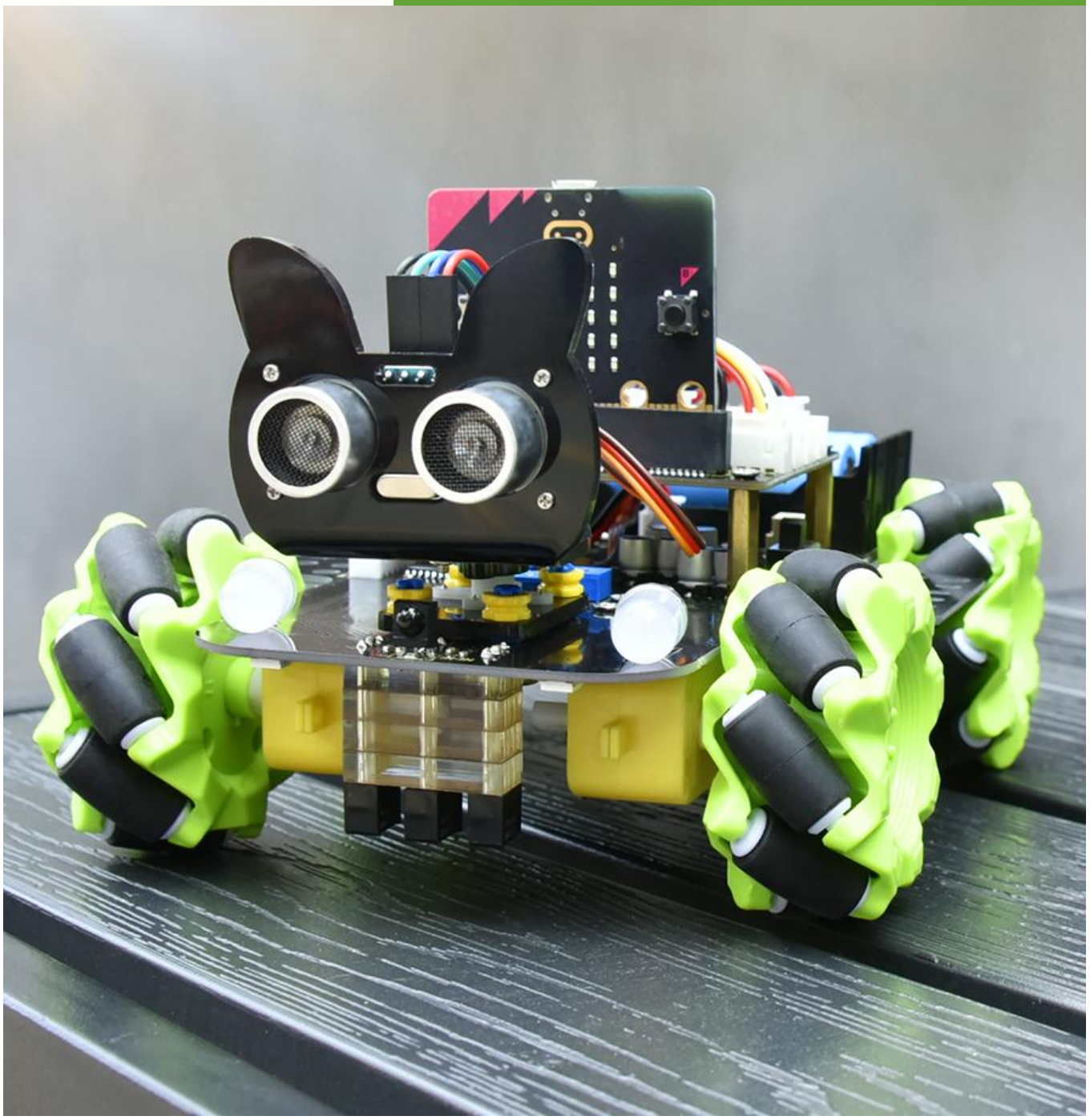
KEYESTUDIO

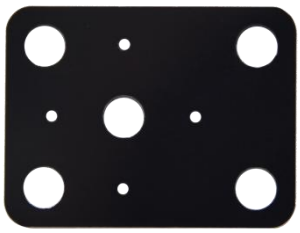
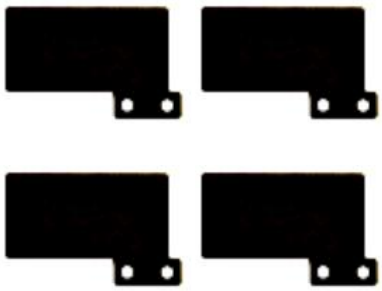
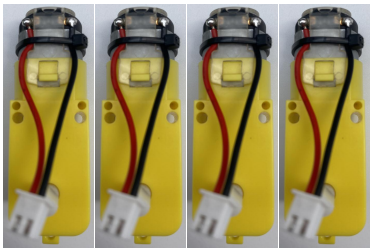
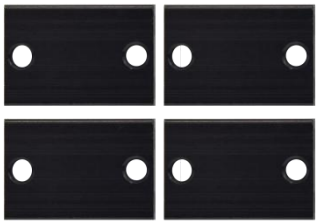
4WD Mecanum Robot Car

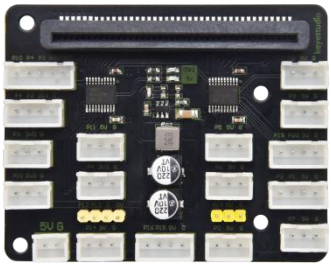
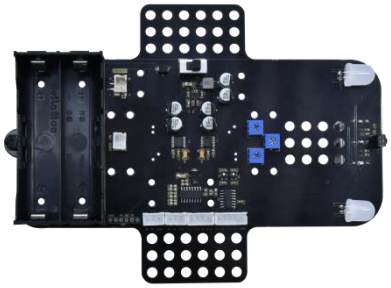
For BBC Microbit V2

Makecode Tutorial

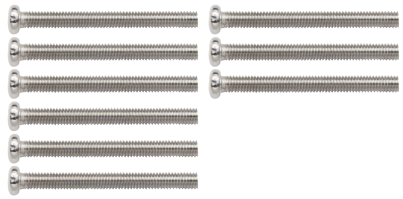


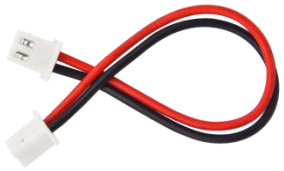
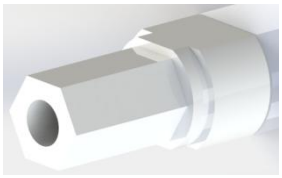
1.What's in the package?

#	Picture	Description	QTY
1		Acrylic Board for Ultrasonic Module	1
2		Acrylic Board for Servo	1
3		Acrylic Board for Motor	4
4		Motor	4
5		23*15*5MM Fixing Board	4

6		180° Servo with Arms and Screws	1
7		58mm Mecanum Wheel 58mm Direction A	2
8		58mm Mecanum Wheel 58mm Direction B	2
9		Micro： bit Shield for Car	1
10		Micro:bit V2.0 Board (Only SKUK S4034 and KS4034F come with a Micro:bit)	1
11		Mecanum Wheel Car Base	1

12		M3*20MM Dual-pass Copper Pillar	4
13		Building Block 4265c	4
14		Building Block 43093	4
15		Acrylic Gasket T=5mm 6 PCS	1
16		M3*6MM Flat Head Screw	10
17		SR04 Ultrasonic Sensor	1
18		M3*8MM Flat Head Screw	10
19		M3 Nut	10

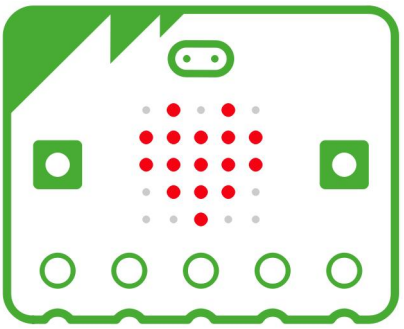
20		M3*30MM Round Head Screw	9
21		M2 Nut	3
22		M2*8MM Round Head Screw	3
23		M1.4 Nut	6
24		M1.4*10MM Round Head Screw	6
25		M2.3*16MM Round Head Screw	4
26		Remote Control(Only the remote controls of SKU KS4034F and KS4035F come with a battery)	1
27		Nylon Cable Tie 3*100MM	5
28		USB Cable	1

29		HX-2.54 2P Cable 100mm	1
30		XH2.54 5P Cable 100mm	1
31		HX-2.54 4P Cable 50mm	1
32		HX2.54mm-4P to 2.54 Dupont 150MM	1
33		XH2.54 3P Cable 50mm	2
34		3*40MM Screwdriver	1
35		TT Coupling	4
36		M1.2*5MM Round Head Self-tapping Screw	6

2.Getting started with micro:bit

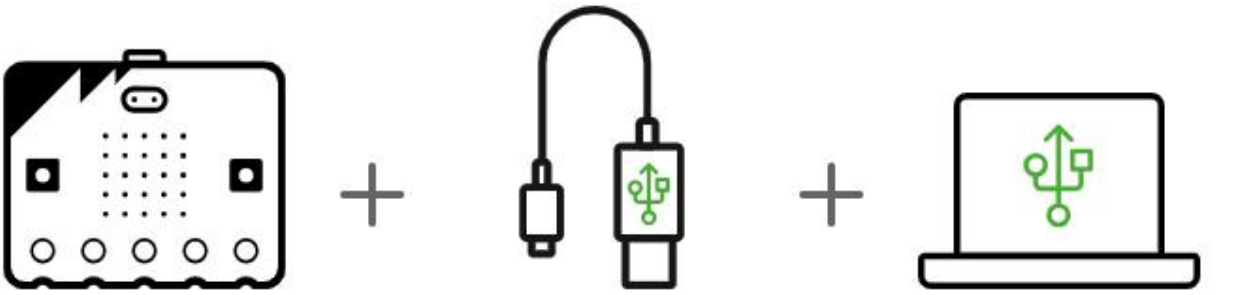
2.1 Introduction

The micro:bit is a pocket sized computer that has sensors and outputs built in. You can use it to create physical computing projects that interact with the real world, from robots to musical instruments and more.



What you will need to prepare now?

Hardware

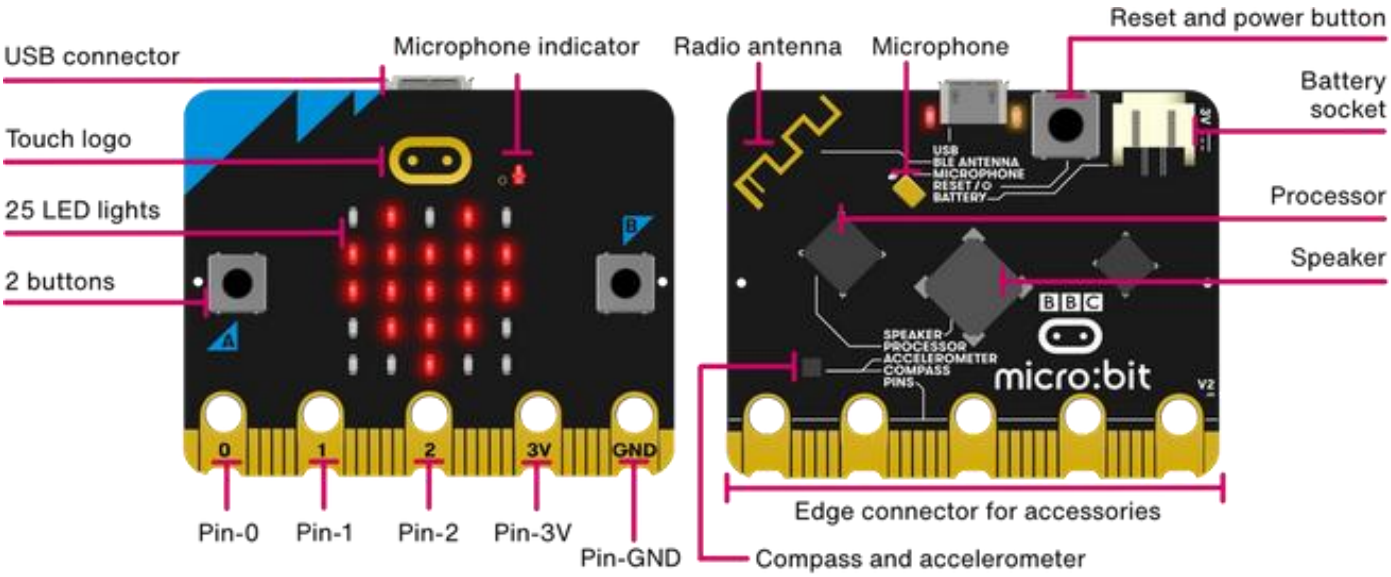


Software

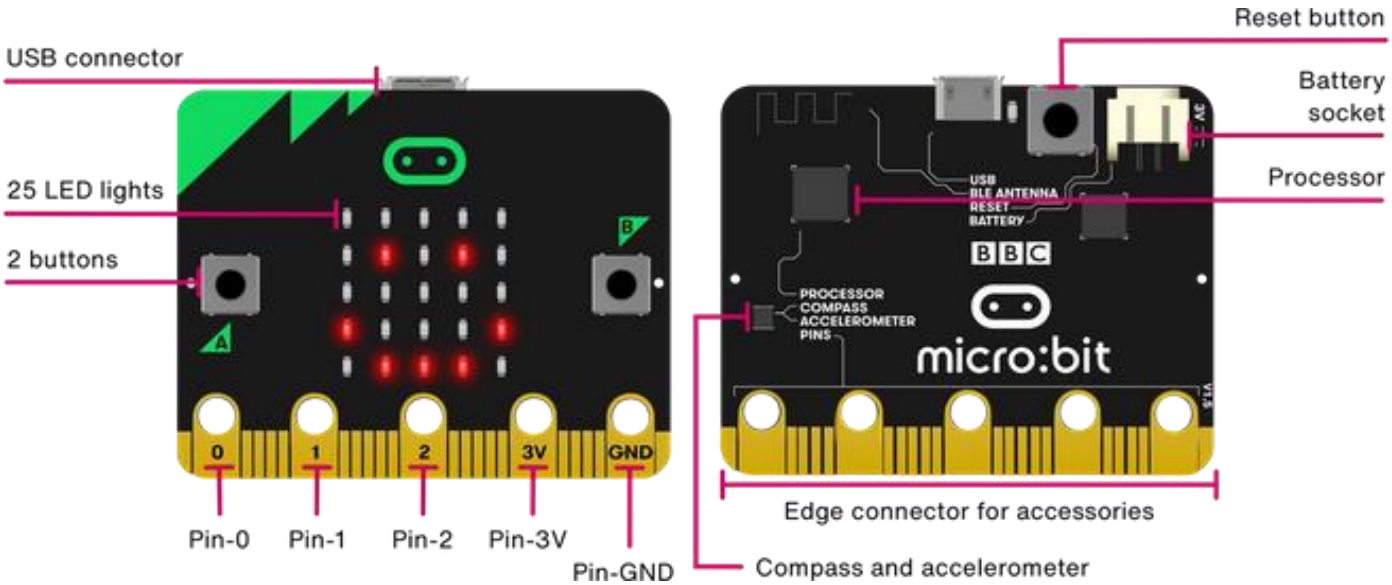
A browser to run the MakeCode editor
https://makecode.microbit.org/

2.2 Meet The Micro:bit

New micro:bit (V2)



Original micro:bit (V1)



The following content is borrowed from Kitronik University micro:bit resources and guides.

The table below lists all of the features for both boards for much easier comparison, new or improved features are highlighted in green.

Original (v1.5)	Feature	Latest (v2)
25 Programmable LEDs in a 5 x 5 grid	LED Matrix	25 Programmable LEDs in a 5 x 5 grid
3-axis motion sensing	Accelerometer	3-axis motion sensing
On-board magnetometer	Compass	On-board magnetometer
On-board temperature sensor	Temperature sensing	On-board temperature sensor
On-board light level sensing	Light level sensing	On-board light level sensing
2 x programmable buttons, A & B	User Buttons	2 x programmable buttons, A & B
BLE Bluetooth 4.0	Bluetooth	BLE Bluetooth 5.0
2.4Ghz Micro:bit Radio	Radio	2.4Ghz Micro:bit Radio
Rear-mounted push button	Reset Button	Rear-mounted push button
N/A	On/Off switch	Power off (push and hold power button)
25 pins	Edge Connector	25 pins
N/A	Microphone	MEMS microphone and LED indicator
N/A	Speaker	Onboard speaker
N/A	Logo touch	Touch-sensitive logo pin
N/A	Power indicator	LED power indicator
C++, MakeCode, Python, Scratch	Software	C++, MakeCode, Python, Scratch

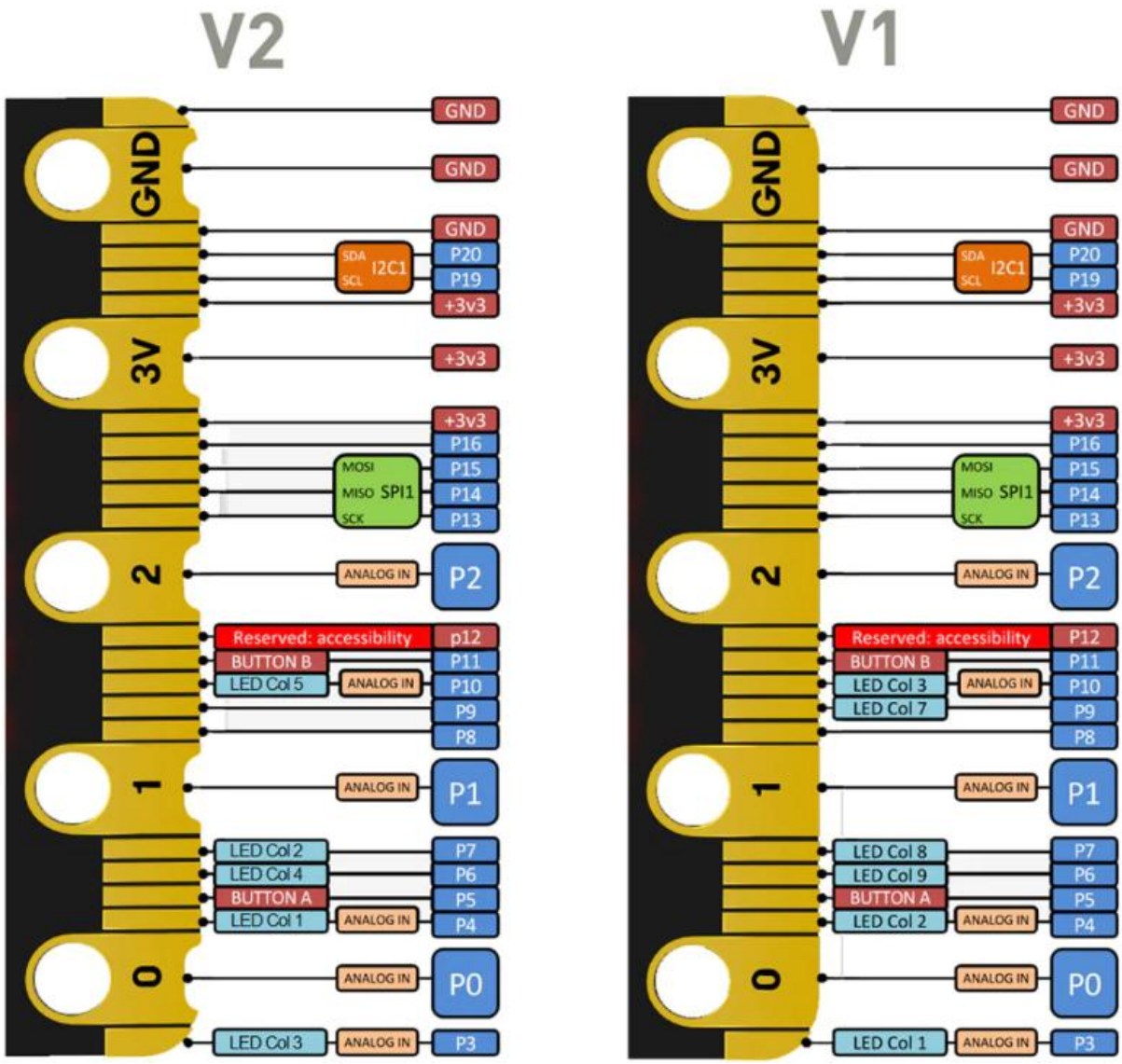
The V2 board also features a number of usability enhancements, they are;

- **Notched edge connector.** To make it easier to connect things like crocodile clips and also conductive thread.
- **Power LED indicator.** In addition to the USB activity indicator.
- **Gold plated antenna.** To easily identify the radio and Bluetooth component.

The new micro:bit speaker works the same as it does when you currently connect a speaker to the micro:bit and the sound output will be on both the speaker and the edge connector. The new microphone blocks in the MakeCode editor will allow you to write code that reacts to sound.

The Tech' Spec's of micro:bit V1 and microbit V2:

Much of the onboard tech has also been enhanced. The processor and memory has been upgraded which has also allowed upgrades to other on-board technologies.



The table below lists key technical specifications for both boards for much easier comparison, new or improved features are highlighted in green.

Current (v1.5)	Feature	Latest (v2).
Nordic Semiconductor nRF51822	Processor	Nordic Semiconductor nRF52833
256kB Flash 16kB RAM	Memory	512kB Flash, 128kB RAM
NXP KL26Z, 16kB RAM	Interface	NXP KL27Z, 32kB RAM
25 pins. 3 dedicated GPIO, PWM, i2c, SPI and ext. power. 3 ring pins for connecting crocodile clips/banana plugs.	Edge Connector	25 pins. 4 dedicated GPIO, PWM, i2c, SPI and ext. power. 3 ring pins for connecting crocodile clips/banana plugs. Notched for easier connection
Shared I2C Bus	I2C	Dedicated I2C bus for peripherals
2.4Ghz Micro:bit Radio/BLE Bluetooth 4.0	Wireless	2.4Ghz Micro:bit Radio/BLE Bluetooth 5.0
5V via Micro USB port, 3V via edge connector or battery pack.	Power	5V via Micro USB port, 3V via edge connector or battery pack, LED power indicator, Power off (push and hold power button)
90mA available for accessories	Current available	200mA available for accessories
ST LSM 303	Motion sensor	ST LSM 303
5cm(w) x 4cm(h)	Size	5cm(w) x 4cm(h)

Coding and MakeCode Compatibility:

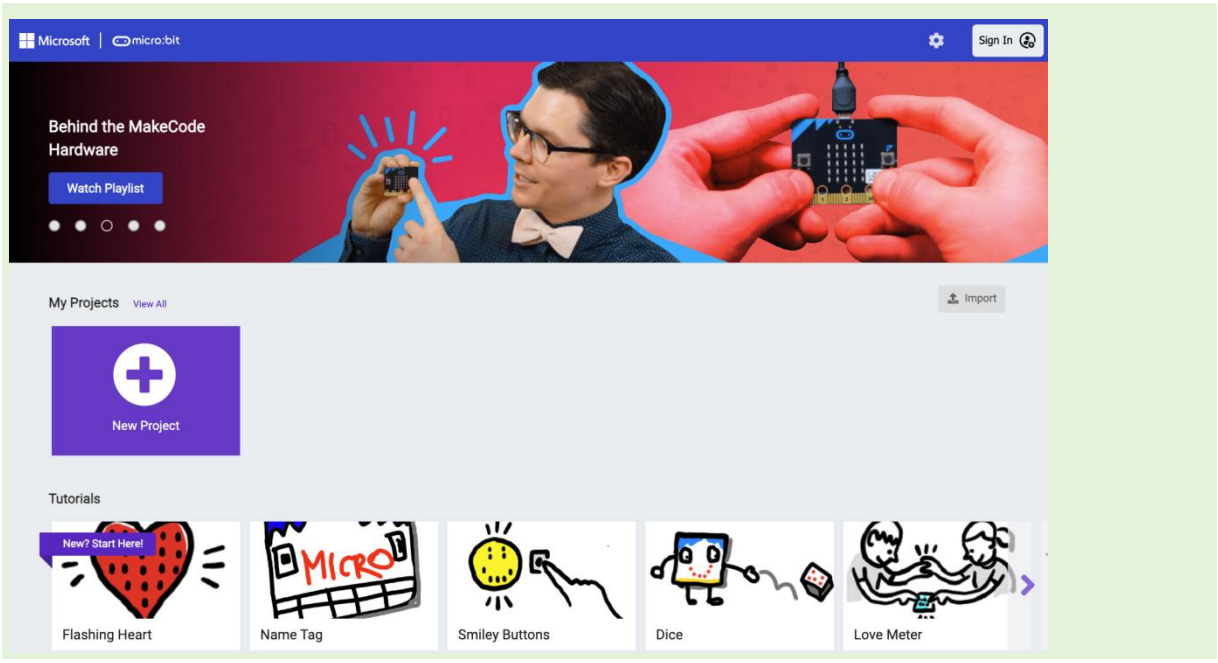
- Users will not need to select which version of the device they have before using MakeCode or the Python Editor.
- Every program that could run on a micro:bit version 1 can be re-built to run on microbit V2.
- The editors will support both versions simultaneously for features common to both boards.
- To convert an old HEX file into a universal HEX file, drag it into the editor it was created in and then export it back to your computer where it can be dragged and dropped onto the micro:bit. If you attempt to use an old .Hex file without updating it, the micro:bit will display a compatibility error.

2.3 Create Your First Project in the MakeCode Editor

In this step you will take a tour of the MakeCode editor, and create your first program for the micro:bit.

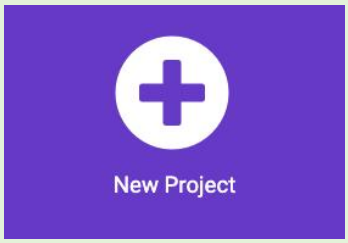
Open the MakeCode editor at https://makecode.microbit.org/

The homepage of the MakeCode

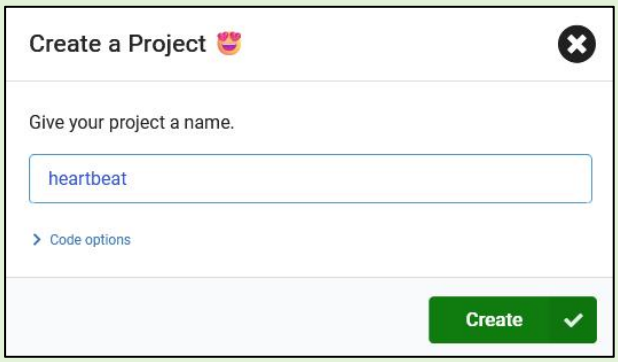


Create a new project

Click on the **New Project** button.



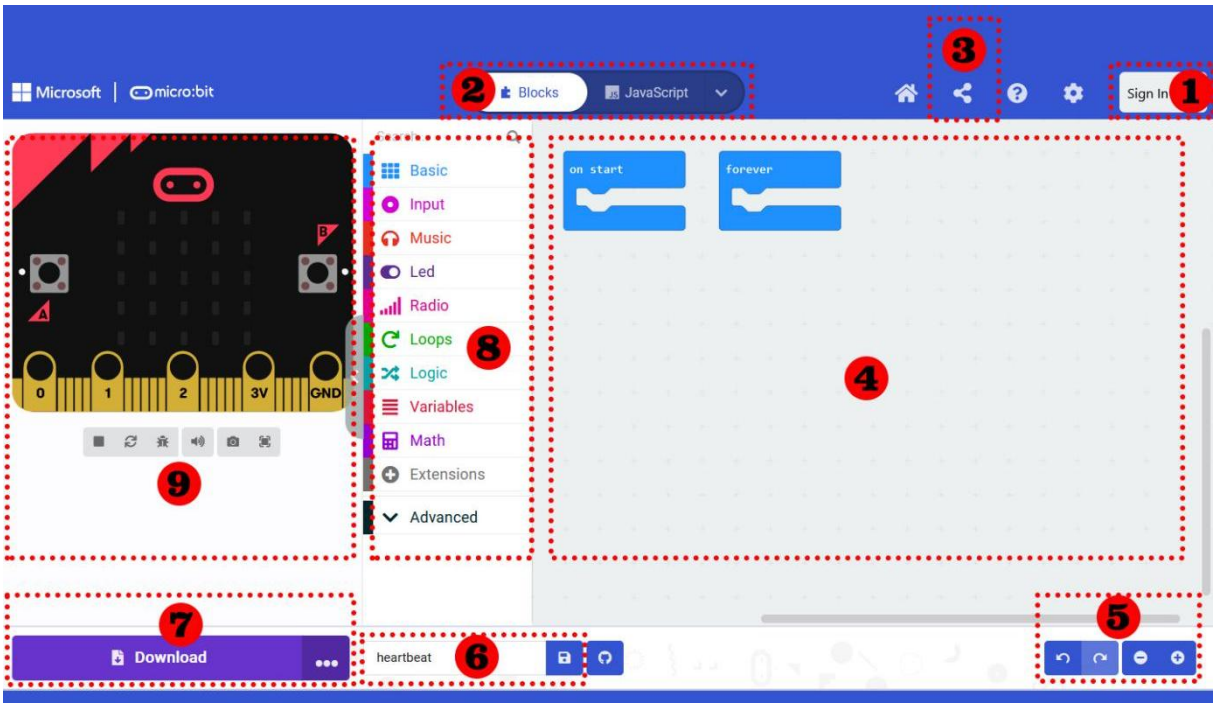
Give your project a name - we are going with **heartbeat**.
Click **Create** to get started with your project.



Introduce to the MakeCode editor

Now you are presented with the MakeCode editor, this is the screen you will use to program your micro:bit.

Below is a quick reminder of the different parts of the editor, to help you better understand the interface.

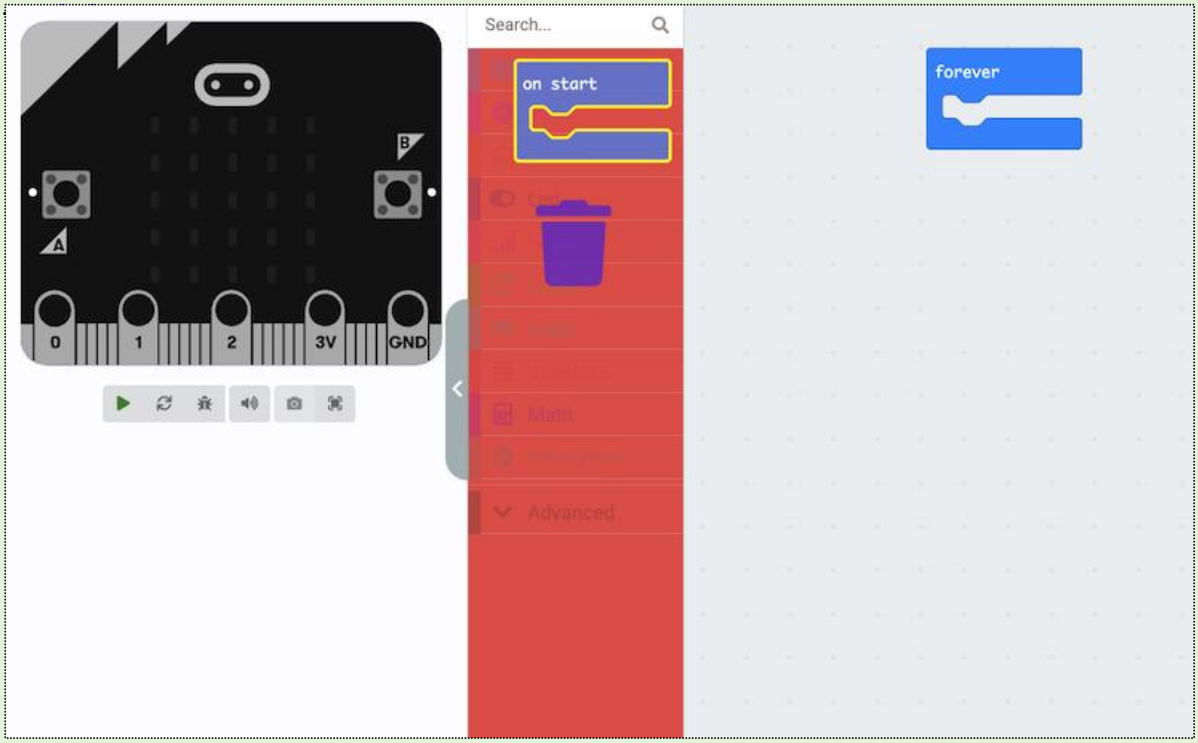


1	**Sign in** --- Sign in to save your progress and access your work anytime, anywhere.
2	**Blocks/JavaScript/Python** --- Choose your own adventure by programming in blocks (default) or in JavaScript. Not shown in the image, Microsoft also eventually added an additional option to use convert the code to MicroPython.
3	**Share** --- Allows you to share your project code in a number of different ways with your friends!
4	**Program Space** --- This is where the magic happens and where you build your program...where you "make code."
5	**Zoom/Undo-Redo** --- Sometimes you need to undo things, or zoom out and look around; these are the buttons for that.
6	**Name & Save** --- Name your program and save it (download it) to your computer as a .hex file. You can drag this file to your micro:bit using your computer's file explorer.
7	**Download** --- Transfer the code directly to your micro:bit using webUSB. Or similar to Save, download your program as a .hex file and drag it into your micro:bit.
8	**Block Library** --- The toolbox is where you get the blocks that make up your program. It is split into categories that are colour coded.
9	**Simulator** --- You don't need hardware! MakeCode has a real-time simulator! As you change your program, you can see what it will do on this virtual micro:bit!

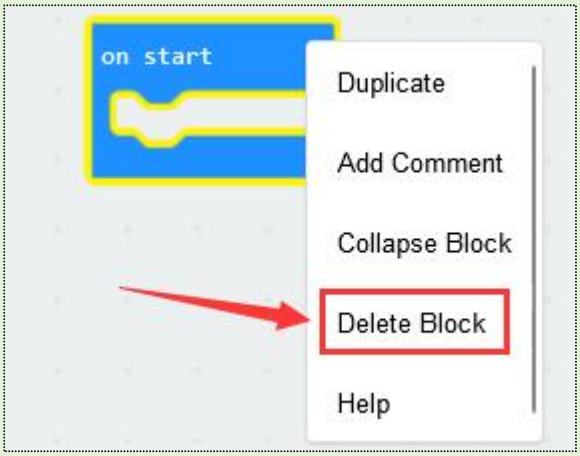
Creating your heartbeat program

In the code area, there are two fixed blocks "on start" and "forever".
The code in the "on start" block will be executed only once after power-on or reset, while the code in "forever" block will be executed circularly.

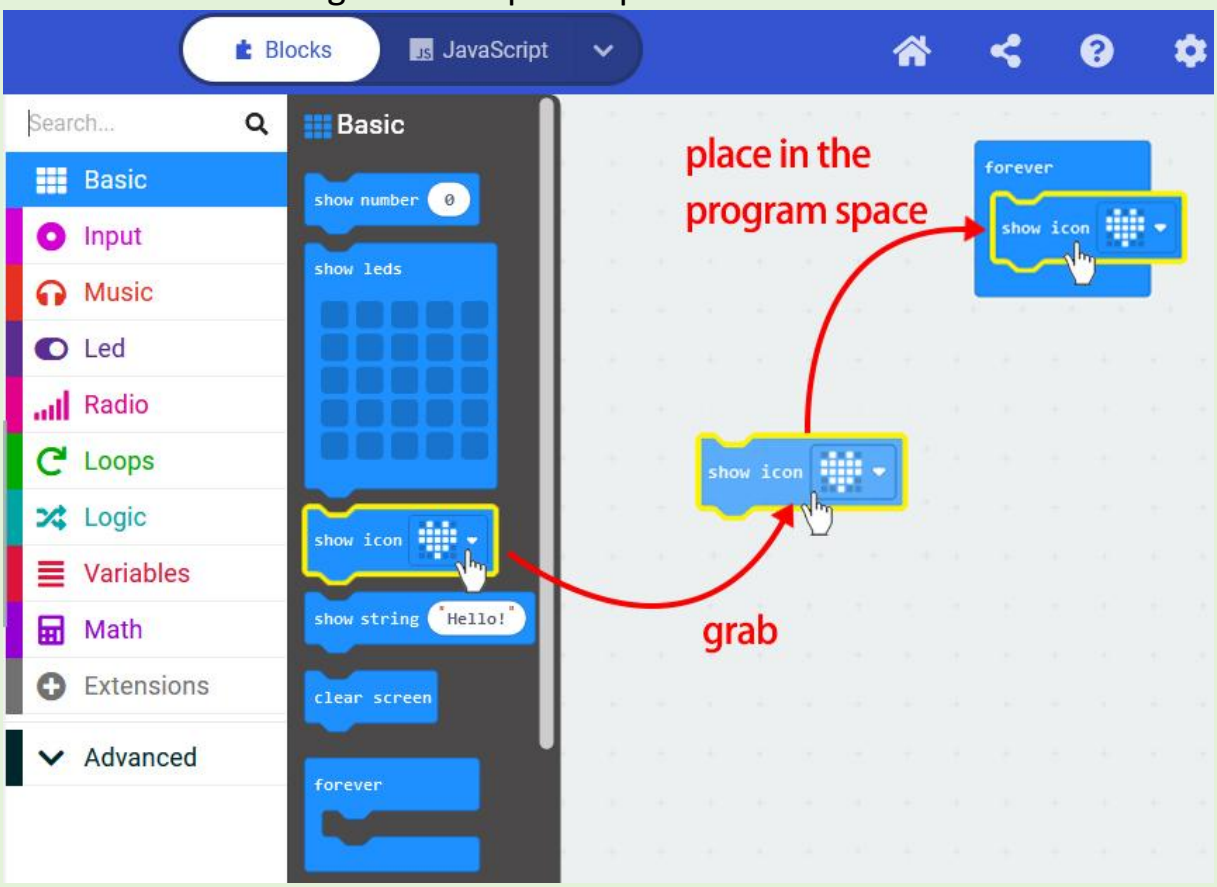
You only need the forever block for this program.
Grab the on start and drag it over the **Toolbox.**
You should see a rubbish bin appear when you do this. Release the block



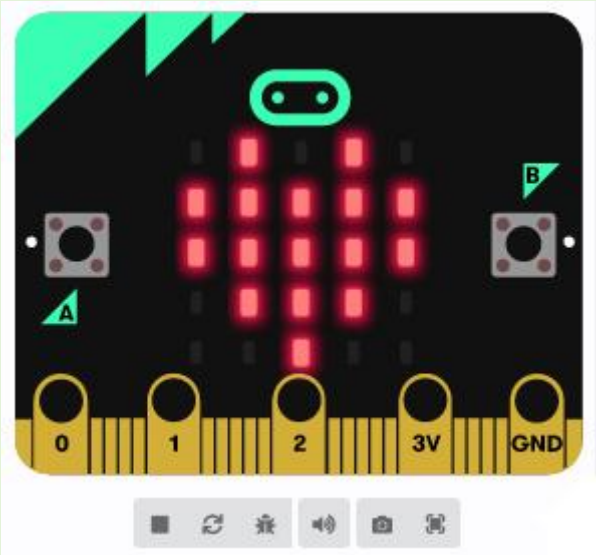
You can also **right-click** on start and click **Delete block** to remove it.



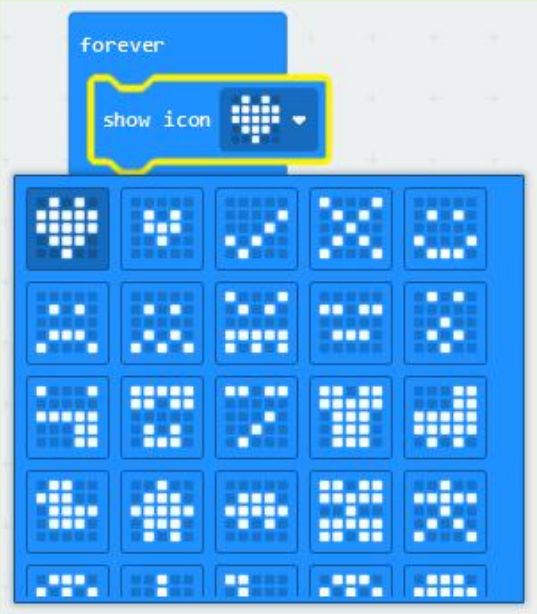
Click the **Basic menu** in the T**oolbox**. Grab a **show icon** block.
Place the **show icon** inside the **forever** block in the **program space**.
The blocks will fit together like puzzle pieces.



The **simulator** will immediately run your program, you will see the heart pattern displayed on the LEDs.

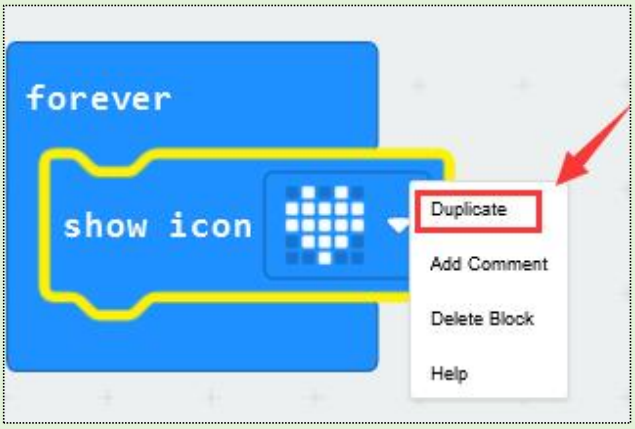


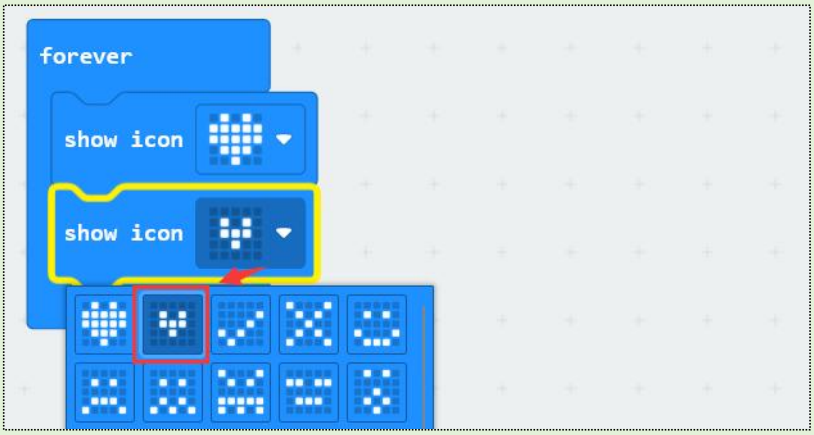
Click on the heart icon at the end of the **show icon** block. You will see a drop down, with all the available pre-made icons.



Click **Basic** again and drag the **show icon** block to place it underneath the **show icon** block you just created.

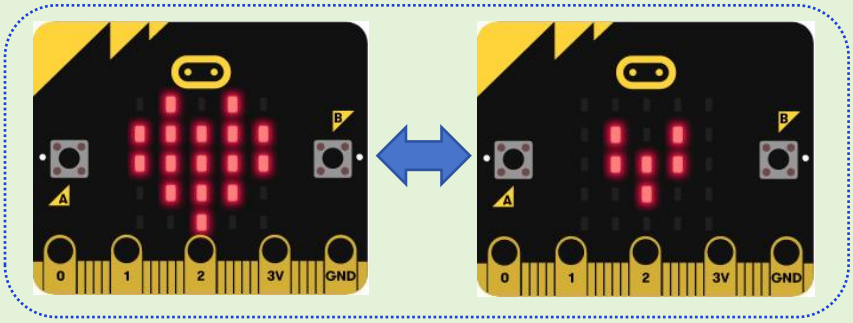
Or you can right click the **show icon** and click **Duplicate**, the same **show icon** block will appear in the program space.





You should notice that when you make a change to your code, the simulator restarts.

The simulator will display a beating heart.



In the next step you will learn how to download your program onto your physical micro:bit.

2.3 Transfer Code to The Micro:bit

The official provides detailed tutorials on how to transferring a program from **multiple devices** to the Microbit**.**
https://microbit.org/get-started/user-guide/transfer-code-to-the-microbit

It covers:

- Transferring a program from MakeCode or the micro:bit Python Editor
  - from a computer
  - from an Apple device (iPad or iPhone)
  - from an Android device
- Transferring a program that has been downloaded as a file

We will introduce you two methods to transfer the program :

2.31 Method 1: Flashing the Micro:bit with WebUSB

WebUSB

WebUSB is a developing web feature that allows you to access a micro:bit directly from a web page. With MakeCode this enables you to flash your micro:bit straight from the browser without the need to save the .hex file first, and use serial communication between the micro:bit and the editor.

WebUSB support for Your Micro:bit

If you're not using a current version of the Chrome or Microsoft Edge browsers, make sure they are this version or newer:

→Chrome (version 61 and newer) browser for Android, Chrome OS, Linux, macOS and Windows 10.

→Microsoft Edge (version 79 and newer) browser for Android, Chrome OS, Linux, macOS and Windows 10.

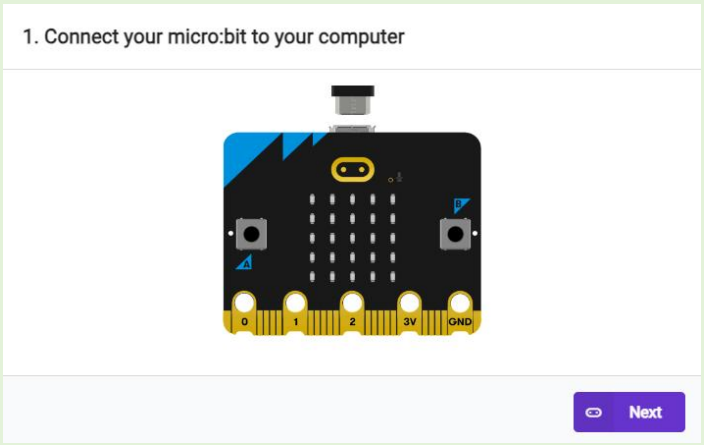
→Also, if you have a micro:bit V1 board, make sure that it is running version 0249 or above of the firmware.

Pair Your Micro:bit

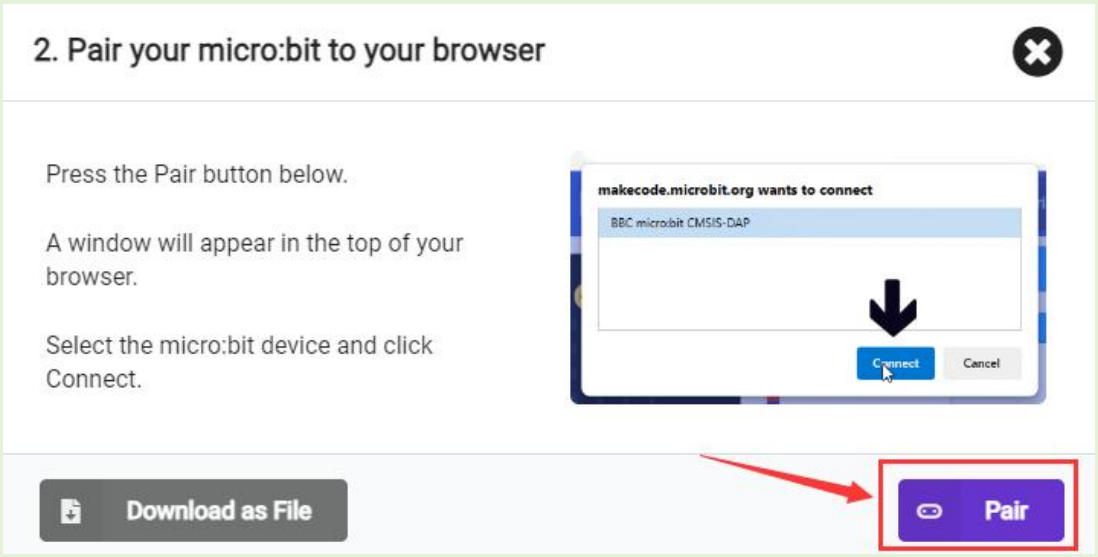
The first time you pair your micro:bit with your computer you'll need to go through a few easy steps to get setup. Here's how to get paired with WebUSB:

- Connect your micro:bit to your computer with a USB cable.
- Open the MakeCode editor at https://makecode.microbit.org/#editor
- Click on the **Download** button below the simulator.

- Click **Next**

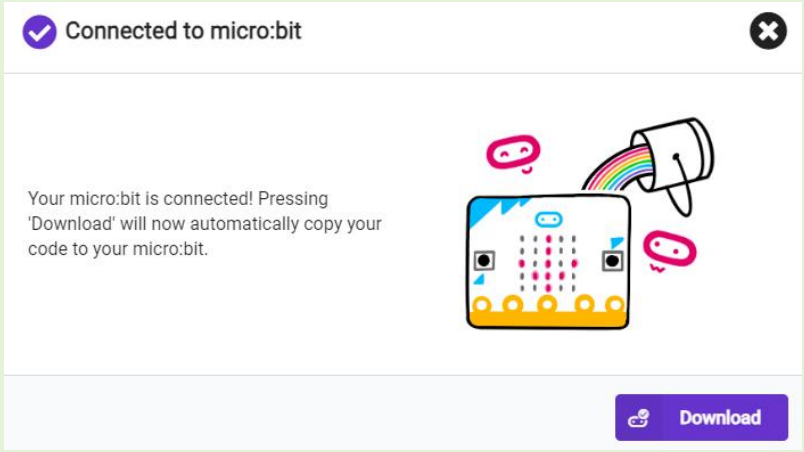


- Click **Pair** and select the microbit device and click **Connect**.



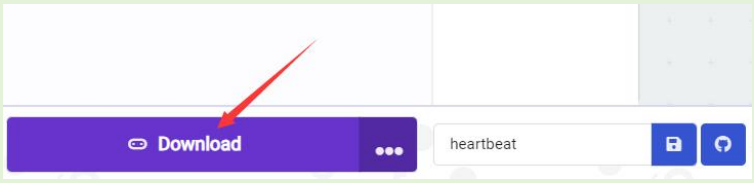
Note: If your micro:bit isn't showing up, try unplugging it and plugging it back in. You can also try a different USB port or USB cable if they are available.

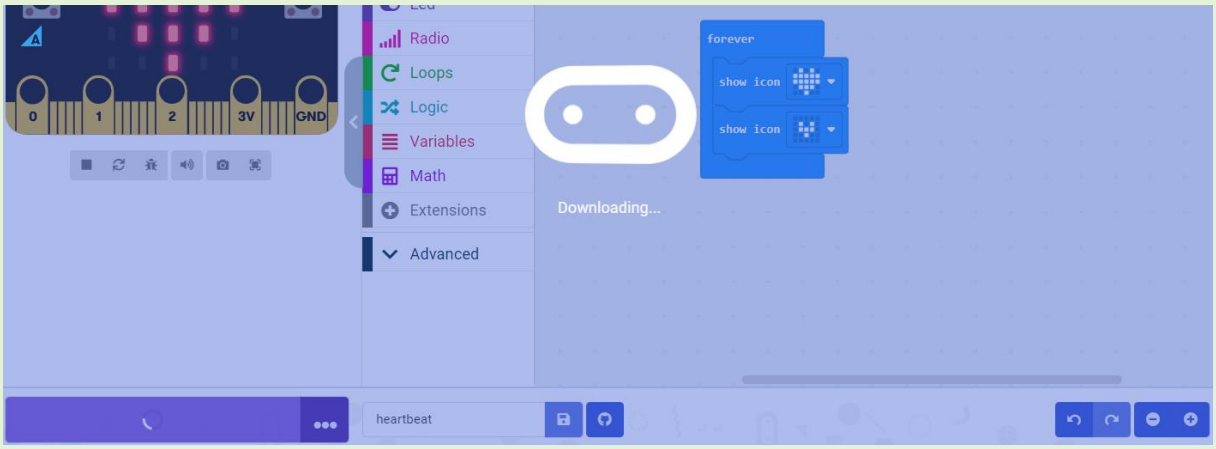
- You will see another box telling you the pairing was successful. Your micro:bit is connected!



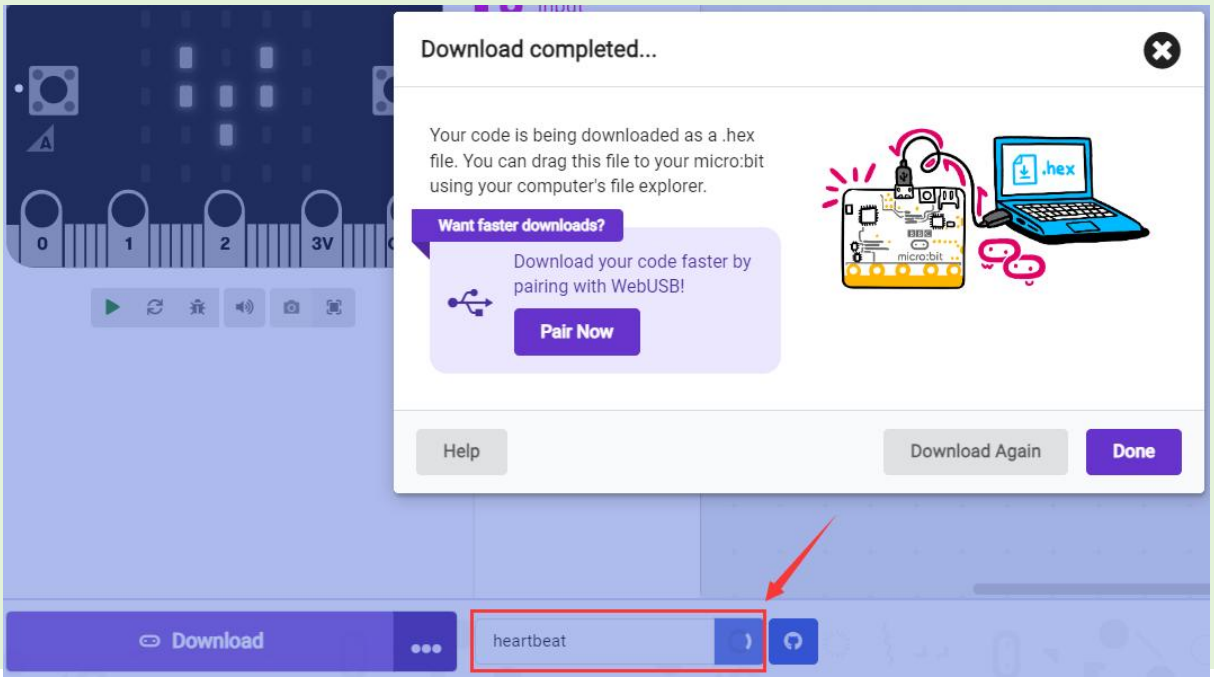
● Once your micro:bit is paired, MakeCode will use WebUSB to transfer the code directly and you won't have to drag and drop .hex files from a folder. Just click the Download button in the editor and your project code will just transfer to the micro:bit.

The yellow light on the back of the micro:it will flash really fast to let you know the program is transferring, then your program will start automatically.





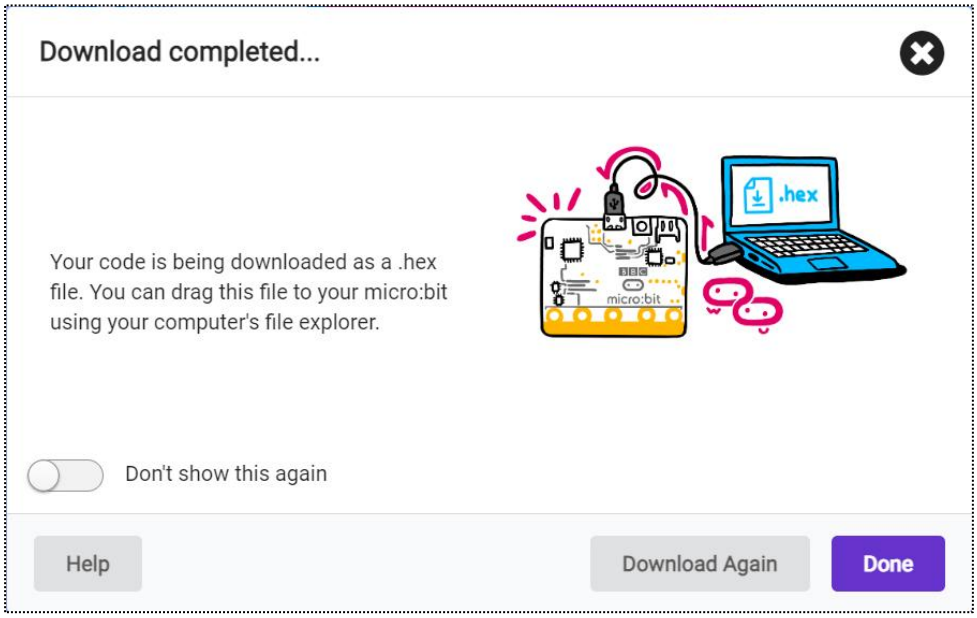
● **This method does not save a version of your code to your local machine**, if you wish to keep a copy of the program you have written, click the save icon (looks like a floppy disk and is located next to your project name in the MakeCode Editor) to save a copy of the hex file to your local machine.



● This will download your program file to your standard download location, probably the Downloads folder on your computer, or whatever location you have set in your download preference. You can copy or move it somewhere safe if you need to keep it to reload into the editor.

Using Safari/Firefox/Other

Note: If you are using any browser other than Chrome or Edge. It may not support WebUSB so you can't pair your micro:bit with your computer. Every time you click on the 'Download' button, your program won't transfer directly to your microbit, your code will be downloaded as a .hexfile. Just like click the save icon to save a copy of the hex file to your computer.You can drag this file to your micro:bit using your computer's file explorer.



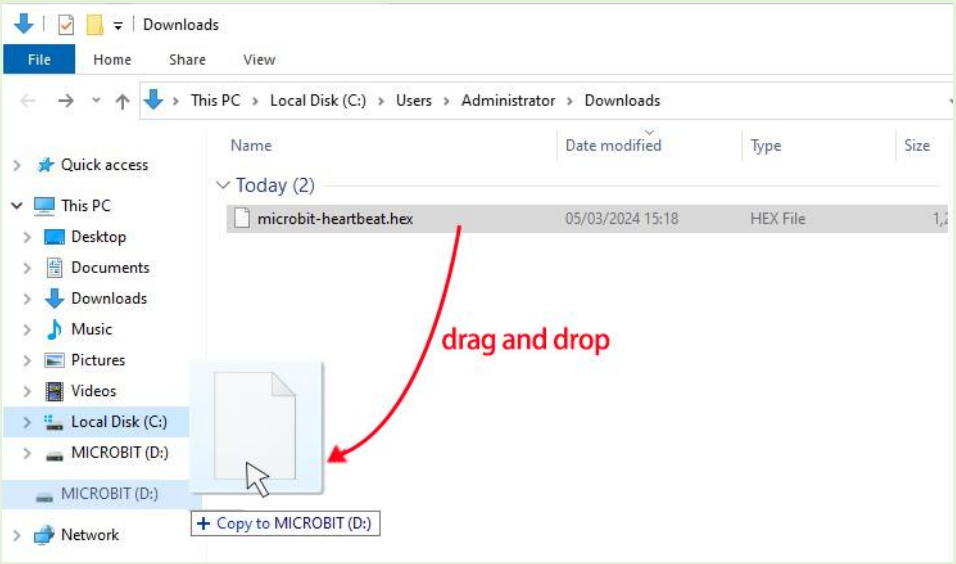
Next step, we will learn how to transfer a program that has been downloaded as a hex file.

2.32 Method 2: Transfer a program that has been downloaded as a hex file

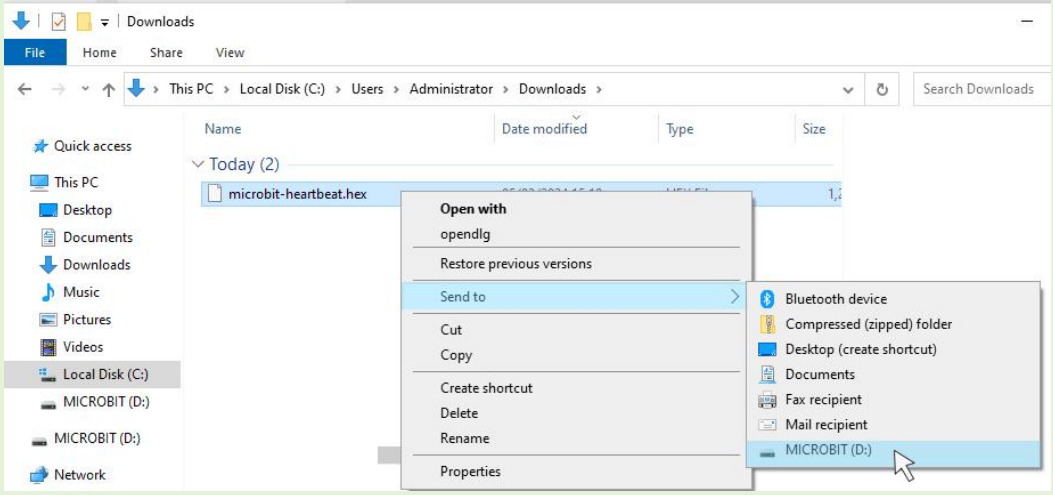
In the previous chapter, we have learned how to download the program to your local computer as a hex file.

- Connect your micro:bit to your computer with a USB cable
- Open Finder (Mac) / File explorer (Windows) and notice that your micro:bit is listed like a USB drive called MICROBIT.

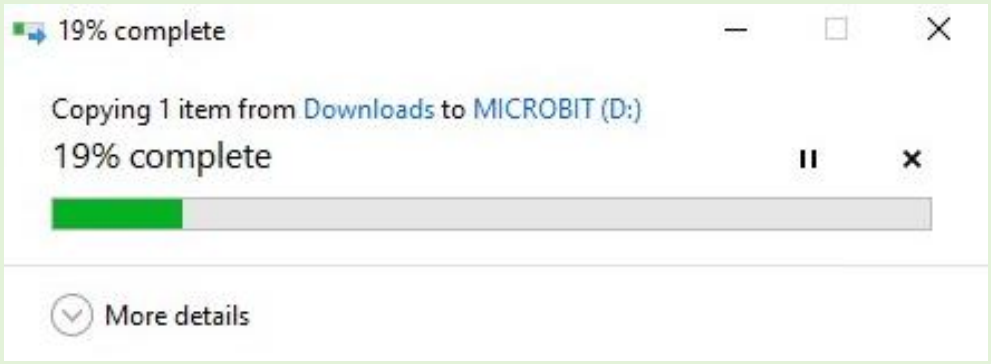
- Find the downloaded program hex file (e.g. in your local downloads folder) and drag and drop this on to the MICROBIT drive.



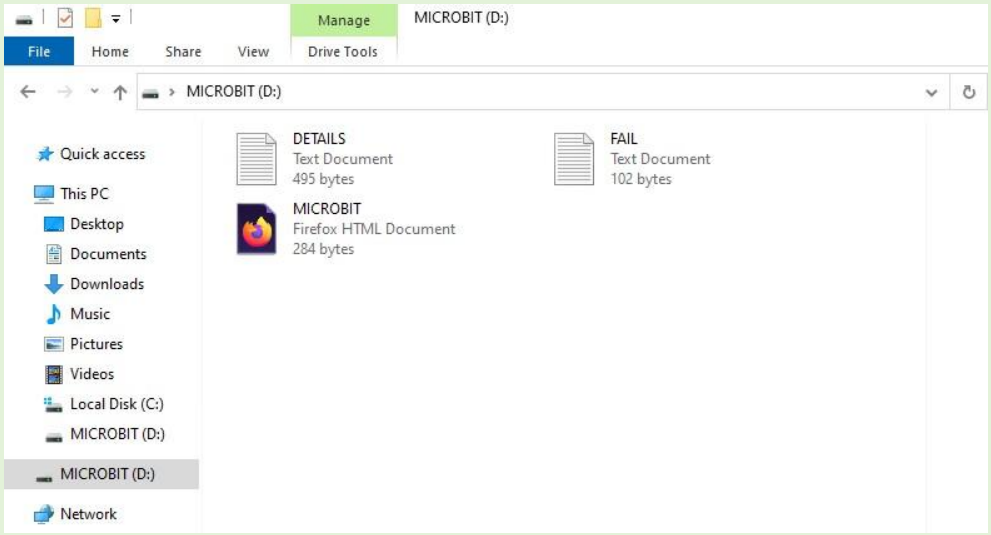
- Or you can right click and choose "Send to→MICROBIT."



● The yellow light on the back of the micro:it will flash really fast to let you know the program is transferring. Once the transfer is completed then your program will start automatically.



● After you transfer your .hex file, the MICROBIT drive will disconnect and reconnect as the micro:bit resets. If you look at the contents of the MICROBIT drive, you will not see the .hex file listed, this is normal, but your hex file will be run.



Note: The micro:bit can only receive hex files and won't store anything else!

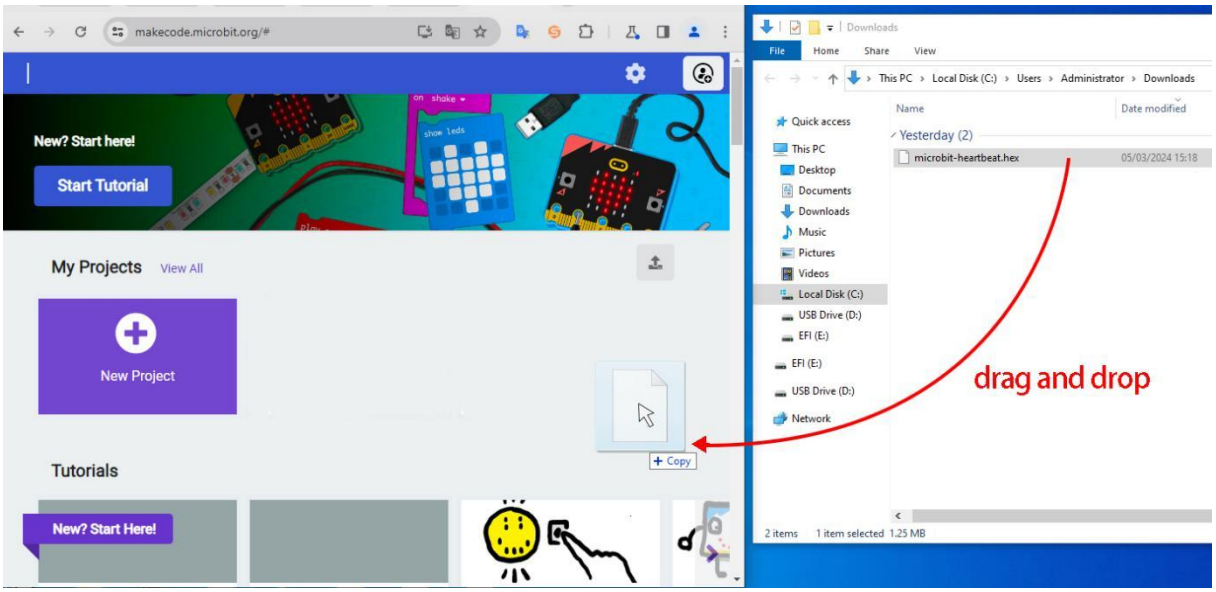
2.4 Import Files from Your Computer with the Makecode.

Steps：

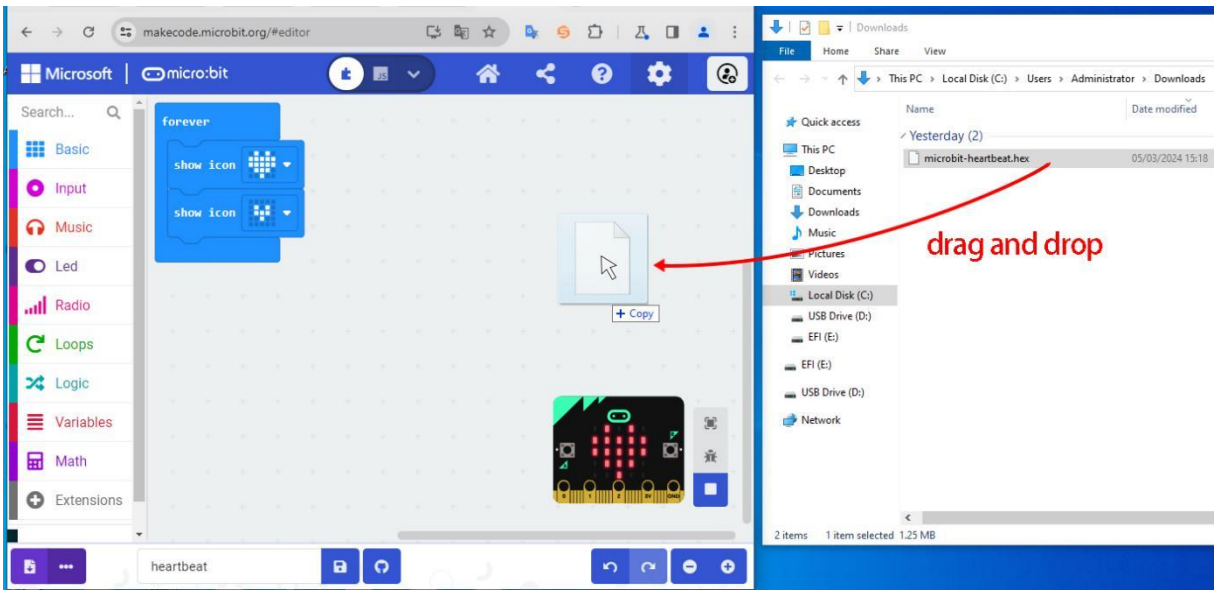
1. In a new browser window open https://makecode.microbit.org

2. There are two ways to import or update a saved hex file in the Makecode editor.

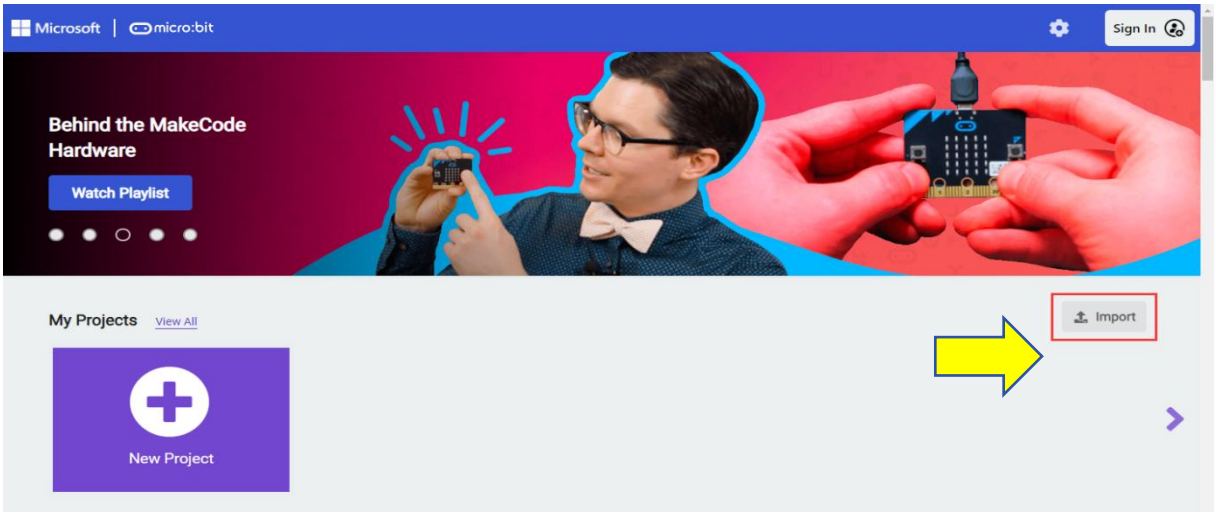
● Drag the HEX file from your computer into the home page or edit window.



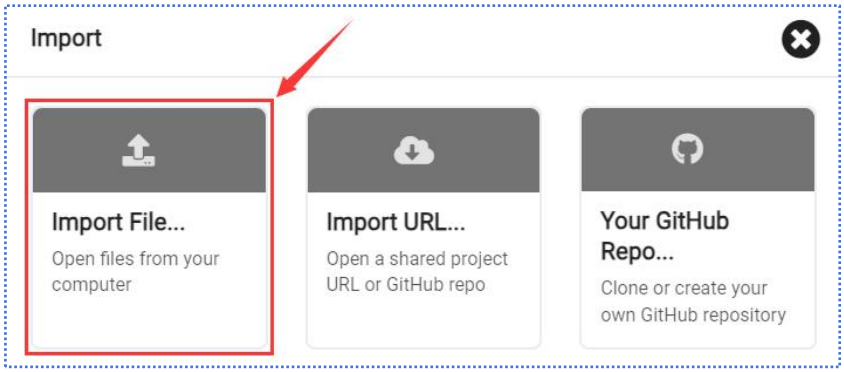
Note: The micro:bit can only run one program at a time - every time you drag-and-drop a hex file onto the device over USB it will erase the current program and replace it with the new one.



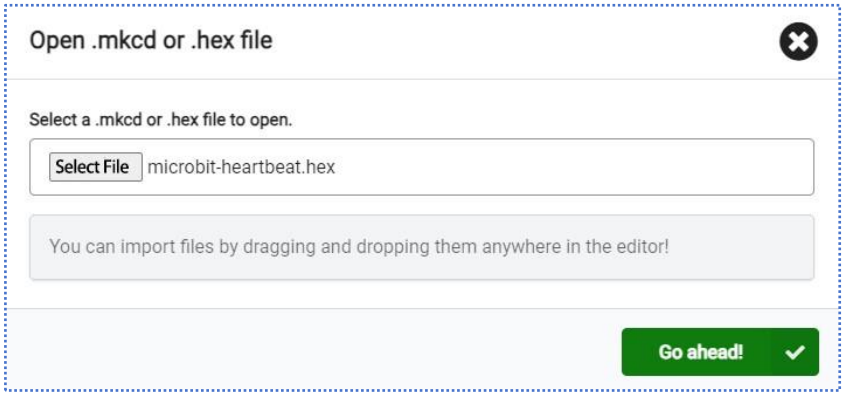
● Click “Import” button on the right side of HOME page.



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



Select file you saved. Then click “Go ahead!”



The file will open in the Makecode editor.

2.5 Micro:bit Project

What you will learn?

- We have learned how to use the MakeCode editor
- We have learned how to connect your micro:bit to your computer and download a MakeCode program to it
- Now we start learning how to use the micro:bit's components like; the LEDs, sensors, buttons and the speaker (V2 only).

2.51 Prepare:

Make sure you have downloaded the files we provide that are required to run the robot.

Link to download:

https://fs.keyestudio.com/KS4034-4035

Or

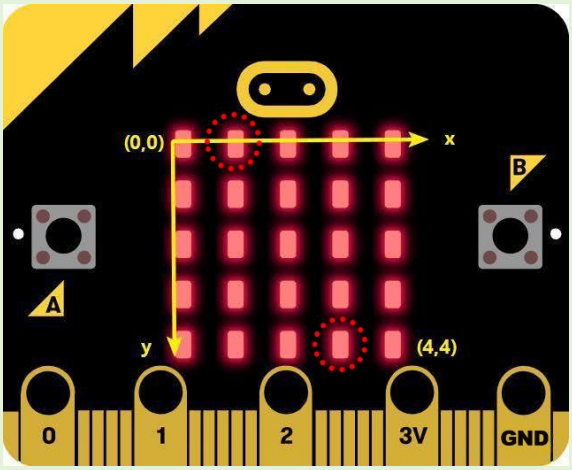
https://drive.google.com/drive/folders/18rCB4j9HJaP5_4ZFNeGA72mQCtmswm-y?usp=drive_link

Download the tutorial package and save it in the folder you want, and unzip it to use.

2.52 LED Matrix

There are 25 LEDs on the front face that you can use to show pictures, numbers, and words.

1 In this step, we will download the code to Microbit to light up the LEDs at coordinates (1,0) and coordinates (3,4).



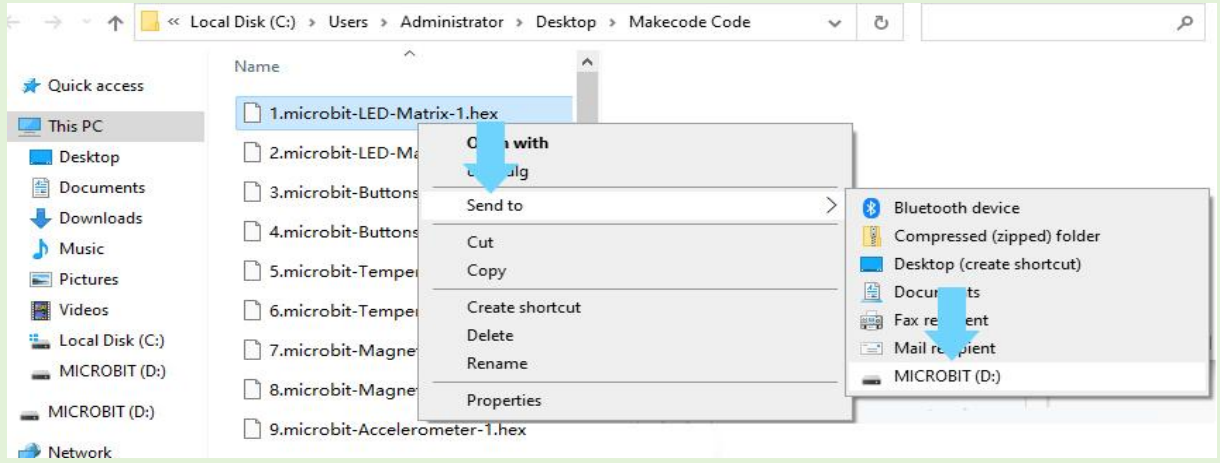
The upper left corner is (0,0) point, the lower right corner is (4,4) point, the horizontal direction (from left to right) is the x-axis direction, increasing in sequence [0-4], and the vertical direction (from top to bottom) is y-axis direction, increasing in sequence [0-4].

Steps:

- Connect your micro:bit to your computer with a USB cable.
- Find the downloaded microbit-LED-Matrix-1.hex file, drag it in the Makecode to view and analyze, and use WebUSB to flash it to the micro:bit from inside editor.

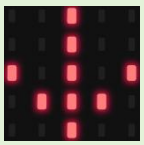
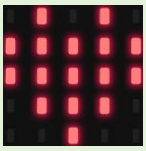
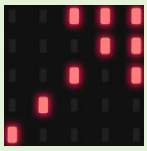
-Or you can right-click on it and choose "Send to→MICROBIT."

File Type	Path	File Name
hex file	.../Makecode/Makecode Code	1.microbit-LED-Matrix-1



Result: LEDs at coordinates (1,0) and coordinates (3,4) of the Microbit will flash alternately.

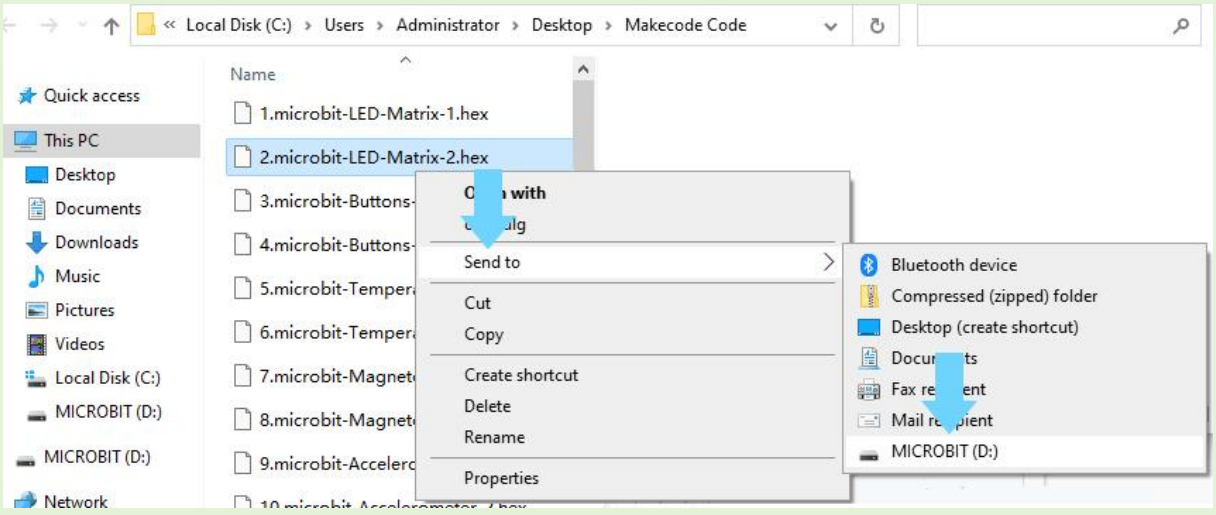
2 Next we will use the Microbit to display the numbers 1, 2, 3, 4, 5, and then cycle through the following patterns in the following order:


"Hello!"




Steps:

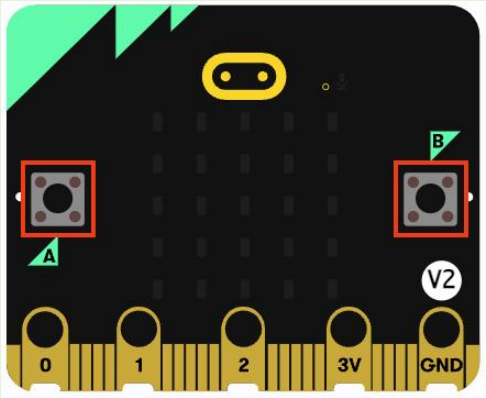
- Connect your micro:bit to your computer with a USB cable.
- Find the downloaded microbit-LED-Matrix-2.hex file, drag it in the Makecode to view and analyze, and use WebUSB to flash it to the micro:bit from inside editor.
- Or you can right-click on it and choose "Send to→MICROBIT."

File Type	Path	File Name
hex file	.../Makecode/Makecode Code	2.microbit-LED-Matrix-2



2.52 Programmable Buttons, A & B

Buttons are a very common input device. All micro:bit have two buttons you can program, and a reset button. The programmable buttons, A & B can trigger pieces of code in your programs.



1 We will make buttons A,B of the Microbit work with the LED matrix.

→Press button A, the LED matrix displays A

→Press button B, the LED matrix displays B

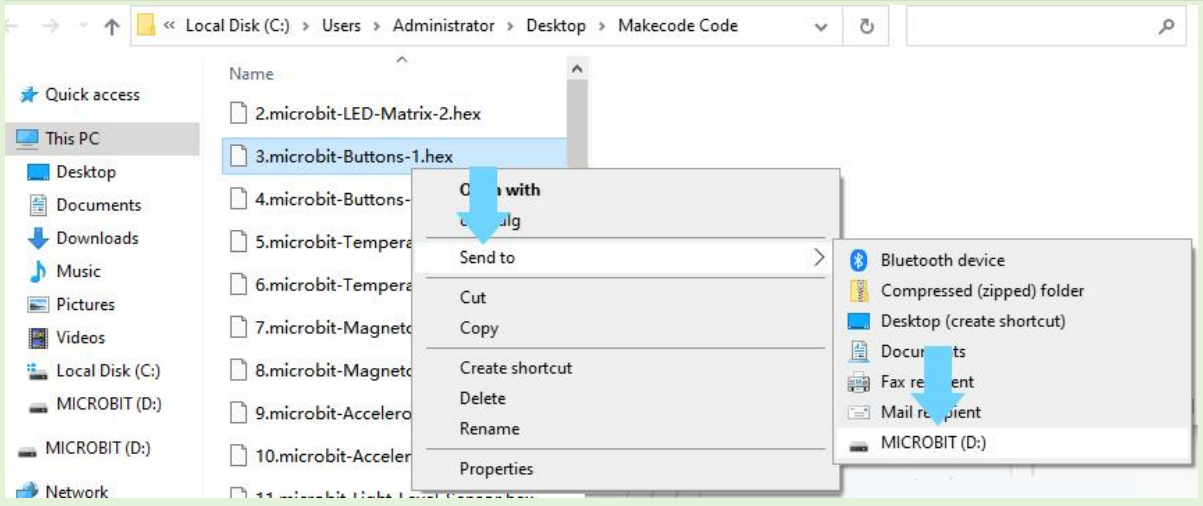
→Press button AB at the same time, the LED matrix displays AB

Steps:

- Connect your micro:bit to your computer with a USB cable.
- Find the downloaded microbit-Buttons-1.hex file, drag it in the Makecode to view and analyze, and use WebUSB to flash it to the micro:bit from inside editor.

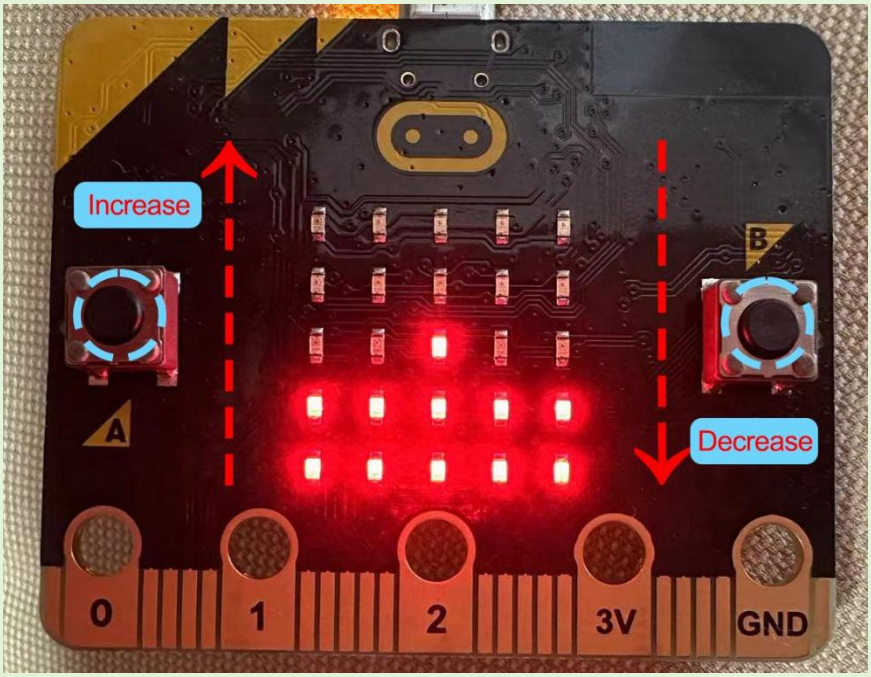
Or you can right-click on it and choose "Send to→MICROBIT."

File Type	Path	File Name
hex file	.../Makecode/Makecode Code	3.microbit-Buttons-1



2 Now let's send another code to Microbit to realize the following function:

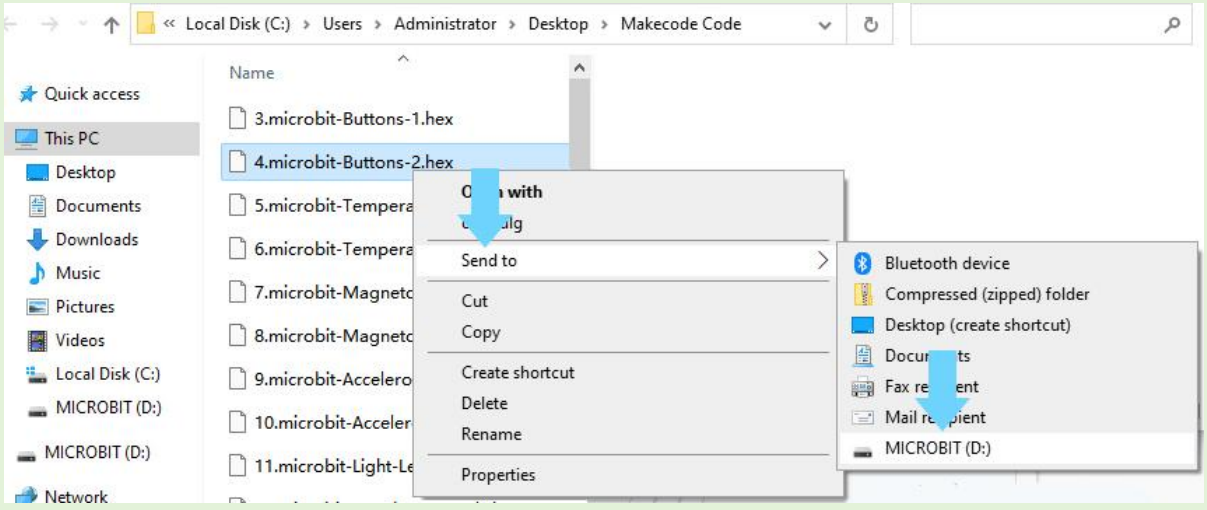
→When you press button A, the LED matrix is lit with more LEDs upwards, and when you press button B, LEDs of the mircrobit turn off downward.



Steps:

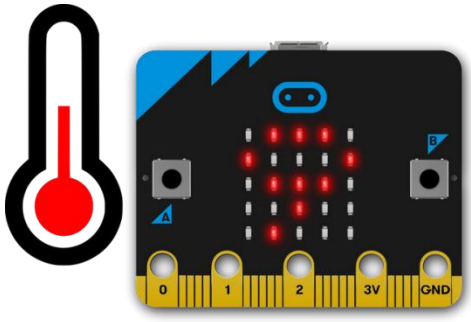
- Connect your micro:bit to your computer with a USB cable.
- Find the downloaded microbit-Buttons-2.hex file, drag it in the Makecode to view and analyze, and use WebUSB to flash it to the micro:bit from inside editor.
- Or you can right-click on it and choose "Send to→MICROBIT."

File Type	Path	File Name
hex file	.../Makecode/Makecode Code	4.microbit-Buttons-2



2.53 Temperature Sensor

The micro:bit does not actually have a temperature sensor. It uses the temperature sensor built into the NFR51822 chip for temperature detection. Therefore, the detected temperature is closer to the temperature of the chip and may have a certain error with the ambient temperature.



1 In this step we will shows how hot or cold your micro:bit is by taking a reading from the temperature sensor in its processor or CPU (central processing unit).

Steps:

- Connect your micro:bit to your computer with a USB cable.
- Find the downloaded microbit-Temperature-Sensor-1.hex file, drag it in the Makecode to view and analyze, and use WebUSB to flash it to the micro:bit from inside editor.

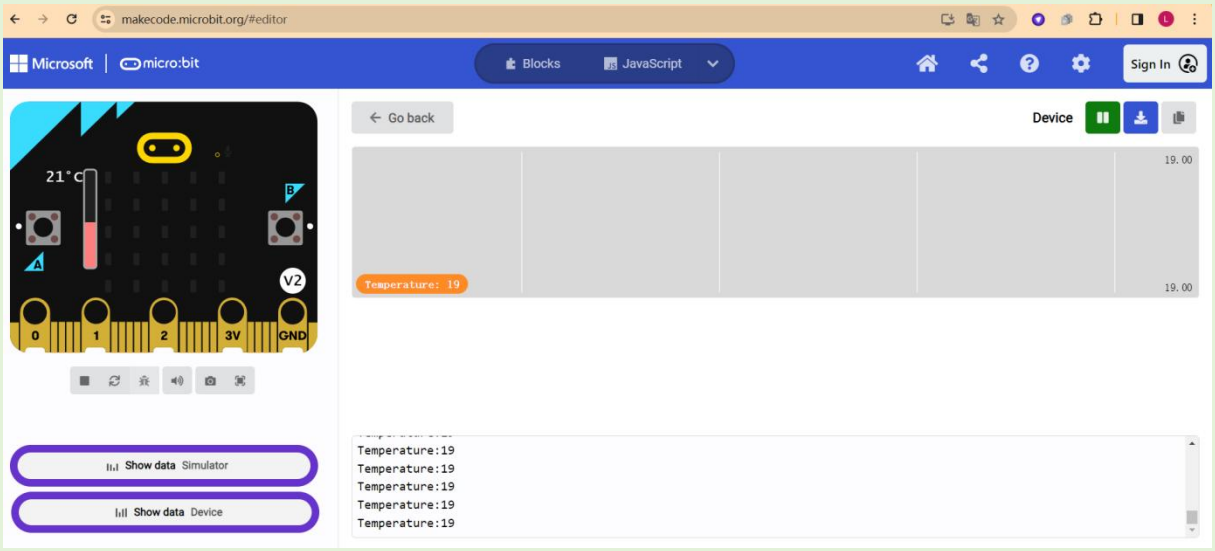
File Type	Path	File Name
hex file	.../Makecode/Makecode Code	5.microbit-Temperature-Sensor-1

- The "Show Data Device" button will appear below the simulator

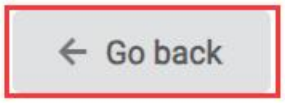
- If you press the Show data button, the editor will switch from the

Blocks or JavaScript view to display a charting window and a text console.

- The Data view window will display the temperature value read by the temperature sensor in the micro:bit's processor chip.



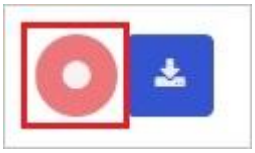
- **Return:** The Go back button switches the view back to previous code window (either Blocks or JavaScript).



Source: Tells you where the data is coming from. If the code writing the data is running in the simulator, then the source is Simulator. If your code is running on the micro:bit and connected by USB, the source is Device.

Pause: The pause button will stop the display of new values and stop scrolling. When you resume, the chart starts again with the current value written.

Resume: The resume button will start displaying new values after the Pause button was pressed.



Download: The download button collects the data your code has written and downloads it to your computer as a file called something like data-11-2018-23-00-0700.csv. The numbers in the filename are the date and time when the file is created. The file may automatically open in an editor or spreadsheet if one of those programs is associated with csv files.

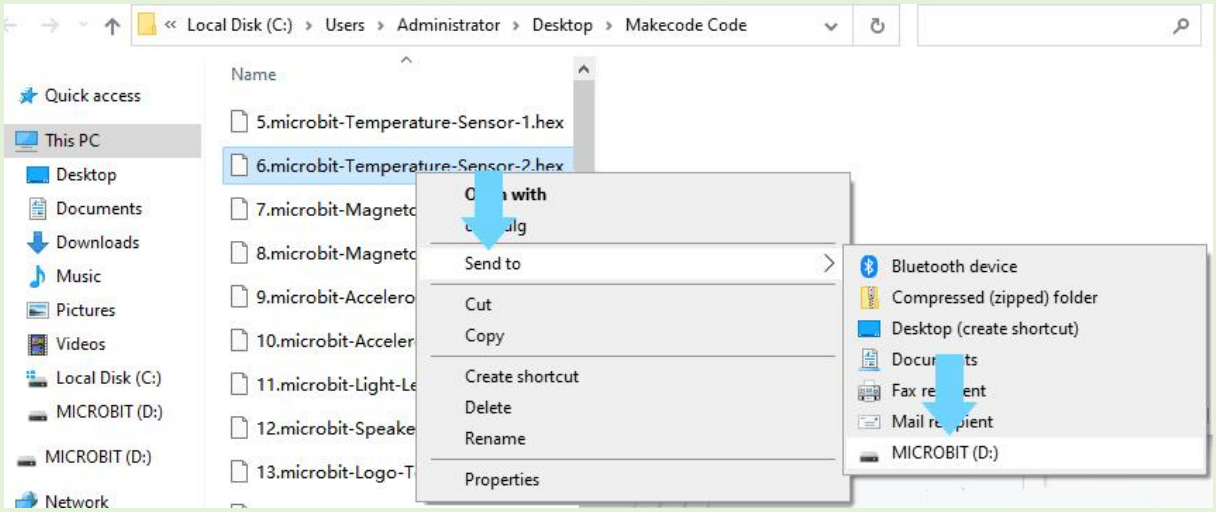
2 Now let's send another code to Microbit to make LED matrix of the microbit display the temperature detected by the temperature sensor.

Steps:

- Connect your micro:bit to your computer with a USB cable.
- Find the downloaded microbit-Temperature-Sensor-2.hex file, drag it in the Makecode to view and analyze, and use WebUSB to flash it to the micro:bit from inside editor.

Or you can right-click on it and choose "Send to→MICROBIT."

File Type	Path	File Name
hex file	.../Makecode/Makecode Code	6.microbit-Temperature-Sensor-2



- **Result:** When your micro:bit turns on, the temperature reading in Celsius will be displayed and scrolled across the LED array followed by the temperature reading in Fahrenheit.

2.54 Magnetometer

The micro:bit's built-in magnetometer chip is intended for use as compass to detect magnetic north. Like the compass app on your phone, this requires calibration.

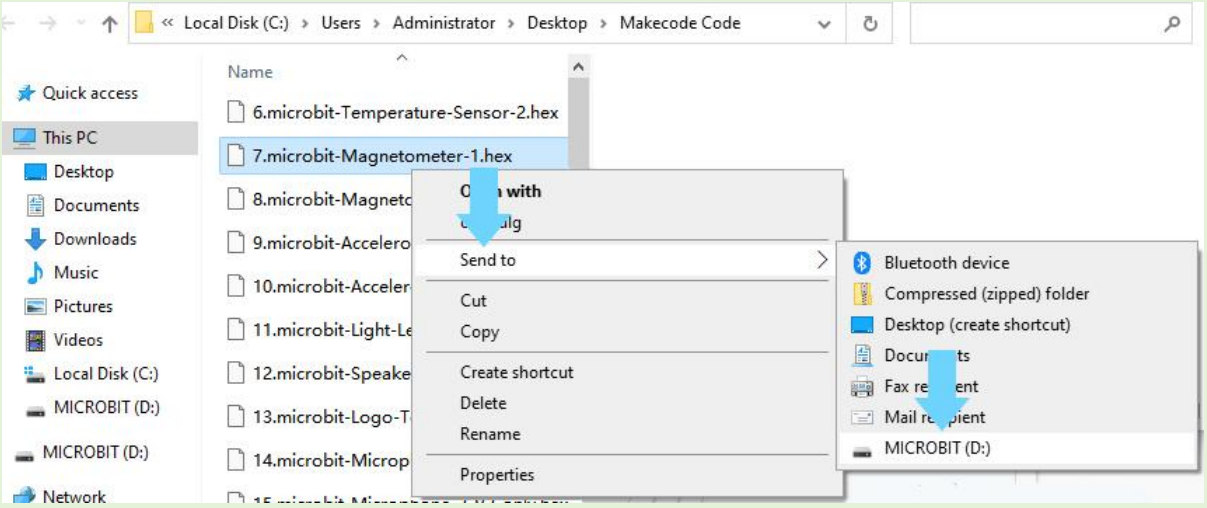
1 We first need to calibrate the compass of the microbit. Then make the led matrix display direction value read by the compass. North, east, south, and west correspond to 0°, 90°, 180°, and 270°.

Steps:

- Connect your micro:bit to your computer with a USB cable.
- Find the downloaded microbit-Magnetometer-1.hex file, drag it in the Makecode to view and analyze, and use WebUSB to flash it to the micro:bit from inside editor.

Or you can right-click on it and choose "Send to→MICROBIT."

File Type	Path	File Name
hex file	.../Makecode/Makecode Code	7.microbit-Magnetometer-1

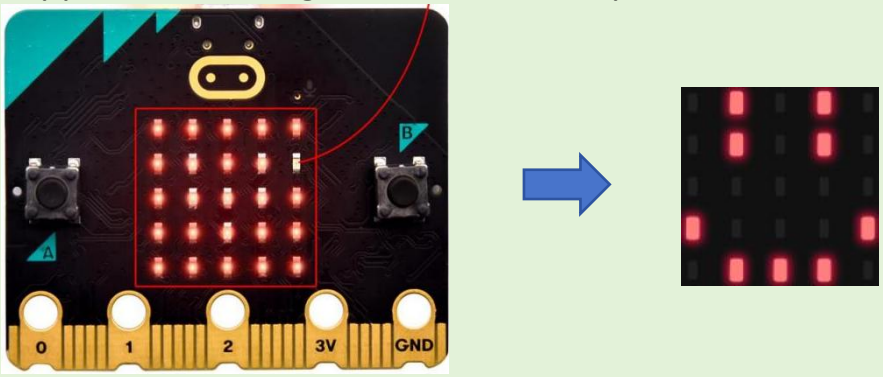


- After the code is downloaded to microbit, the LED matrix of the micro:bit prompts: "TILT TO FILL SCREEN", and then it enters calibration mode.

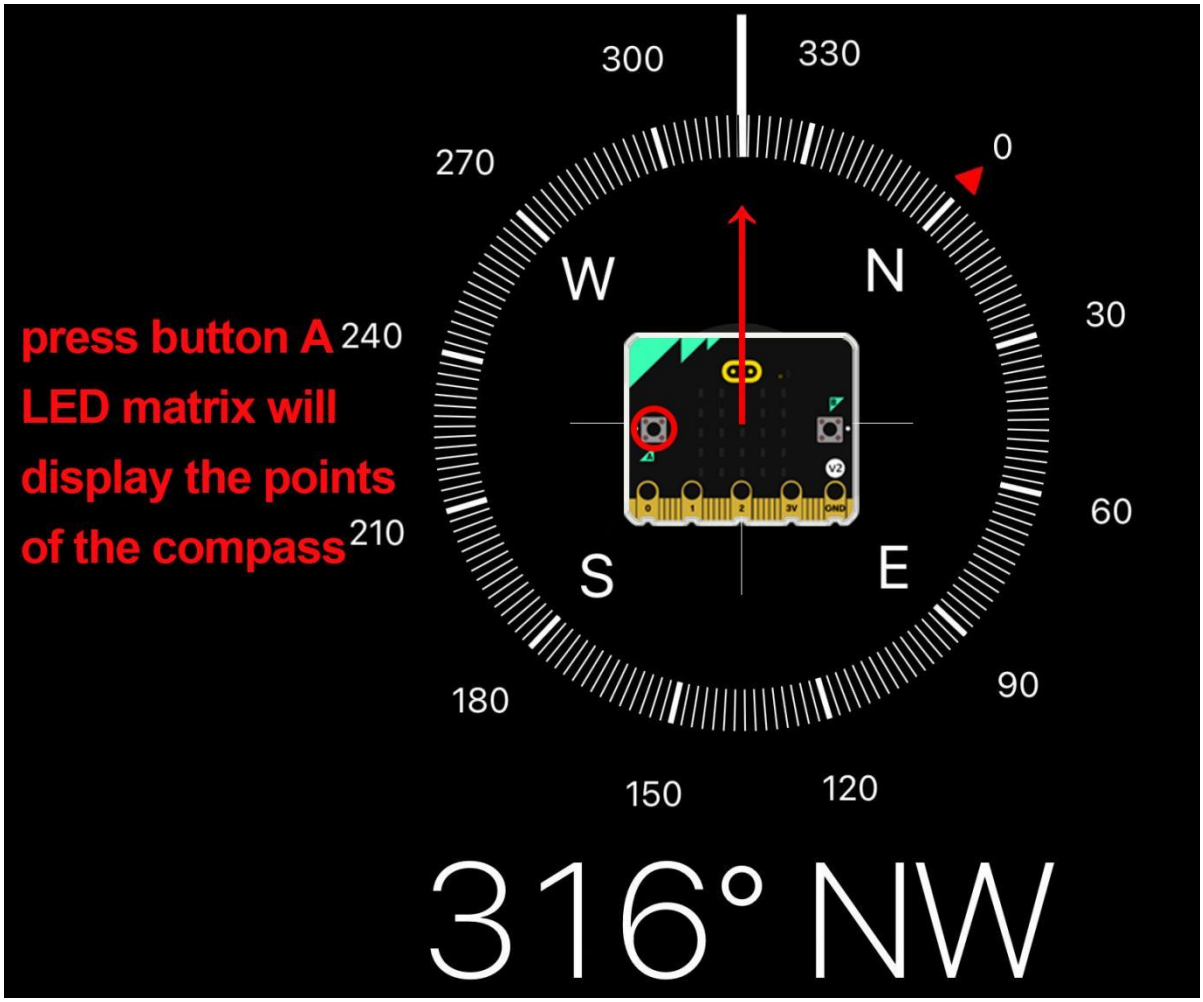
The calibration method is:

When you tilt the microbit in a certain direction, the LED matrix will light up more LEDs in that direction.

Keep tilting the microbit in all directions until all LEDs are lit, and a smiley face appears indicating calibration is complete!



- **Test Result:** Each time you press button A, position value read by the compass will be displayed on the LED matrix. Change the orientation of the microbit and you will notice that the position value changes accordingly.



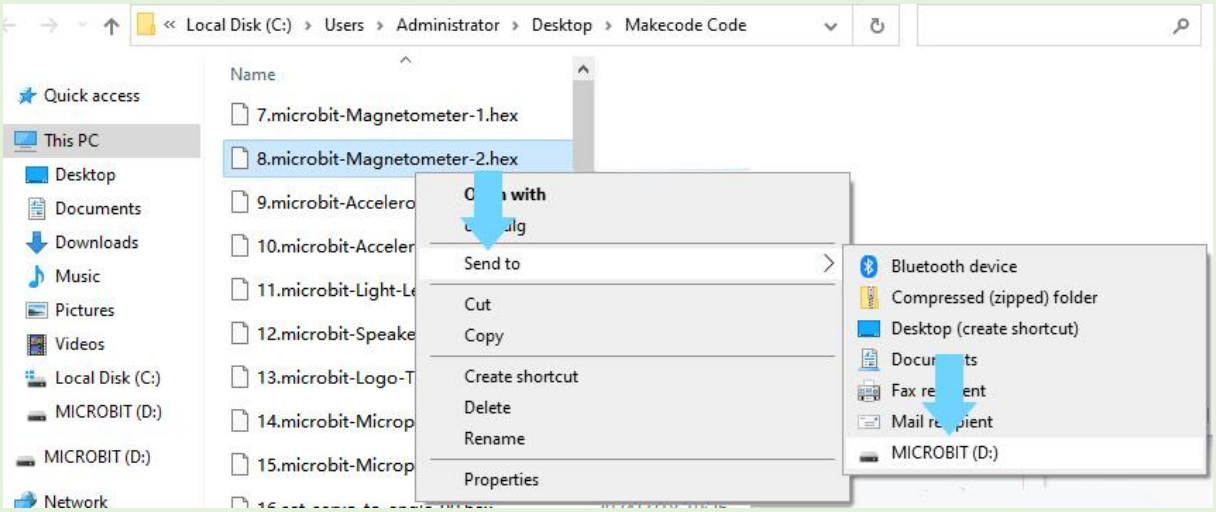
2 Here, we download a new code to make the arrow on the LED matrix of the Microbit always point to the north.

Steps:

- Connect your micro:bit to your computer with a USB cable.
- Find the downloaded microbit-Magnetometer-2.hex file, drag it in the Makecode to view and analyze, and use WebUSB to flash it to the micro:bit from inside editor.

Or you can right-click on it and choose "Send to→MICROBIT."

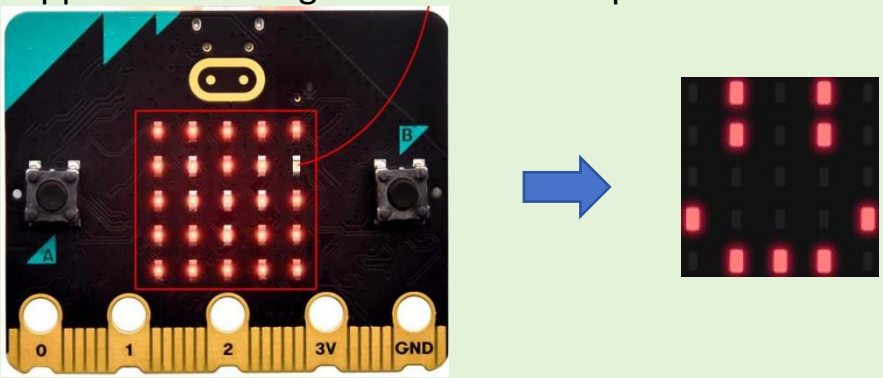
File Type	Path	File Name
hex file	.../Makecode/Makecode Code	8.microbit-Magnetometer-2



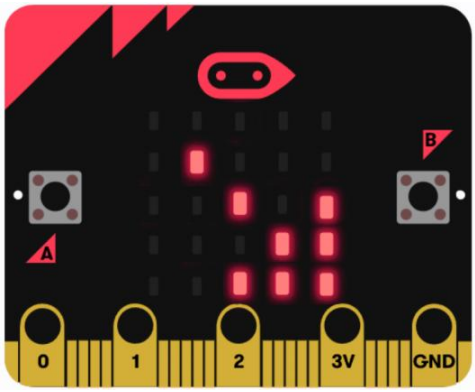
- After the code is downloaded to microbit, the LED matrix of the micro:bit prompts: "TILT TO FILL SCREEN", and then it enters calibration mode.
- **The calibration method is:**

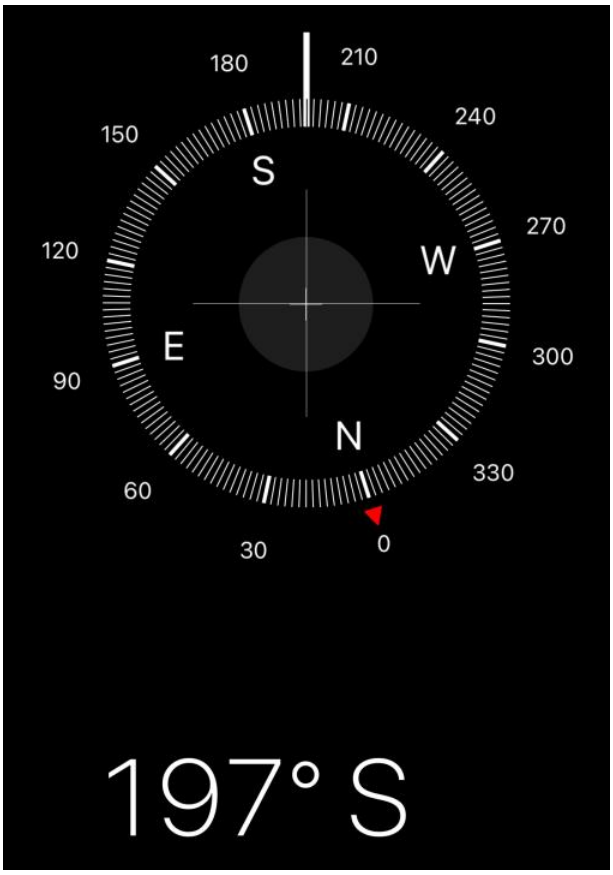
When you tilt the microbit in a certain direction, the LED matrix will light up more LEDs in that direction.

Keep tilting the microbit in all directions until all LEDs are lit, and a smiley face appears indicating calibration is complete!



● **Test Result:** Place the microbit's LED matrix horizontally facing up and change its orientation horizontally, you will find that no matter how you change its orientation, the arrow displayed by its LED matrix points in the same direction. If you have a compass app on your phone, turn the compass app on, hold your microbit horizontally in the same direction as your phone and you will see that the arrow of the microbit is pointing at North9(0 degree).

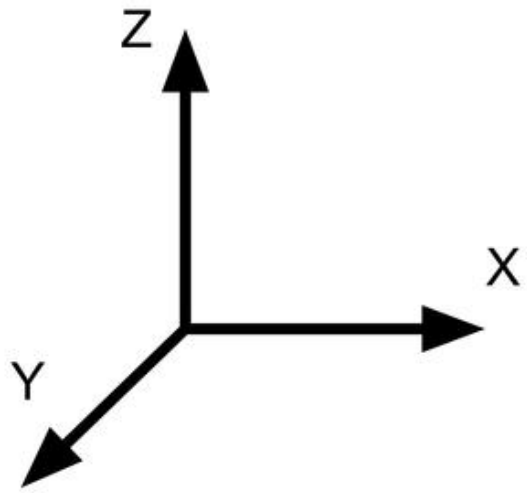




2.54 Accelerometer

An accelerometer is a motion sensor that measures movement. The accelerometer in your BBC micro:bit detects when you tilt it left to right, backwards and forwards and up and down.

If you imagine the micro:bit sitting flat on a desk, the x dimension is left to right; the y dimension front to back and the z dimension into and out of the desk. We can use an accelerometer to detect the orientation of the micro:bit, since gravity will always be acting in the same downwards direction.



X: acceleration in the left and right direction.
Y: acceleration in the forward and backward direction.
Z: acceleration in the up and down direction.
Strength: the resulting strength of acceleration from all three dimensions (directions).

A number that means the amount of acceleration

When the micro:bit is lying flat on a surface with the screen pointing up,
-x is 0,
-y is 0,
-z is -1023,
-and strength is 1023.

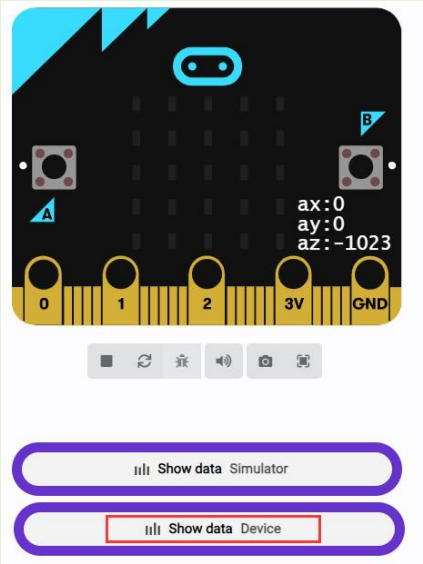
1 In this step we will use the Data View Window to display the values of the x, y, and z axes detected by accelerometer.

Steps:

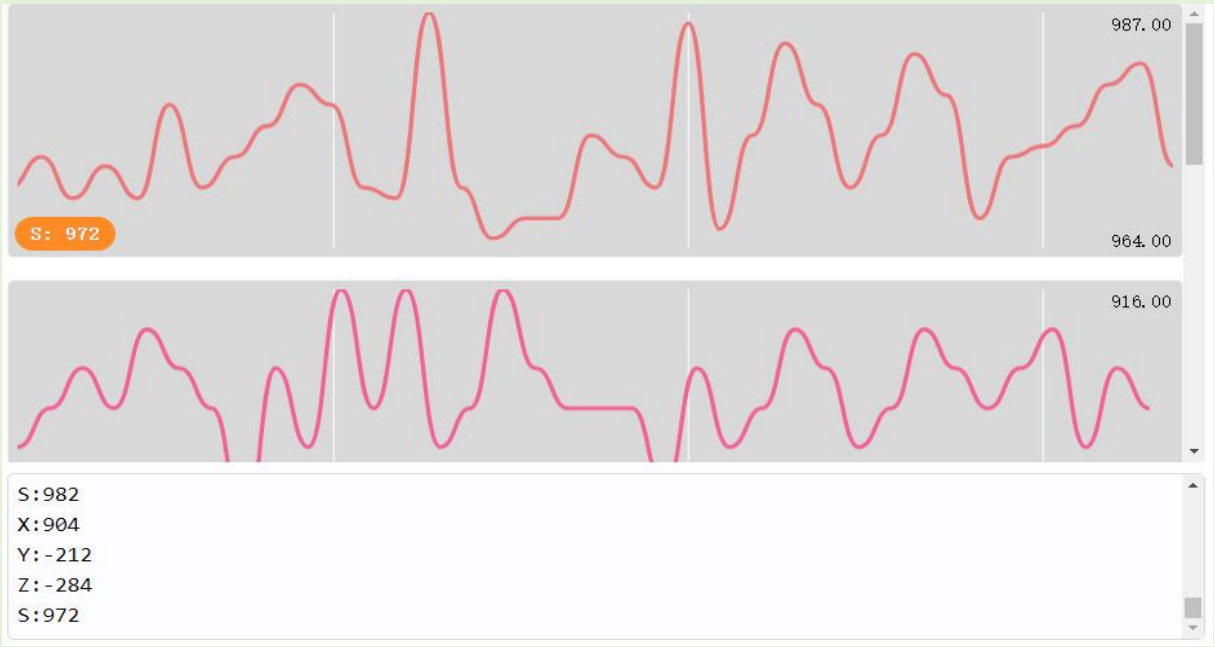
- Connect your micro:bit to your computer with a USB cable.
- Find the downloaded microbit-Accelerometer-1.hex file, drag it in the Makecode to view and analyze, and use WebUSB to flash it to the micro:bit from inside editor.

File Type	Path	File Name
hex file	.../Makecode/Makecode Code	9.microbit-Accelerometer-1

- Click the "Show Data Device" under the simulator



- You will see the values of the X, Y, and Z axes detected by the accelerometer in the Data View Window.



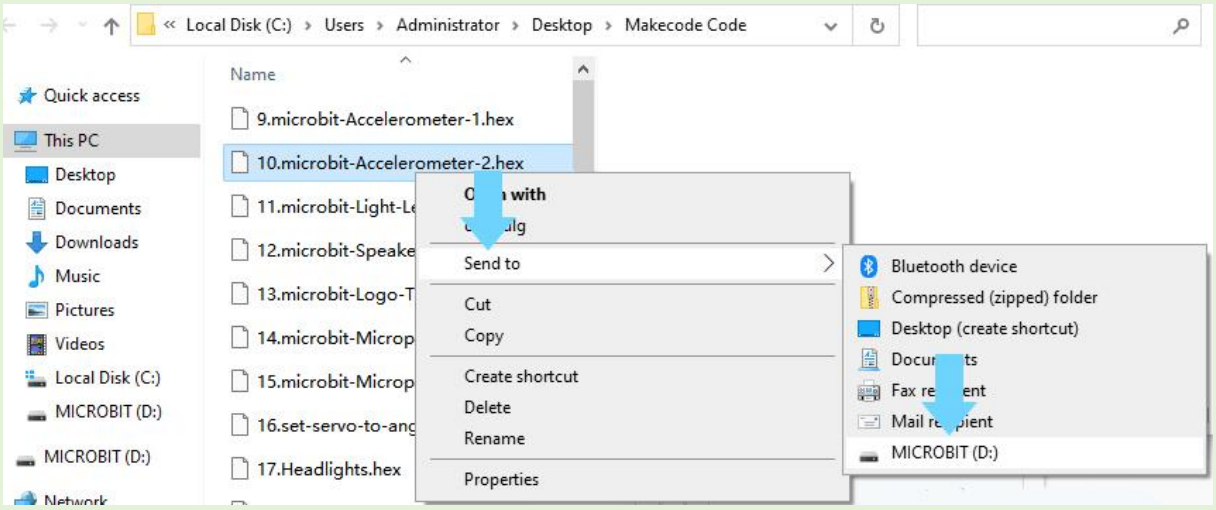
2 Now let us upload new code so that when the accelerometer detects the posture of the Microbit change, the Microbit will display the corresponding number on the LED matrix.

Steps:

- Connect your micro:bit to your computer with a USB cable.
- Find the downloaded microbit-Accelerometer-2.hex file, drag it in the Makecode to view and analyze, and use WebUSB to flash it to the micro:bit from inside editor.

Or you can right-click on it and choose "Send to→MICROBIT."

File Type	Path	File Name
hex file	.../Makecode/Makecode Code	10.microbit-Accelerometer-2



- **Test result:**

Shake the microbit: LED matrix shows number 1

Logo up: LED matrix shows number 2

Logo down: LED matrix shows number 3

Screen up: LED matrix shows number 4

Screen down: LED matrix shows number 5

Tilt left: LED matrix shows number 6

Tilt right: LED matrix shows number 7

Free fall: LED matrix shows number 8

2.55 Light Level Sensor

Find the light level (how bright or dark it is) where you are. The light level 0 means darkness and 255 means bright light. The micro:bit measures the light around it by using some of the LEDs on the LED screen.

The first time you use it, this function will say 0. After that, it will say the real light level. This is because the light sensor (the part that can find the light level) has to be turned on first.

In this step we will use the Data View Window to display the values of the light intensity detected by light level sensor.

Steps:

- Connect your micro:bit to your computer with a USB cable.
- Find the downloaded microbit-Light-Level-Sensor.hex file, drag it in the Makecode to view and analyze, and use WebUSB to flash it to the micro:bit from inside editor.

File Type	Path	File Name
hex file	.../Makecode/Makecode Code	11.microbit-Light-Level-Sensor

- Click the "Show Data Device" under the simulator
- You will see the values of light intensity in the Data View Window.


```
Light intensity:59
Light intensity:58
Light intensity:59
Light intensity:58
Light intensity:59
```

2.56 Speaker (V2 only)

The micro:bit can make sounds, play tunes and the new micro:bit has a built-in speaker to make it even easier to make expressive and useful projects.

Any micro:bit sound project will work with the speaker, but with the new micro:bit you can also express yourself with some new sounds: make your micro:bit giggle, greet you or let you know when it's sleepy or sad.

You can also mute the speaker and sound will still come out of the pins so you can still enjoy micro:bit music on headphones connected to GND and pin 0. In MakeCode, use the music block 'set on-board speaker off'.

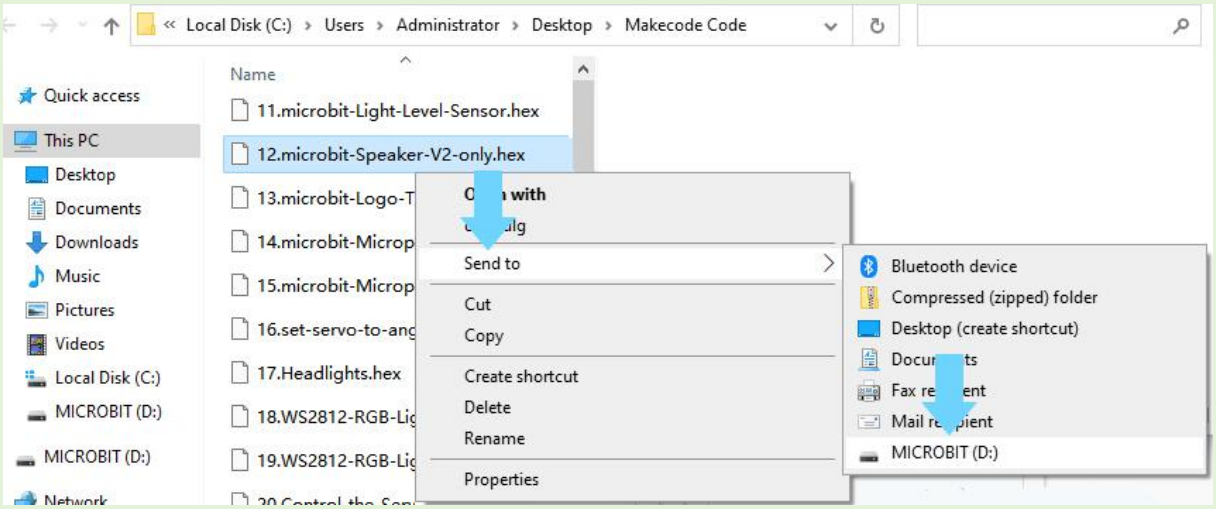
In this step we're going to make the Microbit produce interesting sounds.

Steps:

- Connect your micro:bit to your computer with a USB cable.
- Find the downloaded microbit-Speaker-V2-only. hex file, drag it in the Makecode to view and analyze, and use WebUSB to flash it to the micro:bit from inside editor.

Or you can right-click on it and choose "Send to→MICROBIT."

File Type	Path	File Name
hex file	.../Makecode/Makecode Code	12.microbit-Speaker-V2-only



- Result: the speaker of the micro:bit makes interesting sounds and the LED dot matrix displays the music logo pattern.

2.57 Logo Touch (V2 only)

The gold logo is a touch sensor that works a bit like a touch screen on a mobile phone, measuring tiny changes in electricity. This is also called a capacitive touch sensor, because it uses measurements of electrical capacitance to work.

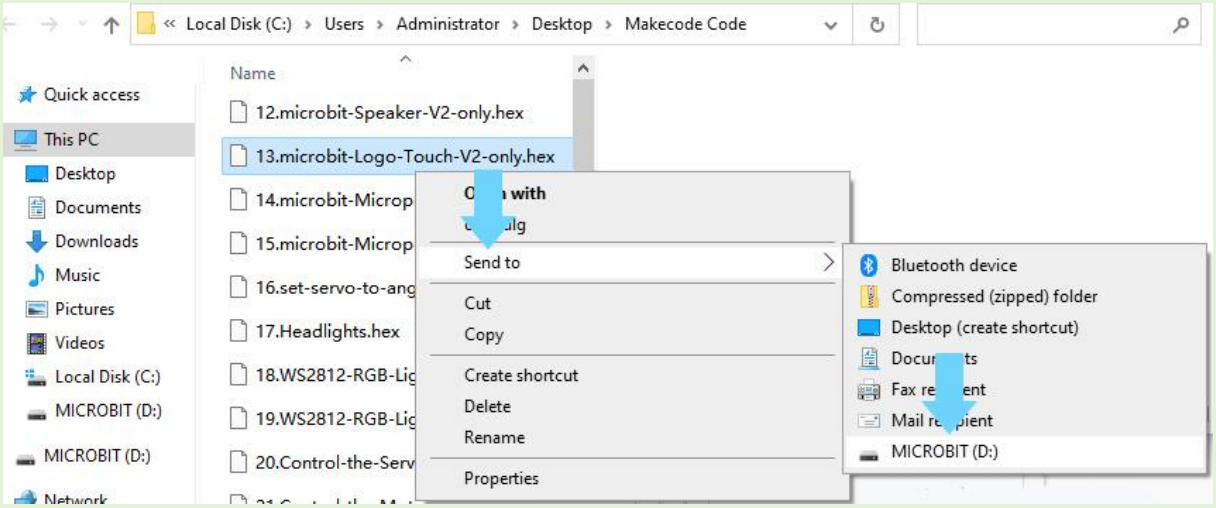
Light up your micro:bit with a heart - but only while you touch it!

Steps:

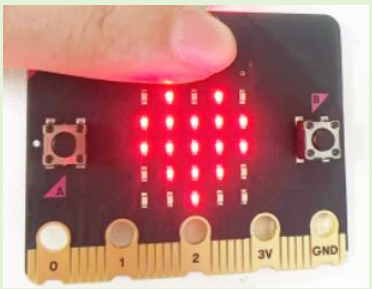
- Connect your micro:bit to your computer with a USB cable.
- Find the downloaded microbit-Logo-Touch-V2-only. hex file, drag it in the Makecode to view and analyze, and use WebUSB to flash it to the micro:bit from inside editor.

Or you can right-click on it and choose "Send to→MICROBIT."

File Type	Path	File Name
hex file	.../Makecode/Makecode Code	13.microbit-Logo-Touch-V2-only



Result: Show a '❤' icon on the LEDs while the logo is pressed. When you release your finger from the "Logo" mark, number will appear.



2.58 Microphone (V2 only)

The new micro:bit with sound has a built-in microphone sensor. It can react to loud and quiet sounds, and also measure how loud your environment is.

The microphone is on the back of the new micro:bit, and on the front you'll find a new microphone LED next to the hole that lets the sound in. It lights up to show you when your micro:bit is measuring sound levels.

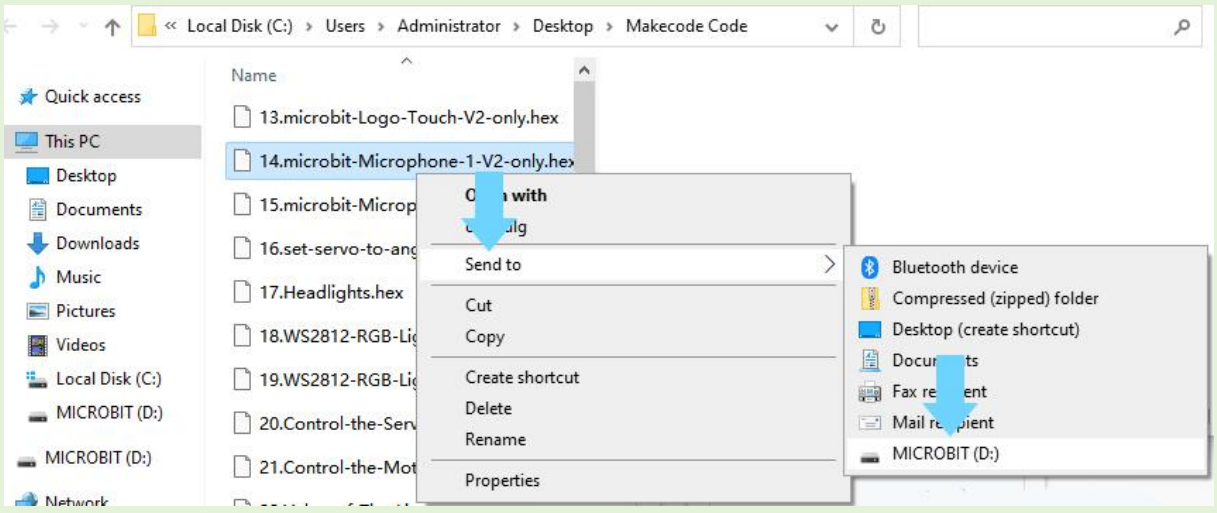
1 We will make the LED matrix of Microbit display a large heart when the microphone detects sound, and a small heart when the surroundings are quiet.

Steps:

- Connect your micro:bit to your computer with a USB cable.
- Find the downloaded microbit-Microphone-1-V2-only.hex file, drag it in the Makecode to view and analyze, and use WebUSB to flash it to the micro:bit from inside editor.

Or you can right-click on it and choose "Send to→MICROBIT."

File Type	Path	File Name
hex file	.../Makecode/Makecode Code	14.microbit-Microphone-1-V2-only



- Result: After the code is downloaded, a small heart will be displayed in the center of the LED matrix of the Microbit. You can try to clap your hands near the Microbit. Each time you clap, the small heart on the LED matrix will turn into a big heart.

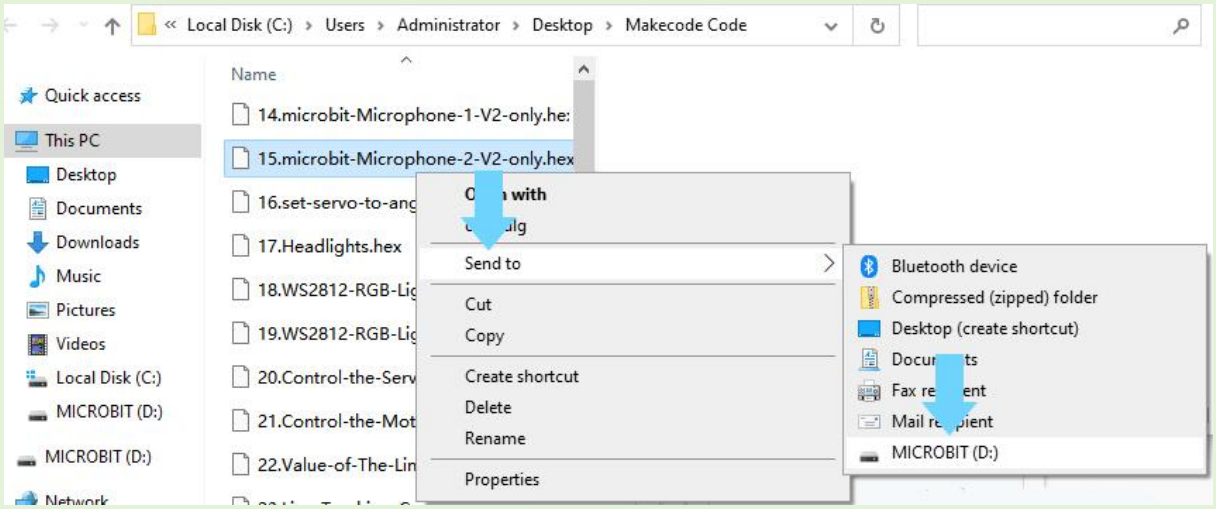
2 Next we will download the new code to the Micro Bit and display the volume of the sound on the LED matrix.

Steps:

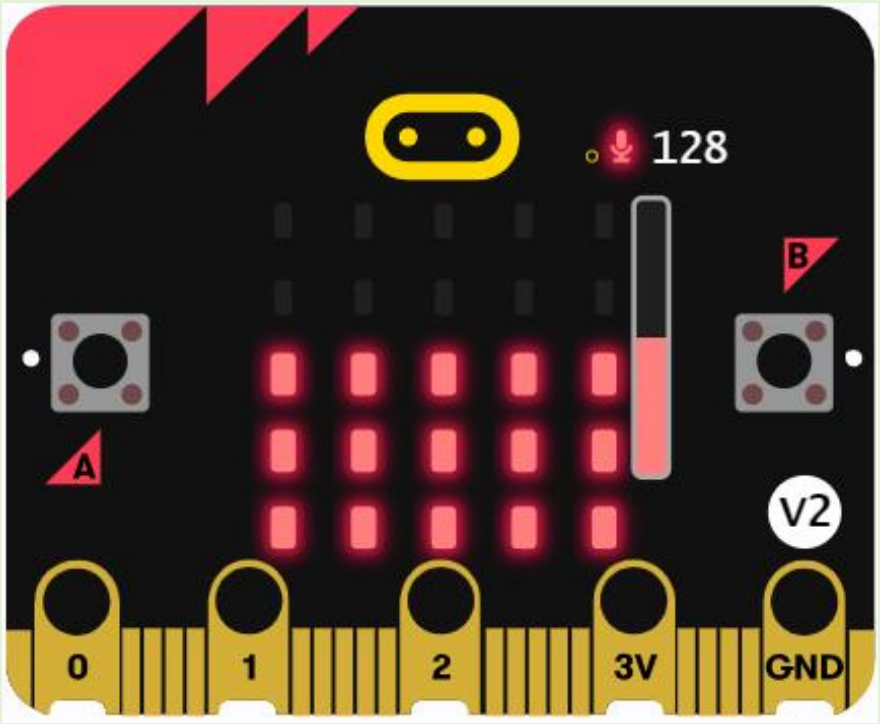
- Connect your micro:bit to your computer with a USB cable.
- Find the downloaded microbit-Microphone-2-V2-only.hex file, drag it in the Makecode to view and analyze, and use WebUSB to flash it to the micro:bit from inside editor.

Or you can right-click on it and choose "Send to→MICROBIT."

File Type	Path	File Name
hex file	.../Makecode/Makecode Code	15.microbit-Microphone-2-V2-only



- Result: After the code is downloaded, You'll notice that the louder the sound around the Microbit, the more LEDs will light up, from bottom to top.



3. Introduction to 4WD Mecanum Wheel Car

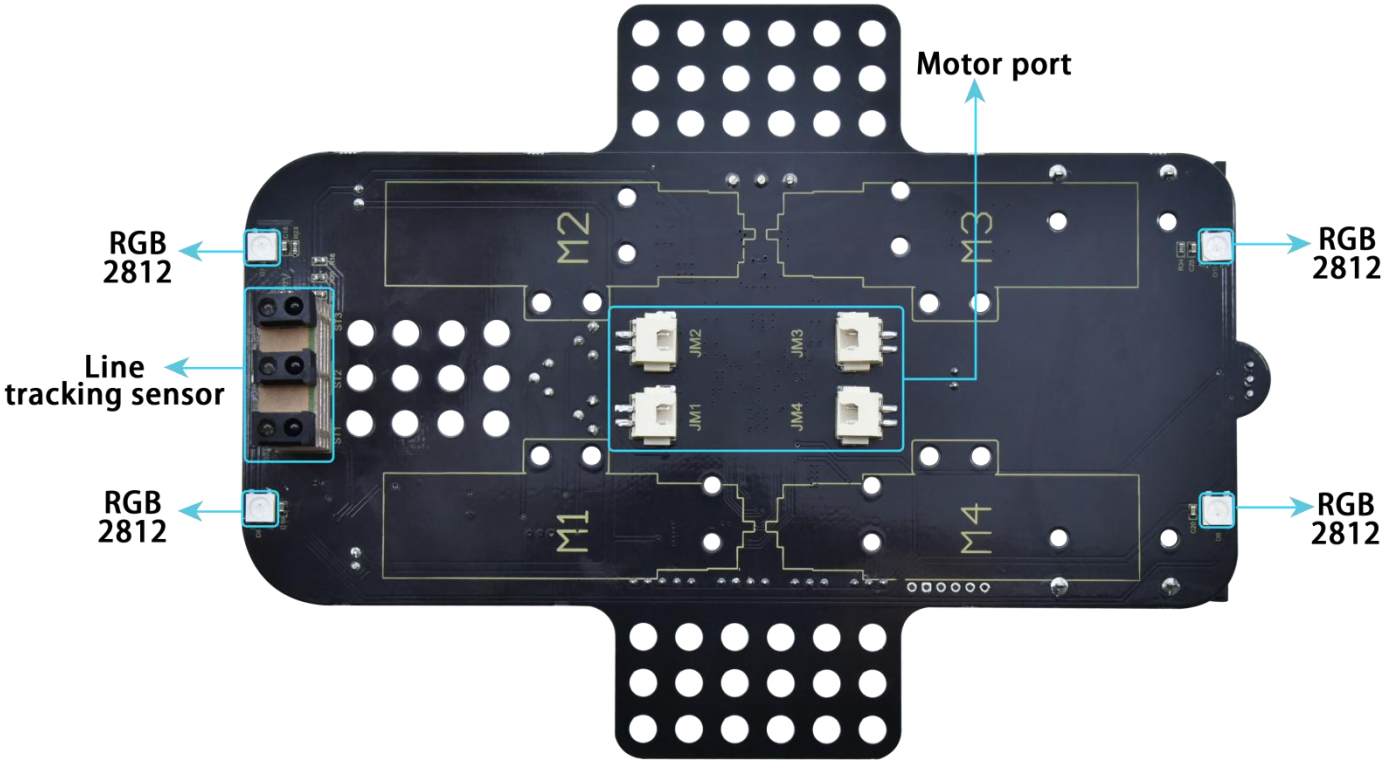
Keyestudio Microbit 4WD Mecanum Robot Car V2.0 is a multifunctional cart based on BBC micro:bit. It is equipped with a wealth of sensors and peripherals to help you understand how to use the micro:bit and learn more about electronics.

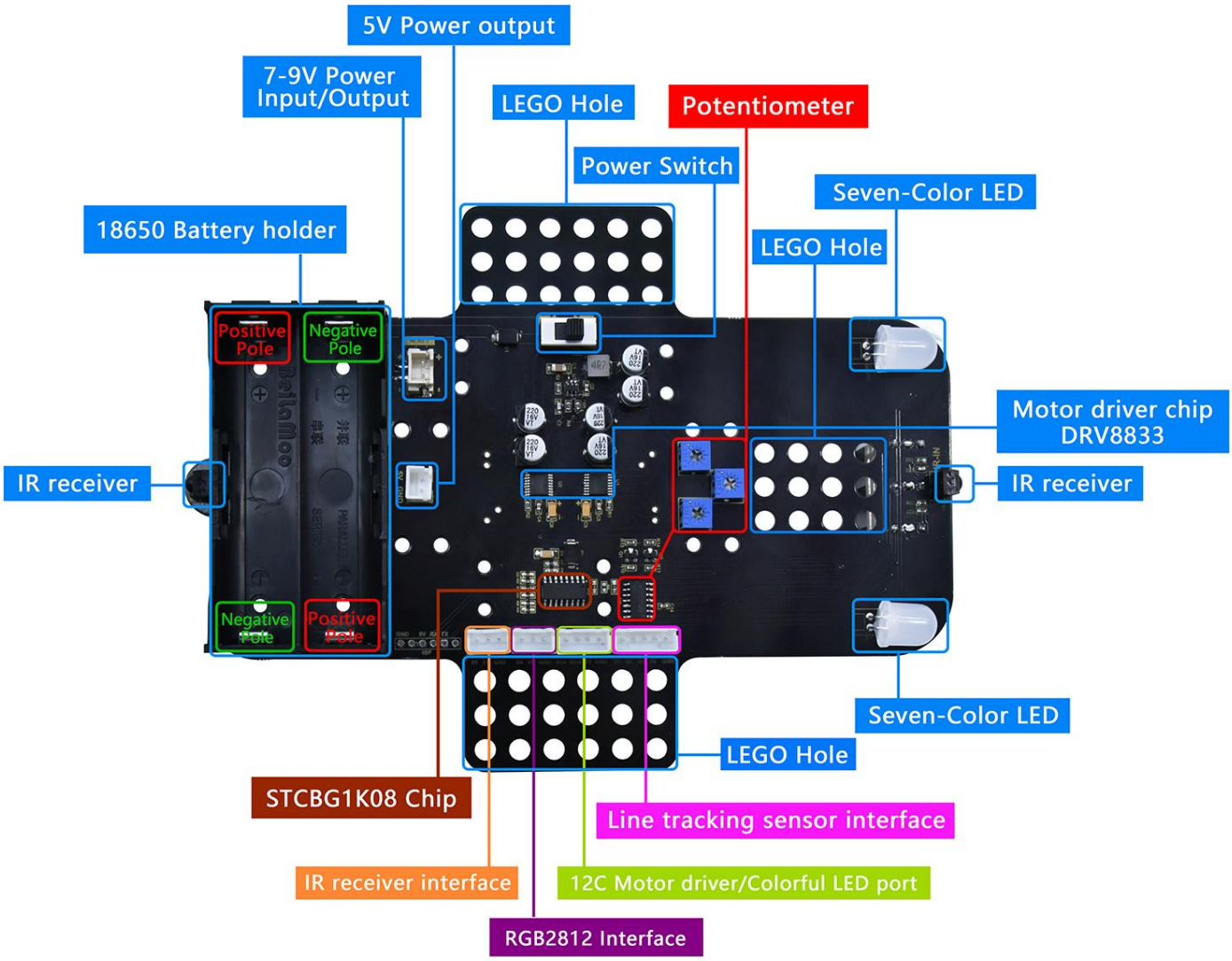
In addition to this it leaves lot of universal jacks of building block for easy connection to other peripheral devices, you can use your experience and imagination to create more interesting inventions.

3.1 Hardware

Electronic Parts	colorful headlight	DC motor	servo	ultrasonic sensor	Line-tracking Sensor	infrared receiver	WS2812 RGB light	battery case
QTY	2	4	1	1	3	2	4	1

Note: the battery case, two colored headlights, four WS2812 RGB lights and two IR receivers are already integrated in the car baseplate.





Connection between the Mecanum Robot and micro:bit GPIO:

GPIO of Micro:bit	4WD Mecanum Robot
P3 P4 P10	line tracking sensor
P14	servo
P0	IR receiver
P15 P16	ultrasonic sensor
P19 P20 (IIC)	4 motors
P19 P20 (IIC)	2 seven-color lights

3.2 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℃
- 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.

3.3 About Battery

The Keyestudio Microbit 4WD Mecanum Robot Car V2.0 is powered by two 18650 battery. This product does not contain 18650 battery. You will need to prepare yourself two 18650 batteries as well as a battery charger.

The following parameters are available for your purchase：

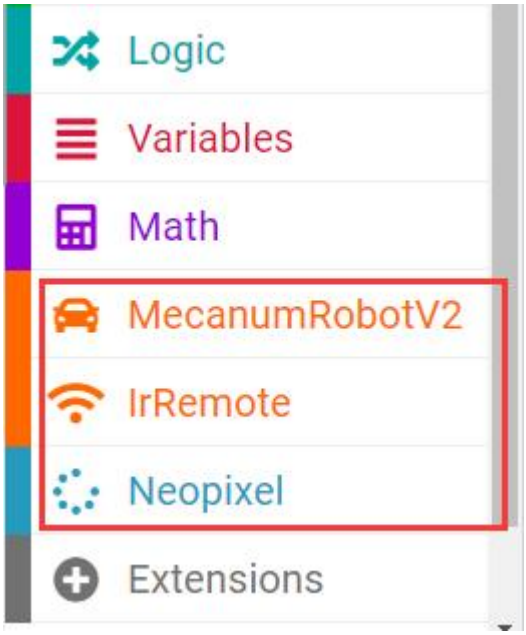
Specifications	
Size	18650
Positive Terminal：	Flat Top or with a top
Capacity	1500-3000mAh
Nominal Voltage	3.7V
Maximum Voltage	4.2V
Discharge cut-off Voltage	2.5V
Rechargeable	Yes
Approx. Dimensions	18.5mm x 65.2mm

3.4 Adding Extension for the Robot Project

We have created a dedicated extension to simplify coding tasks for the mecanum robot.

Extensions are functional code modules that are installed from outside the MakeCode editor and plug new blocks into the Toolbox. If you have used Arduino before, you probably know about a thing called a library; which is a collection of code that extends the functionality of the core programming language. MakeCode extensions work the same way.

Drag the hex files we provide into the Makecode editor or use the "Import" button to open the hex files. You may see three additional extensions in the Toolbox list, which are already included in our code.



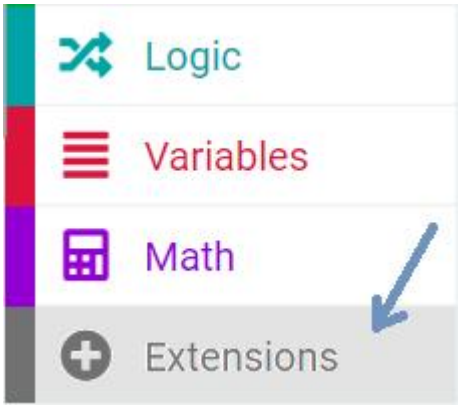
That is to say, when you use the hex file we provide, you do not need to add these extensions.

Note:

1. If you want to Create a New Project to control this robot, you will need to add the extension we provide to use new blocks to complete the code.
2. For every new MakeCode project that you make, you will have to load extensions over again.

How to add an extension

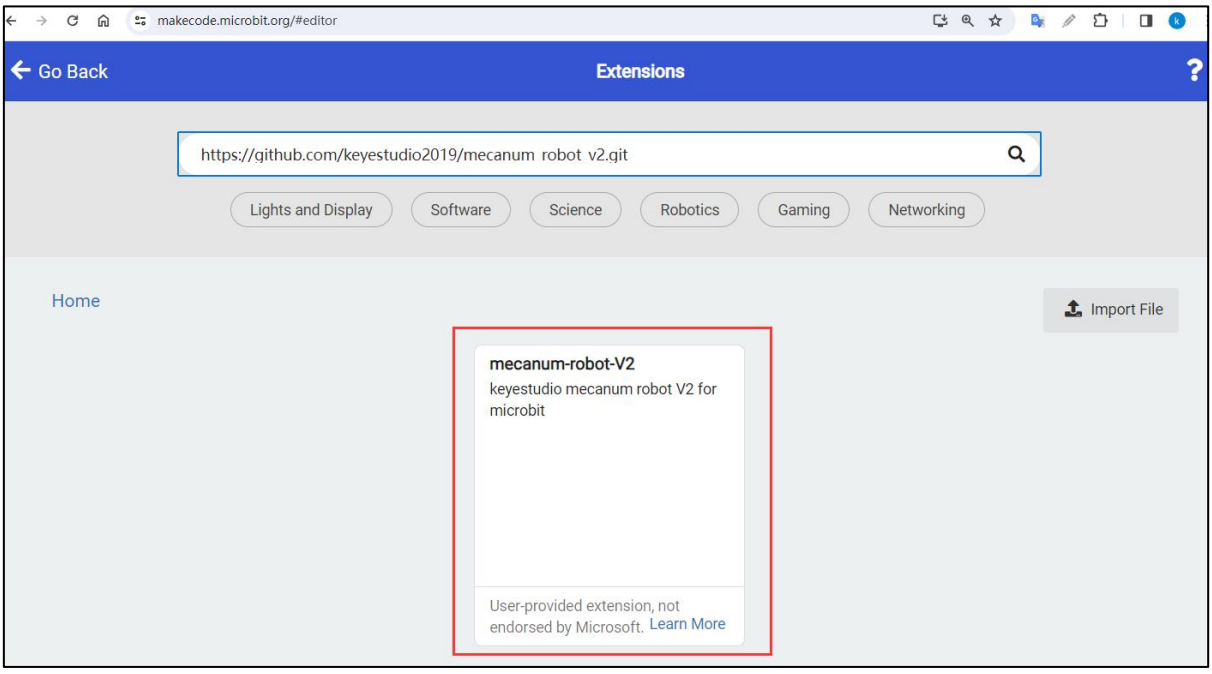
You can add an extension by going to Toolbox and clicking on the Extensions category.



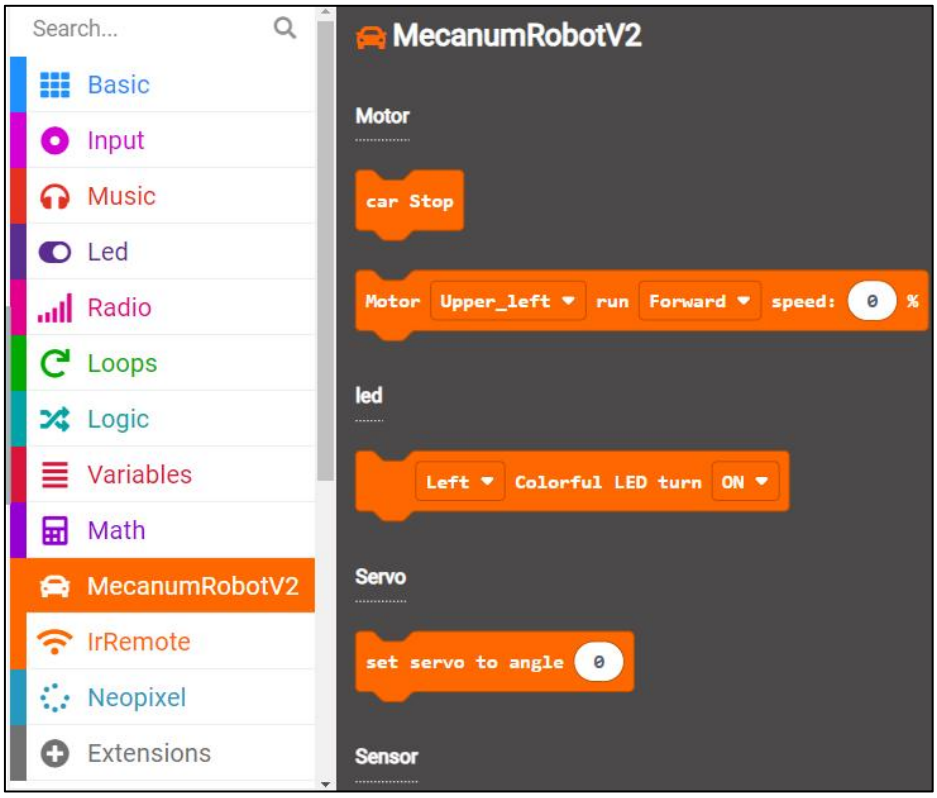
This will open a window giving you a place to search for extensions. Also, a selection of recommended extensions is shown for you to choose from. Copy and paste the following link into the search box and press the "**Enter**" to search.

https://github.com/keyestudio2019/mecanum_robot_v2.git

Click the **“mecanum-robot-v2”** extension.

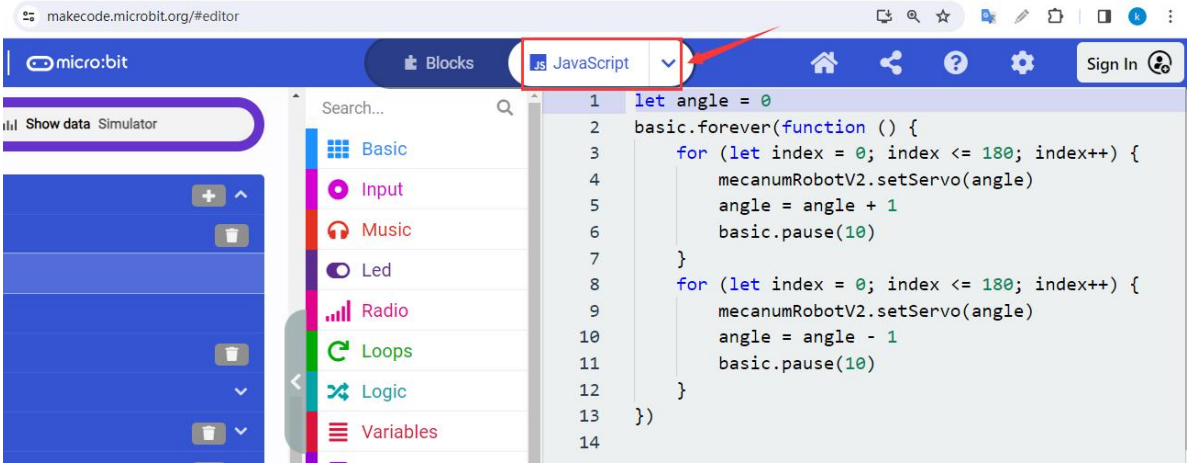


After the extension is added, you should see the **MecanumRobotV2, IrRemote** and **Neopixel** extension category appear in the Toolbox. The extension will contain the new blocks, ready for you to use in the project's code.

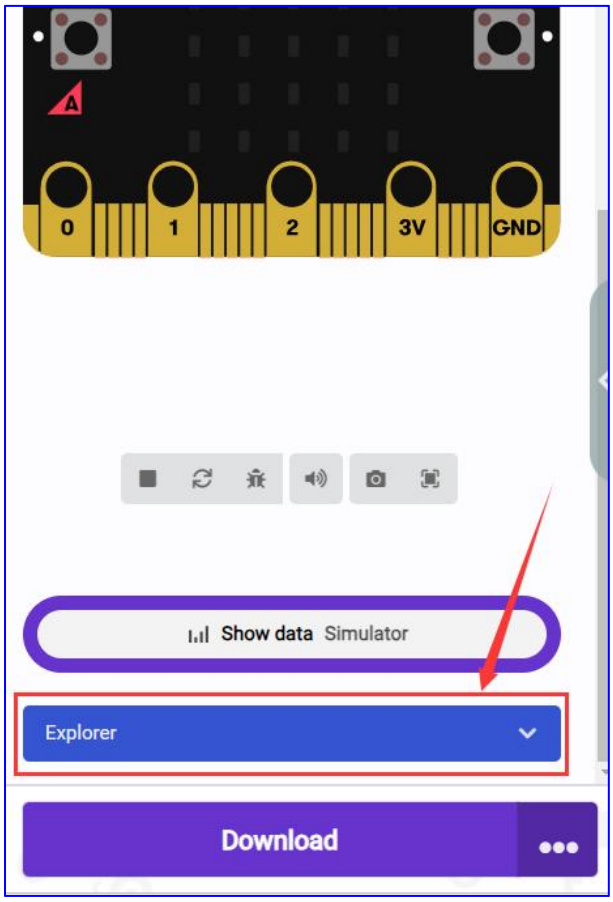


How to Update or Delete an Extension

1.Click the "JavaScript" button to switch to text code.



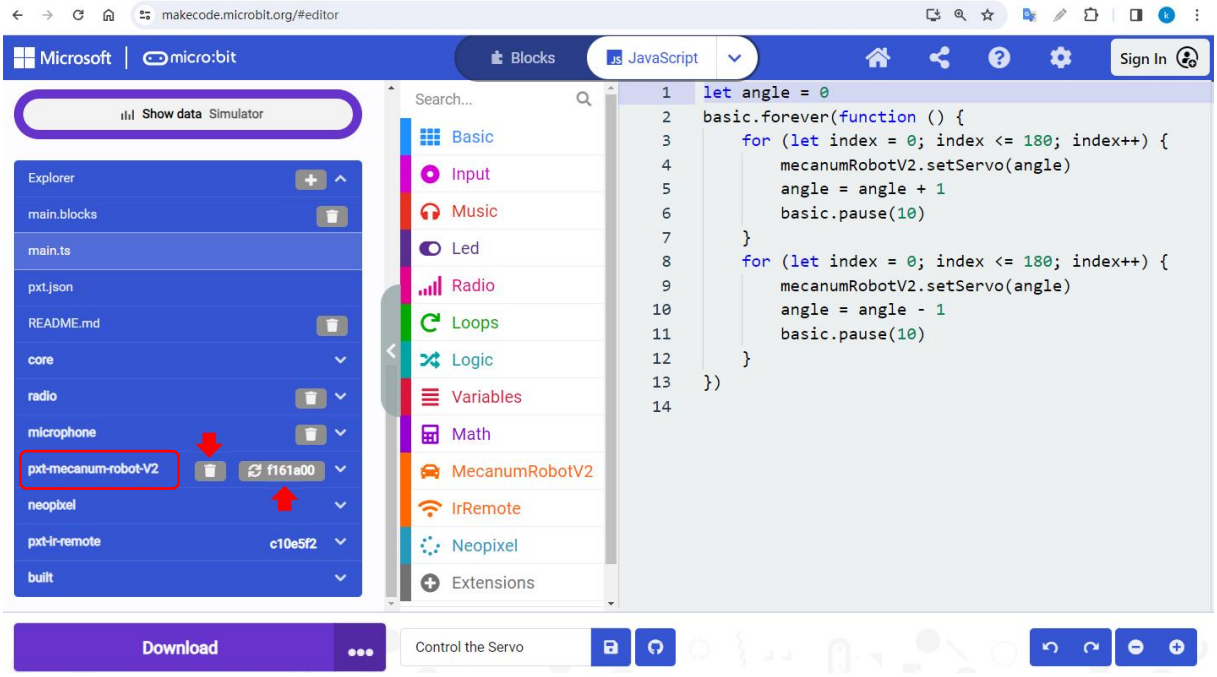
2.Find the Explorer button below the simulator



3.Find pxt-mecanum-robot-V2 in the extended list.

Click on the trash can icon to delete the extension.

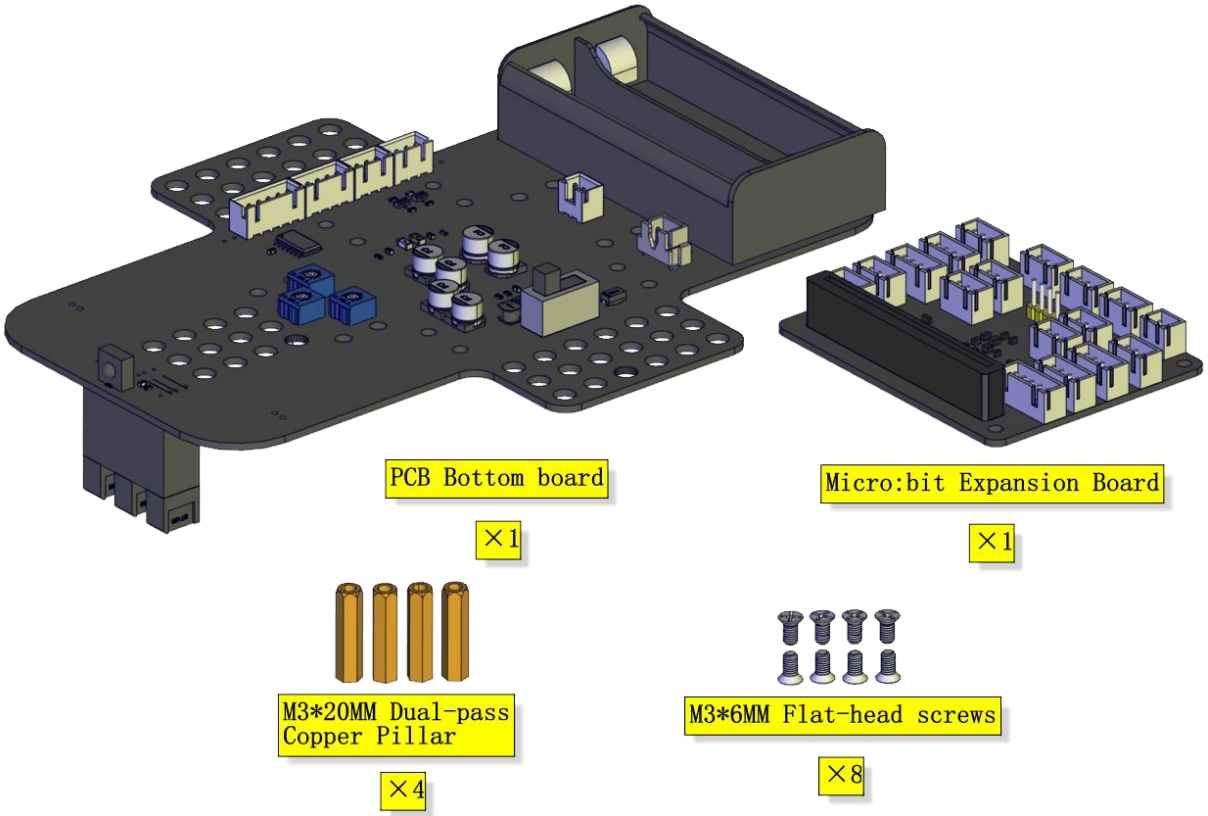
Click on the refresh icon to update Rthe extension.

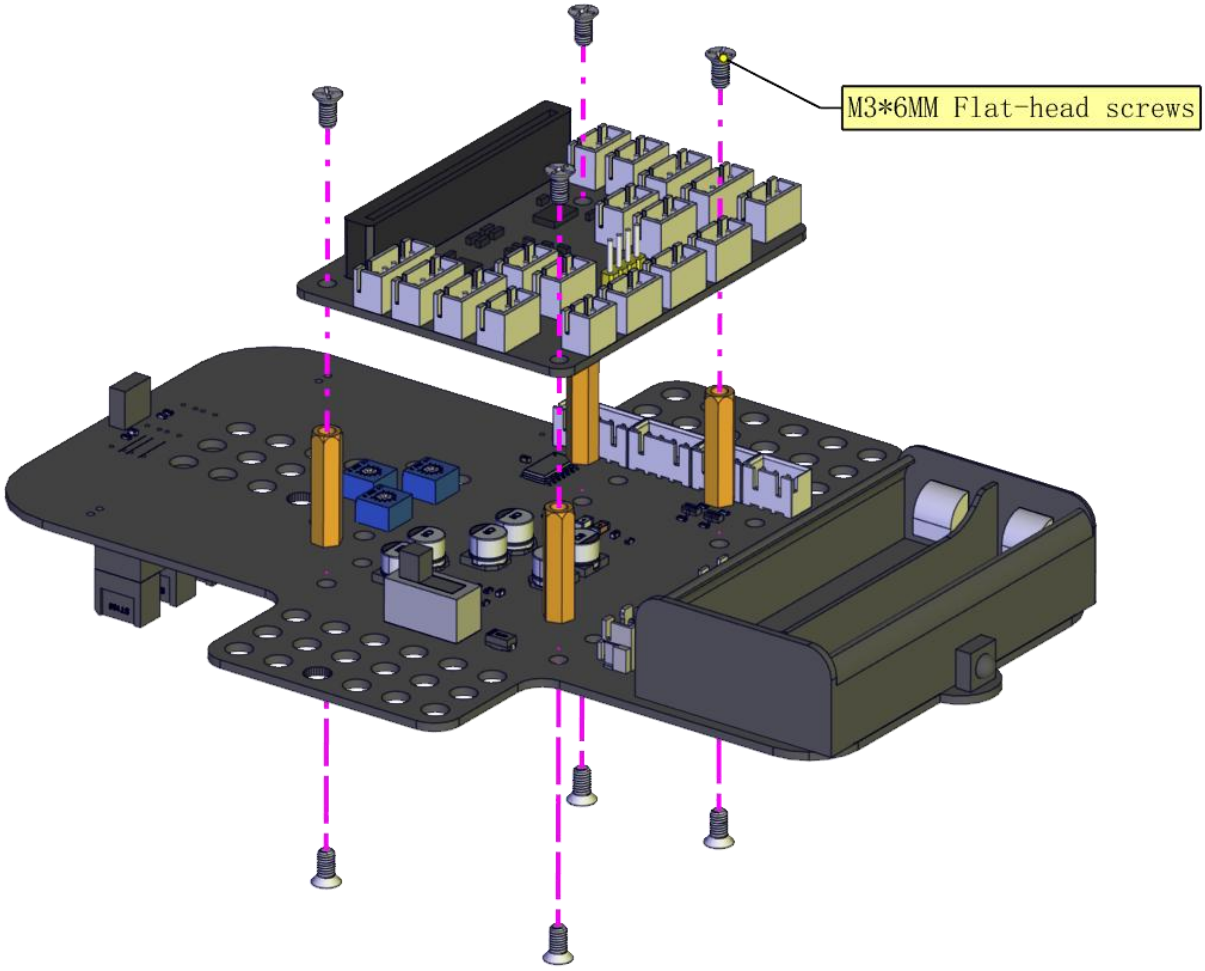


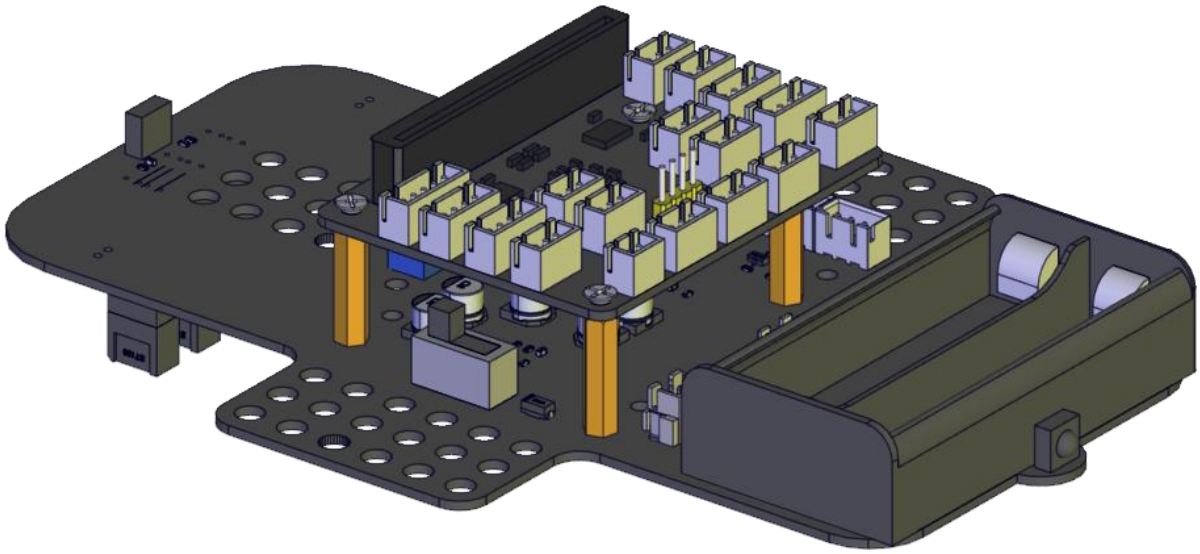
3.5 Assemble the Robot

Before assembly, please tear off the protective film on the acrylic boards.

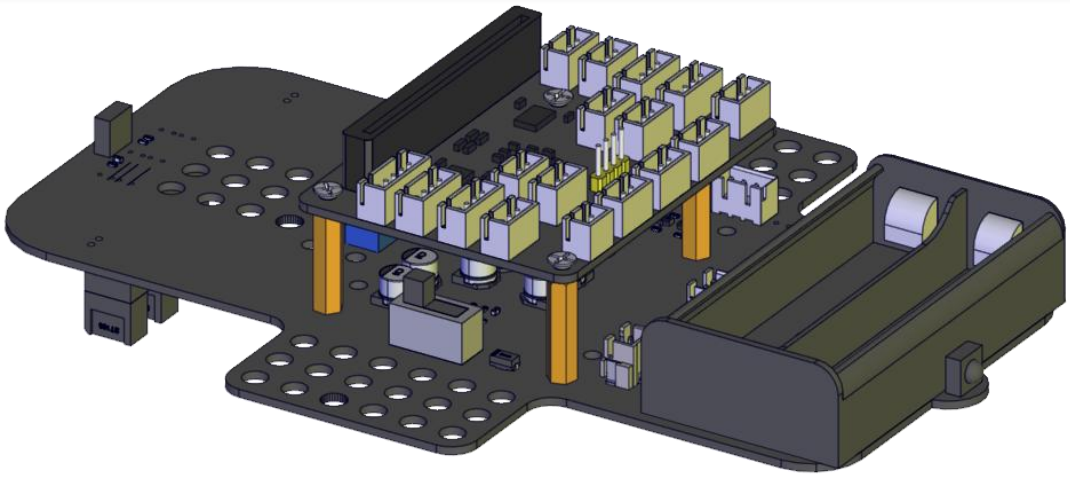
1.Assemble the car base board and microbit shield.

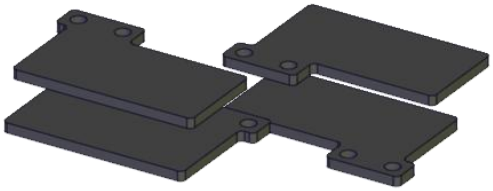



M3*6MM Flat-head screws

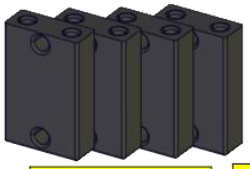


2.Install the motor and wheels





Acrylic Boards

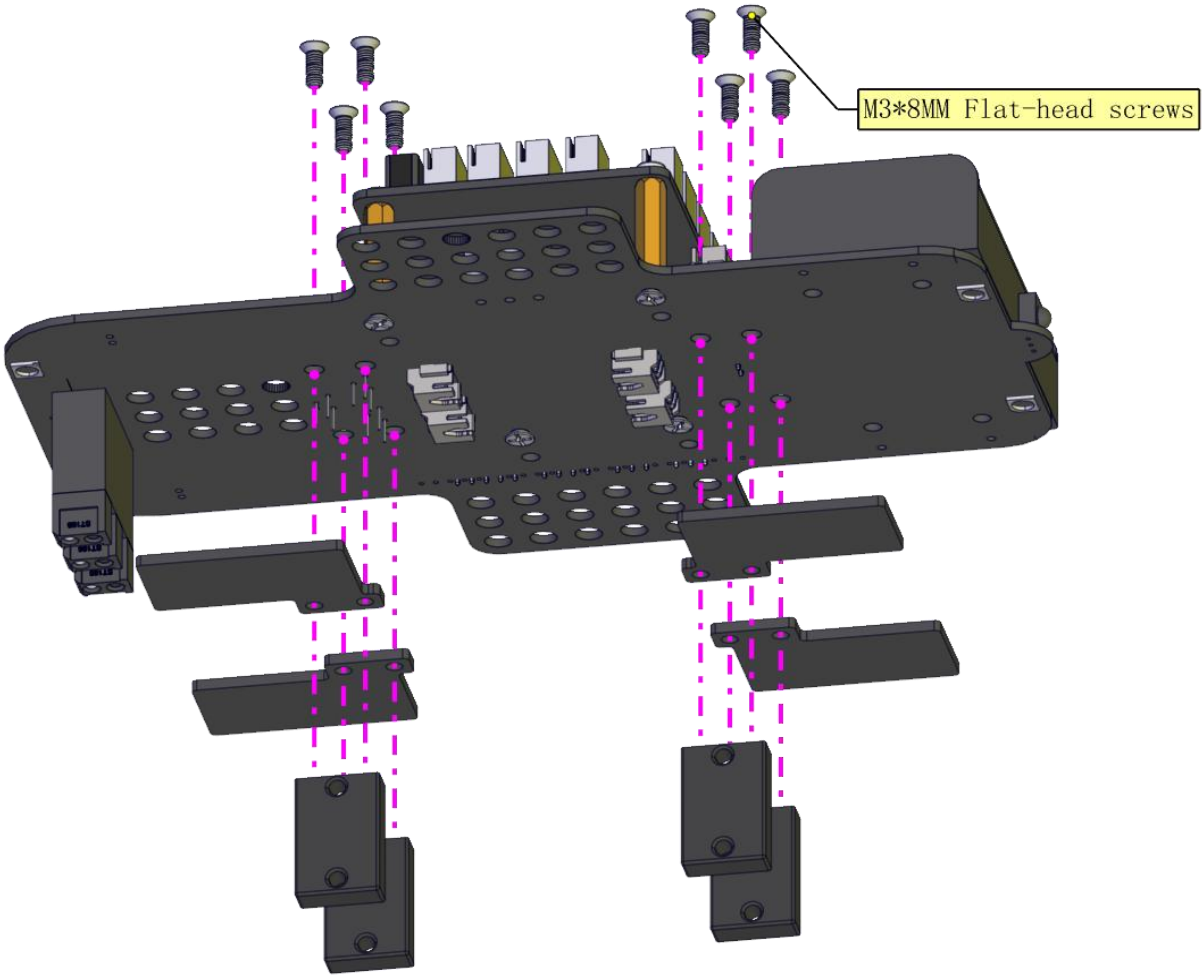


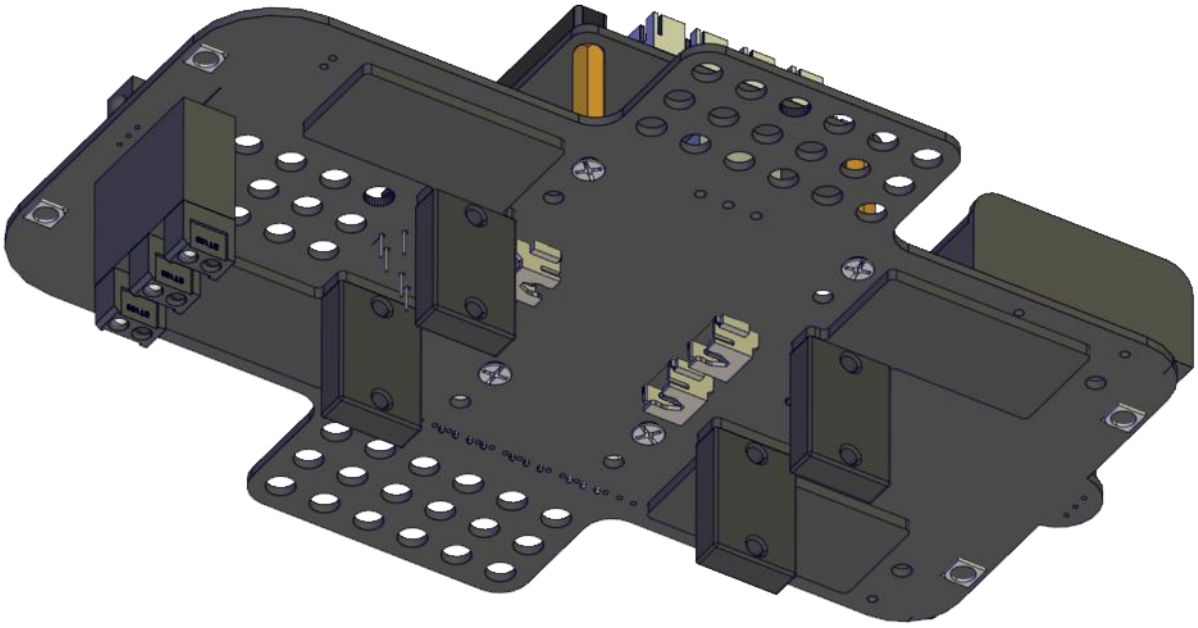
Fixed parts

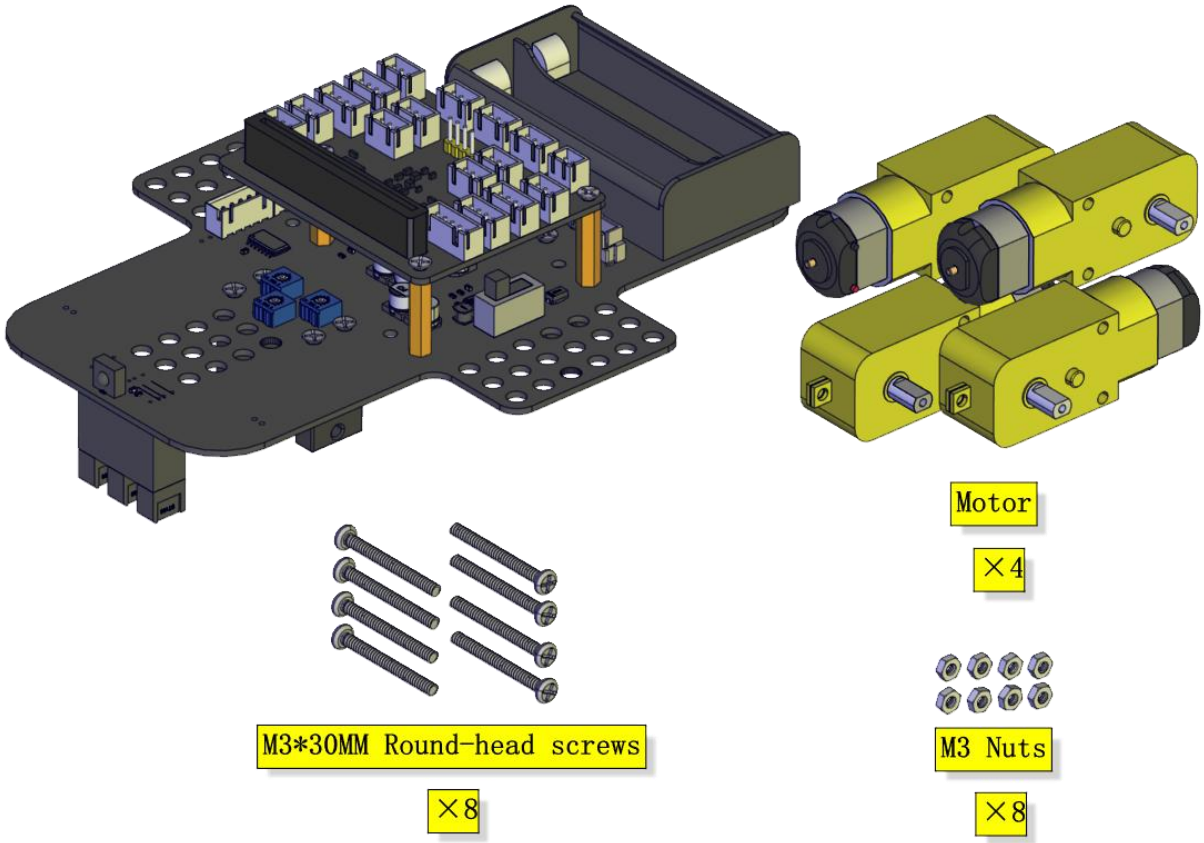
×4

M3*8MM Flat-head screws

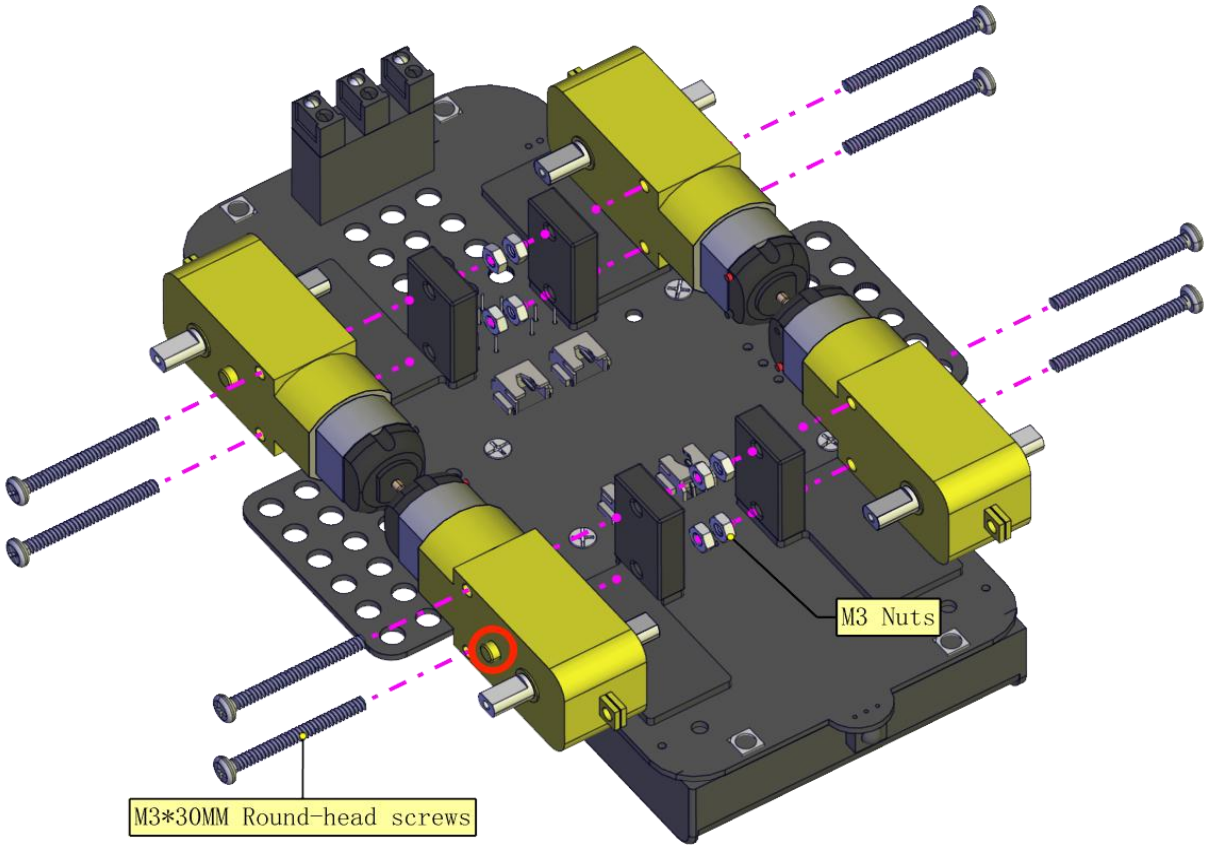
×8

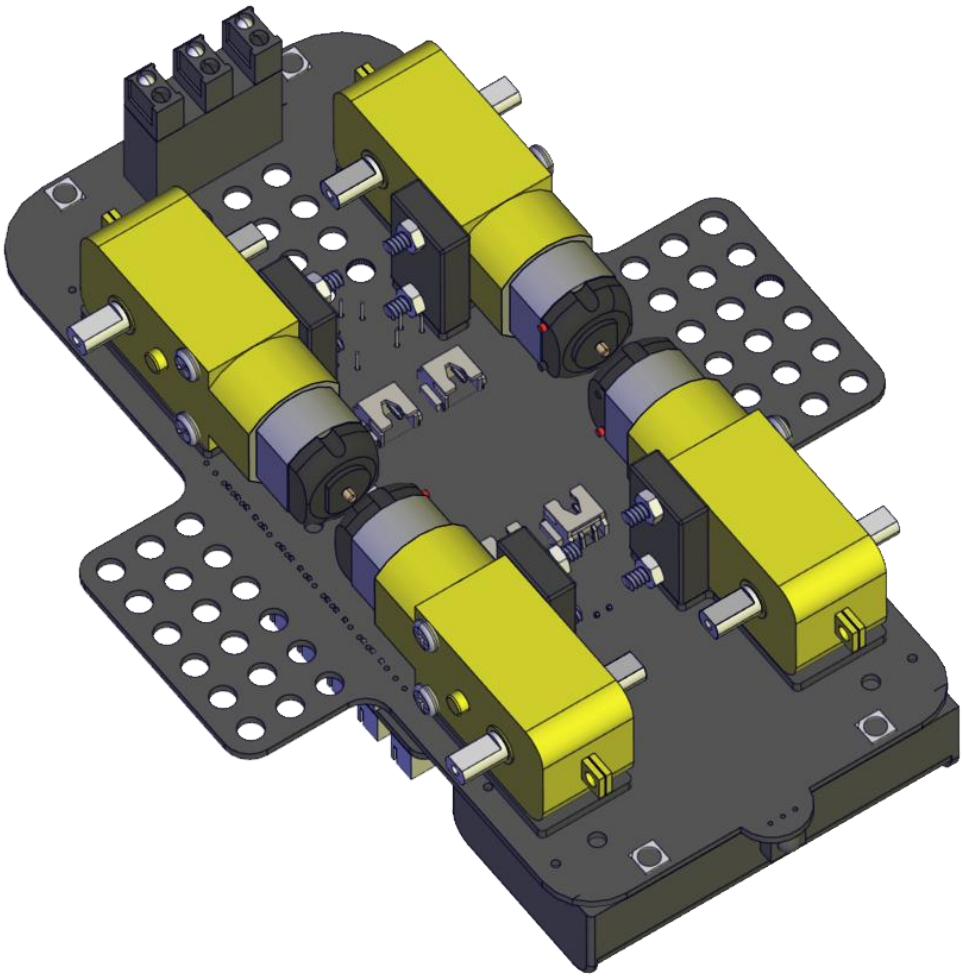


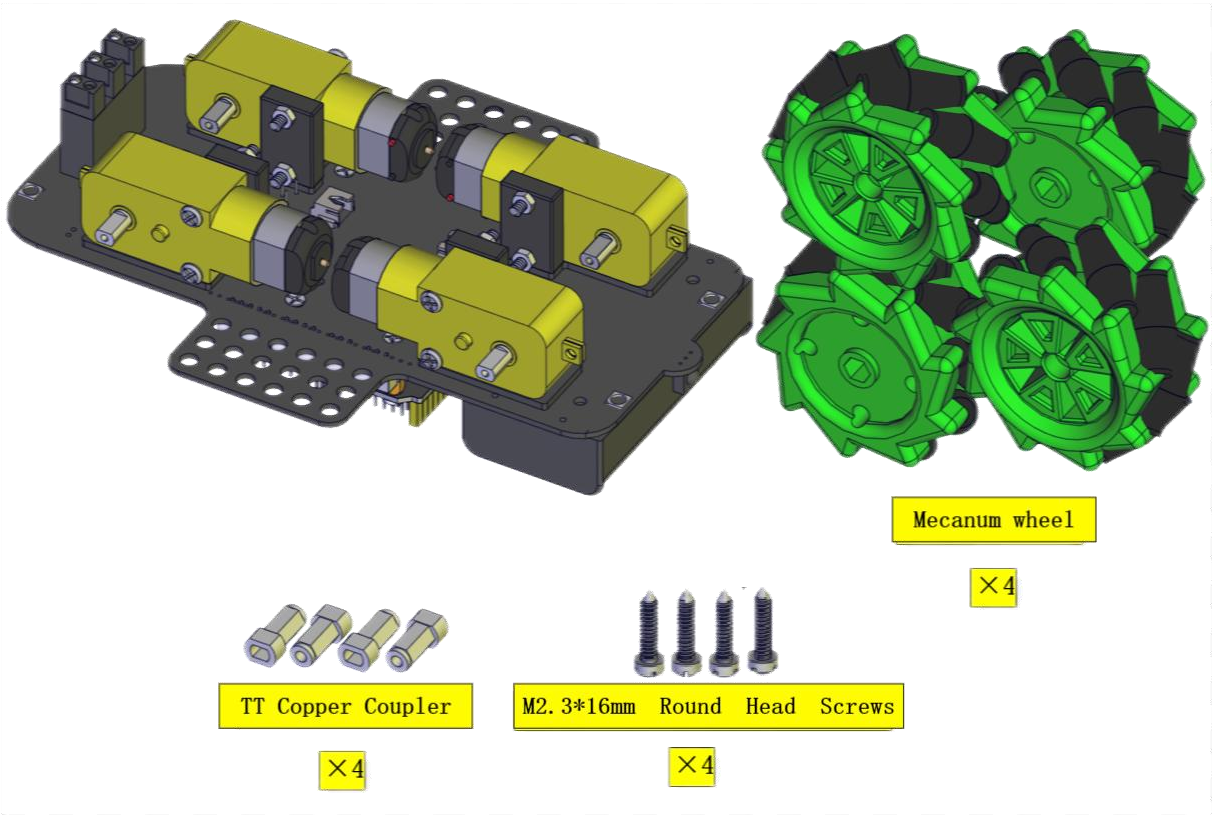



Motor
×4
M3*30MM Round-head screws
×8
M3 Nuts
×8

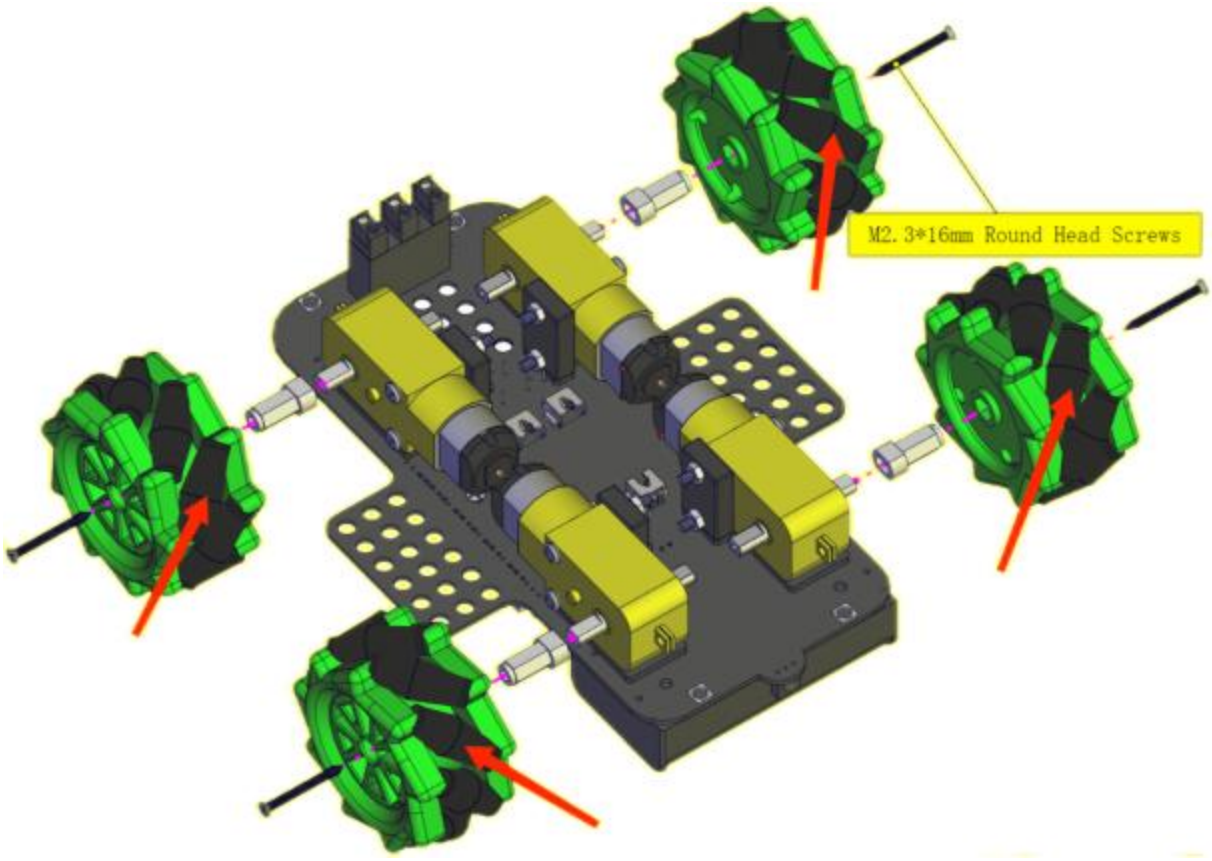
Note: When installing the motor, the wire of the motor are facing inward.

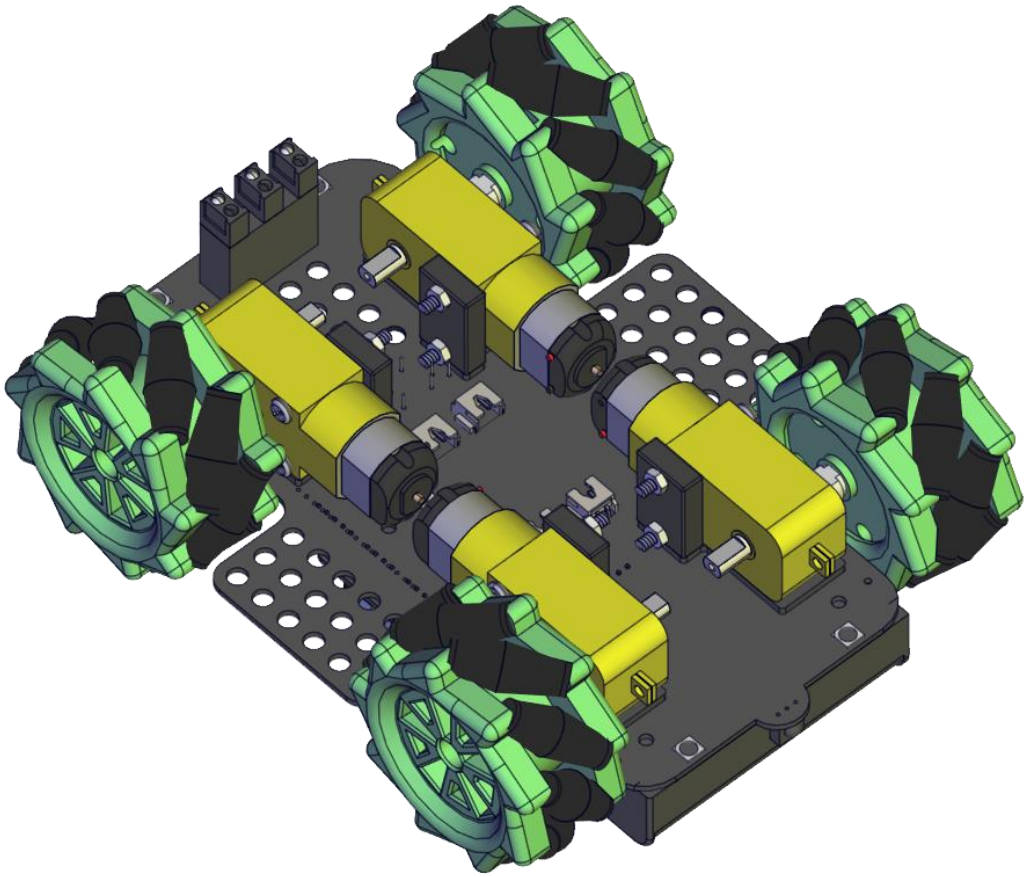






Pay attention to the direction of the small rubber roller on the circumference of the mecanum wheel.



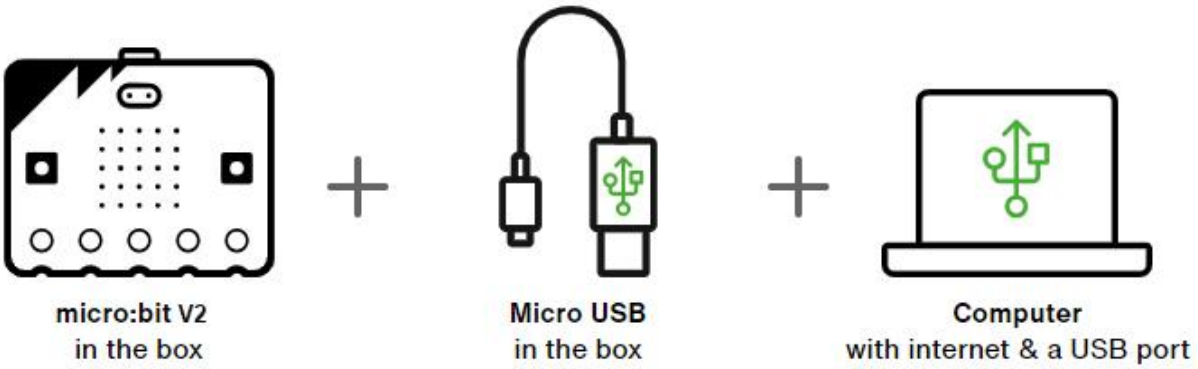


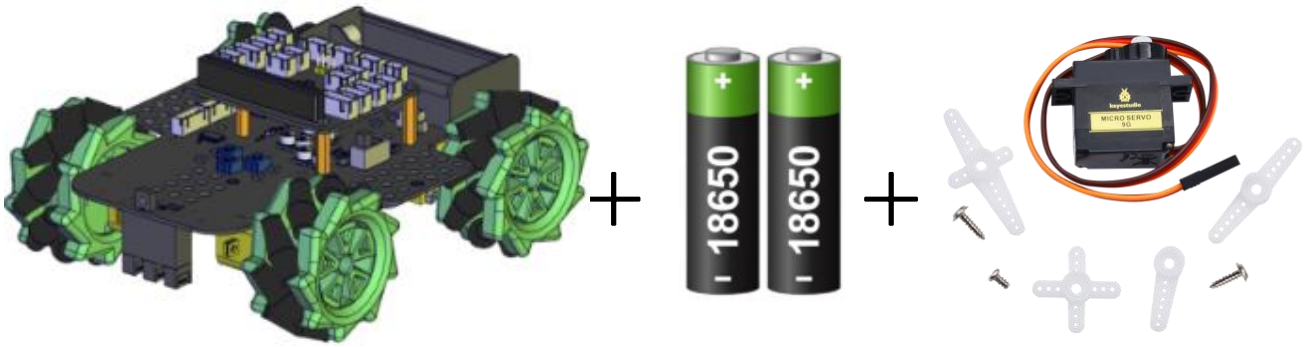
2.Install ultrasonic sensor and servo

Note: Adjust the Angle of the Servo Before Assembly.

Before assembling the servo, we need to adjust it to 90°, so that it can work with the ultrasonic module to make the robot work as intended.

You need to prepare:

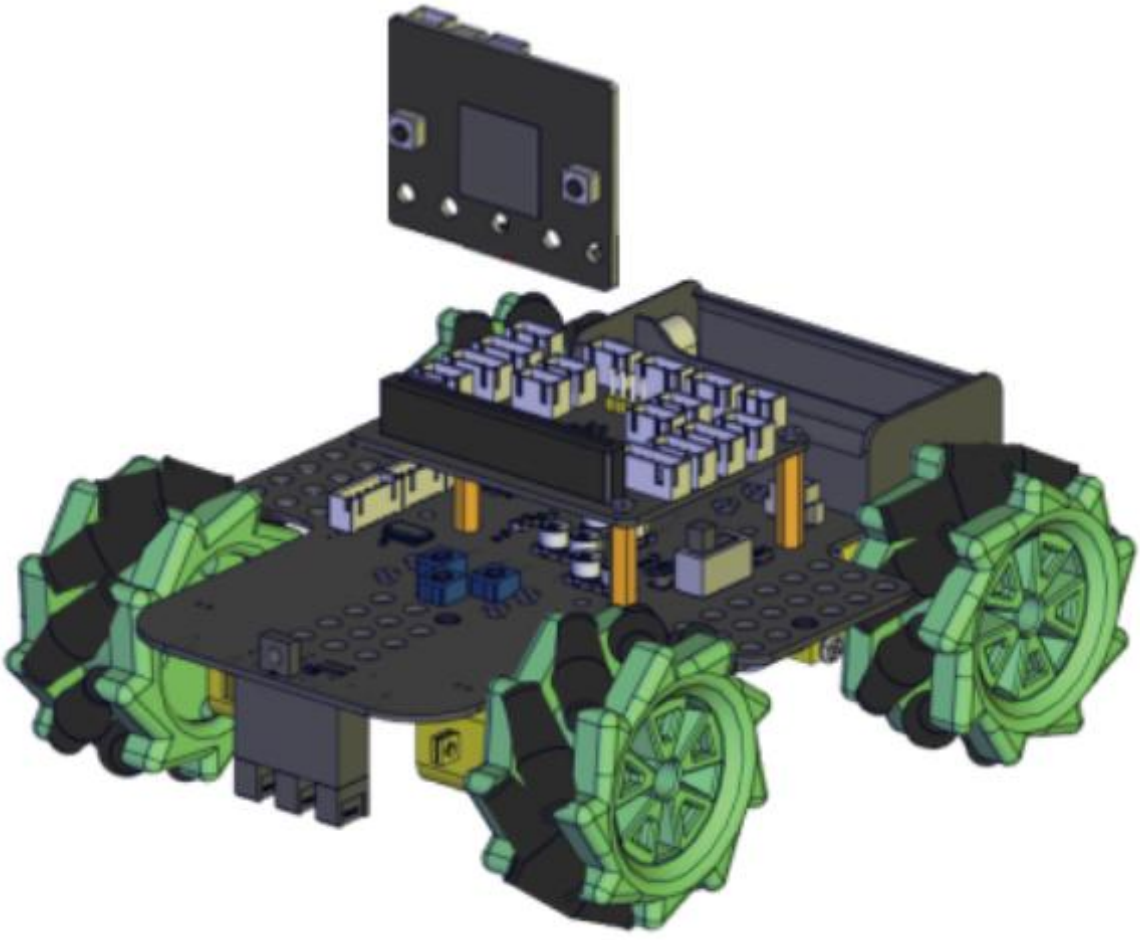




assembled car body two 18650 batteries servo

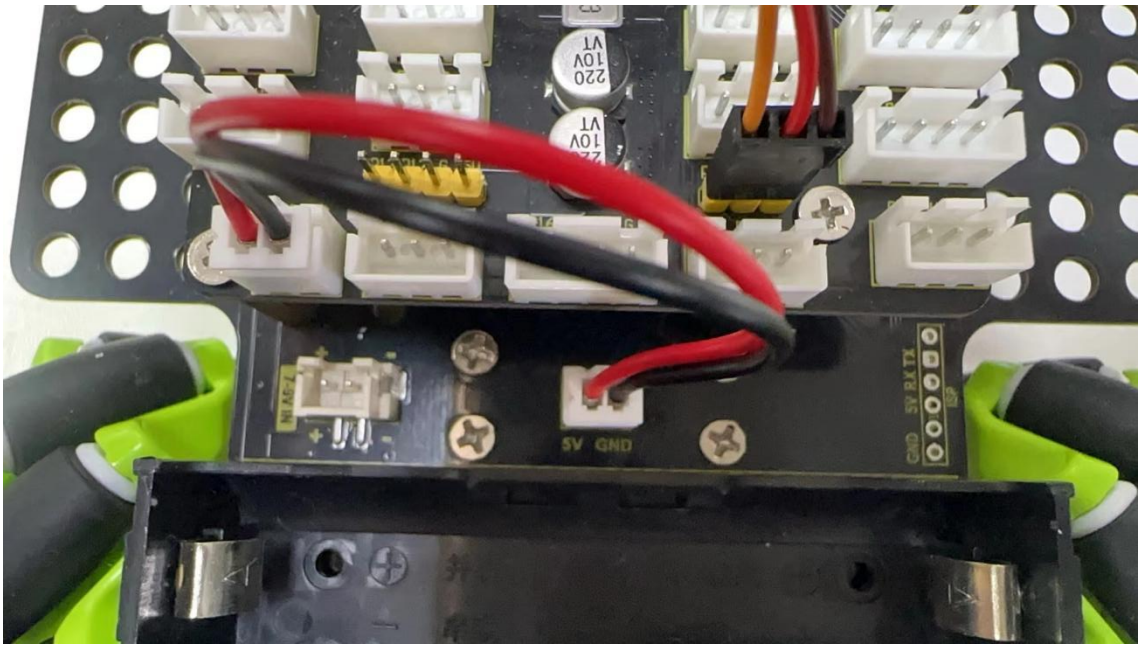
Steps:

1) Insert the microbit into the microbit shield of the car body as shown in the picture below. (Note: The LED matrix and the logo face the front).



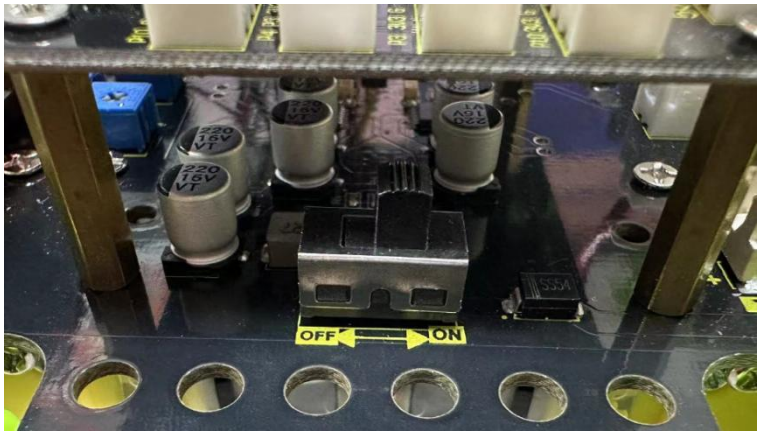
2)Use the HX-2.54 2P duPont wire 100mm to connect the battery case to the microbit shield. And connect servo to microbit shield.

Servo	Microbit shield
Brown wire	G
Red wire	V
Yellow wire	P14



3)Install the 18650 batteries into the battery case of the car body in the direction shown in the figure below.

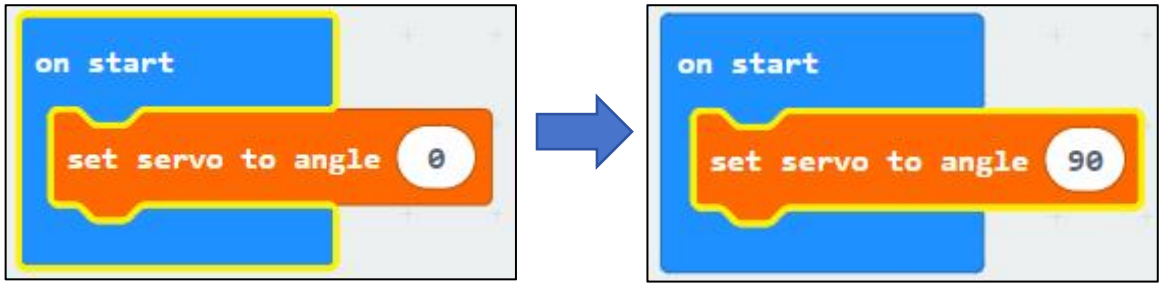
4)Turn on the power switch on the car body, push the button to "**ON**"



4)Set the servo to 90°

- 1.Connect the micro:bit to your computer with the USB cable.
- 2.Click **Create a Project** on the Makecode homepage, and name it "**set servo to angle 90**".
- 3.Adding **MecanumRobotV2** extension in Toolbox (Refer to the previous section to add it).
- 4.Click the MecanumRobotV2 extension and find the "**set servo angle to 0**" block.

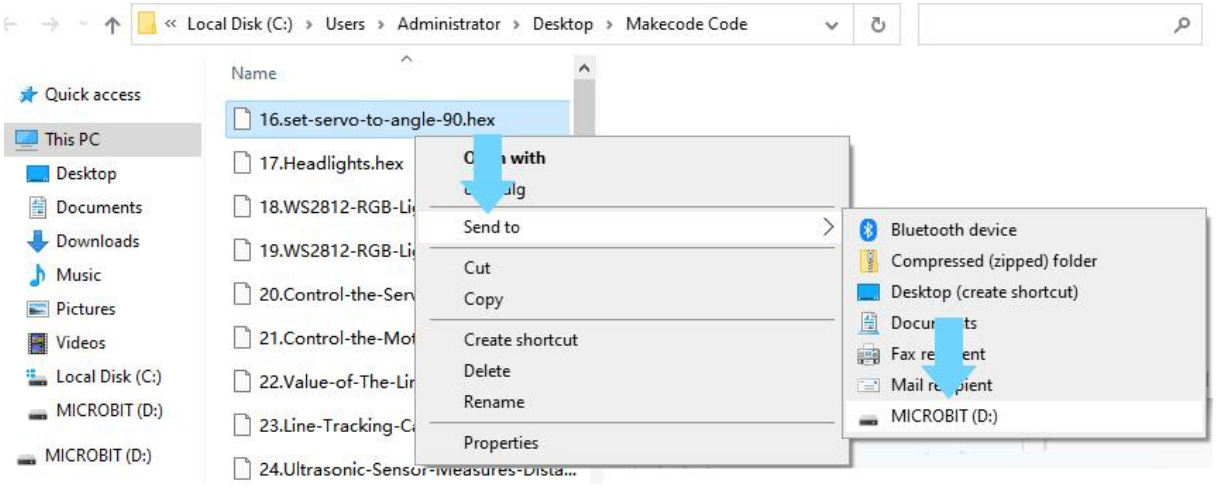
- 5.Drag the "**set servo angle to 0**" block into the edit box and splice it with the "**on start**" block
- 6.Change the angle 0 to **90**



- Pair the micro:bit with your computer using Webusb and download the code to it.
- After the code download is completed, the servo will be adjusted to 90°

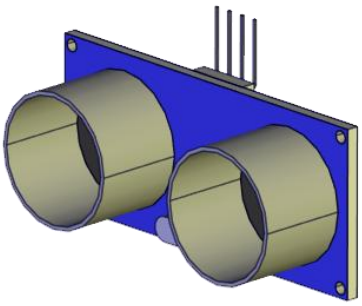
● Or you can find the downloaded set-servo-to-angle-90.hex file, drag it to the Microbit device or right-click on it and choose "**Send to → MICROBIT**."

File Type	Path	File Name
hex file	.../Makecode/Makecode Code	16.set-servo-to-angle-90



Let's continue with the assembly steps

Acrylic Boards



Ultrasonic Sensor

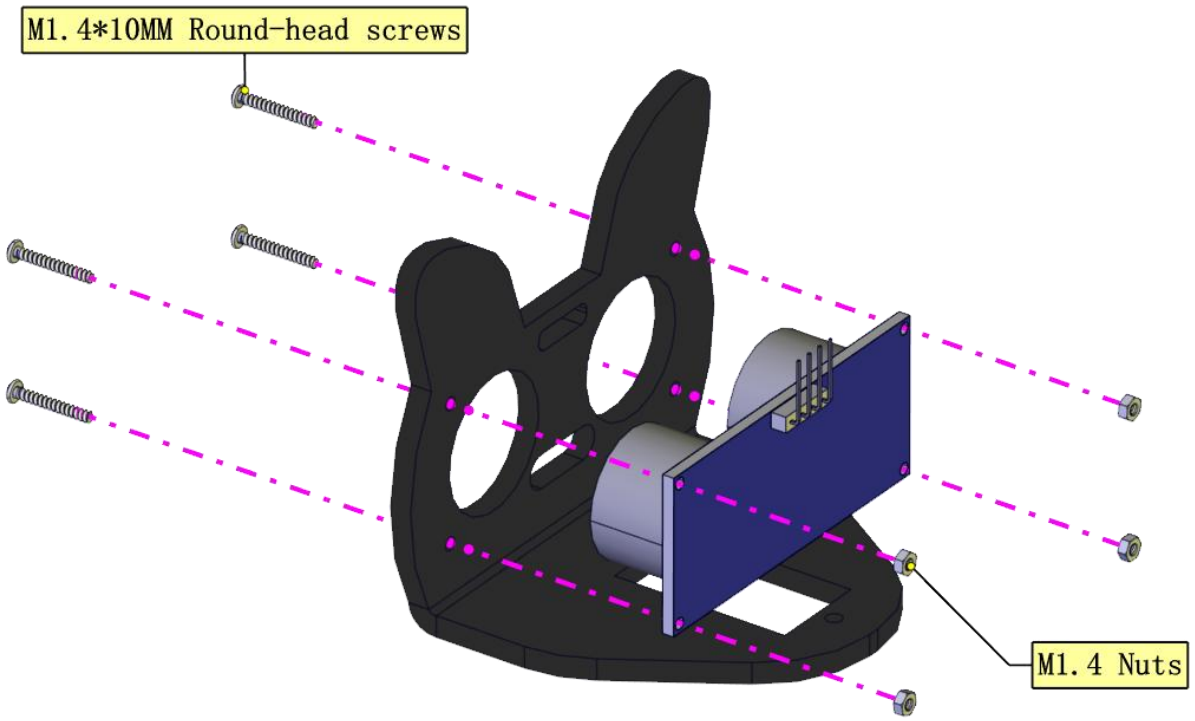
×1

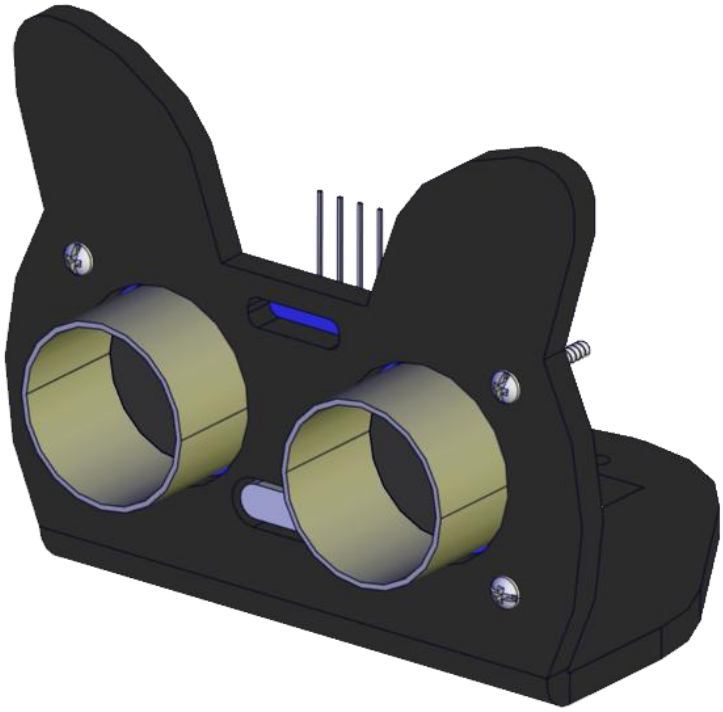
M1.4*10MM Round-head screws

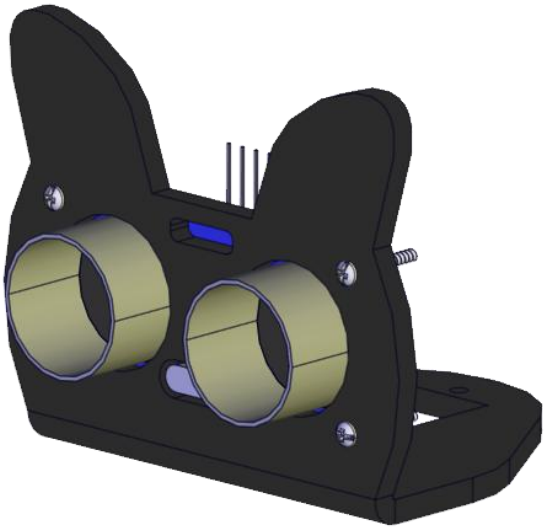
×4

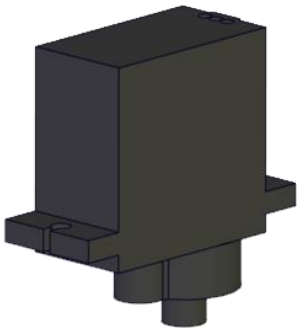
M1.4 Nuts

×4


M1.4*10MM Round-head screws
M1.4 Nuts







Keyestudio Servo

×1

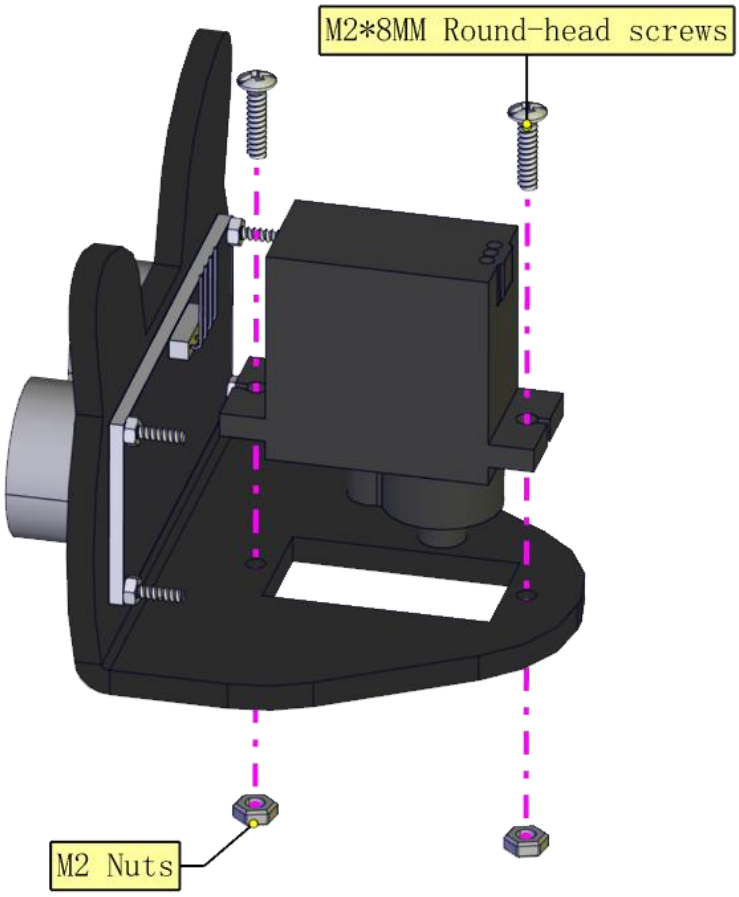
M2*8MM Round-head screws

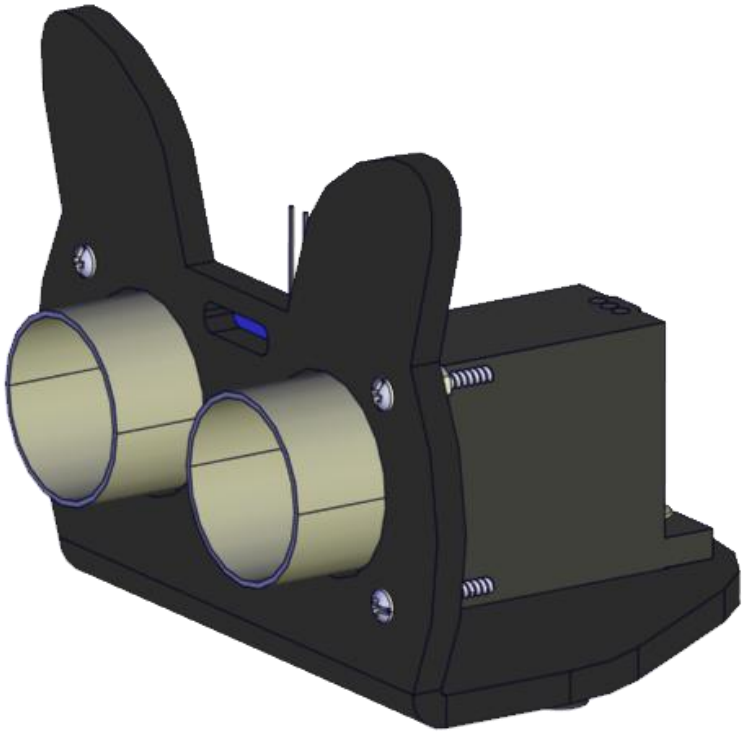
×2

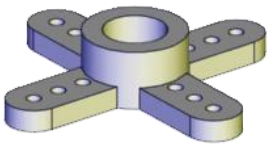
M2 Nuts

×2

Note: The servo needs to be adjusted to 90° before assembly

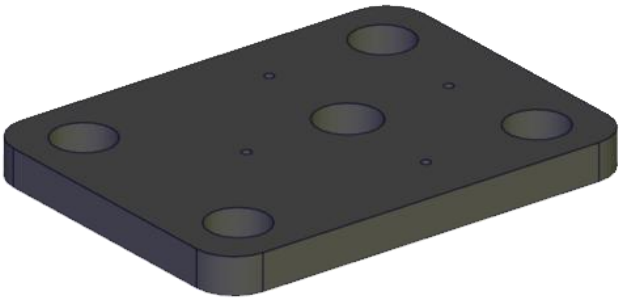






Control horn(belong to servo)

×1

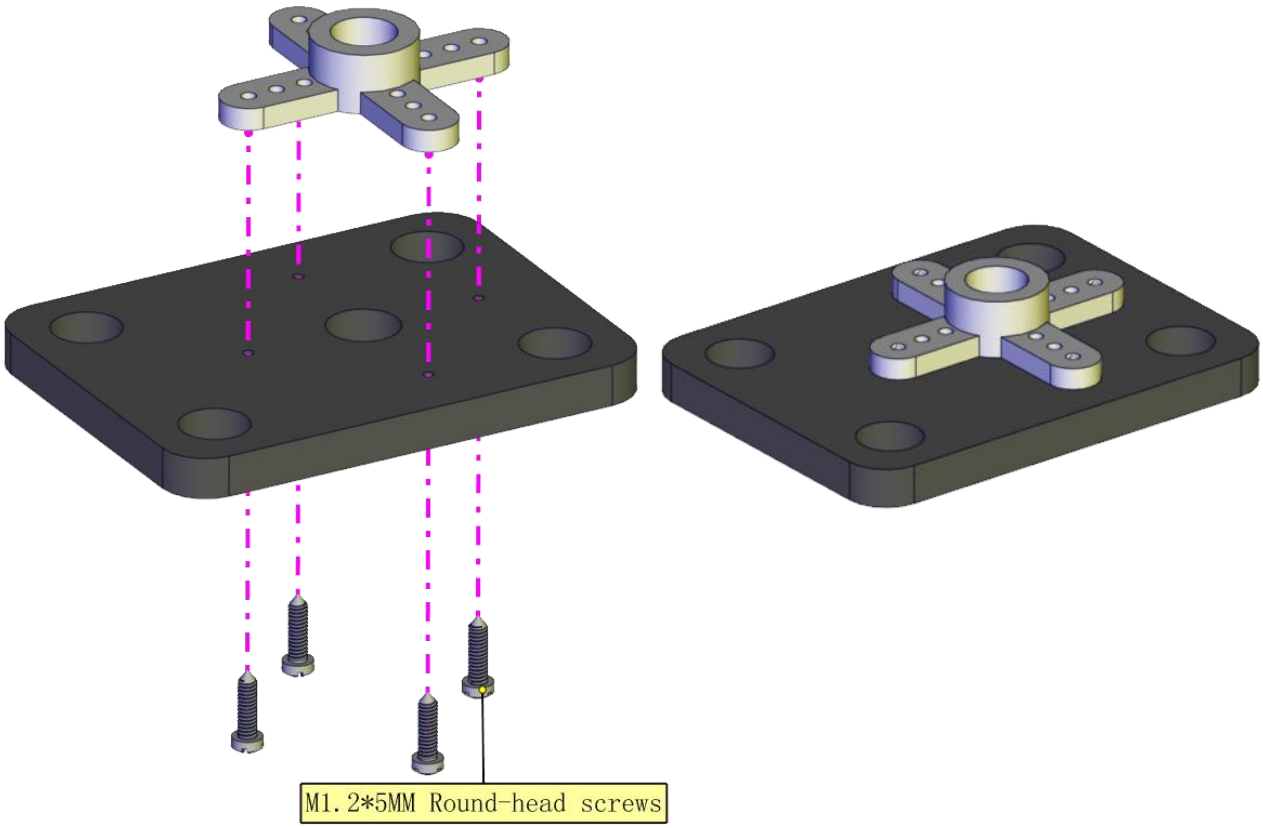


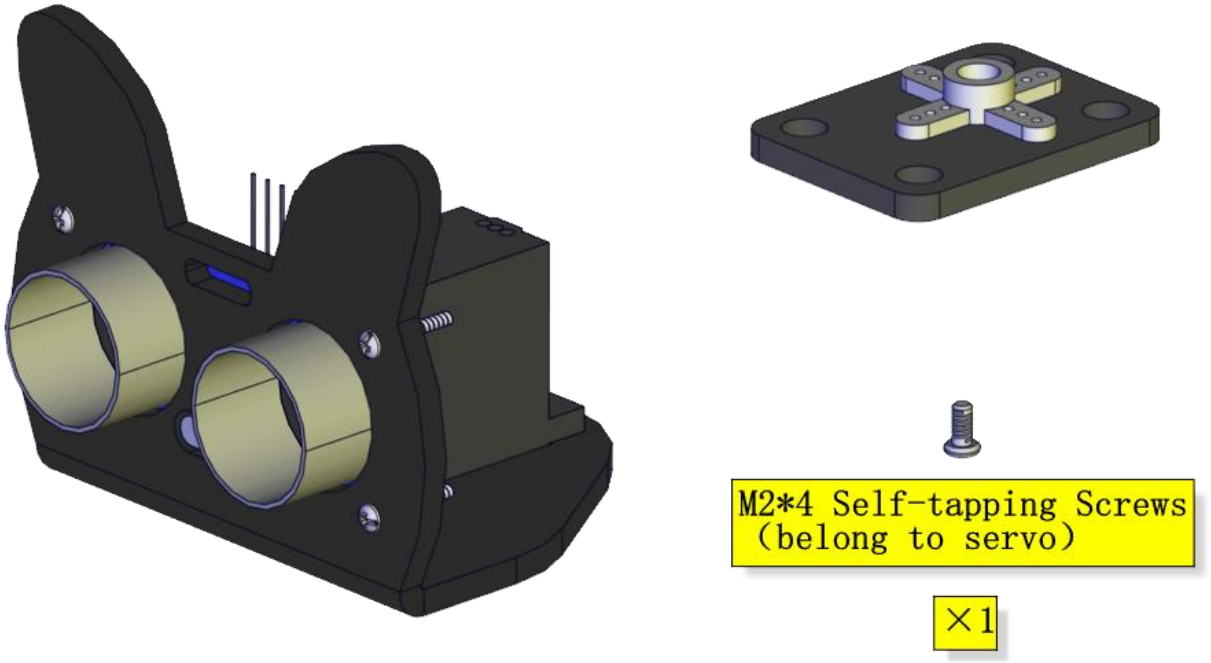
Acrylic Boards

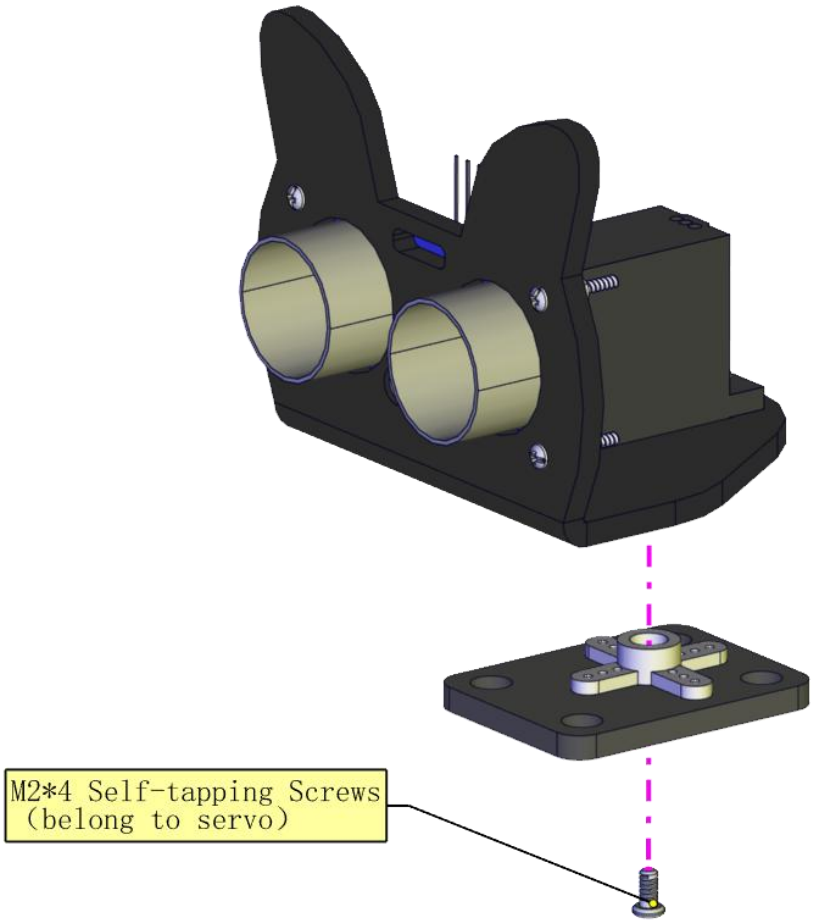
×1

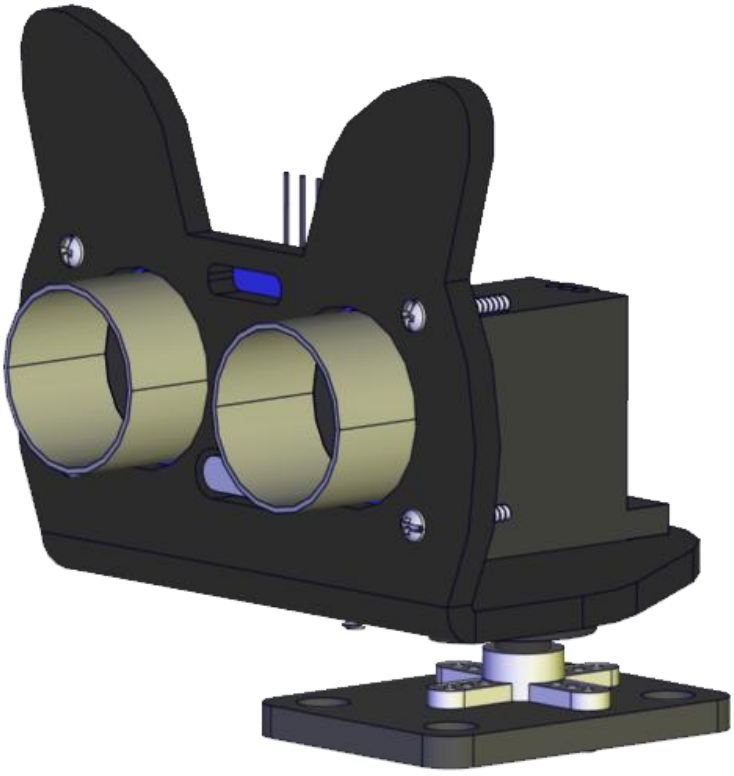
M1.2*5MM Round-head screws

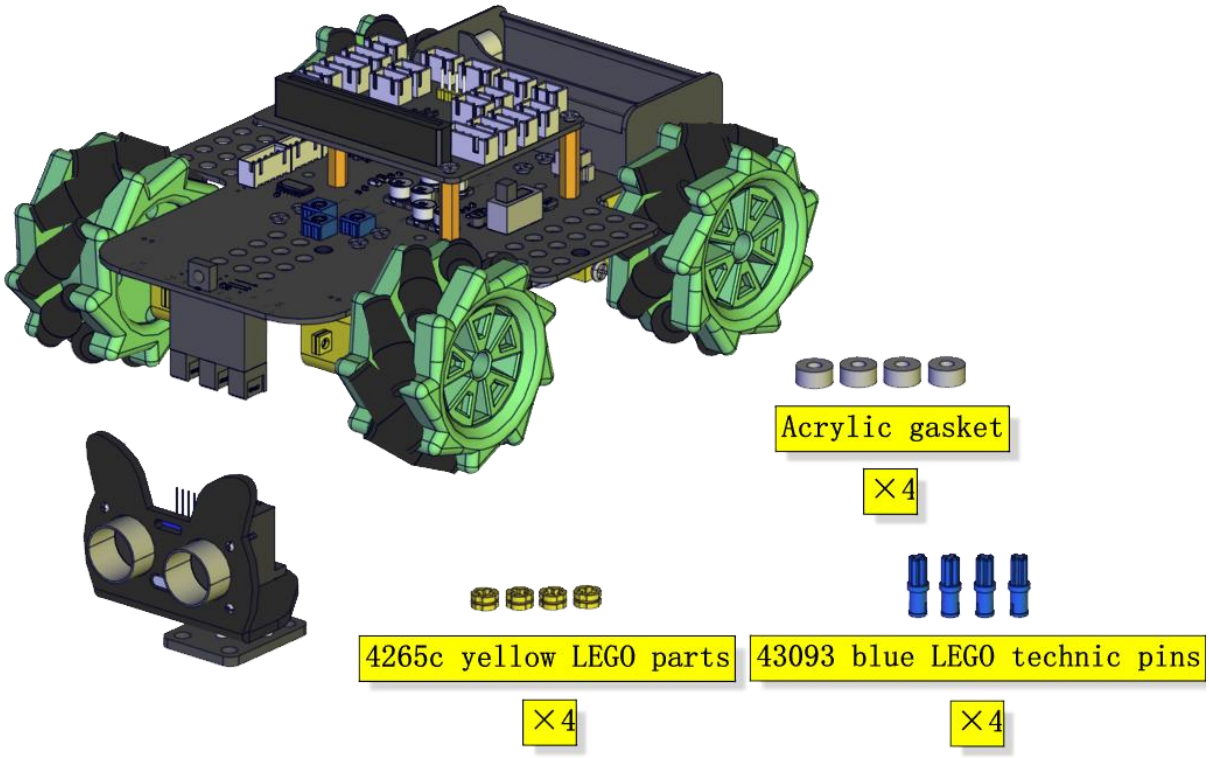
×4

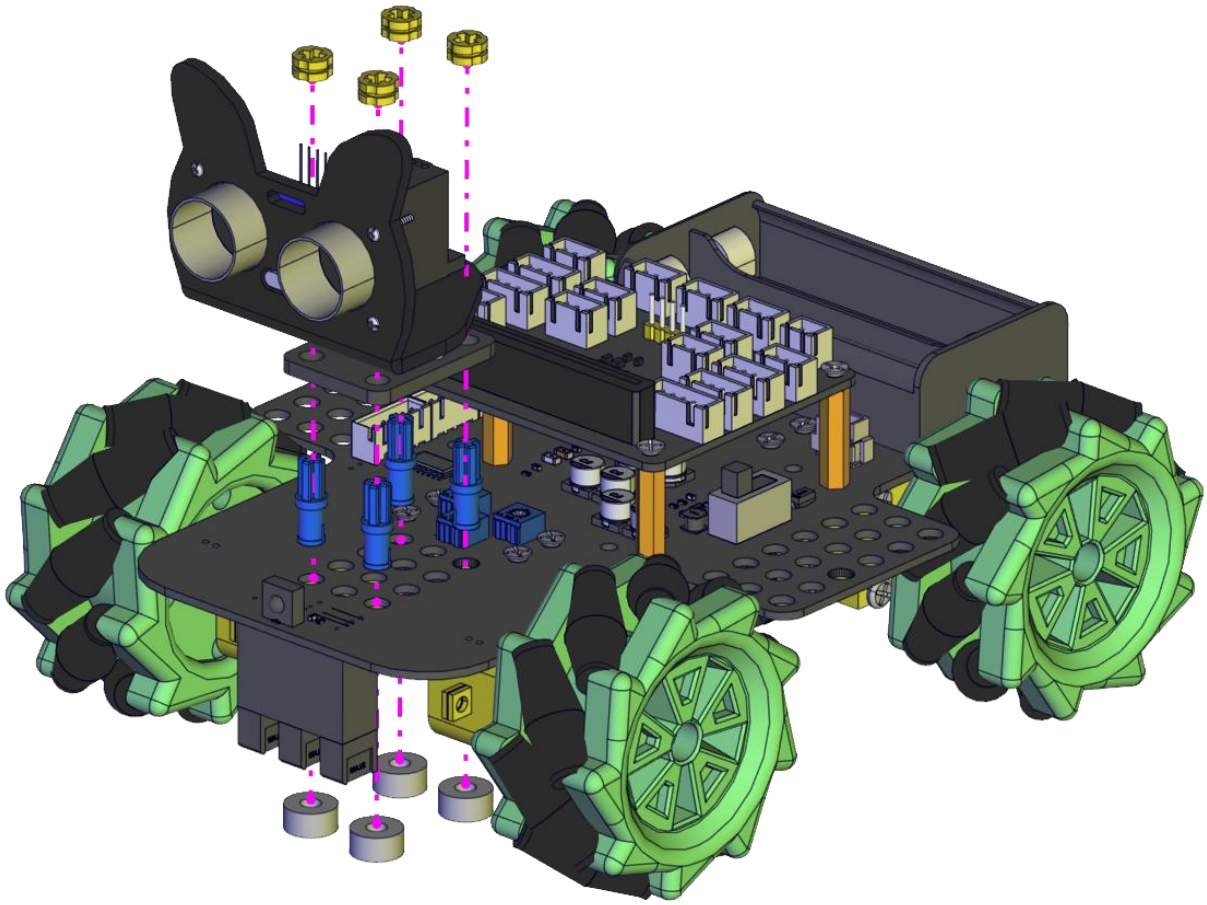

M1.2*5MM Round-head screws

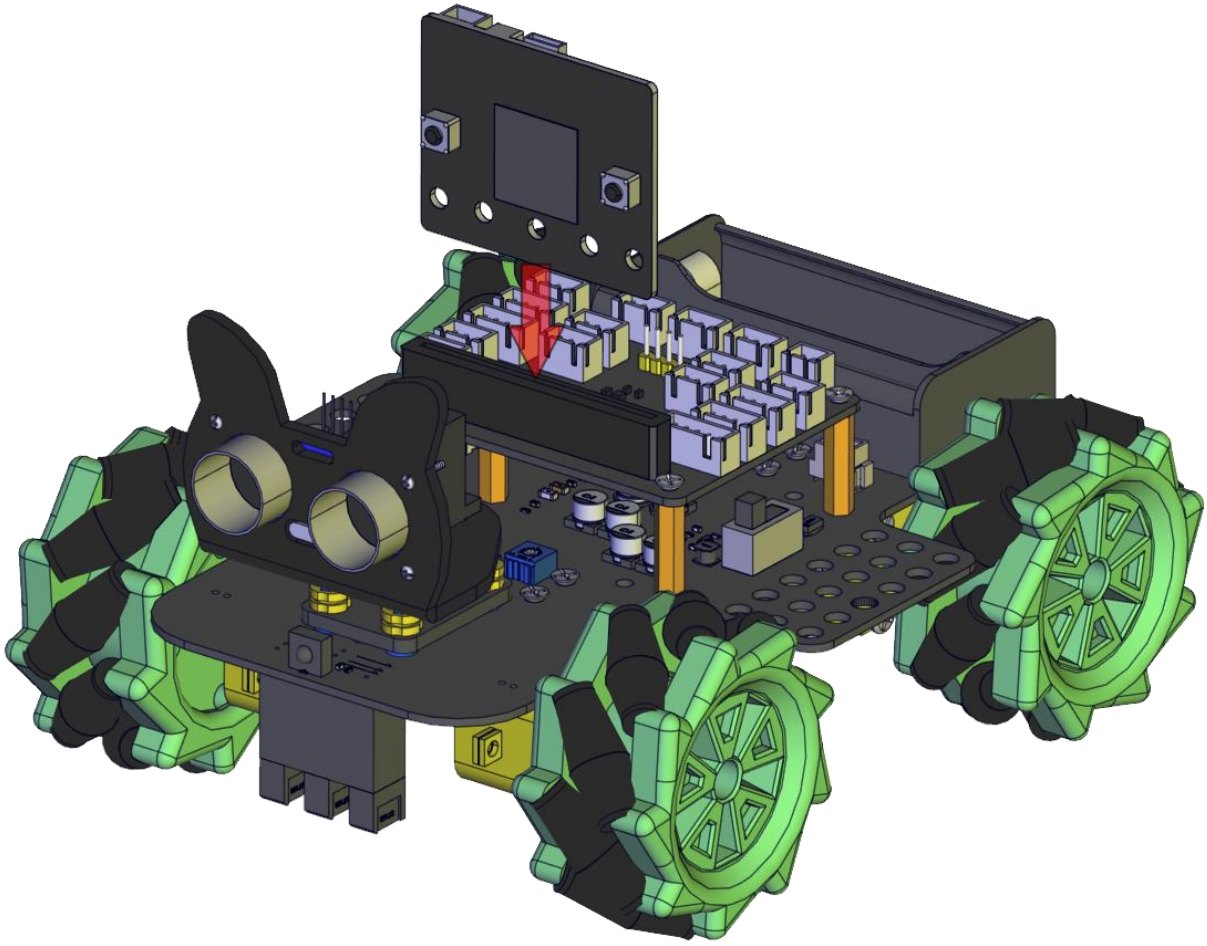

M2*4 Self-tapping Screws
(belong to servo)
×1

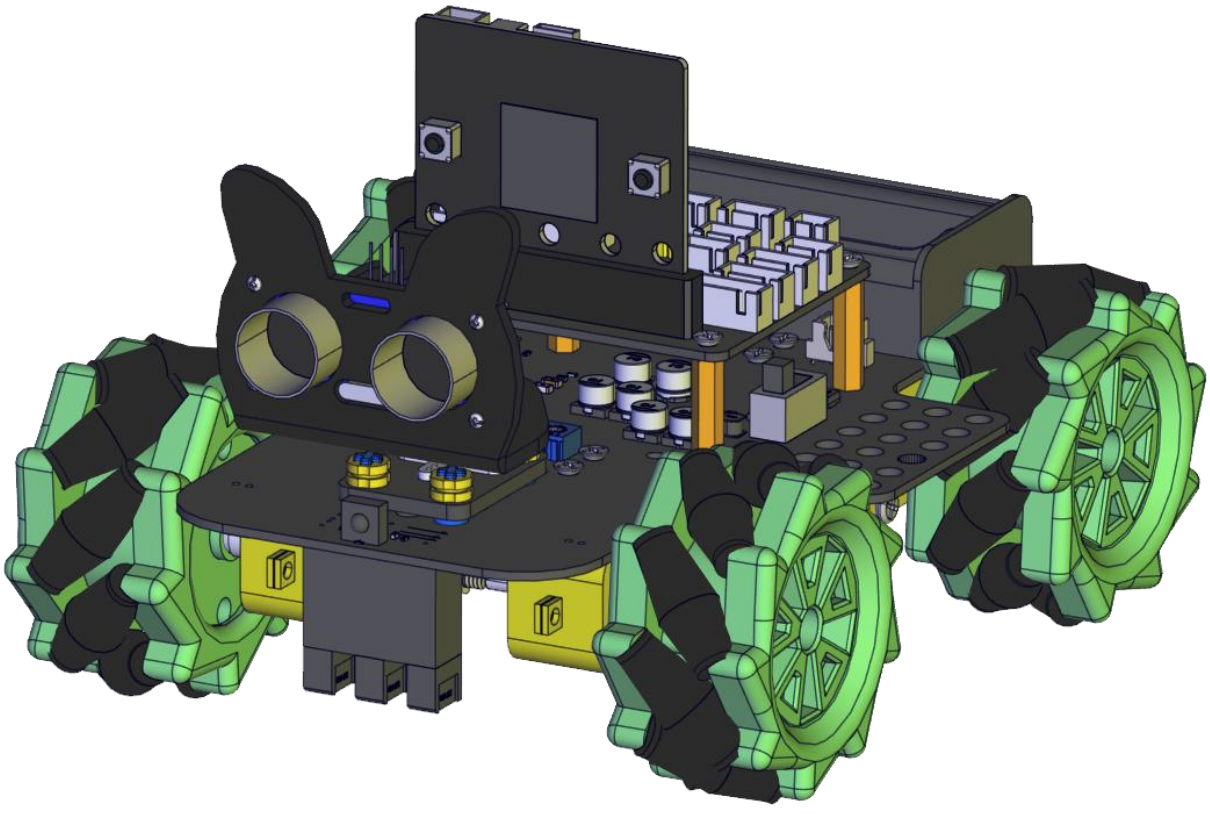

M2*4 Self-tapping Screws
(belong to servo)




Acrylic gasket
×4
4265c yellow LEGO parts
×4
43093 blue LEGO technic pins
×4







3.5 Wiring the Robot

1. Connect the battery case to the microbit shield with 100mm HX-2.54 2P duPont wire.

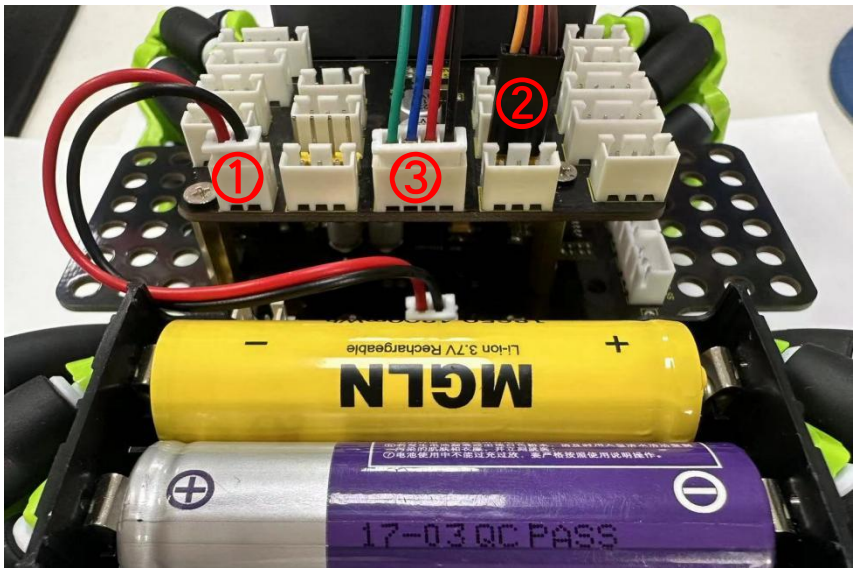
2. Connect servo to microbit shield.

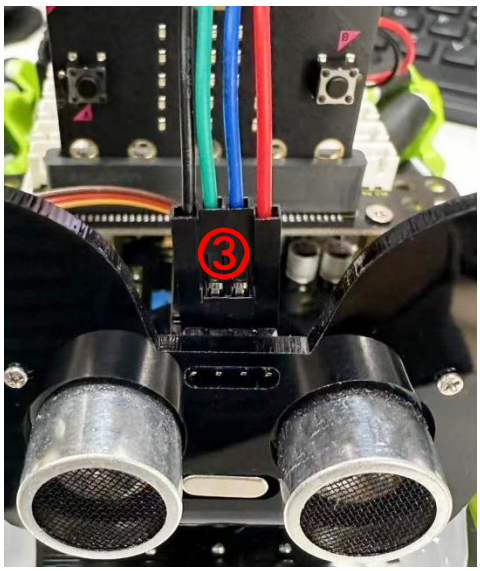
Servo	Microbit shield
Brown wire	G
Red wire	V
Yellow wire	P14

3.Connect the ultrasonic sensor to the microbit shield with 150mm HX2.54mm-4P to 2.54 Dupont wire.

Ultrasonic sensor	Microbit shield
VCC	5V
TRIG	P15
ECHO	P16
GND	G

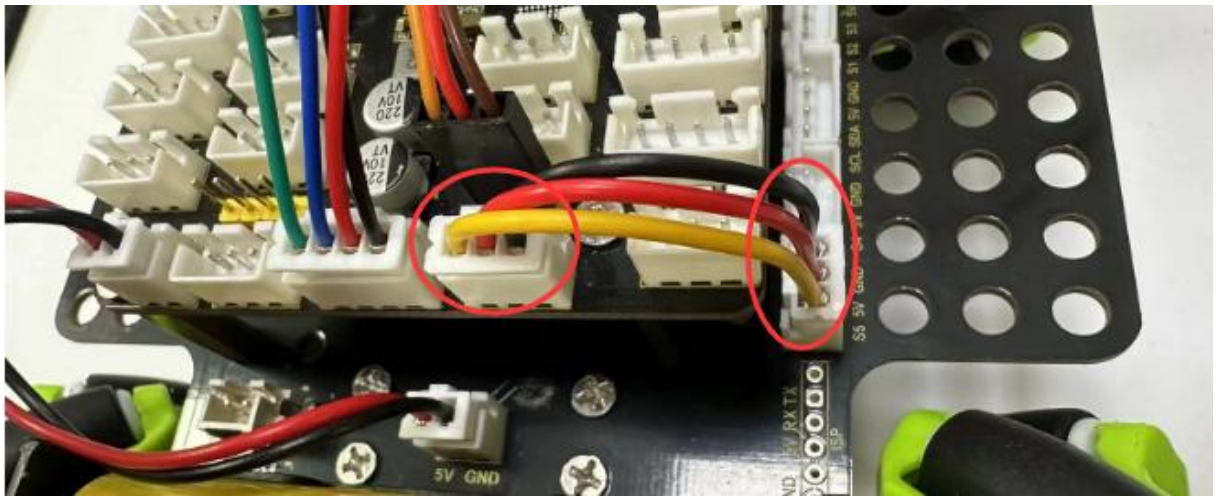
4.The positive and negative terminals of the battery are mounted in the direction shown below





5. Connect the onboard IR module to Microbit shield with 50mm XH2.54 3P wire.

Onboard IR module	Microbit shield
GND	G
5V	5V
S5	P0

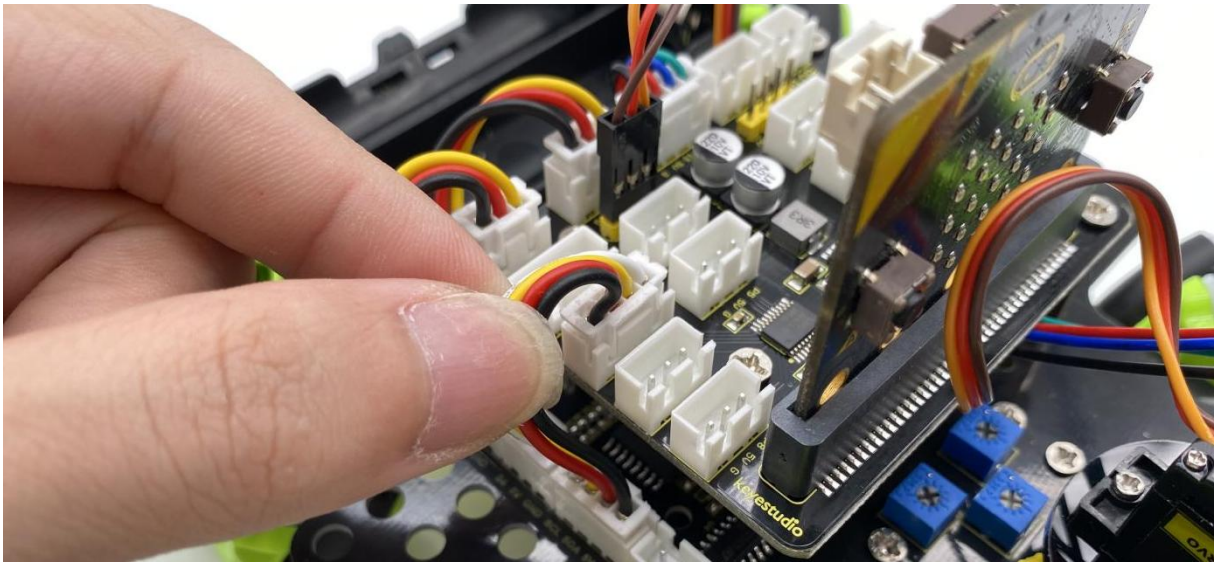


6. Connect the onboard RGB headlights to Microbit shield with 50mm XH2.54 3P wire.

Onboard RGB Headlight	Microbit shield
GND	G
5V	5V
S4	P7

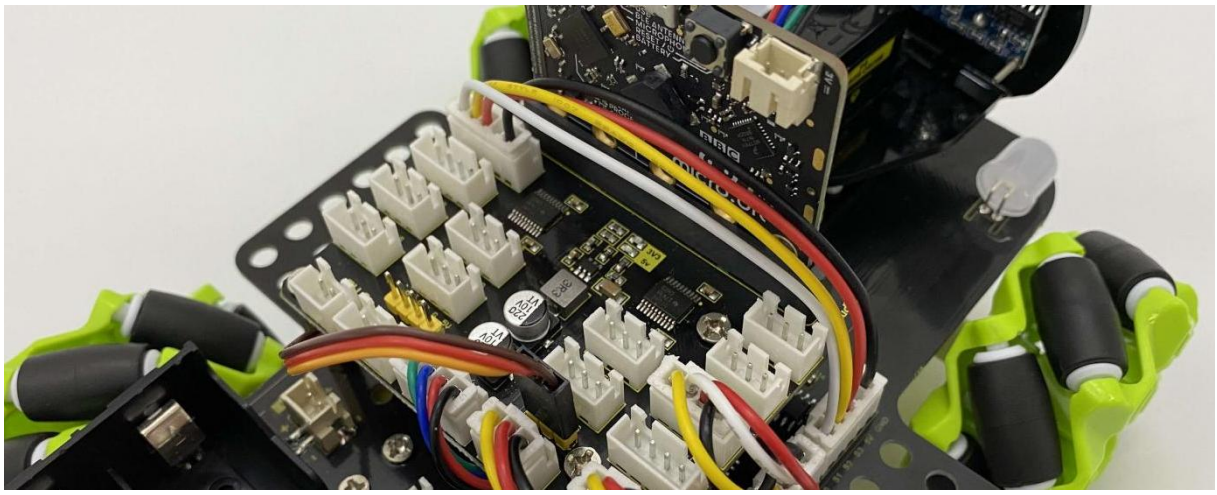
7. Connect the onboard WS2812 RGB lights to Microbit shield with 50mm HX-2.54 4P wire.

Onboard WS2812 RGB light	Microbit shield
SCL	P19
SDA	P20
5V	5V
GND	G

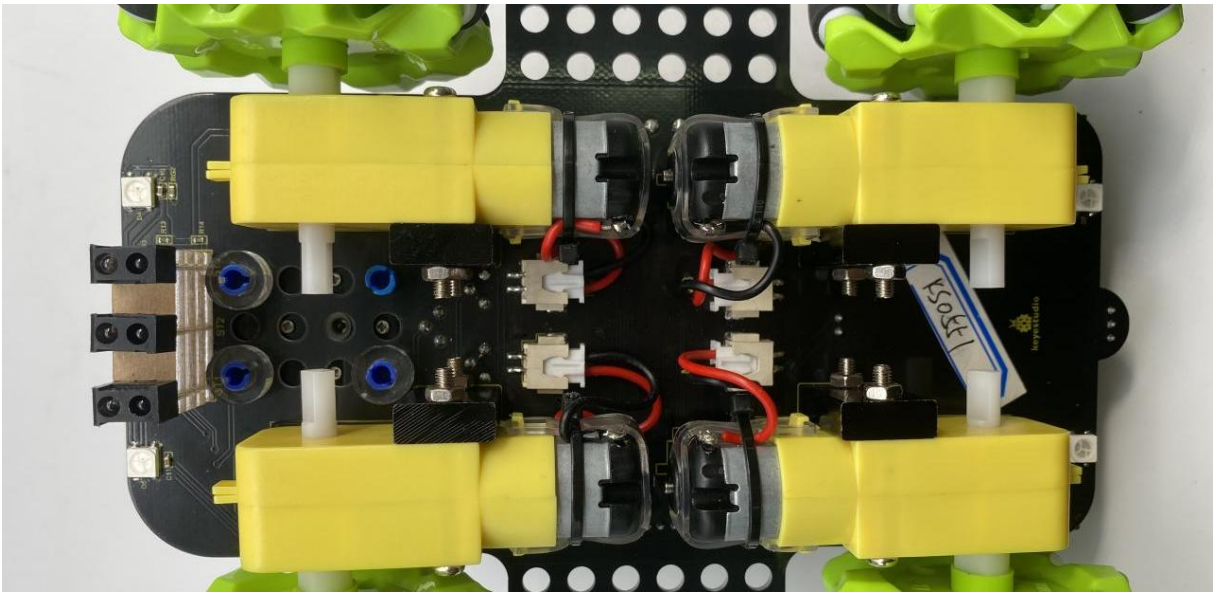


8. Connect the three-way line tracking module to Microbit shield with 100mm XH2.54 5P wire.

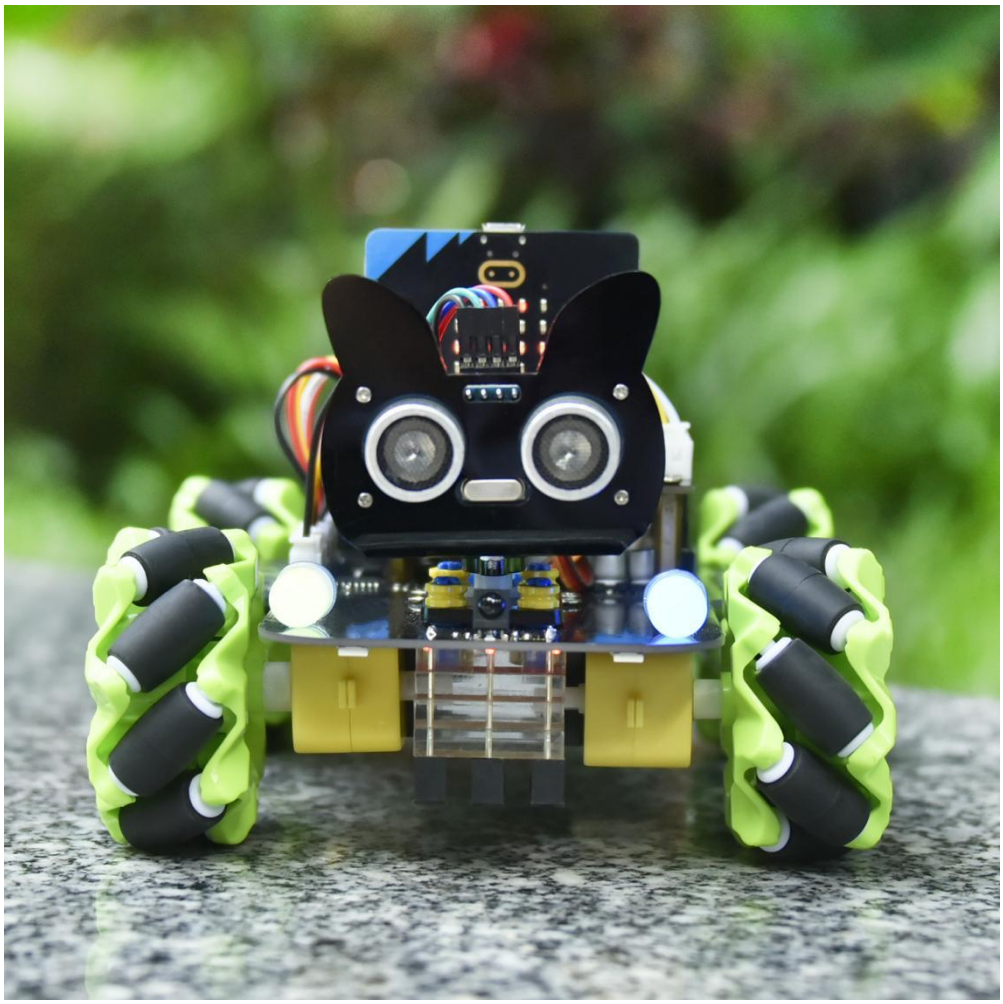
Line Tracking Module	Microbit shield
S1	P10
S2	P4
S3	P3
#	#
GND	G



9. Connect the motors to the car base board.



In the next chapter, we will start our journey to lean and explore the 4WD Mecanum Wheel Car！

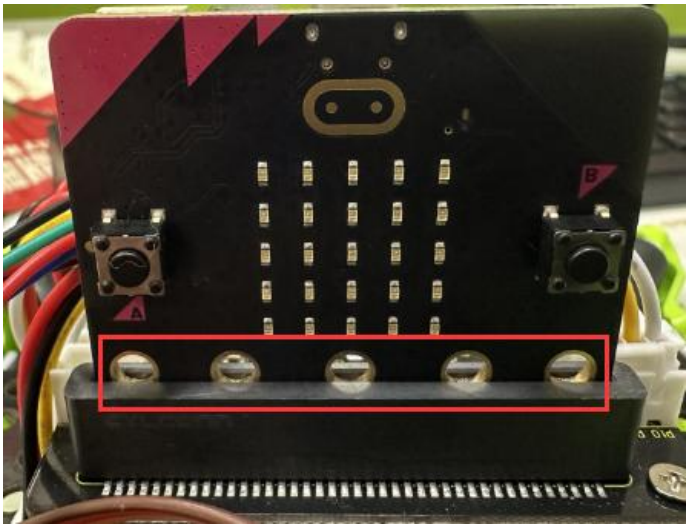


3.5 Robot Project

In this section, we'll start from testing individual electronic modules to making the robot perform complex tasks. Step by step, we'll show you how the mecanum wheel robot works.
It is highly recommended that you complete all of the projects as this will help verify whether each electronic module is working correctly or the robot is wired correctly, which is critical as they can affect whether the entire robot can work as expected in the end.

Things to Note Before Starting the Robot Project:

1. To avoid poor contact between Microbit and robot, please insert the Microbit correctly into the robot shield to make sure the interface of shield cover the edge of the round hole of the microbit.



2. Make sure the 18650 batteries have sufficient charge and is properly placed in the battery case.

3.51 Headlights

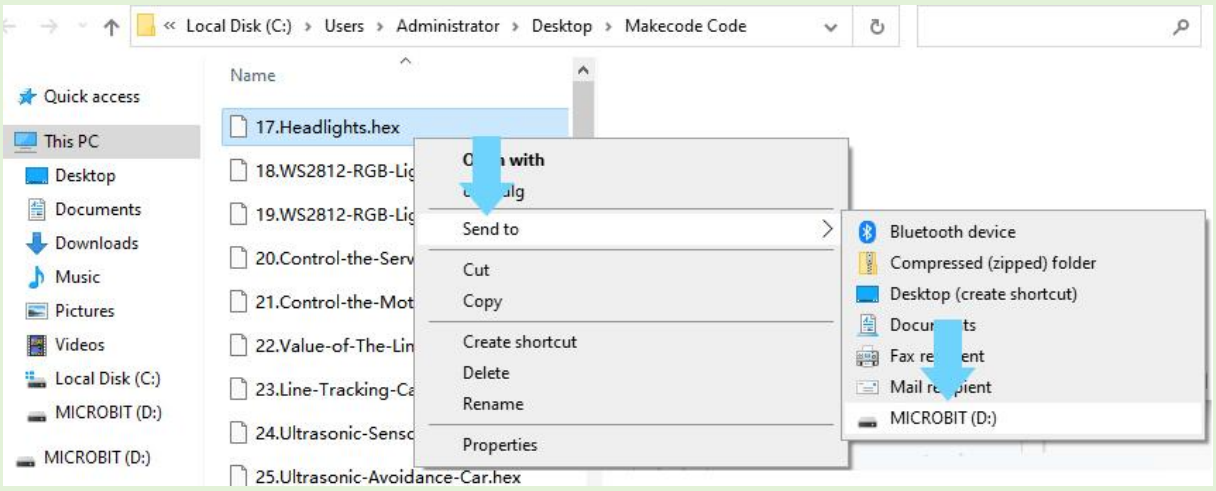
Light up the two RGB headlights on the mecanum car and let them automatically flash in 7 colors: red, green, blue, cyan, crimson, yellow, and white.

Steps:

- Connect your micro:bit to your computer with a USB cable.
- Turn on the POWER DIP switch on the car base.
- Find the downloaded Headlights.hex file, drag it in the Makecode to view and analyze, and use WebUSB to flash it to the micro:bit from inside editor.

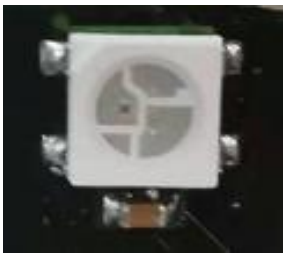
Or you can right-click on it and choose "Send to→MICROBIT."

File Type	Path	File Name
hex file	.../Makecode/Makecode Code	17.Headlights



- **Result:** two RGB headlights on the mecanum car will flash in 7 colors: red, green, blue, cyan, crimson, yellow, and white.

3.52 WS2812 RGB Lights



1 Light up the four WS2812 RGB lights under the car body and make them flash in white, red, orange, green, blue and purple.

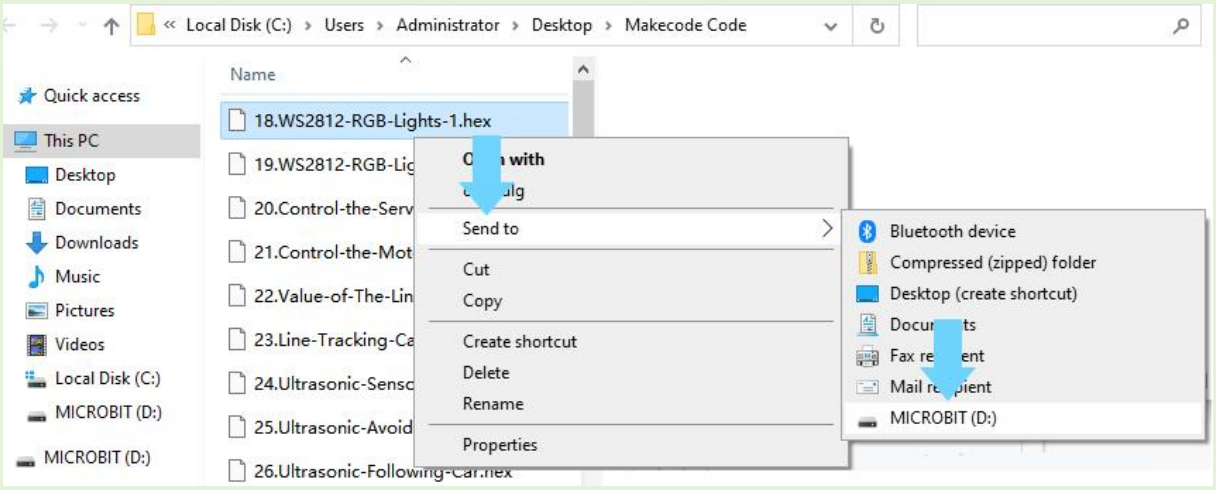
Steps:

- Connect your micro:bit to your computer with a USB cable.
- Turn on the POWER DIP switch on the car base.

Find the downloaded WS2812-RGB-Lights-1.hex file, drag it in the Makecode to view and analyze, and use WebUSB to flash it to the micro:bit from inside editor.

Or you can right-click on it and choose "Send to→MICROBIT."

File Type	Path	File Name
hex file	.../Makecode/Makecode Code	18.WS2812-RGB-Lights-1



- **Result:** The four WS2812 RGB lights under the car body will continuously flash white, red, orange, green, blue and purple.

2 Now let's download load new code to make the four WS2812 RGB lights flash one after another..

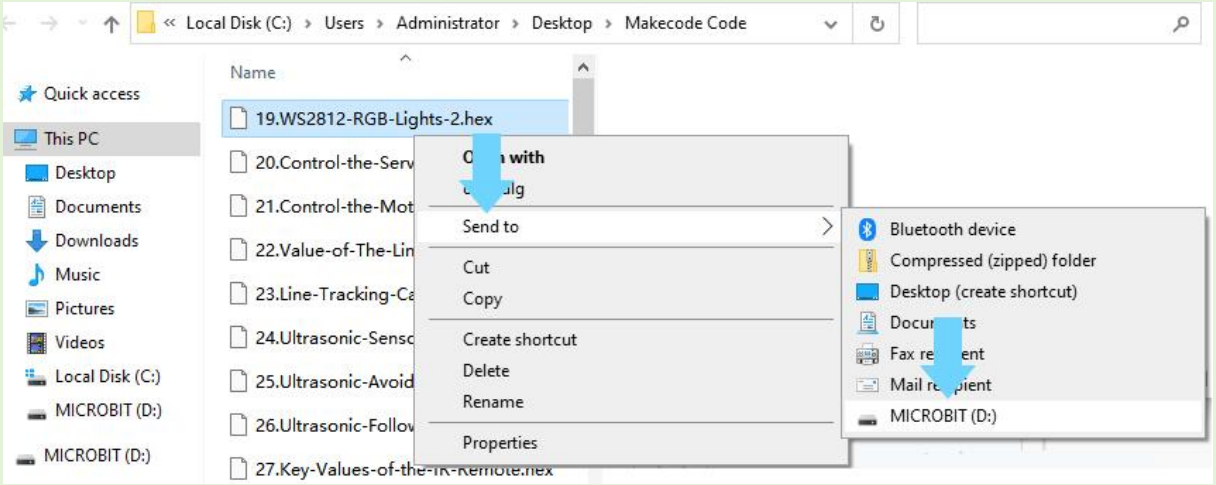
Steps:

- Connect your micro:bit to your computer with a USB cable.
- Turn on the POWER DIP switch on the car base.

Find the downloaded WS2812-RGB-Lights-2.hex file, drag it in the Makecode to view and analyze, and use WebUSB to flash it to the micro:bit from inside editor.

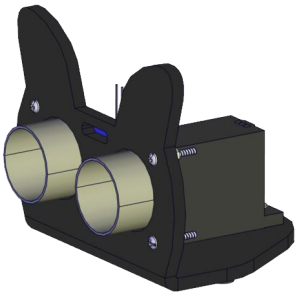
Or you can right-click on it and choose "Send to→MICROBIT."

File Type	Path	File Name
hex file	.../Makecode/Makecode Code	19.WS2812-RGB-Lights-2



- **Result:** The four WS2812 RGB lights under the car body will flash one after another.

3.53 Control the Servo



Let's control the servo of the robot to rotate back and forth between 0 and 180 degrees.

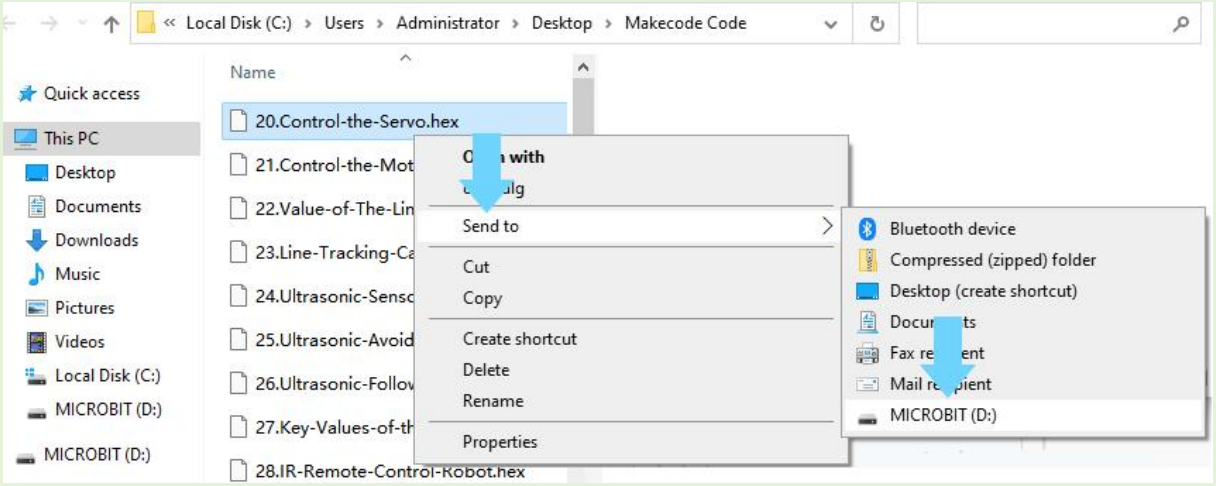
Steps:

- Connect your micro:bit to your computer with a USB cable.
- Turn on the POWER DIP switch on the car base.

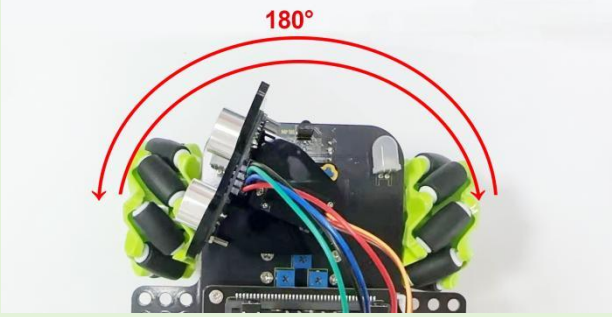
Find the downloaded Control-the-Servo.hex file, drag it in the Makecode to view and analyze, and use WebUSB to flash it to the micro:bit from inside editor.

Or you can right-click on it and choose "Send to→MICROBIT."

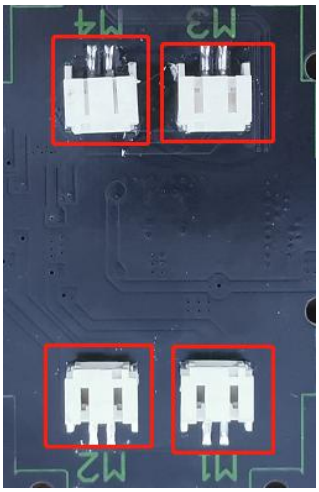
File Type	Path	File Name
hex file	.../Makecode/Makecode Code	20.Control-the-Servo



- **Result:** The robot's head will swing from side to side.

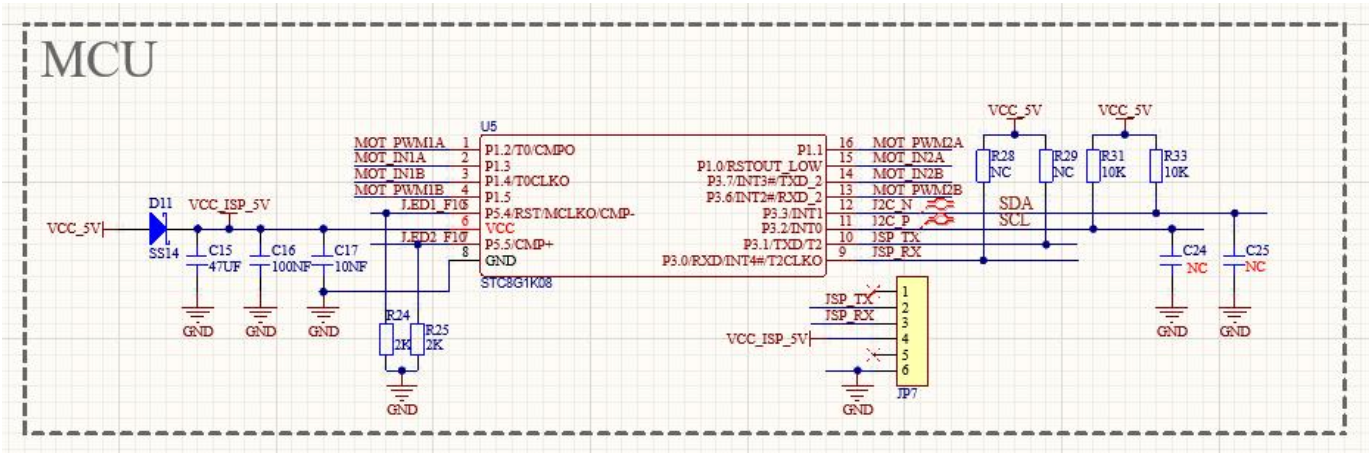


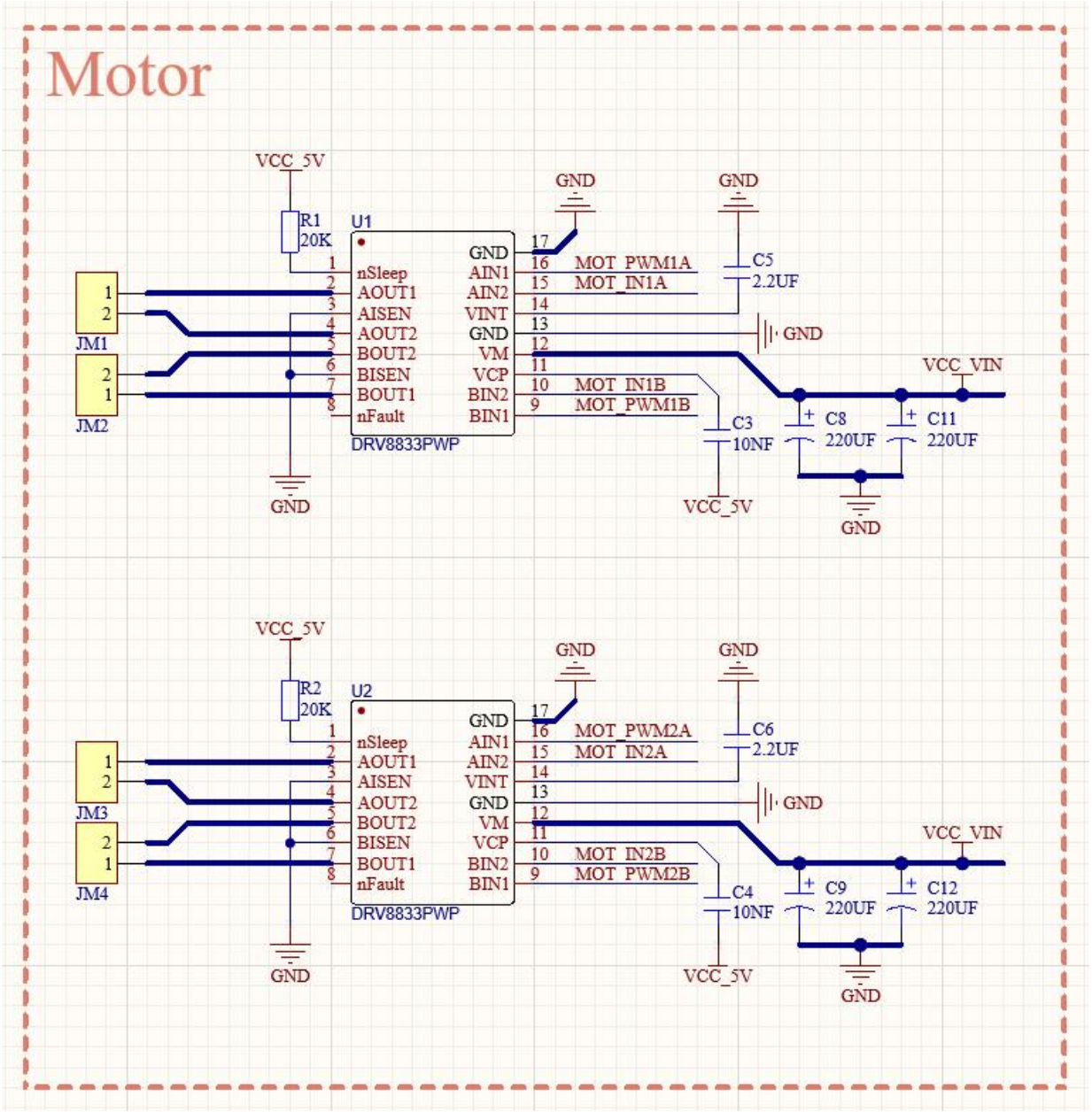
3.54 Control the Motors



The 4WD Mecanum Robot Car V2.0 is equipped with 4 DC reduction motors, namely gear reduction motors, which are based on ordinary DC motors and coupled with matching gear reduction boxes.

The Micro:bit motor driver baseboard has the most commonly used DRV8833 motor driver chip. In order to save IO port resources, we use some pins of the STC8G1K08 chip to control the DRV8833 motor driver chip, and the DRV8833 motor driver chip is used to control the rotation direction and speed of four DC reduction motors. Let’s take a look at the circuit diagram of the motor driver chip:





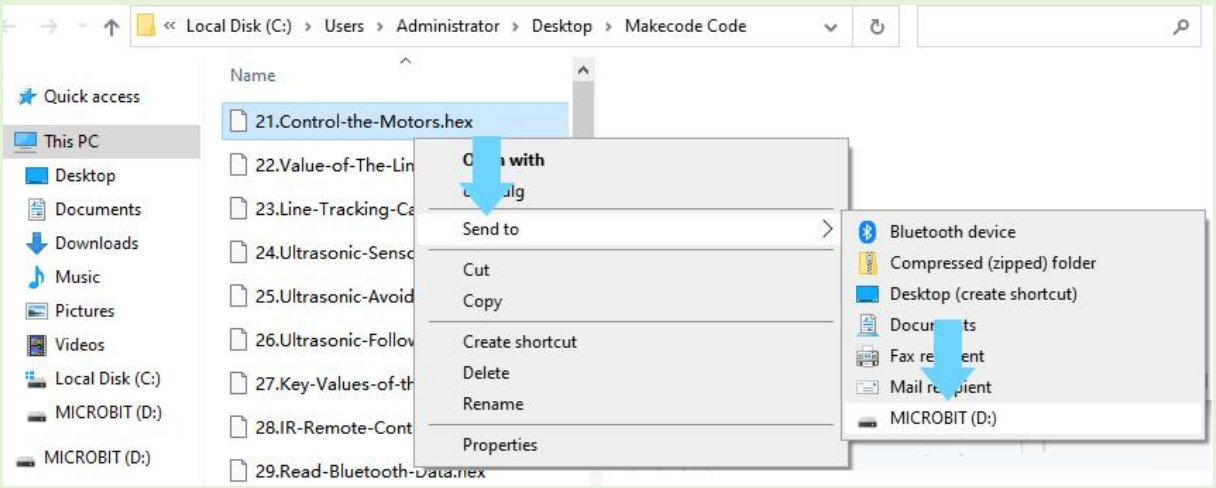
Now let's learn how to control the motor.

Steps:

- Connect your micro:bit to your computer with a USB cable.
- **Hold the robot up** to prevent the it from moving and falling from the table after the code is downloaded successfully.
- Find the downloaded Control-the-Motors.hex file, drag it in the Makecode to view and analyze, and use WebUSB to flash it to the micro:bit from inside editor.

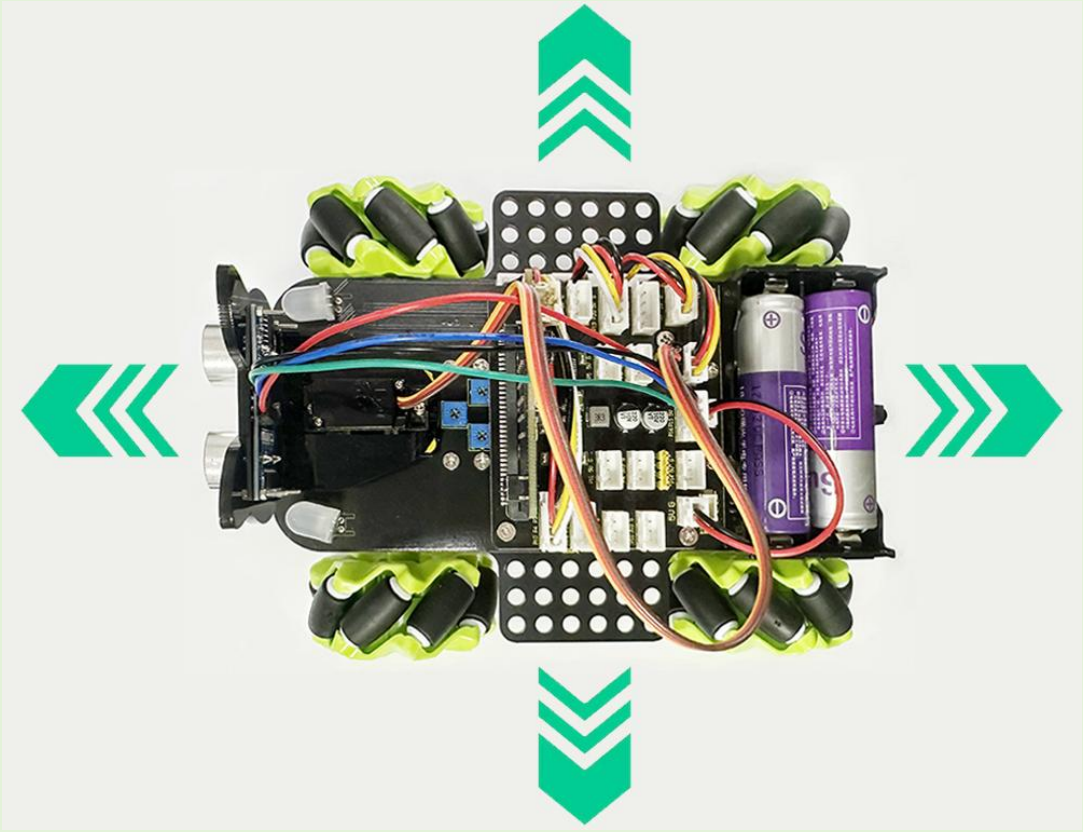
Or you can right-click on it and choose "Send to→MICROBIT."

File Type	Path	File Name
hex file	.../Makecode/Makecode Code	21.Control-the-Motors

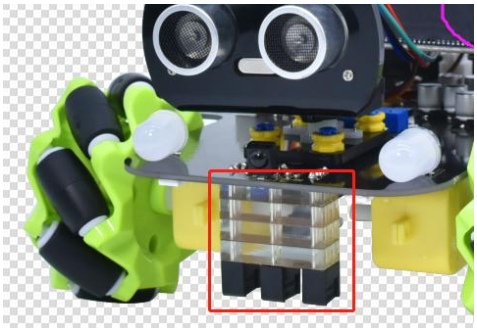


● **Result:** After the code is downloaded, place the robot on the ground and **turn on the power switch** on the car base.

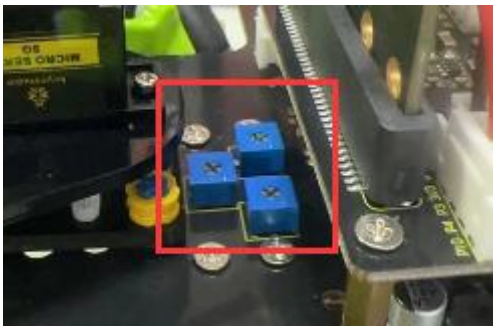
The car will move forward for 2 seconds, go back for 2 seconds, turn left for 2 seconds, turn right for 2 seconds, then stop for 2 seconds; And keep the cycle in this motion mode.



3.55 The Value of The Line Tracking Sensor



The car base board of Keyestudio Microbit 4WD Mecanum Robot Car V2.0 contains a 3-in-1 line tracking sensor module, and it also integrates three potentiometers on the top to allow you to adjust the sensitivity of the 3-in-1 line tracking sensor.



The line tracking sensor is an infrared sensor that uses the TCRT5000 infrared tube. The TCRT5000 Infrared sensor mainly includes an IR Tx (transmitter) & and IR Rx (receiver), the IR Tx generates an infrared signal continuously then reflects through an obstacle and is detected through the IR receiver.

This sensor module is frequently used in different robots like object sorting and line following because this module detects the surface whether that is black or white. Black objects absorb infrared signals, and white objects reflect infrared signals. Once this module is activated through power, then the IR diode will emit the IR signal frequently.
Digital Outputs HIGH (1) when black objects are detected.
Digital Outputs LOW (0) when white objects are detected.

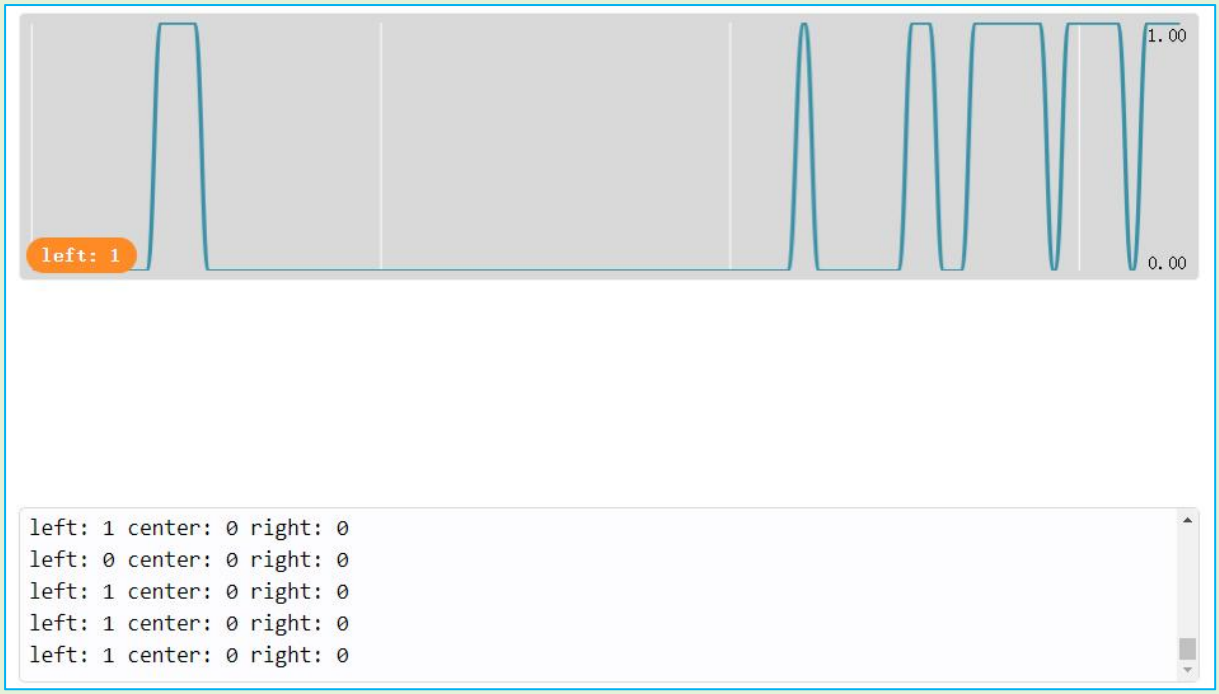
We will read the digital value of the TCRT5000 Infrared sensor in the simulator.

Steps:

- Connect your micro:bit to your computer with a USB cable.
- **Turn on the POWER DIP switch** on the car base and **hold it up** to prevent the robot from moving and falling from the table after the code is downloaded successfully.
- Find the downloaded Value-of-The-Line-Tracking-Sensor.hex file, drag it in the Makecode to view and analyze, and use WebUSB to flash it to the micro:bit from inside editor.

File Type	Path	File Name
hex file	.../Makecode/Makecode Code	22.Value-of-The-Line-Tracking-Sensor

- **Result:** You can use a black and white object or hold your finger close to the sensor. Click the "**Show data Device**" button and the simulator will display the digital value detected by each sensor.



3.56 Line Tracking Robot

Now let's make a line tracking robot.

The robot car takes different actions based on the digital values outputs from the line tracking sensor.

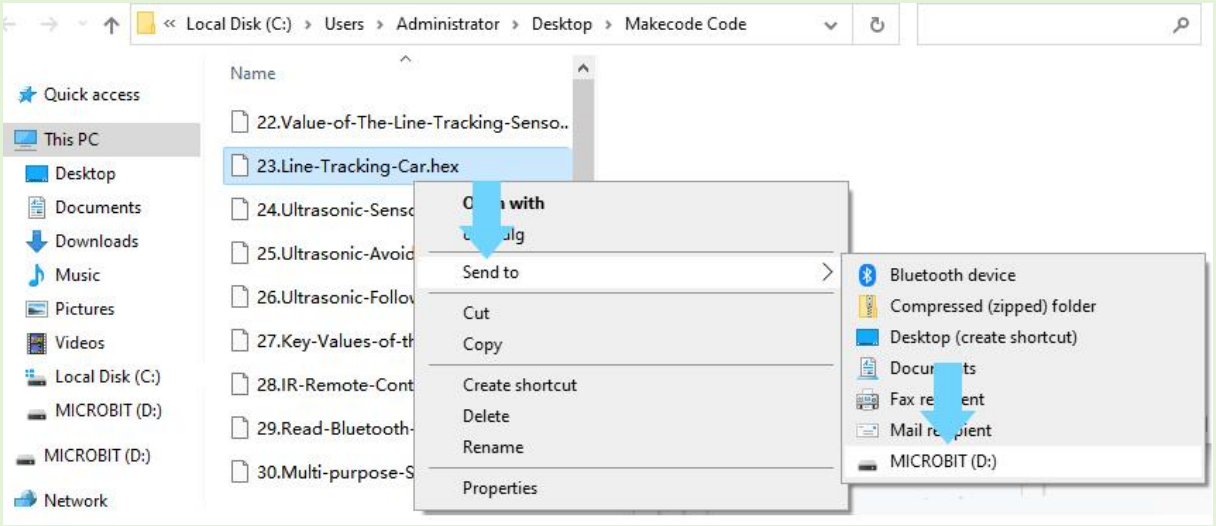
Left tracking sensor	Intermediate tracking sensor	Right tracking sensor	binary value	deci mal value	4WD Mecanum Robot Car
low level (0）	low level (0）	low level (0）	000	0	stop
low level (0）	low level (0）	high level (1）	001	1	Turn right
low level (0）	high level (1）	low level (0）	010	2	go ahead
low level (0）	high level (1）	high level (1）	011	3	Turn right
high level (1）	low level (0）	low level (0）	100	4	Turn left
high level (1）	low level (0）	high level (1）	101	5	go ahead
high level (1）	high level (1）	low level (0）	110	6	Turn left
high level (1）	high level (1）	high level (1）	111	7	stop

Steps:

- Connect your micro:bit to your computer with a USB cable.
- **Turn off the POWER DIP switch** on the car base and **hold it up** to prevent the robot from moving and falling from the table after the code is downloaded successfully.
- Find the downloaded Line-Tracking-Car.hex file, drag it in the Makecode to view and analyze, and use WebUSB to flash it to the micro:bit from inside editor.

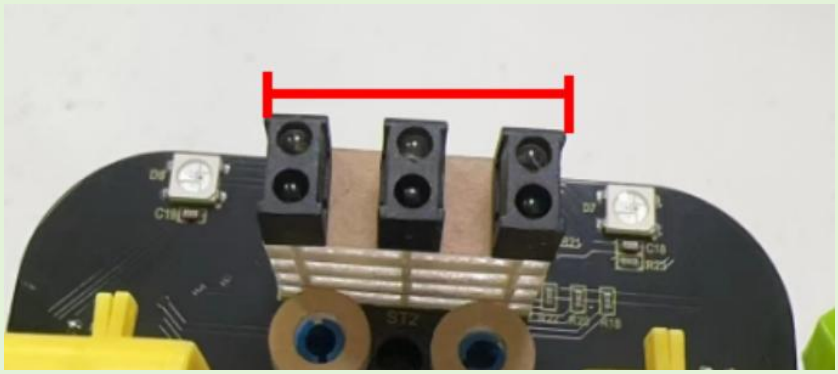
Or you can right-click on it and choose "Send to→MICROBIT."

File Type	Path	File Name
hex file	.../Makecode/Makecode Code	23.Line-Tracking-Car



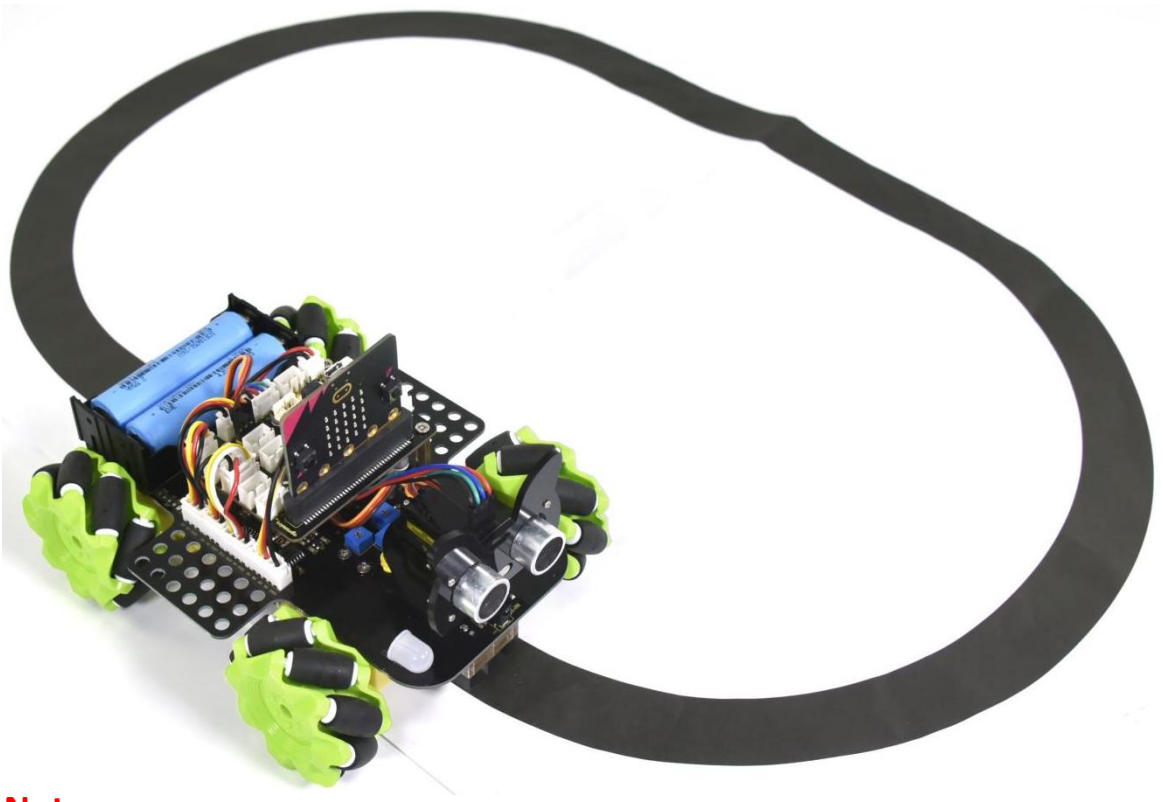
- **Result:**

Draw a black line wider than the 3-in-1 TCRT5000 Infrared sensor on the white paper.



After the code is downloaded, place the robot on the paper and align the TCRT5000 Infrared sensor with the black line.

Turn on the power switch on the car base, the robot car will move following the black line.

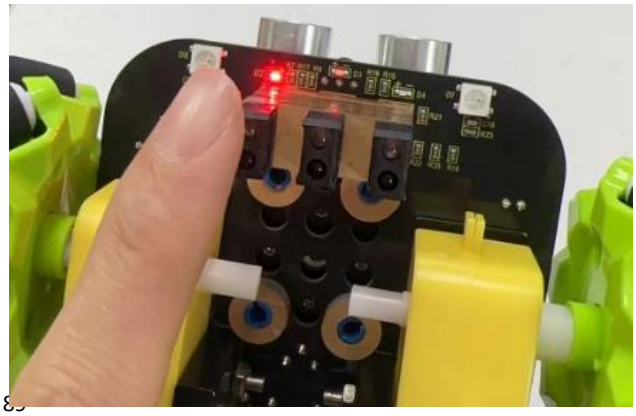


Note:

Avoid performing this experiment in strong sunlight. Sunlight contains a large amount of invisible light, such as infrared rays and ultraviolet rays, which will affect the work of the TCRT5000 Infrared sensor.

What if the robot doesn't work or doesn't follow the line?

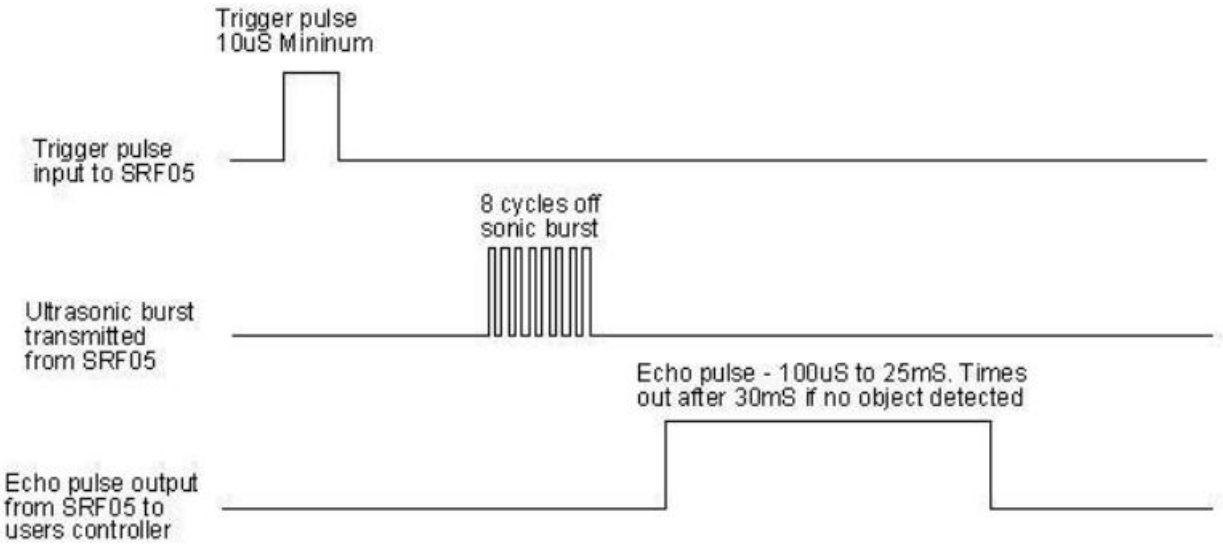
1. Please check whether the battery has sufficient power.
2. You can use a screwdriver to rotate the three potentiometers on the top to adjust the sensitivity of the sensor. When a finger is close to the sensor, its onboard LED lights up, indicating that the sensor's sensitivity is good.



3.57 Ultrasonic Sensor Measures Distance

Let's look at the picture of the ultrasonic sensor module. There are two things like eyes, one is the signal transmitter (TRIG) and the other is the signal receiver (ECHO).

working principle:



(1) We first pull TRIG (T) low, and then give at least 10us of high-level signal to trigger;
(2) After triggering, the module will automatically transmit 8 40KHZ square waves and automatically detect whether there is a signal return;
(3) If a signal returns, a high level is output through ECHO (E). The duration of the high level is the time from transmission to reception of the ultrasonic wave. Then the test distance = the duration of high level * 340m/s * 0.5.

Specifications:

Working voltage: 3-5.5V (DC)
Working current: 15mA
Working frequency: 40khz
Maximum detection distance: about 3m
Minimum detection distance: 2-3cm
High accuracy: up to 0.2cm
Sensing angle: no more than 15 degrees
Input trigger pulse: 10us TTL level
Output echo signal: Output TTL level signal (high), proportional to the range

Now, let's use the ultrasonic sensor to measure the distance of an object in front of the robot.

Steps:

- Connect your micro:bit to your computer with a USB cable.
- Turn on the POWER DIP switch on the car base
- Find the downloaded Ultrasonic-Sensor-Measures-Distance.hex file, drag it in the Makecode to view and analyze, and use WebUSB to flash it to the micro:bit from inside editor.

File Type	Path	File Name
hex file	.../Makecode/Makecode Code	24.Ultrasonic-Sensor-Measures-Distance

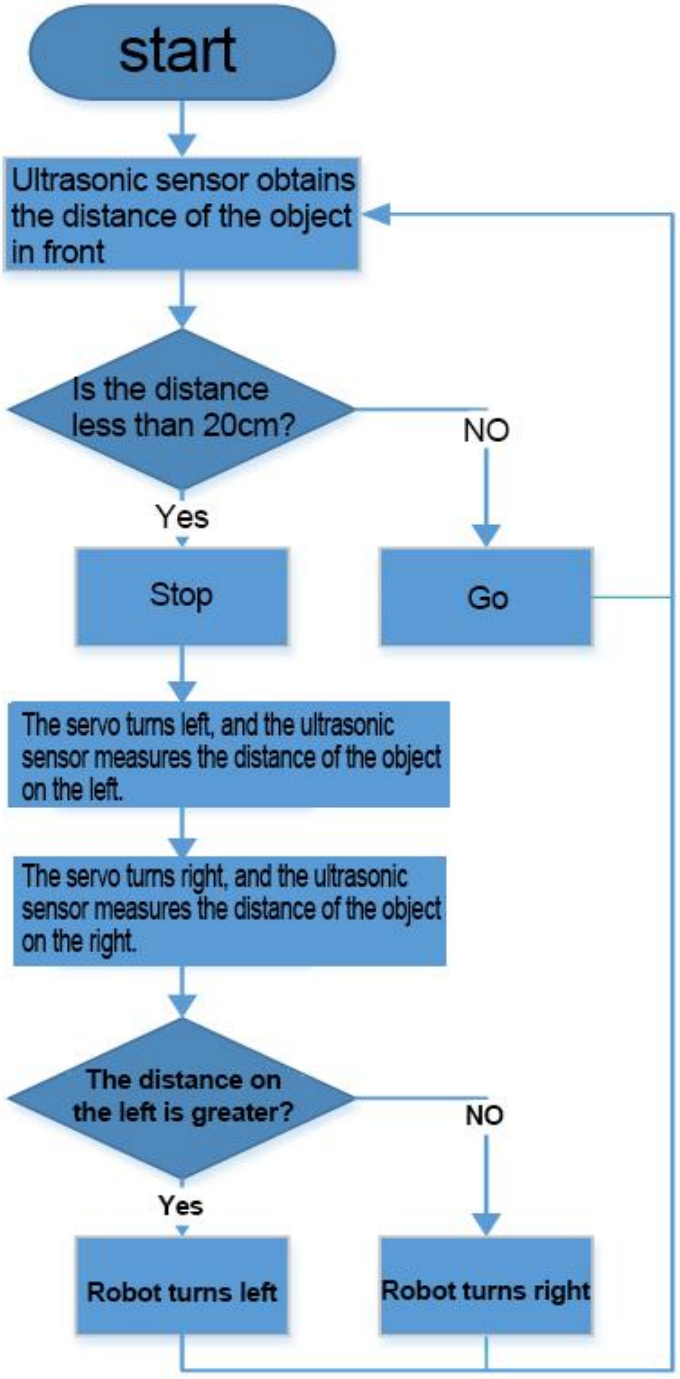
- **Result:** Click the "**Show data Device**" button under the simulator. The distance of the object in front measured by the ultrasonic sensor will be displayed in the Data View Window.

3.58 Obstacle Avoidance Robot

An Obstacle Avoidance Robot is an intelligent robot, which can automatically sense and overcome obstacles on its path.

Let's make an obstacle avoidance robot！

working principle：

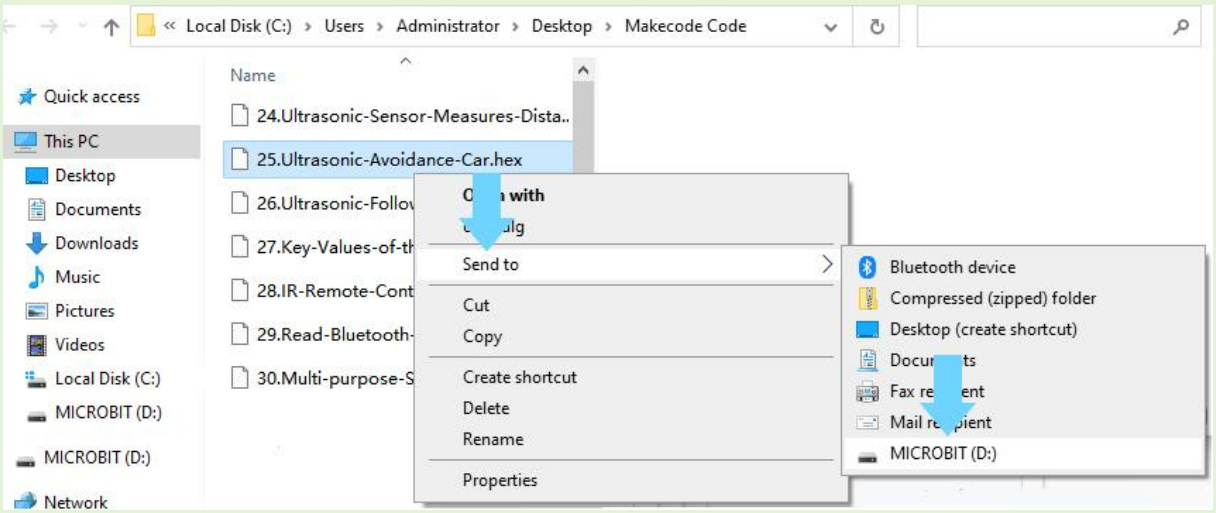


Steps:

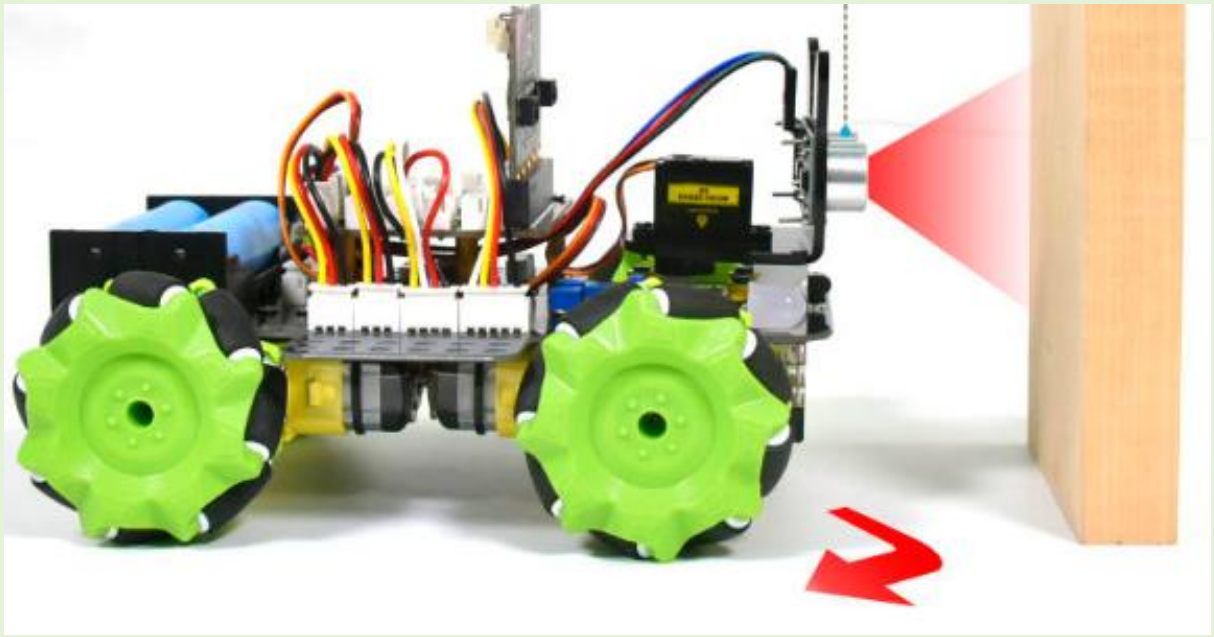
- Connect your micro:bit to your computer with a USB cable.
- **Turn off the POWER DIP switch** on the car base and **hold it up** to prevent the robot from moving and falling from the table after the code is downloaded successfully.
- Find the downloaded Ultrasonic-Avoidance-Car.hex file, drag it in the Makecode to view and analyze, and use WebUSB to flash it to the micro:bit from inside editor.

Or you can right-click on it and choose "Send to→MICROBIT."

File Type	Path	File Name
hex file	.../Makecode/Makecode Code	25.Ultrasonic-Avoidance-Car



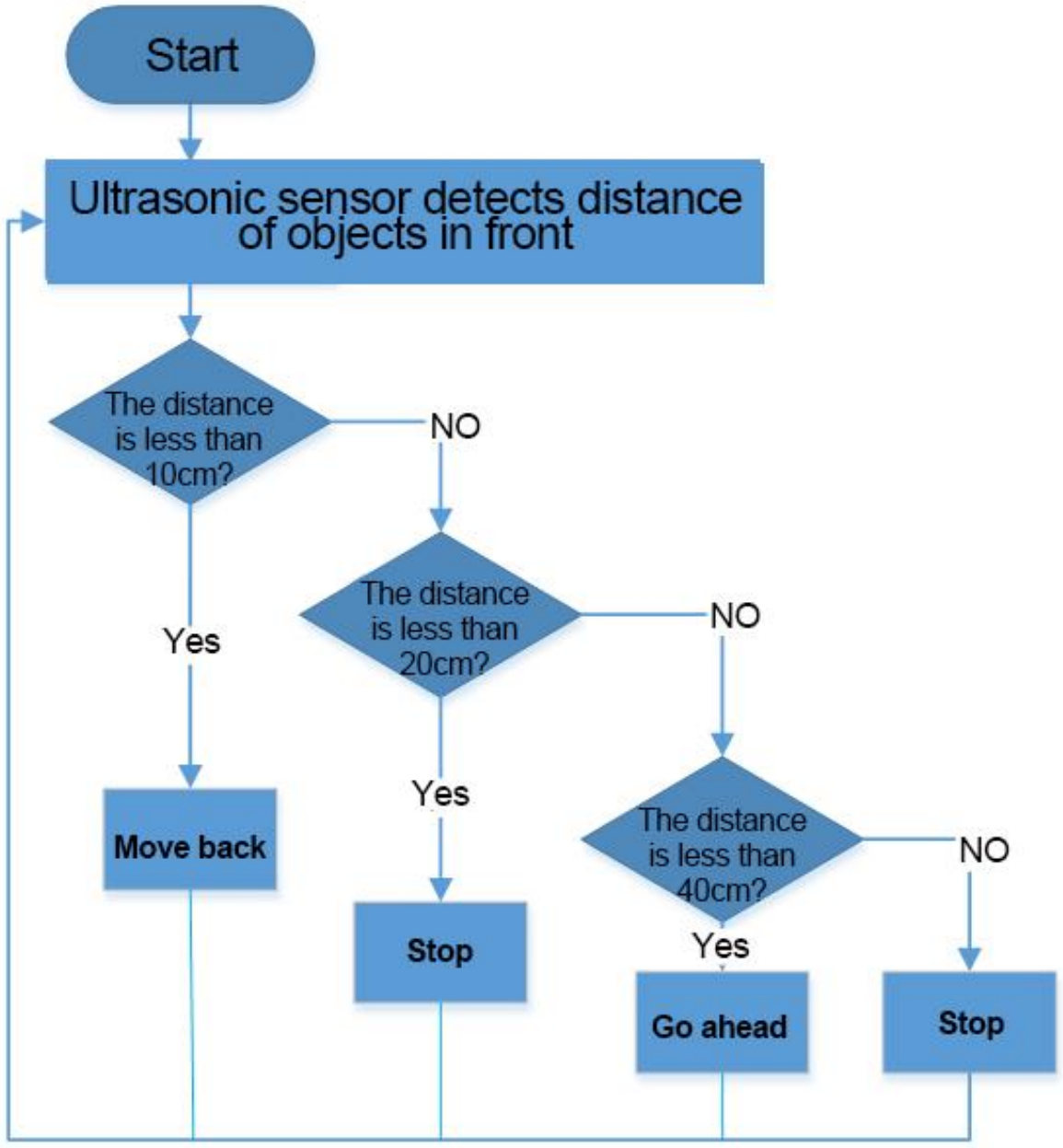
- **Result:** Place the robot on the ground and **turn on the power switch** on the car base, the robot car will will move forward and avoid obstacles.



3.59 Obstacle Following Robot

Contrary to the previous chapter, we will make an ultrasonic following car in this step. The surface of the obstacle in front should be flat and large enough to return ultrasonic signals to the ultrasonic module.

Working principle：

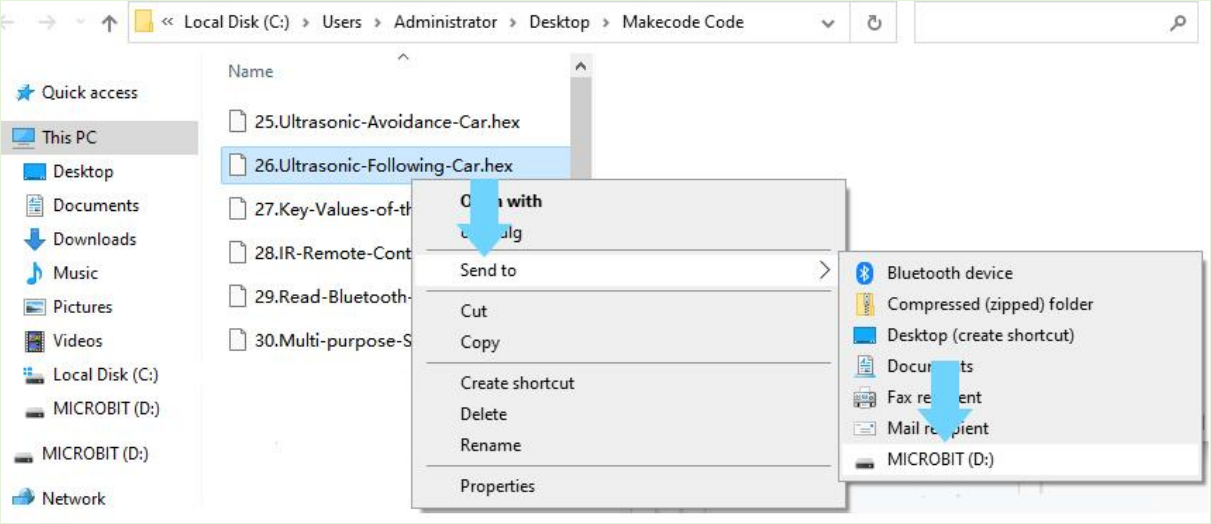


Steps:

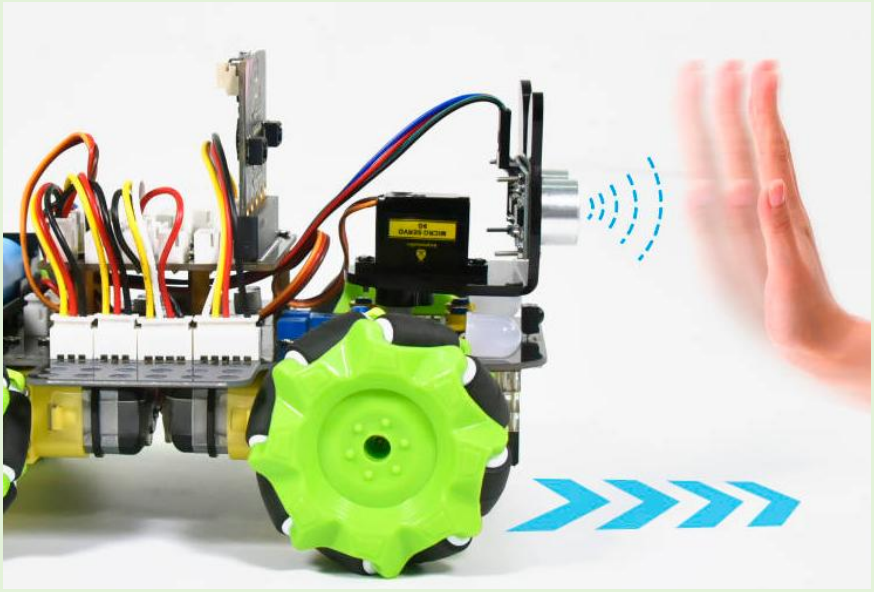
- Connect your micro:bit to your computer with a USB cable.
- **Turn off the POWER DIP switch** on the car base and **hold the robot up** to prevent the robot from moving and falling from the table after the code is downloaded successfully.
- Find the downloaded Ultrasonic-Following-Car.hex file, drag it in the Makecode to view and analyze, and use WebUSB to flash it to the micro:bit from inside editor.

Or you can right-click on it and choose "Send to→MICROBIT."

File Type	Path	File Name
hex file	.../Makecode/Makecode Code	26.Ultrasonic-Following-Car

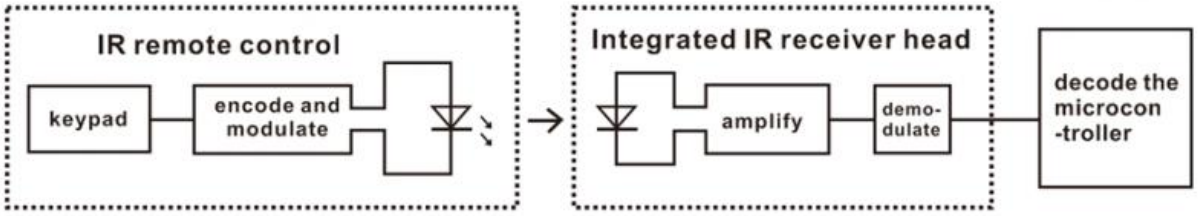


- **Result:** Place the robot on the ground and **turn on the power switch** on the car base, place your phone or palm in front of the ultrasound and the robot will follow it.



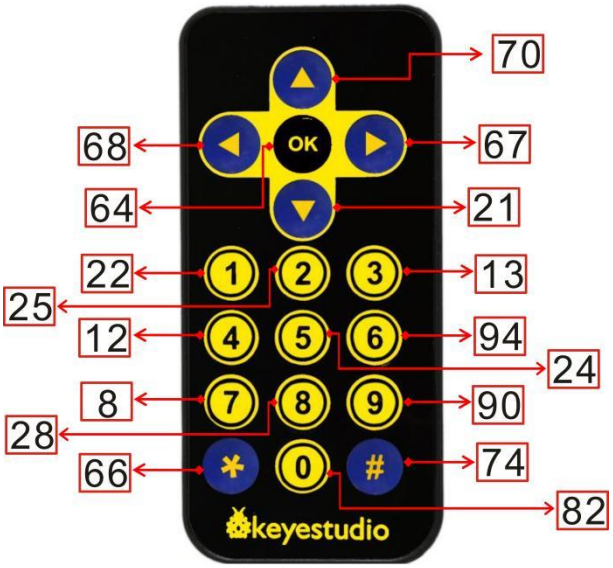
3.60 Key Value of the IR Remote

Infrared remote controls are used to control various home appliances such as televisions, stereos, video recorders, and satellite receivers. The infrared remote control is composed of an infrared remote control, an infrared receiver module, and a microcontroller that can decode.



When a button on the remote control is pressed, the remote control sends out an infrared carrier signal. When the infrared receiver receives the signal, the program decodes the carrier signal and determines which button is pressed based on the difference in data code. The microcontroller decodes the received 0 and 1 signals to determine which key is pressed on the remote control.

The infrared receiving module on the Keyestudio Microbit 4WD Mecanum Robot Car V2.0 is integrated into the P9 5V G collective port of the car base and is controlled by the P9 of micro:bit.



In this step, we will use View Data Window to display the key values sent from the infrared remote control.

Steps:

- Connect your micro:bit to your computer with a USB cable.
- **Turn on the POWER DIP switch** on the car base.
- Find the downloaded Key-Values-of-the-IR-Remote.hex file, drag it in the Makecode to view and analyze, and use WebUSB to flash it to the micro:bit from inside editor.

File Type	Path	File Name
hex file	.../Makecode/Makecode Code	27.Key-Values-of-the-IR-Remote

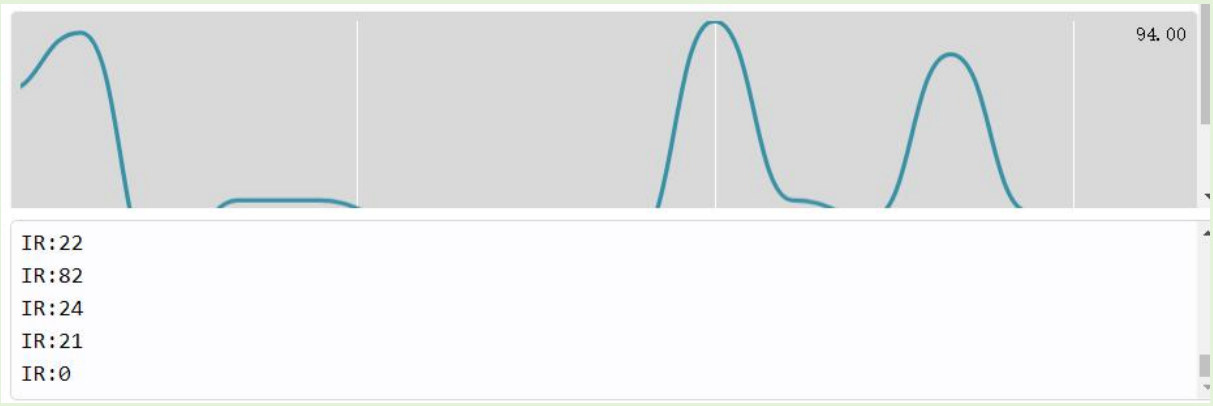
Result:

1.Click the "**Show data Device**" button under the simulator.

2.Pull out the plastic insulation film from the infrared remote control.

3. Make sure there's a battery inside.

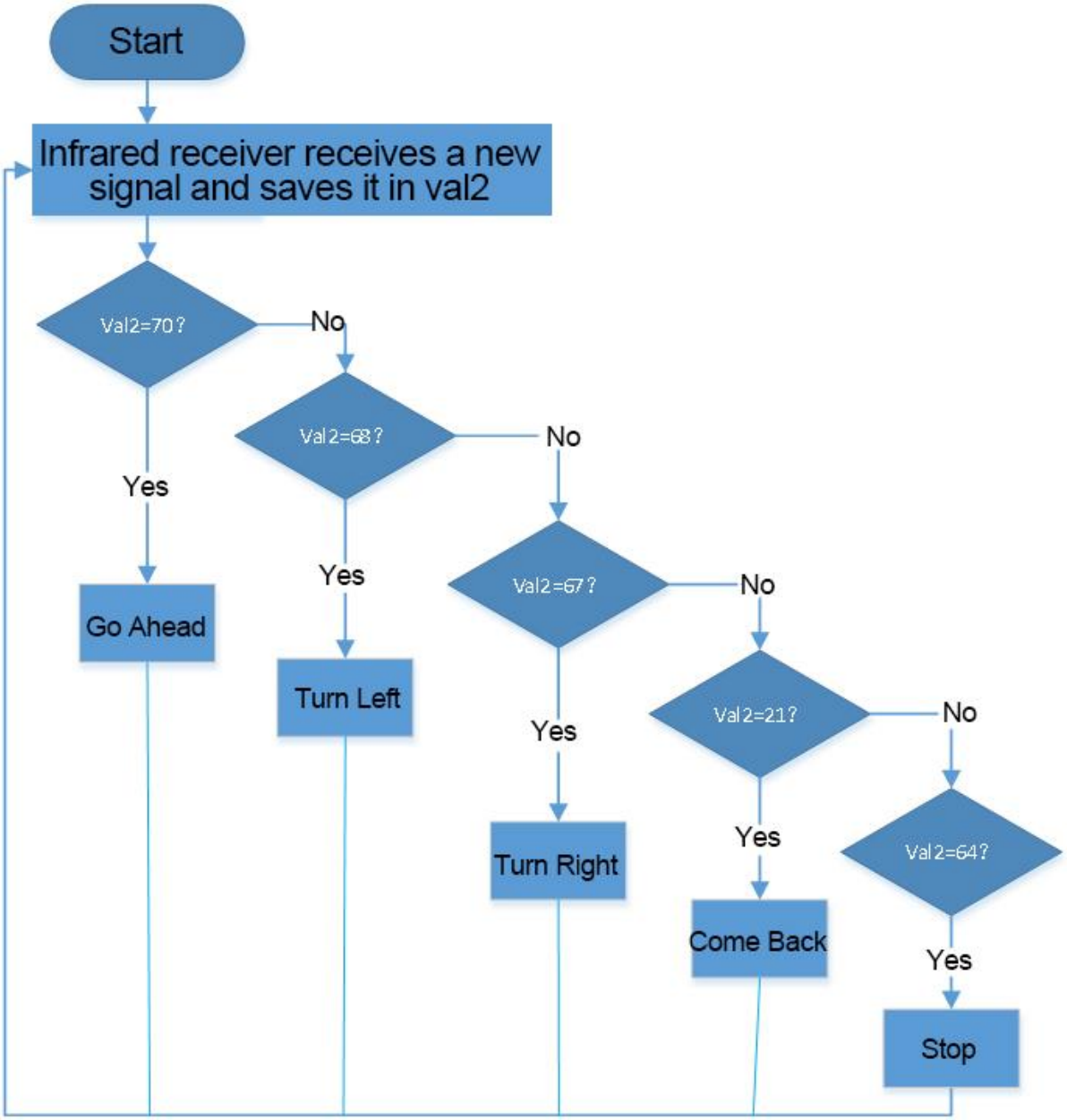
4.Press the button towards the infrared receiver of the robot, the View Data Window will display the value corresponding to the key of the IR remote control.



3.61 IR Remote Control Robot

We are going to make an IR Remote Control Robot. The robot car receives the infrared signal and decodes it to obtain specific values from the IR remote control, and the car performs tasks based on the corresponding values.

Working principle：

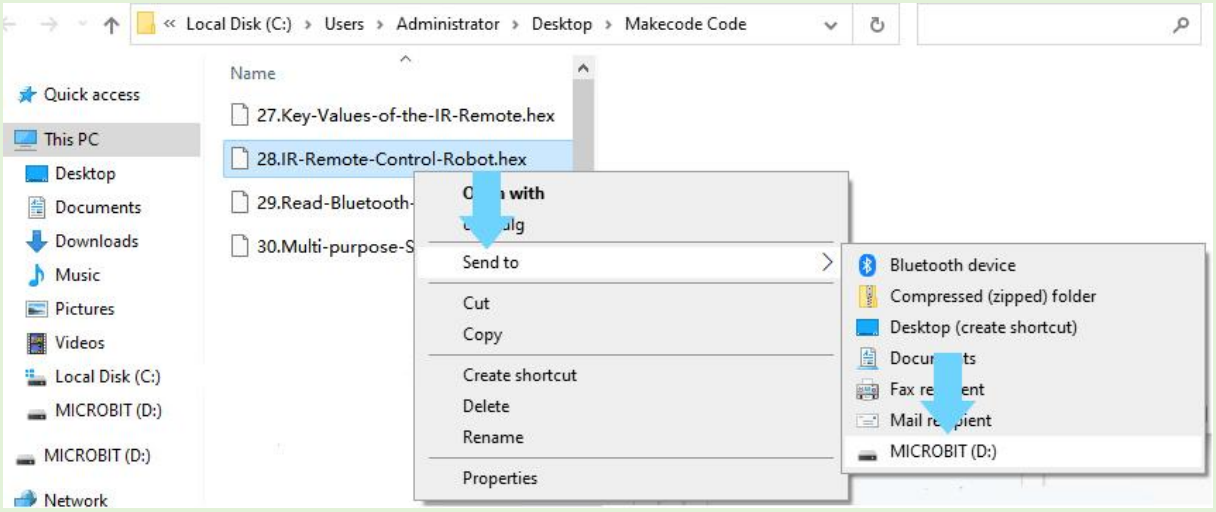


Steps:

- Connect your micro:bit to your computer with a USB cable.
- Turn on the POWER DIP switch on the car base.
- Find the downloaded IR-Remote-Control-Robot.hex file, drag it in the Makecode to view and analyze, and use WebUSB to flash it to the micro:bit from inside editor.

Or you can right-click on it and choose "Send to→MICROBIT."

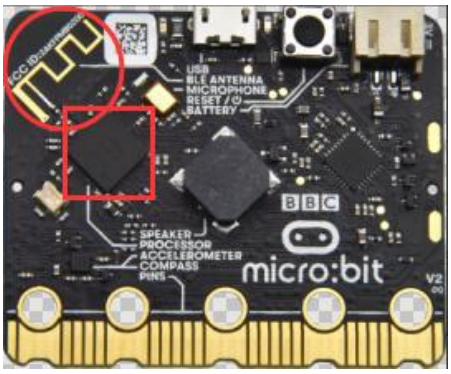
File Type	Path	File Name
hex file	.../Makecode/Makecode Code	28.IR-Remote-Control-Robot



Result: After the code is downloaded successfully, we can use the four direction keys of the infrared remote control to control the movement of the robot. Press the **OK** button and the robot will stop moving.

Note: The infrared remote control needs to send a signal towards one of the infrared receivers of the robot. The distance should not exceed about 5 meters. We configure an infrared receiver before and after the car base to make it easier for the robot to receive infrared signals.

3.62 Install Bluetooth APP and Read Bluetooth Data



We can use the built-in Bluetooth on the micro:bit to communicate with the Bluetooth APP of the mobile phone, and use the Bluetooth APP of the mobile phone to control the external devices of the micro:bit. The built-in Bluetooth on Micro:bit supports both Android phones and IOS devices (mobile phones or iPads).

In this step, we mainly introduce the Bluetooth APP we developed and read the data of the Bluetooth APP.

First we need to download the code to the Microbit.

Steps:

- Connect your micro:bit to your computer with a USB cable.
- Turn on the POWER DIP switch on the car base.
- Find the downloaded Read-Bluetooth-Data.hex file, drag it in the Makecode to view and analyze, and use WebUSB to flash it to the micro:bit from inside editor.

File Type	Path	File Name
hex file	.../Makecode/Makecode Code	29.Read-Bluetooth-Data

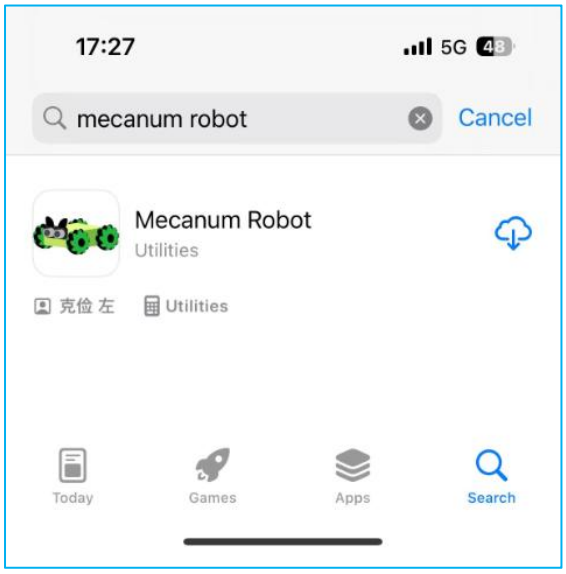
Next we will need to download the Bluetooth App for our phone and connect it to the Microbit.

Download APP and connect the bluetooth:

For iOS system device

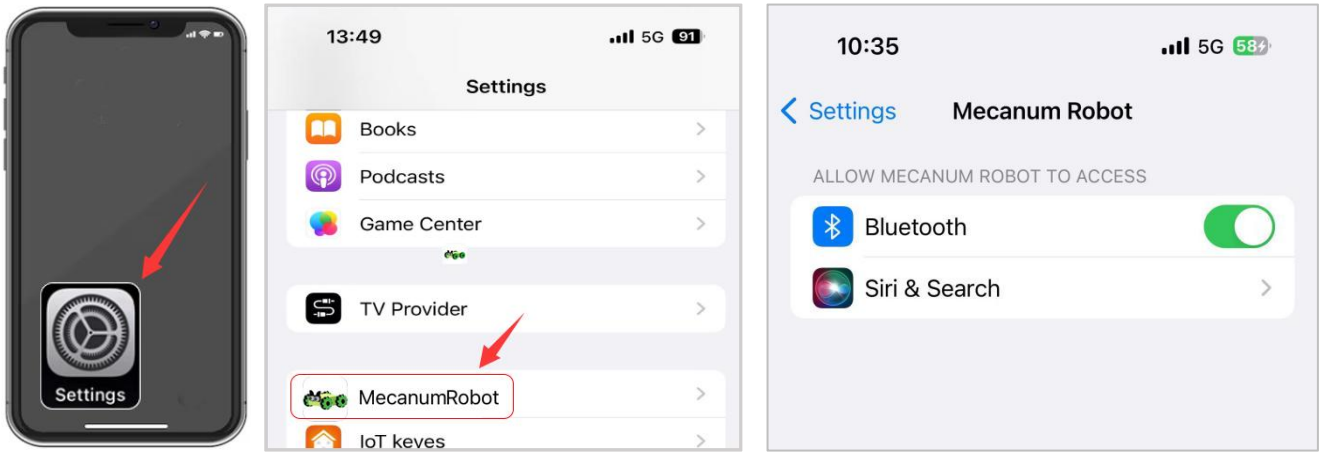
1. Open App Store

2.Search **mecanum robot** in the Apple Store and click download.



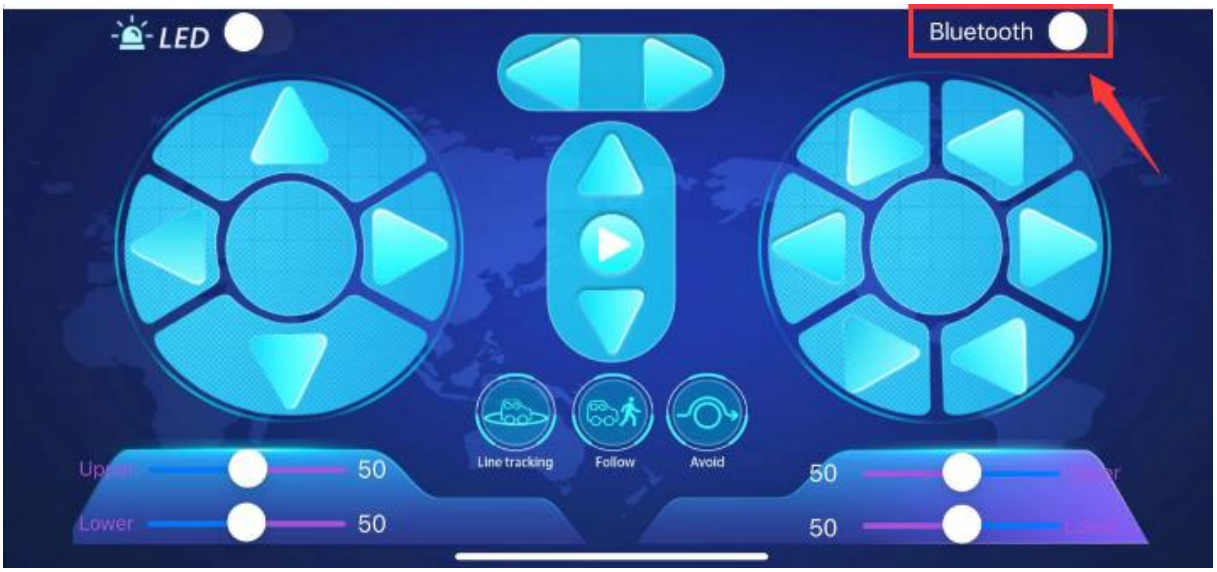
3.Click **Settings**, and turn on the **Bluetooth** and **location** services.

4.Click **Settings,** scroll through the list and find **mecanum robot**. Click it and allow Mecanum Robot to access Bluetooth.



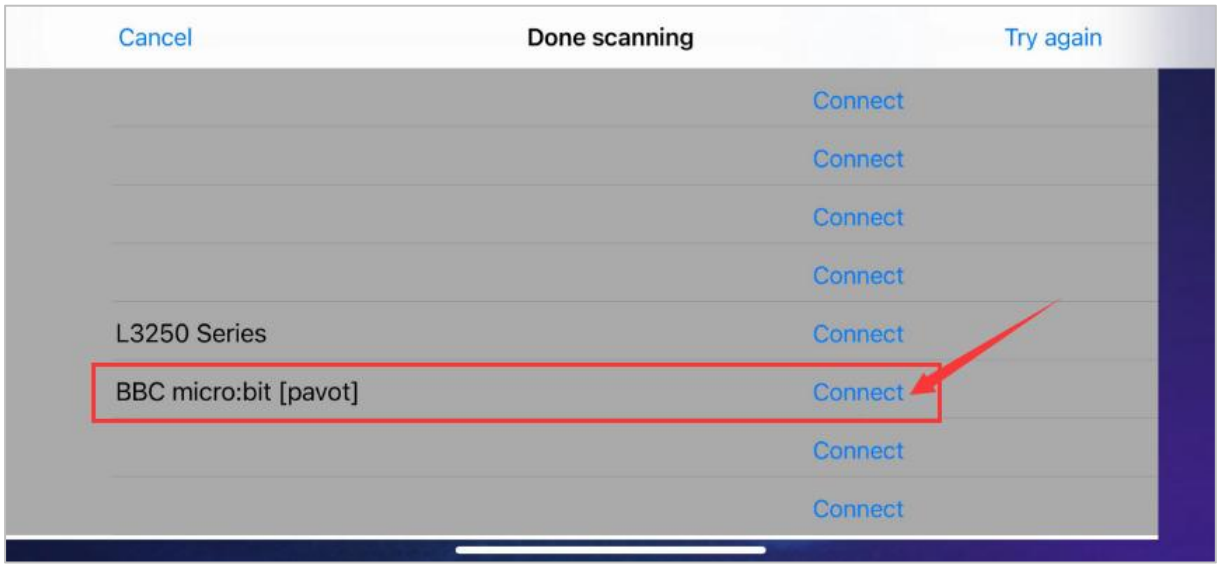
5.After the mecanum robot app is installed, you will see the following icon on your phone desktop. Click to open it.

6.Click the **Bluetooth** button in the upper left corner of the APP homepage.



7. After clicking the **Bluetooth** button, a Bluetooth list will pop up, scroll through the list and find **BBC Micro:bit**, click on the **Connect**.

Note: If a pop-up window prompts "failed to connect", please ignore it and start using this APP.



8. If after clicking the Bluetooth button, the pop-up page is blank and shows that Bluetooth not turned on, please exit the APP and restart it and try again.

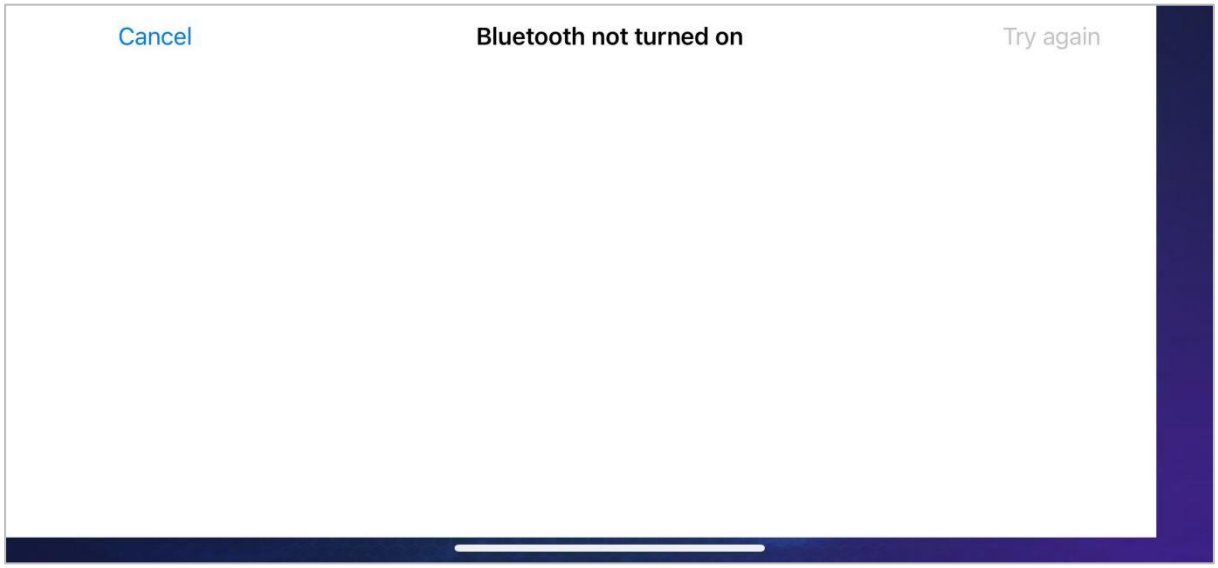
You also need to make sure you have completed the following steps.

1.Code downloaded to Microbit

2.The robot's battery power is sufficient and the power switch on the car base has been turned on.

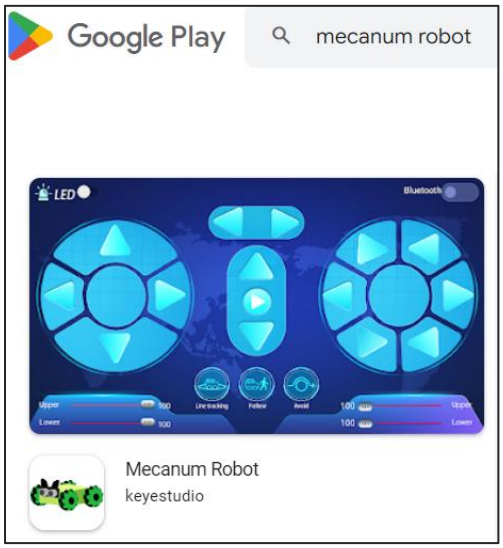
4.Turn on the Bluetooth and location services on phone.

5.Allow the **mecanum robot APP** to access Bluetooth.



For Android system device

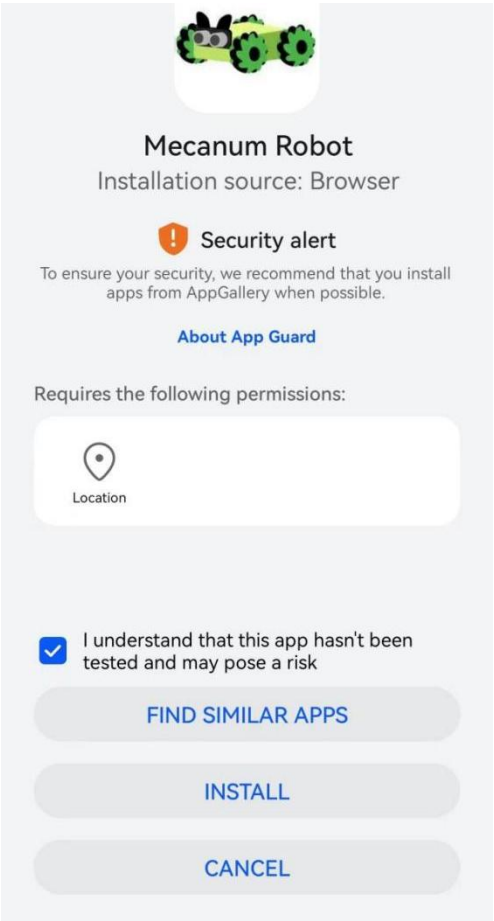
1. Search **mecanum robot** in Google Play.



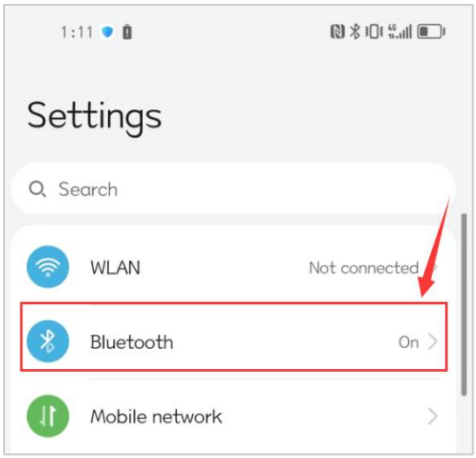
2. Click **INSTALL**

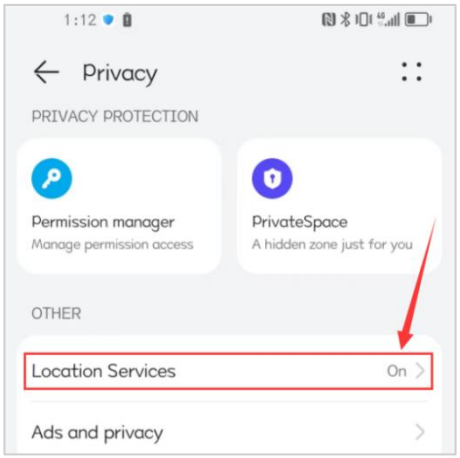
A security warning pop-up will appear. We guarantee that the APP is safe, please use it with confidence.

-**Select** I understand that this app hasn't been tested and may pose a risk

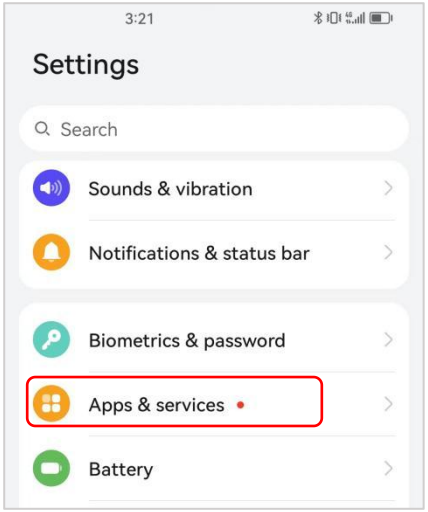


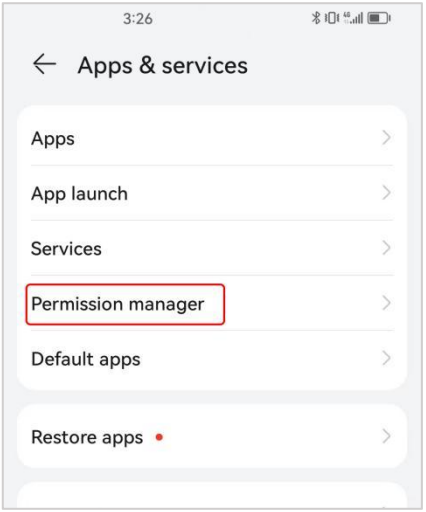
4.Click **Settings** of the mobile phone, and turn on the Bluetooth.
Click **Privacy**, and turn on the location services.



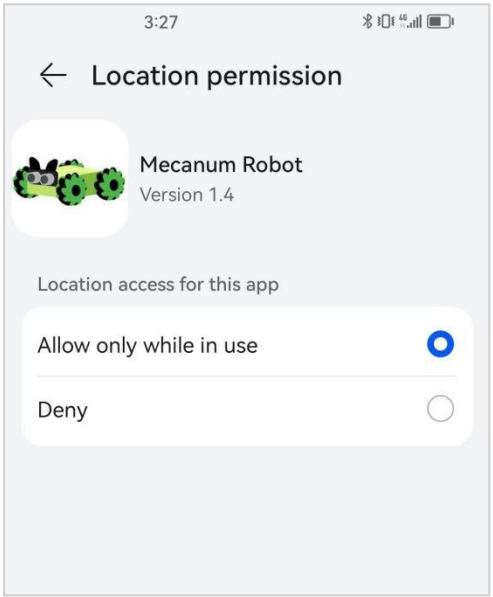


5.Click **Settings** > **Apps & services** > **Permission manager**



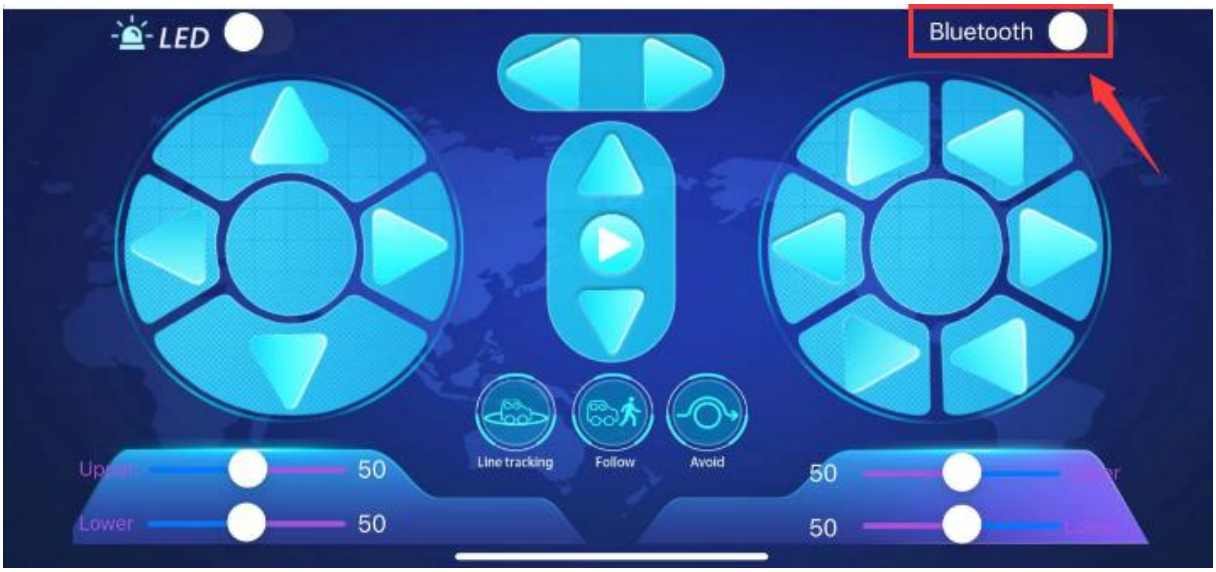


1. Find the **mecanum robot** Bluetooth app, click on the permission options of the app, and enable Location and nearby device permissions.(**Note:** Some mobile phones do not have nearby device permissions function.)



7.After the mecanum robot app is installed, you will see the following icon on your phone desktop. Click to open it.

8.Click the **Bluetooth** button in the upper left corner of the APP homepage.

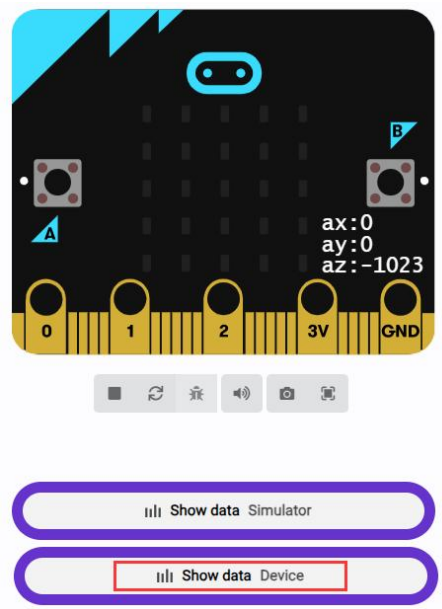


9.After clicking the **Bluetooth** button, a Bluetooth list will pop up, scroll through the list and find **BBC Micro:bit**, click on the **Connect**.

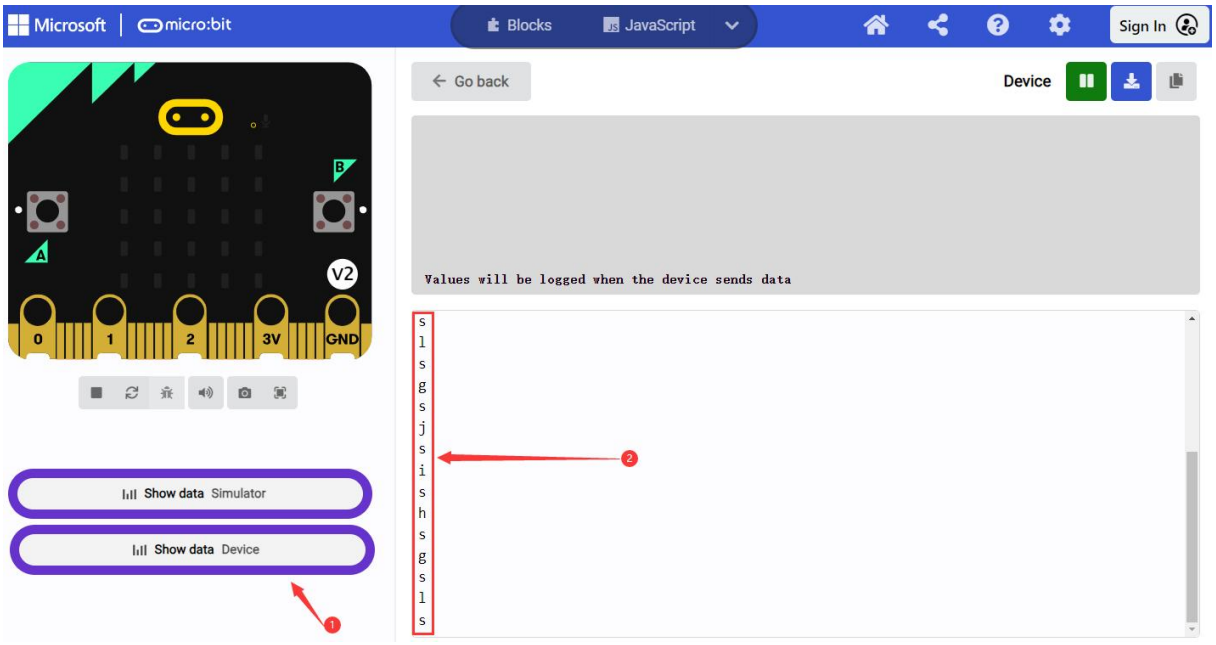
● After uploading the following code to the Microbit and connecting it to the Bluetooth APP.

File Type	Path	File Name
hex file	.../Makecode/Makecode Code	29.Read-Bluetooth-Data

● Click the "Show Data Device" under the simulator



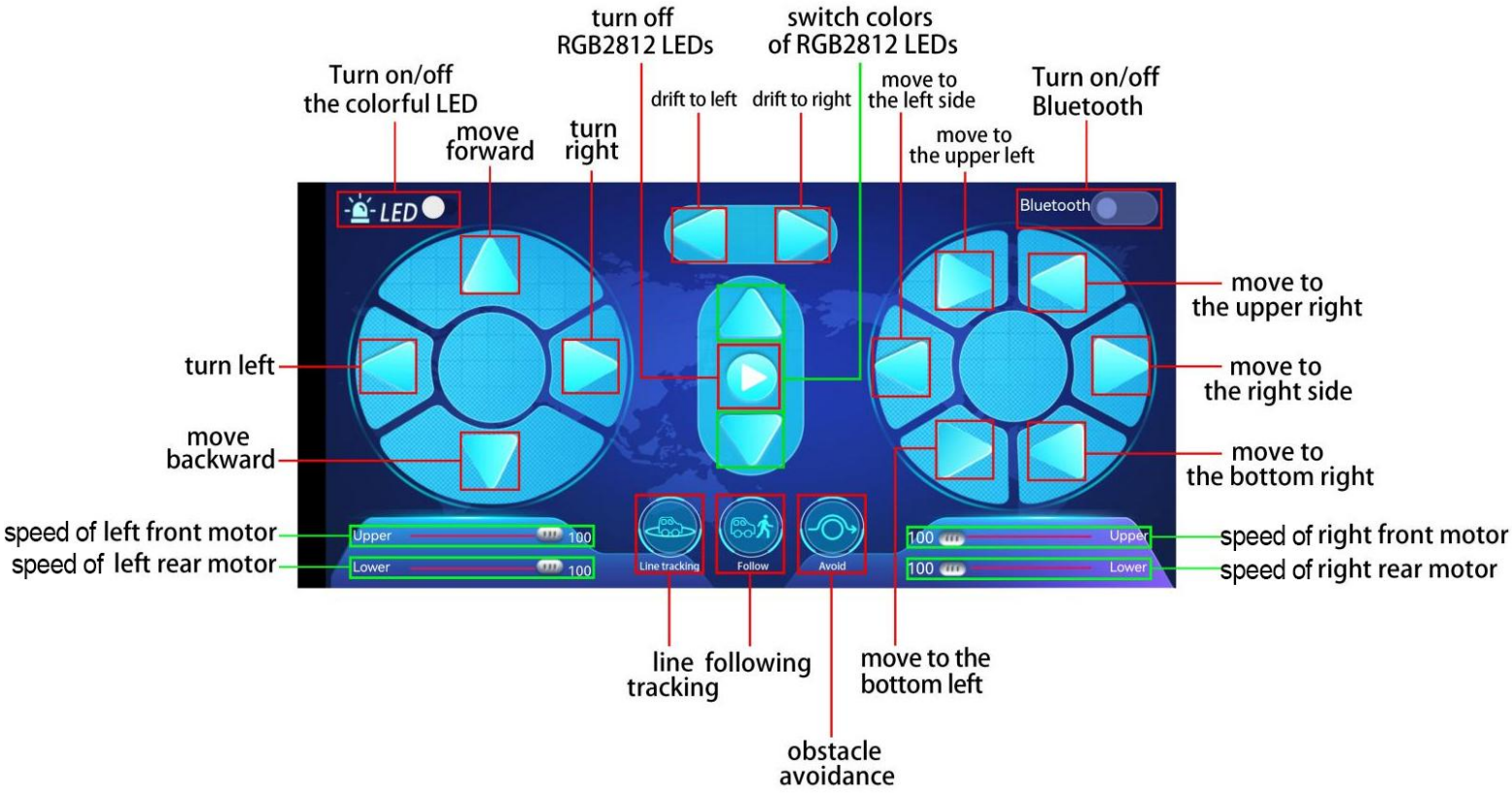
Click the button on the Bluetooth APP, the Data View Window will display its corresponding letter value, and "S" will be displayed when releasing the button.



3.63 Multi-purpose Smart Car

In the previous chapter, we learned how to install and use Bluetooth APP. In this step, we will use Bluetooth APP to control the robot.

The following are the functions corresponding to the buttons on the Bluetooth APP interface.

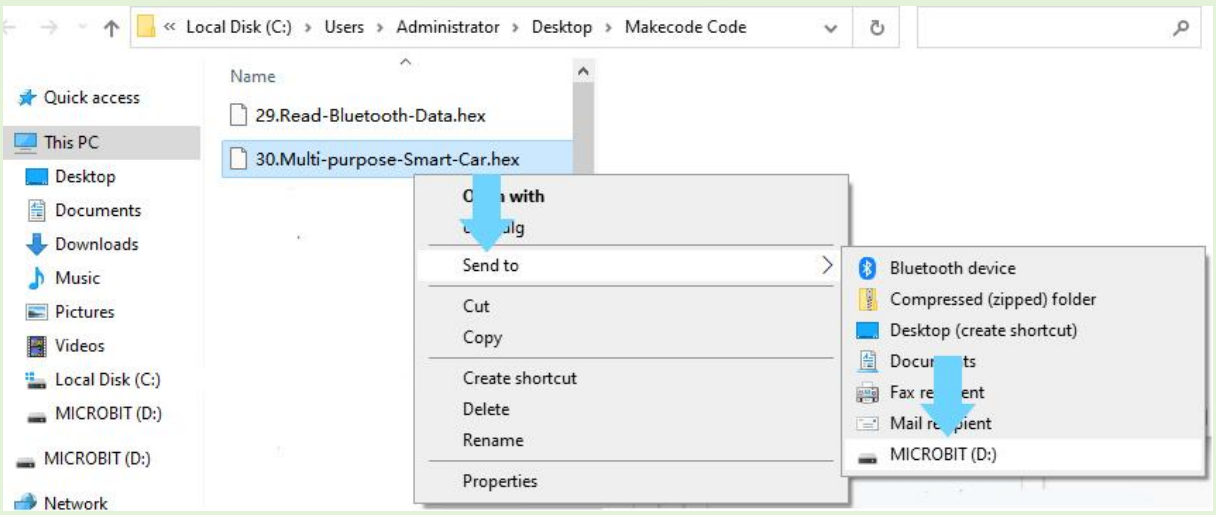


Steps:

- Connect your micro:bit to your computer with a USB cable.
- Turn on the POWER DIP switch on the car base.
- Find the downloaded Multi-purpose-Smart-Car.hex file, drag it in the Makecode to view and analyze, and use WebUSB to flash it to the micro:bit from inside editor.

Or you can right-click on it and choose "Send to→MICROBIT."

File Type	Path	File Name
hex file	.../Makecode/Makecode Code	30.Multi-purpose-Smart-Car



● Result:

Place the robot on the ground and turn on the power switch on the car base, connect the Bluetooth APP to the microbit, then you can use the Bluetooth APP to control the robot.