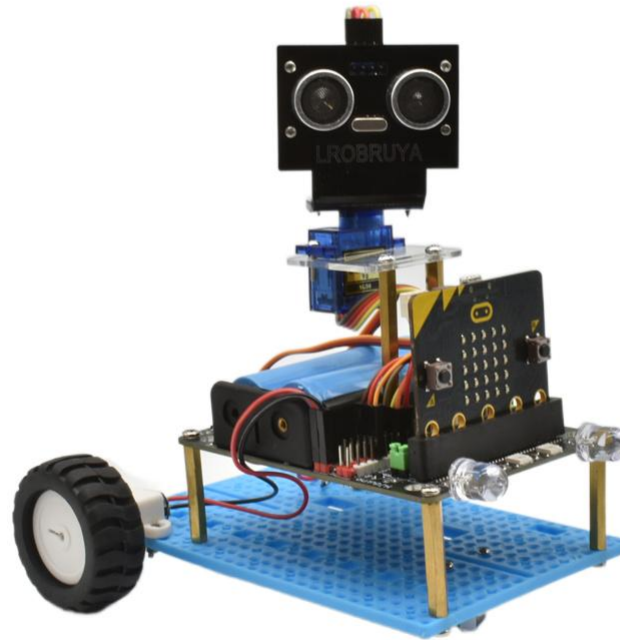


LROBRUYA

Micro bit Smart Robot Car Kit

LROBRUYA





Content

Preface	4
Micro:bit	5
GPIO	8
Introduction to Driver Board Functions	9
Code & Programming	11
Wiring diagram	26
Lesson 1 Light two leds	27
Lesson 2 LED light flashing	30
Lesson 3 RGB colored light strip	31
Lesson 4 The buzzer sounds	33
Lesson 5 Sound light alarm	36
Lesson 6 Car automatic driving	38
Lesson 7 Push start car	42
Lesson 8 Light controlled car	44
Lesson 9 Ultrasonic LED ranging display	45
Lesson 10 Ultrasonic ranging serial port display	48
Lesson 11 Ultrasonic magic hand	51
Lesson 12 Ultrasonic obstacle avoidance car	54



Preface

Company Profile

Founded in 2014, Shenzhen Lonten Technology Co., Ltd. focuses on the design, research production of Electronics Module for robotics related products. Consisting of professional researchers and skilled engineers, our R&D team constantly strives for creative function and excellent user experience. The company's R&D investments on arduino kits raspberry pi kits, as well as 3D printer and robots that back up STEAM education.

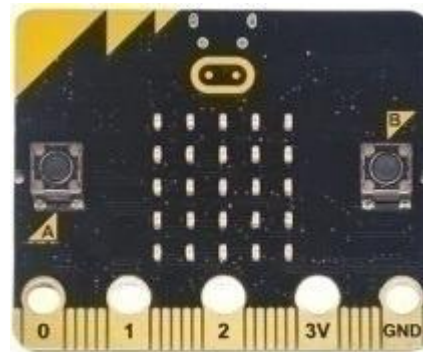
Customer Service

Our self-owned factory is certificated with BSCI and SO, covering an area of 5,000 square meters, and achieving an annual production capacity of over 10,000 units. Our products are all certified to CE, FCC, and ROHS standards, have exported to more than 100 countries including, but not limited to France, the United States of America, Australia, Russia, the United Kingdom, Germany, Singapore, Egypt, and India, bringing technological innovation to all walks of life.

LROB RUYA

Micro:bit

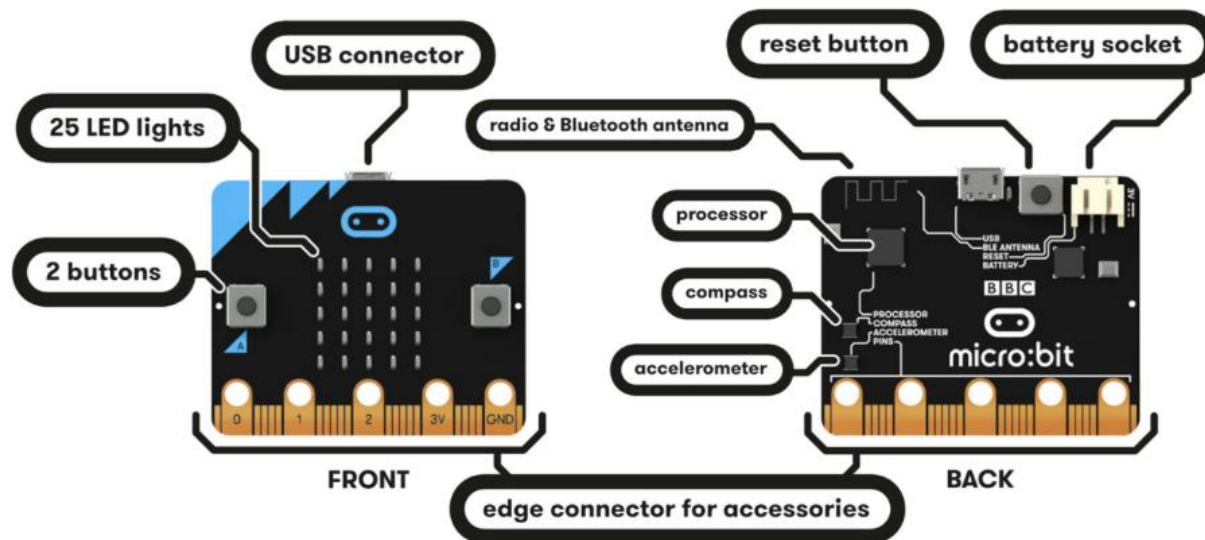
This chapter is the Start Point in the journey to build and explore Micro:bit electronic projects.



The BBC micro:bit is a pocket-size, programmable micro-computer that can be used for all sorts of cool creations, from robots to musical instruments the possibilities are infinite.

For more contents, please refer to: <https://microbit.org/guide/>

LROBRUYA



Your micro:bit has the following physical features:

- 25 individual programmable LEDs
- 2 programmable buttons
- Physical connection pins



-
- Light and temperature sensors
 - Motion sensors (accelerometer and compass)
 - Wireless Communication, via Radio and Bluetooth
 - USB interface

For more details, please refer to: <https://microbit.org/guide/features/>

It is not required for beginners to master this section, but a brief understanding is necessary. However, if you want to be a developer, hardware information will be very helpful. Detailed hardware information about micro:bit can be found here:

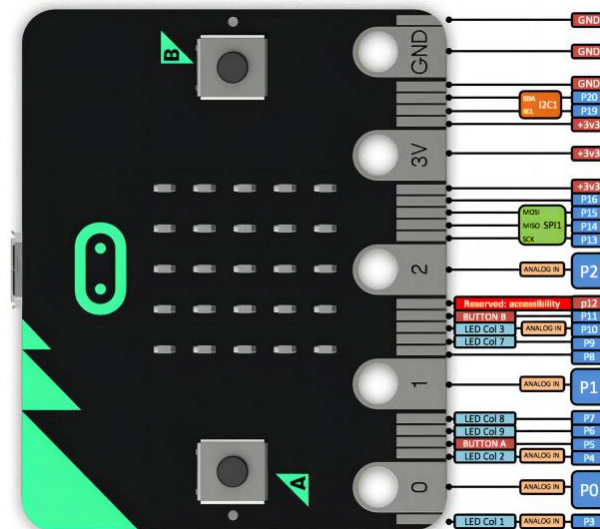
<https://tech.microbit.org/hardware/>

First, get to know the micro:bit GPIO.

LROBRUYA

GPIO

GPIO, namely General Purpose Input/output Pins, is an important part of micro:bit for connecting external devices. All sensors and devices on Rover communicate with each other through micro:bit GPIO. The following is the GPIO serial number and function diagram of micro:bit:



LROBRUYA

Introduction to Driver Board Functions

Hardware and Feature

Driver Board is shown as below:

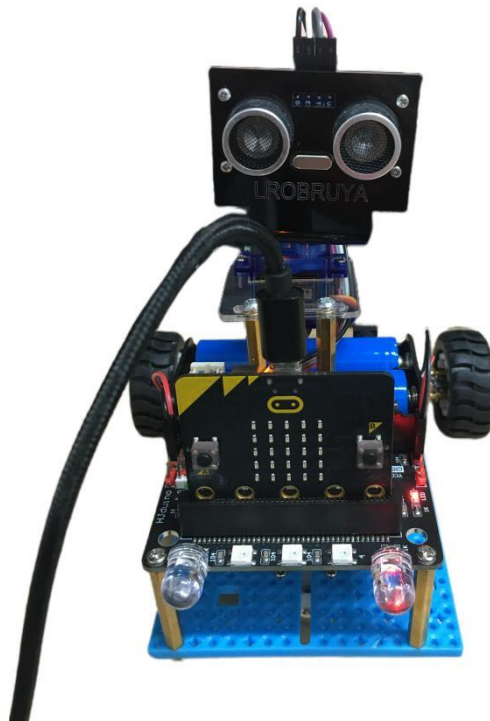


Integrated with 2 driver chips, capable of connecting 4 motors in parallel to form a 2WD car, 4WD car, integrated sensor interface, RGB interface, LED headlights, two 18650 battery sockets.

LROBRUYA

How to connect?

Attention: Install in the direction shown in the diagram. If installed in the wrong direction, it may burn out the motherboard.





Code & Programming

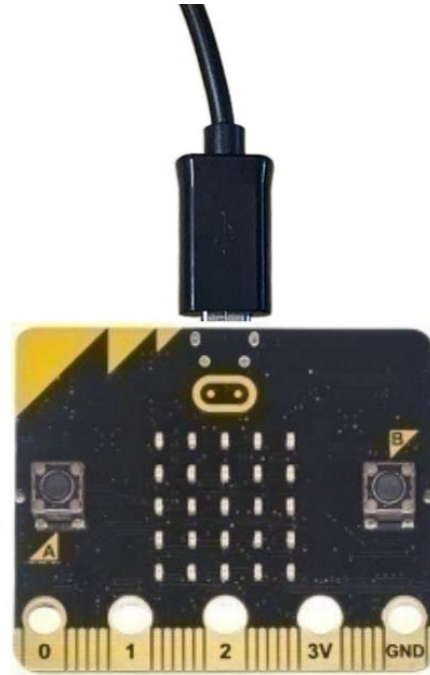
Quick Start

This section describes how to write programs for micro:bit and how to download them to micro:bit. There are very detailed tutorials on the official website. You can refer to: <https://microbit.org/guide/quick/>

Step 1: Connecting Micro:bit

Connect the micro:bit to your computer via a micro USB cable. Macs, PCs, Chromebooks and Linux systems (including Raspberry Pi) are all supported.

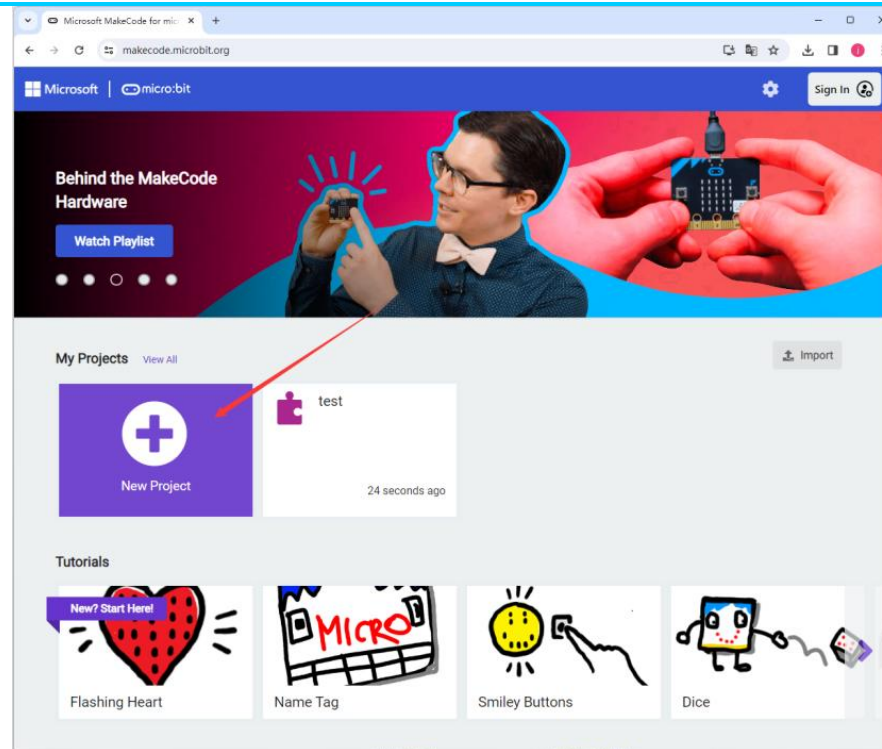
LROBRUYA



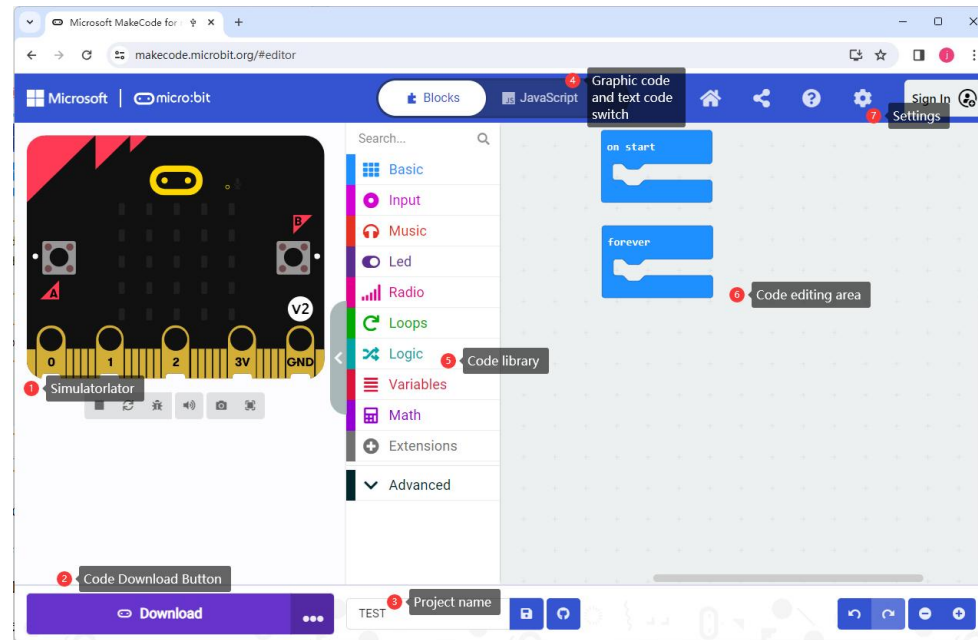
Step 2: Write Program

Visit <https://makecode.microbit.org/> Then click "New Project" and start programming.

LROBROYA



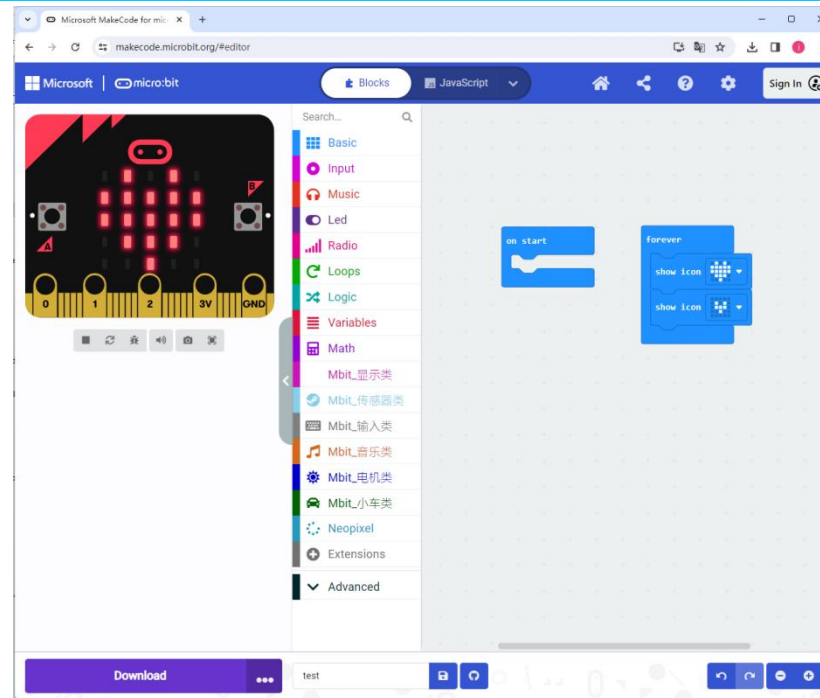
LROBRUYA



Write your first micro:bit code. For example, drag and drop some blocks and try your program on the Simulator in the MakeCode Editor, like in the image below that shows how to program a Flashing Heart.

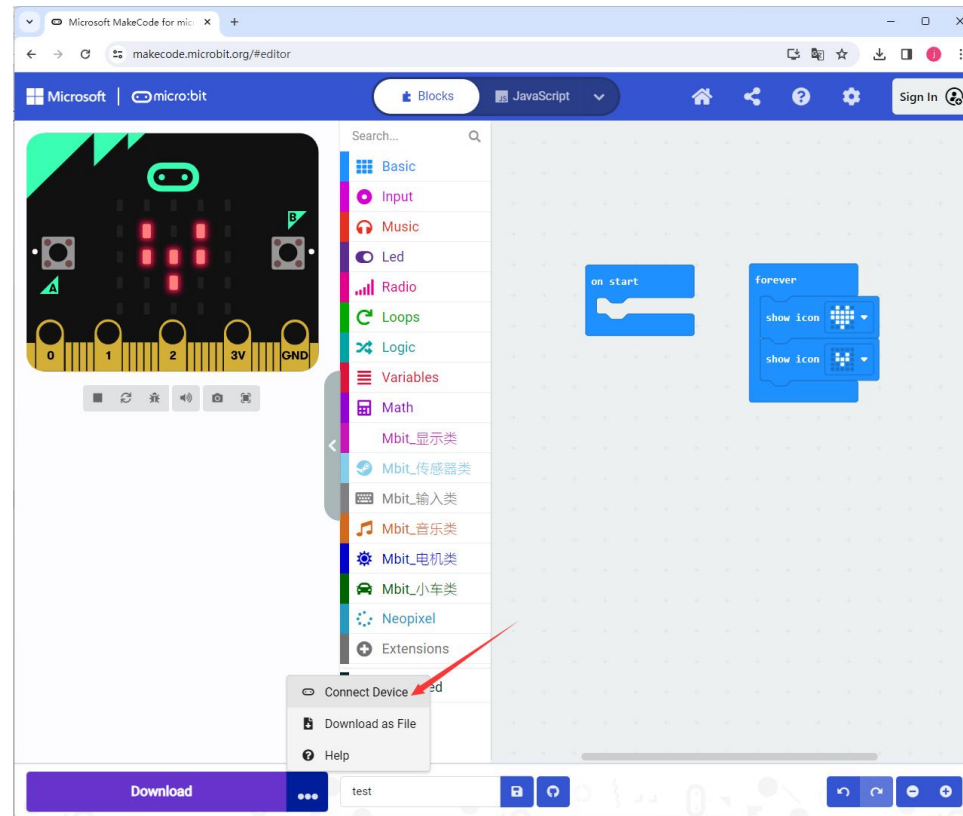
MakeCode will be further introduced in next section.

LROBROYA



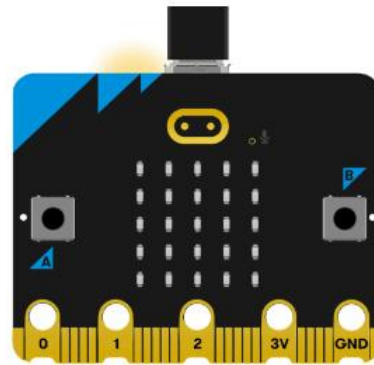
Then continue to click “Connect Device” button.

LROBROYA



LROBROYA

1. Connect your micro:bit to your computer



Next

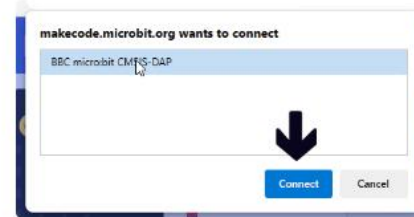
LROBBUYA

2. Pair your micro:bit to your browser

Press the Pair button below.

A window will appear in the top of your browser.

Select the micro:bit device and click Connect.



 Pair

LROBRUYA



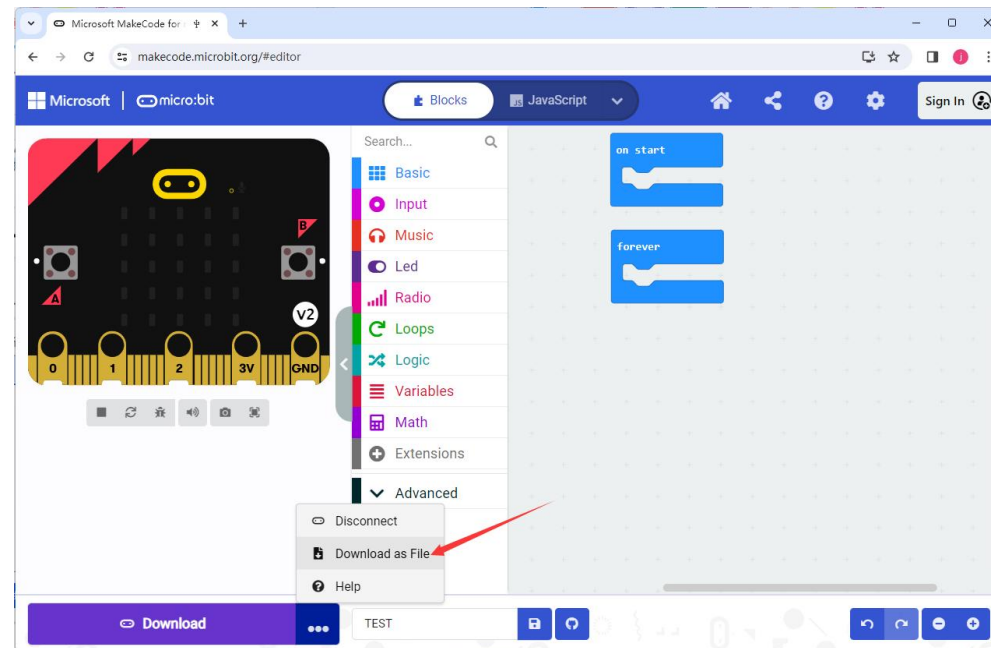
Step 3: Flashing Code to your Micro:bit

The process of transferring the .HEX file to the BBC micro:bit is called flashing.

LROB RUYA

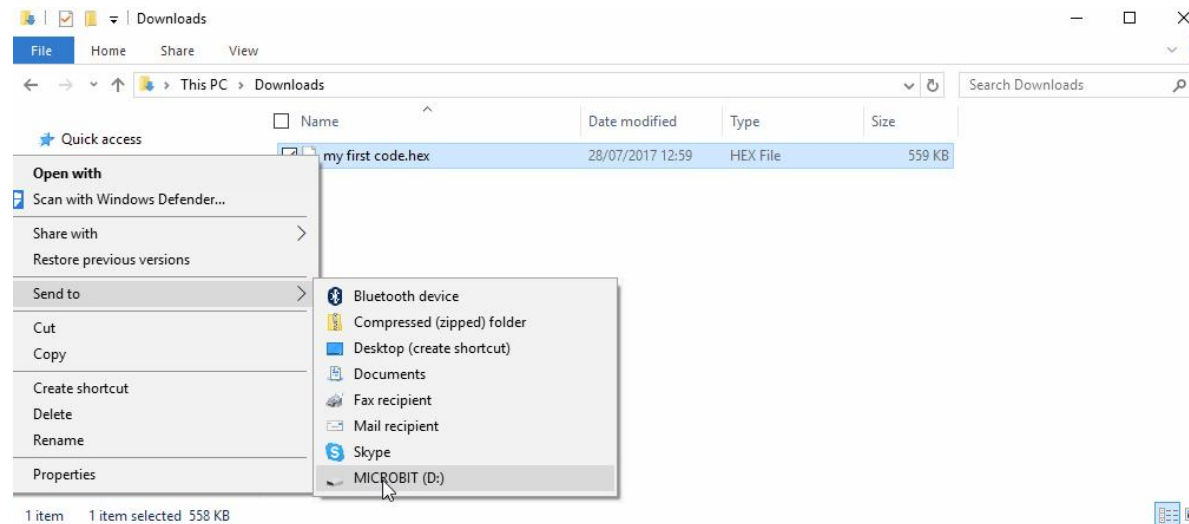
If you write program using Windows 10 App, you just need to click the “Download” button, then the program will be downloaded directly to micro:bit without any other actions.

If you write program using browser, please follow steps below:





Click the Download as File button in the editor. This will download a '.hex' file, which is a compact format of your program that your micro:bit can read. Once the hex file has been downloaded, copy it to your micro:bit just like copying a file to a USB drive. On Windows you can right click and choose "Send to→MICROBIT."



Step 4: Run the Program



The micro:bit will pause and the yellow LED on the back of the micro:bit will blink while your code is flashed. Once that's finished the code will run automatically! The micro:bit can only run one program at a time - every time you drag-and-drop a hex file onto the device over USB it will erase the current program and replace it with the new one.

Warning

The MICROBIT drive will automatically eject and reconnect each time you program it, but your hex file will be gone. The micro:bit can only receive hex files and won't store anything else!

Import Code

We provide hex file (project files) for each project, which contains all the contents of the project and can be imported directly. You can also complete the code of project manually. If you choose to complete the code by dragging code block, you may need to add necessary extensions.

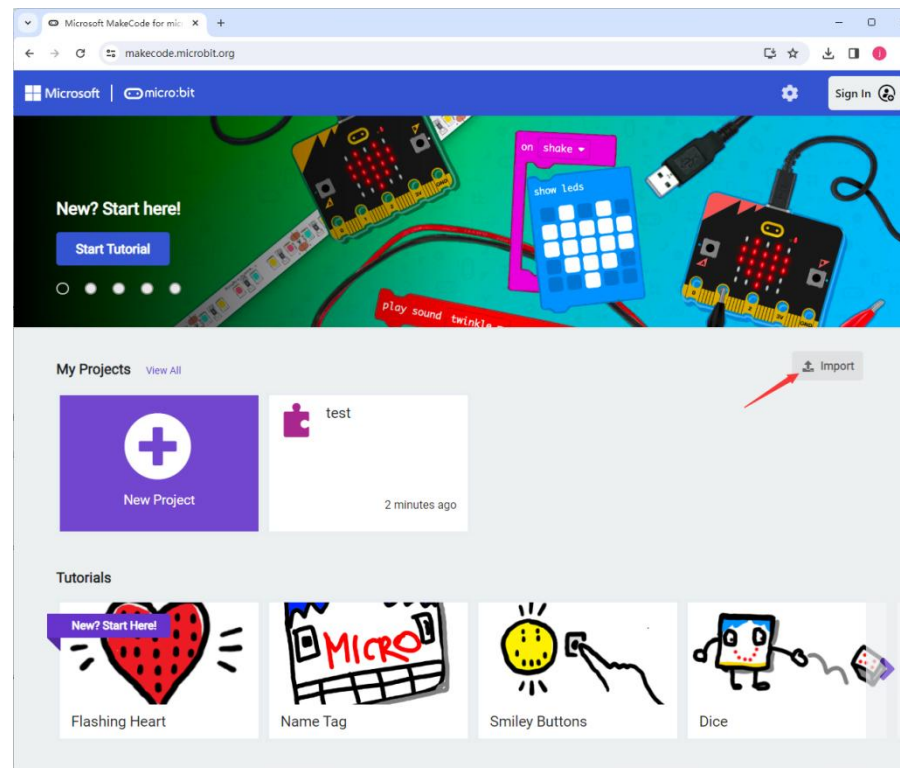
As for simple projects, it is recommended to complete the project by dragging code block.

As for complicated projects, it is recommended to complete the project by importing Hex code file.

LROBROYA

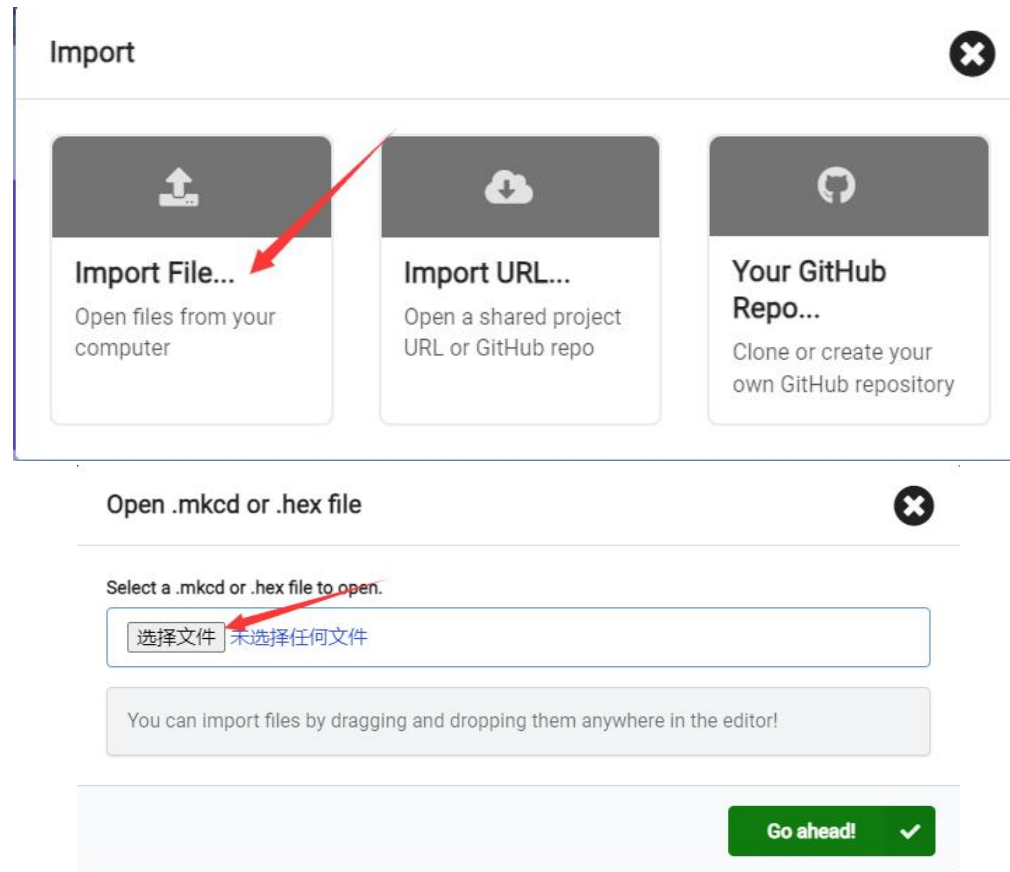
Next, we will take “Heartbeat” project as an example to introduce how to load code. Open web version of makecode.

Click “Import” button on the right of HOME page.



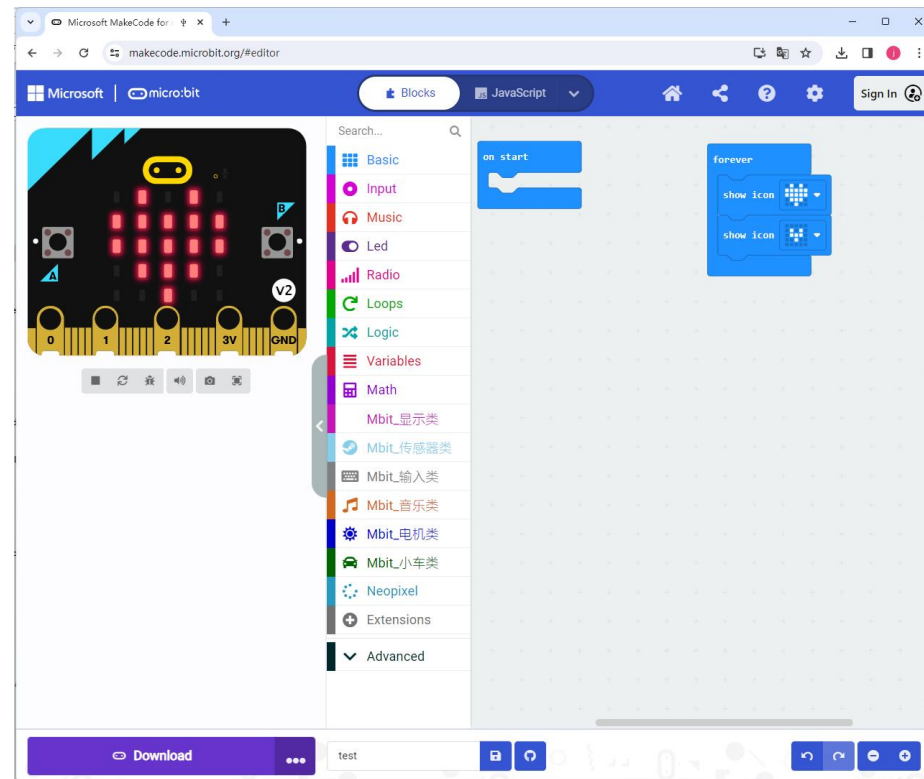
LROBROYA

In the pop-up dialog box, click "Import File".



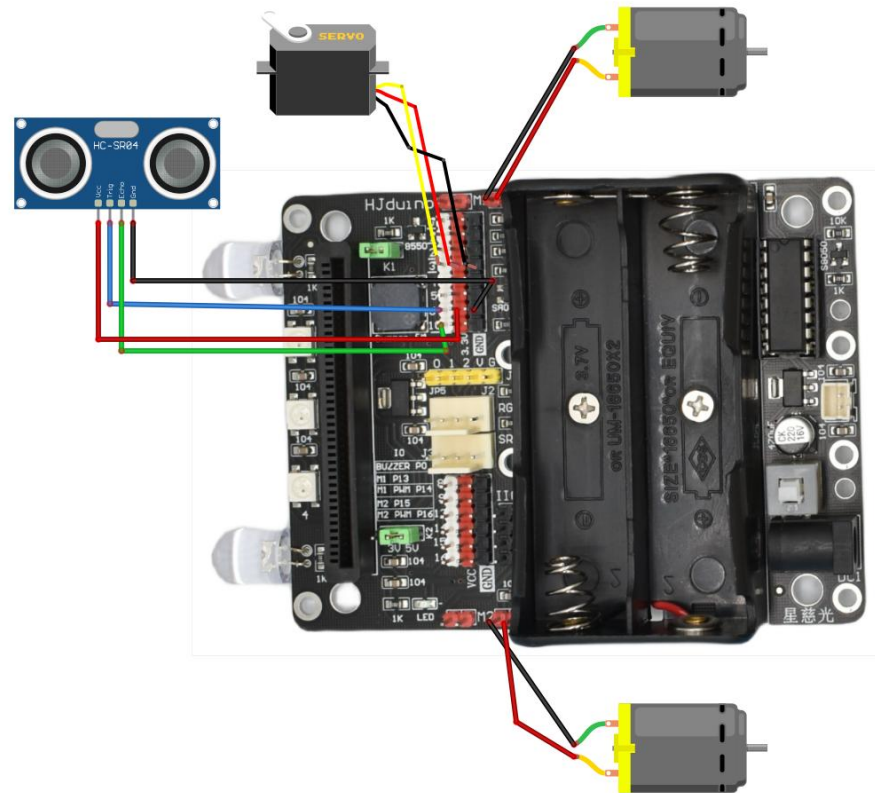
LROBRUYA

A few seconds later, the project is loaded successfully.



LROBRUYA

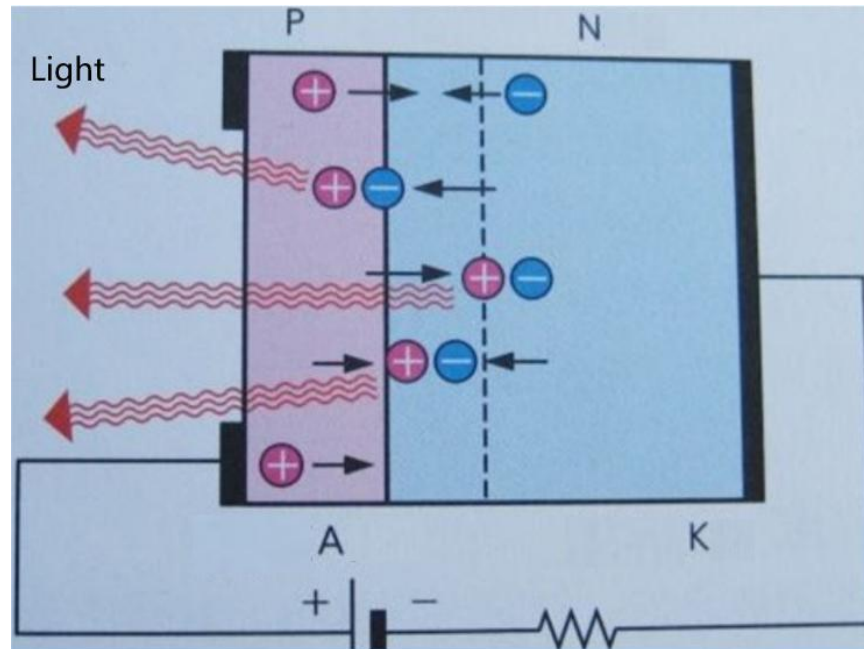
Wiring diagram



LROBRUYA

Lesson 1 Light two leds

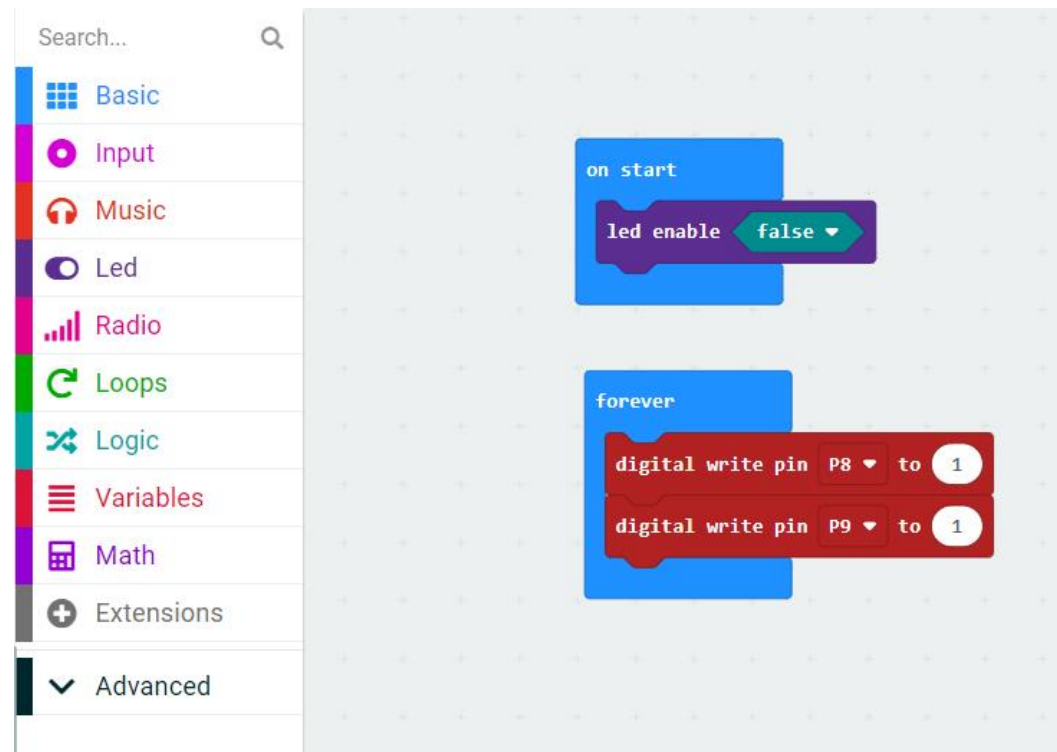
LED light wiring diagram



It can be seen from the figure that high level light is programmed.

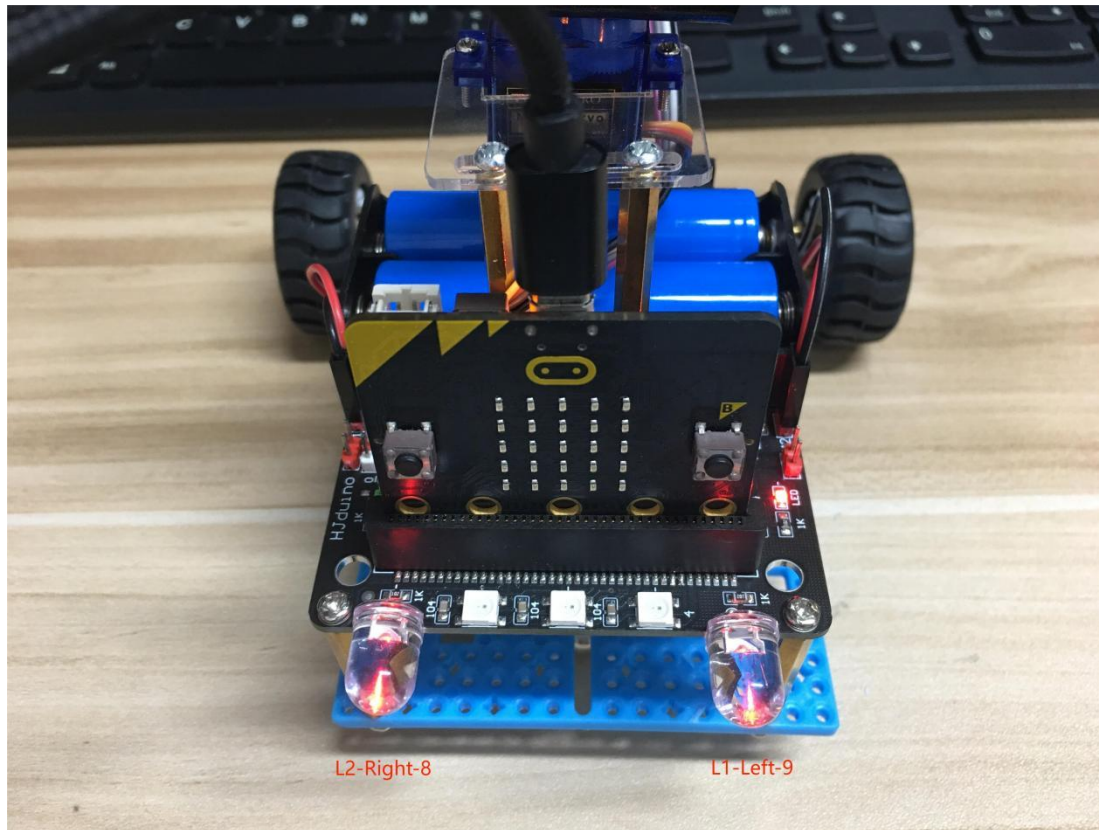
LROBROYA

Code block



LROBRUYA

Experimental demonstration



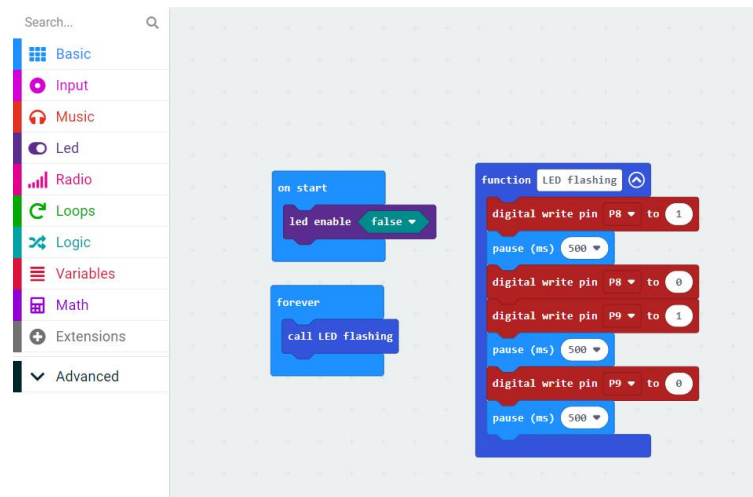
LROBROYA

Lesson 2 LED light flashing

LED light flashing

Through the learning of the first lesson, we know that the LED is lit at a high level, then the LED is extinguished for a low level, and the LED flicker can be realized by programming to alternately light off.

Code block



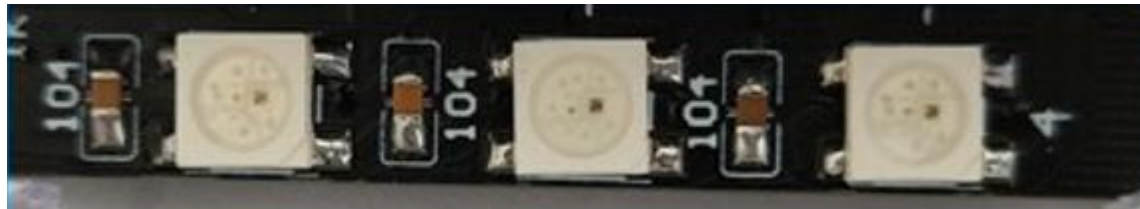


Lesson 3 RGB colored light strip

RGB color light

3 RGB color lights are set on the driver board, which is controlled by pin 12 IO port single bus.

Each RGB has a control chip inside, which means that each RGB lamp can be programmed to independently control the brightness and single color, color change effect it wants to display.



LROBROYA

Code block

The image shows a Scratch code editor with a script for an LED strip. The left sidebar contains a search bar and a list of categories: Basic, Input, Music, Led, Radio, Loops, Logic, Variables, Math, Neopixel, Extensions, and Advanced. The main workspace displays the following code:

```
on start
  set strip to Neopixel at pin P12 with 3 leds as RGB+W

forever
  strip show color purple
  pause (ms) 100
  strip clear
  strip show
  strip show color black
  pause (ms) 100
```

The code is written in a block-based language. The 'on start' block contains a 'set strip to Neopixel at pin P12 with 3 leds as RGB+W' block. The 'forever' loop contains a sequence of blocks: 'strip show color purple', 'pause (ms) 100', 'strip clear', 'strip show', 'strip show color black', and 'pause (ms) 100'.



Lesson 4 The buzzer sounds

Buzzer

Buzzer is an integrated structure of electronic sound device, using DC voltage power supply, widely used in computer, printer, copier, alarm, electronic toys and other electronic products as a sound device. Buzzers are mainly divided into two types: piezoelectric buzzers and electromagnetic buzzers. The buzzer is represented in the circuit by the letter "H" or "HA" (the old standard used "FM", "ZZG", "LB", "JD", etc.).

Attention: The buzzer requires K1 jumper cap gating, and the K1 can be removed when the buzzer is not used.

LROBRUYA



Code block

The buzzer on the driver board is connected to port P0 and integrates a passive buzzer that requires a PWM control signal to drive the sound.

LROBROYA

Search...

Basic

Input

Music

Led

Radio

Loops

Logic

Variables

Math

Extensions

Advanced

on start

forever

play tone Middle C for 1 ▾ beat

pause (ms) 200 ▾

play tone Middle D for 1 ▾ beat

pause (ms) 200 ▾

play tone Middle E for 1 ▾ beat

pause (ms) 200 ▾

play tone Middle F for 1 ▾ beat

pause (ms) 200 ▾

play tone Middle G for 1 ▾ beat

pause (ms) 200 ▾

play tone Middle A for 1 ▾ beat

pause (ms) 200 ▾

play tone Middle B for 1 ▾ beat



Lesson 5 Sound light alarm

Introduction

Through the previous study, we have already known the control of buzzer and LED light. This lesson realizes the experiment of sound and light alarm by combining the previous lessons.

Code block

LROBRUYA

Search...

- Basic
- Input
- Music
- Led
- Radio
- Loops
- Logic
- Variables
- Math
- Extensions
- Advanced

on start

- led enable false

function voice

- play tone High D for 1 beat
- pause (ms) 200
- play tone Middle B for 1 beat
- pause (ms) 200

forever

- call voice
- call LED

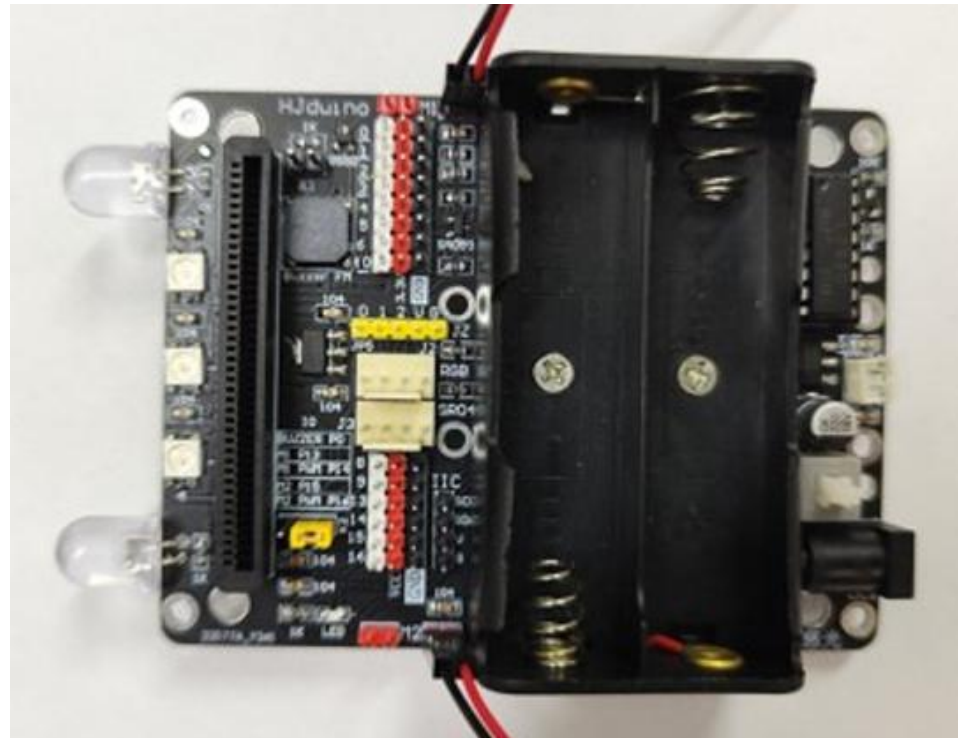
function LED

- digital write pin P8 to 1
- digital write pin P9 to 1
- pause (ms) 200
- digital write pin P8 to 0
- digital write pin P9 to 0

LROBRUYA

Lesson 6 Car automatic driving

Connection mode



LROBROYA

PWM speed regulation

Motor interface IO

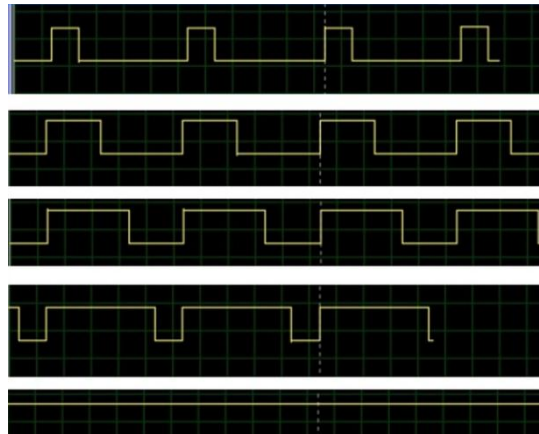
M1	P13
M1 PWM	P14
M2	P15
M2 PWM	P16

Set the pin output PWM value.



LROB RUYA

For the speed adjustment of the motor, we use the pulse width modulation (PWM) method, when controlling the motor, the power supply is not continuously supplied to the motor, but provides electrical energy in the form of square wave pulses at a specific frequency. Square wave signals with different duty ratios can speed regulate the motor, because the motor is actually a large inductor, which has the ability to hinder the input current and voltage mutation, so the pulse input signal is evenly distributed to the action time, so that, Changing the duty cycle of the square wave input on the enabling end ENA and ENB changes the voltage applied to both ends of the motor and thus changes the speed.



LROBRUYA

Code block

The image displays a Scratch code editor with a sidebar on the left and a main workspace on the right. The sidebar contains a search bar and a list of categories: Basic, Input, Music, Led, Radio, Loops, Logic, Variables, Math, Extensions, and Advanced. The main workspace contains the following code blocks:

- on start** (blue block)
 - call PWM** (blue block)
 - repeat 2 times** (green loop block)
 - do** (green block)
 - call advance** (blue block)
 - pause (ms) 1000** (blue block)
 - call back** (blue block)
 - pause (ms) 1000** (blue block)
 - call turnLeft** (blue block)
 - pause (ms) 1000** (blue block)
 - call turnRight** (blue block)
 - pause (ms) 1000** (blue block)
 - call stop** (blue block)
- forever** (blue block)

The workspace also contains several function blocks (blue blocks with a small 'f' icon):

- function PWM** (blue block)
 - analog write pin P14 (write only) to 500** (red block)
 - analog write pin P15 (write only) to 500** (red block)
- function advance** (blue block)
 - digital write pin P13 to 0** (red block)
 - digital write pin P15 to 0** (red block)
- function back** (blue block)
 - digital write pin P13 to 1** (red block)
 - digital write pin P15 to 1** (red block)
- function turnLeft** (blue block)
 - digital write pin P13 to 0** (red block)
 - digital write pin P15 to 1** (red block)
- function turnRight** (blue block)
 - digital write pin P13 to 1** (red block)
 - digital write pin P15 to 0** (red block)
- function stop** (blue block)
 - analog write pin P14 (write only) to 0** (red block)
 - analog write pin P16 (write only) to 0** (red block)



Lesson 7 Push start car

Introduction

When driving, the driver should start the car, and then refuel, the car is driven and the direction is controlled. Add a start button on the smart car to realize that after pressing the case, the car will start to drive.

Code block

Add the key start code to the car free driving program code.

LROBROYA

Search...

- Basic
- Input
- Music
- Led
- Radio
- Loops
- Logic
- Variables
- Math
- Extensions
- Advanced

on start

forever

if button A is pressed then

call PWM

repeat 2 times

do

call advance

pause (ms) 1000

call back

pause (ms) 1000

call turnLeft

pause (ms) 1000

call turnRight

pause (ms) 1000

call stop

function PWM

analog write pin P14 (write only) to 500

analog write pin P16 (write only) to 500

function advance

digital write pin P13 to 0

digital write pin P15 to 0

function back

digital write pin P13 to 1

digital write pin P15 to 1

function turnLeft

digital write pin P13 to 0

digital write pin P15 to 1

function turnRight

digital write pin P13 to 1

digital write pin P15 to 0

function stop

analog write pin P14 (write only) to 0

analog write pin P16 (write only) to 0

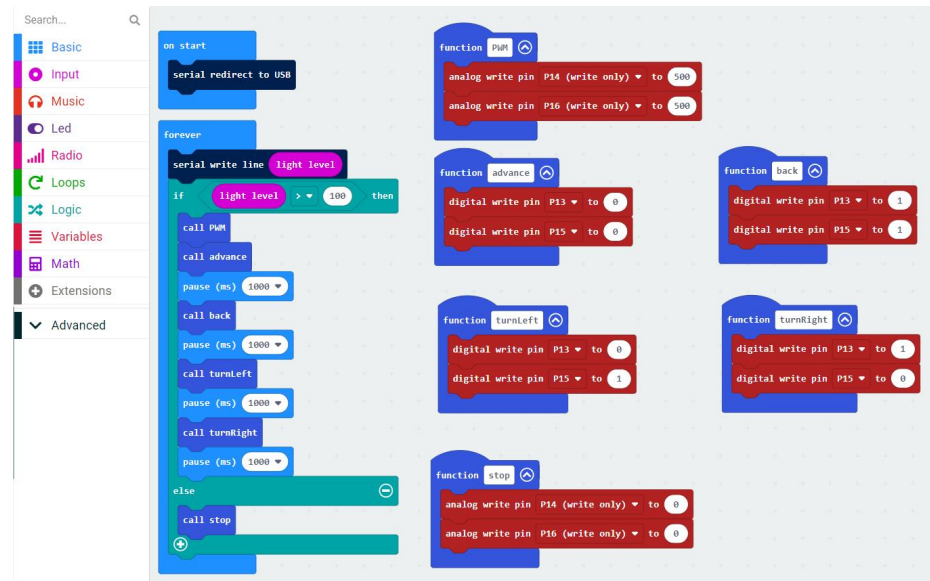
LROBROYA

Lesson 8 Light controlled car

Introduction

Light control code is added on the basis of free line code of car.

Code block





Lesson 9 Ultrasonic LED ranging display

Introduction

Ultrasonic wave is a mechanical wave with a very short wavelength, generally less than 2cm in air. It must rely on a medium to propagate and cannot exist in a vacuum (such as space). It spreads farther in water than in air, and can be used for cleaning, crushing, sterilization and so on. It has many applications in medicine and industry.

In terms of intelligent control, it is used for distance measurement and obstacle avoidance of intelligent car robot.

Connect

Vcc --> 3.3V

Trag --> 6

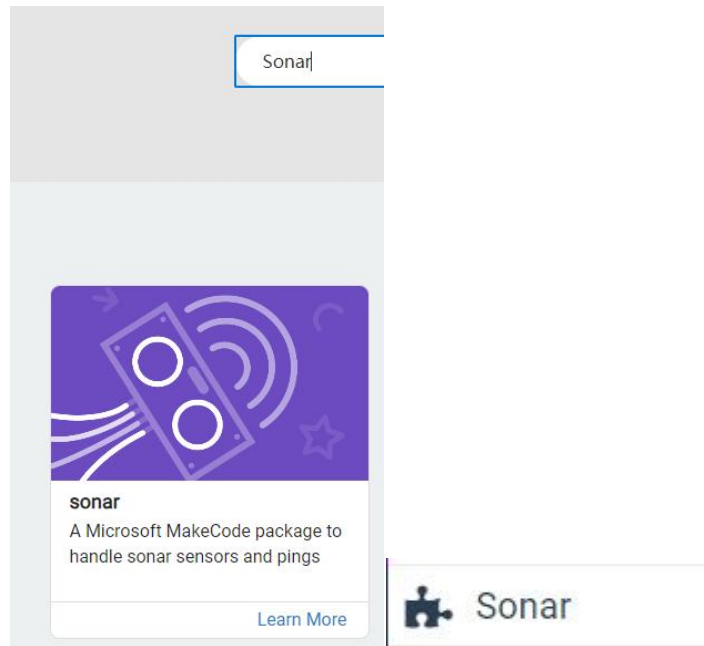
Echo --> 10

Gnd --> Gnd

LROBBUYA

Code block

Need to add Sonar ultrasonic module to programming software.



LROBRUYA





Lesson 10 Ultrasonic ranging serial port display

Introduction

Same as in lesson 9, except here it's using a serial port.

Connect

Vcc --> 3.3V

Trig --> 6

Echo --> 10


Gnd --> Gnd

Code block

Need to add Sonar ultrasonic module to programming software.


LROBROYA

Sonar|



sonar
A Microsoft MakeCode package to handle sonar sensors and pings

[Learn More](#)

 Sonar

LROBROYA





Lesson 11 Ultrasonic magic hand

Introduction

When the ultrasonic detection distance is greater than 20cm, the car moves forward, and when the ultrasonic detection distance is less than 20cm, the car moves backward.

Connect

Vcc --> 3.3V

Trag --> 6

Echo --> 10


Gnd --> Gnd

Code block

Need to add Sonar ultrasonic module to programming software.


LROBROYA

Sonar|

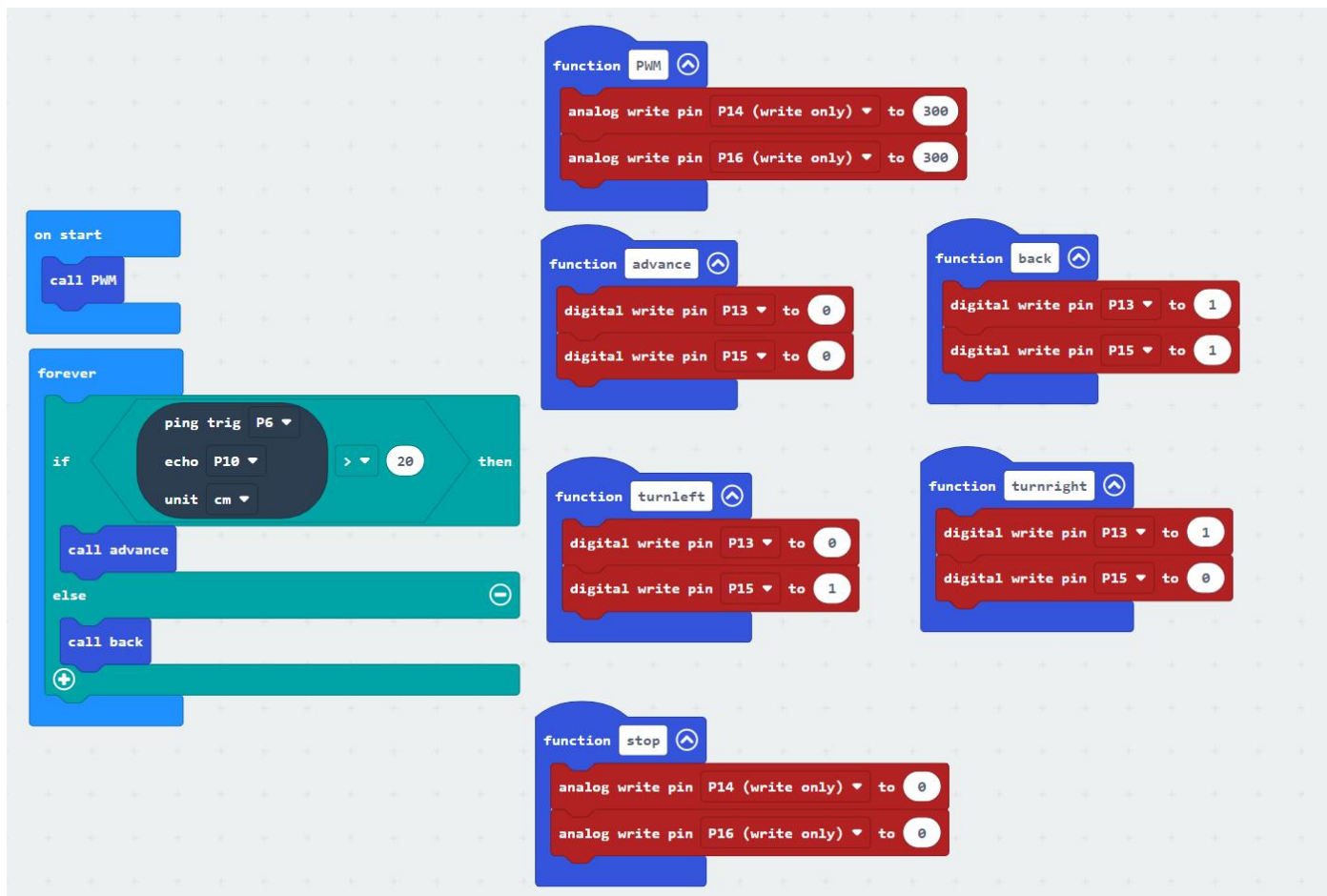


sonar
A Microsoft MakeCode package to handle sonar sensors and pings

[Learn More](#)

 Sonar

LROBRUYA





Lesson 12 Ultrasonic obstacle avoidance car

Introduction

When there is no obstacle in front of the robot car, the tank keeps walking straight. When the distance of the obstacle in front of the robot car is less than 20cm, the robot car stops, then detects whether there is an obstacle in the left front and right front, and then turns in the opposite direction.

Connect

Ultrasonic

Vcc --> 3.3V

Trig --> 6

Echo --> 10

Gnd --> Gnd

LROBROYA

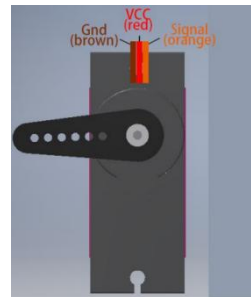
Servo

Gnd --> Gnd

Vcc --> 3.3V

signal --> 5

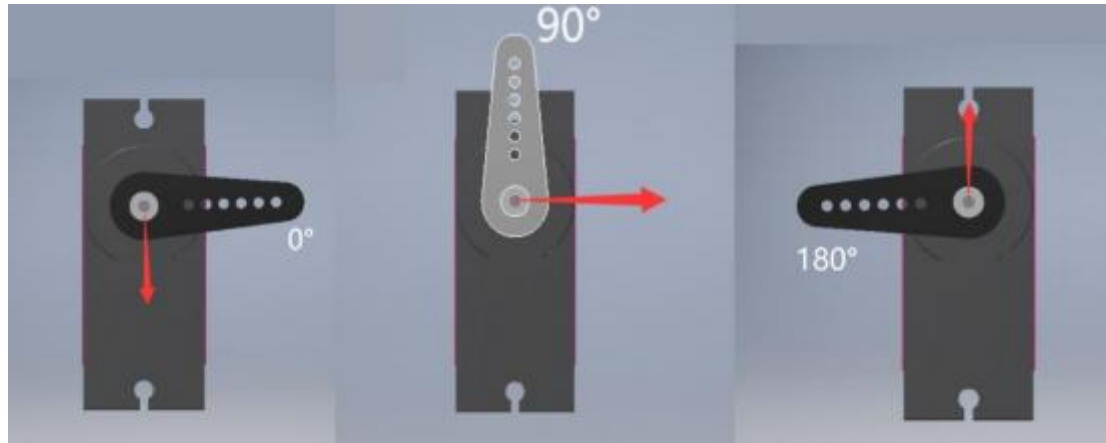
What is a servo motor



Servo motors are great devices that can turn to a specified position.

Usually, they have a servo arm that can turn 180 degrees. Using the Arduino, we can tell a servo to go to a specified position and it will go there. As simple as that!

LROBROYA



Servo motors were first used in the Remote Control (RC) world, usually to control the steering of RC cars or the flaps on a RC plane. With time, they found their uses in robotics, automation, and of course, the Arduino world.

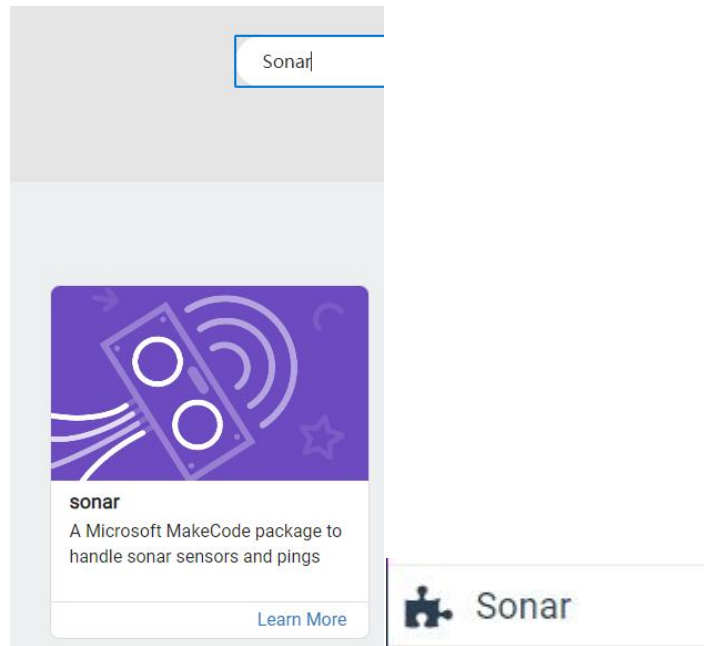
There are two ways to control a servomotor with Arduino. One is to use a common digital sensor port of Arduino to produce square wave with different duty cycle to simulate PWM signal and use that signal to control the positioning of the motor. Another way is to directly use the Servo function of the Arduino to control the motor.

The Arduino drive capacity is limited. So if you need to control more than one motor, you will need external power.

LROB RUYA

Code block

Need to add Sonar ultrasonic module to programming software.



LROBRUYA

