Docs » ring:bit Car Kit V2 » 1. Introduction

1. Introduction

1.1. Introduction of Ring:bit car 2

The ELECFREAKS Ring:bit Car is a small DIY smart car driven by the BBC micro:bit and the ELECFREAKS Ring:bit. The Ring:bit extends the micro:bit's 3 GPIO ports and allow for different sensors and components to be easily attached to the micro:bit. A basic Ring:bit Car can be easily programmed to run autonomously, with a remote control, and even create rainbow beacons of light. Just add one of the many extensions available and your Ring:bit Car can do even more things like line and light following, obstacle avoiding, drawing and more!

Please note that the Ring:bit car version 2 has been upgraded. It is diffrent from the Ring:bit car version 1. All related documents refers to the Ring:bit car version 2. Information of the Ring:bit car version 1

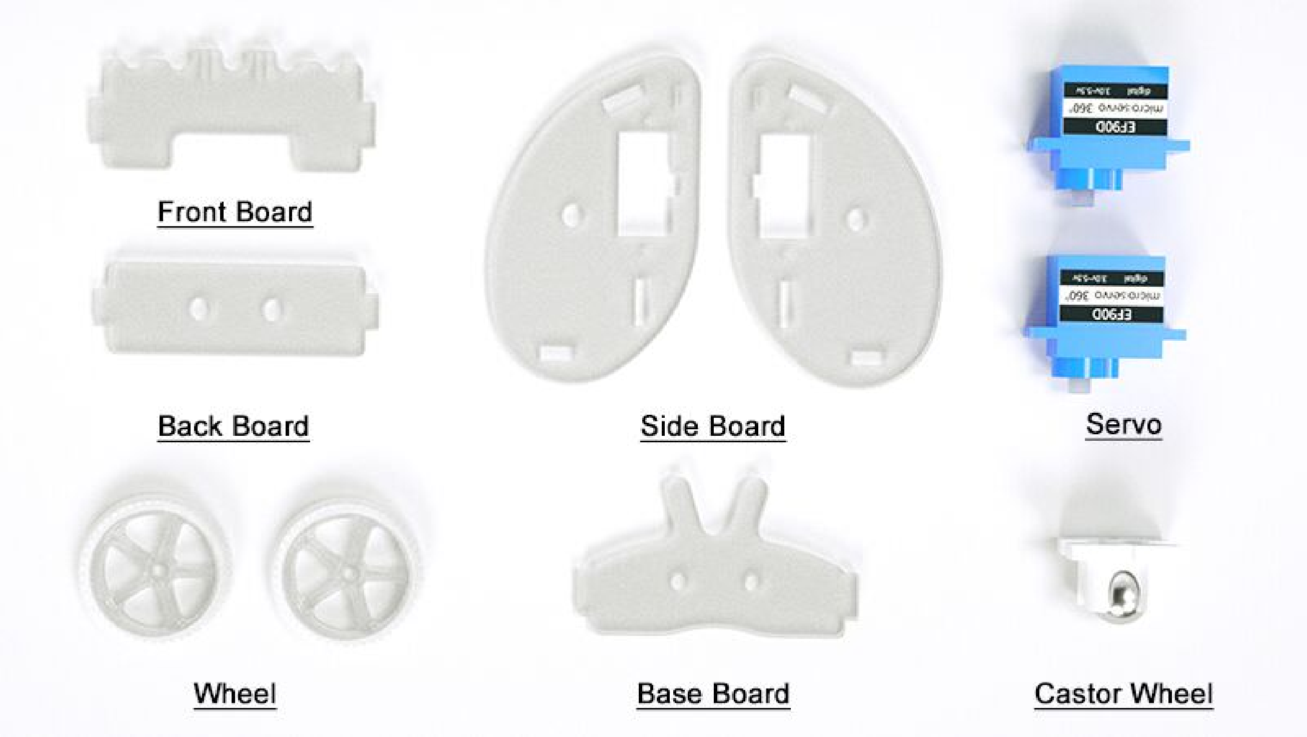
1.2. Components list

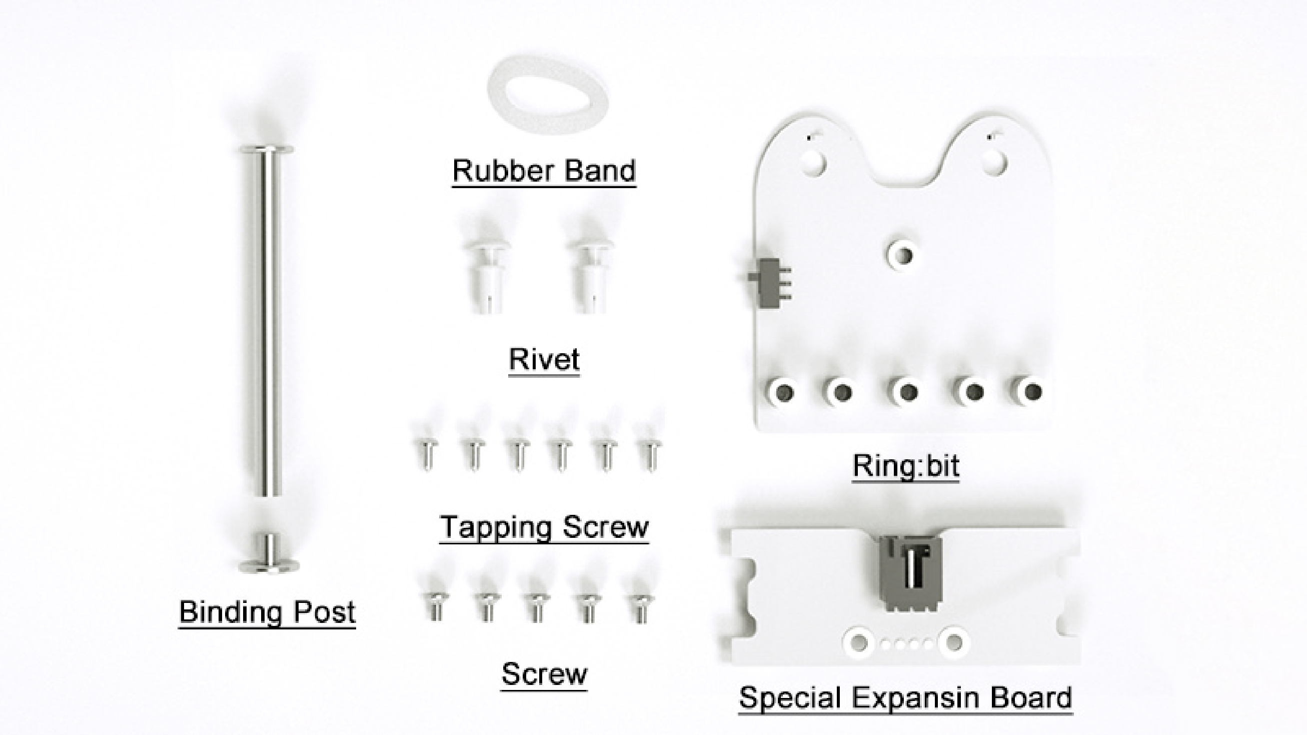
Components	Numbers	Pictures

Components	Numbers	Pictures
micro:bit	Optional	
Ring:bit Expansion Board	1	
Ring:bit Car Expansion Board	1	
Ring:bit Car Front Board	1	
Ring:bit Car Back Board	1	
Ring:bit Car Base Board	1	
Ring:bit Car Side Board	2	
Ring:bit Car Wheel	2	
Binding Post	1	
360° Servo	2	
Castor Wheel	1	
Screw	5	
Tapping Screw	5	
Rivet	2	
Screwdriver	1	
Rubber Band	1	

2. Ring:bit Car Assembly Step

2.1. Ring:bit car 2 Components Lists

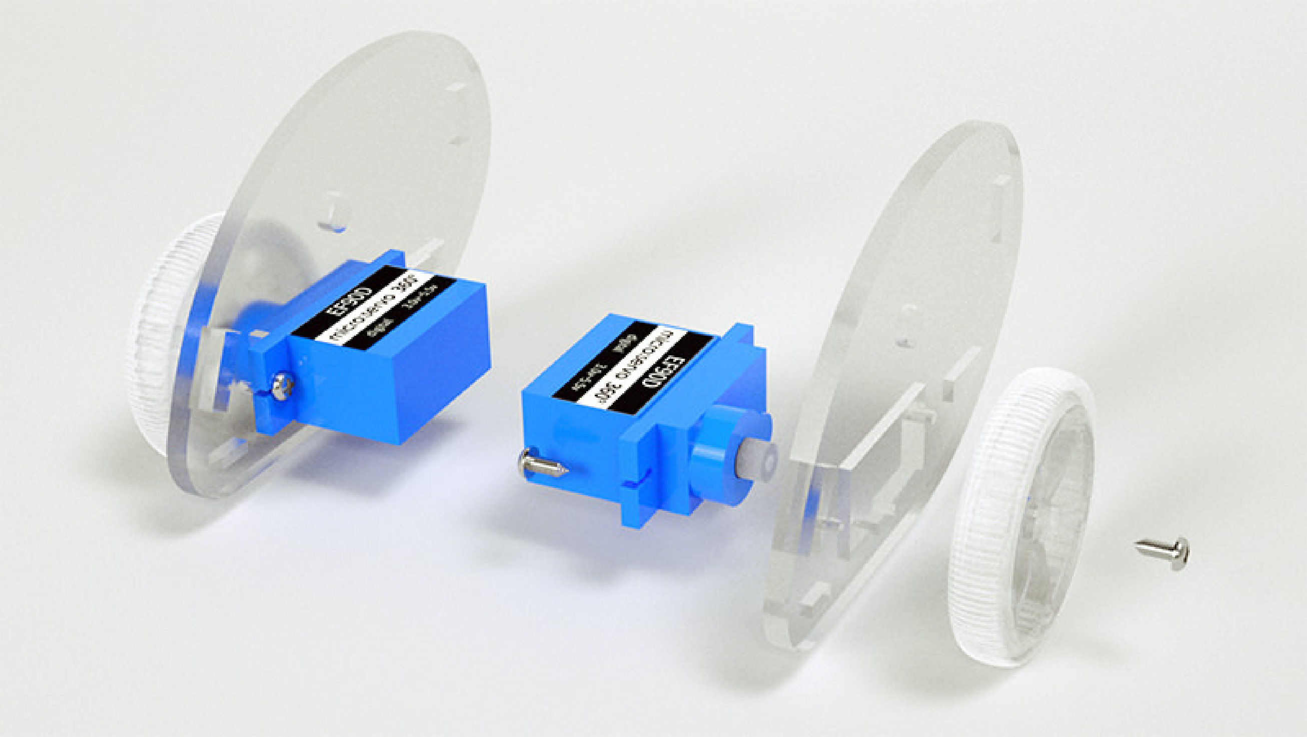




2.2. Assembly Step

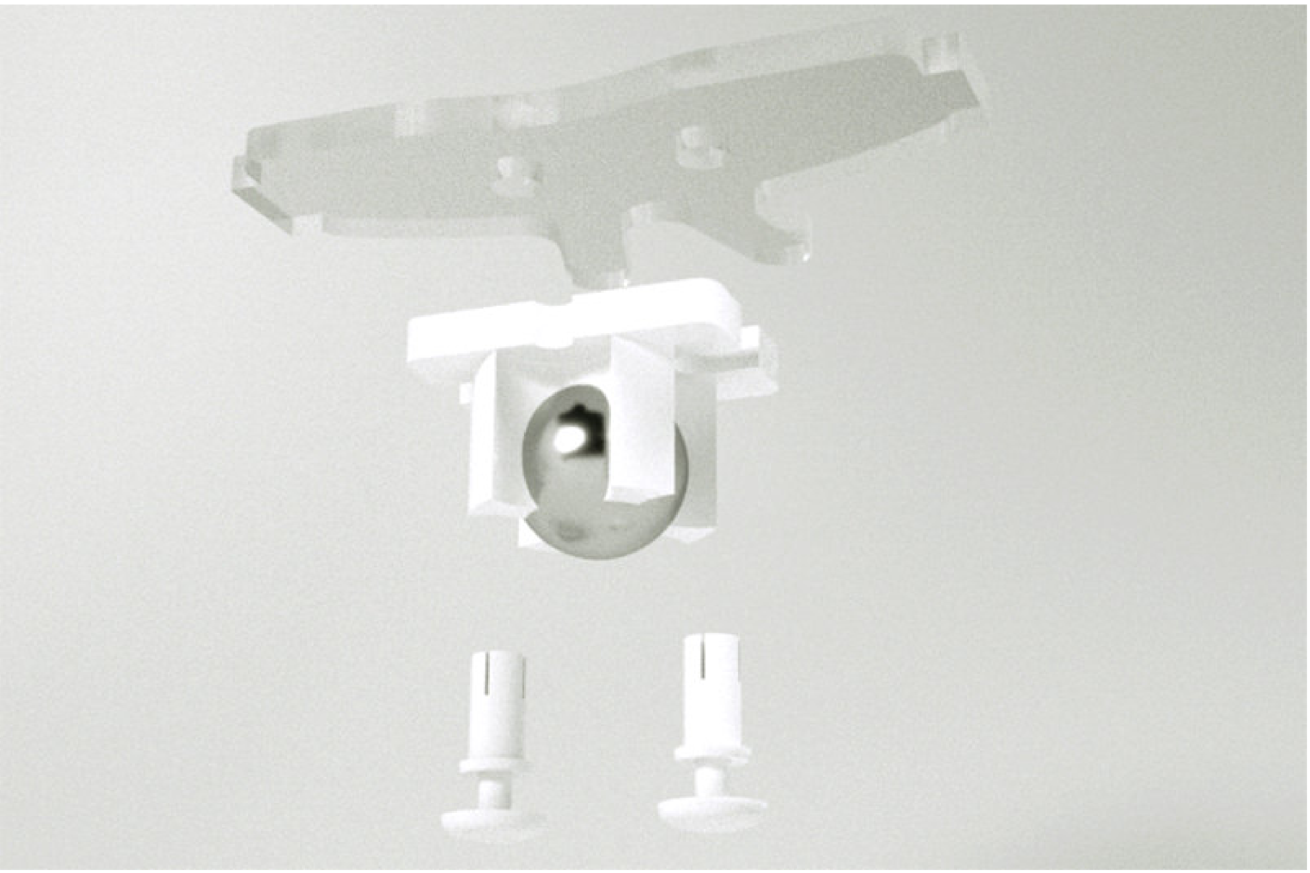
Step-1

- Fisrt, use the tapping screw to fix the servo to the side board.(as below pictures)
- Then, use the screw to fix the wheel to the servo.(as below pictures)



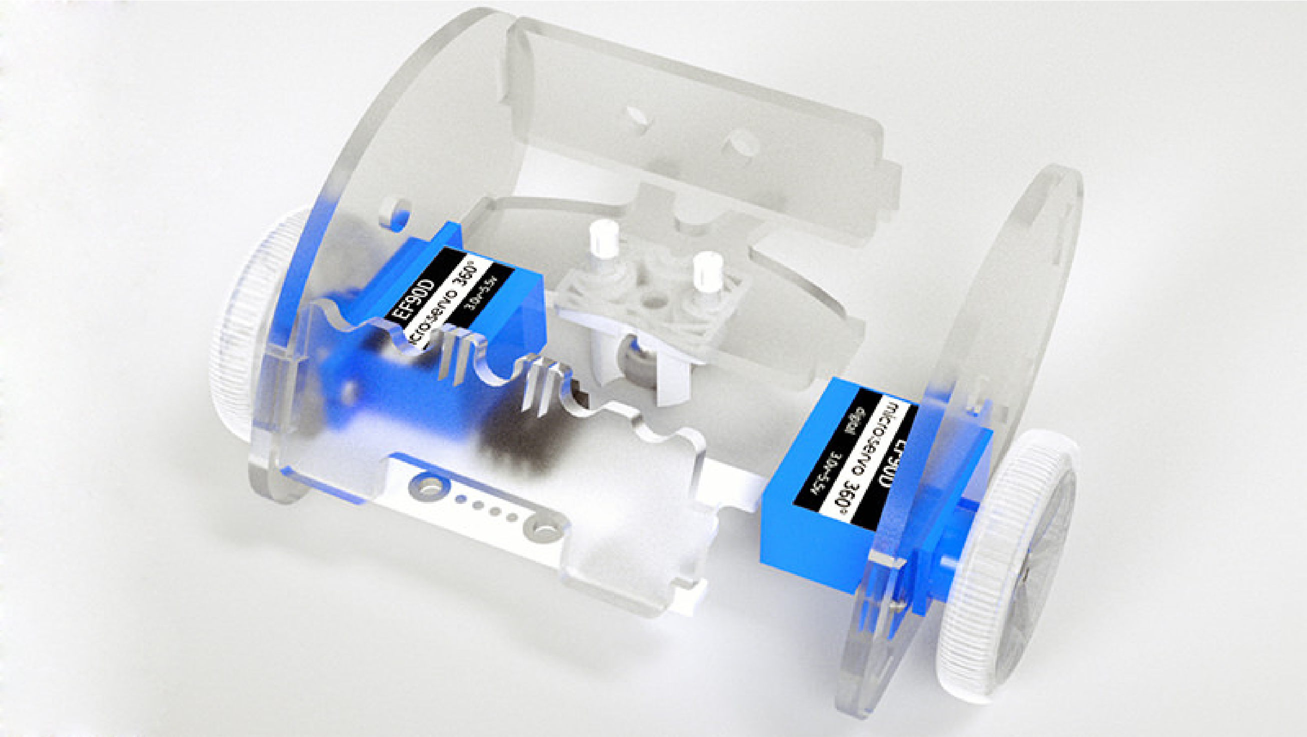
Step-2

- Use the rivet to install the castor wheel on the chassis.



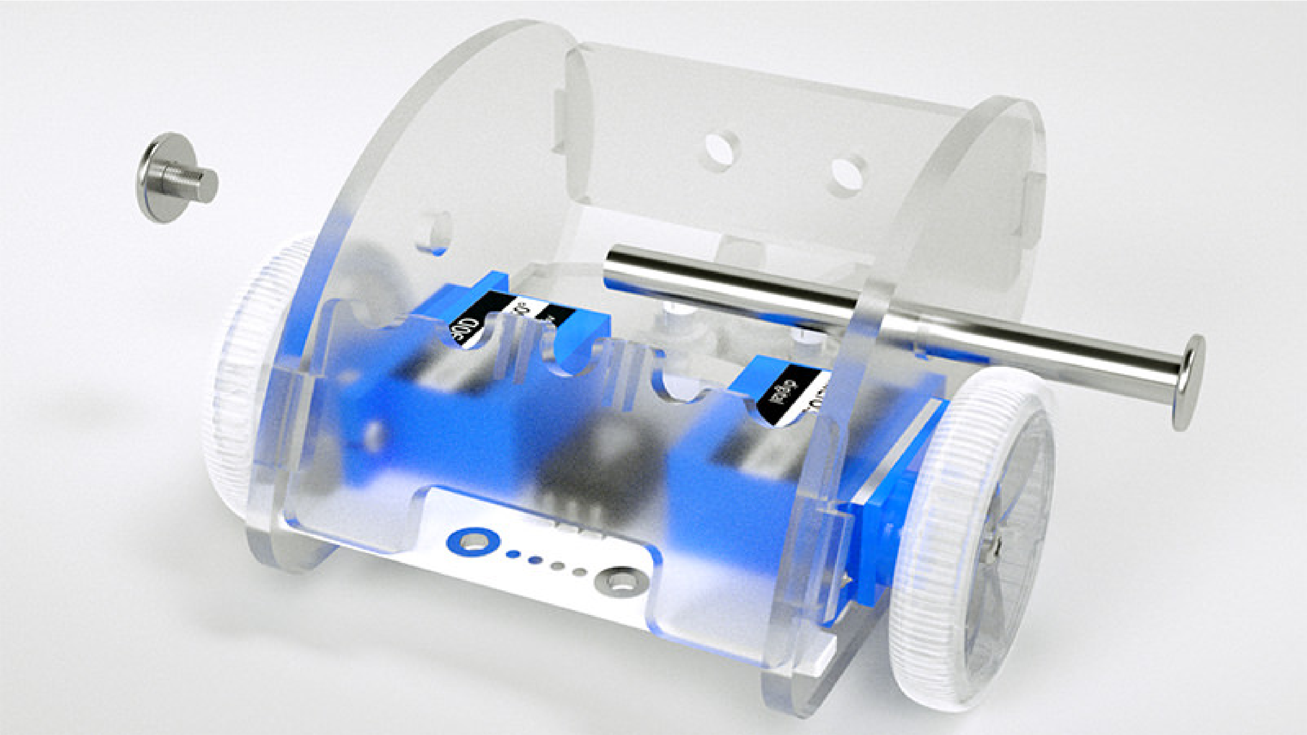
Step-3

- Assemble the front board, the back board and the base board to the side board as below picture.



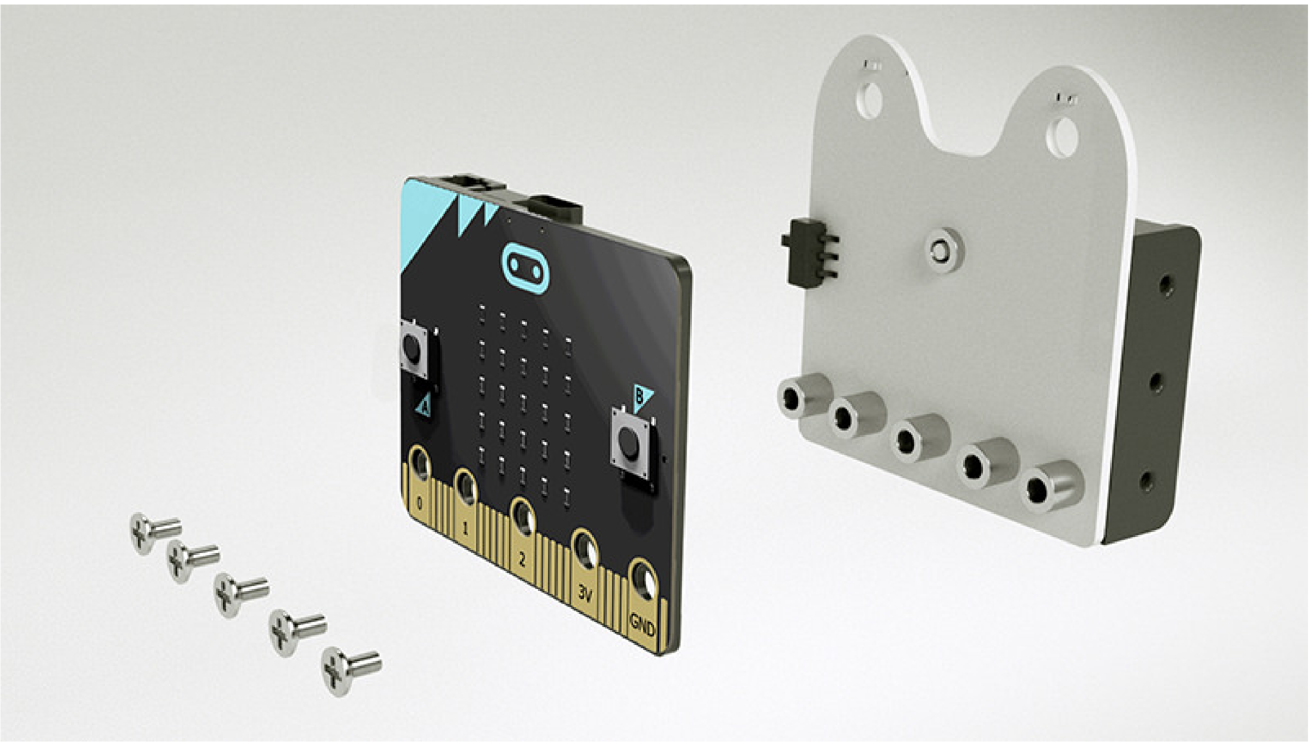
Step-4

- Use the binding post to fix all installed components.



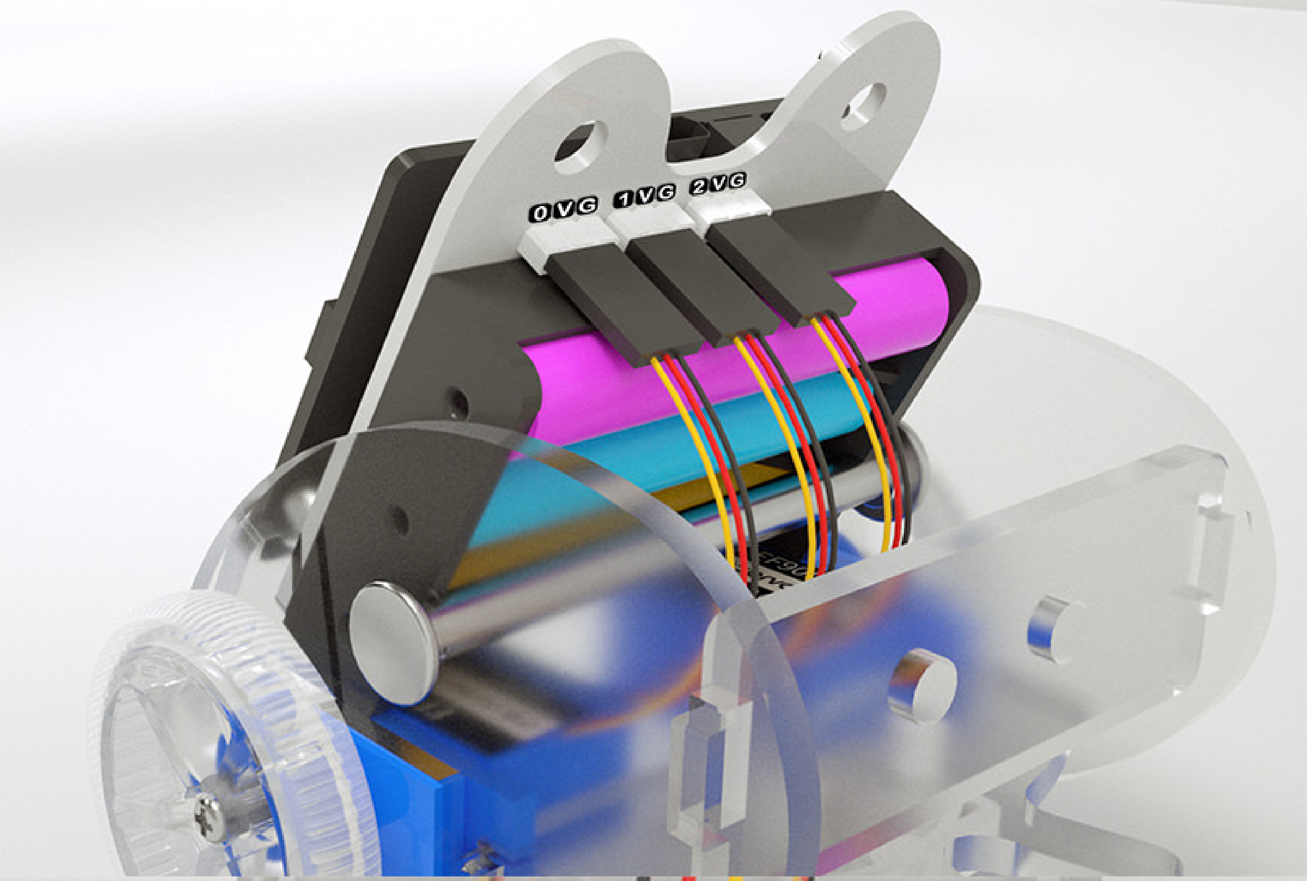
Step-5

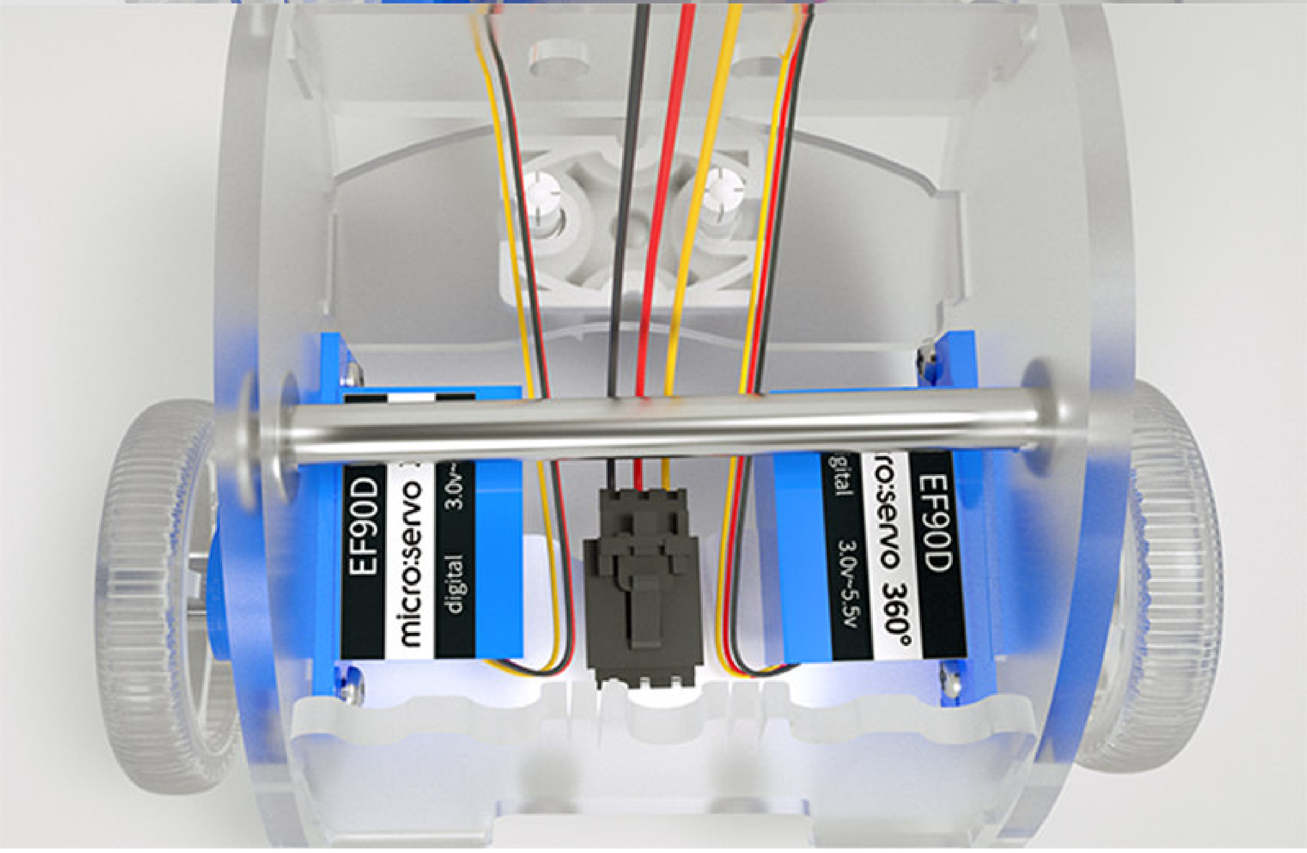
- Use the screw to fix the ring:bit to the micro:bit board.



Step-6

- Connecting wire as below pictures.





Completed

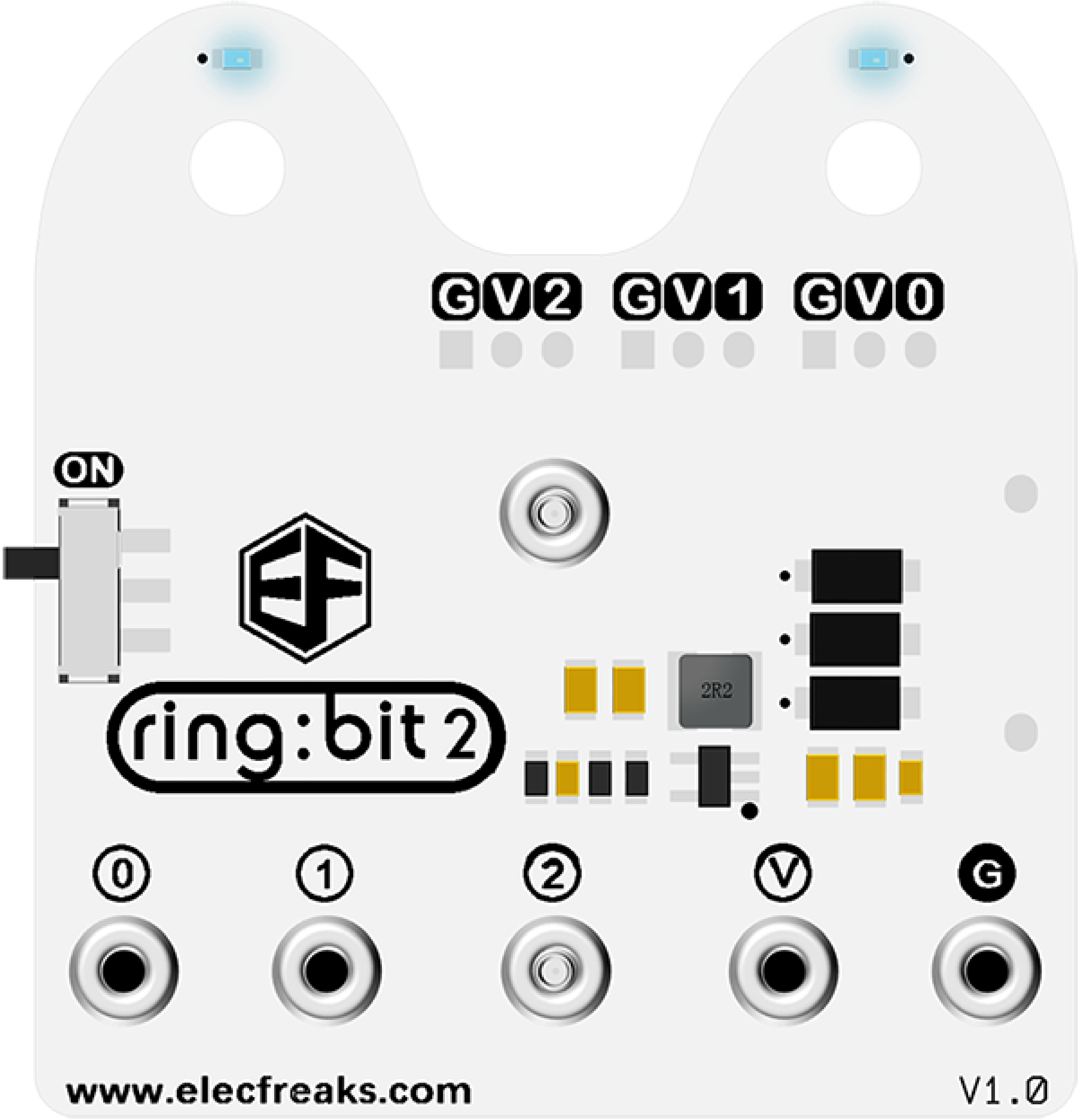
- Fix the assembled ring:bit to the front board.

A
B
0
1
2
3V
GND

3. Ring:bit Introduction

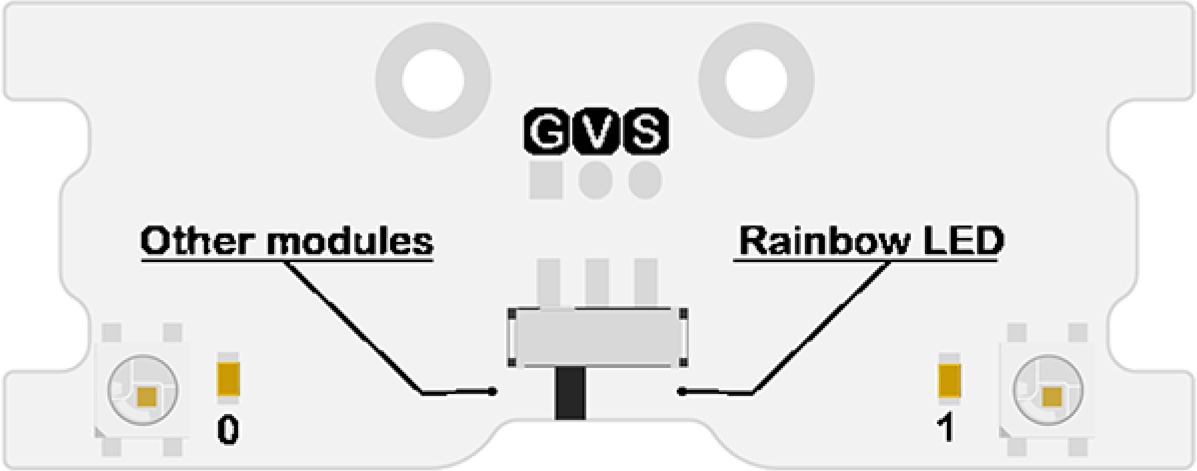
3.1. Expansion Board of Ring:bit car 2

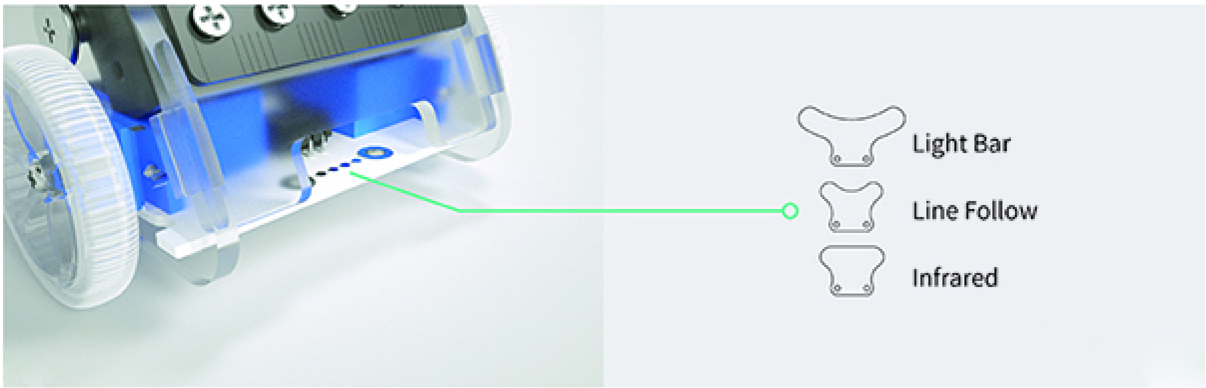
- The Ring:bit expansion board is a simple PCB for the micro:bit. The Ring:bit extends the micro:bit's 3 GPIO ports. We will use this module to convert the P0/P1/P2 port to the common GVS port. It can be loaded 3 AAA batteries to drive the car or other components.



3.2. The Special Expansion Board of Ring:bit car 2

- The Special Expansion Board of Ring:bit car 2 is for adding extendibility of Ring:bit car. With two Rainbow LEDs and a special expanded port for special expansion module connection.
- There is a slide switch on the board, slide it to “Rainbow LED” for using of two Rainbow LEDs; slide the switch to the “Other modules” for using other modules. Please note that these can not be operated at the same time.





Please note that the Ring:bit car version 2 has been upgraded. It is diffrent from the Ring:bit car version 1. All related documents refers to the Ring:bit car version 2. *Information of the Ring:bit car version 1*

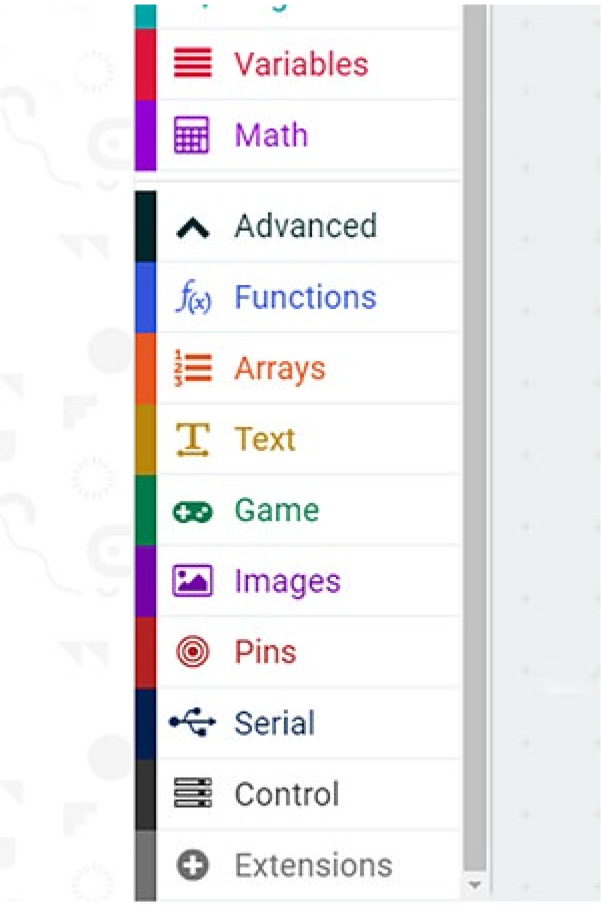
4. Ring:bit Introduction of E-blocks

4.1. Add New Extensions

- We'll need to add a package of code to be able to use our ring:bit car.
- Add new package by following below steps.

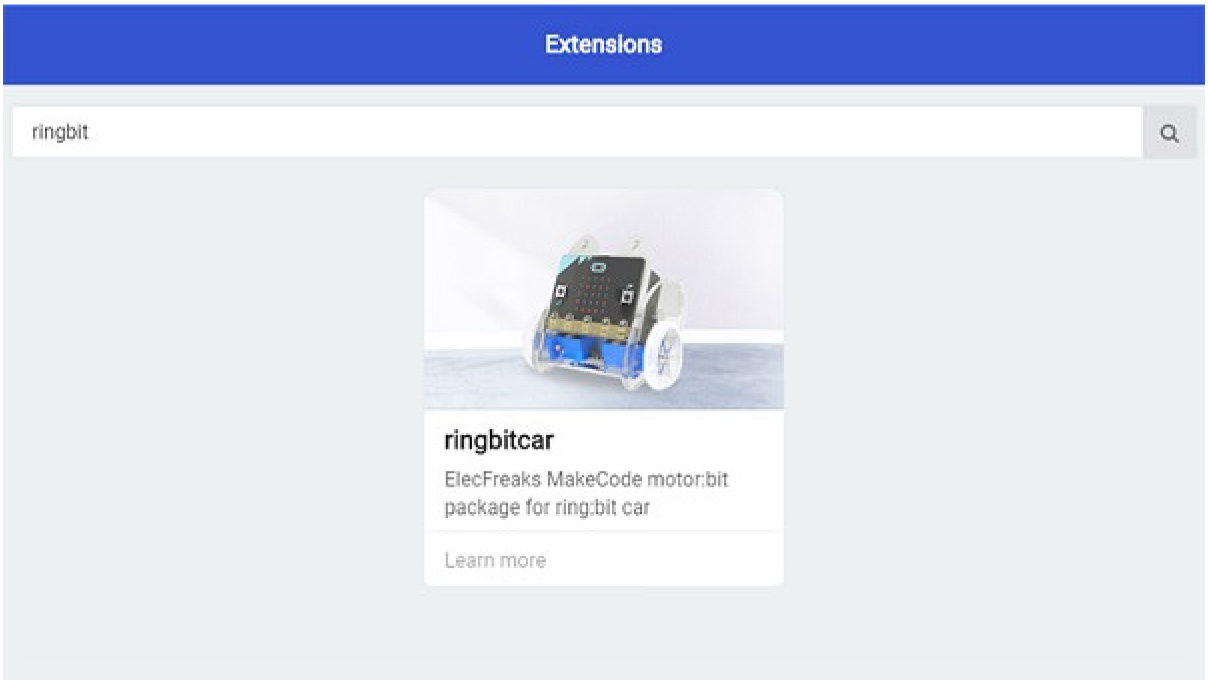
Step 1

- Click on "Advanced" in the Code Drawer to see more code sections and look at the bottom of the Code Drawer for " Extensions".



Step 2

- Search for "ringbitcar" and click on the ring:bit car package to add it to your project. (As below picture)

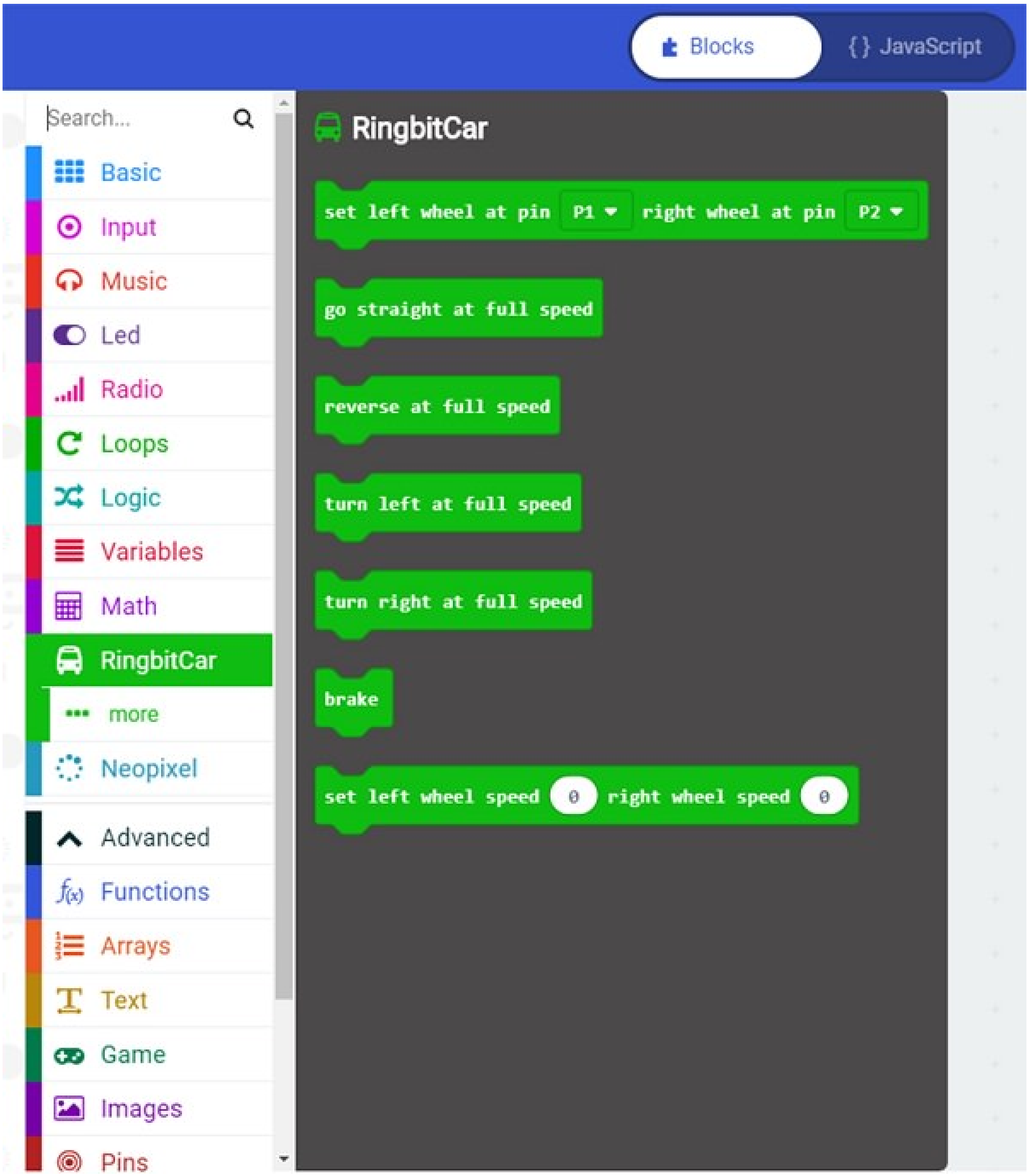


ringbitcar

ElecFreaks MakeCode motor:bit package for ring:bit car

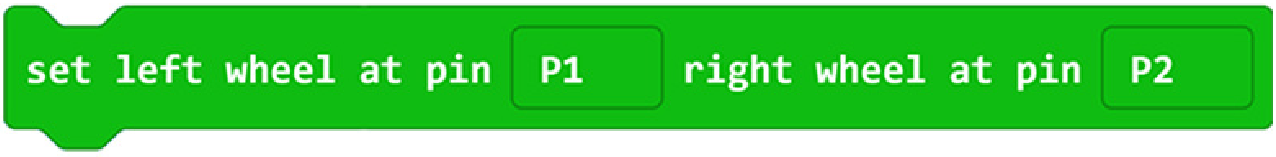
Step 3

- Completed.



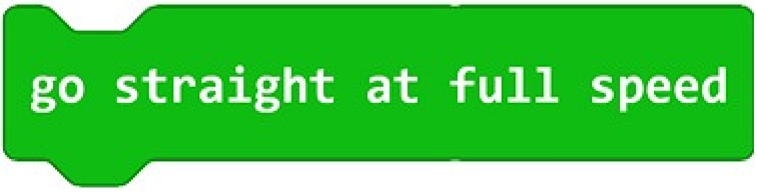
```
set left wheel at pin P1 ▾ right wheel at pin P2 ▾
go straight at full speed
reverse at full speed
turn left at full speed
turn right at full speed
brake
set left wheel speed 0 right wheel speed 0
```

4.2. Introduction



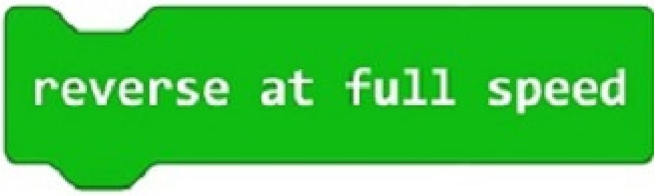
```
set left wheel at pin P1 right wheel at pin P2
```

- Pre-coding, we need to use the block to name the left wheel and the right wheel.



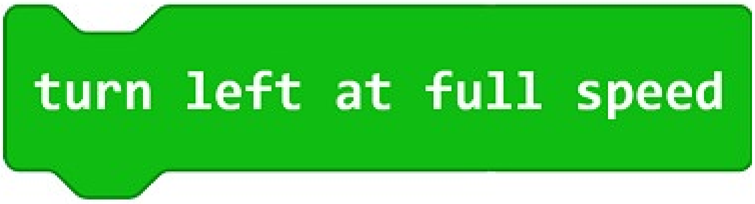
```
go straight at full speed
```

- Above block is for the car goes straight at full speed.



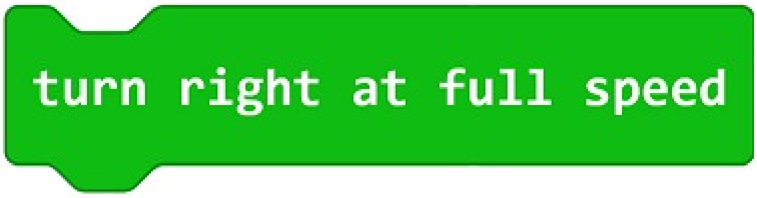
```
reverse at full speed
```

- Above block is for the car reverses at full speed.



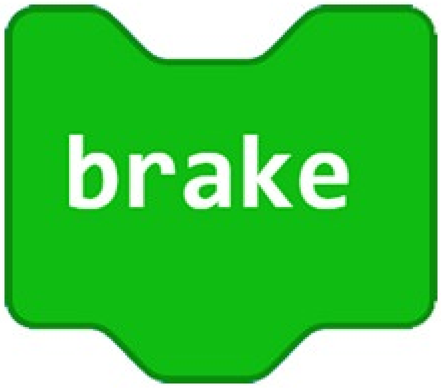
```
turn left at full speed
```

- Above block is for the car turns left at full speed.



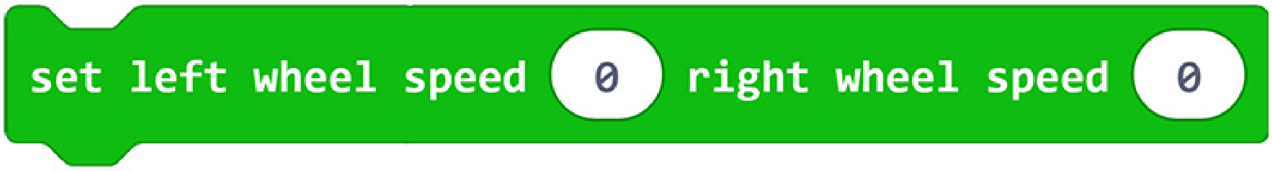
turn right at full speed

- Above block is for the car turns right at full speed.



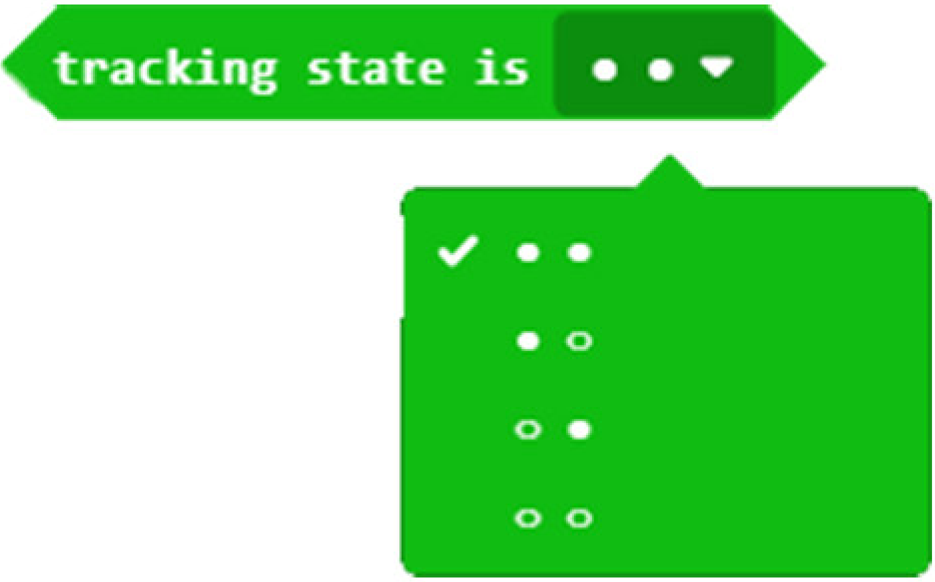
brake

- Above block is for the car brakes.

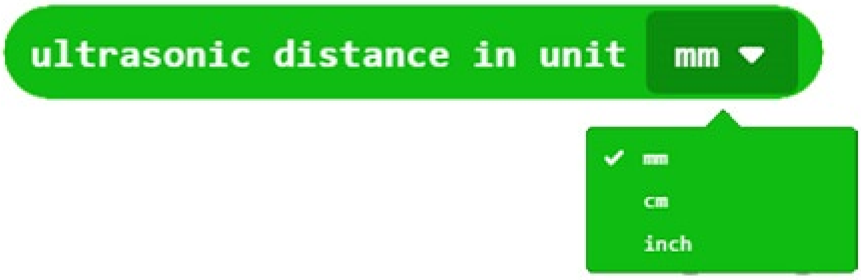


set left wheel speed 0 right wheel speed 0

- Above block is for setting the rotate speed for the left wheel and the right wheel of the car.



- This block can be used to judge the line follow status after you expands the line follow module.

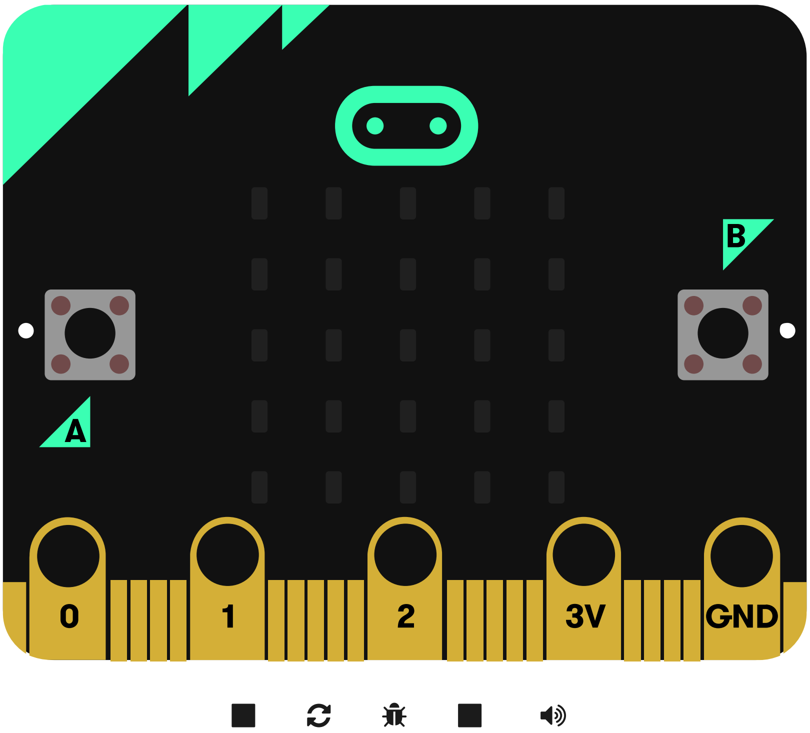


- This block can be used to receive the ultrasonic’s distance after you expands the ultrasonic distance module.

_``

Program

- Program Link： https://makecode.microbit.org/_MitdxCVTbJfz
- If you don’t want to type these code by yourself, you can directly download the whole program from the link below:



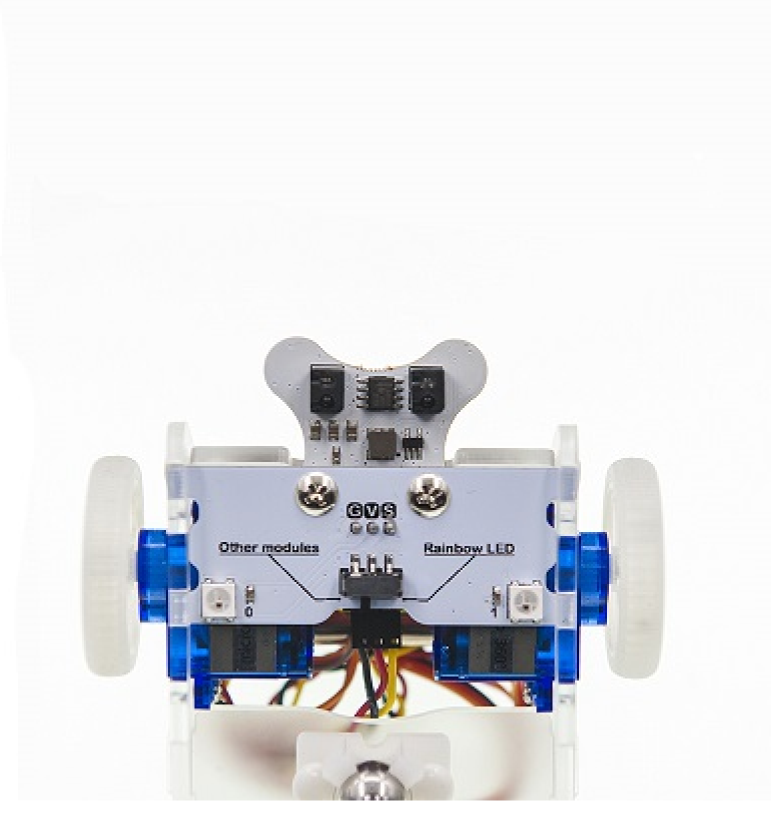
4.3. Questions

4.4. More Information

5. Line flow module

5.1. Introduction

- Ring:bit car V2 line following module is a dedicated module designed for the Ring:bit car V2, it is easy to be installed to achieve the line following function for the Ring:bit car V2.
- Equipped with double infrared probes, it can detect the distance between 2~12mm accurately that can achieve the function of line following around the circles, the detection of the black lines and the detection of the edge.



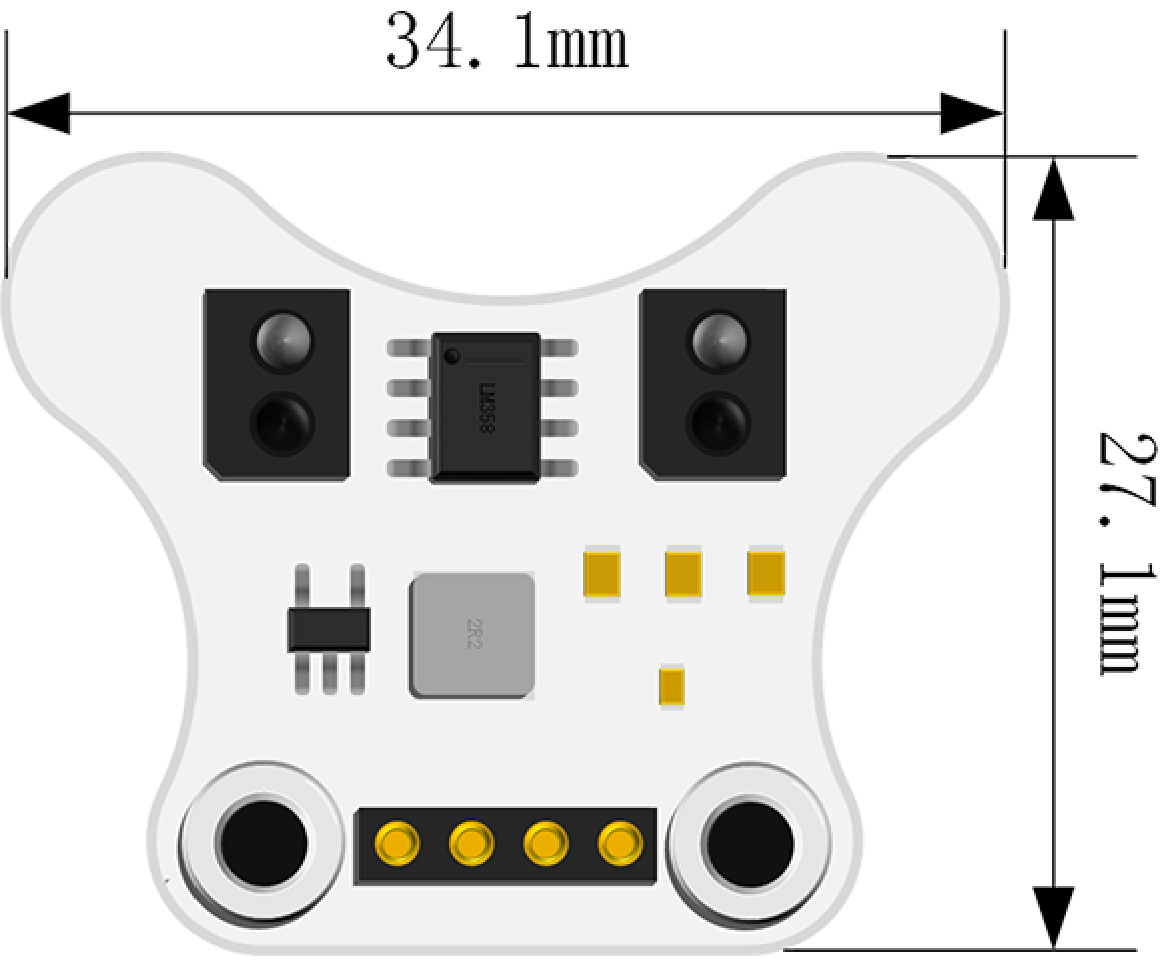
5.2. Features

- The micro:bit can drive it directly with the input voltage of this module in 3V~5V.
- It only needs one IO port it with standard 3-pin GVS ports.
- It uses the infrared light to detect with a strong anti-interference capability.

5.3. Parameter

Items	Parameter	Notes
Name	Ring:bit car V2 line following module	-
SKU	EF03424	-
Working Voltage	DC 3-5V	-
Port	Ring:bit car dedicated pin ports	Fixed by screws
Types of Output Signal	Simulation	-
Effective Distance	2~12mm	-
Dimension	34.15 x 27.20mm	-
Net Weight	4.7g	-

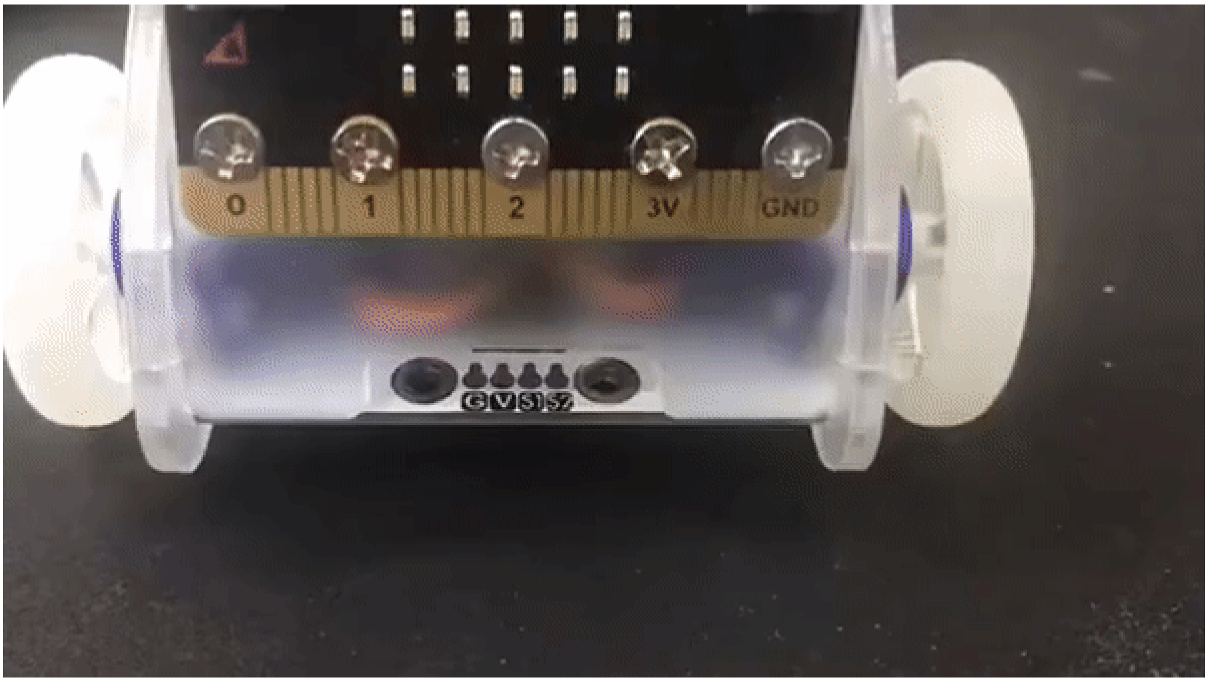
5.4. Dimensions:



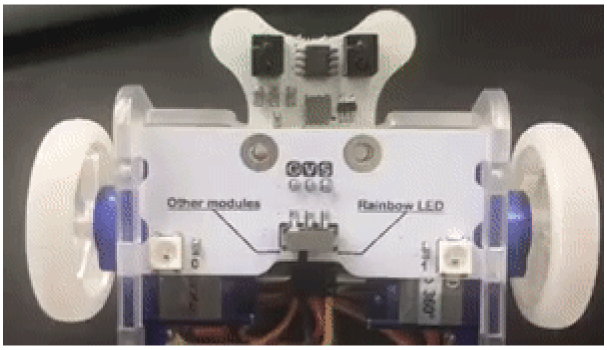
5.5. Quick to Start

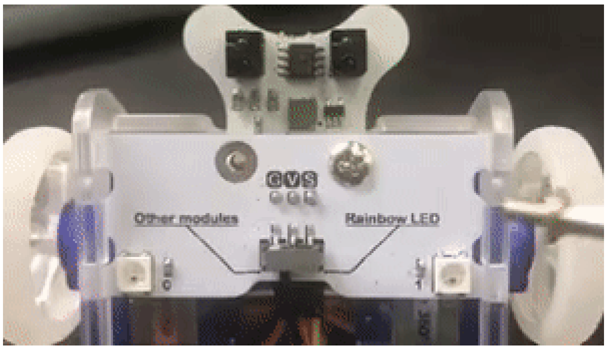
Hardware Connection

- The first step is to insert this module to the baseboard of the Ring:bit car V2.

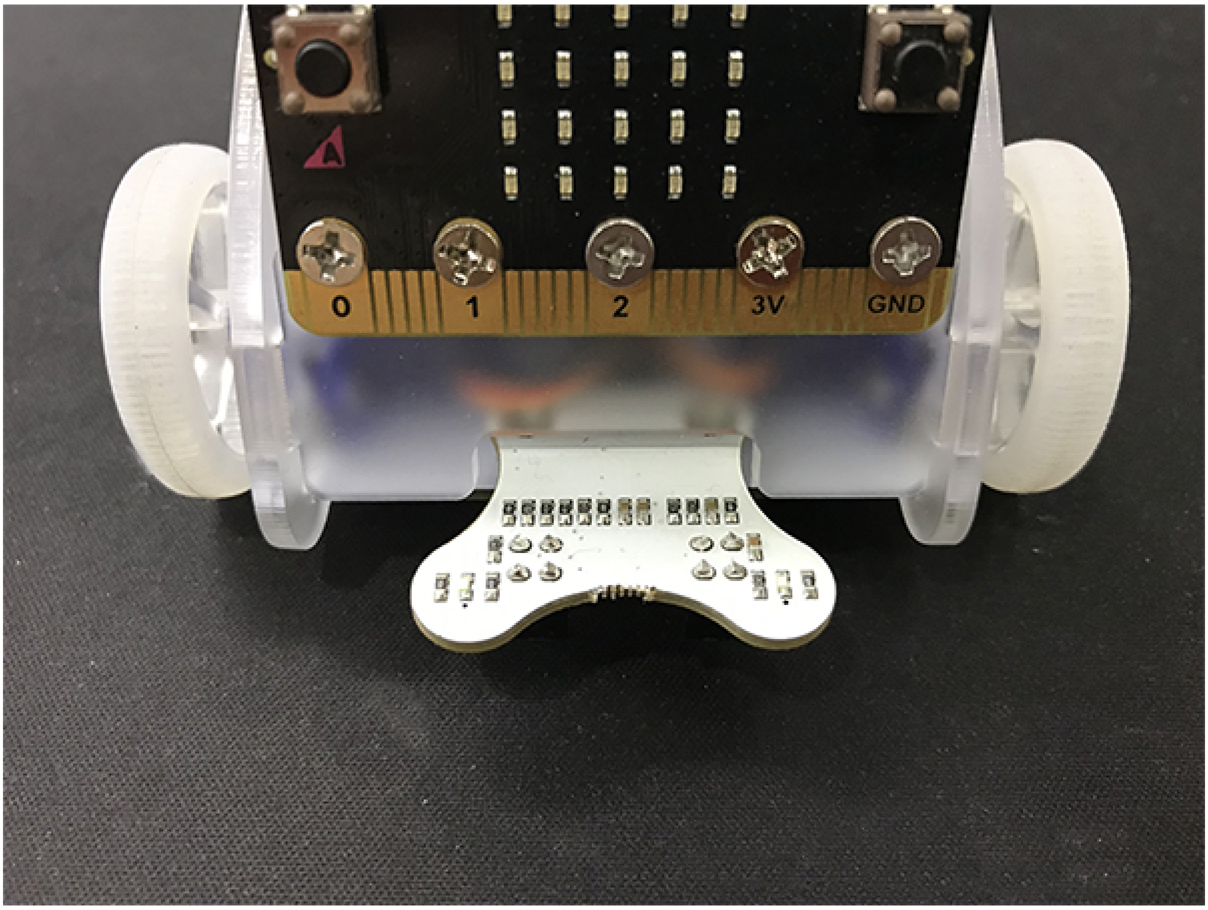


- The next step is to fix the module with the two screws.



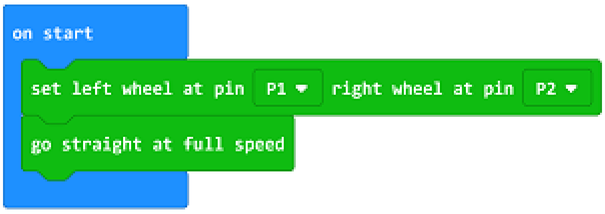


- Completed.

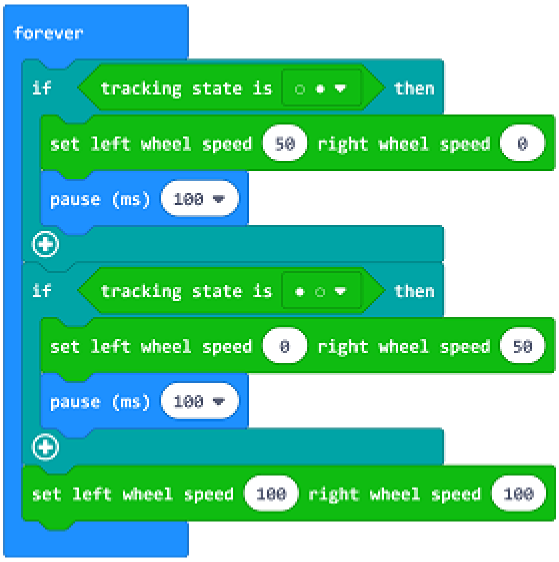


Software Programming

- Program a simple line-following code in the makecode .
- Initiate the connection ports for the left and right wheels to P1 and P2.
- When the left detection probe detects deviation from the black line, the right wheel would stop moving and the left wheel would adjust to go back to the black line at the speed of 50.
- The right detection probe would work as the same as the way of the left detection probe works.



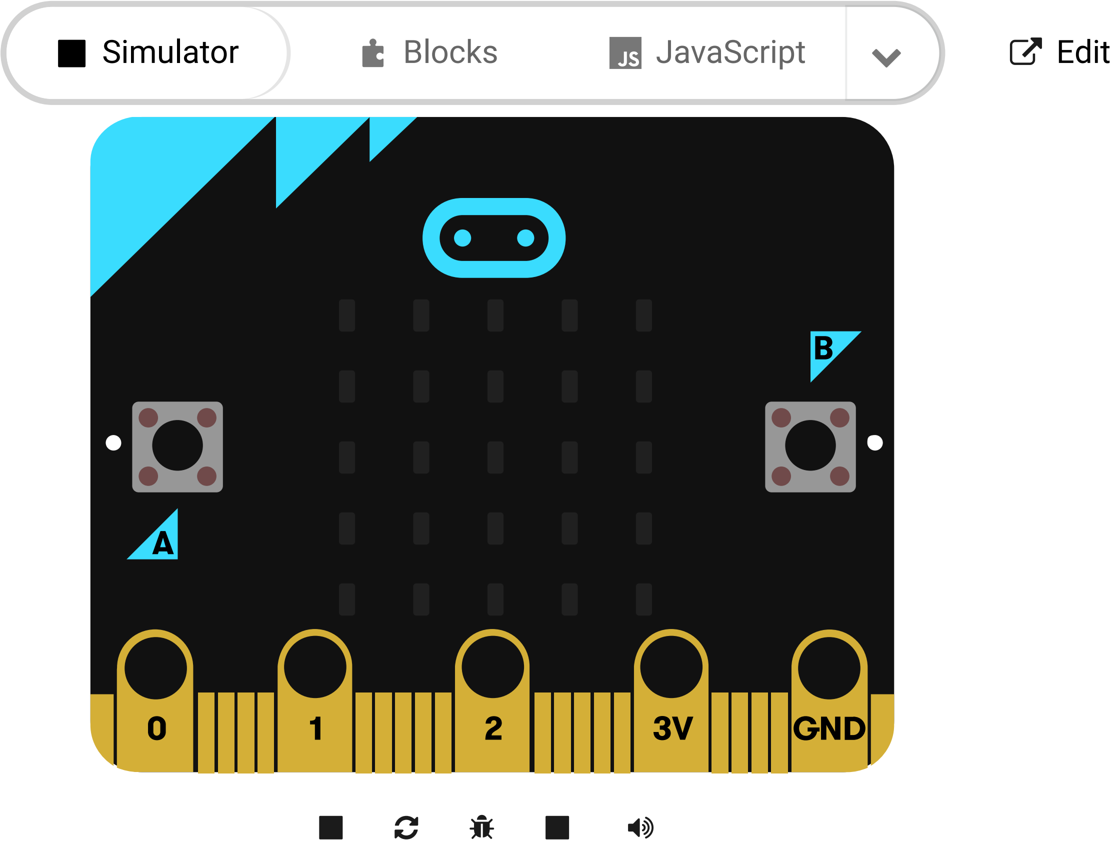
```
on start
  set left wheel at pin P1 ▾ right wheel at pin P2 ▾
  go straight at full speed
```



```
forever
  if  tracking state is ○ ● ▾  then
    set left wheel speed 50 right wheel speed 0
    pause (ms) 100 ▾
  ⊕
  if  tracking state is ● ○ ▾  then
    set left wheel speed 0 right wheel speed 50
    pause (ms) 100 ▾
  ⊕
  set left wheel speed 100 right wheel speed 100
```

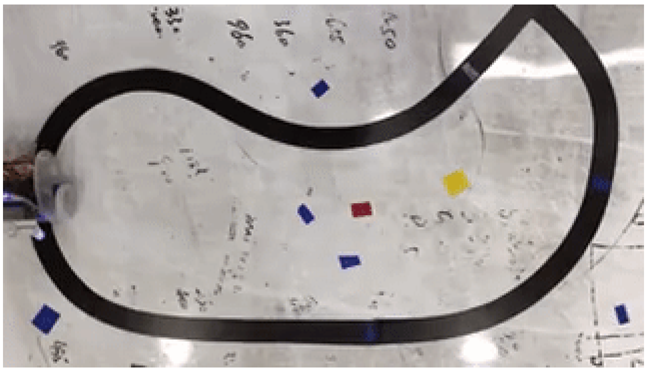
Links:https://makecode.microbit.org/_hYH1Rd697TFf

You can also download the code directly below:



Result

- The Ring:bit car runs around the circle slowly.

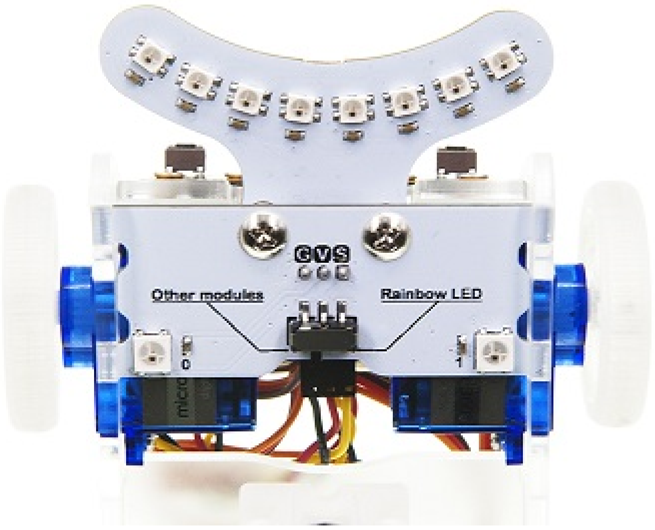


5.6. FAQ

6. Rainbow LED module

6.1. Introduction

- Ring:bit car V2 light-bar module is the dedicated module designed for Ring:Bit Car V2. It brings you mutiple color with simple assemblling methods.
- It comes with 8 Rainbow full color LEDs. You can update it to a car with auto-headlights and rainbow lights.



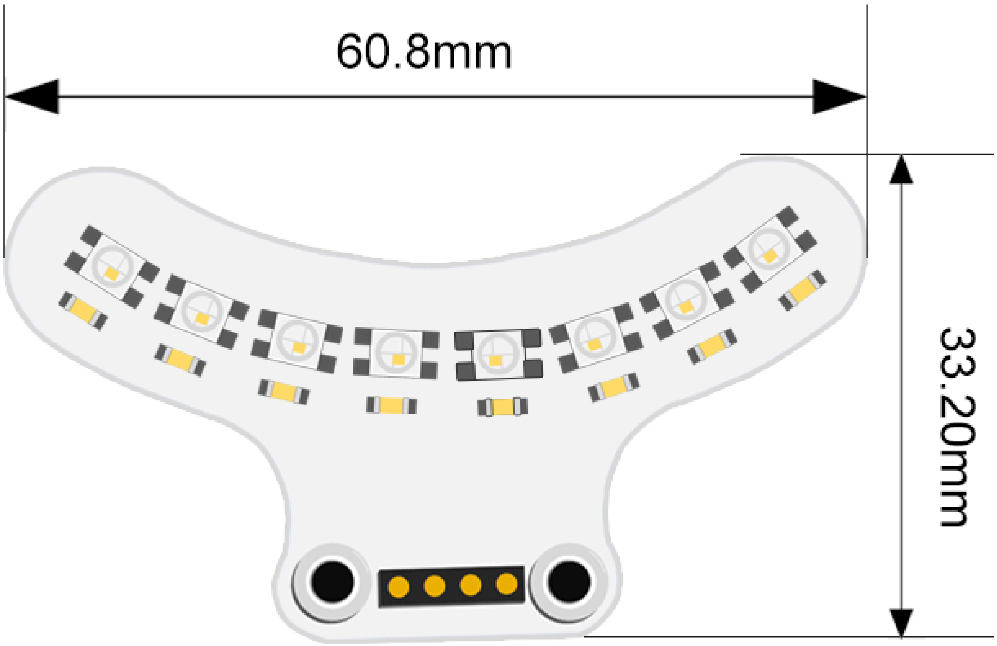
6.2. Features

- Input voltage 3V~5V， driving directly by micro:bit.
- Standard 3 line GVS port, occupying only one I/O port.
- 8 small LEDs, power saving and energy saving.
- Each led can be programmed seperately with RGB color.

6.3. Parameter

Items	Parameter	Remark
Name	Ring:bit car v2 light bar	-
SKU	EF03425	-
Working Voltage	DC 3-5V	-
Connection Port	Ring:bit car dedicated pin ports	Fixed with scews
Output Signal	Simulation	-
LEDs	8 pcs	-
Dimension	60.8 x 33.20mm	-
Net Weight	5.7g	-

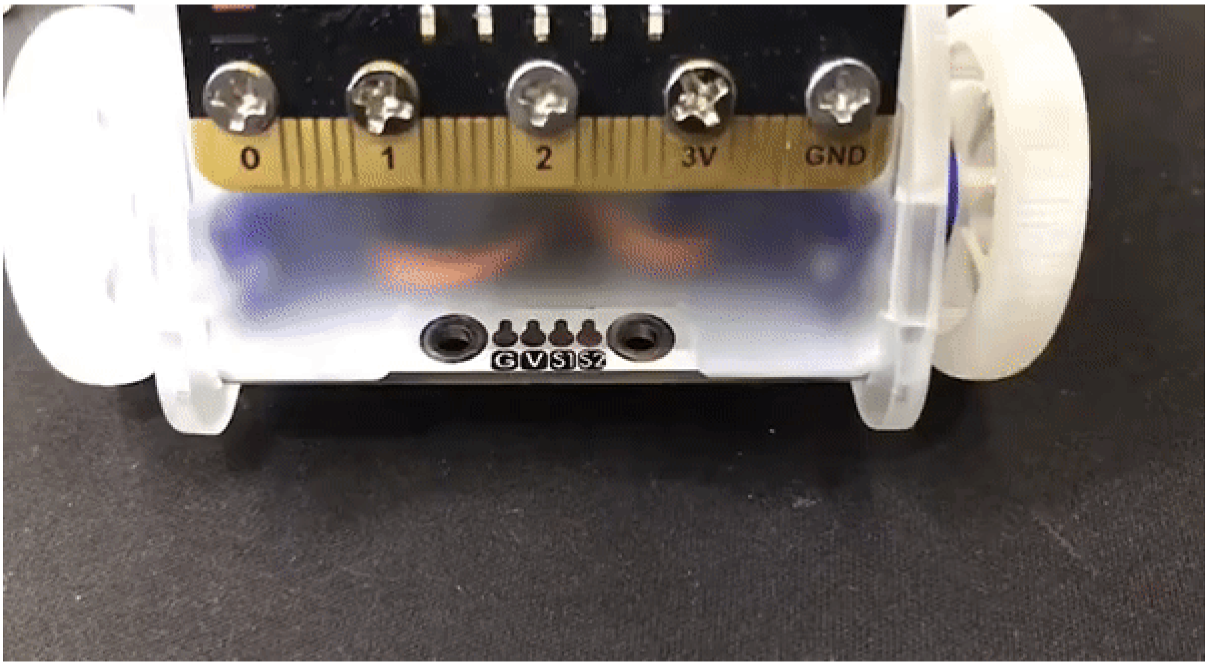
6.4. Dimensions:



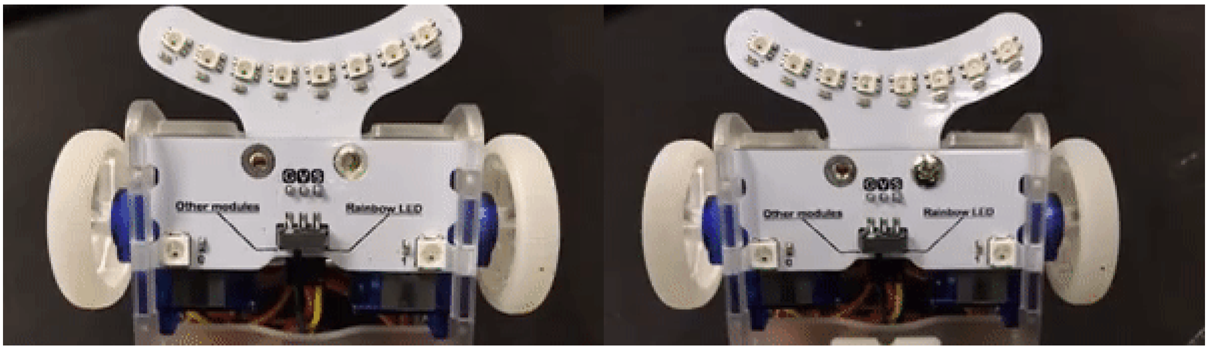
6.5. Quick to Start

Hardware Connection

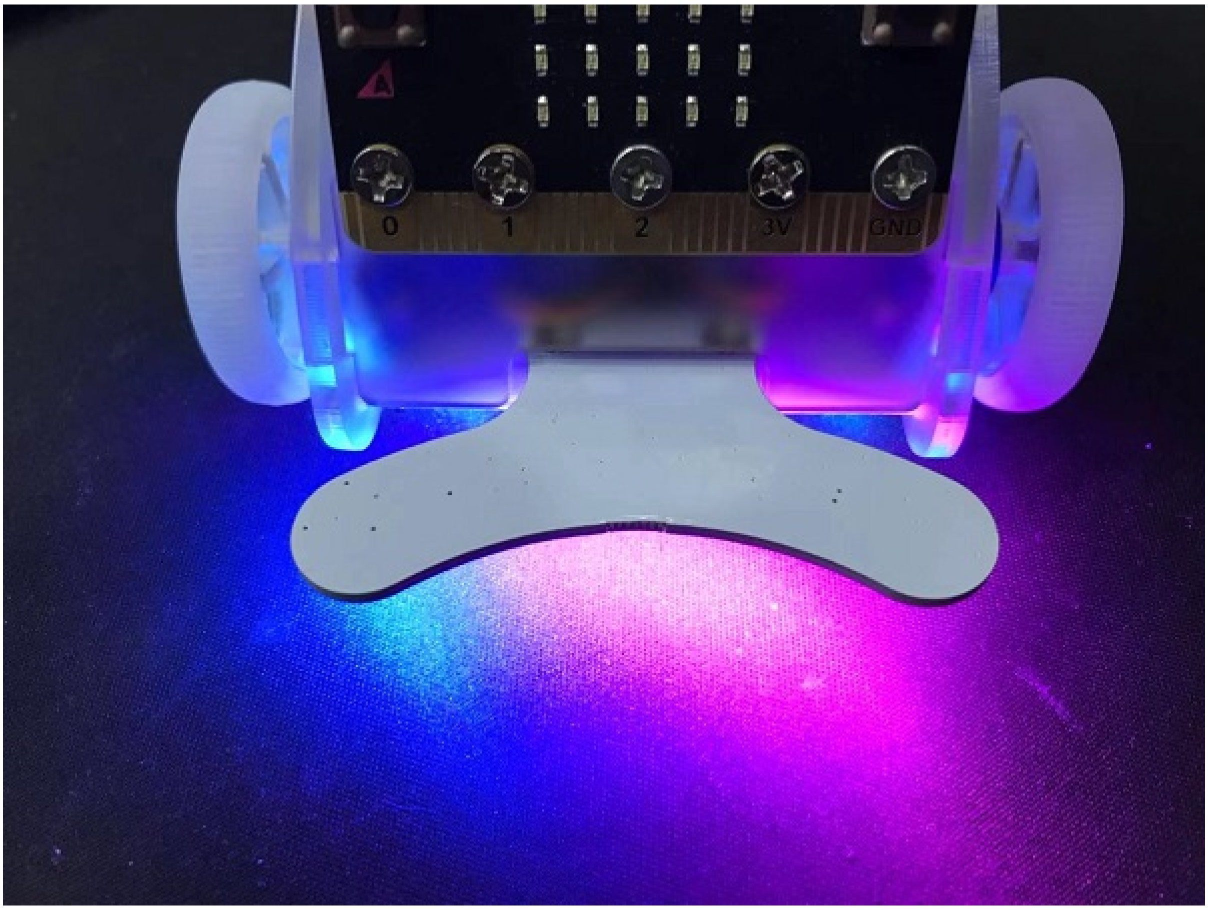
- First, insert this module to the baseboard of the Ring:bit car V2.



- Then, screw up the screws.

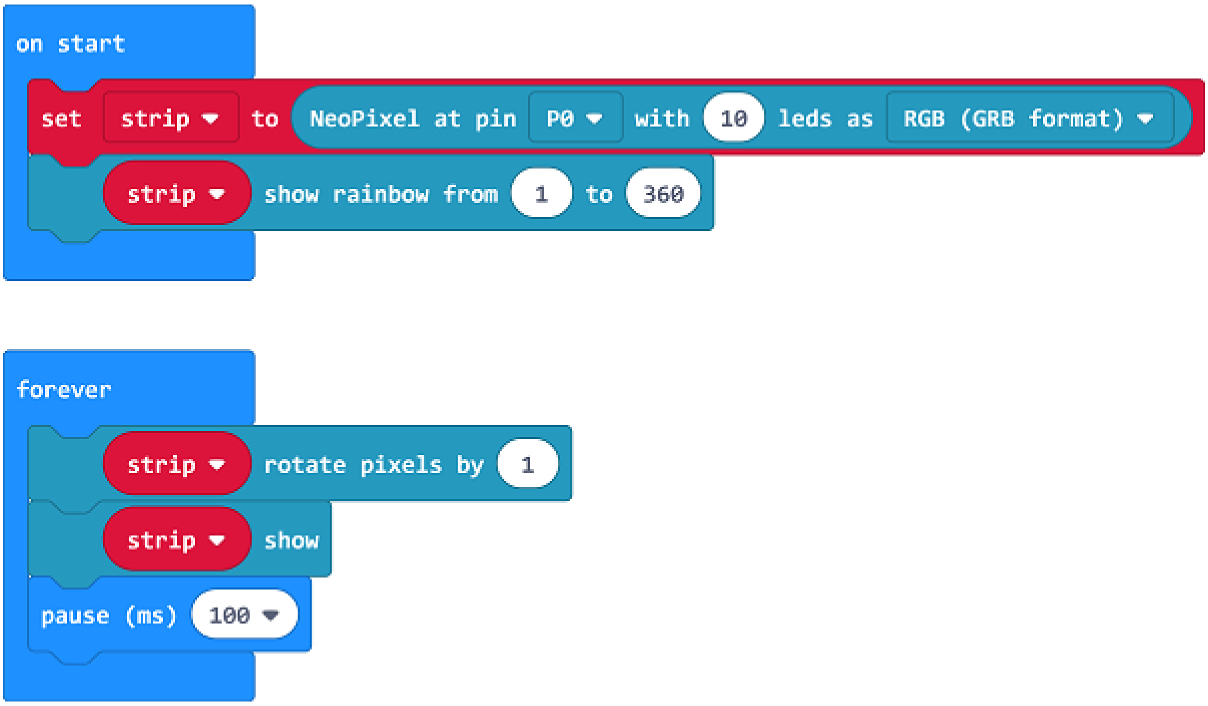


- Completed.



Software Programming

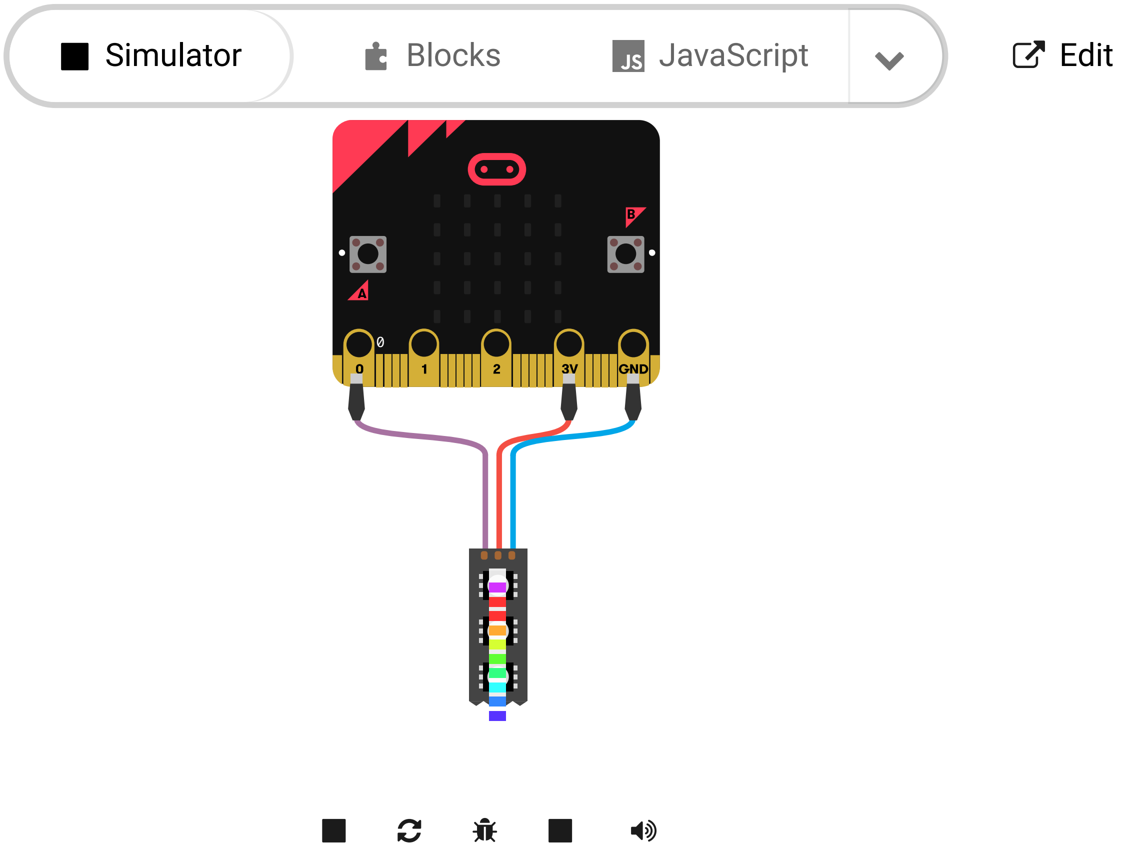
- Program a simple light-bar code in the makecode .
- Set the variable for Strip, initiate the 10pcs LEDs connected to P0.
- Programme the LED to show Rainbow color
- Programme to circulate and displace the color in “forever” brick.
- Show color.



```
on start
    set strip ▾ to NeoPixel at pin P0 ▾ with 10 leds as RGB (GRB format) ▾
    strip ▾ show rainbow from 1 to 360

forever
    strip ▾ rotate pixels by 1
    strip ▾ show
    pause (ms) 100 ▾
```

Links: https://makecode.microbit.org/_3Wc1k8Ckg9vF

You can download it directly below:



Result

-Rainbow lights.

6.6. FAQ

7. Sonar:bit

7.1. Introduction

- Sonar:bit is a 3-wire ultrasonic module with the working voltage between 3-5V. It is available to be used to 3.3V or 5V micro-controller system. With only one 3-wire(GVS) cable, it can work properly. Compared to the normal 4-wire ultrasonic module, it has saved one IO port. The measurement range of sonar:bit is 4cm-400cm. It can output stable and accurate measurement data with ±1cm tolerance only.
- It can connect to the Ring:bit with an expansion board.

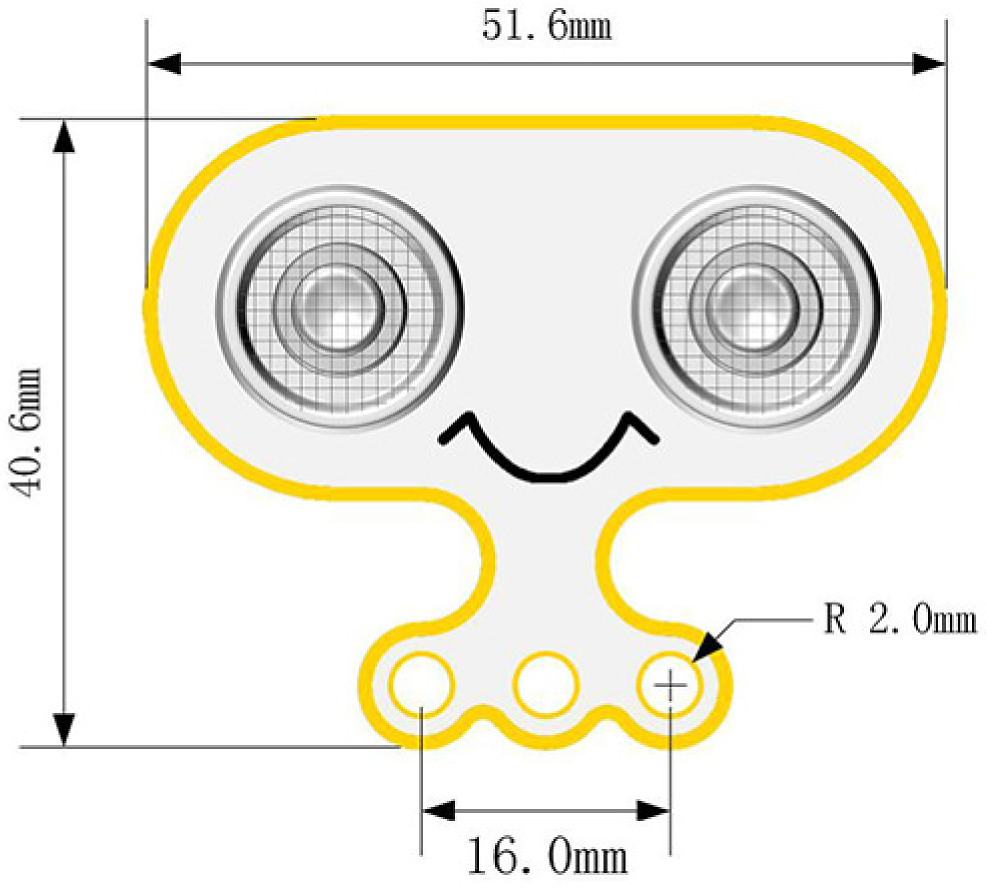
7.2. Characteristics

- Input voltage:3V~5V and can be driven by micro:bit directly.
- Standard 3-wire GVS connecotr, which occupies 1 IO port only.

7.3. Parameter

Item	Parameter	Remark
Name	Ring:bit car v2 sonar:bit	-
SKU	EF04089	-
Working Voltage	DC 3-5V	-
Connection	3pin GVS connection	-
Output Signal	Simulation	-
Measuring Distance	4-400cm	-
Dimmension	40.60mm * 51.60mm	-
Net Weight	12g	-

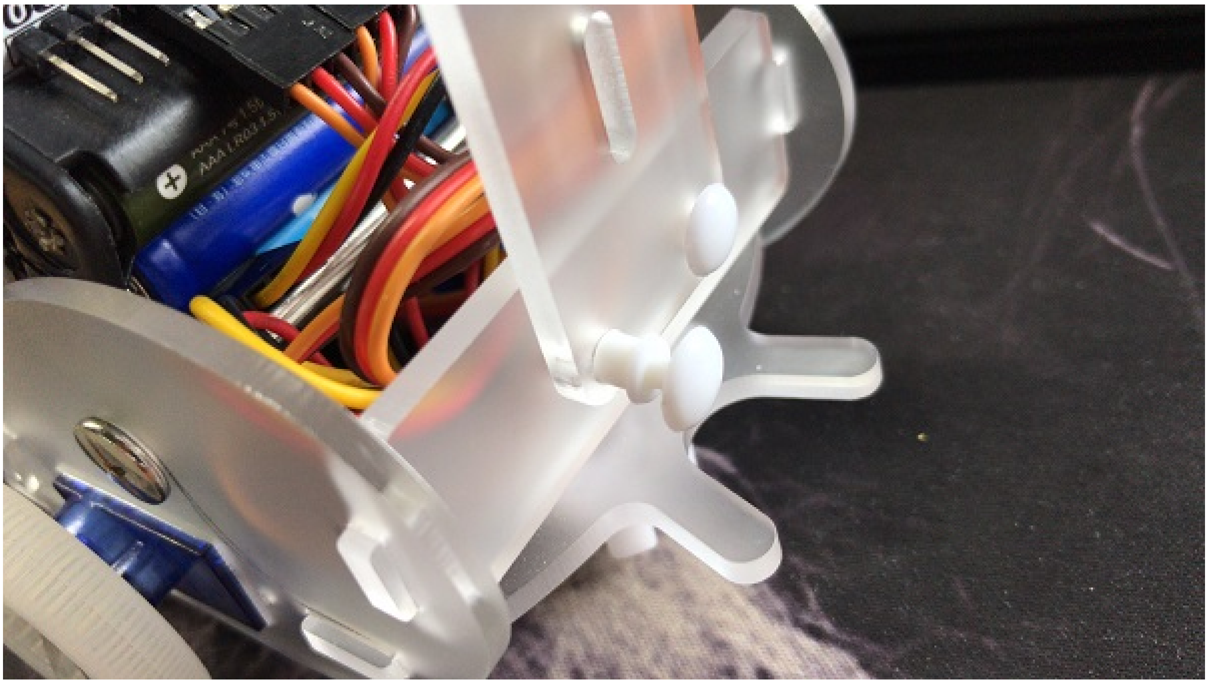
7.4. Outlook



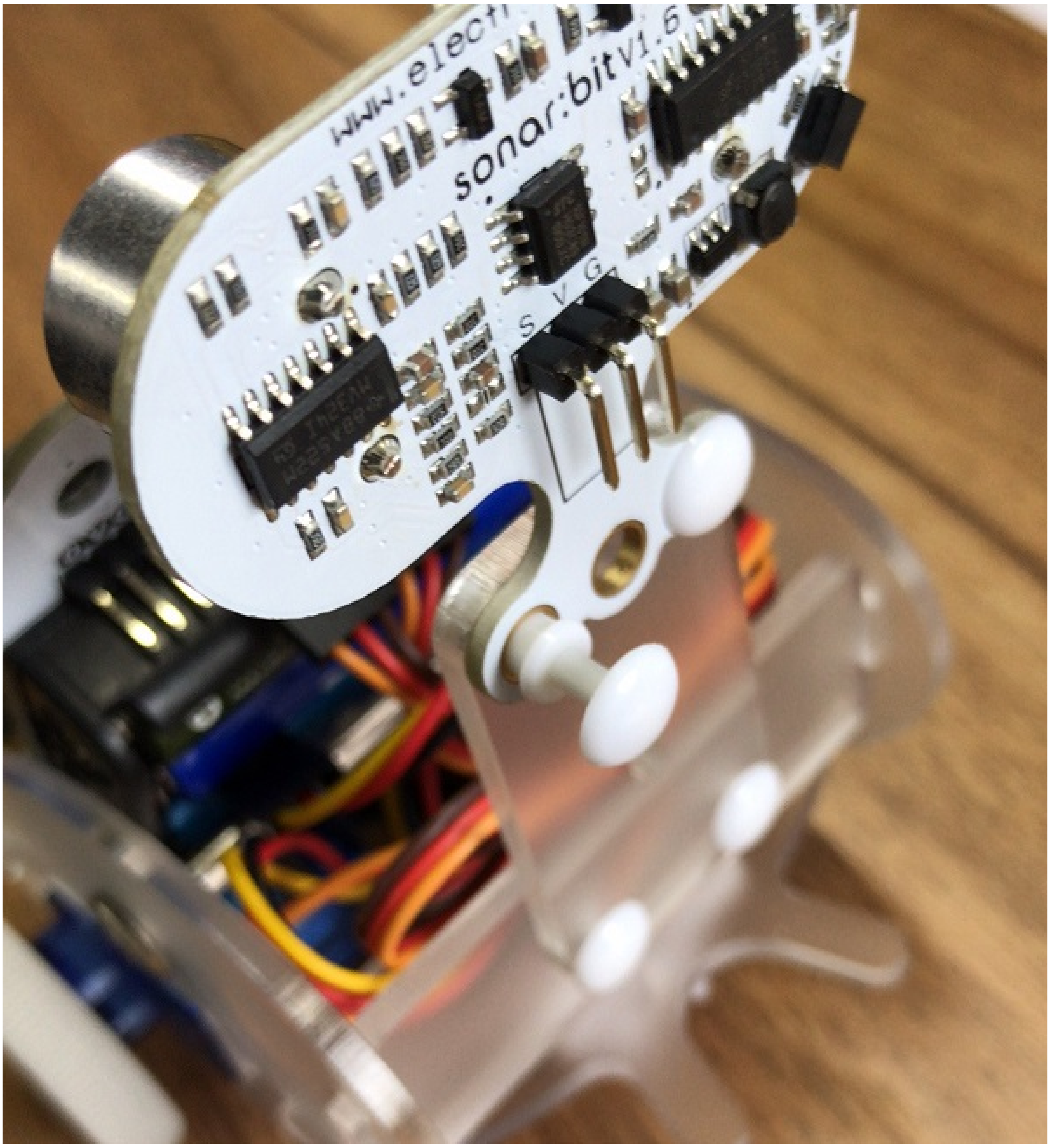
7.5. Quick to Start

Hardware Connection

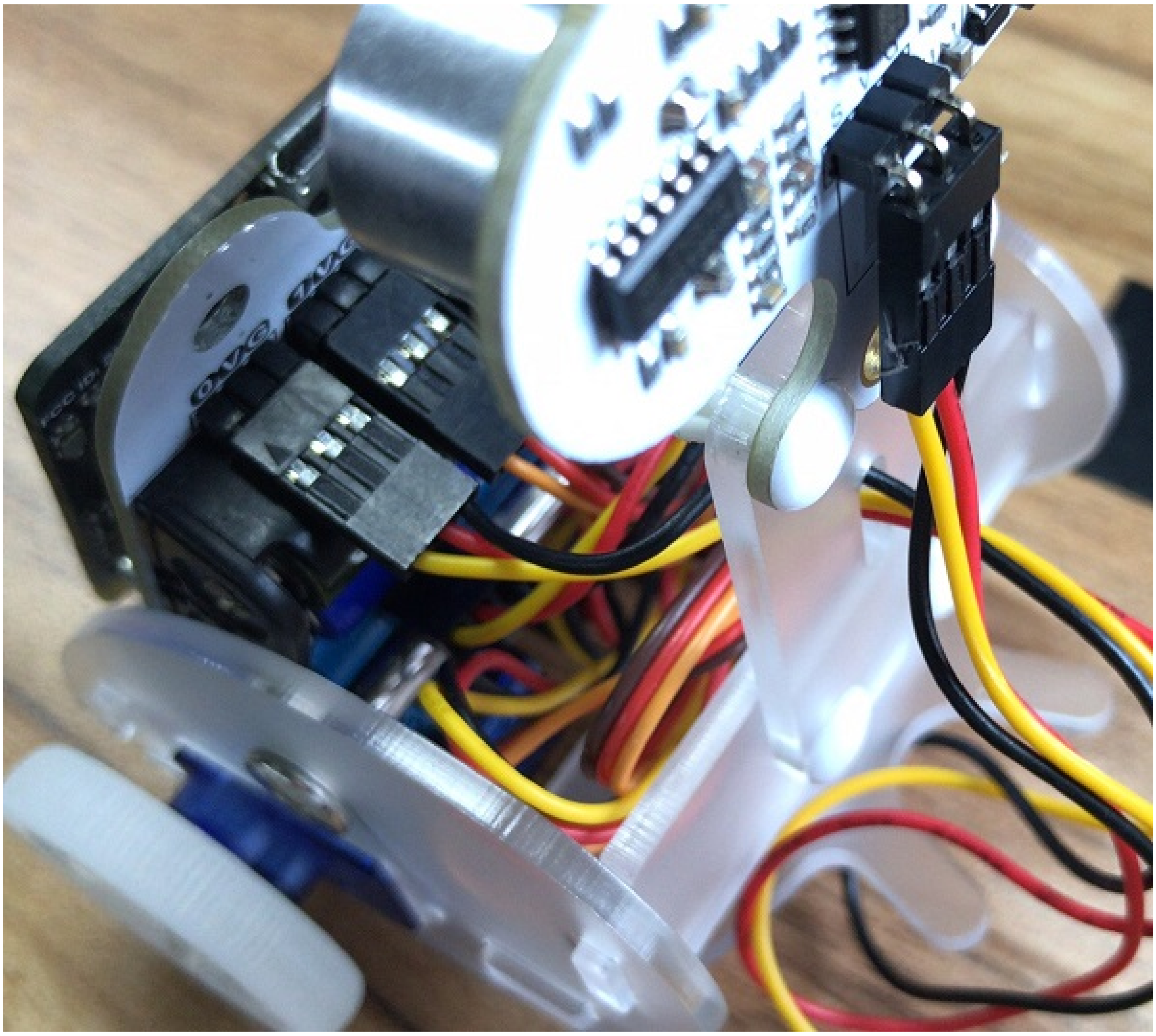
- Connect the acrylic transition board to the back board with the rivets.



- Connect the Sonar:bit to the other side of the acrylic transition board with rivets.

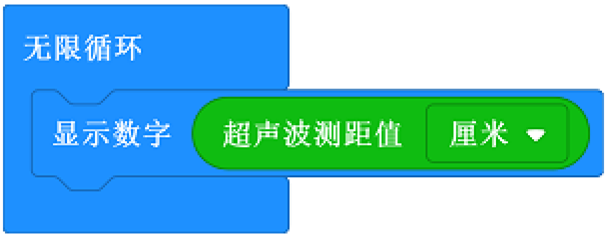


- Connect the Sonar:bit to the Ring:bit breakout board with a 3-pin wire.



Software Programming

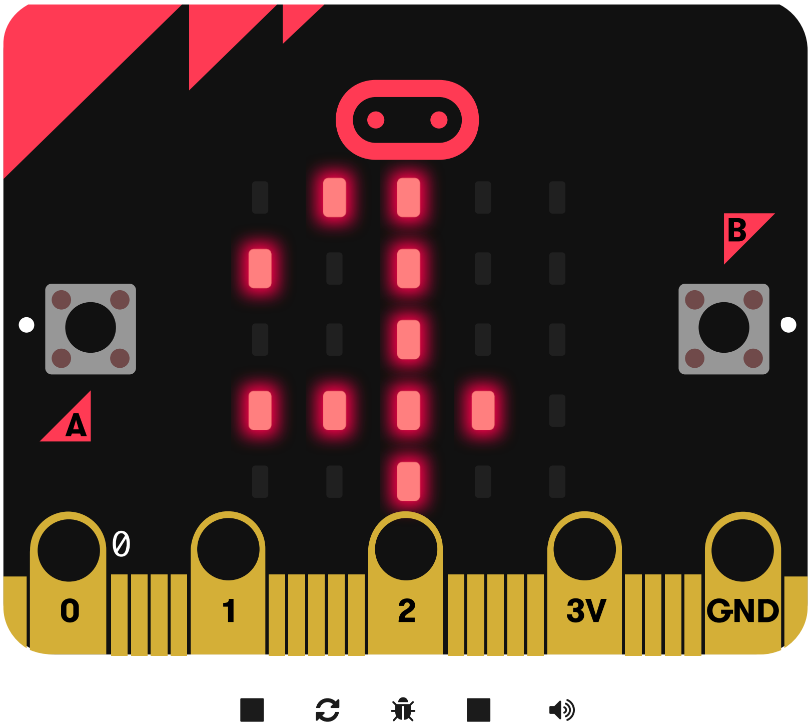
- Choose `sonar:bit` in makecode.
- Search `ringbit` in the dialogue box and click it to download it.
- Programme to measure the distance with ultrasonic in makecode on-line editor.



```
无限循环
  显示数字 超声波测距值 厘米 ▾
```

Links: https://makecode.microbit.org/_iP7DWxFicRox

You can download it directly below:



Result

- The value measured by Sonar:bit is showing on the micro:bit.

7.6. About

case 10

8. ring:bit car v2 accelerometer arithmetic

- Accelerometer arithmetic of the Ring:bit car

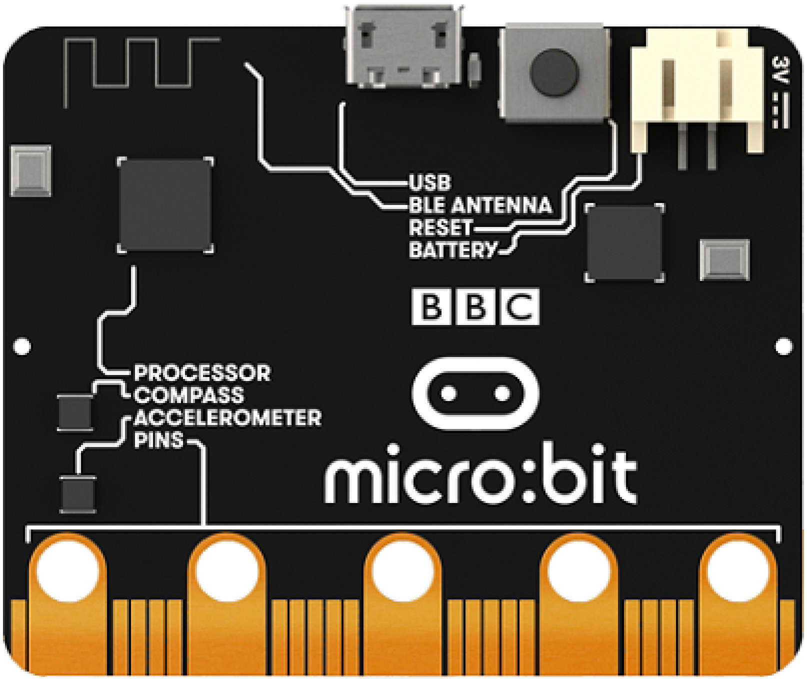
8.1. Required Materials

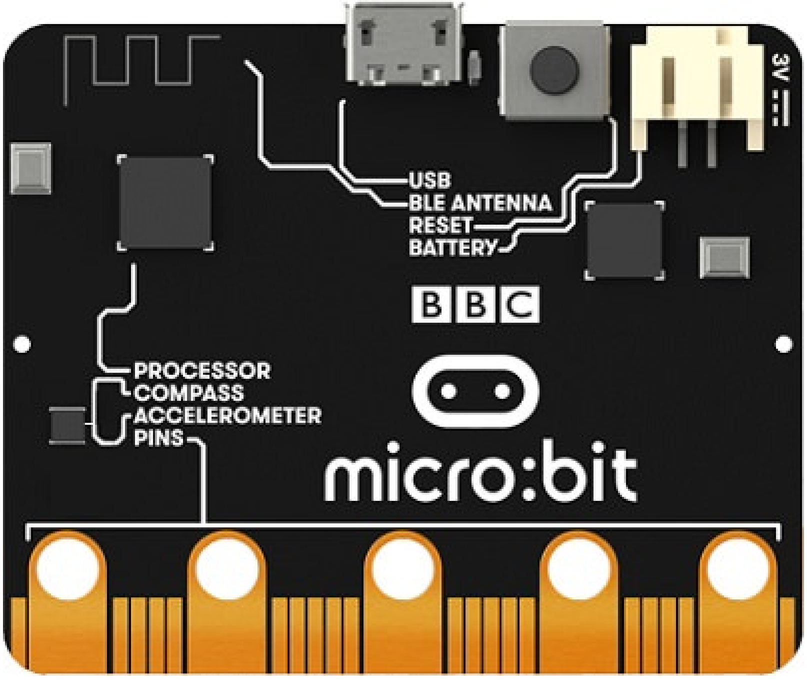
- 1 x Micro:bit
- 1 x Ring:bit Car

8.2. Background

- Micro:bit is a small coding computer which was given by the BBC. It is specially designed for adolescent programming education and to male programming more easy and interesting.
- ELECFREAKS Ring:bit car 2 is a small DIY smart car driven by the BBC micro:bit and the ELECFREAKS Ring:bit. The Ring:bit extends the micro:bit's 3 GPIO ports and allow for different sensors and components to be easily attached to the micro:bit. A basic Ring:bit Car can be easily programmed to run autonomously, with a remote control, and even create rainbow beacons of light. Just add one of the many extensions available and your Ring:bit Car can do even more things like line and light following, obstacle avoiding, drawing and more!
- Wireless Technology Wireless technology sends message by radio wave because changes of electricity will generate waves. We can use this to load the message to the wave. Variation of electromagnetic field will generate the electricity when the electromagnetic is arrving the receiver. Then, to extract the message from wave by demodulation to send the message.
- Acceleration Sensor The accelerometer is a kind of sensor to measure the accelerated speed. It has included mass block, damper, elastic component, sensitive element and optimal tuning circuit. The sensor is measuring the inertia force and getting the acceleration magnitude by newton's second law during its accelerating. According to the difference of sensitive element, the common used accelerometer has included capacitance, strain, piezoresistive and piezoelectric.

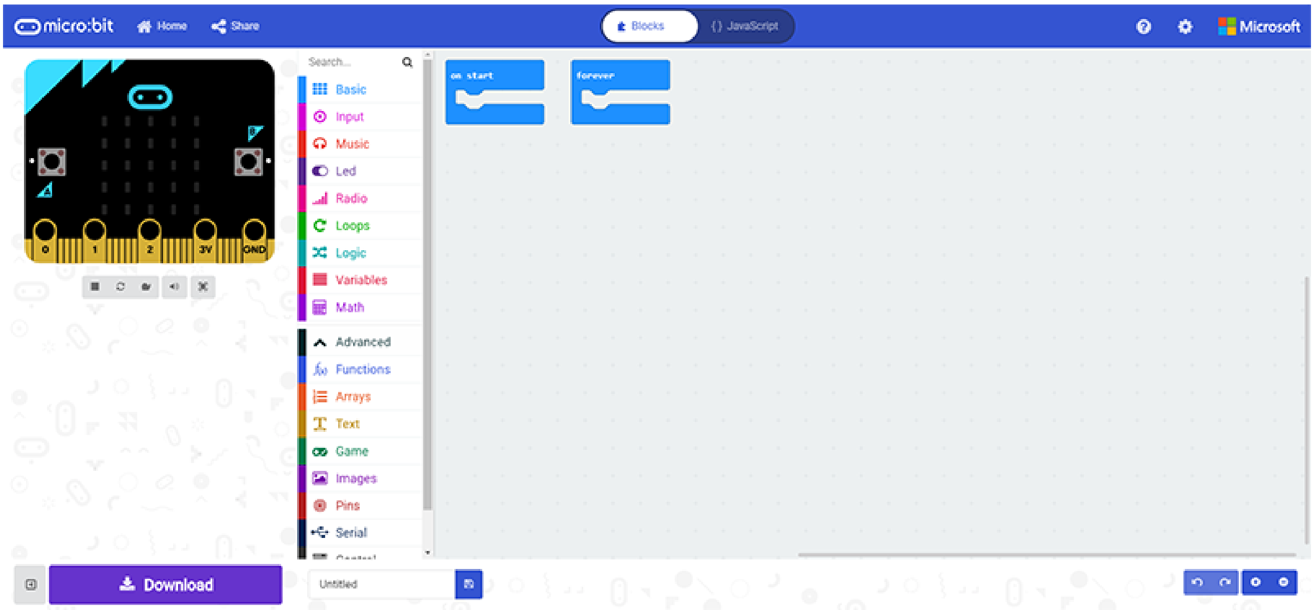
The new version accelerometer chip of micro:bit is different from the old version. The new version has combined the electronic compass and the accelerometer. No change for usage.





8.3. Software

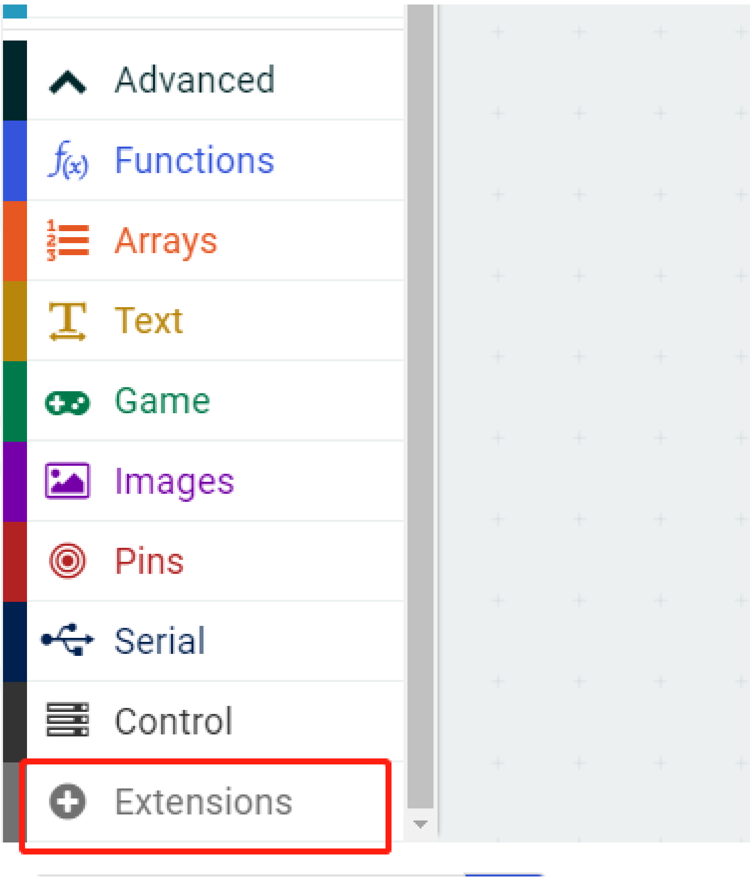
makecodeOnline block codinghttps://makecode.microbit.org/#



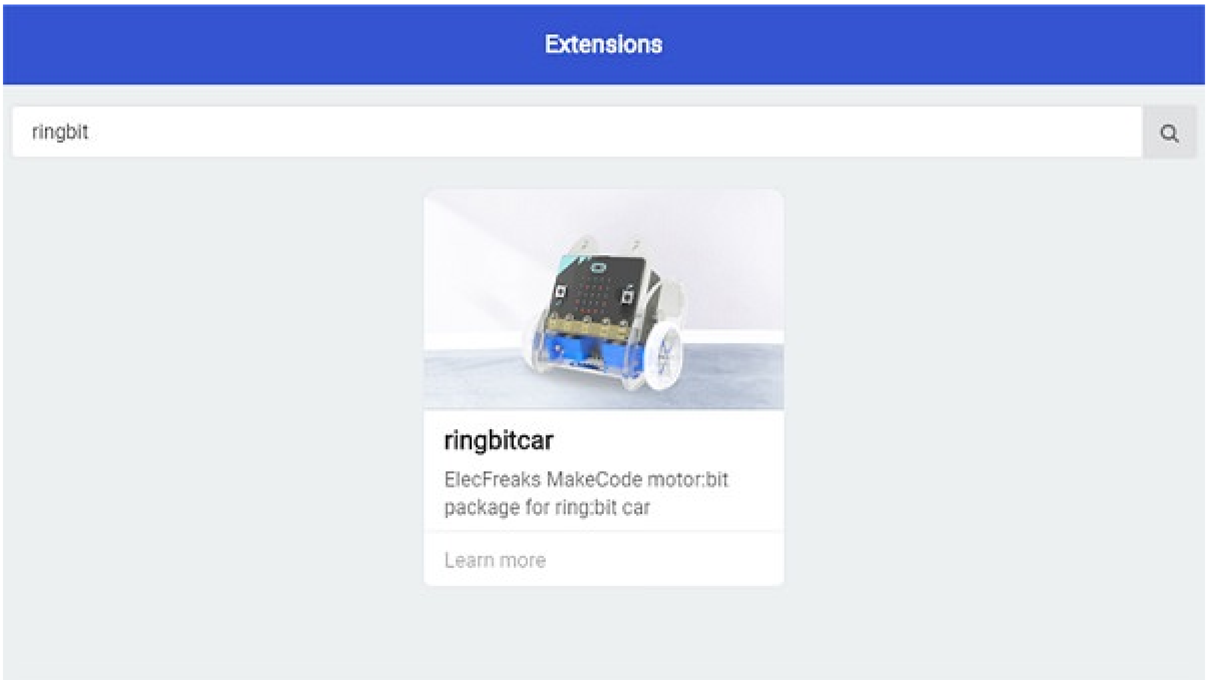
8.4. Coding

Step 1： Add Coding Package

- Click on “Advanced” in the MakeCode Drawer to see more code sections.



- Search for “ringbitcar” and click on the ring:bit car package to add it to your project. (As below picture)



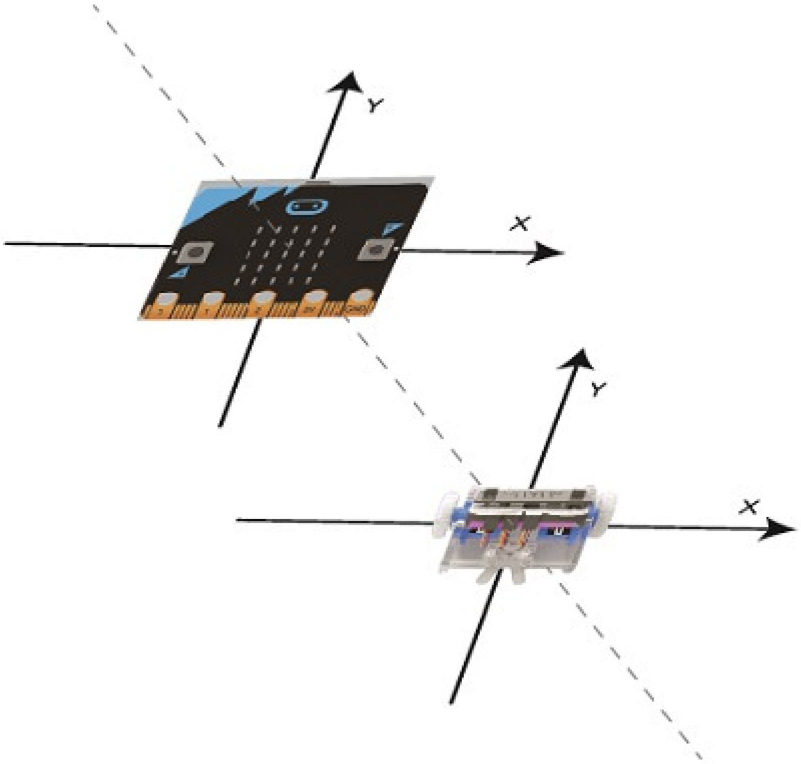
ringbitcar

ElecFreaks MakeCode motor:bit package for ring:bit car

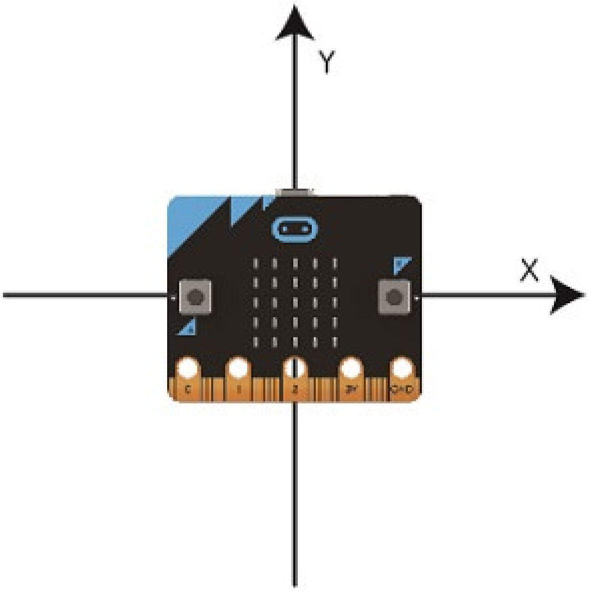
Note： If you get a warning telling you some packages will be removed because of incompatibility issues, either follow the prompts or create a new project in the Project file menu.

Step 2： Principle of arithmetic

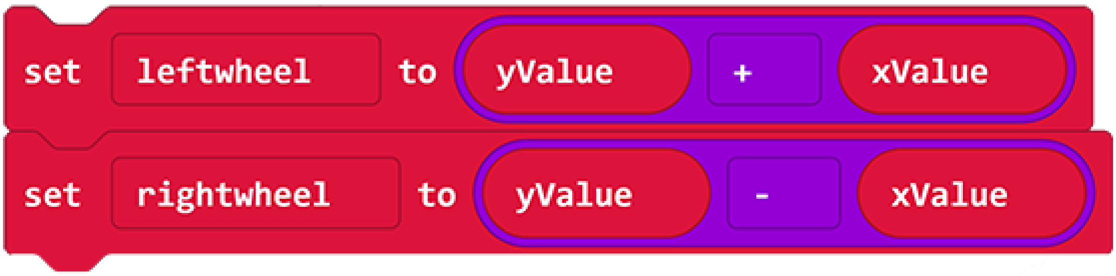
- The accelerometer has 2 directions: x-axis and Y-axis. It is easy to understand when you put the micro:bit and the Ring:bit Car to a same three dimensional coordinate axis in a top view.



Move direction and speed of the car is controlling by micro:bit’s direction.

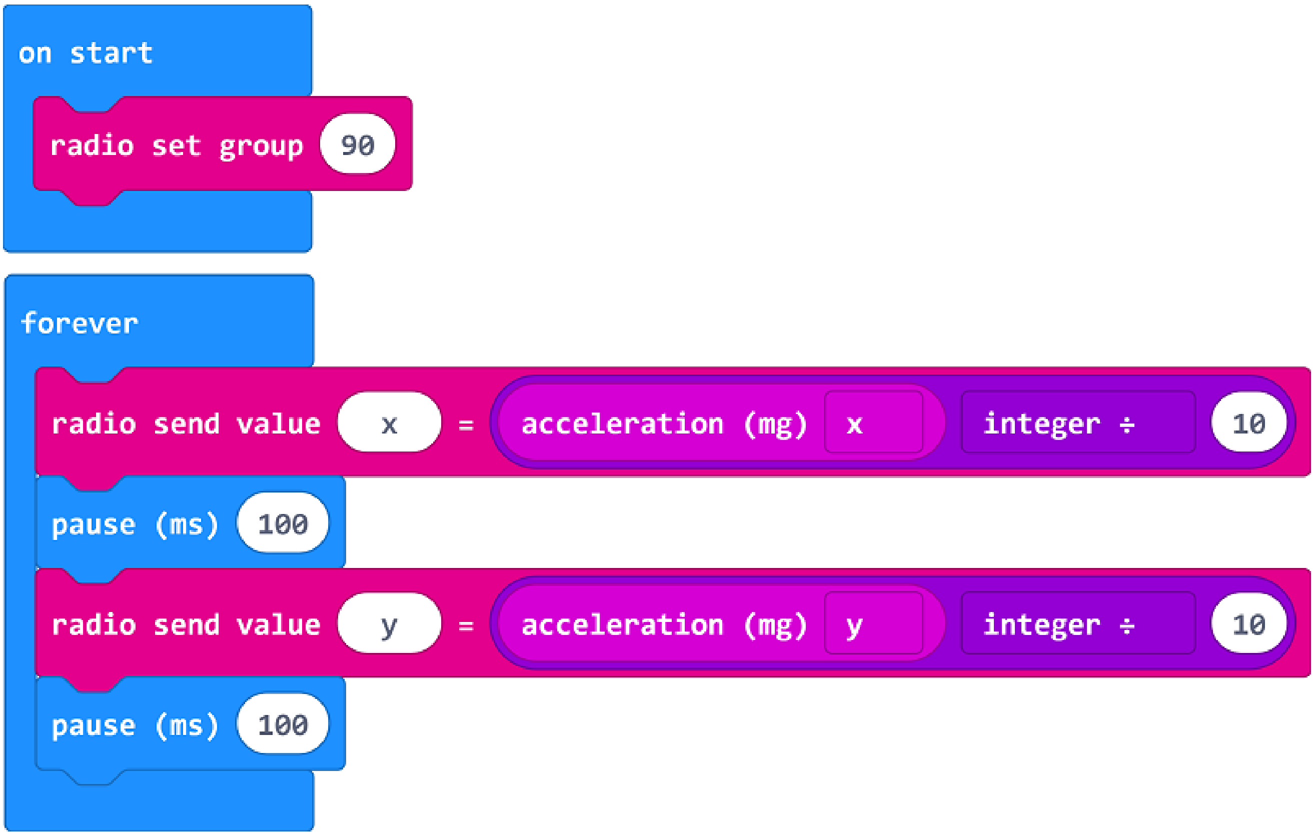


- Move forward to the right, the value of x-axis and Y-axis are positive number. The speed of the left wheel should be greater than the right wheel and complete turn right.
- Others in the same way.
- Result: The value of Y-axis plus the x-axis is the value of the left wheel; the value of Y-axis minus the x-axis is the value of the right wheel.



```
set leftwheel to yValue + xValue
set rightwheel to yValue - xValue
```

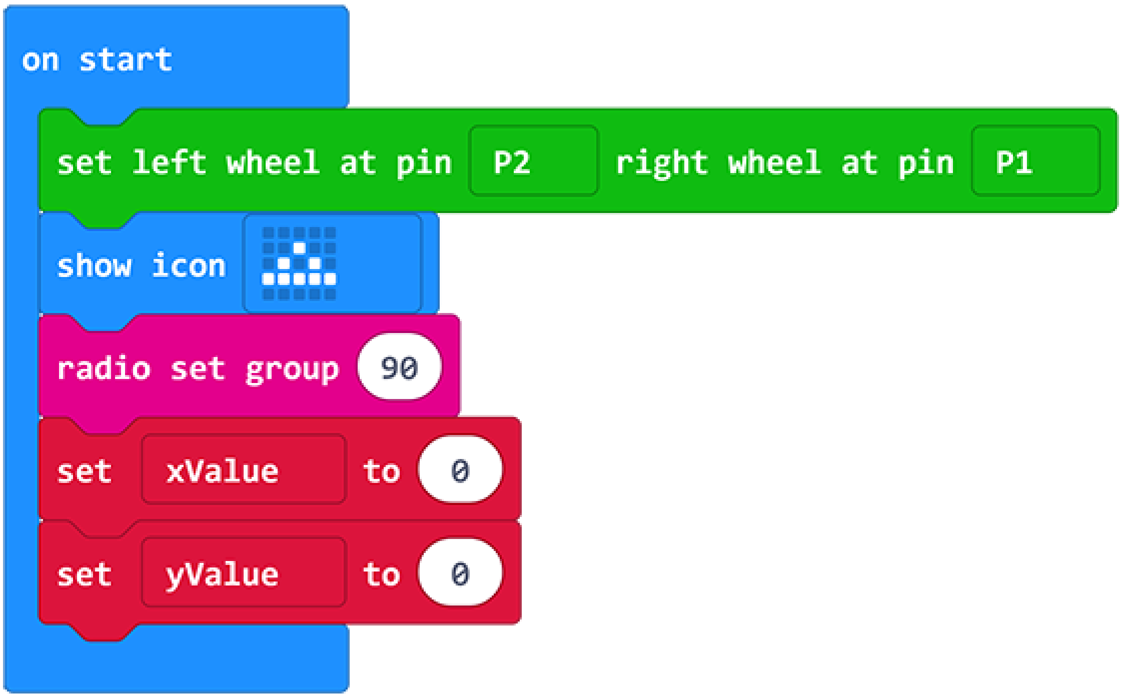
Step 3：Code Interpretation(The controlling end)



```
on start
    radio set group 90

forever
    radio send value x = acceleration (mg) x integer ÷ 10
    pause (ms) 100
    radio send value y = acceleration (mg) y integer ÷ 10
    pause (ms) 100
```

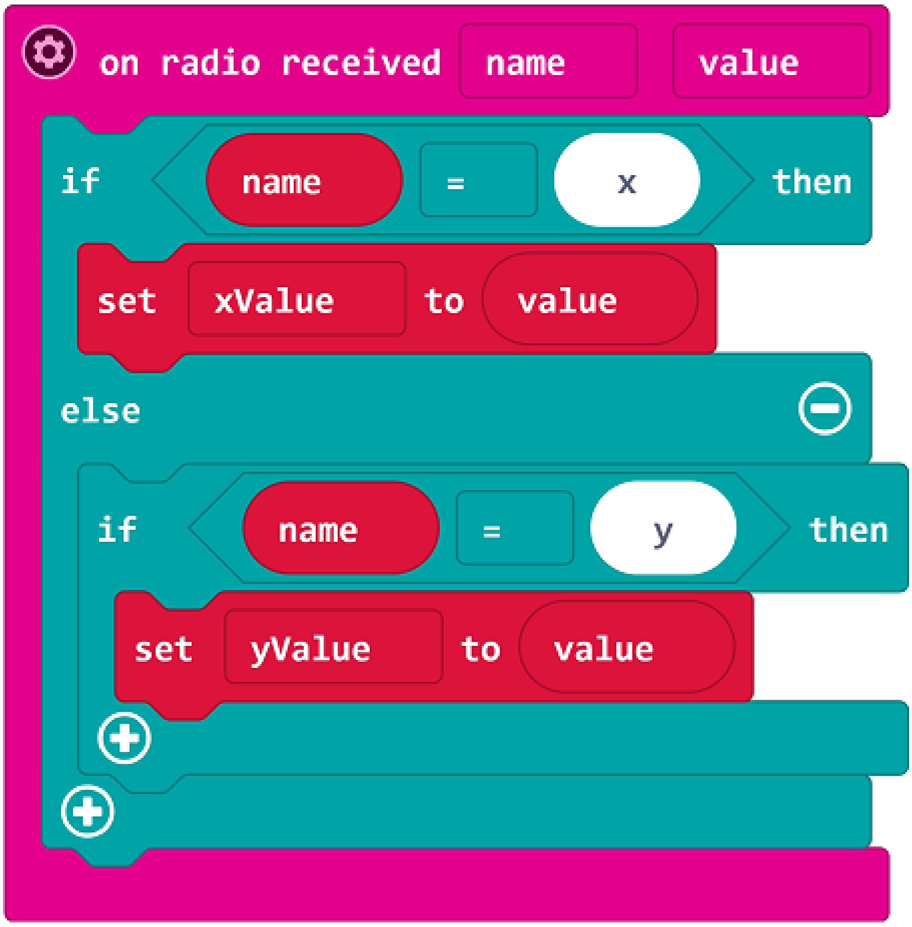
- On start, set wireless to `90`, same with the car.
- In the forever loop, send the acceleration magnitude of the X-axis. The scope of the acceleration magnitude is `-1024~+1024`, the car speed is `-100~+100`, so we need to divided by ten.
- In the forever loop, send the acceleration magnitude of the Y-axis and divided by ten.

Step 4：Code Interpretation(The car)



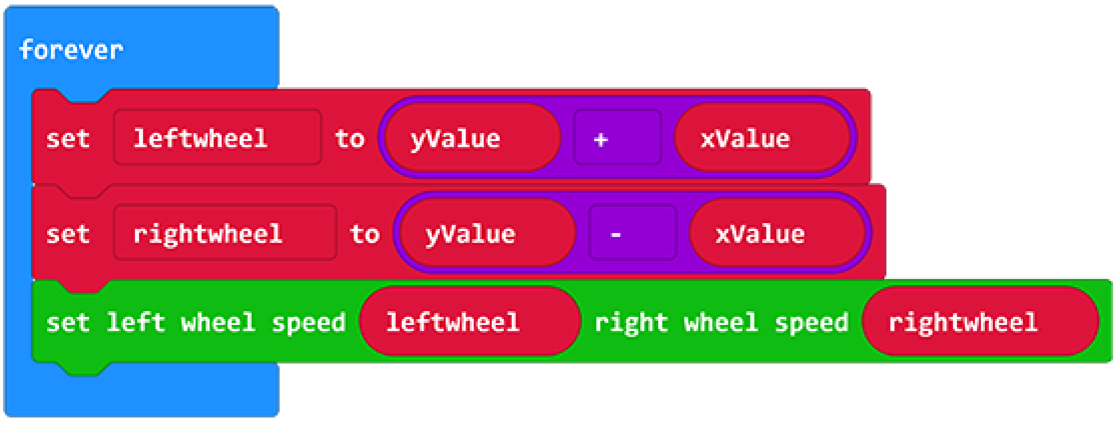
```
on start
    set left wheel at pin P2 right wheel at pin P1
    show icon
    radio set group 90
    set xValue to 0
    set yValue to 0
```

- On start, set the left wheel to P1 and the right wheel to P2 (subject to the actual) and show an icon. Set wireless value to 90 (same with the sending side), to set variable XValue and variable yValue to save the value of X-axis and Y-axis.



```
on radio received name value
    if name = x then
        set xValue to value
    else
        if name = y then
            set yValue to value
```

- On radio received the value, if it is X, save the data to `xValue`, if it is Y, save the data to `yValue`.



```
forever
    set leftwheel to yValue + xValue
    set rightwheel to yValue - xValue
    set left wheel speed leftwheel right wheel speed rightwheel
```

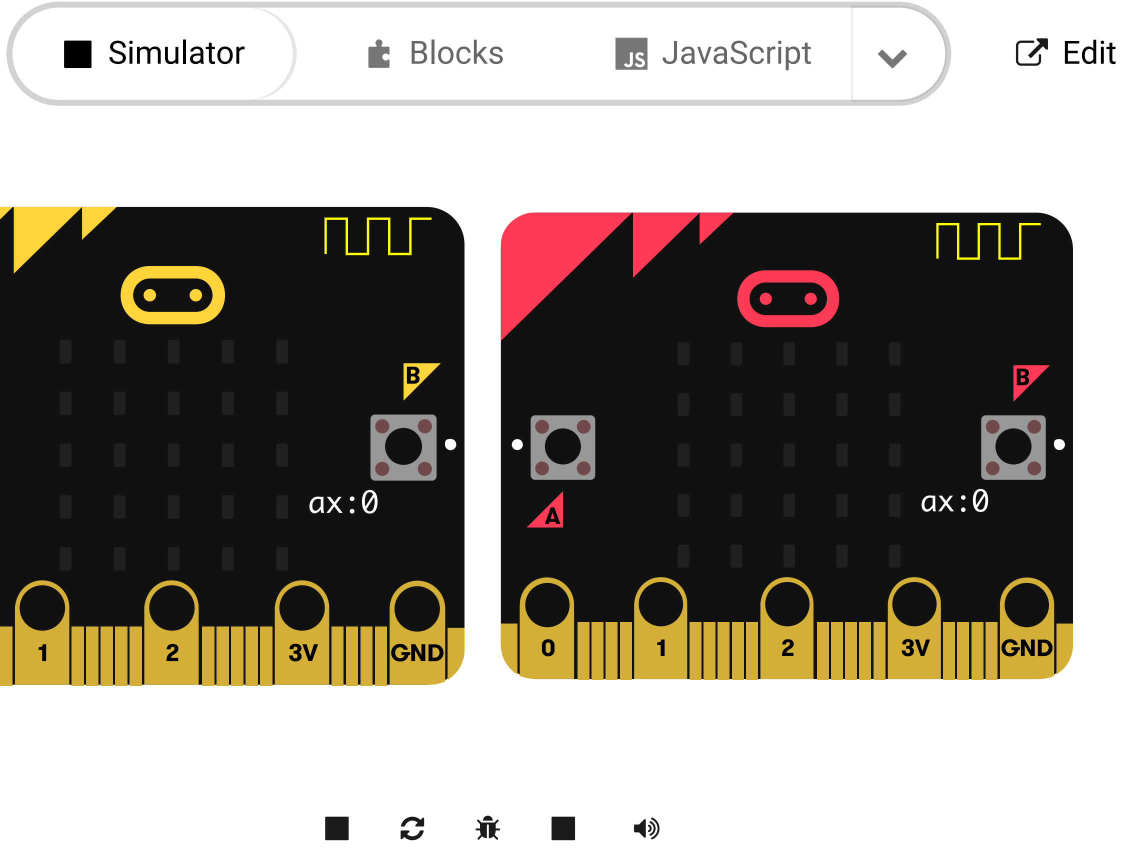
- Then, calculating the speed value of the right and left wheel, set them to corresponding value.

8.5. Reference Program

Remote Control Coding

Program Link： https://makecode.microbit.org/_AT4PoHKdVi6L

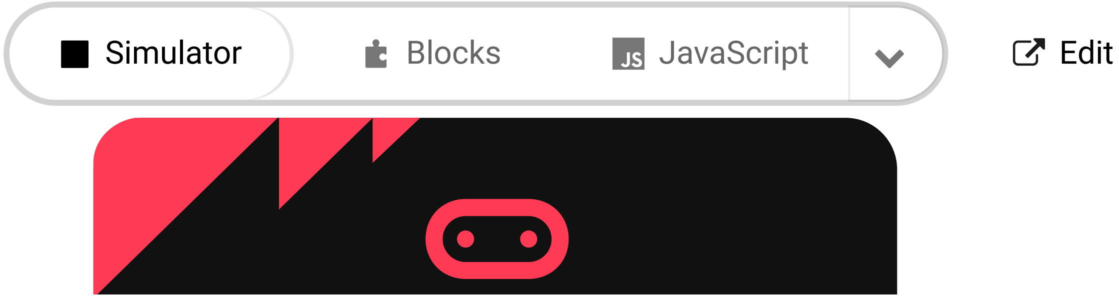
If you don’t want to type these code by yourself, you can directly download the whole program from the link below:

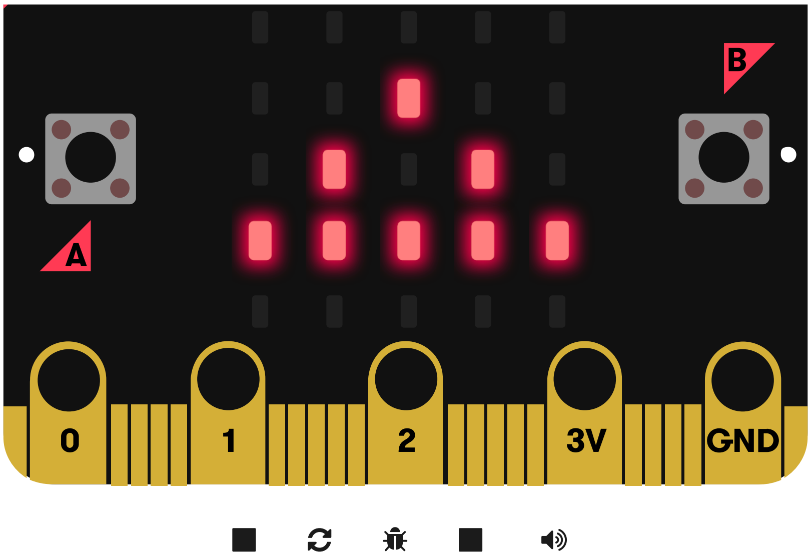


Ring:bit car Coding

Program Link： https://makecode.microbit.org/_e5t6XPHoTiHy

If you don’t want to type these code by yourself, you can directly download the whole program from the link below:





8.6. Result

- The Ring:bit car moves as direction of the gyroscope and oblique angle controls its speed.
-

9. case 01 Full speed ahead

9.1. Our Goal

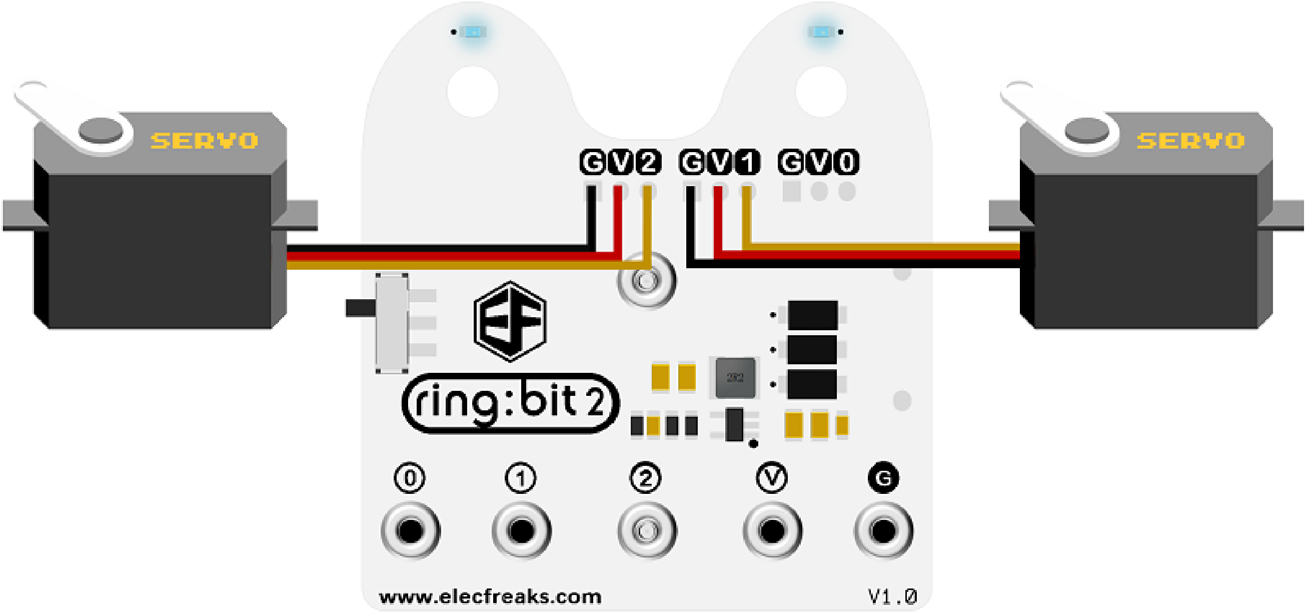
- To make the Ring:bit Car to go ahead and reverse.

9.2. Requiered Materials

- 1 x Ring:bit Car

9.3. Hardware Connect

- Connect the left wheel servo to P1 of the Ring:bit expansion board and the right wheel servo to P2.



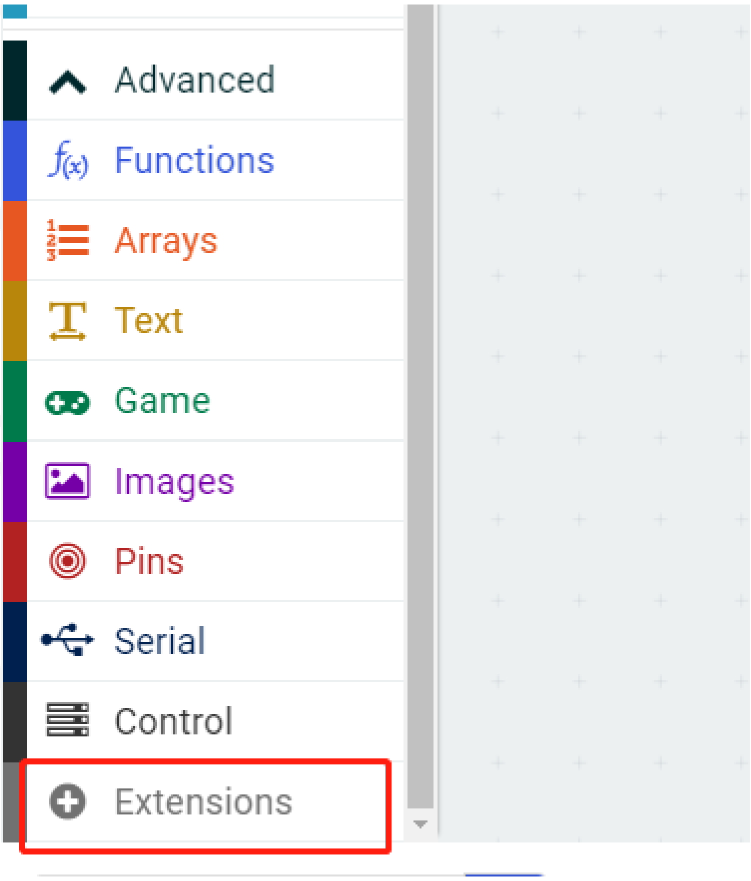
9.4. Software

makecode

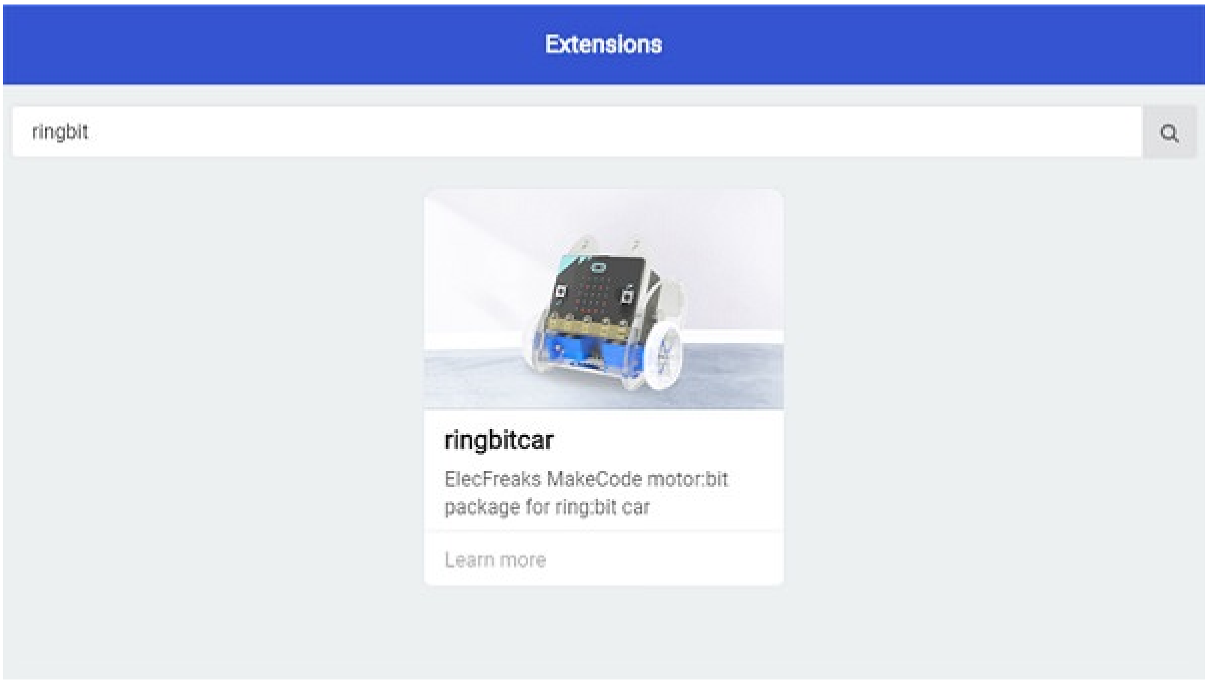
9.5. Coding

Step 1

- Click on “Advanced” in the MakeCode Drawer to see more code sections.



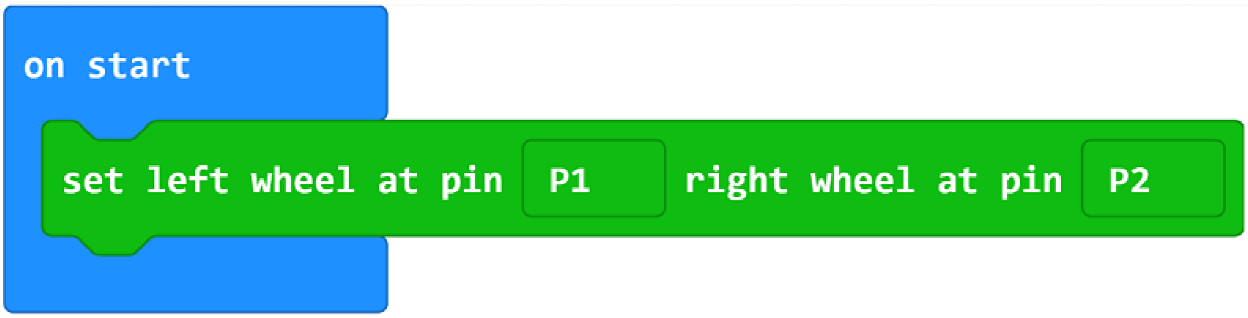
- Search for “ringbitcar” and click on the ring:bit car package to add it to your project. (As below picture)



Note： If you get a warning telling you some packages will be removed because of incompatibility issues, either follow the prompts or create a new project in the Project file menu.

Step 2

- Snap the block `go straight at full speed` into the `On start` block.
- The port number is based on the actual servo connection port.

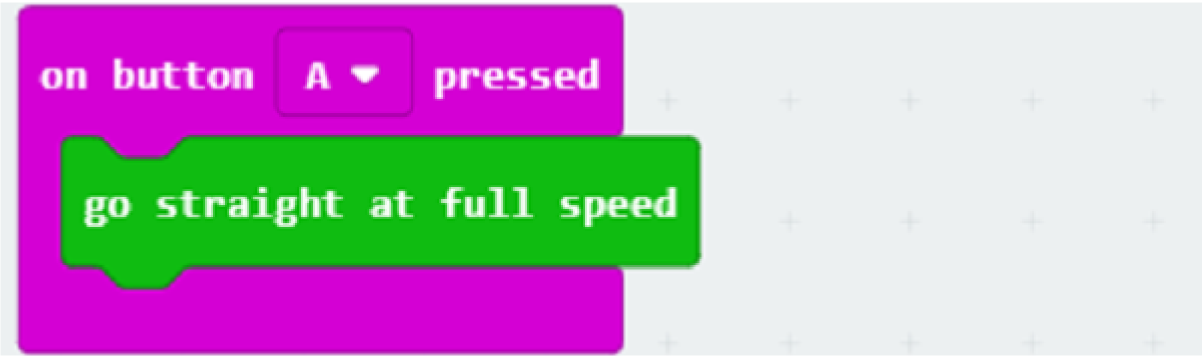


on start

set left wheel at pin P1 right wheel at pin P2

Step 3

- Snap the `go straight at full speed` block into the `on button A pressed` block.

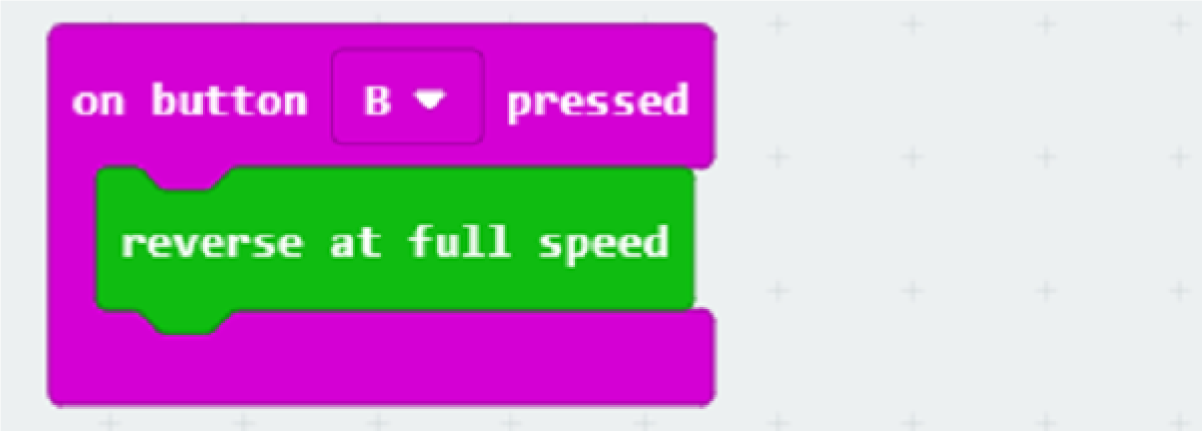


on button A ▾ pressed

go straight at full speed

Step 4

- Snap the `reverse at full speed` block into the `on button B pressed` block.



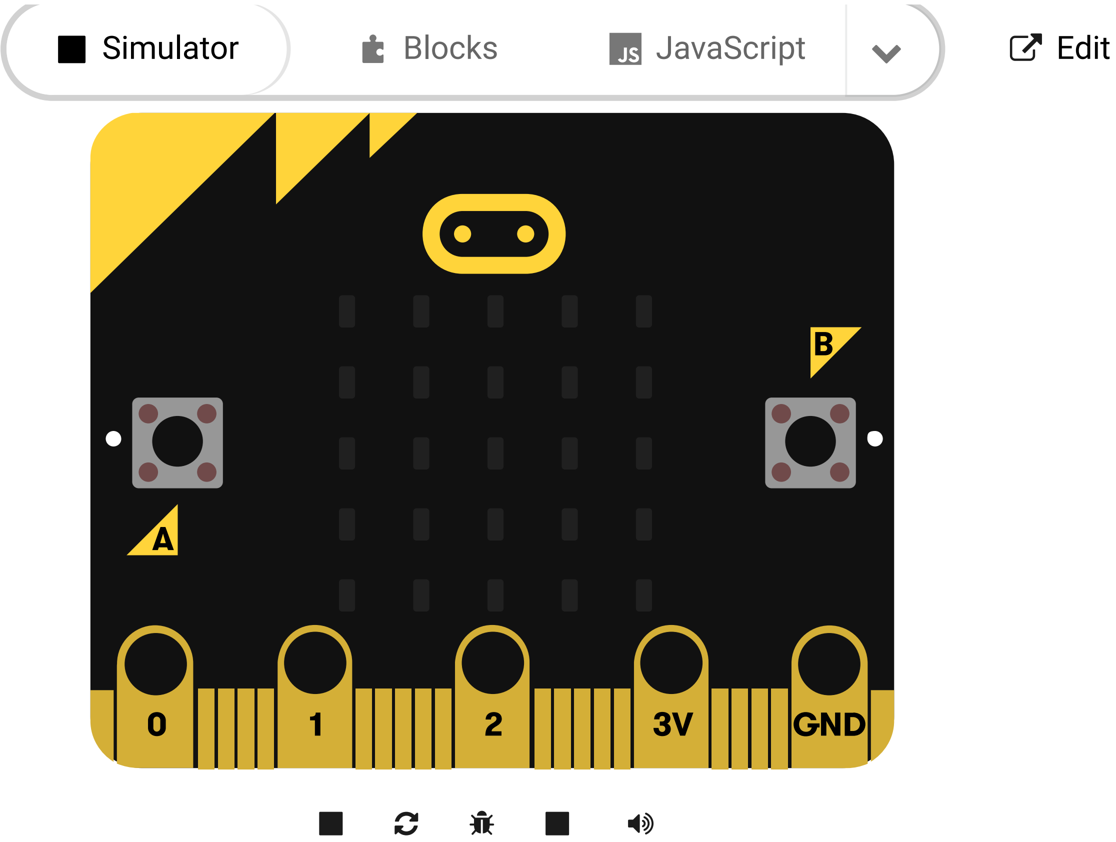
on button B ▾ pressed

reverse at full speed

Program

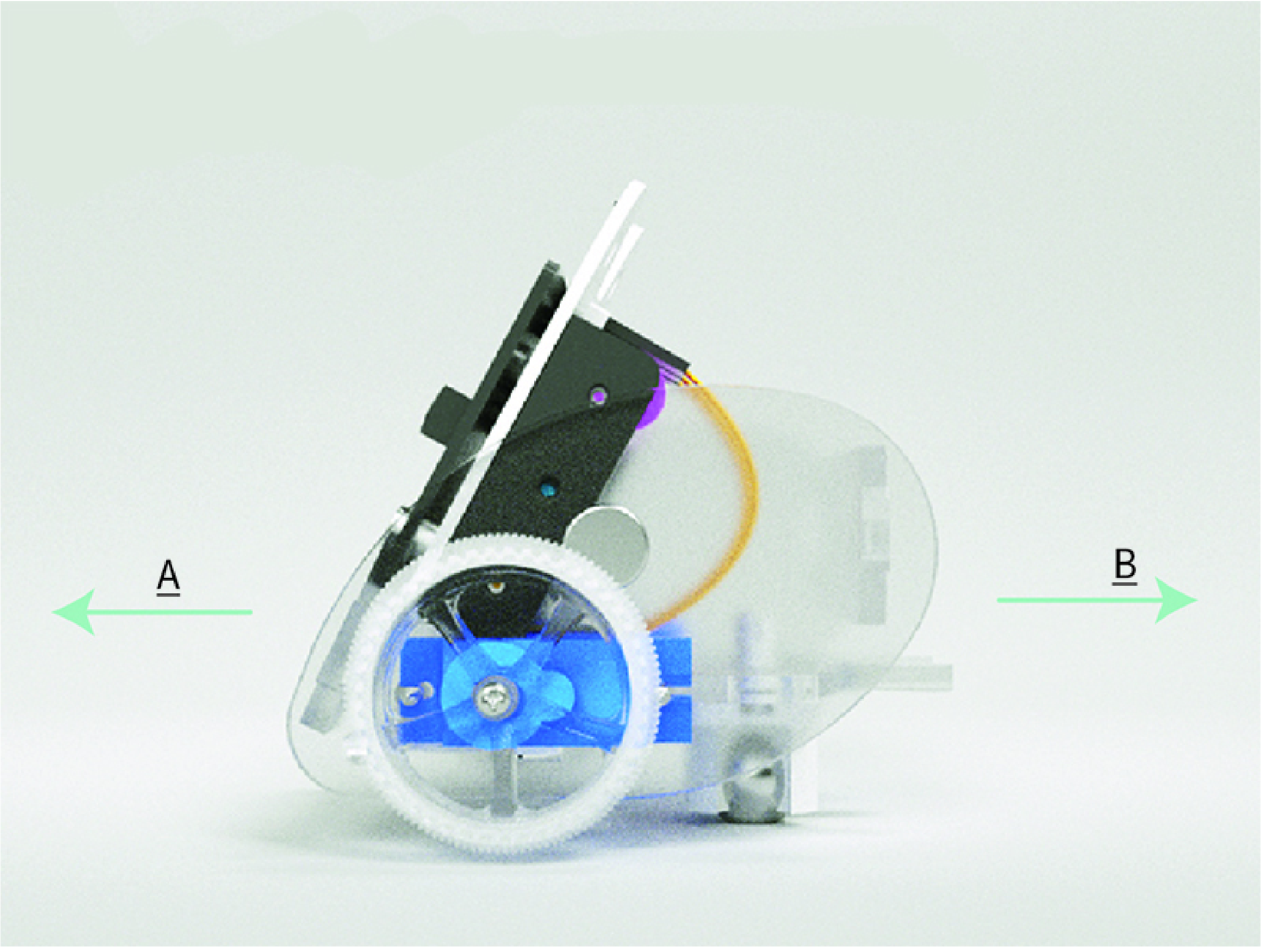
Program Link： https://makecode.microbit.org/_5md9ofDyRigh

If you don’t want to type these code by yourself, you can directly download the whole program from the link below:



9.6. Completed!

- When button A is pressed, the car will go straight at full speed.
- When button B is pressed, the car will reverse at full speed.



9.7. Think

How can you make your car to stop when button A is pressed ?

9.8. Questions

9.9. More Information

10. case 02 Make a shape

10.1. Our Goal

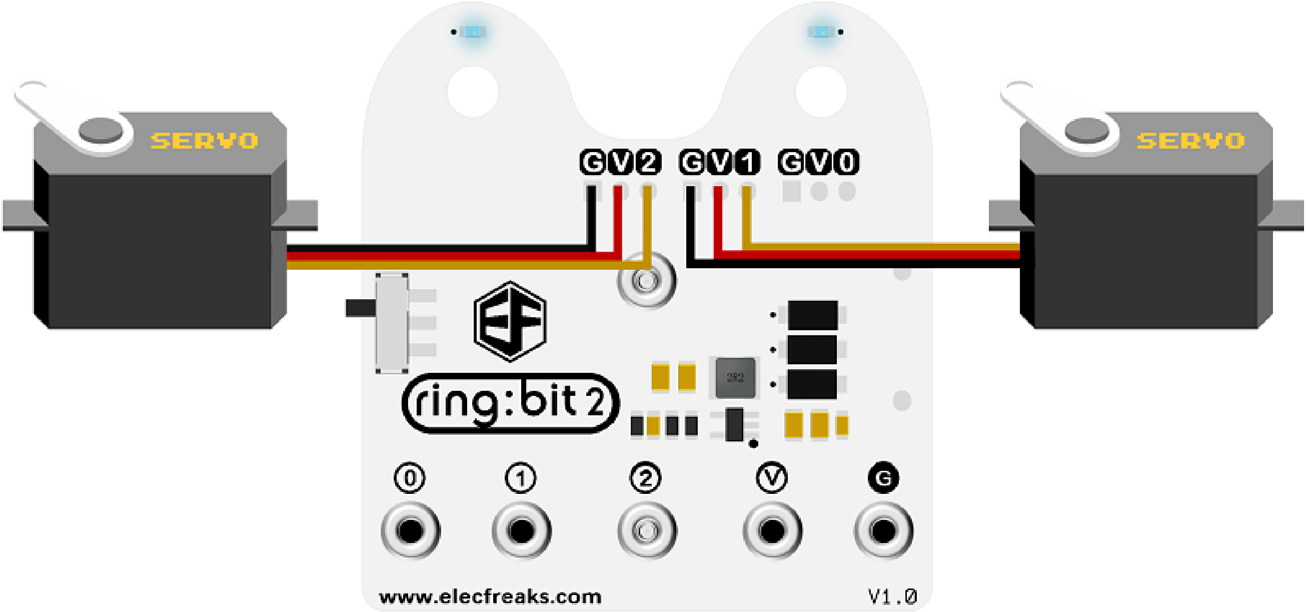
- To make the Ring:bit Car turn at an angle.

10.2. Requiered Materials

- 1 x Ring:bit Car

10.3. Hardware Connect

- Connect the left wheel servo to P1 of the Ring:bit expansion board and the right wheel servo to P2.



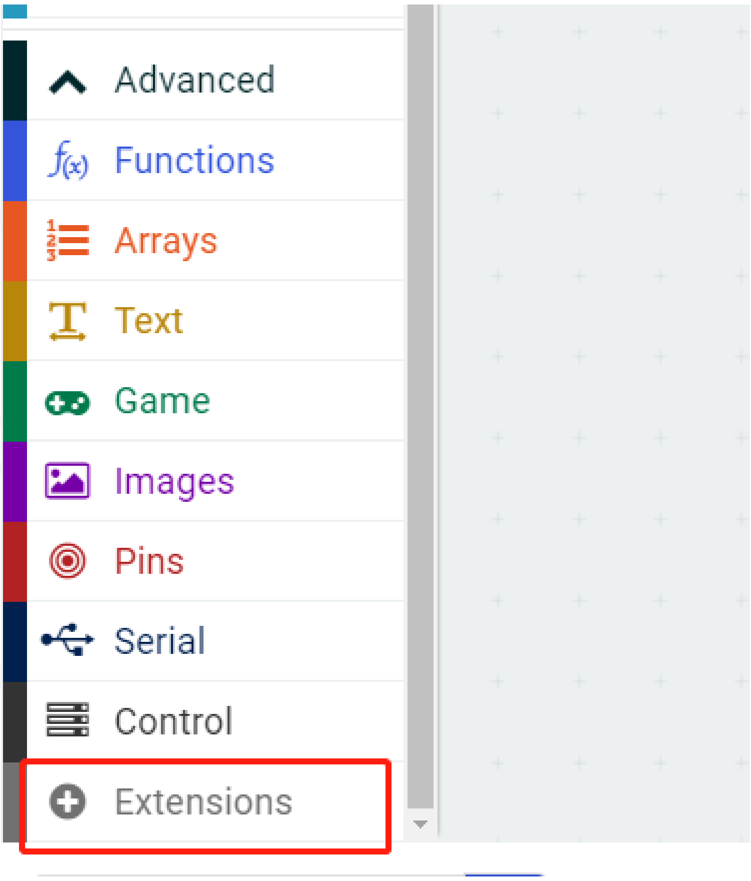
10.4. Software

makecode

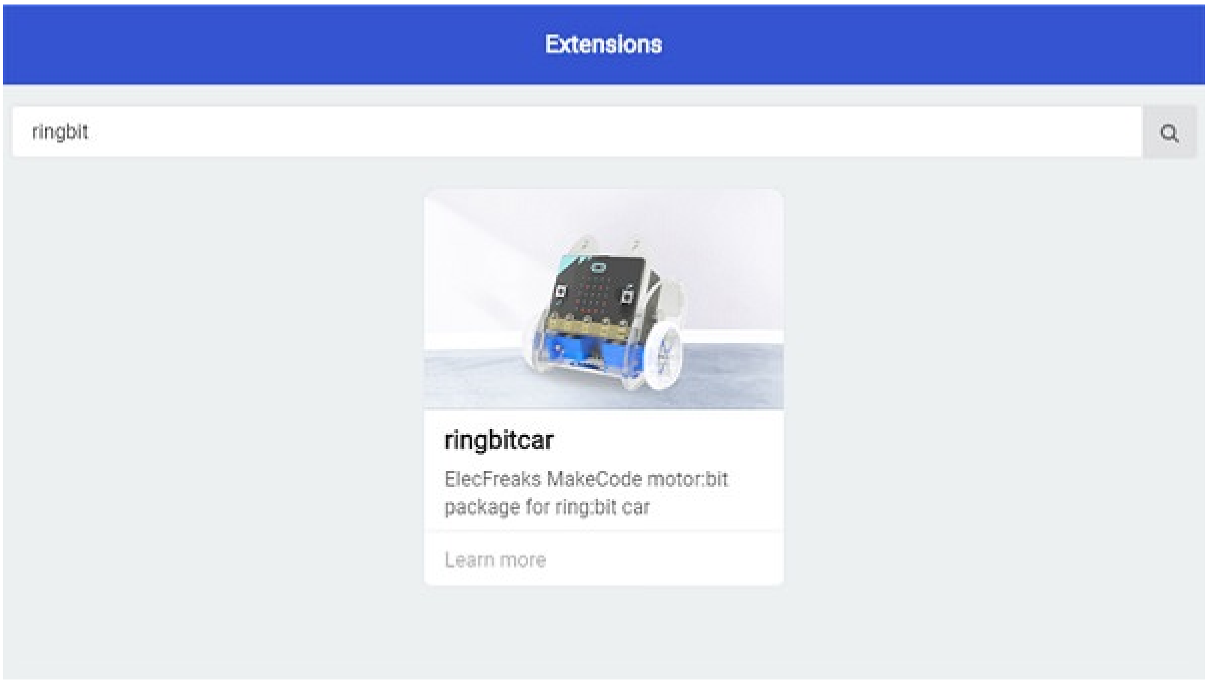
10.5. Coding

Step 1

- Click on “Advanced” in the MakeCode Drawer to see more code sections.



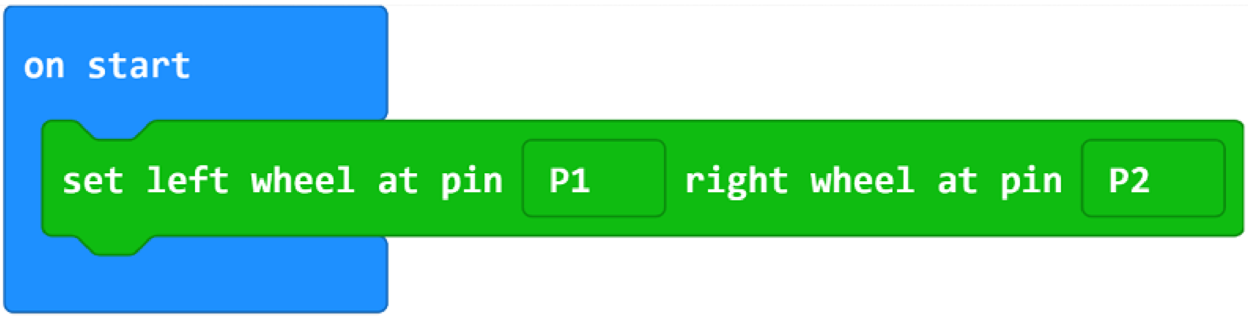
- Search for “ringbitcar” and click on the ring:bit car package to add it to your project. (As below picture)



Note： If you get a warning telling you some packages will be removed because of incompatibility issues, either follow the prompts or create a new project in the Project file menu.

Step 2

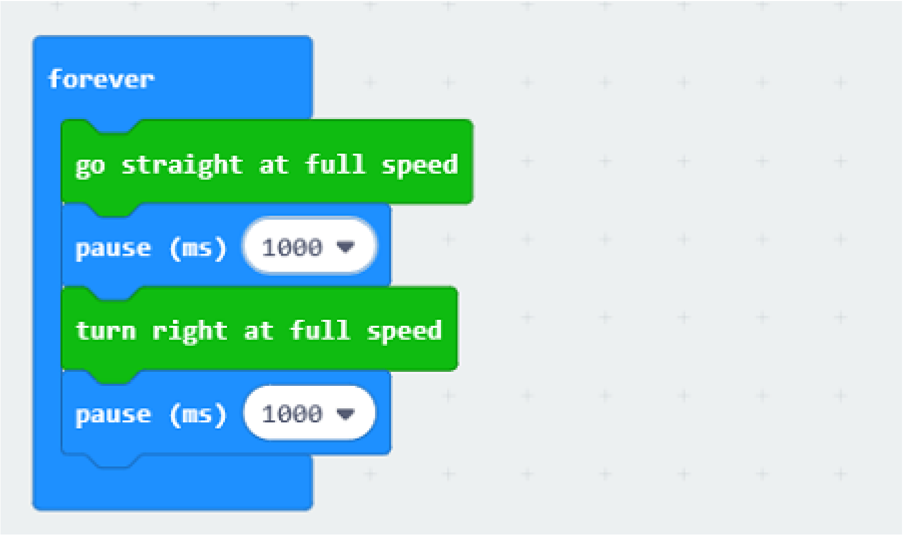
- Snap the block “set left wheel at pin P1 right wheel at pin P2” into the on start block.
- The port number is based on the actual servo connection port.



```
on start
    set left wheel at pin P1 right wheel at pin P2
```

Step 3

- Within the `forever` block, snap below blocks in line:
- `go straight at full speed` block
- `pause` block
- `turn right at full speed` block
- `pause` block

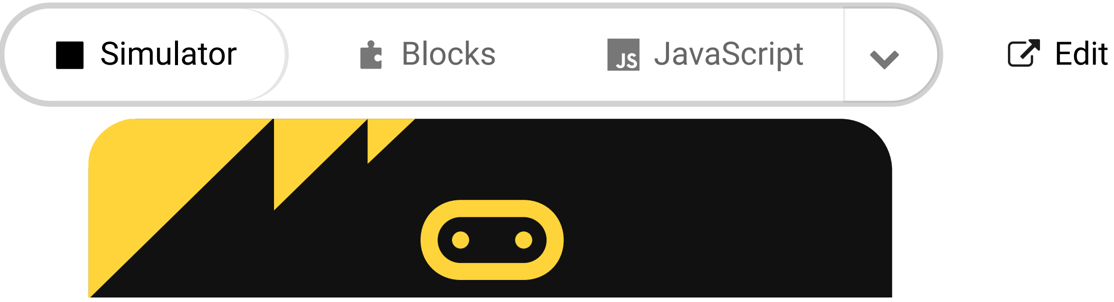


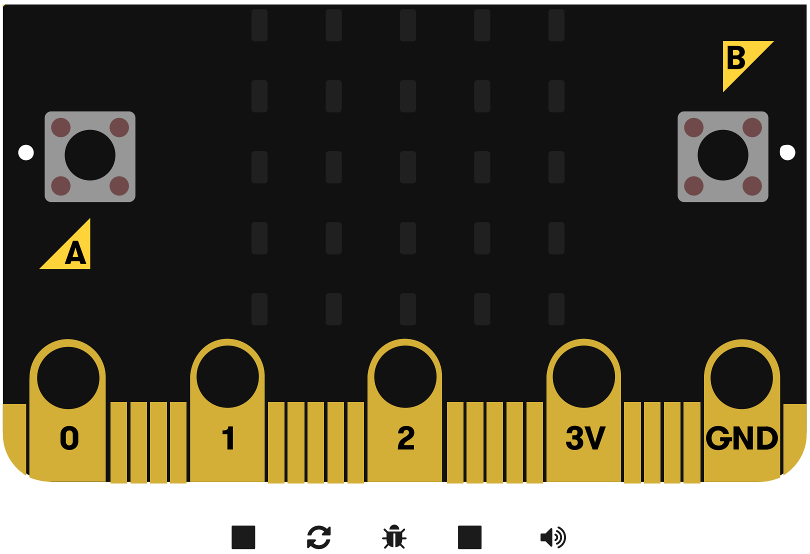
```
forever
    go straight at full speed
    pause (ms) 1000
    turn right at full speed
    pause (ms) 1000
```

Program

Program Link： https://makecode.microbit.org/_iPWL19C26CYe

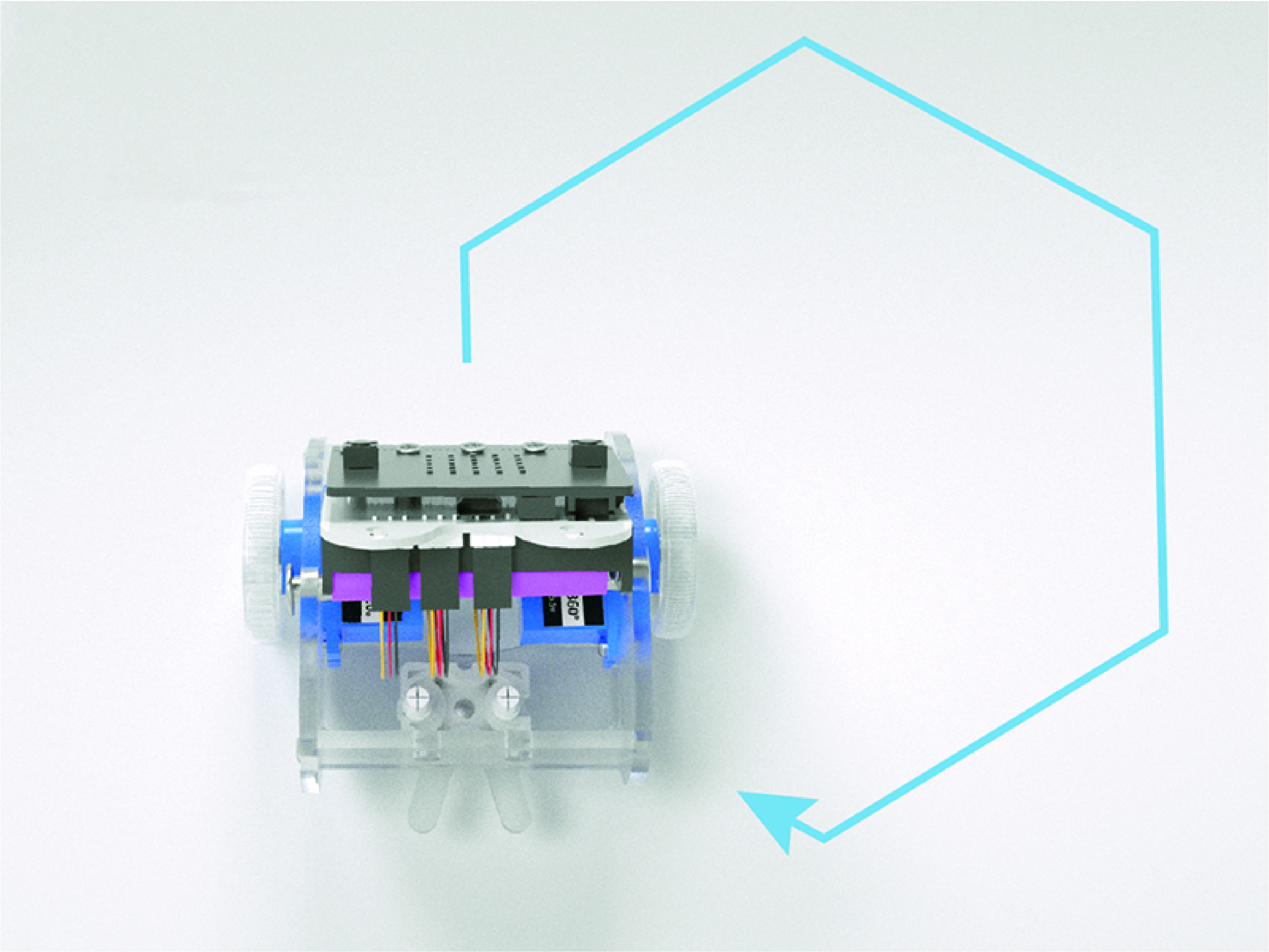
If you don’t want to type these code by yourself, you can directly download the whole program from the link below:





10.6. Result

- The car goes straight and turns right, then goes straight.



10.7. Think

- How can you make your car dance?

10.8. Questions

10.9. More Information

11. case 03 Turn at an angle

11.1. Our Goal

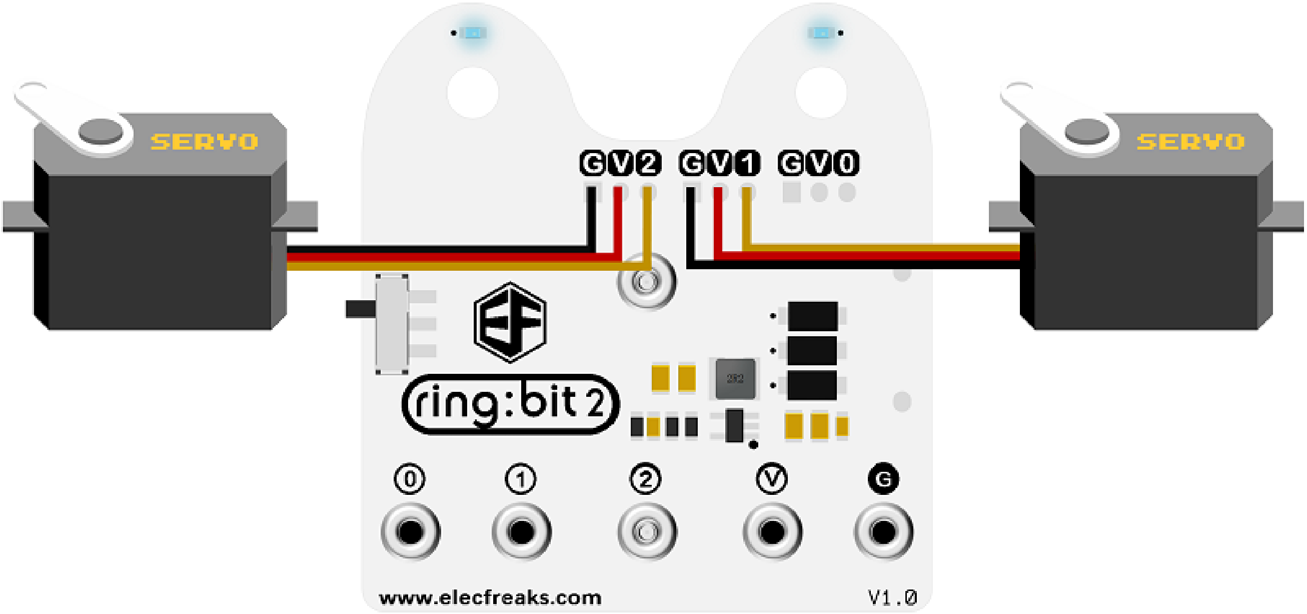
- To make the Ring:bit Car draw a circle in the clockwise direction.

11.2. Requiered Materials

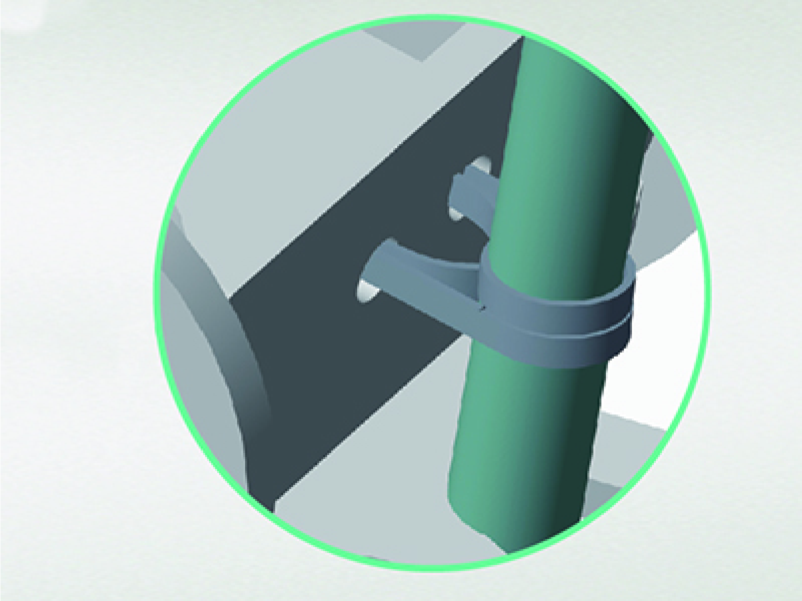
- 1 x Ring:bit Car

11.3. Hardware Connect

- Connect the left wheel servo to P1 of the Ring:bit expansion board and the right wheel servo to P2.



- Use a rubber band to fix a pencil on the base board of the Ring:bit car.



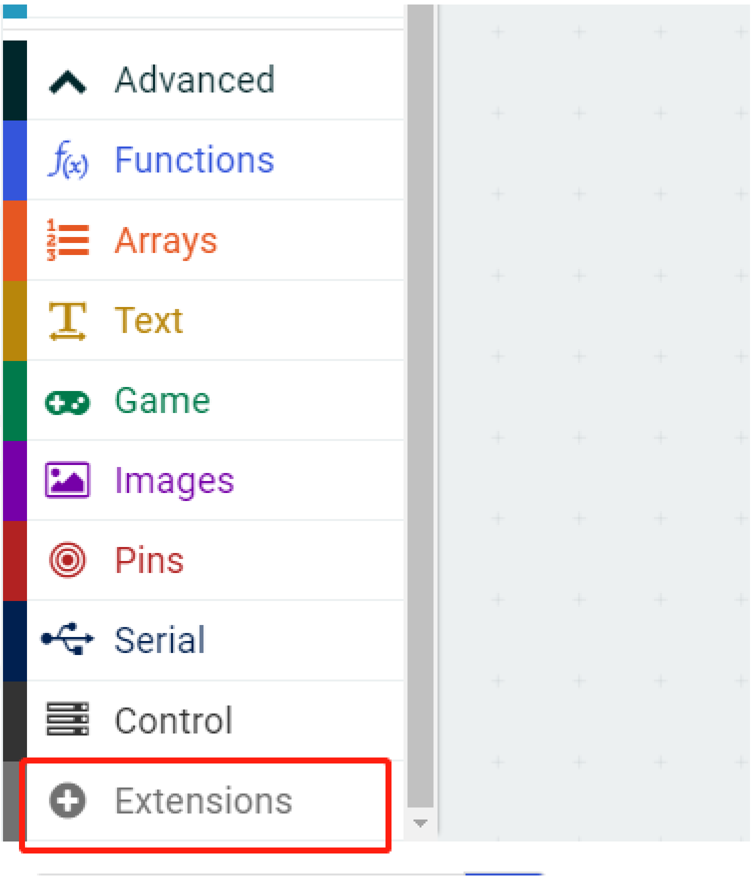
11.4. Software

makecode

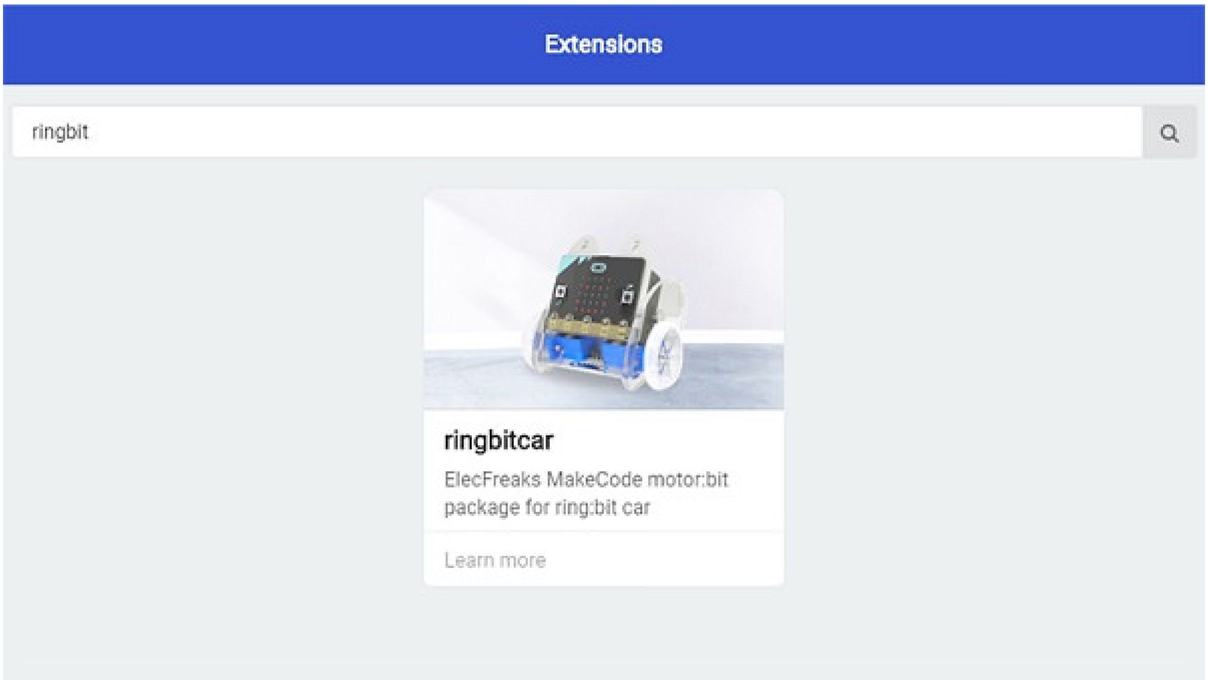
11.5. Coding

Step 1

- Click on “Advanced” in the MakeCode Drawer to see more code sections.



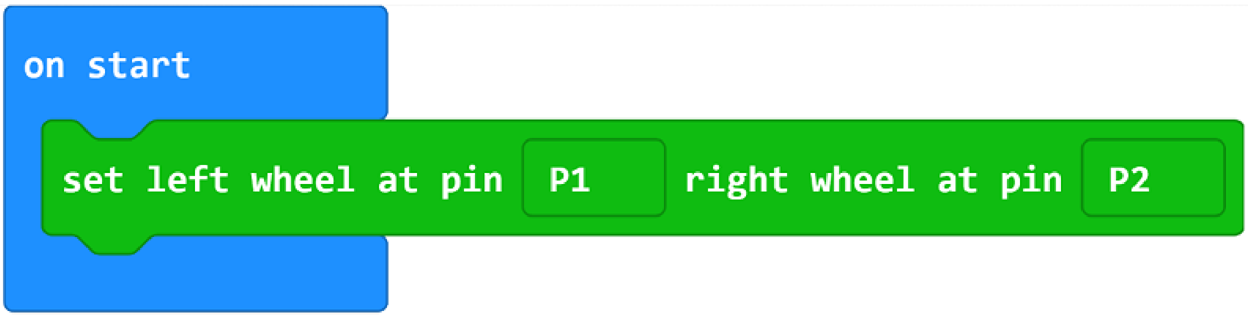
- Search for “ringbitcar” and click on the ring:bit car package to add it to your project. (As below picture)



Note： If you get a warning telling you some packages will be removed because of incompatibility issues, either follow the prompts or create a new project in the Project file menu.

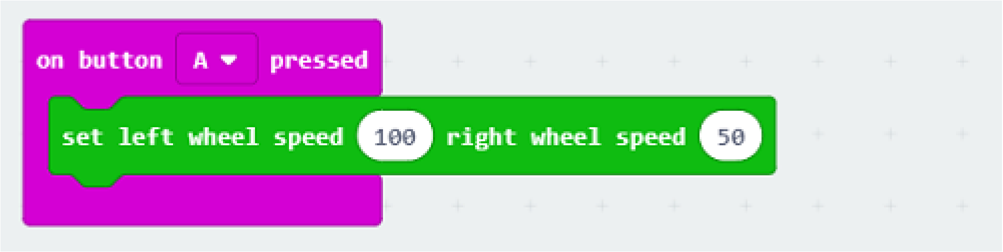
Step 2

- Snap the block `set left wheel at pin P1 right wheel at pin P2` into the on start block.
- The port number is based on the actual servo connection port.



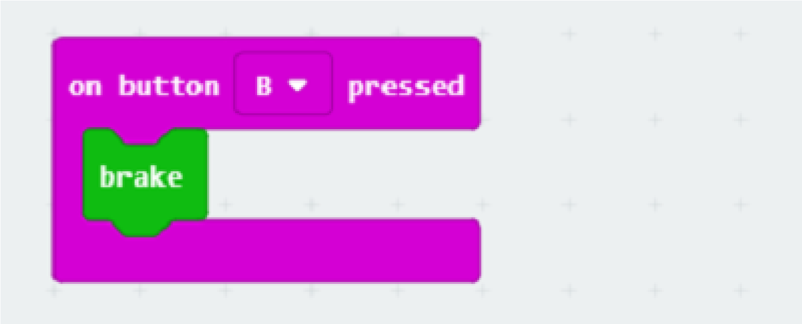
Step 3

- Snap the `set left wheel speed right wheel speed` block into the `on button A pressed` block.
- Set the left wheel’s speed to 10 and the right to 50.



Step 4

- Snap the `brake` block into the `on button B pressed` block.

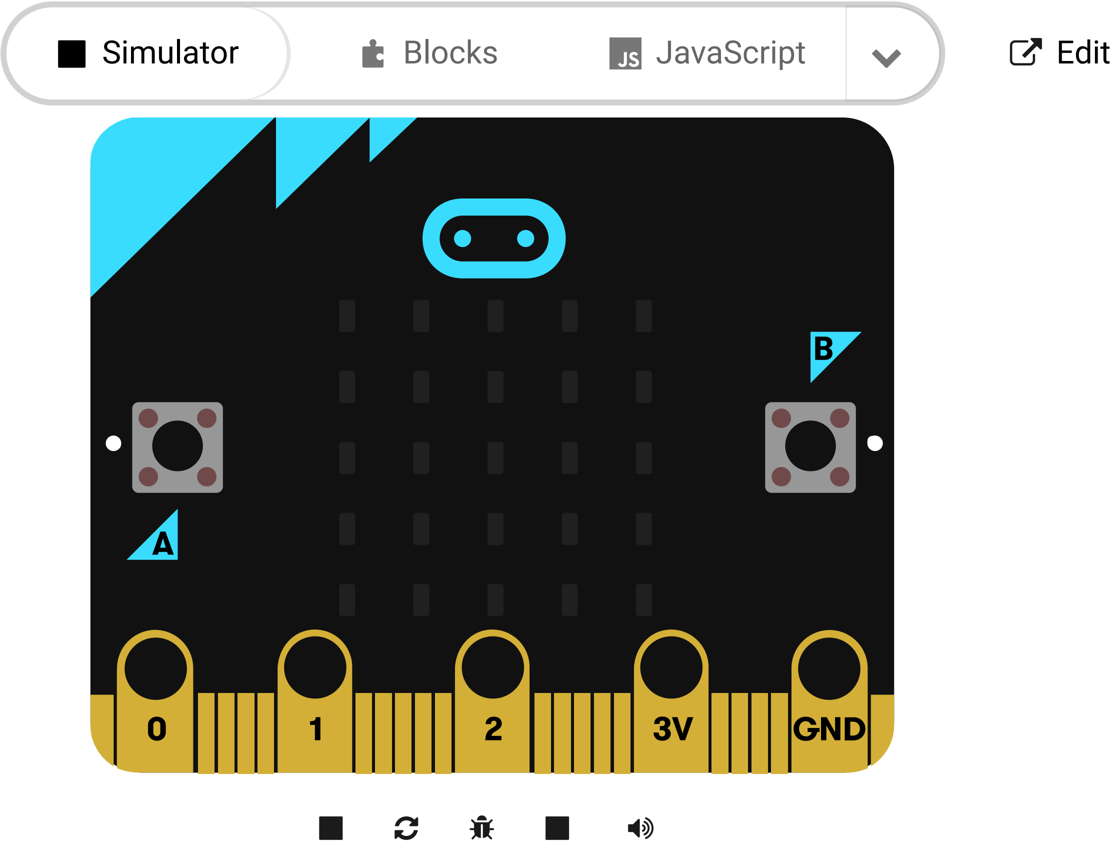


```
on button B ▾ pressed
    brake
```

Program

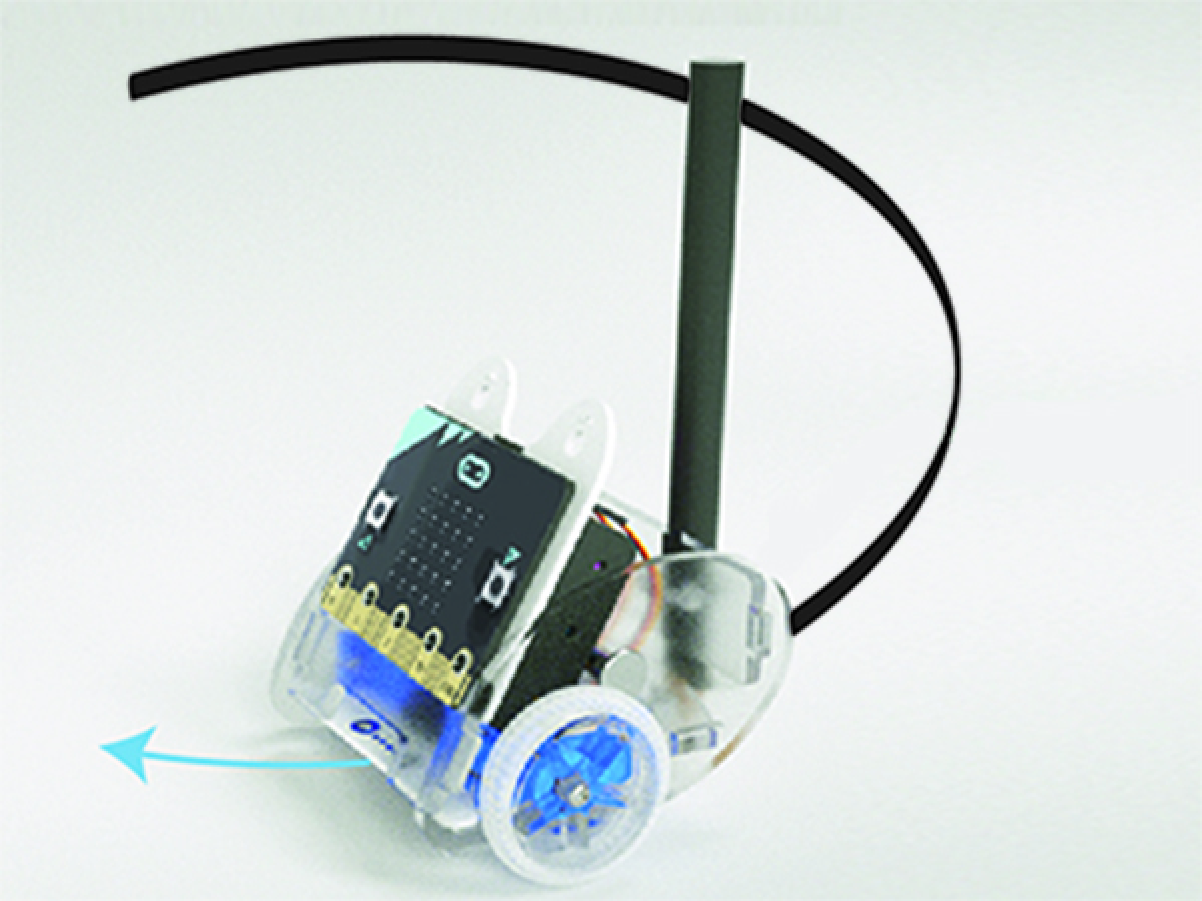
Program Link：https://makecode.microbit.org/_YejCA2AJDDVV

If you don't want to type these code by yourself, you can directly download the whole program from the link below:



11.6. Result

- On button A pressed, the car draws a circle.
- On button B pressed, stop car.



11.7. Think

- How can you make your car draw an eight?

11.8. Questions

11.9. More Information

12. case 04 Here comes the police

12.1. Our Goal

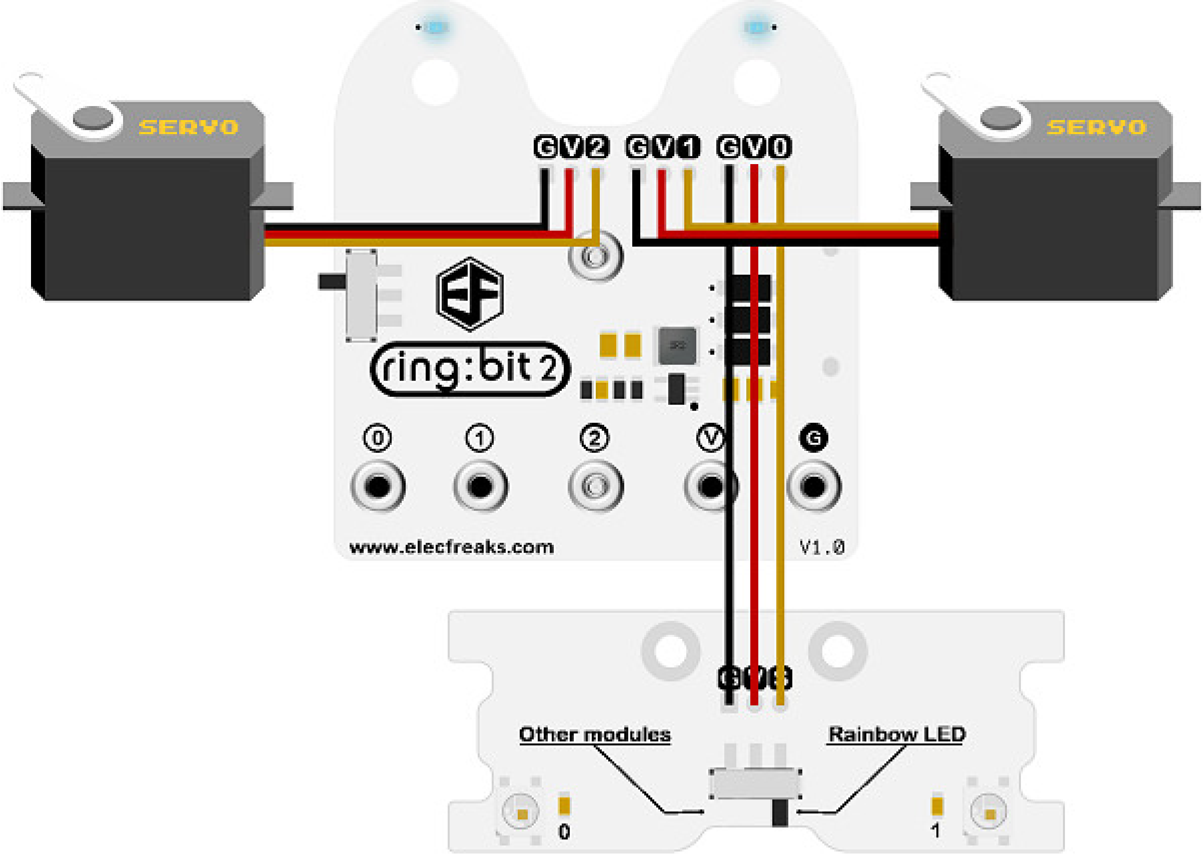
- To change the Ring:bit car to a police car.

12.2. Requiered Materials

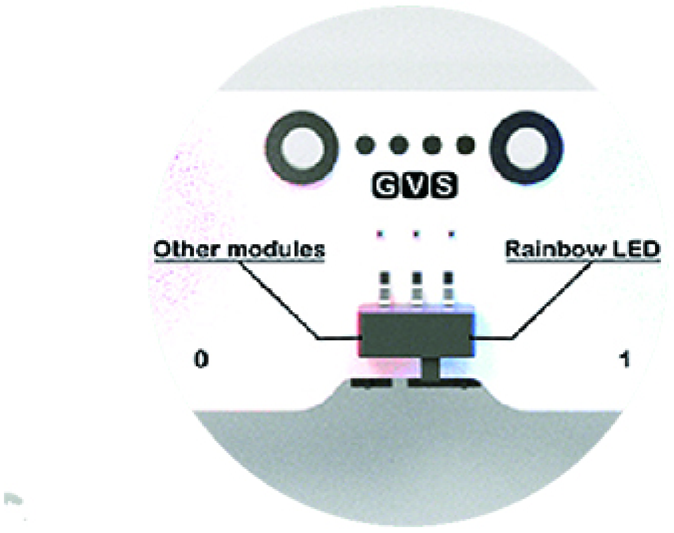
- 1 x Ring:bit Car

12.3. Hardware Connect

- Connect the left wheel servo to P1 of the Ring:bit expansion board and the right wheel servo to P2.
- Connect the ring:bit car special expansion board to the P0.



- Slide the clip switch of the special expansion board to the Rainbow LED.



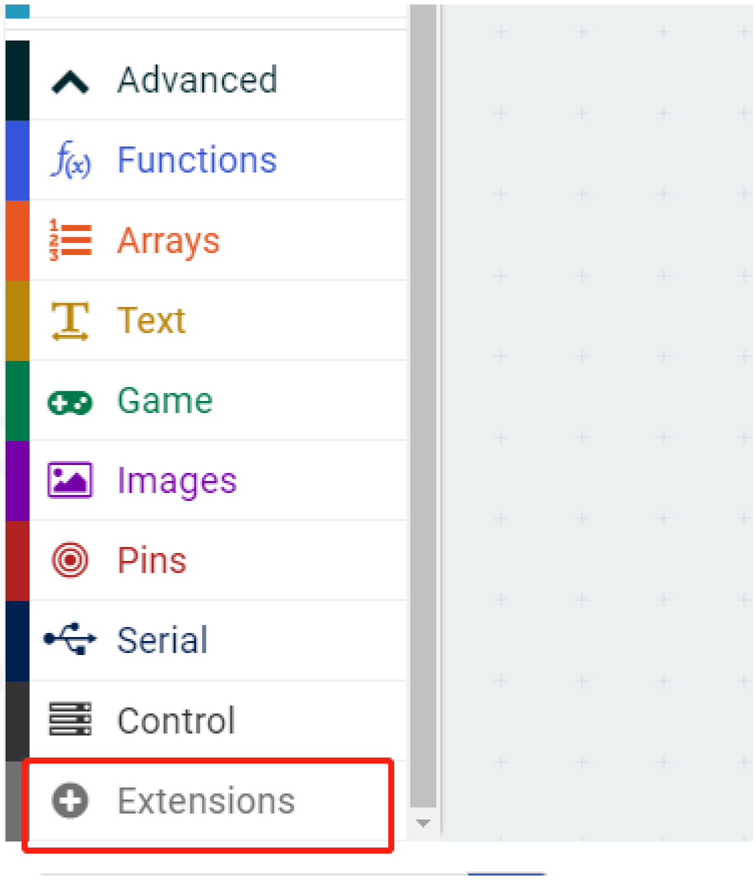
12.4. Software

makecode

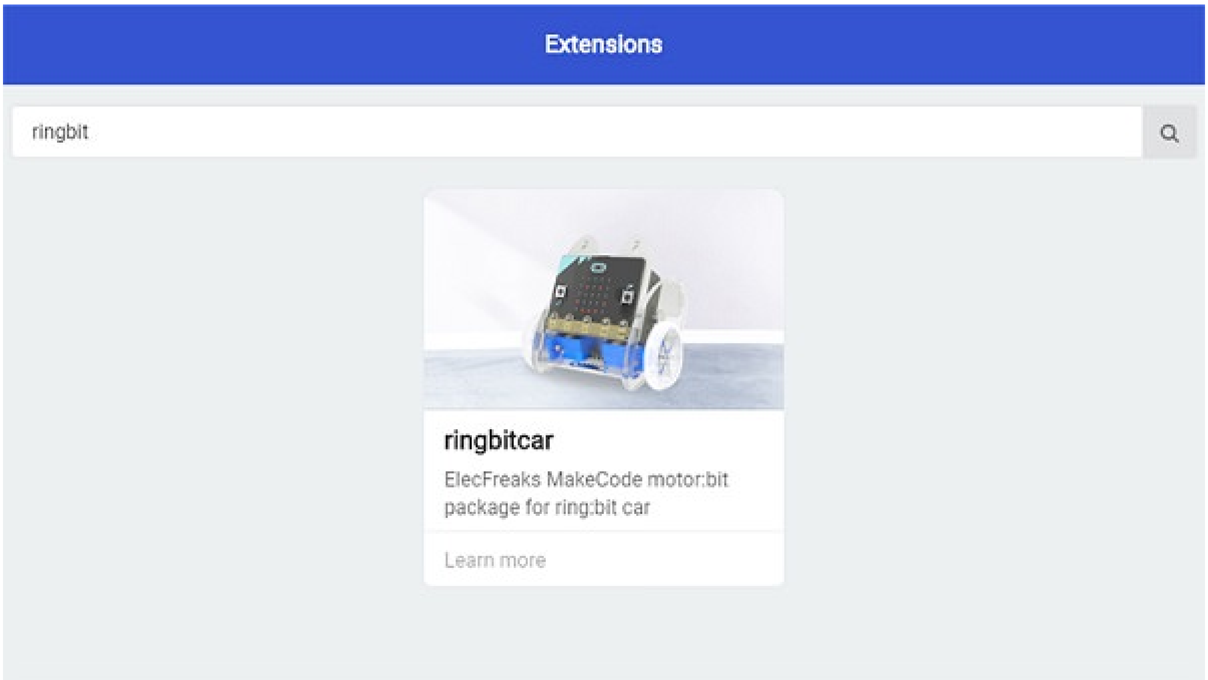
12.5. Coding

Step 1

- Click on “Advanced” in the MakeCode Drawer to see more code sections.



- Search for “ringbitcar” and click on the ring:bit car package to add it to your project. (As below picture)



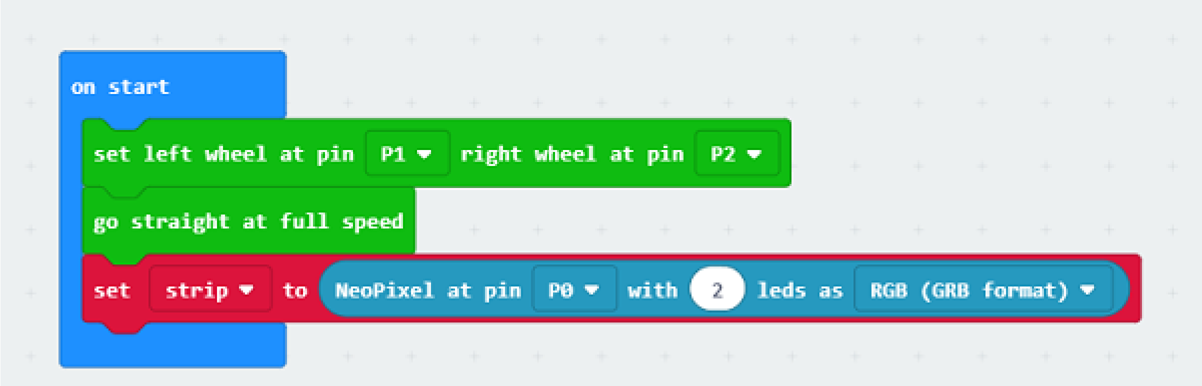
ringbitcar

ElecFreaks MakeCode motor:bit package for ring:bit car

Note： If you get a warning telling you some packages will be removed because of incompatibility issues, either follow the prompts or create a new project in the Project file menu.

Step 2

- Snap the block `set left wheel at pin P1 right wheel at pin P2` to the `On start` block. The port number is based on the actual servo connection port.
- Go straight at full speed.
- Set the `2` Rainbow LED of the `P0` to `RGB` color.

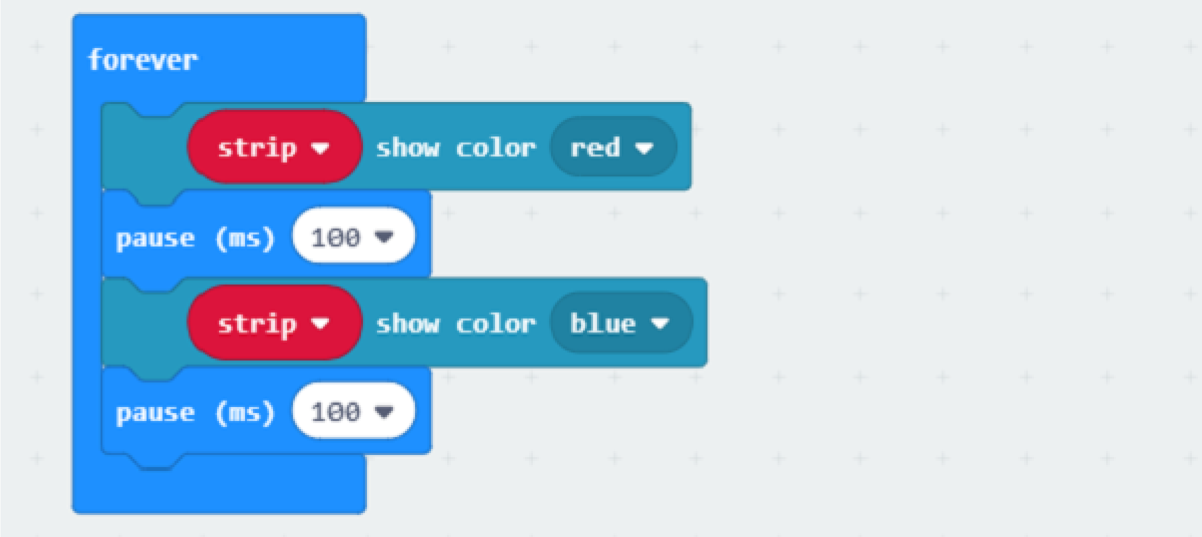


```
on start
    set left wheel at pin P1 ▾ right wheel at pin P2 ▾
    go straight at full speed
    set strip ▾ to NeoPixel at pin P0 ▾ with 2 leds as RGB (GRB format) ▾
```

Step 3

- Within the `forever` block, snap below blocks in line:
- `show color red` block.
- `pause(ms) 100` block.
- `show color blue` block.

- `pause(ms) 100` block.

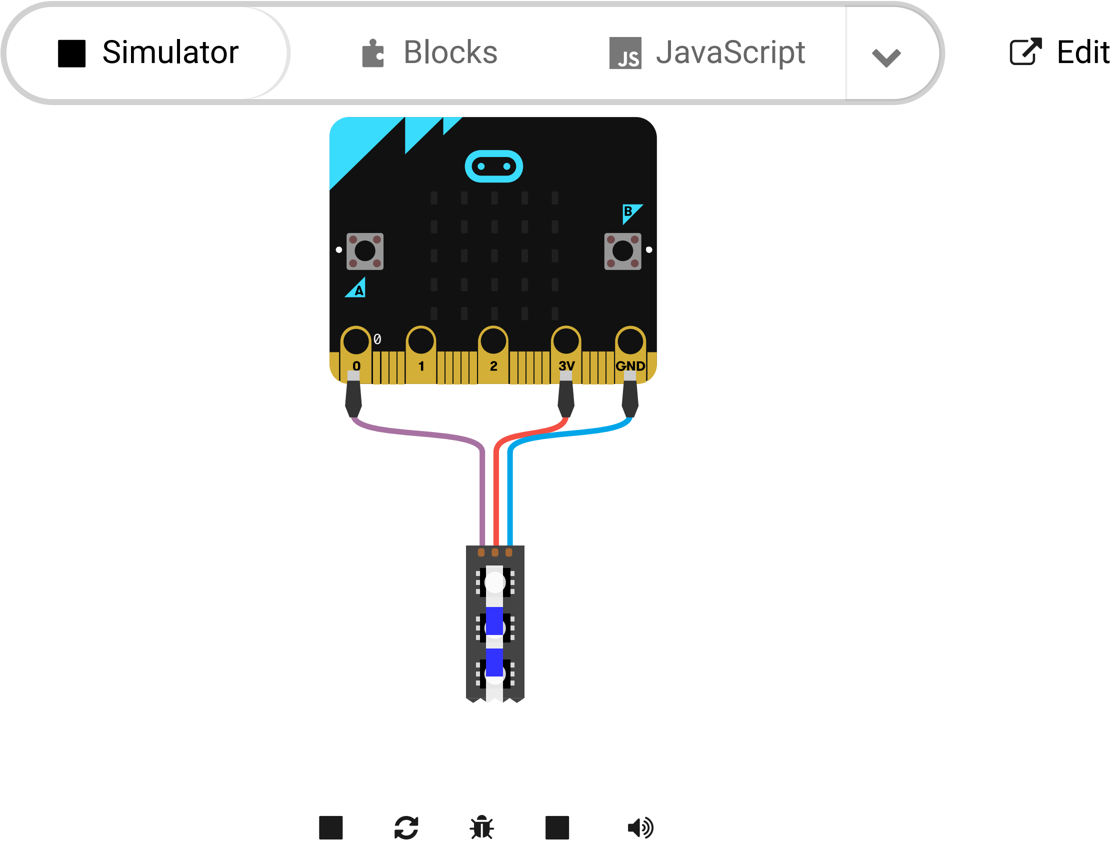


```
forever
    strip ▾ show color red ▾
    pause (ms) 100 ▾
    strip ▾ show color blue ▾
    pause (ms) 100 ▾
```

Program

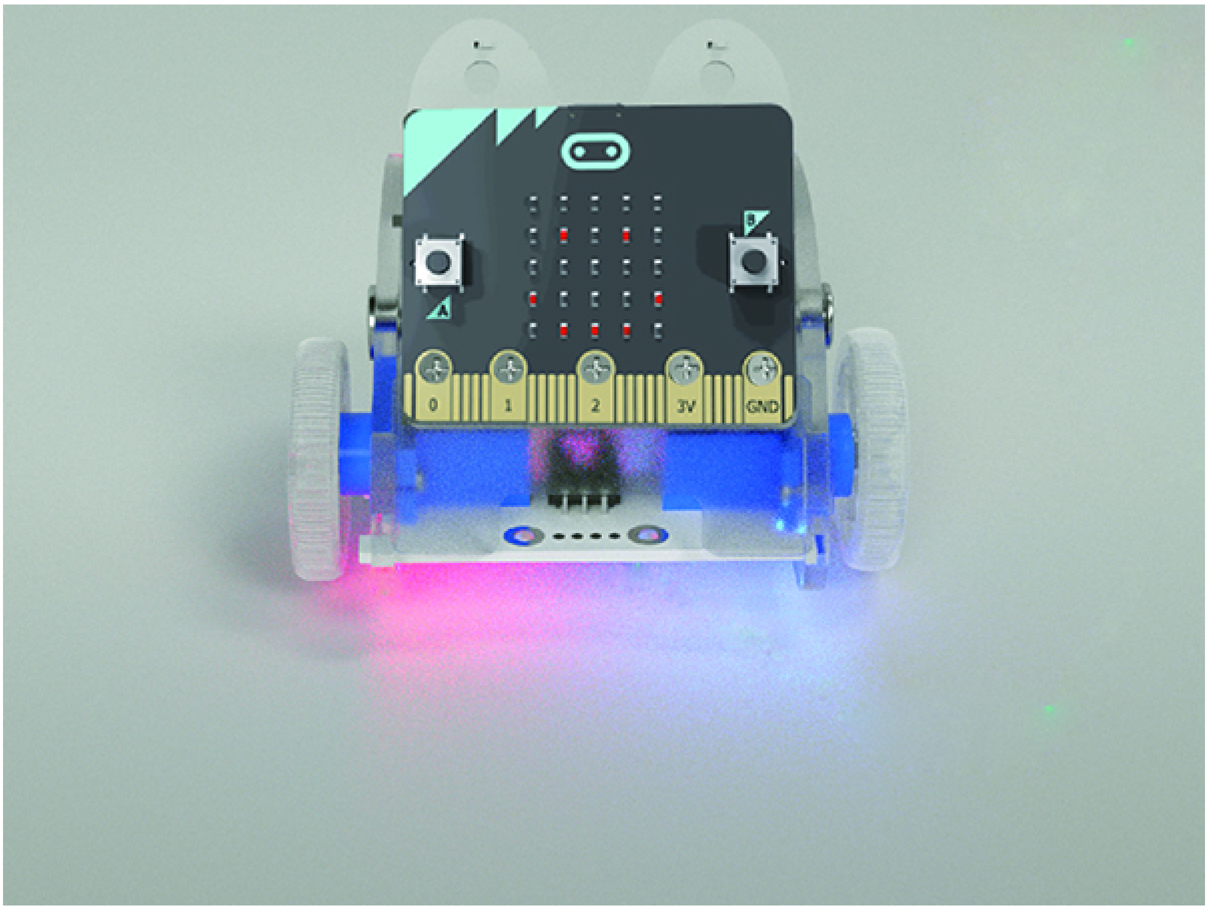
Program Link: https://makecode.microbit.org/_AvMC6j86A5ym

If you don’t want to type these code by yourself, you can directly download the whole program from the link below:



12.6. Result

- The car goes straigt at full speed at red and blue flash.



12.7. Think

- How can you make youe car at yellow and white flash?

12.8. Questions

12.9. More Information

13. case 05 Crazy dance

13.1. Our Goal

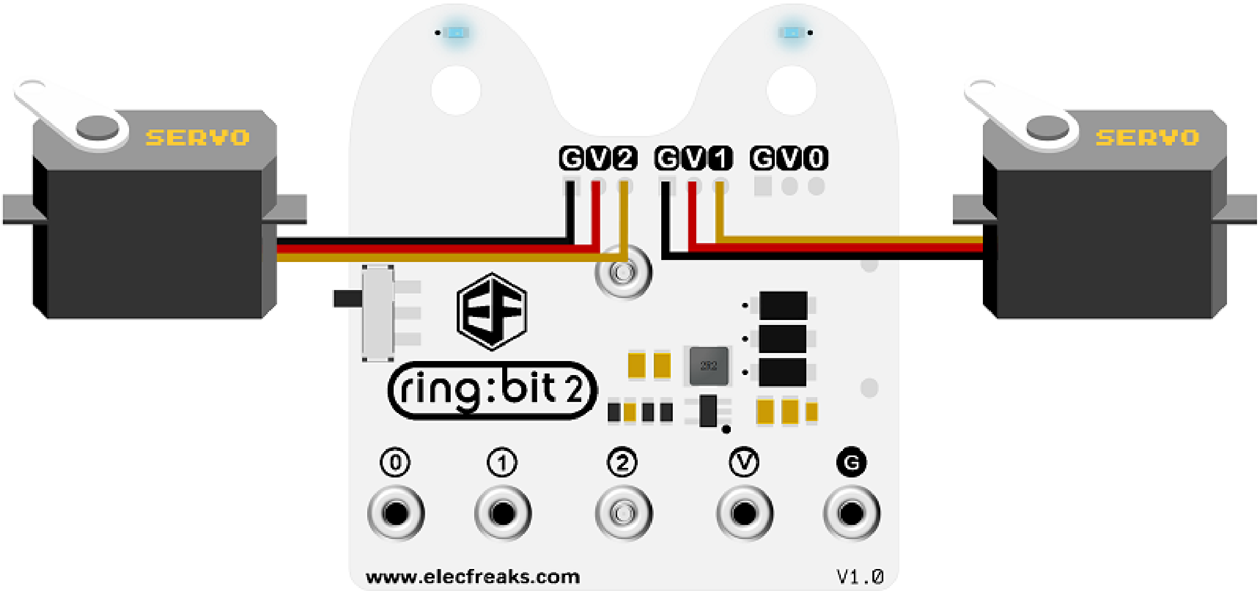
- To make the Ring:bit car to dance in free style.

13.2. Requiered Materials

- 1 x Ring:bit Car

13.3. Hardware Connect

- Connect the left wheel servo to P1 of the Ring:bit expansion board and the right wheel servo to P2.



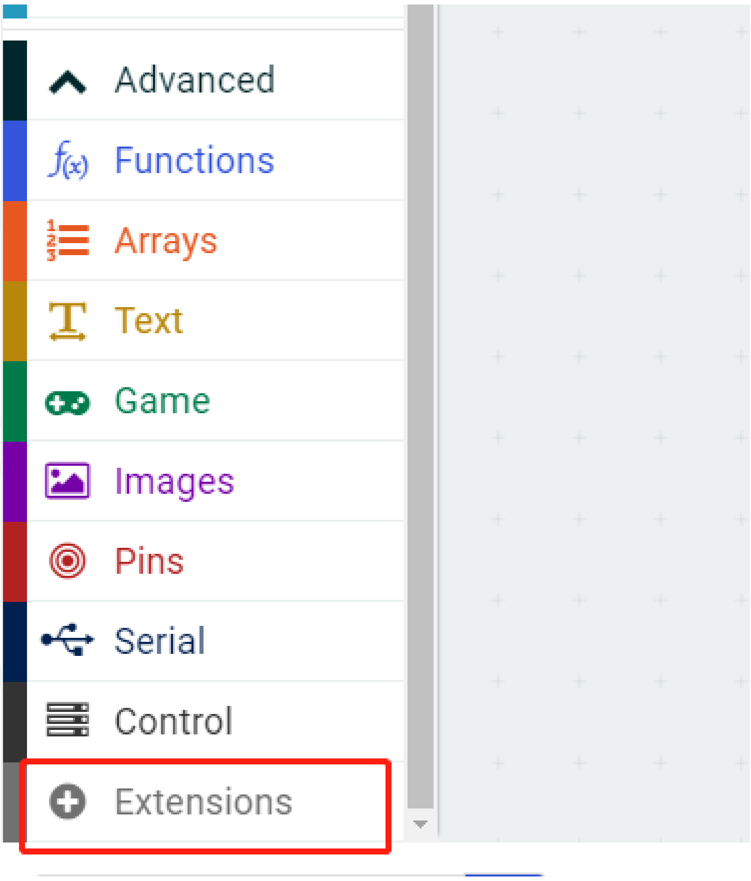
13.4. Software

makecode

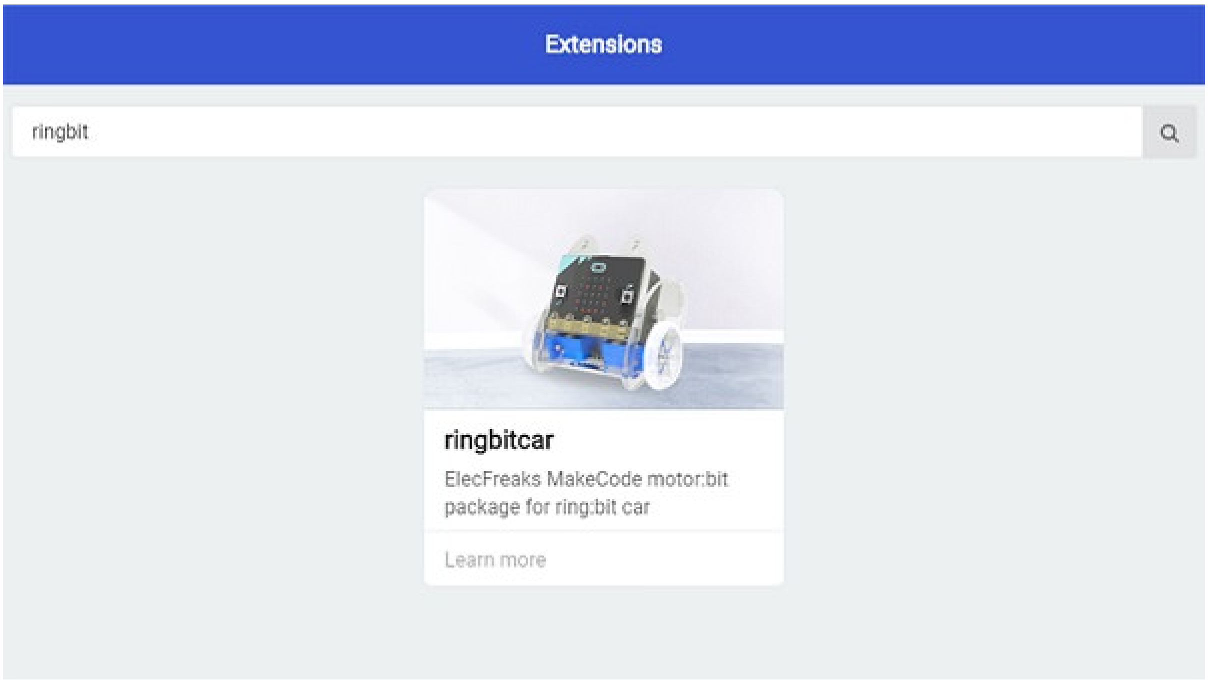
13.5. Coding

Step 1

- Click on “Advanced” in the MakeCode Drawer to see more code sections.



- Search for “ringbitcar” and click on the ring:bit car package to add it to your project. (As below picture)



ringbit

ringbitcar

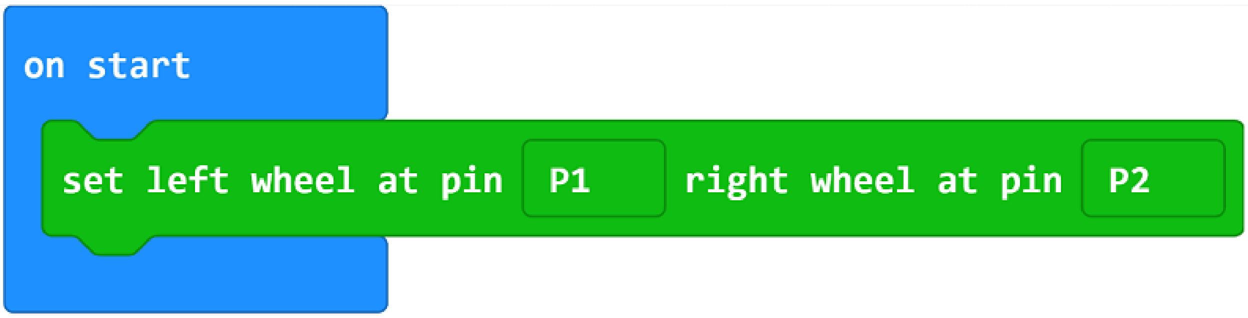
ElecFreaks MakeCode motor:bit package for ring:bit car

Learn more

Note： If you get a warning telling you some packages will be removed because of incompatibility issues, either follow the prompts or create a new project in the Project file menu.

Step 2

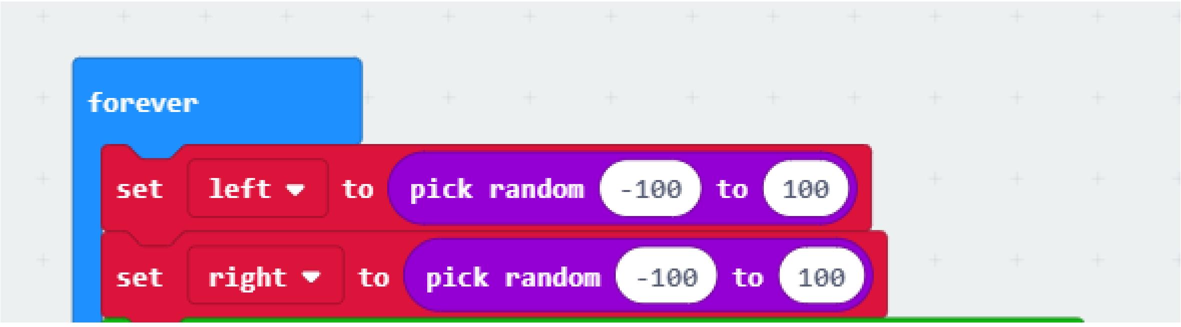
- Snap the block `set left wheel at pin P1 right wheel at pin P2` to the `On start` block.
- The port number is based on the actual servo connection port.



on start
set left wheel at pin P1 right wheel at pin P2

Step 3

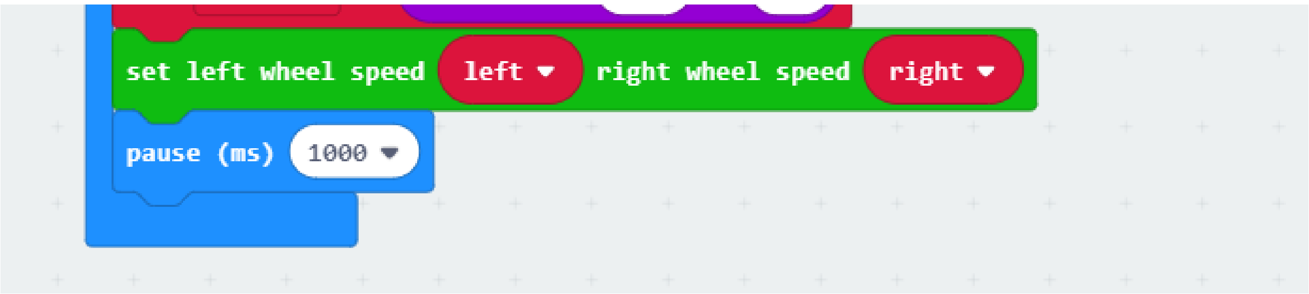
- Snap blocks into `forever` block.
- Set an varieble to `left` and an variable to `right`.
- Generate a random number and assinment to the `left` and the `right`.



forever
set left ▾ to pick random -100 to 100
set right ▾ to pick random -100 to 100

Step 4

- Set the left wheel speed to the variable left abd the right wheel speed to the variable right.
- Pause for one second.

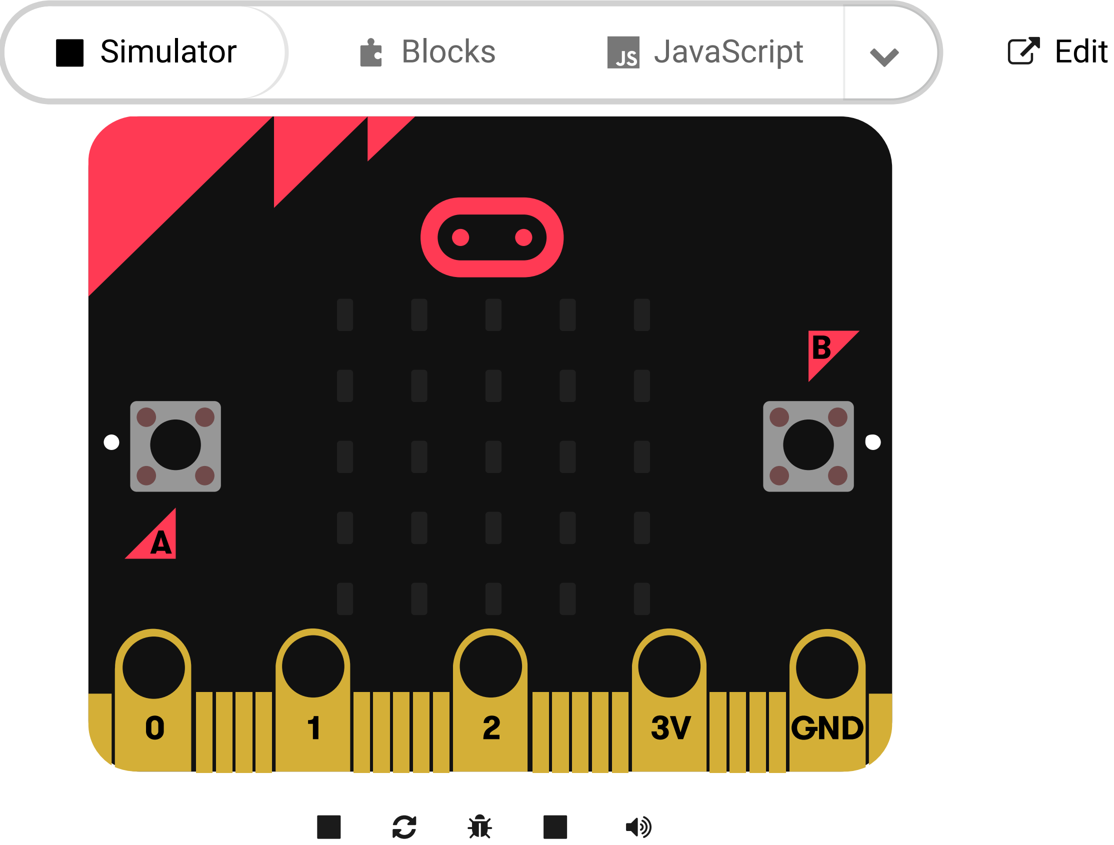


set left wheel speed left ▾ right wheel speed right ▾
pause (ms) 1000 ▾

Program

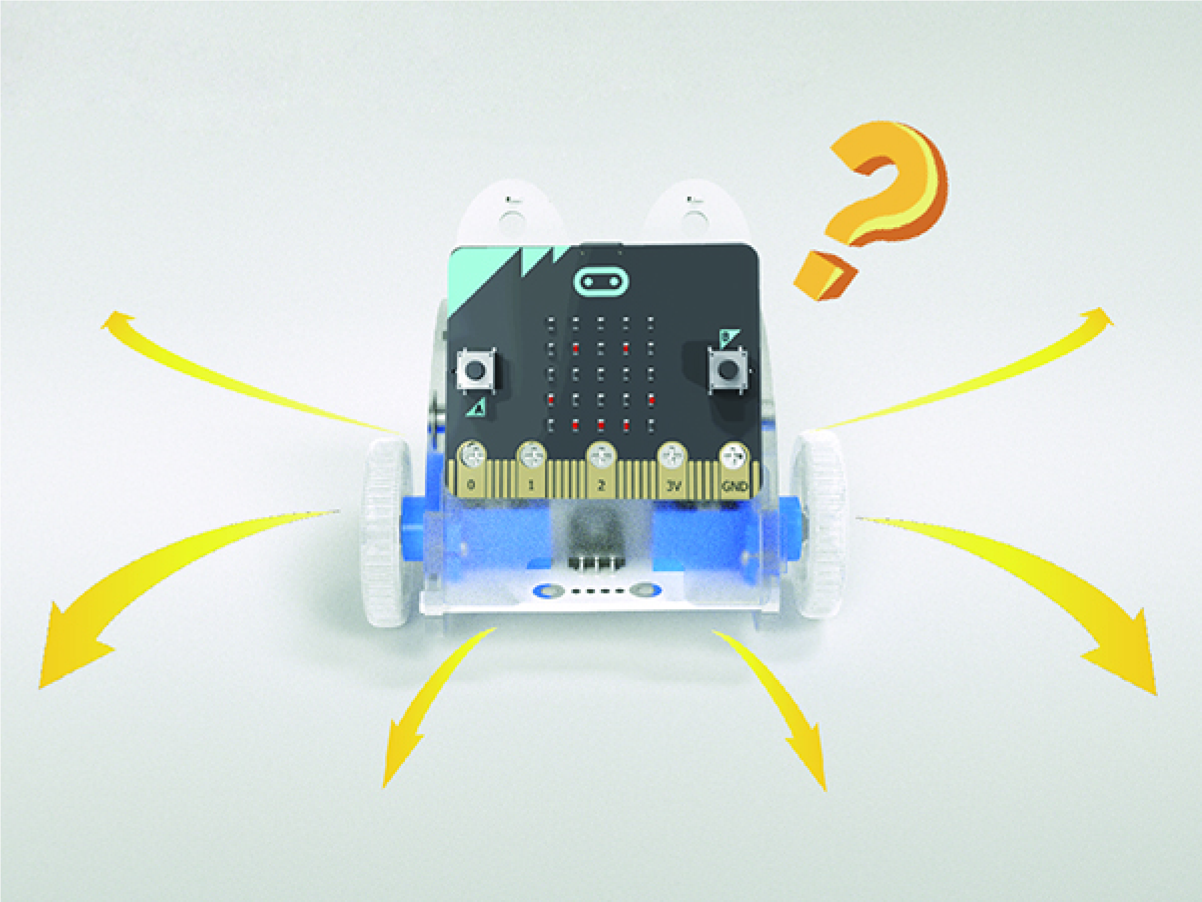
Program Link：https://makecode.microbit.org/_WrC1kWhizc6E

If you don’t want to type these code by yourself, you can directly download the whole program from the link below:



13.6. Result

- The car runs at different speed.



13.7. Think

- How do you ensure the car moves at a slower speed constantly?

13.8. Questions

13.9. More Information

14. case 06 Remote Control

14.1. Our Goal

- Using another micro:bit control the ring:bit car.

14.2. Requiered Materials

- 1 x Ring:bit Car
- 1 x micro:bit

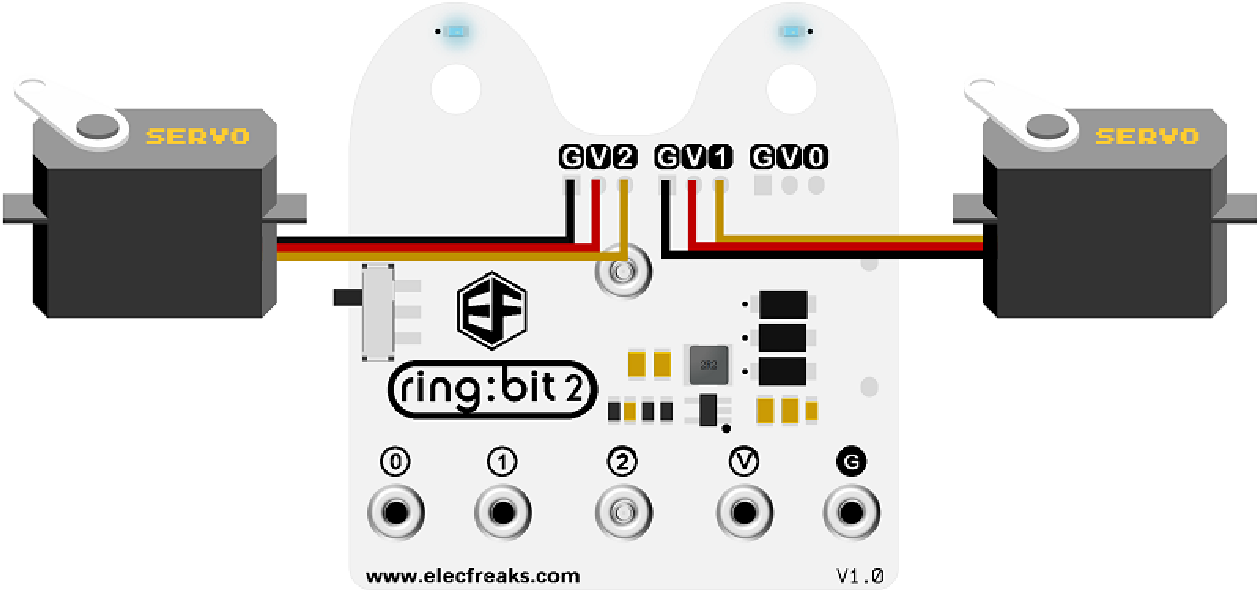
14.3. Background

What is wireless ?

- Wireless technology sends message by radio wave because changes of electricity will generate waves. We can use this to load the message to the wave. Variation of electromagnetic field will generate the electricity when the electromagnetic is arrving the receiver. Then, to extract the message from wave by demodulation to send the message.

14.4. Hardware Connect

- Connect the left wheel servo to P1 of the Ring:bit expansion board and the right wheel servo to P2.



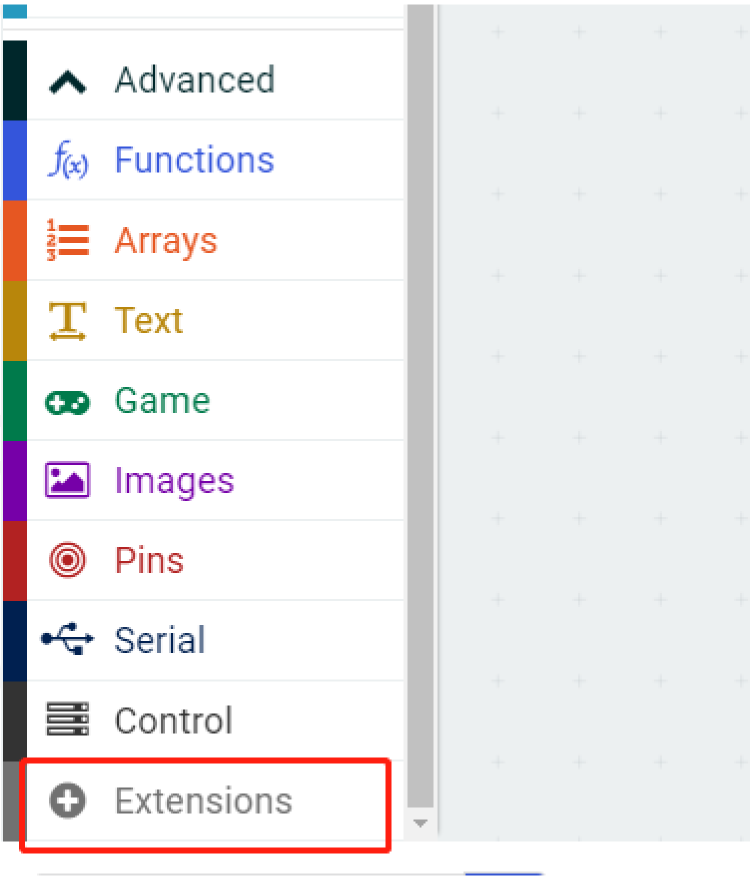
14.5. Software

makecode Online block codinghttps://makecode.microbit.org/#

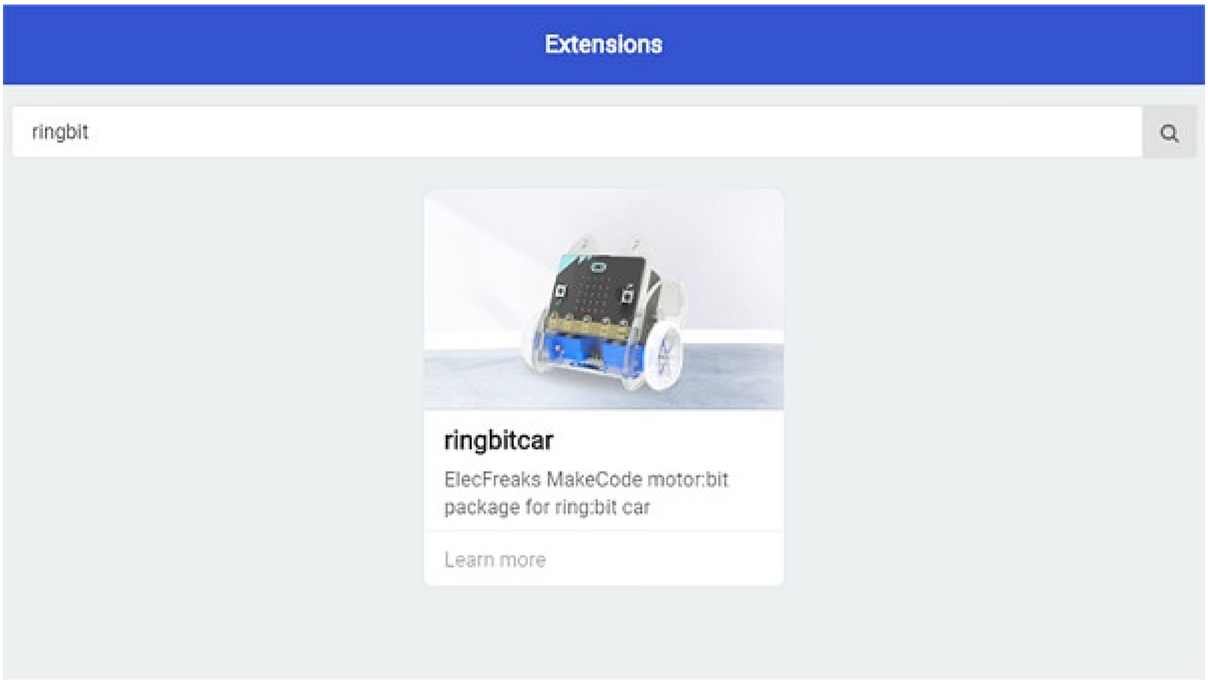
14.6. Coding

Step 1

- Click on “Advanced” in the MakeCode Drawer to see more code sections.



- Search for “ringbitcar” and click on the ring:bit car package to add it to your project. (As below picture)



ringbit

ringbitcar

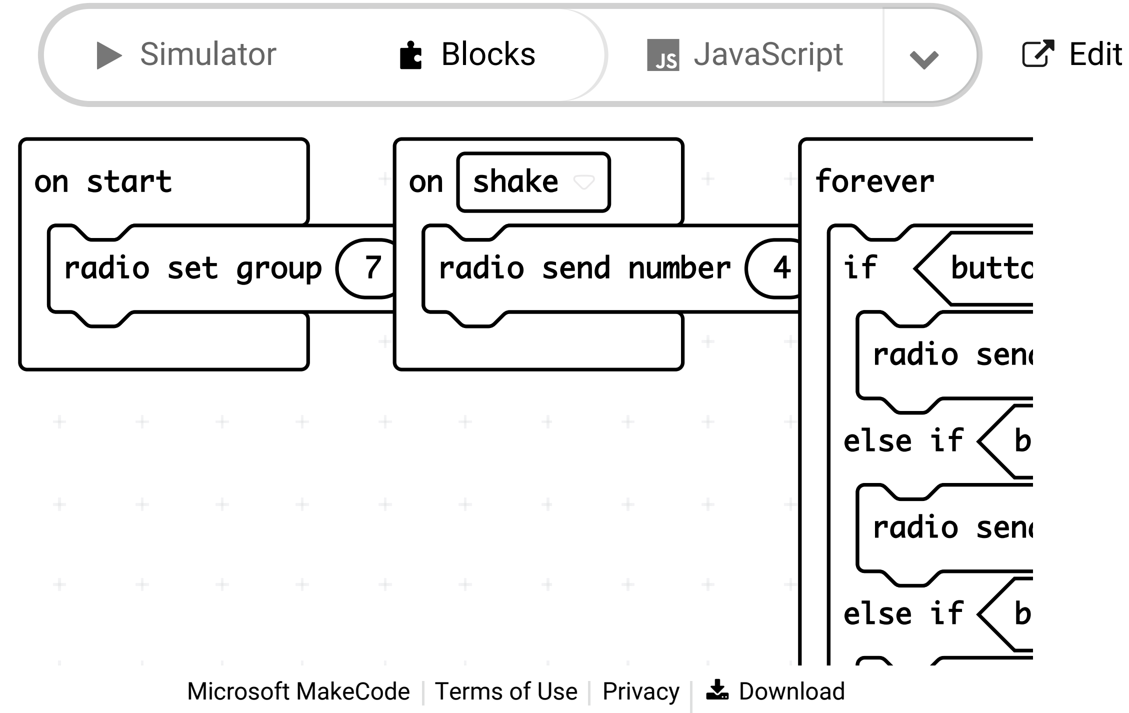
ElecFreaks MakeCode motor:bit package for ring:bit car

Note： If you get a warning telling you some packages will be removed because of incompatibility issues, either follow the prompts or create a new project in the Project file menu.

Remote Control Coding

Program Link： https://makecode.microbit.org/_46FfwWDJwiyy

If you don’t want to type these code by yourself, you can directly download the whole program from the link below:

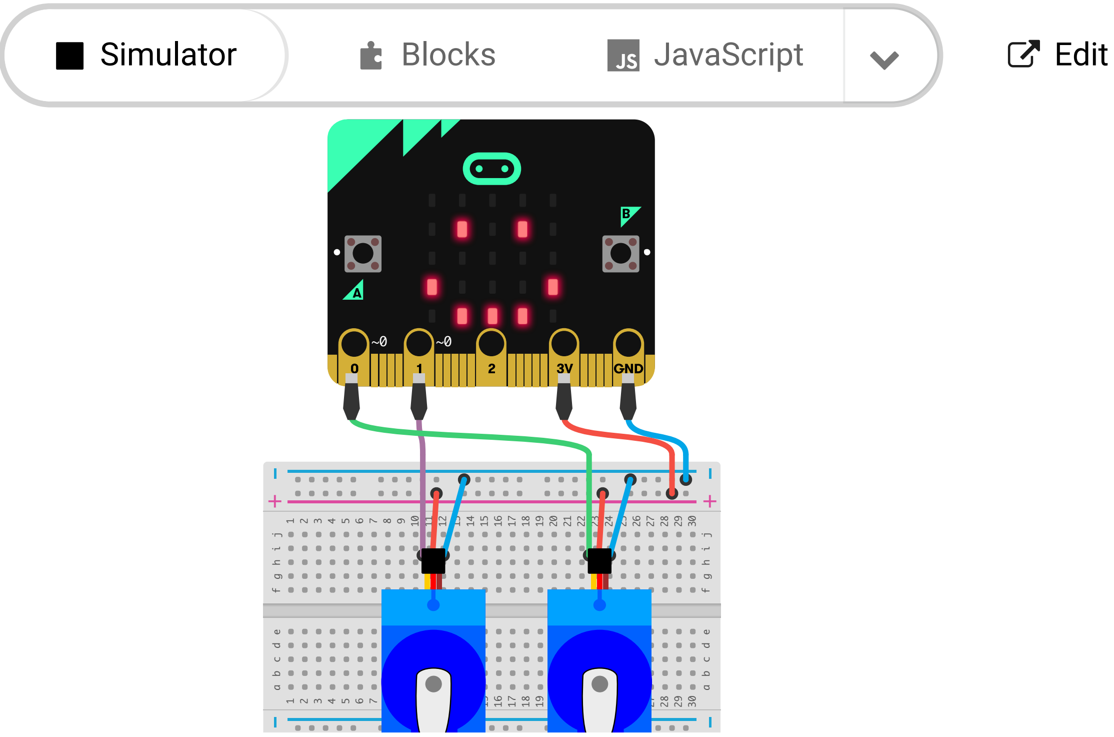


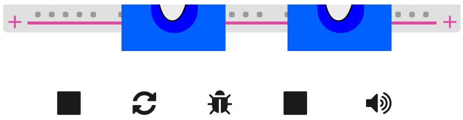
```
on start
    radio set group 7

on shake
    radio send number 4

forever
    if butto
        radio sen
    else if b
        radio sen
    else if b
```

Ring:bit car Coding

Program Link：https://makecode.microbit.org/_4aWeti12K4HR

If you don’t want to type these code by yourself, you can directly download the whole program from the link below:





14.7. Result

- Press the button A, the car turns left; press the button B, the car turns right; press the button A and B together, the car reverses.

14.8. Think

- How can you control running speed of the car in a remote distance?

14.9. Questions

14.10. More Information

15. case 07 Turns over Detection

15.1. Our Goal

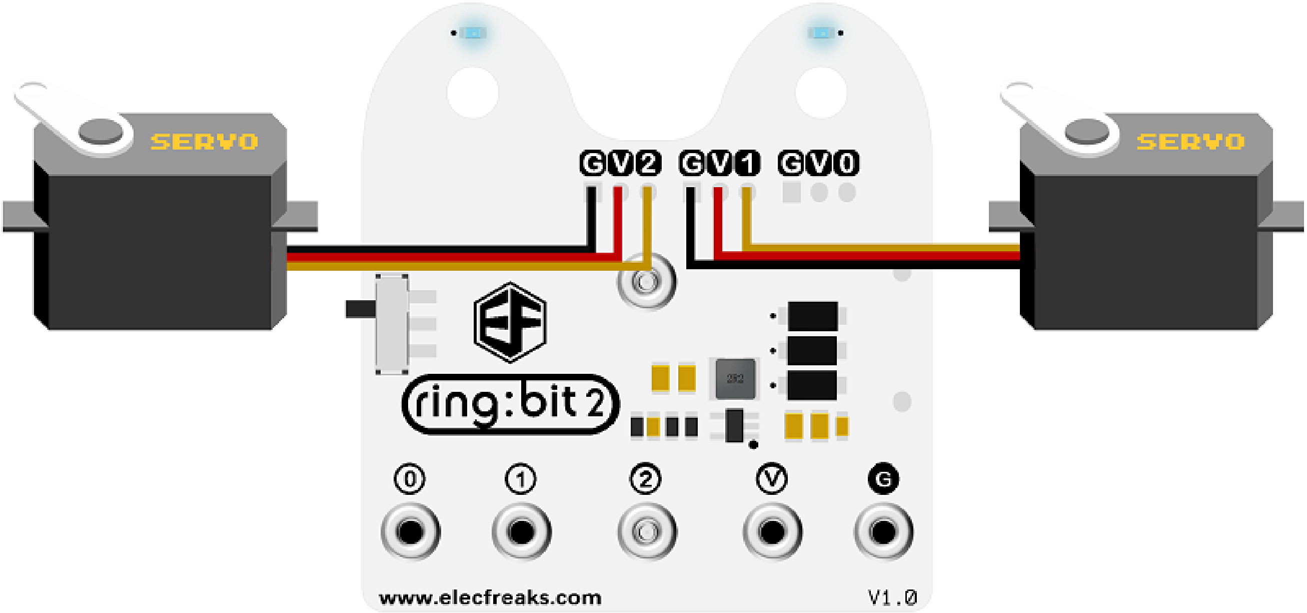
- Make the car to stop when it turns over.

15.2. Requiered Materials

- 1 x Ring:bit Car

15.3. Hardware Connect

- Connect the left wheel servo to P1 of the Ring:bit expansion board and the right wheel servo to P2.



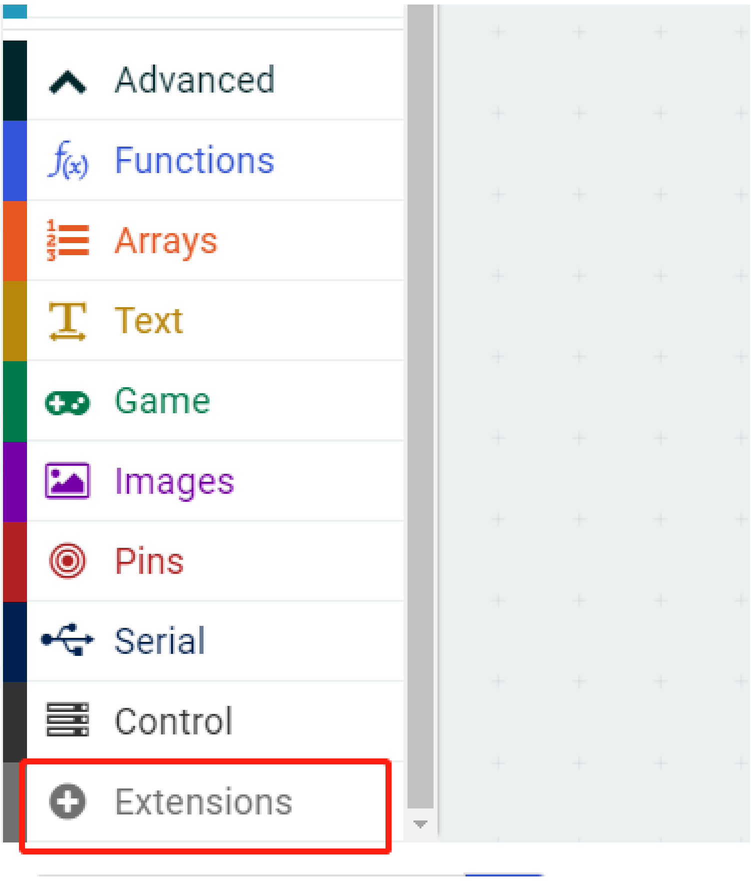
15.4. Software

makecode

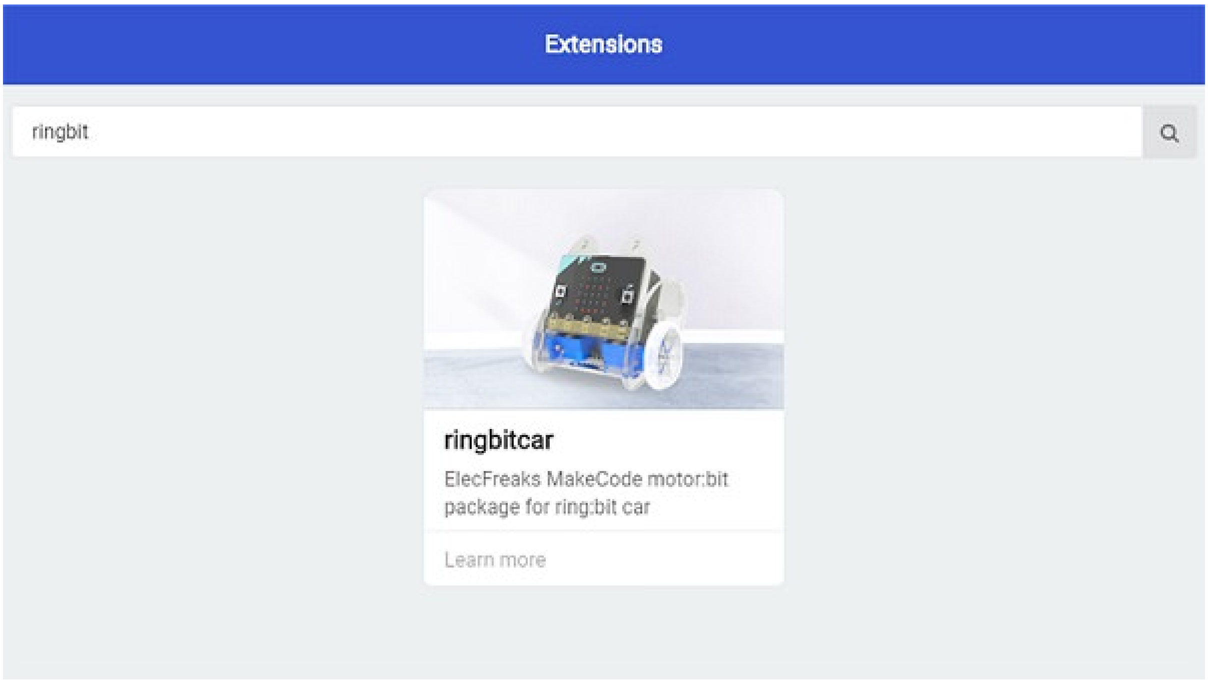
15.5. Coding

Step 1

- Click on “Advanced” in the MakeCode Drawer to see more code sections.



- Search for “ringbitcar” and click on the ring:bit car package to add it to your project. (As below picture)



ringbit

ringbitcar

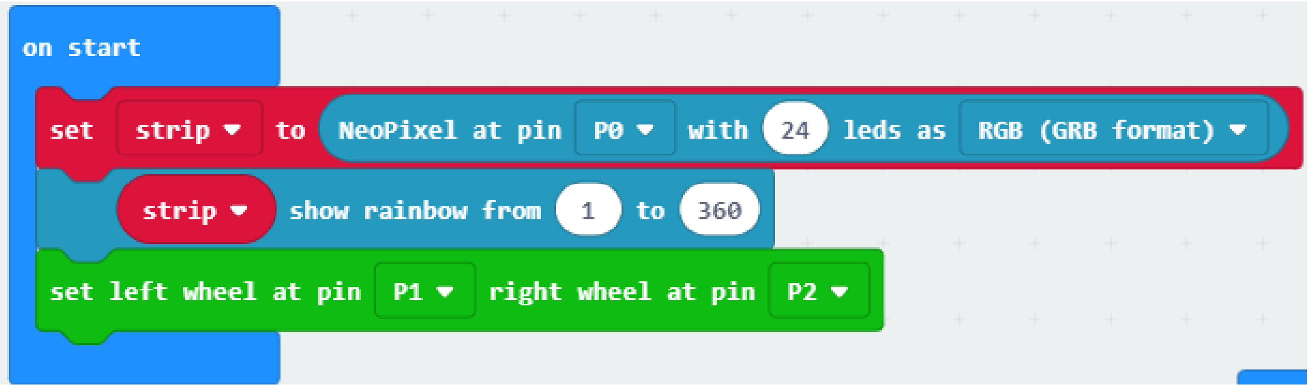
ElecFreaks MakeCode motor:bit package for ring:bit car

Note： If you get a warning telling you some packages will be removed because of incompatibility issues, either follow the prompts or create a new project in the Project file menu.

Step 2

- Snap the `on start` block from the Basic. Set Neopixel with 24 leds in RGB format for colorful light and show rainbow from 1 to 360. Then, initialize the P1 port to left wheel and the P2 port to right wheel.

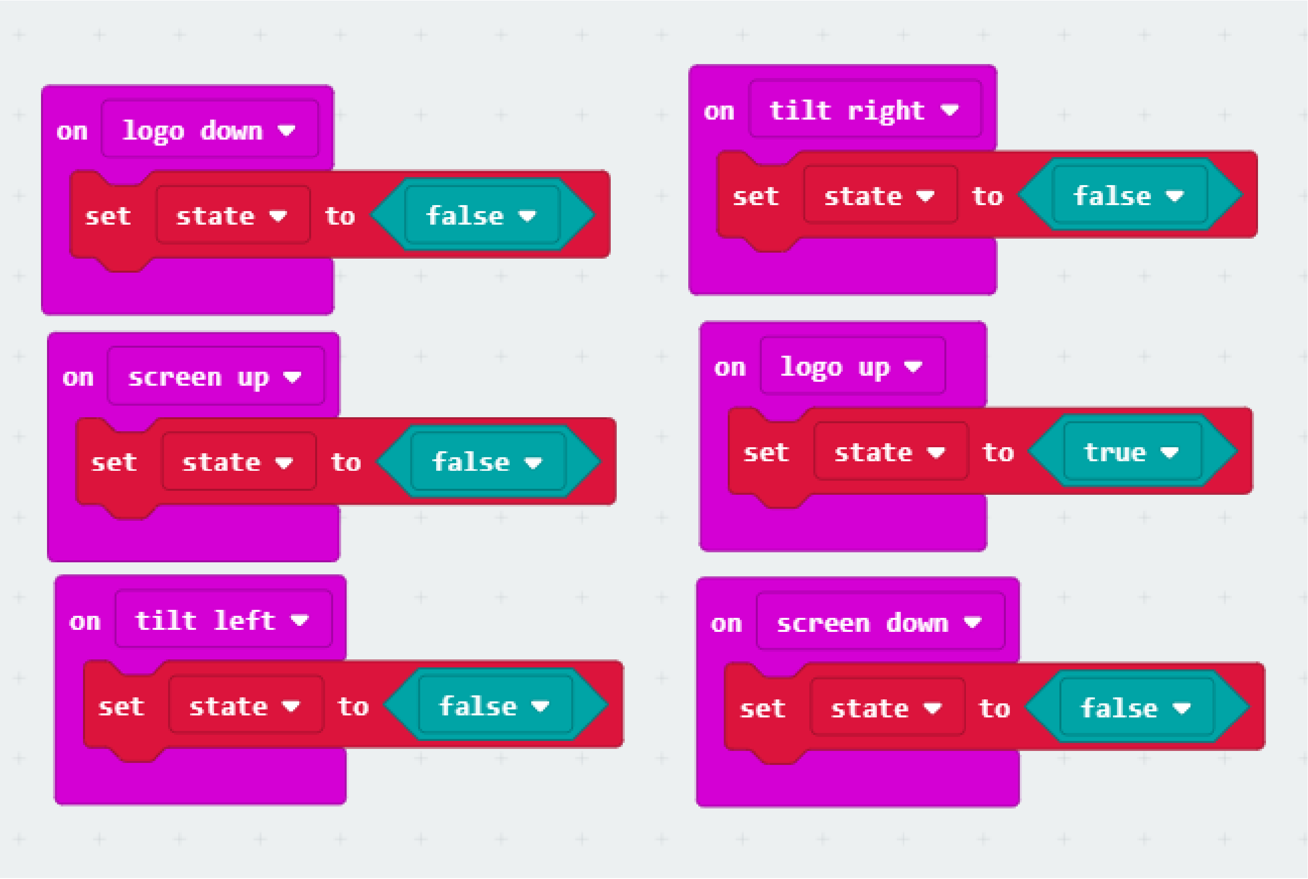
!



```
on start
    set strip ▾ to NeoPixel at pin P0 ▾ with 24 leds as RGB (GRB format) ▾
    strip ▾ show rainbow from 1 to 360
    set left wheel at pin P1 ▾ right wheel at pin P2 ▾
```

Step 3

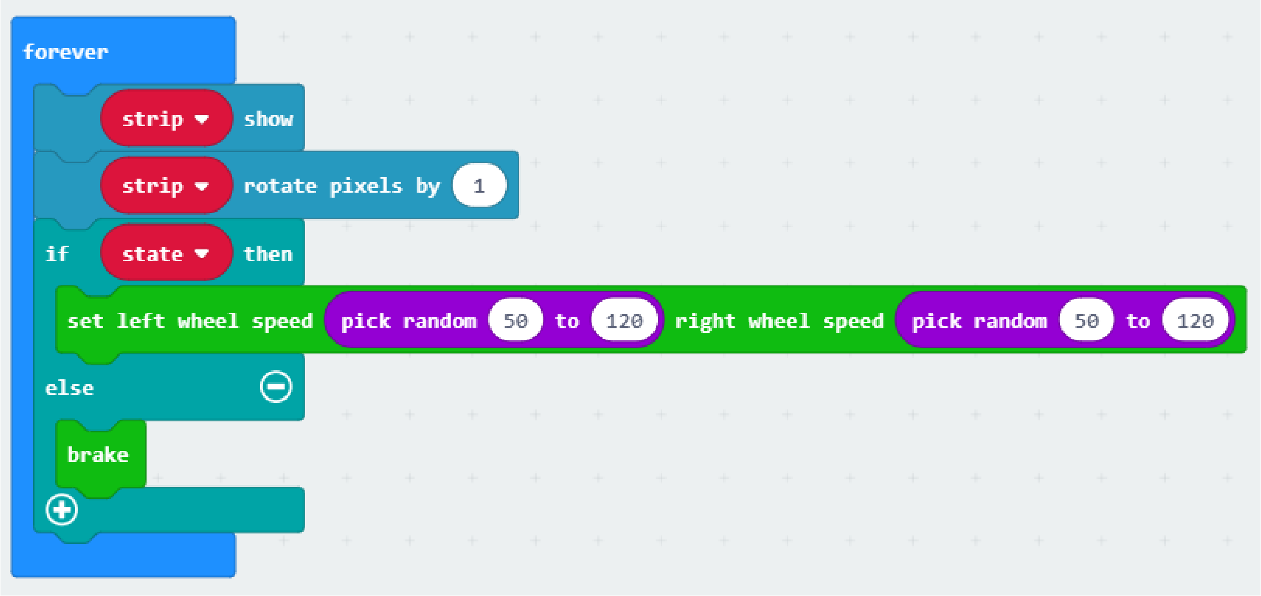
- Set a variable to state for controlling car’s running. Then set state variable unber the logo up block to true and the other to false.



```
on logo down ▾
    set state ▾ to false ▾

on tilt right ▾
    set state ▾ to false ▾

on screen up ▾
    set state ▾ to false ▾

on logo up ▾
    set state ▾ to true ▾

on tilt left ▾
    set state ▾ to false ▾

on screen down ▾
    set state ▾ to false ▾
```

Step 4

- Set the rainblw led to colorful lights under the `forever` loop in gradual change. Use the variable state to control car’s running.

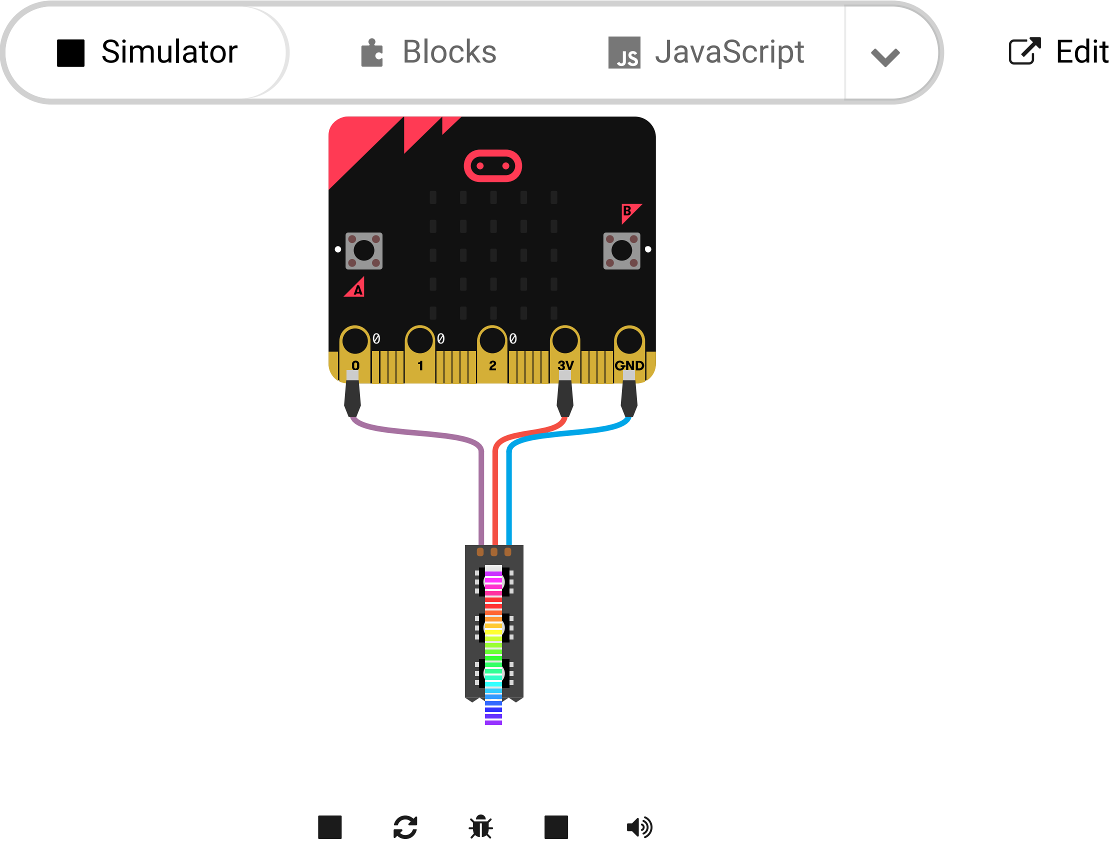


```
forever
    strip ▾ show
    strip ▾ rotate pixels by 1
    if state ▾ then
        set left wheel speed pick random 50 to 120 right wheel speed pick random 50 to 120
    else ⊖
        brake
    ⊕
```

Program

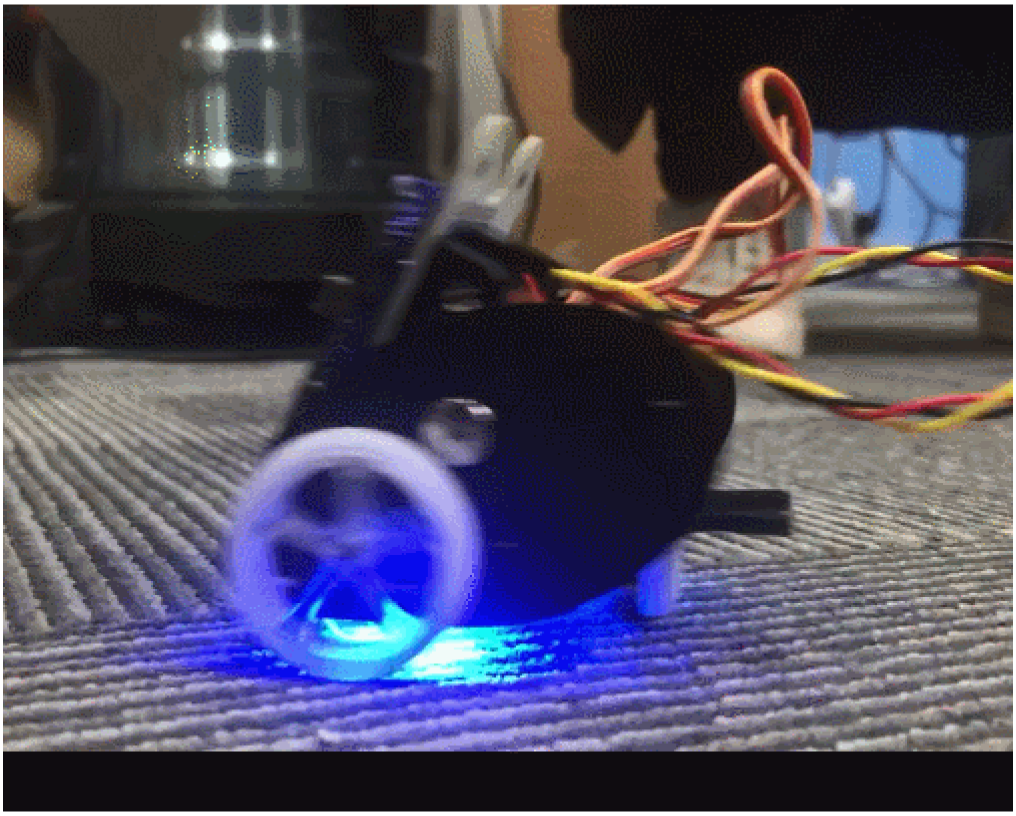
Program Link： https://makecode.microbit.org/_asoLxTRz4Dg2

If you don’t want to type these code by yourself, you can directly download the whole program from the link below:



15.6. Result

- The car runs normally， when it turns over, it stops.



15.7. Think

- How can you design more funny project using other kits?

15.8. Questions

15.9. More Information

16. case 08 Light Follow

16.1. Our Goal

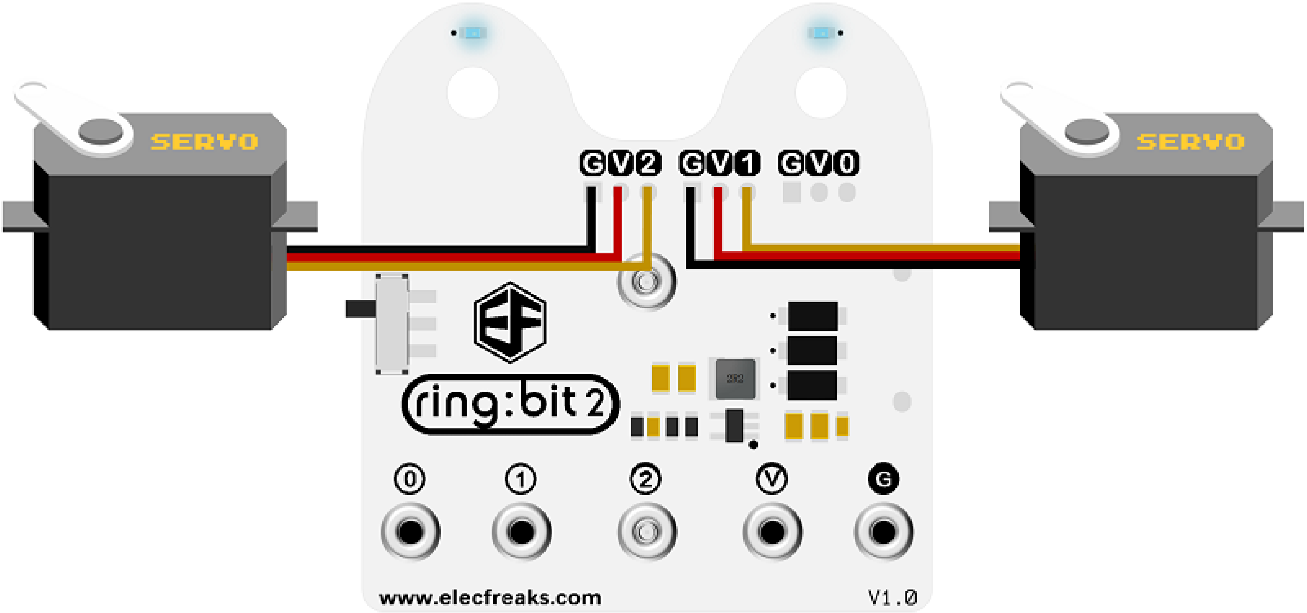
- Make the ring:bit car to light follow.

16.2. Requiered Materials

- 1 x Ring:bit Car

16.3. Hardware Connect

- Connect the left wheel servo to P1 of the Ring:bit expansion board and the right wheel servo to P2.



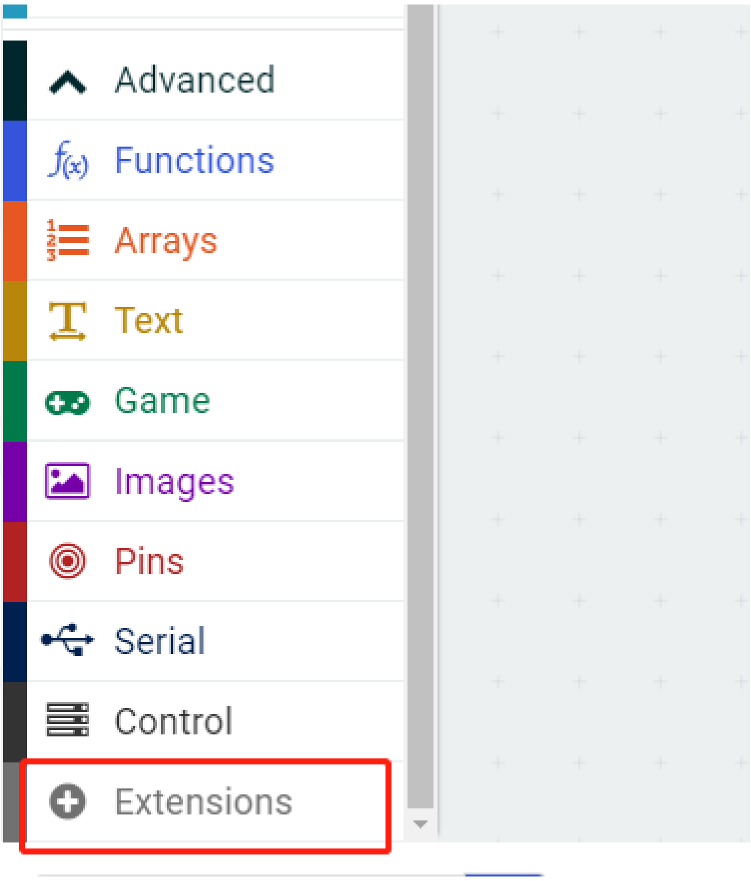
16.4. Software

makecode

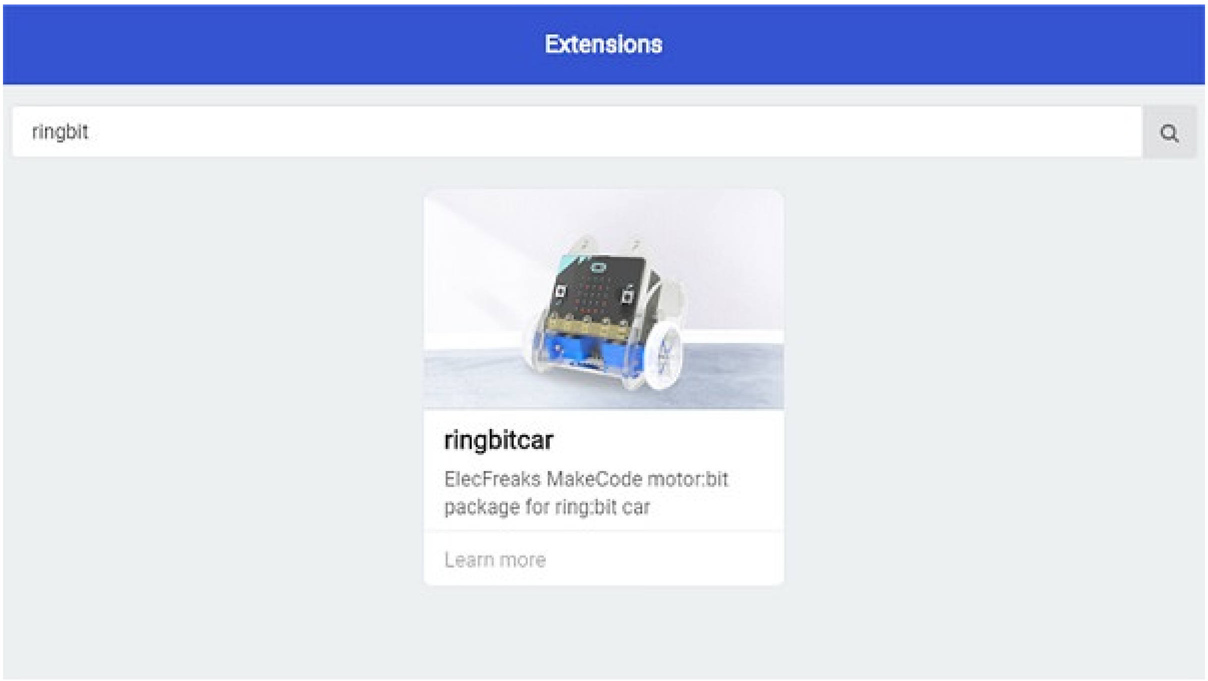
16.5. Coding

Step 1

- Click on “Advanced” in the MakeCode Drawer to see more code sections.



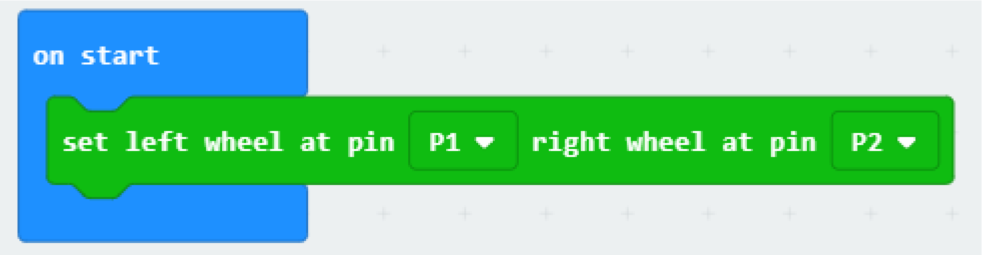
- Search for “ringbitcar” and click on the ring:bit car package to add it to your project. (As below picture)



Note： If you get a warning telling you some packages will be removed because of incompatibility issues, either follow the prompts or create a new project in the Project file menu.

Step 2

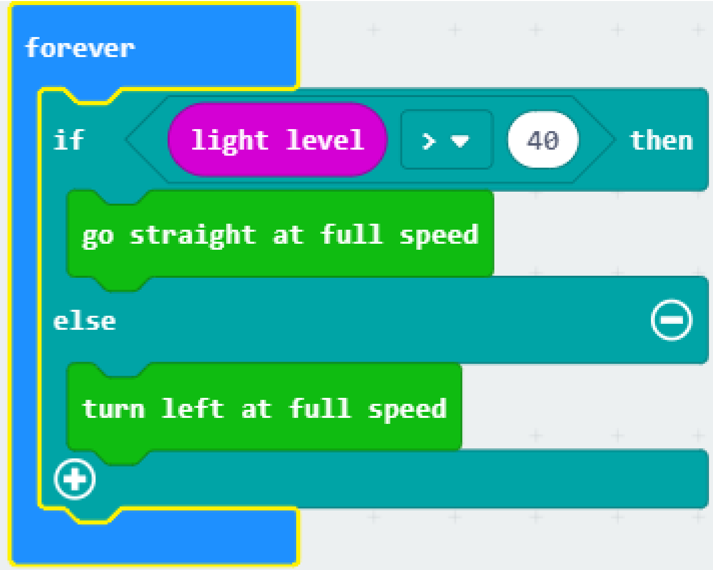
- Snap the `on start` block from the Basic and initialize the P1 port to left wheel and the P2 port to right wheel.



```
on start
    set left wheel at pin P1 ▾ right wheel at pin P2 ▾
```

Step 3

- Snap the `light level` from the Input. The car will move to the light source when the light level is greater than the defined value.
- The car will spin around to find the light source when the light level is less than the defined value.

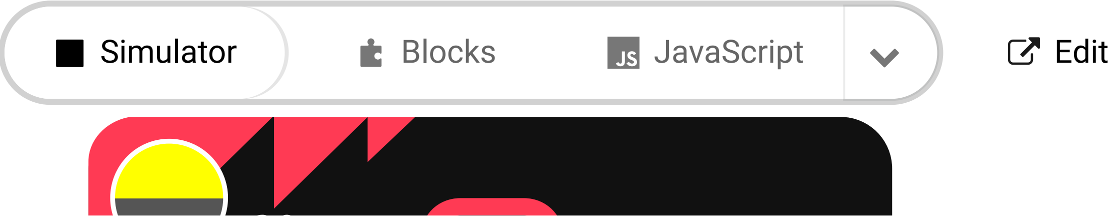


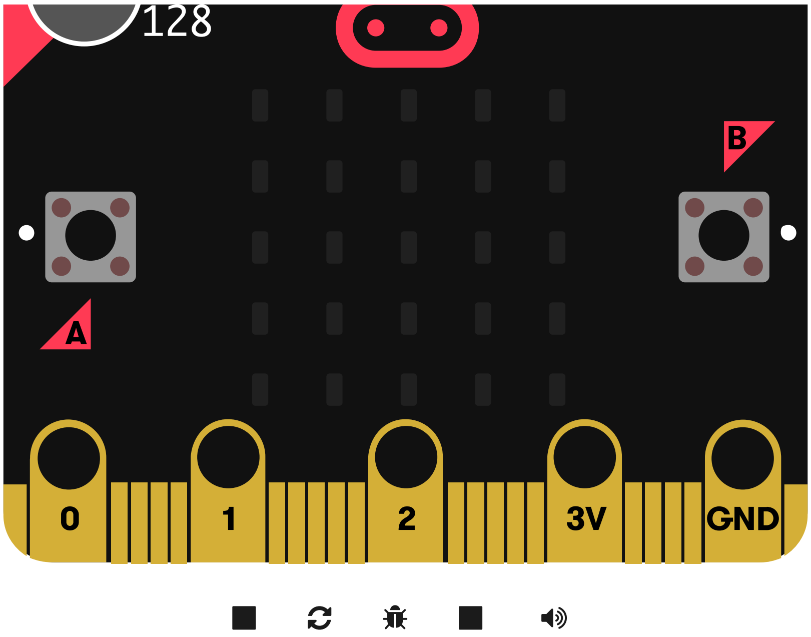
```
forever
    if light level > ▾ 40 then
        go straight at full speed
    else
        turn left at full speed
```

Program

Program Link： https://makecode.microbit.org/_H7c7DPWTPYKY

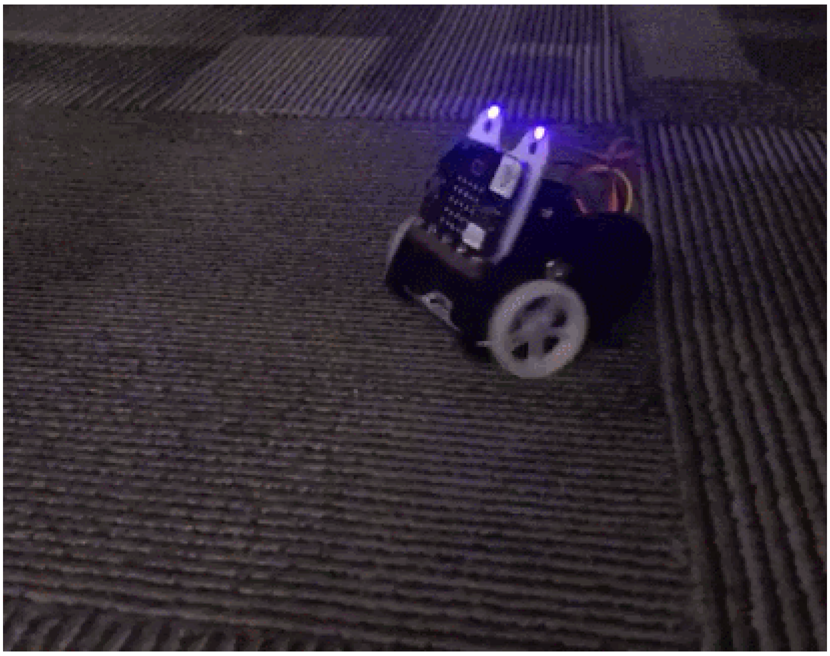
If you don’t want to type these code by yourself, you can directly download the whole program from the link below:





16.6. Result

- The car spins around and will move to the light source when it is detecting the light.



16.7. Think

- How can you make a smart car to detect the fire?

16.8. Questions

16.9. More Information

17. case 09 Remote Control by an accelerometer

17.1. Our Goal

- Use another an accelerometer of the micro:bit to remote control the car.

17.2. Requiered Materials

- 1 x Ring:bit Car
- 2 x micro:bit

17.3. Background

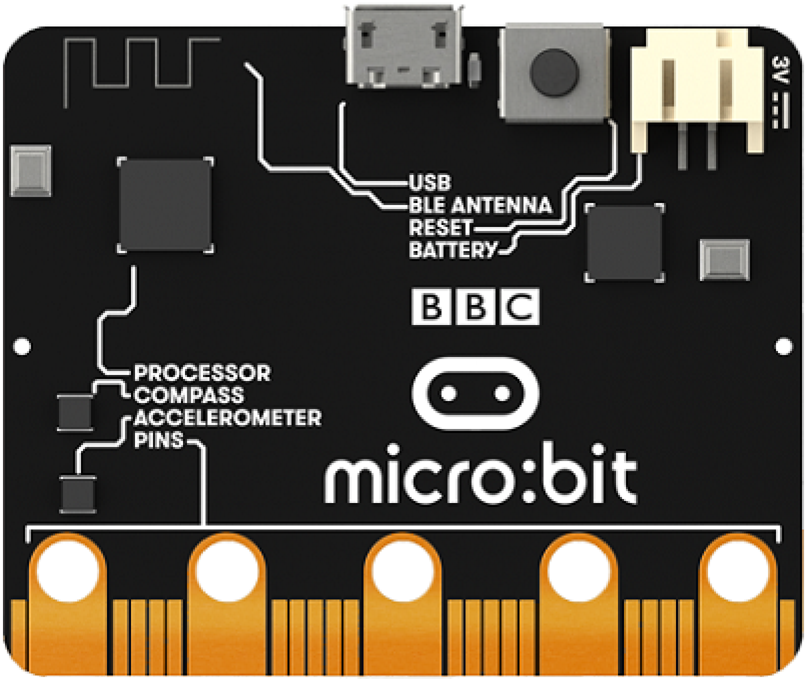
What is wireless

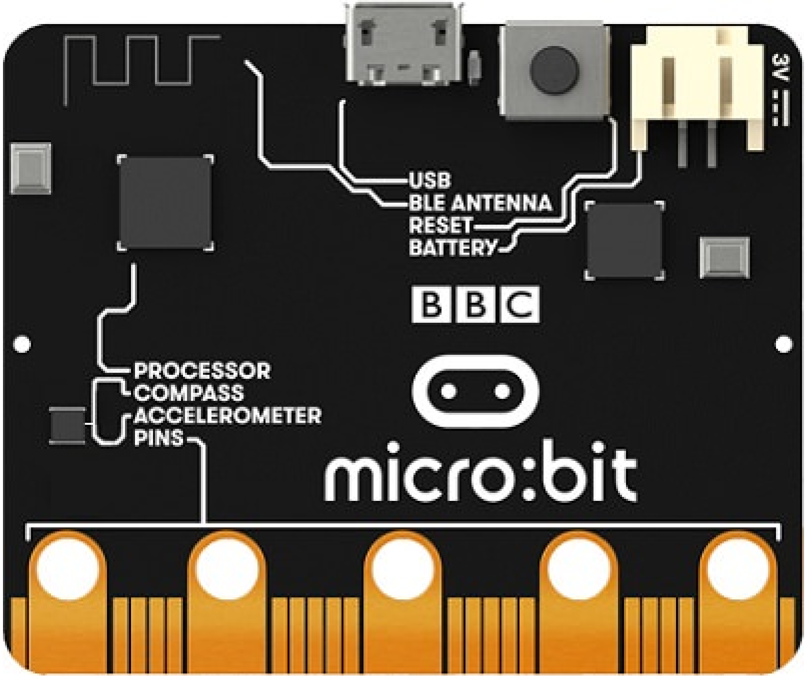
- Wireless technology sends message by radio wave because changes of electricity will generate waves. We can use this to load the message to the wave. Variation of electromagnetic field will generate the electricity when the electromagnetic is arrving the receiver. Then, to extract the message from wave by demodulation to send the message.

What is an accelerometer

- The accelerometer is a kind of sensor to measure the accelerated speed. It has included mass block, damper, elastic component, sensitive element and optimal tuning circuit. The sensor is measuring the inertia force and getting the acceleration magnitude by newton's second law during its accelerating. According to the difference of sensitive element, the common used accelerometer has included capacitance, strain, piezoresistive and piezoelectric.

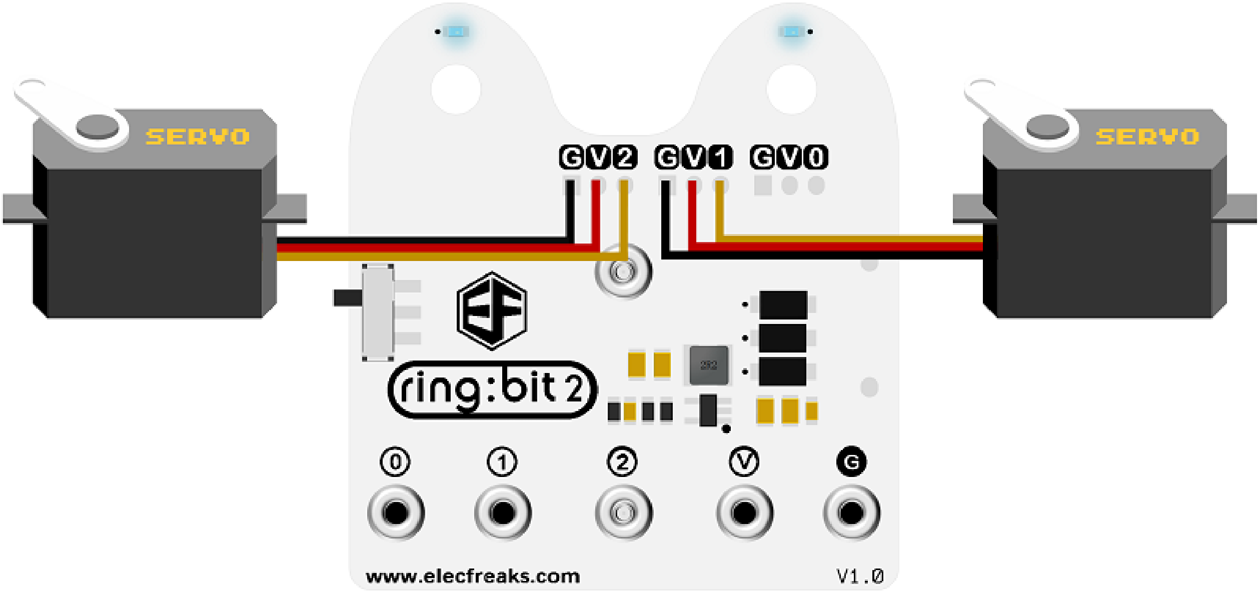
The new version accelerometer chip of micro:bit is different from the old version. The new version has combined the electronic compass and the accelerometer. No change for usage.





17.4. Hardware Connect

- Connect the left wheel servo to P1 of the Ring:bit expansion board and the right wheel servo to P2.



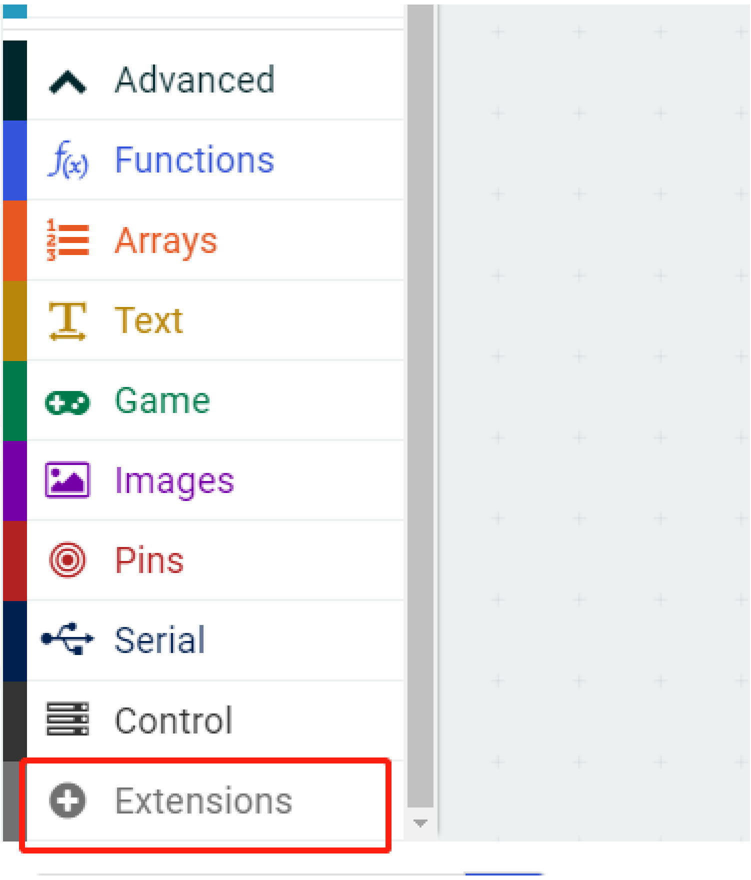
17.5. Software

makecodeOnline block codinghttps://makecode.microbit.org/#

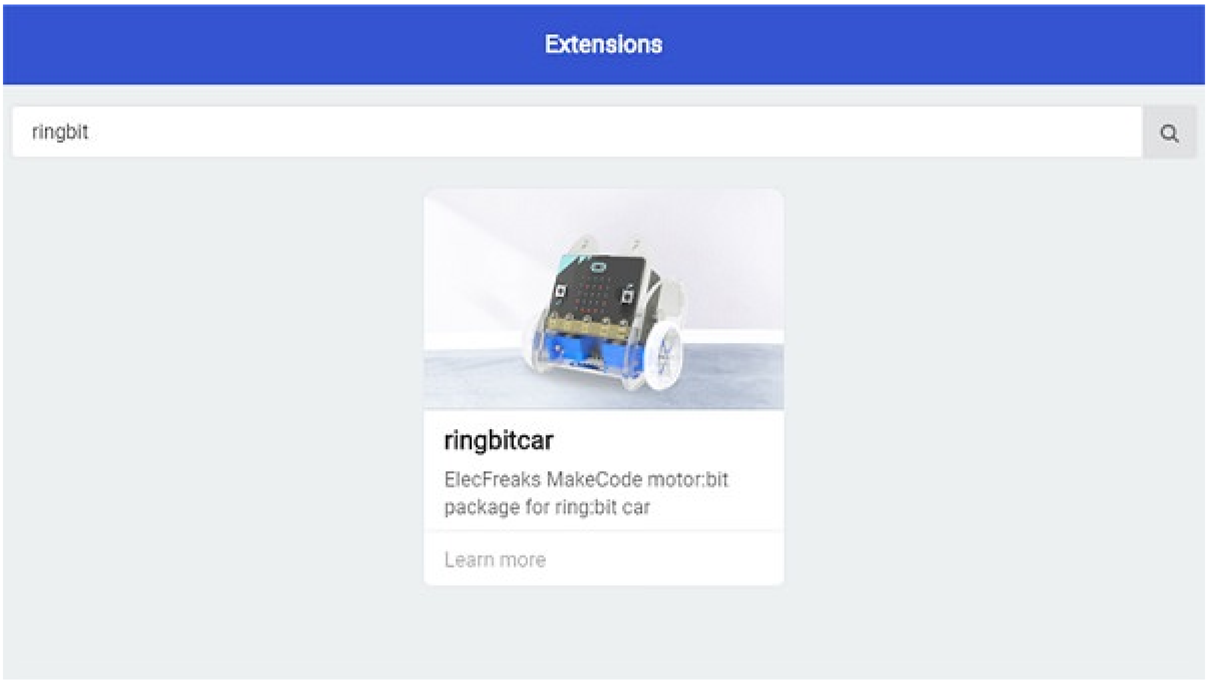
17.6. Coding

Step 1

- Click on “Advanced” in the MakeCode Drawer to see more code sections.



- Search for “ringbitcar” and click on the ring:bit car package to add it to your project. (As below picture)



Note：If you get a warning telling you some packages will be removed because of incompatibility issues, either follow the prompts or create a new project in the Project file menu.

Step 2

- It is difficult to calculate the control of the gyroscope, please refer more details by the link：Accelerometry

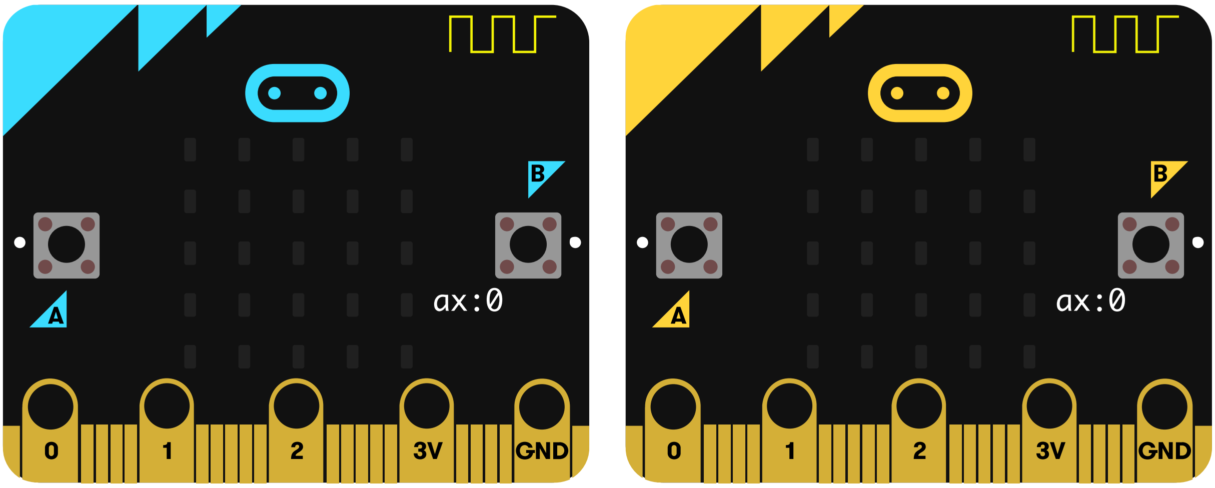
- If you don’t want to type these code by yourself, you can directly download the whole program from the link below:

Remote Control Coding

Program Link：https://makecode.microbit.org/_AT4PoHKdVi6L

If you don’t want to type these code by yourself, you can directly download the whole program from the link below:

Simulator Blocks JavaScript Edit

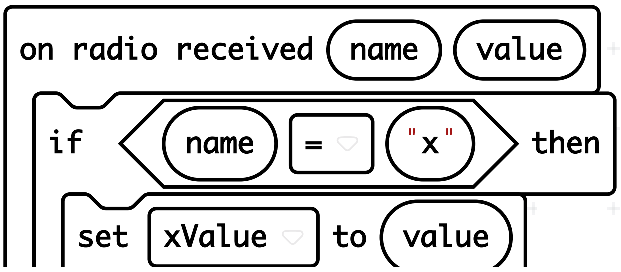


Ring:bit car Coding

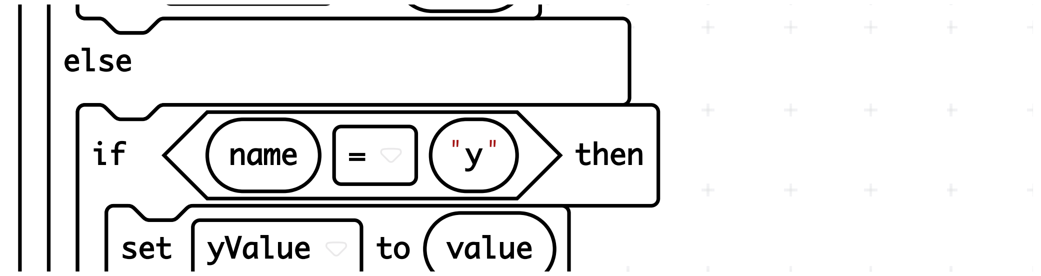
Program Link：https://makecode.microbit.org/_e5t6XPHoTiHy

If you don’t want to type these code by yourself, you can directly download the whole program from the link below:

Simulator Blocks JavaScript Edit



```
on radio received name value
  if name = "x" then
    set xValue to value
```



```
else
  if name = "y" then
    set yValue to value
```

Microsoft MakeCode | Terms of Use | Privacy | Download

17.7. Result

- The Ring:bit car will move as direction of the gyroscope and its speed will as the angle of inclination of the gyroscope.

17.8. Think

-

17.9. Questions

17.10. More Information

18. Case 10 Smart Crashproof Car

18.1. Purpose

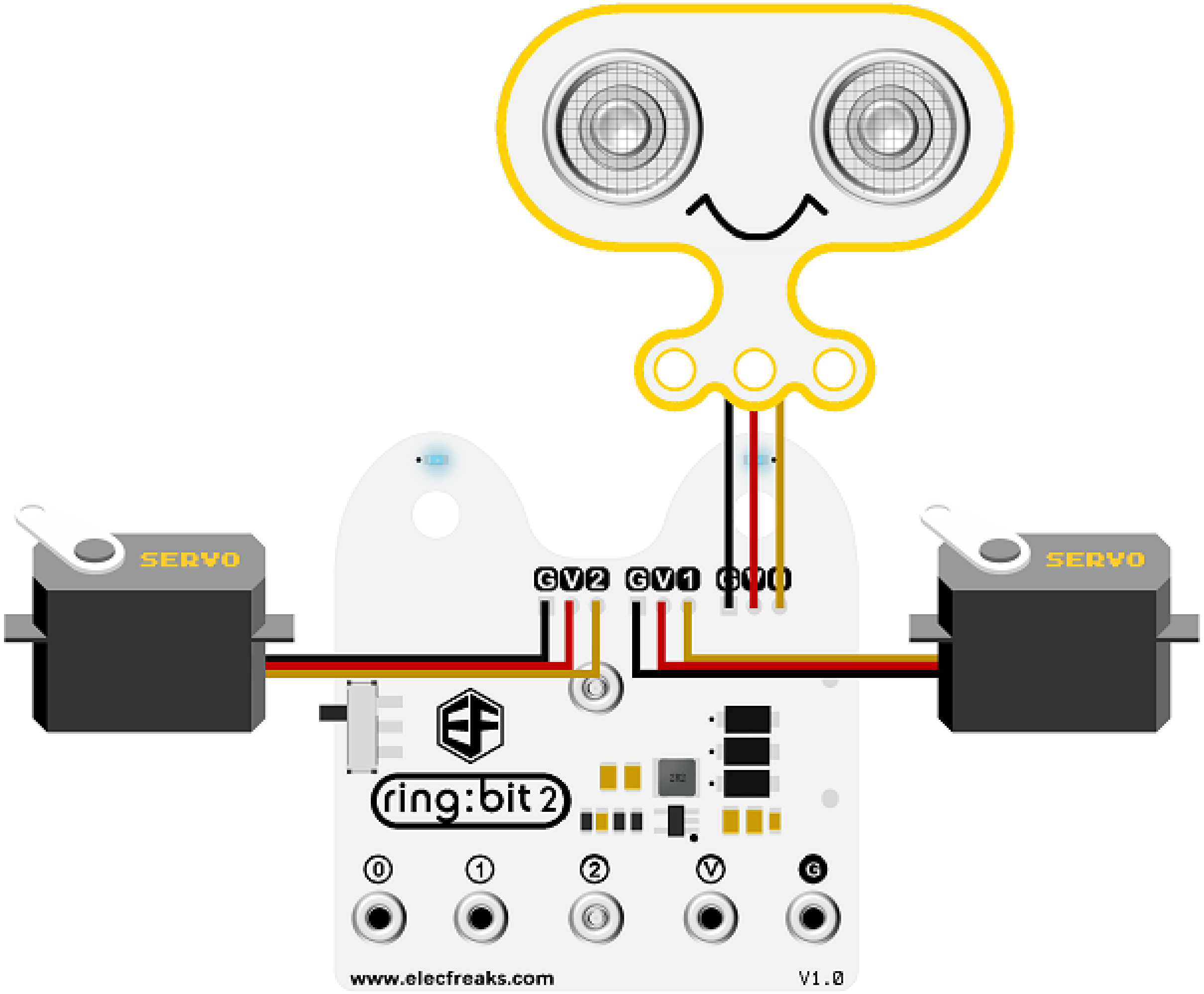
- Make a smart crashproof car with a Sonar:bit.

18.2. Materials

- 1 x Ring:bit Car kit
- 1 x micro:bit
- 1 x Sonar:bit

18.3. Hardware Connection

- Connect the left servo to P1, right servo to P2 and Sonar:bit to P0 of the Ring:bit.



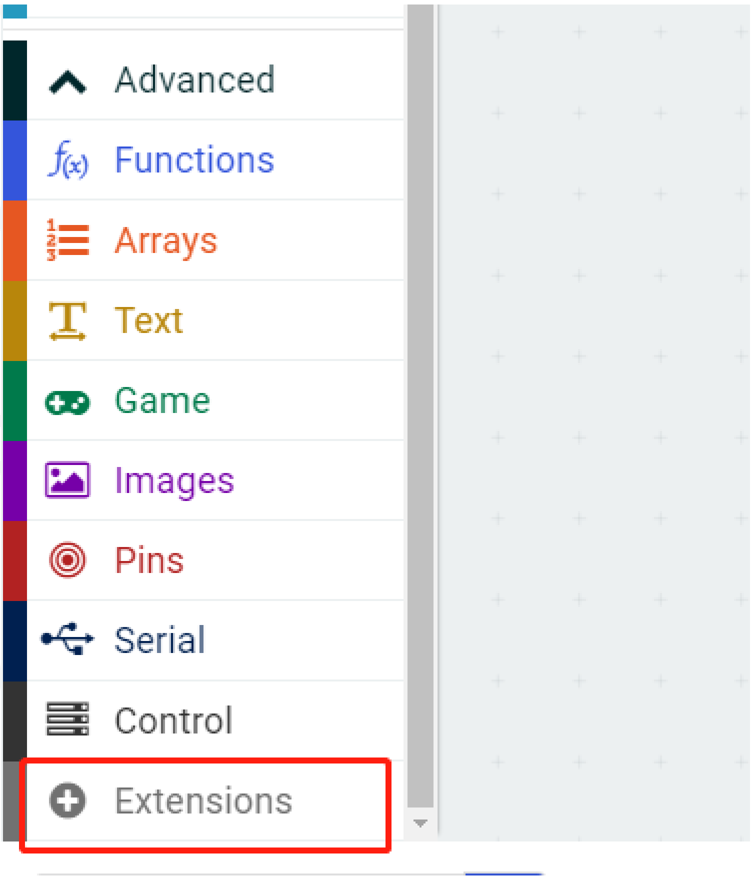
18.4. Software Programming

Microsoft makecode online coding https://makecode.microbit.org/#

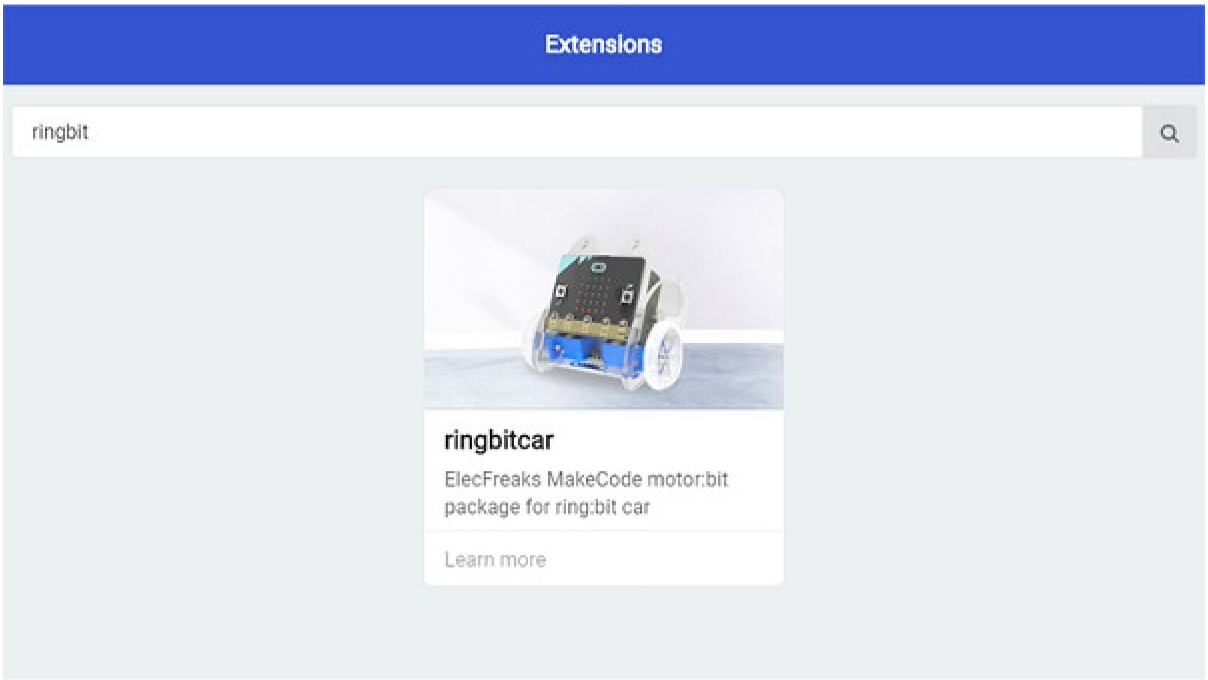
18.5. Software

Step 1

- Click “Advanced” in the drawer of MakeCode to see more choices.



- We need to add a package for programming the kit. Click “Extensions” in the drawer and search “ringbit” in the dialogue box to download it.

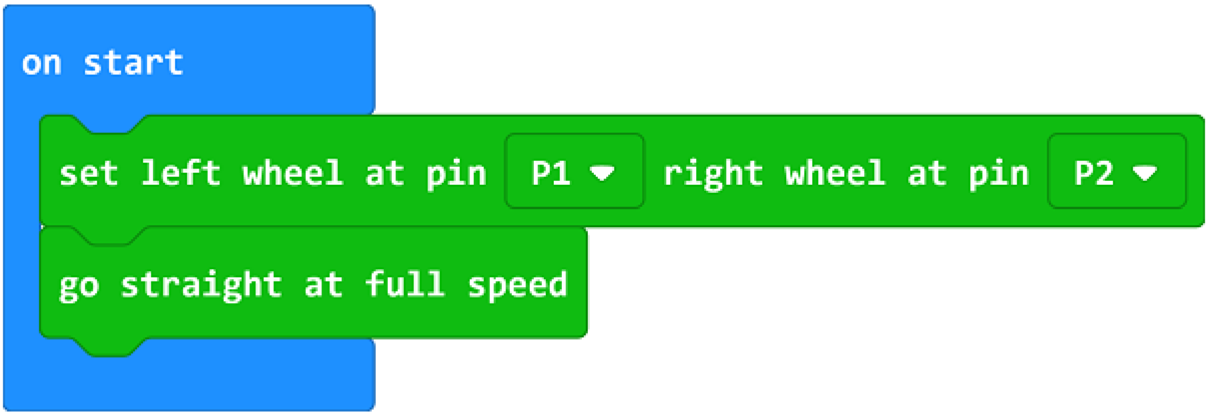


- We also need to add a package for programming the Sonar:bit. Click “Extensions” in the drawer and search “`https://github.com/elecfreaks/pxt-sonarbit`” in the dialogue box to download it.

Note: if you are informed that it will be deleted due to incompatibility of the codebase, you can go on with the tips or create a new project in the menu.

Step 2

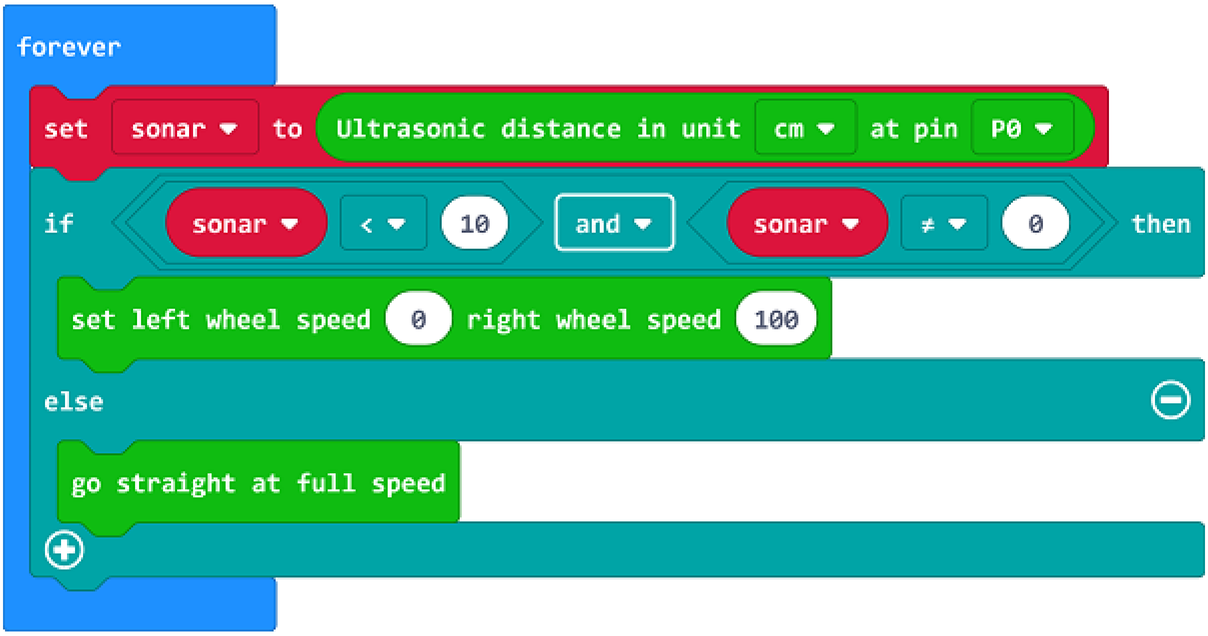
- Drag the pins selection bricks for servos in the `On start` brick, the pins numbers are set according to the actual connection port.
- Move forward at full speed.



```
on start
    set left wheel at pin P1 ▾ right wheel at pin P2 ▾
    go straight at full speed
```

Step 3

- Set variate `sonar` in the `forever` brick and read the detected value from the Sonar:bit to it.
- Drag the `if` brick and judge if the value given by `sonar` is below 10 and not equal to 0.
- If yes, set the speed of right wheel to 100 and the left to 0 to turn left with 500ms for obstacle avoidance.
- If not, move forward at full speed.

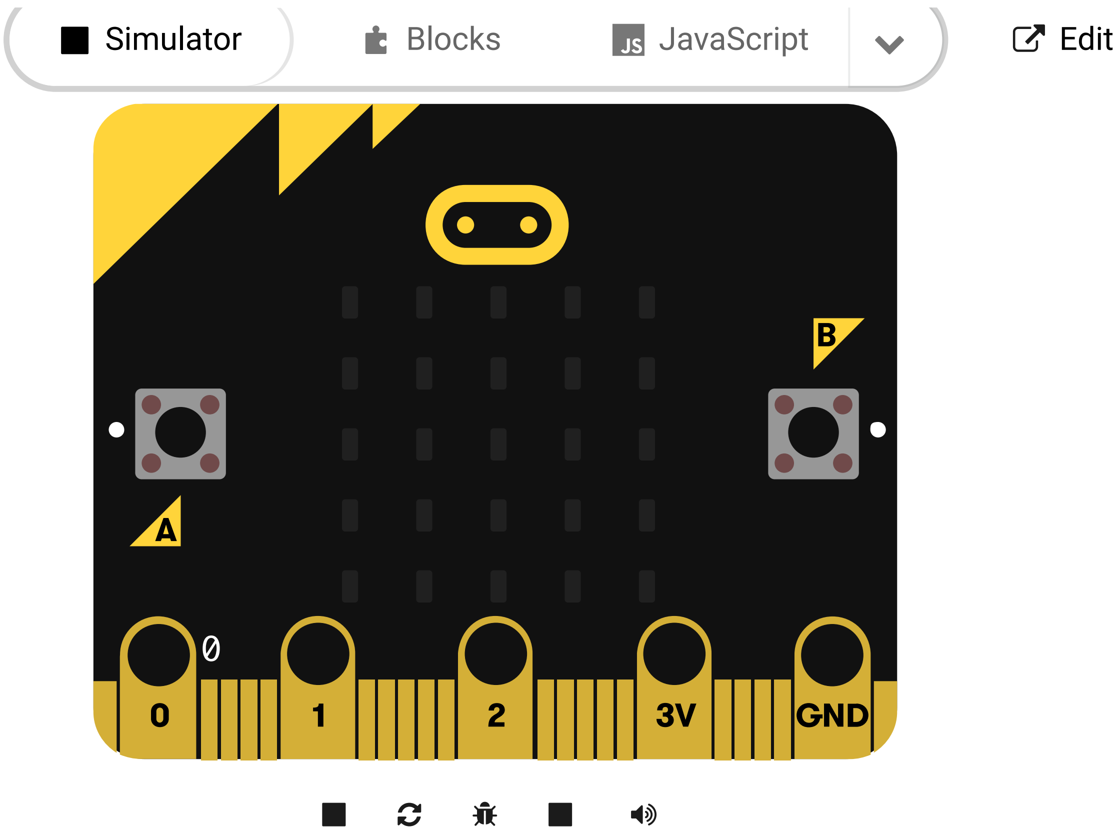


```
forever
    set sonar ▾ to Ultrasonic distance in unit cm ▾ at pin P0 ▾
    if sonar ▾ < ▾ 10 and ▾ sonar ▾ ≠ ▾ 0 then
        set left wheel speed 0 right wheel speed 100
    else
        go straight at full speed
```

Reference

Links: https://makecode.microbit.org/_RTwFcMeA1MMY

You can also download it below:



--- ## Result --- - The Ring:bit car turns left automatically when detecting any obstacle 10cm in front of it.

18.6. Exploration

- Question: Why we need to judge if the value is not 0 ?
- Answer: The detection value is also 0 if beyond the detection scope of Sonar:bit.

18.7. FAQ

19. Case 11: Control the Ring:bit Car with the Joystick:bit

19.1. Purpose

- Control the ring:bit car via the Joystick:bit.

19.2. Materials

Ring:bit car V2 × 1

Joystick:bit × 1

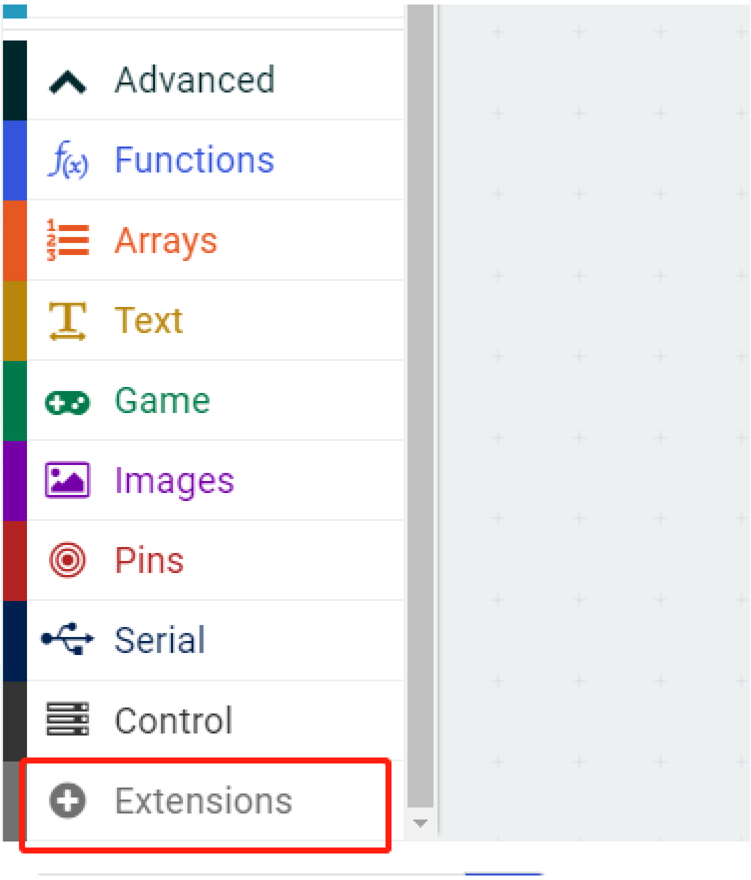
19.3. Software

MicroSoft makecode

19.4. Programming

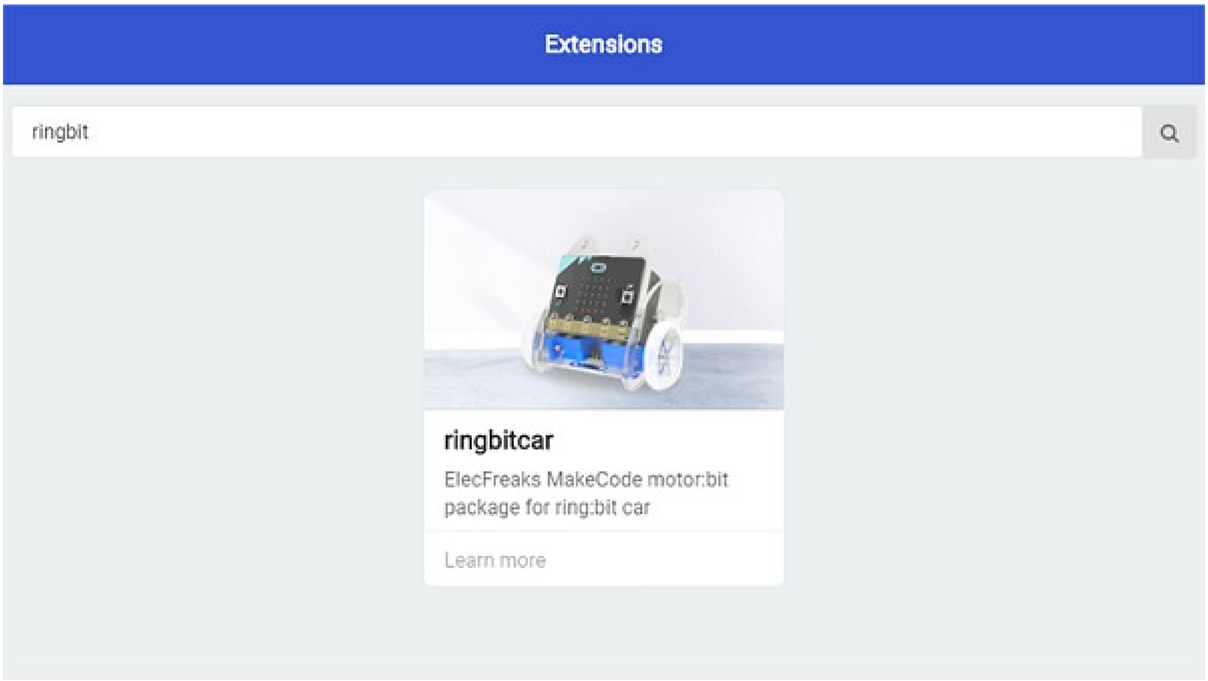
Step 1

Click “Advanced” in the MakeCode drawer to see more choices.



Step 2 Coding

We need to add a package for programming. Click “Extensions” in the bottom of the drawer and search with “Ringbit” in the dialogue box to download it.



Extensions

ringbit

ringbitcar

ElecFreaks MakeCode motor:bit package for ring:bit car

Learn more

*Note: If you met a tip indicating it might be deleted due to incompatibility, you may continue as it indicates or create a new project in the menu.

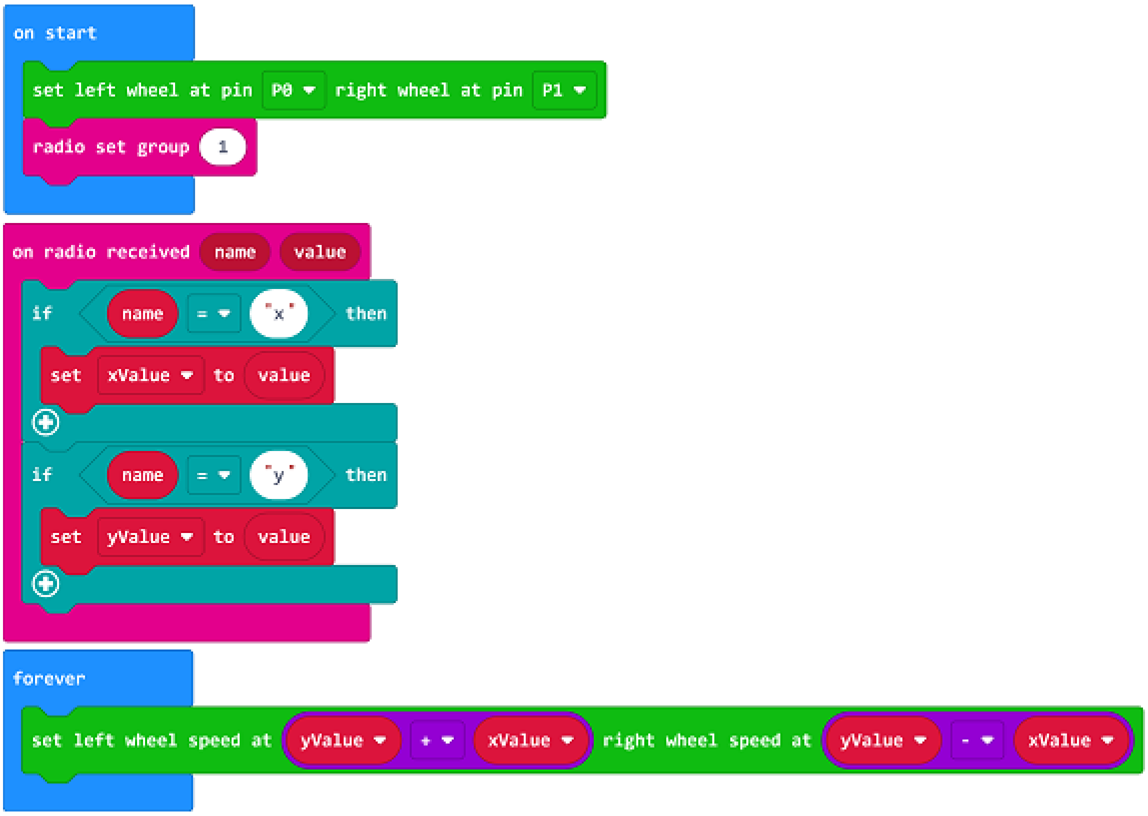
In the on start brick, set the radio group as 1, please make sure it is in the same radio group with the remote controlling end so they can match.

Drag two “if…else…” bricks into the on radio data received brick, judge if the recevied value “name” is X or Y.

If “name” is x, set it as the data of xValue.

If “name” is y, set it as the data of yValue.

In forever brick, set the speed of the left wheel as yValue+xValue and the speed of the right wheel as yValue-xValue.

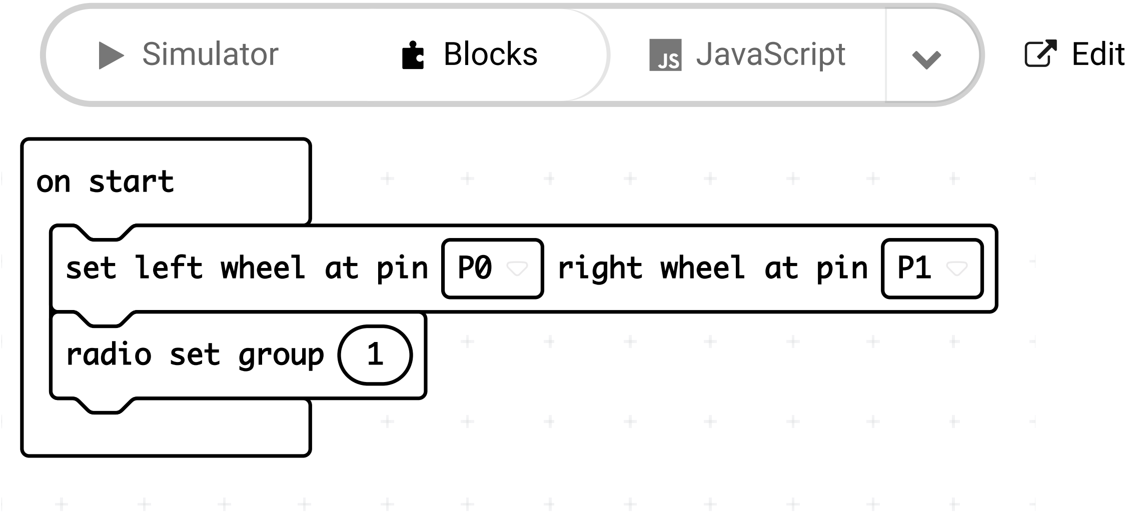


```
on start
    set left wheel at pin P0 ▾ right wheel at pin P1 ▾
    radio set group 1

on radio received name value
    if name = ▾ "x" then
        set xValue ▾ to value
    ⊕
    if name = ▾ "y" then
        set yValue ▾ to value
    ⊕

forever
    set left wheel speed at yValue ▾ + ▾ xValue ▾ right wheel speed at yValue ▾ - ▾ xValue ▾
```

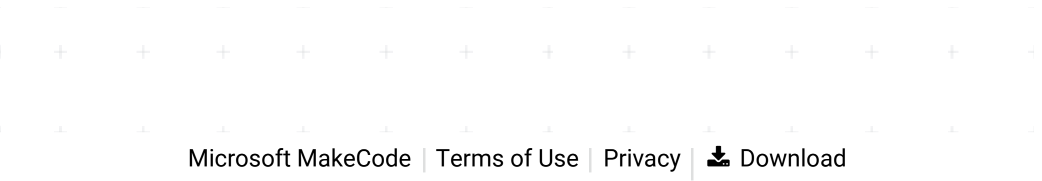
Link

Link: https://makecode.microbit.org/_1vAgLo3Ky5Rm

You may download it directly below:

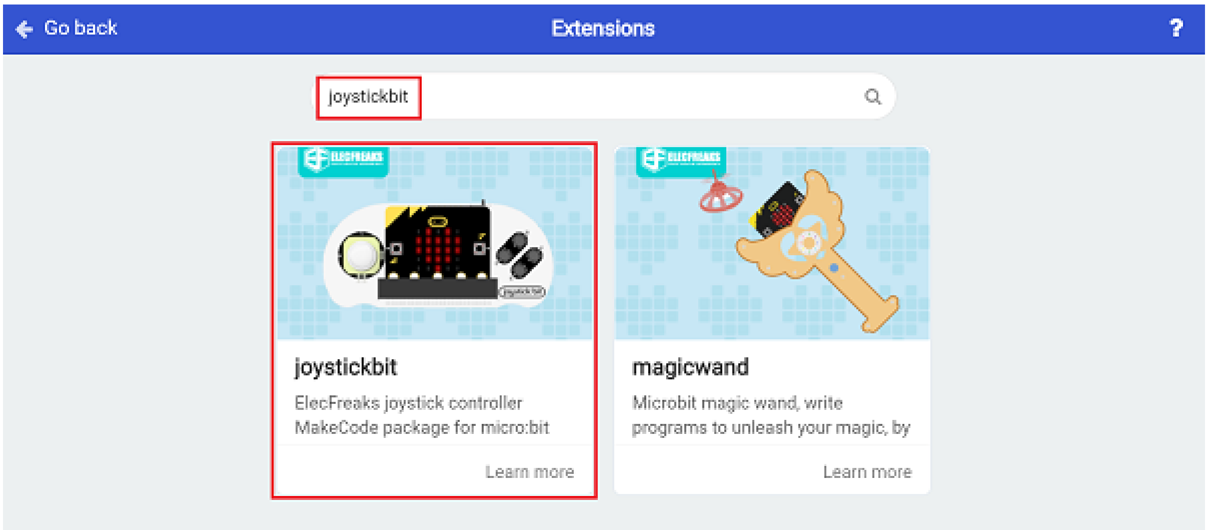


```
on start
    set left wheel at pin P0 right wheel at pin P1
    radio set group 1
```



Step 3 Coding the Joystick:bit

We need to add a package for programming. Click “Extensions” in the bottom of the drawer and search with “joystickbit” in the dialogue box to download it.


Go back
Extensions
?
joystickbit
joystickbit
ElecFreaks joystick controller MakeCode package for micro:bit
Learn more
magicwand
Microbit magic wand, write programs to unleash your magic, by
Learn more

Note: If you met a tip indicating it might be deleted due to incompatibility, you may continue as it indicates or create a new project in the menu.

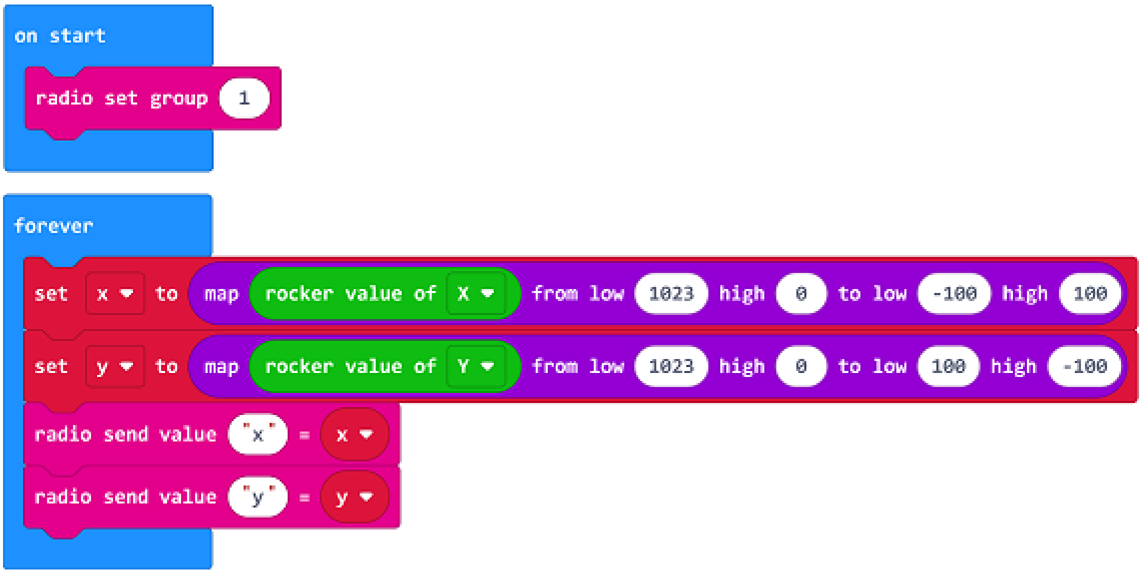
In the on start brick, set the radio group as 1;

The value of X and Y ranges from 0~1023, the theoretical value is 512 while the Joystick is placed in the middle position, thus we need to map that range to -100~100;

In forever brick, set the value of x is among -100~100 mapping from x-axis;

In forever brick, set the value of y is among -100~100 mapping from y-axis;

Send the value of x and y via radio.

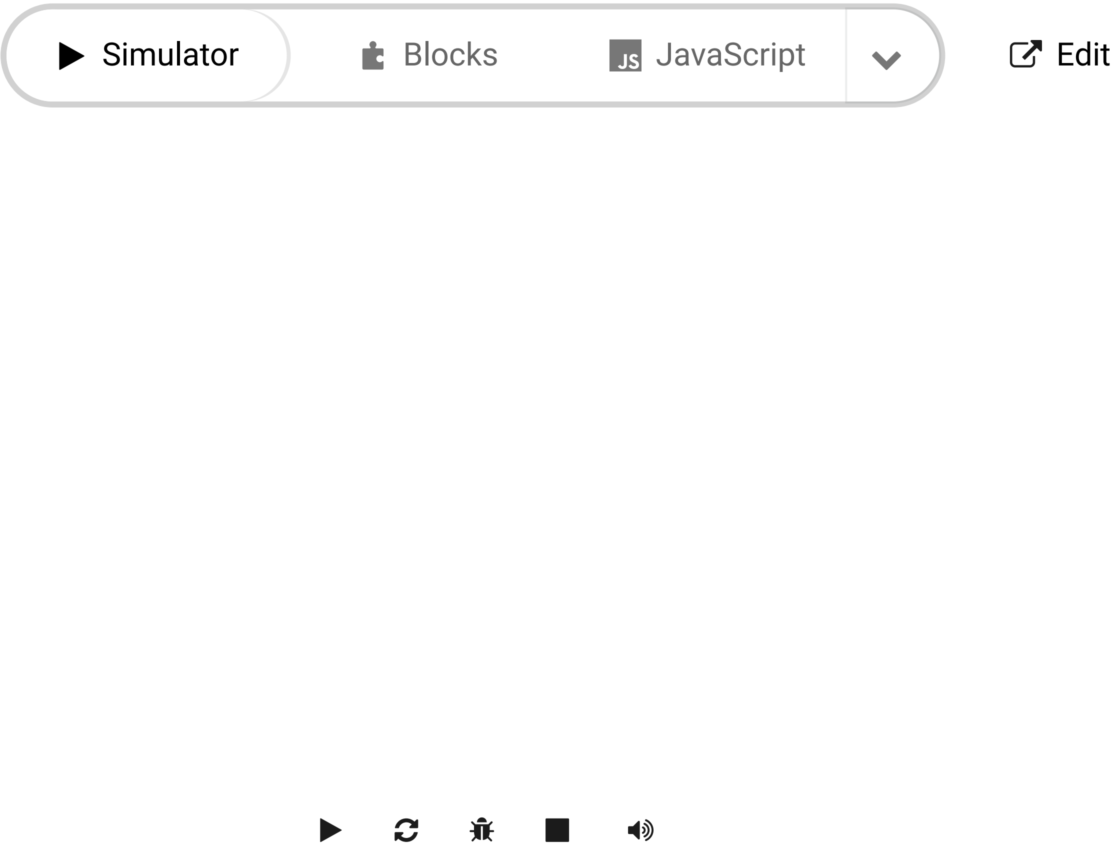


```
on start
    radio set group 1

forever
    set x to map rocker value of X from low 1023 high 0 to low -100 high 100
    set y to map rocker value of Y from low 1023 high 0 to low 100 high -100
    radio send value "x" = x
    radio send value "y" = y
```

Link

Link： https://makecode.microbit.org/_Ct3UpWKx3eb0

You may also download it directly below:



Result --- The Joystick:bit is able to control the movement of the ring:bit car.

19.5. Exploration

19.6. FAQ

19.7. Relevant File