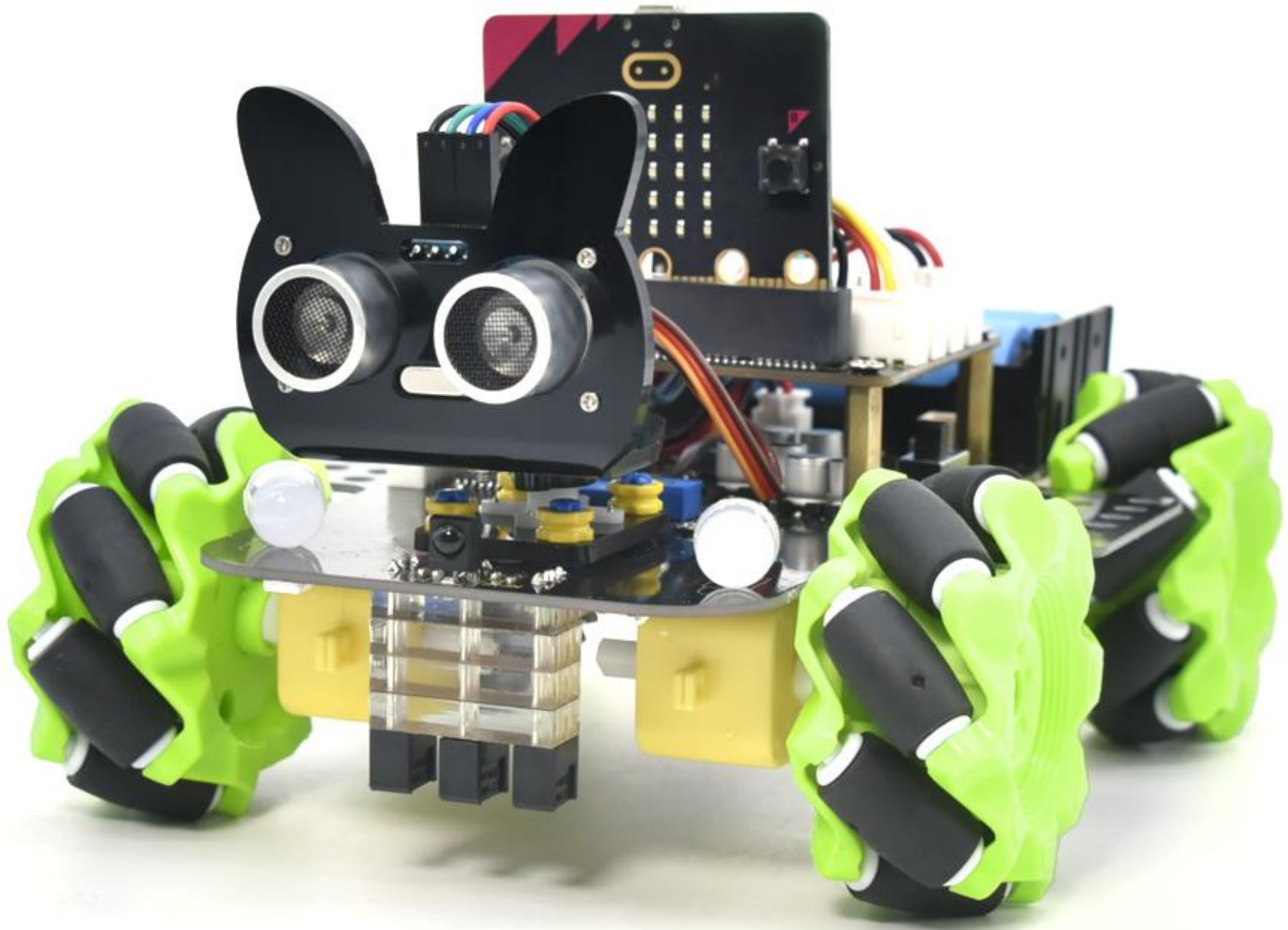


Keystudio Microbit 4WD Mecanum Robot Car V2.0



Contents

1. Introduction	3
2. Description	4
3.Parameters	4
4.Kit	4

5.Preparations	8
5.1BBC Micro:bit	8
(1)What is Micro:bit?	8
(2) Layout	8
(3) Pin out	9
(4) Notes for the Micro:bit V2 main board	10
5.2.Install Micro:bit driver	11
6.Keyestudio 4WD Mecanum Robot Car V2.0	11
6.1. Keyestudio 4WD Mecanum Robot Car V2.0	11
6.2. The Installation of Keyestudio 4WD Mecanum Robot Car V2.0	12
7.Get Started with Micro:bit	27
7.1 Write code and program	27
7.2.Makecode	33
7.3.Quick Download	34
7.4.Makecode Extension Libraries	38
7.5.Resources and Test Code	42
7.6.Import test code	42
7.7.Install CoolTerm	45
8. Projects	47
Project 1: Heart beat	47
Project 2: Light A Single LED	49
Project 3: 5 x 5 LED Dot Matrix	52
Project 4: Programmable Buttons	58
Project 5: Temperature Measurement	64
Project 6: Geomagnetic Sensor	71
Project 7: Accelerometer	79
Project 8: Light Detection	87
Project 9: Speaker	90
Project 10: Touch-sensitive Logo	92
Project 11: Microphone	96
Project 12: Bluetooth Wireless Communication	102
Project 13: Seven-Color LED	115
Project 14: 4 WS2812 RGB LEDs	118
Project 15: Servo	133

Project 16: Motor	138
Project 17: Line Tracking Sensor	145
Project 17.1: Detect Line Tracking Sensor	145
Project 17.2: Tracking Smart Car	150
Project 18: Ultrasonic Sensor	160
Project 18.1: Ultrasonic Ranging	160
Project 18.2: Ultrasonic Avoidance	165
Project 18.3: Ultrasonic Following	174
Project 19: IR Remote Control	180
Project 19.1: Decode IR Remote Control	180
Project 19.2: IR Remote Control	186
Project 20: Bluetooth Multi-purpose Smart Car	196
Project 20.1: Read Bluetooth Data	196
Project 20.2: Multi-purpose Smart Car	204
9. Common Problems	207
10. Resources	208

1. Introduction

Have you wondered to learn programming or have your own programming robot? Nowadays, programming has developed to a lower age group, and it will be a trend for everyone thanks to the spread of simple graphical programming platforms, from micro:bit to Arduino and Raspberry Pi. Maybe you haven't heard of them before. However, with the help of this product and tutorial, you can easily install a multi-functional programming car and experience the fun of being a maker.

Micro:bit is a highly integrated microcontroller with powerful functions and small size. It is very suitable to be applied in STEAM education for its functions to make robots, wearable devices and electronic interactive games via the combination of code programming and graphical programming.

This Keyestudio 4WD Mecanum Robot Car V2.0 is a smart DIY car dedicated to micro:bit. The smart car consists of a car body with extended functions, a PCB base plate with integrated motor drive sensors, 4 decelerating DC motors, Mecanum wheels, various modules and sensors as well as acrylic boards. Therefore, you can easily assemble a cool Mecanum wheel 4WD smart car by yourself, and then use Microsoft's online graphical programming platform MakeCode to program the micro:bit control board to control the car. In the process, you can not only experience the fun of creation but enhance hands-on ability and learn programming skills as well.

MakeCode for micro:bit is the most widely used graphical programming environment on the micro:bit official website. It is based on the graphical programming environment developed by Microsoft's open source project

MakeCode. This graphical programming can also be converted to code languages, python and javascript language, making it more accessible to learn programming. At the same time, MakeCode programming can be simulated or programmed for actual electronic components.

For your convenience, source code has been provided in every project, as well as code programming steps and code explanation in details. Hope you can better understand them.

2. Description

This product is a smart car based on Micro:bit. It integrates a host of functions such as ultrasonic following, line tracking, infrared control as well as Bluetooth control. There is a passive buzzer to play music, 4 WS2812RGB LEDs to display different colors, 2 seven-color lights to make direction lights for the car. This product uses two 18650 lithium batteries for power supply.

When installing and disassembling the battery, please pay attention to the positive and negative poles of the battery, and be sure not to reverse them. By the way, the motor speed of this product is adjustable.

In order to provide you with better experience, corresponding documents about installation and test code are also provided.

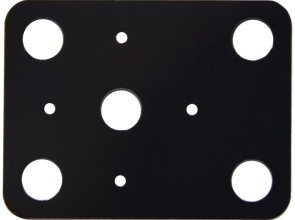
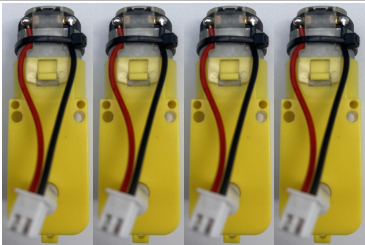
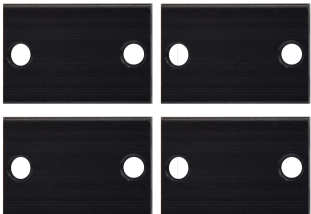
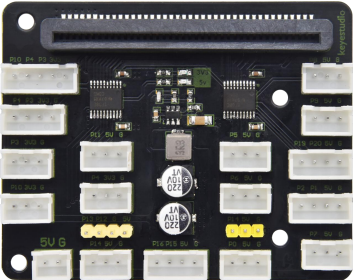
3.Parameters

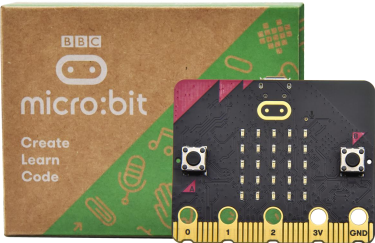
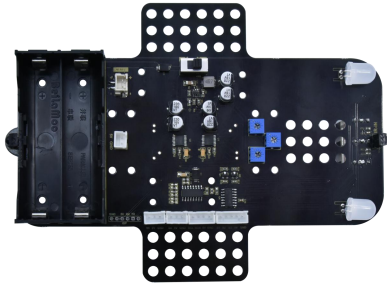
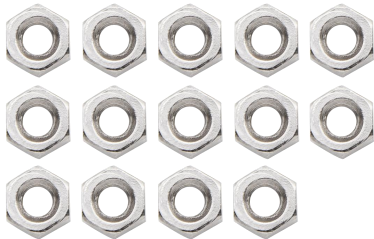
- ◆ Connector port input: DC 6V---9V
- ◆ Operating voltage of driver board system: 5V
- ◆ Standard operating power consumption: about 2.2W
- ◆ Maximum power: 12W
- ◆ Motor speed: 200RPM
- ◆ Working temperature range: 0-50°C
- ◆ Size: 120*120*120mm
- ◆ Environmental protection attributes: ROHS

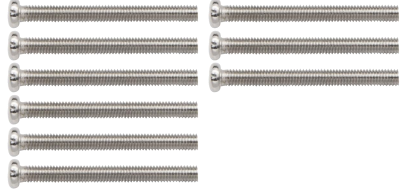
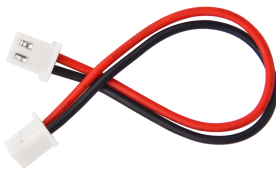
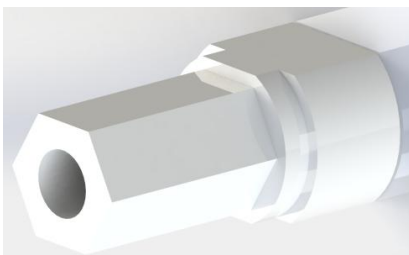
Note: The working voltage of micro:bit is 3.3V, and the driver shield integrates a 3.3V/5V communication conversion circuit.

4.Kit

#	Picture	Name	QTY
---	---------	------	-----

1		Acrylic Board T=3mm	1
2		Acrylic Board with Lego Holes T=3mm	1
3		Motor Plate	4
4		Motor	4
5		23*15*5MM Fixing Board	4
6		Servo	1
7		Mecanum Wheels (A direction)	2
8		Mecanum Wheels (B direction)	2
9		keyestudio Micro: bit Expansion Board	1

10		micro:bit V2.0 Mainboard (KS4034、KS4034F)	1
11		Keystudio Mecanum Car Lower Plate (include motor driver)	1
12		M3*20MM Dual-pass Copper Pillar	4
13		4265c Lego Part	4
14		43093 Lego Part	4
15		Acrylic Gasket	1
16		M3*6MM Flat Head Screw	10
17		HC-SR04 Ultrasonic Sensor	1
18		M3*8MM Flat Head Screw	10
19		M3 Nickle-plated Nut	10

20		M3*30MM Round Head Screw	9
21		M2 Nickle-plated Nut	3
22		M2*8MM Round Head Screw	3
23		M1.4 Nickle-plated Nut	6
24		M1.4*10MM Round Head Screw	6
25		M2.3*16MM Round Head Screw	4
26		Remote Control (KS4034F, KS4035F)	1
		Remote Control (without batteries) (KS4034, KS4035)	
27		Plastic String 3*100MM	5
28		USB Cable	1
29		HX-2.54 2P DuPont Wire 100mm	1
30		XH2.54 5P DuPont Wire 100mm	1
31		HX-2.54 4P DuPont Wire 50mm	1
32		HX2.54mm-4P to 2.54 F-F DuPont Wire 150mm	1
33		XH2.54 3P DuPont Wire 50mm	2
34		3*40MM Screwdriver	1
35		TT Coupling	4

36		M1.2*5MM Round Head Self-tapping Screw	6
----	---	---	---

5.Preparations

5.1BBC Micro:bit

(1)What is Micro:bit?

Micro:bit is an open source hardware platform based on the ARM architecture launched by British Broadcasting Corporation (BBC) together with ARM, Barclays, element14, Microsoft as well as other institutions. The core device is a 32-bit Arm Cortex-M4 with FPU micro-processing.

It is just the size of a credit card but it's very powerful. The Micro:bit main board is equipped with a host of components such as a 5*5 LED dot matrix, 2 programmable buttons, an accelerometer, a compass, a thermometer, a touch-sensitive logo and a MEMS microphone, a Bluetooth module of low energy as well as a buzzer and so on, making it empower to play a variety of sounds without external devices.

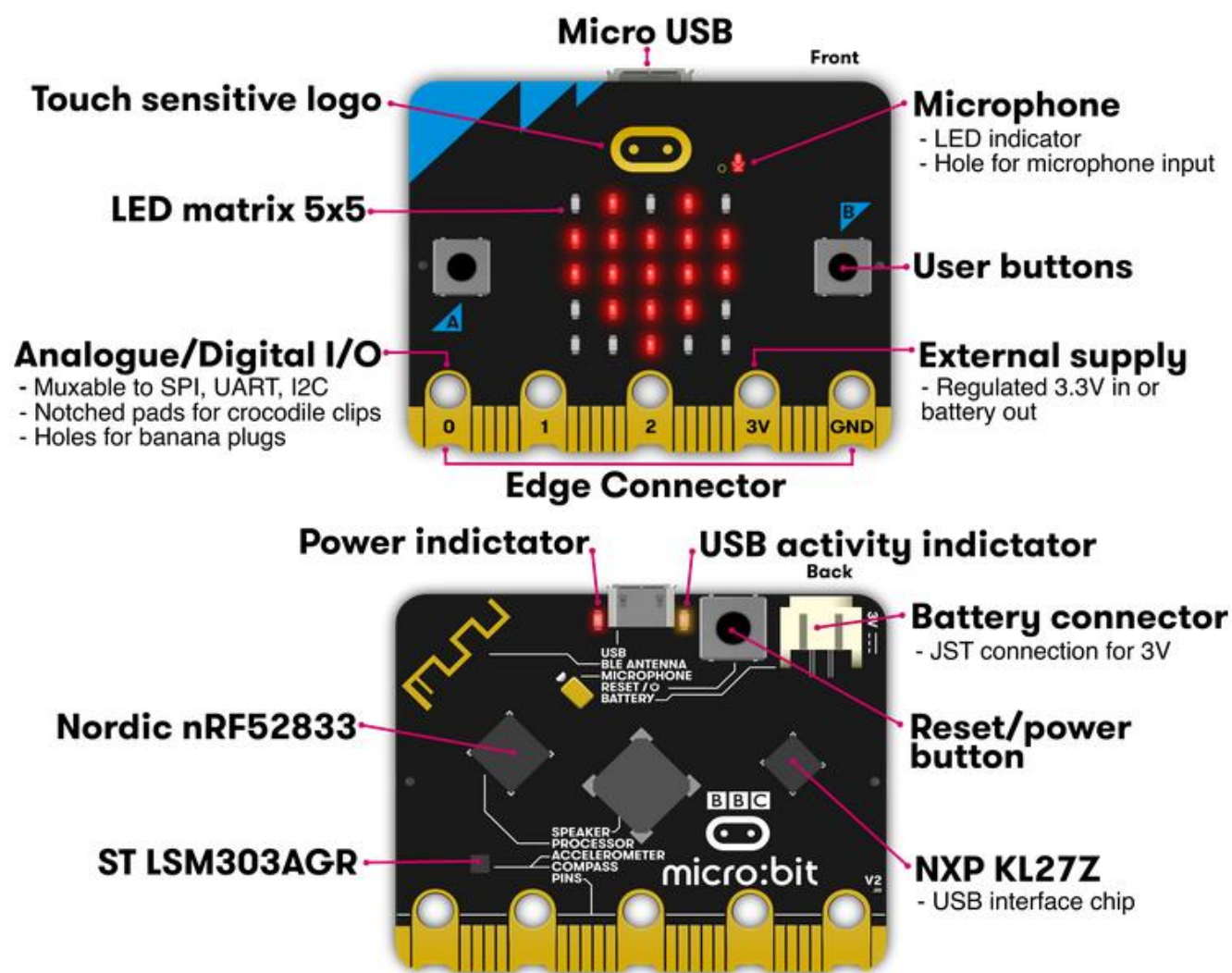
Moreover, this board supports a sleeping mode to lower power consumption of batteries and it can be entered if users long press the Reset & Power button on the back of it.

Micro: Bit development board is easy to use and expand, the bottom gear design of the gold finger can be used to interact with various electronic components via fixed alligator clip. It is capable of reading the data of sensors, controlling servos and RGB lights and inserting an expansion board so as to connect various sensors.

Furthermore, it also supports a variety of codes and graphical programming platforms, and is compatible with almost all PCs and mobile devices and a free-installation driver. It has high integration electronic modules and a serial port monitoring function for easy debugging.

The board is widely used in programming video games, interactions between light and sound, robots controls, scientific experiments, wearable devices as well as some cool inventions like robots and musical instruments.

(2) Layout



For more information, please resort to following links:

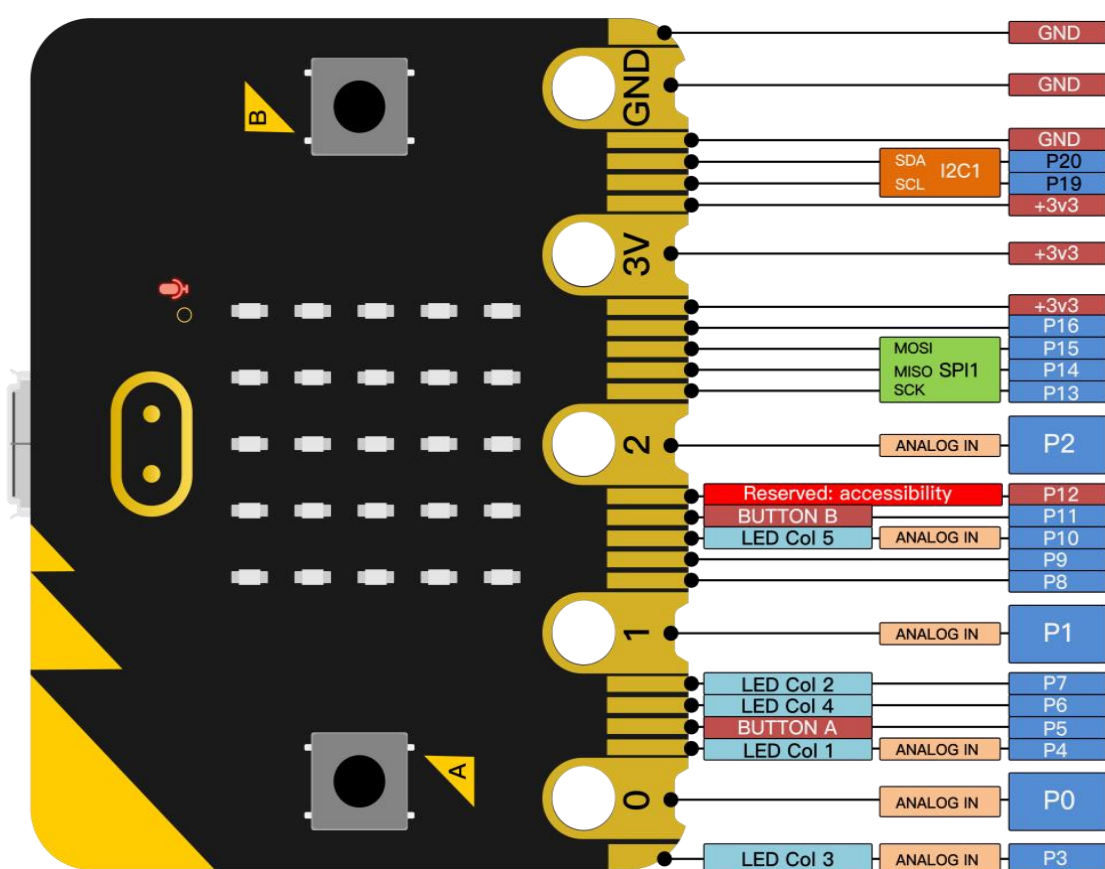
<https://tech.microbit.org/hardware/>

<https://microbit.org/new-microbit/>

<https://www.microbit.org/get-started/user-guide/overview/>

<https://microbit.org/get-started/user-guide/features-in-depth/>

(3) Pin out



Functions:

GPIO	P0, P1, P2, P3, P4, P5, P6, P7, P8, P9, P10, P11, P12, P13, P14, P15, P16, P19, P20
------	---

ADC/DAC	P0, P1, P2, P3, P4, P10
IIC	P19 (SCL) , P20 (SDA)
SPI	P13 (SCK) , P14 (MISO) , P15 (MOSI)
PWM (used frequently)	P0, P1, P2, P3, P4, P10
PWM (not frequently used)	P5、 P6、 P7、 P8、 P9、 P11、 P12、 P13、 P14、 P15、 P16、 P19、 P20
Occupied	P3(LED Col3), P4(LED Col1), P5(Button A), P6(LED Col4), P7(LED Col2), P10(LED Col5), P11(Button B)

Please browse the official website for more details: <https://tech.microbit.org/hardware/edgeconnector/>
<https://microbit.org/guide/hardware/pins/>

(4) Notes for the Micro:bit V2 main board

- It is recommended to cover with a silicone protector to prevent short circuit for its sophisticated electronic components.
- Its IO port is very weak in driving since it can merely handle current less than 300mA. Therefore, do not connect it with devices operating in a large current, such as MG995 servo and DC motor or it will get burnt. Furthermore, you must figure out the current requirements of the devices before you use them and it is generally recommended to use the board together with a Micro:bit expansion board.
- It is recommended to power the main board via the USB interface or the battery of 3V. The IO port of this board is 3V, so it does not support sensors of 5V. If you need to connect sensors of 5 V, a Micro: Bit expansion board is required.
- When using pins(P3, P4, P6, P7 and P10)shared with the LED dot matrix, blocking them from the matrix or the LEDs may display randomly and the data about sensors connected maybe wrong.
- Pin 19 and 20 can not be used as IO ports though the Makecode shows they can. They can only be used as I2C communication.
- The battery port of 3V cannot be connected with battery more than 3.3V or the main board will be damaged.
- Forbid to operate it on metal products to avoid short circuit.

To put it simple, Micro:bit V2 main board is like a microcomputer, which has made programming at our fingertips and enhanced digital innovation. And as for programming environment, BBC provides a website: <https://microbit.org/code/>, which has a graphical MakeCode program easy for use.

5.2.Install Micro:bit driver

Micro: Bit can be installed without the USB driver. However, if your computer fails to recognize the main board, you can install the diver too.

Just enter the file folder  2.Micro: bit Driver  mbed_usb_2020_x64_1212.exe  microbit driver installation manual

6.Keyestudio 4WD Mecanum Robot Car V2.0

It is a programmable car based on BBC micro:bit. It integrates a motor driver, a line tracking sensor and an IR receiver into the base plate, which also contains an ultrasonic sensor, a servo, 2 seven-color lights as well as 4 WS2812 RGB lights. The wiring is not complicated and it has Lego jacks to facilitate connection with other peripheral devices. Abundant hardware resources will enable you to master more knowledge and skills to create more technological inventions.

6.1. Keyestudio 4WD Mecanum Robot Car V2.0

This car can help you to better learn how to use the Micro:bit and make electronic knowledge accessible to you.

Functions

Sensor	Seven-color light	Decelerating DC motor	Servo	Ultrasonic sensor	Line Tracking Sensor	IR Receiver	WS2812 RGB light	Power switch
QTY	2	4	1	1	3	1	4	1

Note: the line tracking sensor, WS2812 RGB lights, IR receiver and motor driver are integrated in the base plate.

Pins:

Pin on Micro:bit	Sensor
P10 P4 P3	Line Tracking Sensor
P14	Servo
P7	4 WS2812RGB Lights
P0	IR Receiver
P15P16	Ultrasonic Sensor

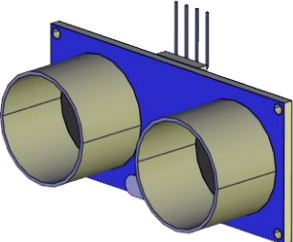
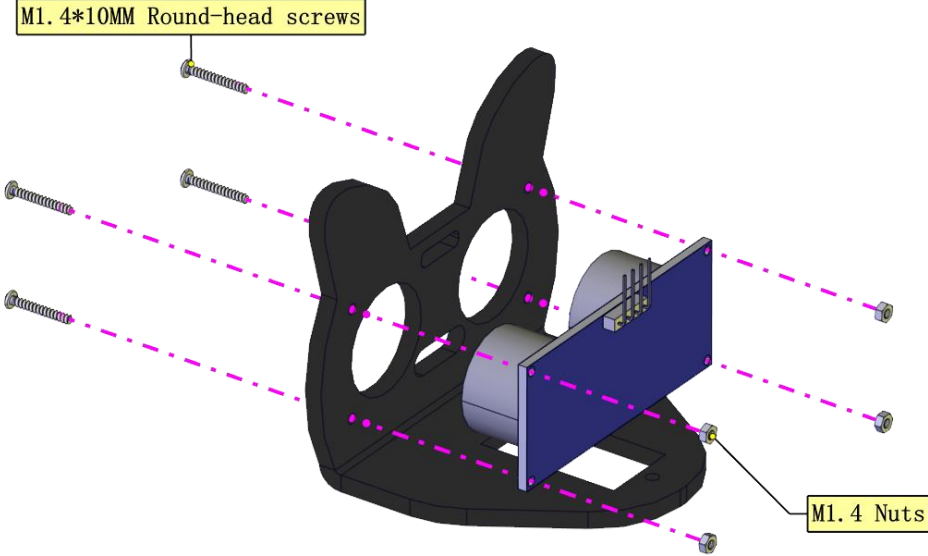
P19 P20 (IIC)	4 Motor drivers
P19 P20 (IIC)	2 Seven-color lights

Power supply and Battery

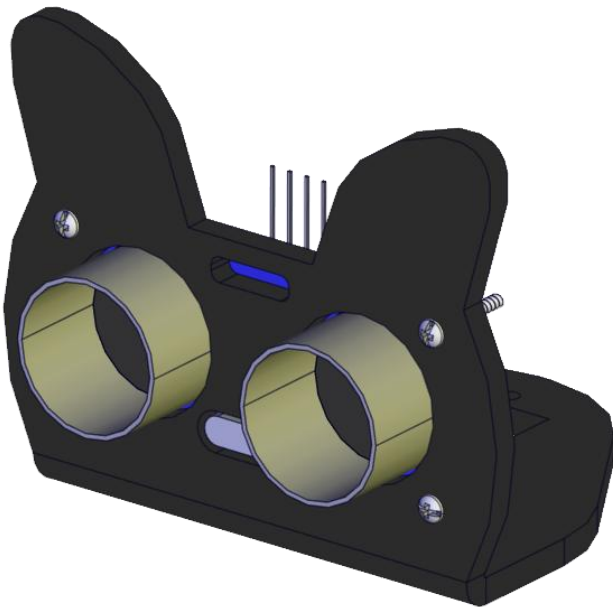
The keystudio 4WD Mecanum Robot Car is powered by two 18650 batteries. The battery holder of the car is compatible with any type of 18650 lithium battery (rechargeable). You can use a universal battery charger to charge the 18650 lithium battery.

Note: This product does not contain batteries.

6.2. The Installation of Keystudio 4WD Mecanum Robot Car V2.0

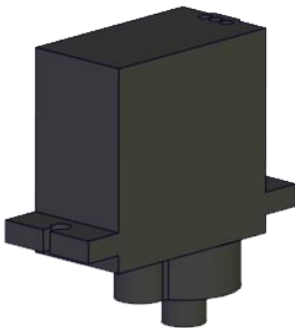
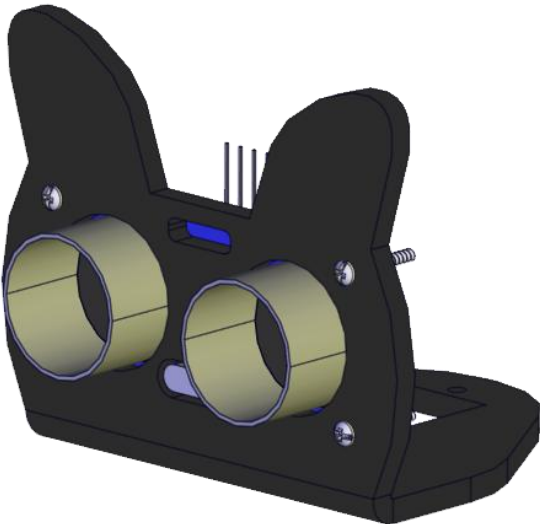
Part 1	
Compo nents Needed	 Acrylic Boards  Ultrasonic Sensor ×1  M1.4*10MM Round-head screws ×4  M1.4 Nuts ×4
Installati on Diagram	 M1.4*10MM Round-head screws M1.4 Nuts

Prototy
pe



Part 2

Compo
nents
Needed



Keyestudio Servo

×1



M2*8MM Round-head screws

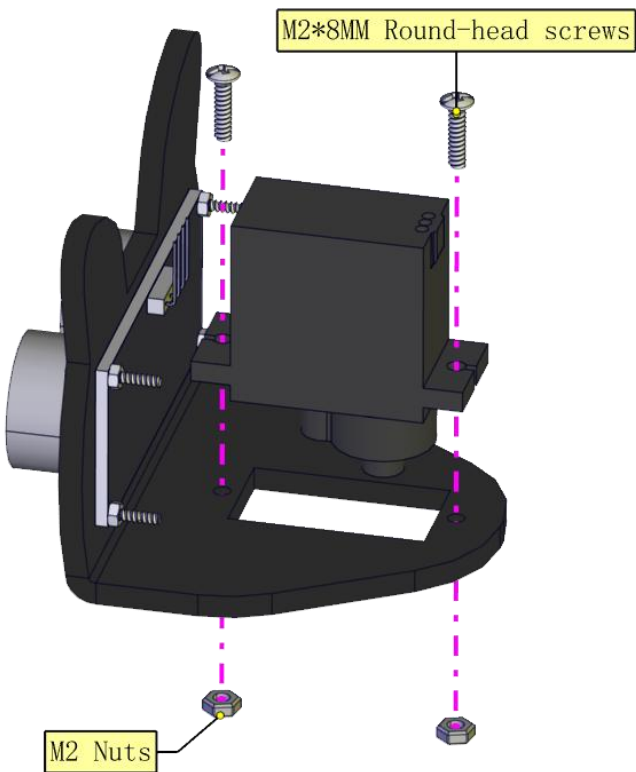
×2



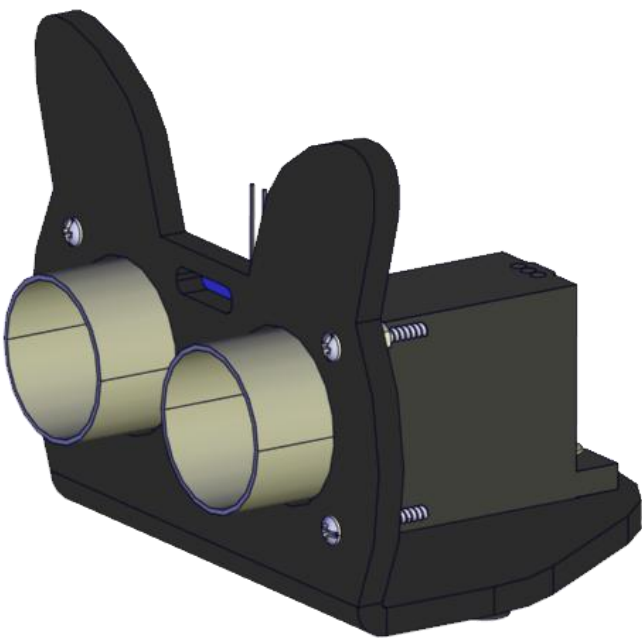
M2 Nuts

×2

Installati
on
Diagram

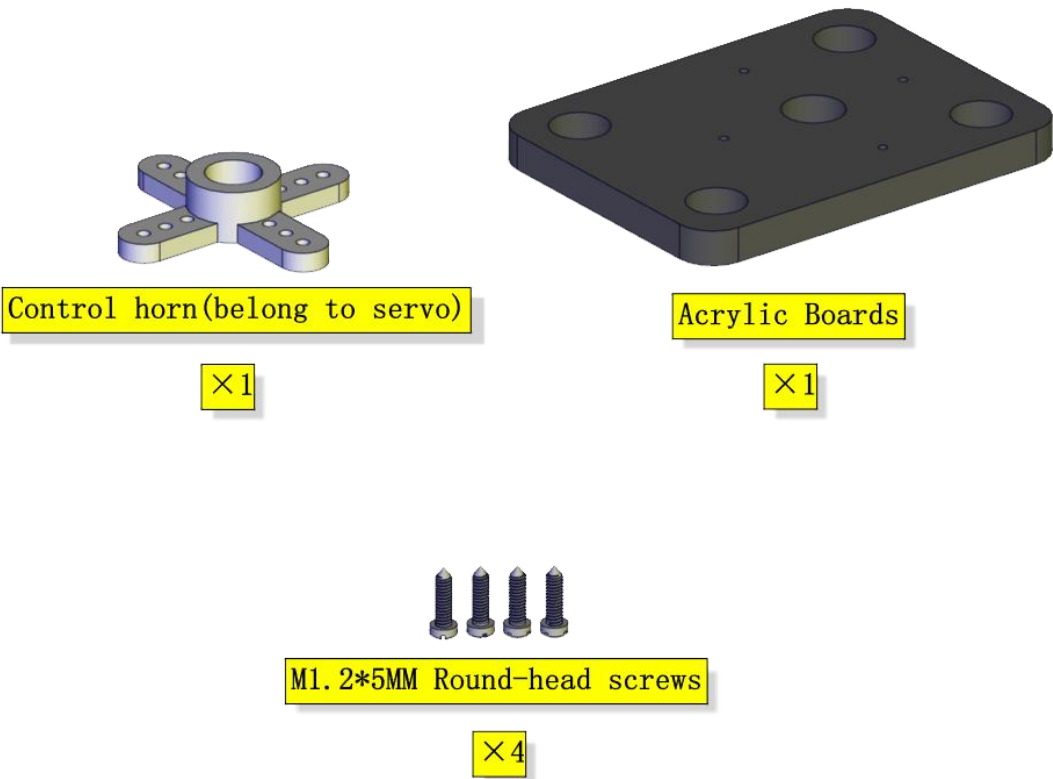


Prototype

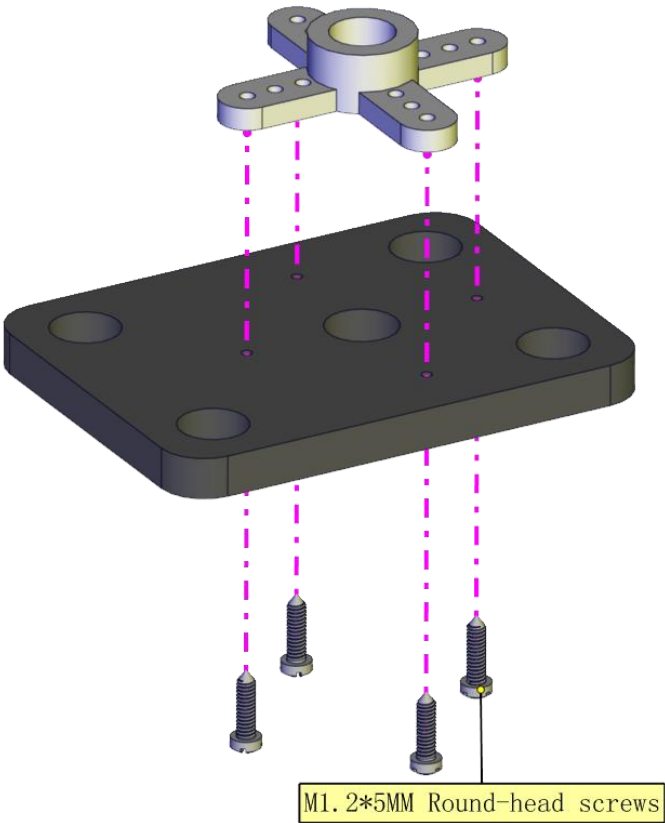


Part 3

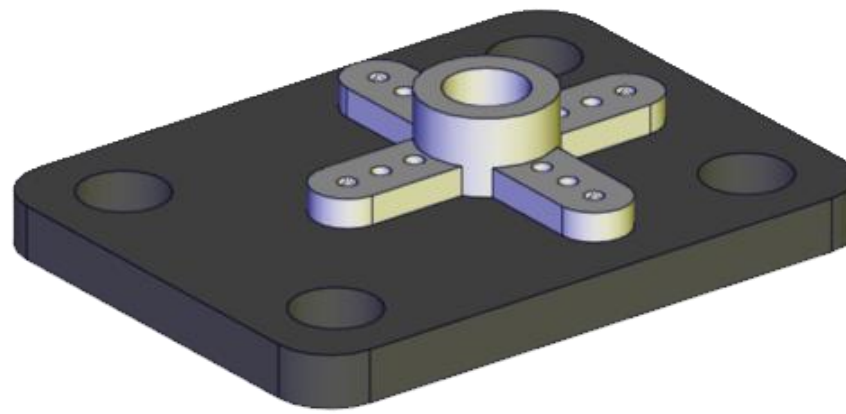
Components Needed



Installation Diagram

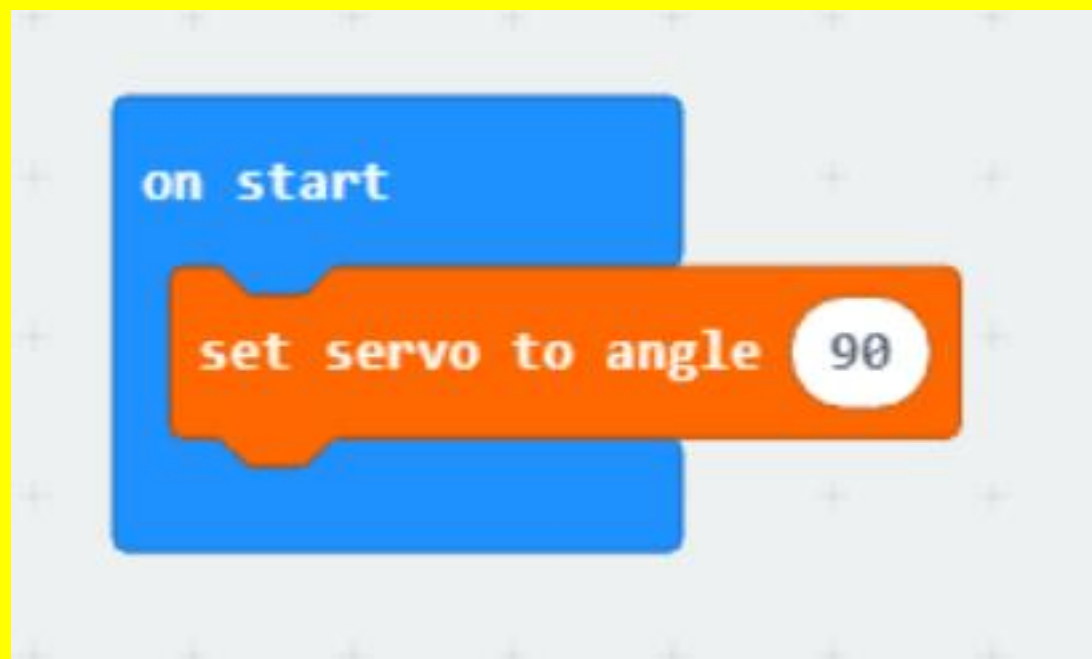


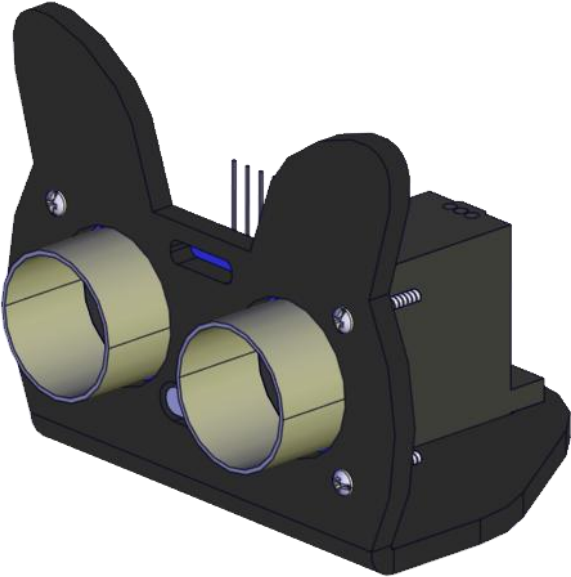
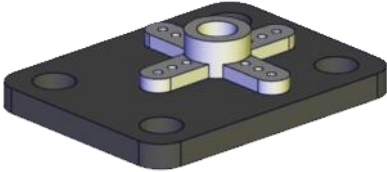
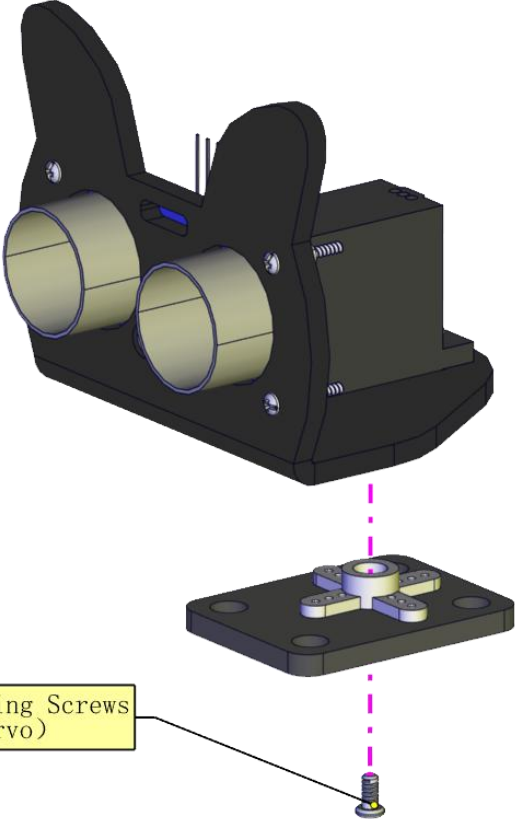
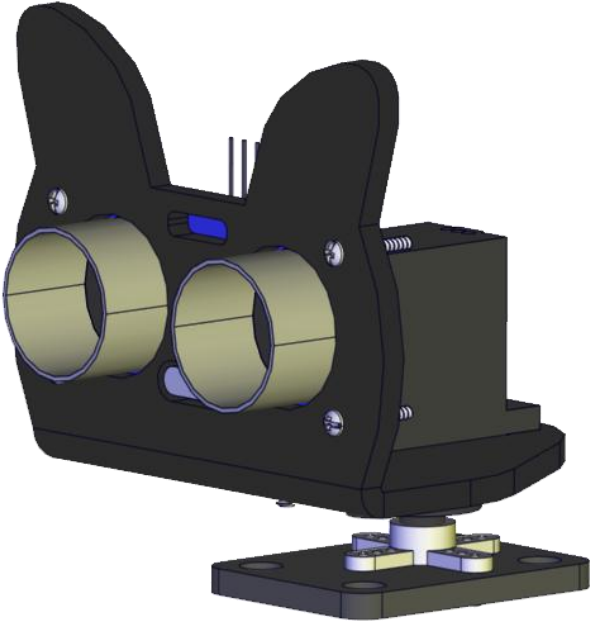
Prototype



Part 4 (adjust the angle of the servo first)

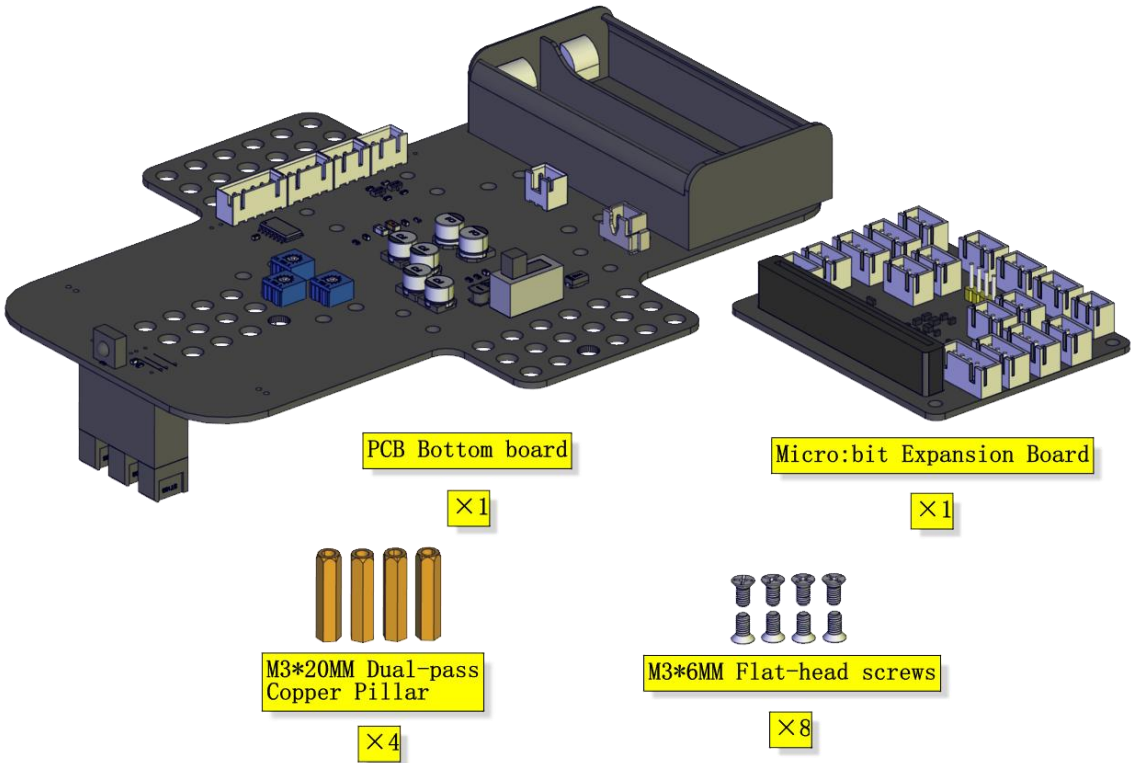
Adjust the angle of the servo to 90 degrees



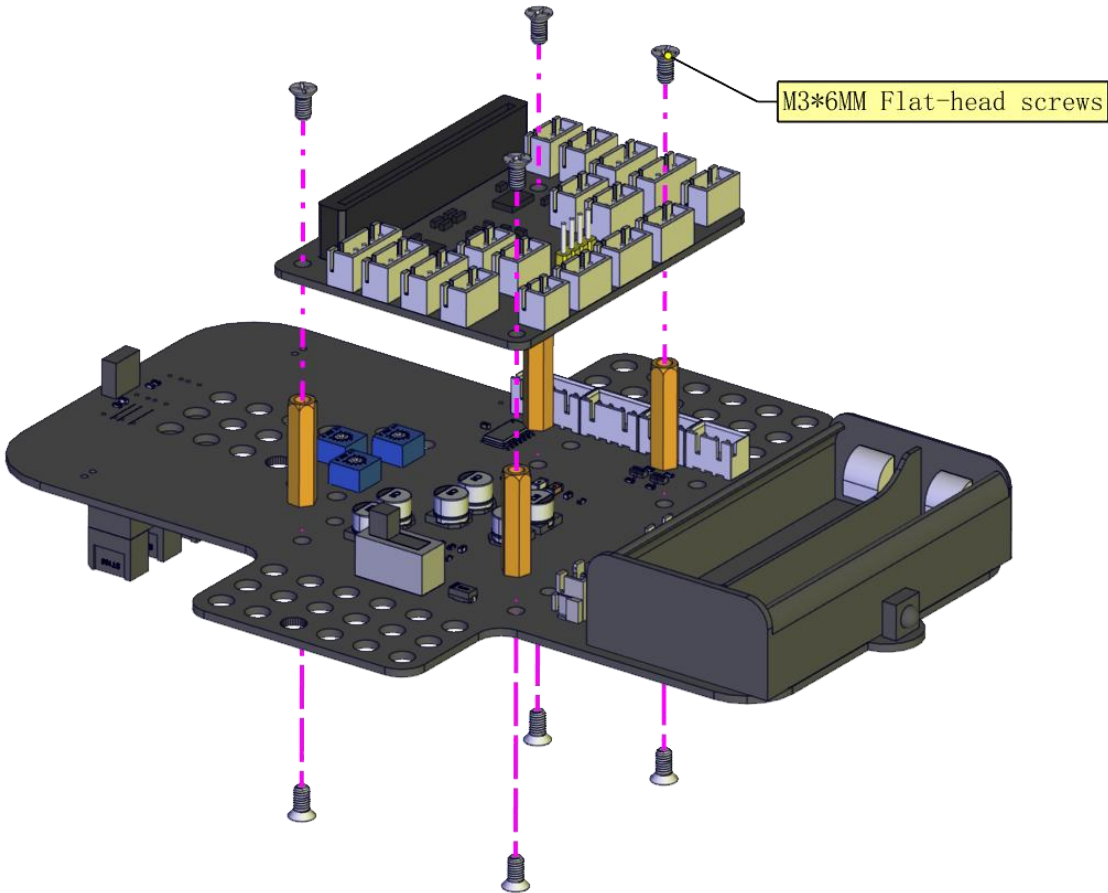
Compon ents Needed	   M2*4 Self-tapping Screws (belong to servo) ×1
Installati on Diagram (mind the installati on directio n)	 M2*4 Self-tapping Screws (belong to servo)
Prototy pe	

Part 5

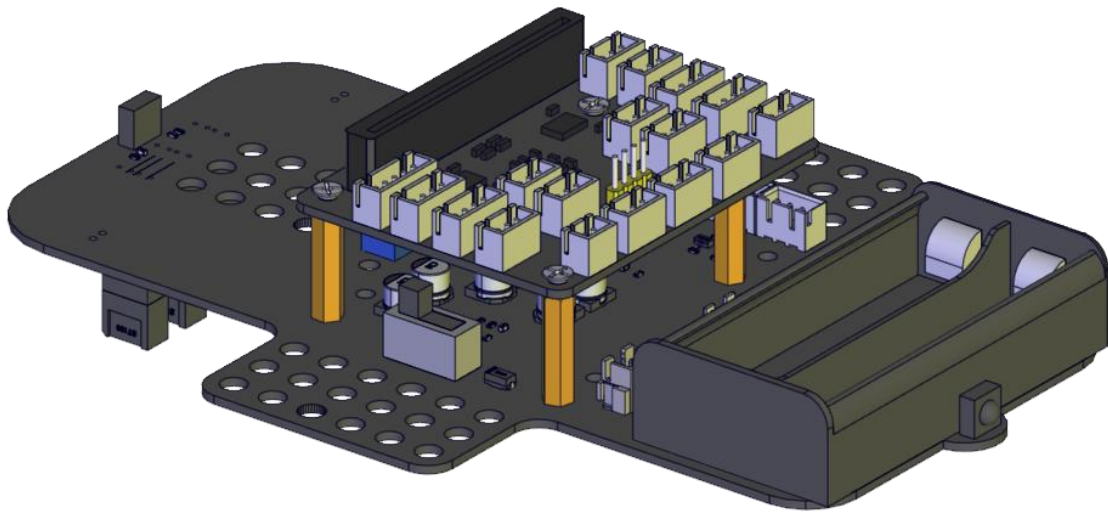
Compon
ents
Needed



Installati
on
Diagram

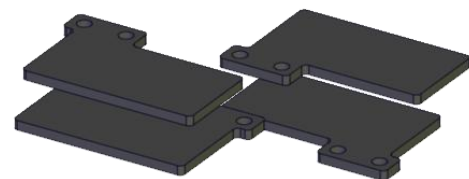
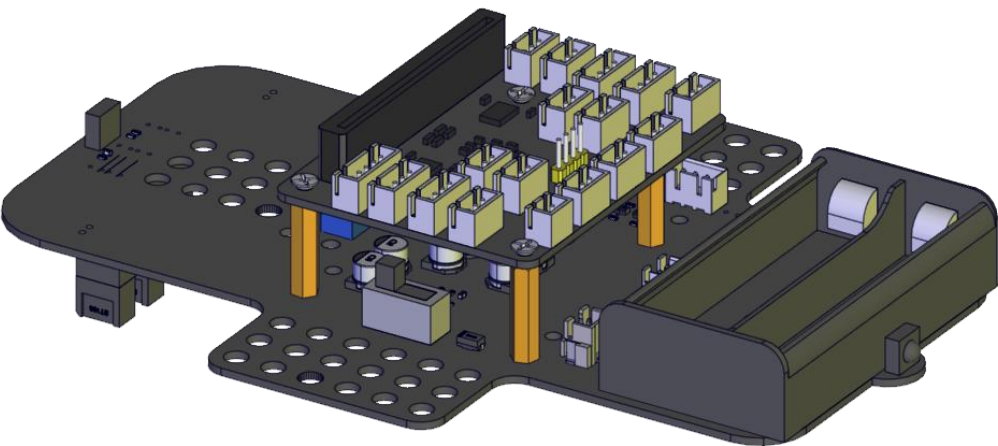


Prototy
pe



Part 6

Compon
ents
Needed



Acrylic Boards



Fixed parts

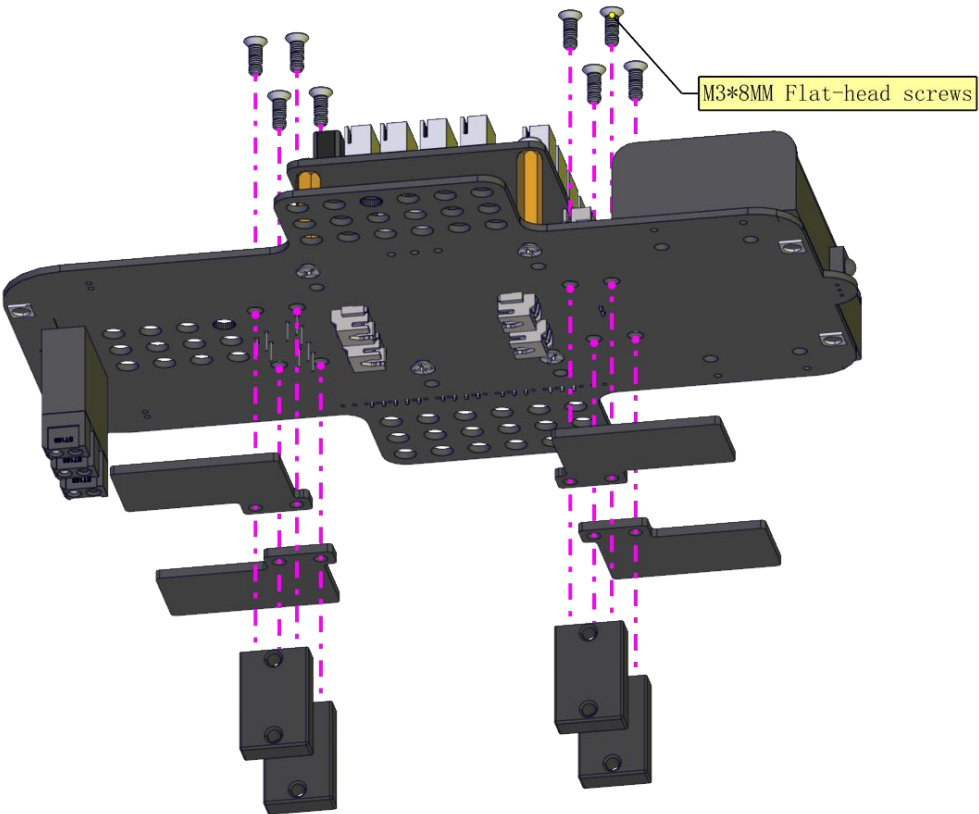
× 4



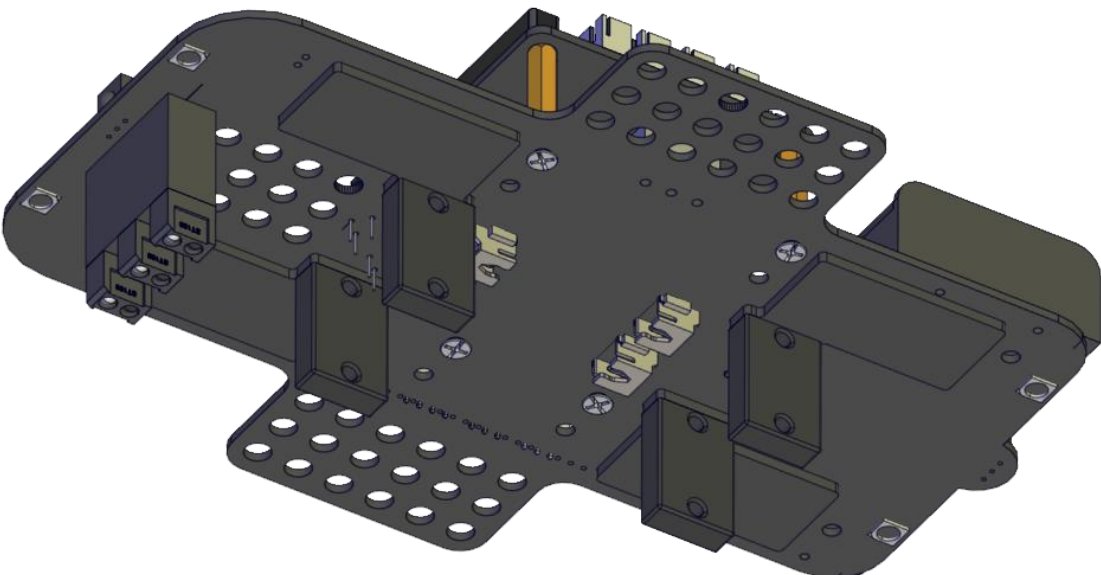
M3*8MM Flat-head screws

× 8

Installati
on
Diagram

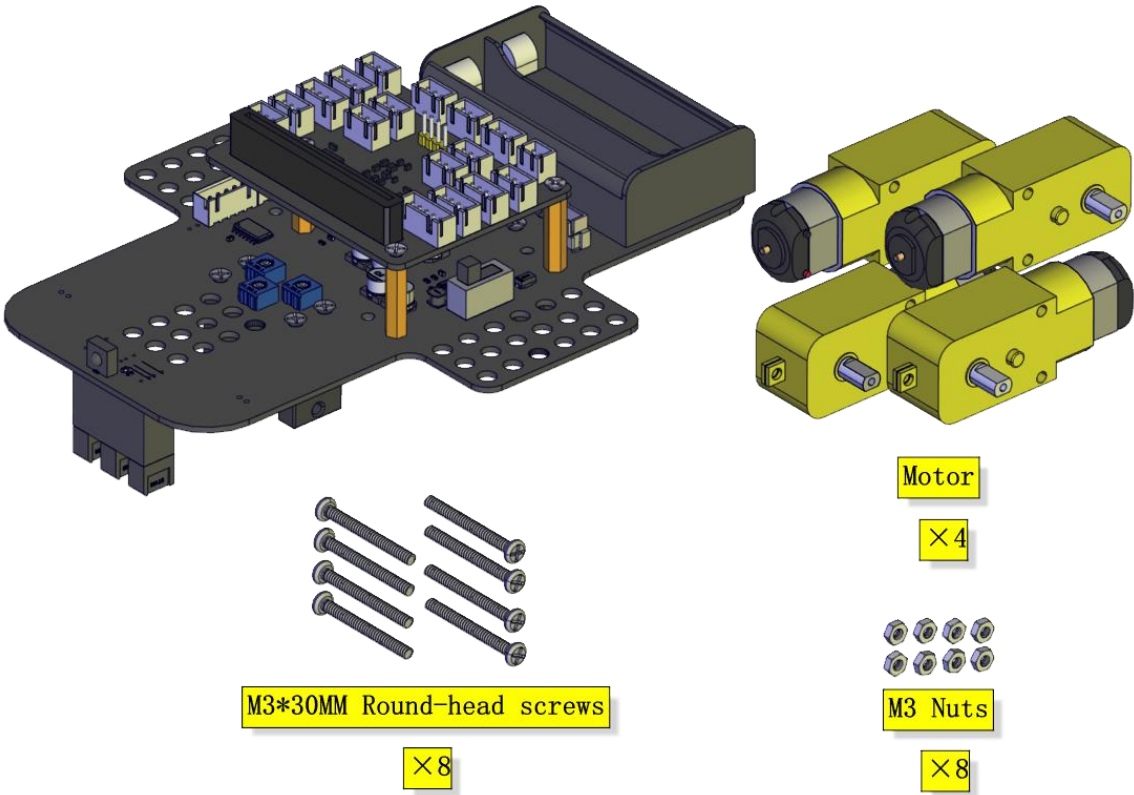


Prototy
pe

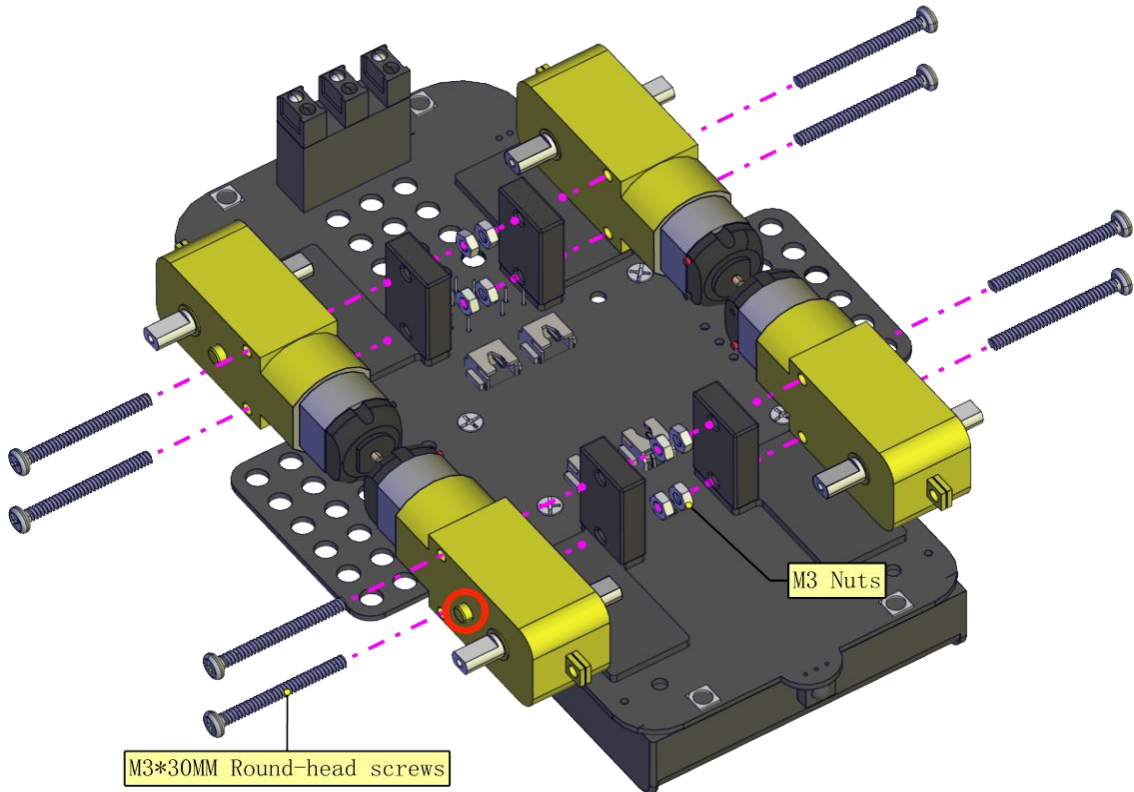


Part 7

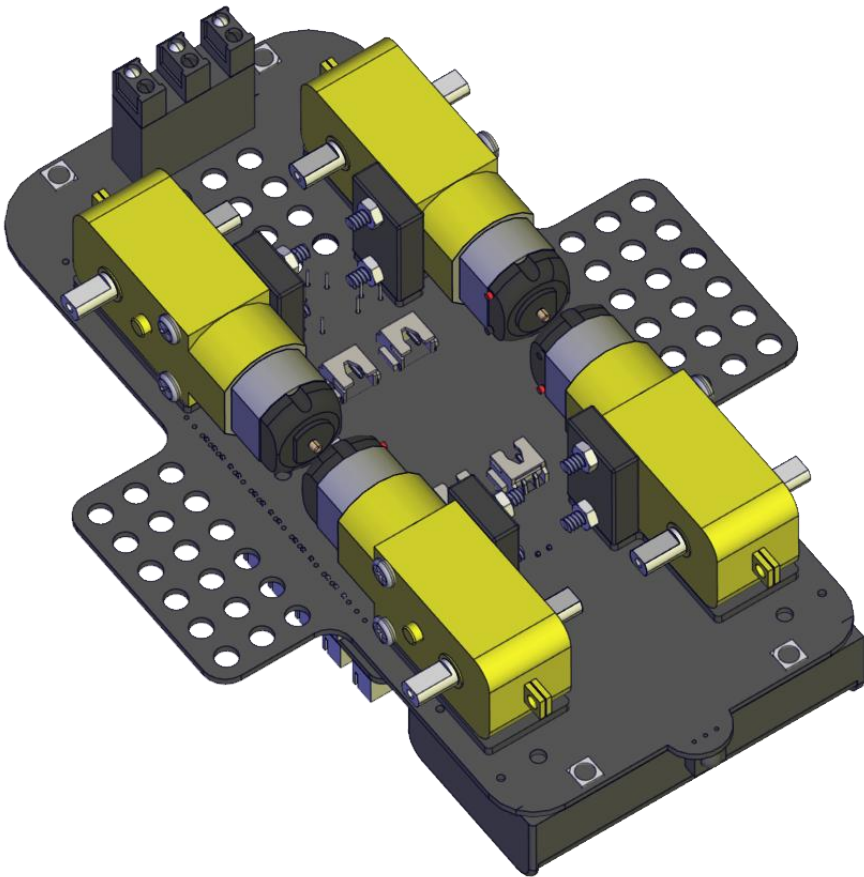
Compon
ents
Needed



Installati
on
Diagram
(mind
the
directio
n of the
motor)

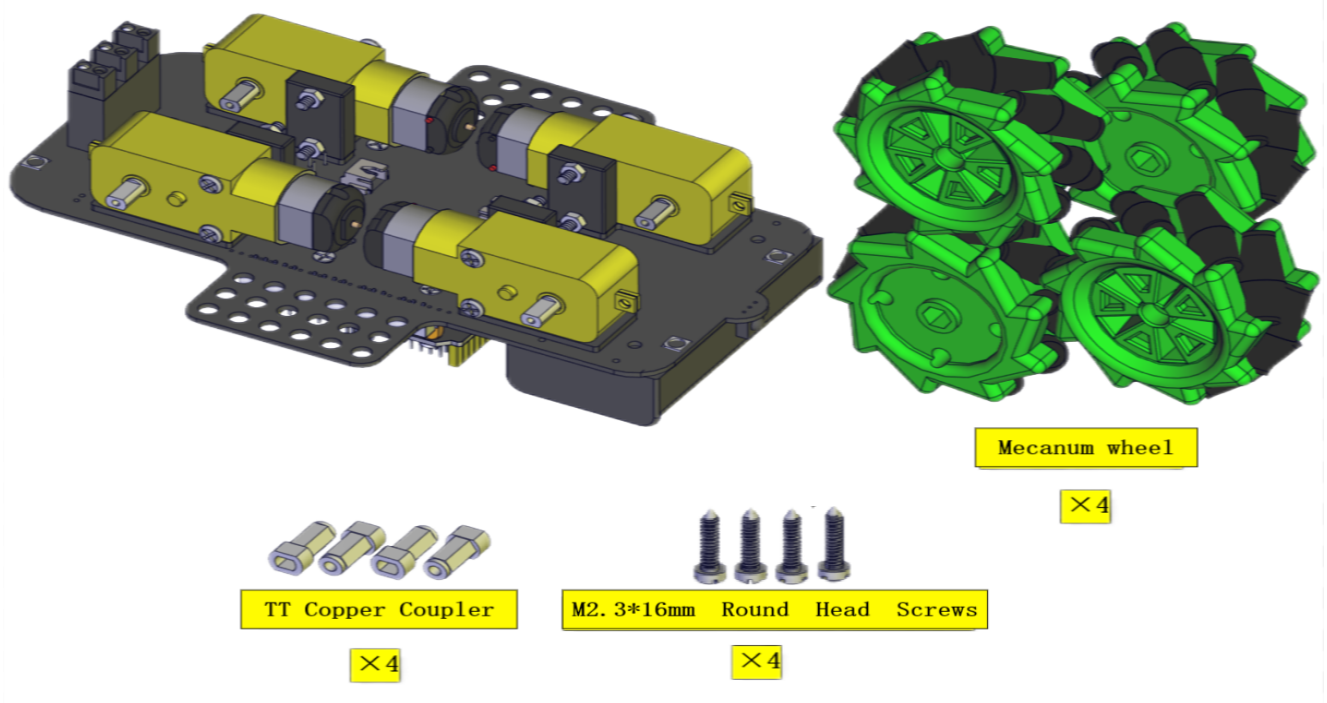


Prototy
pe

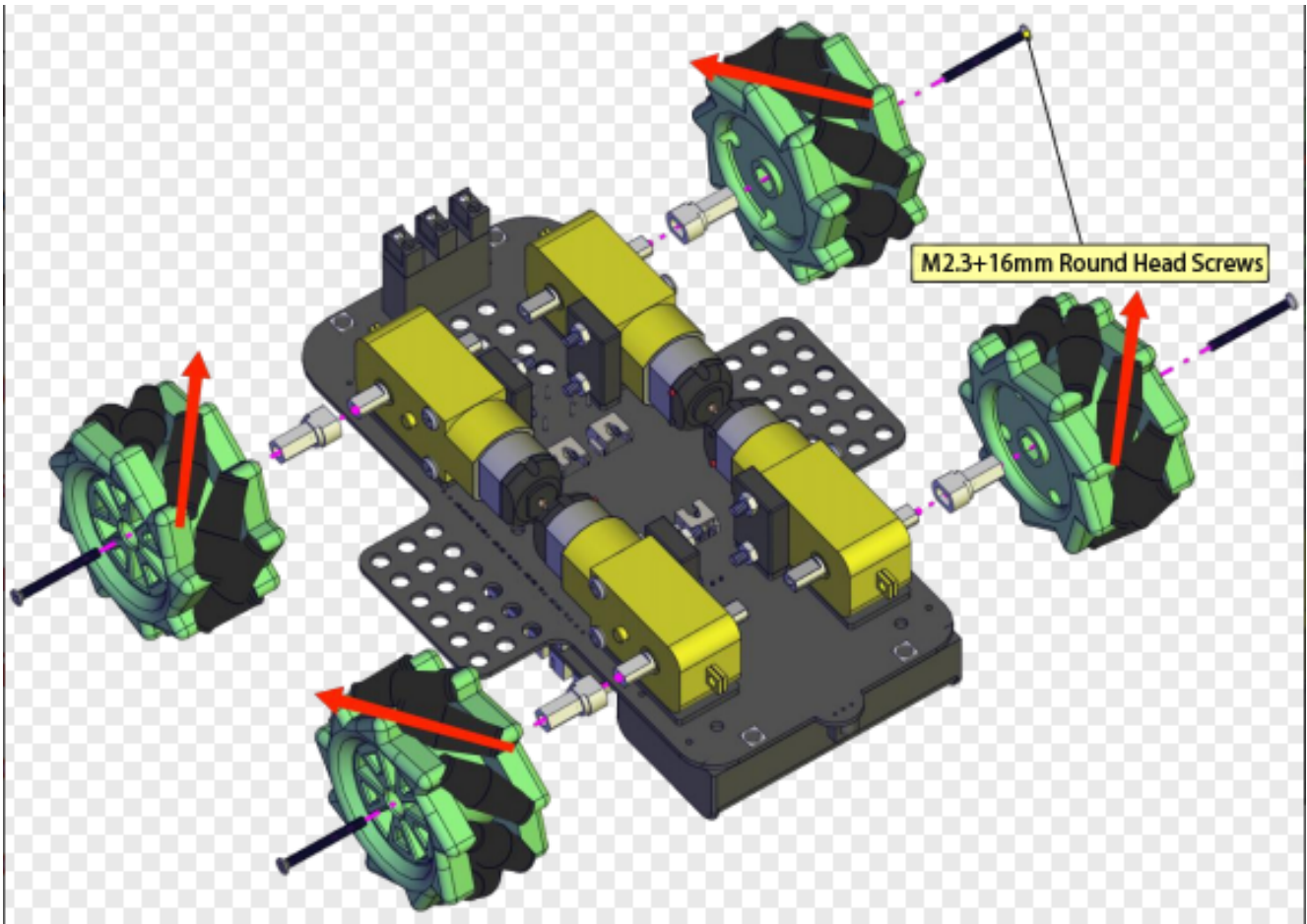


Part 8

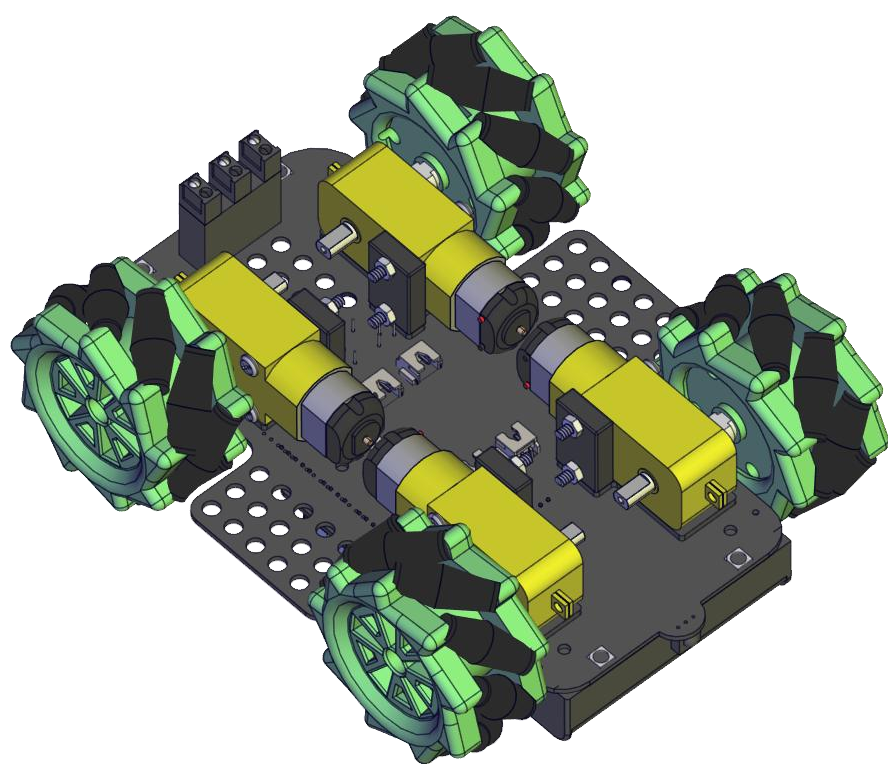
Compon
ents
Needed



Installati
on
Diagram
(Pay
attentio
n to the
installati
on
directio
n of the
mecanu
m
wheel)

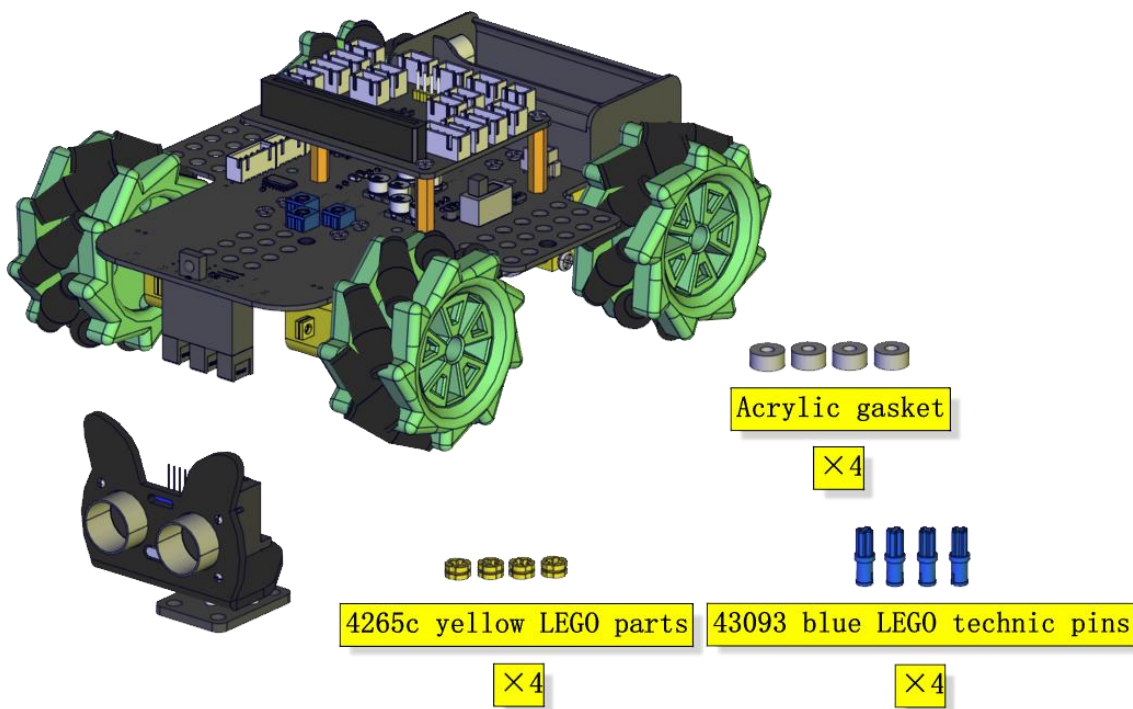


Prototy
pe

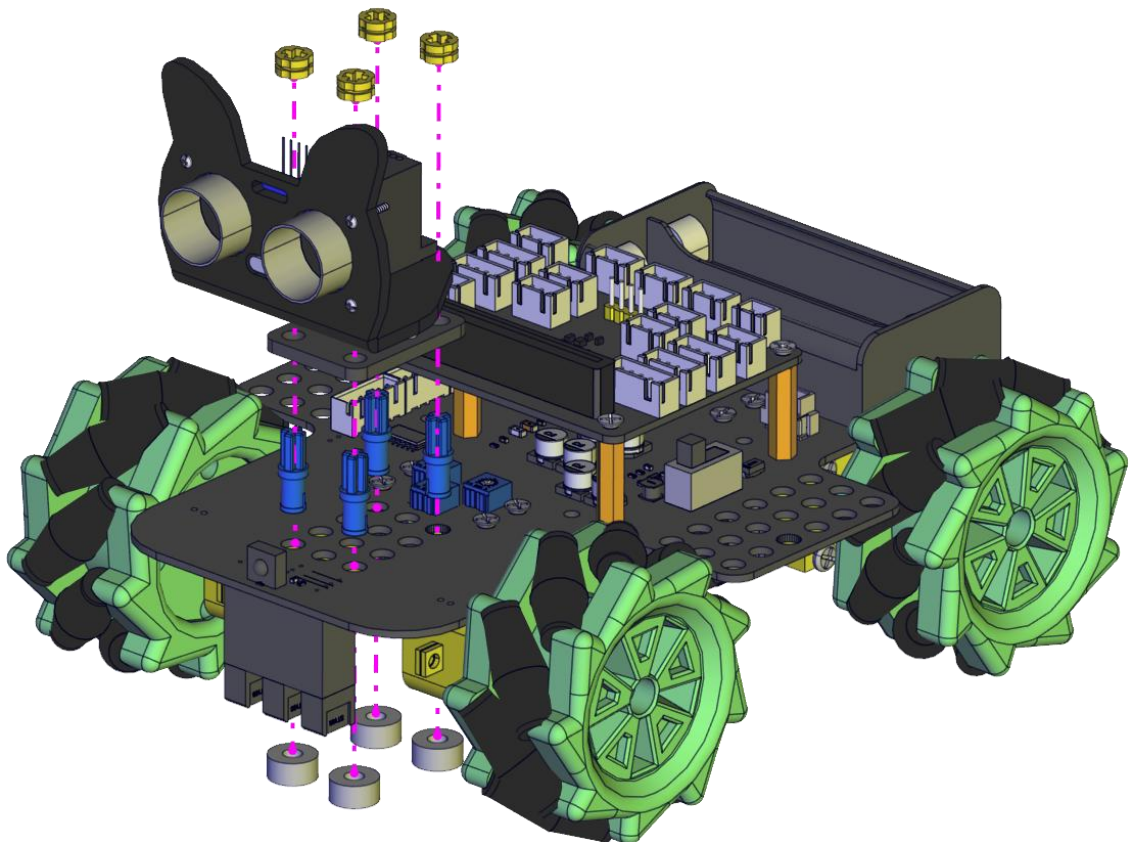


Part 9

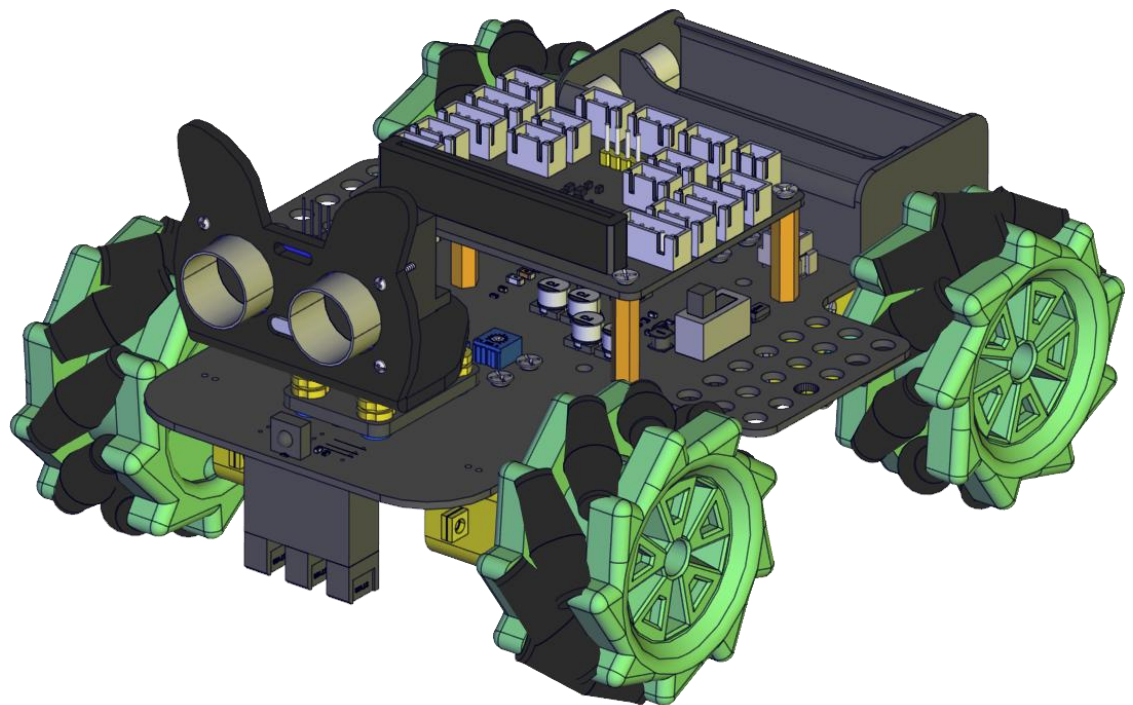
Compon
ents
Needed



Installati
on
Diagram

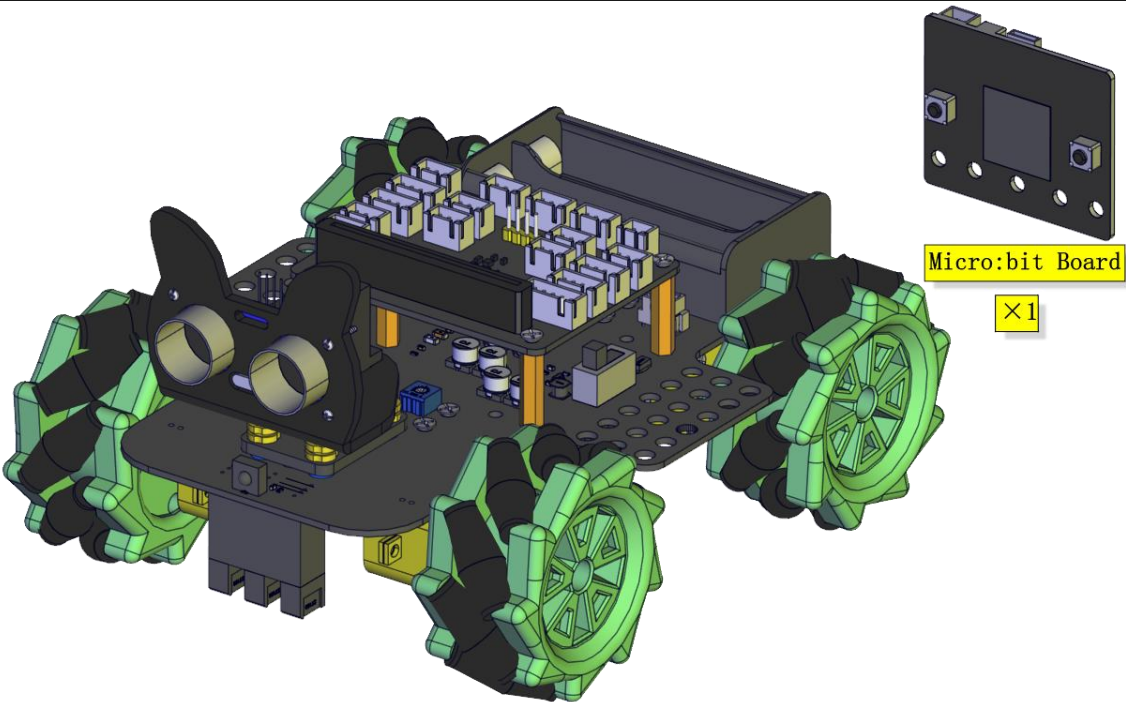


Prototy
pe

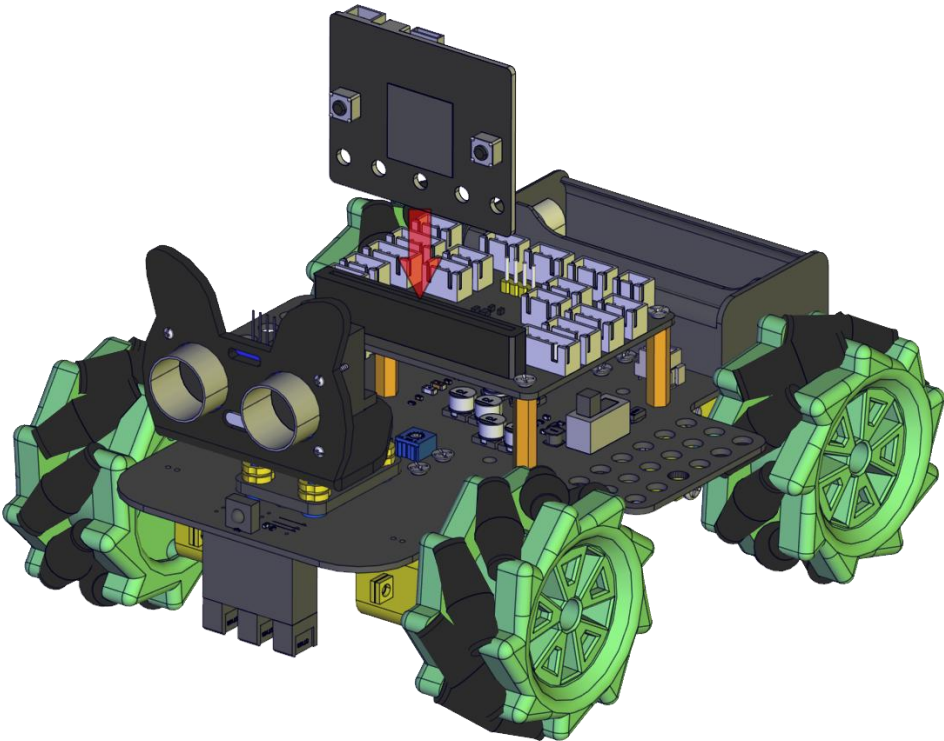


Part 10

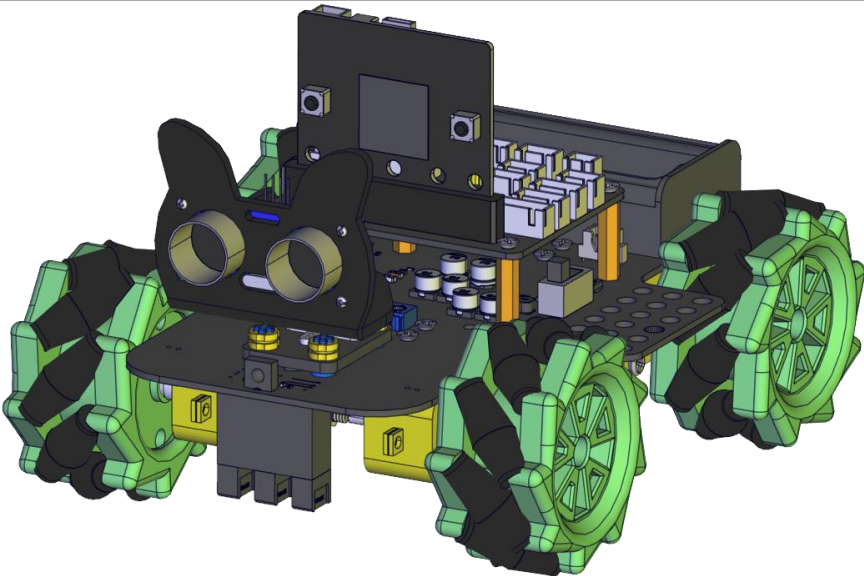
Compon
ents
Needed



Installati
on
Diagram



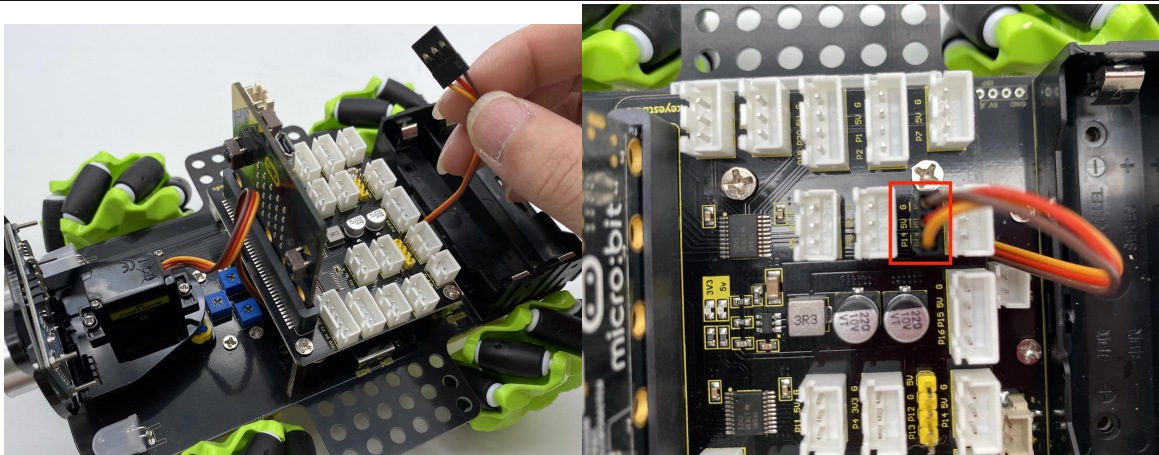
Prototy
pe



keystudio

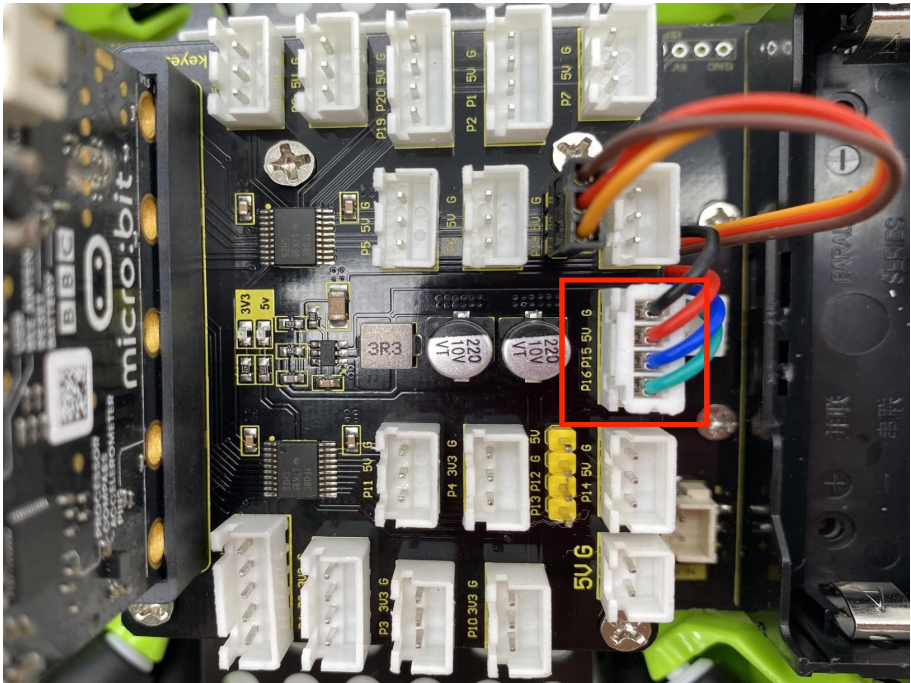
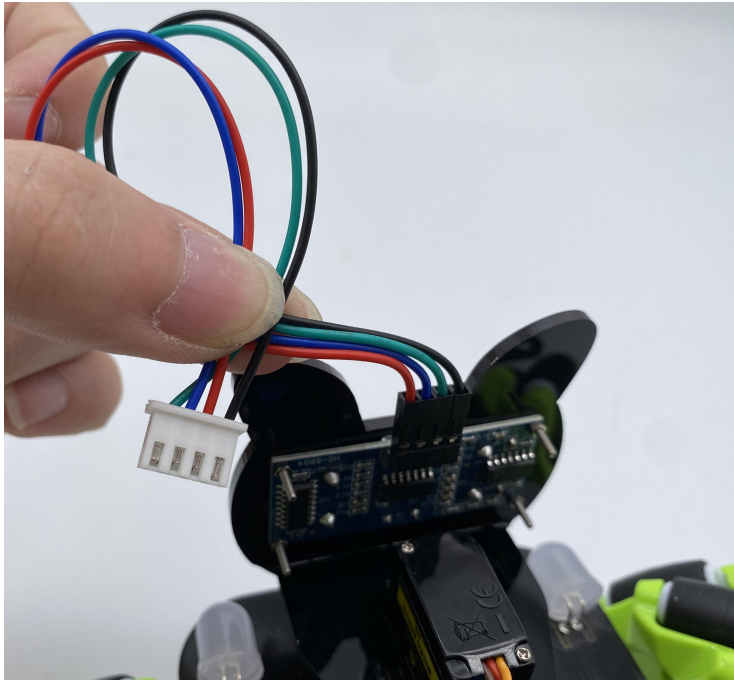
Wiring Diagram

The wiring of the servo



Servo	Expansion Board
Brown	G
Red	V
Yellow	P14

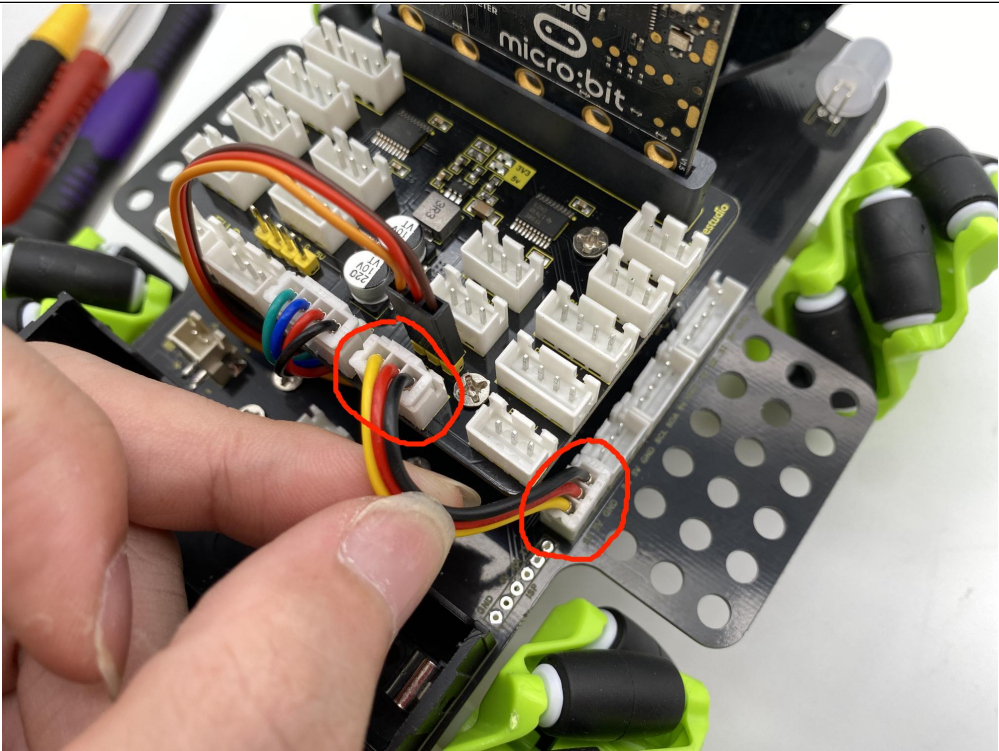
The wiring of the ultrasonic sensor



Ultrasonic Sensor	Expansion Board
VCC	5V

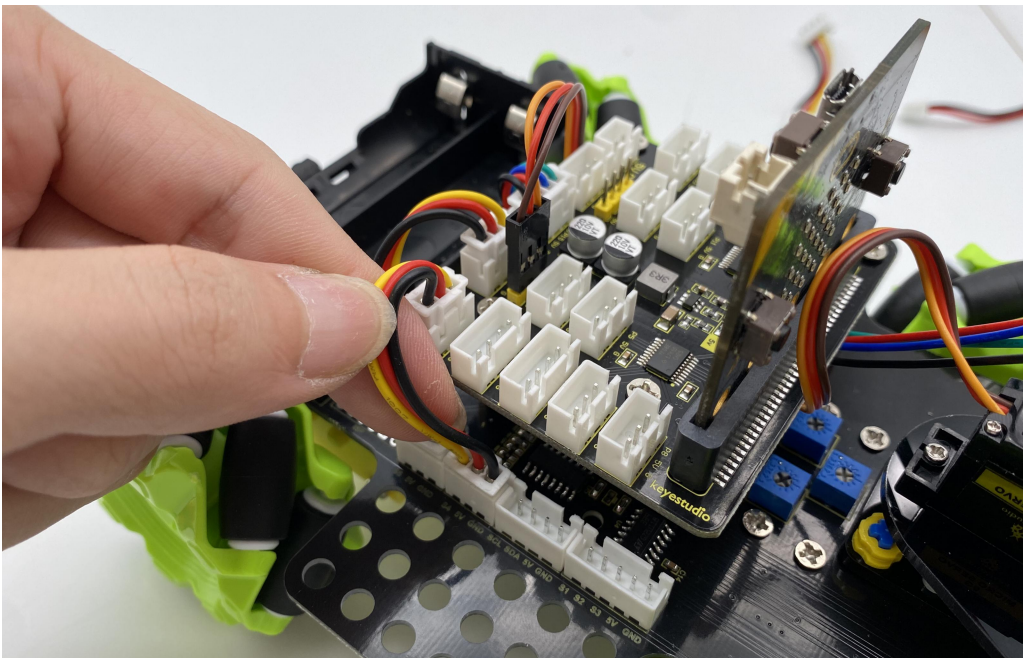
TRIG	P15
ECHO	P16
GND	G

The wiring of the IR receiver module



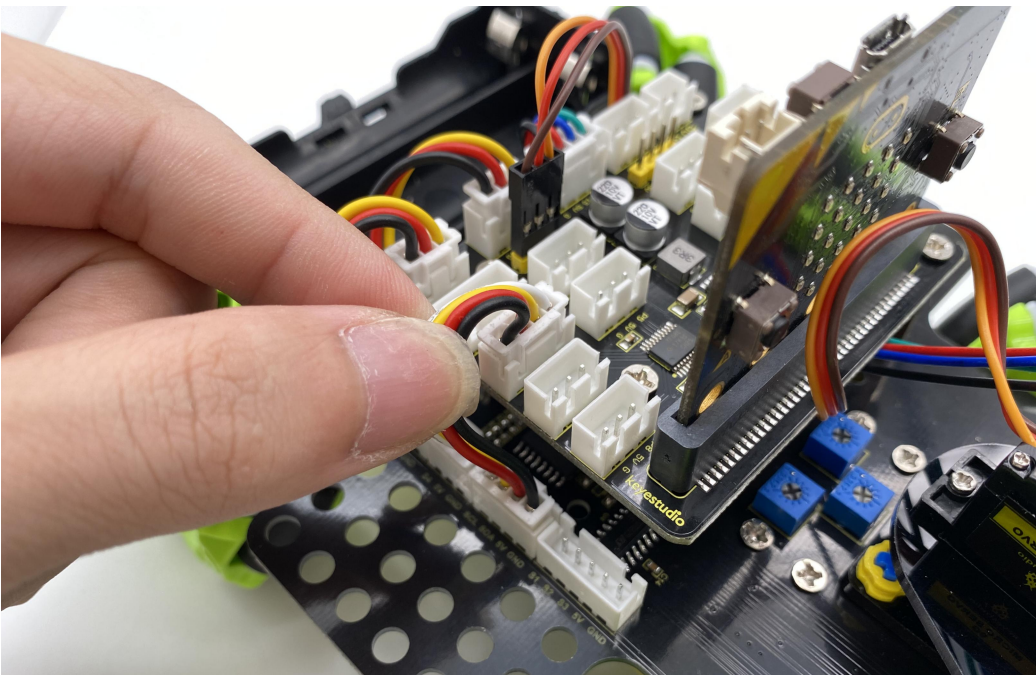
Driver Board	Expansion Board
GND	G
5V	5V
S5	P0

The wiring of the RGB



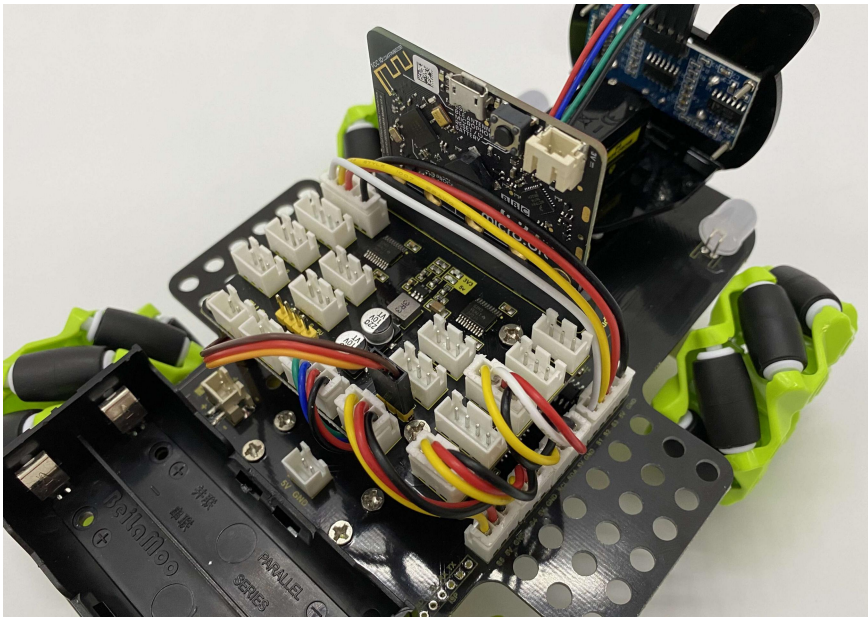
Driver Board	Expansion Board
GND	G
5V	5V
S4	P7

The wiring of controlling the motor and seven-color light



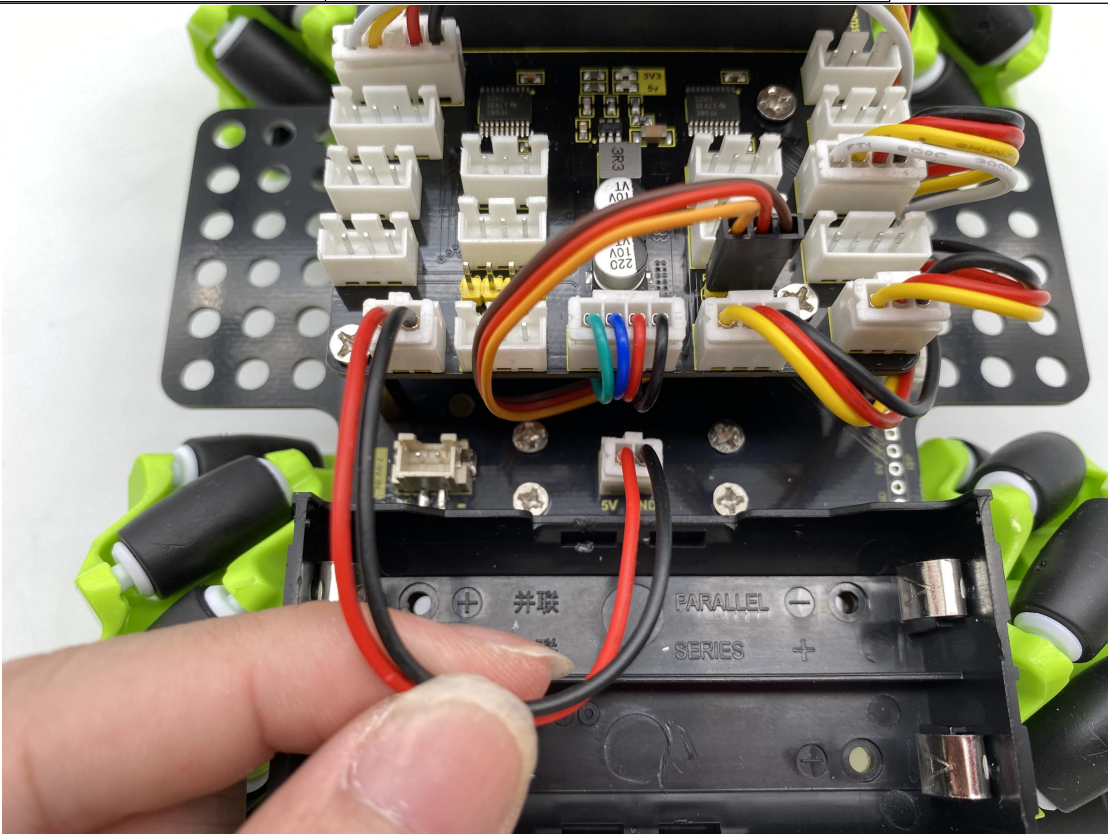
Driver Board	Expansion Board
SCL	P19
SDA	P20
5V	5V
GND	G

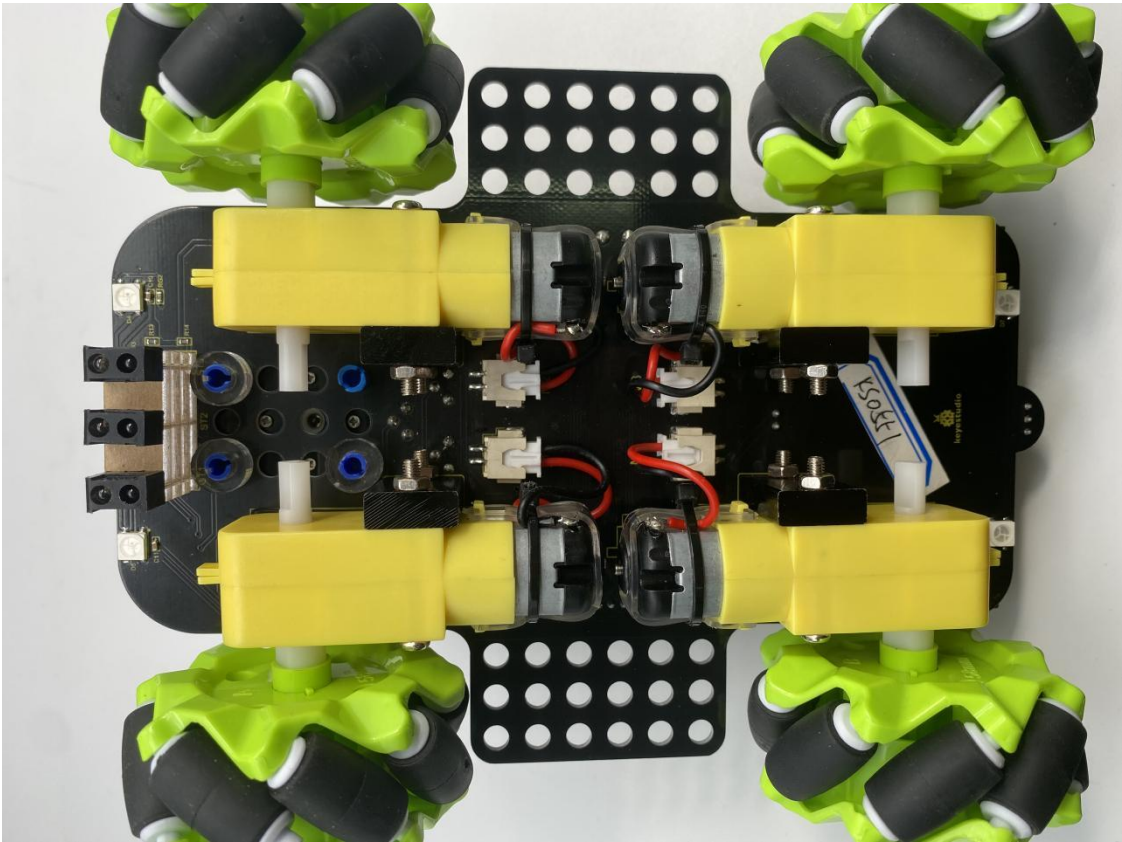
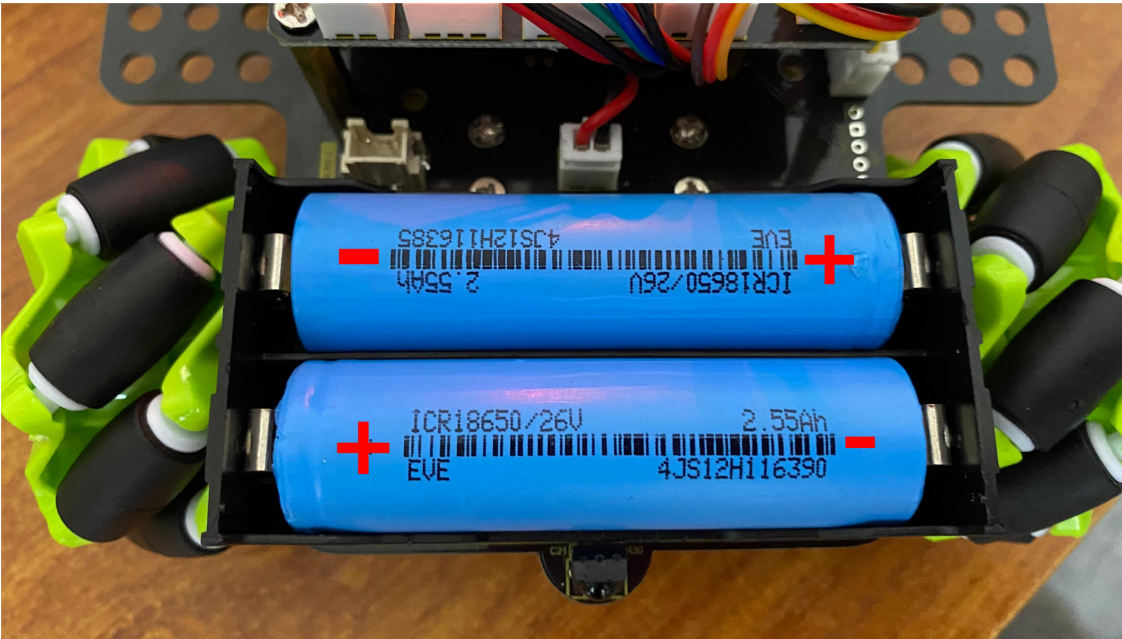
The wiring of controlling the 3-channel line-tracking sensor



Driver Board	Expansion Board
S1	P10
S2	P4
S3	P3
GND	G

The wiring of the power supply



The corresponding interface of the motor	
The installation of the battery	

7.Get Started with Micro:bit

The following instructions are applied for Windows system but can also serve as a reference if you are using a different system.

7.1 Write code and program

This chapter describes how to write program and load it to the Micro: Bit main board V2.

Please browse the link for more details:

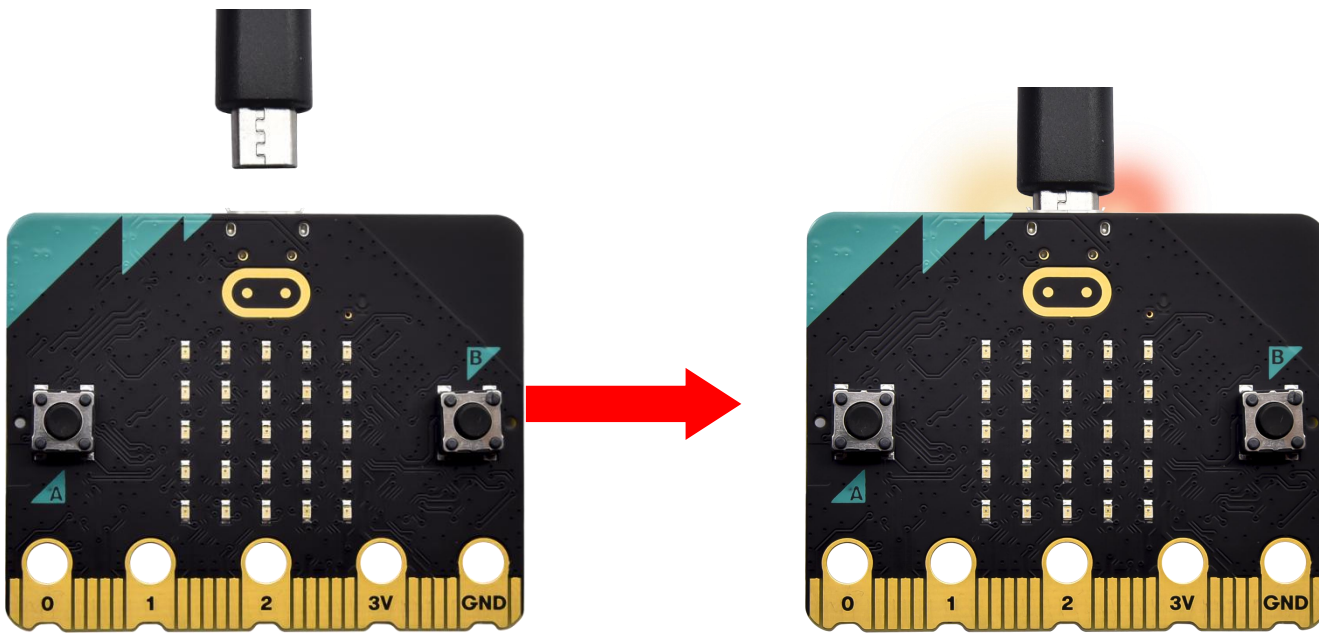
<https://microbit.org/guide/quick/>

Step 1: Connect the Micro: Bit

Firstly, link the Micro: Bit main board with your computer via the USB cable. Macs, PCs, Chromebooks and Linux (including Raspberry Pi) systems are all compatible with the Micro: Bit main board.

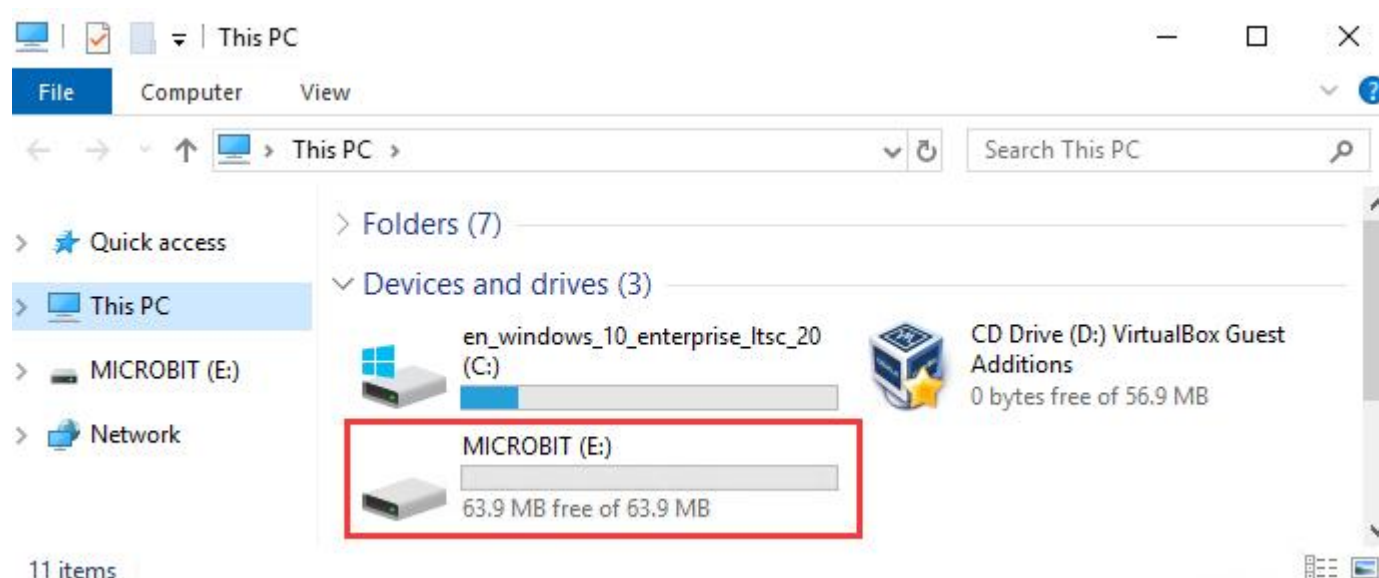
Note that if you are about to pair the board with your phone or tablet, please refer to this link:

<https://microbit.org/get-started/user-guide/mobile/>



Secondly, if the red LED on the back of the board is on, that means the board is powered. When your computer communicates with the main board via the USB cable, the yellow LED on it will flash. For example, it will flicker when you burn a "hex" file.

Then Micro: bit main board will display a driver named "MICROBIT(E:)" on your computer. Please note that it is not an ordinary USB disk, as shown below.



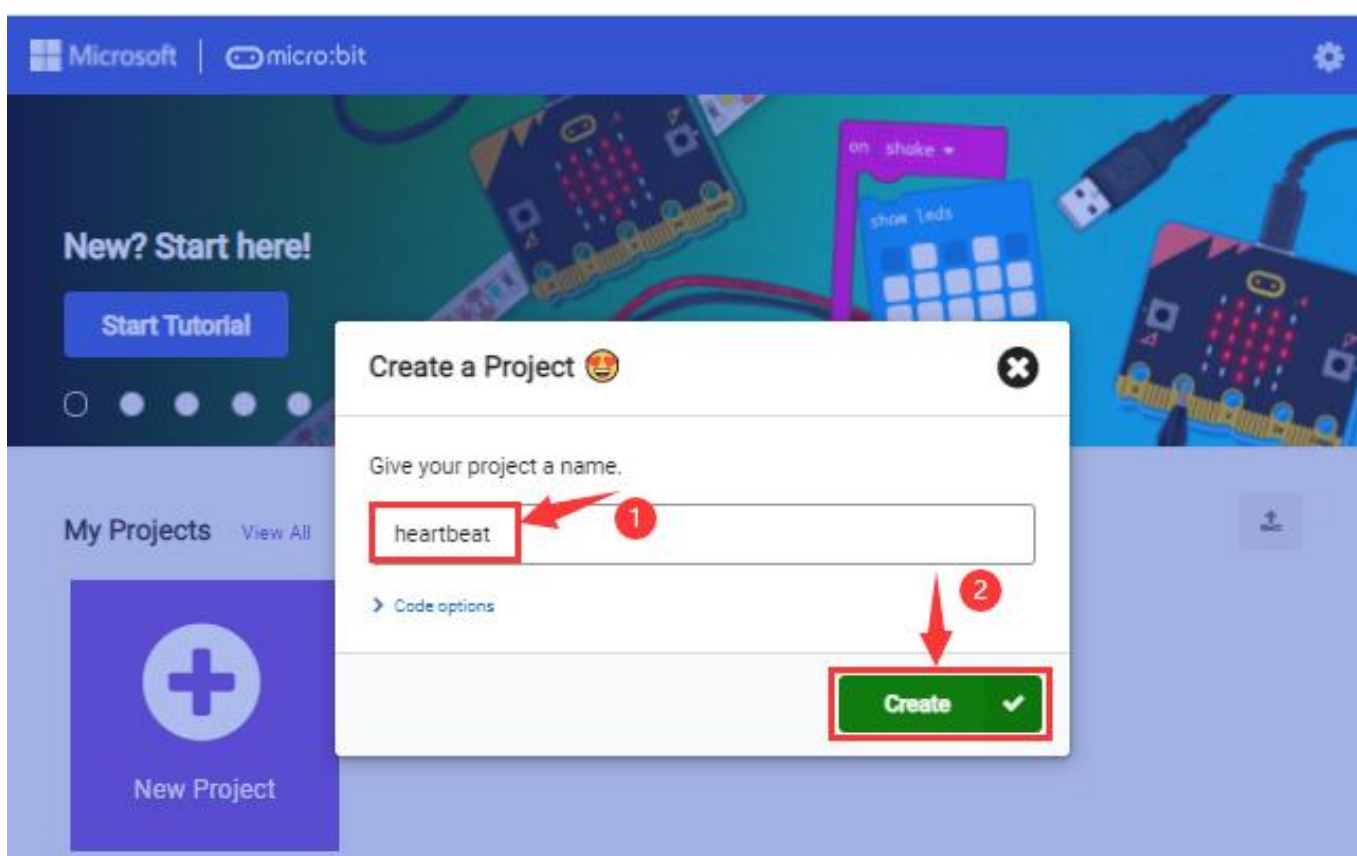
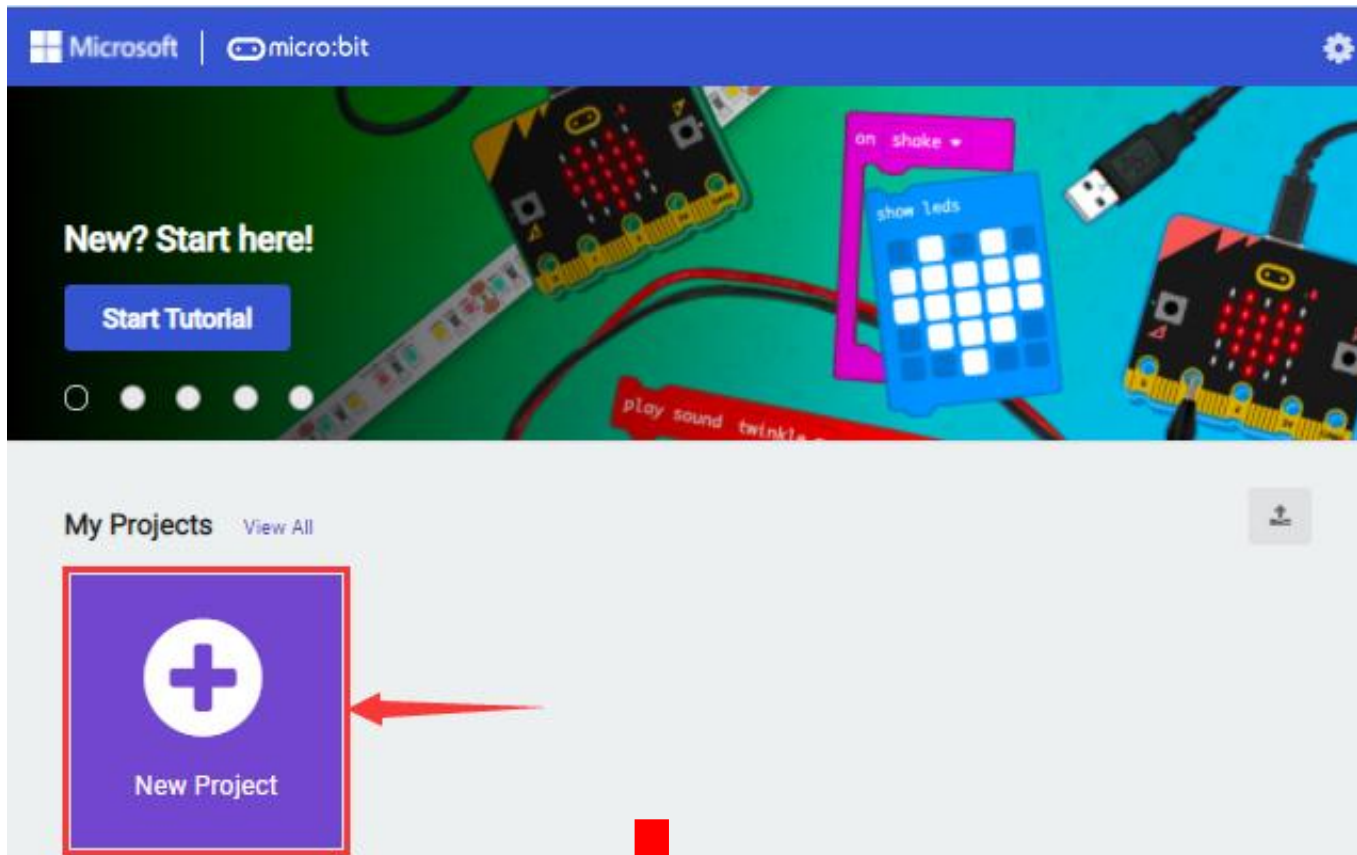
Step 2: Write programs

Access the link: <https://makecode.microbit.org/>, then click "New Project", when the dialog box 'Create a Project' appears, input 'heartbeat' in it and click 'Create ✓' to edit.

(If you are running Windows 10 system, it is also viable to edit on the Windows 10 APP, which is exactly like editing in the website.

Link of the APP :

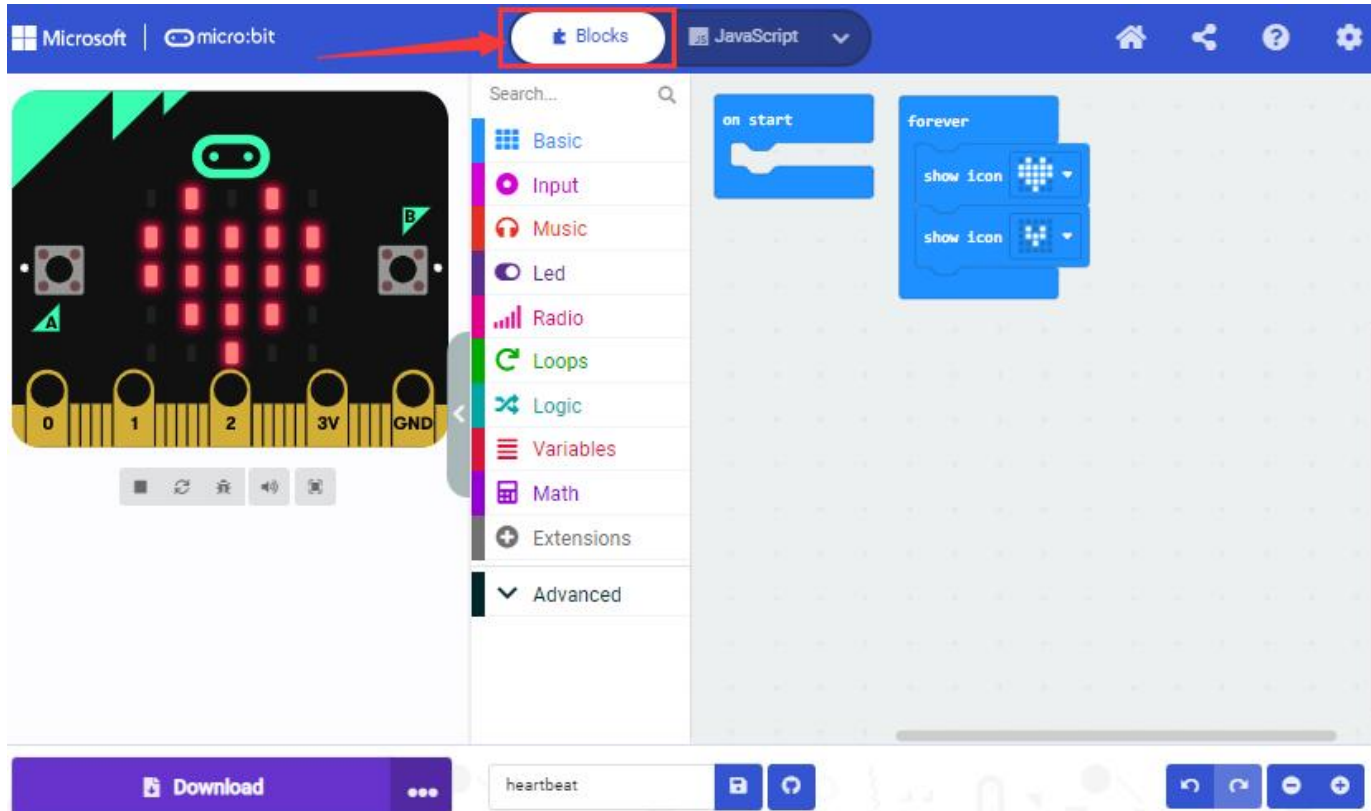
<https://www.microsoft.com/zh-cn/p/makecode-for-micro-bit/9pjc7sv48lcx?ocid=badge&rtc=1&activetab=pivot:overviewtab>



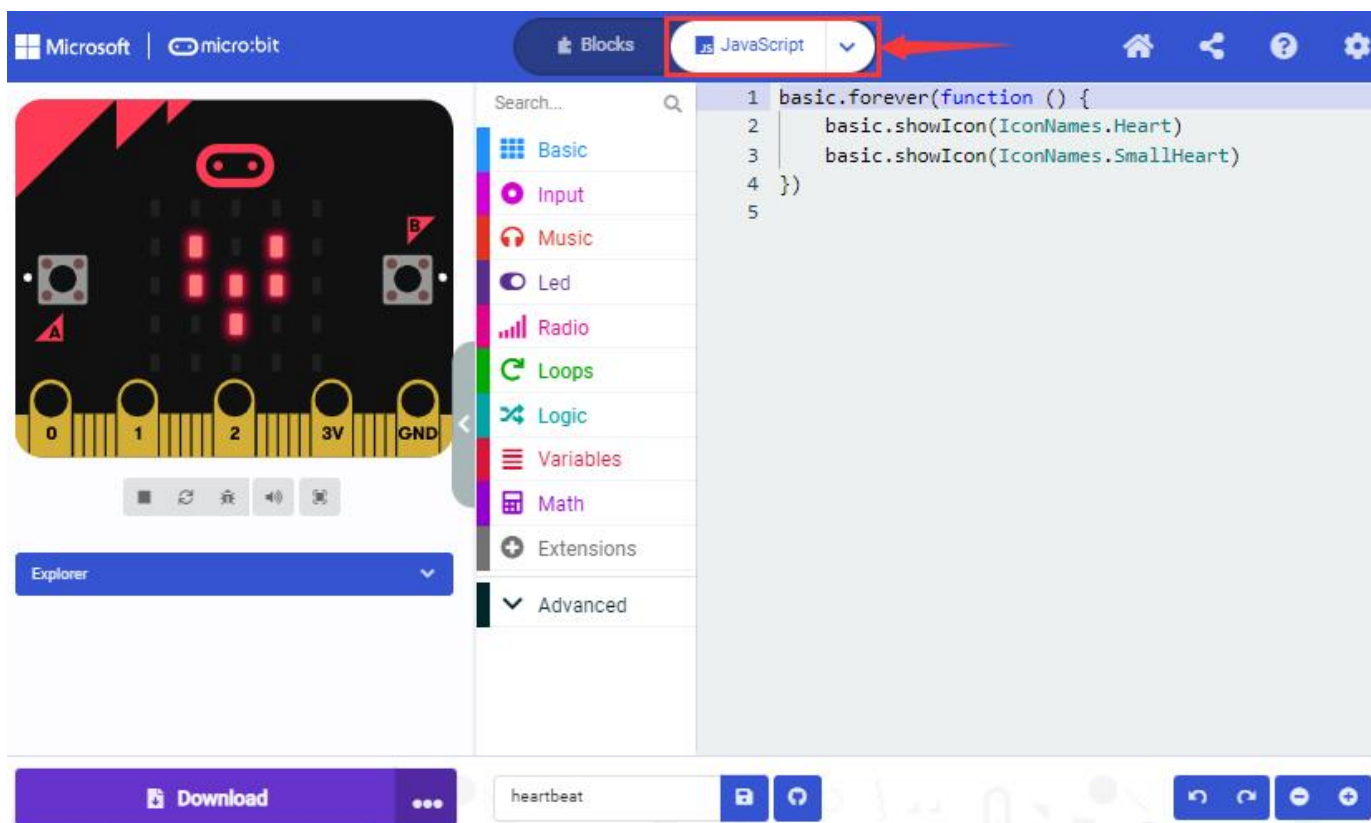
Write a set of micro:bit code. You can drag some blocks from the modules to the editing area and then run your program in Simulator of MakeCode editor, as shown in the picture below, which demonstrates how to edit 'heartbeat' program .

Here's a demo video(path): .../2. Makecode Tutorial\Makecode Code\Project 1: Heart beat\heartbeat.mp4

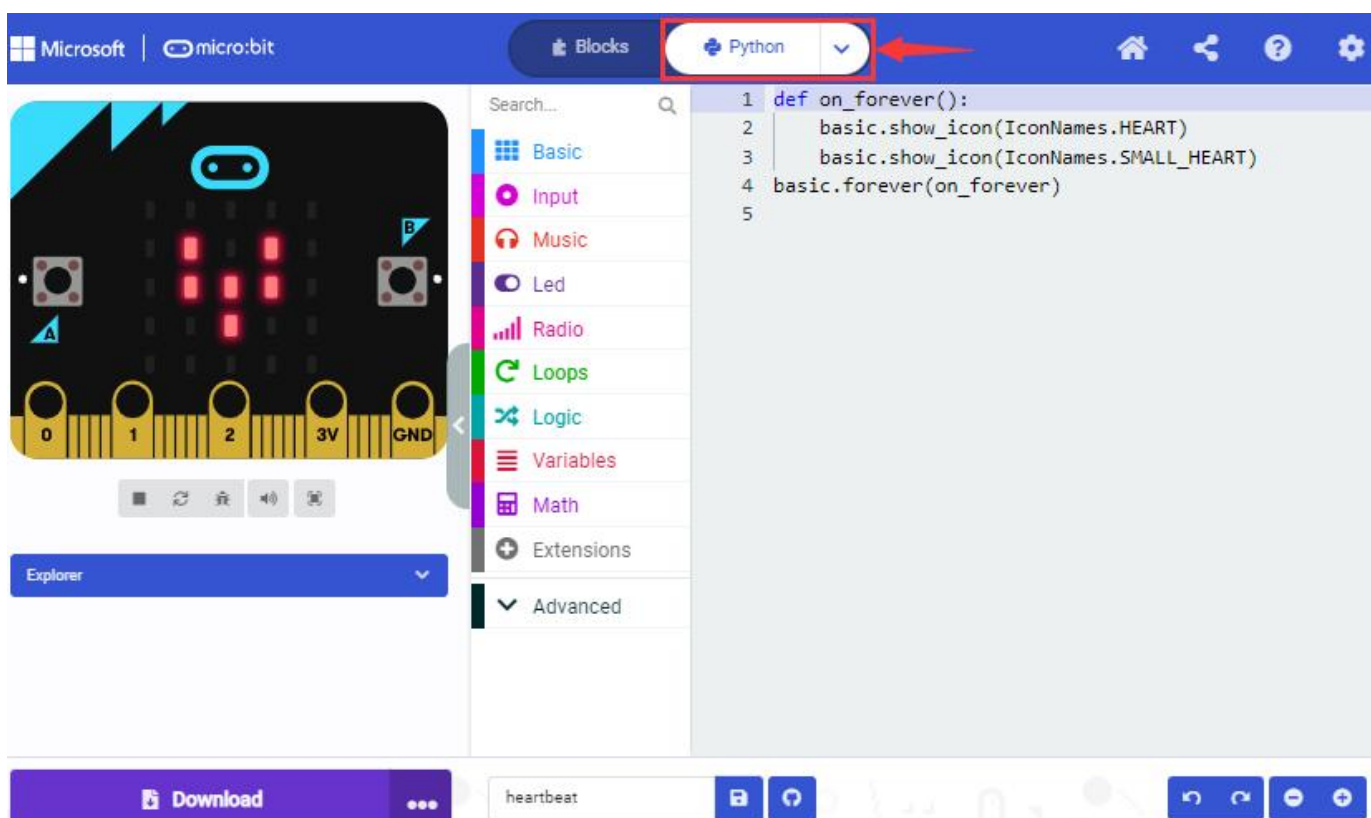
keystudio



Click " JS JavaScript" , then we enable to find the corresponding program in JavaScript language, as shown below:



You also can click the "JS JavaScript" and slide the button to choose "Python" and you will find the corresponding Python language program, as shown below:

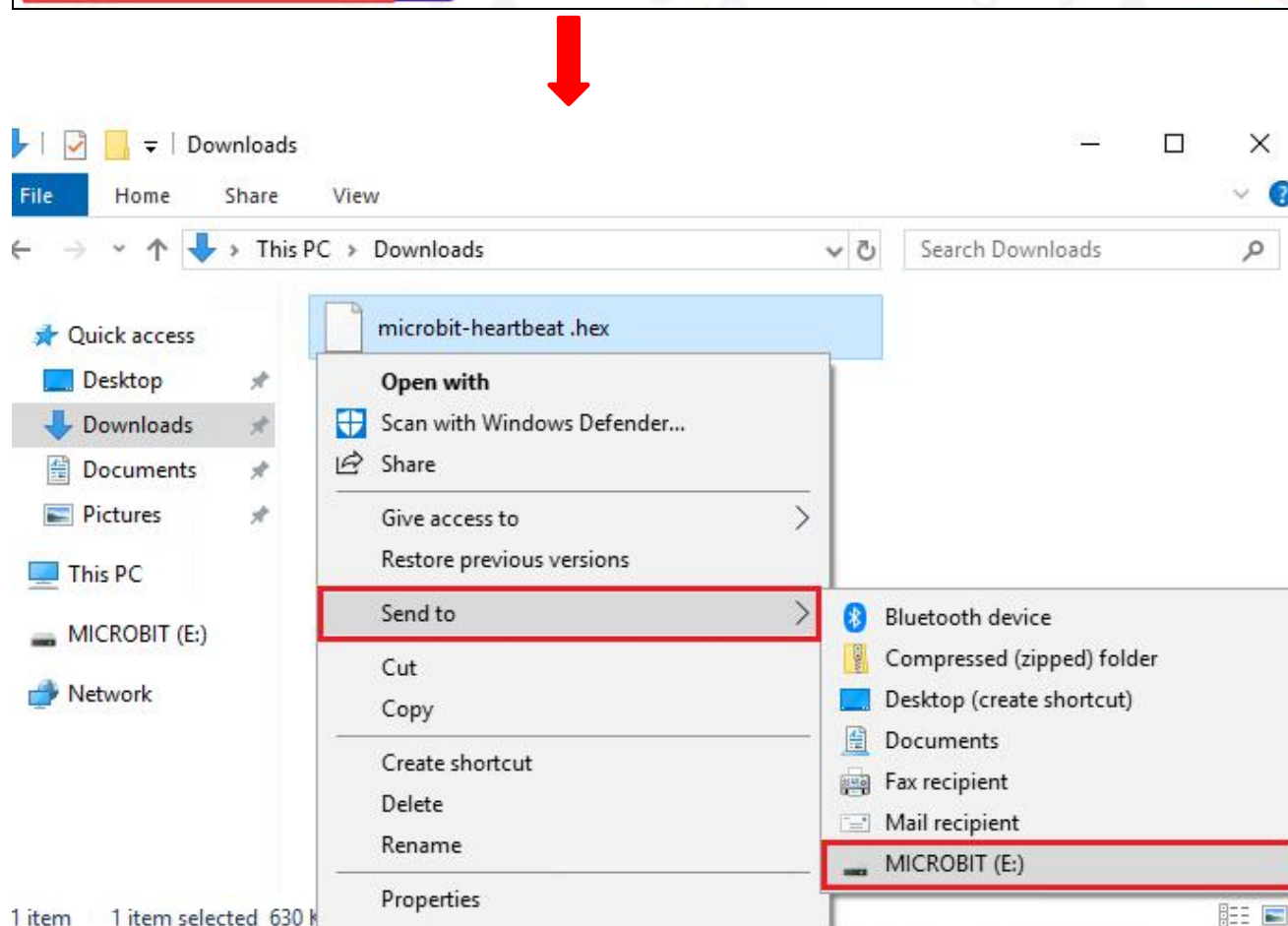
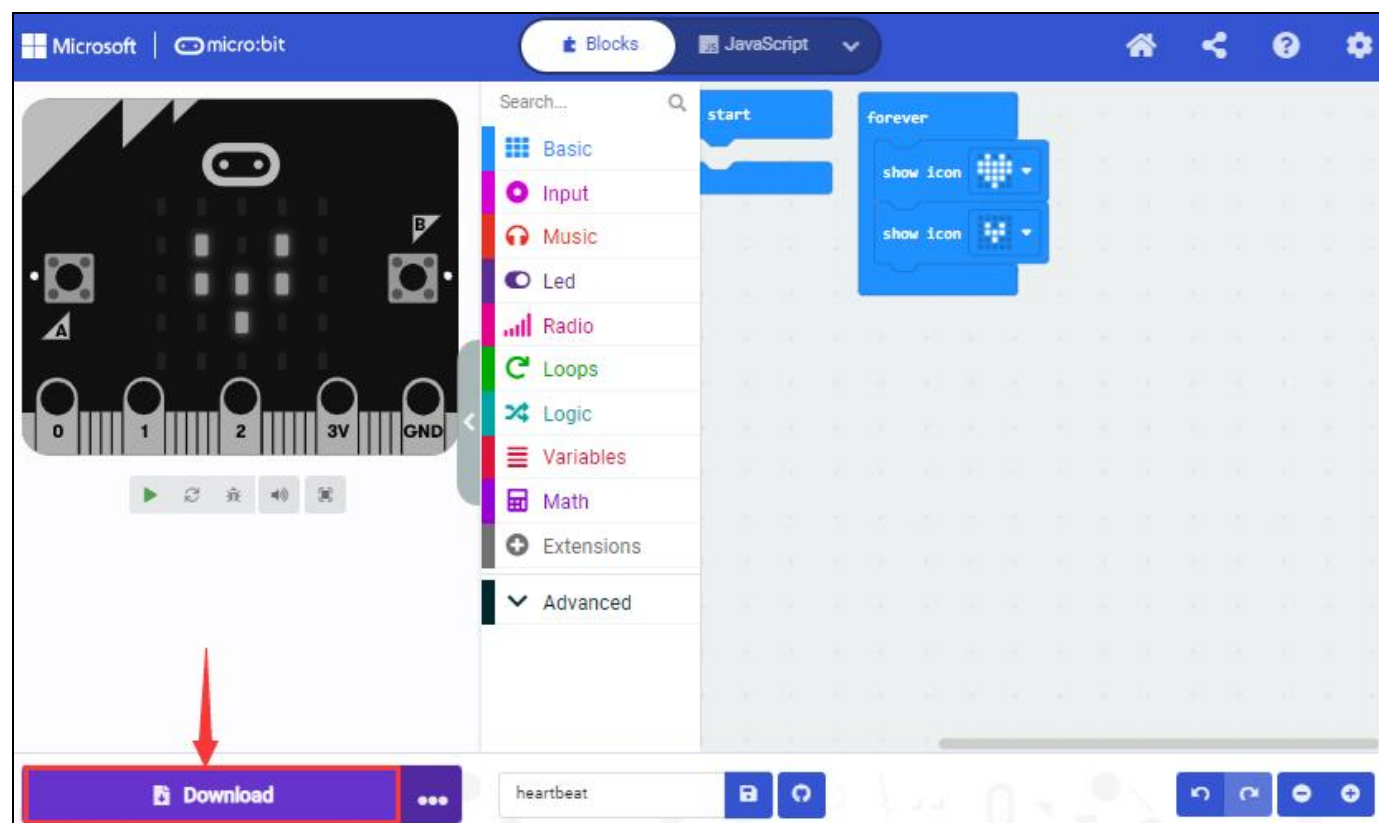


Step 3: Download Code

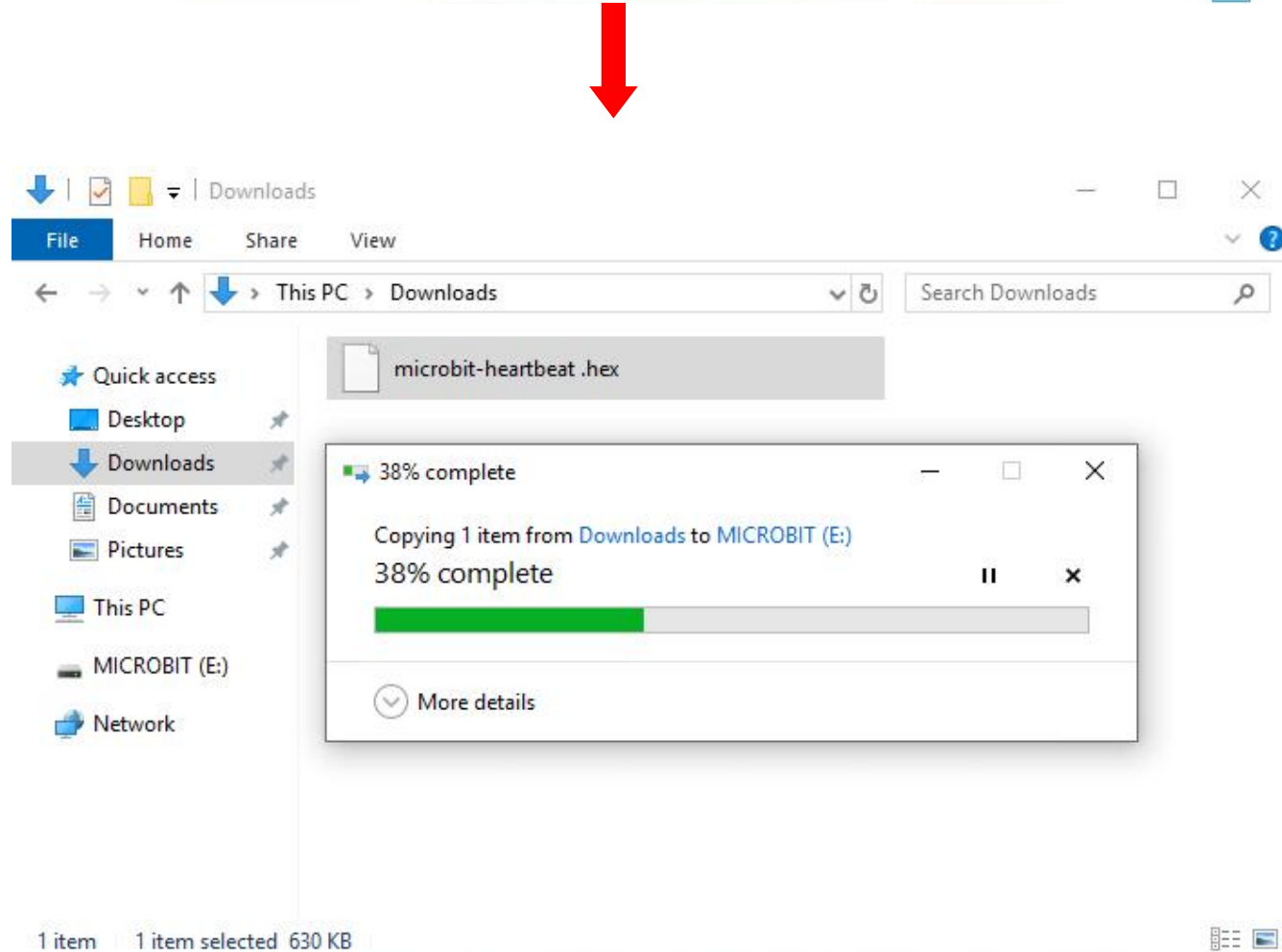
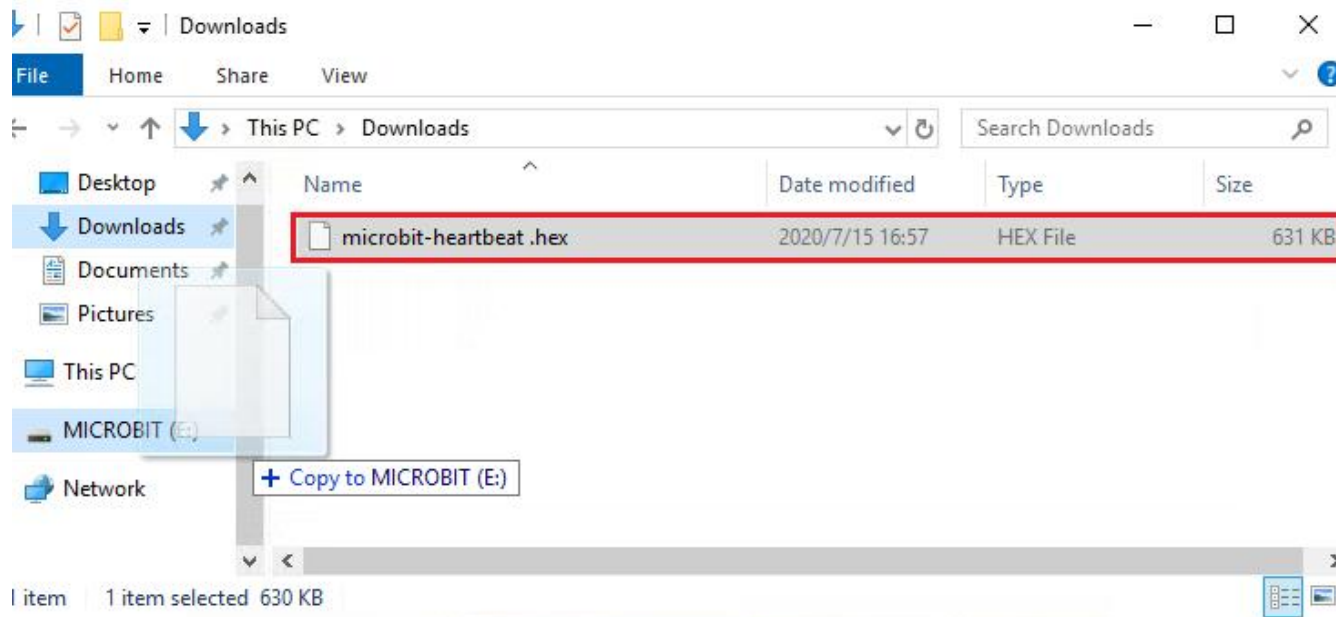
If your computer is Windows 10, you solely need to click the 'Download' button to download the program to your micro: bit main board.

If you are writing program through the website, following these steps:

Click the 'Download' button in the editor to download a "hex" file, which can be read by the Micro: Bit main board. Once the hexadecimal file is downloaded, copy it to your board just like the process that you copy the file to the USB driver. If you are running Windows system, you can also right-click and select 'Send to → MICROBIT(E:)' to copy the hex file to the Micro: Bit main board.



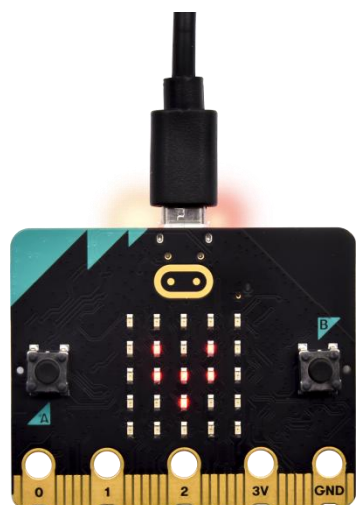
You can also directly drag the "hex" file onto the MICROBIT (E:) disk.



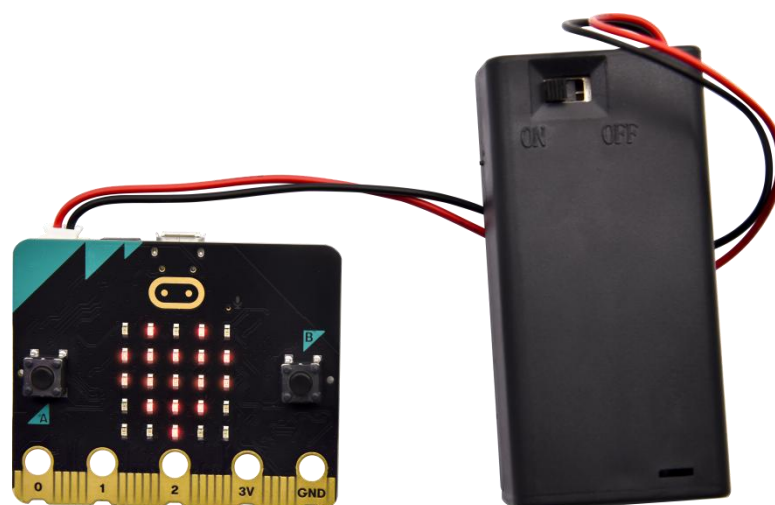
During the process of copying the downloaded hex file to the Micro: bit main board, the yellow signal light on the back side of the board flashes. When the copy is completed, the yellow signal light will stop flashing and remain on.

Step 4: Run the program

After the program is uploaded to the Micro: bit main board, you could still power it via an USB cable or an external power. The 5 x 5 LED dot matrix on the board displays the heartbeat pattern.



Power via USB cable



Power via external power (3V)

Caution:

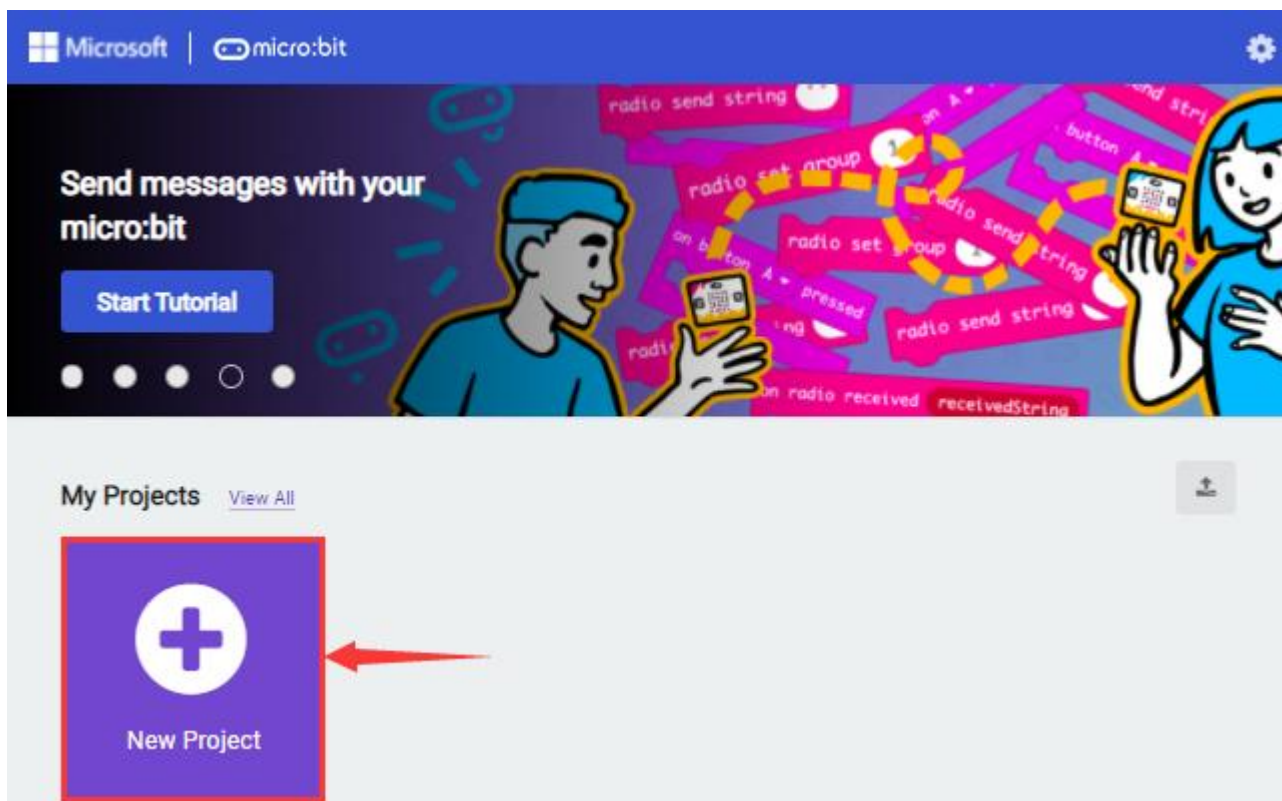
When programming, the MICROBIT driver will automatically eject and return and your hexadecimal files will disappear. And the board can only have access to hexadecimal files (hex) and won't save other files.

Step5: Other programming languages

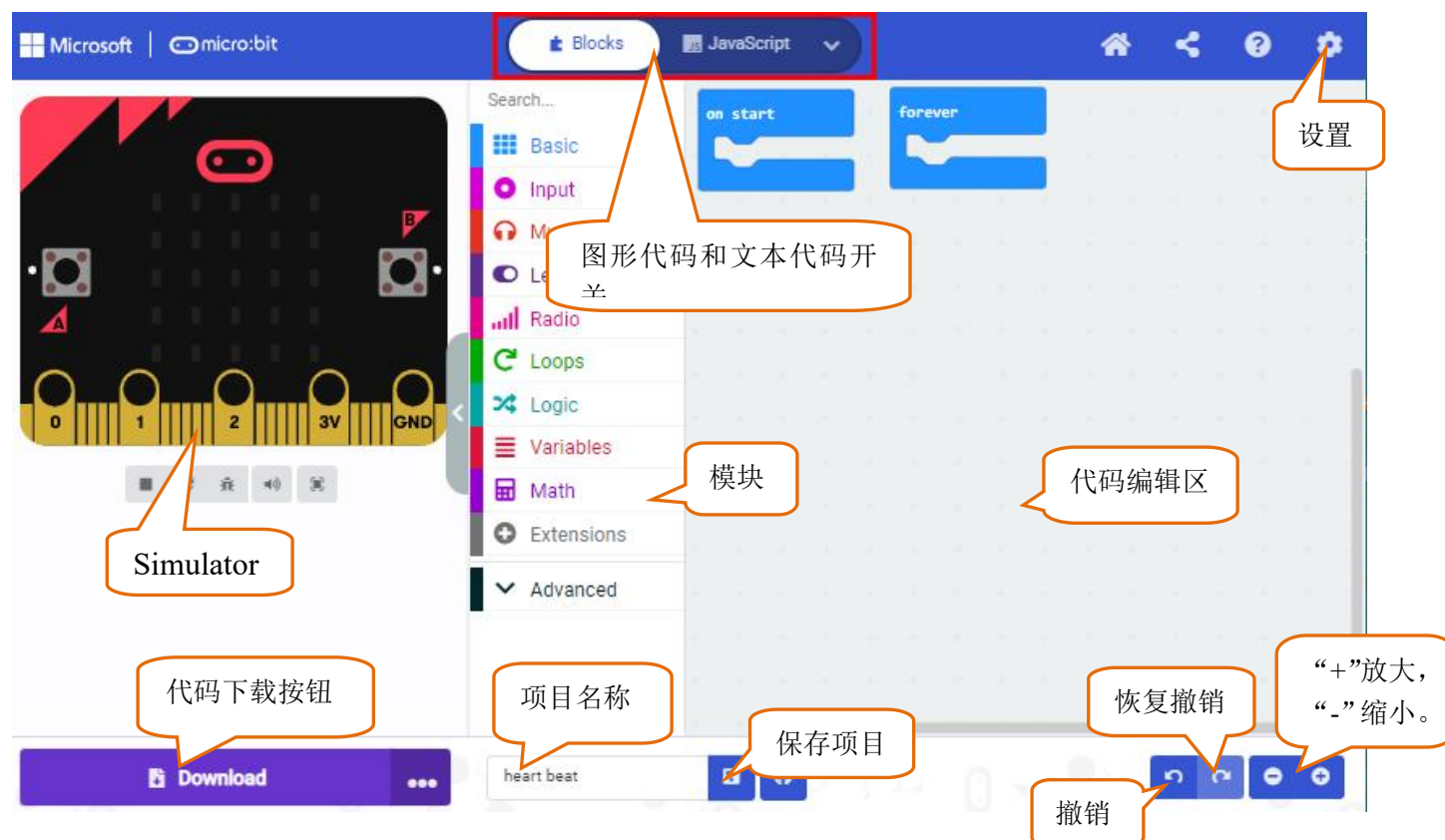
Go to the link: : <https://microbit.org/code/> for different programming languages or : <https://microbit.org/projects/> to learn what you are interested.

7.2.Makecode

Access the link: <https://makecode.microbit.org/>, and enter the online editor of [makecode](#) or open the APP makecode for micro:bit Windows 10.



Click "New Project" , and input "heartbeat" in the dialog box, then click "create ✓" to enter Makecode editor, as shown below:



There are blocks “on start” and “forever” in the code editing area.

When the power is plugged or reset, “on start” means that the code in the block only executes once, while “forever” implies that the code runs cyclically.

7.3.Quick Download

As mentioned before, if your computer is Windows 10 and you have downloaded the MakeCode APP for micro:bit, then the code can be quickly downloaded to the Micro: Bit main board by clicking ‘Download’ button.

While it is a little more trickier if you are using a browser to enter Makecode. However, if you use Google Chrome for Android, ChromeOS, Linux, macOS and Windows 10, the process can be quicker too.

We use the webUSB function of Chrome to access the micro USB hardware device.

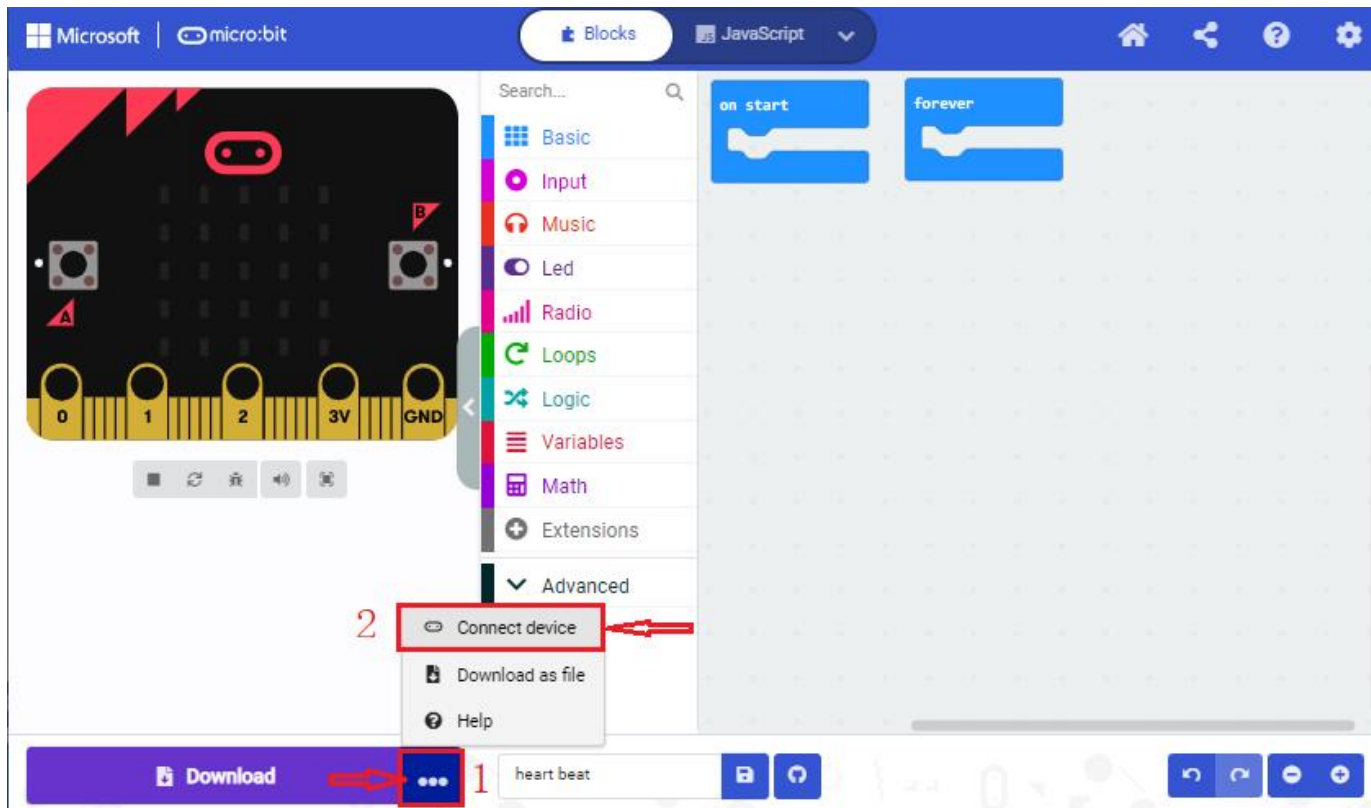
You could refer to the following steps to connect and pair devices.

Device pairing:

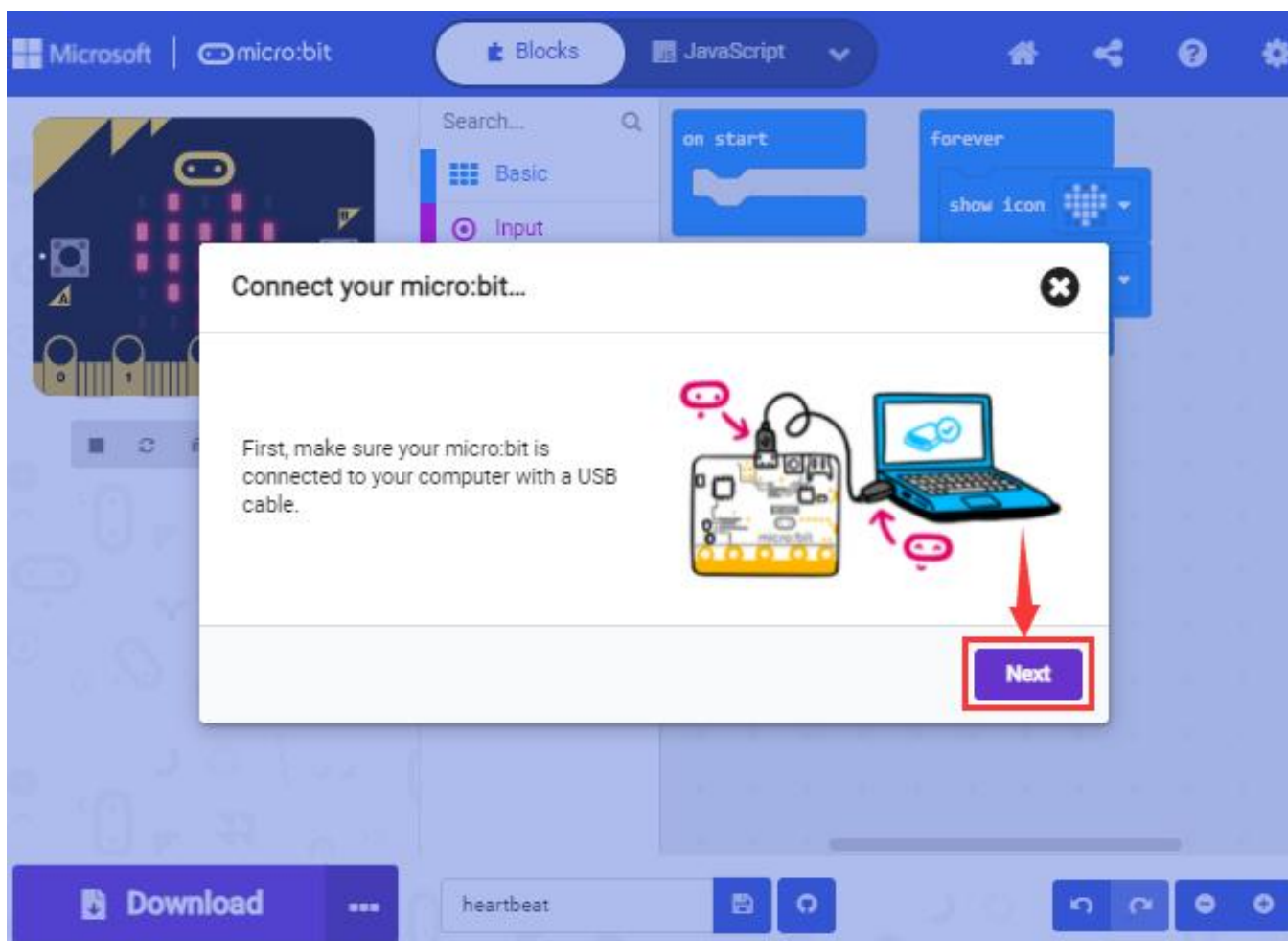
Connect micro:bit to your computer by USB cable.

Click “...” beside to “Download” and tap “Connect device” ;

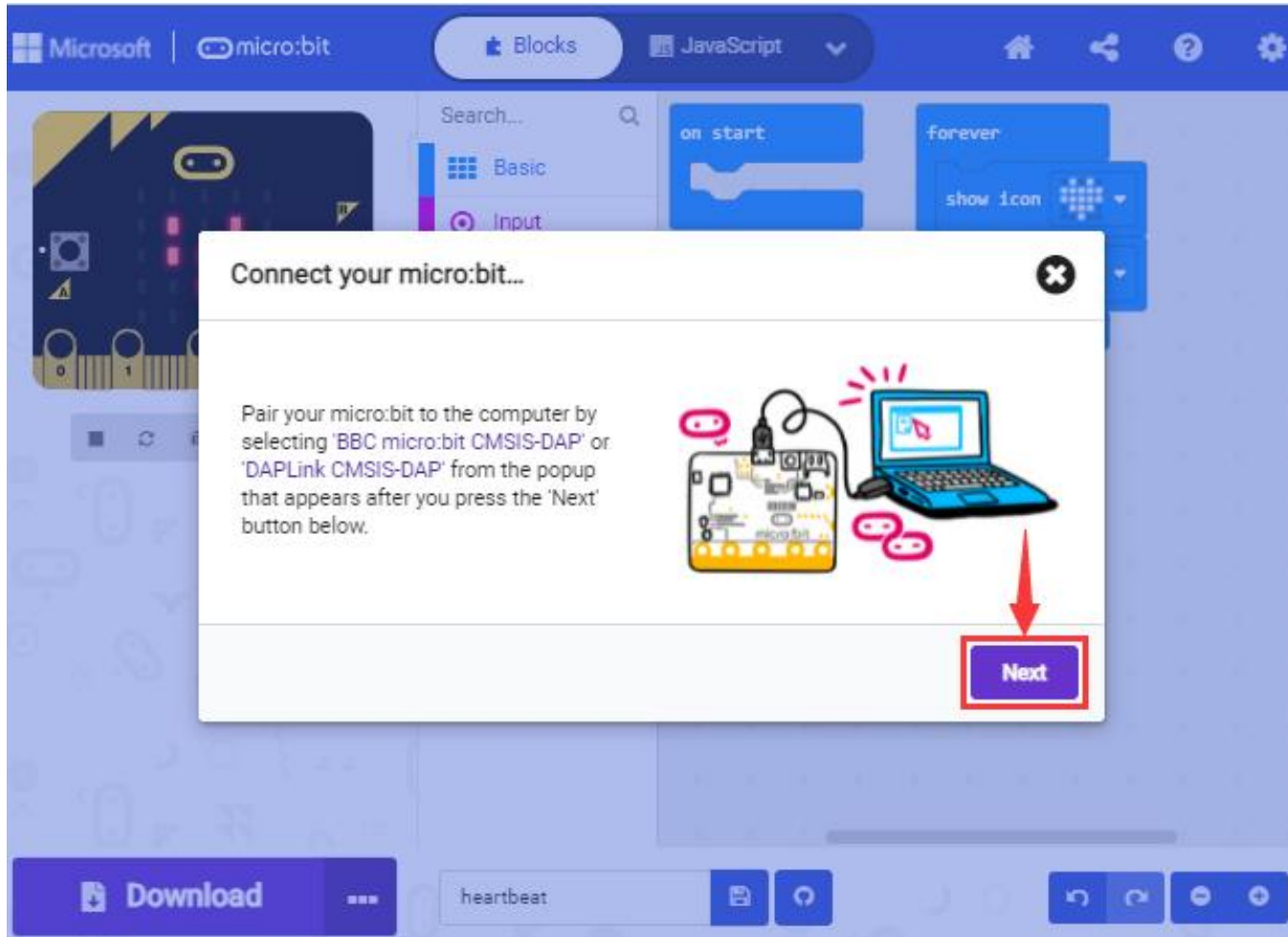
keyestudio



Click "Next" ;



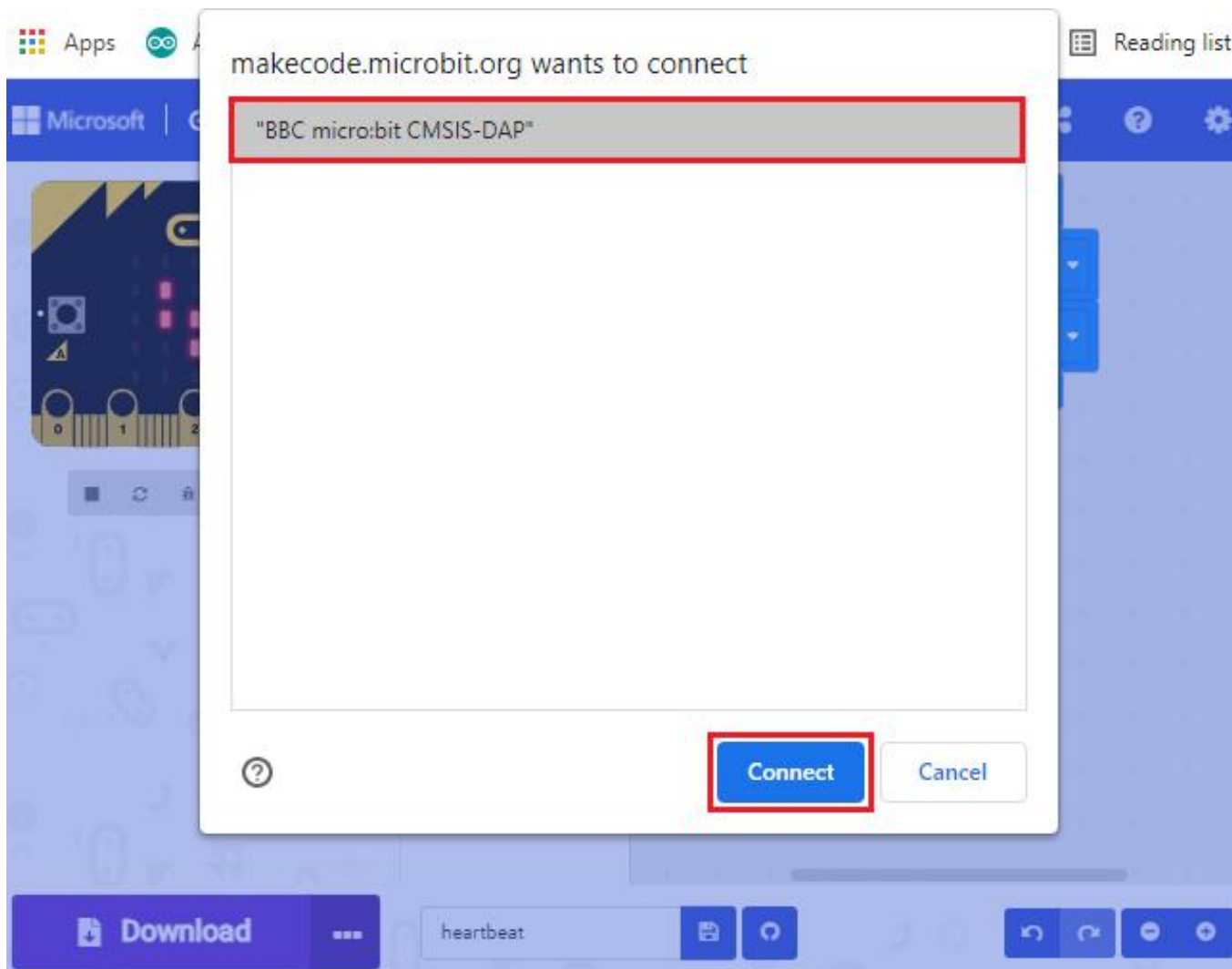
Click "Next" again



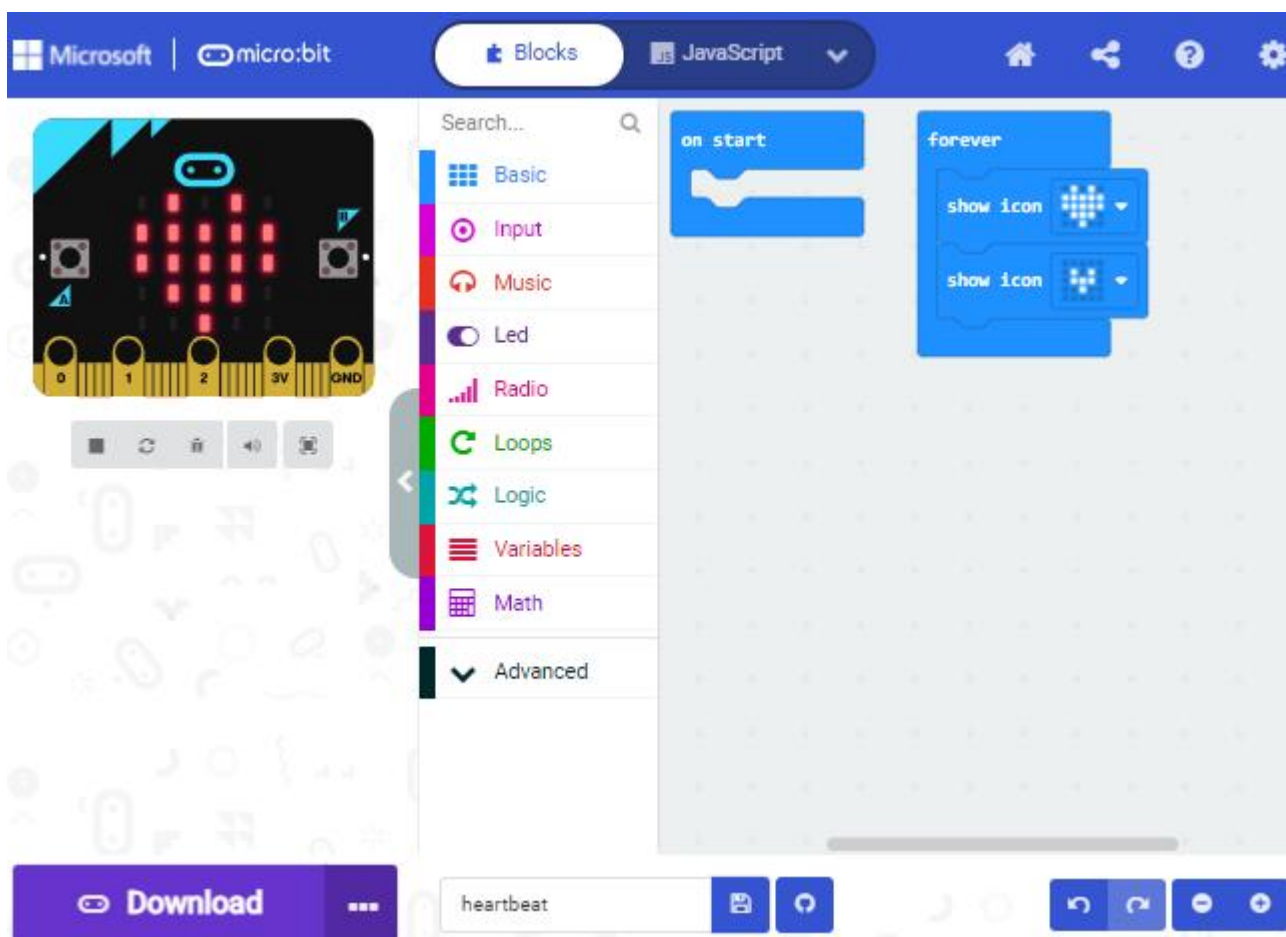
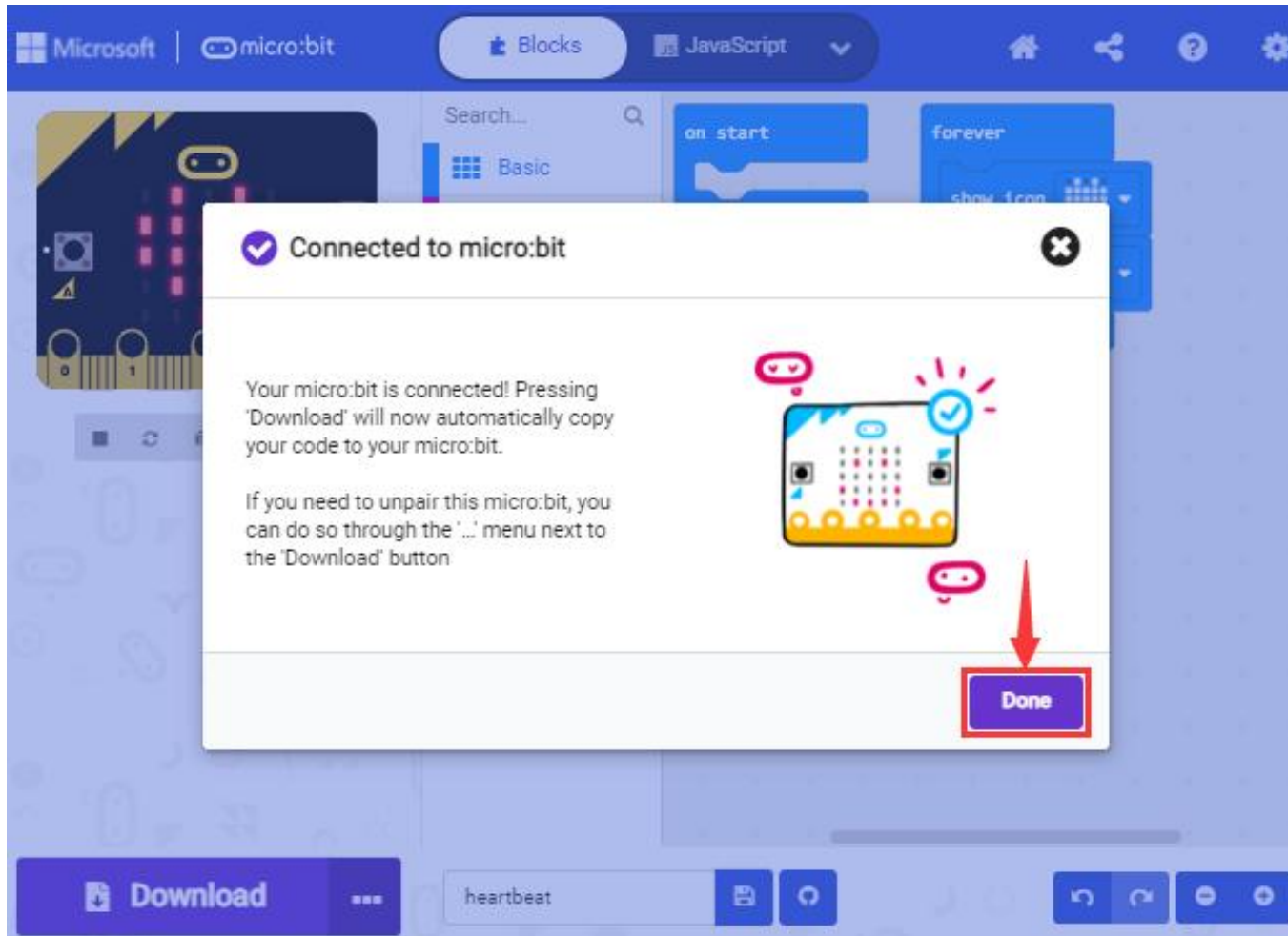
Then select the corresponding device and click "Connect". If no device shows up for selection, please refer to the link: <https://makecode.microbit.org/device/usb/webusb/troubleshoot>

And for updating the firmware of the Micro:bit: <https://microbit.org/guide/firmware/> .

If the links are too troublesome for you , then you can also turn to our **'Troubleshooting Downloads with WebUSB.pdf'**



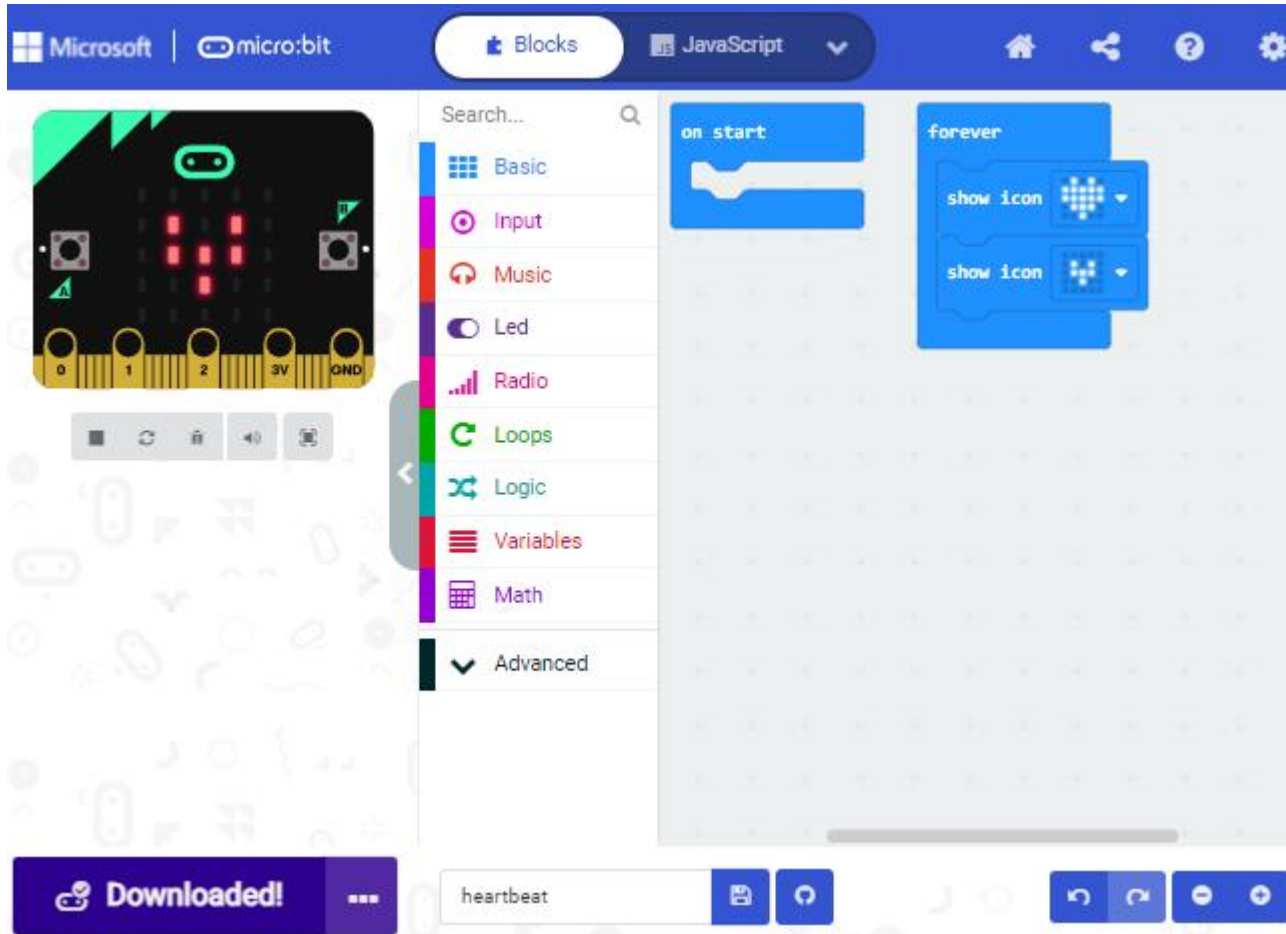
Click "Done" to finish the pairing.



Download Program

After pairing, click “download” to directly download the program to the board. If it is successfully downloaded, the

icon  will shift to .



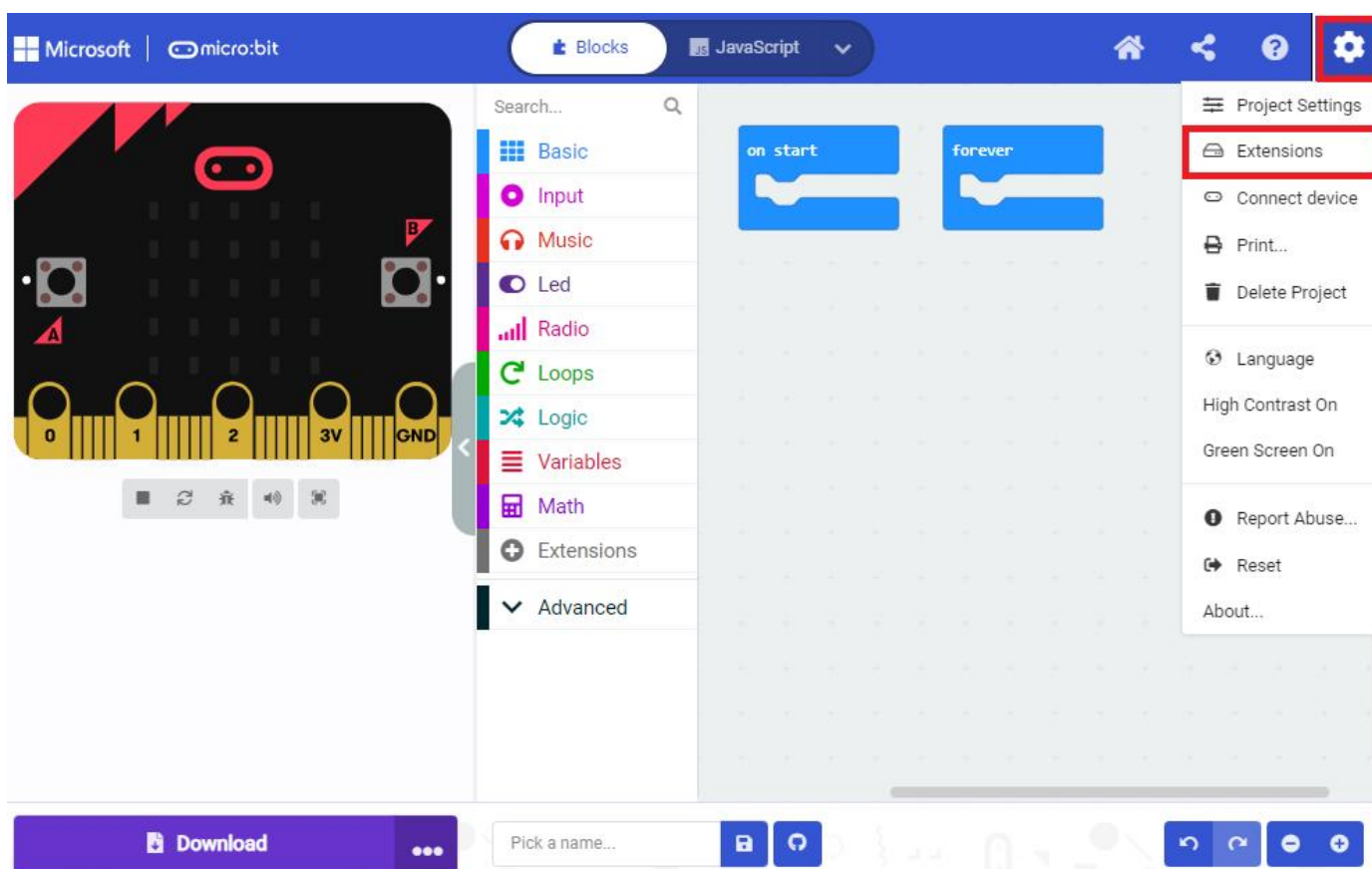
7.4. Makecode Extension Libraries

We have made a makecode extension library for this Mecanum robot car V2.0

Add an extension library of the Mecanum robot car V2.0

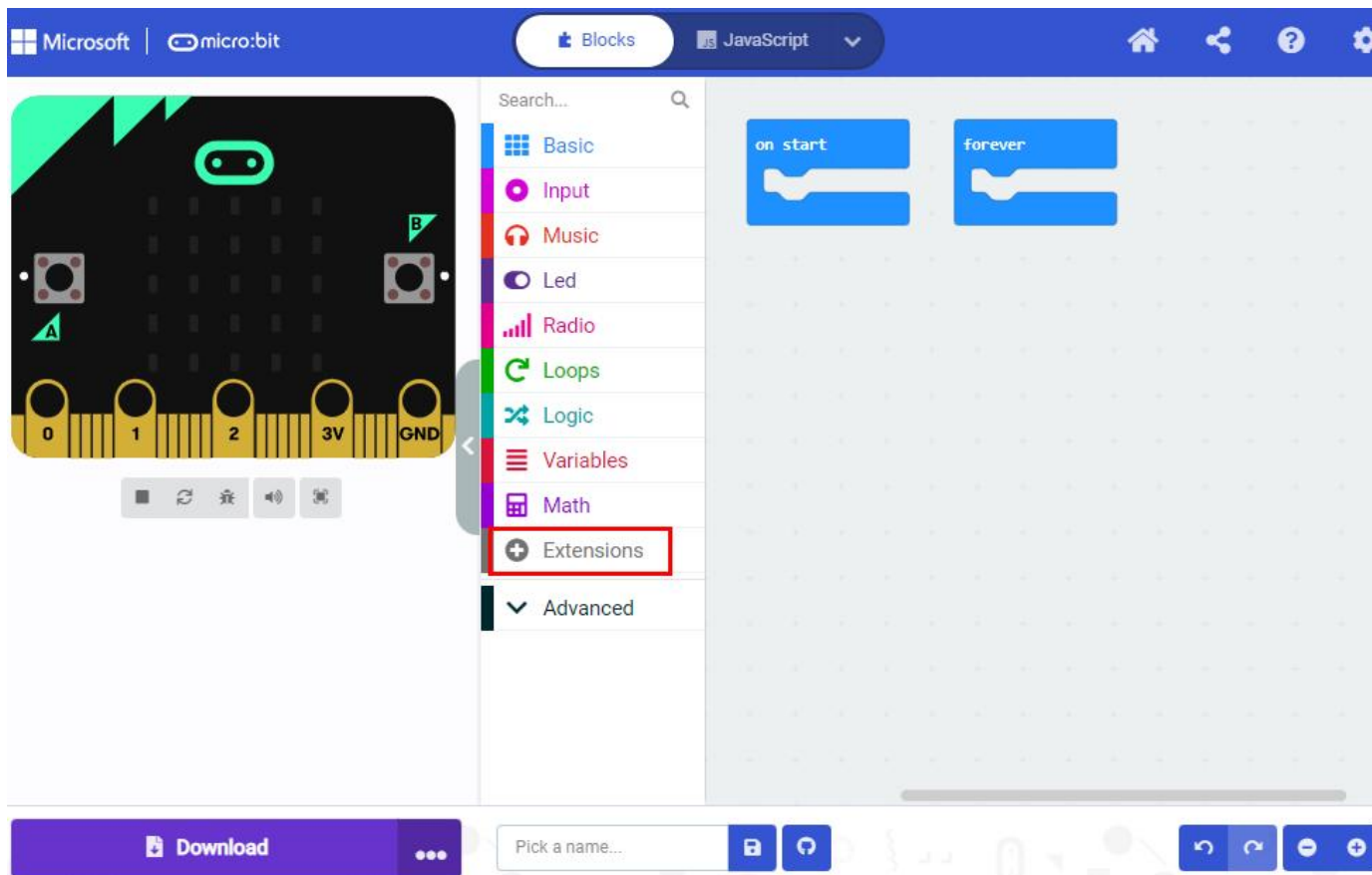
Please follow the following steps to add extension files:

Open Makecode to enter a certain project → click the gear-shaped icon (for setting) in the upper right corner → choose “Extensions” ;



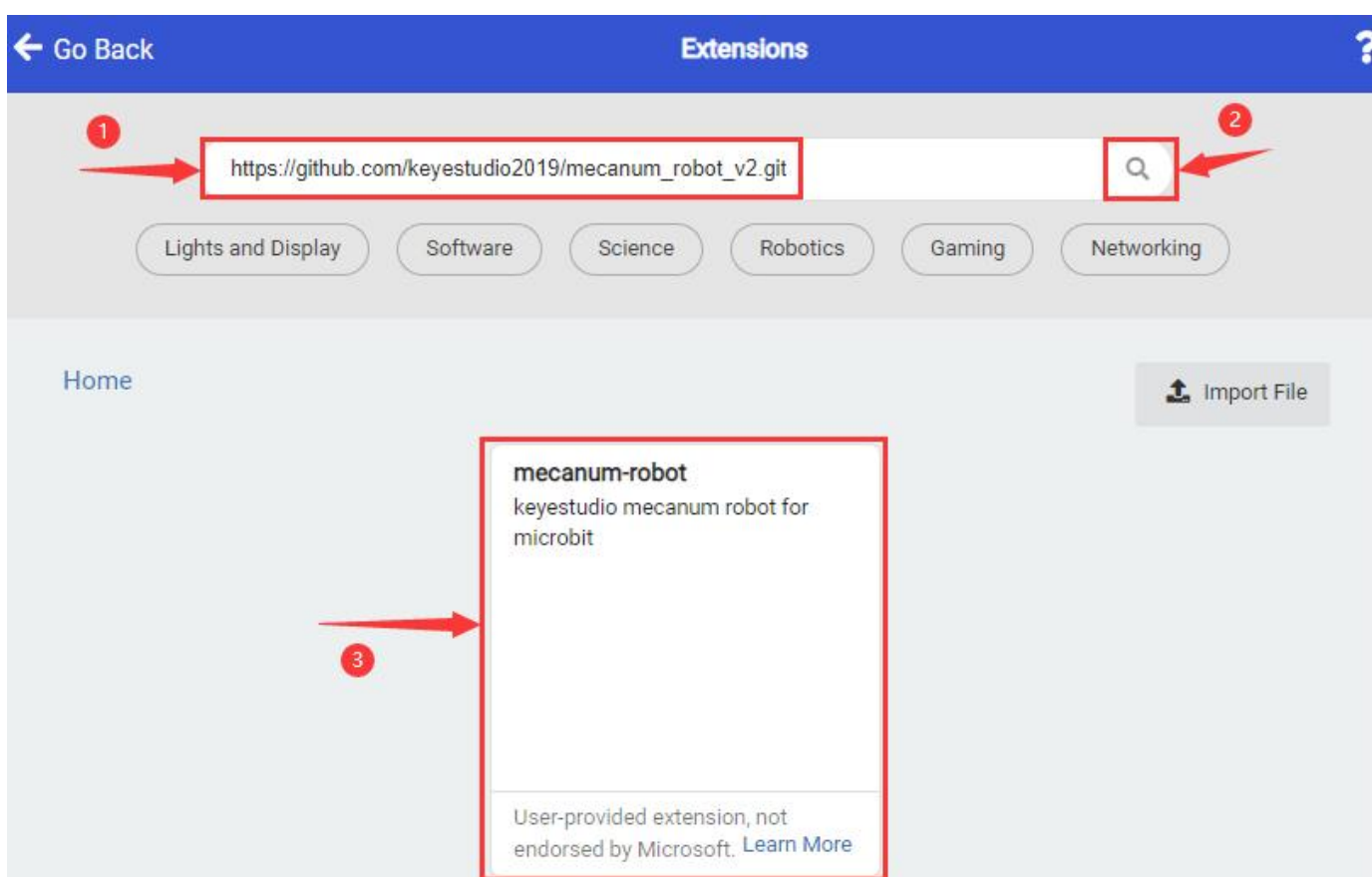
Or click “Extensions” , as shown below:

keyestudio



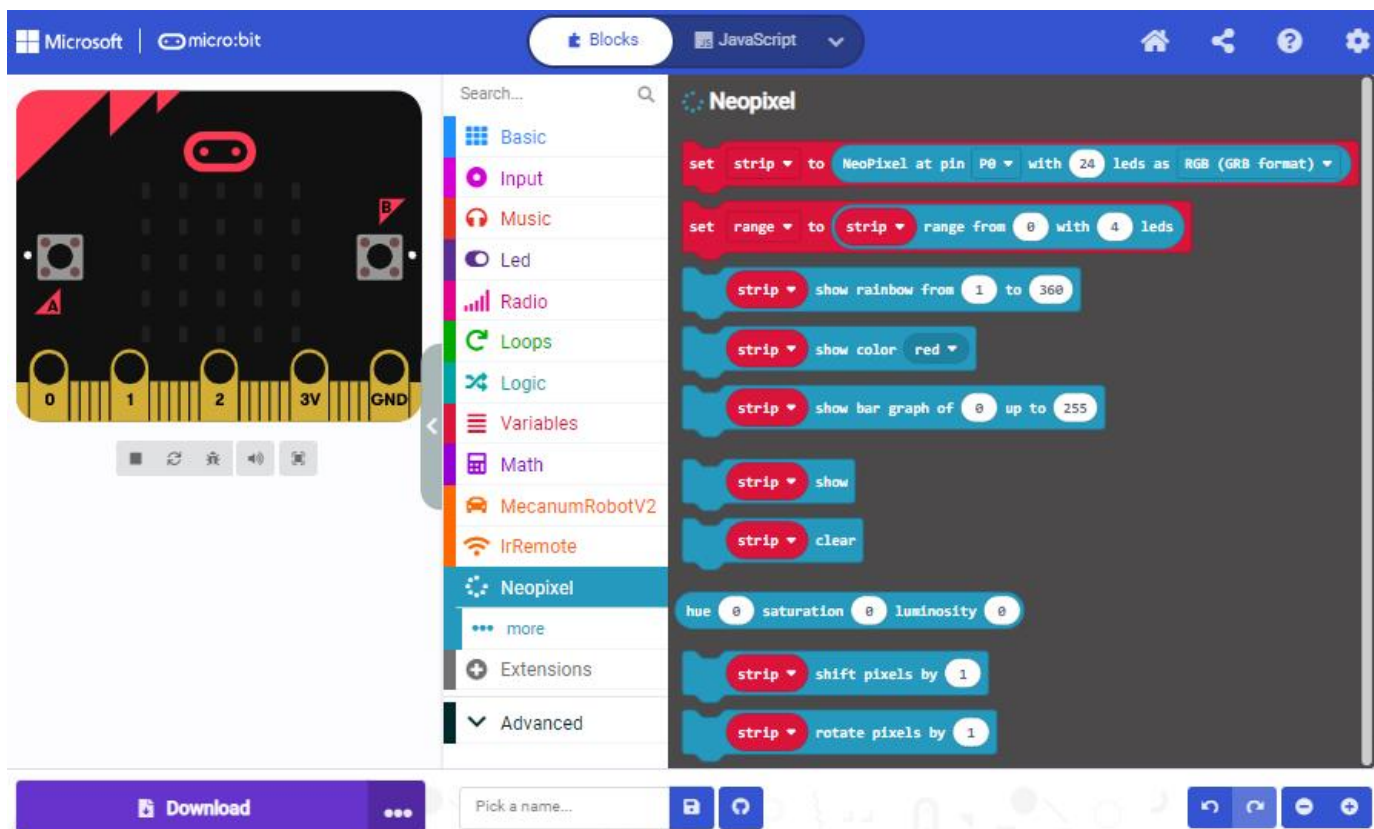
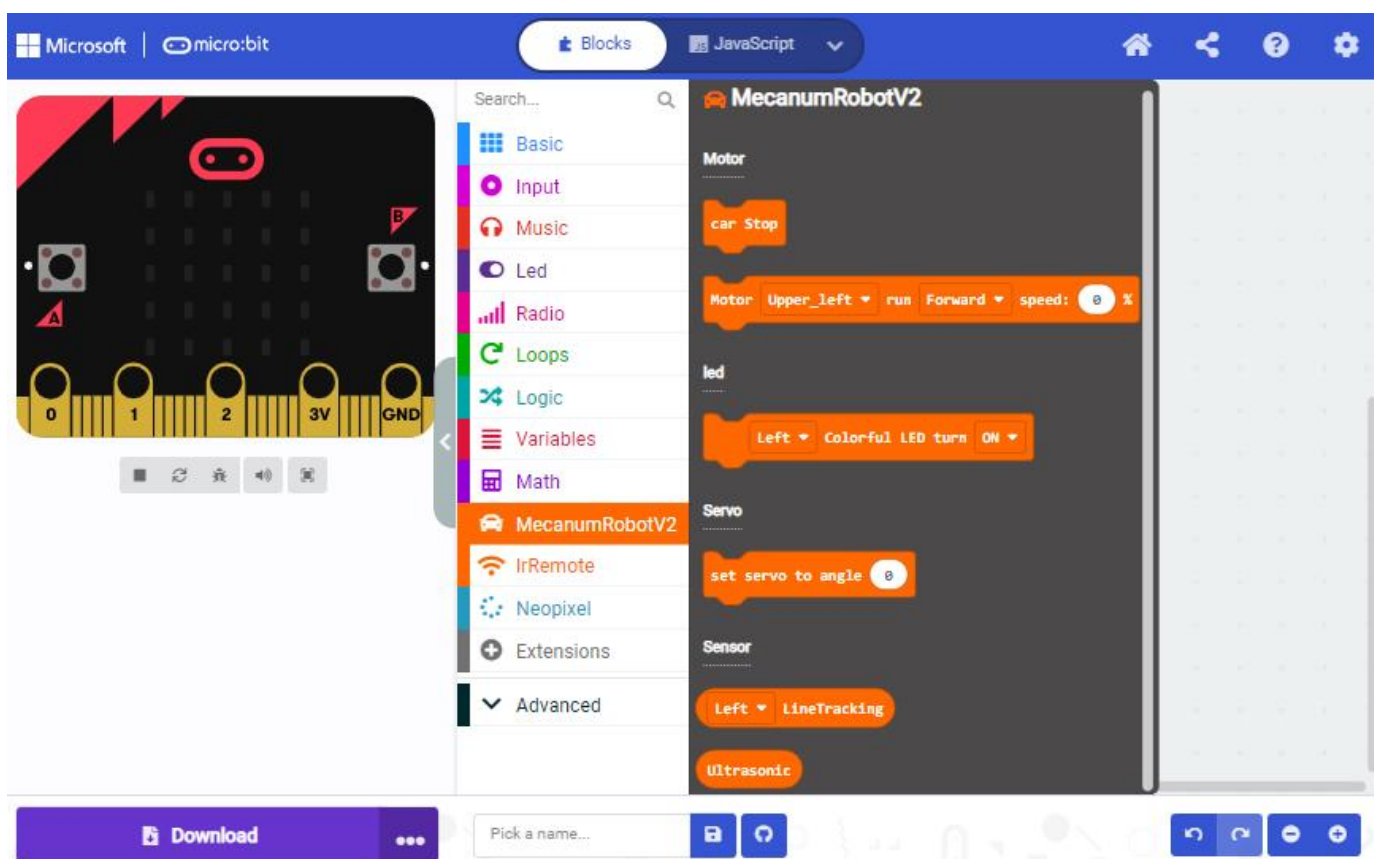
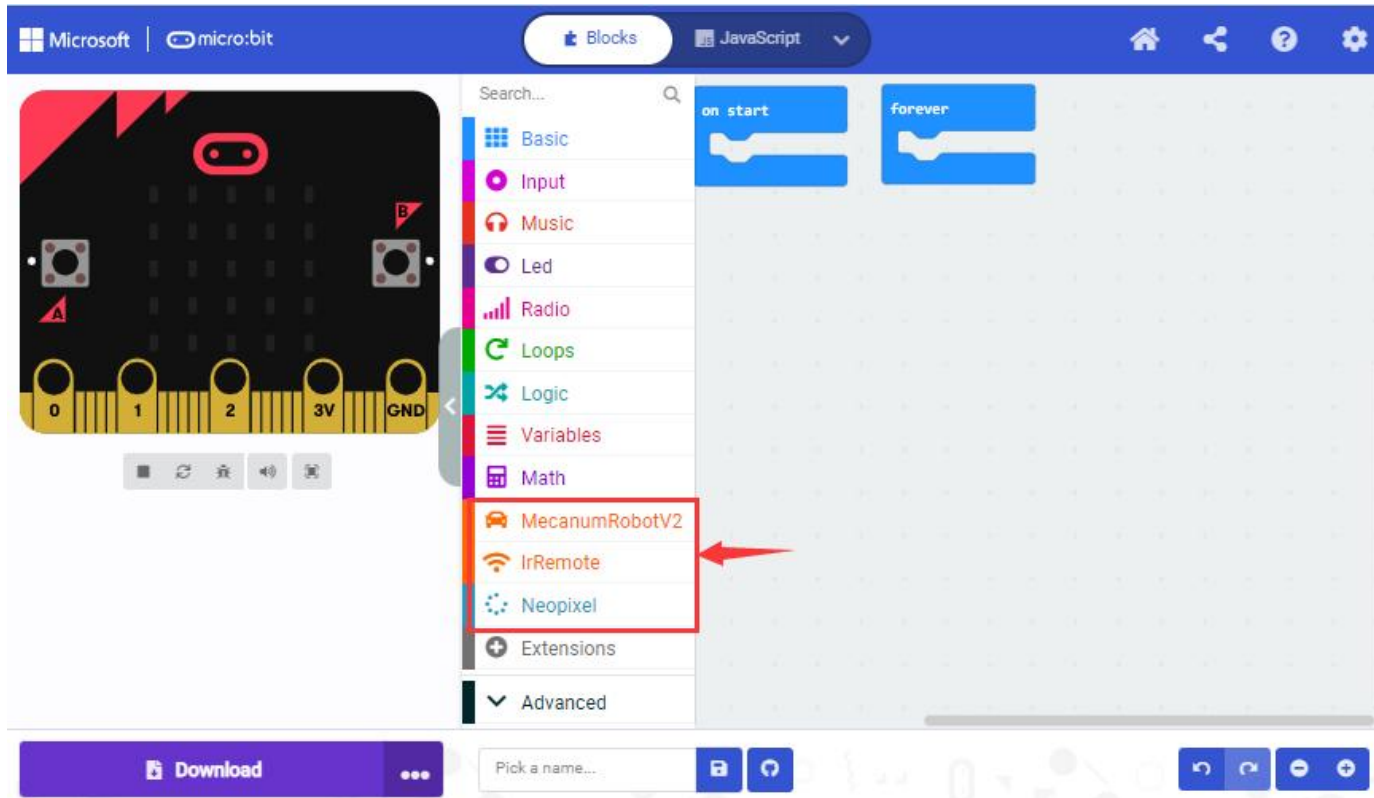
Input the link : https://github.com/keyestudio2019/mecanum_robot_v2.git to search;

Tap the searching result “MecanumRobotV2” to download and install it, This process may take a few seconds.



After the installation, you can find the extension files **Mecanum RobotV2** and **IrRemote** on the left side. And extension file Neopixel is also installed.

keystudio



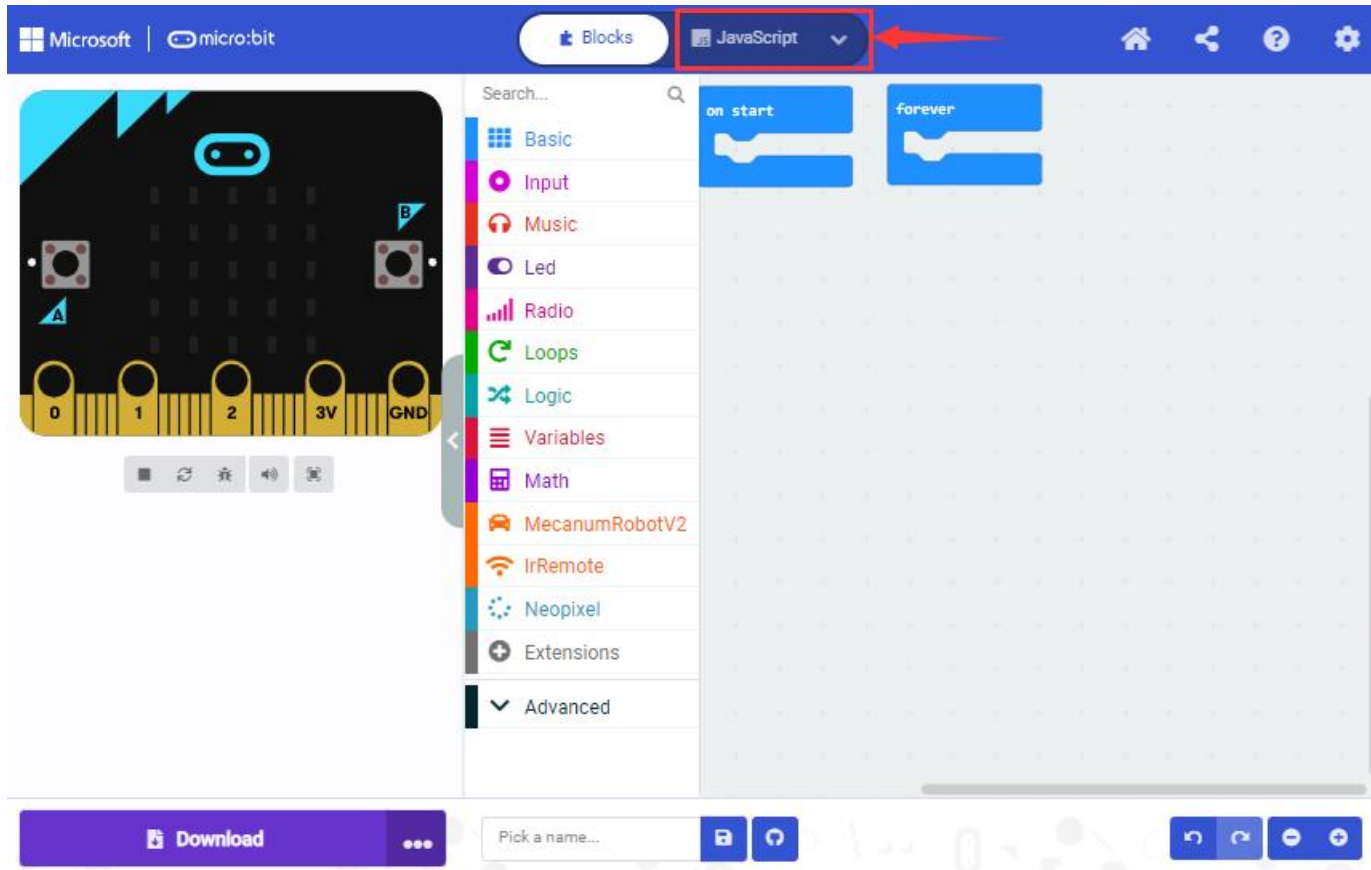
Note: The extension files added are only available for this project. Therefore, when you create a new **MecanumRobotV2** project, you will need to add these extension files again.

Update or delete the MecanumRobotV2 extension files:

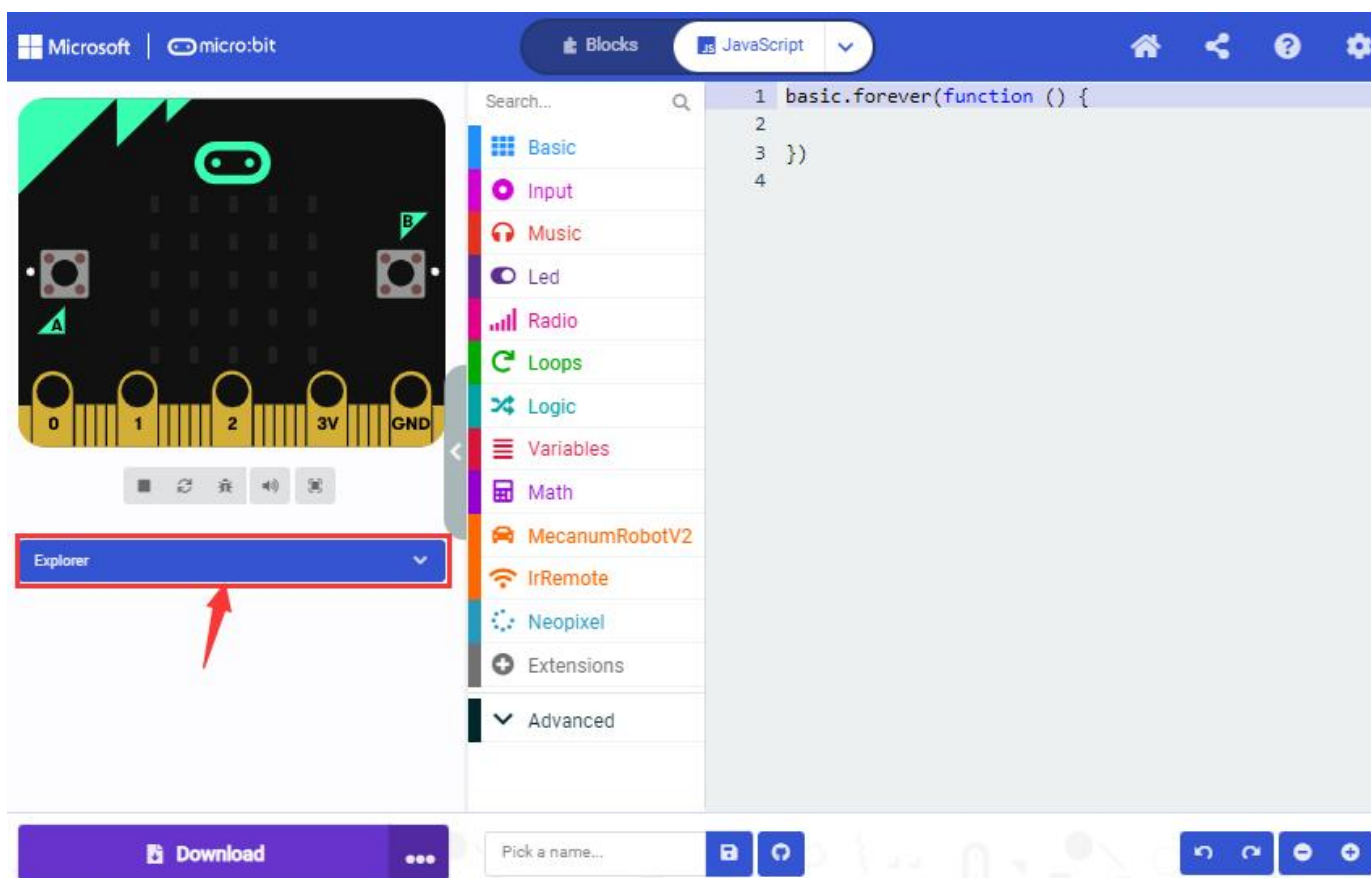
keyestudio

Please follow the following steps to update or delete extension files:

Click "Js JavaScript" to change to textual version:



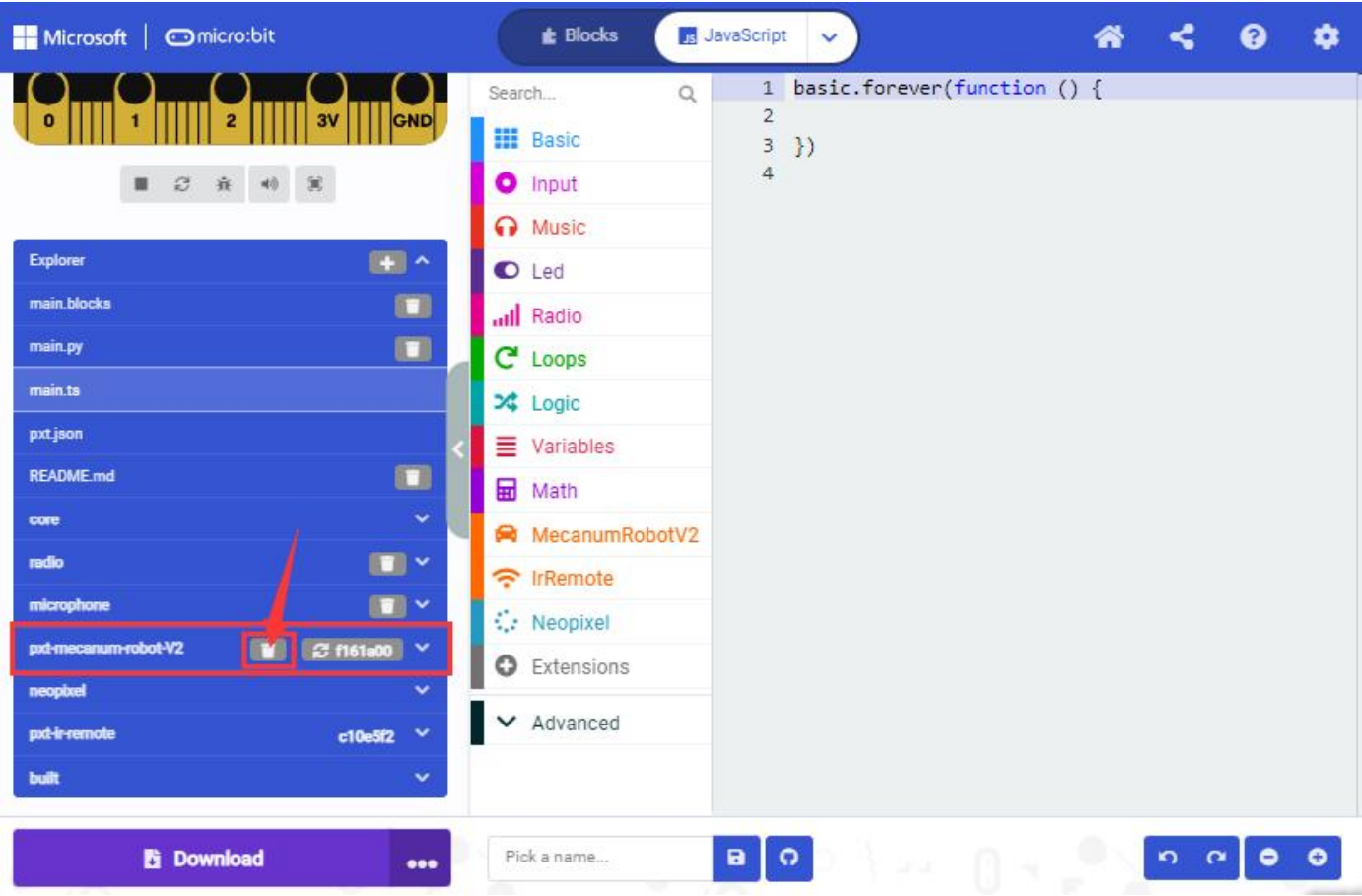
Click the "Explorer" on the left side:



You can find these added files in the list;

Click the dustbin icon beside the file to delete the MecanumRobotV2 file and tap the refresh icon to update it.

keystudio



7.5.Resources and Test Code

Download link: <https://fs.keyestudio.com/KS4034-4035>

Once downloaded and unzipped, a file named KS4034(KS4035) Keyestudio Micro bit 4WD Mecanum Robot Car Kit V2.0 will lie in sight.

Open it , as shown in the figure below:

1. 安装Microbit 驱动	5/20/2021 5:59 PM	File folder
2. Makecode 课程	7/9/2021 2:56 PM	File folder
3. Python 课程	7/9/2021 2:53 PM	File folder
4. 如何更新Microbit固件	5/20/2021 5:57 PM	File folder
5. MAINTENANCE 模式故障解决	7/9/2021 2:48 PM	File folder
6. 使用 WebUSB 进行下载的故障排除	7/9/2021 2:48 PM	File folder
7. Cool Term 软件	5/20/2021 5:07 PM	File folder

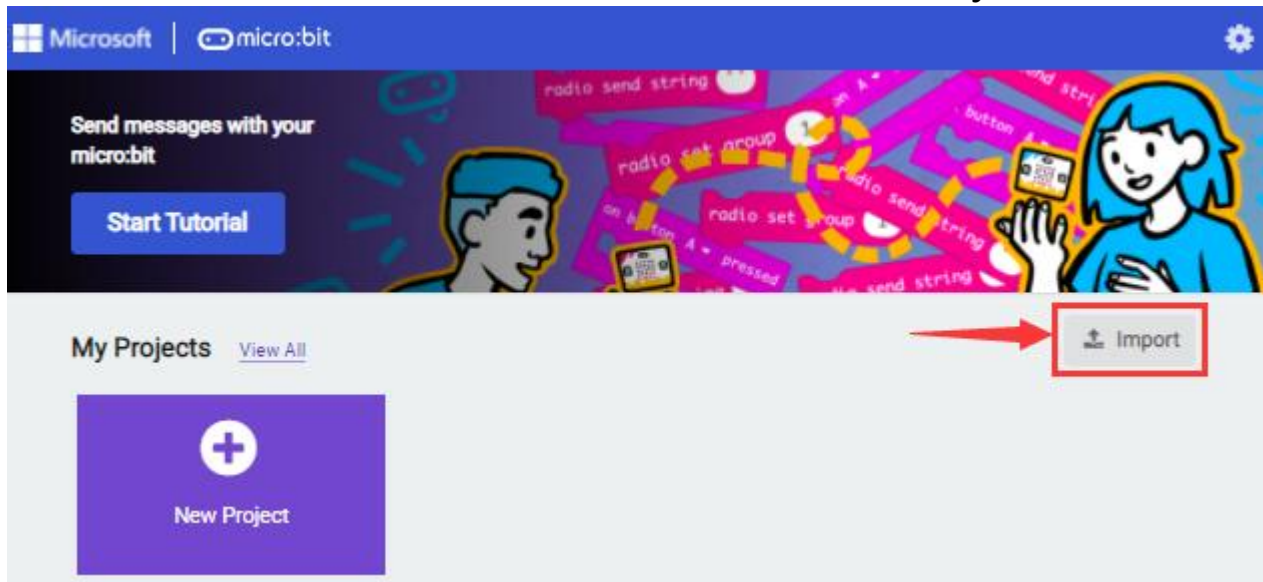
7.6.Import test code

We provide hexadecimal code files (project files) for each project. The file contains all the contents of the project and can be imported directly, and you can manually drag the code blocks to complete the program.

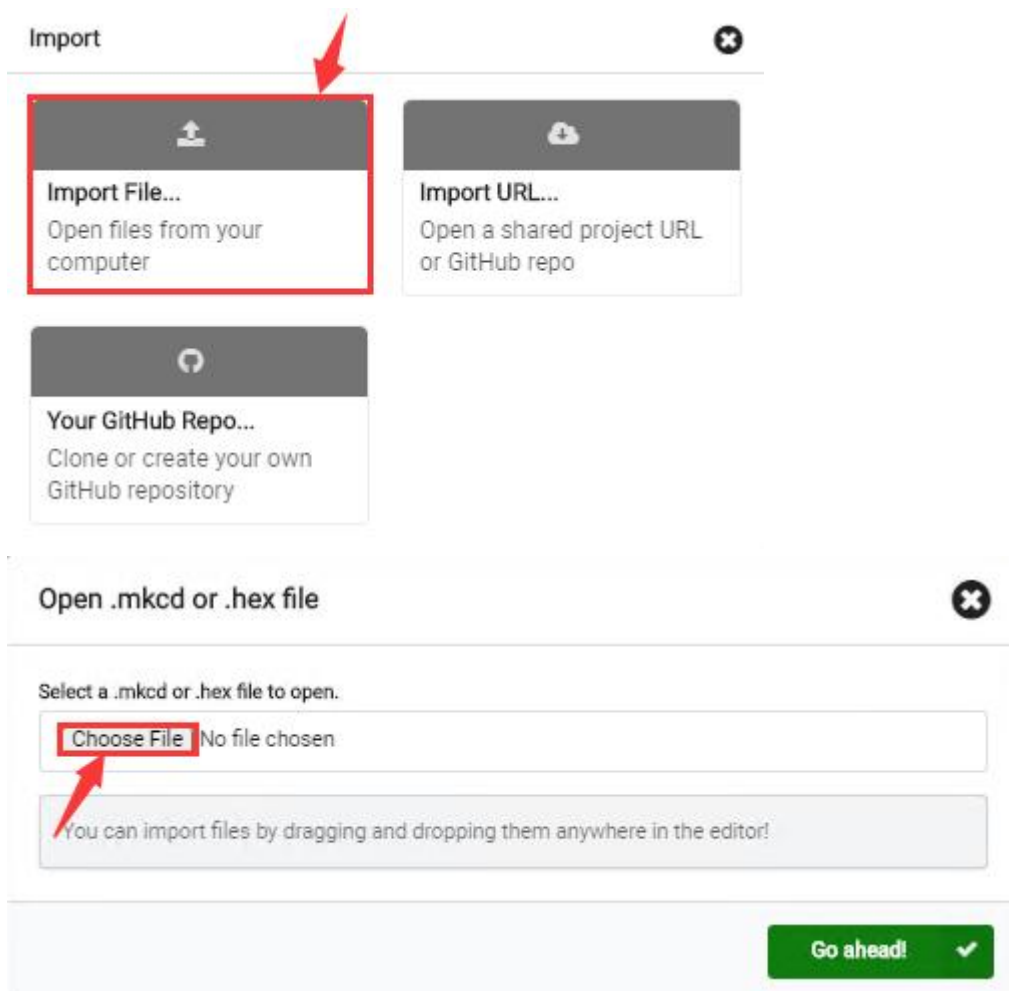
For simple projects, dragging a block of code to complete the program is recommended. For complex projects, it is recommended to conduct the program by importing the provided hexadecimal code file.

Let's take the "Heatbeat" project as an example to show how to load the code.

Open the Web version of Makecode or the Windows 10 App version of Makecode then click “Import” .

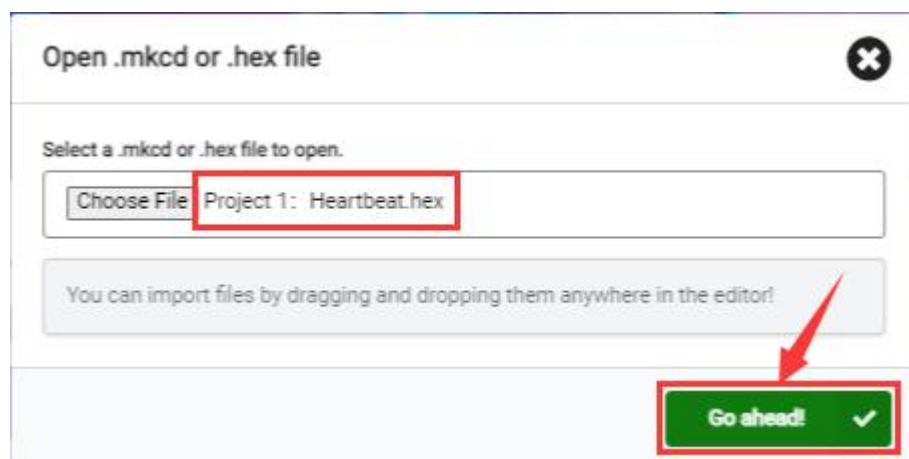
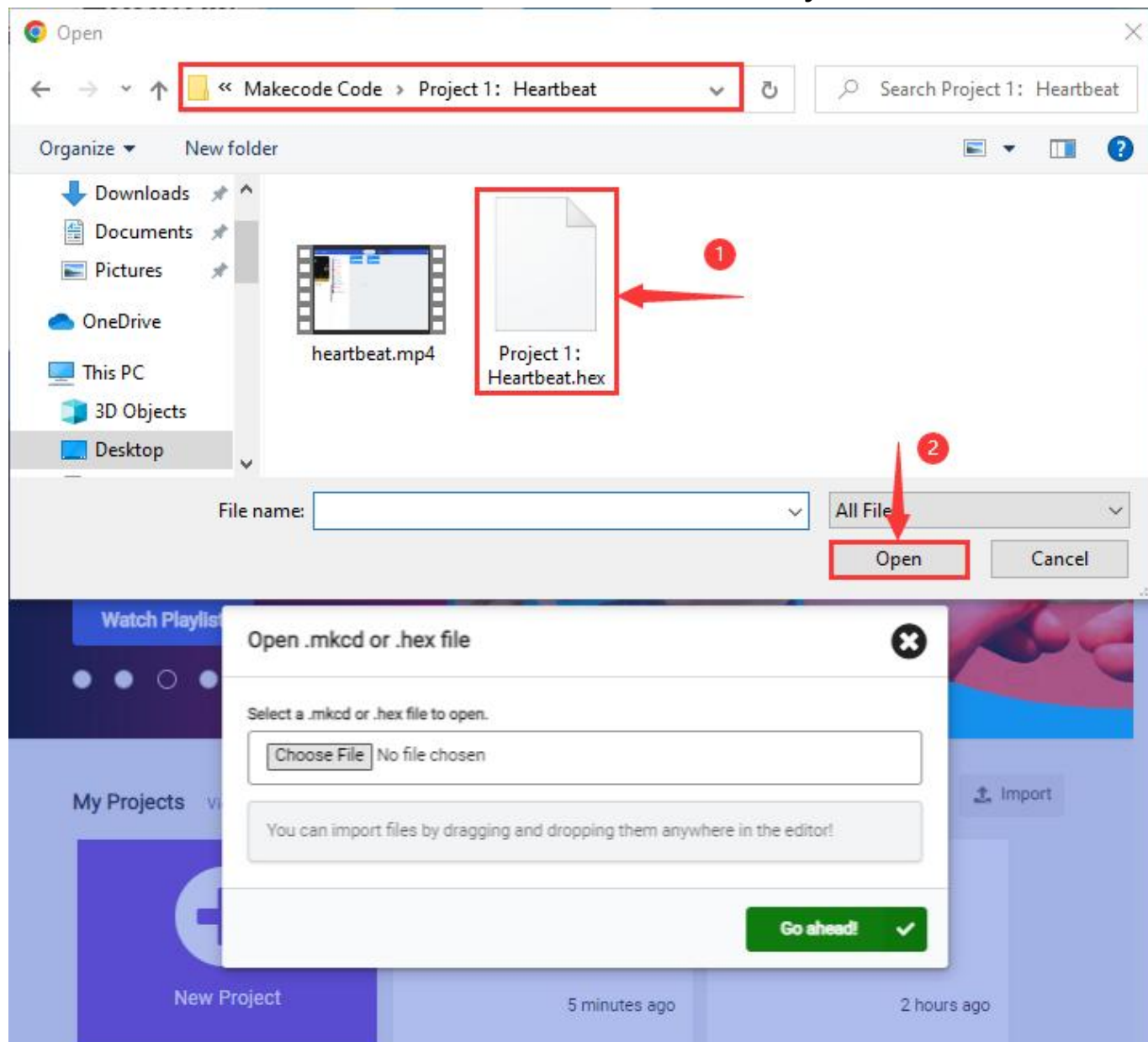


Click "Import File" ;



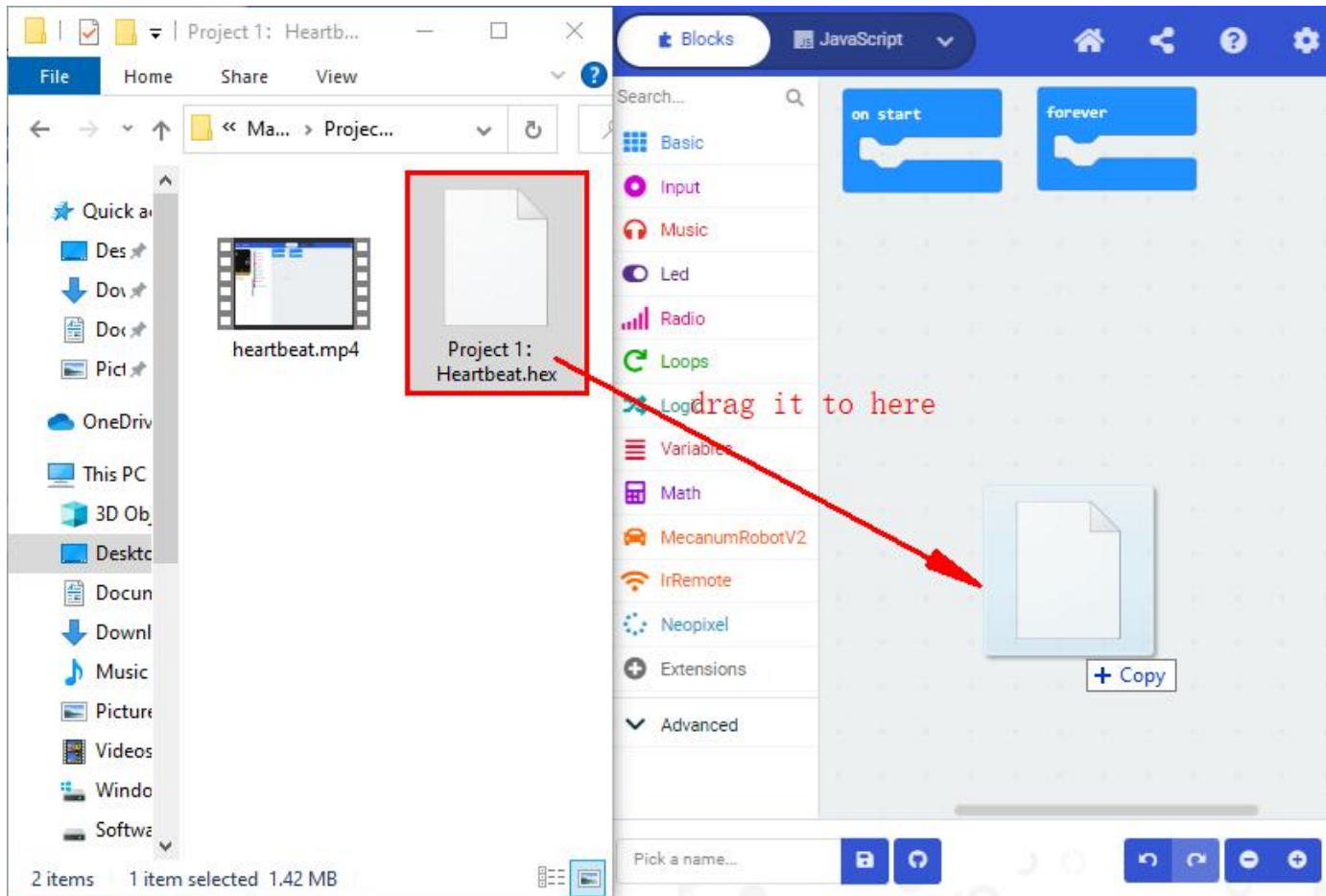
Select "..\2. Makecode Tutorial\Makecode Code\Project 1: Heartbeat\Project 1: Heart beat.hex" , then click "Go ahead" .

keystudio

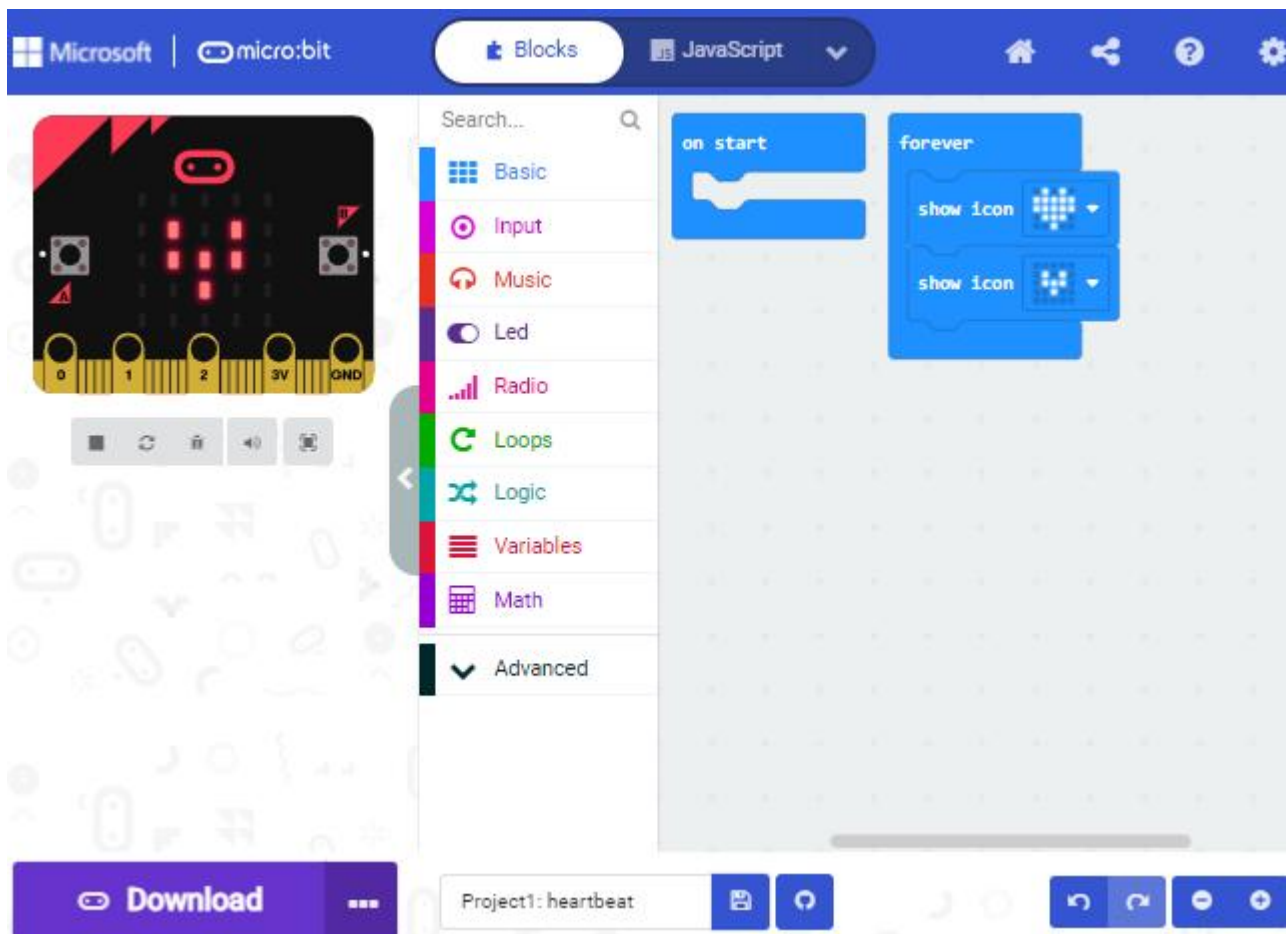


In addition to importing the provided test code file into the Makecode compiler above, you can also drag them to the code editing area of the Makecode compiler, as shown in the figure below:

keyestudio



After a few seconds, it has done.



Note: If your computer system is not Windows 10, the pairing cannot be done via Google Chrome. Therefore, digital or analog signals of sensors and modules cannot be read. However, we can use the CoolTerm software to read the serial port data. Next chapter is about how to install the CoolTerm.

7.7.Install CoolTerm

CoolTerm program is used to read the data on serial port.

Download CoolTerm program: <https://freeware.the-meiers.org/>

keyestudio

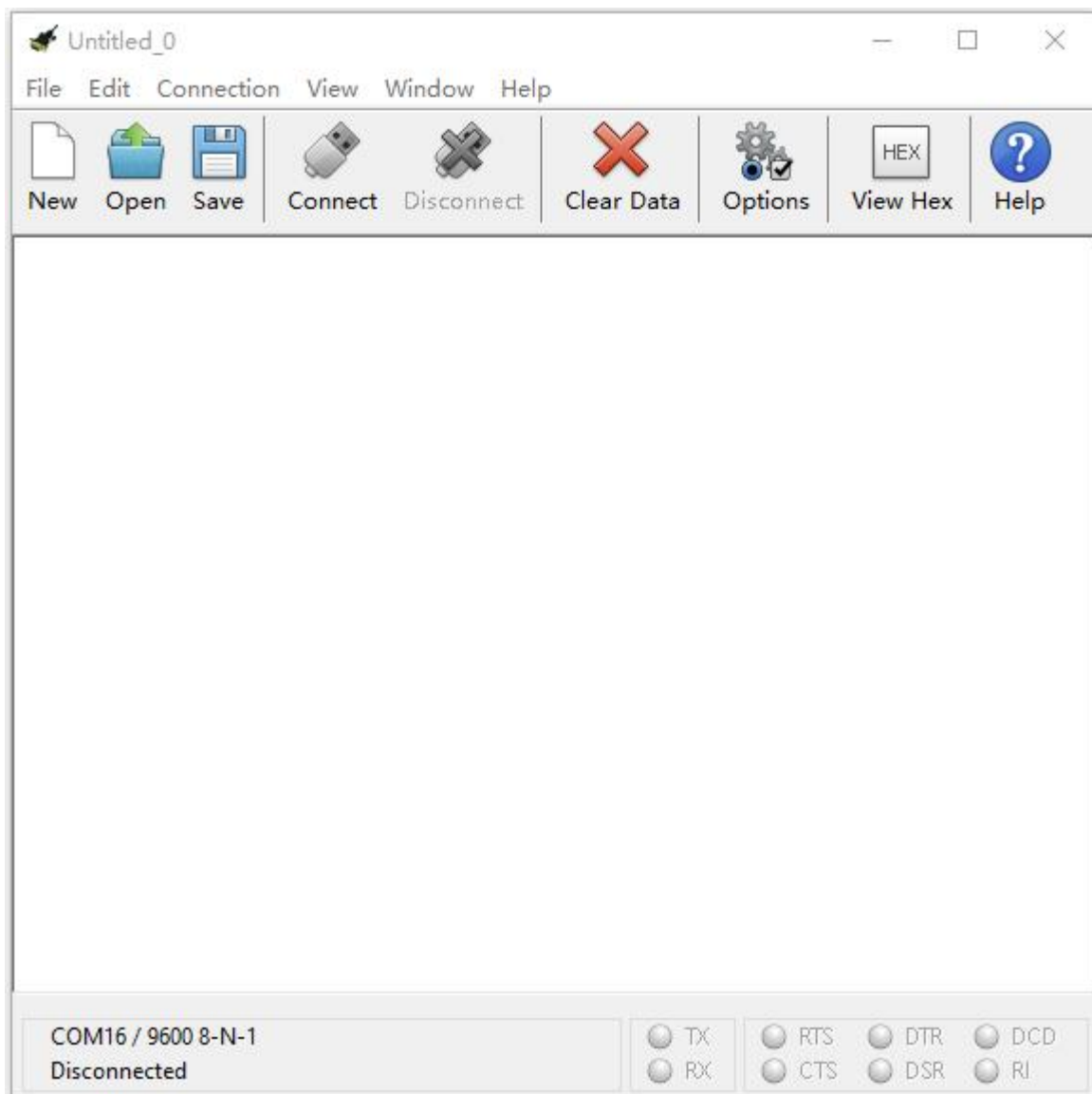
After downloading, we need to install **CoolTerm** program file, and we take **PC Window system** as an example.

- (1) Choose "win" to download the zip file of CoolTerm
- (2) Unzip the file and open it. (also suitable for Mac and Linux system)



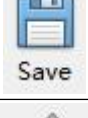
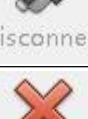
CoolTerm Libs	2020/4/21 11:20	File folder	
CoolTerm Resources	2020/4/21 11:20	File folder	
CoolTerm.exe	2019/5/17 22:56	Application	5,314 KB
msvcp120.dll	2019/4/3 14:33	Application extension	645 KB
msvcp140.dll	2019/4/3 14:33	Application extension	625 KB
msvcr120.dll	2019/4/3 14:33	Application extension	941 KB
ReadMe.txt	2019/5/18 20:35	Text Document	31 KB
vccorlib140.dll	2019/4/3 14:33	Application extension	387 KB
vcruntime140.dll	2019/4/3 14:33	Application extension	88 KB
Windows System Requirements.txt	2018/1/7 14:29	Text Document	1 KB
XojoGUIFramework64.dll	2019/4/3 14:33	Application extension	30,801 KB

- (3) Double-click CoolTerm.exe . (Please make sure that the driver of Micro:bit is installed and the main board is connected to the computer.)



The functions of each button on the Toolbar are listed below:

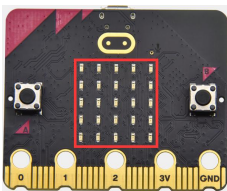


	Opens up a new Terminal
	Opens a saved Connection
	Saves the current Connection to disk
	Opens the Serial Connection
	Closes the Serial Connection
	Clears the Received Data
	Opens the Connection Options Dialog
	Displays the Terminal Data in Hexadecimal Format
	Displays the Help Window

8. Projects

(Note: project 8.1 to 8.12 are basic courses conducted with the built-in sensors and LED dot matrix of the Micro:bit main board V2, while project 8.13 to 8.20 are extended courses)

Project 1: Heart beat



1. Description

This project is easy to conduct solely with a micro:bit main board, a micro USB cable and a computer. The micro:bit will display a “flickering heart” pattern. This experiment serves as a starter for your to entry to the magical world

of the micro:bit.

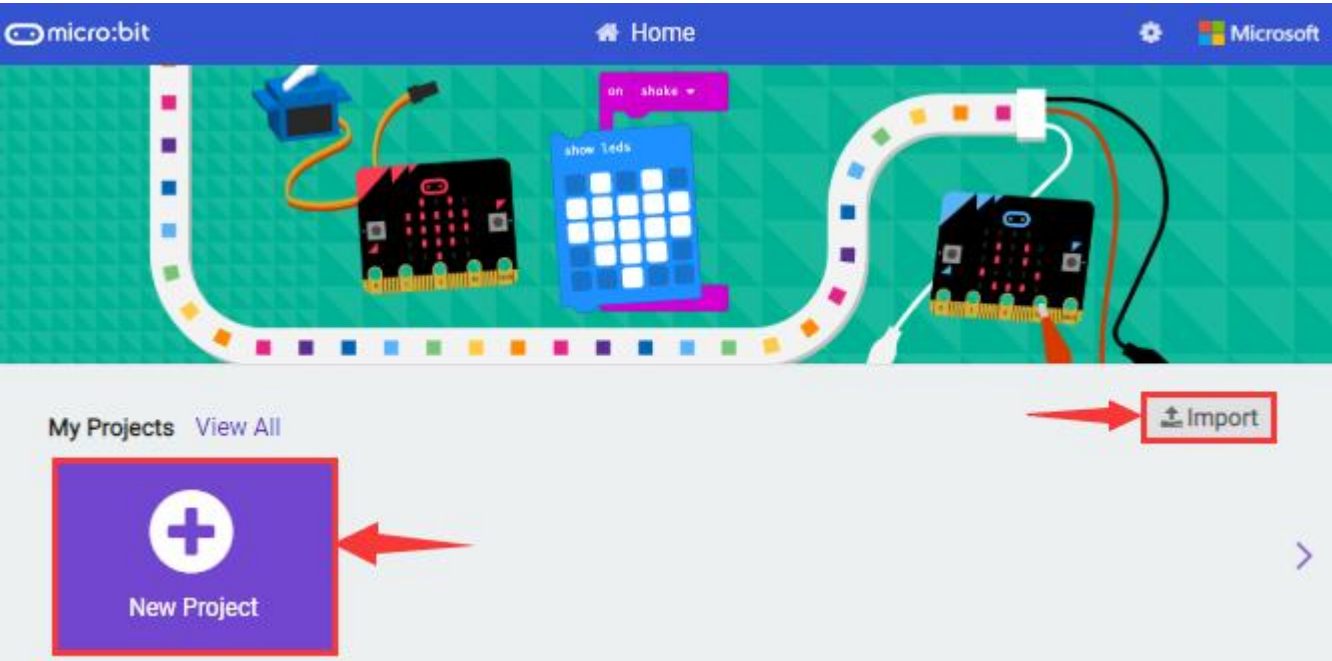
2. Preparation

A. Attach the micro:bit main board to your computer via the USB cable

B. Open the Web version of Makecode.

Import Hex profile (How to import?)

Click “New Project” to drag blocks step by step



3. Test Code

Route to get test code (**How to load?**)

File Type	Path	File Name
Hex file	..\2. Makecode Tutorial\Makecode Code\Project 1: Heartbeat	Project 1: Heartbeat.hex

Or you could edit code in the editing area.

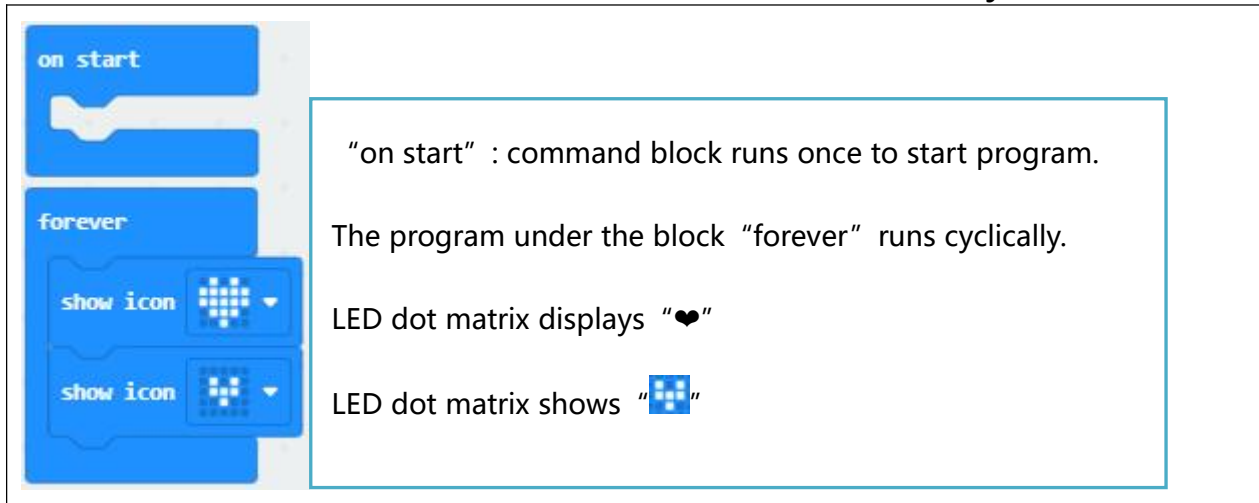
Go to “Basic” → “show icon” .

Copy it again and place it into “forever” block.

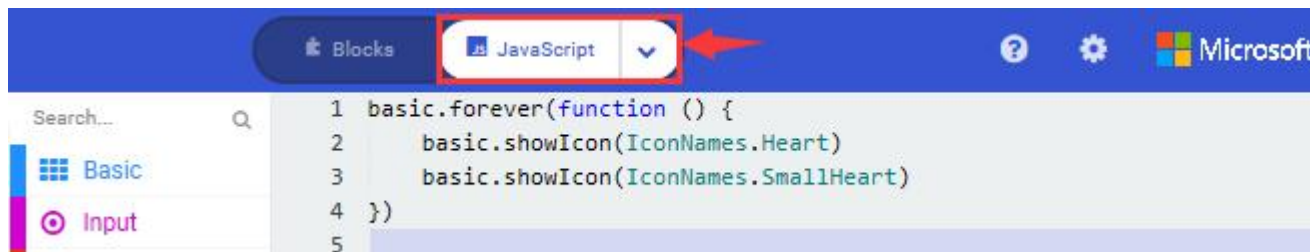
Click “♥” to select “” .



Complete Program:



Click "JavaScript" to view the corresponding JavaScript code:



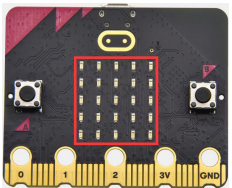
4. Test Result

Download code to micro:bit and keep the USB cable connected. The LED dot matrix will display ♥ and 5x5 grid of dots ceaselessly.

(How to download? How to quick download?)

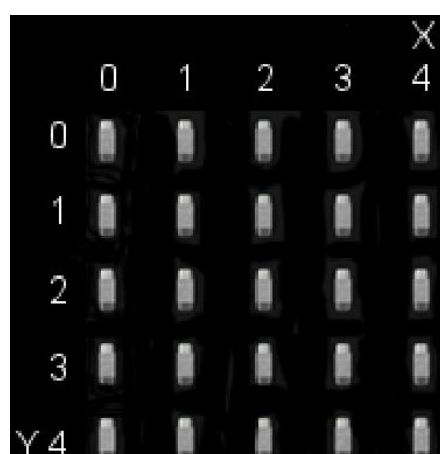
If the download is not success, try to disconnect the micro:bit from your computer and then reconnect them and reopen Makecode to try again.

Project 2: Light A Single LED



1. Description

The LED dot matrix consists of 25 diodes arranged in a 5 by 5 square and placed at the intersection of row lines (X) and column lines (Y). We can control one of the 25 LEDs by setting coordinate points. For example, the first LED sits in the first line is (0,0) and the third LED positioned in the first line is (2,0) and others likewise.



2. Preparation

- A. Attach the micro:bit main board to your computer via the USB cable
- B. Open the Web version of Makecode.

Import Hex profile (How to import?)
Click “New Project” to drag blocks step by step

3. Test Code

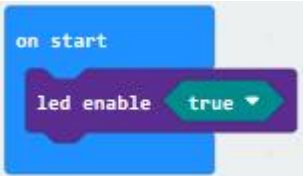
Route to get test code ([How to load?](#))

File Type	Path	File Name
Hex file	..\2. Makecode Tutorial\Makecode Code	Project 2: Light A Single LED.hex

Or you could edit code in the editing area.

A. Click “Led” → “more” → “led enable false”

B. Put it into the “on start” block, and click the drop-down triangle button to select “true” .



C. Enter “Led” → “toggle x 0 y 0” block, and drag it into “forever” , then alter “x 0” into “x 1” .



D. Enter “ Basic ” → “ pause (ms) 100 ” from “ ” , then move it to the “forever” block, and set to delay



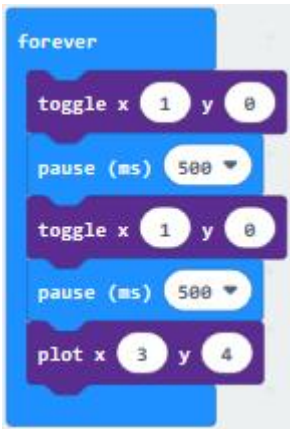
500ms.



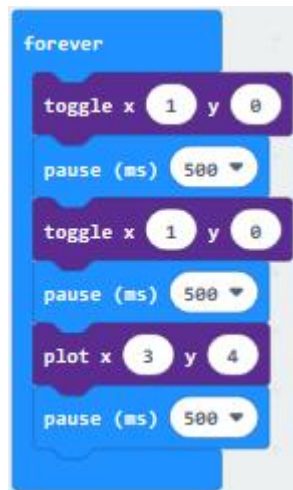
E. Duplicate code string once and place it into “forever” block.



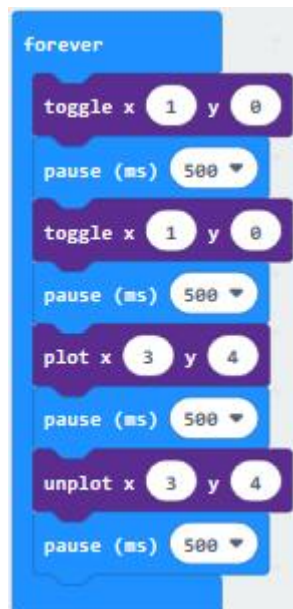
F. Enter “Led” , and drag “plot x 0 y 0” into “forever” block, then alter “x 0 y 0” into “x 3 y 4” .



G. Replicate "pause (ms) 500" once and place it into "forever" block.



H. Click "Led" → "unplot x 0 y 0" and set to "unplot x3 y 4" then copy "pause (ms) 500" block once, and place it into "forever" block.



Complete Program:



```

on start
  led enable true

forever
  toggle x 1 y 0
  pause (ms) 500
  toggle x 1 y 0
  pause (ms) 500
  plot x 3 y 4
  pause (ms) 500
  unplot x 3 y 4
  pause (ms) 500
        
```

“on start”: command block only runs once to start program.

Turn on LED dot matrix.

The program under the block “forever” runs cyclically.

Toggle the LED brightness at coordinate point “x 1 y 0”.

Toggle the LED brightness at coordinate point “x 1 y 0”.

Turn on the LED at coordinate point “x3, y4”.

Delay in 500ms

Turn off the LED at coordinate point “x3 y4”.

Click “JavaScript” to switch into corresponding JavaScript code:



```

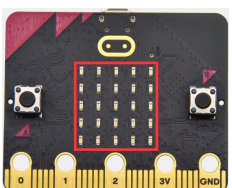
1 led.enable(true)
2 basic.forever(function () {
3   led.toggle(1, 0)
4   basic.pause(500)
5   led.toggle(1, 0)
6   basic.pause(500)
7   led.plot(3, 4)
8   basic.pause(500)
9   led.unplot(3, 4)
10  basic.pause(500)
11 })
12
        
```

4. Test Result

After uploading the test code to micro:bit main board V2 and powering it via the USB cable, the LED in (1,0) lights up for 0.5s and the one in (3,4) shines for 0.5s and repeat this sequence.

(How to download? How to quick download?)

Project 3: 5 x 5 LED Dot Matrix



1. Description

Dot matrix is very commonplace in daily life, which has found wide applications in LED advertisement screens, elevator floor display, bus stop announcement and so on.

The LED dot matrix of Micro: Bit main board contains 25 Diodes. Previously, we have succeeded in controlling a certain LED via its position point. Supported by the same theory, we can turn on many LEDs at the same time to showcase patterns, digits and characters.

What’ s more, we can also click” show icon ” to choose the pattern we like to display. Last but not the least, we can design patterns by ourselves as well.

2. Preparation

- A. Attach the micro:bit main board to your computer via the USB cable
- B. Open the Web version of Makecode.

3. Test Code

Test Code1:

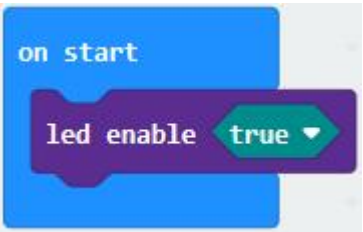
The route to get test code ([How to load?](#))

File Type	Path	File Name
Hex file	..\2. Makecode Tutorial\Makecode Code	Project 3: LED Dot Matrix-1.hex

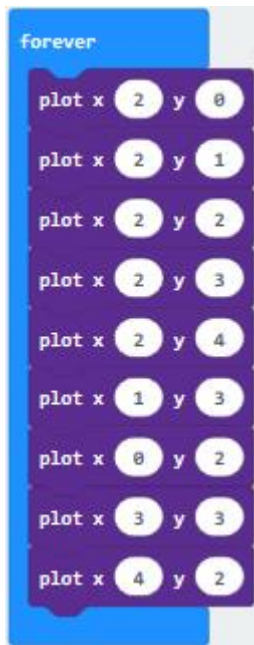
Or you could edit code step by step in the editing area.

Enter “Led” → “more” → “led enable false” n and place it into “on start” block

Click the drop-down triangle button to select “true”



Click “Led” , and move “plot x 0 y 0” into “forever” , then replicate “plot x 0 y 0” for 8 times, respectively set to “x 2” y 0” , “x 2” y 1” , “x 2” y 2” , “x 2” y 3” , “x 2” y 4” , “x 1” y 3” “x 0” y 2” , “x 3” y 3” , “x 4” y 2” .



Complete Program:

"on start" : command block only runs once to start program.

Turn on LED dot matrix.

The program under the block "forever" runs cyclically.

Toggle the LED brightness at coordinate point "x 2, y 0" , "x 2, y 1" , "x 2, y 2", "x 2, y 3" , "x 2, y 4" , "x 1, y 3" , "x 0, y 2" , "x 3, y 3" and "x 4, y 2"

Select "JavaScript" and "Python" to switch into JavaScript and Python language code:



Code 2:

The route to get test code ([How to load?](#))

File Type	Path	File Name
Hex file	..\2. Makecode Tutorial\Makecode Code	Project 3: LED Dot Matrix-2.hex

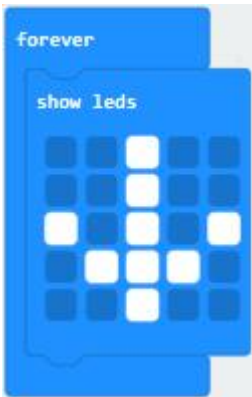
Or you could edit code step by step in the editing area.

Enter “Basic” → “show number 0” block,

Duplicate it for 4 times, then separately set to “show number 1” , “show number 2” , “show number 3” , “show number 4” , “show number 5” .



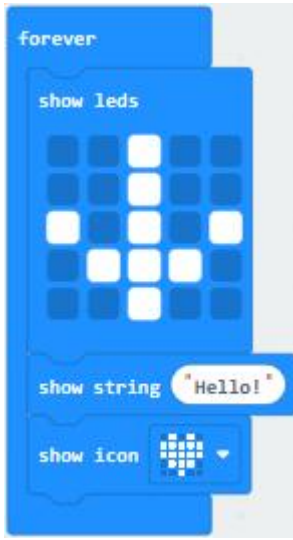
Click “Basic” → “show leds” , then put it into “forever” block, tick blue boxes to light LED and generate “↓” pattern.



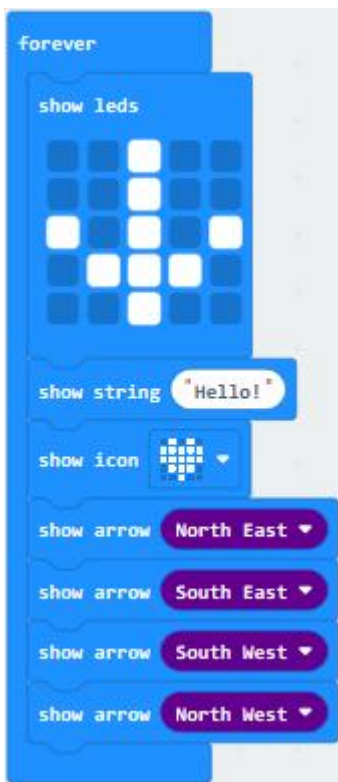
Move out the block “show string” from “Basic” block, and leave it beneath the “show leds” block



Choose “show icon” from “Basic” block, and leave it in the “forever” block



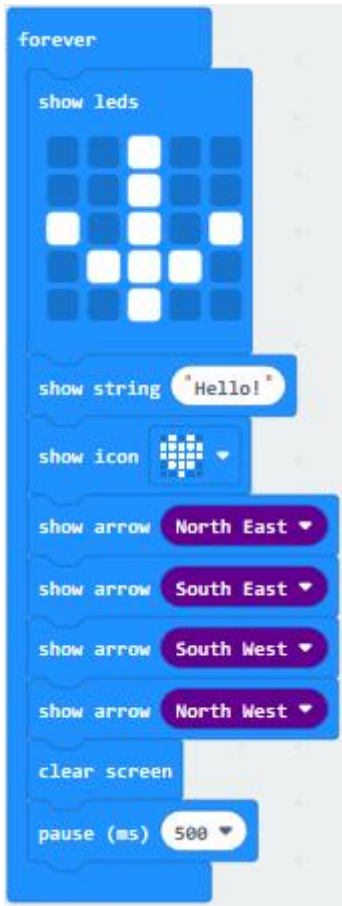
Enter “Basic” → “show arrow North” and leave it into “forever” block, then replicate “show arrow North” for 3 times, respectively set to “North East” , “South East” , “South West” , “North West” .



Click “Basic” and drag the block “clear screen” into “forever”



Drag “pause (ms) 100” block from “Basic” block and set the delay to 500ms, then place it in “forever” below “clear screen” block.



Complete Program:

“on start”: command block only runs once to start program.

LED dot matrix displays 1,2,3,4,5

Under the block “forever”, program runs cyclically.

Dot matrix shows the “↓” pattern

Dot matrix scrolls to show “Hello!”

“♥” is shown on dot matrix

LED dot matrix displays “North East” arrow.

The “South East” arrow shows up on LED dot matrix

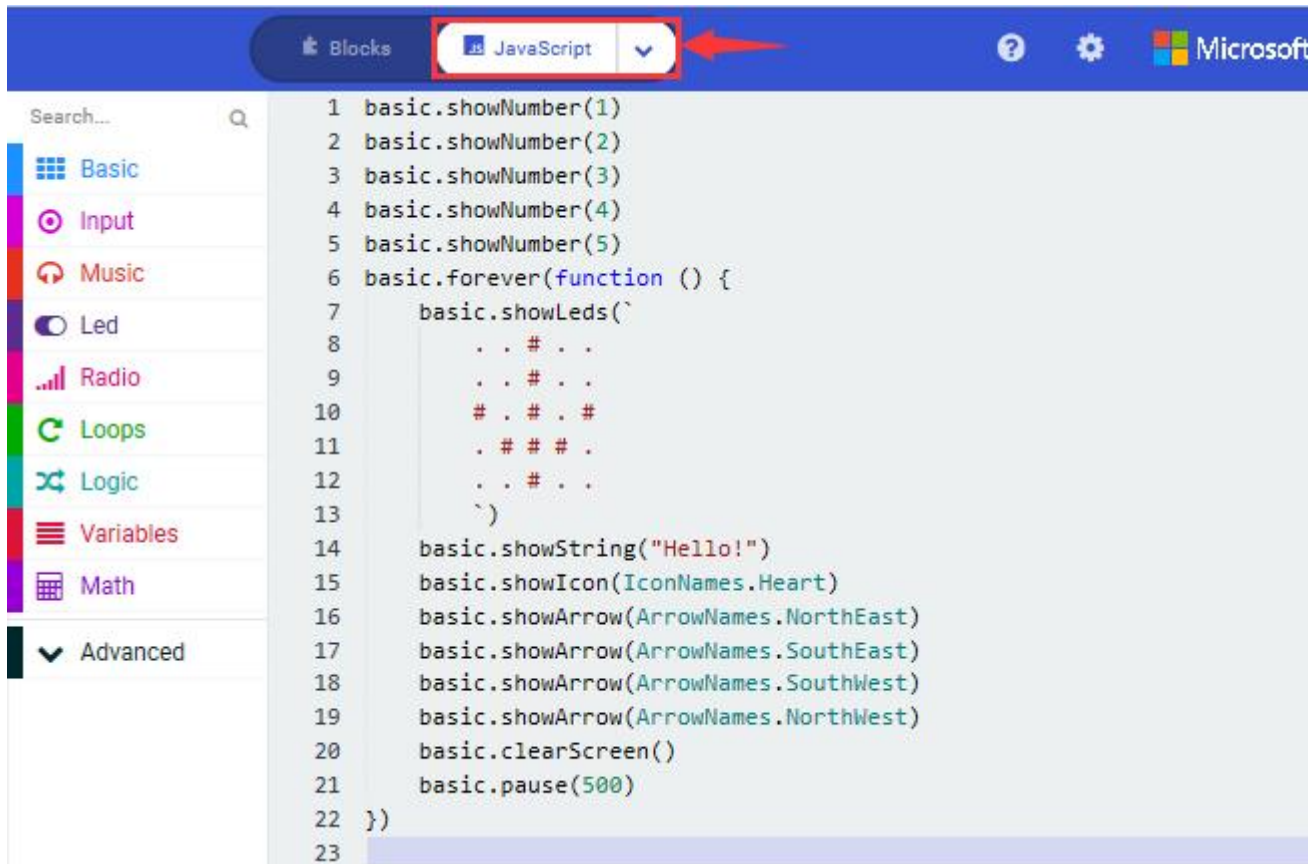
The “South West” arrow appears up on LED dot matrix

The “North West” arrow is displayed on LED dot matrix

Clear the screen

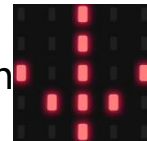
Delay in 500ms

Select “JavaScript” and “Python” to switch into JavaScript and Python language code:

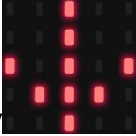


4. Test Result

Upload code 1 and power the board , we will see the icon

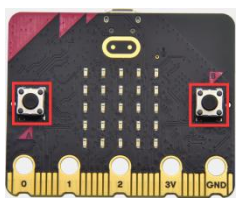


Upload code 2 and plug micro:bit in power, Micro: bit starts showing number 1, 2, 3, 4, and 5, then cyclically

display , "Hello!" , , , ,  and  patterns.

(How to download? How to quick download?)

Project 4: Programmable Buttons



1. Description

Buttons can be used to control circuits. In an integrated circuit with a push button, the circuit is connected when pressing the button and but after release, it will break again.

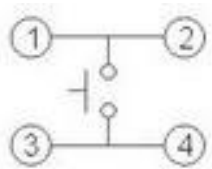
Both ends of the button like two mountains. There is a river in between.

The internal metal piece connect the two sides to let the current pass, just like building a bridge to connect two mountains.



The internal structure of the button is shown as follows: before pressing the button, 1 ,2 , 3 and 4 are turned on.

However, 1, 3 or 1, 4 or 2, 3 or 2 and 4 are disconnected, which is only enabled when the button is pressed.



Micro: Bit main board boasts three push buttons, two are programmable buttons(marked with A and B), and the one on the other side is a reset button. By pressing the two programmable buttons can input three different signals. We can press button A or B alone or press them together and the LED dot matrix shows A,B and AB respectively. Let’ s get started.

2. Preparation

- A. Attach the micro:bit main board to your computer via the USB cable
- B. Open the Web version of Makecode.

3. Test Code

Test Code 1:

Press buttons on micro:bit, micro:bit will display character strings.

The route to get test code ([How to load?](#))

File Type	Path	File Name
Hex file	..\2. Makecode Tutorial\Makecode Code	Project 4 : Programmable Buttons-1.hex

Or you could edit code step by step in the editing area.

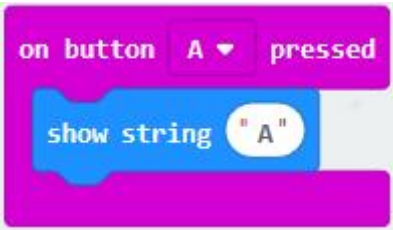
Delete “on start” and “forever” firstly, then click “Input” → “on button A pressed”

A. Click “Basic” → “show string” ;

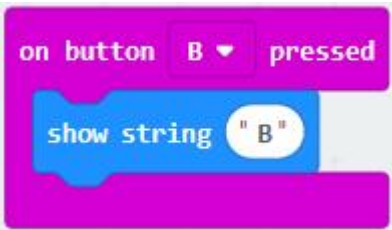
B. Then place it into “on button A pressed” block, change “Hello!” into “A” .



Copy code string once, tap the drop-down button “A” to select “B” and modify character “A”



into “B” .





Copy once, and pull down the triangle button to select "A+B" and "show string "AB"



Complete Code:

Press button A on Micro: bit main board
Show the character "A"

Press button B on Micro: bit main board
Show the character "B"

Press button A and B at same time
Display the character "AB"

Select "JavaScript" and "Python" to switch into JavaScript and Python language code:



Code 2:

The route to get test code (How to load?)

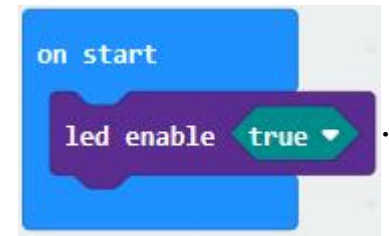
File Type	Path	File Name
Hex file	..\2. Makecode Tutorial\Makecode Code	Project 4: Programmable Buttons-2.hex

keyestudio

Or you could edit code step by step in the editing area.

A. Click "Led" → "more" → "led enable false" ,

B. Put it into the block "on start" , click drop-down triangle button to select "true"



A. Tap "Variables" → "Make a Variable..." → "New variable name: "

B. Enter "item" in the dialog box and click "OK" , then variable "item" is produced. And move "set item to 0" into "on start" block.



A. Click "Input" → "on button A pressed" .

B. Go to "Variables" → "change item by 1"

C. Place it into "on button A pressed" and 1 is modified into 5.



Duplicate



-5.



A. Enter "Led" → "plot bar graph of 0 up to 0"

B. Keep it into "forever" block

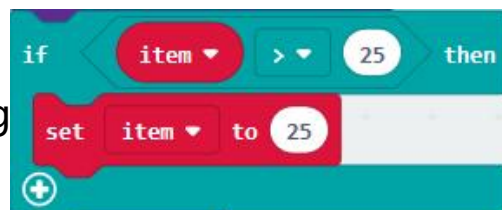
C. Go to "Variables" to move "item" into 0 box, change 0 into 25.



- A. Go to "Logic" to move out "if...true...then..." and "=" blocks,
- B. Keep "=" into "true" box and set to ">"
- C. Select "item" in the "Variables" and lay it down at left box of ">" , change 0 into 25;
- D. Enter "Variables" to drag "set item to 0" block into "if...true..then..." , alter 0 into 25.

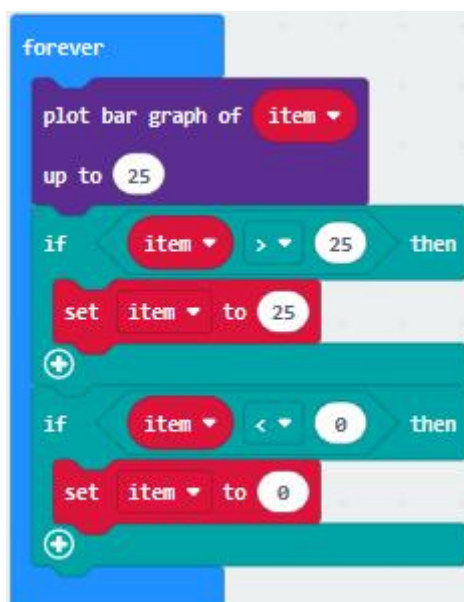
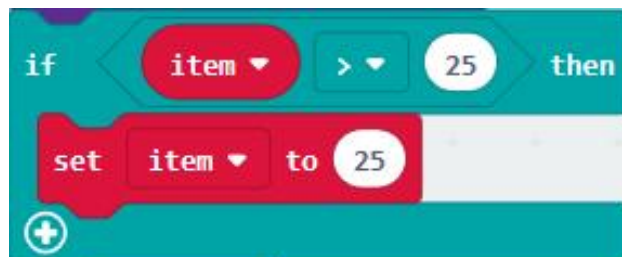


- (7) A. Replicate code string

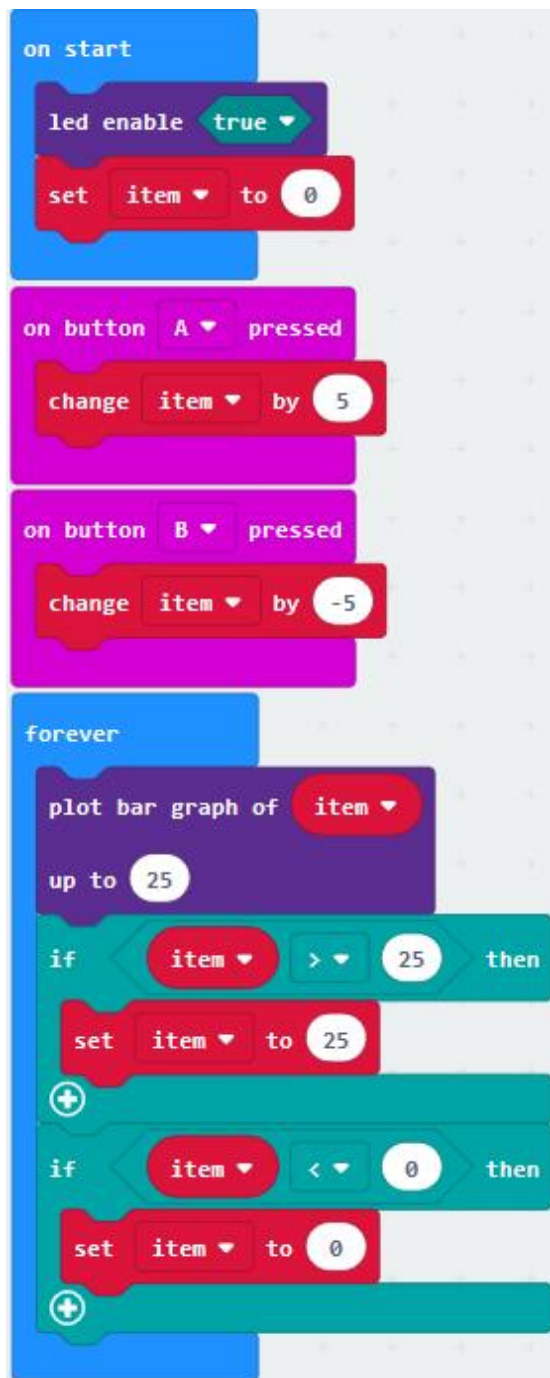


- B. ">" is modified into "<" and 25 is changed into 0,

- C. Leave it beneath



Complete Program:



"on start": command block runs once to start program.

Turn on LED dot matrix

Set the initial value of item to 0

Press button A on Micro:bit board

Change item by 5

Press button B on Micro:bit board

Change item by -5

The program under the block "forever" runs cyclically. Light on LED in dot matrix to draw bar graph, light up up to 25 LEDs

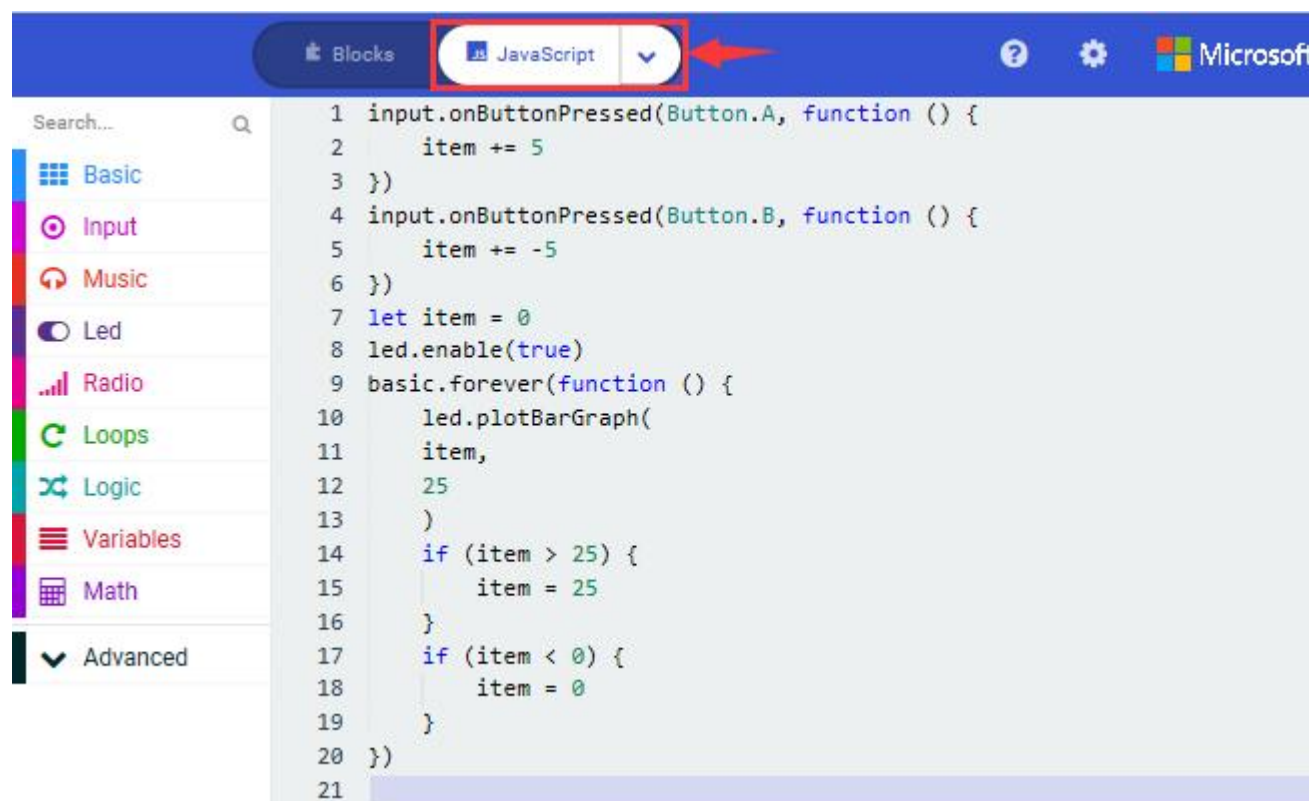
If item is greater than 25

Then set item to 25

If item is less than 0

Then set item to 0

Click "JavaScript" to switch into JavaScript code:



4. Test Result

After uploading test code 1 to micro:bit main board V2 and powering the board via the USB cable, the 5*5 LED dot matrix shows A if button A is pressed, B if button B pressed, and AB if button A and B are pressed together.

After uploading test code 2 to micro:bit main board V2 and powering the board via the USB cable, when pressing the button A , the LEDs turning red increase by 5 while when pressing the button B the LEDs turning red reduce.
(How to download? How to quick download?)

Project 5: Temperature Measurement

1. Description

The Micro:bit main board is not equipped with a temperature sensor, but uses the built-in temperature sensor in NFR52833 chip for temperature detection. Therefore, the detected temperature is more closer to the temperature of the chip, and there maybe deviation from the ambient temperature.

In this project, we will seek to use the sensor to test the temperature in the current environment, and display the test results in the display data (device). And then control the LED dot matrix to display different patterns by setting the temperature range detected by the sensor.

Note: the temperature sensor of Micro:bit main board is shown below:



2. Preparation

- A. Attach the micro:bit main board to your computer via the USB cable
- B. Open the Web version of Makecode.

3. Test Code

Test Code 1:

Micro:bit detects temperature

The route to get test code ([How to load?](#))

File Type	Path	File Name
Hex	..\2. Makecode	Project 5 : Temperature

file	Tutorial\Makecode Code	Measurement-1.hex
------	------------------------	-------------------

Or you could edit code step by step in the editing area.

Go to "Advanced" → "Serial" → "serial redirect to USB"

Place it into "on start"



Click "Serial" to drag out "serial write value x=0"

Move it into "forever" block



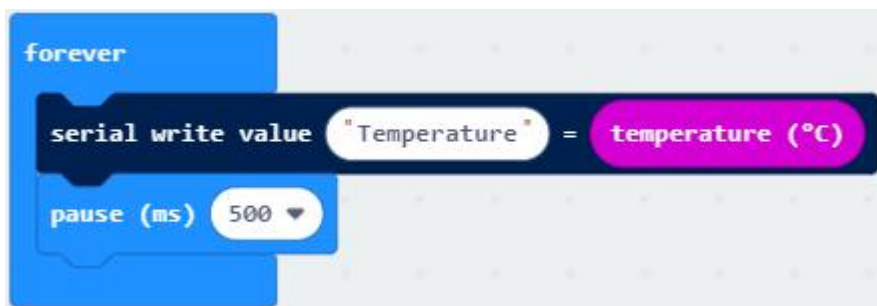
Go to "Input" → "temperature(°C)"

Place it into 0 box

Change x into Temperature



Move "pause (ms) 100" from "Basic" block and place it in the "forever" block, and set the delay to 500ms.



Complete Program:



“on start” : command block runs once to start program.

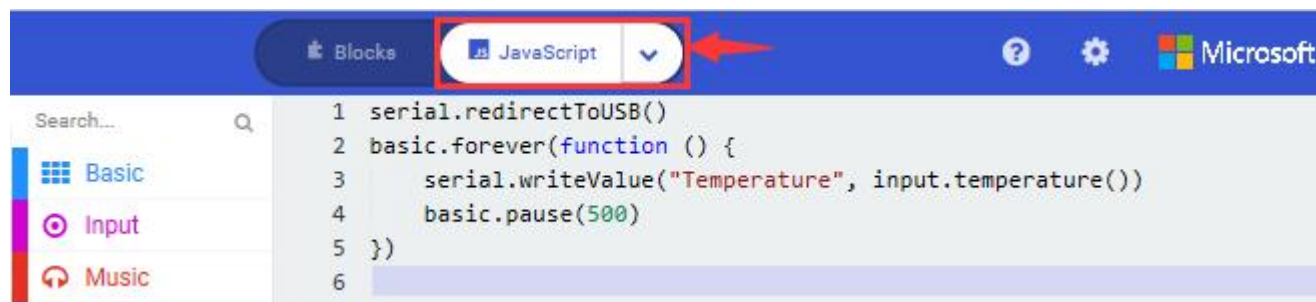
Serial redirect to USB

The program under the block “forever” runs cyclically.

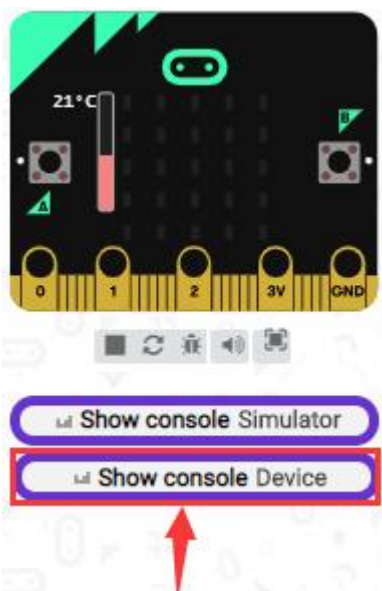
Serial writes Temperature

Delay in 500ms

Click “JavaScript” to view the corresponding JavaScript code:

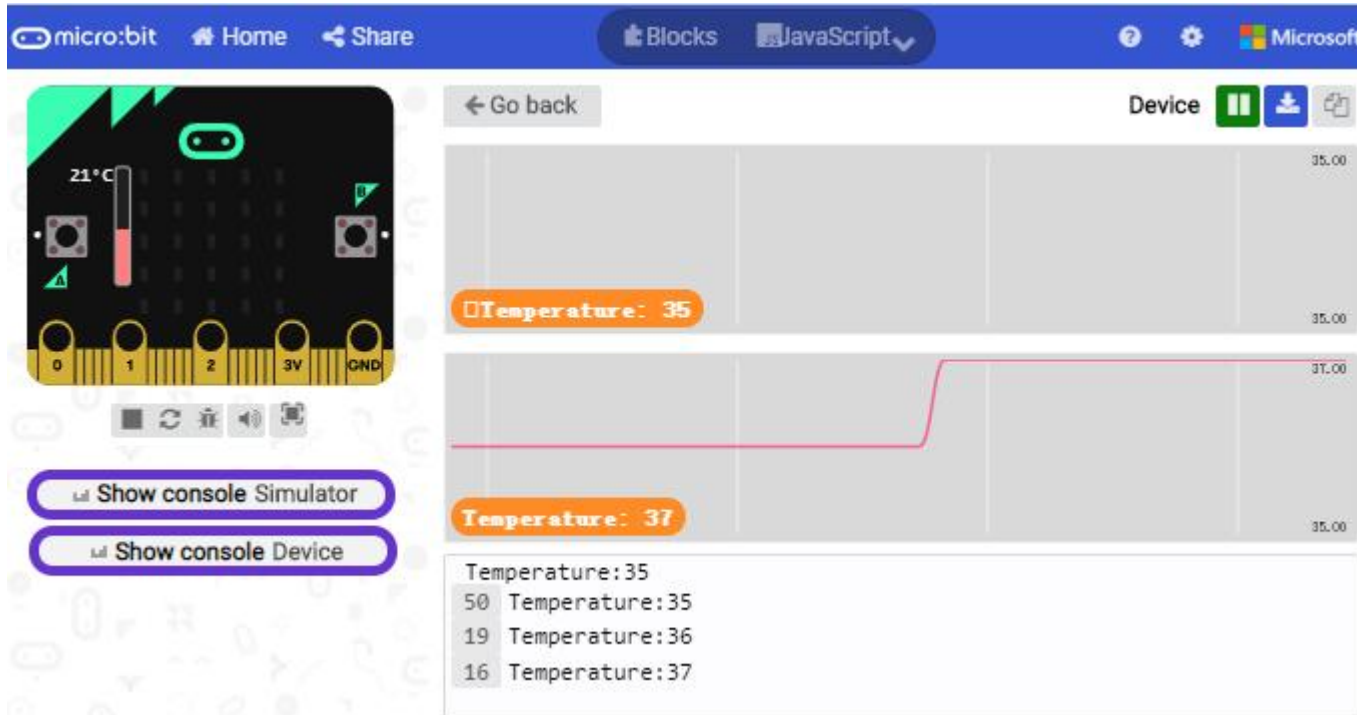


Download code 1 to micro:bit board and keep USB cable connected, then tap the button :
(How to quick download?)



Temperature data is shown below:

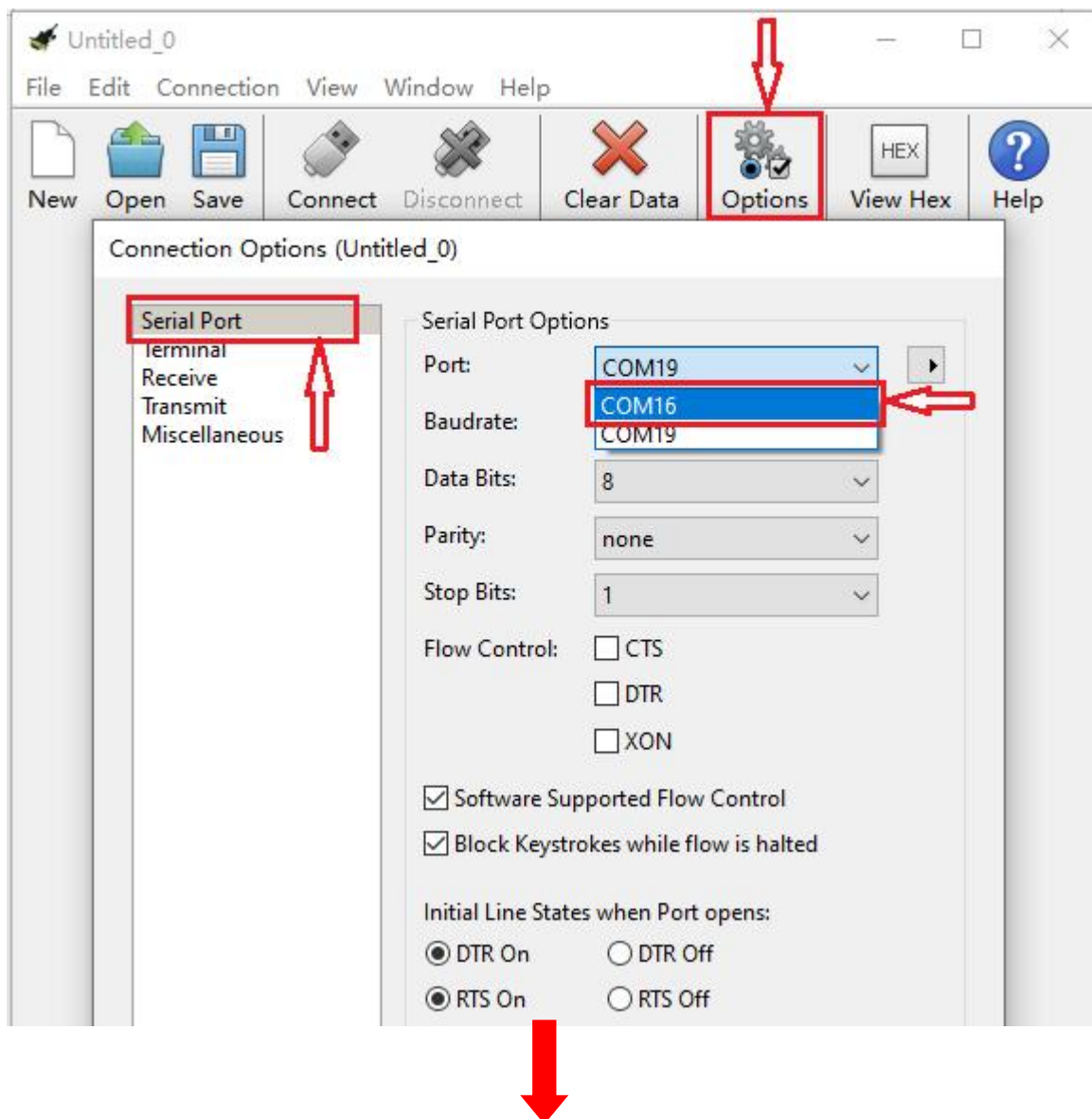
keystudio

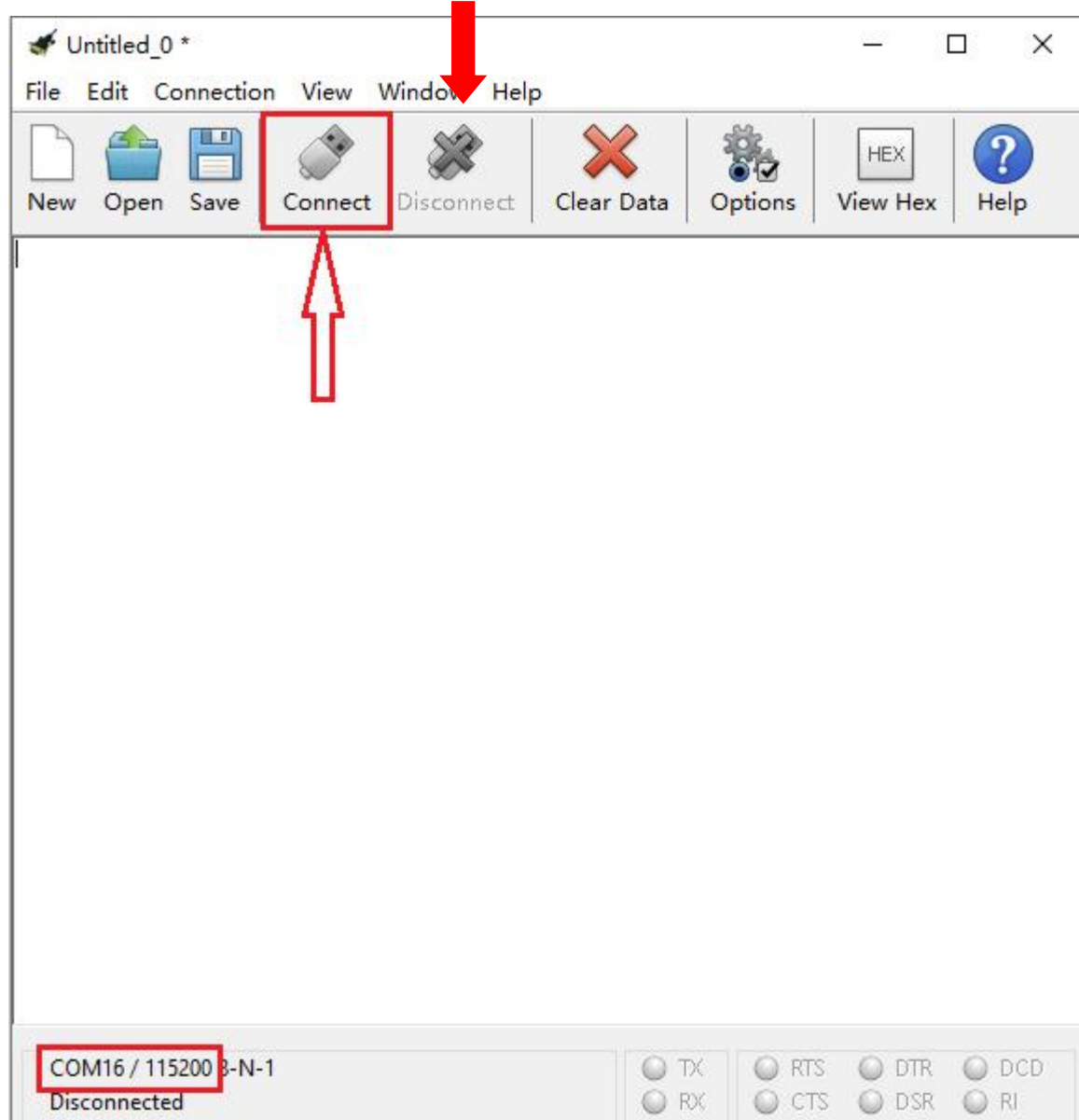
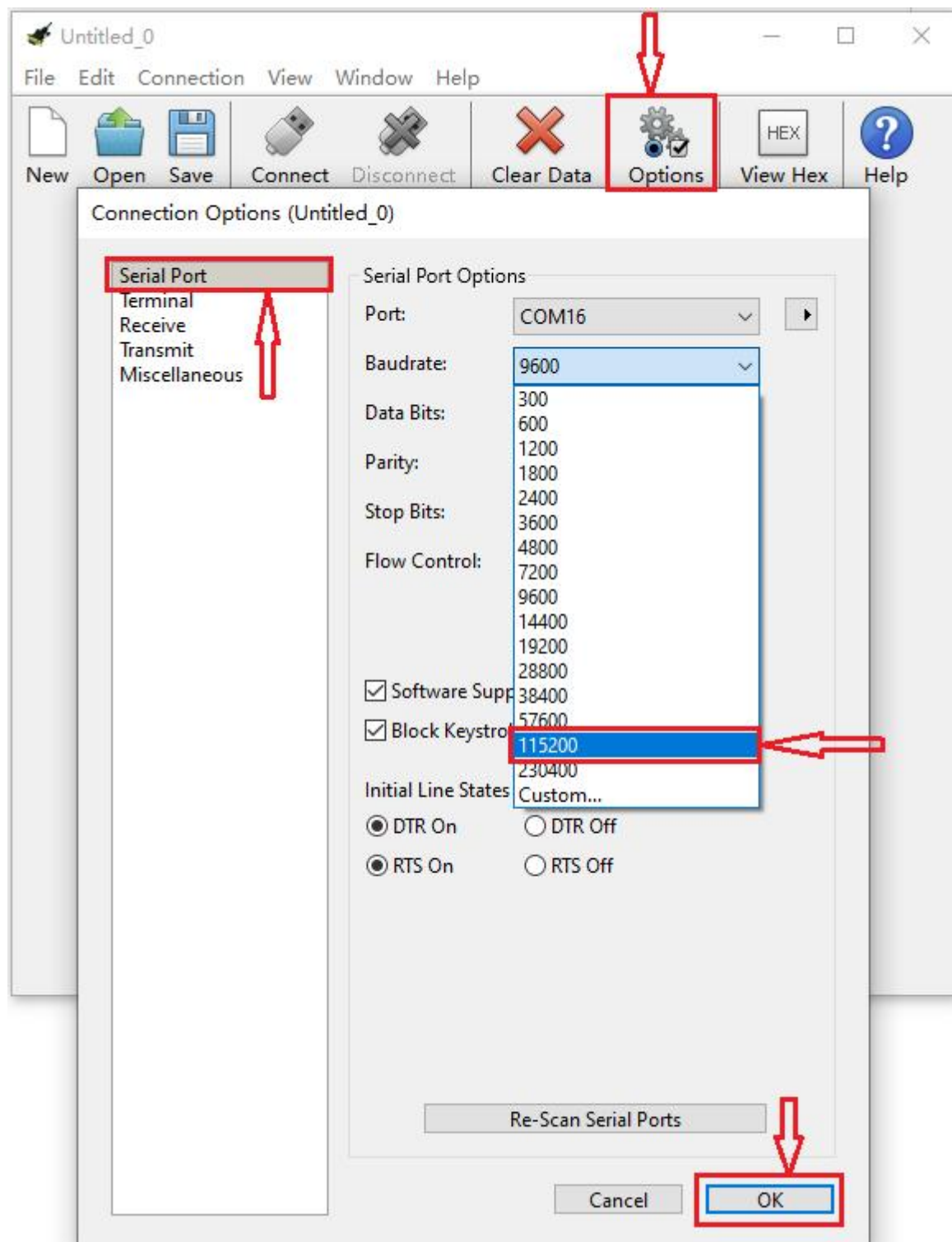


Through the test, the room temperature is 35 °C when touching the NFR51822 chip of micro:bit; however, the temperature rises to 37°C when it touches water cup.

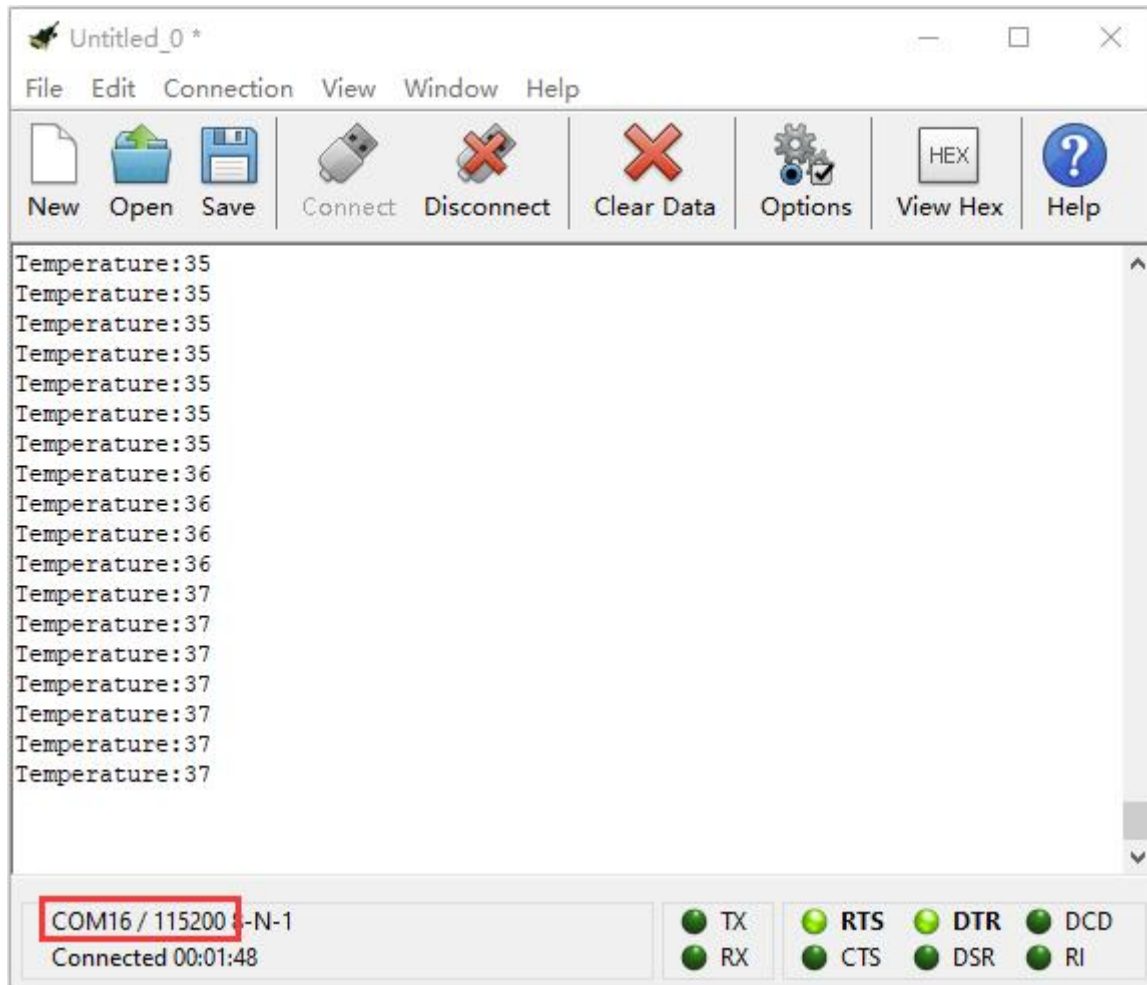
Open CoolTerm, click Options to select SerialPort. Set COM port and 115200 baud rate(the baud rate of USB serial communication of Micro:bit is 115200 through the test). Click "OK" and "Connect" .

The serial monitor shows the current ambient temperature value, as shown below:





keyestudio



Code 2:

Micro:bit display different pictures by temperature(the temperature value in the code could be adjusted).

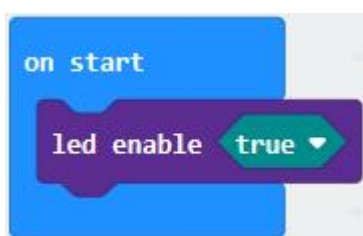
The route to get test code ([How to load?](#))

File Type	Path	File Name
Hex file	..\2. Makecode Tutorial\Makecode Code	Project 5: Temperature Measurement-2.hex

Or you could edit code step by step in the editing area.

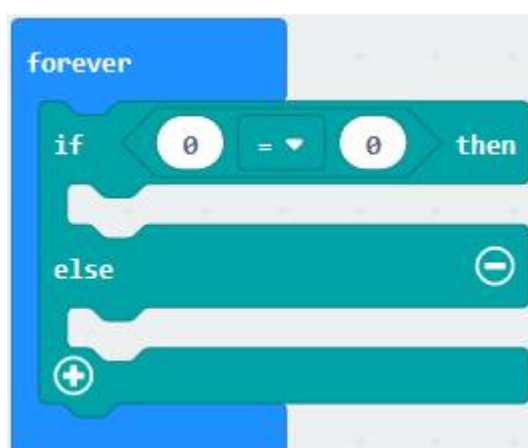
You could set temperature based on real situation.

Click "Led" → "more" → "led enable false" into "on start" , click drop-down triangle button to select "true"



A. Go to "Logic" → "if..true...then...else" and "=" block;

B. Move "if..true...then...else" into "forever" block, then place "=" into "true" box.



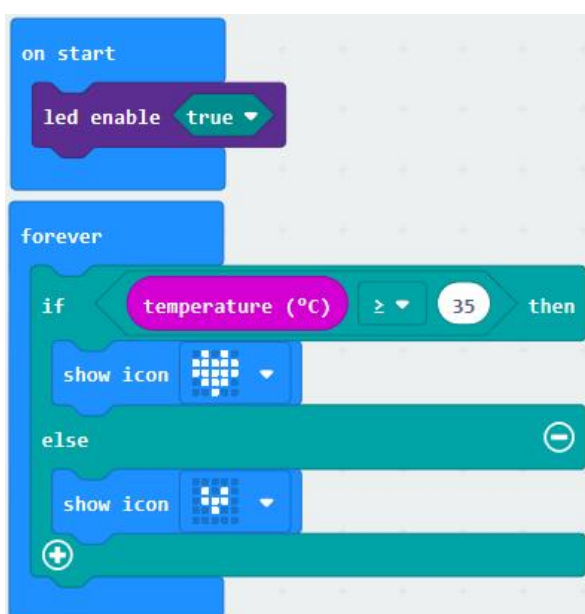
- Change "=" into "≥"
- Go to "Input" → "temperature(°C)"
- Change 0 into 35.



Tap "Basic" → "show icon" , copy it once and lay down them under the "if ...then" and else blocks, then click the drop-down triangle button to select " " .



Complete Program:



"on start" : command block runs once to start program.

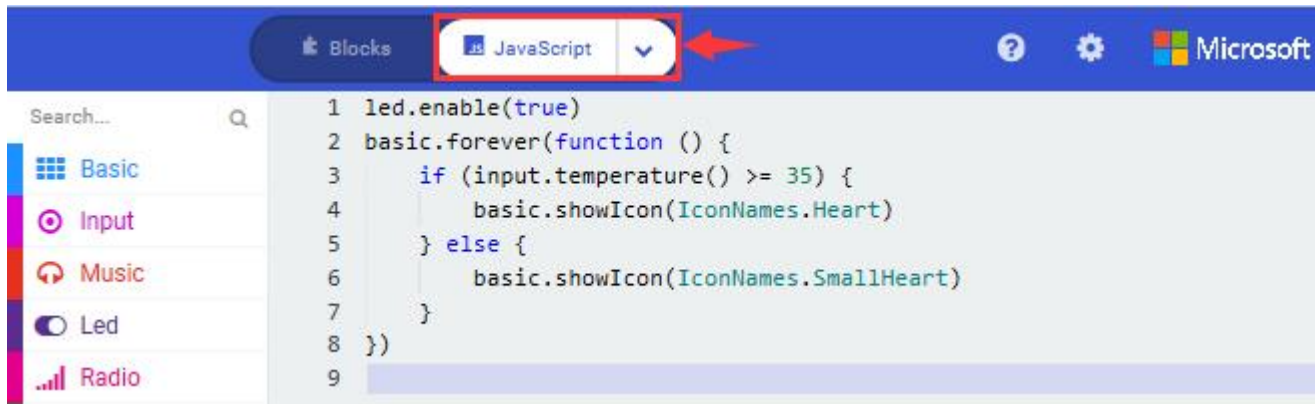
Turn on LED dot matrix

Under the block "forever" , program runs cyclically.

If the detected temperature $\geq 35^{\circ}$, the next program will be executed.

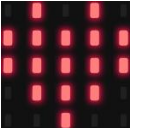
Dot matrix shows " "

Click "JavaScript", the corresponding JavaScript code is shown below:



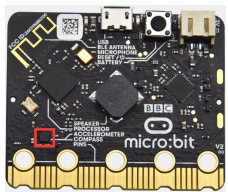
4. Test Result

Upload the Code 1 and plug in power. And the 5*5 LED displays the ambient temperature. When pressing the temperature sensor, the temperature will grow on dot matrix. When the ambient temperature is less than 35°C, the

5*5LED will show . When the temperature is equivalent to or greater than 35°C, the pattern  will appear.

(How to download? [How to quick download?](#))

Project 6: Geomagnetic Sensor



1. Description

This project mainly introduces the use of the micro:bit's geomagnetic sensor. In addition to detecting the strength of the magnetic field, it can also be used to determine the direction, which is an important part of the heading and attitude reference system (AHRS) as well.

It uses FreescaleMAG3110 three-axis magnetometer. Its I2C interface communicates with the outside, and the range is $\pm 1000\mu\text{T}$, the maximum data update rate is 80Hz. Combined with an accelerometer, it can calculate the position. Additionally, it is applied to magnetic detection and compass blocks.

Then we could read the value detected by it to determine the location. We need to calibrate the micro:bit board when the magnetic sensor works. The correct calibration method is to rotate the micro:bit board.

In addition, the objects nearby may affect the accuracy of readings and calibration.

2. Preparation

A. Attach the micro:bit main board to your computer via the USB cable

B. Open the Web version of Makecode.

3. Test Code

Test Code 1:

Press A on micro:bit, the value of compass is shown.

The route to get test code ([How to load?](#))

File Type	Path	File Name
Hex file	..\2. Makecode Tutorial\Makecode Code	Project 6: Geomagnetic Sensor-1.hex

Or you could edit code step by step in the editing area.

A. Click "Input" → "more" → "calibrate compass"

B. Lay down it into block "on start" .



A. Go to "Input" → "on button A pressed" .

B. Enter "Basic" → "show number" , put it into "on button A pressed" block;

C. Tap "Input" → "compass heading(°)" , and place it into "show number"



Complete Program:

① "on start" : command block only runs once to start program.

② Calibrate compass

③ Press button A on Micro:bit main board

④ Dot matrix shows the direction of compass heading

Select "JavaScript" and "Python" to switch into JavaScript and Python language code:



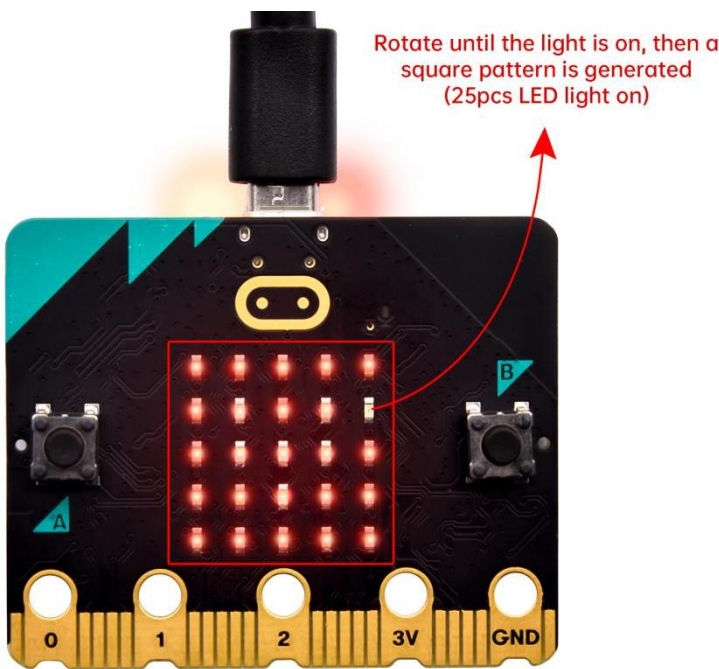
Code Description:

Upload the code 1, plug in micro:bit via an USB cable.

As the button A is pressed, LED dot matrix indicates the “TILT TO FILL SCREEN” then enter the calibration interface.

The calibration method: rotate the micro:bit to make the LED dot matrix draw a square (25 LEDs are on), as shown in the following figure:

(How to download? How to quick download?)



The calibration will be finished until you view the smile pattern  appear.

The serial monitor will show 0°, 90°, 180° and 270° when pressing A.

Code 2:

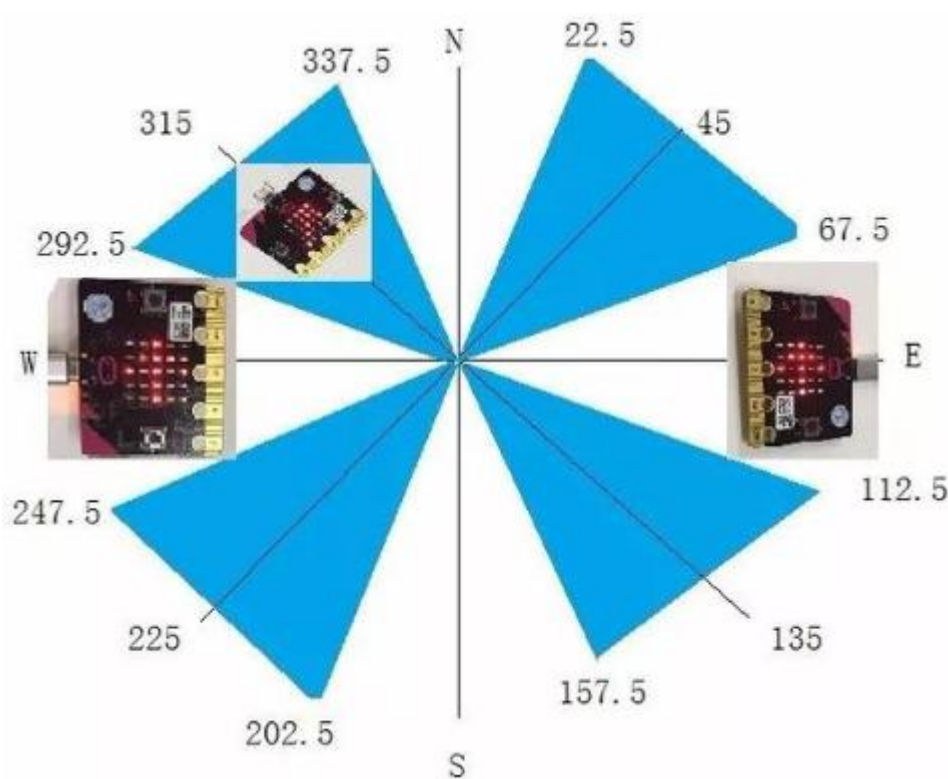
Make micro: bit board point to the north, south, east and west horizontally , LED dot matrix displays the corresponding direction patterns

The route to get test code ([How to load?](#))

File Type	Path	File Name
Hex file	..\2. Makecode Tutorial\Makecode Code	Project 6: Geomagnetic Sensor-2.hex



This module can keep reading data to determine direction, and let arrow point to the current magnetic North Pole.

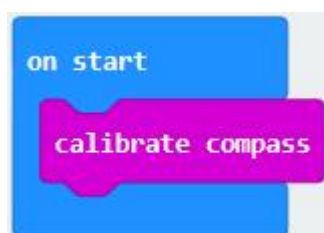


For the above picture, the arrow will point to the upper right when the value ranges from 292.5 to 337.5. Because 0.5 can't be input in the code, the values we get are 293 and 338.

Link computer with micro:bit board by a micro USB cable, and program in MakeCode editor:

Enter "Input" → "more" → "calibrate compass"

Move "calibrate compass" into "on start"



- Click "Variables" → "Make a Variable..." → "New variable name: "
- Input "x" in the blank box and click "OK" , and the variable "x" is generated.
- Drag out "set x to" into "forever" block



A. Go to "Input" → "compass heading(°C)" , and keep it into "0" box



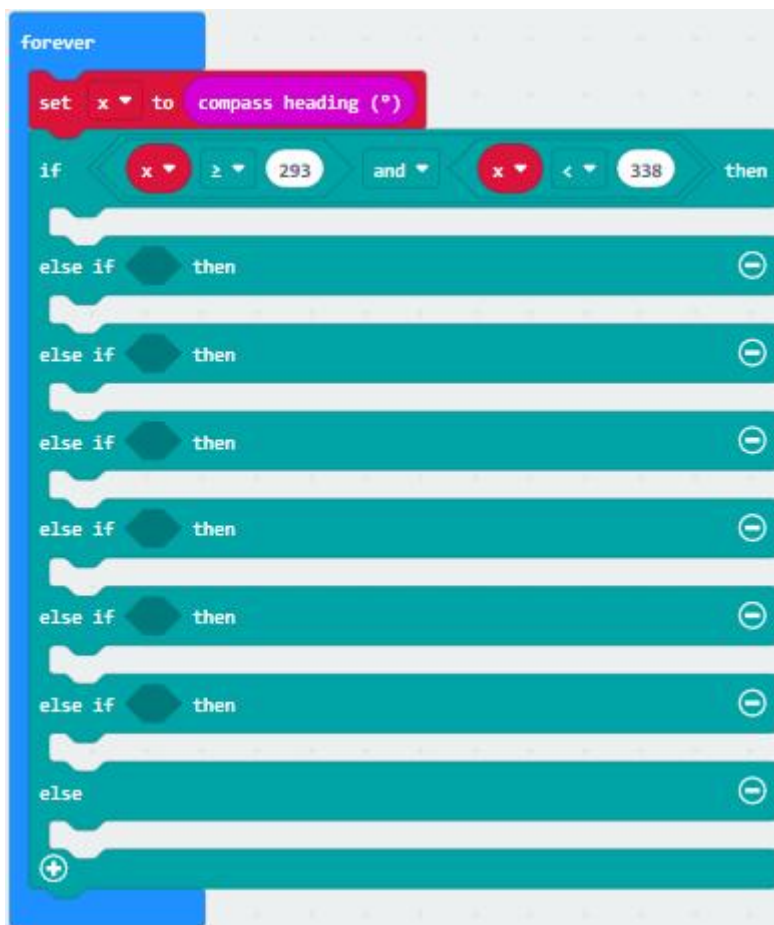
Tap "Logic" → "if...then...else" , and place it into "forever" block, then click  icon for 6 times.

A. Place "and" into "true" block

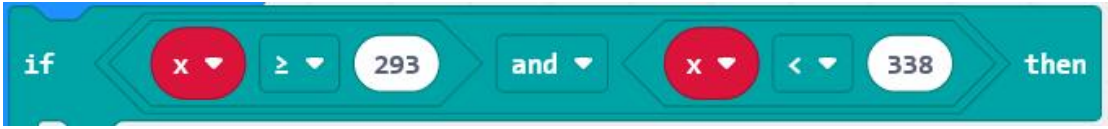
B. Then move "=" block to the left box of "and"

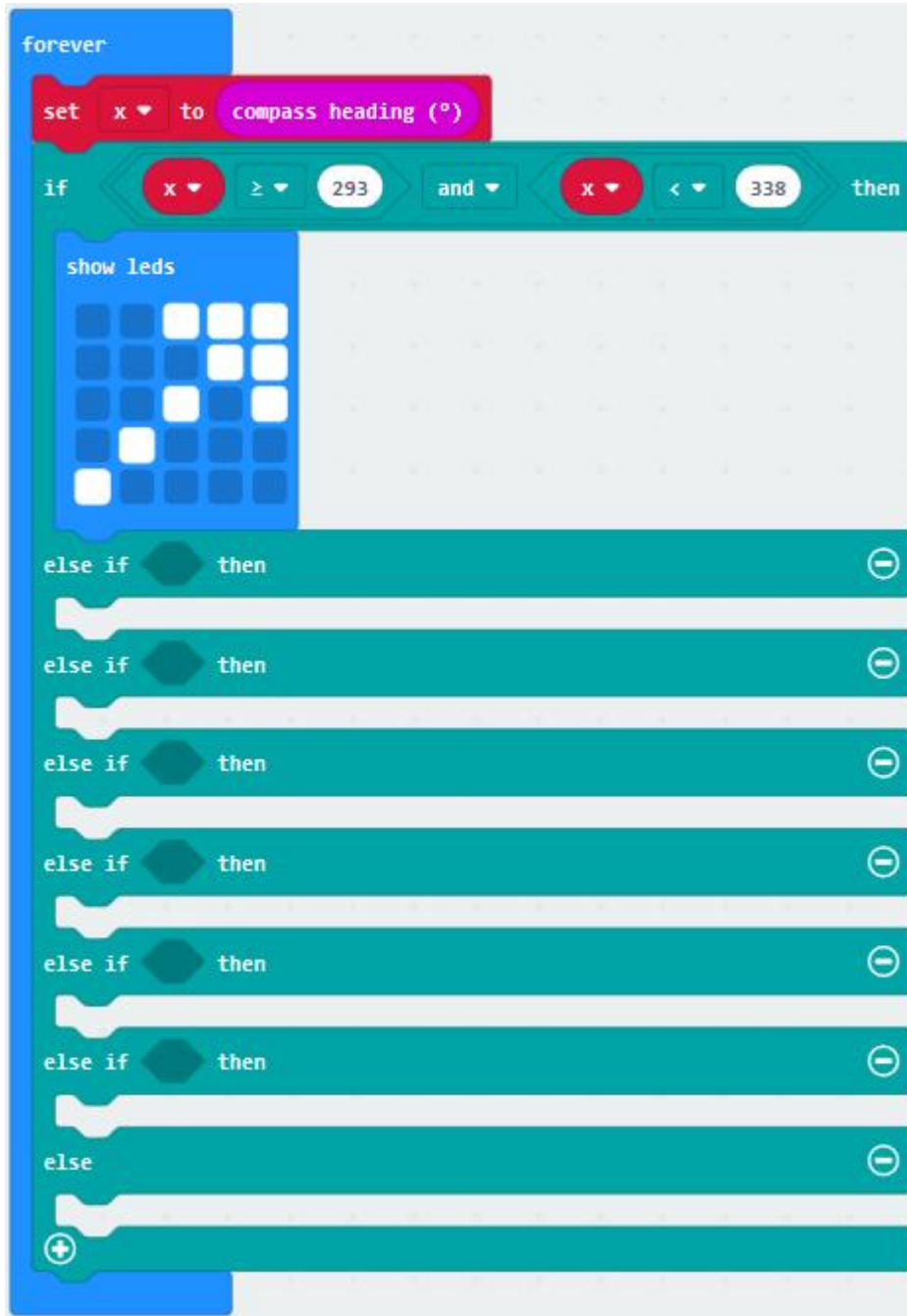
C. Click "Variables" to drag "x" to the left "0" box, change 0 into 293 and set to "≥" ;

D. Then copy "x≥293" once and leave it to the right "0" box and set to "x<338"



A. Go to "Basic" → "show leds"

B. Lay it down beneath  block, then click "show leds" and the pattern  appears.



A. Duplicate  for 6 times.

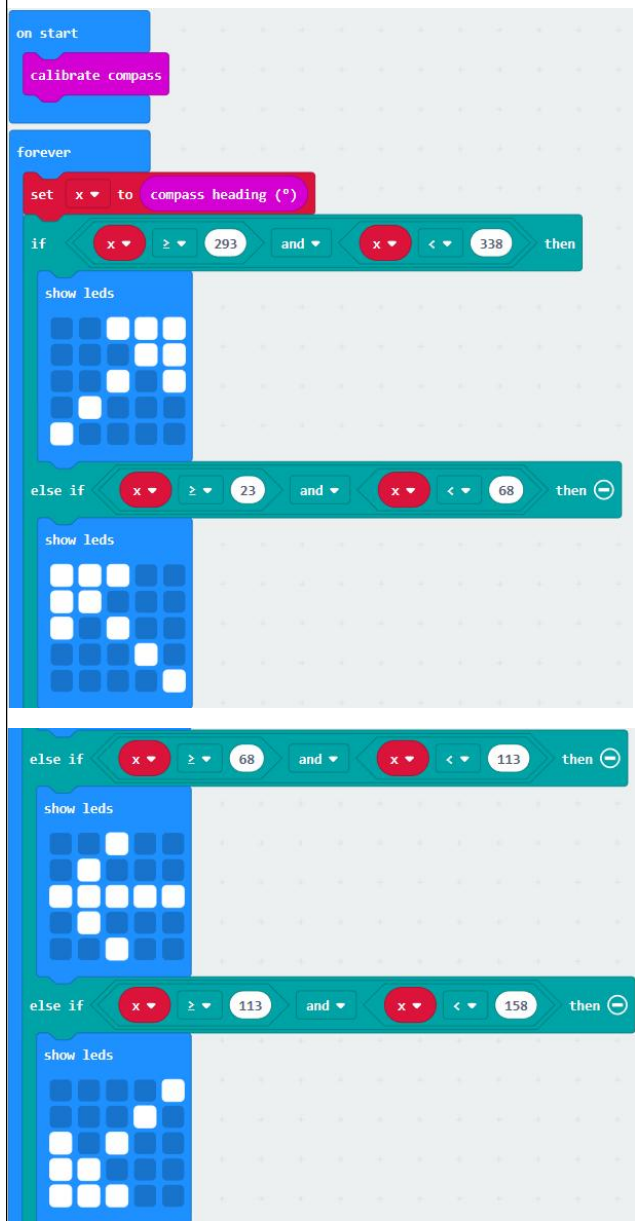
B. Separately leave them into the blank boxes behind "else if" .

C. Set to "x ≥ 23 and x < 68" , "x ≥ 68 and x < 113 " , "x ≥ 113 and x < 158 " , "x ≥ 158 and x < 203 " , "x ≥ 203 and x < 248 " , "x ≥ 248 and x < 293 " respectively.

D. Then copy "show leds" for 7 times and keep them below the "else if.....then" block respectively.

E. Click the blue boxes to form the pattern  ,  ,  ,  ,  ,  and  .

Complete Program:



“on start”: command block only runs once to start program.
Calibrate compass
The program under the block “forever” runs cyclically.

Store the angle of the compass heading into the variable x

When $293 \leq x < 338$, the next program will be executed

 appears on the dot matrix

When $23 \leq x < 68$, the next program will be executed

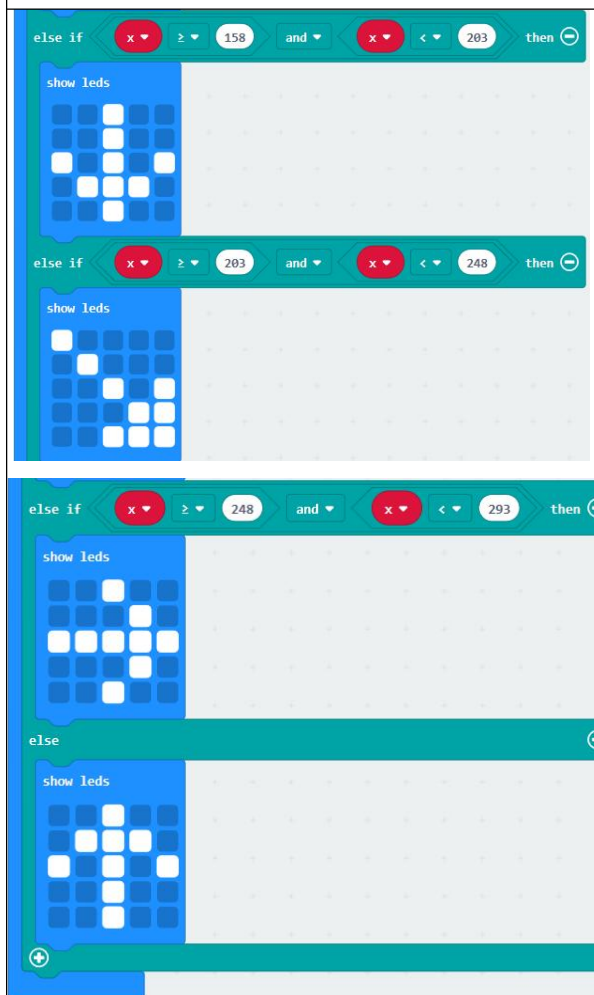
 is displayed on dot matrix

When $68 \leq x < 113$, the next program will be executed

 is shown on dot matrix

When $113 \leq x < 158$, the next program will be executed

 pattern appears



When $158 \leq x < 203$, the next program will be executed.

Dot matrix shows 

When $203 \leq x < 248$, the next program will be executed.

Dot matrix displays 

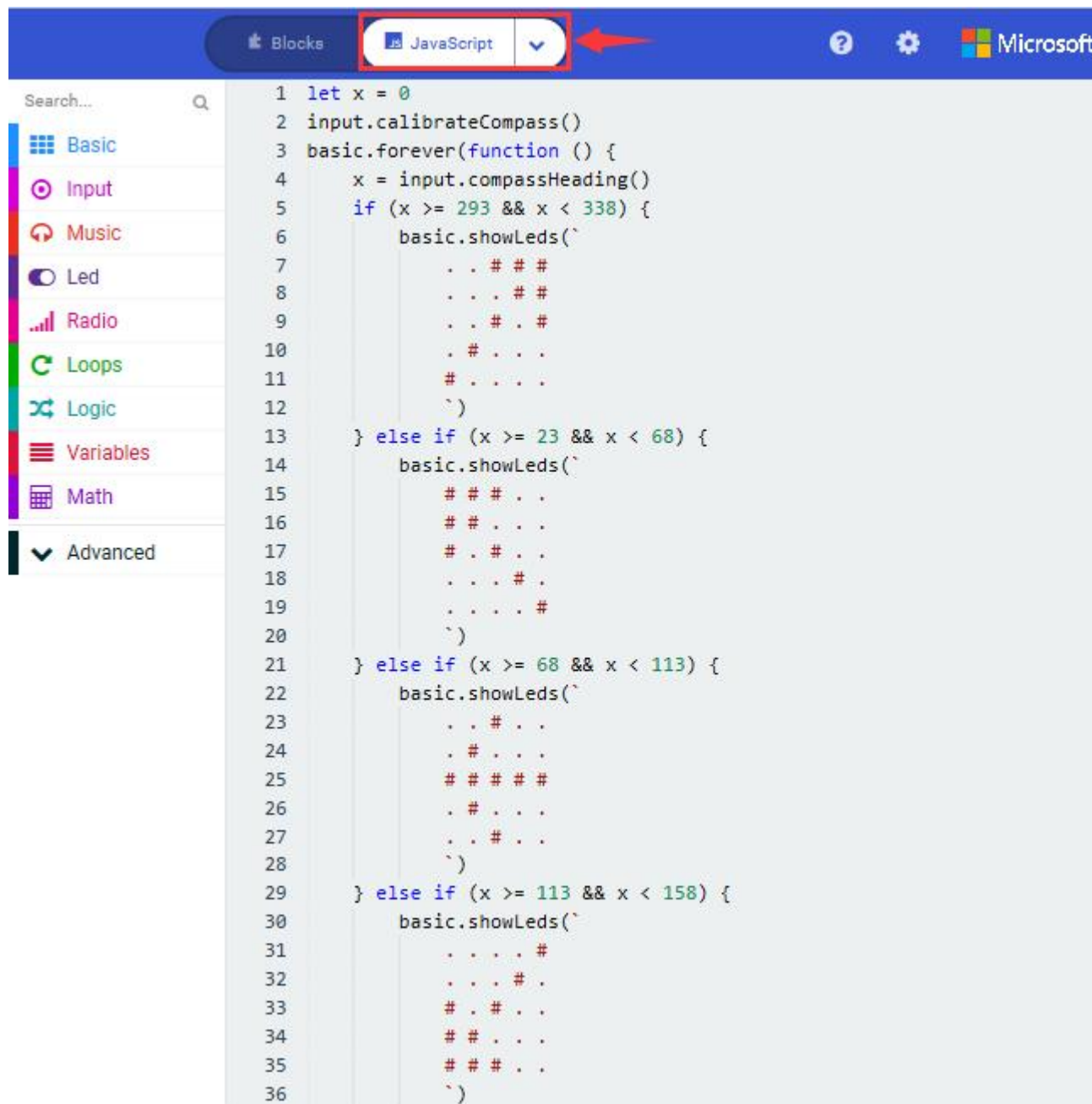
When $248 \leq x < 293$, the next program will be executed.

Dot matrix shows 

When x is not among the above rang, the next program will be executed under else block

keystudio

Select "JavaScript" and "Python" to switch into JavaScript and Python language code:



```
1 let x = 0
2 input.calibrateCompass()
3 basic.forever(function () {
4   x = input.compassHeading()
5   if (x >= 293 && x < 338) {
6     basic.showLeds(`
7       . . # # #
8       . . . # #
9       . . # . #
10      . # . . .
11      # . . . .
12      `)
13   } else if (x >= 23 && x < 68) {
14     basic.showLeds(`
15       # # # . .
16       # # . . .
17       # . # . .
18       . . . # .
19       . . . . #
20       `)
21   } else if (x >= 68 && x < 113) {
22     basic.showLeds(`
23       . . # . .
24       . # . . .
25       # # # # #
26       . # . . .
27       . . # . .
28       `)
29   } else if (x >= 113 && x < 158) {
30     basic.showLeds(`
31       . . . . #
32       . . . # .
33       # . # . .
34       # # . . .
35       # # # . .
36       `)
```

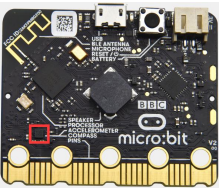
```
37   } else if (x >= 158 && x < 203) {
38     basic.showLeds(`
39       . . # . .
40       . . # . .
41       # . # . #
42       . # # # .
43       . . # . .
44       `)
45   } else if (x >= 203 && x < 248) {
46     basic.showLeds(`
47       # . . . .
48       . # . . .
49       . . # . #
50       . . . # #
51       . . # # #
52       `)
53   } else if (x >= 248 && x < 293) {
54     basic.showLeds(`
55       . . # . .
56       . . . # .
57       # # # # #
58       . . . # .
59       . . # . .
60       `)
61   } else {
62     basic.showLeds(`
63       . . # . .
64       . # # # .
65       # . # . #
66       . . # . .
67       . . # . .
68       `)
69   }
70 })
71
```

4. Test Result

Upload code 2 and plug micro:bit into power. After calibration, tilt the micro:bit board, and the LED dot matrix displays the direction signs.

(How to download? How to quick download?)

Project 7: Accelerometer



1. Description

The micro:bit board has a built-in Freescale MMA8653FC three-axis acceleration sensor (accelerometer). Its I2C interface works on external communication, the range can be set to $\pm 2g$, $\pm 4g$, and $\pm 8g$, and the maximum data update rate can reach 800Hz.

When the Micro:bit is stationary or moving at a constant speed, the accelerometer only detects the gravitational acceleration; when the Micro:bit is slightly shaken, the acceleration detected is much smaller than the gravitational acceleration and can be ignored. Therefore, in the process of using Micro:bit, the main purpose is to detect the changes of the gravitational acceleration on the x, y, and z axes when the attitude changes.

For this project, we will introduce the detection of several special postures by the accelerometer.

2. Preparation

- A. Attach the micro:bit main board to your computer via the USB cable
- B. Open the Web version of Makecode.

3. Test Code

Test Code 1:

The route to get test code ([How to load?](#))

File Type	Path	File Name
Hex file	..\2. Makecode Tutorial\Makecode Code	Project 7 : Accelerometer-1.hex

Or you could edit code step by step in the editing area.

- (1) A. Enter "Input" → "on shake" ,

B. Click “Basic” → “show number” , place it into “on shake” block, then change 0 into 1.



(2) A. Copy code string  for 7 times, and separately click the triangle button to select “logo up” ,

“logo down” , “screen up” , “screen down” , “tilt left” , “tilt right” and “free fall” , then respectively change 1 into 2, 3, 4, 5, 6, 7, 8.

Complete Program:



Shake the Micro:bit board
LED dot matrix displays 1
The log is up
LED dot matrix displays 2
The logo is down
LED dot matrix displays 3
The screen is up
LED dot matrix displays 4



on screen down ▾
show number 5

on tilt left ▾
show number 6

on tilt right ▾
show number 7

on free fall ▾
show number 8

The screen is down

Number 5 is shown

The Micro:bit board is tilt to the left

Number 6 is displayed

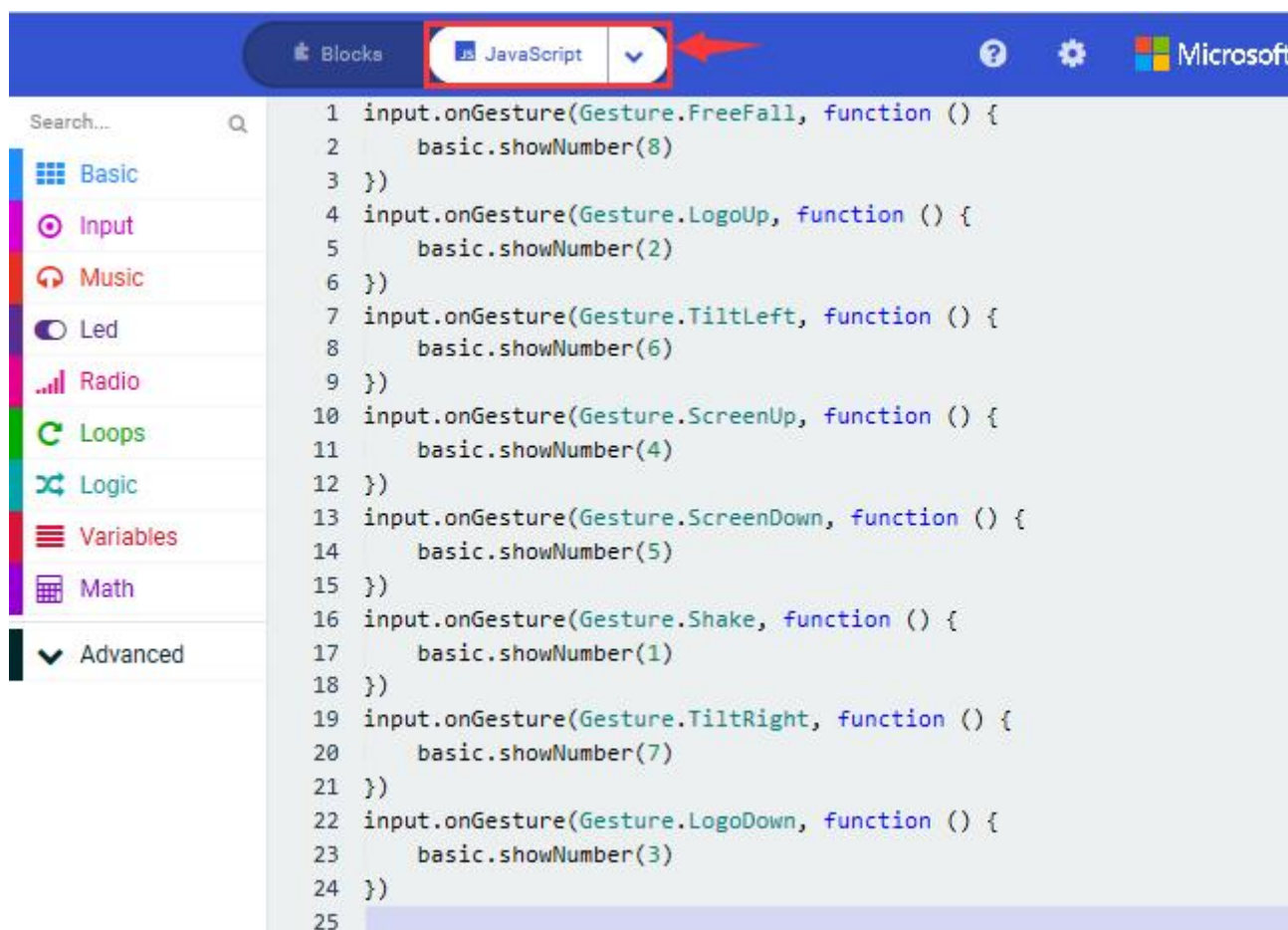
The Micro:bit board is tilt to the right

Number7 is displayed

When the Micro:bit board is free fall

LED dot matrix shows 8

Click "JavaScript", you will view the corresponding JavaScript code:



Code 2:

Detect the value of acceleration speed at x, y and z axis

The route to get test code ([How to load?](#))

File Type	Path	File Name

Hex file	..\2. Makecode Tutorial\Makecode Code	Project 7: Accelerometer-2.hex
----------	--	-----------------------------------

Or you could edit code step by step in the editing area.

A. Go to "Advanced" → "Serial" → "serial redirect to USB"

B. Drag it into "on start"



A. Enter "Serial" → "serial write value x =0"

B. Leave it into "forever" block



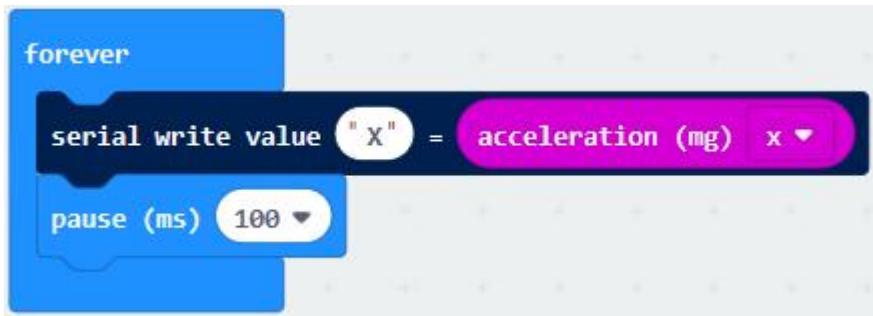
.....

A. Click "Input" → "acceleration(mg) x" ;

B. Keep it into "0" box and capitalize the "x"



Go to "Basic" and place "pause (ms) 100" into "forever" , then set the delay to 100ms.

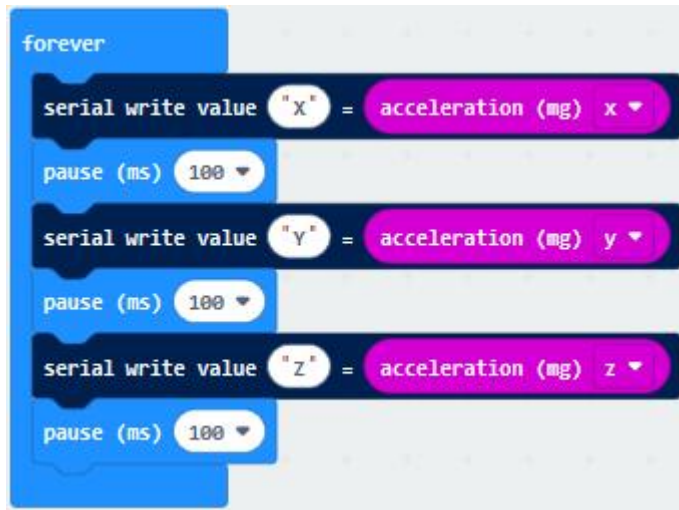


Replicate code string

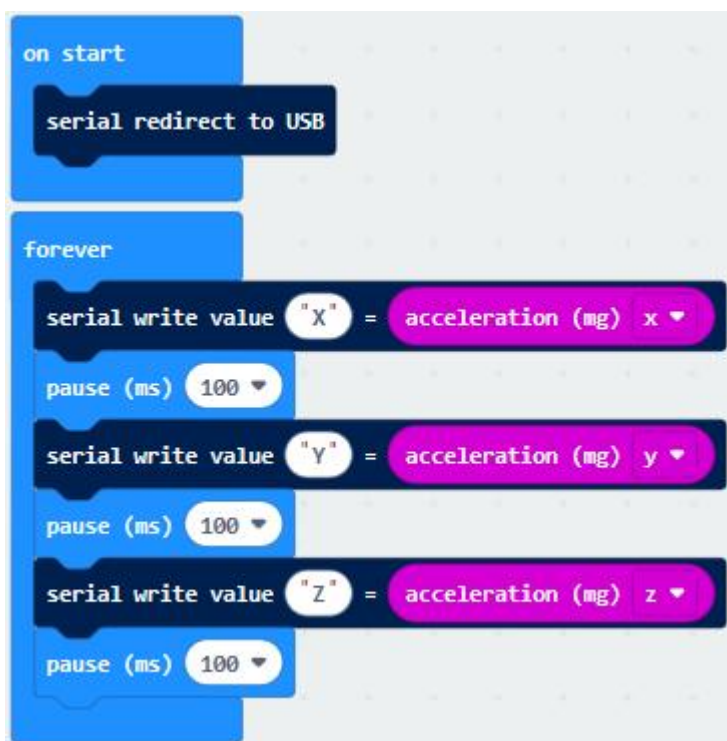


for 2 times and keep them into "forever" block, separately set the whole code string as follows:

keyestudio



Complete Program:



“on start”: command block runs once to start program.

Serial redirects to USB

The program under the block “forever” runs cyclically.

Serial write value “X”=acceleration value on x axis

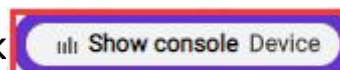
Serial write value “Y”=acceleration value on y axis

Serial write value “Z”=acceleration value on z axis

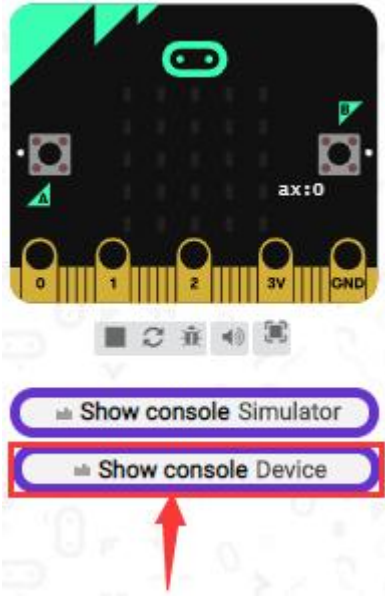
Click “JavaScript” to view the corresponding JavaScript code:



Download code 1 to micro:bit board, keep USB cable connected and click

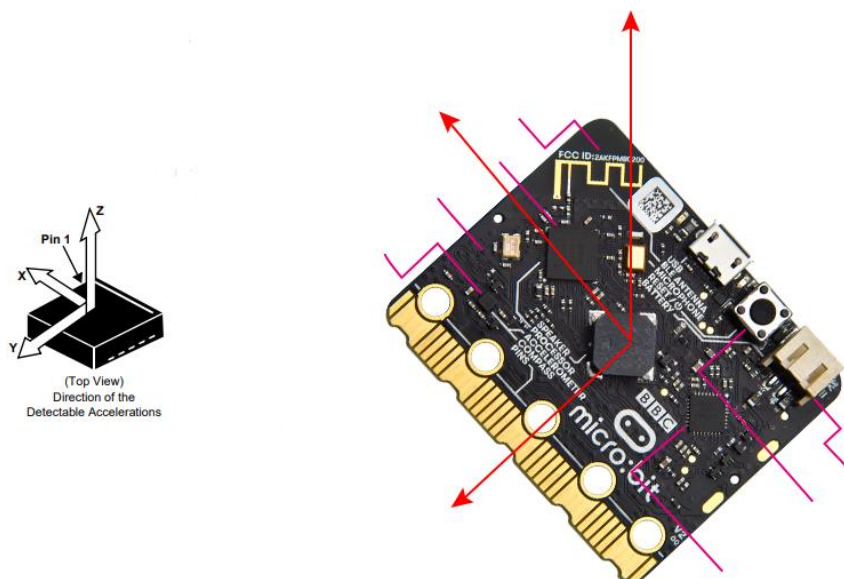


(How to quick download?)



After referring to the MMA8653FC data manual and the hardware schematic diagram of the Micro: Bit main board V2, the accelerometer coordinate of the Micro: Bit V2 motherboard are shown in the figure below:

The following interface shows the decomposition value of acceleration in X axis, Y axis and Z axis respectively, as well as acceleration synthesis (acceleration synthesis of gravity and other external forces).

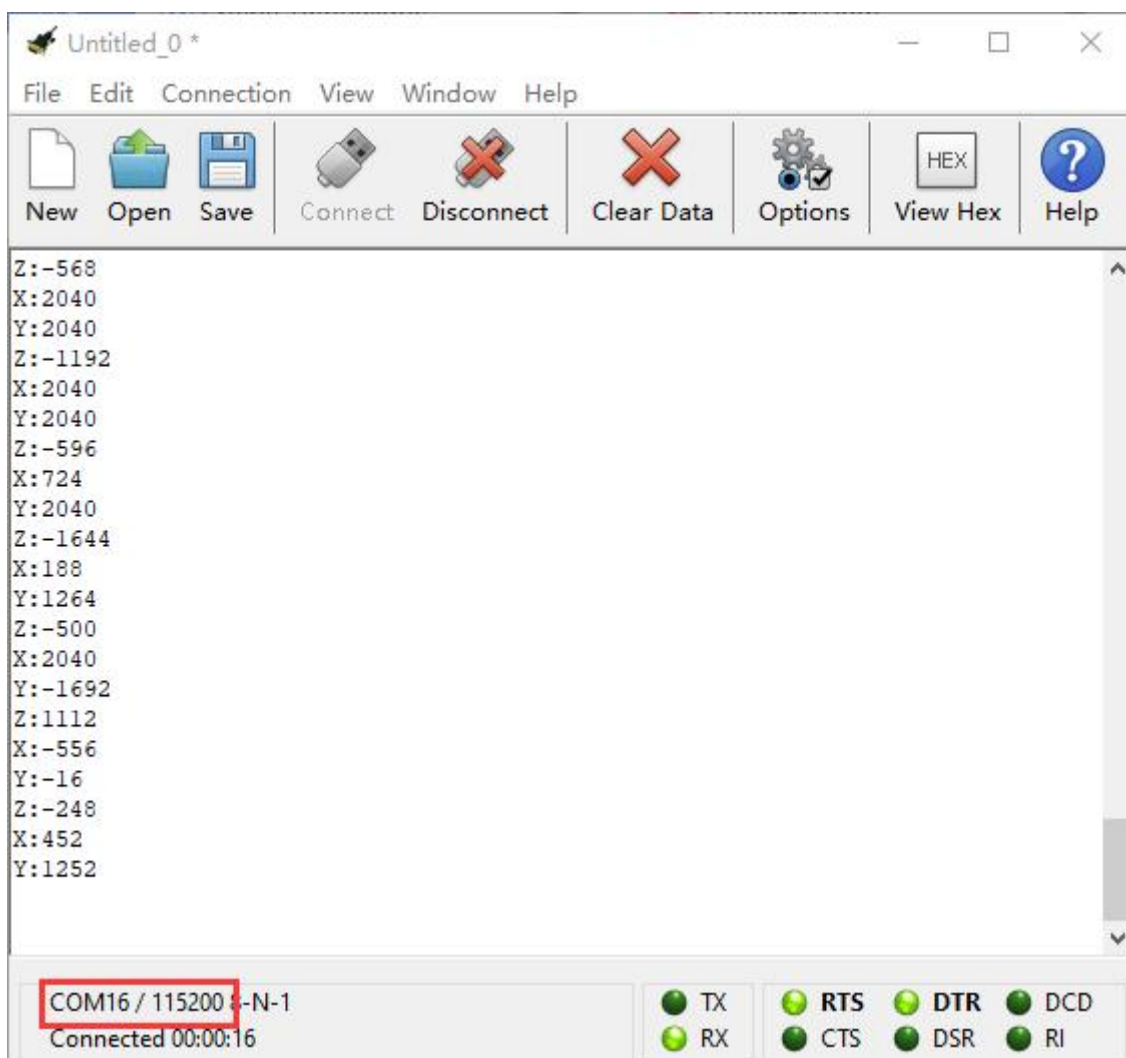


keystudio



If you're running Windows 7 or 8 instead of Windows 10, it won't be able to match devices via Google Chrome. You'll need to use the CoolTerm serial monitor software to read data.

You could open CoolTerm software, click Options, select SerialPort, set COM port and put baud rate to 115200 (after testing, the baud rate of USB SerialPort communication on Micro: Bit main board V2 is 115200), click OK, and Connect. The CoolTerm serial monitor shows the data of X axis, Y axis and Z axis , as shown in the figures below :



4. Test Result

After uploading the test code 1 to micro:bit main board V2 and powering the board via the USB cable, if we shake the Micro: Bit main board V2, no matter at any direction, the LED dot matrix displays the digit "1" .

(How to download? How to quick download?)

When it is kept upright (make its logo above the LED dot matrix) , the number 2 shows.



When it is kept upside down(make its logo below the LED dot matrix) , it shows as below.



When it is placed still on the desk, showing its front side, the number 4 appears.

When it is placed still on the desk, showing its back side, the number 5 exhibits.

When the board is tilted to the left , the LED dot matrix shows the number 6, as shown below.



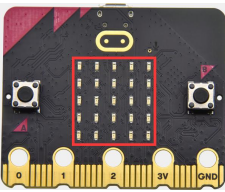
When the board is tilted to the right , the LED dot matrix displays the number 7, as shown below:



When the board is knocked to the floor, this process can be considered as a free fall and the LED dot matrix shows the number 8. (Please note that this test is not recommended for it may damage the main board.)

Attention: if you' d like to try this function, you can also set the acceleration to 3g, 6g or 8g.

Project 8: Light Detection



1. Description

In this project, we focus on the light detection function of the Micro: Bit main board V2. It is achieved by the LED dot matrix since the main board is not equipped with a photoresistor.

When the light irradiates the LED matrix, the voltage change will be produced. Therefore, we could determine the light intensity by voltage change.

2. Preparation

- A. Attach the micro:bit main board to your computer via the USB cable
 - B. Open the Web version of Makecode.
- Import Hex profile (How to import?)
- Or click “New Project” and drag blocks step by step

3. Test Code

The route to get test code ([How to load?](#))

File Type	Path	File Name
Hex file	..\2. Makecode Tutorial\Makecode Code	Project 8: Light Detection.hex

Or you could edit code step by step in the editing area.

(1)A. Enter “Advanced” → “Serial” → “serial redirect to USB” ;

B. Drag it into “on start” block.



(2) A. Go to “Serial” → “serial write value x =0” ;

B. Move it into “forever”



A. Click "Input" → "acceleration(mg) x"

B. Put "acceleration(mg) x" in the "0" box and change "x" into "Light intensity" .



A. Click "Basic" → "pause (ms) 100" ;

B. Lay it down into "forever" and set the delay to 100ms.



Complete Program:



"on start": command block runs once to start program.

Serial redirects to USB

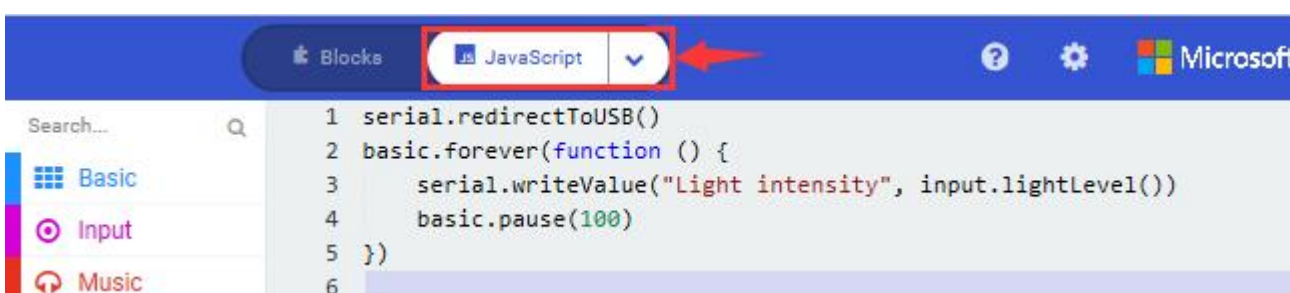
The program under the block "forever" runs cyclically.

Serial write value "Light intensity"

= light level

Delay in 100ms

Click "JavaScript" to switch into the corresponding JavaScript code:

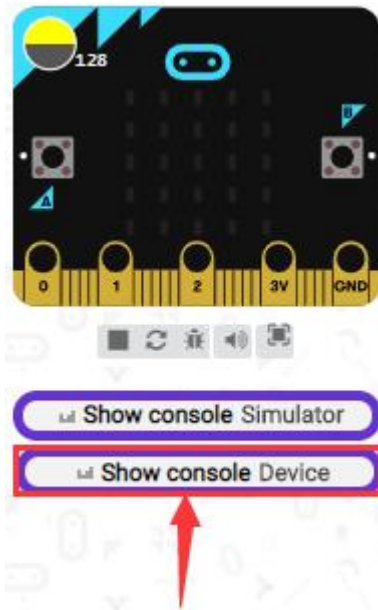


4. Test Result

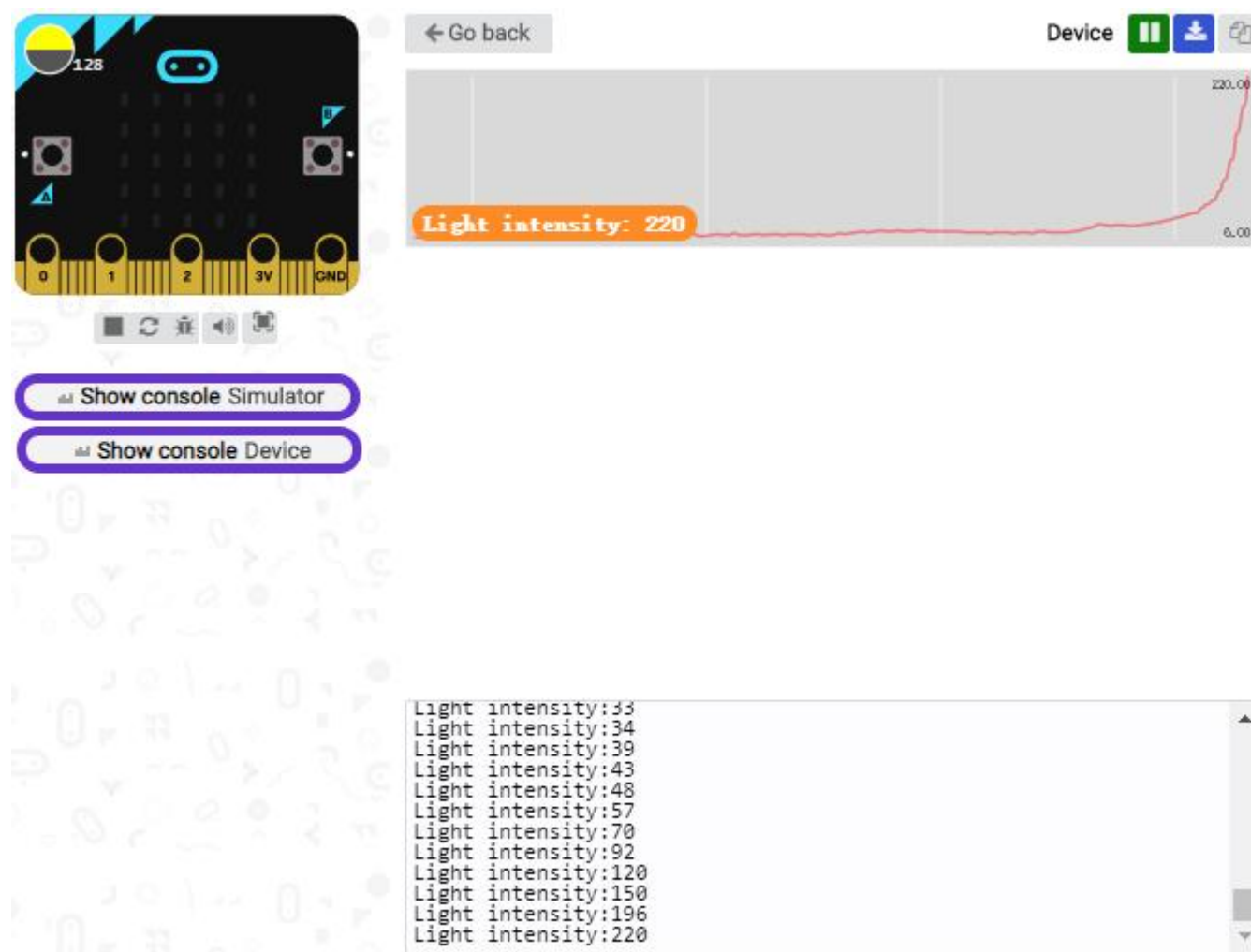
Download code to micro:bit board don't plug off the USB cable and click

Show console Device

(How to quick download?)



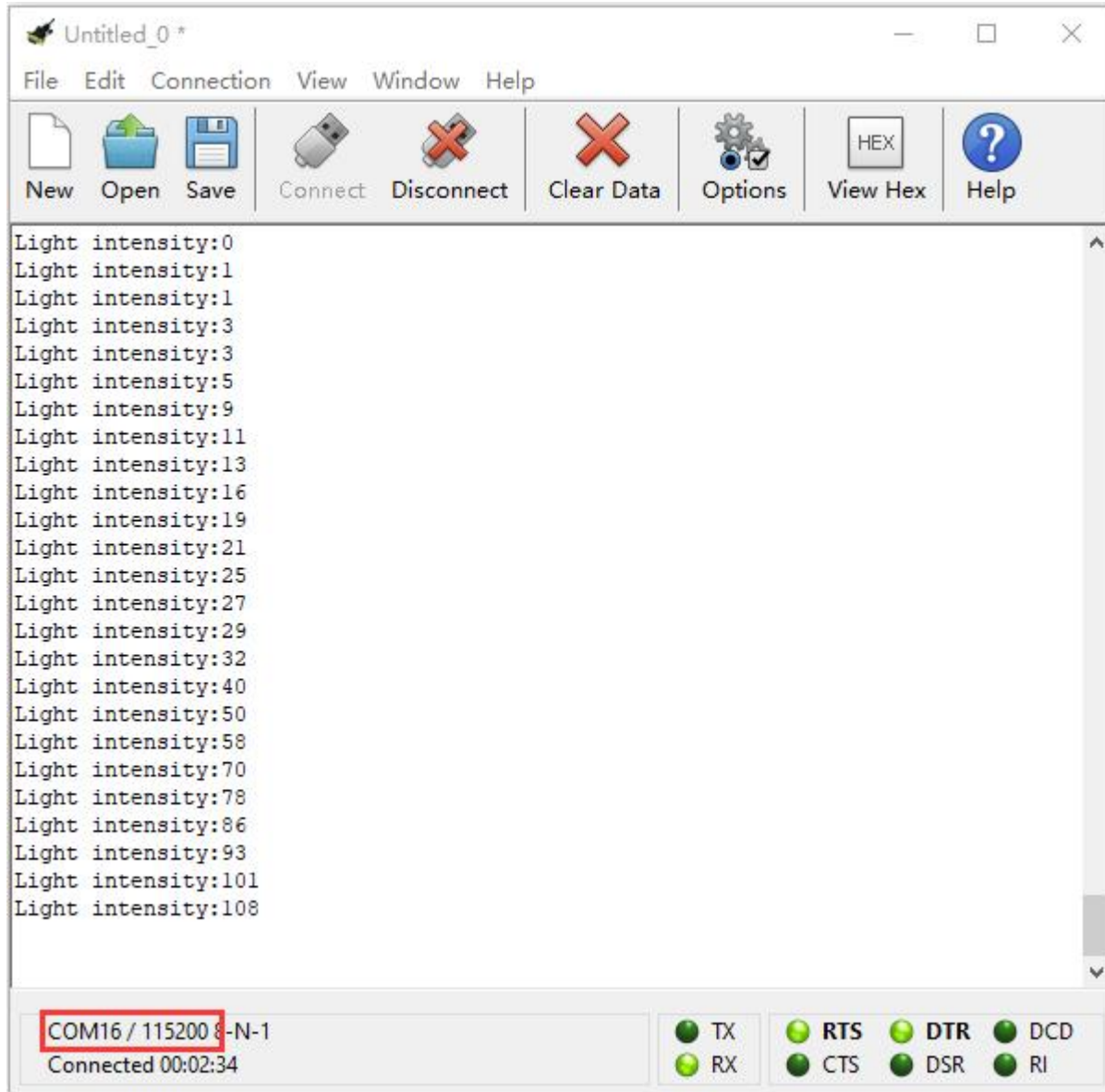
The intensity value is 0 when covering the LED dot matrix. And the value varies with the light intensity. When placing micro:bit under the sunlight, the stronger the light is, the larger the intensity value will be. As shown below:



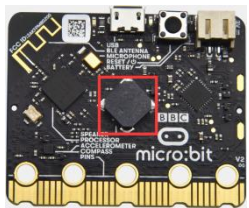
Open "CoolTerm" , click "Options" to select "SerialPort" , and set "COM" port and 115200 baud rate(the baud rate of USB serial communication of micro:bit is 115200 through the test).

Then click "OK" and "Connect" .

The light intensity value is shown below:



Project 9: Speaker



1. Description

Micro: Bit main board has an built-in speaker, which makes adding sound to the programs easier. It can also be programmed to produce all kinds of tones, like playing the song *Ode to Joy*.

2. Preparation

A. Attach the micro:bit main board to your computer via the USB cable

B. Open the Web version of Makecode.

Import Hex profile (How to import?)

Or click "New Project" and drag blocks step by step

3. Test Code

The route to get test code ([How to load?](#))

File Type	Path	File Name

keyestudio

Hex file	..\2. Makecode Tutorial\Makecode Code	Project 9: Speaker.hex
----------	--	------------------------

Or you could edit code step by step in the editing area.

Enter "Basic" module to find "show icon" and drag it into "on start" block;

Click the little triangle to find 



(2) Enter "Music" module to find and drag "play sound giggle until done" into "forever" block;

Enter "Basic" module to find and drag "pause(ms) 100" into "forever" block ;

Change 100 into 1000;



(3) Copy



three times and place it into "forever" block ;

Click the little triangle to select "happy" , " hello" , " yawn" ;



Complete Program:

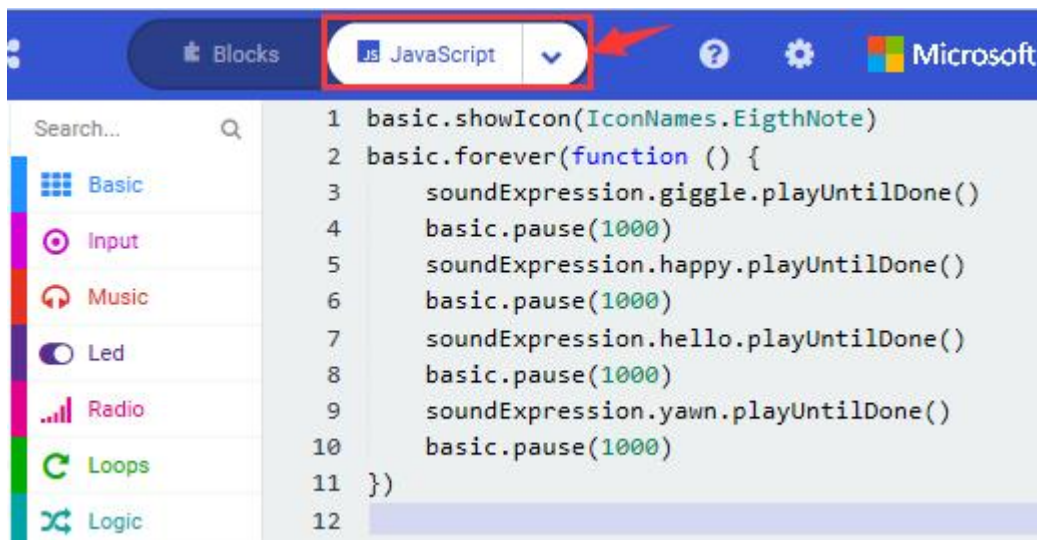
The diagram shows a Scratch script with the following blocks:

- on start** (blue block):
 - show icon** (blue block) with a 4x4 grid icon.
- forever** (blue loop block):
 - play sound** (red block) with **giggle** selected and **until done** checked.
 - pause (ms)** (blue block) with **1000** selected.
 - play sound** (red block) with **happy** selected and **until done** checked.
 - pause (ms)** (blue block) with **1000** selected.
 - play sound** (red block) with **hello** selected and **until done** checked.
 - pause (ms)** (blue block) with **1000** selected.
 - play sound** (red block) with **yawn** selected and **until done** checked.
 - pause (ms)** (blue block) with **1000** selected.

The right side of the image contains a list of 11 steps explaining the script:

- ① "on start" : command block runs once to start program.
- ② LED displays " " (with a 4x4 grid icon).
- ③ The program under the block "forever" runs cyclically.
- ④ The buzzer will sound "giggle"
- ⑤ Delay in 1000ms
- ⑥ The buzzer will sound "happy"
- ⑦ Delay in 1000ms
- ⑧ The buzzer will sound "hello"
- ⑨ Delay in 1000ms
- ⑩ The buzzer will sound "yawn"
- ⑪ Delay in 1000ms

Select "JavaScript" and "Python" to switch into JavaScript and Python language code:



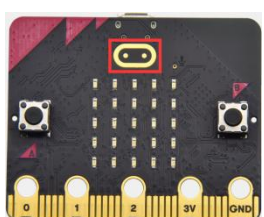
4. Test Result

After uploading the test code to micro:bit main board V2 and powering the board via the USB cable, the speaker utters sound and the LED dot matrix shows the logo of music.

(How to download? How to quick download?)

(How to download? How to quick download?)

Project 10: Touch-sensitive Logo



1. Description

The Micro: Bit main board V2 is equipped with a golden touch-sensitive logo, which can act as an input component like an button.

It contains a capacitive touch sensor that senses small changes in the electric field when pressed (or touched), just like your phone or tablet screen. When you press it , the program can be activated.

2. Preparation

- A. Attach the micro:bit main board to your computer via the USB cable
- B. Open the Web version of Makecode.

Import Hex profile (How to import?)

Or click “New Project” and drag blocks step by step

3. Test Code

The route to get test code ([How to load?](#))

File Type	Path	File Name
Hex file	..\2. Makecode Tutorial\Makecode Code	Project 10: Touch-sensitive Logo.hex

Or you could edit code step by step in the editing area.

- (1) Delete block “on start” and “forever” ;
- (2)Enter “Input” module to find and drag “on logo pressed” ;
- Click the little triangle to find “touched” ;



- (3) Enter module “Variables” →choose “Make a Variable” →input “start” →click “OK”

The variable “start” is established;

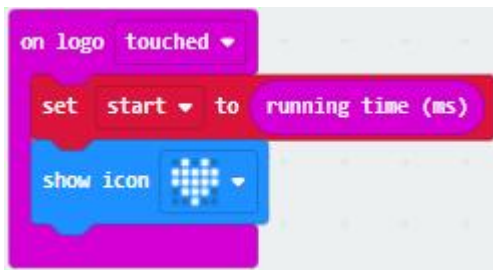
Enter “Variables” module to find and drag “set start to 0” into “on logo touched” block;



(4)Enter "Input" module →click "more" → find and drag "running time(ms)" into the "0" of "set start to 0" block;



(5)Enter "Basic" module to find and drag "show icon♥" into "on logo touched" block;



(6)Enter "Input" module to find and drag "on logo pressed" →choose "released" → establish variable "time";
Enter "Variables" module to find and drag "set time to 0" into "on logo pressed" block;
Enter "Math" module to find and drag "0-0" into the "0" of "set start to 0" block;



(7)Enter "Input" module→ "more" → find and drag "running time(ms)" into "0" on the left side of "0-0" ;

Enter "Variables" module to find and drag "start" into "0" on the right side of "0-0" ;



(8)Enter "Basic" module to find and drag "show number" into "on logo released" block;

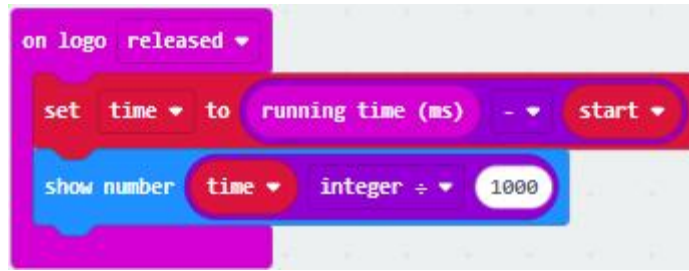
Enter "Math" module to find and drag "square root 0" into "0" ; Click the little triangle to find" integer÷" ;



(9) Enter "Variables" module to find and drag "time" into "0" on the left side of "0-0" and change the "0" on the

keystudio

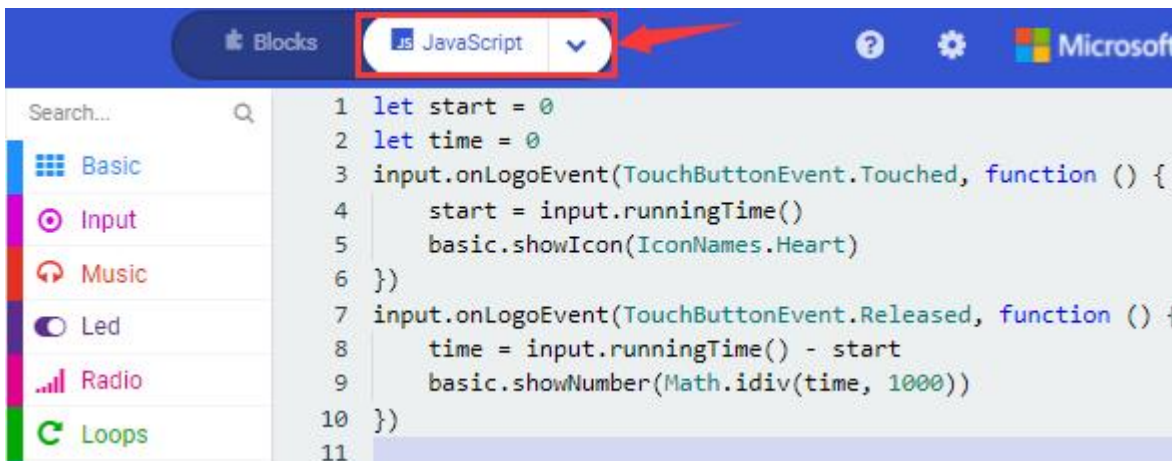
right side to " 1000" ;



Complete Program:

	①Touch the logo on the Microbit mainboard ② Assign the running time value to the variable start ③LED display "♥" pattern ③Release the logo by hand ④Assign the value of running time- variable start to variable time ⑤LED dot matrix display variable time divided by 1000 rounded
--	--

Select "JavaScript" and "Python" to switch into JavaScript and Python language code:

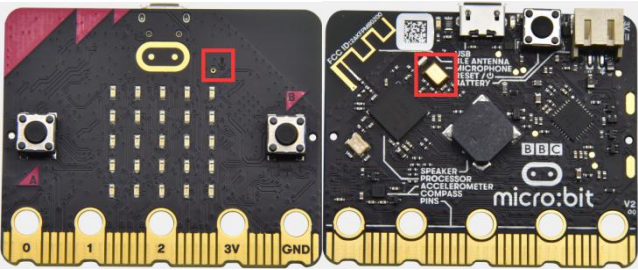


After uploading the test code to micro:bit main board V2 and powering the board via the USB cable, the LED dot matrix exhibits the "♥" pattern when the touch-sensitive logo is pressed or touched and displays digit when the

logo is released.

(How to download? How to quick download?)

Project 11: Microphone



1. Description

The Micro: Bit main board has a built-in microphone, which can test the volume of ambient environment. When you clap, the microphone LED indicator turns on. Furthermore, it can measure the intensity of sound, thereby you can make a noise scale or disco lighting changing with music.

The microphone is placed on the opposite side of the microphone LED indicator and in proximity with holes that lets sound pass. When the board detects the sound, the LED indicator lights up.

2. Preparation

- A. Attach the micro:bit main board to your computer via the USB cable
 - B. Open the Web version of Makecode.
- Import Hex profile (How to import?)
- Or click “New Project” and drag blocks step by step

3. Test Code

The route to get test code (**How to load?**)

File Type	Path	File Name
Hex file	..\2. Makecode Tutorial\Makecode Code	Project 11: Microphone-1.hex

Or you could edit code step by step in the editing area.

- (1) Delete block “on start” and “forever” ;
 - (2) Enter “Input” module to find and drag “on loud sound” ;
- Enter “Basic” module to find and drag “show number” into “on loud sound” block ;



(3)Copy once;

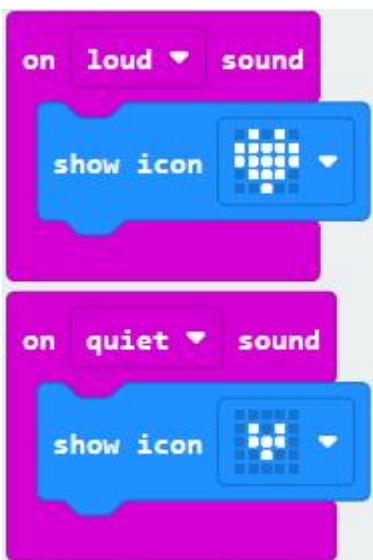


Click the little triangle of "loud" to choose "quiet" ;

Click the little triangle of "♥" to choose "🧡" ;

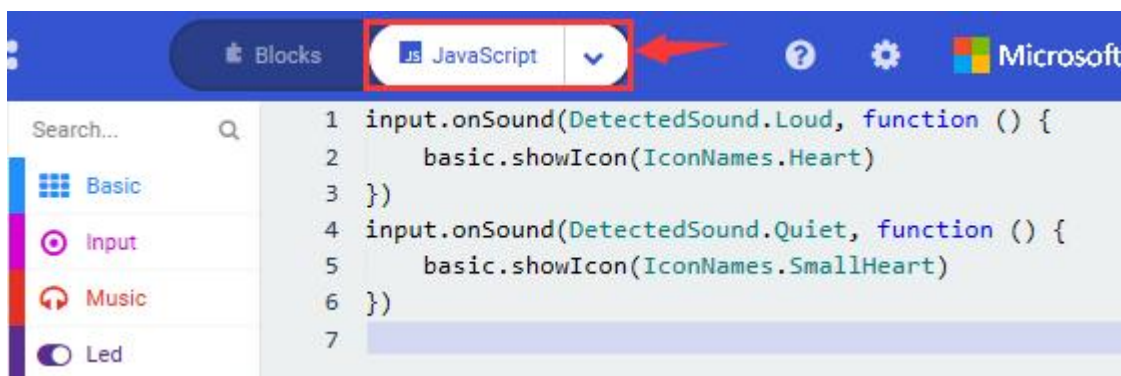


Complete Program:



-①The microphone detects sound
-②LED dot matrix display "♥"
-③The microphone detects sound
-④LED dot matrix display "🧡"

Select "JavaScript" and "Python" to switch into JavaScript and Python language code:



4. Test Results 1

After uploading test code1 to micro:bit main board V2 and powering the board via the USB cable, the LED dot matrix displays pattern "♥" when you claps and pattern 🧡 when it is quiet around.

(How to download? How to quick download?)

Code 2:

keyestudio

The route to get test code ([How to load?](#))

File Type	Path	File Name
Hex file	..\2. Makecode Tutorial\Makecode Code	Project 11: Microphone-2.hex

Or you could edit code step by step in the editing area.

Enter “Advanced” module→ choose “Serial” to find and drag “serial redirect to USB” into “on start” block ;



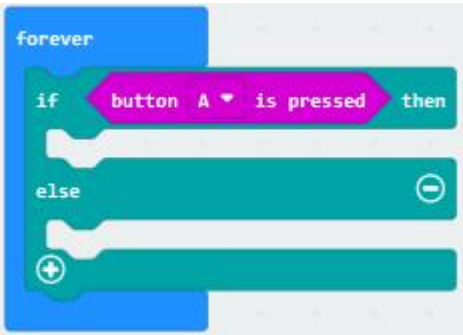
Enter “Variables” module → choose “Make a Variable” → input “maxSound” → click “OK” ,variable “ maxSound” is established;

Enter “Variables” module to find and drag “set maxSound to 0” into “on start” block ;



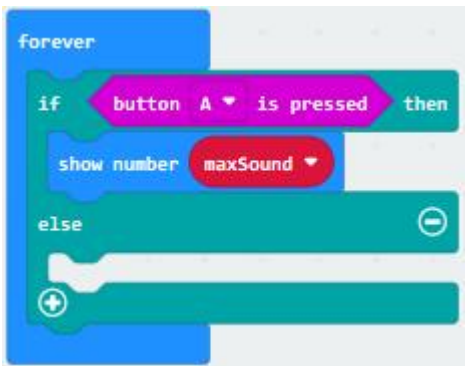
Enter “Logic” module to find and drag “if true then...else” into “forever” block ;

Enter “Input” module to find and drag “button A is pressed” into “then” ;



Enter “Basic” module to find and drag “show number” into “then” ;

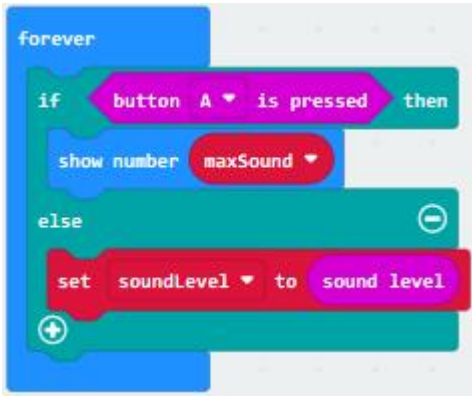
Enter “Variables” module to find and drag “maxSound” into “0” ;



Establish variable “soundLevel” ;

Enter “Variables” module to find and drag “set soundLevel to 0” into “else” ;

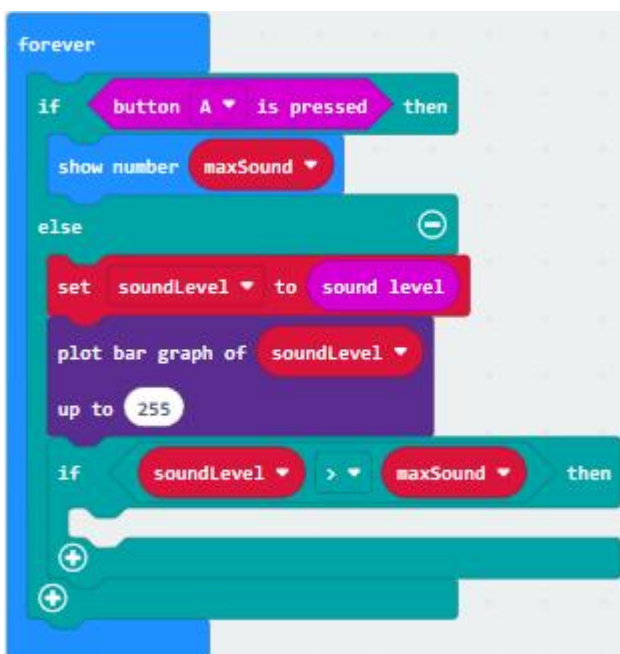
Enter “Input” module to find and drag “sound level” into “0” ;



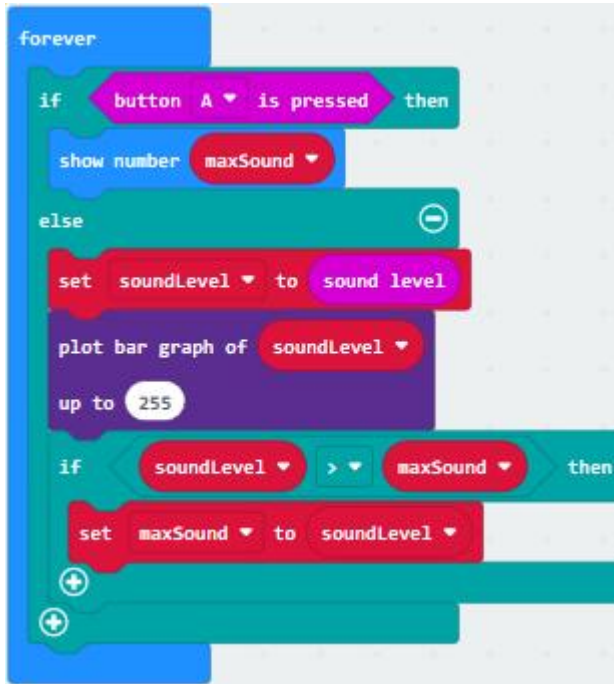
Enter "Led" module to find and drag "plot bar graph of 0 up to 0" into "else" ;
 Enter "Variables" module to find and drag "soundLevel" into the "0" behind "of" ;
 Change the "0" behind "up" to "255" ;



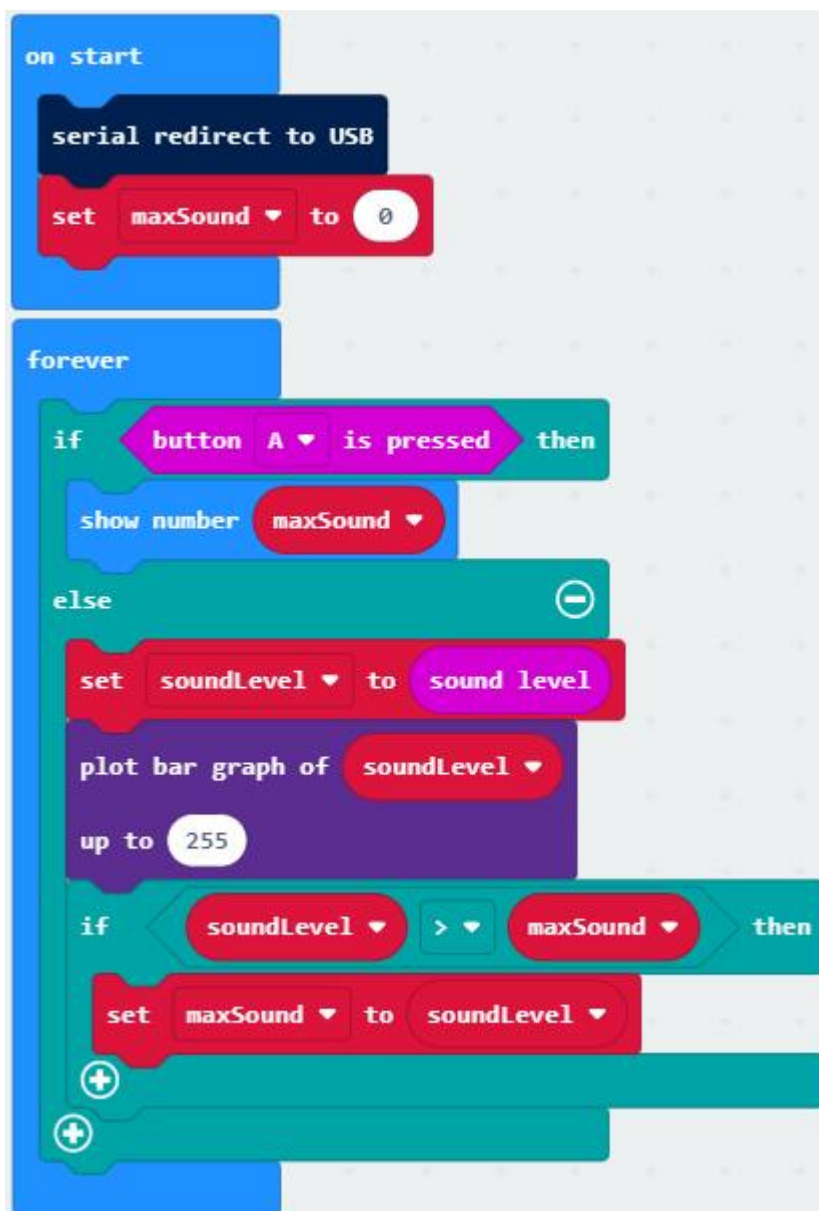
Enter "Logic" module to find and drag "if true then" into "else" block ;
 Enter "Logic" module to find and drag "0 > 0" into "then" ;
 Enter "Variables" module to find and drag "soundLevel" into "0" on the left side of "0-0" ;
 Enter "Variables" module to find and drag "maxSound" into "0" on the right side;



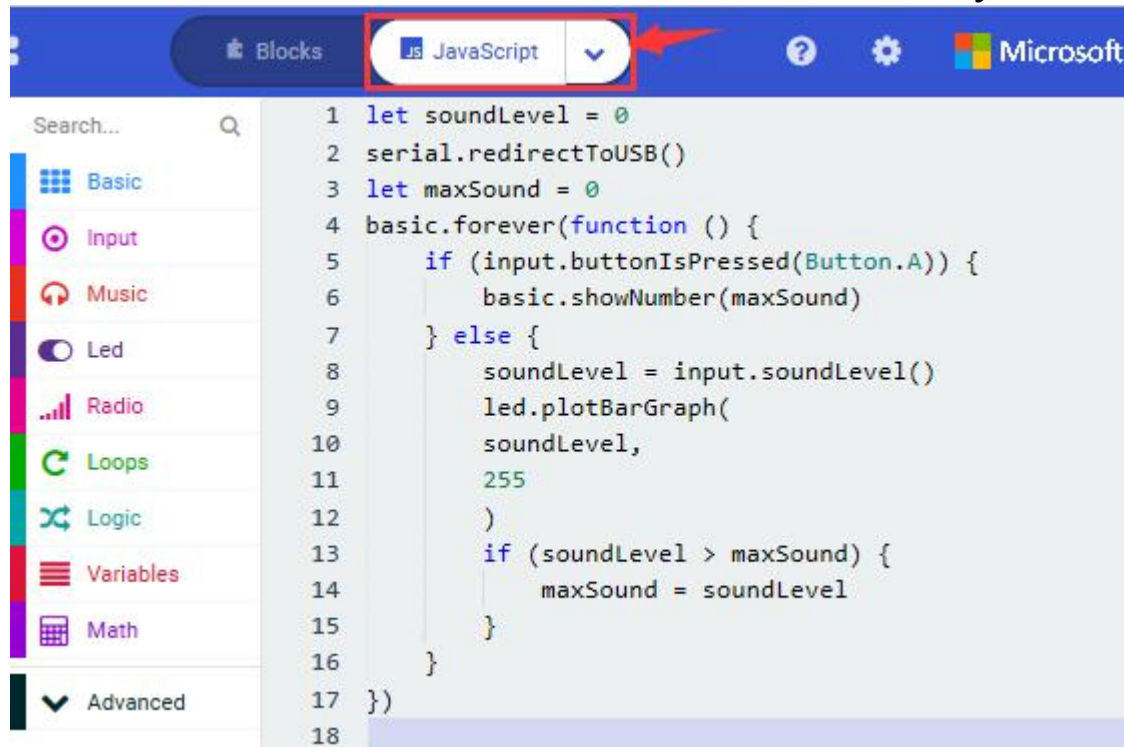
Enter "Variables" module to find and drag "set maxSound to 0" into the second "then" ;
 Enter "Variables" module to find and drag "soundLevel" into the "0" ;



Complete Program:



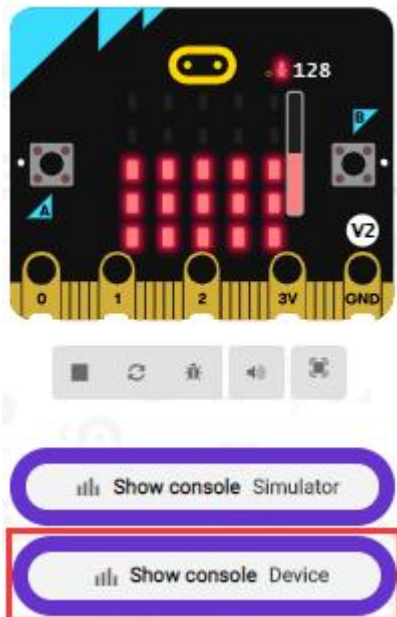
Select "JavaScript" and "Python" to switch into JavaScript and Python language code:



5. Test Results 2

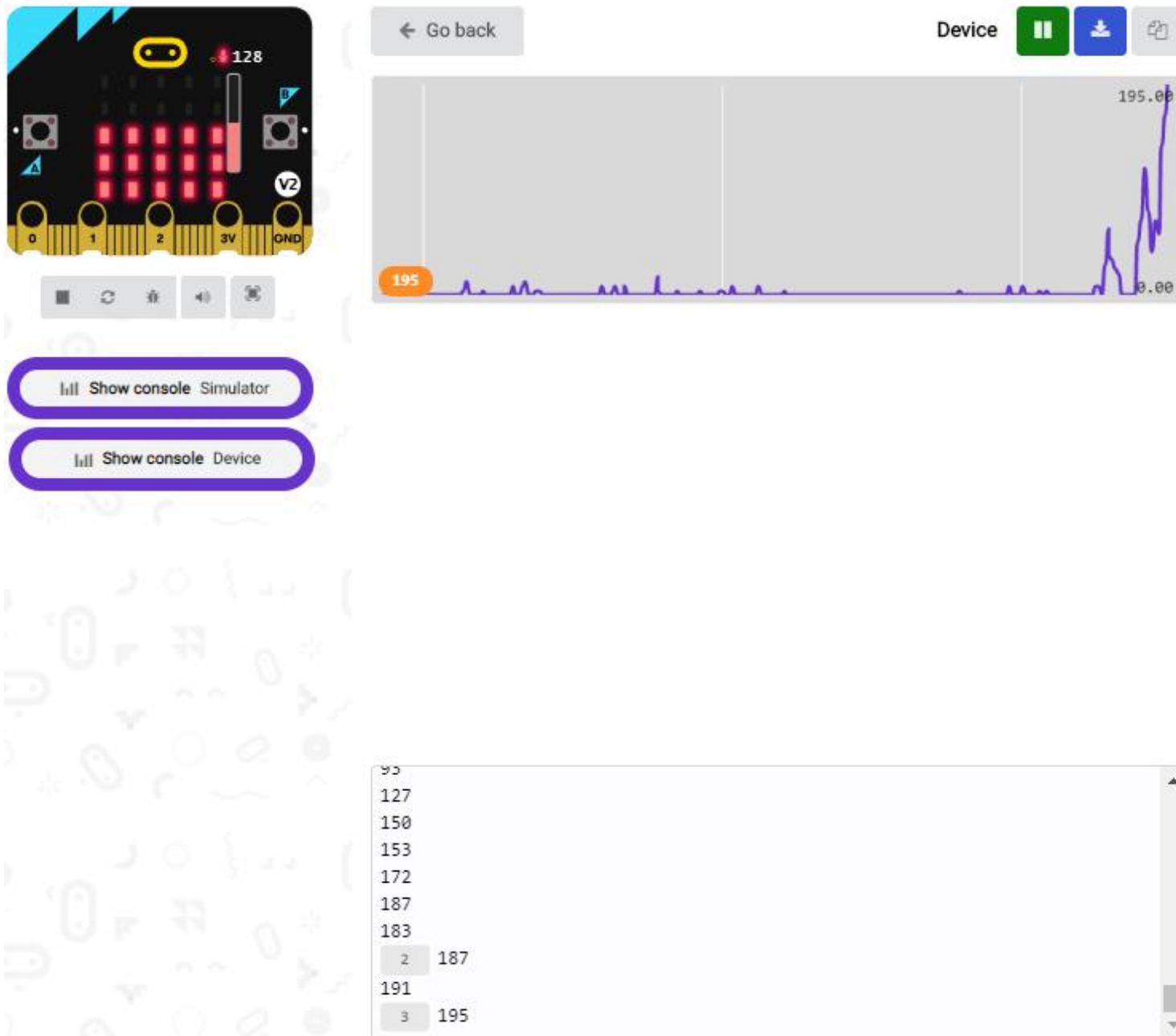
Upload test code 2 to micro:bit main board V2, power the board via the USB cable and click "Show console Device" as shown below.

(How to quick download?)



When the sound is louder around, the sound value shows in the serial port is bigger as shown below.

keyestudio



What' s more, when pressing the button A, the LED dot matrix displays the value of the biggest volume(please note that the biggest volume can be reset via the Reset button on the other side of the board) while when clapping, the LED dot matrix shows the pattern of the sound.

Project 12: Bluetooth Wireless Communication



1. Description

The Micro: Bit main board V2 comes with a nRF52833 processor (with a built-in BLE(Bluetooth Low Energy) device Bluetooth 5.1) and a 2.4GHz antenna for Bluetooth wireless communication and 2.4GHz wireless communication. With the help of them, the board is able to communicate with a variety of Bluetooth devices, including smart phones and tablets.

In this project, we mainly concentrate on the Bluetooth wireless communication functions of this main board. Linked

keyestudio

with Bluetooth, it can transmit code or signals. To this end, we should connect an Apple device (a phone or an iPad) to the board.

Since setting up Android phones to achieve wireless transmission is similar to that of Apple devices, so we don't need to illustrate again.

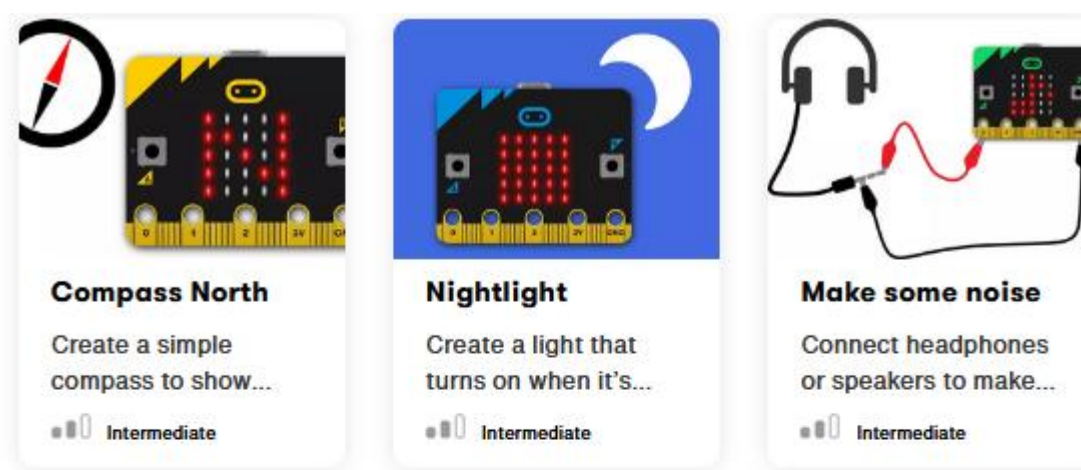
2. Preparation

A. Attach the micro:bit main board to your computer via the USB cable

B. iPhone device (phone /iPad) or Android phone.

3. Procedures

For Apple devices, enter this link: <https://www.microbit.org/get-started/user-guide/ble-ios/> with your computer first, and then click "Download pairing HEX file" to download the Micro: Bit firmware to a folder or desk, and upload the downloaded firmware to the Micro: Bit main board V2.



If you need help

If you're having problems flashing code from your iOS device to your micro:bit, download this HEX file and transfer it to your micro:bit from a computer, or visit our support site.

[Download pairing HEX file](#)

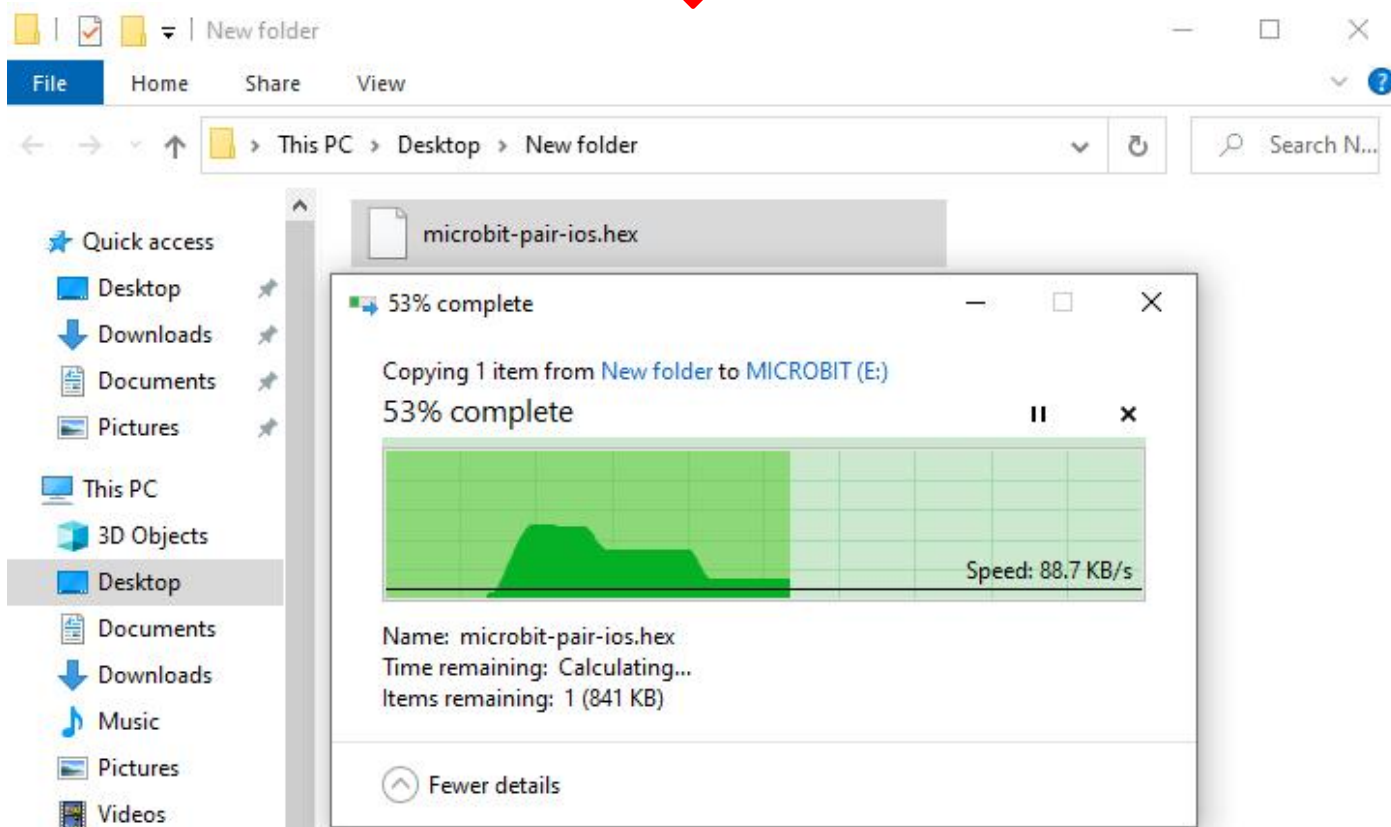
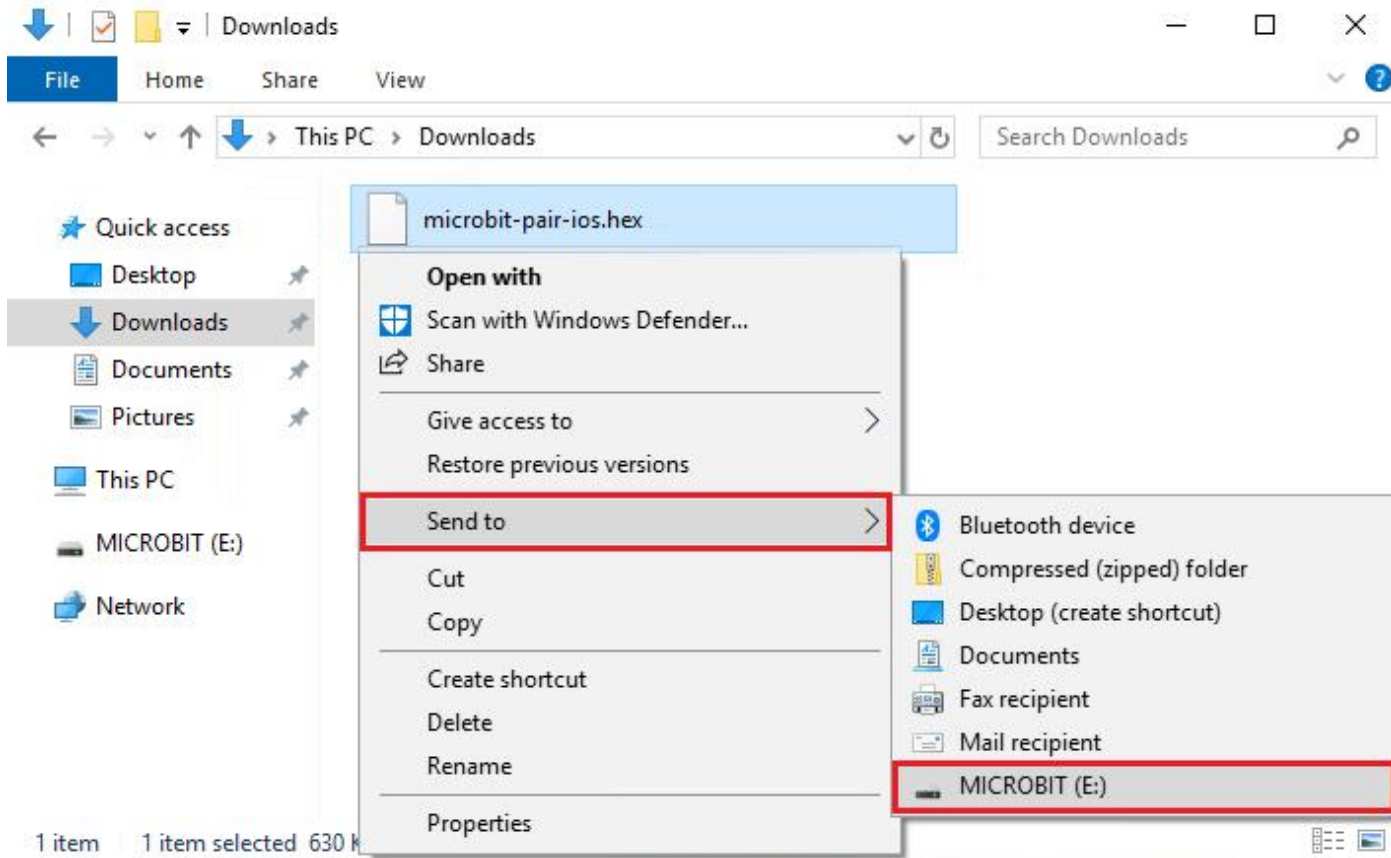
[iOS app support](#)

Monitor and control

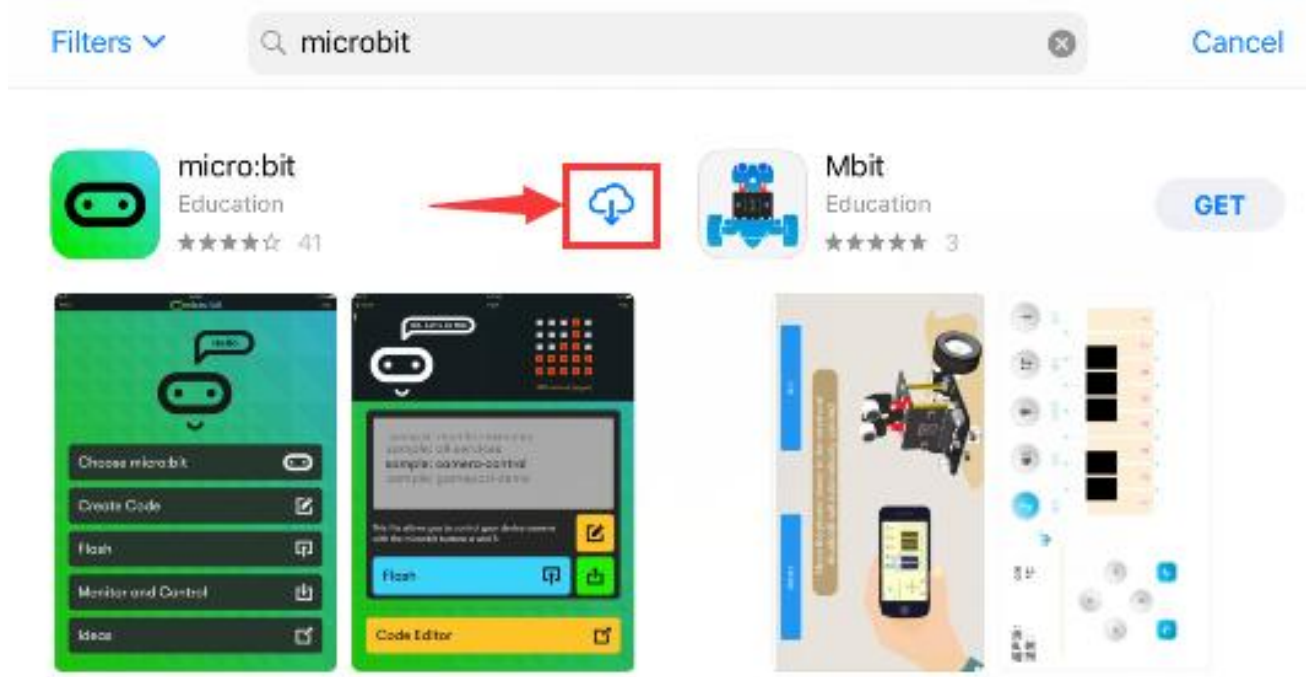
The 'Monitor and control' section of the iOS app allows you to observe real-time data from the micro:bit sensors, send messages directly to the LEDs and control the micro:bit buttons and pins from your iPad or iPhone.



keystudio



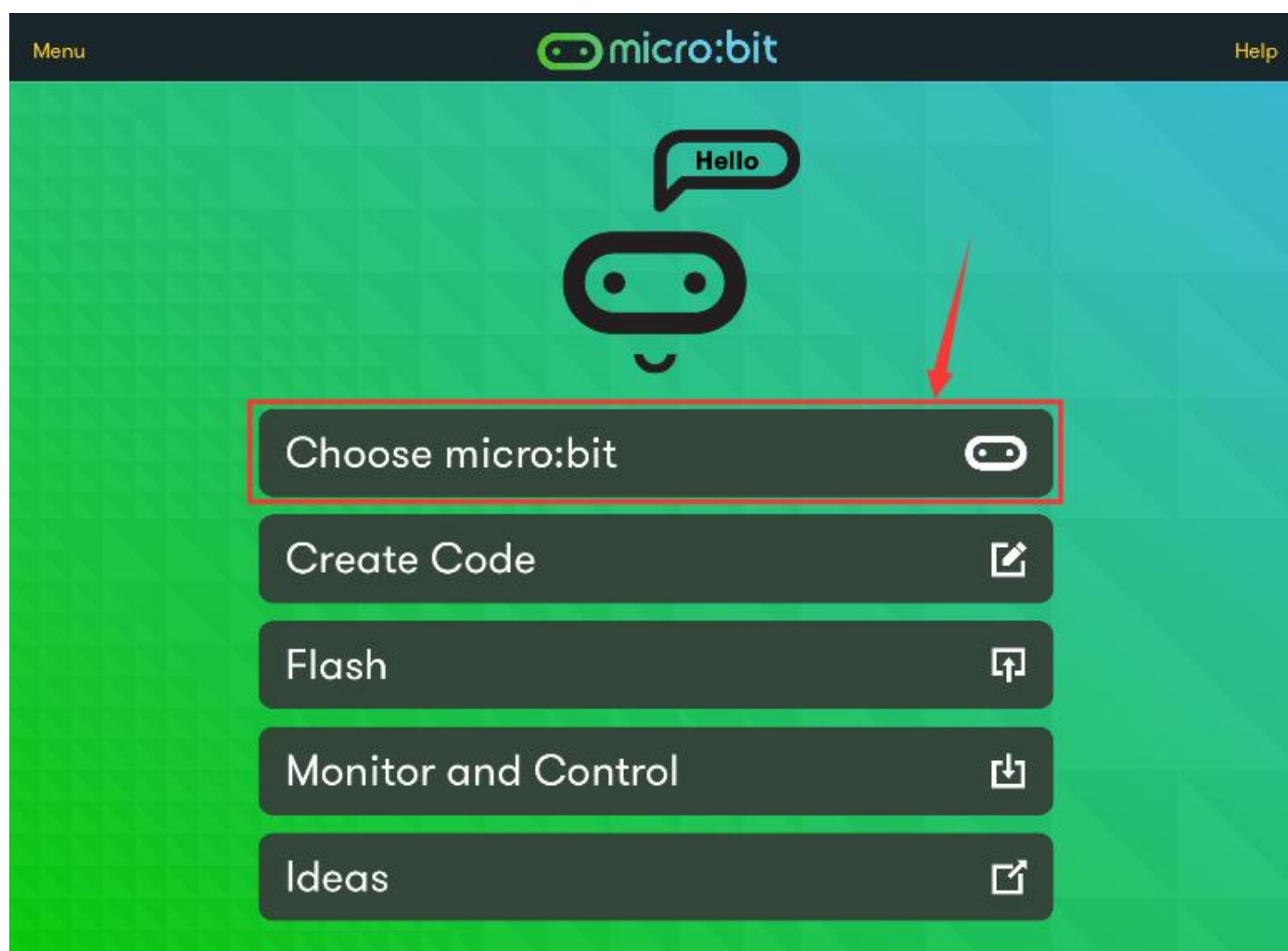
Open App Store to search "micro bit" in your App Store to download the APP micro:bit then click " " .



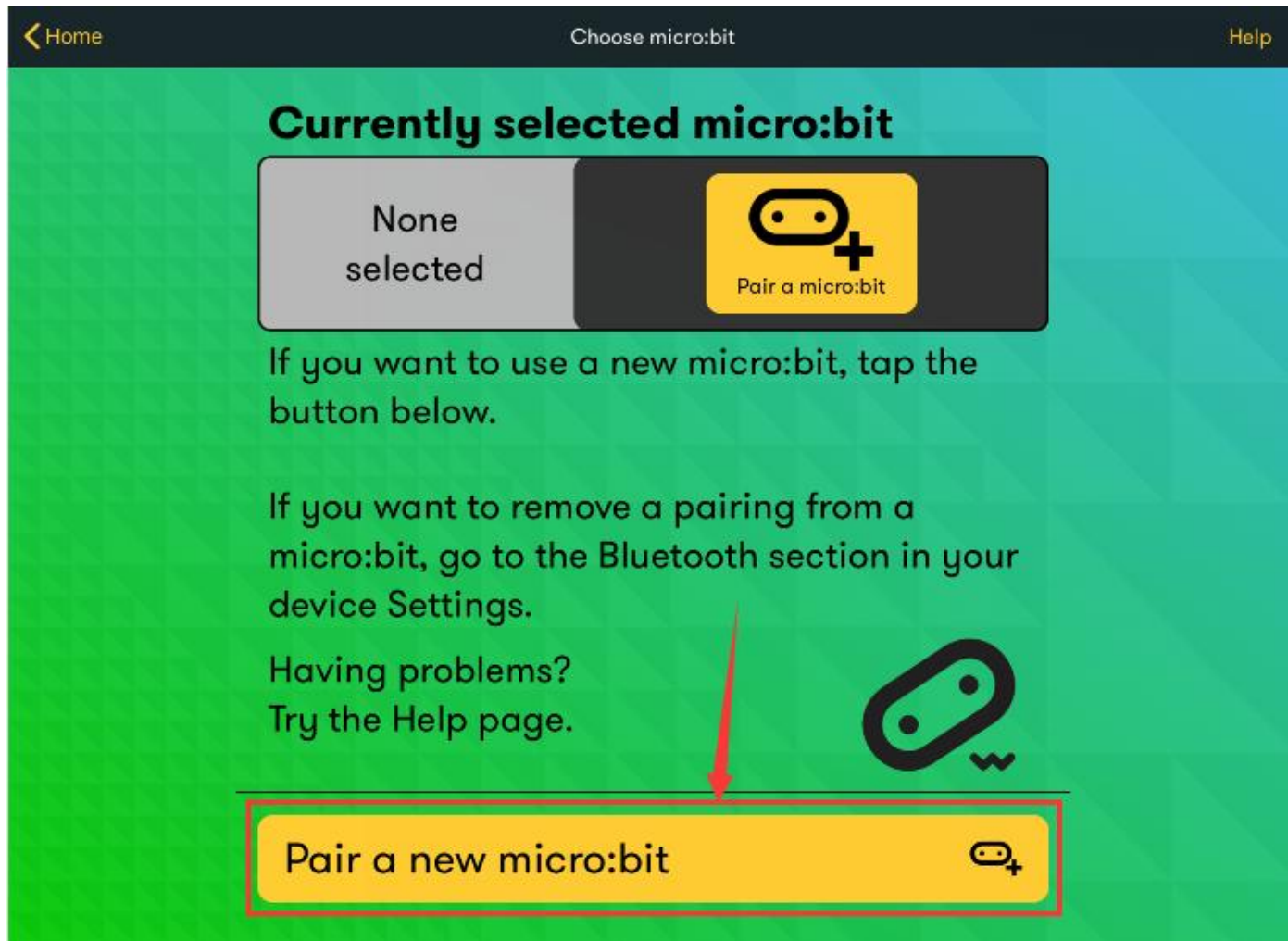
Connect your Apple device with Micro: Bit main board V2:

Firstly, turn on the Bluetooth of your Apple device and open the APP micro:bit to select item "Choose micro:bit" to start pairing Bluetooth.

Please make sure that the Micro: Bit main board V2 and your computer are still linked via the USB cable.



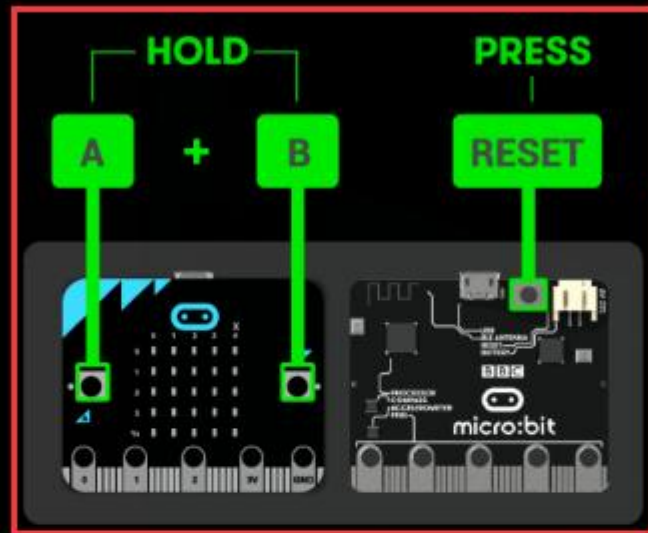
Secondly, click "Pair a new micro:bit" ;



Following the instructions to press button A and B at the same time(do not release them until you are told to) and press Reset & Power button for a few seconds.

Release the Reset & Power button, you will see a password pattern shows on the LED dot matrix. Now , release buttons A and B and click Next.

How to pair your micro:bit



Let's do this

Step 1

HOLD the A and B buttons and
PRESS and RELEASE RESET

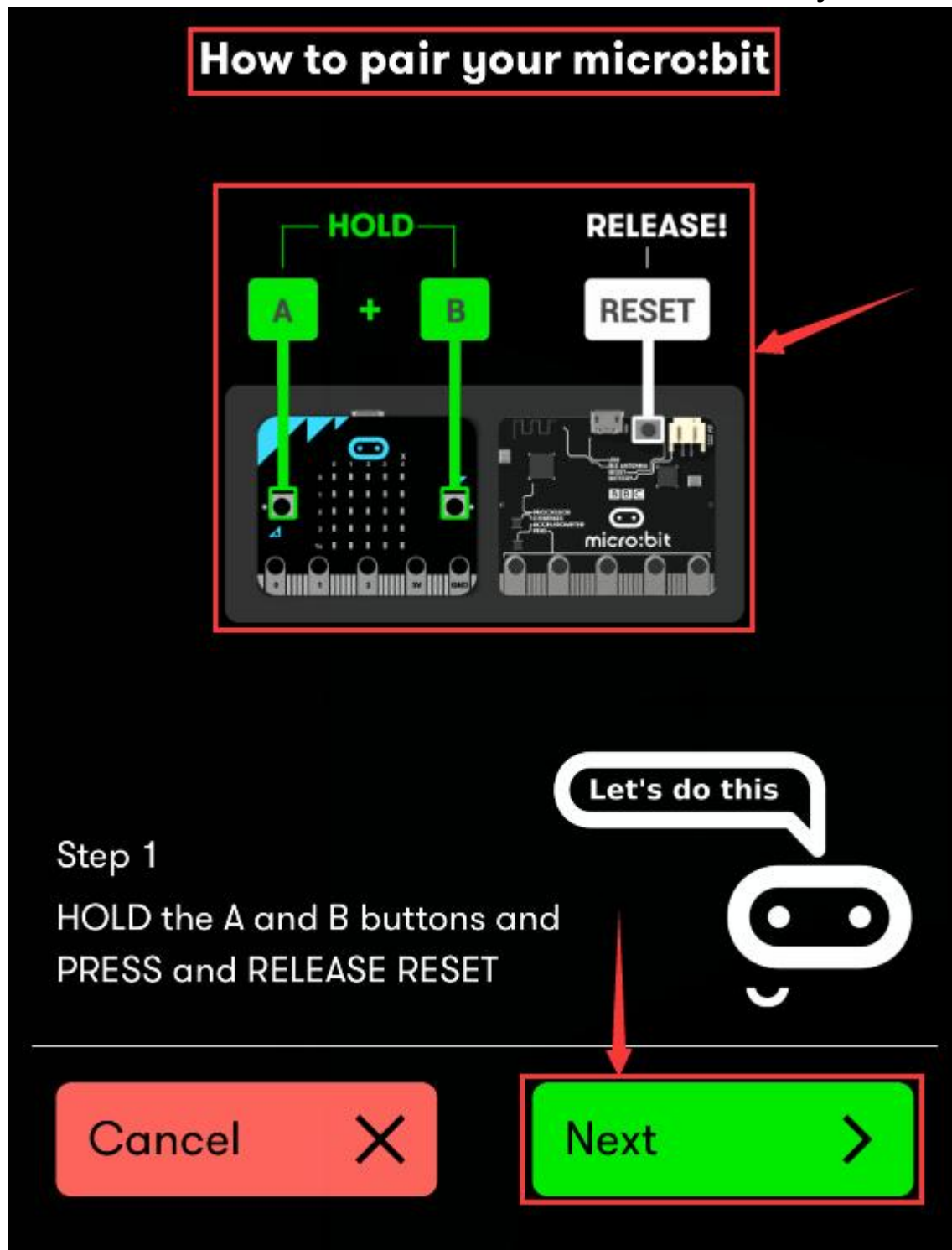


Cancel

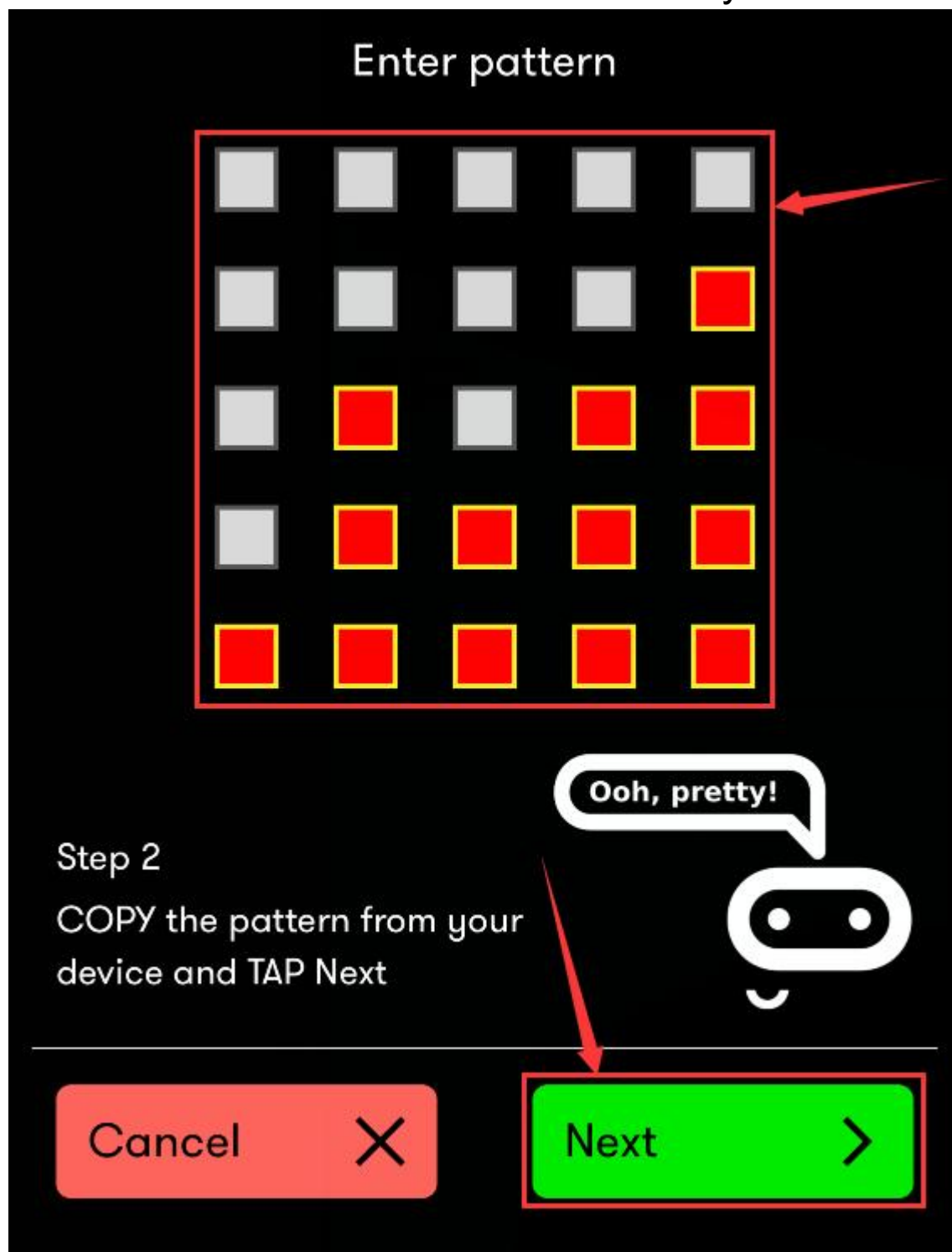


Next

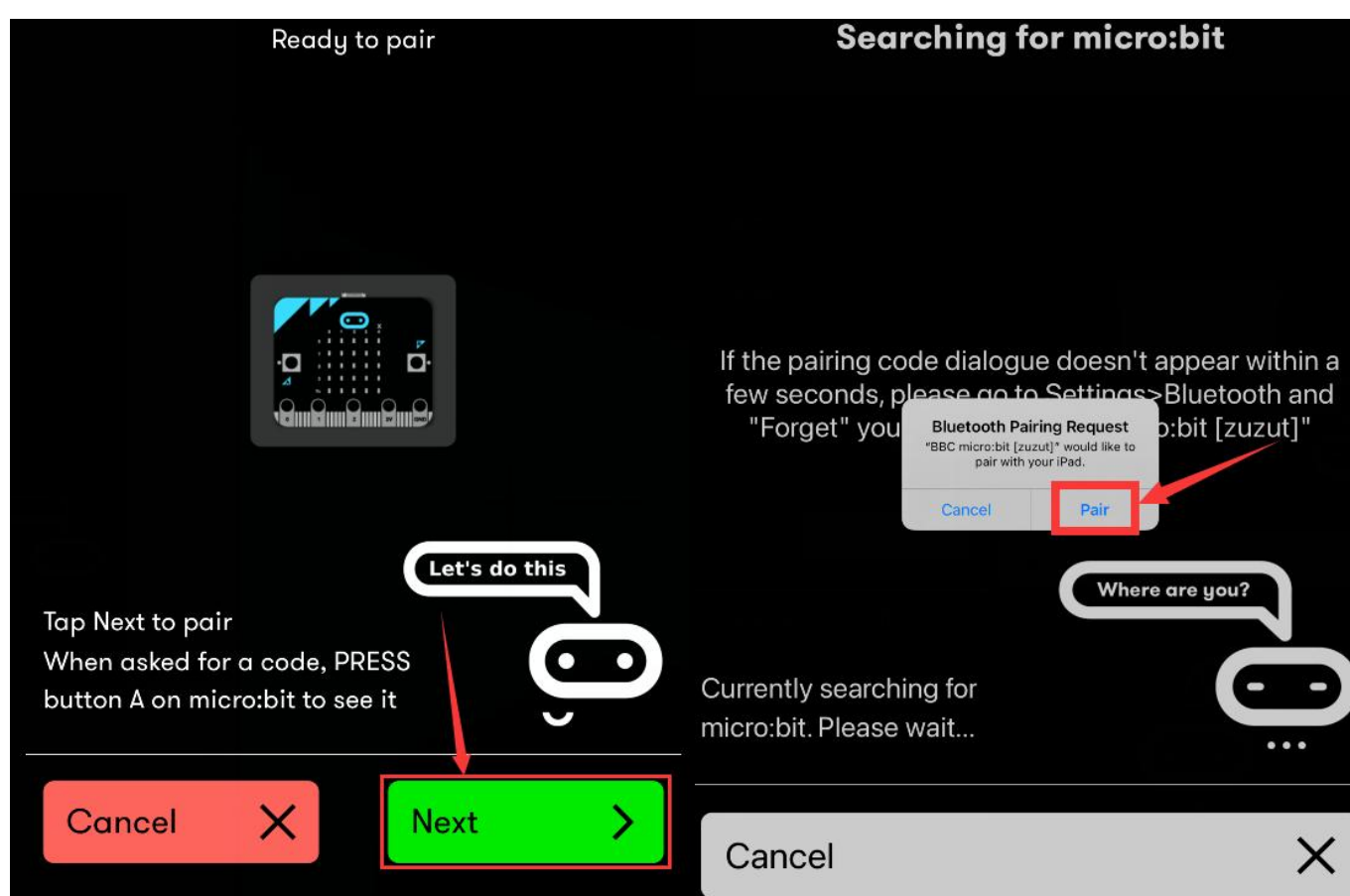


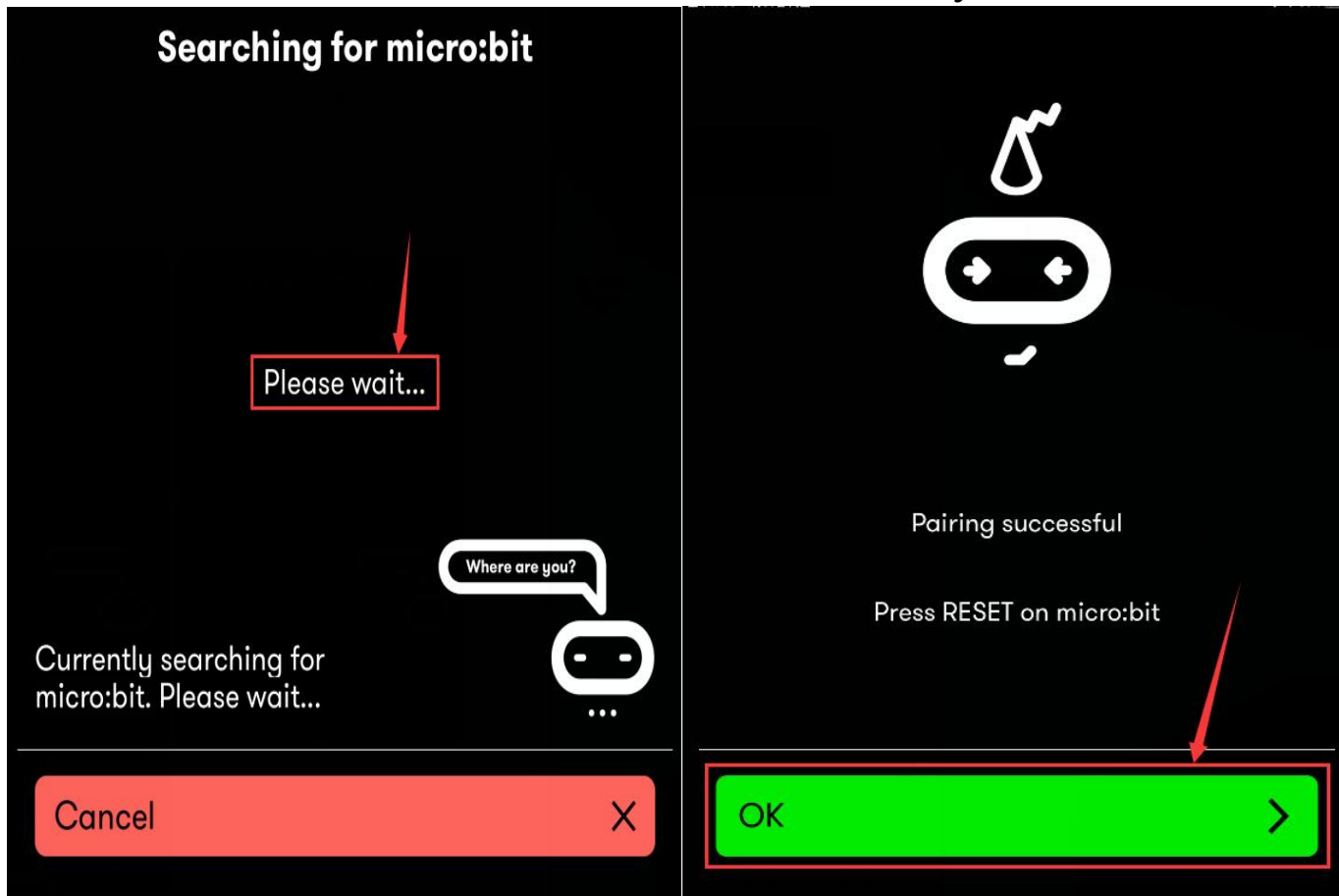


Set the password pattern on your Apple device as the same pattern showed on the matrix and click Next.



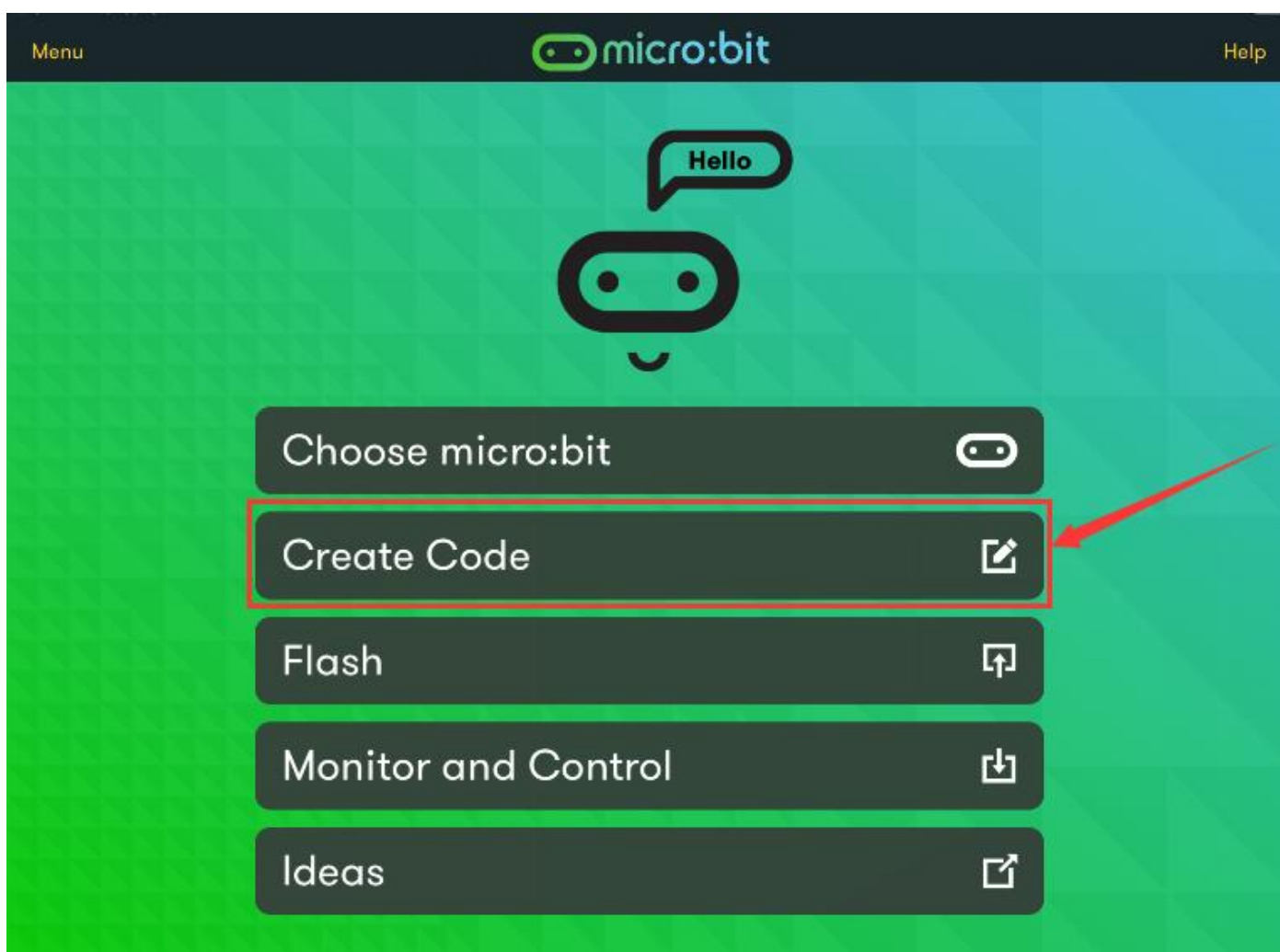
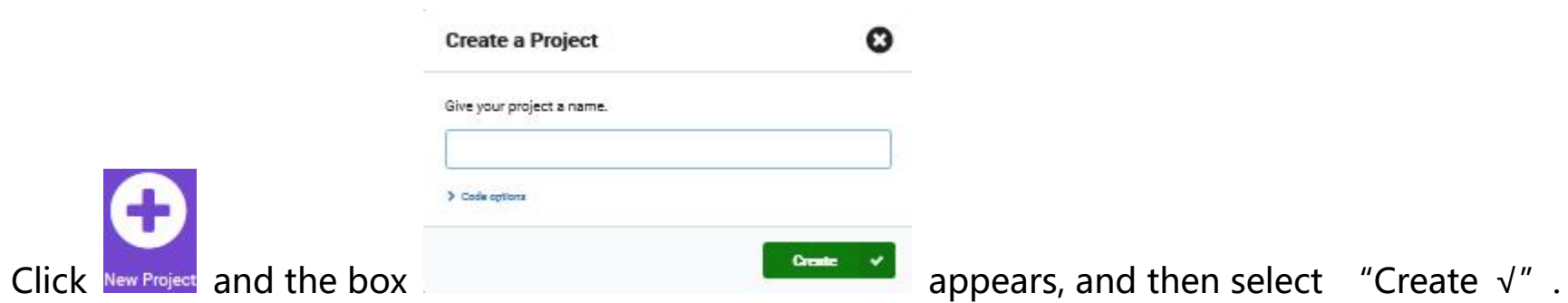
Still click Next and a dialog box props up as shown below. Then click "Pair". A few seconds later, the match is done and the LED dot matrix displays the "√" pattern.



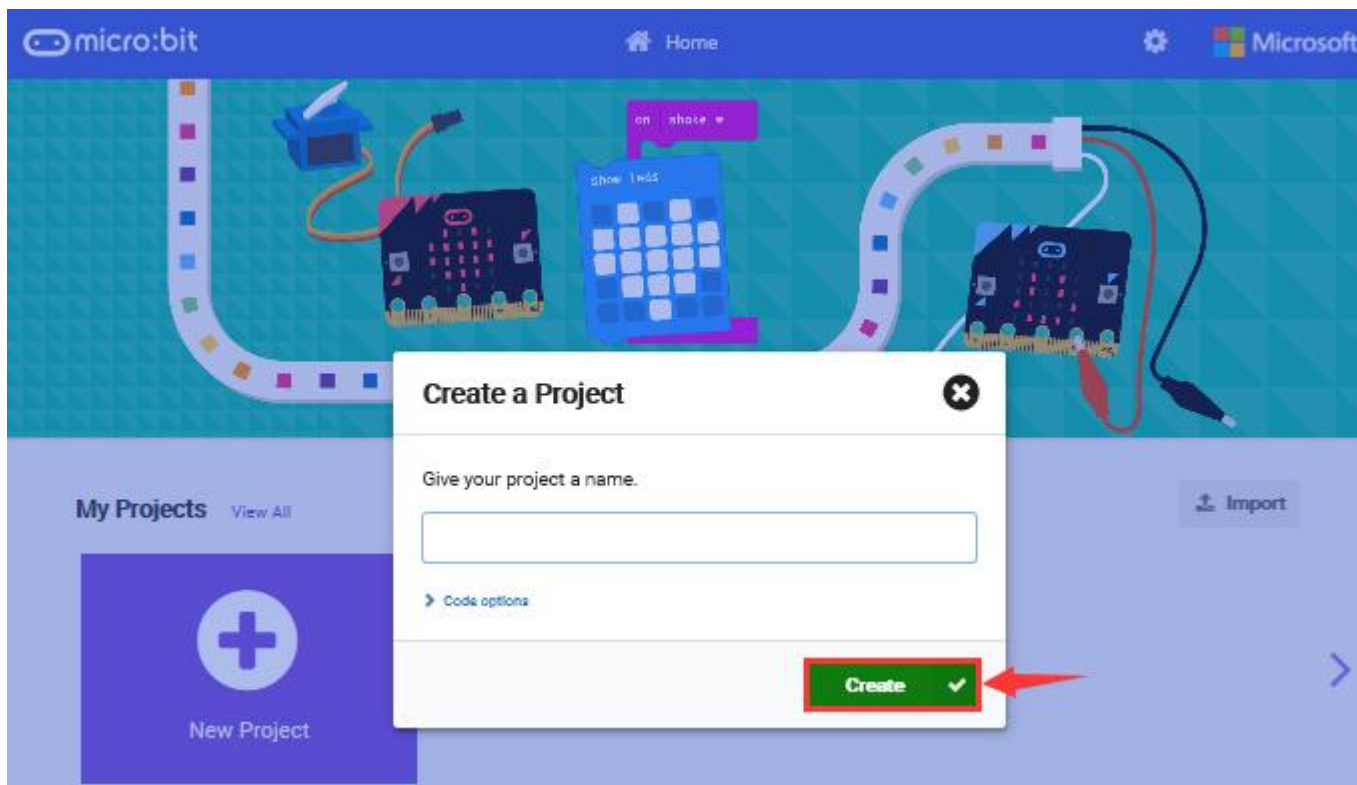
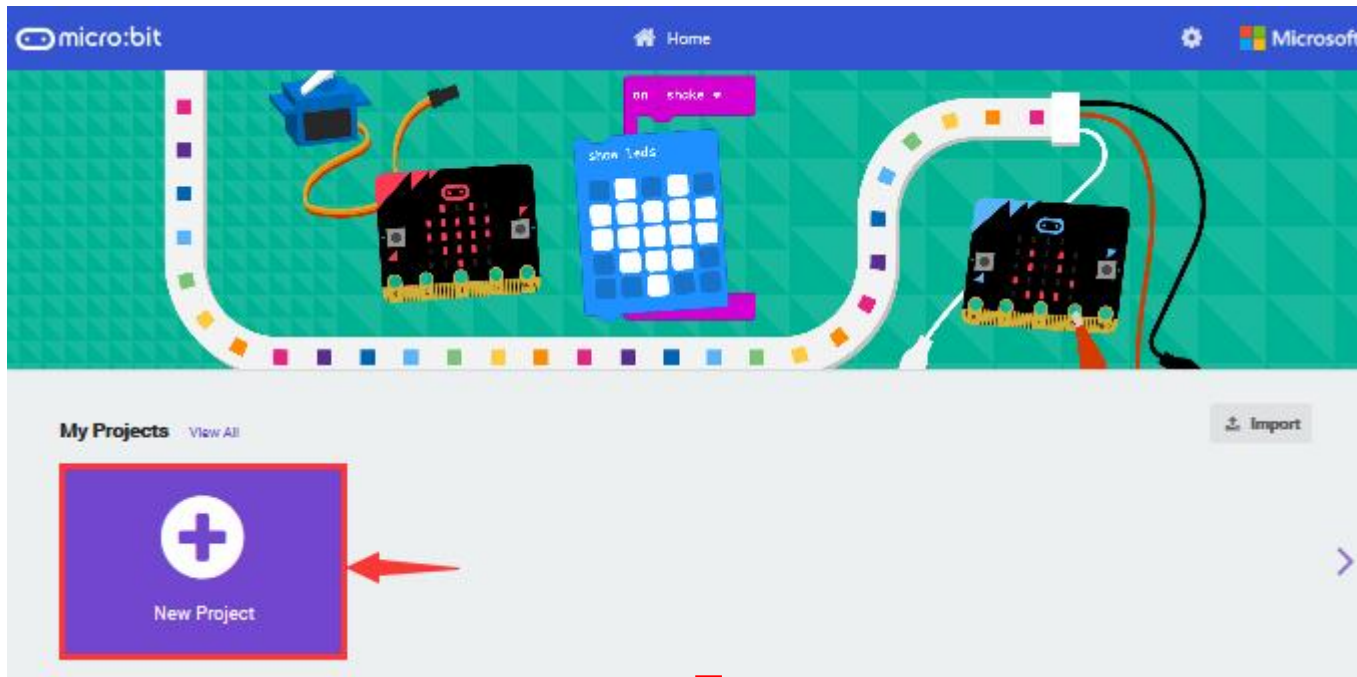


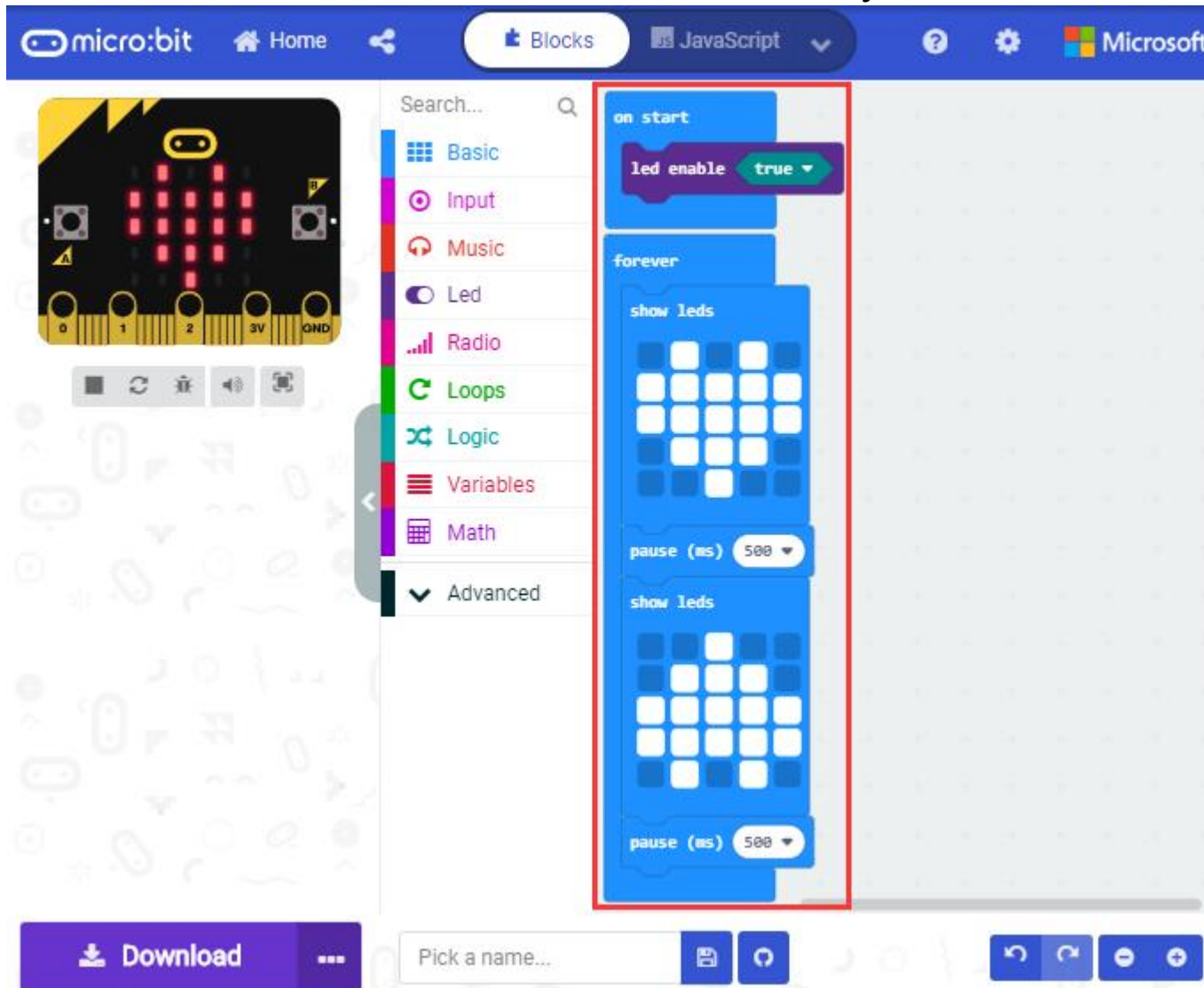
After the match with Bluetooth, write and upload code with the App.

Click "Create Code" to enter the programming page and write code.

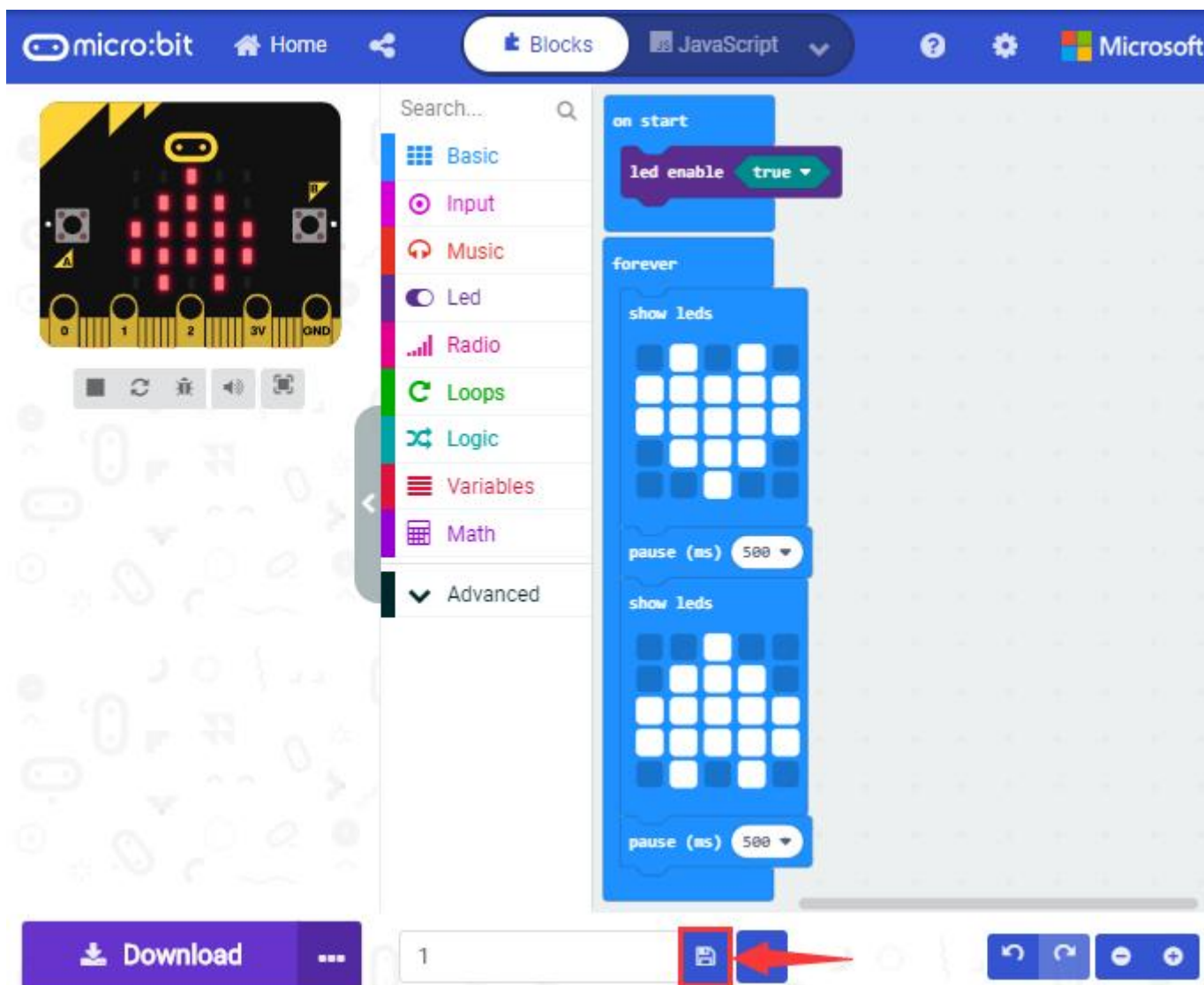


keystudio

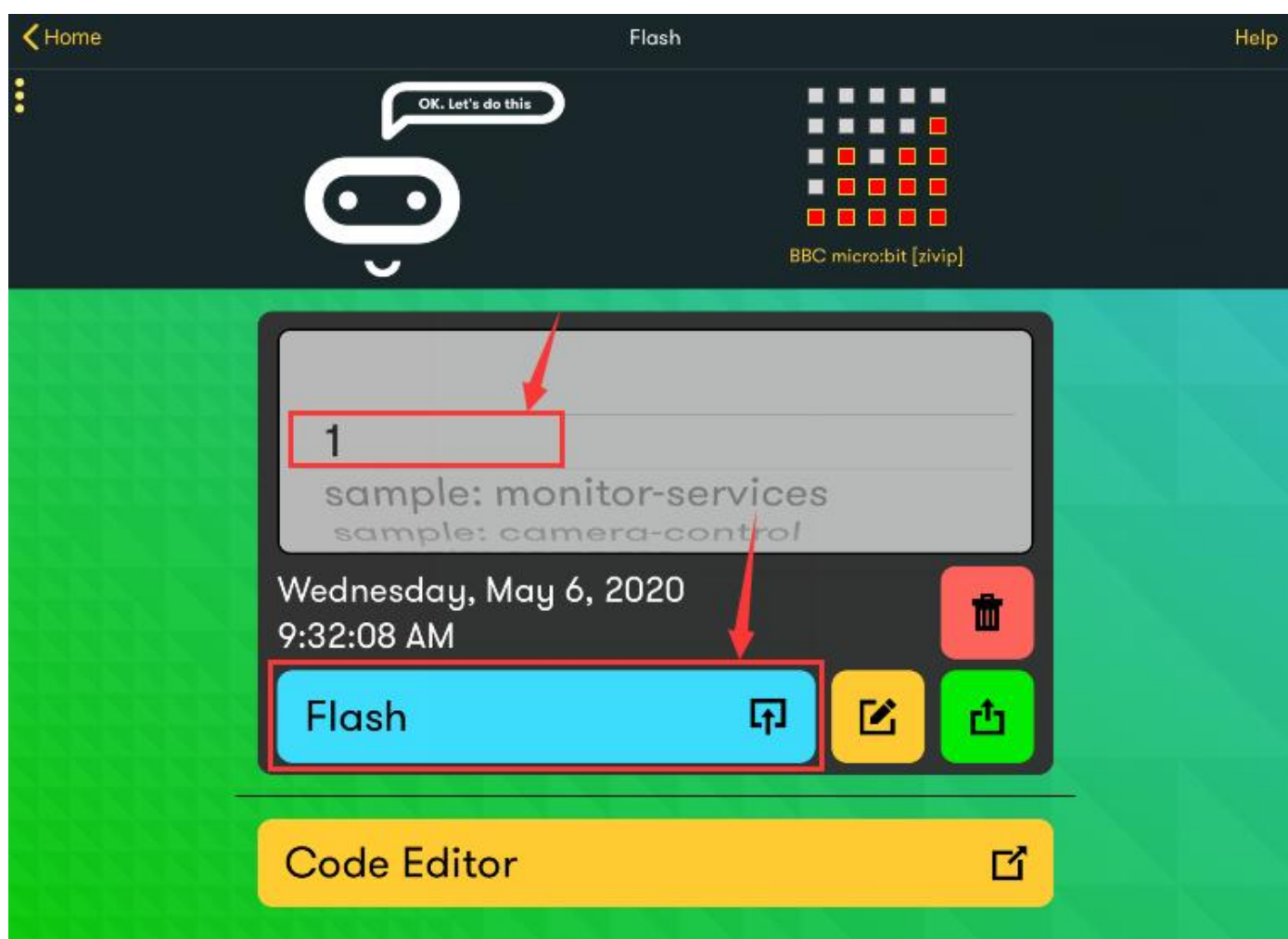
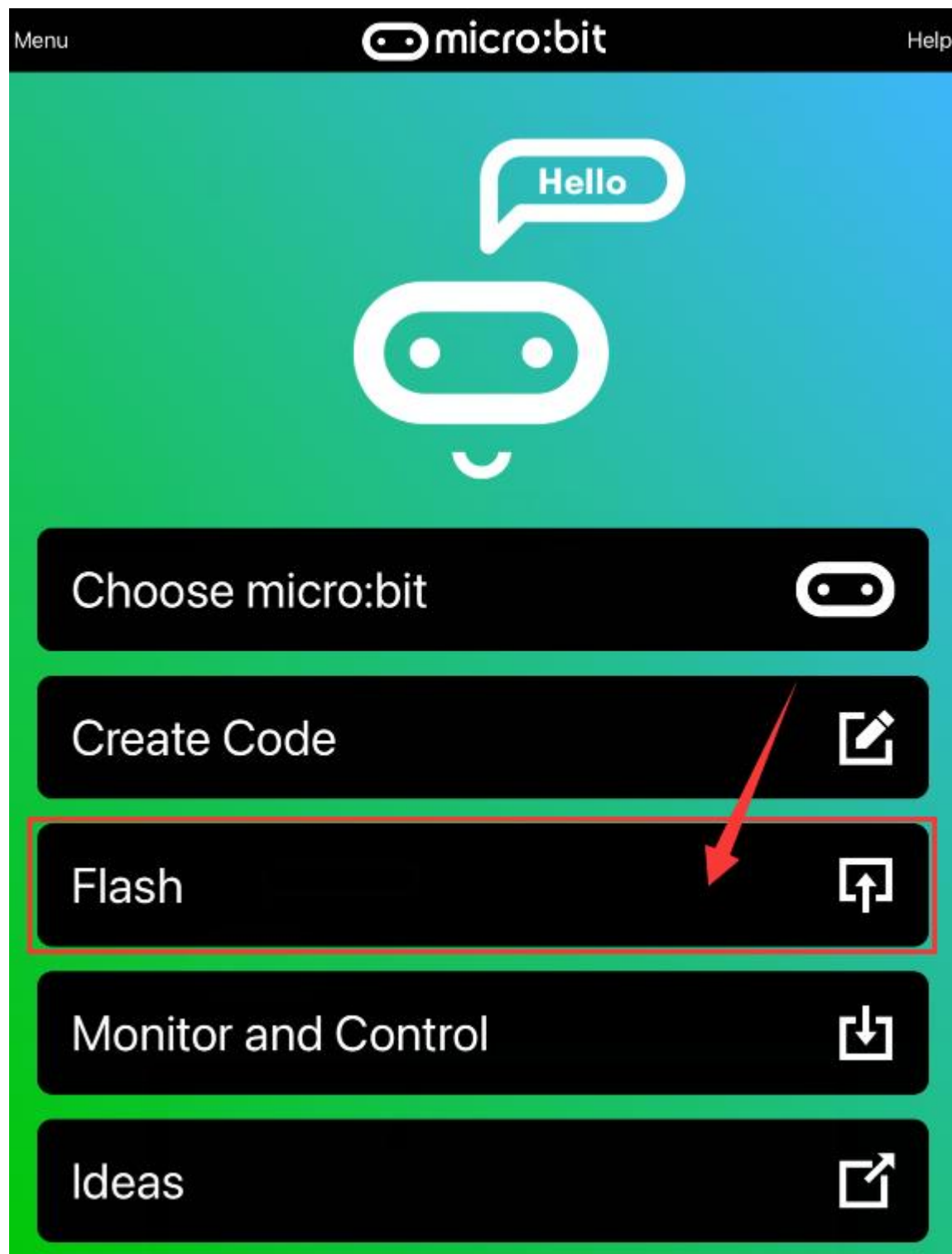


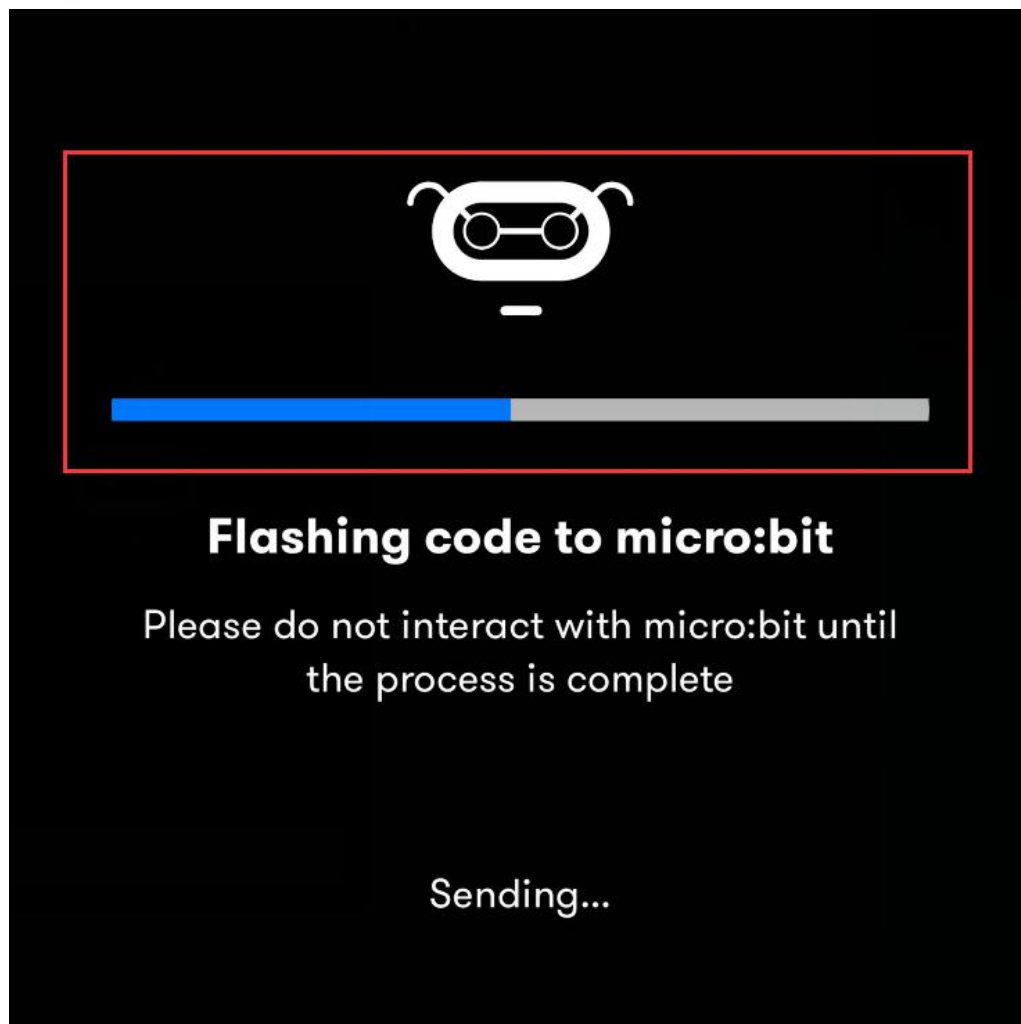


Name the code as "1" and click  to save it.

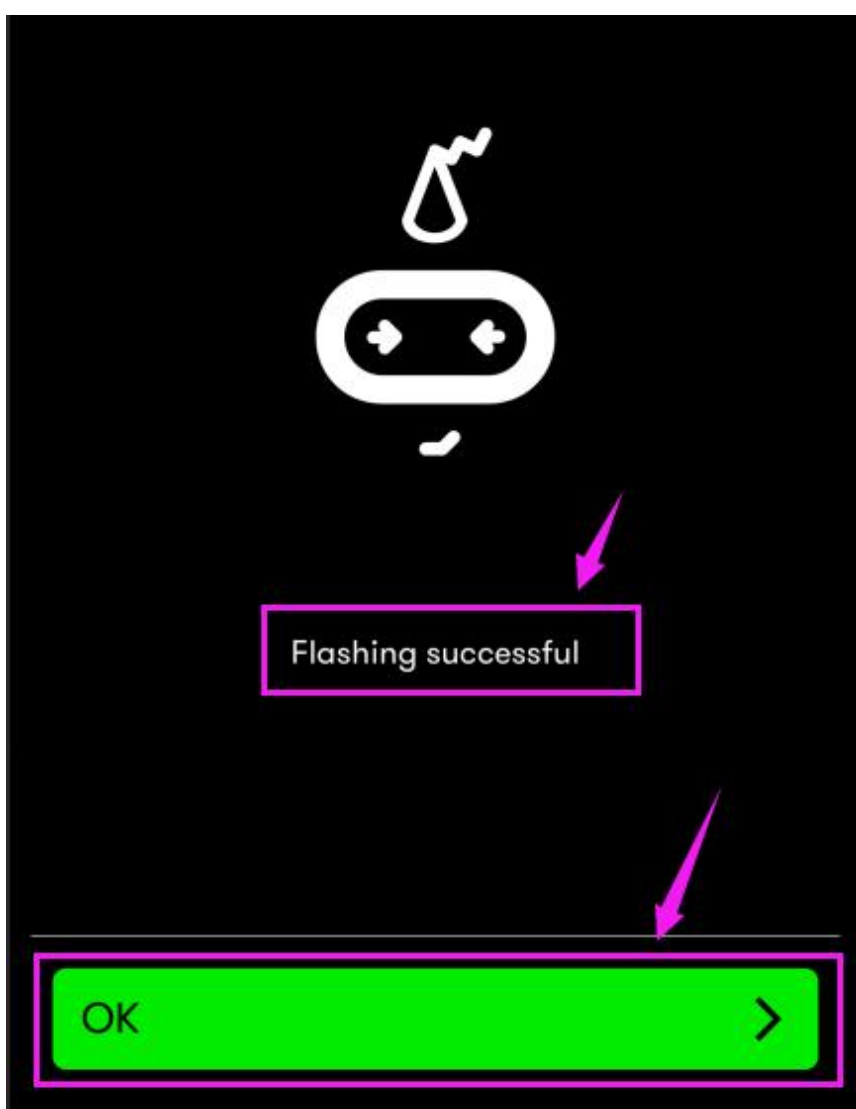


Click the third item "Flash" to enter the uploading page. The default code program for uploading is the one saved just now and named "1" and then click the other "Flash" to upload the code program "1".





If the code is uploaded successfully a few seconds later, the App will emerge as below and the LED dot matrix of the Micro: Bit main board V2 will exhibit a heart pattern.



Projects above all conduct with the built-in sensors and the LED dot matrix of the main board while the following ones will carry out with the help of external sensors of this turtle car. **(Attention: In order to avoid burning the the Micro:bit main board V2, please remove the USB cable and the external power from the board before fix it with the shield of the car; likewise, the USB cable and the external power should be cut from the main board before disconnect the shield from the board.)**

Project 13: Seven-Color LED



1. Description

This module consists of a commonly used LED with 7colors but in white appearance. It can automatically flash different colors to create fantastic light effects when high level is input like a normal LED.

2. Preparation

- Insert the micro:bit board into the slot of keyestudio 4WD Mecanum Robot Car V2.0
- Place batteries into battery holder
- Dial power switch to ON end
- Connect the micro:bit to your computer via an USB cable
- Open the Web version of Makecode.

Import Hex profile (How to import?) , or click “New Project” and drag blocks step by step(add MecanumRobot extension library first)

(How to add Mecanum_Robot extension?)

3. Test Code

Code1:

Make the RGB light flash 7 lights alternatively.

Code path:

File Type	Path	File Name

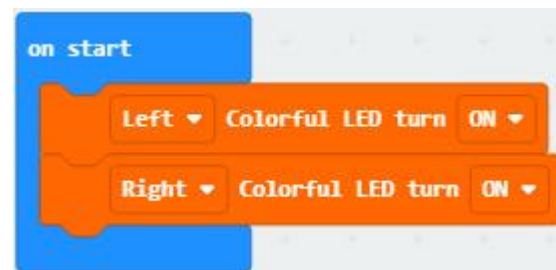
keyestudio

Hex file	..\2. Makecode Tutorial\Makecode Code	Project 13: Colorful Lights-1.hex
----------	--	--------------------------------------

Or you could edit code step by step in the editing area.

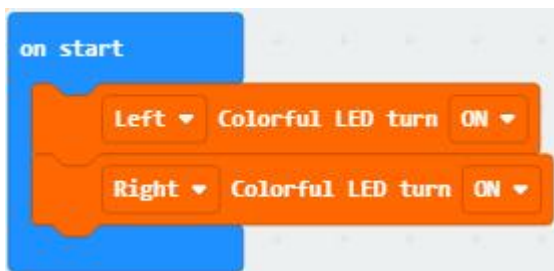
(1) Click "MecanumRobotV2" → find and drag  to "on start" ;

Copy  once;



Click the little triangle behind "Left" to choose "Right" :

Complete Program:

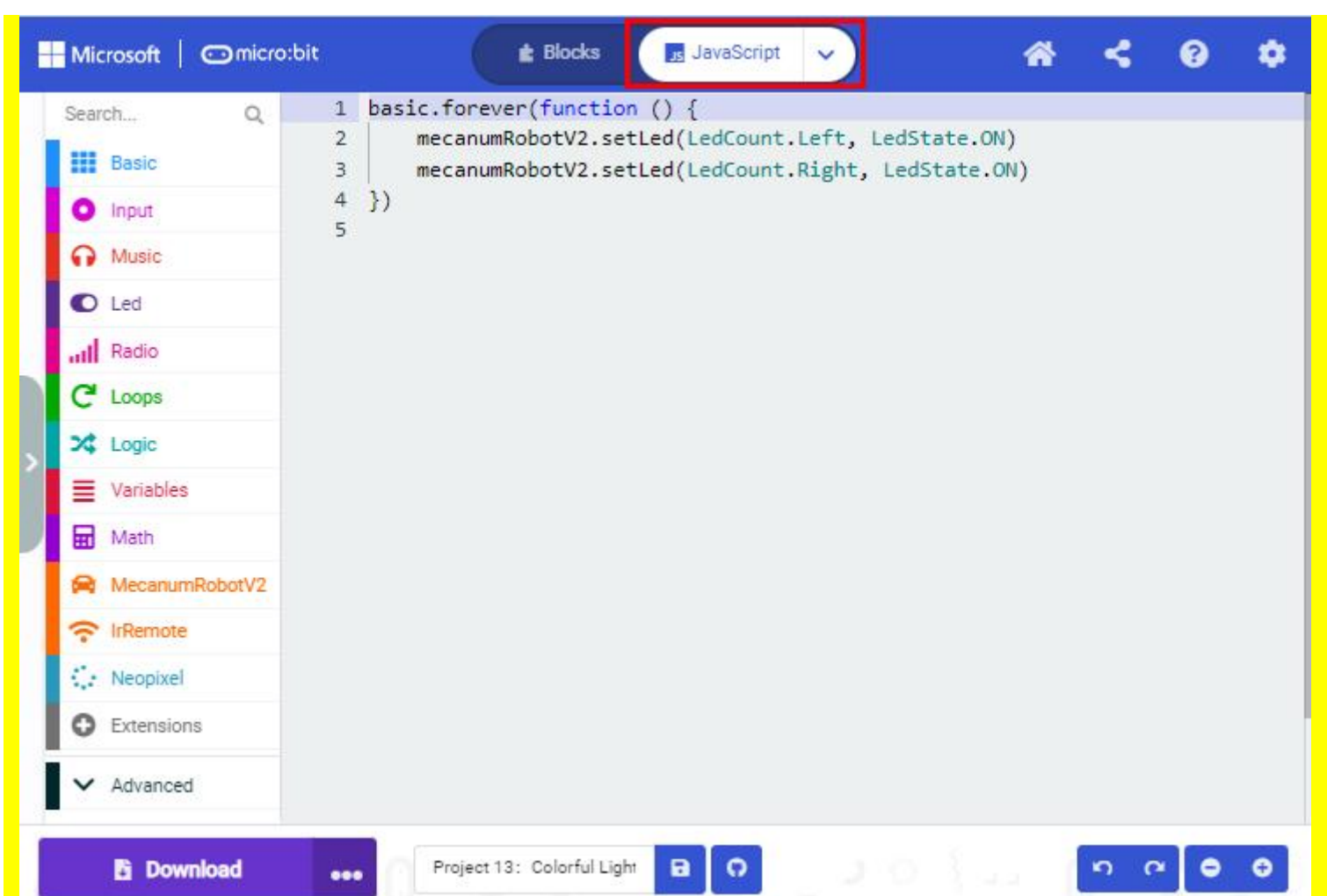


.....①run "on start" once to start the program

.....②open the colorful light on the left of the car

.....③open the colorful light on the right of the car

Click "JavaScript" to view the corresponding JavaScript code: :



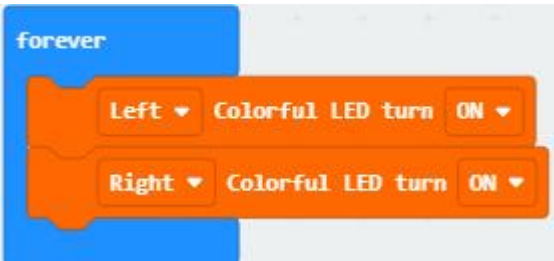
Code2:

File Type	Path	File Name
Hex file	..\2. Makecode Tutorial\Makecode Code	Project 13: Colorful Lights-2.hex

Or you could edit code step by step in the editing area.

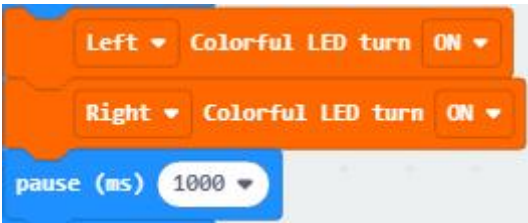
(1) click "MecanumRobotV2" to find and drag  to "forever" ;

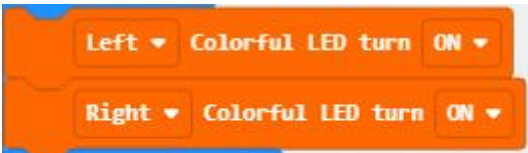
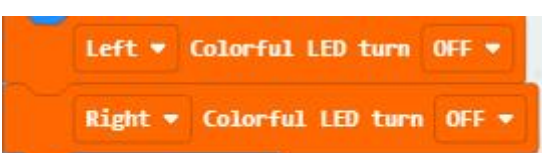
Copy  once;

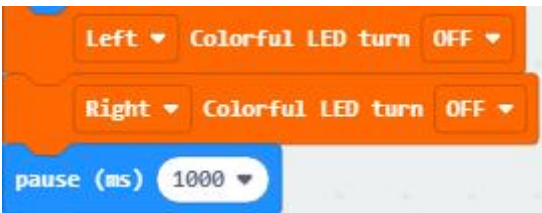


Click the little triangle behind "Left" to choose "Right" :

(2) Click "Basic" to find and drag  and tap the little triangle to choose "1 second"  ;



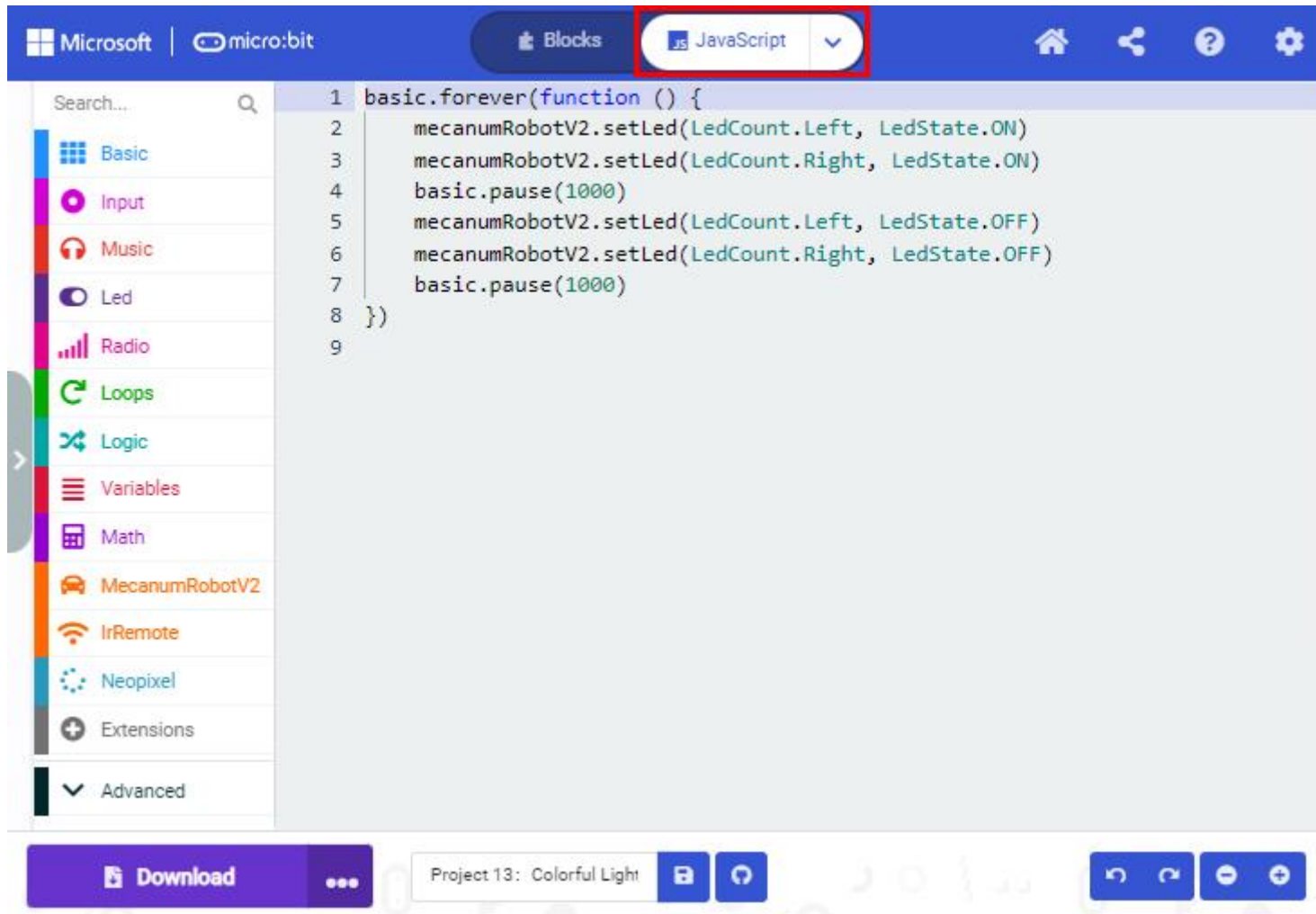
Copy  once and choose OFF  and put them in forever.



Complete Program:



- ① In the "forever" instruction block, the program runs cyclically.
- ②Turn on the 2 colorful lights of the car.
- ③Wait for 1 second
- ④Turn off the 2 colorful lights of the car.
- ⑤Wait for 1 second



4. Test Result

Download code 1 to micro:bit board and dial POWER switch to ON end, 2 RGB lights of smart car emit red, green, blue, indigo, dark red, yellow and white color cyclically.

Download code 2 to micro:bit board, 2 RGB lights will flash for 1 second and then stop flashing for 1 second, cyclically.

(How to download? How to quick download?)

Project 14: 4 WS2812 RGB LEDs



1. Description

The driver shield cooperates 4 pcs WS2812 RGB LEDs, compatible with micro:bit board and controlled by P7. In this lesson, we will make the RGB LEDs display different colors by P7. In this lesson, 3 sets of test code are provided to make the 4 WS2812 RGB LEDs display different effects.

2. Preparation

- Insert the micro:bit board into the slot of keyestudio 4WD Mecanum Robot Car V2.0

- Place batteries into battery holder
- Dial power switch to ON end
- Connect the micro:bit to your computer via an USB cable
- Open the Web version of Makecode.

Import Hex profile (How to import?) , or click “New Project” and drag blocks step by step(add MecanumRobot extension library first)

(How to add Mecanum_Robot extension?)

3. Test Code

Code 1:

File Type	Path	File Name
Hex file	..\2. Makecode Tutorial\Makecode Code	Project 14: WS2812 RGB LEDs-1.hex

Or you could edit code step by step in the editing area.

- a. Enter “Neopixel” → “set strip to Neopixel at pin P0 with 24 leds as RGB (GRB format)”
- b. Place it into “on start” block,
- c. Signal end P7 of WS2812 RGB is controlled by P7 of micro:bit . So we set to P7.
- d. Smart car has 4 pcs WS2812 RGB lights, so set to 4 leads

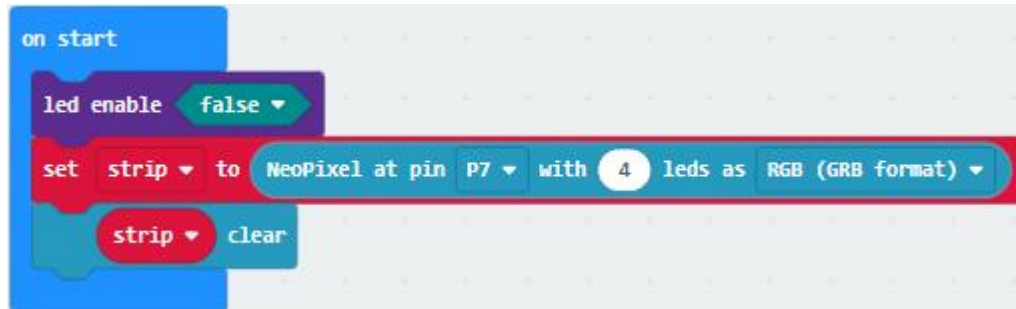


Click “Neopixel” to move block “strip clear” into “on start” block.

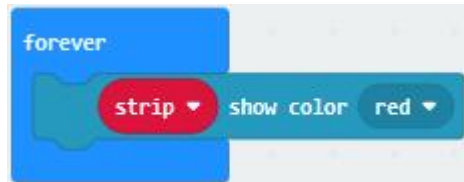


Click “Led” → “more” → “led enable(false)” and place it into “on start” block,

keystudio



Enter "Neopixel" to move block "strip show color red" into "forever" block



Click "Basic" to move "pause (ms) 100" block into "forever" block

Then set the delay to 1000ms

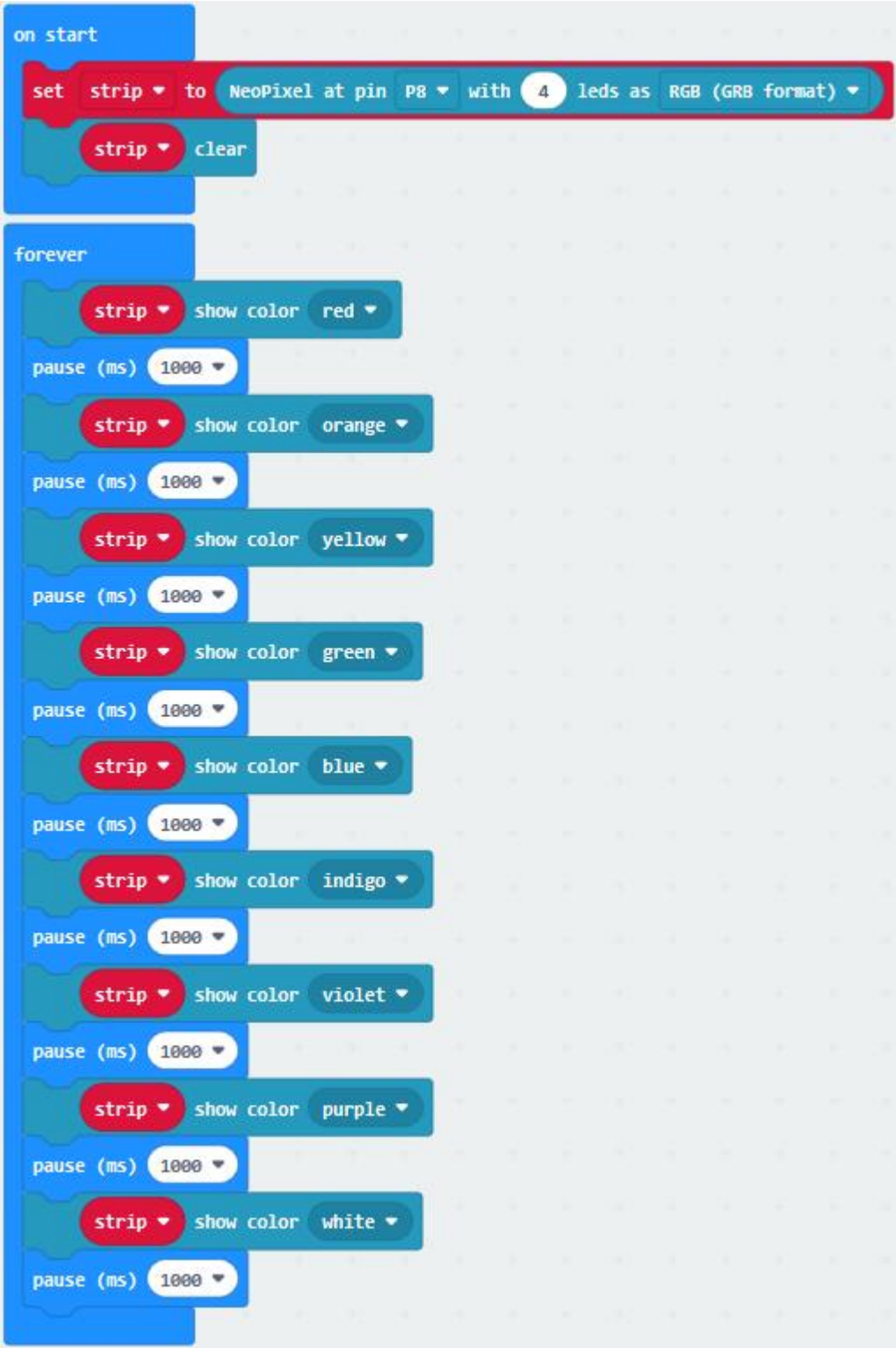


Copy code string `strip show color red` for eight times, and click red to respectively set to orange, yellow, green, blue, indigo, violet, purple and white.

Tap the triangle icon to select orange, yellow, green, blue, indigo, violet, purple and white.

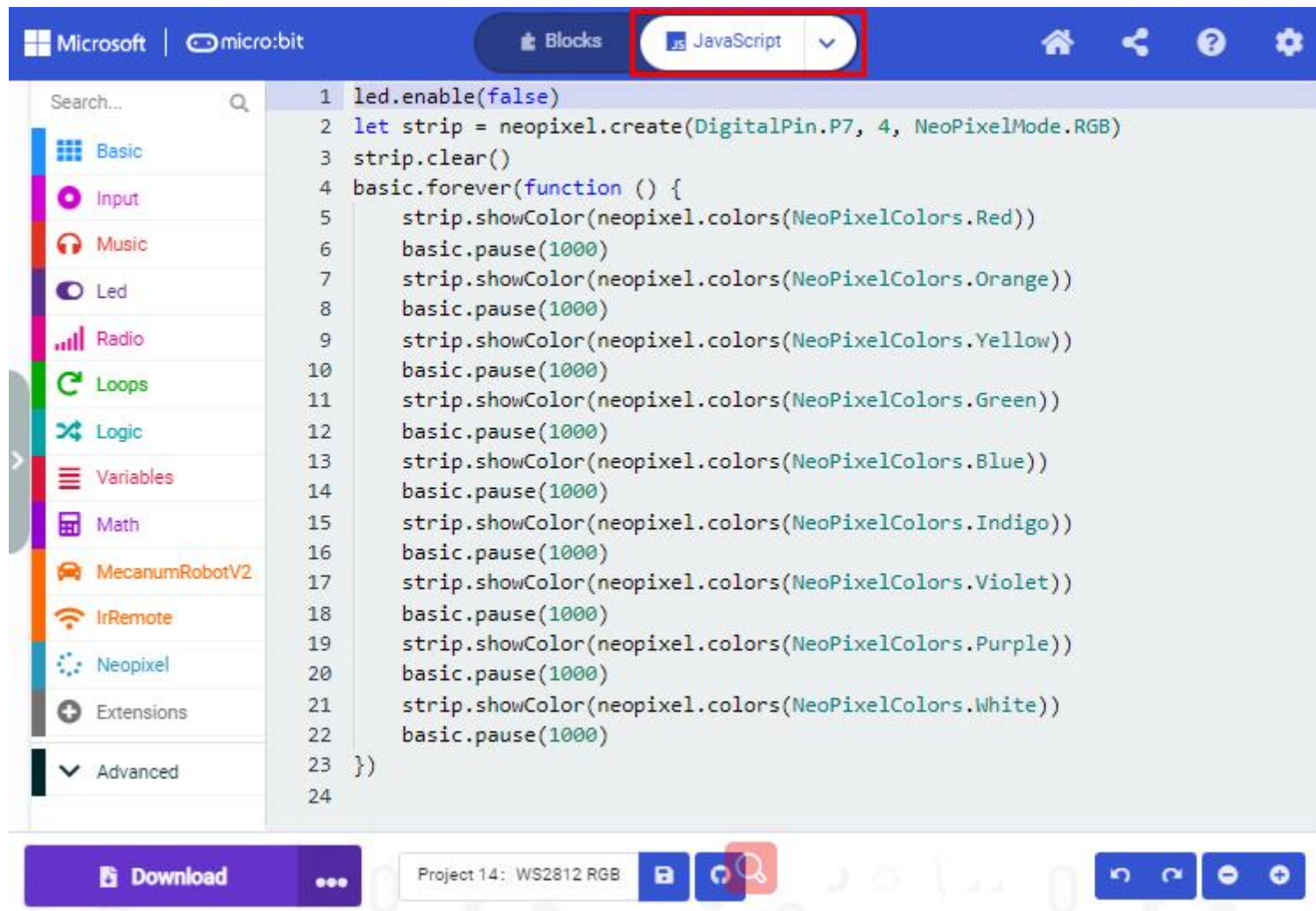


Complete Code:



. "on start" : command block runs once to start program.
Set strip to Neopixel at pin P7 with 4 leads as RGB
Turn off 4pcs WS2812 RGB lights
The program under the block "forever" runs cyclically.
All RGB lights show red color
Delay in 1000ms
All RGB lights show orange color
Delay in 1000ms
All RGB lights show yellow color
Delay in 1000ms
All RGB lights show green color
Delay in 1000ms
All RGB lights show blue color
Delay in 1000ms
All RGB lights show indigo color
Delay in 1000ms
All RGB lights show violet color
Delay in 1000ms
All RGB lights show purple color
Delay in 1000ms
All RGB lights show white color
Delay in 1000ms

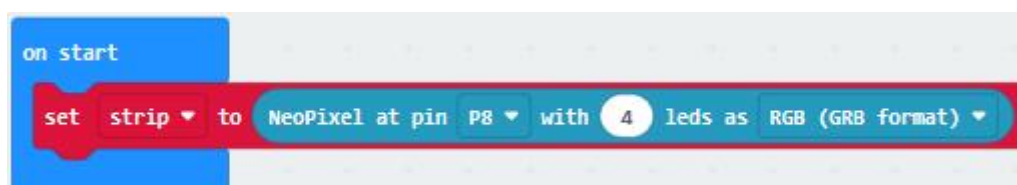
Click "JavaScript" to switch into the corresponding JavaScript code:



Code 2:

File Type	Path	File Name
Hex file	..\2. Makecode Tutorial\Makecode Code	Project 14: WS2812 RGB LEDs-2.hex

- Enter "Neopixel" → "set strip to Neopixel at pin P0 with 24 leds as RGB (GRB format)"
- Place it into "on start" block,
- Signal end P7 of WS2812 RGB is controlled by P7 of micro:bit . So we set to P7.
- Smart car has 4 pcs WS2812 RGB lights, so set to 4 leads



Click "Led" → "more" → "led enable(false)" and place it into "on start" block,



Click "Loops" to drag "for index from 0 to 4...do" into "forever" block

Change 4 into 3



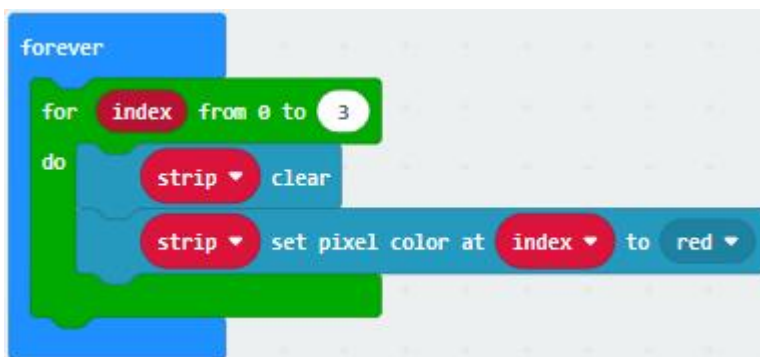
Click "Neopixel" to move block "strip clear" into block "for index from 0 to 3...do"



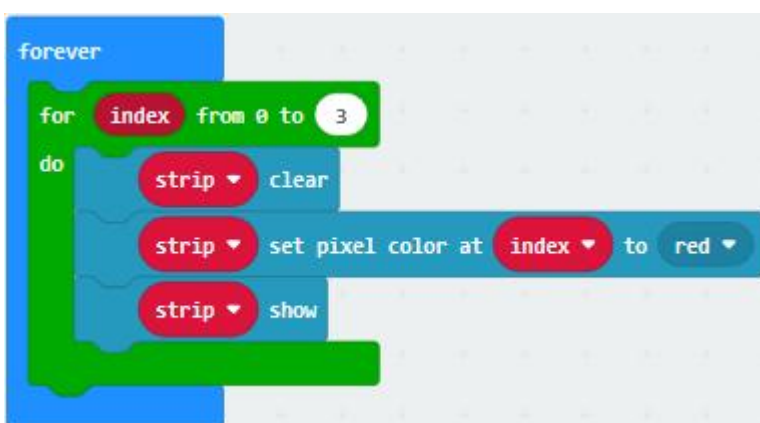
Tap "Neopixel" → "more" → "strip set pixel color at 0 to red"

Place it into "for index from 0 to 3...do" block

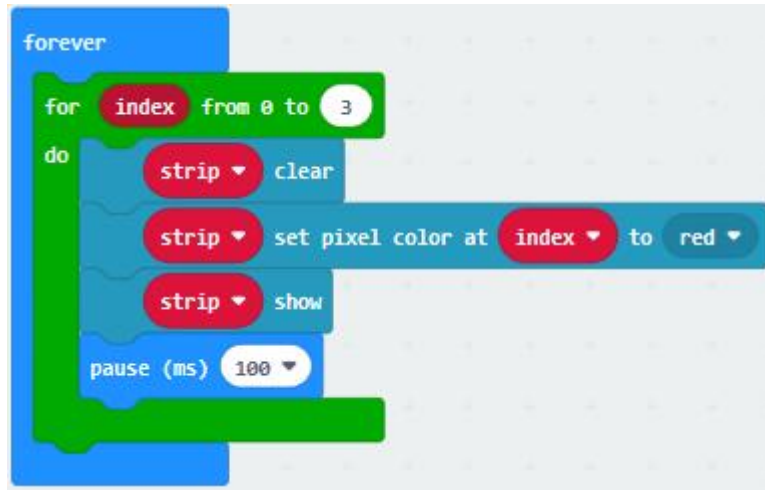
Click "Variables" to move "index" into 0 box



Click "Neopixel" to move "strip show" into "for index from 0 to 3...do" block

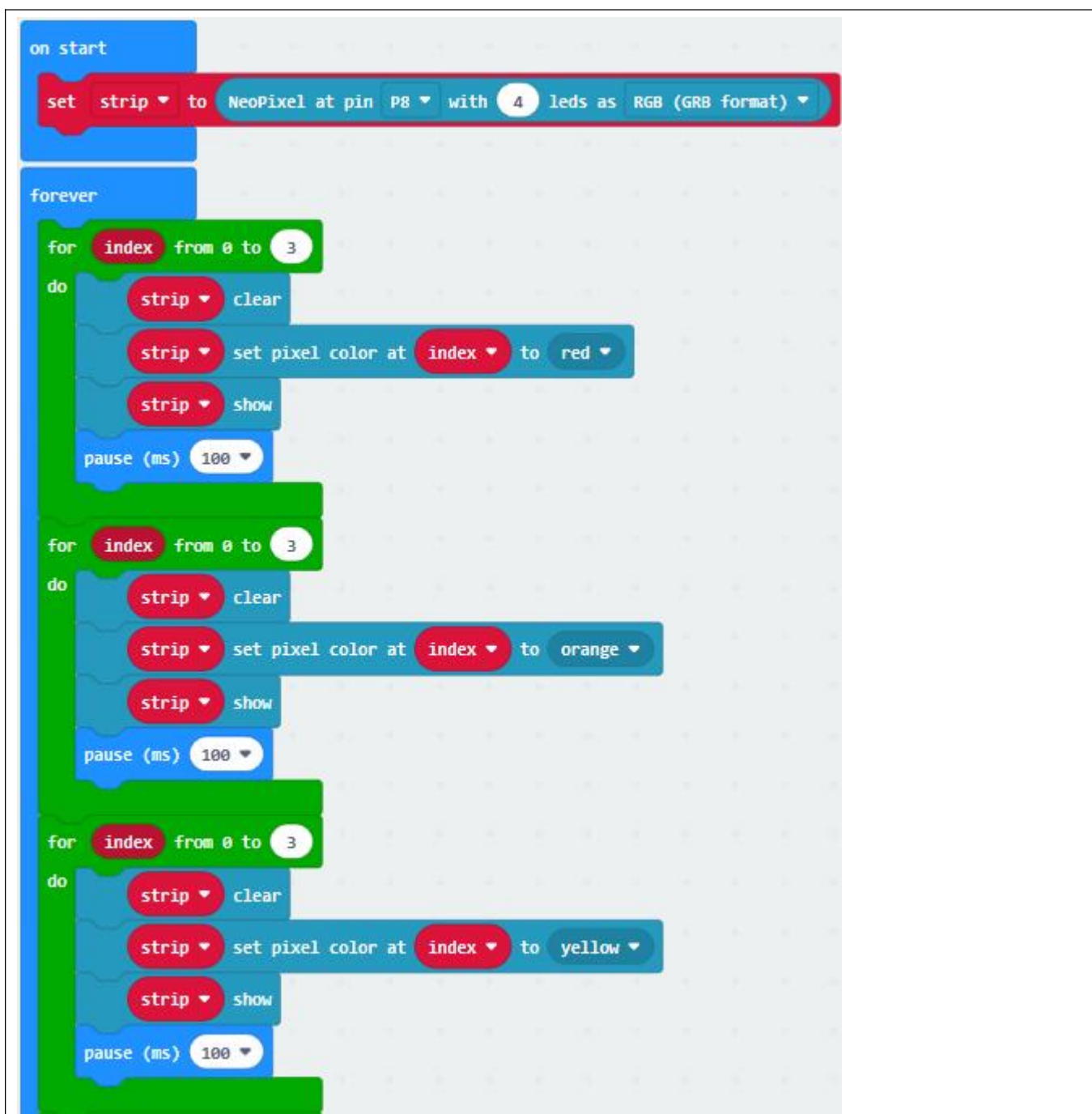


Tap "Basic" to move "pause (ms) 100" block into "index from 0 to 3...do" and set the delay to 100ms.



Replicate code string for eight times and place them into “forever” block.
Click red to respectively choose orange, yellow, green, blue, indigo, violet, purple and white.

Complete Code:



"on start" : command block runs once to start program.

Set strip to Neopixel at pin p7 with 4 leads as RGB

The program under the block "forever" runs cyclically.

For index from 0 to 3, execute the program under do block

Turn off 4 pcs WS2812 RGB lights

Set index of WS2812 RGB lights to red color

Strip shows

Delay in 100ms

For index from 0 to 3, execute the program under do block

Turn off 4 pcs WS2812 RGB lights

Set index of WS2812 RGB lights to orange color

Strip shows

Delay in 100ms

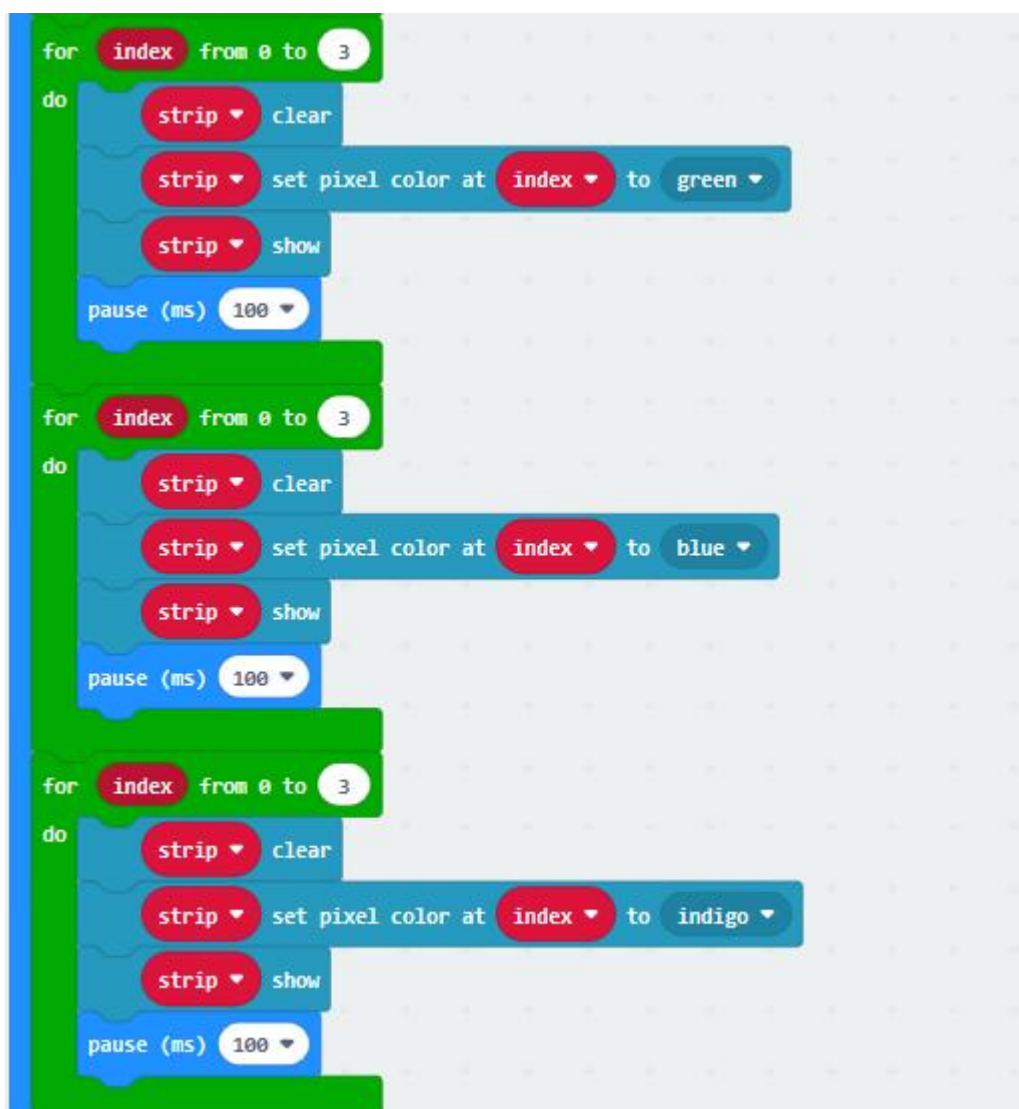
For index from 0 to 3, execute the program under do block

Turn off 4 pcs WS2812 RGB lights

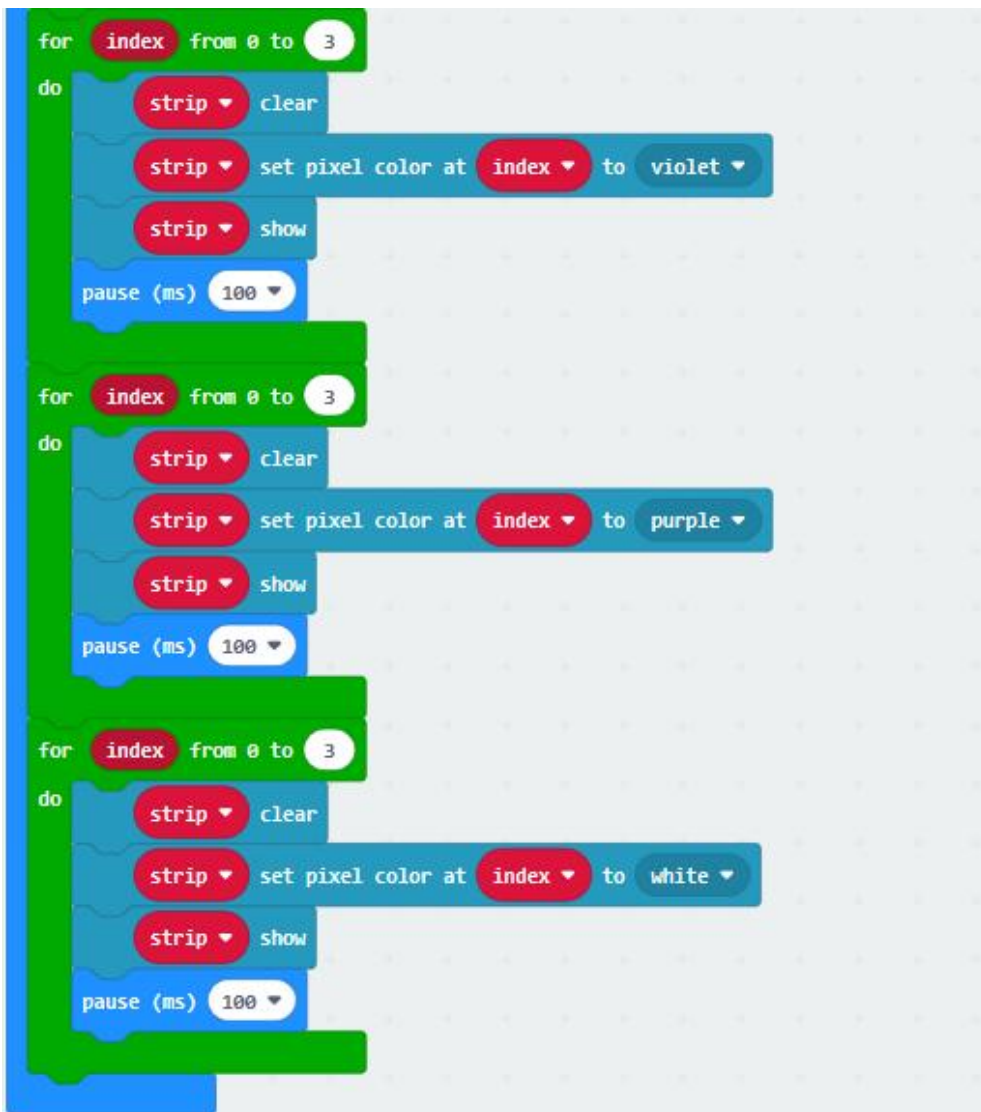
Set index of WS2812 RGB lights to yellow color

Strip shows

Delay in 100ms



.For index from 0 to 3, execute the program under do block
Turn off 4 pcs WS2812 RGB lights
Set the index of WS2812 RGB lights to green color
Strip shows
Delay in 100ms
For index from 0 to 3, execute the program under do block
Turn off 4 pcs WS2812 RGB lights
Set the index of WS2812 RGB lights to blue color
strip shows
Delay in 100ms
For index from 0 to 3, execute the program under do block
Turn off 4 pcs WS2812 RGB lights
Set the index of WS2812 RGB lights to indigo color
Strip shows
Delay in 100ms
For index from 0 to 17, execute the program under do block
Turn off all RGB on strip
Set the index of WS2812 RGB lights to violet color
Set all RGB lights to show violet color
Strip displays all changes
Delay in 100ms
For index from 0 to 17, execute the program under do block
Turn off all RGB on strip
Set the index of WS2812 RGB lights to purple color
Strip displays all changes
Delay in 100ms
For index from 0 to 17, execute the program under do block
Turn off all RGB on strip
Set the index of WS2812 RGB lights to white color
Strip displays all changes
Delay in 100ms



For index from 0 to 3, execute the program under do block
 Turn off 4 pcs WS2812 RGB lights
 Set the index of WS2812 RGB lights to violet color
 Strip shows
 Delay in 100ms
 For index from 0 to 3, execute the program under do block
 Turn off 4 pcs WS2812 RGB lights
 Set the index of WS2812 RGB lights to purple color
 Strip shows
 Delay in 100ms
 For index from 0 to 3, execute the program under do block
 Turn off 4 pcs WS2812 RGB lights
 Set the index of WS2812 RGB lights to white color
 Strip shows
 Delay in 100ms

Click "JavaScript" to switch into the corresponding JavaScript code:

```

1 led.enable(false)
2 let strip = neopixel.create(DigitalPin.P7, 4, NeoPixelMode.RGB)
3 basic.forever(function () {
4   for (let index = 0; index <= 3; index++) {
5     strip.clear()
6     strip.setPixelColor(index, neopixel.colors(NeoPixelColors.Red))
7     strip.show()
8     basic.pause(100)
9   }
10  for (let index = 0; index <= 3; index++) {
11    strip.clear()
12    strip.setPixelColor(index, neopixel.colors(NeoPixelColors.Orange))
13    strip.show()
14    basic.pause(100)
15  }
16  for (let index = 0; index <= 3; index++) {
17    strip.clear()
18    strip.setPixelColor(index, neopixel.colors(NeoPixelColors.Yellow))
19    strip.show()
20    basic.pause(100)
21  }
22  for (let index = 0; index <= 3; index++) {
23    strip.clear()
24    strip.setPixelColor(index, neopixel.colors(NeoPixelColors.Green))
25    strip.show()
26    basic.pause(100)
27  }
28  for (let index = 0; index <= 3; index++) {
29    strip.clear()
30    strip.setPixelColor(index, neopixel.colors(NeoPixelColors.Blue))
31    strip.show()
32    basic.pause(100)
33  }
34  for (let index = 0; index <= 3; index++) {
35    strip.clear()
36    strip.setPixelColor(index, neopixel.colors(NeoPixelColors.Indigo))
37    strip.show()
38    basic.pause(100)
39  }
}

```

keystudio

```

40   for (let index = 0; index <= 3; index++) {
41       strip.clear()
42       strip.setPixelColor(index, neopixel.colors(NeoPixelColors.Violet))
43       strip.show()
44       basic.pause(100)
45   }
46   for (let index = 0; index <= 3; index++) {
47       strip.clear()
48       strip.setPixelColor(index, neopixel.colors(NeoPixelColors.Purple))
49       strip.show()
50       basic.pause(100)
51   }
52   for (let index = 0; index <= 3; index++) {
53       strip.clear()
54       strip.setPixelColor(index, neopixel.colors(NeoPixelColors.White))
55       strip.show()
56       basic.pause(100)
57   }
58 })
59

```

Code 3:

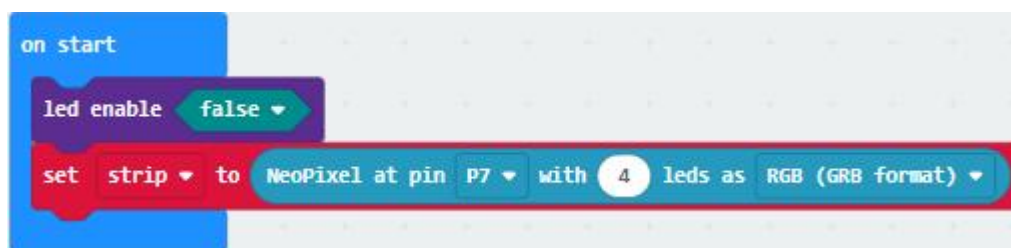
File Type	Path	File Name
Hex file	..\2. Makecode Tutorial\Makecode Code	Project 14: WS2812 RGB LEDs-3.hex

Or you could edit code step by step in the editing area.

- Enter "Neopixel" → "set strip to Neopixel at pin P0 with 24 leds as RGB (GRB format)"
- Place it into "on start" block,
- Signal end P7 of WS2812 RGB is controlled by P7 of micro:bit . So we set to P7.
- Smart car has 4 pcs WS2812 RGB lights, set to 4 leads



Click "Led" → "more" → "led enable(false)" and place it into "on start" block,



Click "Variables" → "Make a Variable..."

Input R to build up variable R

We create variable "G" and "B" in same way

Drag "set B to 0" into "on start" block

Copy "set B to 0" twice and click triangle button to choose G and B



Click "Loops" to get block "for index from 0 to 4...do"

Leave it into "forever" and change 4 into 3



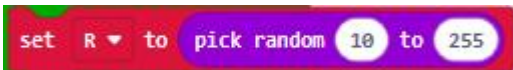
Move block "set B to 0" into "for index from 0 to 3...do" block,

Click B to choose R

Go to "Math" to drag block "pick random 0 to 10" into 0 box

Change 0 into 10, 10 into 255

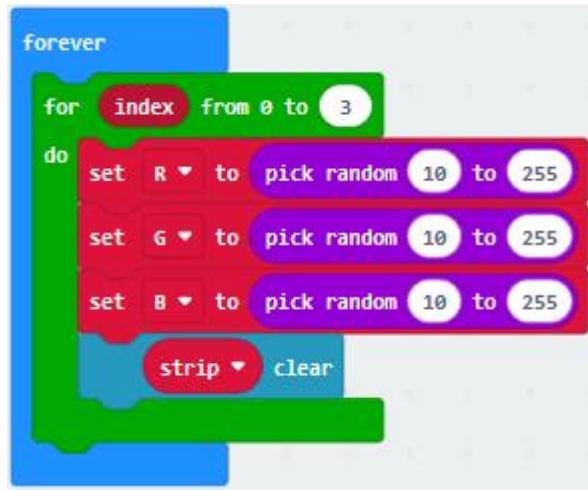


Replicate block  twice and place them into "for index from 0 to 3...do" block.

Click R to select G and B



Tap "Neopixel" and move "strip clear" into "for index from 0 to 3...do" block.



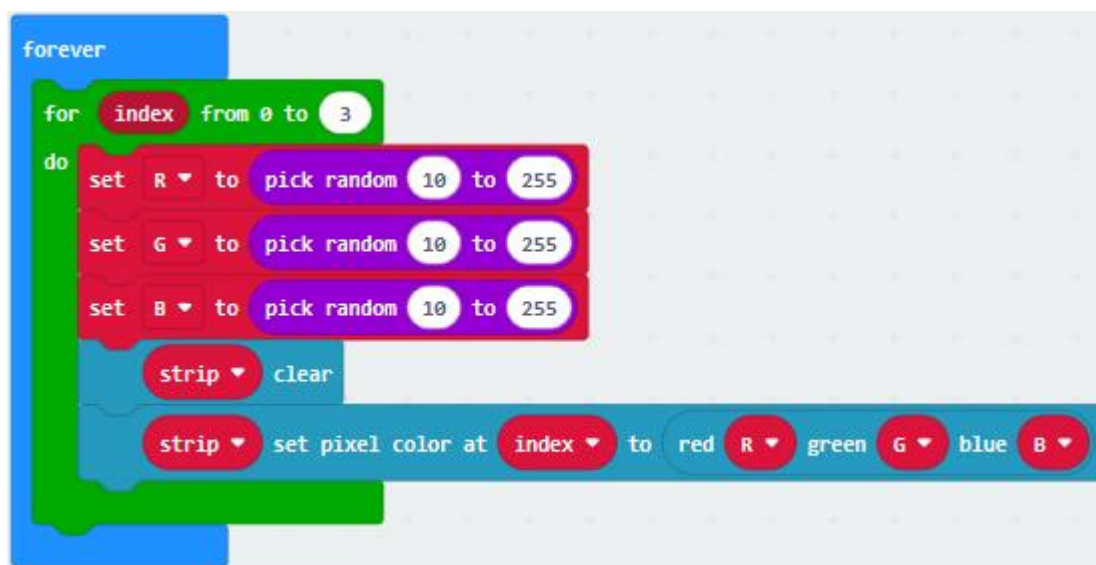
Go to "Neopixel" → "more" → "strip set pixel color at 0 to red"

Leave it in the block "for index from 0 to 3...do" block

Drag block "red 255 green 255 blue 255" into "red" box

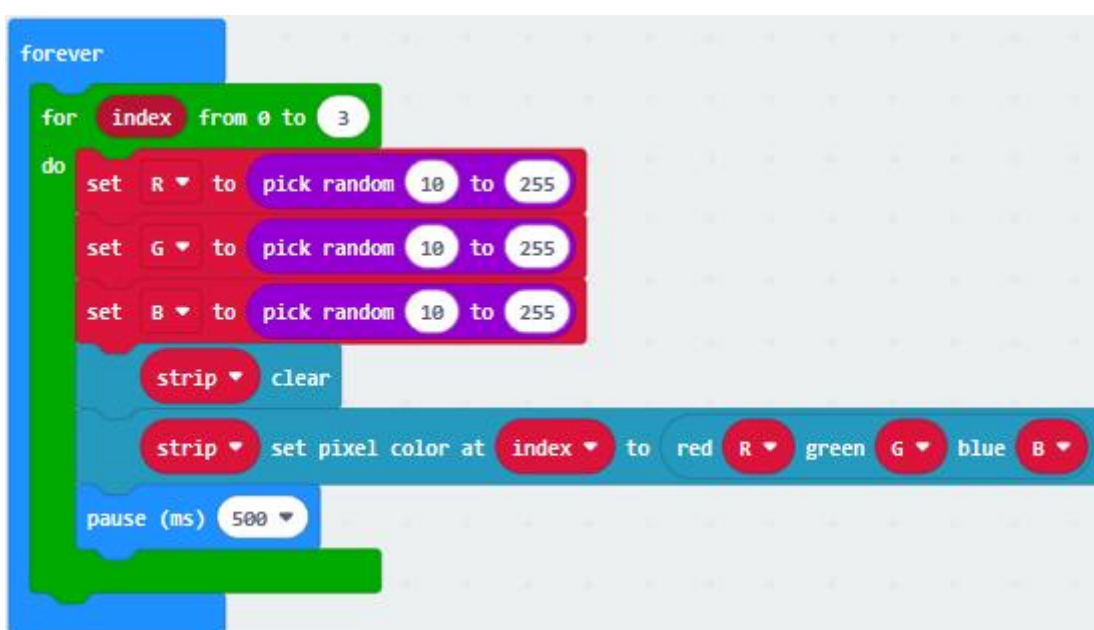
Tap "Variables" to move "index" block into 0 box

Separately drag R , G and B into 255 box, as shown below:

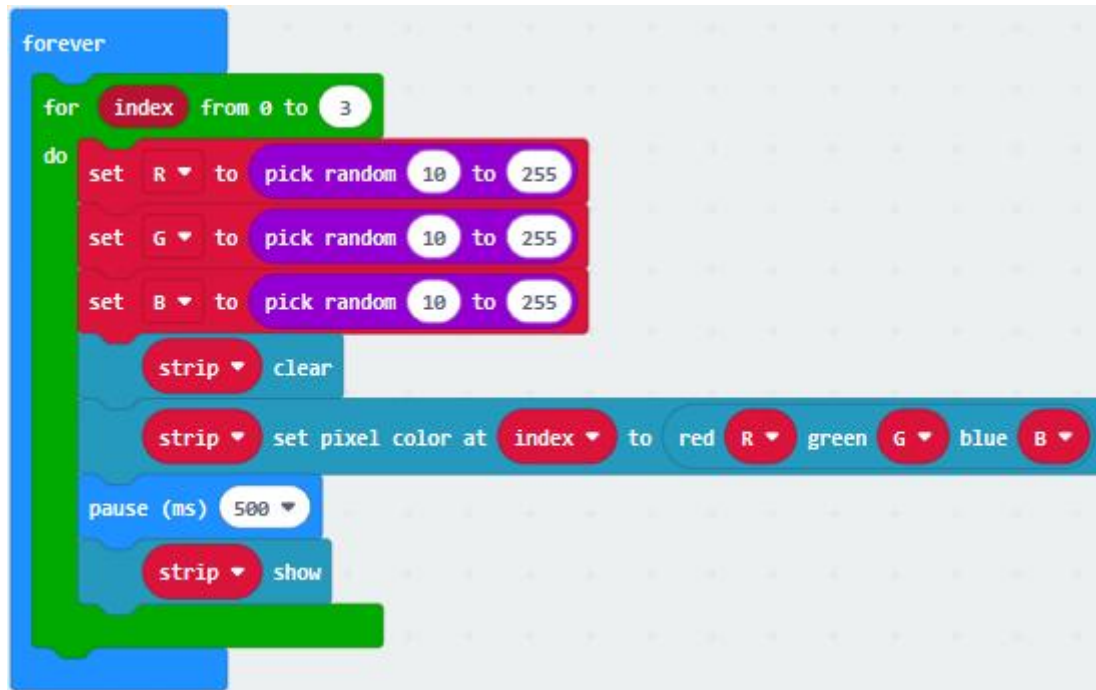


Click "Basic" to drag "pause (ms) 100" under block "strip.....B"

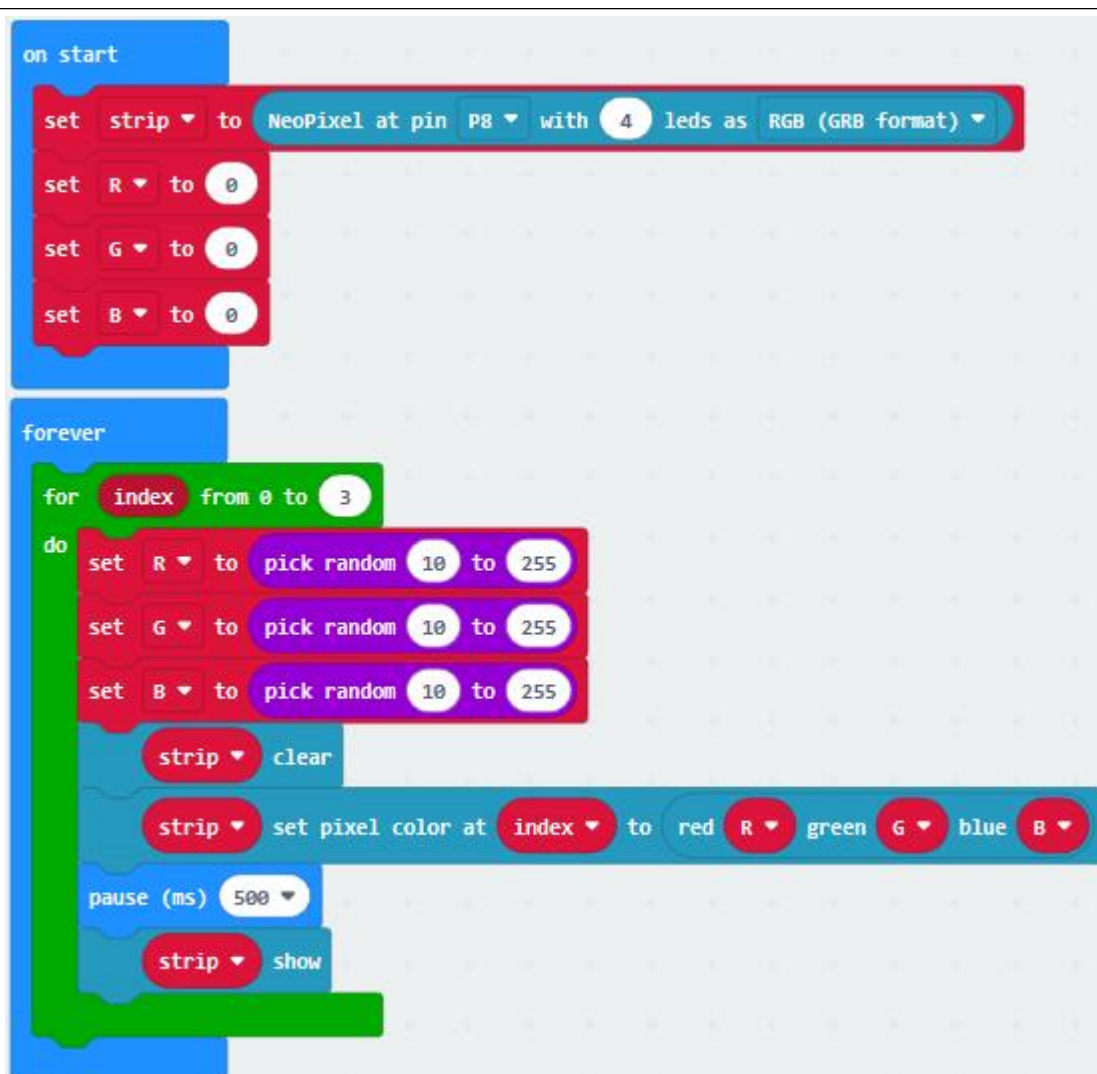
Set the delay to 500ms.



Click "Neopixel" to move "strip show" block under "pause(as) 500"



Complete Code:



“on start” : command block runs once to start program.

Set strip to Neopixel at pin p7 with 4 leads as RGB(GRB format)

Set variable R to 0

Set variable G to 0

Set variable B to 0

The program under the block “forever” runs cyclically.

When the value of index is in 0-3, execute the program under do block

Set variable R to random number in 10-255

Set variable G to random number in 10-255

Set variable B to random number in 10-255

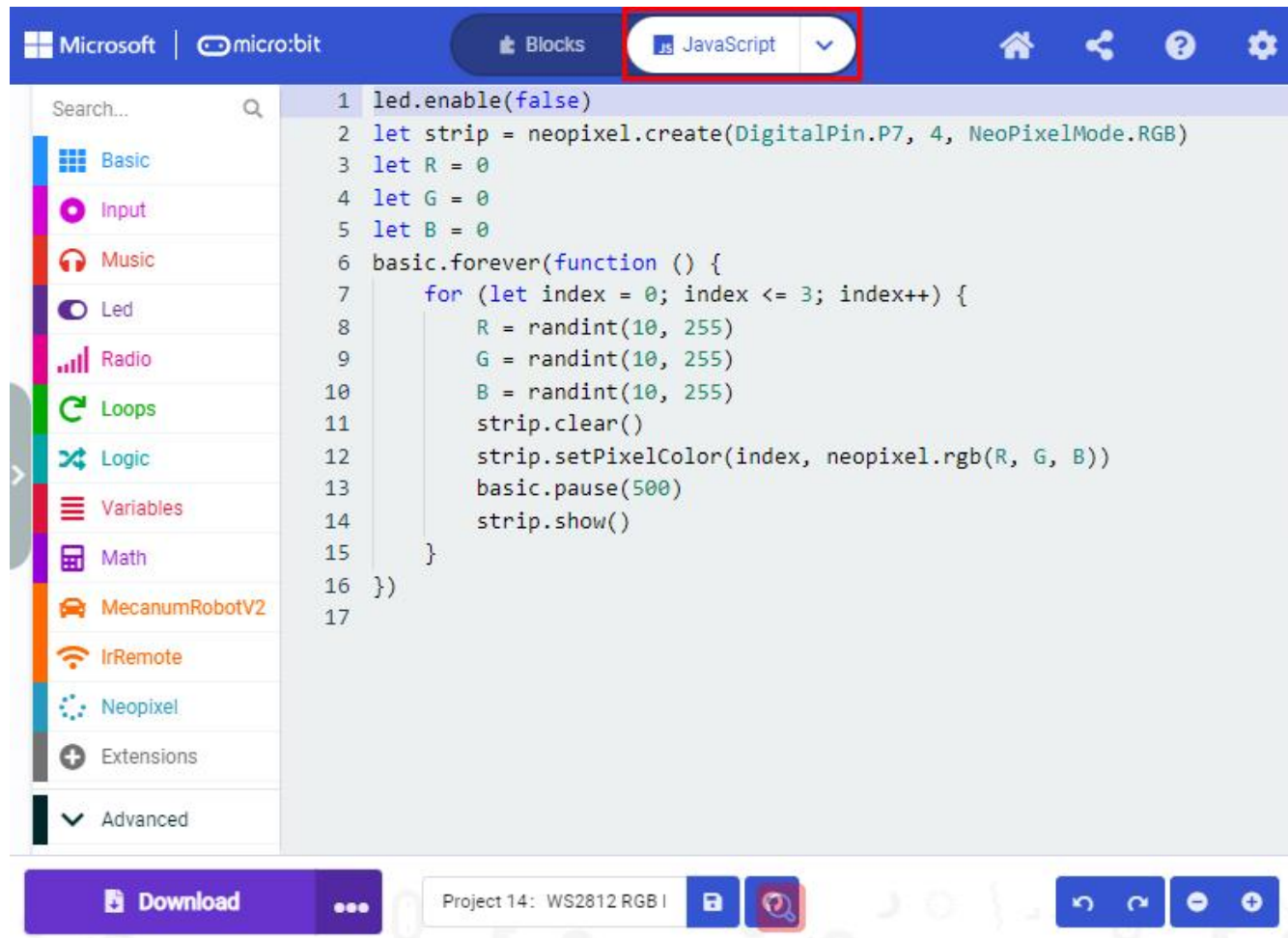
Turn off all RGB on strip

Set index of 4 pcs WS2812 RGB lights to RGB(red, green, blue)

Delay in 500ms

Strip shows

Click “JavaScript” to switch into the corresponding JavaScript code:



4. Test Result

Download code 1 to micro: bit, and dial POWER to ON end. All four WS2812RGB LEDs light up a different color a time cyclically.

Download code 2 to micro: bit, WS2812RGB LEDs display like flow light.

Download code 3 to micro: bit, every WS2812RGB light shows random color one by one.

(How to download? How to quick download?)

Project 15: Servo



1. Description

For those DIY smart cars, they often have the function of automatic obstacle avoidance. In the DIY process, we need to use a servo to control the ultrasonic module to rotate left and right, and then detect the distance between the car and the obstacle, so as to control the car to avoid the obstacle. If other microcontrollers are used to control the rotation of

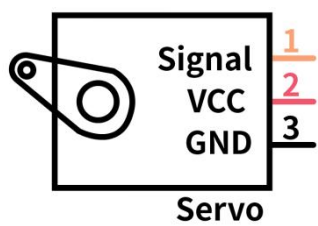
the servo, we need to set a certain frequency and a certain width of pulse to control the servo angle.

However, if the micro:bit main board is used to control the servo angle, we only need to set the control angle in the development environment where the corresponding pulse will be automatically set to control the servo rotation. In

this project, you will learn how to control the servo to rotate back and forth between 0° and 90°.

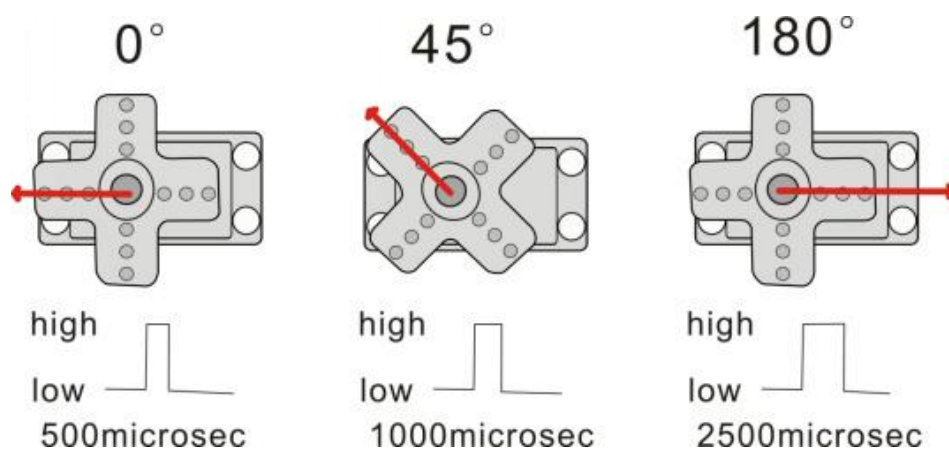
2. Information of the Servo

Servo motor is a position control rotary actuator. It mainly consists of housing, circuit board, core-less motor, gear and position sensor. Its working principle is that the servo receives the signal sent by MCU or receiver, 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 obtains the voltage difference output.



For the servo used in this project, the brown wire is the ground, the red one is the positive wire, and the orange one is the signal wire.

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 to the rotation angle from 0° to 180°. But note that for different brand motor, the same signal may have different rotation angle.



More details:

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

3. Parameters

- ◆ Working voltage: DC 4.8V ~ 6V
- ◆ Operating angle range: about 180 ° (at 500 → 2500 μsec)
- ◆ Pulse width range: 500 → 2500 μsec

keystudio

- ◆ No-load speed: 0.12 ± 0.01 sec / 60 (DC 4.8V) 0.1 ± 0.01 sec / 60 (DC 6V)
- ◆ No-load current: 200 ± 20mA (DC 4.8V) 220 ± 20mA (DC 6V)
- ◆ Stopping torque: 1.3 ± 0.01kg · cm (DC 4.8V) 1.5 ± 0.1kg · cm (DC 6V)
- ◆ Stop current: ≤ 850mA (DC 4.8V) ≤ 1000mA (DC 6V)
- ◆ Standby current: 3 ± 1mA (DC 4.8V) 4 ± 1mA (DC 6V)

4. Preparation

- Insert the micro:bit board into the slot of keystudio 4WD Mecanum Robot Car V2.0
- Place batteries into battery holder
- Dial power switch to ON end
- Connect the micro:bit to your computer via an USB cable
- Open the Web version of Makecode

Import Hex profile **(How to import?)** , or click “New Project” and drag blocks step by step(add MecanumRobot extension library first)

(How to add Mecanum_Robot extension?)

5. Test Code

File Type	Path	File Name
Hex file	..\2. Makecode Tutorial\Makecode Code	Project 15: Servo.hex

Or you could edit code step by step in the editing area.

(1) Click “Variables” ; motor “Make a Variable name” create a variable named “angle” ; set the value to 0 :



; and then put it into “on start”



;

keystudio



(2)Click "Loops" to find and drag to "forever" and change the number to "180" ; click



" MecanumRobotV2 " to find and drag and put variable " angle " into



; put into and change the digit 0 to



180 ;



Click of "Variable" and of "Math" ; put variable" angle" on the left and



change the number on the right to 1:



;put it into : ;Put



behind



and add delay in



10ms ;

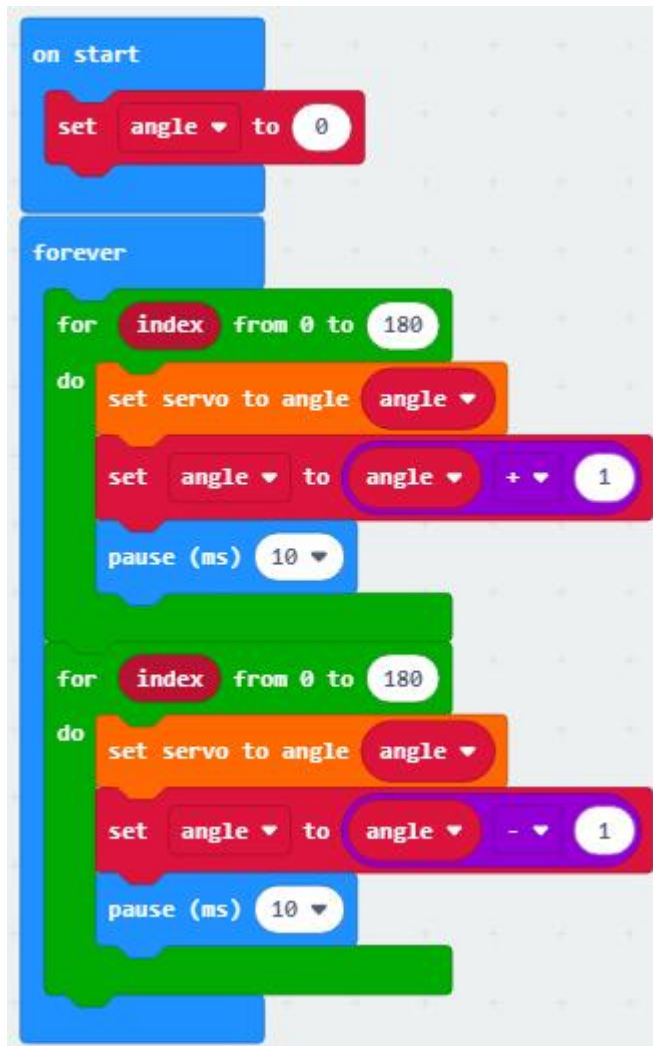


Copy once and change the "+" of to "-" :





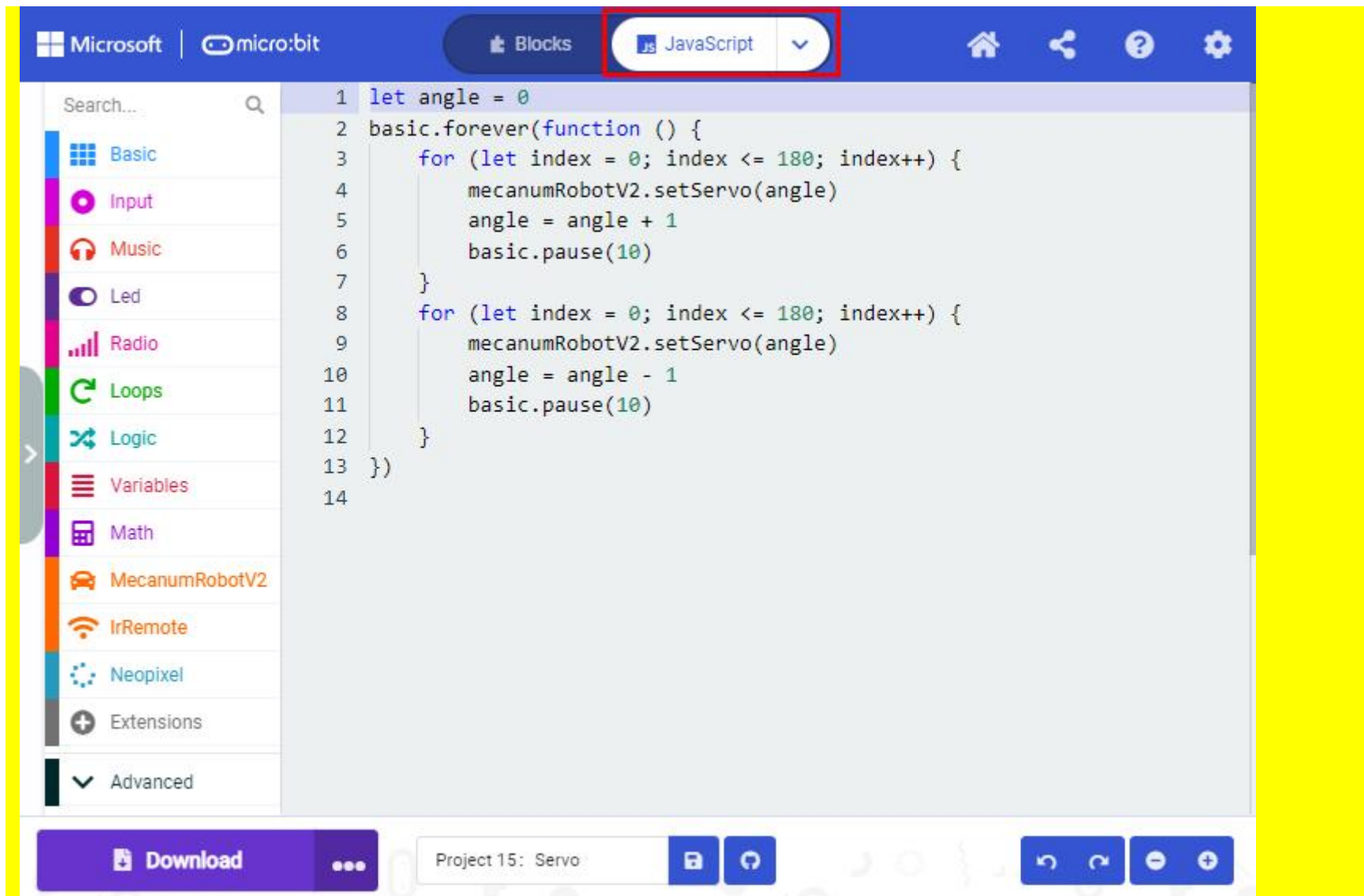
Complete Program:



- ① The "on start" command block runs only once to start the program.
- ② Set the initial value of the angle variable to 0.
- ③ In the "forever" command box, the program runs cyclically
- ④ Cycle 180 times
- ⑤ Rotate the servo to angle
- ⑥ Angle variable increases 1
- ⑦ Delay in 10ms
- ⑧ Cycle 180 times
- ⑨ The servo rotates to angle
- ⑩ Angle angle variable minus 1
- ⑪ Delay in 10ms

Click "JavaScript" to view the corresponding JavaScript code: :

keyestudio

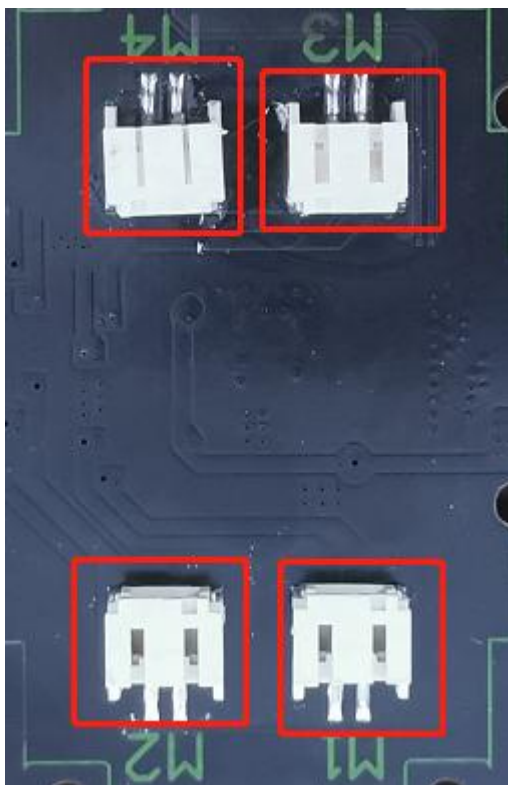


6. Test Result

After uploading the test code and dial POWER switch to ON end, the servo rotates from 0 degree to 180 degrees.

(How to download? How to quick download?)

Project 16: Motor

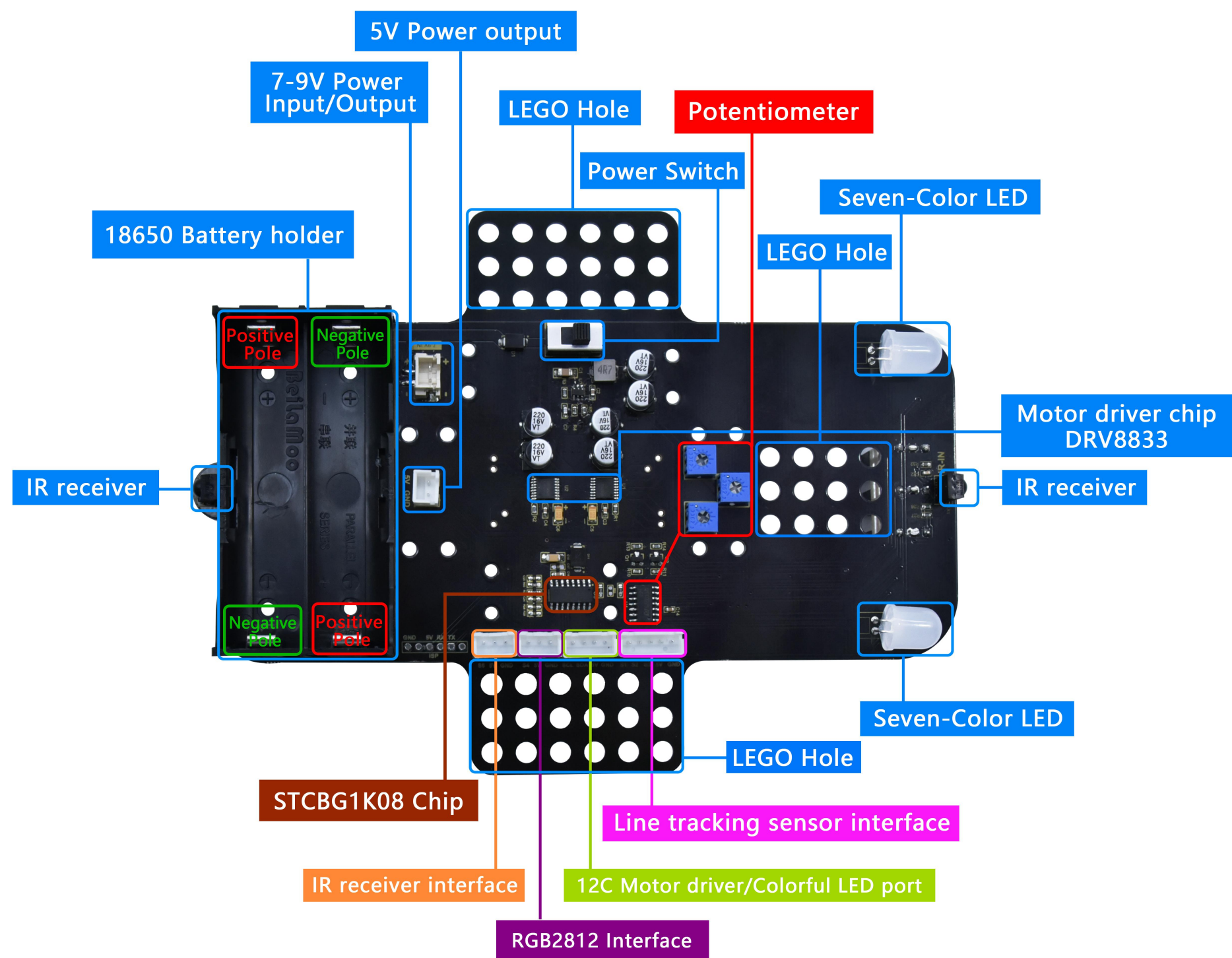


1. Description

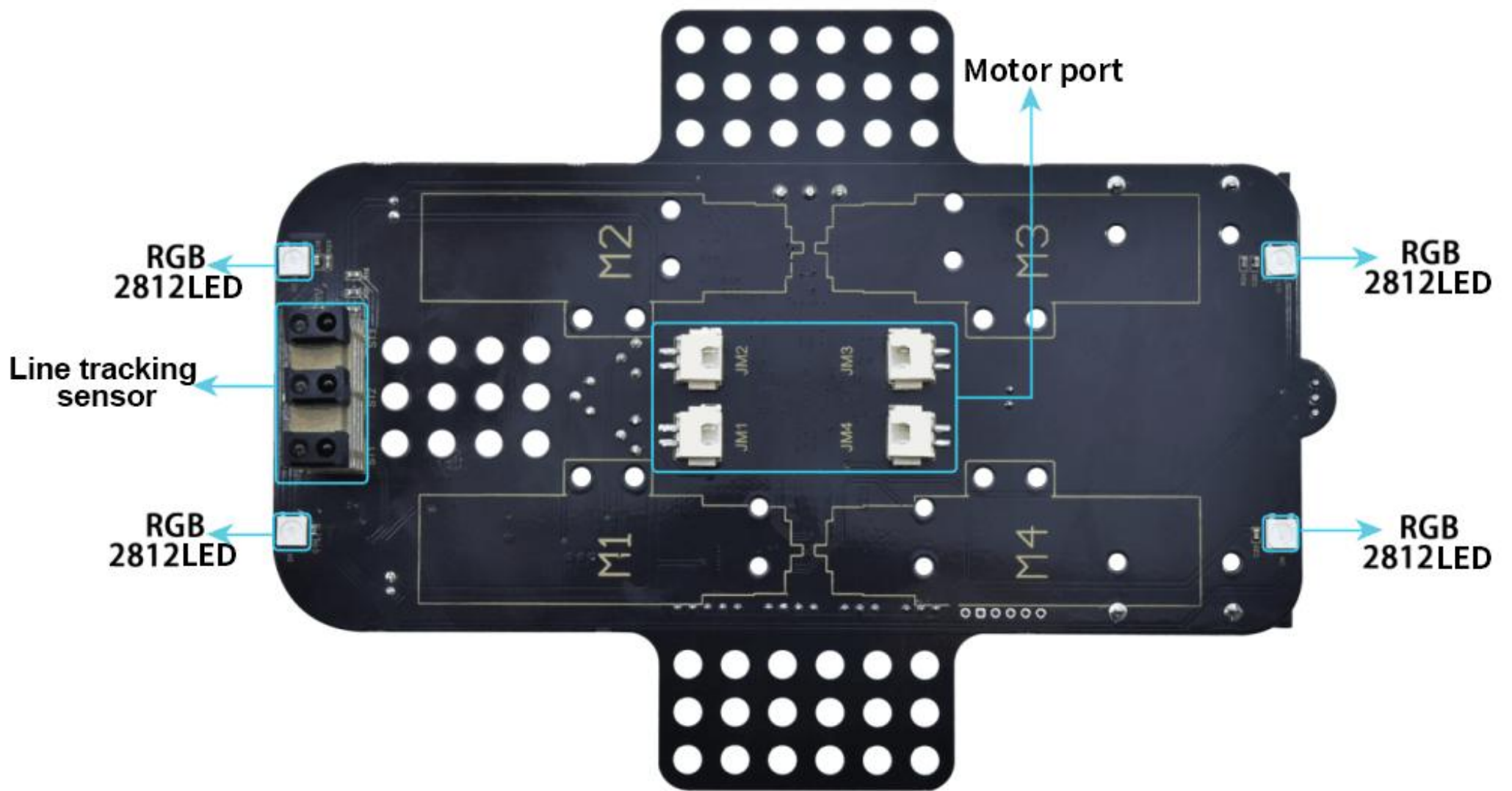
The Keyestudio 4WD Mecanum Robot Car is equipped with 4 DC reduction motors, also called gear reduction motor,

which is developed on the ordinary DC motor. It has a matching gear reduction box which provides a lower speed but a larger torque. Furthermore, different reduction ratios of the box can provide different speeds and torques. Gear motor is the integration of gearmotor and motor, which is applied widely in steel and machine industry

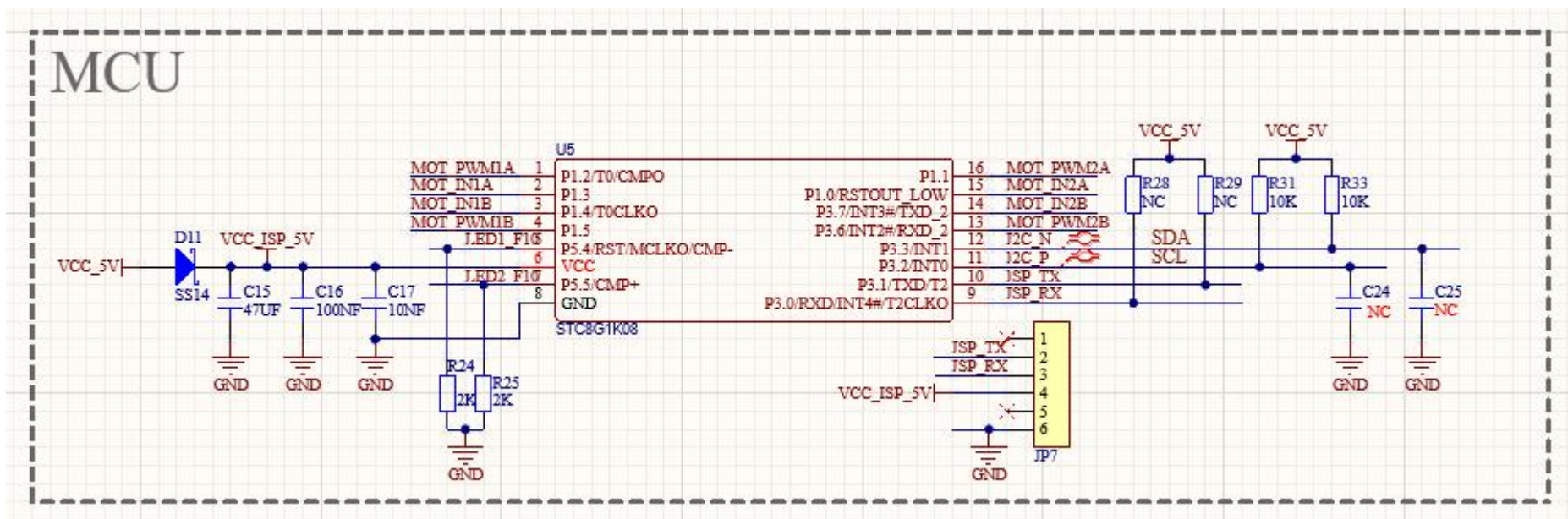
Micro:bit motor driver shield comes with a DRV8833 chip. In order to save the IO port resource, we control the rotation direction and speed of 4 DC gear motors with the DRV8833 chip.

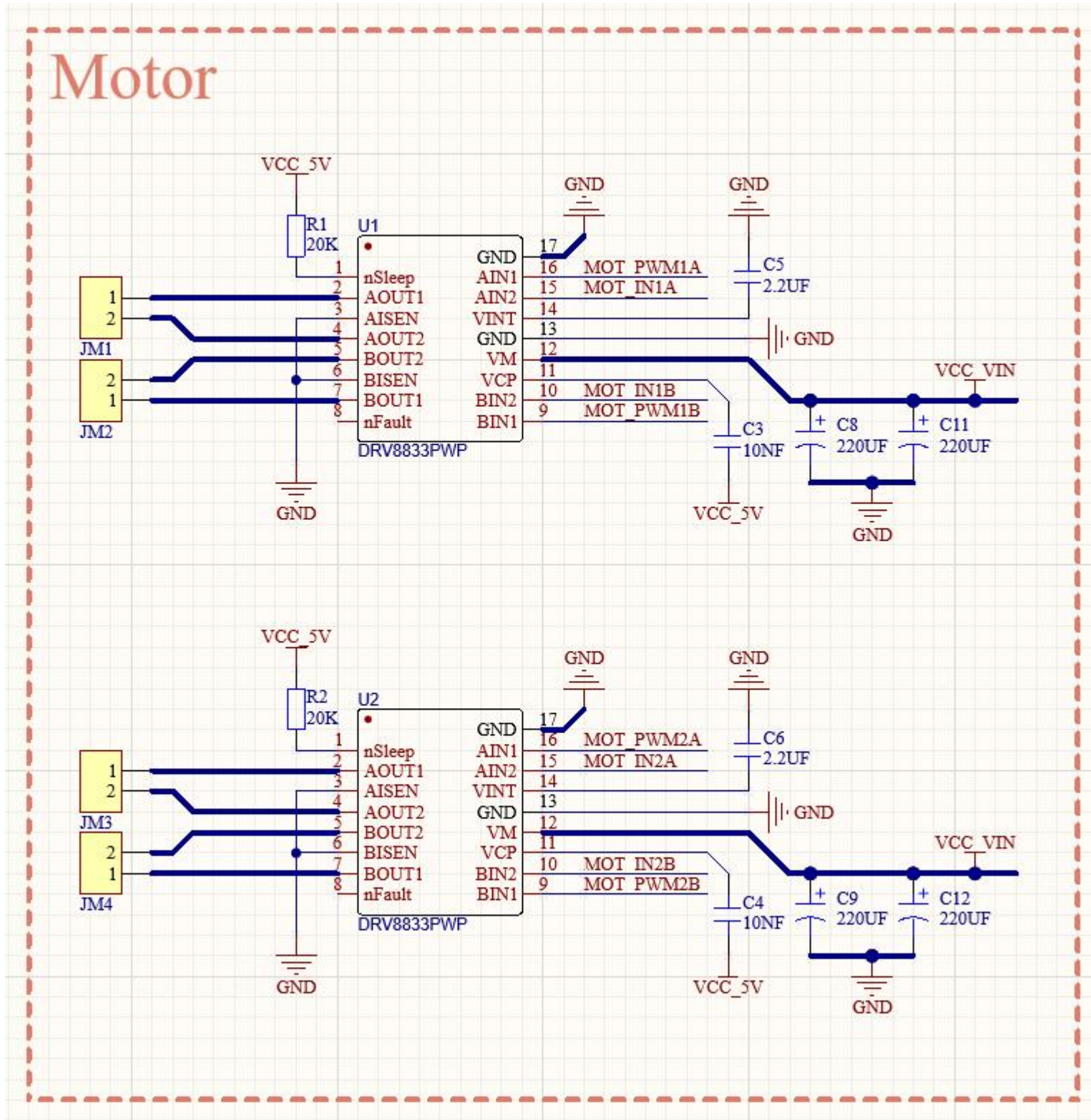


Front



Back





2. Preparation

- Insert the micro:bit board into the slot of keyestudio 4WD Mecanum Robot Car V2.0
- Place batteries into battery holder
- Dial power switch to ON end
- Connect the micro:bit to your computer via an USB cable
- Open the Web version of Makecode

Import Hex profile (**How to import?**) , or click "New Project" and drag blocks step by step(add MecanumRobot extension library first)

(How to add Mecanum_Robot extension?)

3. Test Code

Code 1:

File Type	Route	File Name
-----------	-------	-----------

Python file	..\2. Makecode	Project 16: Motor-1.hex
	Tutorial\Makecode Code	

Or you could edit code step by step in the editing area.

(1)Click " MecanumRobotV2" to find and drag  into "forever" ;click the number behind speed to choose 75  ;

(2)Copy  four times; click the little triangle behind "Motor" to choose Lower_left, Upper_right, Lower_right respectively; and put them all in

 ;

(3)Click "Basic" to find and drag "pause (ms) 100" to "forever" ;set delay in 2000ms  ;

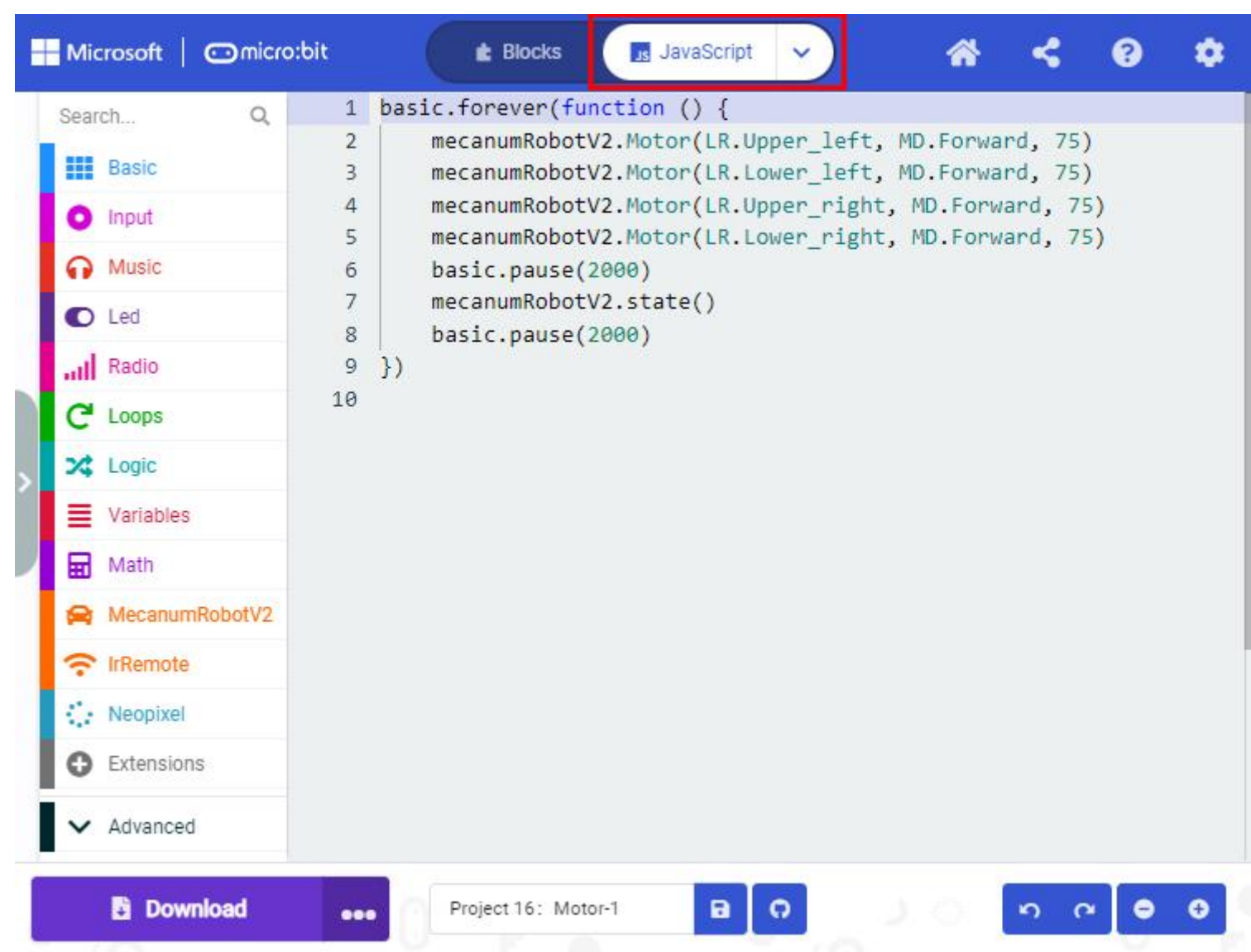
(4)Click "MecanumRobot" to find and drag  to  ; copy  once and put it behind .

Complete Program:



- ① In the "forever" instruction block, the program runs cyclically.
- ② Set the left front motor speed to 75, and rotate clockwise.
- ③ Set the speed of the left rear motor to 75 and the direction to rotate clockwise.
- ④Set the right front motor speed to 75 and the direction to rotate clockwise.
- ⑤Set the right rear motor speed to 75 and the direction to rotate clockwise
- ⑥The delay time is 2000 milliseconds
- ⑦4 motors stop rotating
- ⑧ Delay time 2000 milliseconds

Click "JavaScript" to view the corresponding JavaScript code: :



Code2:

Code path:

File Type	Path	File Name
Hex file	..\2. Makecode Tutorial\Makecode Code	Project 16: Motor-2.hex

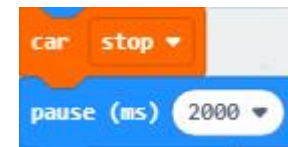
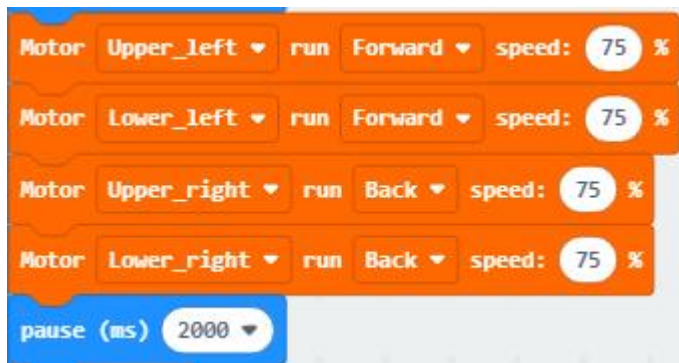
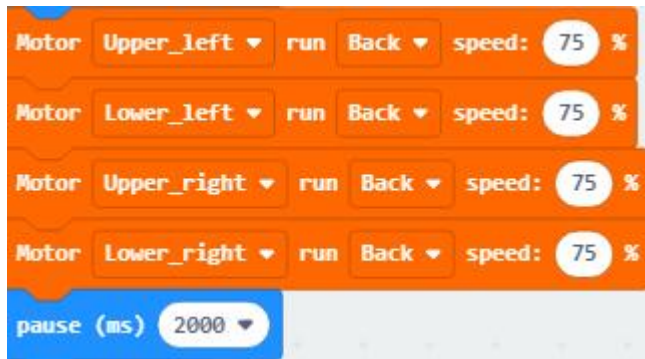
Or you could edit code step by step in the editing area.



Drag and copy three times; click the little triangle behind" run" to choose as

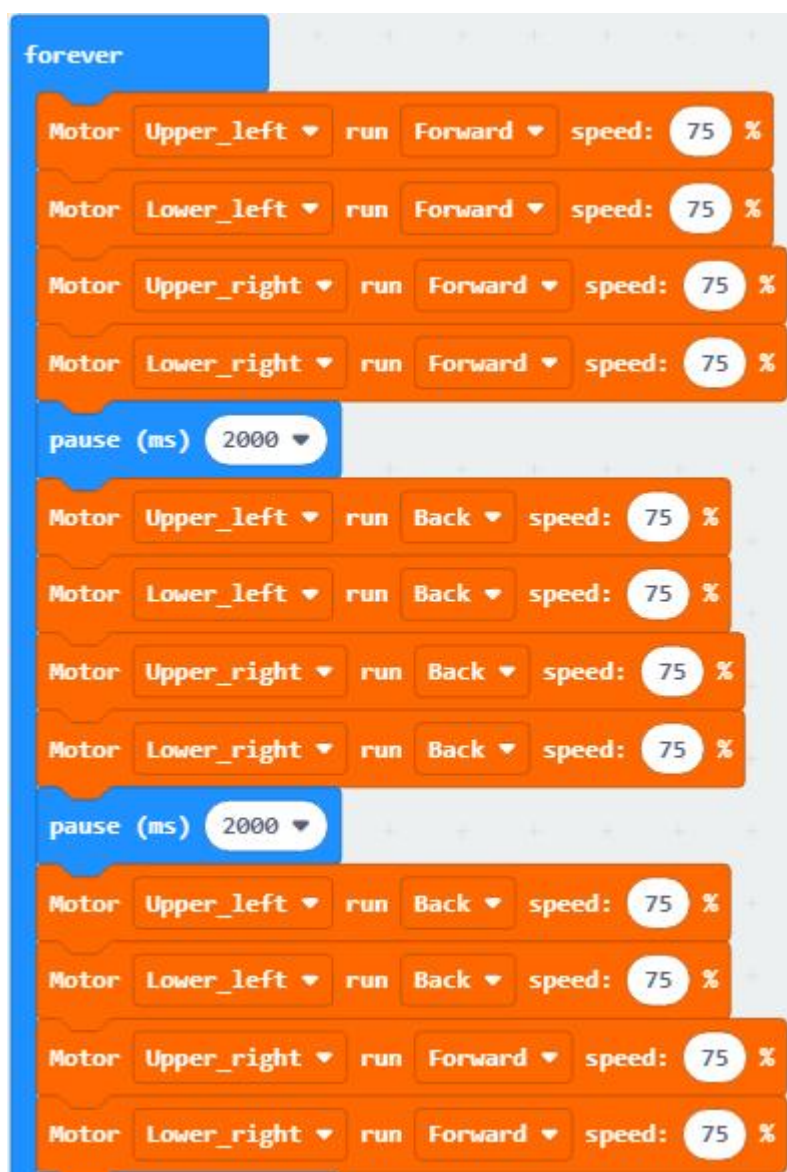
keystudio

shown



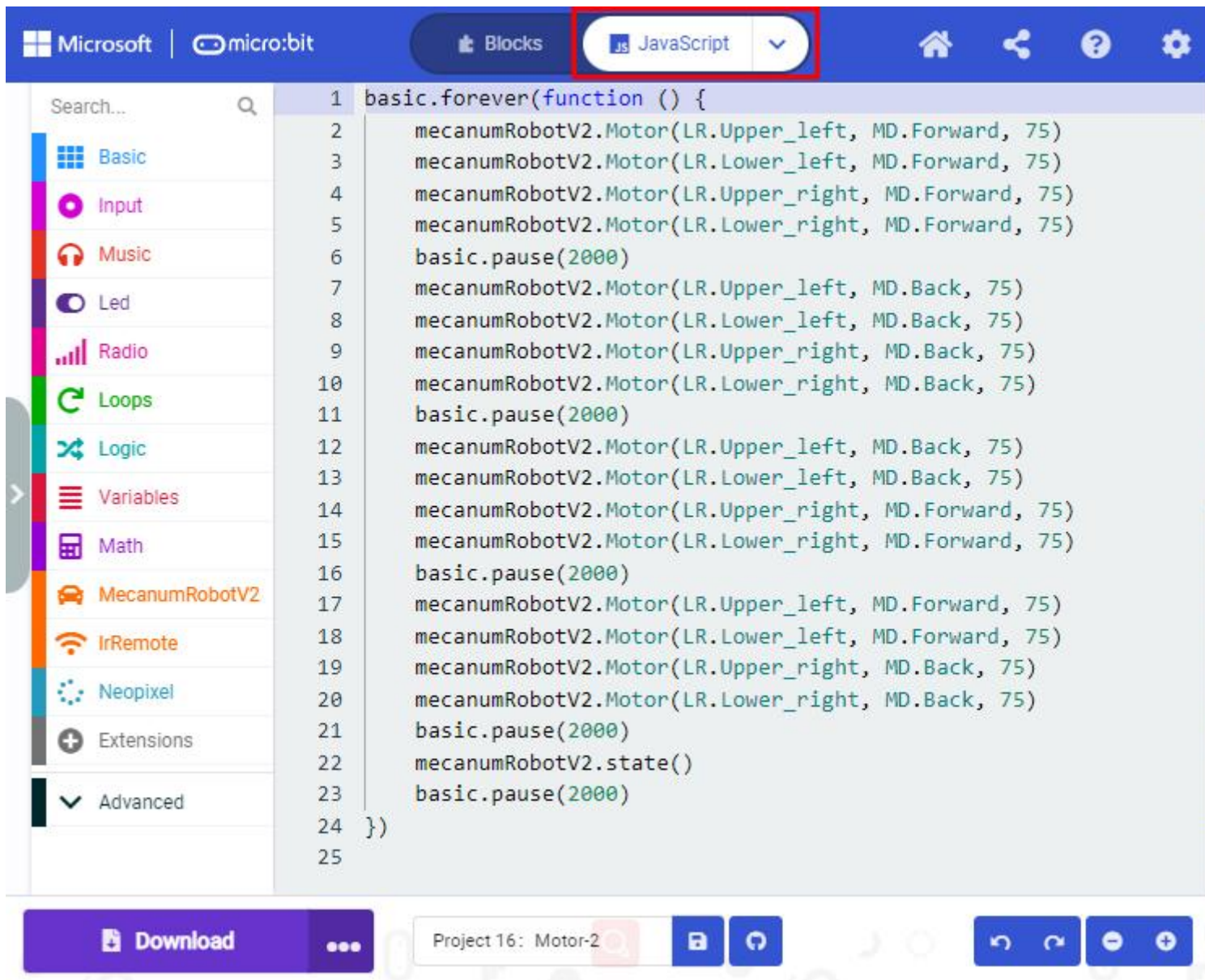
And then put they all in forever and add

Complete Program:



- ① In the "forever" instruction block, the program runs cyclically.
- ② Set the left front motor speed to 75 and the direction to rotate forward.
- ③ Set the speed of the rear left rear motor to 75 and the direction to rotate forward.
- ④ Set the right front motor speed to 75 and the direction to rotate forward.
- ⑤ Set the right rear motor speed to 75 and the direction to rotate forward.
- ⑥ Wait for 2 seconds
- ⑦ Set the left front motor speed to 75, and the direction is reversed.
- ⑧ Set the motor speed at the left rear to 75, and the direction is reversed.
- ⑨ Set the right front motor speed to 75, and the direction is reversed.
- ⑩ Set the right rear motor speed to 75, and the direction is reversed.
- ⑪ Wait for 2 seconds
- ⑫ Set the left front motor speed to 75, and the direction is reversed.
- ⑬ Set the motor speed at the left rear to 75, and the direction is reversed.
- ⑭ Set the right front motor speed to 75 and the direction to rotate forward.
- ⑮ Set the right rear motor speed to 75, the direction is forward.
- ⑯ Wait for 2 seconds
- ⑰ Set the left front motor speed to 75 and the direction to rotate forward.
- ⑱ Set the speed of the rear left motor to 75 and the direction to rotate forward.
- ⑲ Set the front right motor speed to 75, and the direction is reversed.
- ⑳ Set the right rear motor speed to 75, and the direction is reversed.
- ㉑ Wait for 2 seconds
- ㉒ The car stops
- ㉓ Wait for 2 seconds

Click "JavaScript" to view the corresponding JavaScript code: :



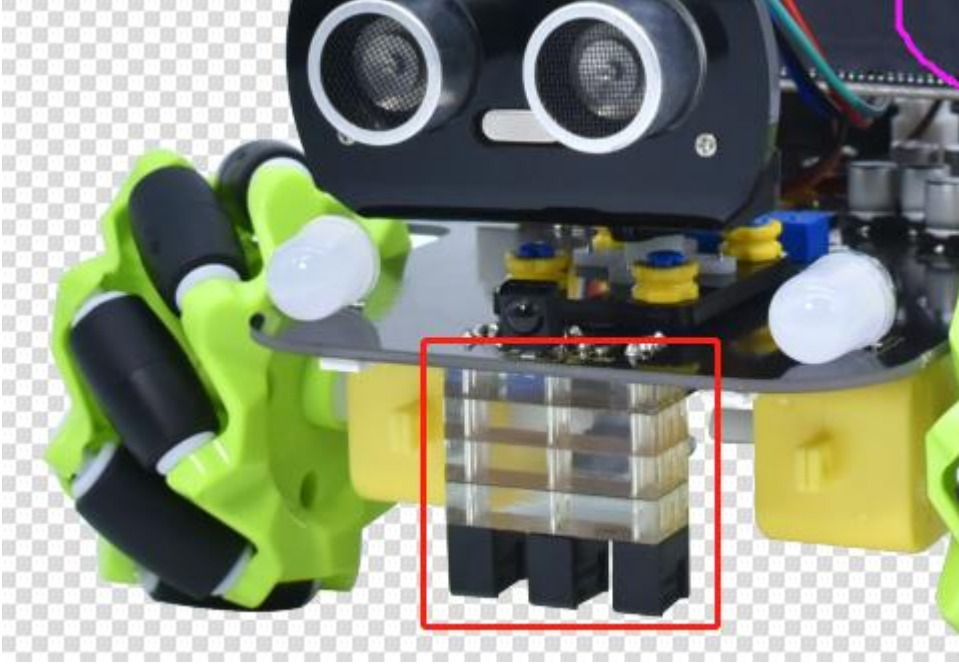
4. Test Result

Download code 1 to micro:bit board, dial POWER switch to ON end. Smart car goes forward for 2s and stops for 2s. Download code 2 to micro:bit board, the car goes forward for 2s, turns back for 2s, turn left for 2s, turn right for 2s and stops for 2s and repeats this pattern.

(How to download? How to quick download?)

Project 17: Line Tracking Sensor

Project 17.1: Detect Line Tracking Sensor



1. Description

The motor driver board of the Keyestudio 4WD Mecanum Robot Car comes with a 3-channel line tracking sensor, which adopts TCRT5000 IR tubes and 3 potentiometers.

The TCRT5000 IR tube contains an IR emitting tube and an IR receiving tube. When the infrared signals of the emitting tube is received by the receiving tube through reflection, the resistance of the receiving tube will change, which is generally reflected in the voltage change on the circuit.

The resistance varies depending on the intensity of the infrared signals received by the receiving tube, which is often in the color of the reflecting surface and the distance of the reflecting surface receiving tube. At the time of detection, black is high level active and white is low level active.

2. Working Principle

When the car runs above a white road, the IR emitting tube installed under the car emits infrared signals to detect the road and the receiving tube will receive signals sending back. Then the output end outputs low level(0); when it detects black lines, it outputs high level(1).

After putting a white paper on the bottom of the 4WD Mecanum Robot Car, we will rotate the potentiometers on the 3-way tracking sensor. When the indicator light on the sensor module is on, pick up the car to make the two wheels on the 4WD Mecanum Robot Car separate. The height of the white paper is about 1.5cm, when the indicator light on the sensor module is off, and then the sensitivity is adjusted.

3. Preparation

- Insert the micro:bit board into the slot of keyestudio 4WD Mecanum Robot Car V2.0
- Place batteries into battery holder
- Dial power switch to ON end
- Connect the micro:bit to your computer via an USB cable
- Open the Web version of Makecode

Import Hex profile (**How to import?**) , or click “New Project” and drag blocks step by step(add MecanumRobot extension library first)

(How to add Mecanum_Robot extension?)

4. Test Code

Code 1:

File Type	Path	File Name
Hex file	..\2. Makecode Tutorial\Makecode Code	Project 17.1: Detect Line Tracking Sensor-1.hex

Or you could edit code step by step in the editing area.

Click “Advanced” → “Serial” → “serial redirect to USB”

Place it into “on start”

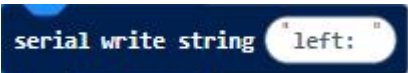


Click “Led” → “more” → “led enable(false)” and place it into “on start” block.

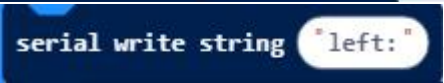


Enter “Advanced” → “Serial” → 

Leave it into “forever” block.

Input “left” and drag out block 

Click “ MecanumRobotV2 ” , and drag out block  and place it behind the .

Copy  once, and change Left to Center .

Copy  once, and change Left to Center .

keystudio

Copy  once, and change "left: " to "right: " , and drag out  block.

Find and drag out  from "MecanumRobotV2" and place it into , and then change Left to Right.



Click "Basic" to find and drag "pause (ms) 100" to "forever" and set the delay to 200ms



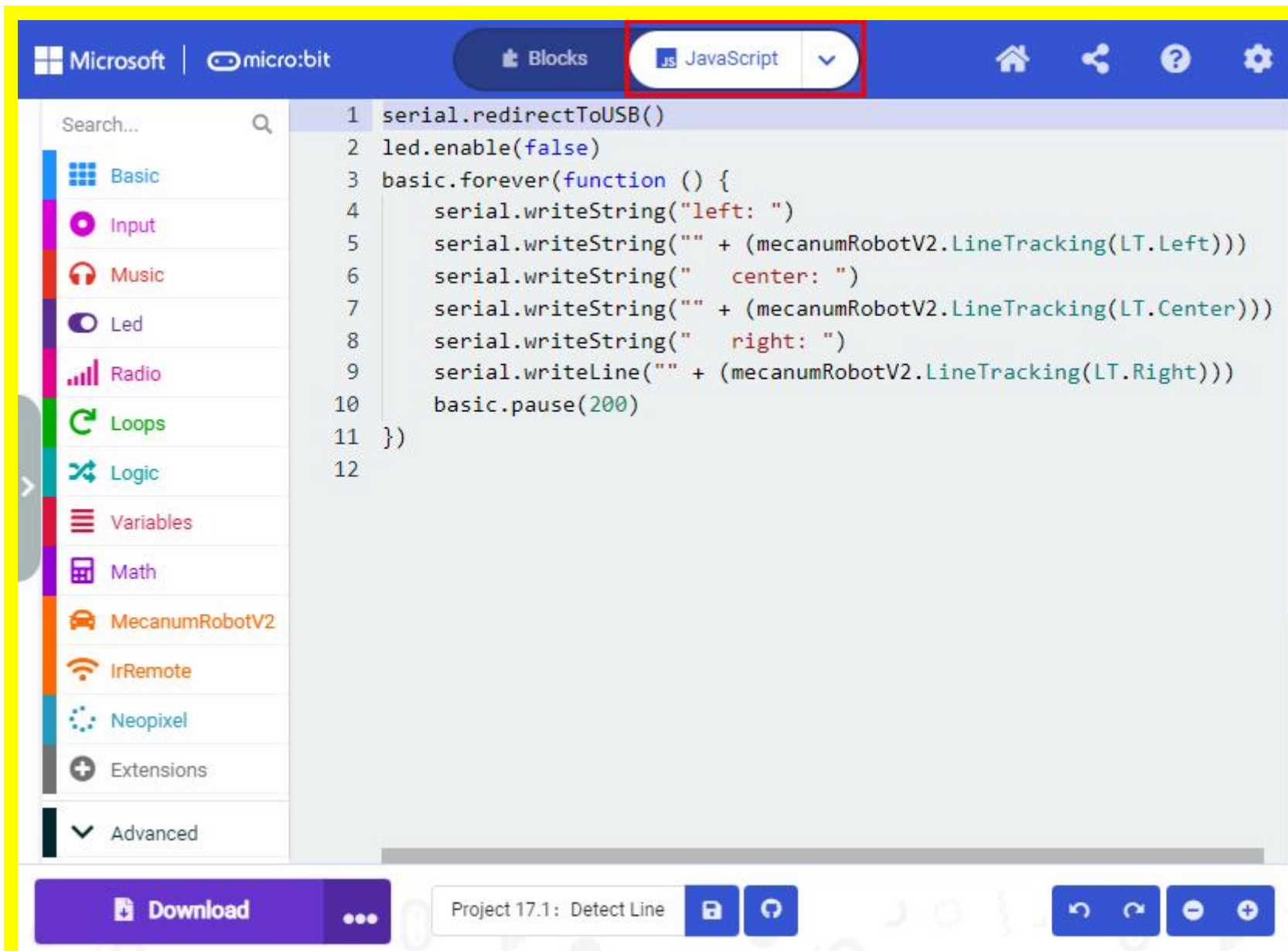
Complete Program:



- ① The "on start" command block runs only once to start the program.
- ② Serial redirection USB.
- ③ In the "forever" instruction block, the program runs cyclically.
- ④ Write string "left:" serially
- ⑤ Serially write the output value of the tracking sensor on the left
- Serial write string "center:"
- Serially write the output value of the middle tracking sensor
- ⑥ Serial write string "right:"
- ⑦ Serially write the output value of the tracking sensor on the right
- ⑧ Writing in line break
- ⑨ Delay time 200 milliseconds

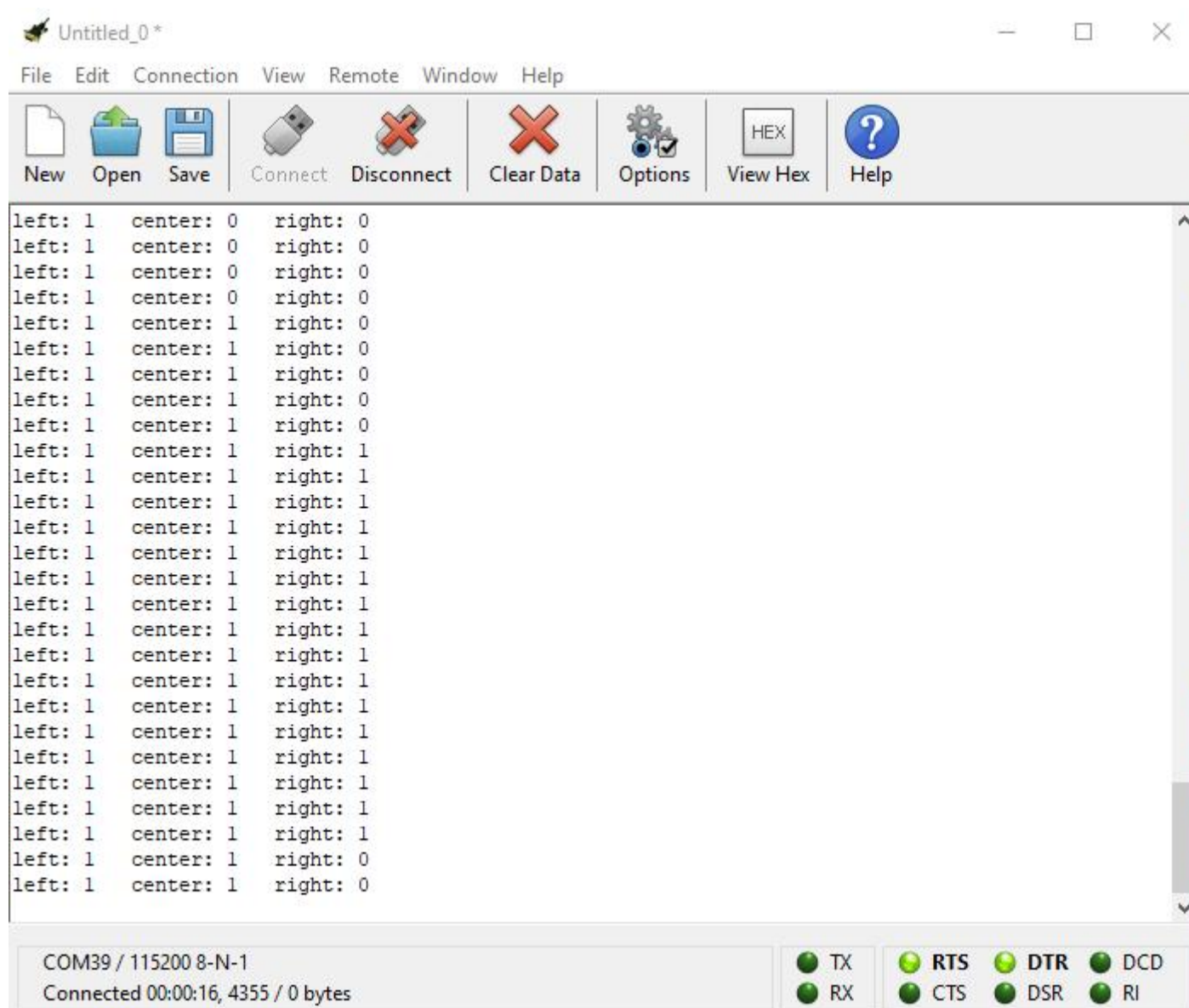
keyestudio

Click "JavaScript" to view the corresponding JavaScript code: :

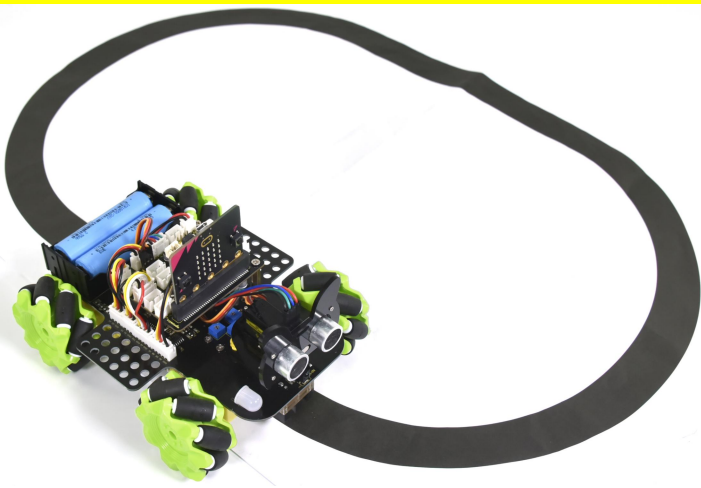


Open CoolTerm, click Options to select SerialPort. Set COM port and 115200 baud rate. Click "OK" and "Connect".

The CoolTerm serial monitor displays the digital signals read by the line tracking sensors.



Project 17.2: Tracking Smart Car



1. Description

In this lesson we will combine a line tracking sensor with a motor to make a line tracking smart car. The micro:bit board will analyze the signals and control the smart car to show the line tracking function.

2. Working Principle

The smart car will make different moves according to the value received by the 3-channel line tracking sensor.

Left line tracking sensor	Middle line tracking sensor	Right line tracking sensor	Binary values	Decimal value	Microbit 4WD Mecanum Robot Car V2.0 的行动
LOW (0)	LOW (0)	LOW (0)	000	0	Stop
LOW (0)	LOW (0)	HIGH (1)	001	1	Turn Right
LOW (0)	HIGH (1)	LOW (0)	010	2	Go forward
LOW (0)	HIGH (1)	HIGH (1)	011	3	Turn Right
HIGH (1)	LOW (0)	LOW (0)	100	4	Turn Left
HIGH (1)	LOW (0)	HIGH (1)	101	5	Go forward
HIGH (1)	HIGH (1)	LOW (0)	110	6	Turn Left
HIGH (1)	HIGH (1)	HIGH (1)	111	7	Stop

3. Preparation

- Insert the micro:bit board into the slot of keystudio 4WD Mecanum Robot Car V2.0
- Place batteries into battery holder
- Dial power switch to ON end
- Connect the micro:bit to your computer via an USB cable

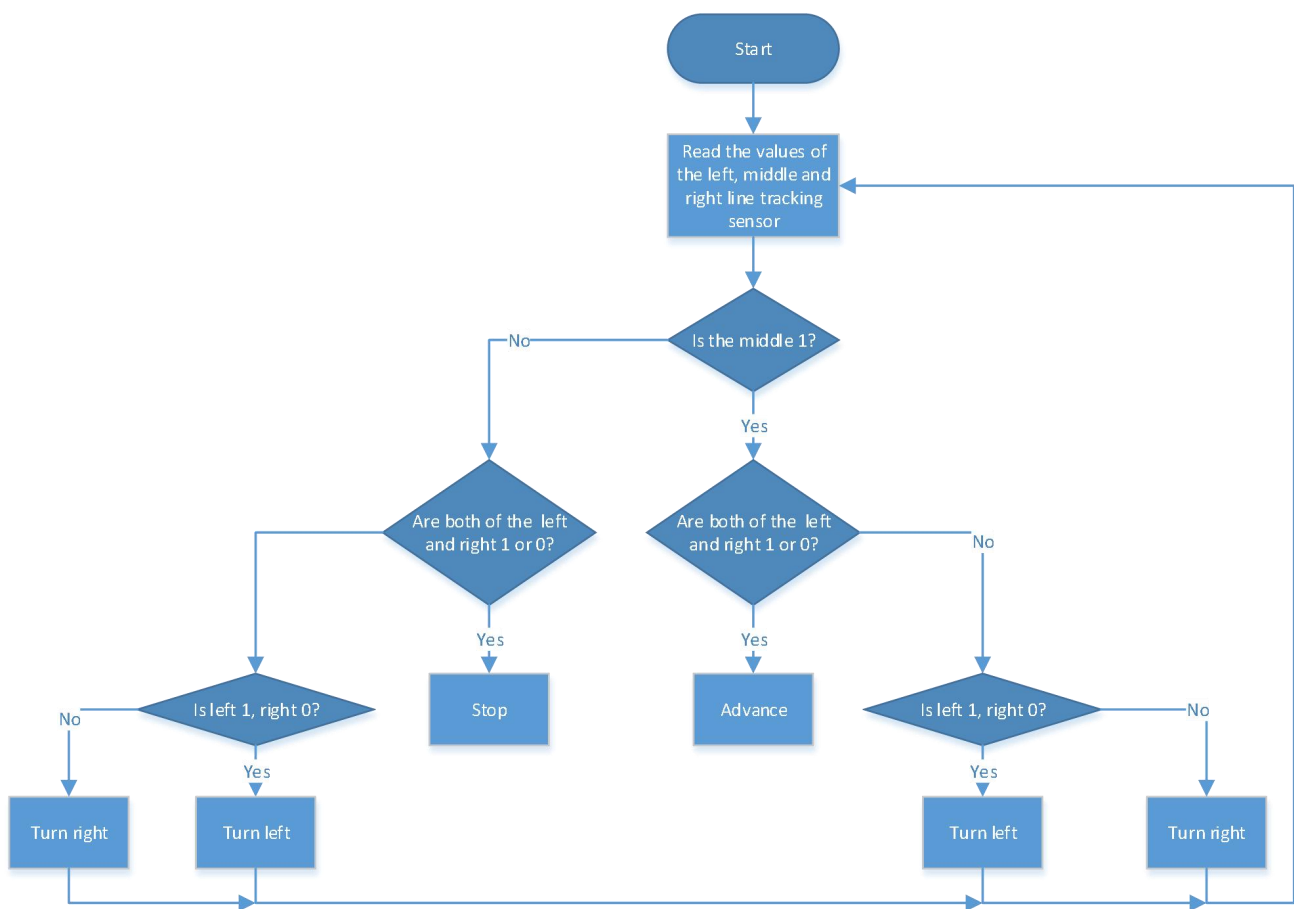
➤ Open the Web version of Makecode

Import Hex profile (**How to import?**) , or click “New Project” and drag blocks step by step(add MecanumRobot extension library first)

(How to add Mecanum_Robot extension?)

Warning: The 3-way tracking sensor should be used in environments without infrared interference such as sunlight. Sunlight contains a lot of invisible light, such as infrared and ultraviolet. In an environment with strong sunlight, the 3-way tracking sensor cannot work properly.

4. Flow Chart



5. Test Code

Code path:

File Type	Path	File Name
Hex file	..\2. Makecode Tutorial\Makecode Code	..\2. Makecode Tutorial\Makecode Code

Or you could edit code step by step in the editing area.

Click “Variables” → “Make a Variable...” → “New variable name: ” .

keyestudio

Input LL → "OK" , then the variable "LL" is established, and variable "CC" and "RR" are likewise.

Drag out "set RR to 0" and place into "on start" block, copy "set RR to 0" 2 times and place into "on start" block.

Click the drop-down triangle button after RR to select LL and CC.



Click "Led" → "more" → "led enable(false)" and place into "on start" block.



Click "Variables" , find and drag out "set RR to 0" into "forever" block.

Click the drop-down triangle button after RR to select LL and click "MecanumRobotV2" .

Find and drag out  and put it at the number 0 after to.

Duplicate  2 times and place into "forever" block.

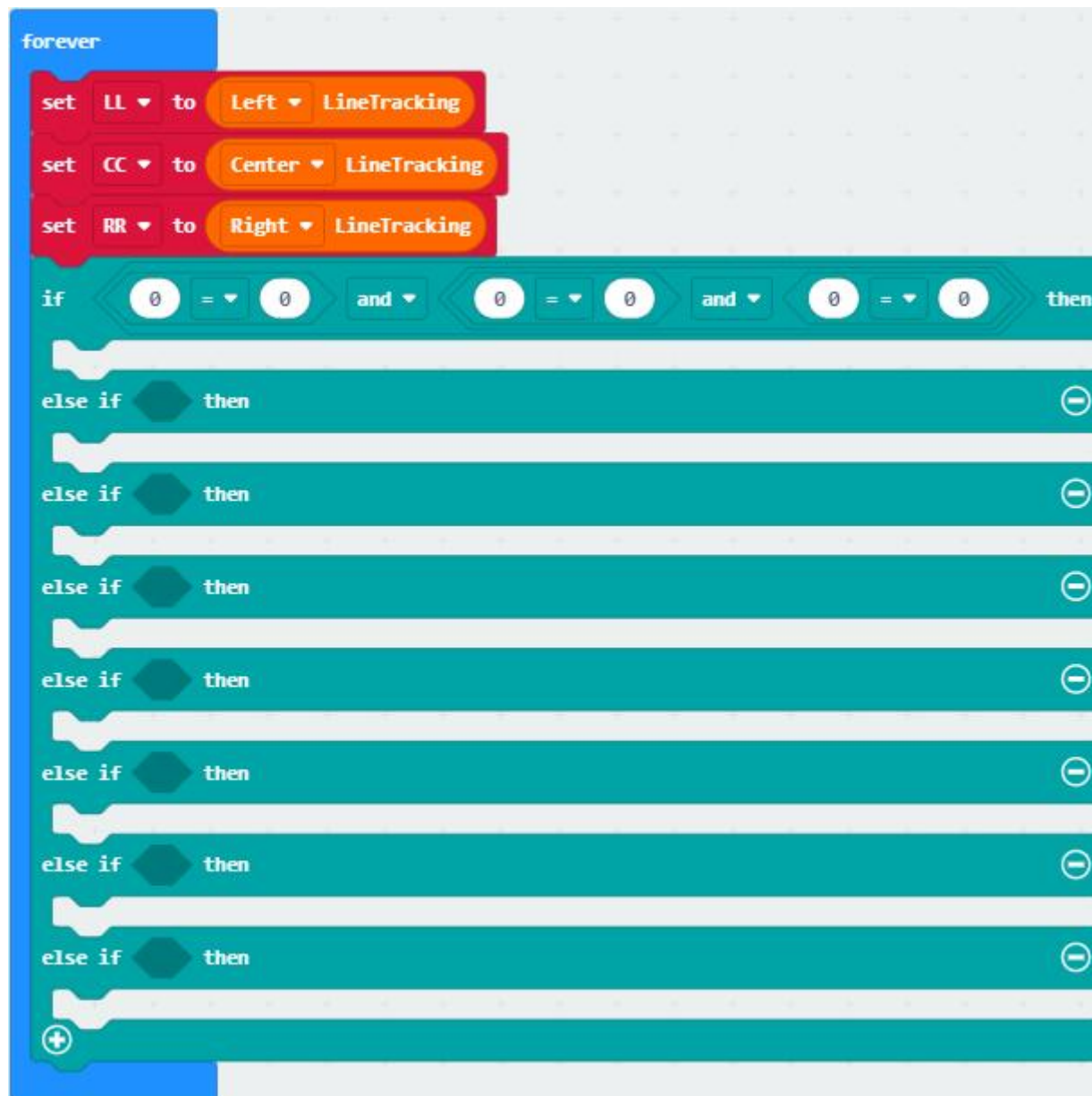
Change the second LL to CC, " Left" to " Center" and the third LL to RR, " Left" to " Right" .



Click "Logic" , find and drag out "if true then...else" into "forever" block.

Click  7 times to add else if, then click else  once to cancel else.

Find and drag out 2 "and" block into true block, and drag out 3 "=" block into and block.



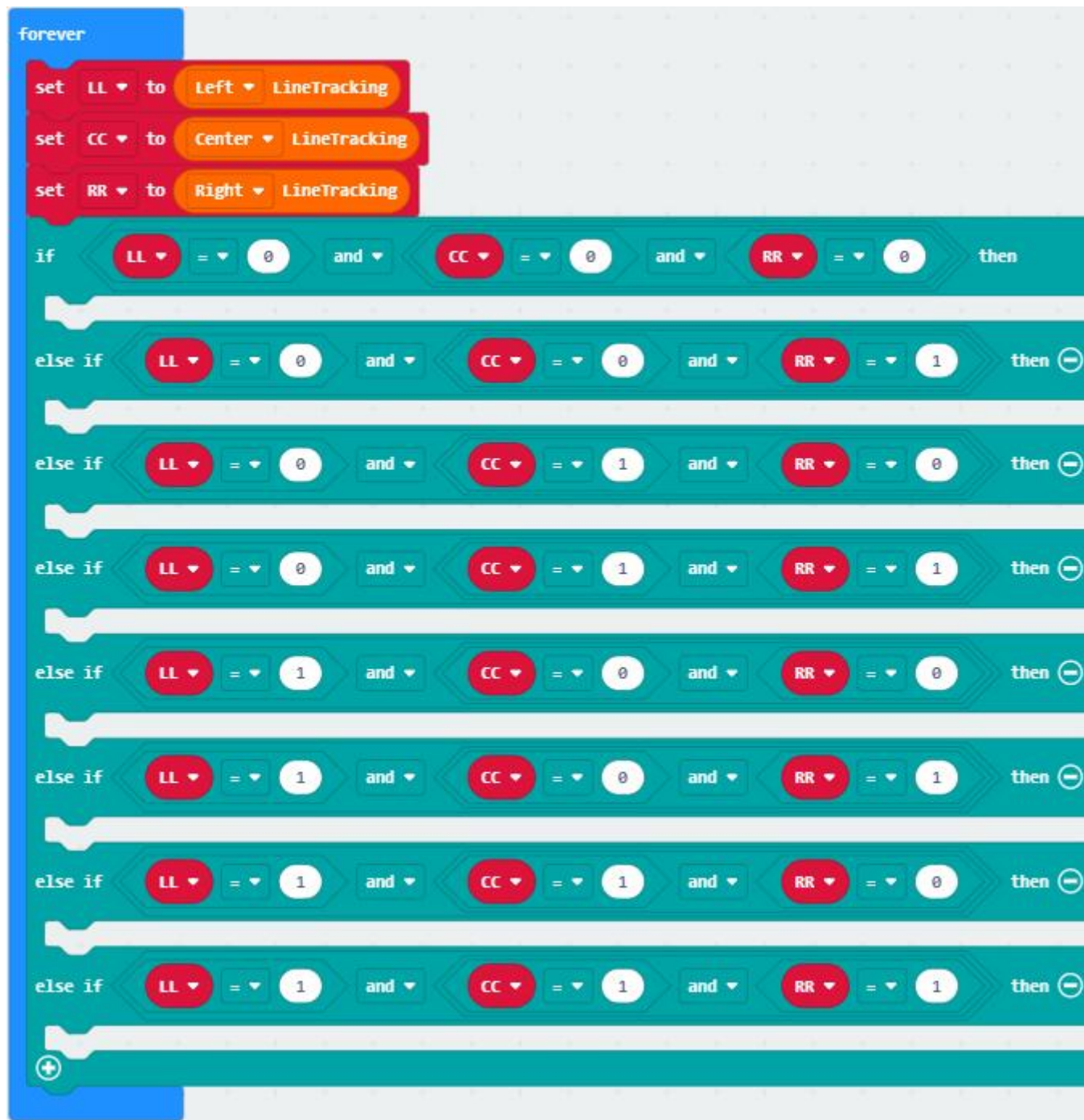
Click "Variables" , find and drag out "LL" into "=" left side, "=" right side 0 is invariant.

Duplicate "LL" 2times and place it into and block.

Click the drop-down triangle button after to select CC and RR, "=" right side 0 is invariant.



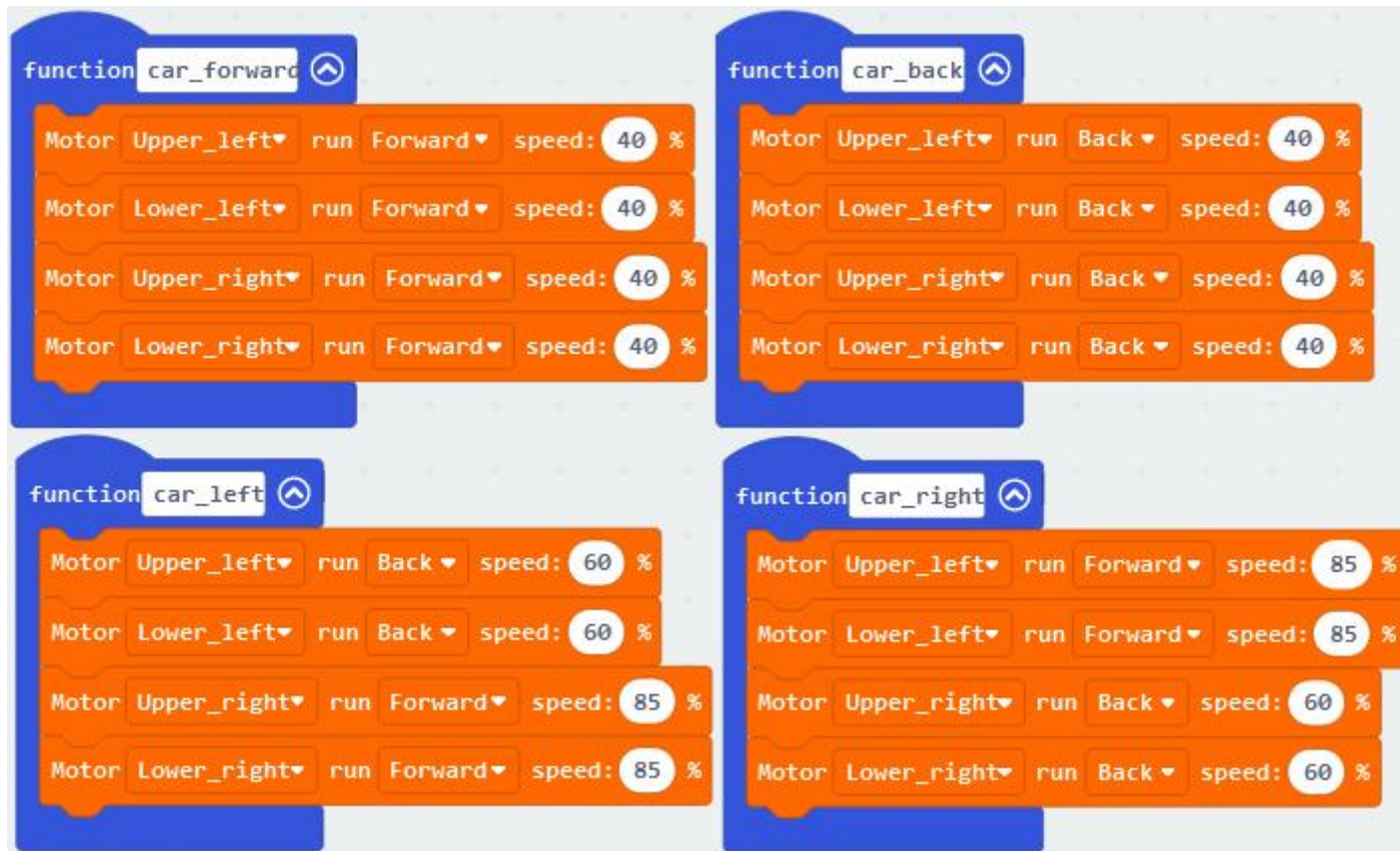
Duplicate  7 times, and change o according to the below:



Click "Functions" of "Advance" and click .

Change "doSomething" to "car_forward" , "car_back" , "car_left" , "car_right" respectively.

keystudio



Click "Functions" of "Advance" to find and drag  2 times to the first and third else if.

Click "Functions" of "Advance" to find and drag  2 times to the the fourth and the sixth else if and find and drag out  2 times and put in the second one .

In the fifth and seventh Else if, click "MecanumRobotV2" ,

if and drag  to the first else if, find and drag  once, and place it into the first if.



Complete Program:

keyestudio



.....The "on start" command block runs only once to start the program.

.....Close the LED dot matrix

.....Set the variable LL to 0

.....Set variable CC to 0

.....Set variable RR to 0

.....In the "forever" instruction block, the program runs cyclically.

.....The left tracking sensor reads the high and low levels and assigns the variable LL

.....The middle tracking sensor reads the high and low levels and assigns the variable CC

.....The right tracking sensor reads the high and low levels and assigns the variable RR

.....The left, middle and right tracking sensor all read 0

.....The car stop

.....The left and middle tracking sensor all read 0 and the right tracking sensor reads 1

.....Turn right function

.....The left and right tracking sensor all read 0 and the middle tracking sensor reads 1

.....Advance.function

.....The middle .and right tracking sensor all read 1 and the left tracking sensor reads 0

.....Turn right function

.....The middle .and right tracking sensor all read 0 and the left tracking sensor reads 1

.....Turn left function

.....The left and right tracking sensor all read 1 and the middle tracking sensor reads 0

.....Advance.function

.....The left and middle tracking sensor all read 1 and the right tracking sensor reads 0

.....Turn left function

.....The left, middle .and right tracking sensor all read 1

.....Advance.function

keyestudio



- ①forward function
- ②The left front motor rotates forward at a speed of 40
- ③The motor at the left rear rotates forward at a speed of 40
- ④The right front motor rotates forward at a speed of 40
- ⑤The right rear motor rotates forward at a speed of 40
- ⑥Backward function
- ⑦The left front motor reverses, the speed is 40
- ⑧The left rear motor reverses, the speed is 40
- ⑨The right front motor reverses, the speed is 40
- ⑩The right rear motor reverses, the speed is 40
- ⑪Left turn function
- ⑫The left front motor reverses, the speed is 60
- ⑬The left rear motor reverses at a speed of 60
- ⑭The right front motor rotates forward, the speed is 85
- ⑮ The right rear motor rotates forward, the speed is 85
- ⑯ right turn function
- ⑰ The left front motor rotates forward, the speed is 85
- ⑱ The left rear motor rotates forward, the speed is 85
- ⑲The right front motor reverses, the speed is 60
- ⑳ The right rear motor reverses, the speed is 60

Click "JavaScript" to view the corresponding JavaScript code:

```

1 function car_back () {
2     mecanumRobotV2.Motor(LR.Upper_left, MD.Back, 40)
3     mecanumRobotV2.Motor(LR.Lower_left, MD.Back, 40)
4     mecanumRobotV2.Motor(LR.Upper_right, MD.Back, 40)
5     mecanumRobotV2.Motor(LR.Lower_right, MD.Back, 40)
6 }
7 function car_left () {
8     mecanumRobotV2.Motor(LR.Upper_left, MD.Back, 60)
9     mecanumRobotV2.Motor(LR.Lower_left, MD.Back, 60)
10    mecanumRobotV2.Motor(LR.Upper_right, MD.Forward, 85)
11    mecanumRobotV2.Motor(LR.Lower_right, MD.Forward, 85)
12 }
13 function car_forward () {
14     mecanumRobotV2.Motor(LR.Upper_left, MD.Forward, 40)
15     mecanumRobotV2.Motor(LR.Lower_left, MD.Forward, 40)
16     mecanumRobotV2.Motor(LR.Upper_right, MD.Forward, 40)
17     mecanumRobotV2.Motor(LR.Lower_right, MD.Forward, 40)
18 }
19 function car_right () {
20     mecanumRobotV2.Motor(LR.Upper_left, MD.Forward, 85)
21     mecanumRobotV2.Motor(LR.Lower_left, MD.Forward, 85)
22     mecanumRobotV2.Motor(LR.Upper_right, MD.Back, 60)
23     mecanumRobotV2.Motor(LR.Lower_right, MD.Back, 60)
24 }

```

```

25 led.enable(false)
26 let LL = 0
27 let CC = 0
28 let RR = 0
29 basic.forever(function () {
30     LL = mecanumRobotV2.LineTracking(LT.Left)
31     CC = mecanumRobotV2.LineTracking(LT.Center)
32     RR = mecanumRobotV2.LineTracking(LT.Right)
33     if (LL == 0 && (CC == 0 && RR == 0)) {
34         mecanumRobotV2.state()
35     } else if (LL == 0 && (CC == 0 && RR == 1)) {
36         car_right()
37     } else if (LL == 0 && (CC == 1 && RR == 0)) {
38         car_forward()
39     } else if (LL == 0 && (CC == 1 && RR == 1)) {
40         car_right()
41     } else if (LL == 1 && (CC == 0 && RR == 0)) {
42         car_left()
43     } else if (LL == 1 && (CC == 0 && RR == 1)) {
44         car_forward()
45     } else if (LL == 1 && (CC == 1 && RR == 0)) {
46         car_left()
47     } else if (LL == 1 && (CC == 1 && RR == 1)) {
48         car_forward()
49     }
50 })
51

```

5. Test Result

Download code to micro:bit and dial POWER to ON end, line tacking car goes forward along black line .

Note: turn on the switch at the back of micro:bit car.

the width of black line should be larger than the width of line tracking sensor.

Avoid to test smart car under the strong light.

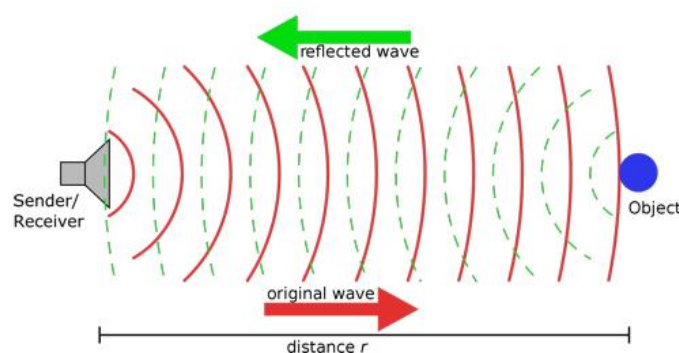
Project 18: Ultrasonic Sensor

Project 18.1: Ultrasonic Ranging

1. Description

The ultrasonic sensor uses sonar to determine distance to an object like 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 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.



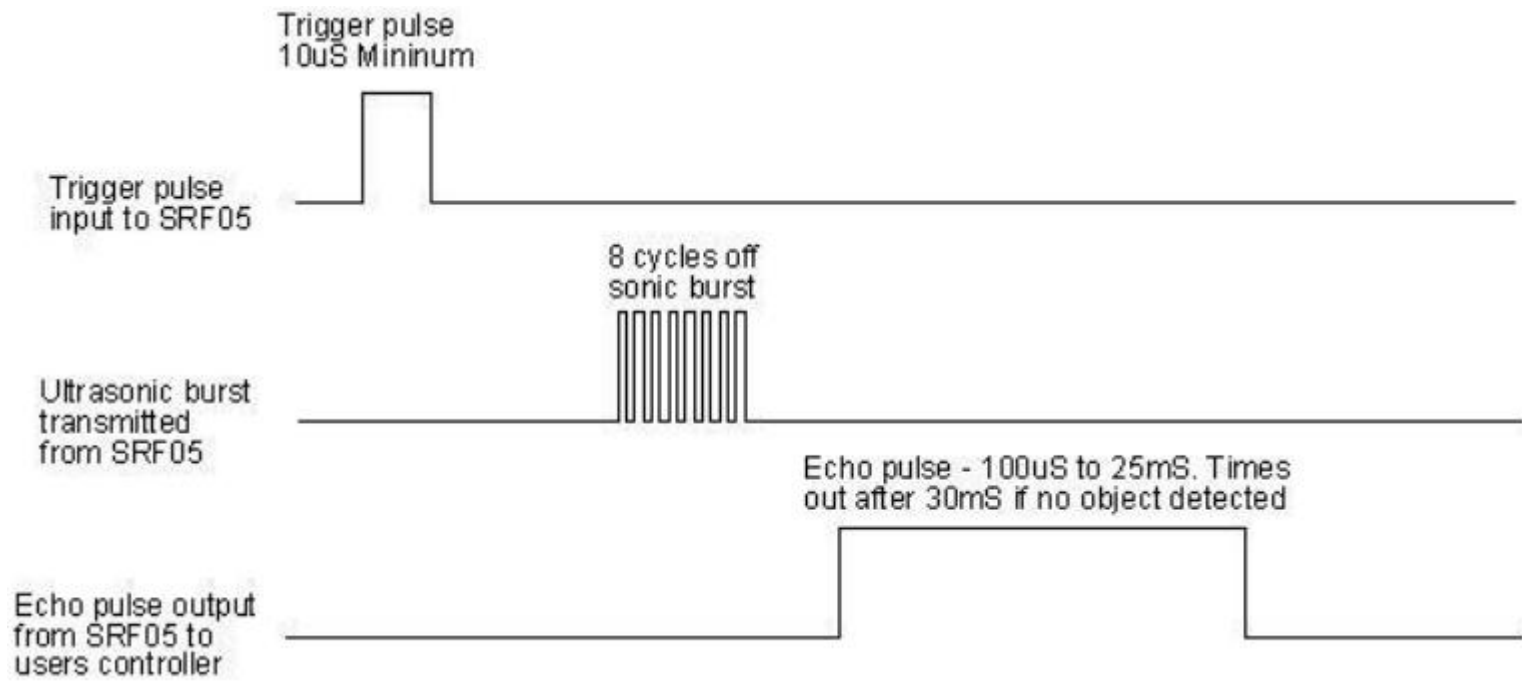
The ultrasonic module will emit the ultrasonic waves after trigger signals. When the ultrasonic waves encounter the object and are reflected back, the module outputs an echo signal, so it can determine the distance of object from the time difference between trigger signal (TRIG) and echo signal (ECHO).

As the picture shows, it is like two eyes. One is transmitting end, the other is receiving end.

According to the above wiring diagram, the integrated port of the ultrasonic sensor module is connected to the 5V G P15 P16 port on the micro:bit motor driver base plate. The Trig (T) pin is controlled by P15 of the micro:bit and the pin of Echo (E) the P16.



2. Working Principle



- (1) Pull down TRIG then trigger high level signals with least 10us;
- (2) After triggering, the module will automatically send eight 40KHz ultrasonic pulses and detect whether there is a signal return;
- (3) If there is a signal return, when ECHO (E) outputs a high level, then the duration of the high level is the time from transmission to reception of the ultrasonic waves. Then test distance = high level duration * 340m/s * 0.5.

3. Parameters

- ◆ Working voltage: 3-5.5V (DC)
- ◆ Working current: 15mA
- ◆ Working frequency: 40khz
- ◆ Maximum detection distance: about 3m
- ◆ Minimum detection distance: 2-3cm
- ◆ Precision: up to 0.2cm
- ◆ Sensing angle: less than 15 degrees
- ◆ Input trigger pulse: 10us TTL level
- ◆ Output echo signal: output TTL level signal (high), proportional to range

4. Preparation

- Insert the micro:bit board into the slot of keystudio 4WD Mecanum Robot Car V2.0
- Place batteries into battery holder
- Dial power switch to ON end
- Connect the micro:bit to your computer via an USB cable
- Open the Web version of Makecode

Import Hex profile (**How to import?**) , or click “New Project” and drag blocks step by step(add MecanumRobot extension library first)

(How to add Mecanum_Robot extension?)

5. Test Code

File Type	Path	File Name
Hex file	..\2. Makecode Tutorial\Makecode Code	Project 18.1: Ultrasonic Ranging.hex

Or you could edit code step by step in the editing area.

(1)Tap “Advanced” → “Serial” → “serial redirect to USB”

Combine it with “on start” block



(2)Click “Advanced” → “Serial” to find and drag “serial write value x=0” into “forever” ; Click “MecanumRobot” to find and drag “Ultrasonic” to the 0 on the right side of “serial write value x=0” and change the x on the left side of “=” to distance:



(3) find and drag  of “Basic” ,and change 100 to 200and place it



behind

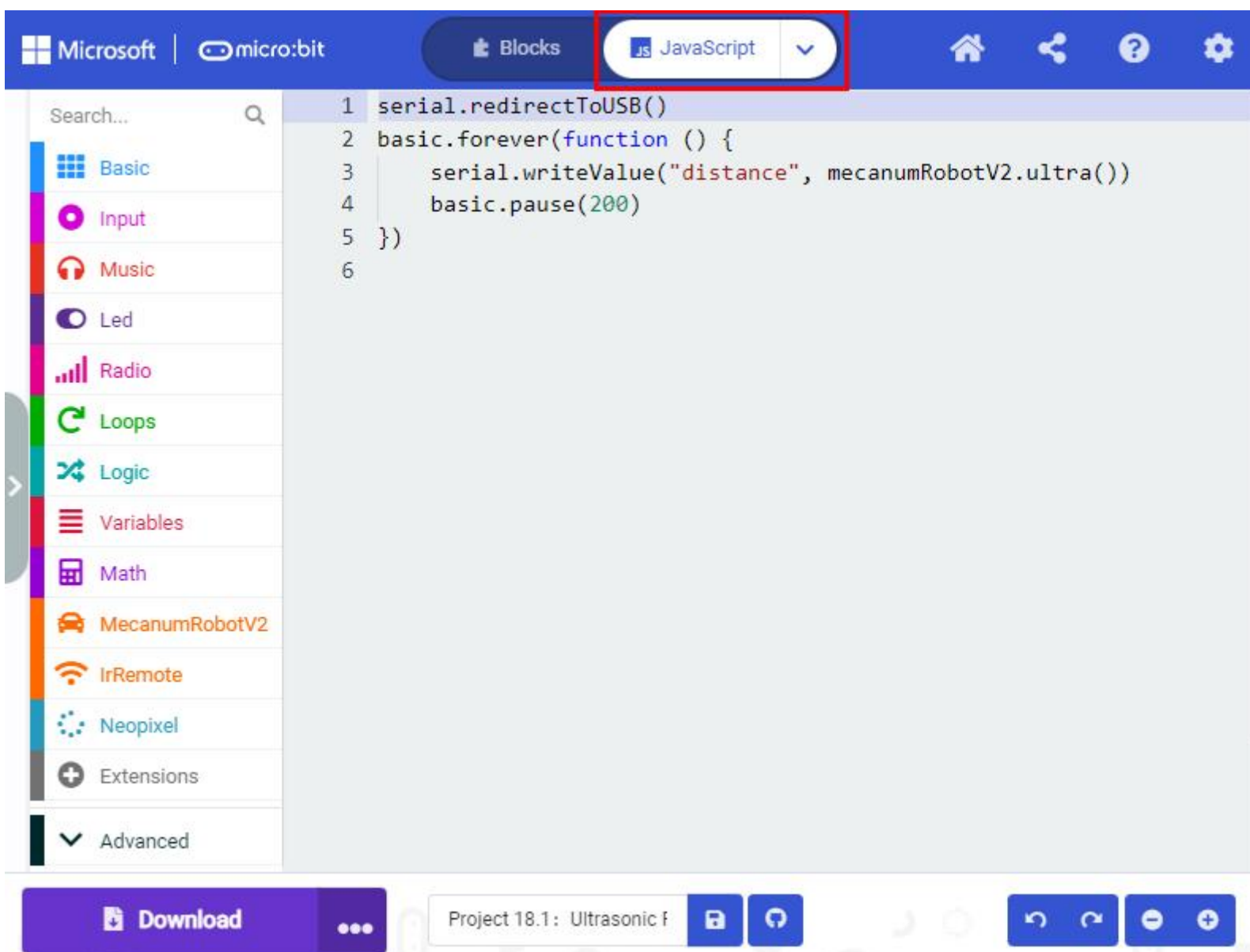
Complete Program:

keystudio



- ① The "on start" command block runs only once to start the program.
- ② Serial redirection USB
- ③ In the "forever" instruction block, the program runs cyclically.
- ④ Serial write value distance=Ultrasonic
- ⑤ Delay time 200 ms

Click "JavaScript" to view the corresponding JavaScript code: :



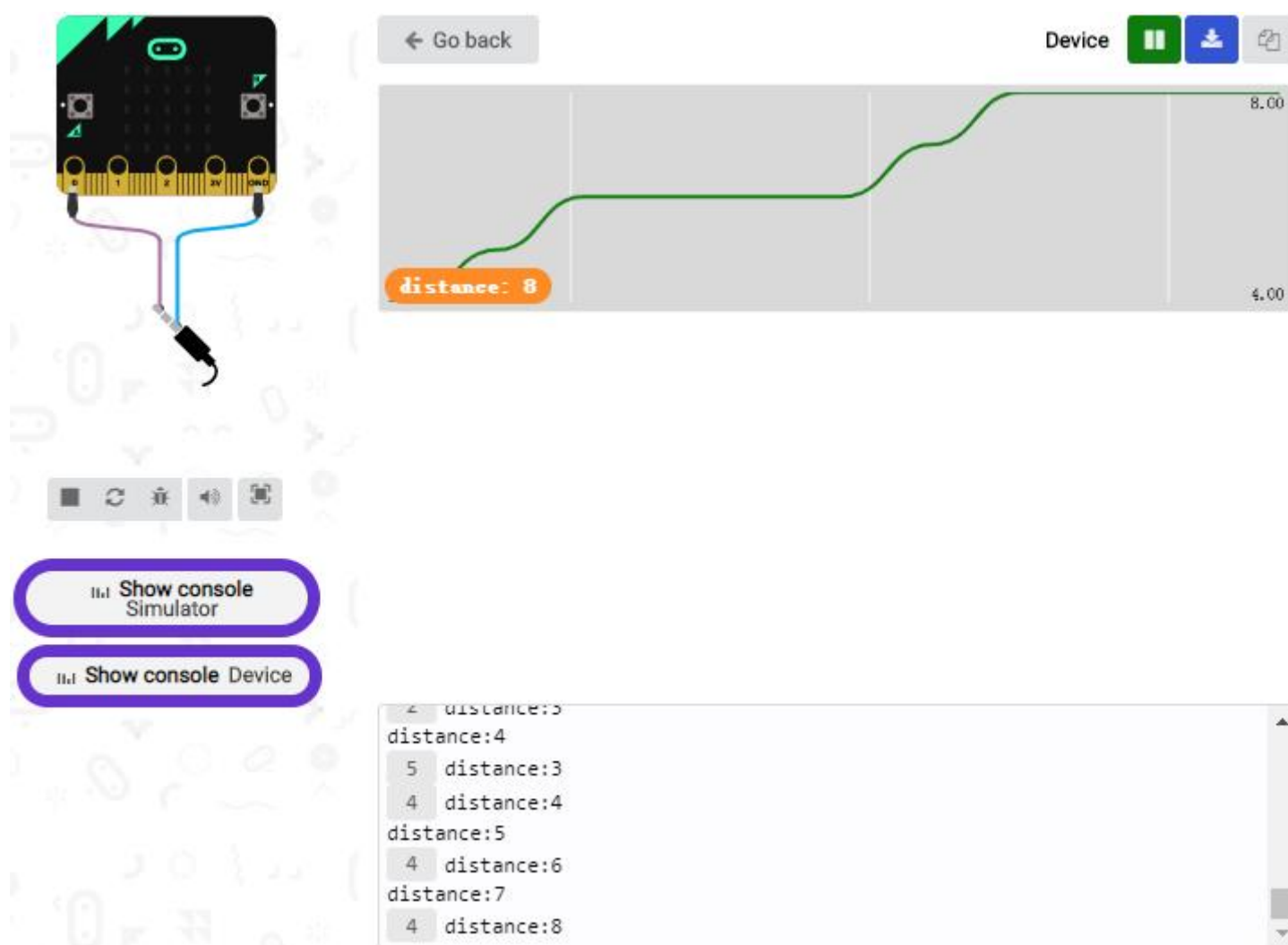
6. Test Result

Download code to micro:bit, keep USB cable connected, dial POWER switch to ON end. The distance value will be displayed on monitor.

(How to quick download?)

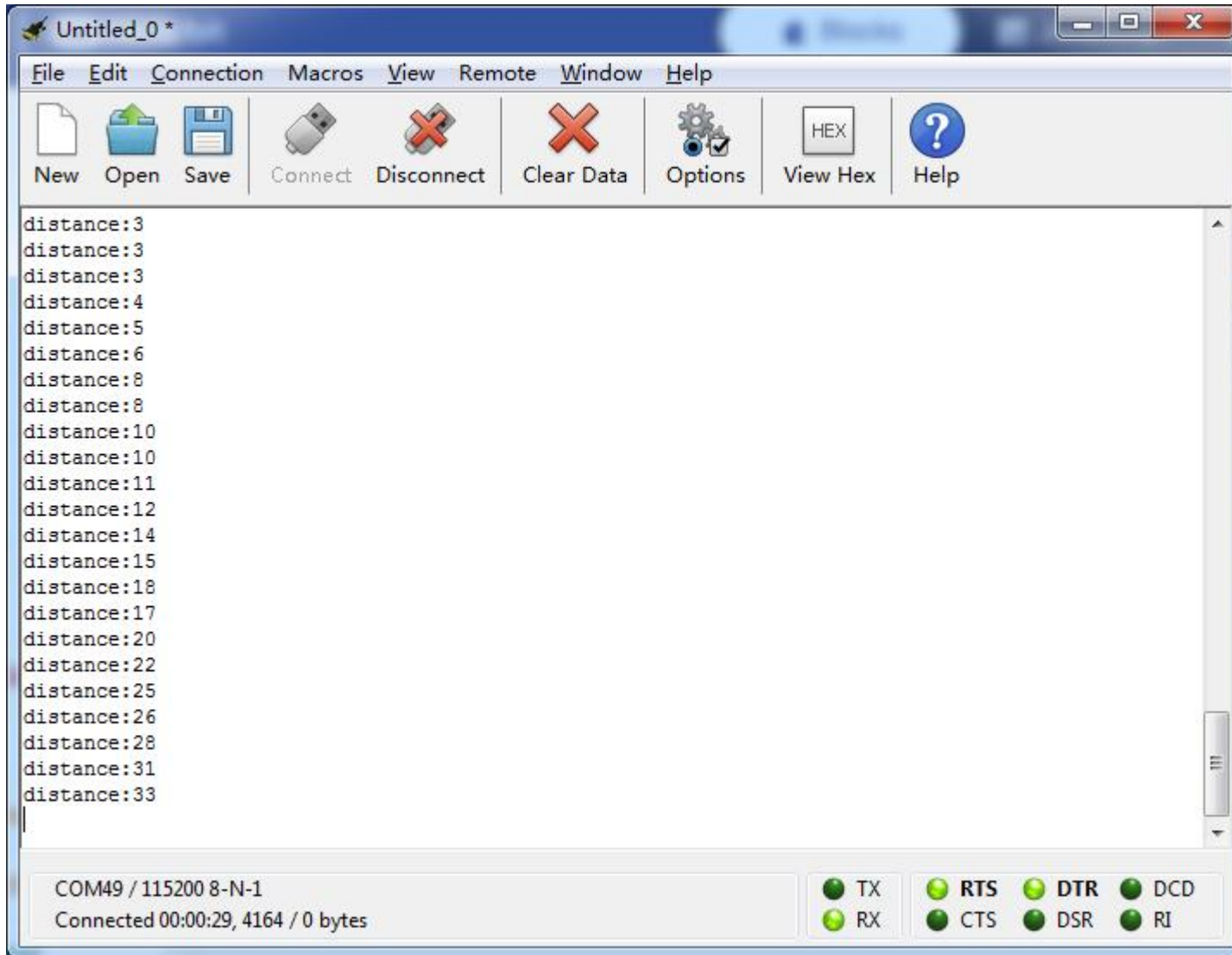


The monitor shows the distance between the obstacle and ultrasonic sensor(as shown below).

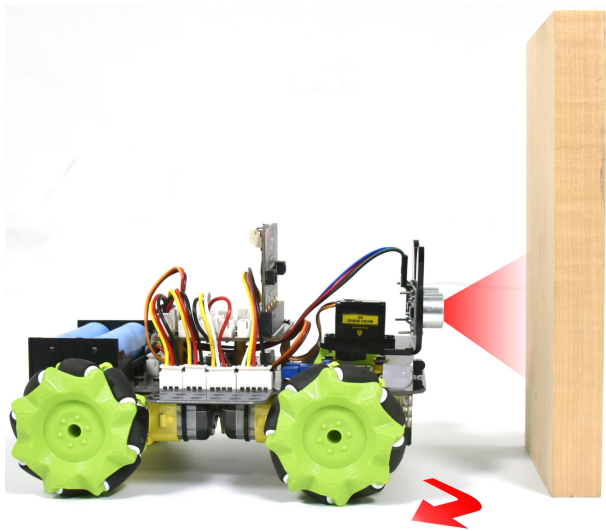


Open CoolTerm, click Options to select SerialPort. Set COM port and 115200 baud rate(the baud rate of USB serial communication of Micro:bit is 115200 through the test). Click "OK" and "Connect" .

CoolTerm serial monitor displays the distance value as follows:



Project 18.2: Ultrasonic Avoidance



1. Description

In this project, we will integrate an ultrasonic sensor and a car to make an ultrasonic avoidance car.

Its principle is to detect the distance between the car and obstacle via the ultrasonic sensor to control the motion of smart car.

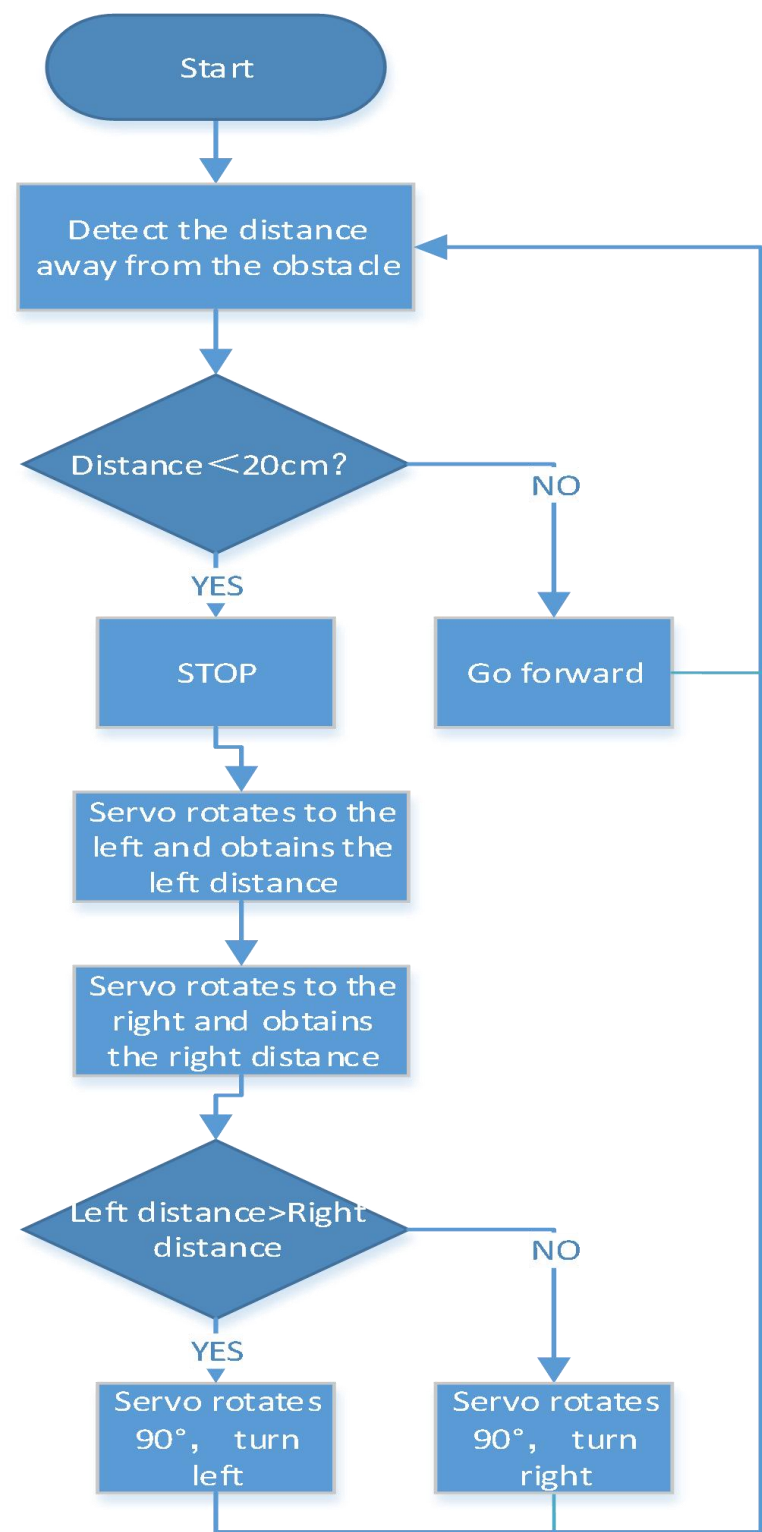
2. Preparation

- Insert the micro:bit board into the slot of keyestudio 4WD Mecanum Robot Car V2.0
- Place batteries into battery holder
- Dial power switch to ON end
- Connect the micro:bit to your computer via an USB cable
- Open the Web version of Makecode

Import Hex profile (**How to import?**) , or click “New Project” and drag blocks step by step(add MecanumRobot extension library first)

(How to add Mecanum_Robot extension?)

3. Flow Chart



4. Test Code

Code path:

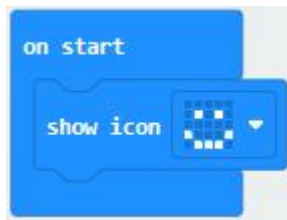
File Type	Path	File Name
Hex file	..\2. Makecode Tutorial\Makecode Code	Project 18.2: Ultrasonic Avoidance Car.hex

keystudio

Or you could edit code step by step in the editing area.

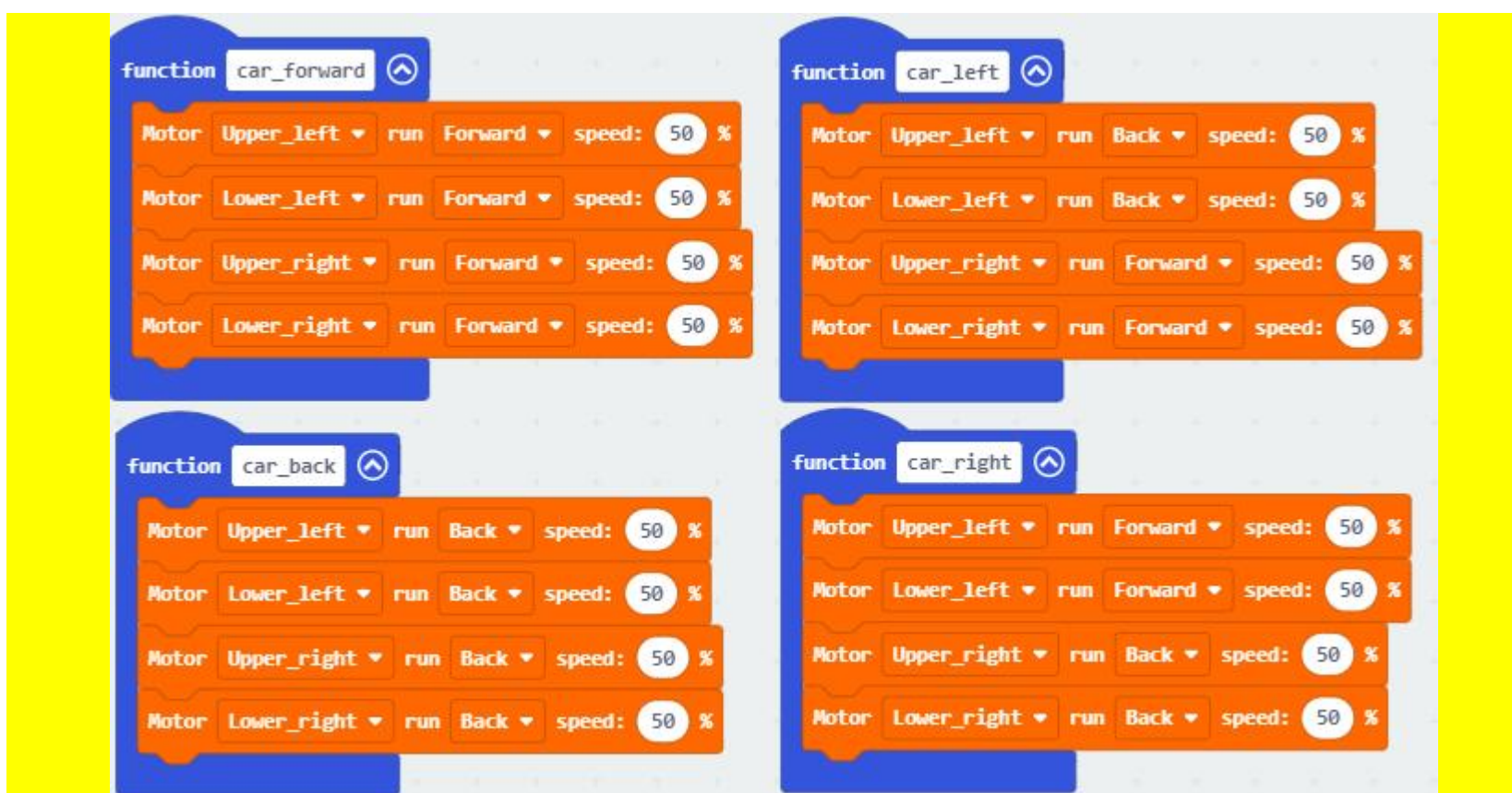
(1)Enter "Basic" → "show icon ♥ "

Place it into "on start" and click the triangle button to select " " pattern



Click "Functions" of "Advance" and click .

Change "doSomething" to "car_forward" , "car_back" , "car_left" , "car_right" respectively.



Click "Variables" and then click "Make a Variable...", dialog box "New variable name: " pops up;

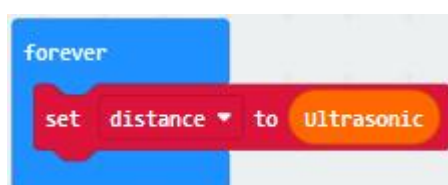
Fill it with "distance" ;

Click "OK" to establish variable "distance;

Set the functions of servo:

Click "Variables" , drag out "set distance to 0" and place it into "forever" .

Click "MecanumRobotV2" to find and drag "Ultrasonic" to the 0 behind the " to" :



keyestudio

Click "Logic" to find and drag "if true then...else" to "forever" ;

Find and drag "=" to "true" ;

Click "Variables" to find and drag "distance" on the left of "=" ;

Click the little triangle behind "=" to choose "<" ;

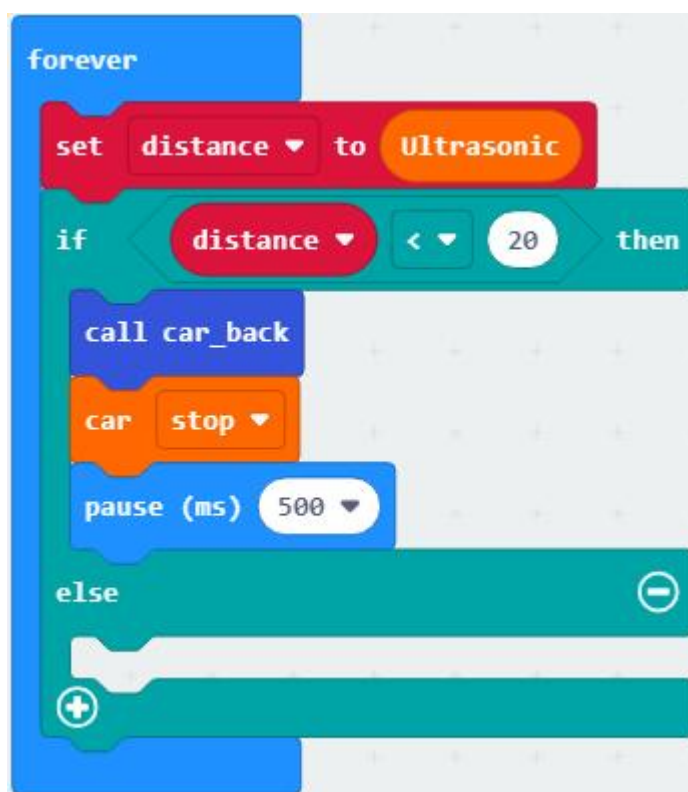
Change the 0 behind ">" to 20:



Click Functionsto of "Advance" to find and drag  ;

Click "MecanumRobot" to find and drag  to then;

Click "Basic" to find  and change the 100 to 500:



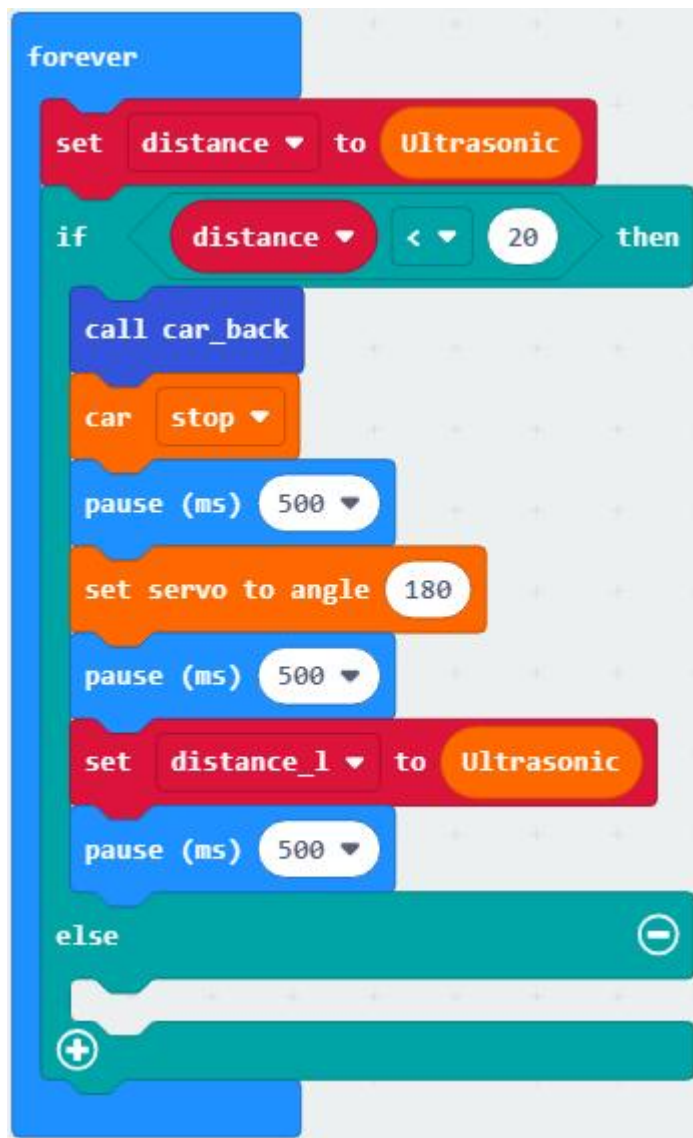
Click "MecanumRobot" to find and drag  and change the 0 to 180;

Copy  once;

Click "Variables" to find and drag "set distance_l to 0" ;

Click "MecanumRobotV2" to find and drag "Ultrasonic" to 0 behind "to"  ;

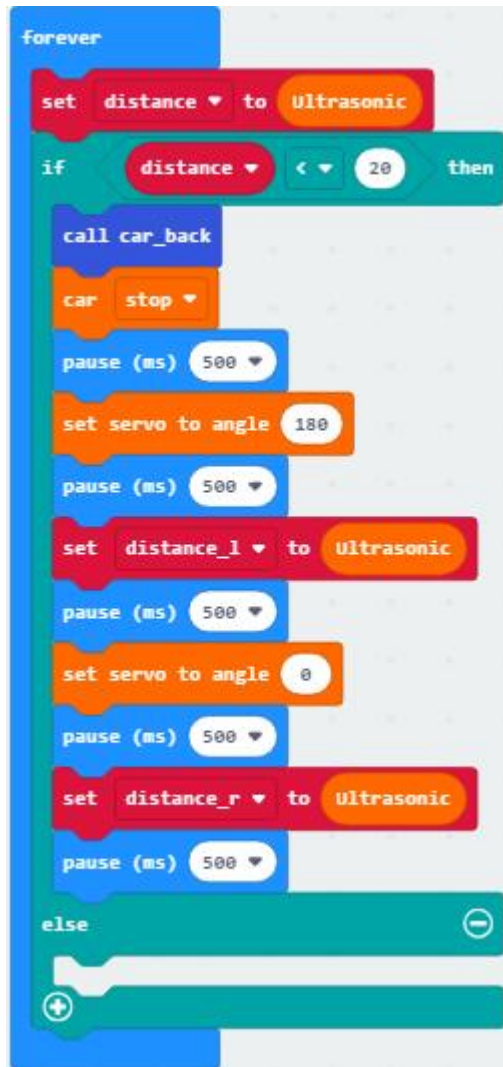
Copy  once;



Copy 


 once;

Change the 180 to 0, distance_l to distance_r and others remain unchanged:

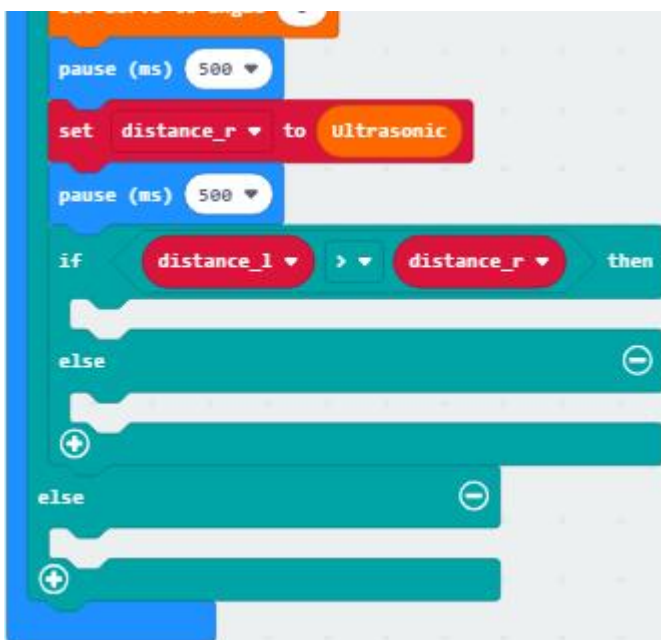


Click "Logic" to find and drag "if true then...else" ;

Find and drag "=" to true;

Click "Variables" to find and drag "distance_l to the left of "=" ; Click the little triangle behind "=" to choose ">" ;

Change the 0 behind ">" to "distance_r" :

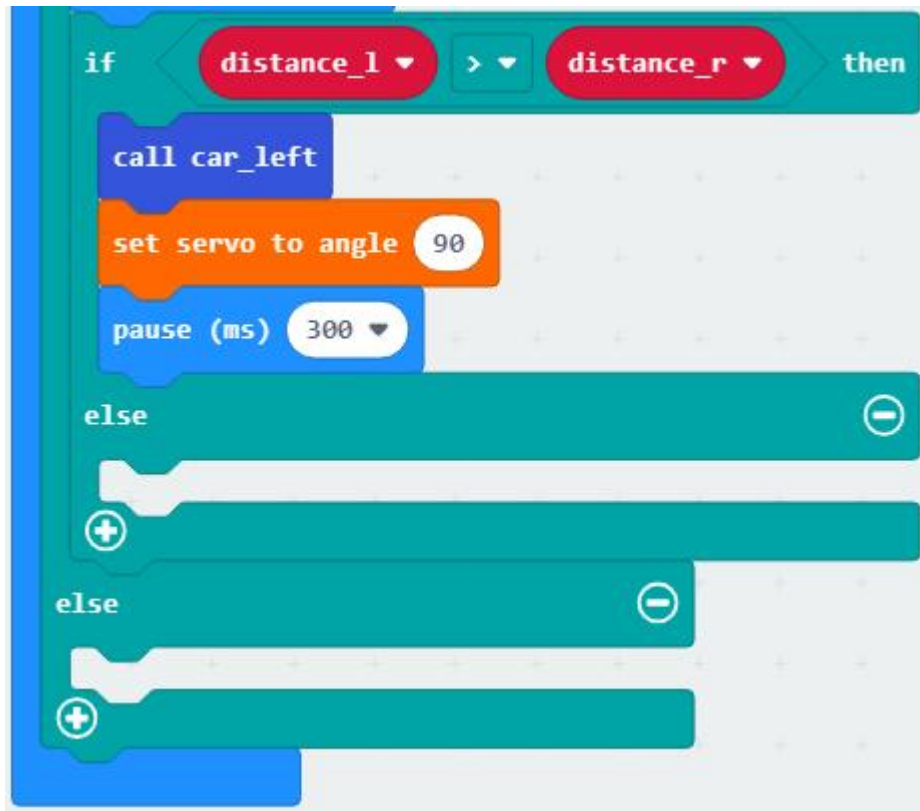


Click Funtions of "Advance" to find and drag call car_left ;

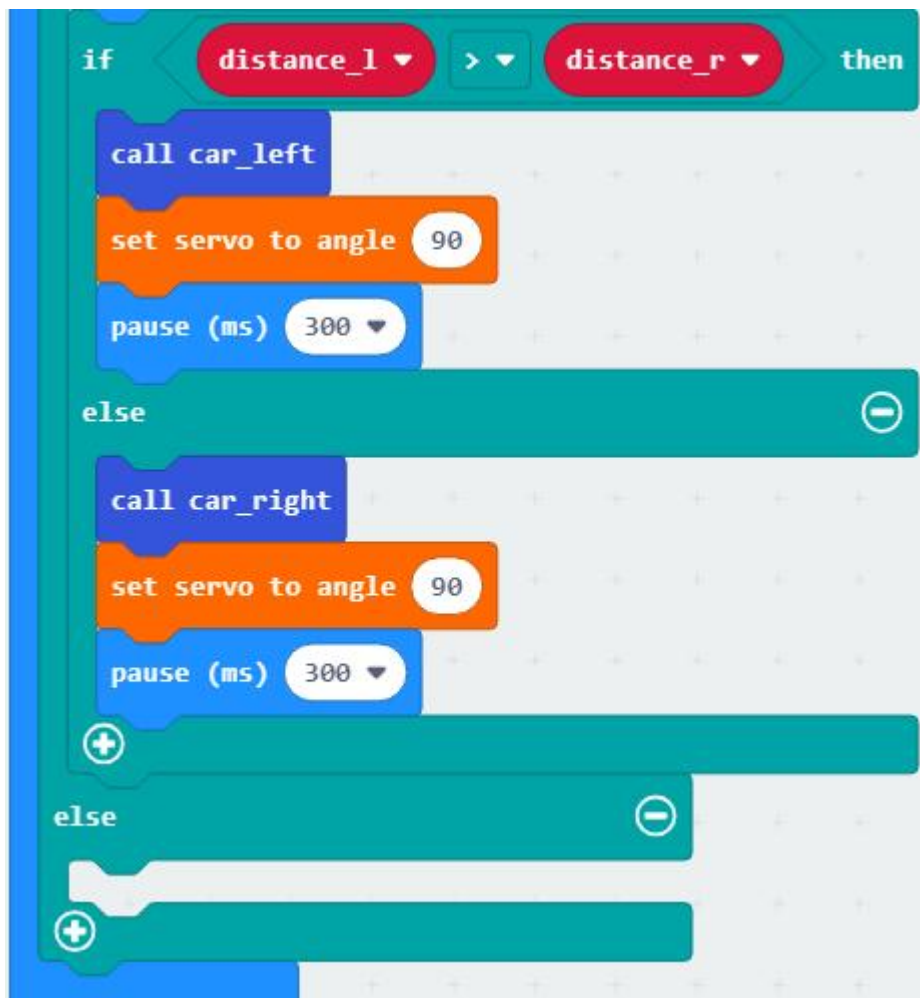
Click "MecanumRobotV2" to find and drag set servo to angle 0 ;

Change the 0 to 90;

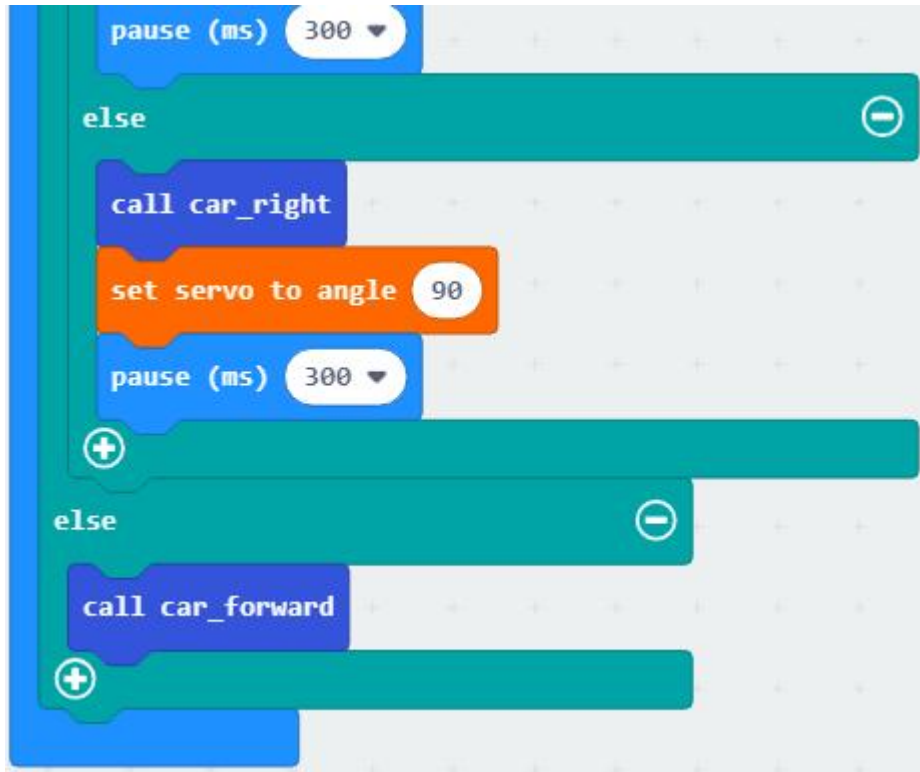
Click "Basic" to find and drag pause (ms) 100 and change the 100 to 300:



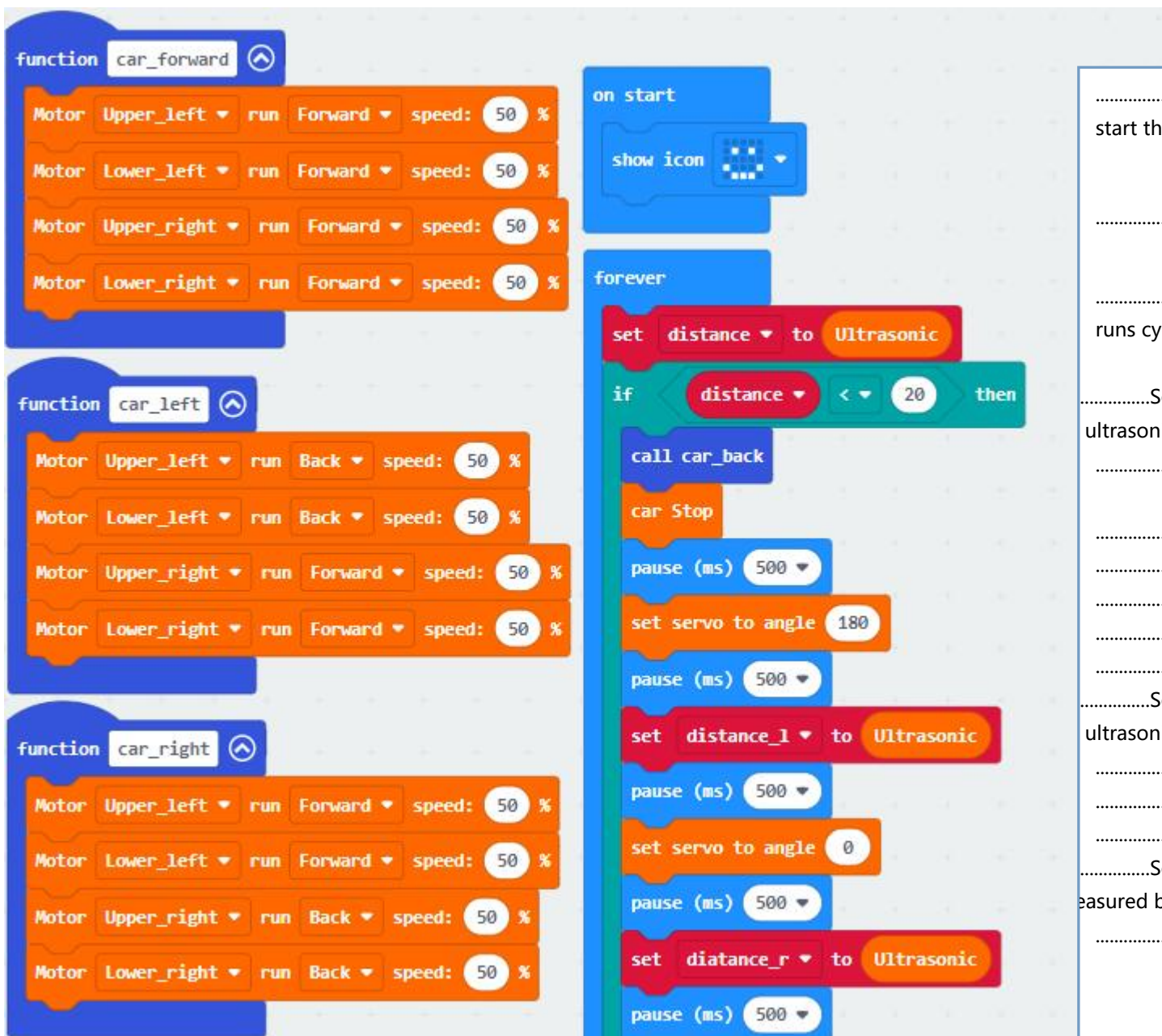
Change `call car_left` to `call car_right` and place it in the first "else" :



Click "Funtions" of "Advance" to find and drag `call car_forward` , and place it to the second "else" :



Complete Program:



....."The "on start" command block runs only once to start the program.

.....LED dot matrix display " " 

.....In the "forever" instruction block, the program runs cyclically.

.....Set variable distance to store the distance measured ultrasonic

.....If the distance is less than 20cm

.....The car moves back

.....The car stop

.....Delay in 500ms

.....Servo rotates left

.....Delay in 500ms

.....Set variable distance_l to store the distance measured ultrasonic

.....Delay in 500ms

.....Servo rotates right

.....Delay in 500ms

.....Set variable distance_r to store the distance measured by ultrasonic

.....Delay in 500ms.

keystudio

.....If the left distance is bigger than right

.....The car turns left

.....Servo rotates to 90 degrees

.....Delay in 300ms

.....If the left distance is equal to or less than right

.....The car turns right

.....Servo rotates to 90 degrees

.....Delay in 300ms

.....If the front distance is equal to or bigger than 20cm

.....The car advance

Click "JavaScript" to view the corresponding JavaScript code: :

```

1 function car_back () {
2     mecanumRobotV2.Motor(LR.Upper_left, MD.Back, 50)
3     mecanumRobotV2.Motor(LR.Lower_left, MD.Back, 50)
4     mecanumRobotV2.Motor(LR.Upper_right, MD.Back, 50)
5     mecanumRobotV2.Motor(LR.Lower_right, MD.Back, 50)
6 }
7 function car_left () {
8     mecanumRobotV2.Motor(LR.Upper_left, MD.Back, 50)
9     mecanumRobotV2.Motor(LR.Lower_left, MD.Back, 50)
10    mecanumRobotV2.Motor(LR.Upper_right, MD.Forward, 50)
11    mecanumRobotV2.Motor(LR.Lower_right, MD.Forward, 50)
12 }
13 function car_forward () {
14     mecanumRobotV2.Motor(LR.Upper_left, MD.Forward, 50)
15     mecanumRobotV2.Motor(LR.Lower_left, MD.Forward, 50)
16     mecanumRobotV2.Motor(LR.Upper_right, MD.Forward, 50)
17     mecanumRobotV2.Motor(LR.Lower_right, MD.Forward, 50)
18 }
19 function car_right () {
20     mecanumRobotV2.Motor(LR.Upper_left, MD.Forward, 50)
21     mecanumRobotV2.Motor(LR.Lower_left, MD.Forward, 50)
22     mecanumRobotV2.Motor(LR.Upper_right, MD.Back, 50)
23     mecanumRobotV2.Motor(LR.Lower_right, MD.Back, 50)
24 }
  
```

```

25 let distance_r = 0
26 let distance_l = 0
27 let distance = 0
28 basic.showIcon(IconNames.Happy)
29 basic.forever(function () {
30     distance = mecanumRobotV2.ultra()
31     if (distance < 20) {
32         car_back()
33         mecanumRobotV2.state()
34         basic.pause(500)
35         mecanumRobotV2.setServo(180)
36         basic.pause(500)
37         distance_l = mecanumRobotV2.ultra()
38         basic.pause(500)
39         mecanumRobotV2.setServo(0)
40         basic.pause(500)
41         distance_r = mecanumRobotV2.ultra()
42         basic.pause(500)
43         if (distance_l > distance_r) {
44             car_left()
45             mecanumRobotV2.setServo(90)
46             basic.pause(300)
47         } else {
48             car_right()
49             mecanumRobotV2.setServo(90)
50             basic.pause(300)
51         }
52     } else {
53         car_forward()
54     }
55 })
56

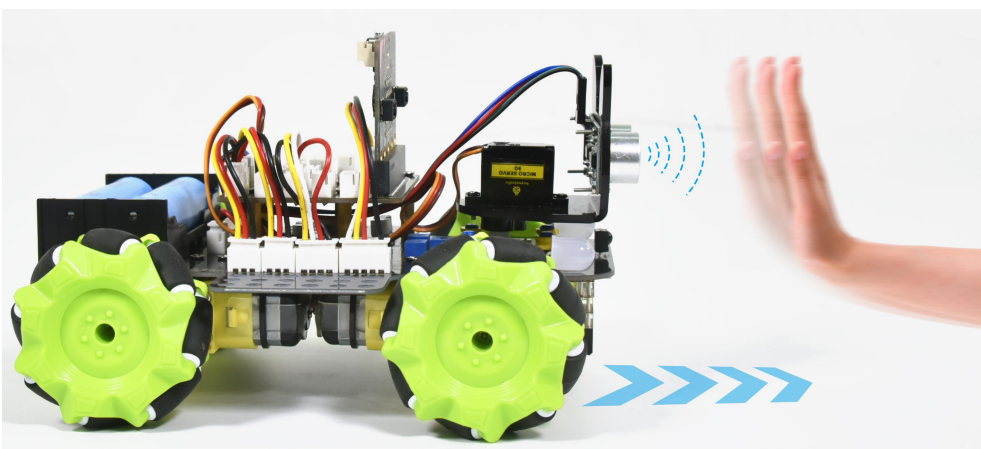
```

5. Test Result

Download code to micro:bit, dial to ON end, and dial POWER to ON end. When the obstacle distance is greater than 20cm, the car goes forward ; on the contrary, smart car turns left.

(How to download? How to quick download?)

Project 18.3: Ultrasonic Following



1. Description

In previous lesson, we' ve learned the basic principle of line tracking sensor. Next, we will combine the ultrasonic sensor with the car to make an ultrasonic following car.

The ultrasonic sensor detects the obstacle distance and control the motion status of car.

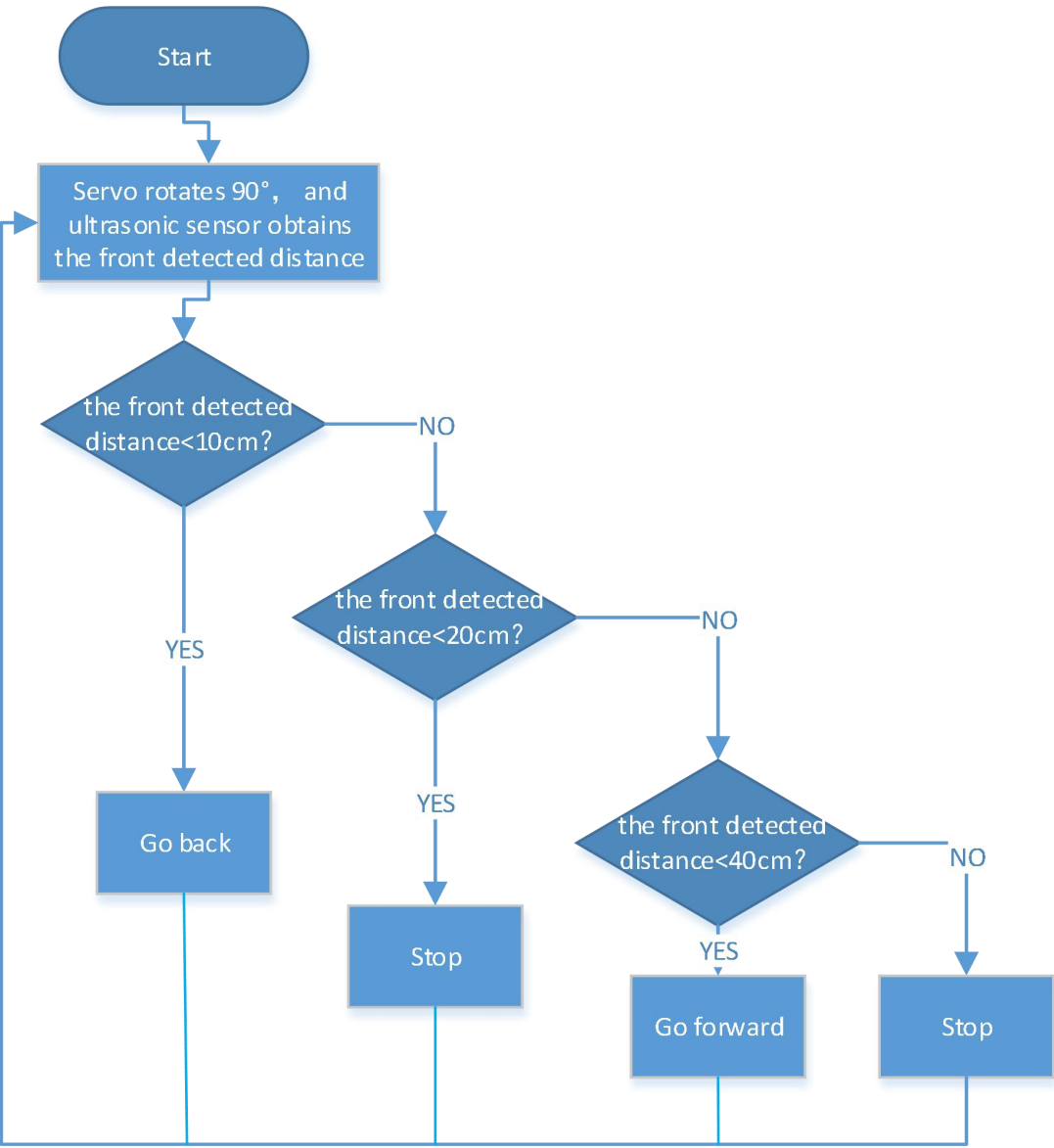
2. Preparation

- Insert the micro:bit board into the slot of keystudio 4WD Mecanum Robot Car V2.0
- Place batteries into battery holder
- Dial power switch to ON end
- Connect the micro:bit to your computer via an USB cable
- Open the Web version of Makecode

Import Hex profile (**How to import?**) , or click “New Project” and drag blocks step by step(add MecanumRobot extension library first)

(How to add Mecanum_Robot extension?)

3. Flow Chart



4. Test Code

Code path:

File Type	Path	File Name
-----------	------	-----------

keystudio

Hex file	..\2. Makecode Tutorial\Makecode Code	Project 18.3: Ultrasonic Follow Smart Car.hex
----------	--	--

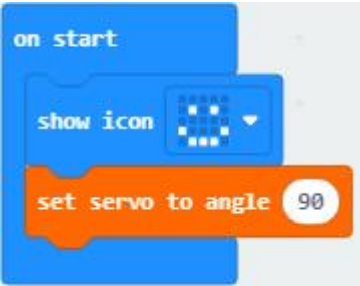
Or you could edit code step by step in the editing area.

(1)Enter “Basic” → “show icon ♥ ”

Place it into “on start” and click the triangle button to select “ ” pattern.

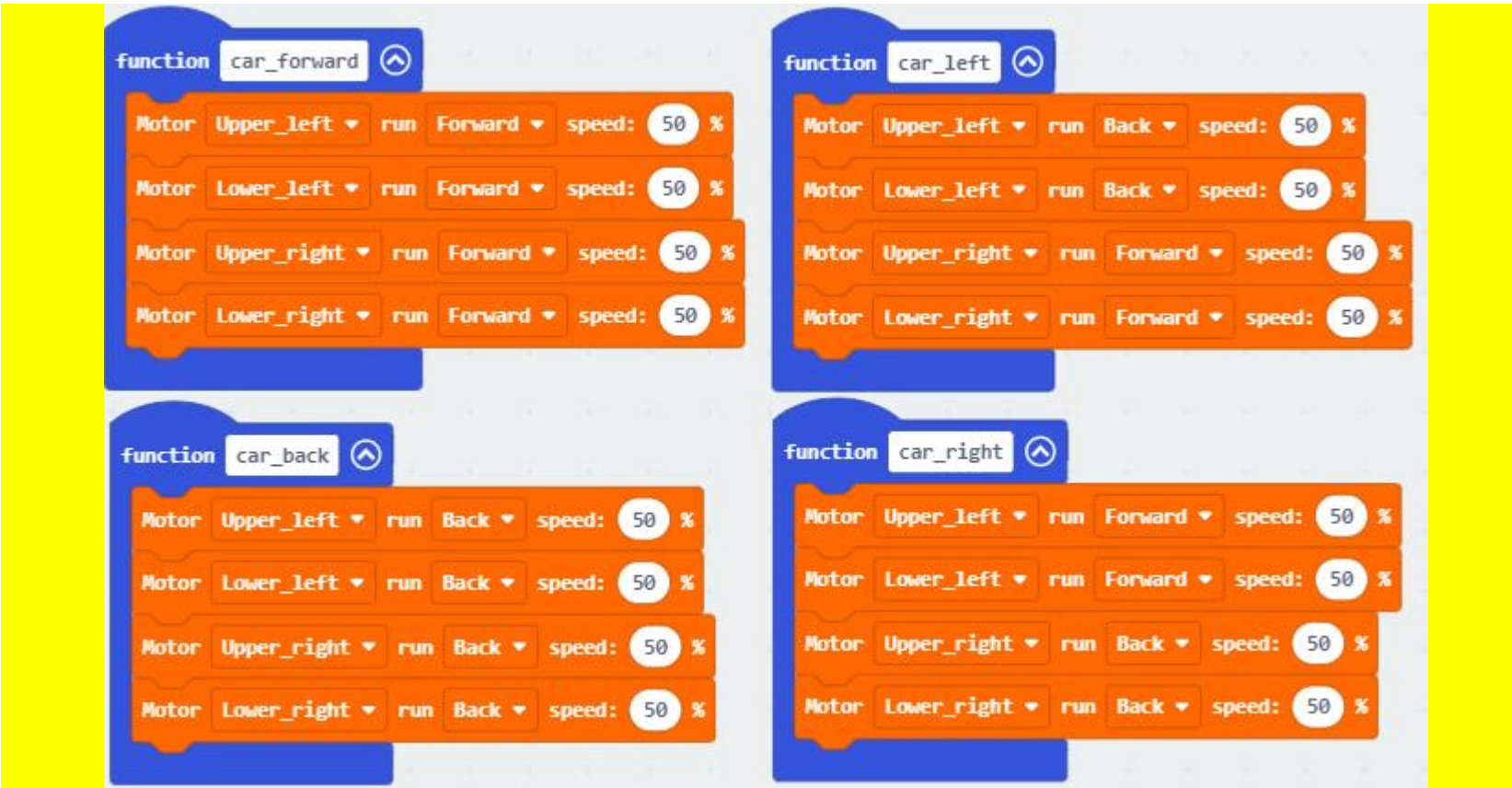


(2)Click “ MecanumRobotV2” to find and drag  to “on start” and change the angle 0 to 90:



Click “Functions” of “ Advance” and click “Make a Function” , then  will appear.

Change “doSomething” to “car_forward” , “car_back” , “car_left” , “car_right” respectively.



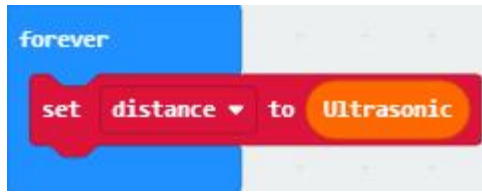
Click “Variables” and then click “Make a Variable...” , the dialog box “New variable name: ” pops up; fill it with “distance” ;

keyestudio

Click "OK" to establish variable "distance" ;

Drag "set distance to 0" to "forever" ;

Click "MecanumRobotV2" to find and drag  to the "0" of "set distance to 0" :



Click "Logic" to find and drag "if true then...else" to "forever" ;

Find and drag "=" to true;

Click "Variables" to find and drag "distance" to the left side of "=" ;

Click the little triangle behind "=" to choose "<" ;

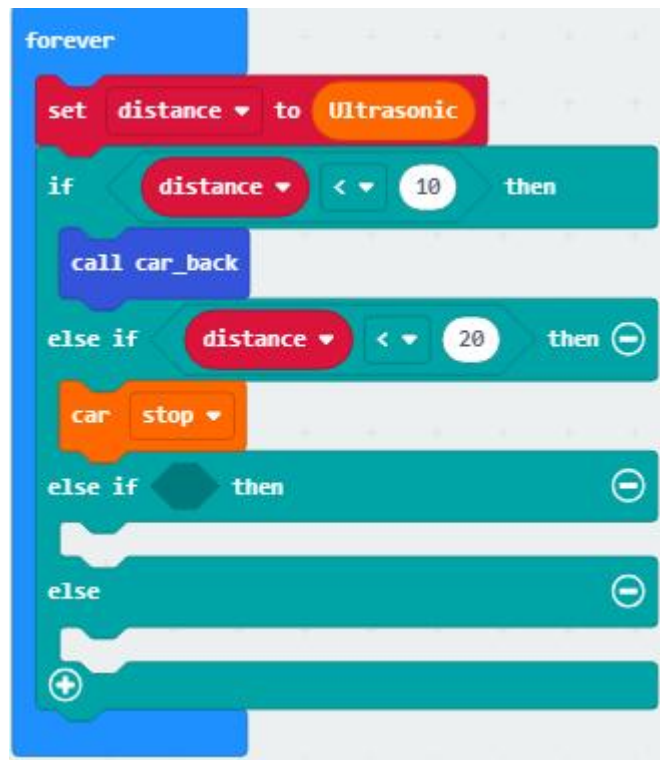
Change the 0 behind "" to 10:



Click "Functions" of "Advance" to find and drag  to "then" :



Change the 10 to 20, car_back to car stop:

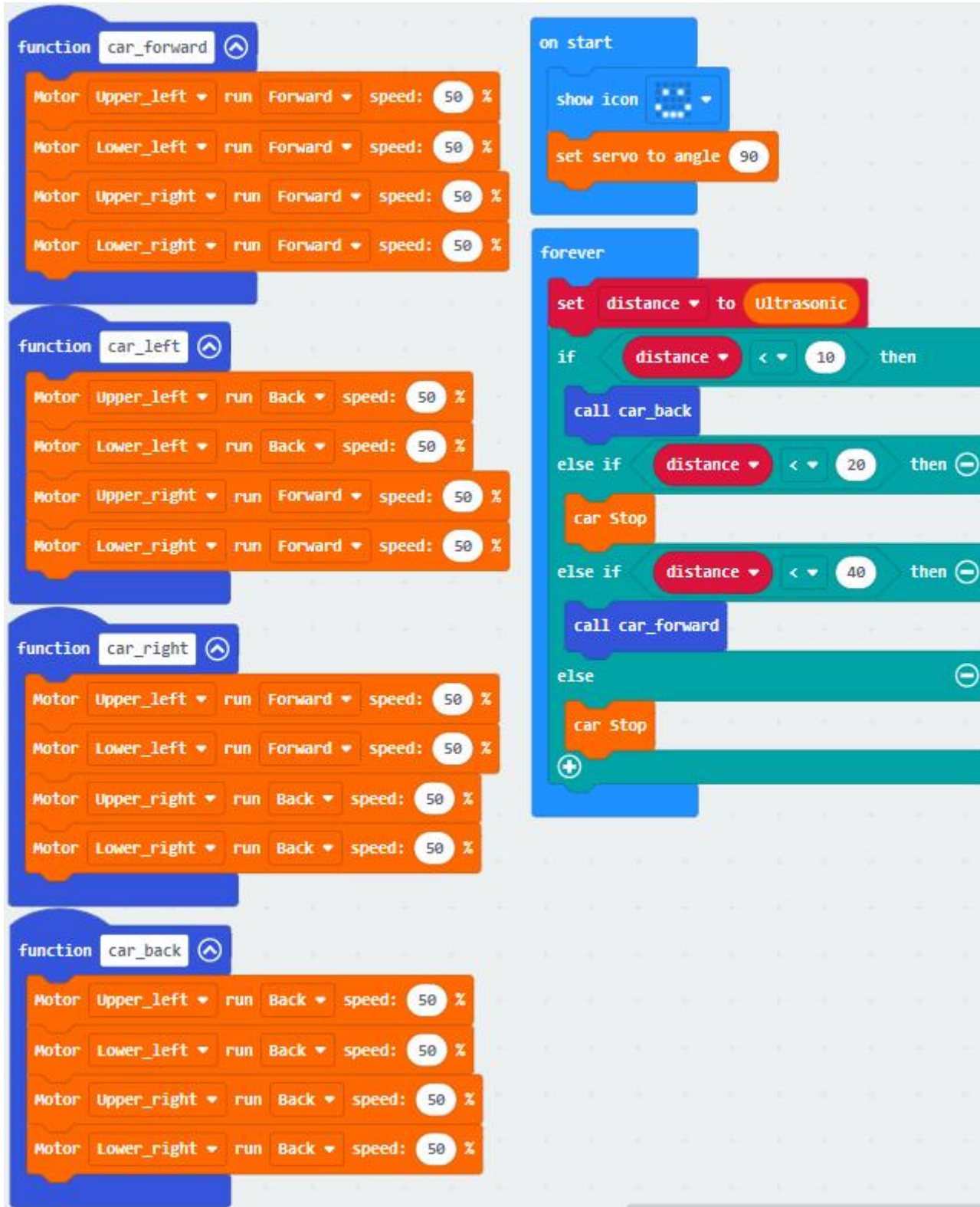


Change the 20 to 40, car stop to car forward; Place car stop to the last " else" :



Complete Program:

keystudio



....."The "on start" command block runs only once to start the program.

.....LED dot matrix display "  "

.....Servo rotates to 90 degrees

.....In the "forever" instruction block, the program runs cyclically.

.....Set variable distance to the value read by ultrasonic

.....If the distance is less than 10cm

.....The car moves back

.....If the distance is less than 20cm

.....The car stop

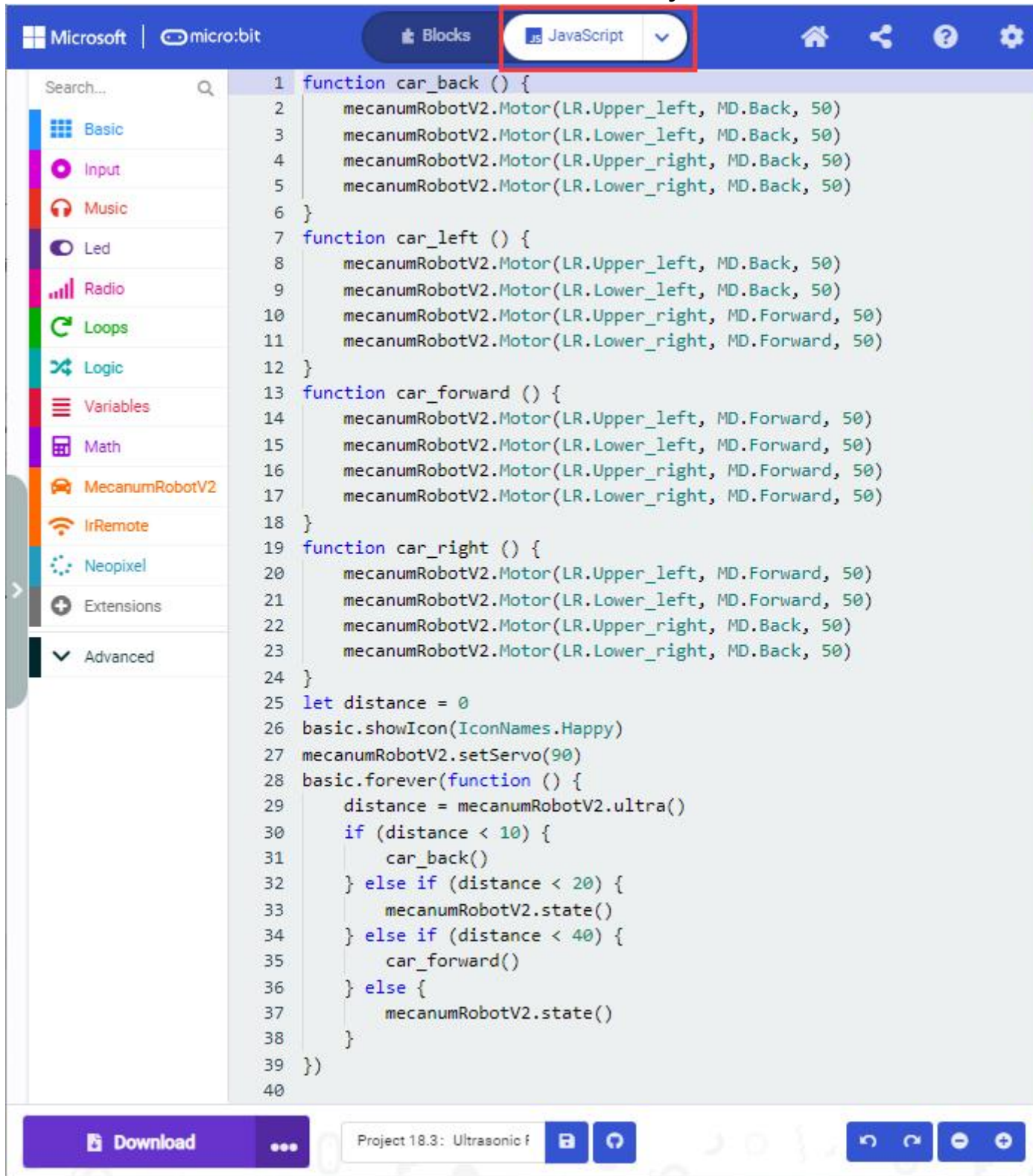
.....If the distance is less than 40cm

.....The car advance

.....Otherwise, none of the above conditions holds

.....The car stop

Click "JavaScript" to view the corresponding JavaScript code: :



5. Test Result

Download code to micro:bit, dial POWER switch to ON end on shield, smart car could follow the obstacle to move.

(How to download? How to quick download?)

Project 19: IR Remote Control

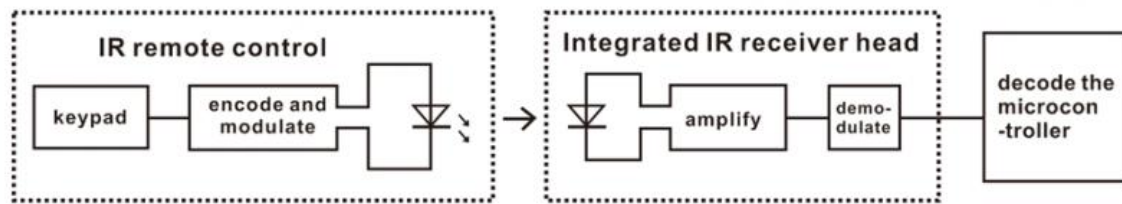
Project 19.1: Decode IR Remote Control



1. Description

There is no doubt that infrared remote control is ubiquitous in daily life. It is used to control various household

appliances, such as TVs, stereos, video recorders and satellite signal receivers. Infrared remote control is composed of infrared transmitting and infrared receiving systems, that is, an infrared remote control, an infrared receiving module and a single-chip microcomputer capable of decoding.



The 38K infrared carrier signal emitted by remote controller is encoded by the encoding chip in the remote controller. It is composed of a section of pilot code, user code, user inverse code, data code, and data inverse code. The time interval of the pulse is used to distinguish whether it is a 0 or 1 signal and the encoding is made up of these 0, 1 signals.

The user code of the same remote control is unchanged. The data code can distinguish the key.

When the remote control button is pressed, the remote control sends out an infrared carrier signal. When the IR receiver receives the signal, the program will decode the carrier signal and determines which key is pressed. The MCU decodes the received 01 signal, thereby judging what key is pressed by the remote control.

Infrared receiver we use is an infrared receiver module. Mainly composed of an infrared receiver head, it is a device that integrates reception, amplification, and demodulation. Its internal IC has completed demodulation, and can achieve from infrared reception to output and be compatible with TTL signals. Additionally, it is suitable for infrared remote control and infrared data transmission. The infrared receiving module made by the receiver has only three pins, signal line, VCC and GND.

According to the picture above, the integrated port of the infrared receiver is connected to the P9 5V G port on the motor driver board and controlled by the the P9 of the micro:bit.

2. Parameters:

- Operating Voltage: 3.3-5V (DC)
- Interface: 3PIN
- Output Signal: Digital signal
- Receiving Angle: 90 degrees
- Frequency: 38khz
- Receiving Distance: about 5m

3. Preparation

- Insert the micro:bit board into the slot of keyestudio 4WD Mecanum Robot Car V2.0
- Place batteries into battery holder

- Dial power switch to ON end
- Connect the micro:bit to your computer via an USB cable
- Open the Web version of Makecode

Import Hex profile (**How to import?**) , or click “New Project” and drag blocks step by step(add MecanumRobot extension library first)

(How to add Mecanum_Robot extension?)

4. Test Code

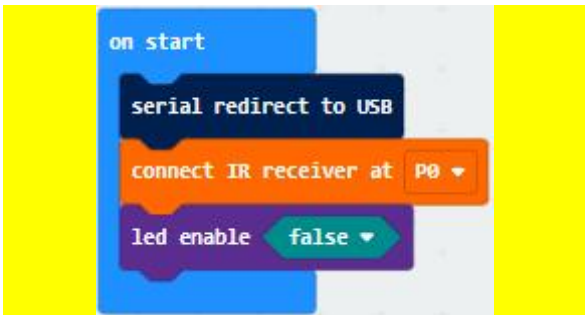
File Type	Path	File Name
Hex file	..\2. Makecode Tutorial\Makecode Code	Project 19.1 : Decode IR Remote Control.hex

Click “Advanced” → “Serial” → “serial redirect to USB”
Place it into “on start” block.

Enter “IrRemote” → “connect IR receiver at P0”
Put it into “on start” block
IR receiving module is controlled by P0 of micro:bit board



Click “Led” → “more” → “led enable(false)” , and place it into “on start” .



Go to “Variables” → “Make a Variable...” → “New variable name: ” dialog box,
Enter “val” and click “OK” to create variable “val”
Then drag out “set val to 0” block into “forever” block.



Go to "Ir Remote" → "IR button"

Place it into 0 box



Click "Advanced" → "Serial" → "serial write value "x" =0"

Put it into "forever" block

Change "x" into "IR"

Enter "Variables" to move block "val" into 0 box behind "="



Drag out block "pause (ms) 100" from "Basic" and delay in 1000ms

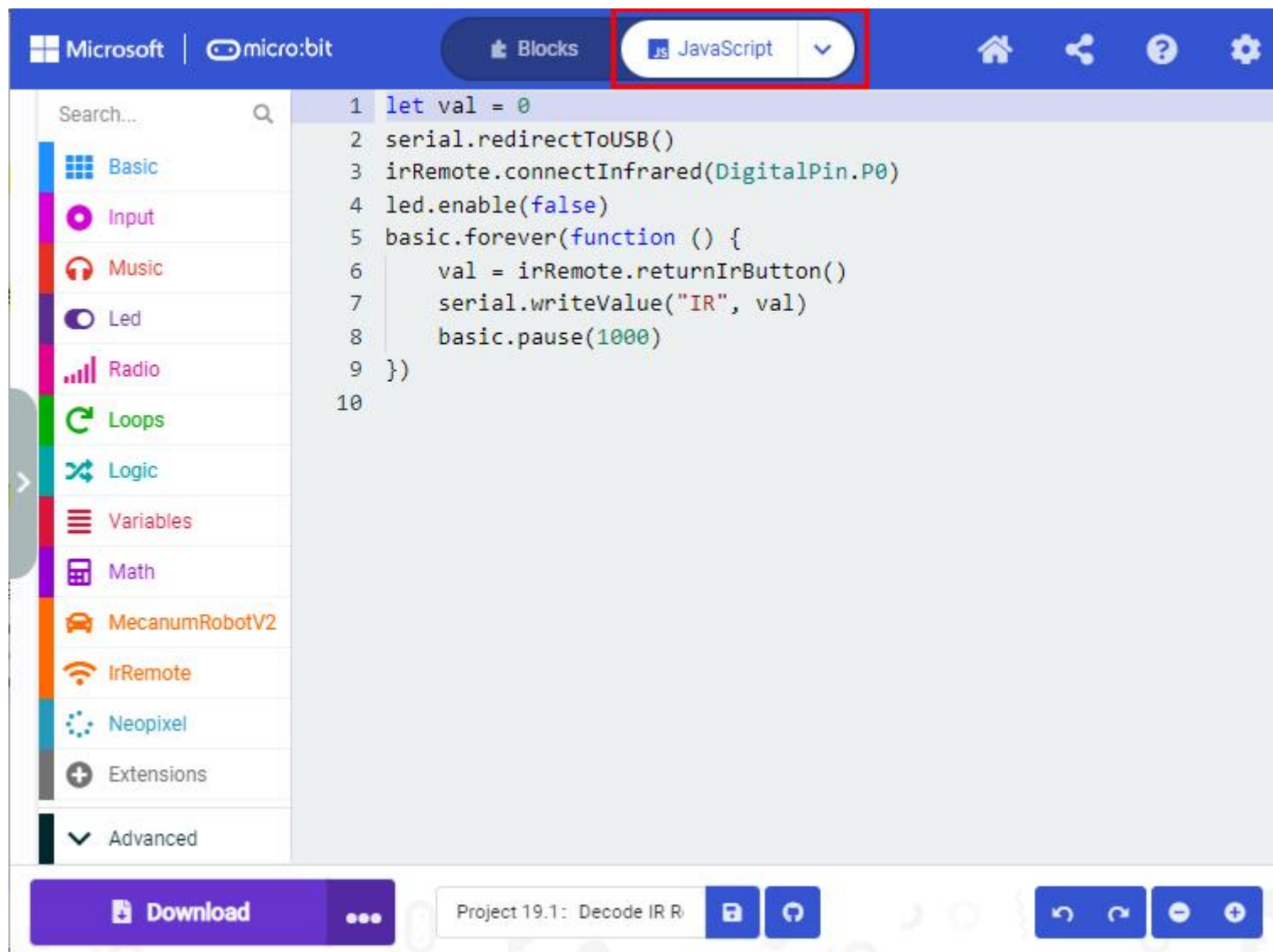
Leave it into "forever" block



Complete Program:

	"on start" : command block runs once to start program. Serial redirect to USB Connect IR receiver to P0 The program under the block "forever" runs cyclically. Set val to IR button Serial port prints IR=val Delay in 1000ms
--	--

Click "JavaScript" to switch into the corresponding JavaScript code:



Code explanation: If the buttons are not pressed, the serial monitor constantly shows 0; when pressed, the corresponding key values are displayed.

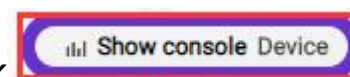
Notes:

The remote control in this kit is not inclusive of batteries. We recommend you to purchase them online.(battery type:CR2025).

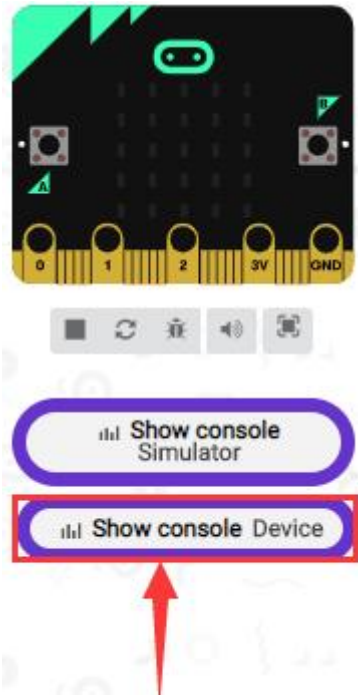
Make sure IR remote is good before test. There is a tip for you to check it.

Open the cellphone camera , make IR remote control point at camera and press button. The remote control is good if you see the purple flashing light in the camera.

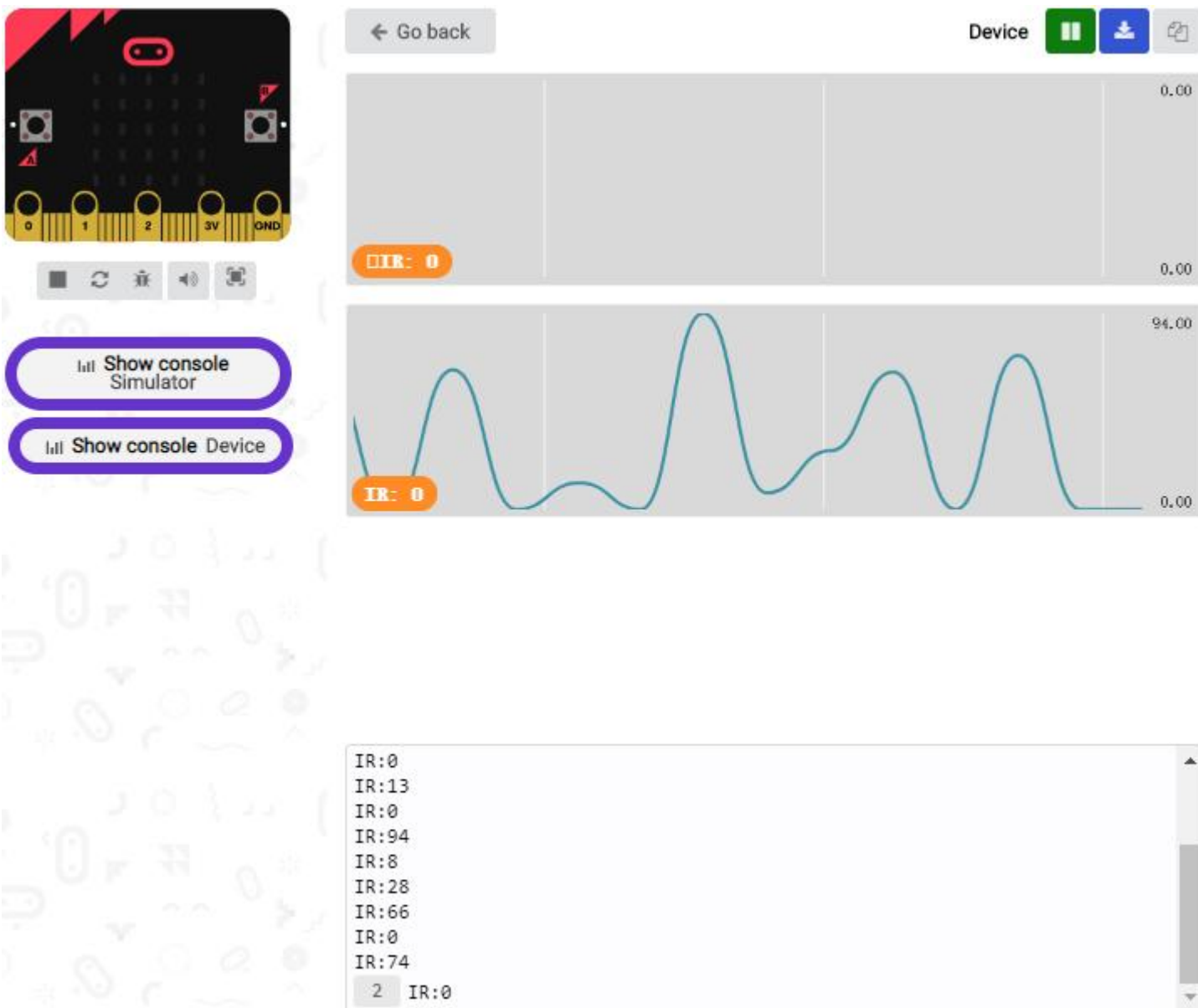
Download code to micro: bit board and don' t plug off USB cable Click



(How to quick download?)

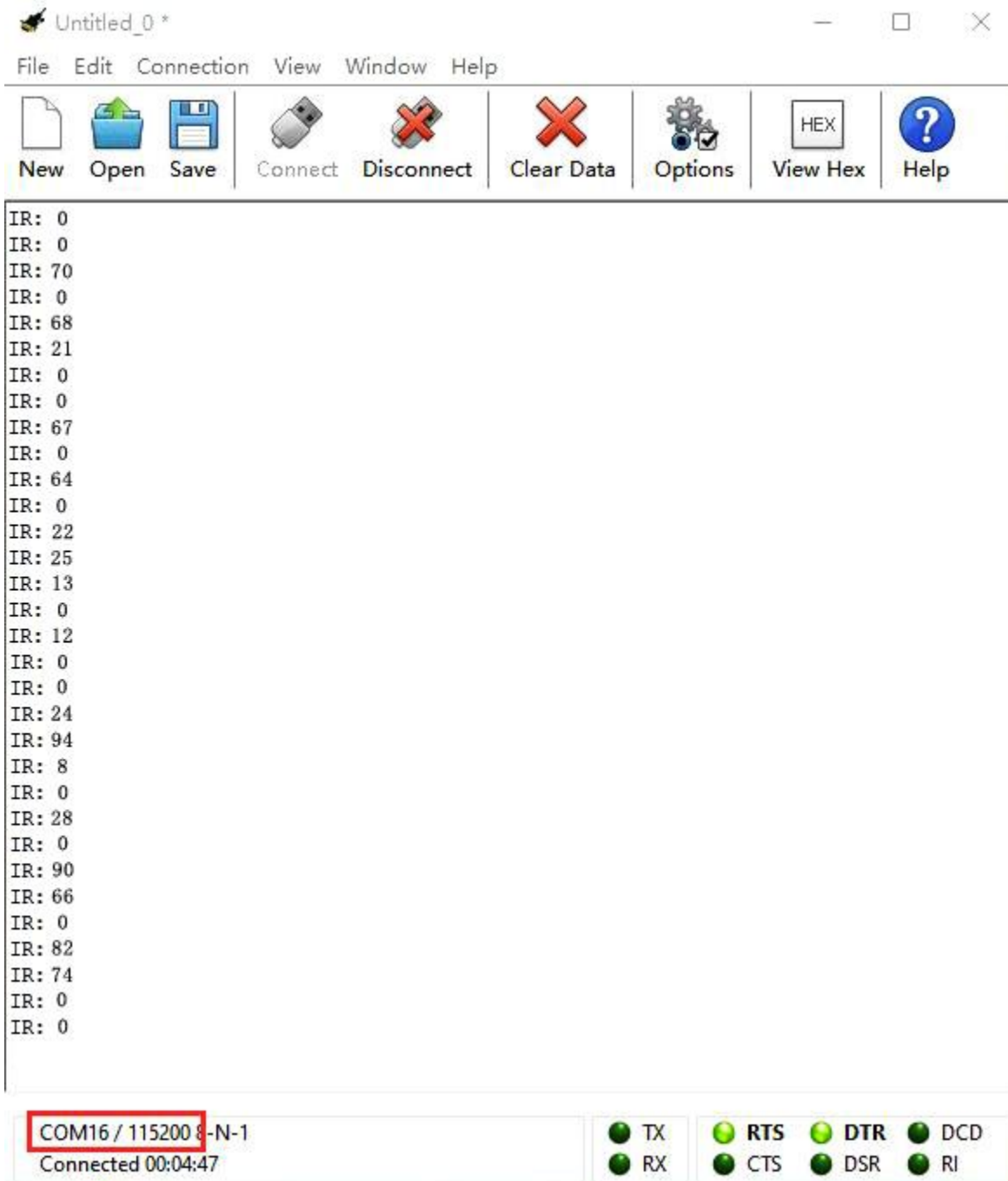


Make IR remote control point at IR receiver and press the button, the serial monitor will display the corresponding key values, as shown below:

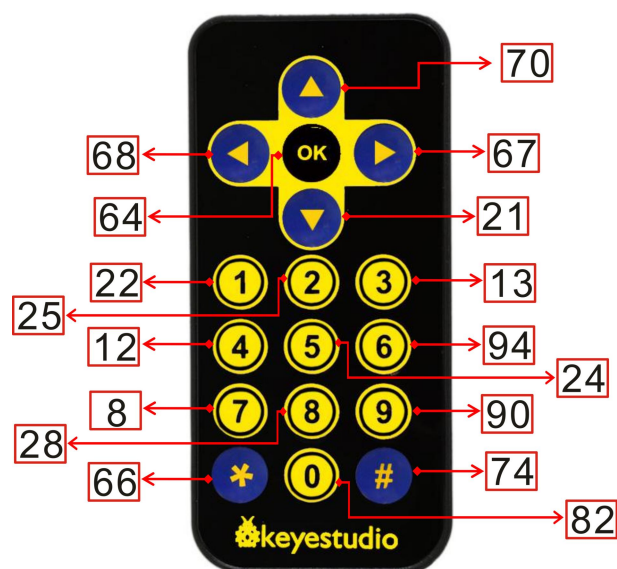


Open CoolTerm, click Options to select SerialPort. Set COM port and 115200 baud rate. Click "OK" and "Connect". CoolTerm serial monitor shows the key value as follows:

keyestudio



The key value is displayed as for your reference:



Project 19.2: IR Remote Control



1. Description

In this project, we combine IR remote control with car shield to make an IR remote smart car. Its principle is to control the motion of car by sending key signals from IR remote control to IR receiving module of car shield.

2. Preparation

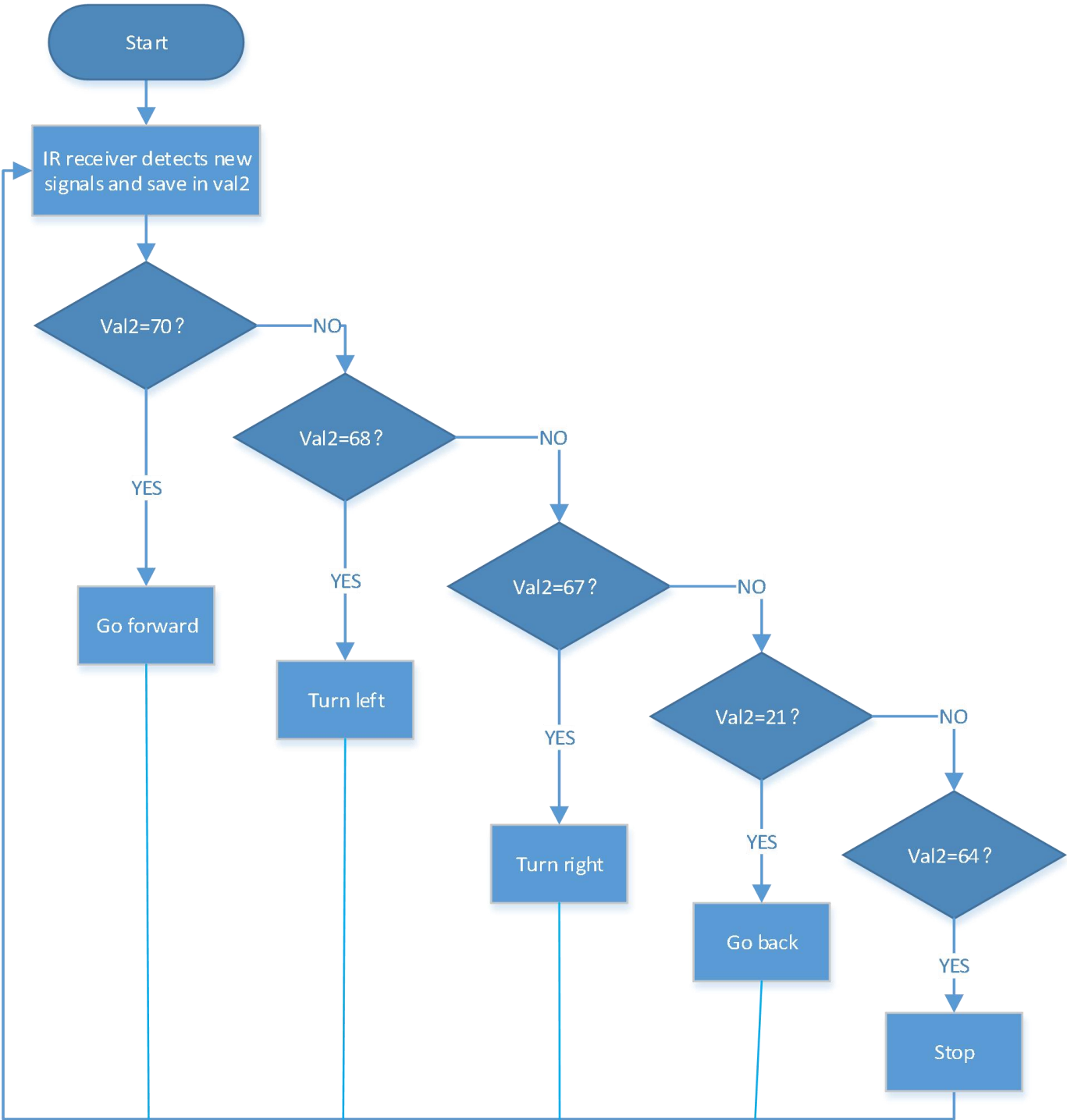
- Insert the micro:bit board into the slot of keystudio 4WD Mecanum Robot Car V2.0
- Place batteries into battery holder
- Dial power switch to ON end
- Connect the micro:bit to your computer via an USB cable
- Open the Web version of Makecode

Import Hex profile (**How to import?**) , or click “New Project” and drag blocks step by step(add MecanumRobot extension library first)

(**How to add Mecanum_Robot extension?**)

Note: The infrared sensor and infrared remote control should not be used in environments with infrared interference such as sunlight for it contains a lot of invisible lights, such as infrared and ultraviolet. In an environment with strong sunlight, they cannot work normally.

3. Flow Chart



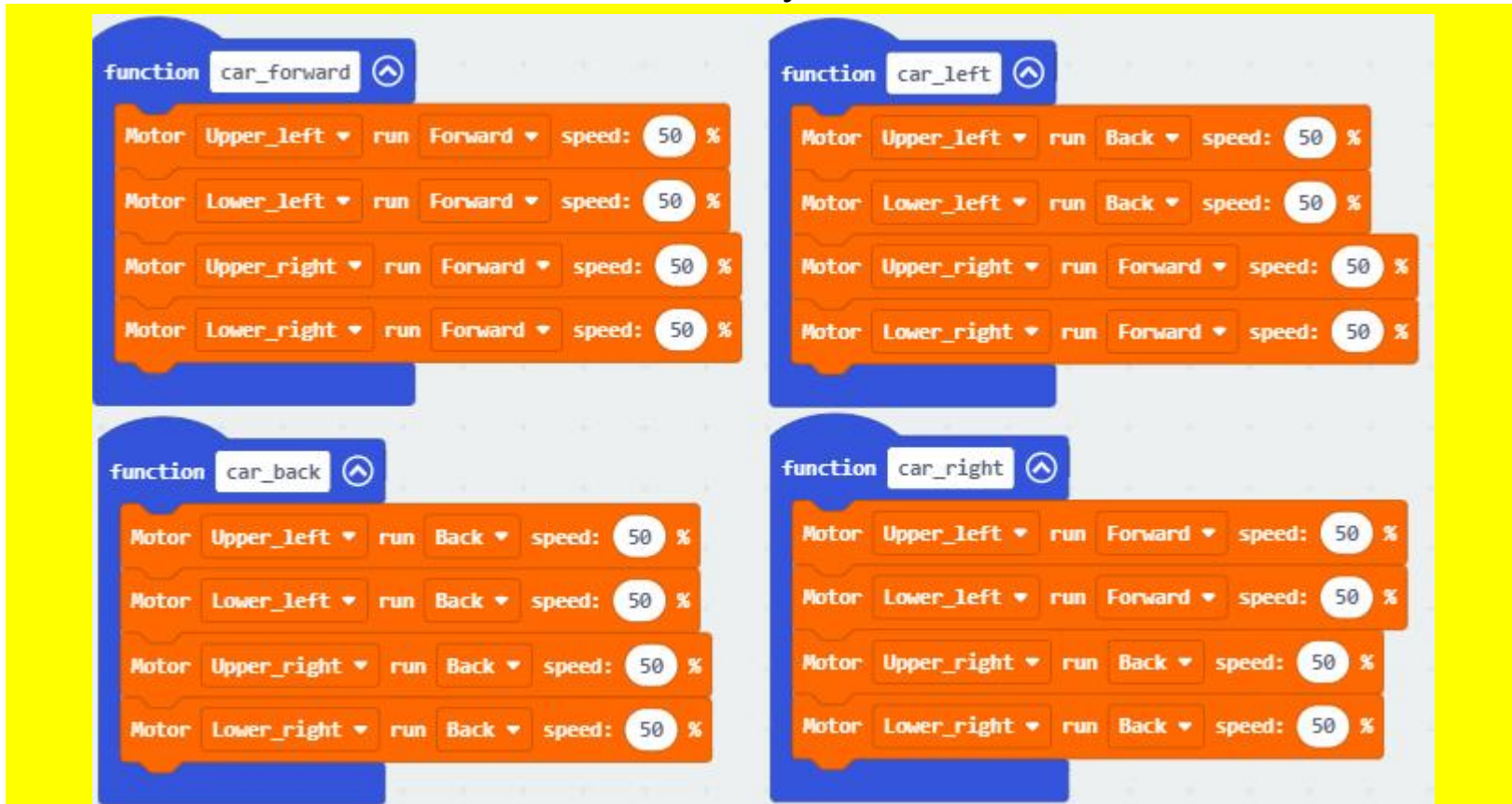
4. Test Code

Code path:

File Type	Path	File Name
Hex file	..\2. Makecode Tutorial\Makecode Code	Project 19.2: IR Remote Control .hex

Or you could edit code step by step in the editing area.

(1)Create four functions to control the car to move forward and back and turn left and right:



(2)Click “Ir Remote” to find and drag “connect IR receiver at P0” into “on start” ; Click the little triangle behind P0 to choose P0;



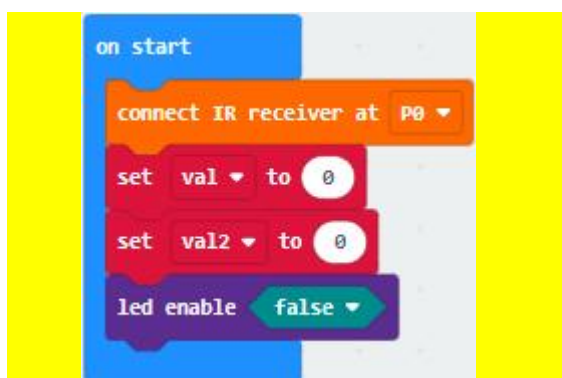
(3)Click “Variables” then click “Make a Variable...” , the dialog box “New variable name: ” pops up; fill it with “val” and click “OK” to create variable “val” ;

Create variable “val2” with the same method; find and drag “set val2 to 0” to “on start” and copy it once to put into “on start” too;

Click the little triangle behind the first val2 to choose “val” ;



(4)Click “Led” → “more” → “led enable(false)” and place it into “on start” .



(5)Click “Variables” , find and drag “set val2 to 0” into “forever” .

Click the little triangle behind the val2 to choose “val” .

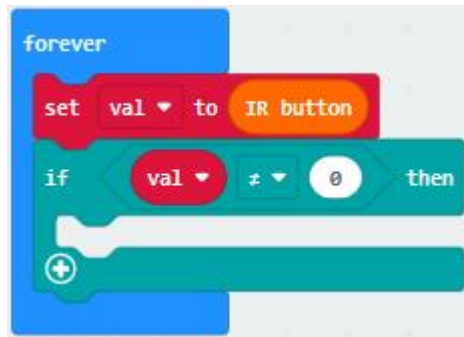
keyestudio

Click "IrRemote" → "IR button" and place it into 0 behind the to.

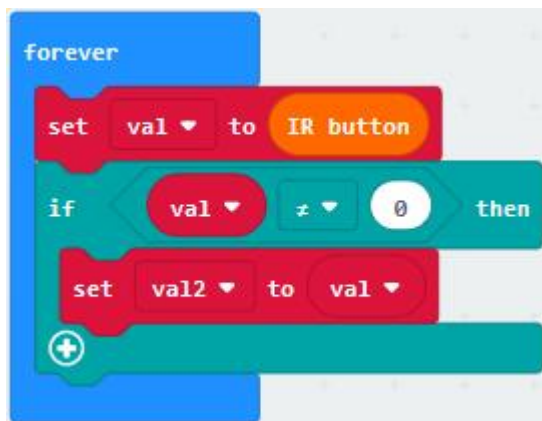


(6)Click "Logic" to find and drag "if true then" into "forever" ; find and drag "=" into "true" ;

Click "Variables" to find and drag "val" to the left side of "="; the 0 on the right side of "=" remain unchanged;
click the little triangle behind "=" to choose "≠" ;



(7)Click "Variables" to find and drag "set val2 to 0" into "then" ;find and drag "val" into the o behind "to" of
"set val2 to 0" ;

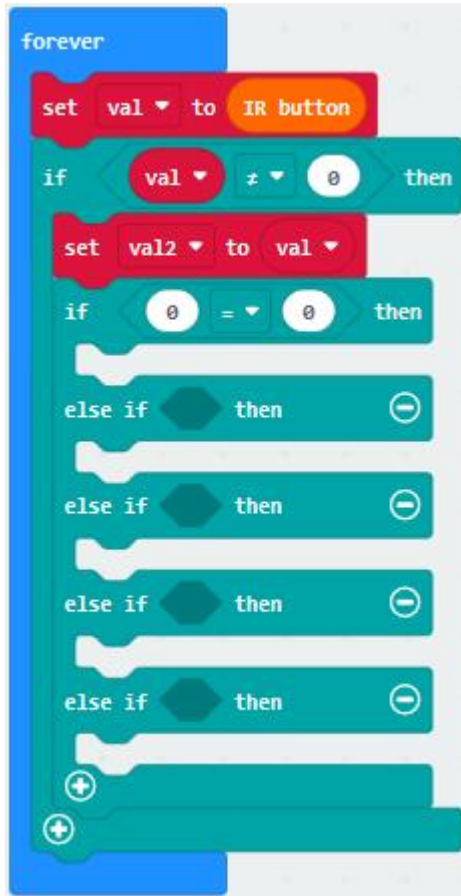


(8)Click "Logic" to find and drag "if...then...else" to then;

Click "+" of "if...then...else" four times;

Click "-" behind "else" once to delete else;

Find and drag "=" to "true" ;



(9)Click "Variables" to find and drag "val2" to the left side of "=" and change the 0 on the right of "=" to 70:



(10)Click "Functions" of "Advance" to find and drag  to the second "then" :



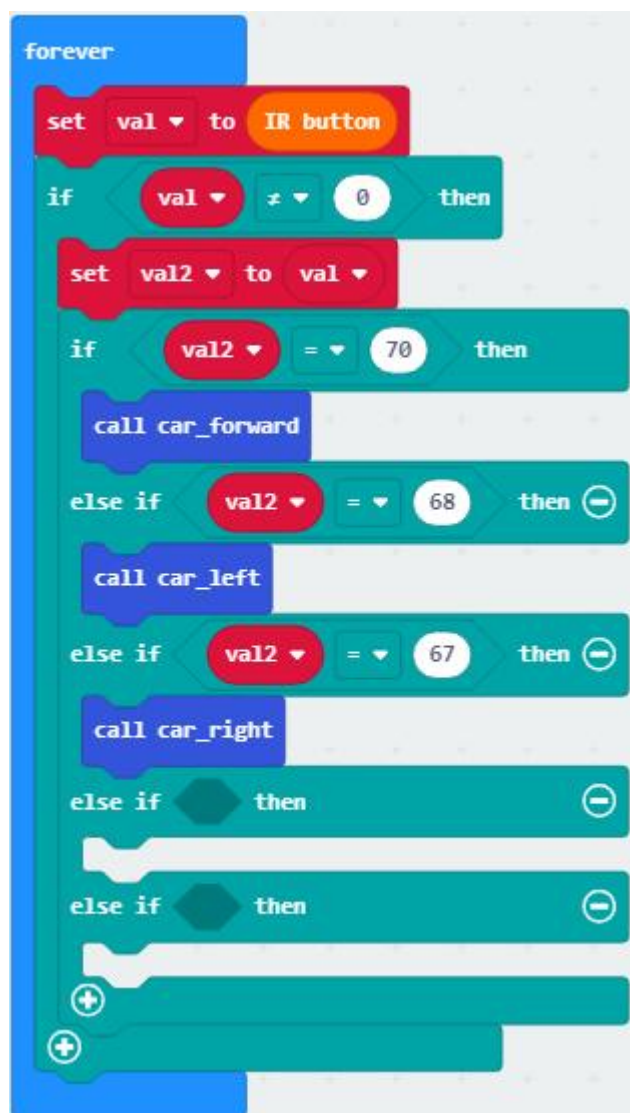
(11)Copy "val2=70" once and place it behind the first "if" ;change the 70 behind "=" to 68.



(12)Click "Functions" of "Advance" to find and drag  to the second "then" .



(13) Copy "val2=68" once and place it behind the second "else if "; change the 68 behind "=" to 67; place it in the forth "then" ; Click "Functions" of "Advance" to find and drag  to the forth "then" :



(14) Copy "val2=67" once and put it behind "=" of the third "else if ; change the number 67 to 21; click "Functions" of "Advance" to find and drag  to the fifth "then" :



(15) Copy "val2=21" once and place it behind the fourth "else if "; change the the number 21 behind "=" to 64;



Click "MecanumRobotV2" to find and drag to the sixth "then" :



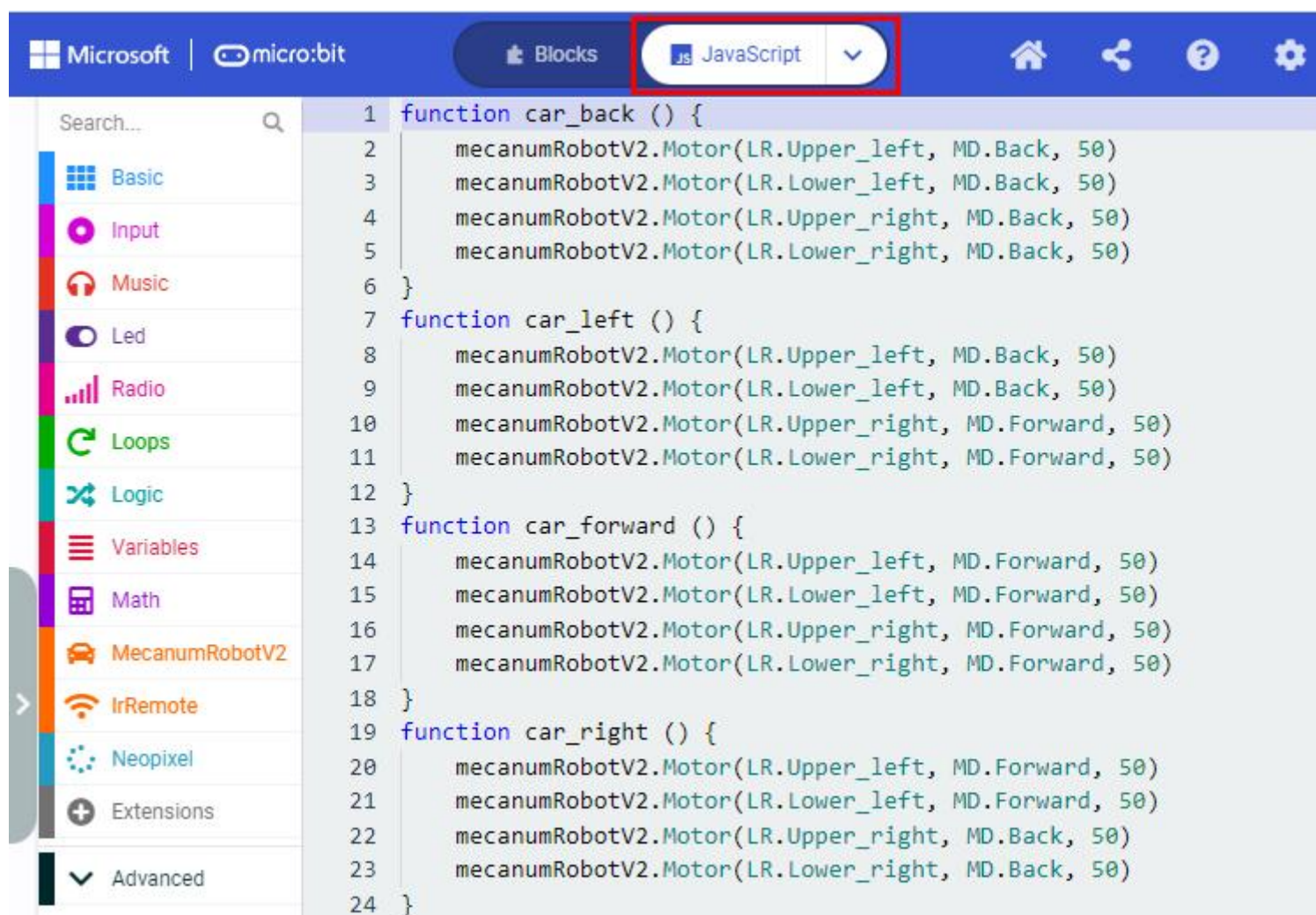
Complete Program:

keystudio



- ① The "on start" command block runs only once to start the program.
- ② Connect the IR receiver to P0
- ③ Set the variable val to 0
- ④ Set the variable val2 to 0
- ⑤ In the "forever" instruction box, the program runs cyclically
- ⑥ Set val to IR button
- ⑦ When the variable val $\neq$ 0 is established, execute the program under then
- ⑧ Set variable val2 to val
- ⑨ When val2=70 is established, execute the program under then
- ⑩ The car goes forward
- ⑪ When val2=68 is established, execute the program under then
- ⑫ Turn left
- ⑬ When val2=67 is established, execute the program under then
- ⑭ Car turn right
- ⑮ When val2=21 is established, execute the program under then
- ⑯ The car goes back
- ⑰ When val2=64 is established, execute the program under then
- ⑱ The car stops

Click "JavaScript" to switch into the corresponding JavaScript code:



```

25 irRemote.connectInfrared(DigitalPin.P0)
26 let val = 0
27 let val2 = 0
28 led.enable(false)
29 basic.forever(function () {
30     val = irRemote.returnIrButton()
31     if (val != 0) {
32         val2 = val
33         if (val2 == 70) {
34             car_forward()
35         } else if (val2 == 68) {
36             car_left()
37         } else if (val2 == 67) {
38             car_right()
39         } else if (val2 == 21) {
40             car_back()
41         } else if (val2 == 64) {
42             mecanumRobotV2.state()
43         }
44     }
45 })
46

```

5. Test Result

Download code to micro:bit board, and dial POWER to ON end.

Make IR remote control point at micro:bit and press the button to control smart car to move.



button makes smart car move forward , stands for turning left , implies rightward turning,



indicates moving backward, stops car.

(How to download? How to quick download?)

Note: The distance between IR remote control and IR receiving head of smart car are supposed less than 5m during the test.

Project 20: Bluetooth Multi-purpose Smart Car

Project 20.1: Read Bluetooth Data



1. Description

Micro:bit main board comes with a built-in Bluetooth which can be used to communicate with it. And the Micro:bit can also be controlled by Bluetooth or transmit signals back to smartphone or computer via it. This Bluetooth can communicate with the Bluetooth equipped in other devices or with Bluetooth App to control other equipment.

It is compatible with both Android system and IOS system. And we have designed two Bluetooth Apps for both systems.

The connection of the Bluetooth on the board with these two Apps is similar. In this lesson, we will introduce the functions of all keys and patterns on the Apps and control the smart car via Bluetooth App.

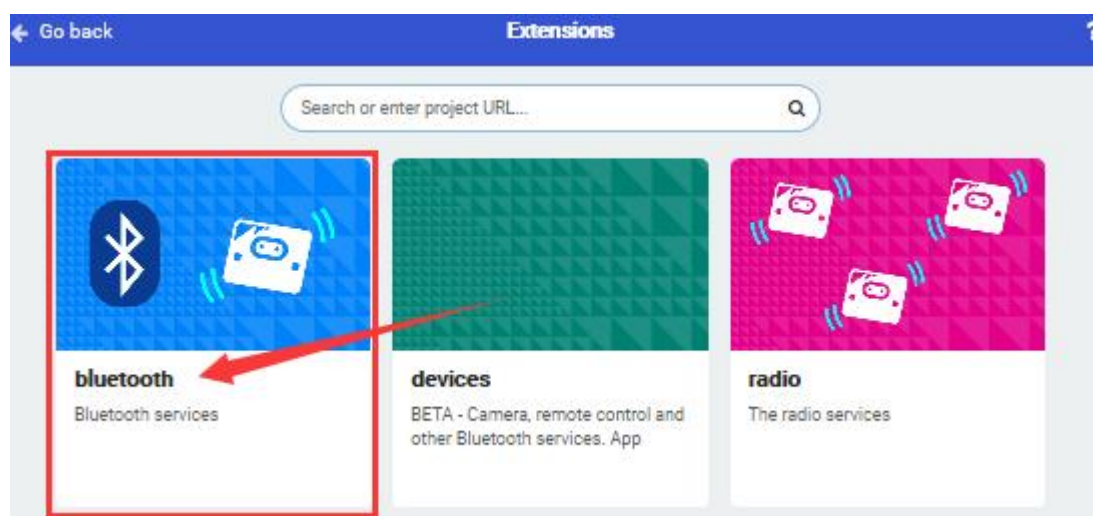
2. Preparation

- Insert the micro:bit board into the slot of keyestudio 4WD Mecanum Robot Car V2.0
- Place batteries into battery holder
- Dial power switch to ON end
- Connect the micro:bit to your computer via an USB cable
- Open the Web version of Makecode

Import Hex profile (**How to import?**) , or click "New Project" and drag blocks step by step(add MecanumRobot extension library first)

(How to add Mecanum_Robot extension?)

If you choose to drag the code manually, you need to add the Bluetooth extension library first. Click the gear icon (Settings) in the upper right corner, then click on Extensions to go to the library file selection screen, and then click on the "Bluetooth" extension library (if it doesn't exist, search Bluetooth to find it), as shown below:



As the Bluetooth and extension radio can't work together, therefore, their extension libraries are not compatible. Therefore, remove extension(s) and add Bluetooth please if you see the following prompt box pop up.



3. Test Code

Code Path:

File Type	Path	File Name
Hex file	..\2. Makecode Tutorial\Makecode Code	Project 20.1: Read Bluetooth Data.hex

Or you could edit code step by step in the editing area.

Enter "Advanced" → "Serial" → "serial redirect to USB"

Place it into "on start"



Click "Bluetooth" → "on bluetooth connected"

Go to "Basic" to move "show icon" block into "on bluetooth connected" block.



Click "Variables" → "Make a Variable..." → "New variable name: " dialog box.

Input "connected" and click "OK" to create variable "connected" .

Drag "set connected to 0" under block "show icon" and change 0 into 1.

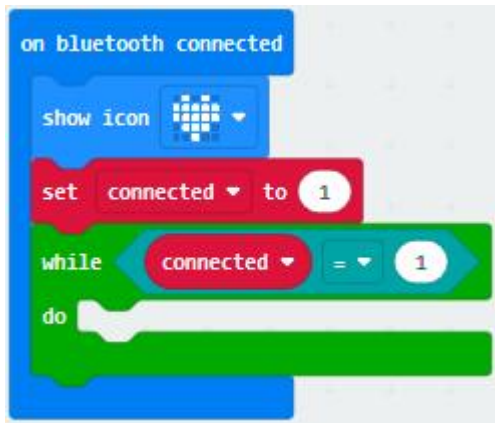


keyestudio

Go to "Loops" to move block "while true do..." into "on bluetooth connected" block.

Enter "Logic" to drag out "=" block.

Click "Variables" to drag "connected" into left box of "=" block and change 0 into 1.

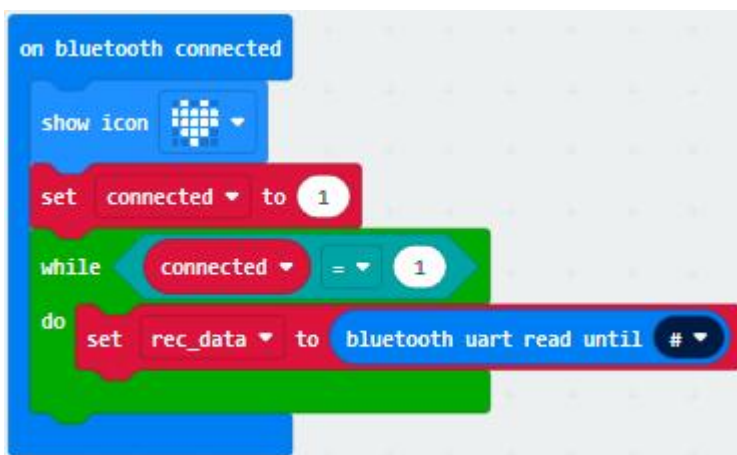


Then we generate variable "rec_data" in same way.

Then drag out "set rec_data to 0" and place it into block "while connected=1 do..." block.

Click "Bluetooth" → "more" → "bluetooth uart read until new line()"

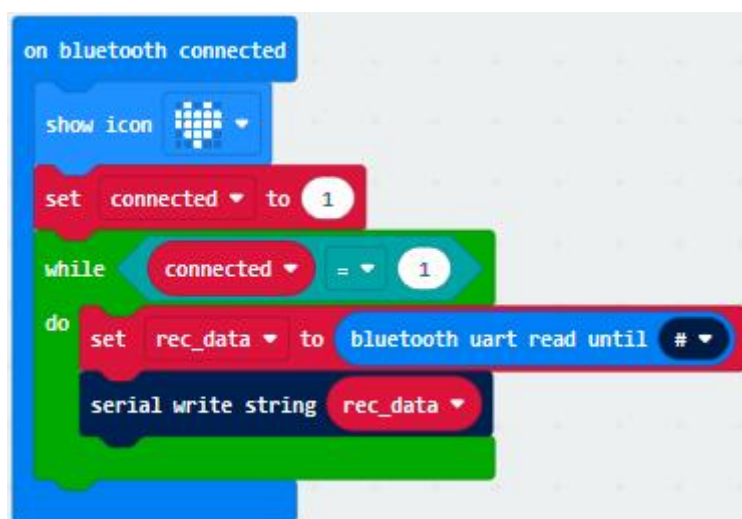
Keep it into 0 box and click triangle button to select #.



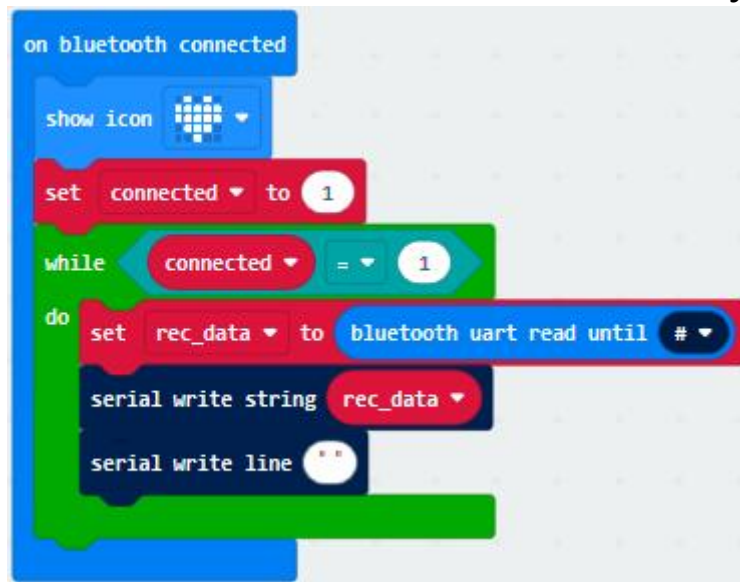
Go to "Advanced" → "Serial" → "serial write string"

Move it below "set rec_data...until#" block

And combine variable "rec_data" with "serial write string" block.



Click "Advanced" → "Serial" → "serial write line" and place it into "while connected=1 do..." .



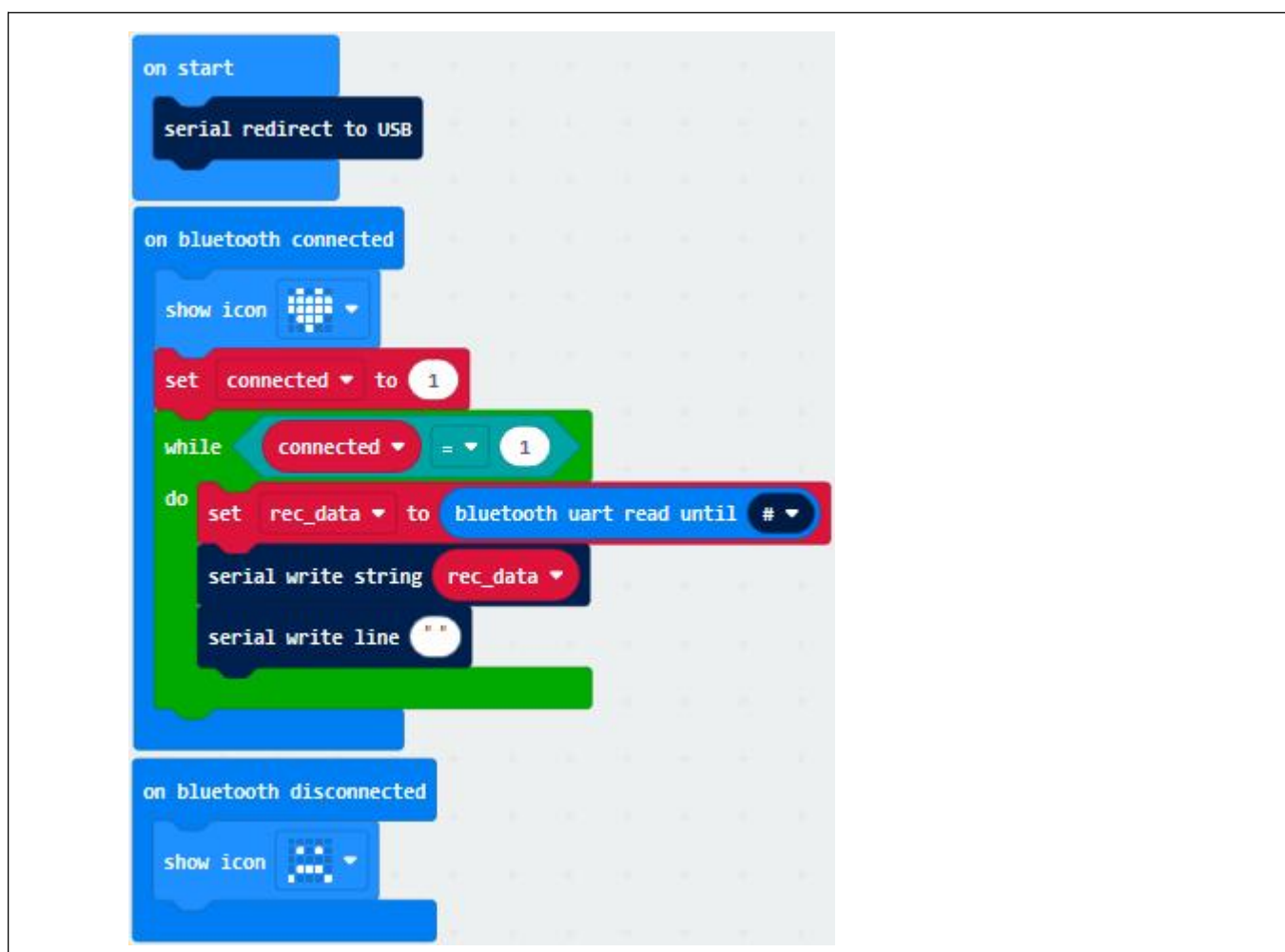
Click "Bluetooth" to drag out "on bluetooth disconnected" .

Click "Basic" , copy "show icon" block and keep it into block "on bluetooth disconnected"

Click triangle button to select "4x4" pattern.



Complete Program



keyestudio

“on start” : command block runs once to start program.

Serial redirect to USB

Connect Bluetooth

LED dot matrix shows “♥” pattern

Set variable connected to 1

When connected=1, the code under do block will be executed.

Set rec_data to bluetooth uart read until #

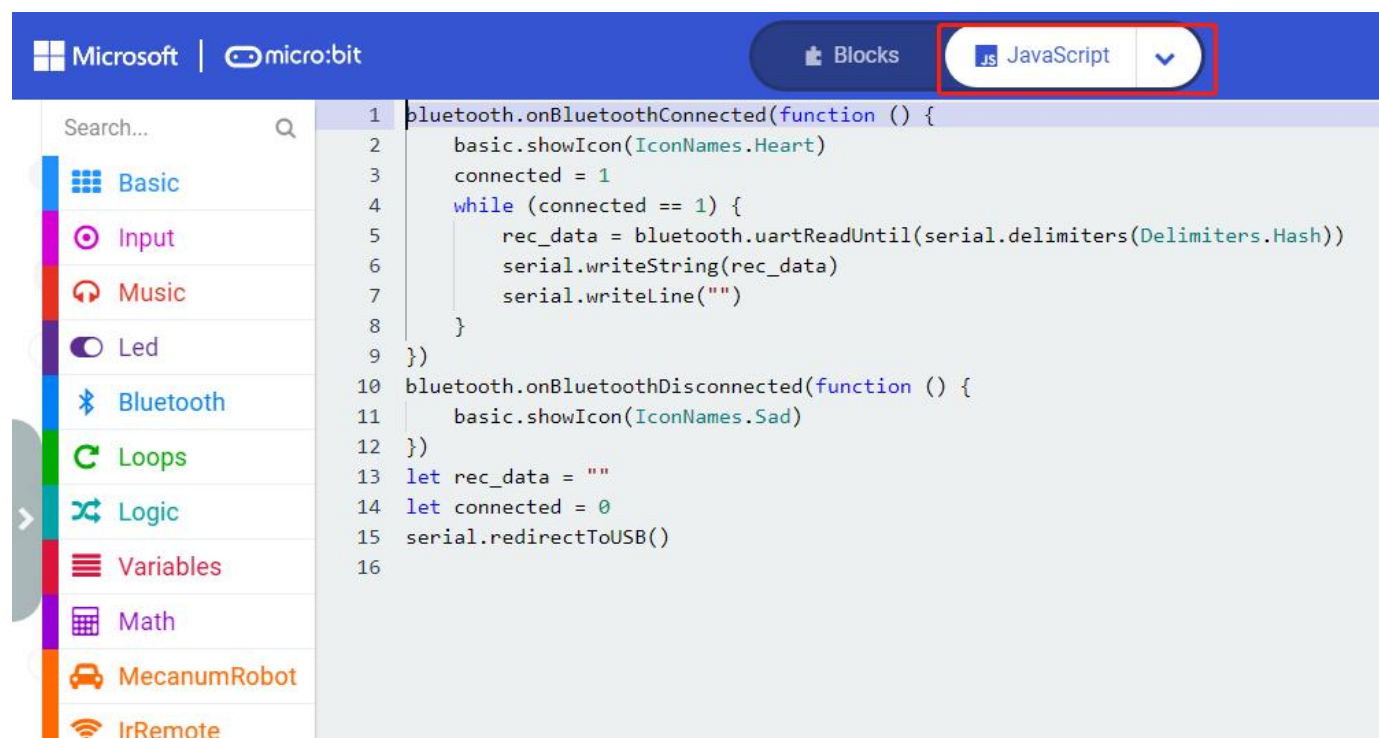
Serial port prints rec_data

Print a blank space

Disconnect Bluetooth

LED dot matrix displays “📶” pattern.

Click “JavaScript” to view the corresponding JavaScript code:



4.Test Result

If you drag blocks step by step, you need to set as follows after finishing test code.



However, you could skip this step if you directly import test code.

After setting, download code to micro:bit board, don't plug off the USB cable(How to download? How to quick download?)

Next to download App.

For IOS System:

a.Open App Store;



b.Search mecanum_robot and click " " to download the Bluetooth App of mecanum_robot;

c. After downloading the APP, click "OPEN" or click the application mecanum_robot on the phone/iPad desktop to open the APP. A dialog box appears on the APP interface, and click "OK" in the dialog box.

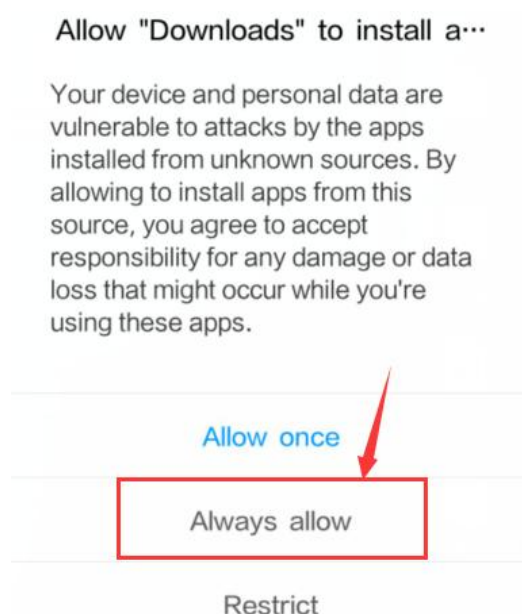
d. First turn on the Bluetooth of the mobile phone/iPad, and then click the connect button (control) in the upper left corner of the APP interface to perform a Bluetooth search. In the search results, click "BCC micro:bit". After a few seconds, the Bluetooth is connected.

For Android System:

a. Use the scanning function in the browser to scan and identify the QR code or enter the http://8.210.52.206/mecanum_robot.apk to download. After the identification is successful, click "go to website" to enter the download mecanum_robot.apk page , click "Download" to download the mecanum_robot application.

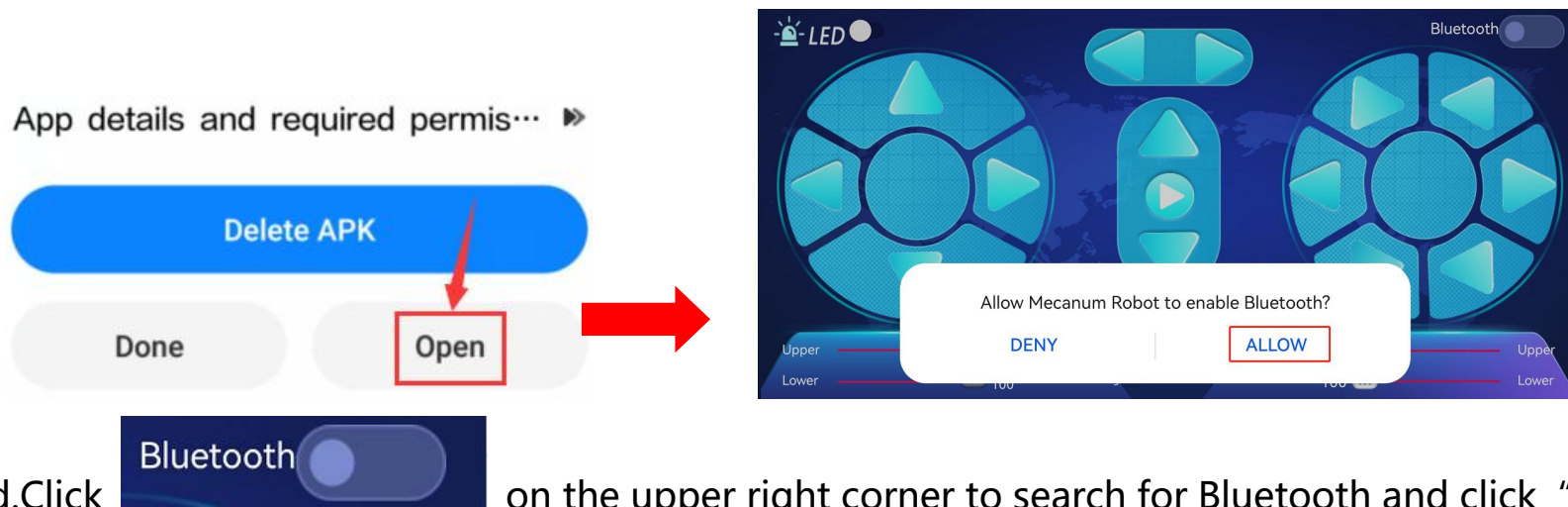


b.Click "Allow allow" to enter Installation Diagram; click "install" to install the App.

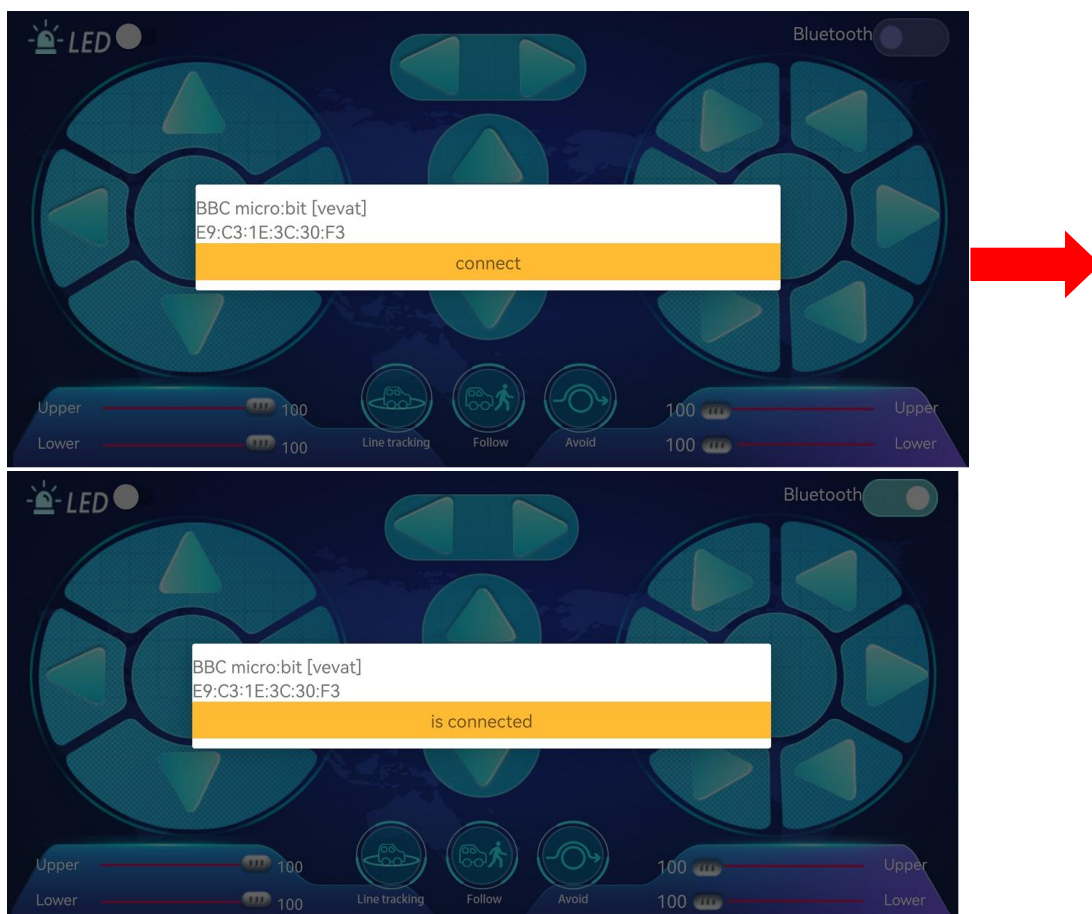


keyestudio

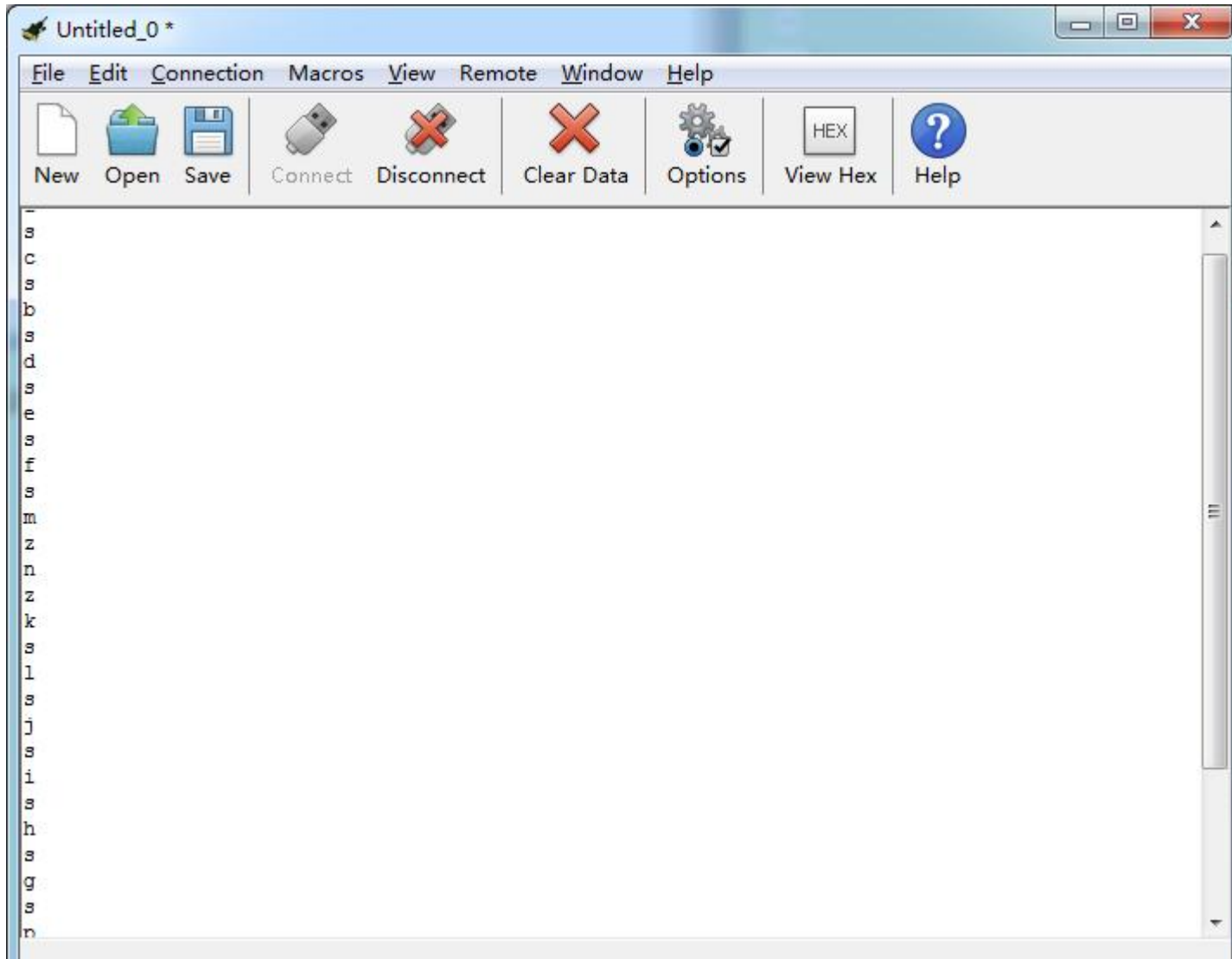
c. Click "Open" or click the application mecanum_robot on the mobile phone desktop to open the APP, and a dialog box appears. In the dialog box, click "Allow" to turn on the Bluetooth of the mobile phone. You can also turn on the phone's Bluetooth before opening the APP.



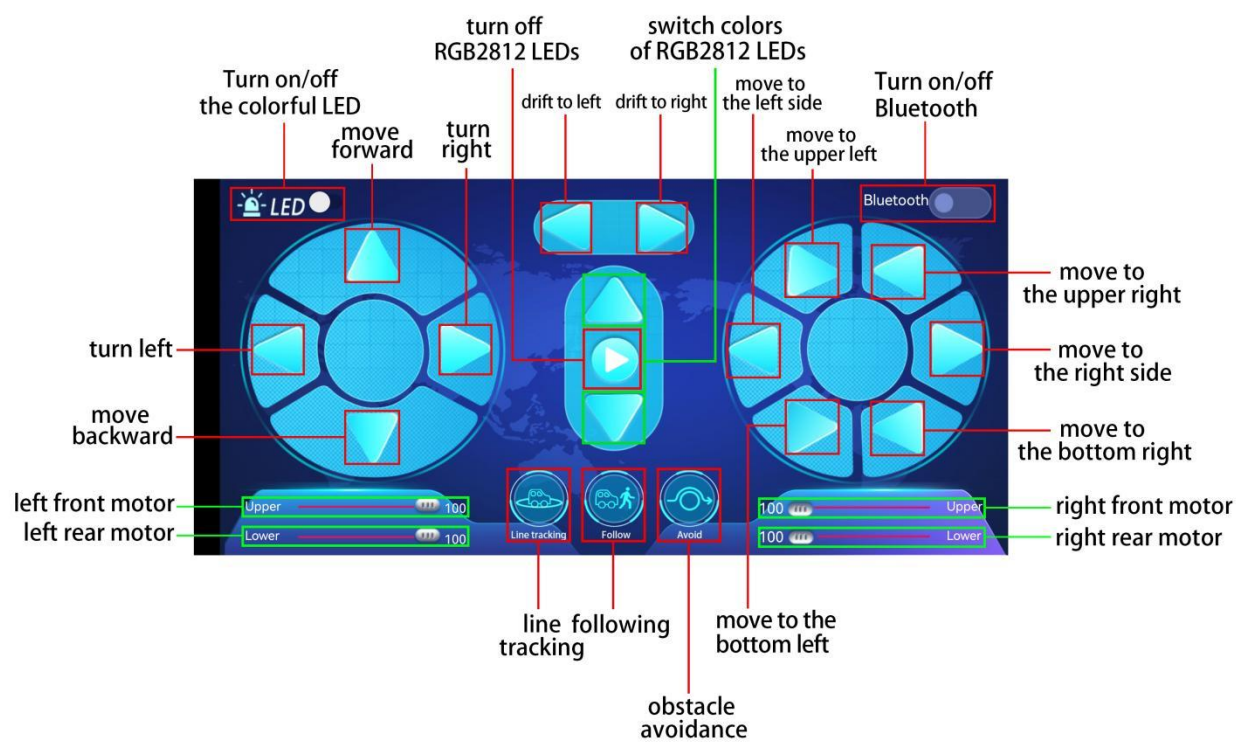
d. Click  on the upper right corner to search for Bluetooth and click "connect" ; a few seconds later, the Bluetooth is paired.



Open CoolTerm, click Options to select SerialPort. Set COM port and 115200 baud rate. Click "OK" and "Connect". Point at micro:bit board and press the icons on APP, the corresponding characters are shown on CoolTerm monitor.



Through the test, we get the functions of every icon, as shown below:



Project 20.2: Multi-purpose Smart Car



1. Description

In this lesson, we will control the smart car to perform multipurpose functions.

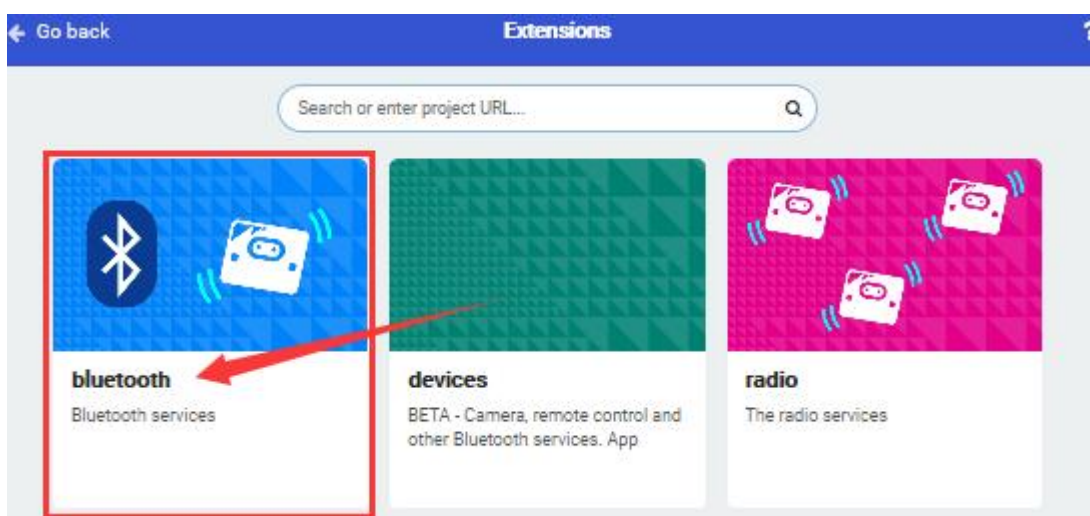
2. Preparation

- Insert the micro:bit board into the slot of keyestudio 4WD Mecanum Robot Car V2.0
- Place batteries into battery holder
- Dial power switch to ON end
- Connect the micro:bit to your computer via an USB cable
- Open the Web version of Makecode

Import Hex profile (**How to import?**) , or click "New Project" and drag blocks step by step(add MecanumRobot extension library first)

(How to add Mecanum_Robot extension?)

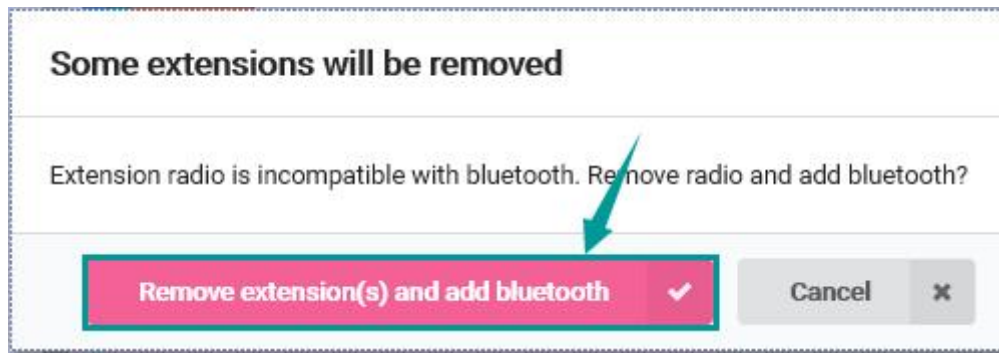
Steps: Click the gear icon (Settings) in the upper right corner, then click on Extensions to go to the library file selection screen, and then click on the "Bluetooth" extension library (if it doesn't exist, search Bluetooth to find it), as shown below:



As the Bluetooth and extension radio can't work together, therefore, their extension libraries are not compatible.

keyestudio

Therefore, remove extension(s) and add Bluetooth please if you see the following prompt box pop up.

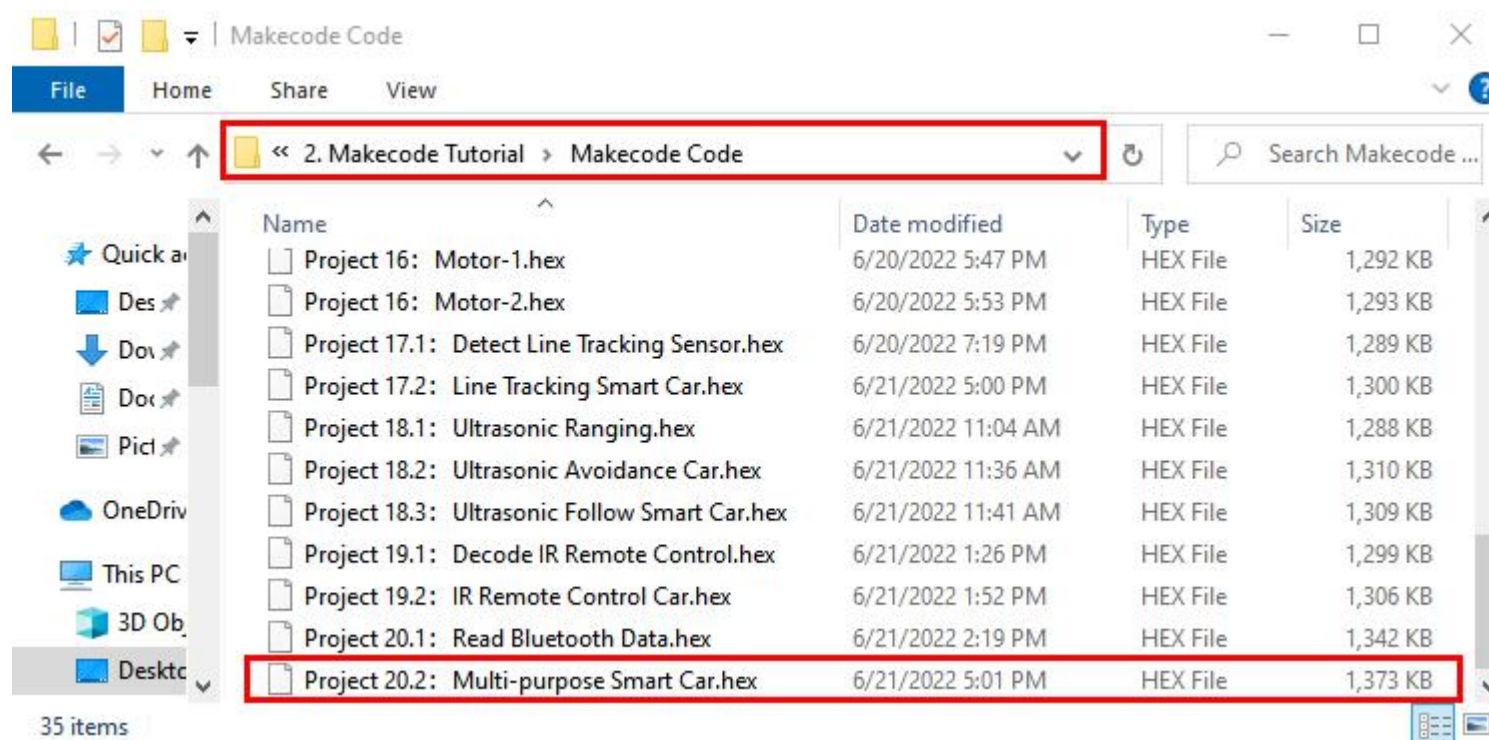


3. Test Code

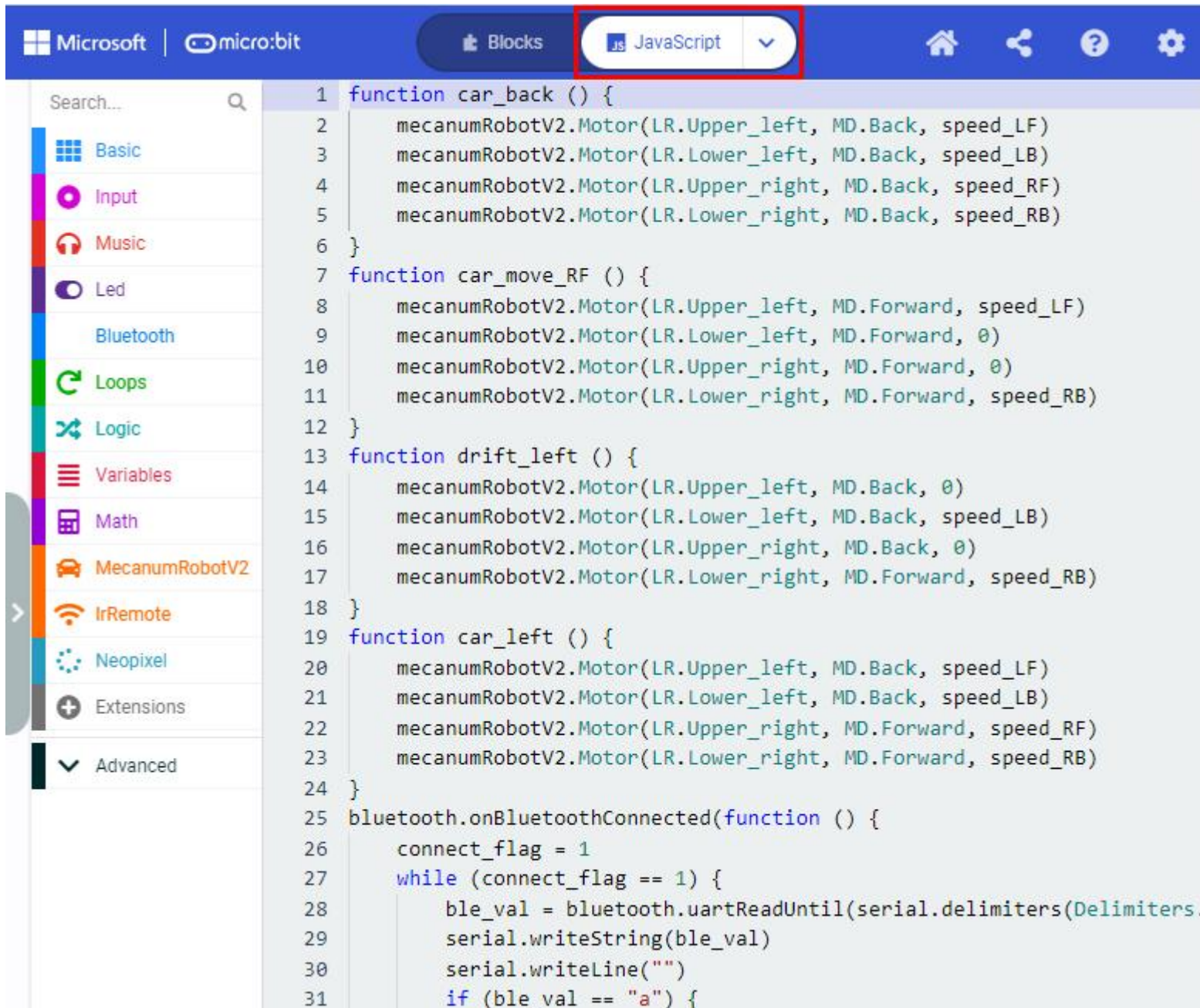
Code path:

File Type	Path	File Name
Hex file	..\2. Makecode Tutorial\Makecode Code	Project 20.2: Multi-purpose Smart Car.hex

Complete Code:

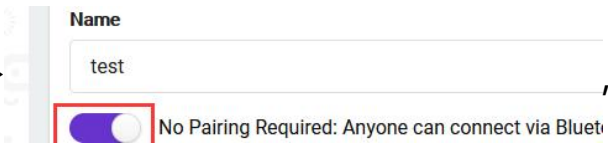


Click "JavaScript" to view the corresponding JavaScript code: :



4. Test Result

This experiment combines the previous projects to make the car to perform actions via Bluetooth.

Enter Makecode online editor→Projecting Settings→, enable “No Pairing....” (you could skip this step if you import test code directly)

Download code to micro:bit board, dial POWER to ON end, and connect the Bluetooth, then you can control the car via the Bluetooth App of mecanum_robot.

(How to download? How to quick download?)

Note:  is used to adjust the speed, and  can only be dragged.

9. Common Problems

1. The car has no reaction

Please check whether the batteries are sufficient

Please check whether the wirings are correct

2. Computers can't recognize the USB ports

Please ensure whether the microbit driver is installed

Please check whether the USB wire is in good condition.

3. Code fails to burn and dot matrix displays expressions

Please check whether the Mecanum Robot Car_V2.py library file is imported

10. Resources

<https://fs.keyestudio.com/KS4034-4035>