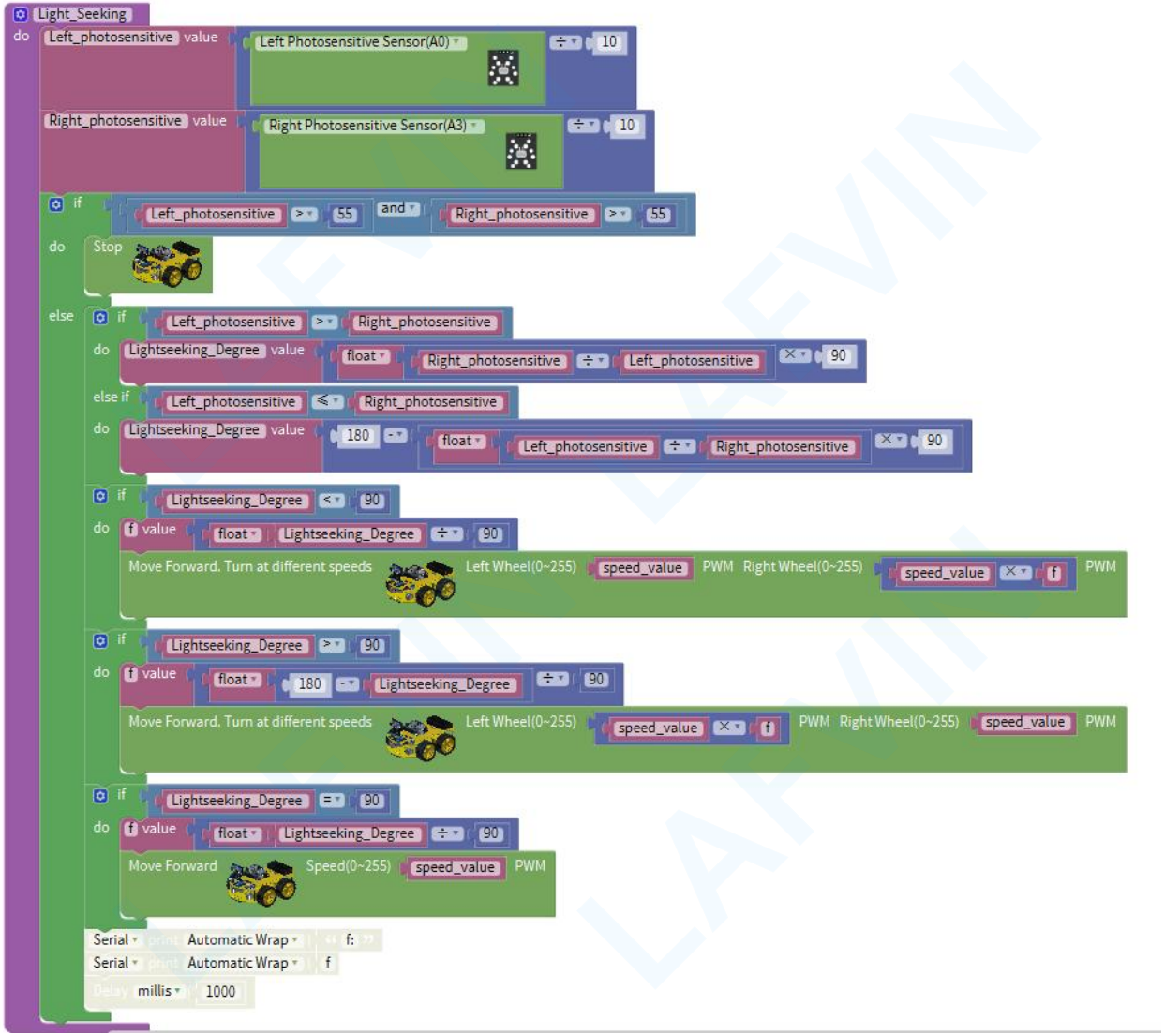
Think about the code logic.Well, figure out the logic,you can have a try to write out the code logic of infrared obstacle avoidance.Open mixly software.Click “New” to add new project, then start your programming .

if you want to refer to the program we provide.open the reference code for this lesson "**Test_2_Light_Seeking_Robot_Car.mix**" in the reference files we provided on CD.

Programming Thinking

The specific software design idea is as shown in Figure




Light_Seeking
do
Left_photosensitive value
Left Photosensitive Sensor(A0)
÷ 10
Right_photosensitive value
Right Photosensitive Sensor(A3)
÷ 10
if
Left_photosensitive > 55 and Right_photosensitive > 55
do
Stop
else
if
Left_photosensitive > Right_photosensitive
do
Lightseeking_Degree value
float
Right_photosensitive ÷ Left_photosensitive
× 90
else if
Left_photosensitive ⩽ Right_photosensitive
do
Lightseeking_Degree value
180 -
float
Left_photosensitive ÷ Right_photosensitive
× 90
if
Lightseeking_Degree < 90
do
f value
float
Lightseeking_Degree ÷ 90
Move Forward. Turn at different speeds
Left Wheel(0~255)
speed_value
PWM
Right Wheel(0~255)
speed_value × f
PWM
if
Lightseeking_Degree > 90
do
f value
float
180 - Lightseeking_Degree
÷ 90
Move Forward. Turn at different speeds
Left Wheel(0~255)
speed_value × f
PWM
Right Wheel(0~255)
speed_value
PWM
if
Lightseeking_Degree = 90
do
f value
float
Lightseeking_Degree ÷ 90
Move Forward
Speed(0~255)
speed_value
PWM
Serial print Automatic Wrap "f:"
Serial print Automatic Wrap f
Delay millis 1000

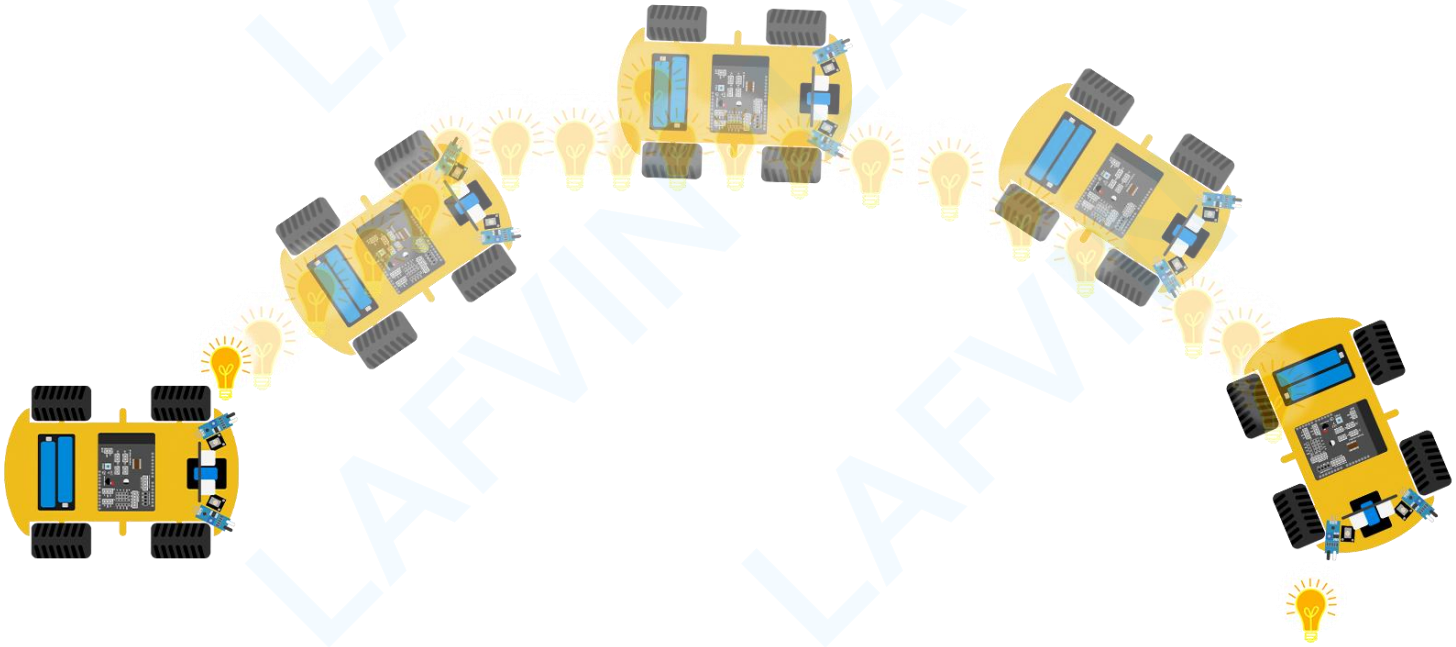
Arduino Code

if you want to refer to the program we provide.open the reference code for this lesson "**Test_2_Light_Seeking_Robot_Car.ino**" in the reference files we provided on CD.

```
void Light_Seeking() {
  Left_photosensitive = analogRead(A0) / 10;
  Right_photosensitive = analogRead(A3) / 10;
  if (Left_photosensitive > 55 && Right_photosensitive > 55) {
    digitalWrite(2,LOW);
    analogWrite(5,0);
    digitalWrite(4,HIGH);
    analogWrite(6,0);
  } else {
    if (Left_photosensitive > Right_photosensitive) {
      Lightseeking_Degree = ((float)(Right_photosensitive / Left_photosensitive)) * 90;
    } else if (Left_photosensitive <= Right_photosensitive) {
      Lightseeking_Degree = 180 - ((float)(Left_photosensitive / Right_photosensitive)) * 90;
    }
    if (Lightseeking_Degree < 90) {
      f = ((float)(Lightseeking_Degree)) / 90;
      digitalWrite(2,HIGH);
      analogWrite(5,speed_value);
      digitalWrite(4,LOW);
      analogWrite(6,(speed_value * f));
    }
    if (Lightseeking_Degree > 90) {
      f = ((float)(180 - Lightseeking_Degree)) / 90;
      digitalWrite(2,HIGH);
      analogWrite(5,(speed_value * f));
      digitalWrite(4,LOW);
      analogWrite(6,speed_value);
    }
    if (Lightseeking_Degree == 90) {
      f = ((float)(Lightseeking_Degree)) / 90;
      digitalWrite(2,HIGH);
      analogWrite(5,speed_value);
      digitalWrite(4,LOW);
      analogWrite(6,speed_value);
    }
```

What will you see

Upload the code to UNO R3 control board,and turn the POWER switch ON. Test in a dark environment and turn off the lights in the room. The light of the flashlight faces the photoelectric sensor on the right, and the robot car turns right. The light of the flashlight faces the photoelectric sensor on the left, and the robot car turns left.



Lesson 9 Bluetooth Control Robot Car

Overview

In this lesson, we mainly study two experimental tests.

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

In Test 2, use the app remote control to control the direction of the robot car.


Remote Control
Gravity Sensor

What is BLE

The meaning of BLE is low energy Bluetooth, which has low working power consumption and long standby time. However, BLE 4.0 is a new branch and is not backward compatible.

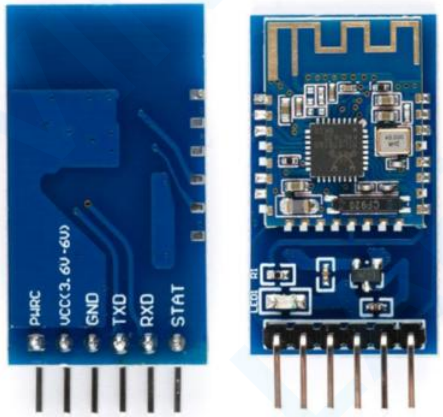
Supported Android devices: Android phones equipped with Bluetooth 4.0 and upgraded to Android 4.3 Or higher system.

You need to complete the search and connection of BLE devices through the App.

BLE 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.



JDY-16

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)

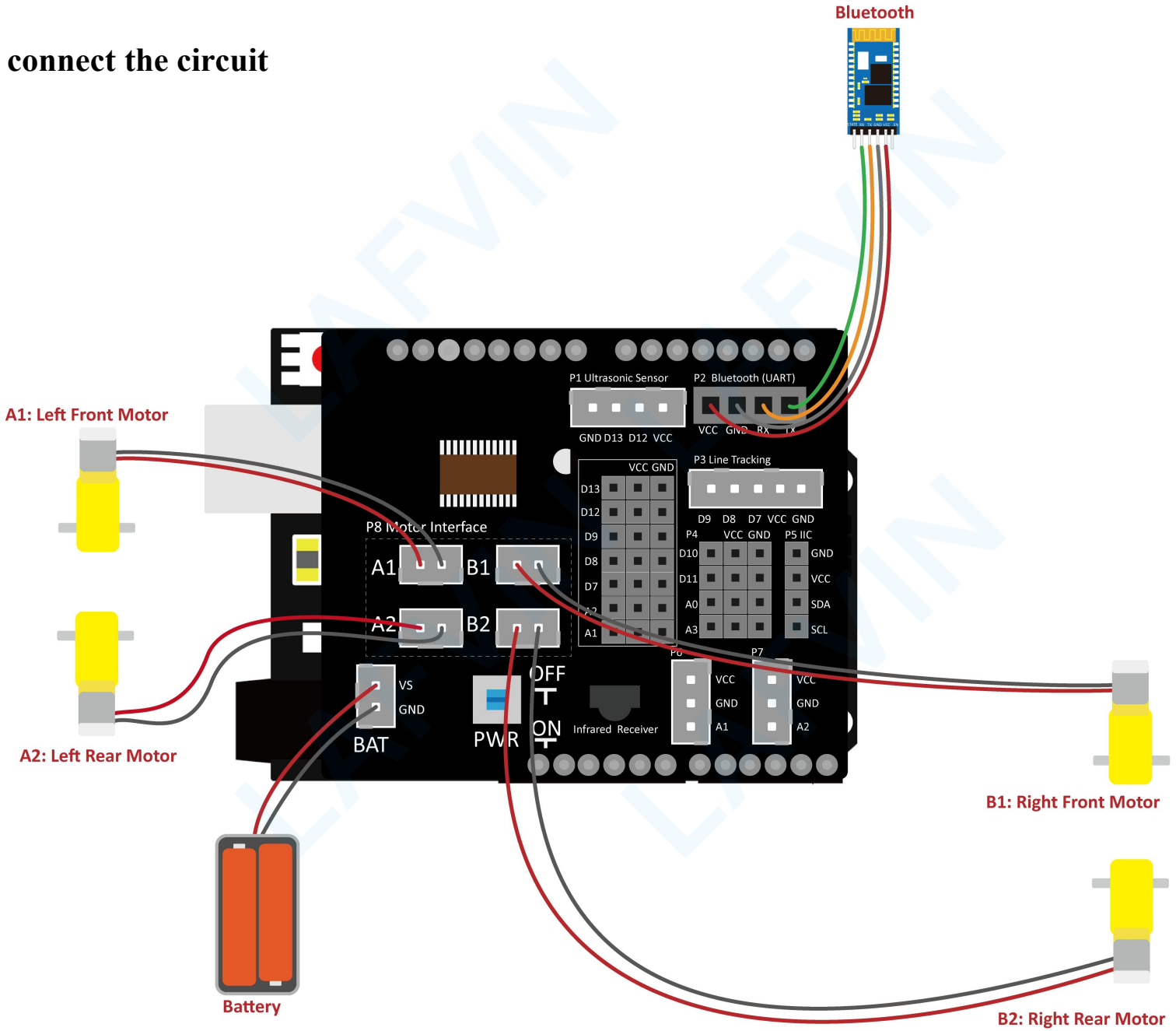
Light sleep state current: <300uA (with broadcast)

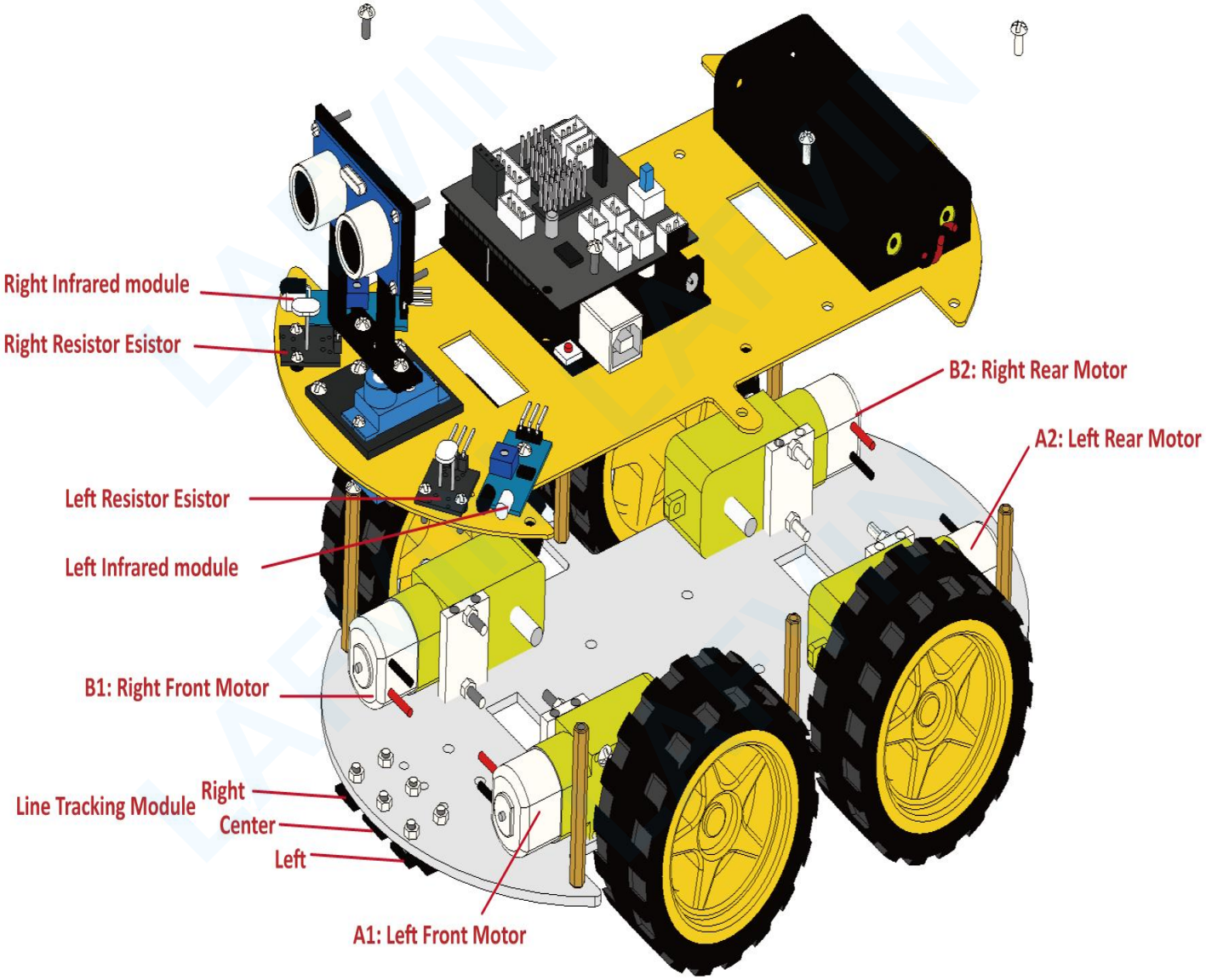
Deep sleep current: 1.8uA (no broadcast)

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

STM welding temperature: <300℃

How to connect the circuit



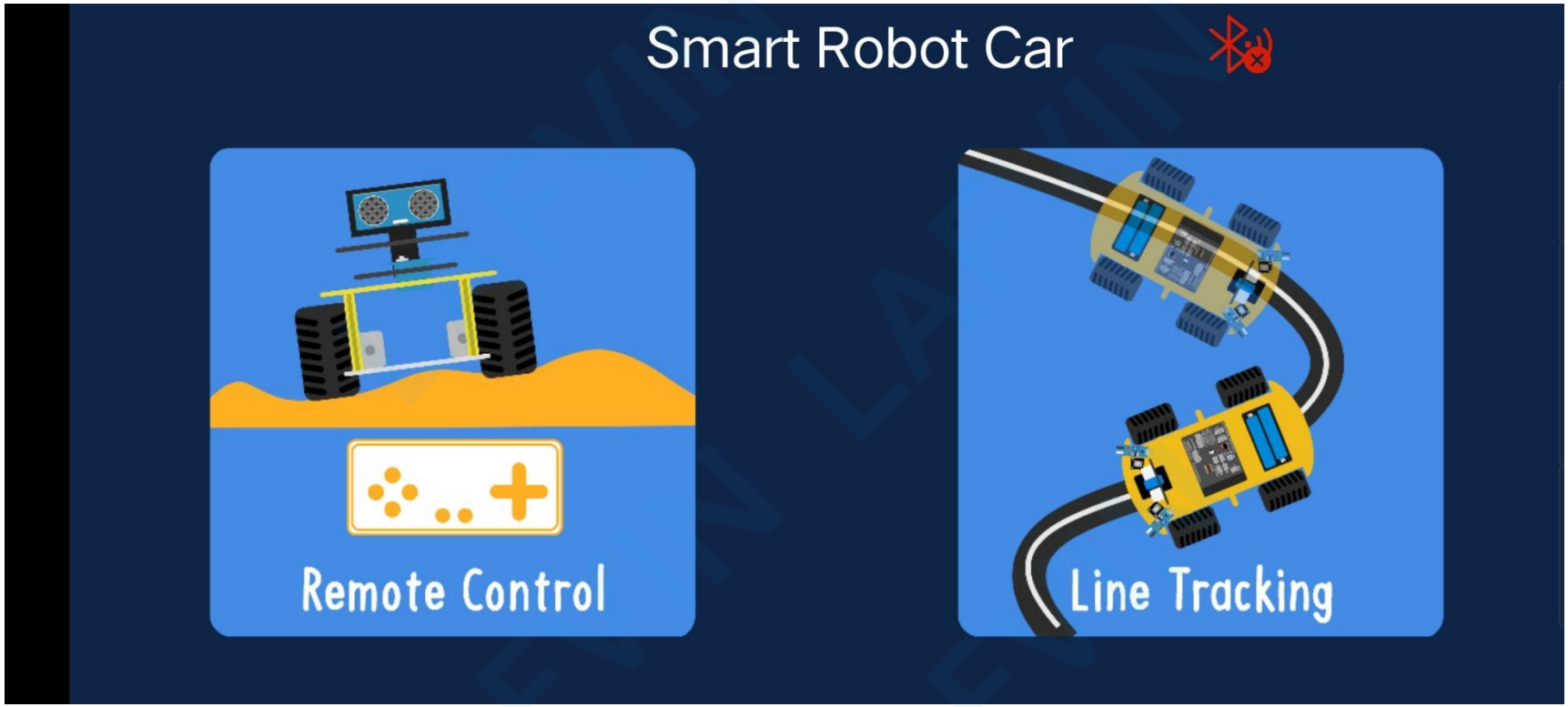

Right Infrared module
Right Resistor Esistor
B2: Right Rear Motor
A2: Left Rear Motor
Left Resistor Esistor
Left Infrared module
B1: Right Front Motor
Line Tracking Module
Right
Center
Left
A1: Left Front Motor

How to connect APP with bluetooth module

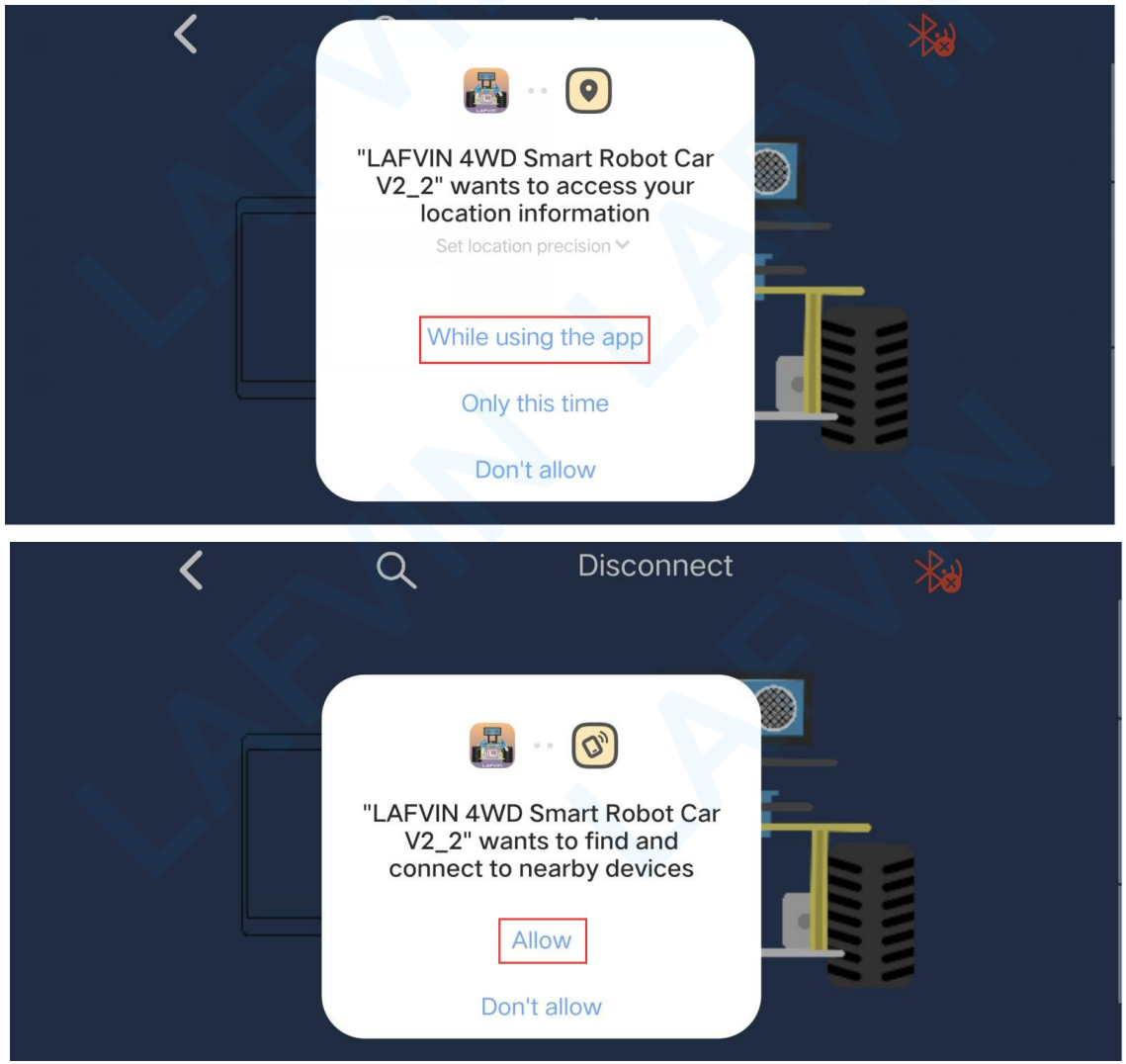
Firstly, copy the “LAFVIN_4WD_Smart_Robot_Car_BLE_V2_2.apk” file from the APP folder to your mobile phone and install it into an application software.Android phones equipped with Bluetooth 4.0 and upgraded to Android 4.3 Or higher system.

LAFVIN 4WD Robot Car

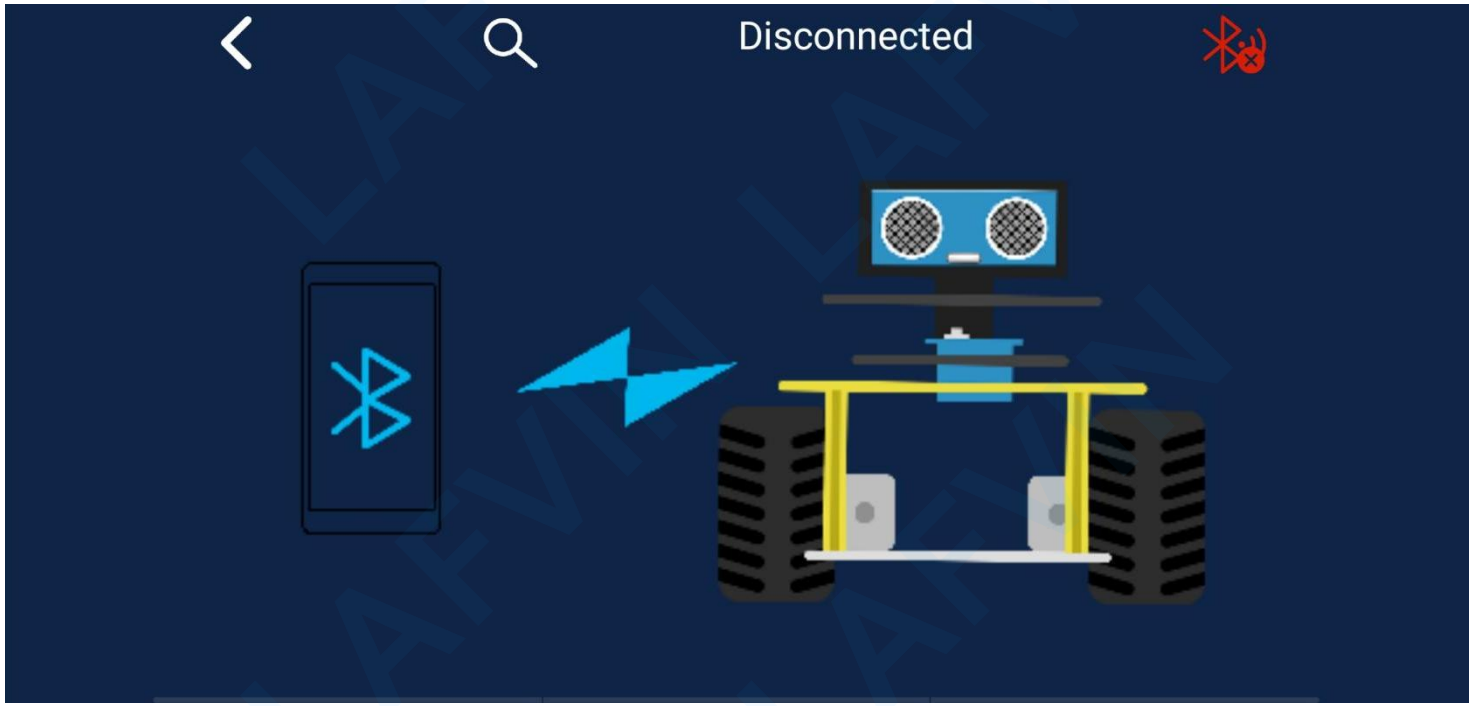
Open the app, you will see the following interface, swipe left and right to switch pages.



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.

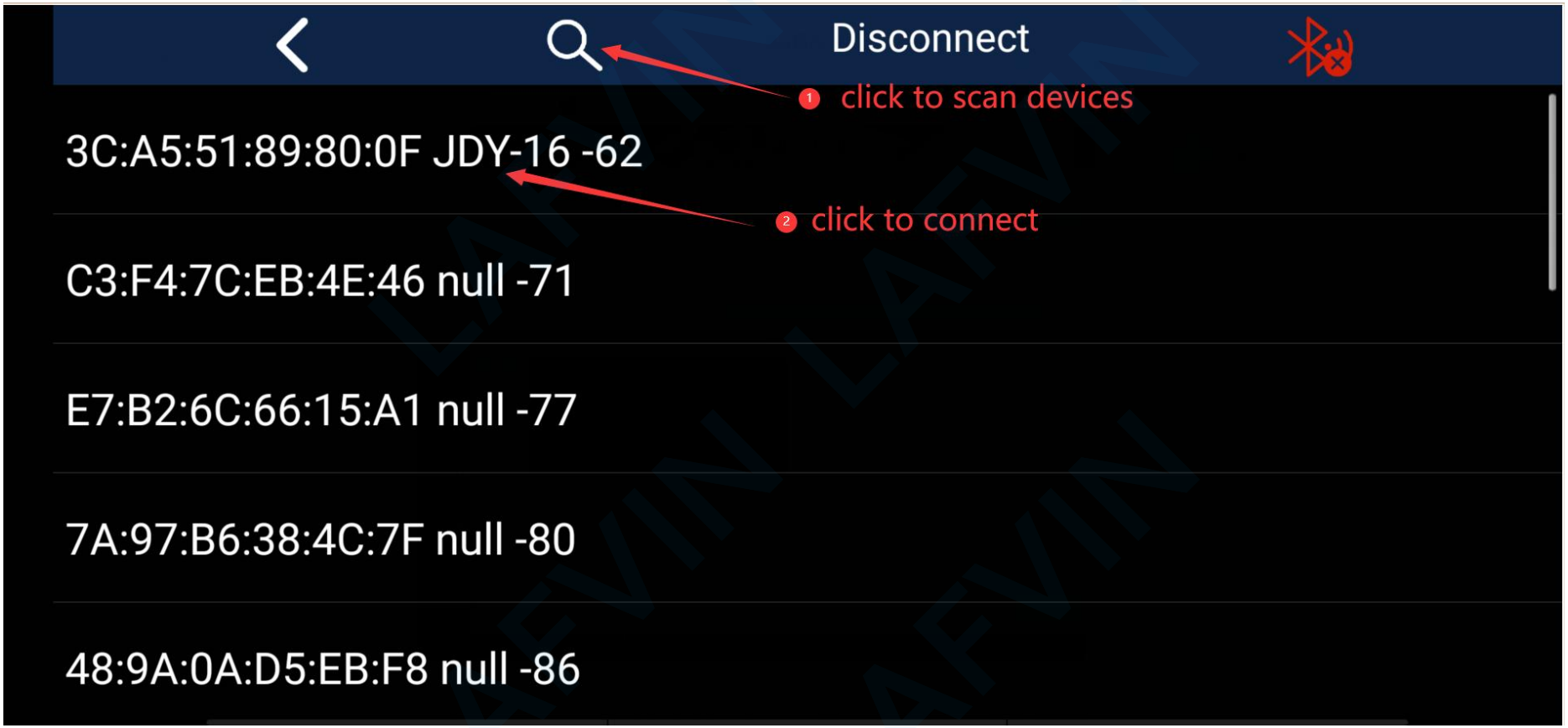


Install the Bluetooth module to the Arduino UNO Motor Driver Shied, turn on the power switch, and the indicator light of the Bluetooth module starts to flash, which means it is in a connectable state.Click the Bluetooth icon  to enter the Bluetooth pairing function page.



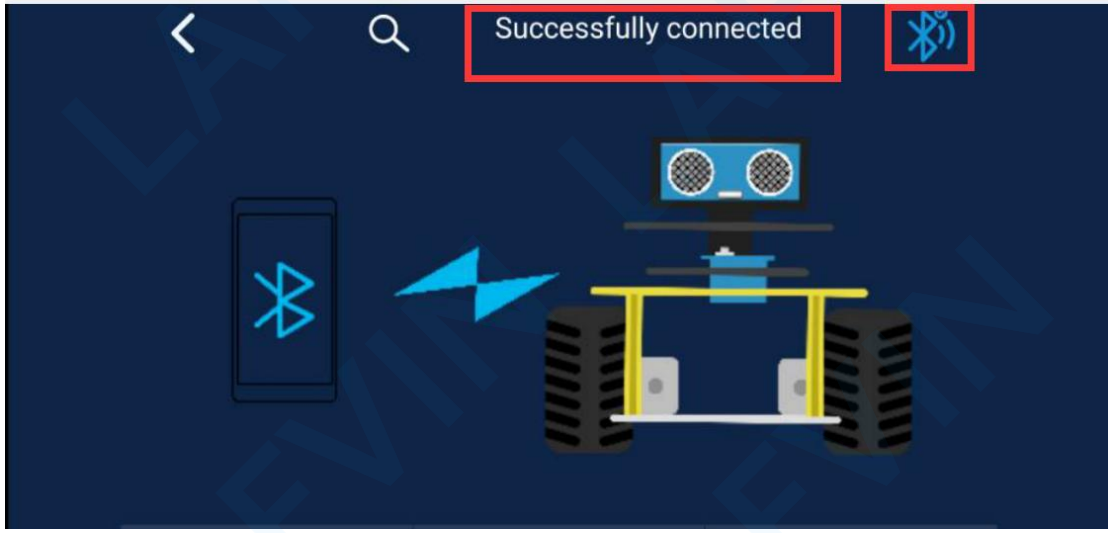
and then click device scan  ,After successfully searching for the device JDY-16, click the device address in the device list

to automatically connect.



Report: Successful connected.This means that the correct device has been connected

Report: Device Error. This means that the connected Bluetooth device does not belong to the robot car, search for JDY-16 and reconnect.



If App does not find the "JDY-16" device, restart the Bluetooth module (unplug the Bluetooth module from the Arduino UNO Motor Driver Shied, disconnect the power of the Bluetooth module; then reinstall it to the Arduino UNO Motor Driver Shied, connect to the power supply).the red LED flashes, which means it is in a connectable state. Then search for

Bluetooth devices again in the APP.

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 .

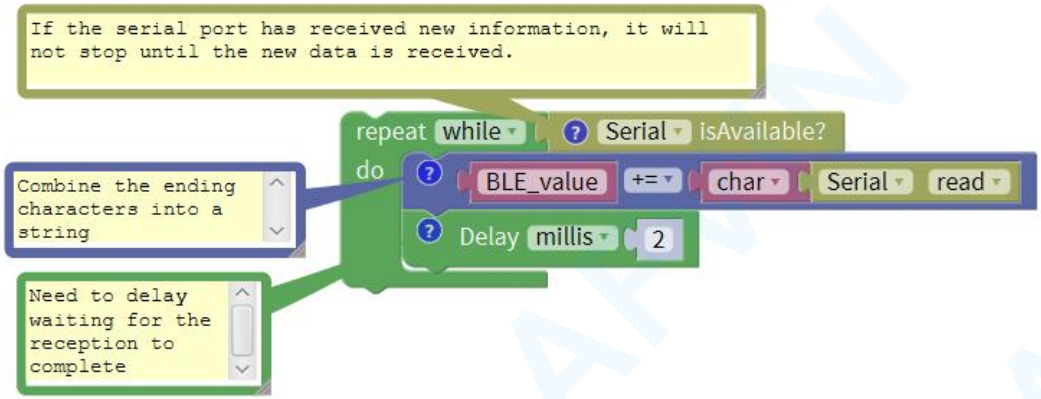
Mixly Code

Think about the code logic.Click “New” to add new project, then start your programming .

if you want to refer to the program we provide.open the reference code for this lesson "**Test_1_Bluetooth_Module_Receives_Information.mix**" in the reference materials we provided.

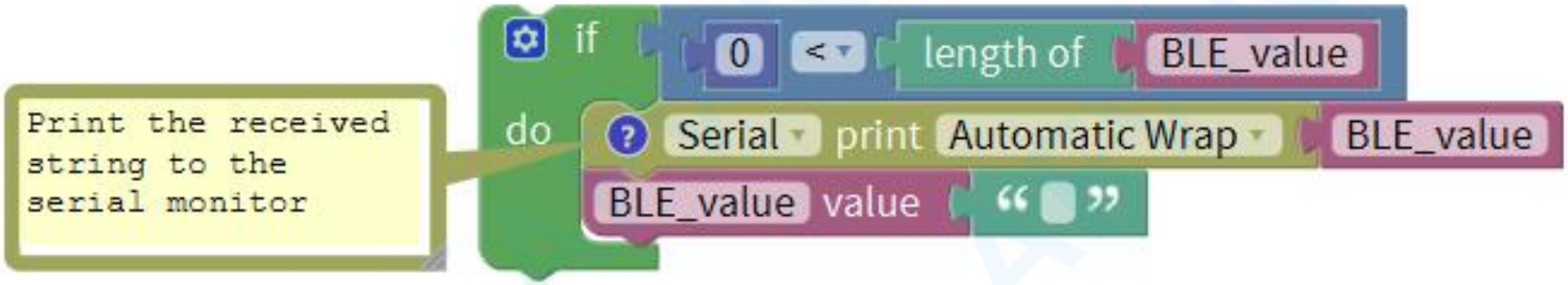
Programming Thinking

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 Bluetooth module is 9600 by default.



If the serial port has received new information, it will not stop until the new data is received.

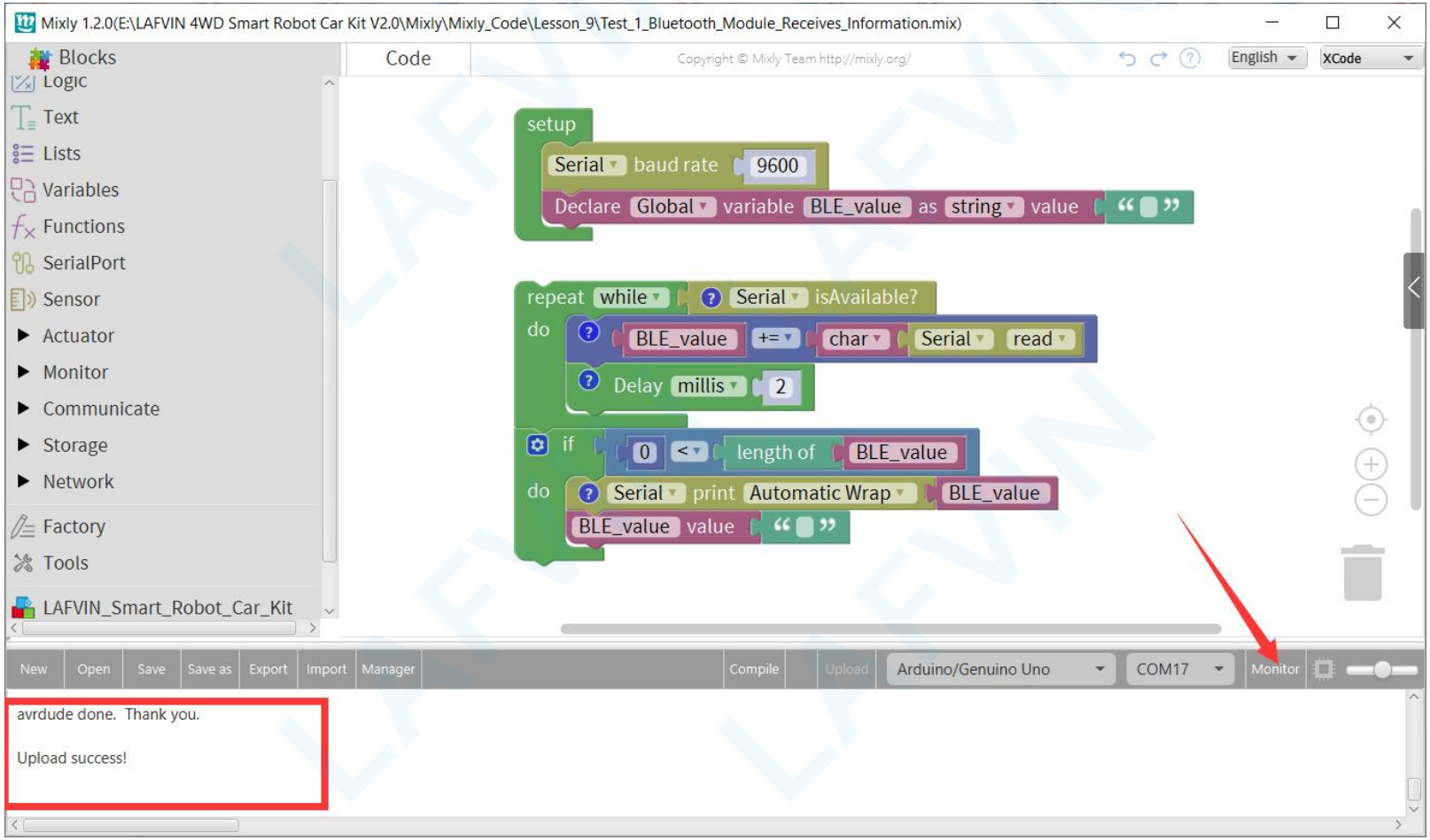
Combine the ending characters into a string.Need to delay waiting for the reception to complete.



If 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.

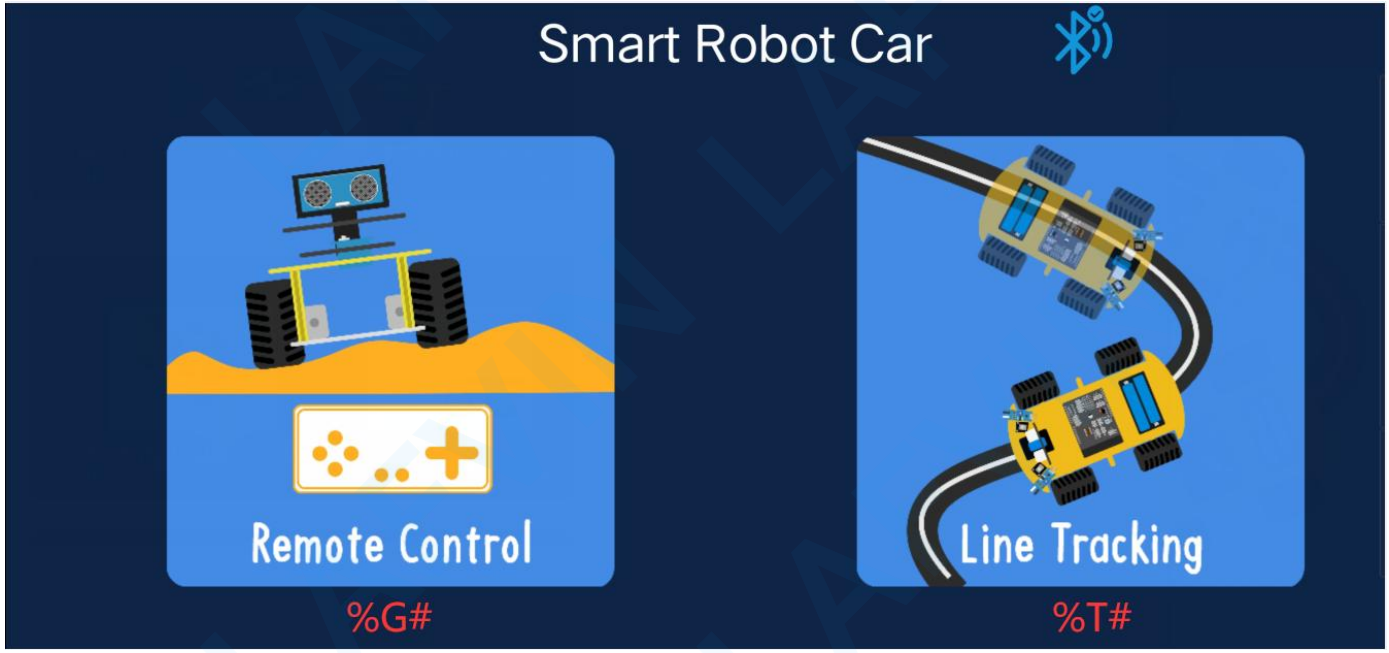


After the Bluetooth module of the mobile phone and the Bluetooth module of the robot car are successfully connected，

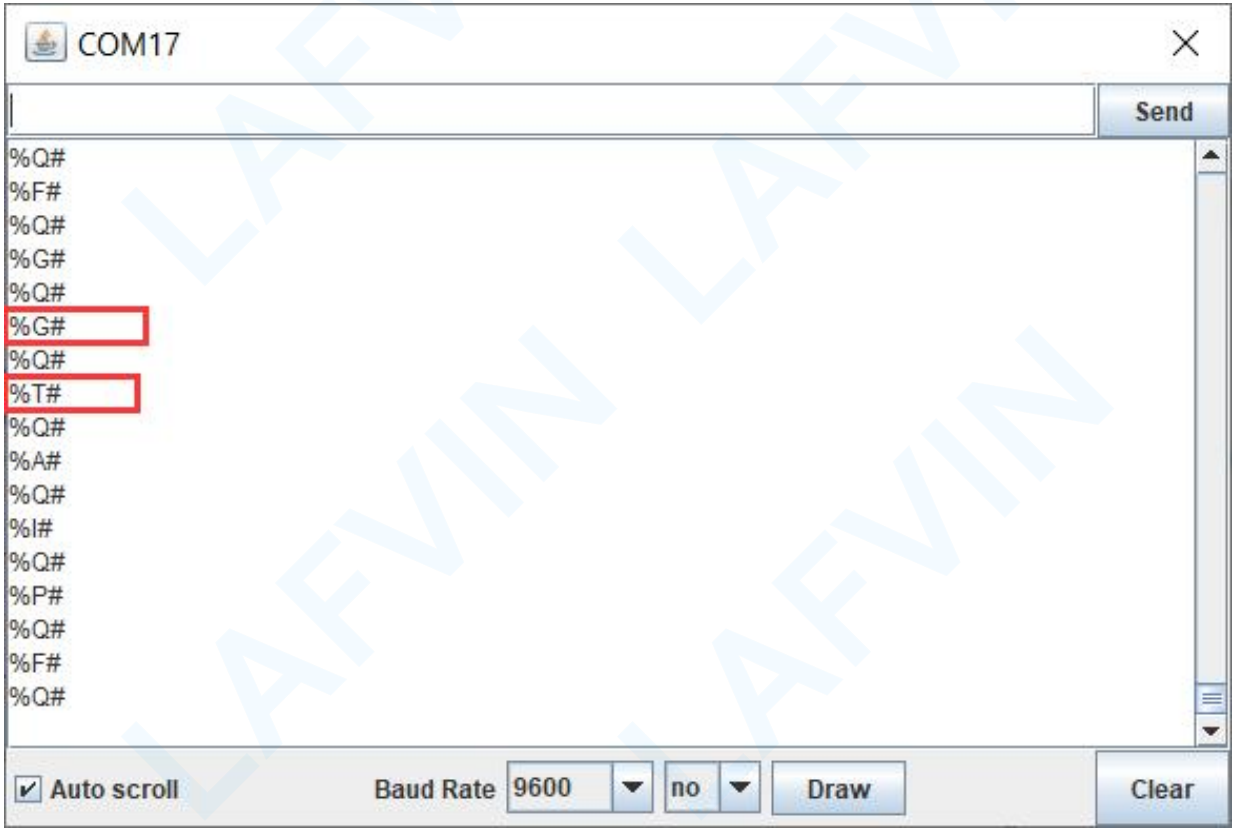
When you press the up button  , the character received by Bluetooth is “%G#”,

When you press the down button  , the character received by Bluetooth is“%T#”.

Similarly, you can use this method to test the commands of other buttons in the APP.



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 Bluetooth module is 9600 by default.



Arduino Code

if you want to refer to the program we provide.open the reference code for this lesson "**Test_1_Bluetooth_Module_Receives_Information.ino**" in the reference materials we provided.

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 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);
   }
```

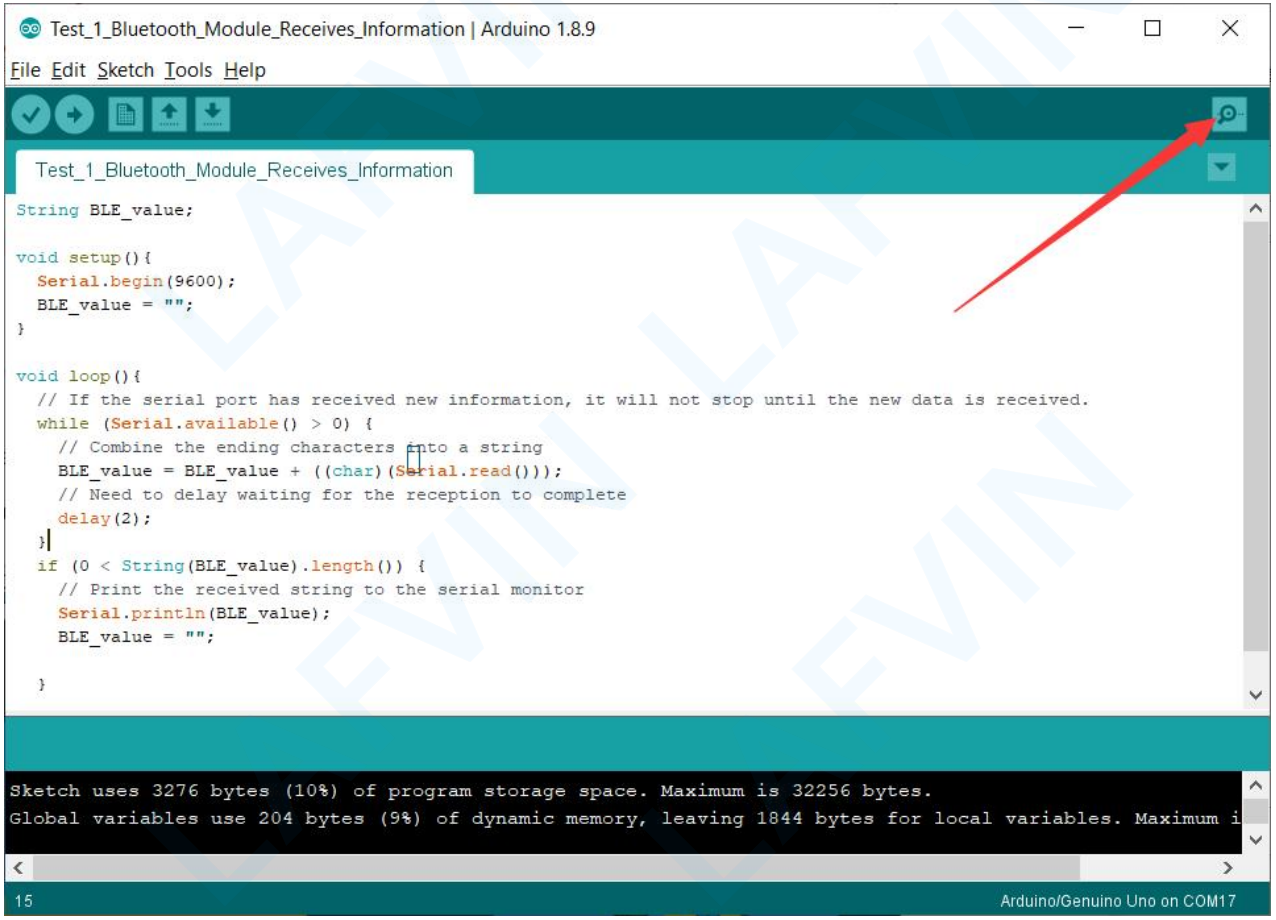
If the serial port has received new information, it will not stop until the new data is received.Combine the ending characters into a string.Need to delay waiting for the reception to complete.

```
   if (0 < String(BLE_value).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.

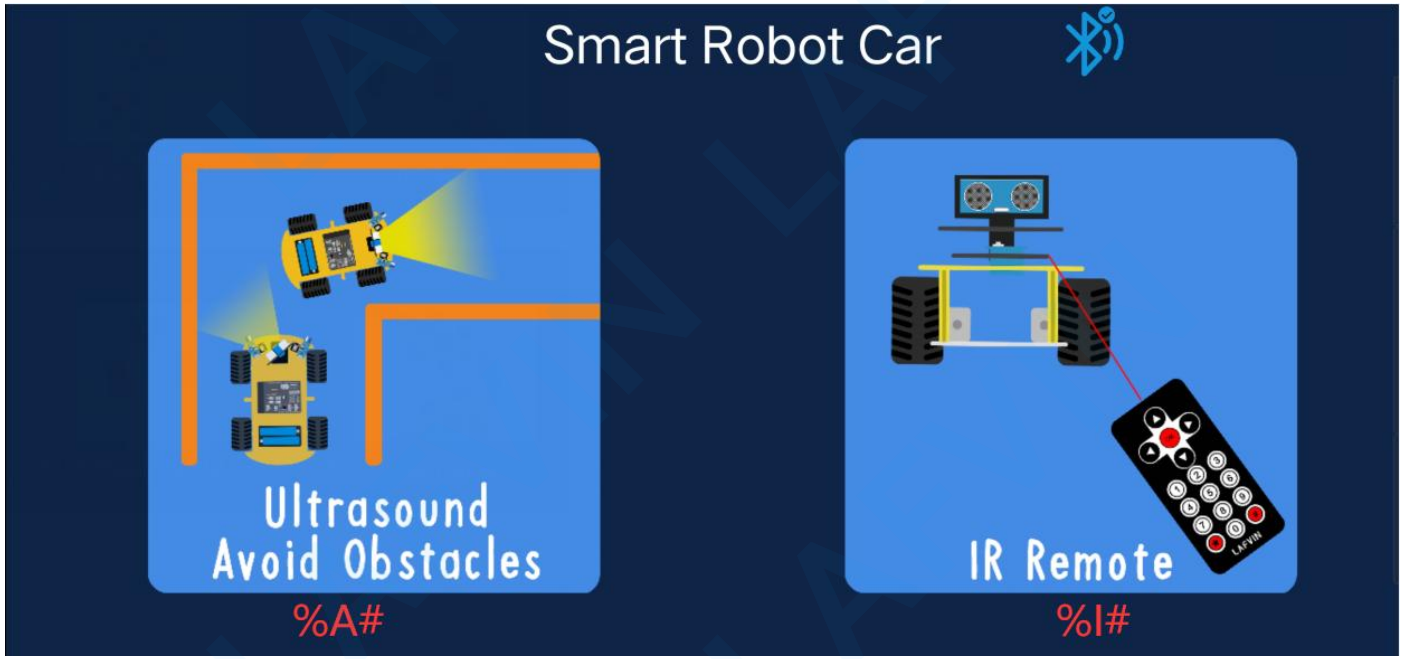


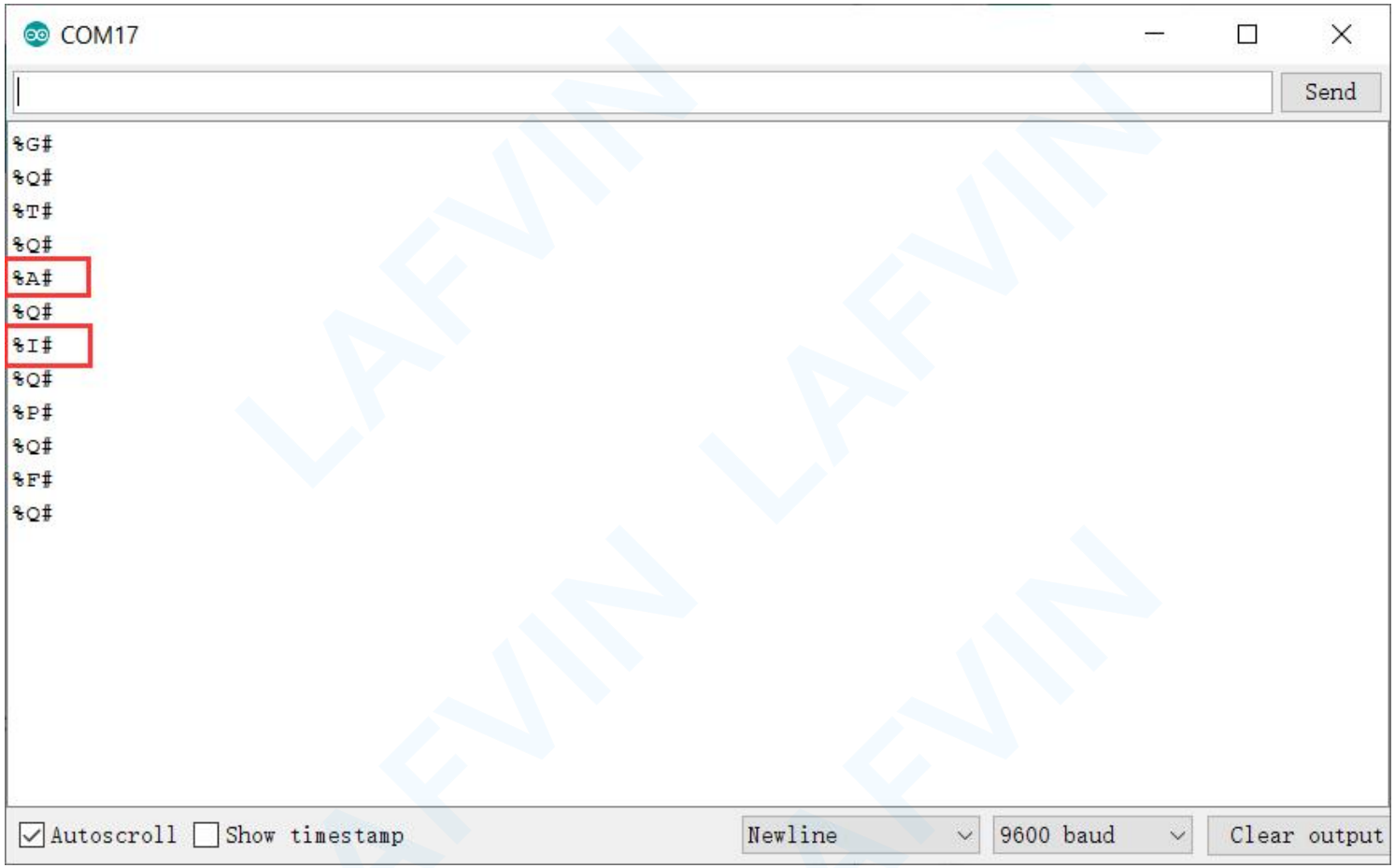
After the Bluetooth module of the mobile phone and the Bluetooth module of the robot car are successfully connected，

When you press the up button  , the character received by Bluetooth is “%A#”,

When you press the down button  , the character received by Bluetooth is“%I#”.

Similarly, you can use this method to test the commands of other buttons in the APP.



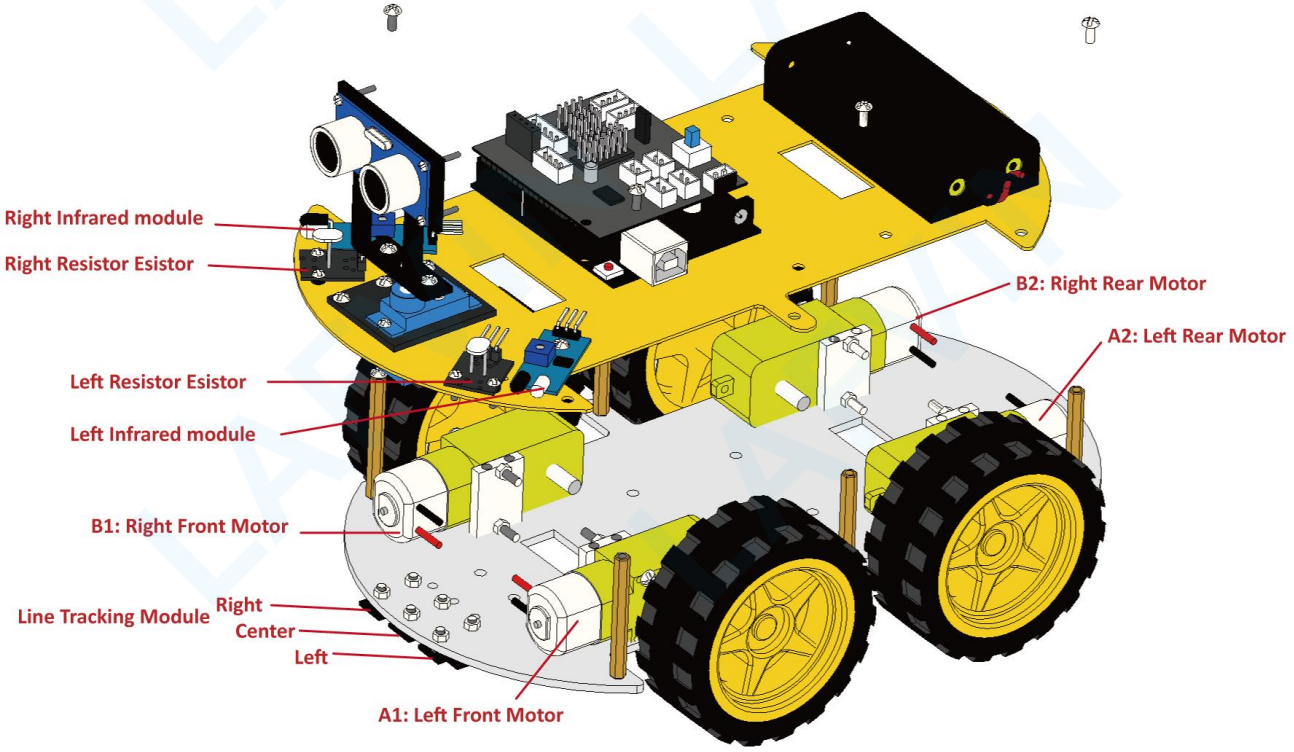


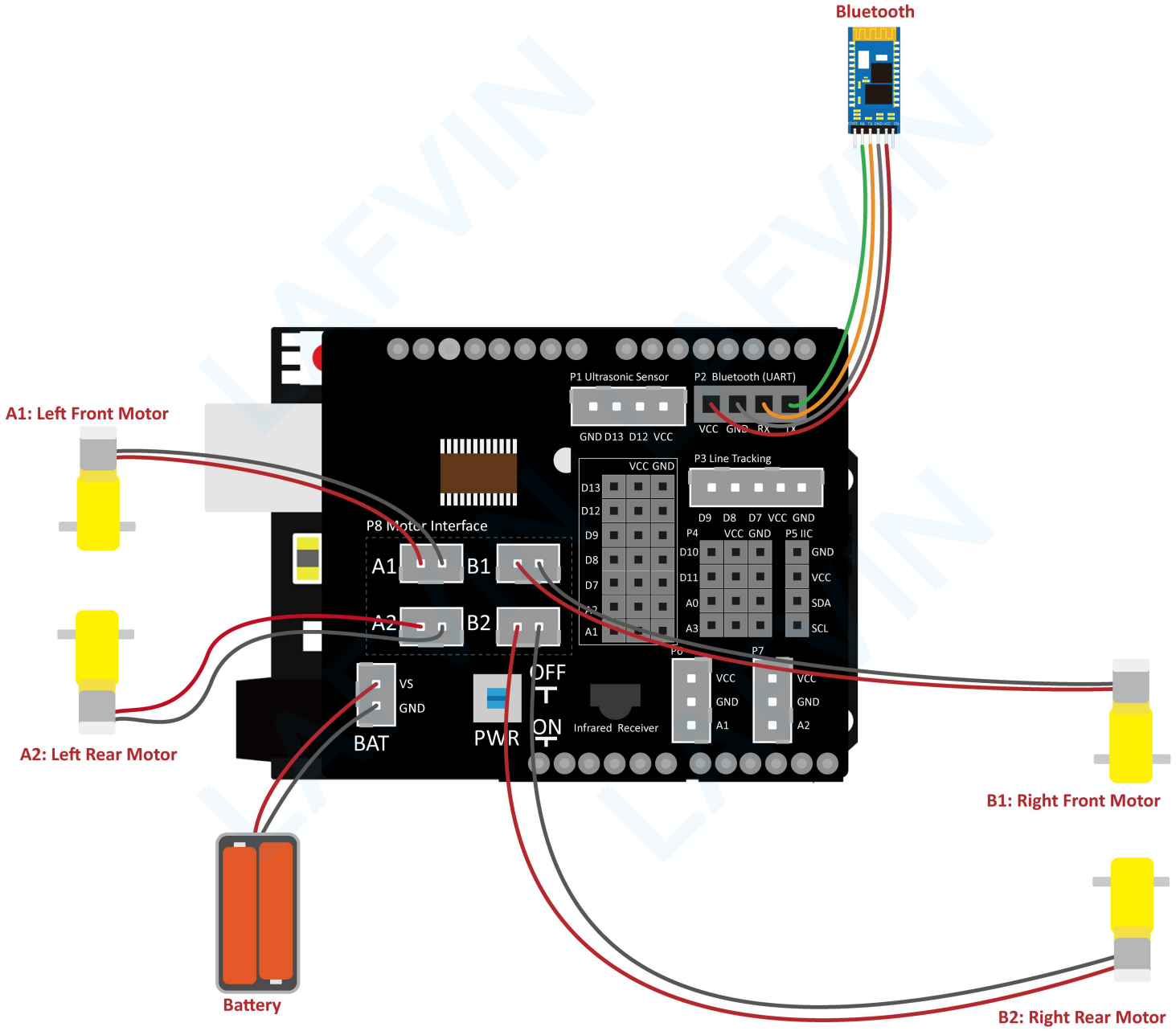
Note:that you need to close both the Arduino IDE serial monitor and the serial monitor of the Mixly software before uploading the code, otherwise the code will fail to upload because the communication serial port is occupied.

Test 2 Bluetooth Control Robot Car

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.




Bluetooth
A1: Left Front Motor
A2: Left Rear Motor
B1: Right Front Motor
B2: Right Rear Motor
Battery
P1 Ultrasonic Sensor
P2 Bluetooth (UART)
GND D13 D12 VCC
VCC GND RX TX
P3 Line Tracking
D9 D8 D7 VCC GND
P8 Motor Interface
A1
B1
A2
B2
P4
VCC GND
P5 IIC
D10
D11
A0
A3
GND
VCC
SDA
SCL
P6
P7
VCC
GND
A1
A2
VS
GND
BAT
PWR
OFF
ON
Infrared Receiver

Step 2:Upload code

Mixly Code

If you use the graphical programming software Mixly as the development environment.Open this source program in Mixly_Code>Lesson_9>"**Test_2_Bluetooth_Control_Robot_Car.mix**".Select the correct serial port to upload the Mixly code.

How to Install Mixly Software How to Add Mixly Libraries How to Upload the Reference Program

Arduino Code

If you use the Arduino IDE as the development environment.Open this source program in Arduino_Code>Lesson_9>"**Test_2_Bluetooth_Control_Robot_Car.ino**".Select the correct serial port to upload the code under the "Tools" menu.If you are missing some steps, Arduino IDE may report some errors. You can refer to the following solutions.

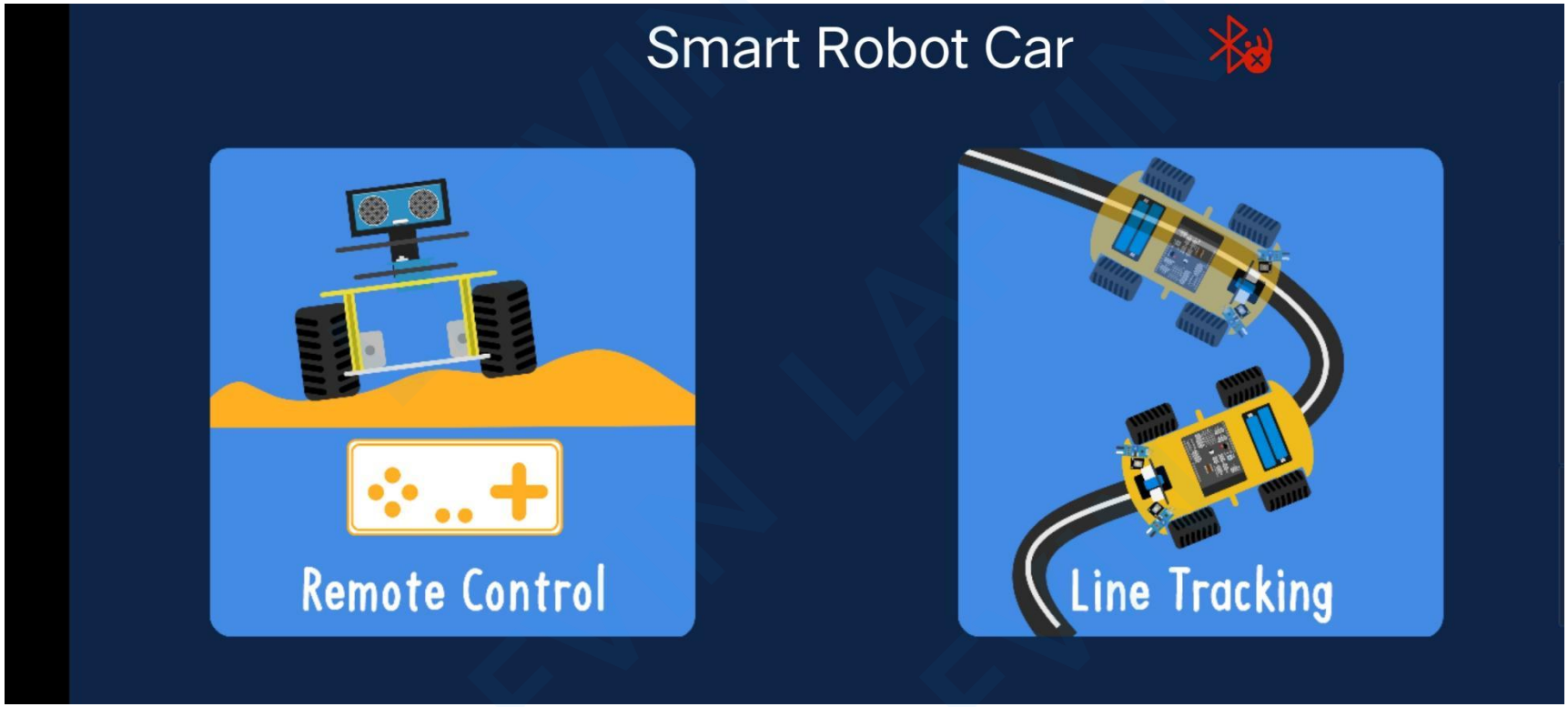
How to Unload Arduino Code.

Step 3:Install the app

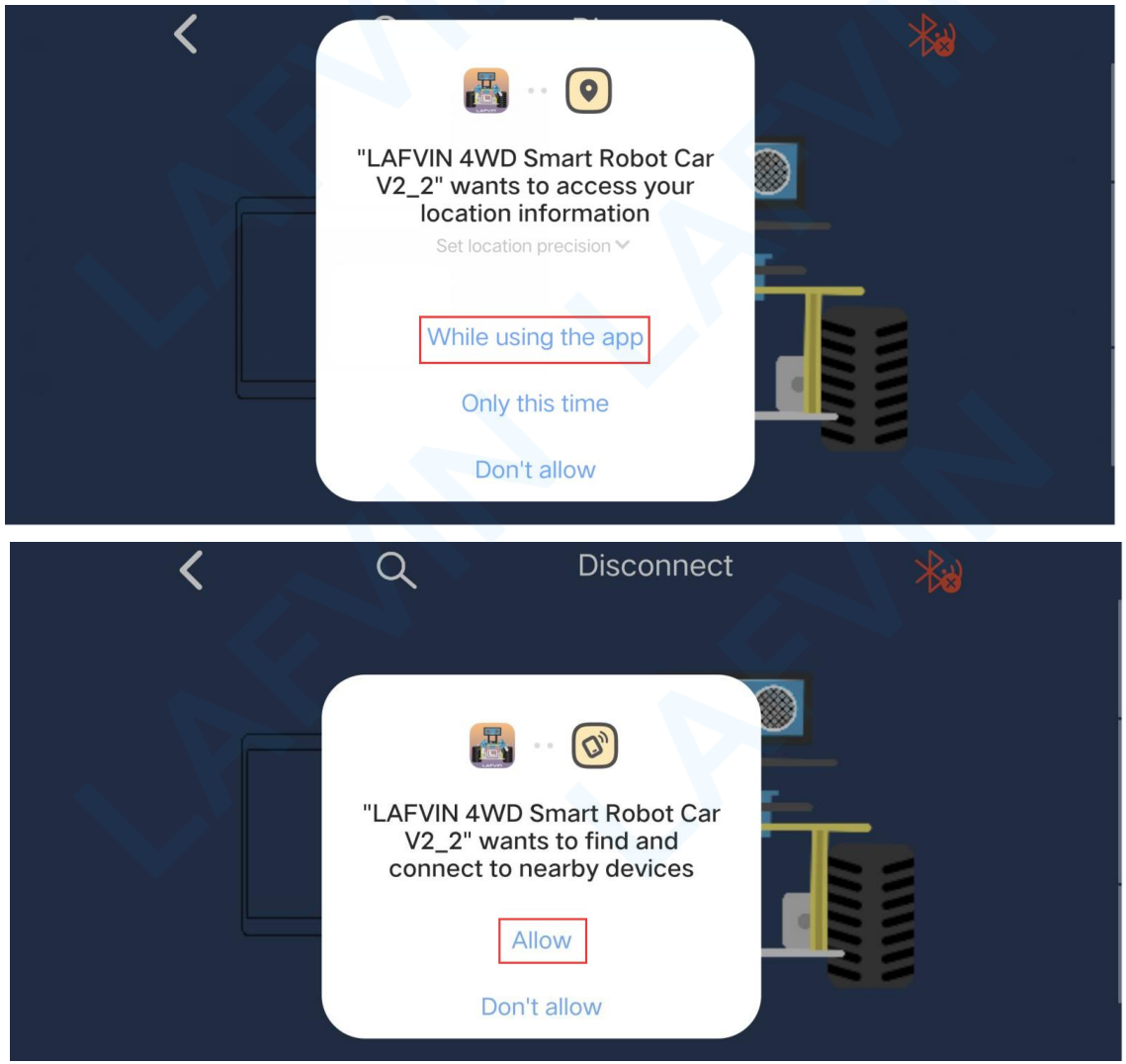
Firstly, copy the “LAFVIN_4WD_Smart_Robot_Car_BLE_V2_2.apk” file from the APP folder to your mobile phone and install it into an application software.Android phones equipped with Bluetooth 4.0 and upgraded to Android 4.3 Or higher system.

LAFVIN 4WD Smart Robot Car

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

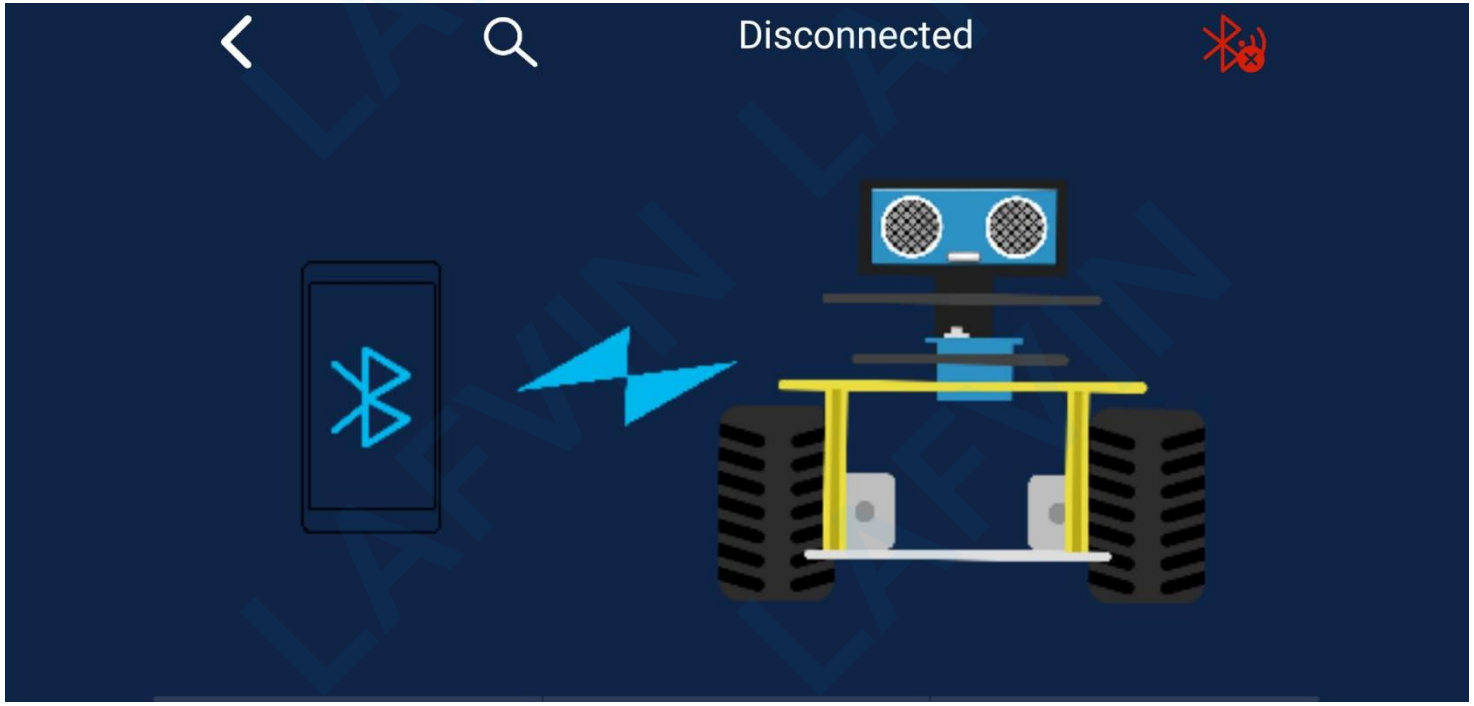


Important:Allows access to location permissions and permission to connect to nearby devices, otherwise it may not be possible to search for nearby Bluetooth devices.

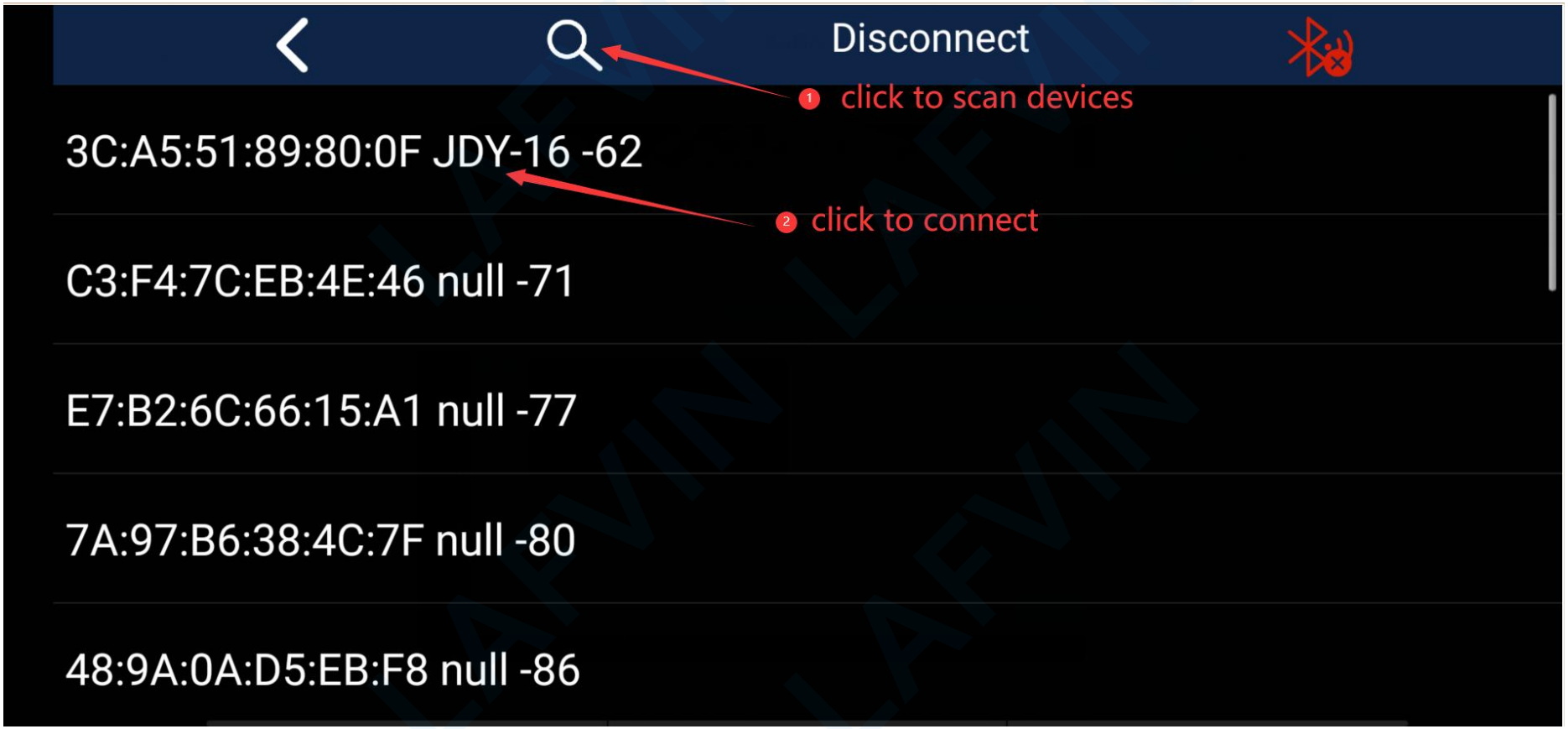


Step 4:Bluetooth connection

Install the 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, which means it is in a connectable state.Click the Bluetooth icon  to enter the Bluetooth pairing function page.

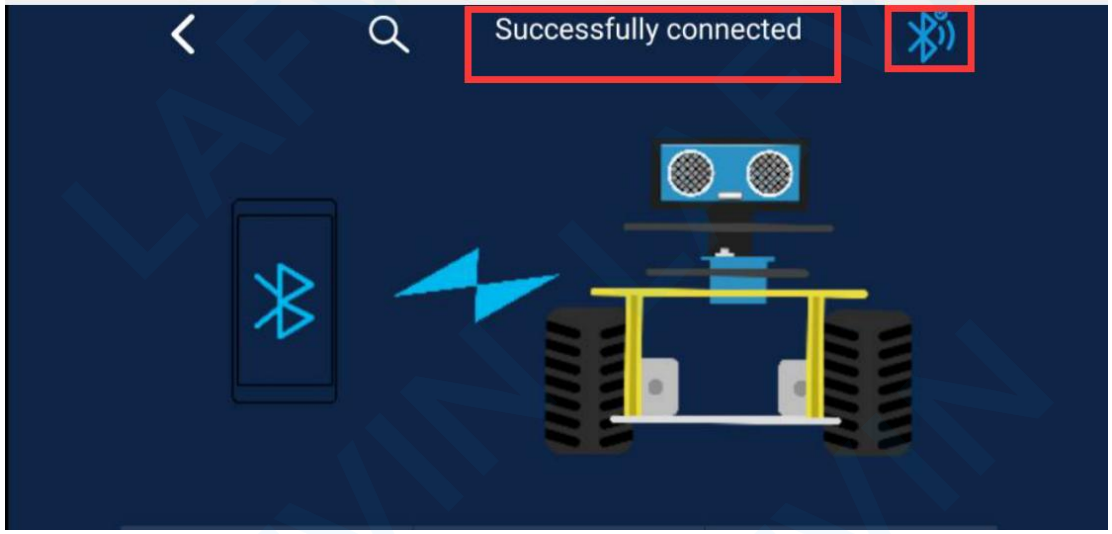


and then click device scan,After successfully searching for the device JDY-16, click the device address in the device list to automatically connect.



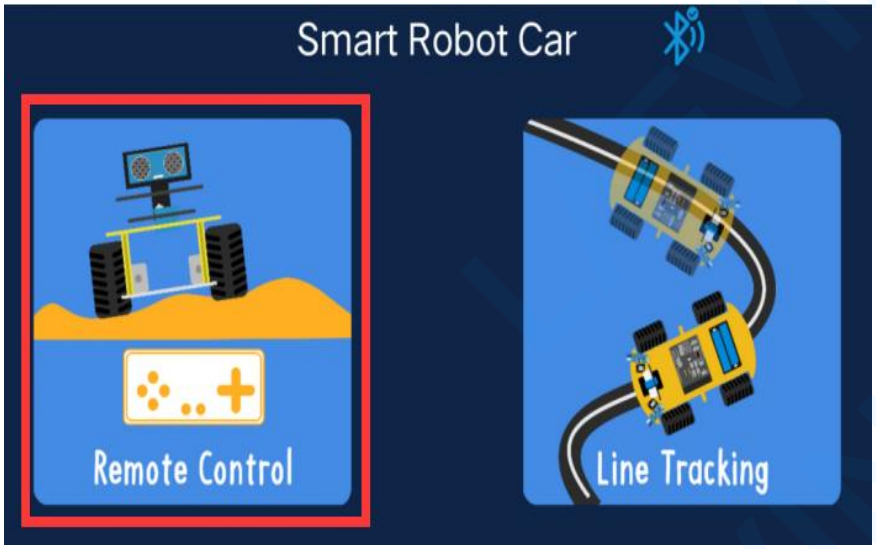
Report: Successful connected.This means that the correct device has been connected

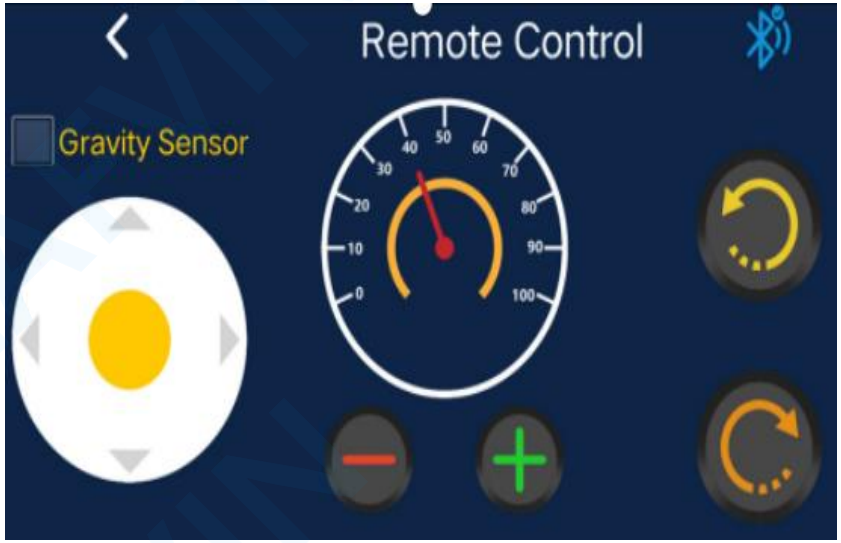
Report: Device Error. This means that the connected Bluetooth device does not belong to the robot car, search for JDY-16 and reconnect.



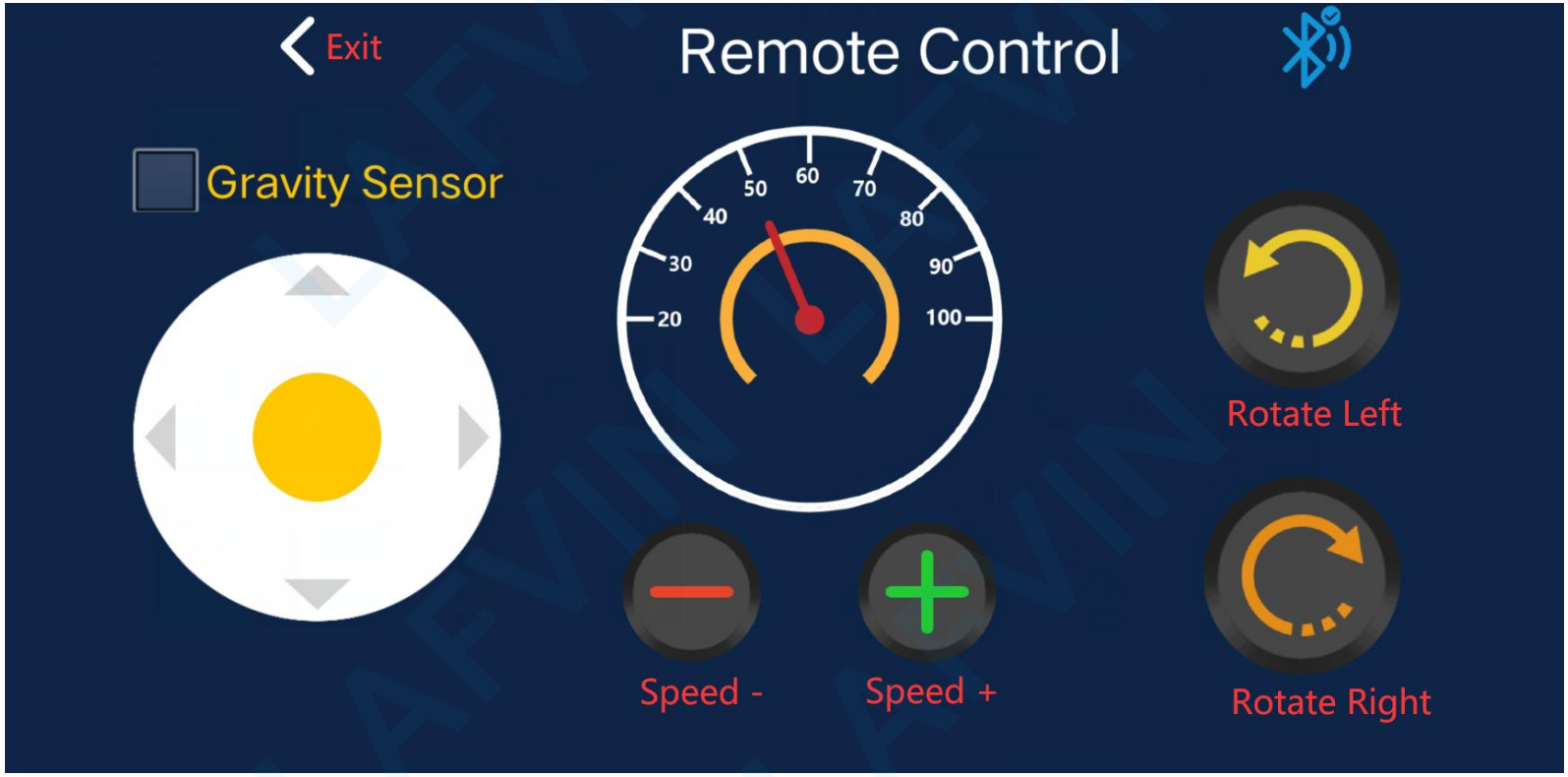
If App does not find the "JDY-16" device, restart the Bluetooth module (unplug the Bluetooth module from the Arduino UNO Motor Driver Shied, disconnect the power of the Bluetooth module; then reinstall it to the Arduino UNO Motor Driver Shied, connect to the power supply).the red LED flashes, which means it is in a connectable state. Then search for Bluetooth devices again in the APP.

Step 5:App control interface





The main function of the Remote Control interface is to control the direction and speed of the robot car. When the battery voltage is lower than 7.4V, replace it in time.

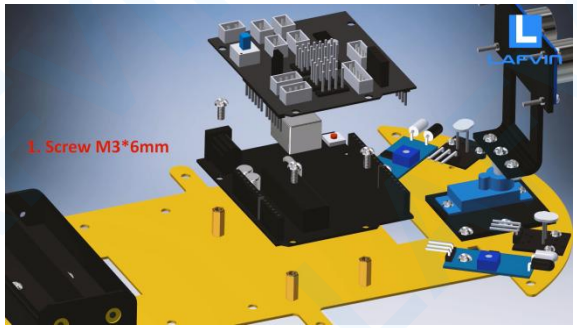


6 in 1 Multi-purpose Bluetooth Robot Car

All functions are collected into the same program, including Line Tracking,Ultrasonic Infrared Avoid Obstacles, Infrared Remote, Light Seeking,Follow Me,Remote Control, Gravity Sensor Remote Control, and any function can be freely switched through the app.

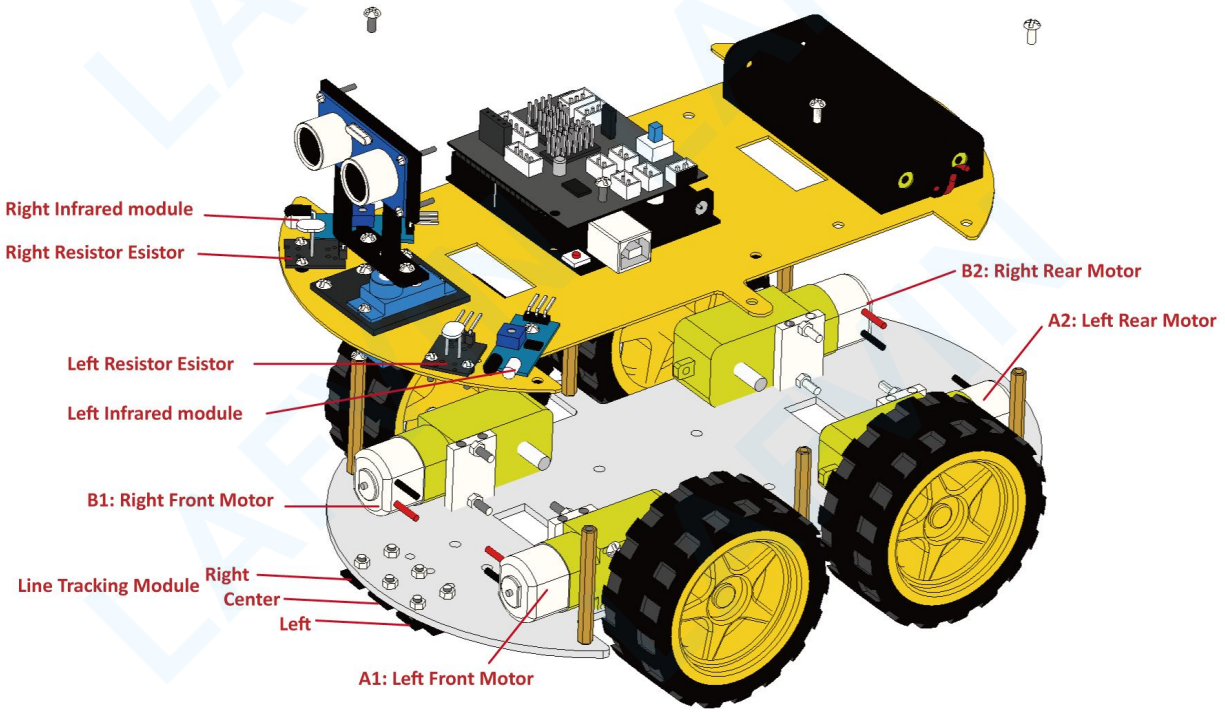
Step 1:Assemble the Robot Car

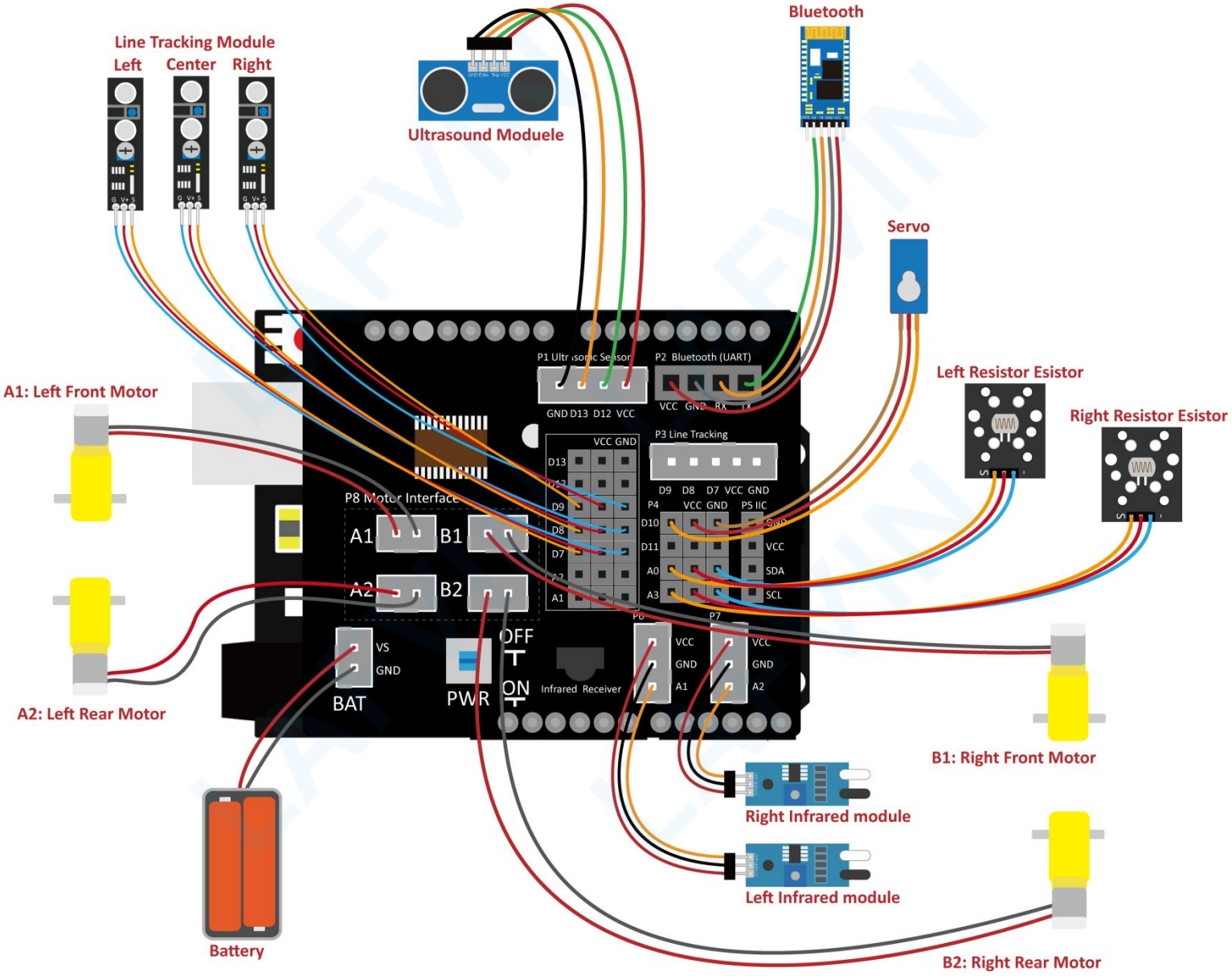
Refer to the assembly video to complete the installation of the robot car. Tip: The length of the screw is marked in the video, you should use the same length. Before installation, you need to identify M3*6mm M3*8mm M3*10mm M3*12mm. If you use the wrong screw length in the first half, there will be missing screws in the lower part of the installation.



Step 2: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.If you connect some parts by mistake, it will affect the normal execution of the robot car.




Line Tracking Module
Left
Center
Right
Ultrasound Moduele
Bluetooth
Servo
Left Resistor Esistor
Right Resistor Esistor
A1: Left Front Motor
A2: Left Rear Motor
B1: Right Front Motor
B2: Right Rear Motor
Right Infrared module
Left Infrared module
Battery
P1 Ultrasonic Sensor
P2 Bluetooth (UART)
P3 Line Tracking
P8 Motor Interface
A1
B1
A2
B2
BAT
VS
GND
PWR
OFF
ON
Infrared Receiver
VCC
GND
SDA
SCL

Step 3:Upload Code

Mixly Code

If you use the graphical programming software Mixly as the development environment.Open this source program in Mixly_Code>"**6_in_1_Multi_purpose_Bluetooth_Robot_Car_V2_2.mix**".Select the correct serial port to upload the Mixly code.

How to Install Mixly Software How to Add Mixly Libraries How to Upload the Reference Program

Arduino Code

If you use the Arduino IDE as the development environment.Open this source program in Arduino_Code>"**6_in_1_Multi_purpose_Bluetooth_Robot_Car_V2_2.ino**".Select the correct serial port to upload the code under the "Tools" menu.If you are missing some steps, Arduino IDE may report some errors. You can refer to the following solutions.

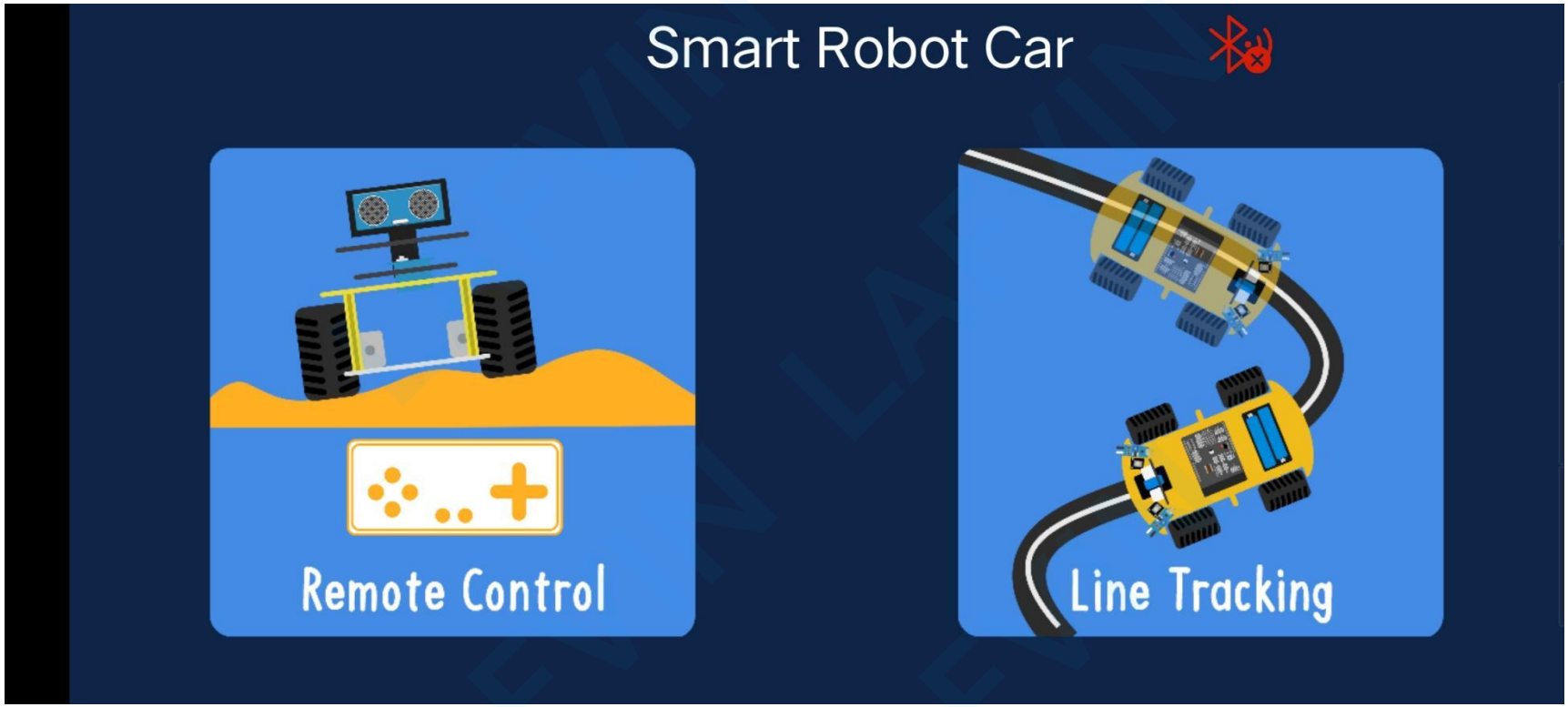
How to Upload Arduino Code How to Install Arduino IDE How to Install Arduino Driver

Step 4:Install the App

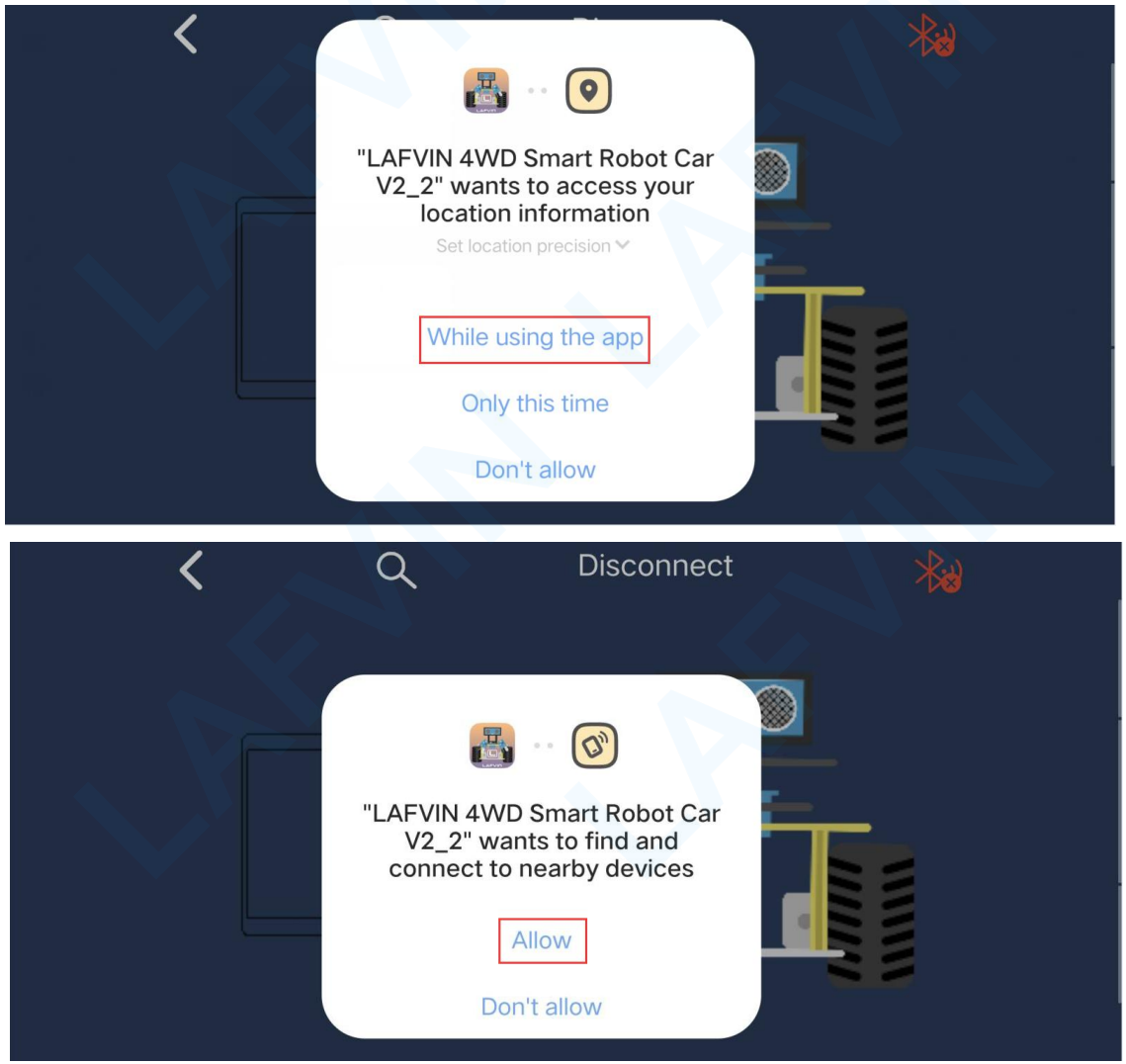
Firstly, copy the “LAFVIN_2WD_Smart_Robot_Car_BLE_V2_2.apk” file from the APP folder to your mobile phone and install it into an application software.Android phones equipped with Bluetooth 4.0 and upgraded to Android 4.3 Or higher system.

LAFVIN 4WD Smart Robot Car

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

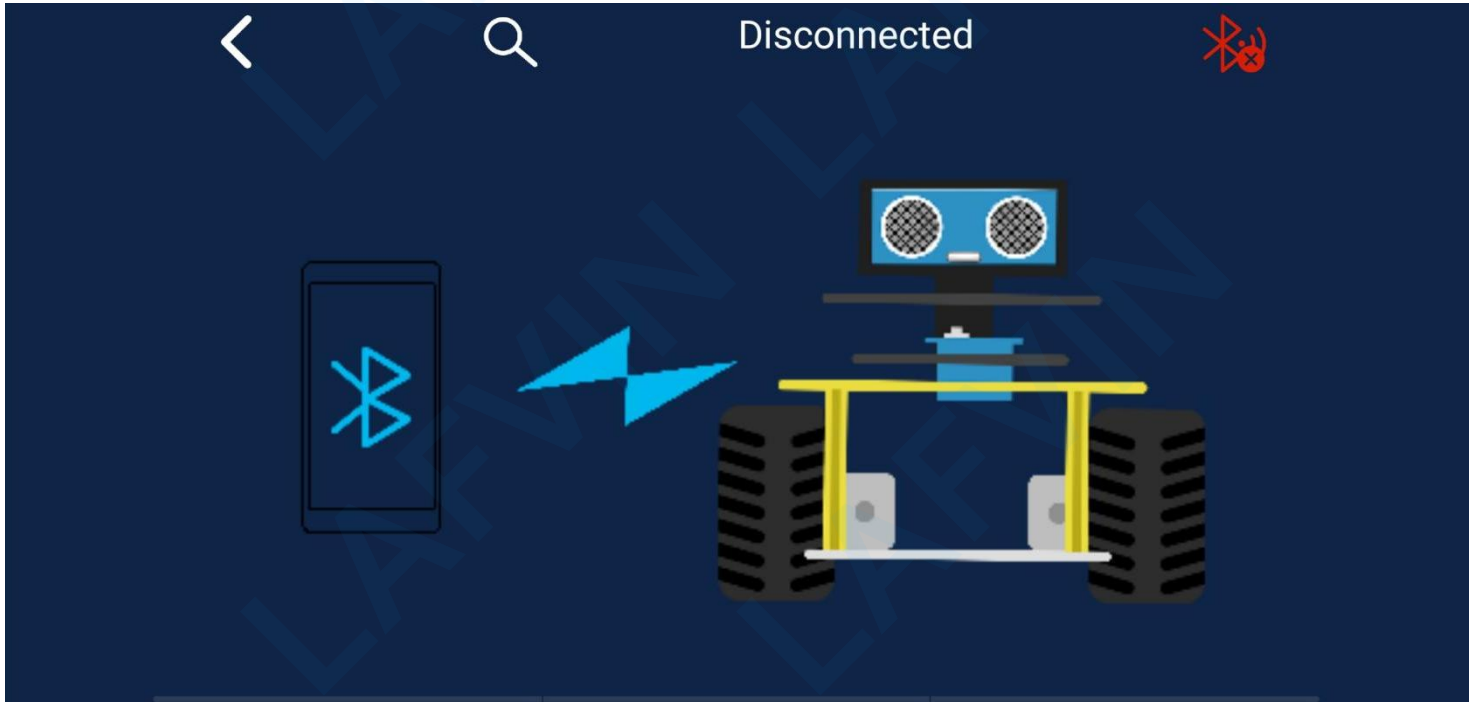


Important:Allows access to location permissions and permission to connect to nearby devices, otherwise it may not be possible to search for nearby Bluetooth devices.

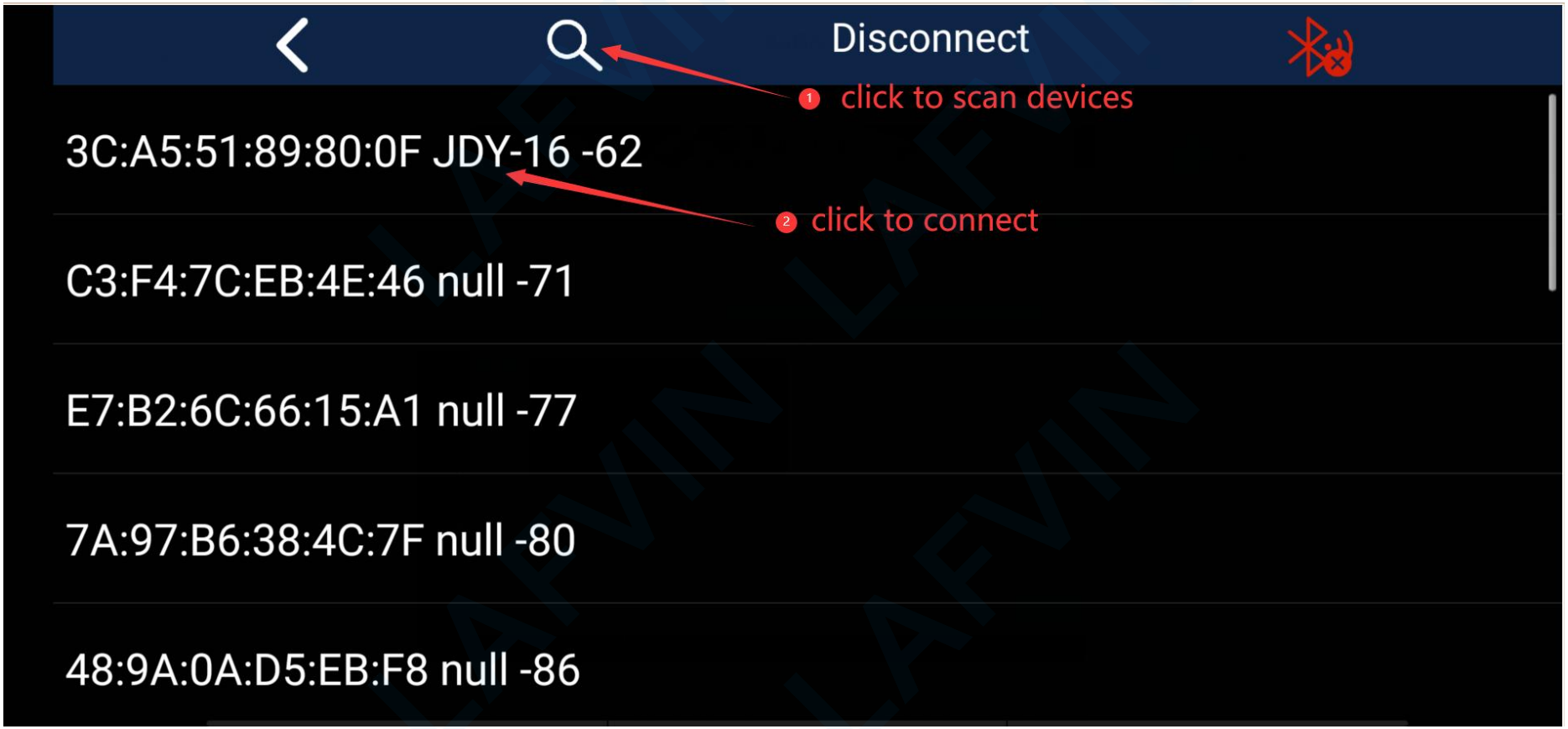


Step 5:Bluetooth Connection

Install the 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, which means it is in a connectable state.Click the Bluetooth icon  to enter the Bluetooth pairing function page.

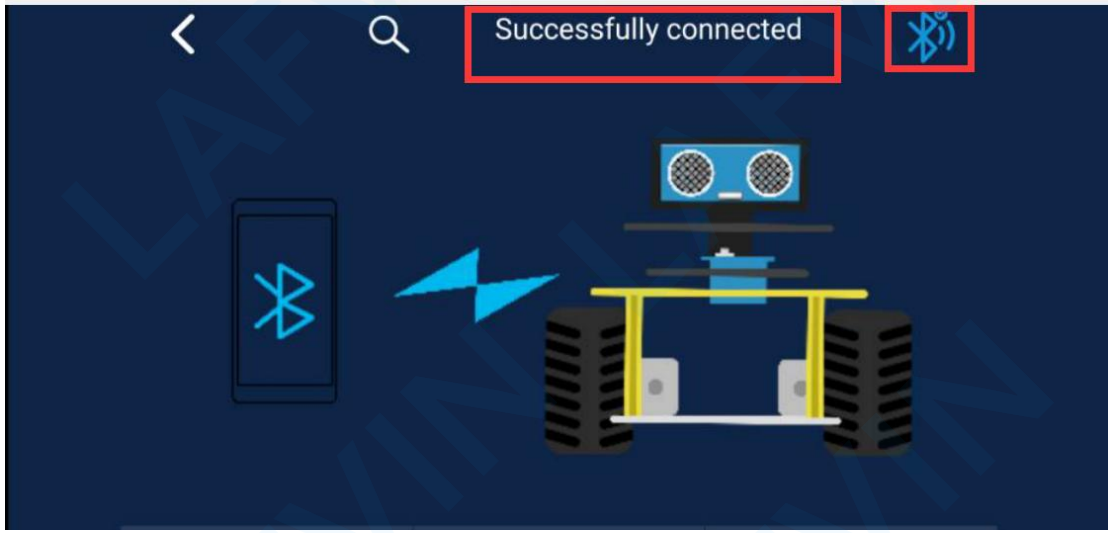


and then click device scan,After successfully searching for the device JDY-16, click the device address in the device list to automatically connect.



Report: Successful connected.This means that the correct device has been connected

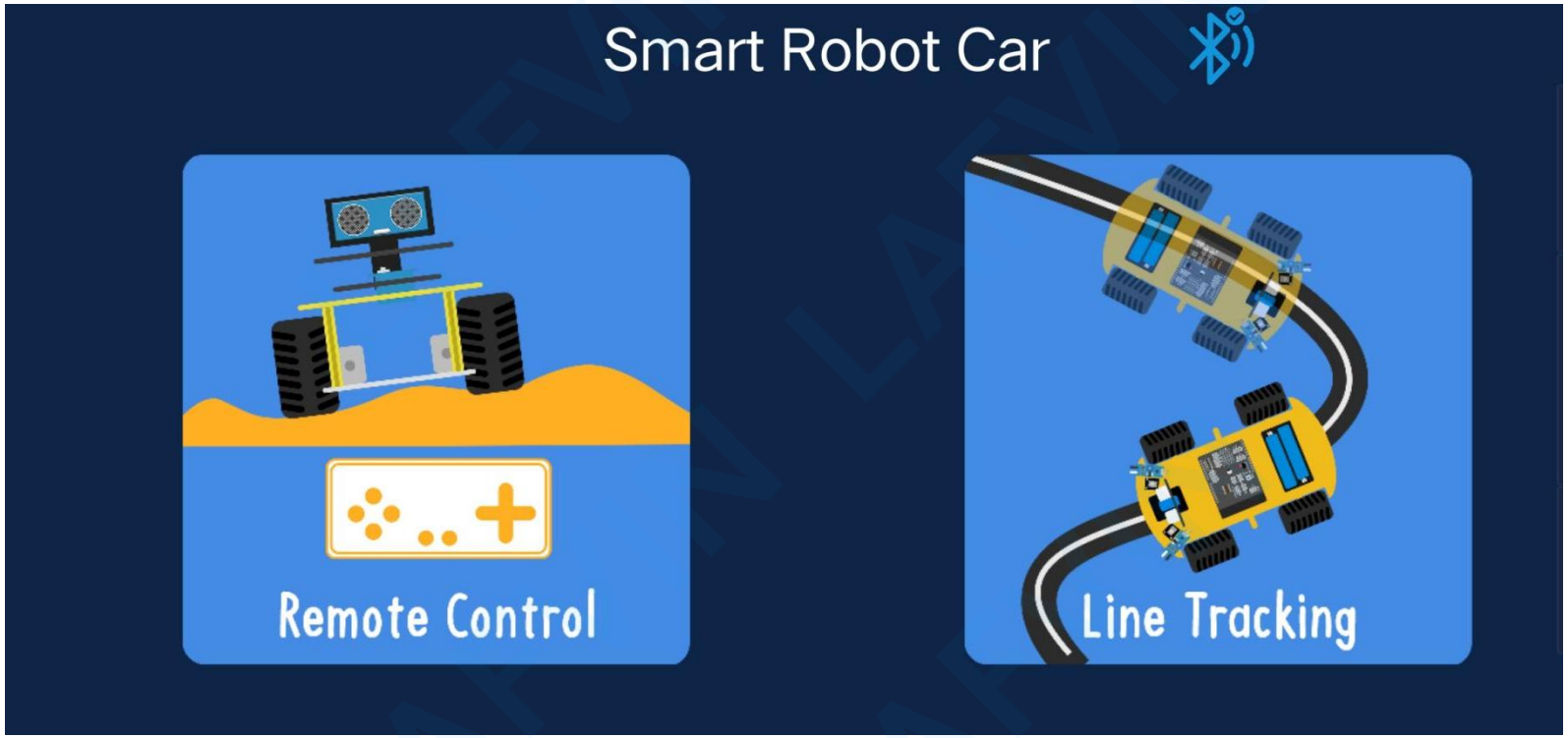
Report: Device Error. This means that the connected Bluetooth device does not belong to the robot car, search for JDY-16 and reconnect.

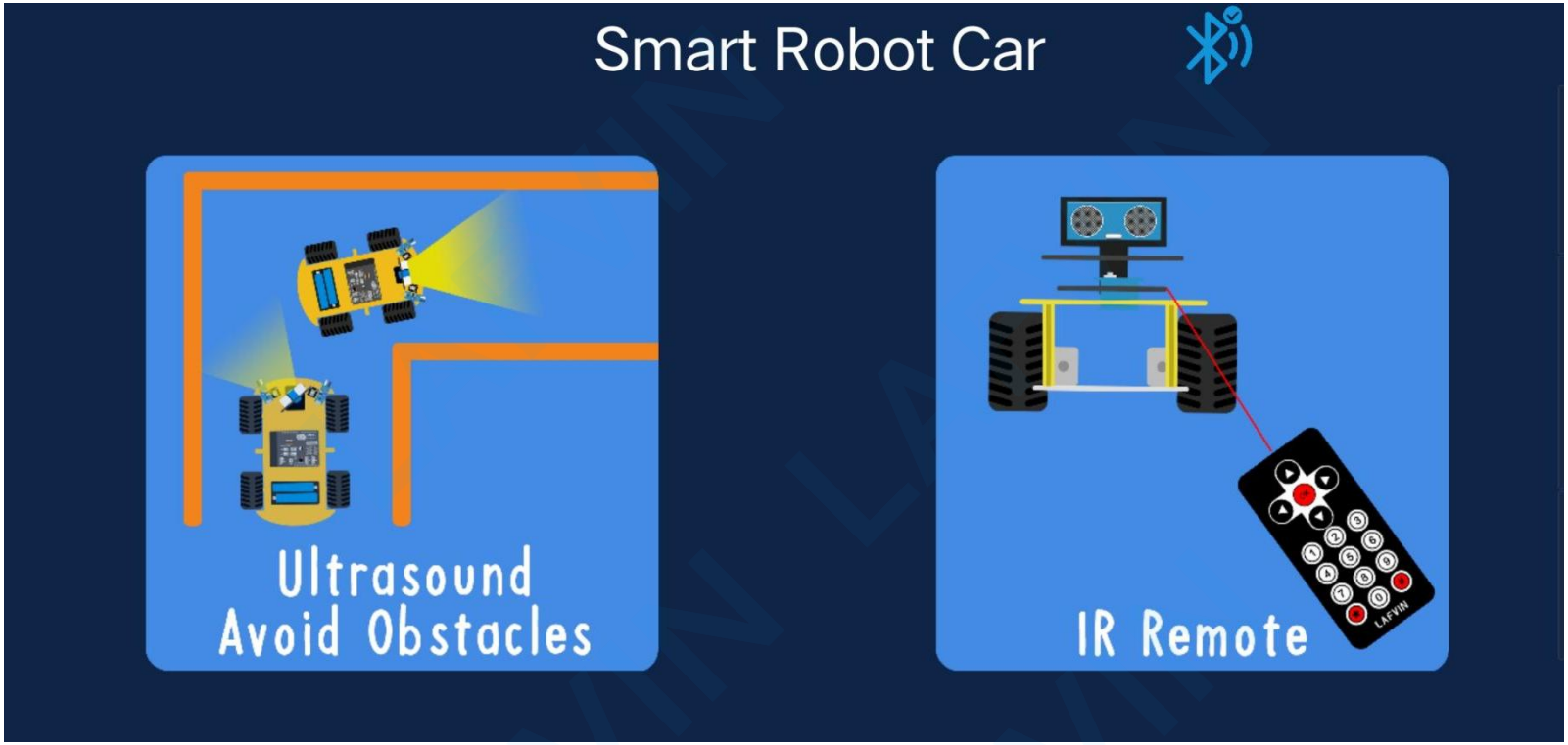


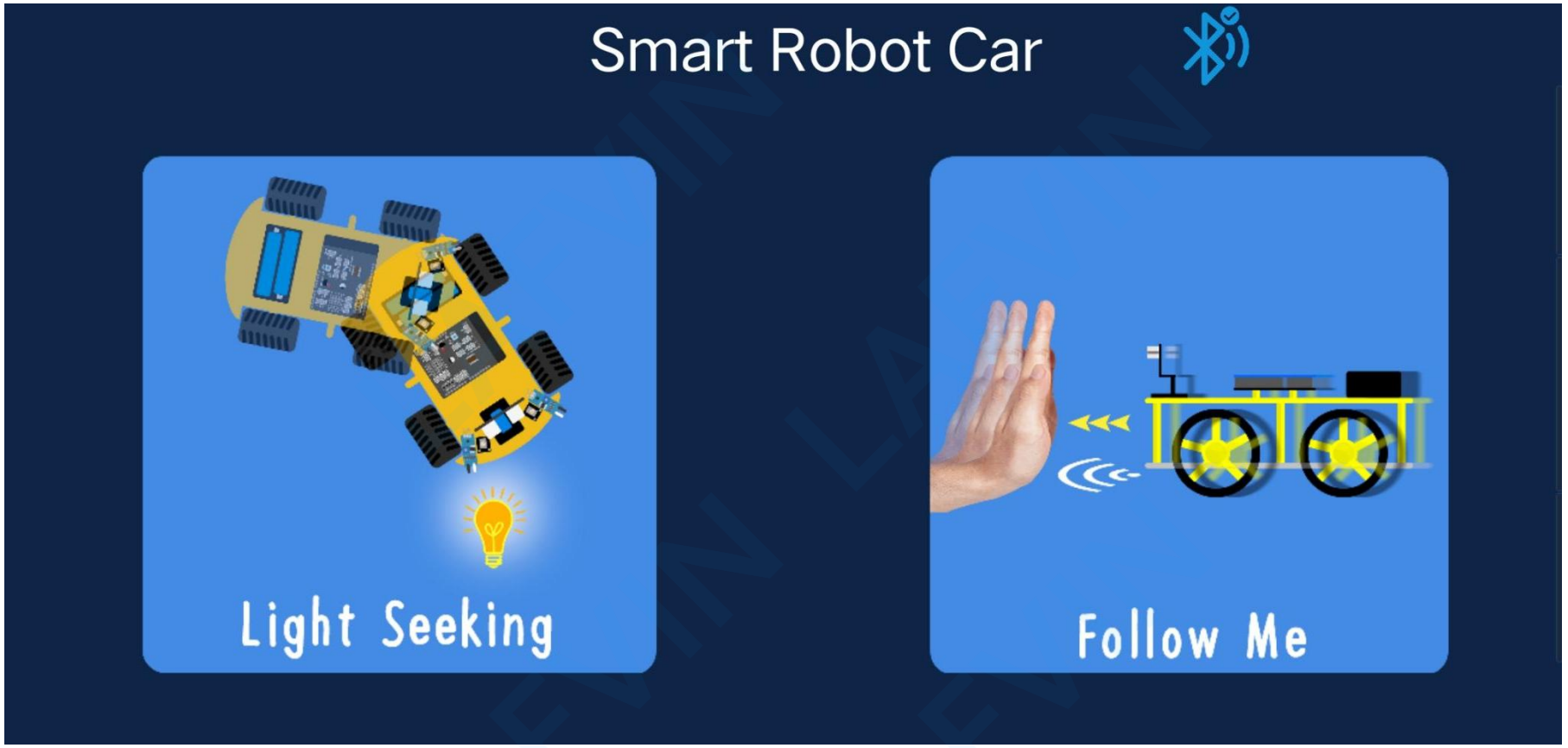
If App does not find the "JDY-16" device, restart the Bluetooth module (unplug the Bluetooth module from the Arduino UNO Motor Driver Shied, disconnect the power of the Bluetooth module; then reinstall it to the Arduino UNO Motor Driver Shied, connect to the power supply).the red LED flashes, which means it is in a connectable state. Then search for Bluetooth devices again in the APP.

Step 6:App control interface

Swipe the screen from left to right

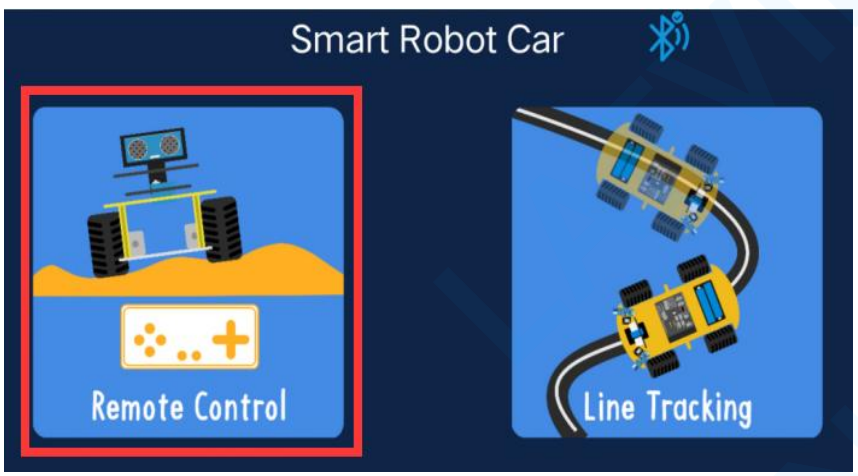


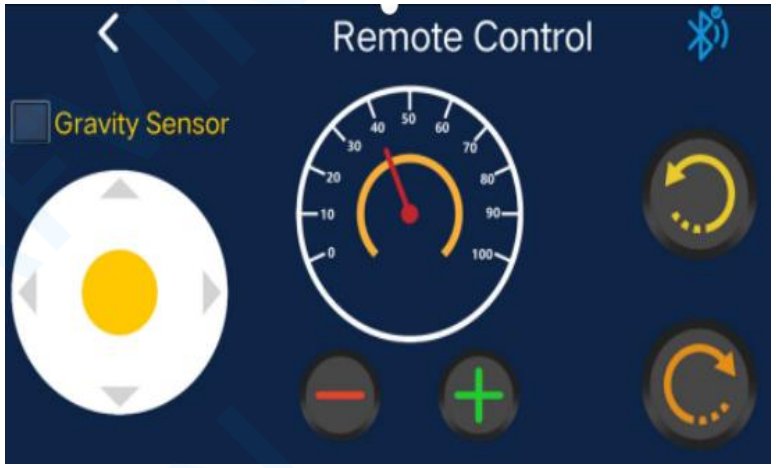

Smart Robot Car
Ultrasound
Avoid Obstacles
IR Remote


Smart Robot Car
Light Seeking
Follow Me

You can freely switch the various functional modes through the mobile phone app, such as ultrasonic obstacle avoidance, infrared obstacle avoidance, infrared remote control, gravity sensor remote control.When the battery voltage is lower than 7.4V, replace it in time.

There are 6 icons on the homepage, and each icon represents a function. The robot car immediately executes the function after clicking.

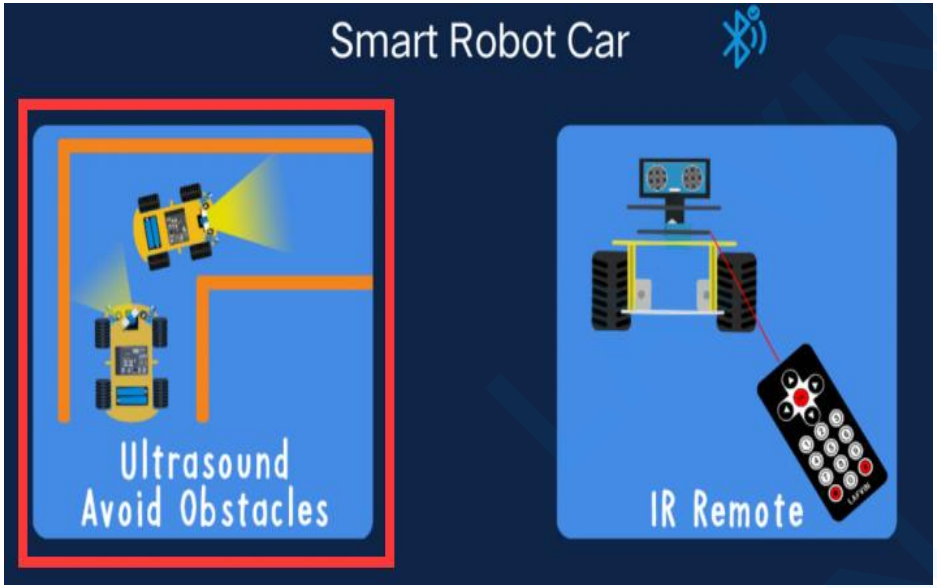


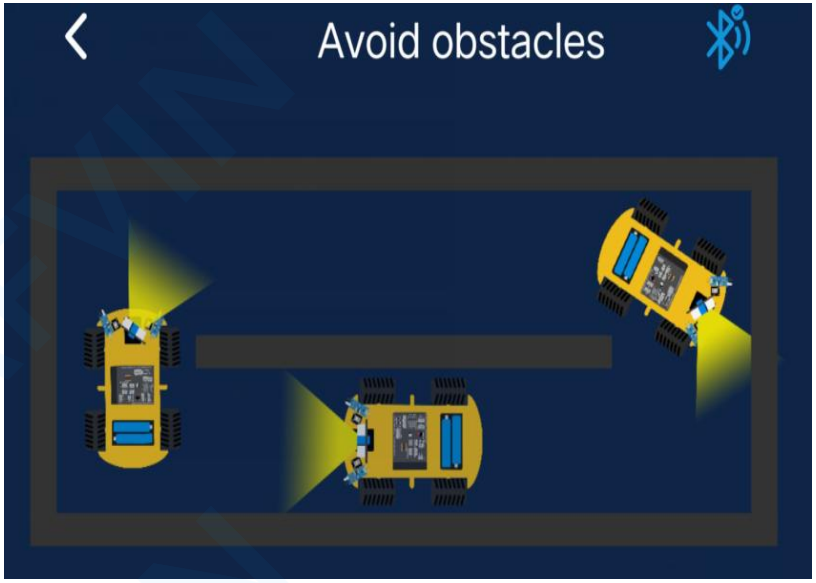


If you have questions about the use of the **Remote Control** function**,** or you want to add your ideas to this function, you can test this function separately. Lesson 9 Bluetooth Control Robot Car.The reason for this is to avoid other factors affecting the execution of this function.

Note: Use black tape to draw the line on the white floor. The width of the black line is 15cm. Adjust the sensitivity through the potentiometer on the line-following sensor.

If you have questions about the use of the **Line Tracking** function, or you want to add your ideas to this function, you can test this function separately.Lesson 4 Line Tracking Smart Car.The reason for this is to avoid other factors affecting the execution of this function.



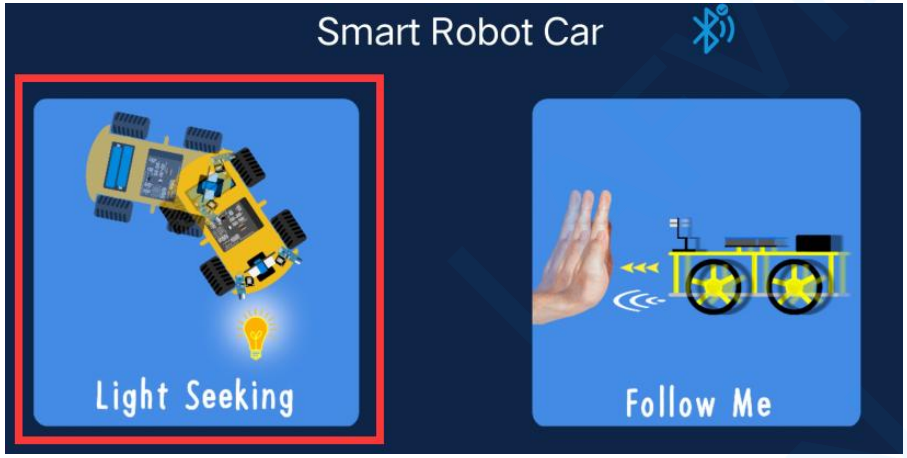


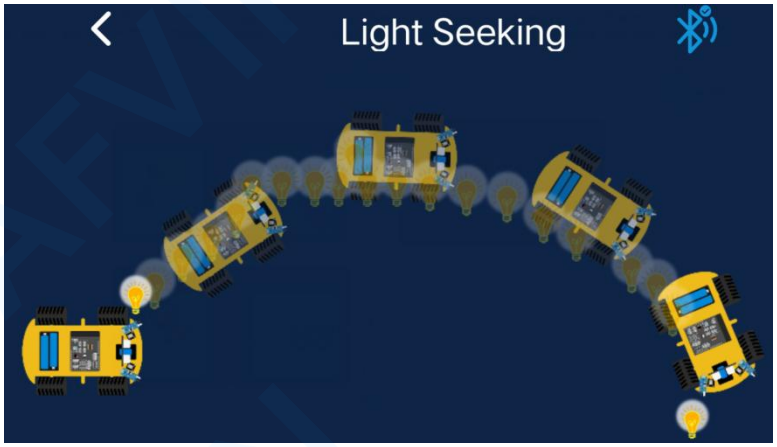
Note: It is recommended that the height of the obstacle is greater than 15cm.which means that the height of the obstacle is greater than the height of the ultrasonic sensor.

If you have questions about the use of the **Avoid Obstacles** function, or you want to add your ideas to this function, you can test this function separately.Lesson 5 Ultrasonic Infrared Obstacle Avoidance Robot Car. The reason for this is to avoid other factors affecting the execution of this function.

Note: Due to air transportation, the remote control does not contain batteries, you need to prepare a button battery CR2025 before use.

If you have questions about the use of the **IR Remote** function, or you want to add your ideas to this function, you can test this function separately. Lesson 7 Infrared Remote Control Robot Car.The reason for this is to avoid other factors affecting the execution of this function.





Note:Test in a dark environment and turn off the lights in the room.

If you have questions about the use of the **Light Seeking** function, or you want to add your ideas to this function, you can test this function separately. Lesson 8 Light Seeking Robot Car.The reason for this is to avoid other factors affecting the execution of this function.

Note: It is recommended to use a rectangular object box to guide the robot car. Length>10cm.Width>15cm.

If you have questions about the use of the **Follow Me** function, or you want to add your ideas to this function, you can test this function separately. Lesson 6 Ultrasonic Follow Robot Car.The reason for this is to avoid other factors affecting the execution of this function.