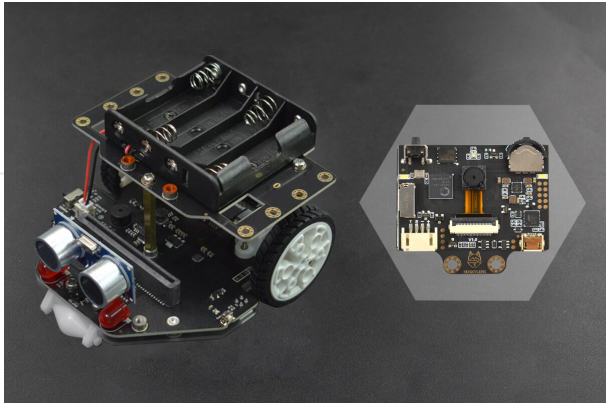
SKU:MBT0021-EN (https://www.dfrobot.com/product-2027.html)

(https://www.dfrobot.com/product-2027.html)



1. Introduction

This is the latest version of Maqueen Plus, a programming robot for STEAM education. Optimized with more expansion ports, larger capacity power supply and larger body, the Maqueen Plus V2.0 can be perfectly compatible with more peripheral components like HuskyLens AI camera and Maqueen Mechanic kits, which makes it an accessible STEAM robot teaching tool for primary and secondary students. Besides, it can be not only suitable for classroom teaching, but also can be used for after-school extended

exercises and robot competitions. Besides all the functions of Maqueen Lite, it offers richer and more flexible functions and stronger performance. Whether you have ever used Maqueen series products or not, you'll find it very easy to get started.

2. Specification

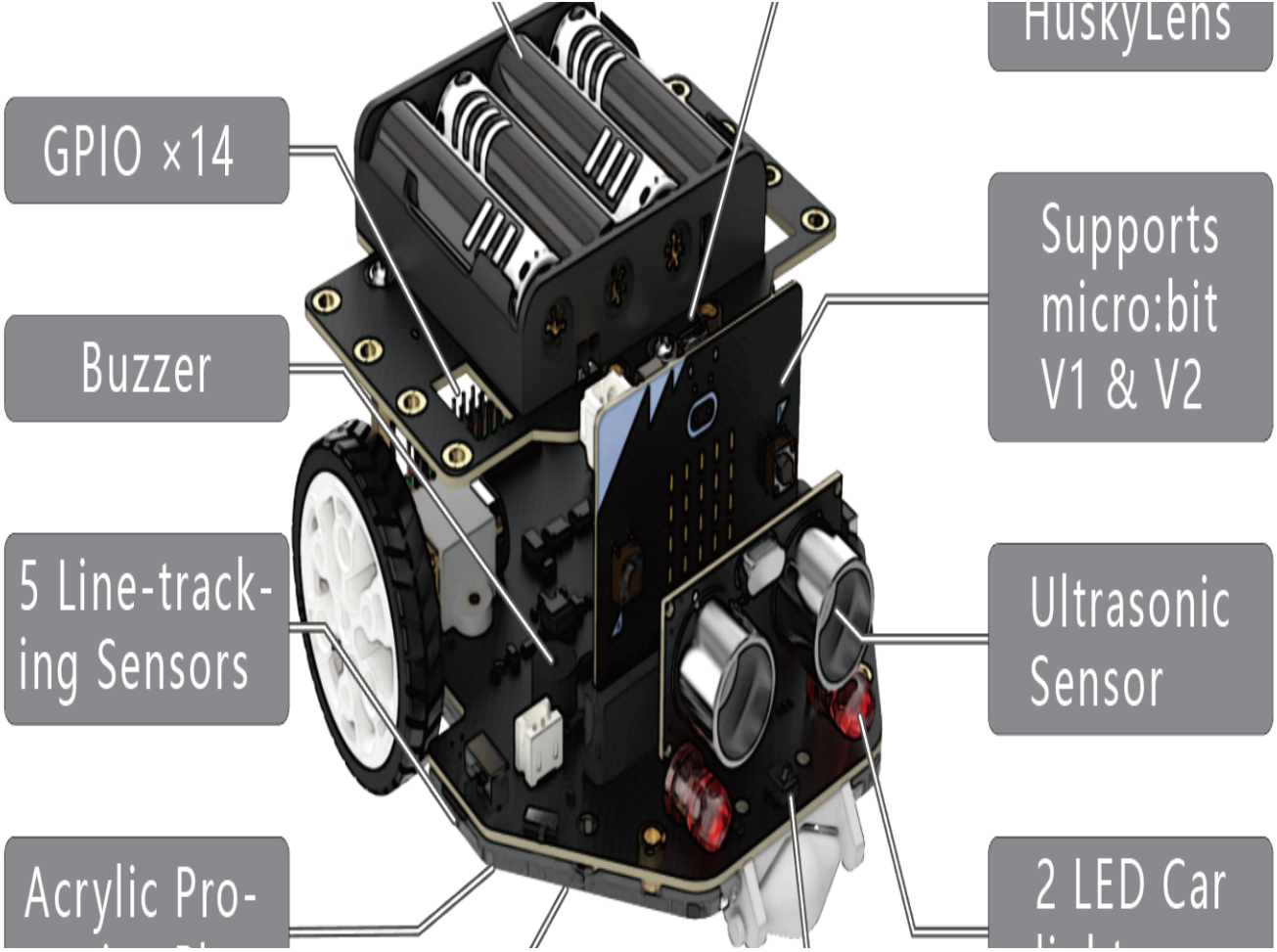
- Power Supply:
  - MBT0021-EN: 1.5V AA alkaline battery x 4 or 1.2V AA Ni-MH battery x 4
  - MBT0021-EN-18650: 18650 Li-ion battery (3.6V or 3.7V) x 1, onboard charging circuit, MicroUSB and TYPE-C charging interface, 4 hours to charge fully, last about 15h
- Support micro:bit V1 and V2
- N20 All-metal Gear Motor x 2
- Motor Reduction Ratio: 1:150
- Motor Rated Rotation Speed: 133 rpm
- Buzzer x 1
- 5V IO (P0 P1 P2) x 3
- 3.3V IO (P0 P1 P2 P8 P9 P12 P16) x 7
- 3.3V I2C x 2

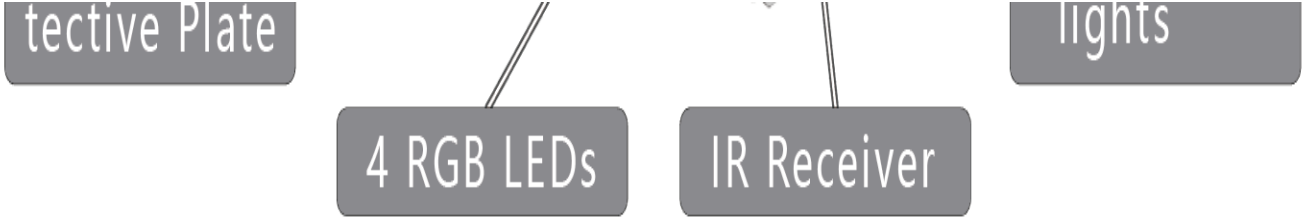
- 5V I2C x 1
- Large Size LED Car Lights x 2
- RGB Ambient Lights x 4

- Line-tracking Sensor x 5
- Line-tracking Sensor with One-key Calibration
- IR Receiver x 1
- SRO4 Ultrasonic Module x 1
- Dimension: 136mm×65mm/5.35×2.56"
- Programming Platform: Mind+, MakeCode

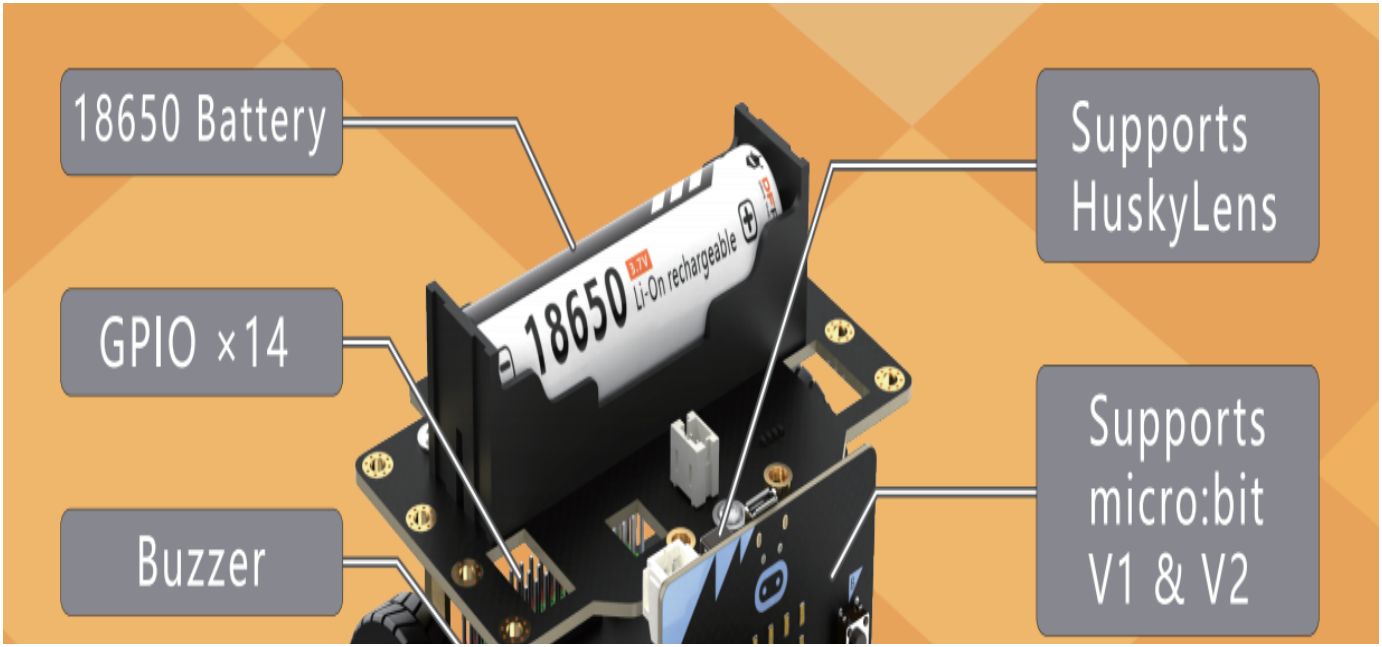
3. Board Overview

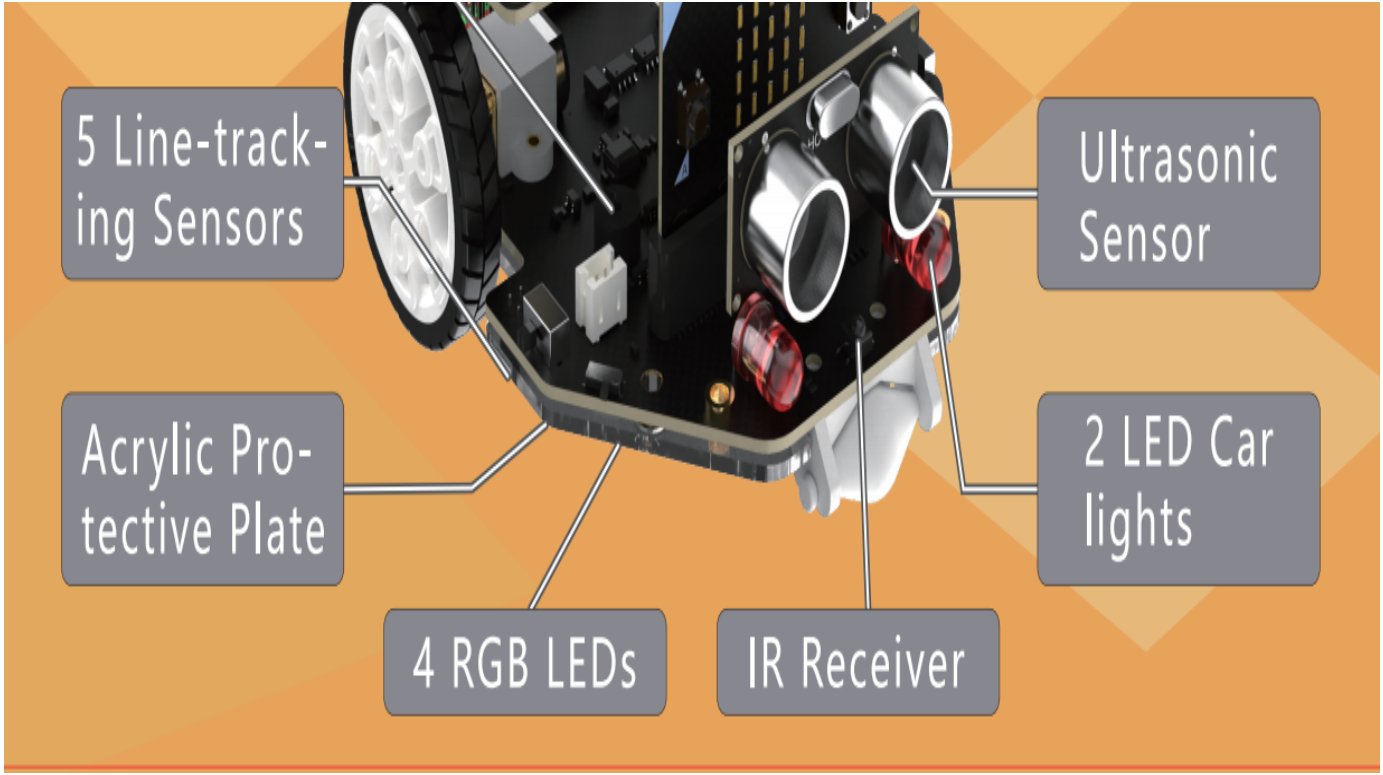
MBT0021-EN (AA Battery Version):


HuskyLens
GPIO ×14
Supports micro:bit V1 & V2
Buzzer
5 Line-tracking Sensors
Ultrasonic Sensor
Acrylic Pro-
2 LED Car



MBT0021-EN-18650(18650 Version):

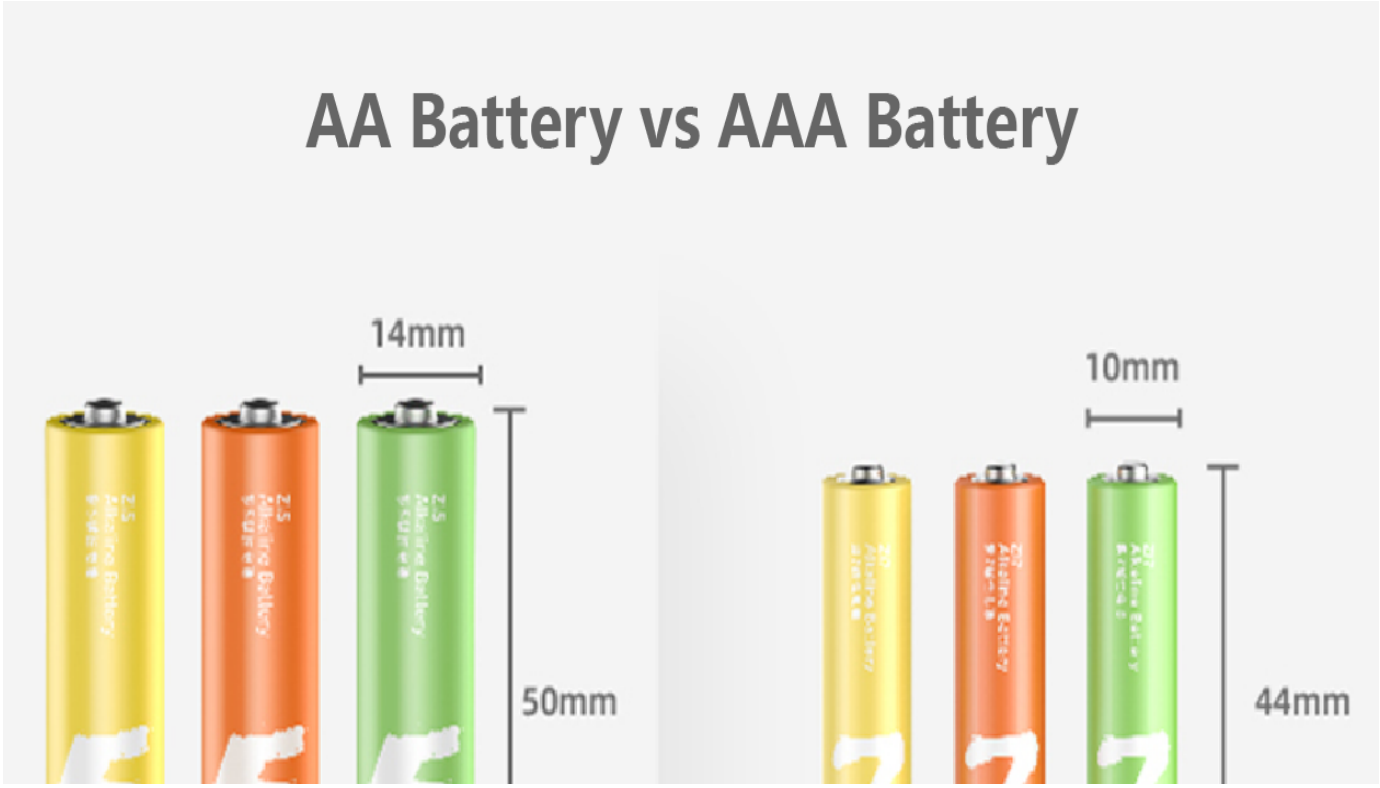


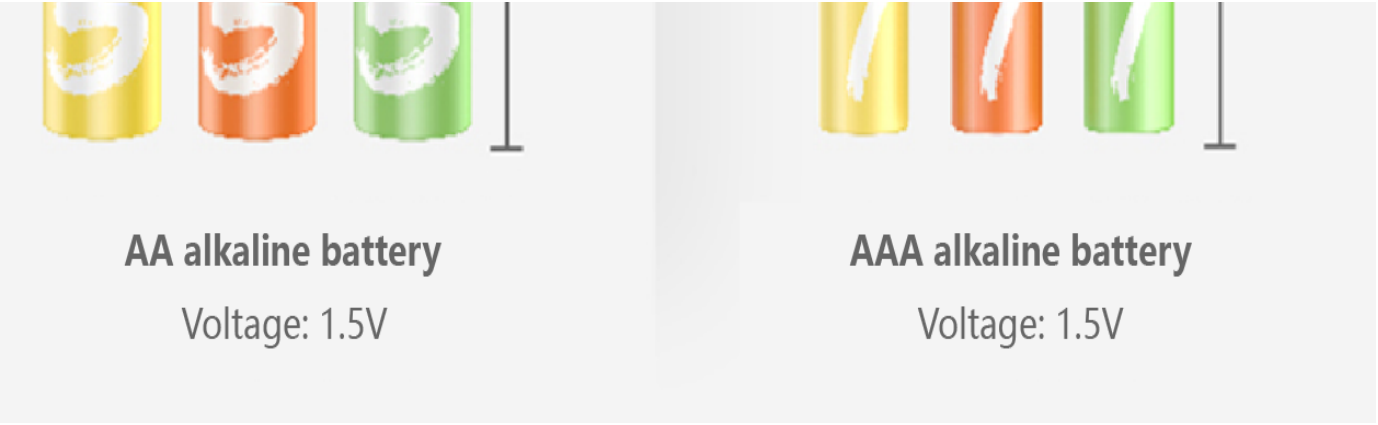


4. Battery Usage and Charging

The Maqueen PlusV2 AA battery version supports 1.5V AA batteries and 1.2V Ni-MH

rechargeable AA batteries. Please be careful not to use AAA batteries. AA batteries are available in most supermarkets. It should be noted that the 1.5V alkaline battery is disposable and cannot be recharged, otherwise it may explode or catch fire.





Ni-MH rechargeable batteries are safer than lithium batteries and are less prone to explosion and fire. The voltage is 1.2V per cell. The size of the Ni-MH battery is the same as that of the AA alkaline battery. There is no onboard charging circuit designed on Maqueen Plus V2.0, considering that it is dangerous if the user accidentally recharges the alkaline battery. Therefore, if you need to use Ni-MH batteries, you need to buy Ni-MH rechargeable batteries and charger sets.

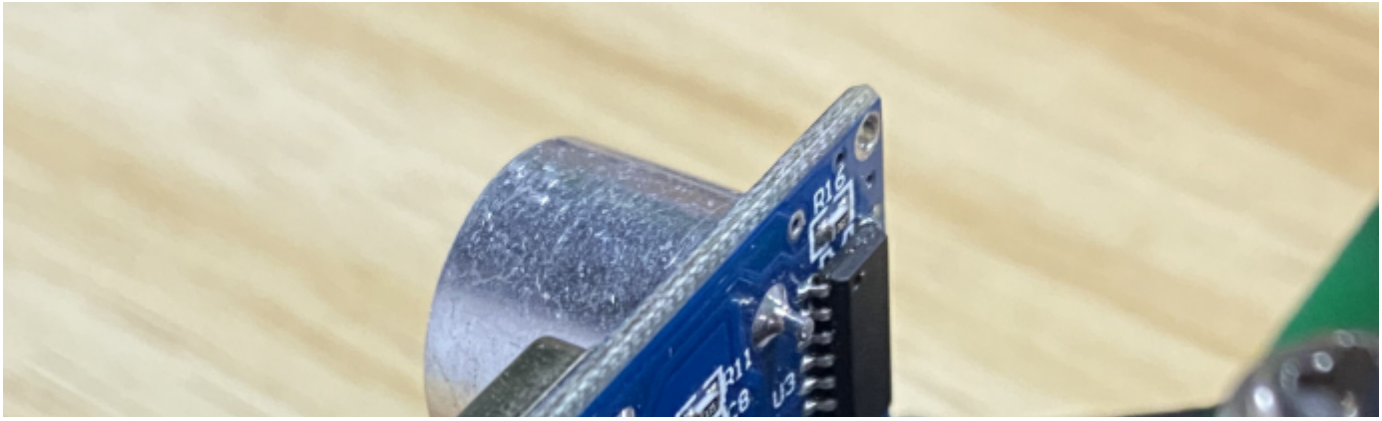
Note: Please pay attention to the polarity when installing battery.

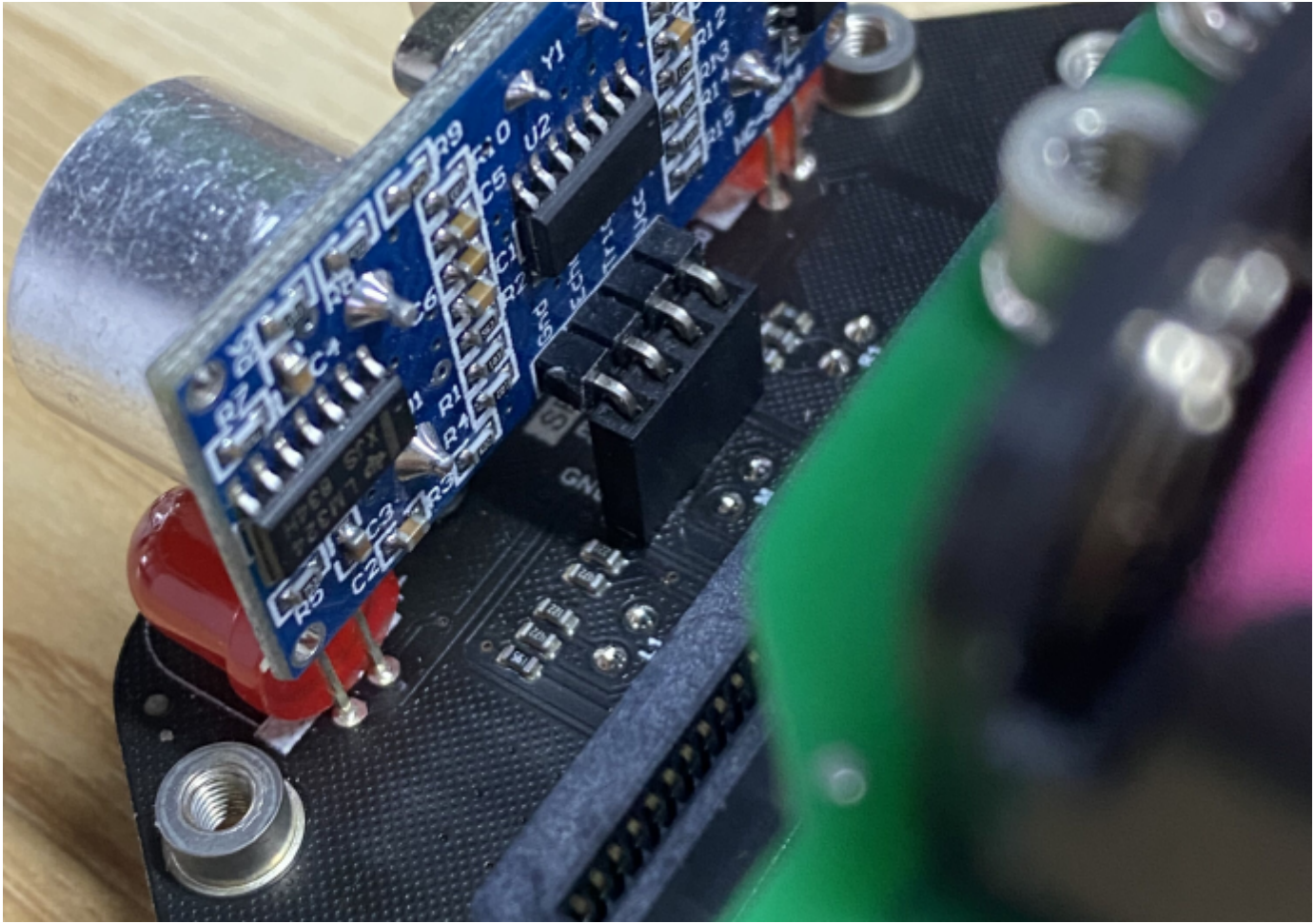
5. Quick Start Guide

Step 1. Install batteries

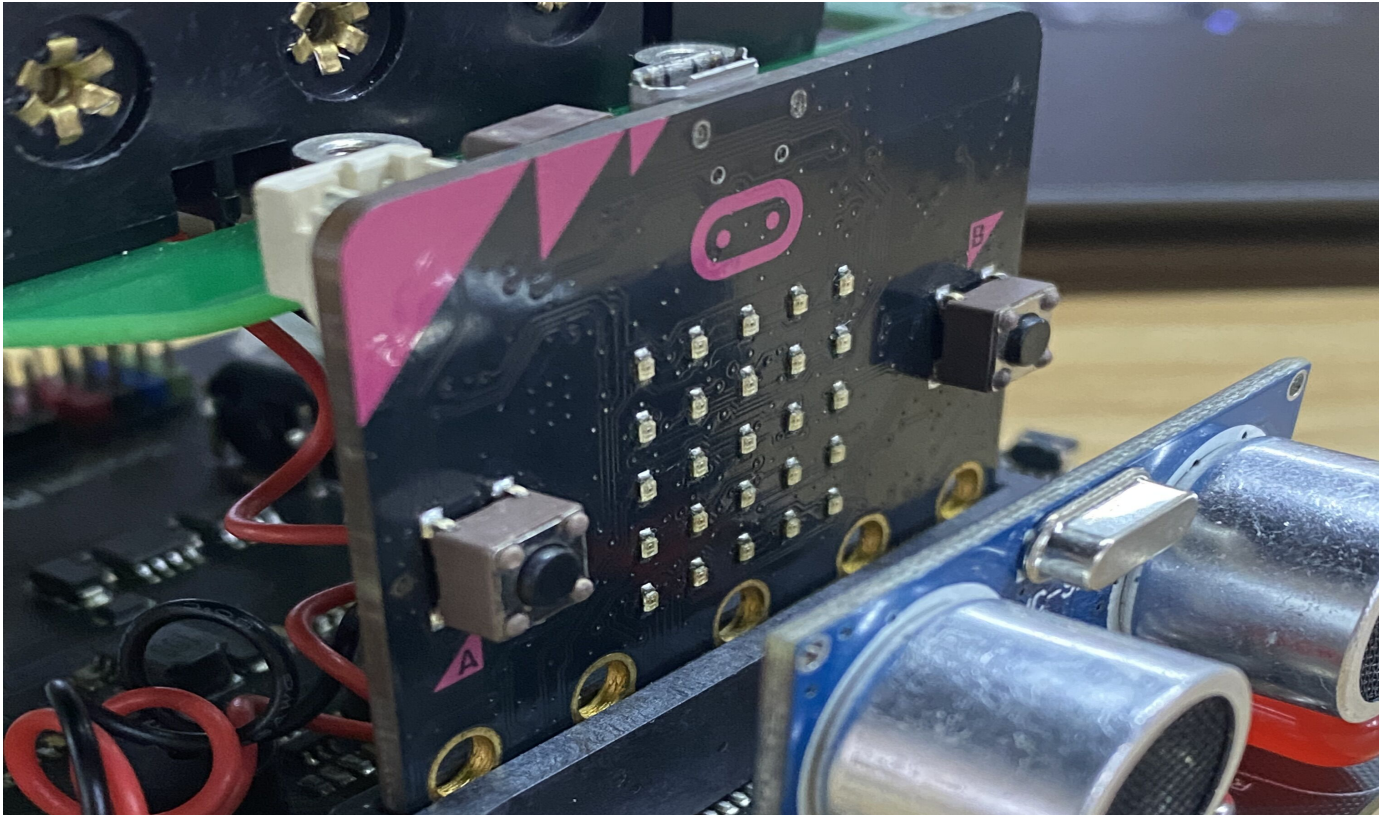
Step 1: Install battery

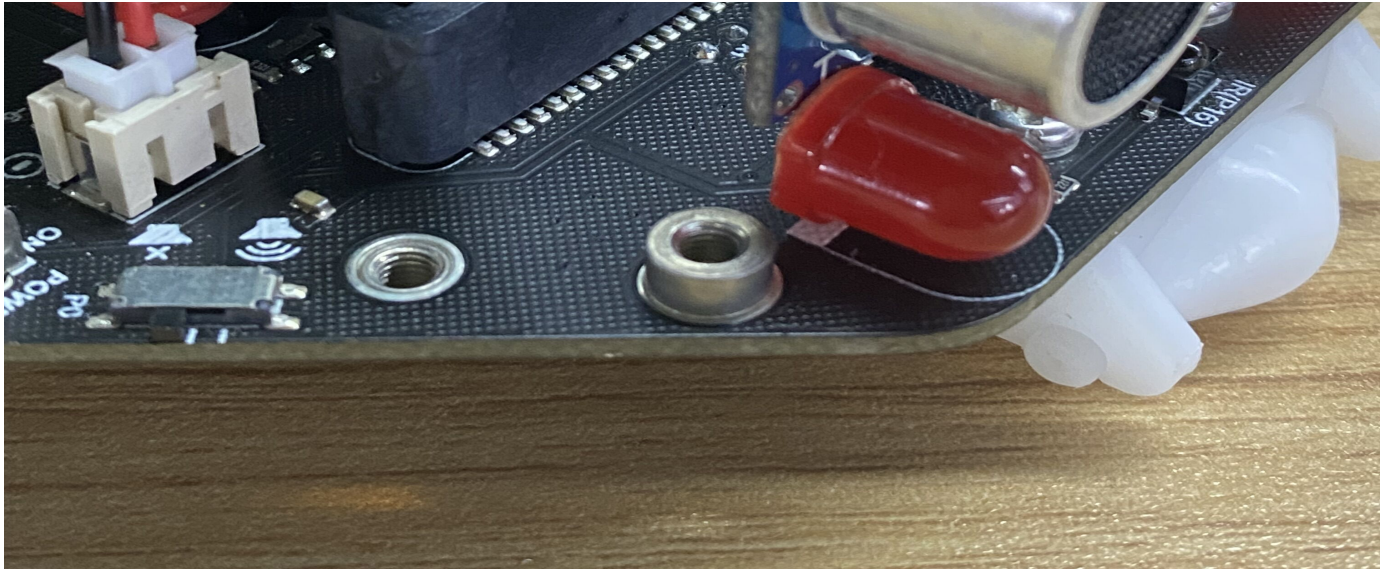
Step 2. Plug in the ultrasonic sensor



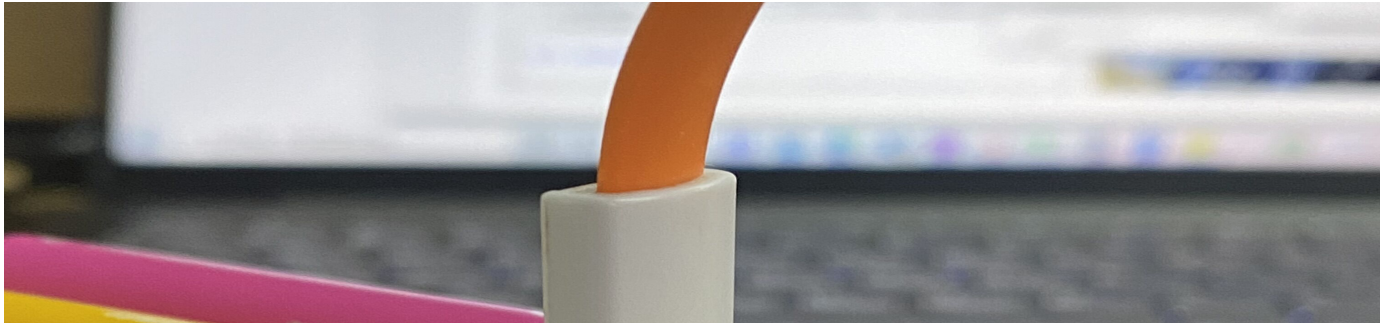


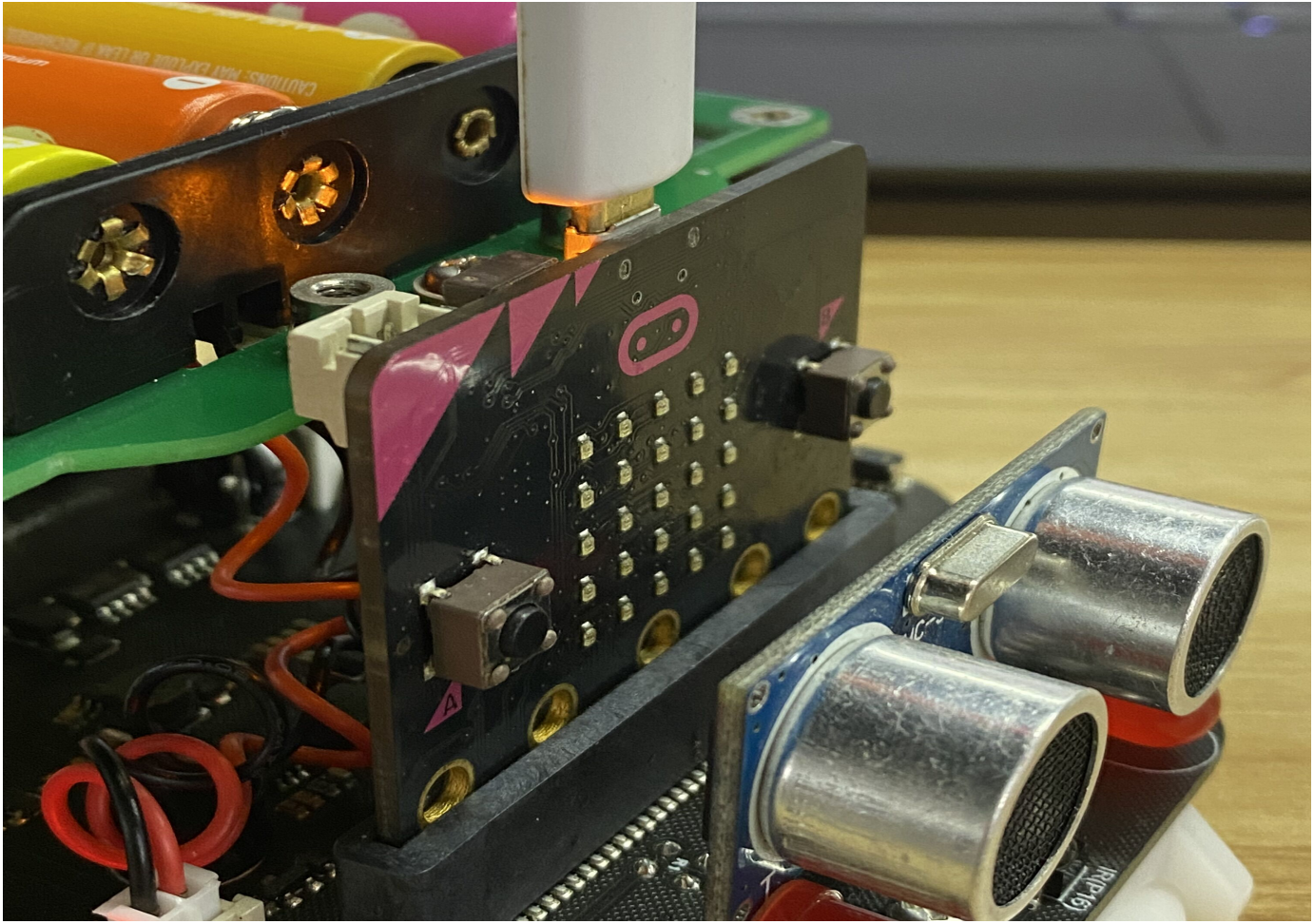
Step 3. Insert the micro:bit board





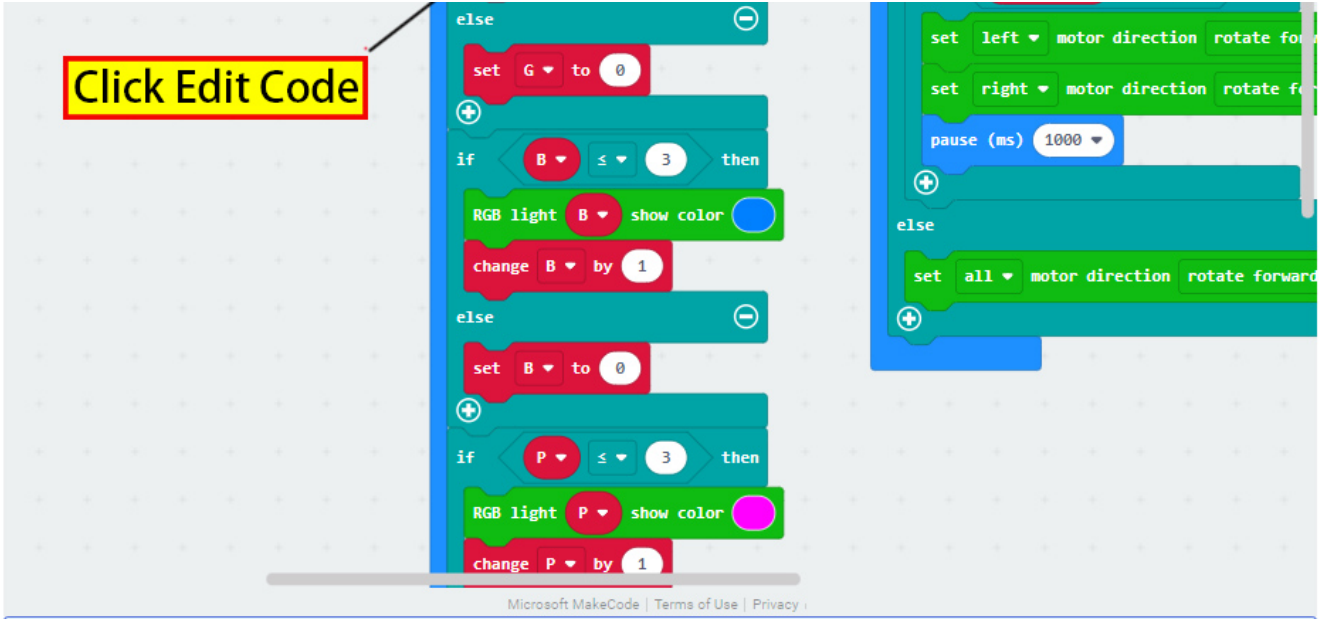
Step 4. Connect the board to a PC via USB cable

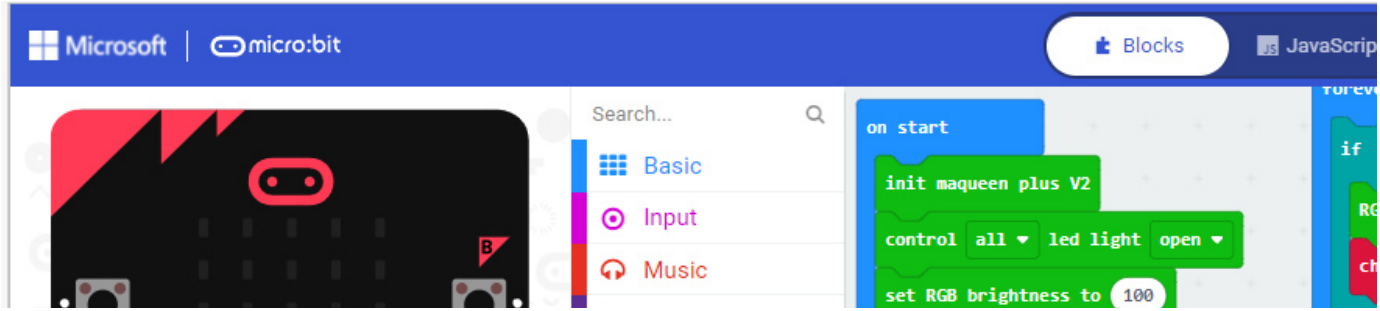


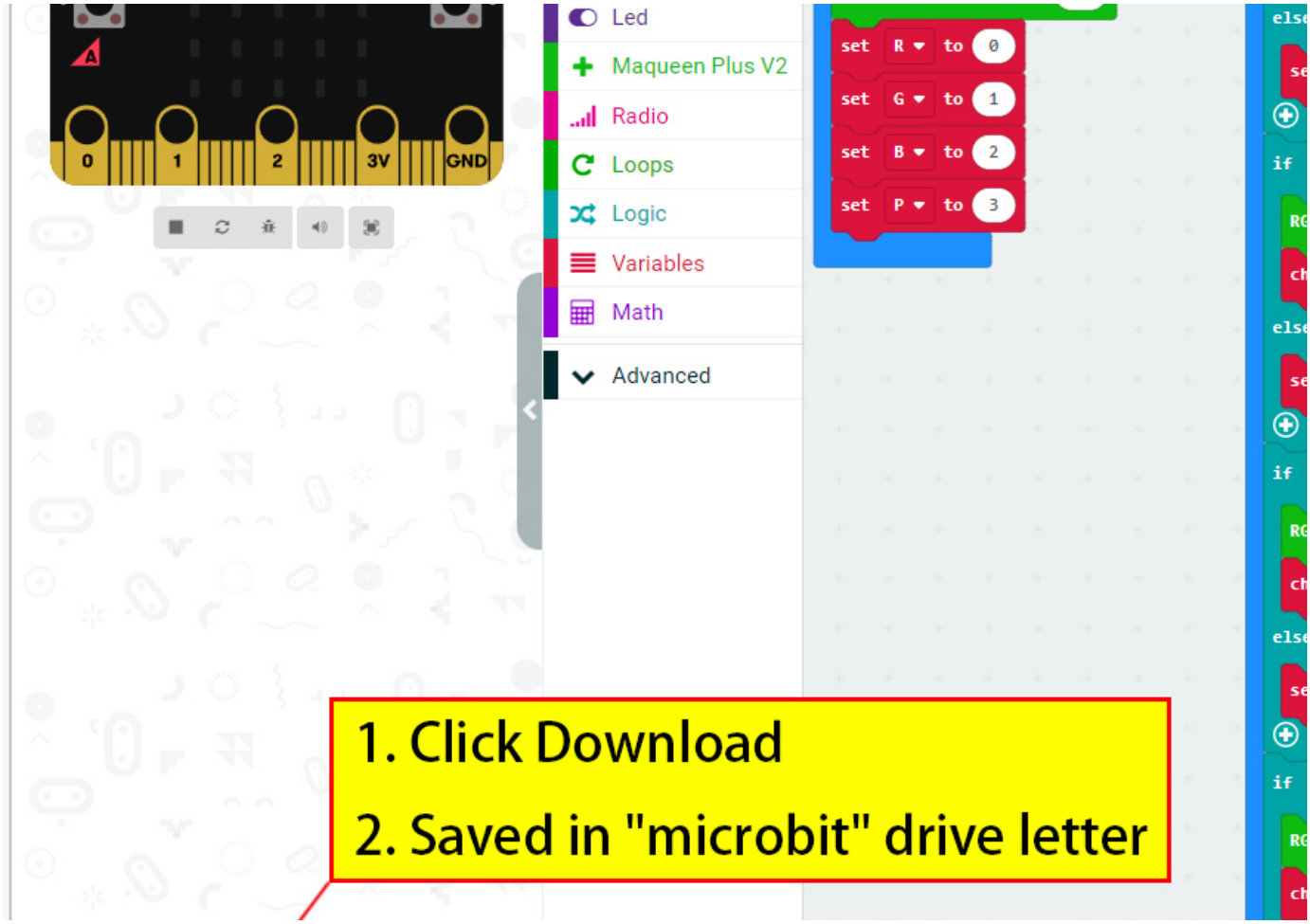


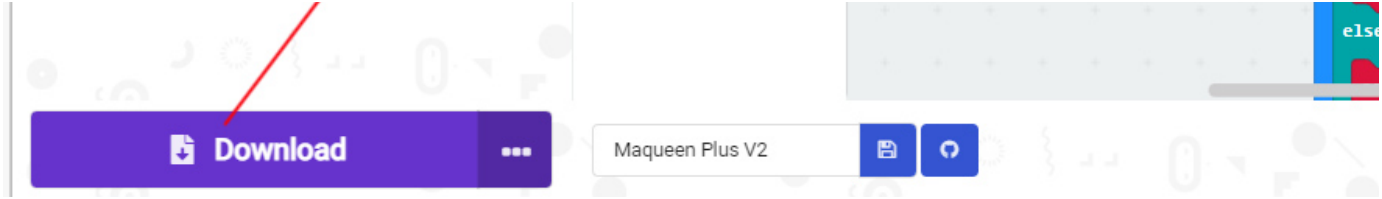
Step 5. Click the sample program link, select "Edit" on the opened webpage, and download program into micro:bit.

Sample Program: https://makecode.microbit.org/_dshFJ6f3gfaK (https://makecode.microbit.org/_dshFJ6f3gfaK)

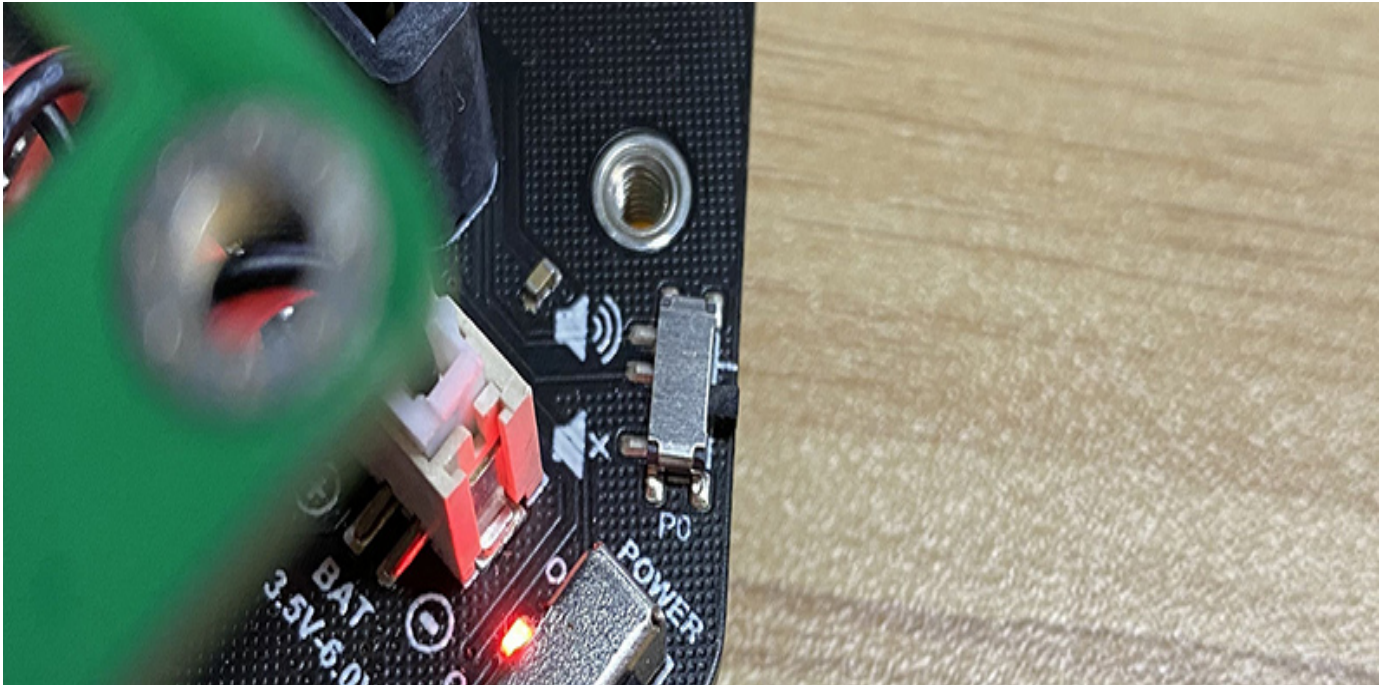





A
0
1
2
3V
GND
Led
Maqueen Plus V2
Radio
Loops
Logic
Variables
Math
Advanced
set R to 0
set G to 1
set B to 2
set P to 3
if
else
1. Click Download
2. Saved in "microbit" drive letter



Step 6. Unplug the USB cable and turn on the power switch.



After the above operations, we downloaded an automatic obstacle avoidance robot program to the board. Put Maqueen Plus on the ground, it will automatically drive forward and automatically detect whether there are obstacles within 30cm in front. If there are obstacles, Maqueen Plus will automatically turn around to avoid obstacles and continue driving.

6. MakeCode Graphical Programming

6. MakeCode Graphical Programming

- How to use MakeCode: https://wiki.dfrobot.com/Makecode%20Get-started%20Tutorial (https://wiki.dfrobot.com/Makecode%20Get-started%20Tutorial)
- MakeCode Library Address: https://github.com/DFRobot/pxt-DFRobot_MaqueenPlus_v20 (https://github.com/DFRobot/pxt-DFRobot_MaqueenPlus_v20)

6.1 Light-controlled Maqueen Plus

The following program calls the light sensor of the micro:bit motherboard. Normally, the car is in a stopped state, and the RGB light at the bottom lights up in color. When the flashlight illuminates the front side of the micro:bit, the car goes forward, the bottom RGB light goes out, and the front lights turn on.

The value of the light sensor is an analog value between 0 and 255, which is used to indicate the intensity of light. In the program, we take a light intensity value of 100 as the demarcation point. When the light intensity is greater than 100, the car is started, otherwise, the car is stopped.

Program Link: https://makecode.microbit.org/_26hJo3HLbaqw (https://makecode.microbit.org/_26hJo3HLbaqw)

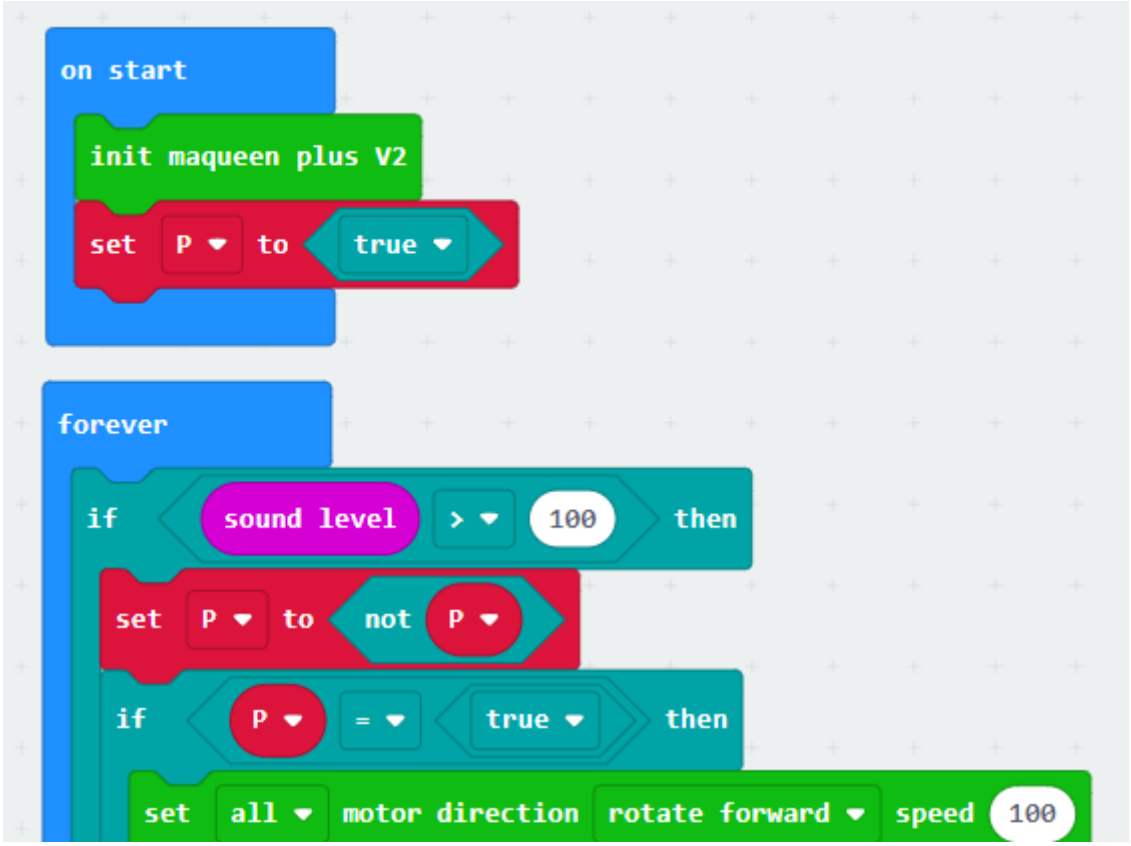

on start
init maqueen plus V2
forever
if
light level
>
100
then
set
left
motor direction
rotate forward
speed
100
control
all
led light
open
clear all RGB
else

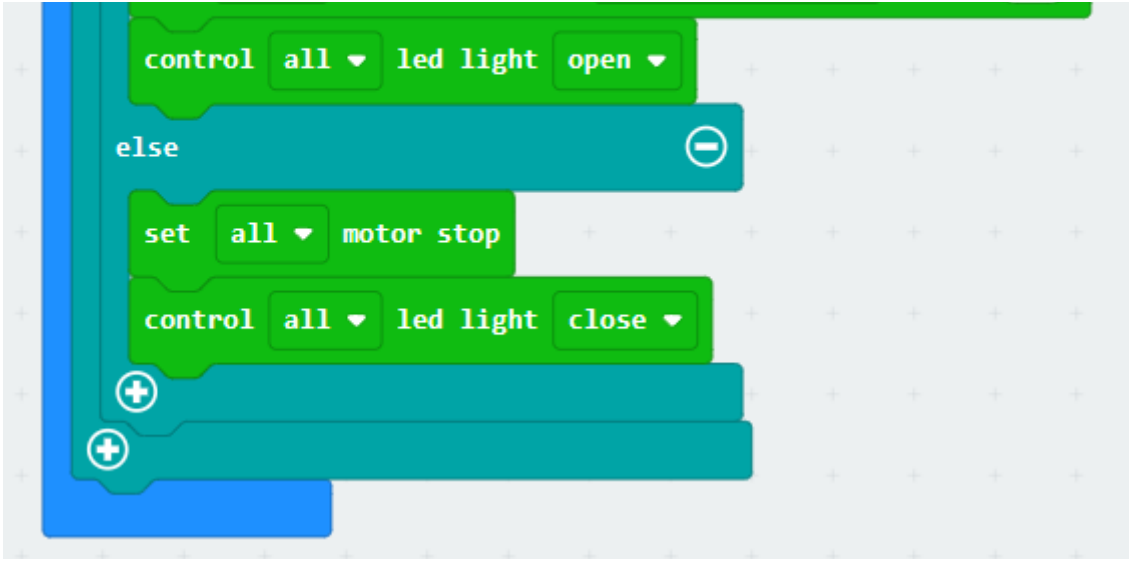
6-2 Sound-controlled Maqueen Plus(only supports micro:bit V2.0)

The following program calls the sound sensor on the micro:bit V2 motherboard. When you clap your hands, the car starts to drive and turns on the lights. When you clap your hands again, the car stops and turns off the lights. Repeat in this way.

In the program, when the sound is detected to be greater than 100, the state of a Boolean value is reversed. Next, the program checks whether the Boolean value is true, if it is true, the car moves forward, if it is false, the car stops.

Program Link: https://makecode.microbit.org/_T89YbzEyeF2m (https://makecode.microbit.org/_T89YbzEyeF2m)





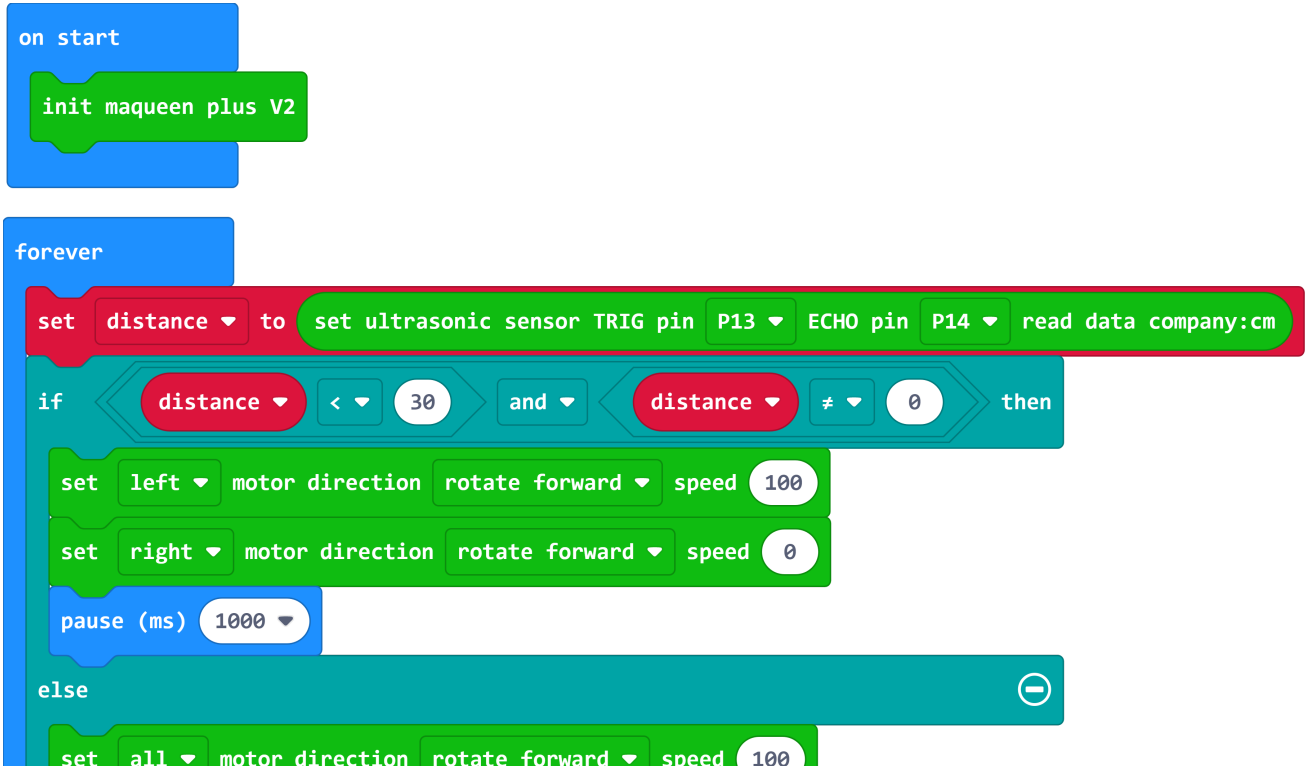
6-3 obstacle Avoidance Robot

The function realized by the following program is: Maqueen Plus detects whether there is an obstacle in front of it during driving. If there is an obstacle about 30cm ahead, turn to avoid the obstacle and continue driving.

In the program, the distance read by the ultrasonic is assigned to a variable, and then it is judged that if the value of this variable is less than 30, the turning action is performed for one second.

After turning for one second, it will continue to detect whether the distance is less than 30cm. If there are no obstacles 30cm ahead, the car will go straight ahead.

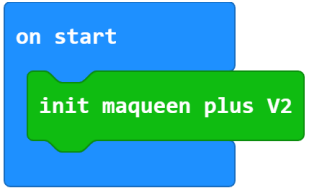
Program Link: https://makecode.microbit.org/_AR3X7gP4cVim
(https://makecode.microbit.org/_AR3X7gP4cVim)

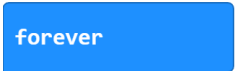

on start
init maqueen plus V2
forever
set distance to set ultrasonic sensor TRIG pin P13 ECHO pin P14 read data company:cm
if distance < 30 and distance ≠ 0 then
set left motor direction rotate forward speed 100
set right motor direction rotate forward speed 0
pause (ms) 1000
else
set all motor direction rotate forward speed 100

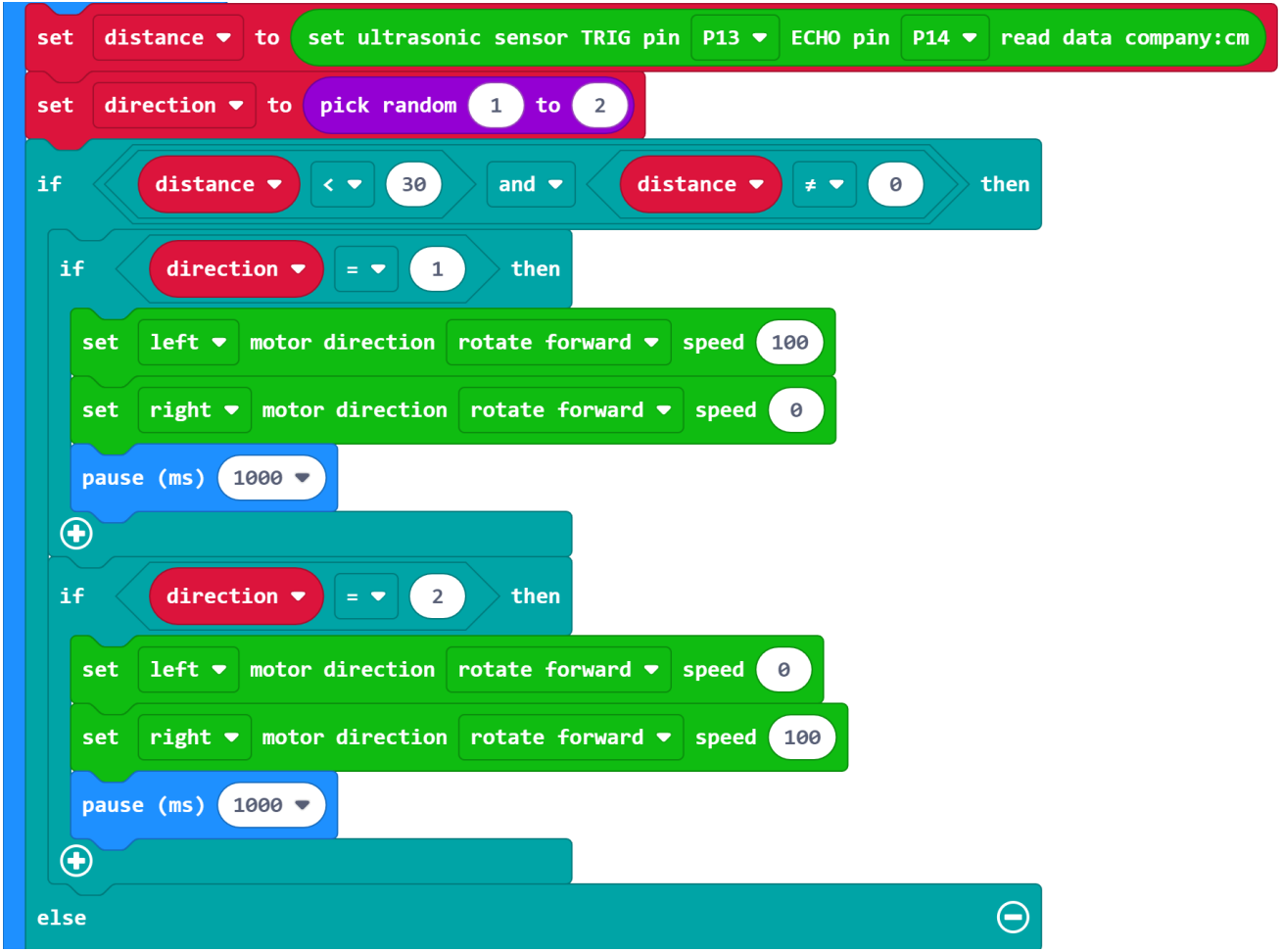
After the above procedure is executed, you will find that the car always turns in one direction after encountering an obstacle. Below we will improve the program as follows, let the car encounter obstacles and randomly choose a turning direction to continue driving.

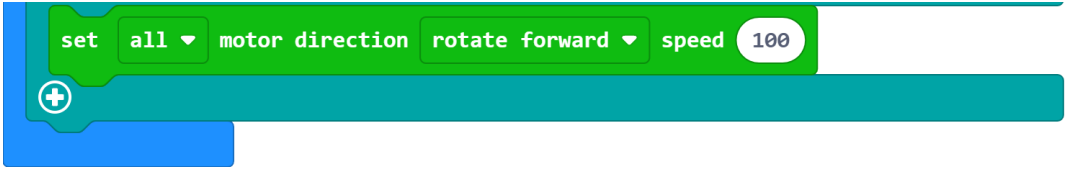
In the program, a module that generates integers randomly is used to randomly produce 1 or 2. If it is judged that the random number is 1, then turn right, if the random number is 2, then turn left.

Program Link: https://makecode.microbit.org/_JyKALkaeh5j5 (https://makecode.microbit.org/_JyKALkaeh5j5)






set distance to set ultrasonic sensor TRIG pin P13 ECHO pin P14 read data company:cm
set direction to pick random 1 to 2
if distance < 30 and distance ≠ 0 then
if direction = 1 then
set left motor direction rotate forward speed 100
set right motor direction rotate forward speed 0
pause (ms) 1000
if direction = 2 then
set left motor direction rotate forward speed 0
set right motor direction rotate forward speed 100
pause (ms) 1000
else



6-4 Line-tracking Robot

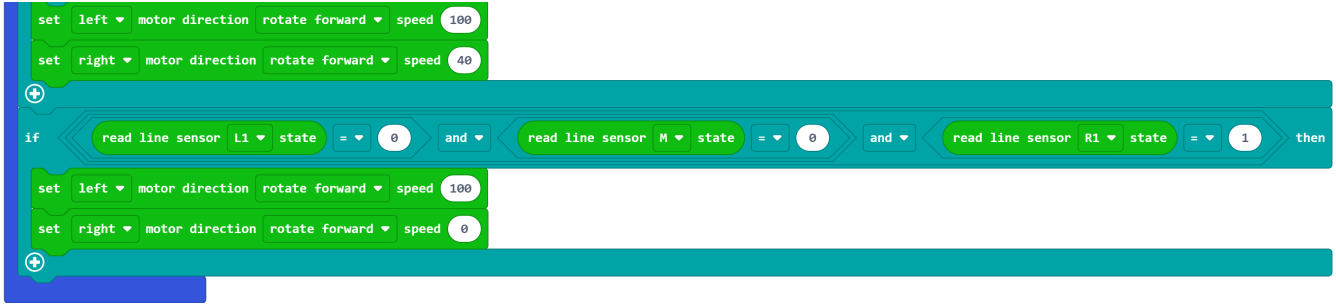
The function of the following program is: put the car on the circular black line map, the car will use the three line-tracking sensor probes set at the bottom to detect whether the car is driving on the black line. If it deviates from the black line, the car will correct its driving direction in real time so that the car will always follow the black line.

There are 5 line-tracking sensor probes in Maqueen Plus V2. In this program, three probes, L1, R1, and M, are used to detect black lines. L2 and R2 are not used yet.

When the line-tracking probe detects a black line, the output value is 1, and when it detects a white line, the output value is 0. With this feature, three line-tracking sensors can be used to detect the black lines, and the status of multiple probes can be used to determine the position of the black line in real time and how the car should turn to correct deviation.

Program Link: https://makecode.microbit.org/_TMa48UKa5XAA (https://makecode.microbit.org/_TMa48UKa5XAA)


on start
init maqueen plus V2
forever
while
read line sensor L1 state = 0 or read line sensor M state = 0 or read line sensor R1 state = 0
do
call Line patrol
function Line patrol
if
read line sensor L1 state = 0 and read line sensor M state = 1 and read line sensor R1 state = 0 then
set all motor direction rotate forward speed 100
if
read line sensor L1 state = 1 and read line sensor M state = 1 and read line sensor R1 state = 0 then
set left motor direction rotate forward speed 40
set right motor direction rotate forward speed 100
if
read line sensor L1 state = 1 and read line sensor M state = 0 and read line sensor R1 state = 0 then
set left motor direction rotate forward speed 0
set right motor direction rotate forward speed 100
if
read line sensor L1 state = 0 and read line sensor M state = 1 and read line sensor R1 state = 1 then

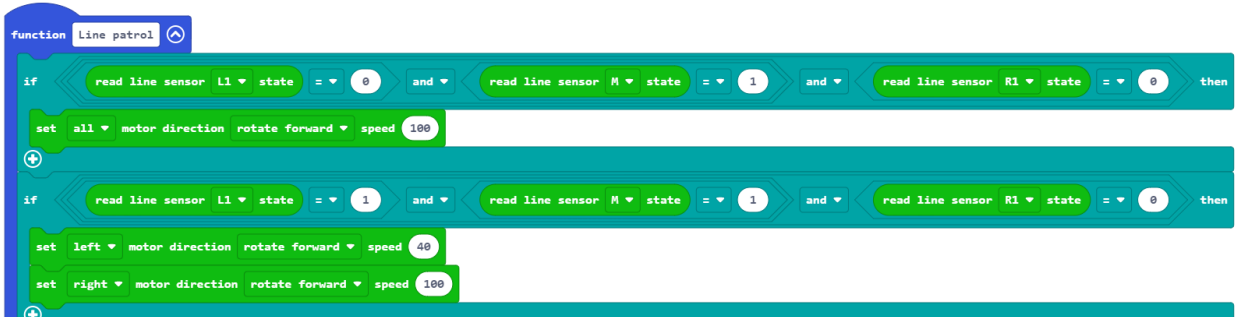


In the above procedure, we did not use the two probes L2 and R2 on the two sides. If the car is driven out by mistake, it will be difficult to return to find the black line to continue driving. In the following procedure, we will use five probes and add another layer of detection on the left and right sides, so that the car can return in time after it finds that it was on the wrong path.

Program Link: https://makecode.microbit.org/_52di8s3WKMoF (https://makecode.microbit.org/_52di8s3WKMoF)

https://makecode.microbit.org/_XucP5zJyeCba (https://makecode.microbit.org/_XucP5zJyeCba)

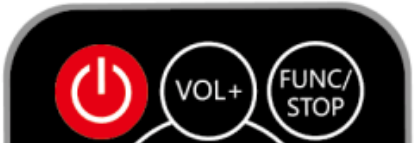

do
if read line sensor L2 state = 1 and read line sensor R2 state = 0 then
set left motor direction rotate forward speed 0
set right motor direction rotate forward speed 100
if read line sensor L1 state = 1 or read line sensor M state = 1 or read line sensor R1 state = 1 then
call Line patrol
if read line sensor L2 state = 0 and read line sensor R2 state = 1 then
set left motor direction rotate forward speed 100
set right motor direction rotate forward speed 0
if read line sensor L1 state = 1 or read line sensor M state = 1 or read line sensor R1 state = 1 then
call Line patrol
call Line patrol


function Line patrol
if read line sensor L1 state = 0 and read line sensor M state = 1 and read line sensor R1 state = 0 then
set all motor direction rotate forward speed 100
if read line sensor L1 state = 1 and read line sensor M state = 1 and read line sensor R1 state = 0 then
set left motor direction rotate forward speed 40
set right motor direction rotate forward speed 100

6-5 IR-controlled Maqueen Plus

First, let us get familiar with the infrared remote control and key values. Each button on the infrared remote control corresponds to a unique key value. The key value has two systems of numeration, hexadecimal and decimal. In Maqueen PlusV2, we use decimal. as the picture shows:

Key	Value (In hexadecimal)	Value (In decimal)
Red Key	0xff00	0

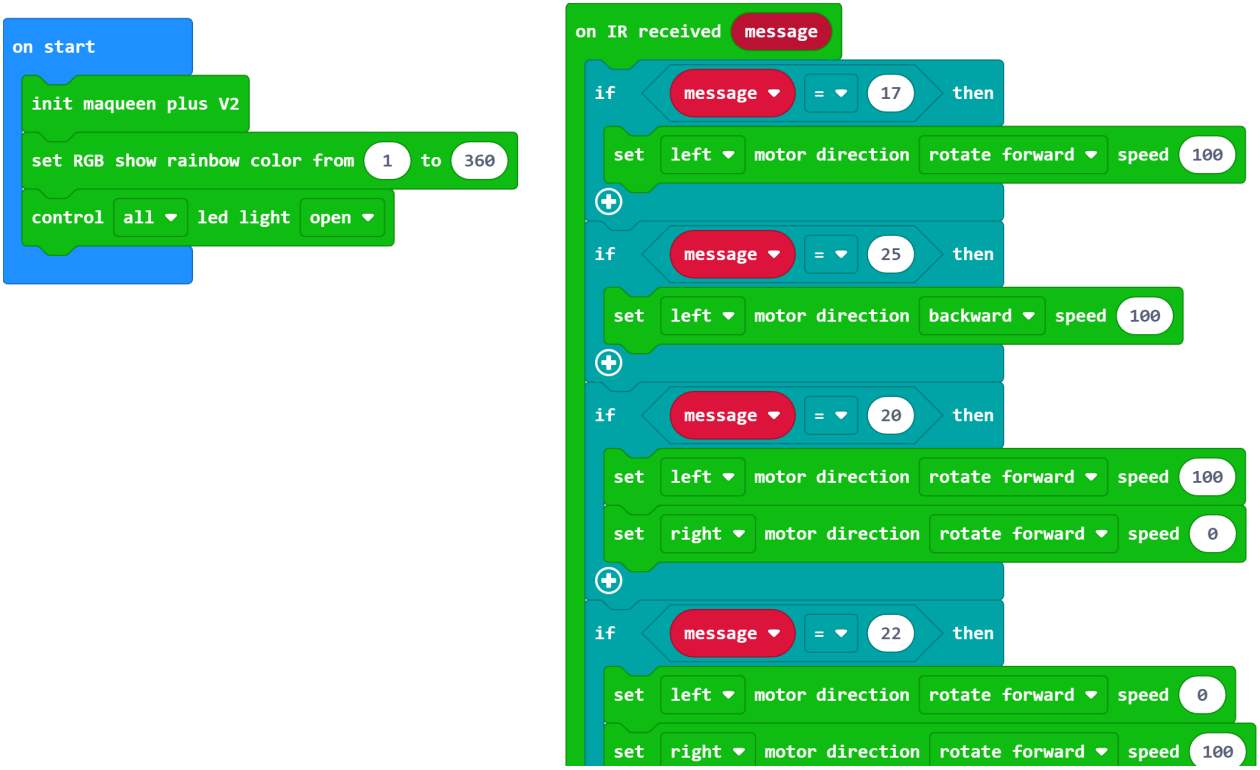


VOL+	0xfe01	1
FUNC/STOP	0xfd02	2
Left Arrow	0xfb04	4
Pause	0xfa05	5
Right Arrow	0xf906	6
Down Arrow	0xf708	8
VOL-	0xf609	9
Up Arrow	0xf50a	10
0	0xf30c	12
EQ	0xf20d	13
ST/REPT	0xf10e	14
1	0xef10	16
2	0xee11	17
3	0xed12	18
4	0xeb14	20
5	0xea15	21
6	0xe916	22
7	0xe718	24

8	0xe619	25
9	0xe51a	26

Example 1: In the following example, we use the four buttons 2, 4, 6, and 8 of the infrared remote control to control the car to go forward, turn left, turn right, and back separately. At the same time, we will let the bottom RGB lights and LED car lights light up.

Program Link: https://makecode.microbit.org/_6DgH54fhb3sE (https://makecode.microbit.org/_6DgH54fhb3sE)


on start
init maqueen plus V2
set RGB show rainbow color from 1 to 360
control all led light open
on IR received message
if message = 17 then
set left motor direction rotate forward speed 100
if message = 25 then
set left motor direction backward speed 100
if message = 20 then
set left motor direction rotate forward speed 100
set right motor direction rotate forward speed 0
if message = 22 then
set left motor direction rotate forward speed 0
set right motor direction rotate forward speed 100

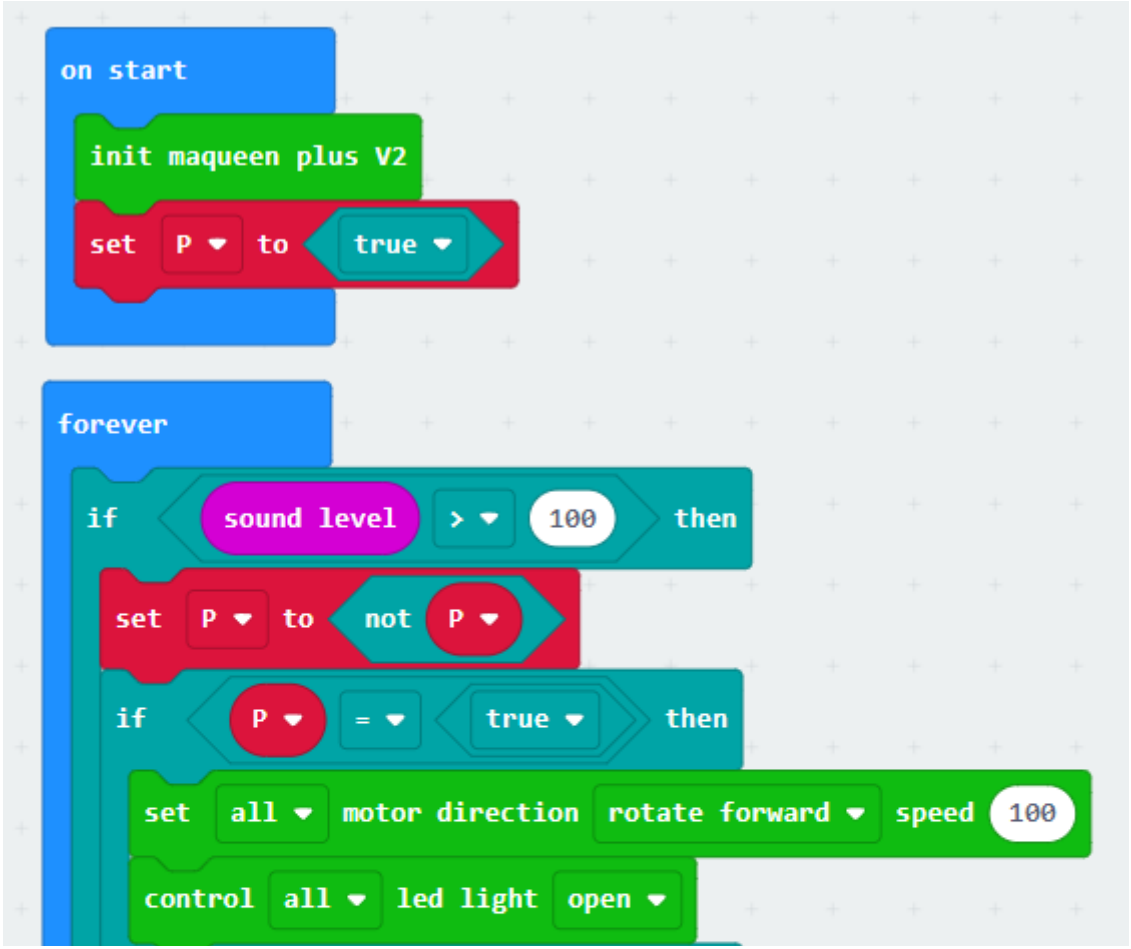
Download the program to the micro:bit, press the “2” button of the infrared remote control, and Maqueen PlusV2 will drive forward. Press the “4” button, Maqueen PlusV2 turns to the left, press the “6” button, Maqueen PlusV2 turns to the right, press the “8” button, Maqueen PlusV2 drives backwards.

However, during the operation, we found two problems:

1.When I let the car turn, as long as I press the “4” button or the “6” button, the car will spin on the spot, the control feels inflexible, and it does not conform to the actual vehicle control method.

2.When the car is moving backwards, press the “2” button to make the car move forward, and the car will fall backward due to the reaction force. We try to optimize the code and operating experience:

Program Link: https://makecode.microbit.org/_TboWd2ihjD2k (https://makecode.microbit.org/_TboWd2ihjD2k)


on start
init maqueen plus V2
set P to true
forever
if sound level > 100 then
set P to not P
if P = true then
set all motor direction rotate forward speed 100
control all led light open

In this example, we have made some changes to the remote control method. When the “4” key (key value 20) or “6” key (key value 22) is pressed, the car is only allowed to turn for a short time (200 milliseconds). In this way, we press once, the car rotates a bit, and then continues straight ahead. If we need a big turn, press it a few more times. When backing, the car will not go back quickly, which will cause the car to roll over due to inertia. Therefore, we changed the back function of pressing the “8” key (key value 25) to stop. In this program, we did not write a part of the program to make the car back. When you write your own program, try to use another button to realize the back function.

6-6. Control Maqueen Plus by Remote Controller

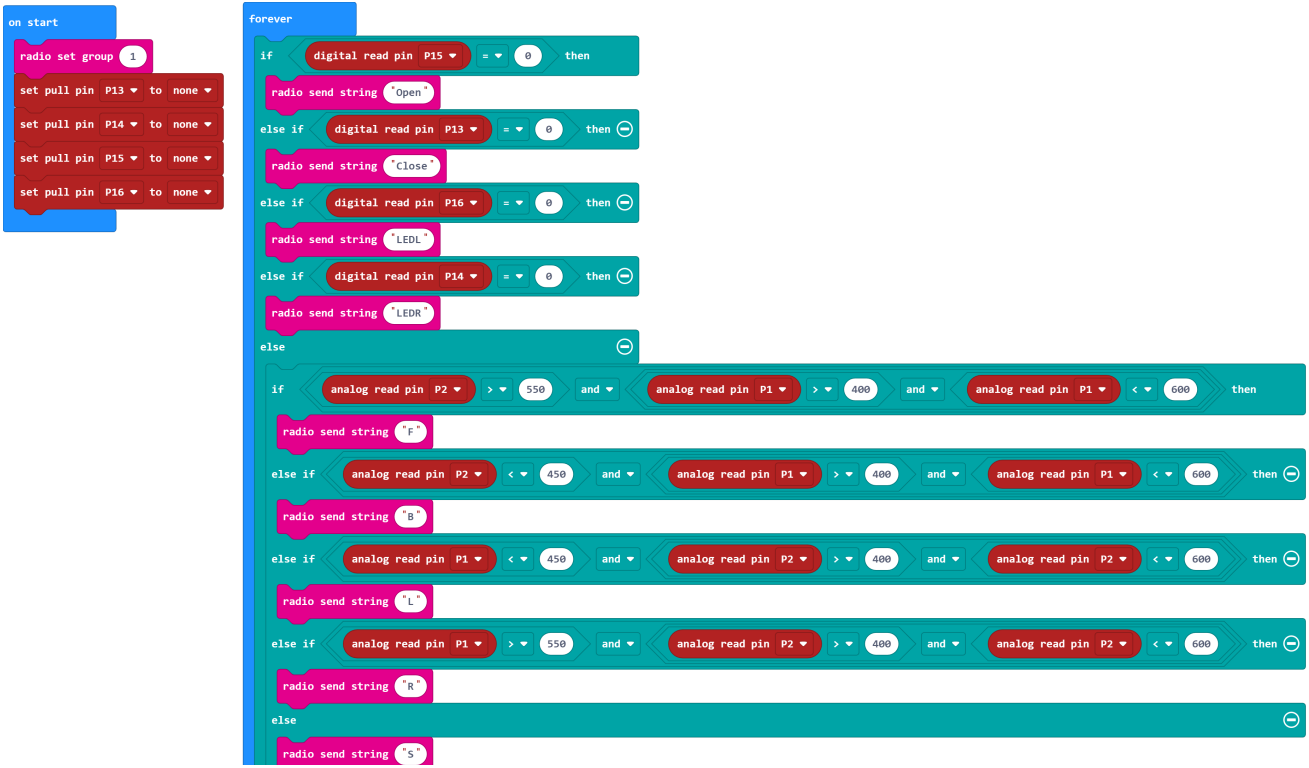
Here, another way and product to control Maqueen will be used: remote control handle.

Program Link: https://www.dfrobot.com.cn/goods-1674.html
(https://www.dfrobot.com.cn/goods-1674.html)

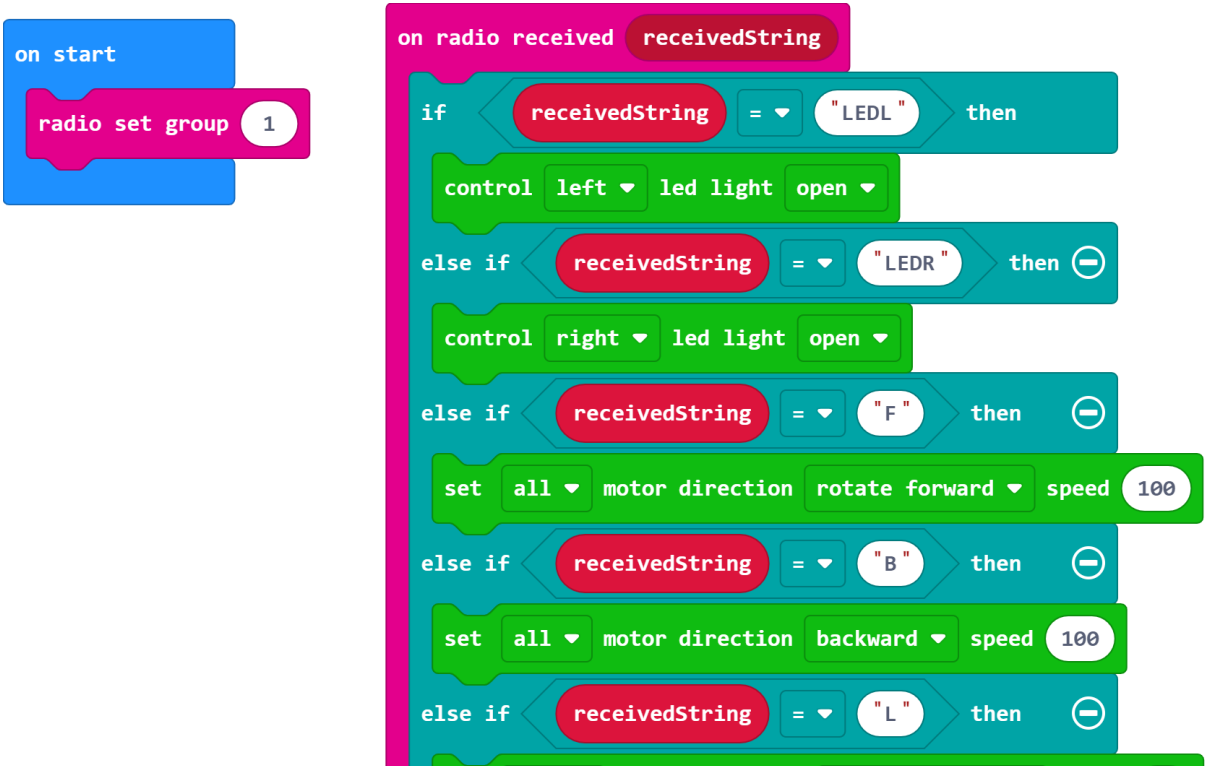
Wiki Link: https://wiki.dfrobot.com/Micro_bit_Gamepad_Expansion_Board_SKU__DFR0536
(https://wiki.dfrobot.com/Micro_bit_Gamepad_Expansion_Board_SKU__DFR0536)

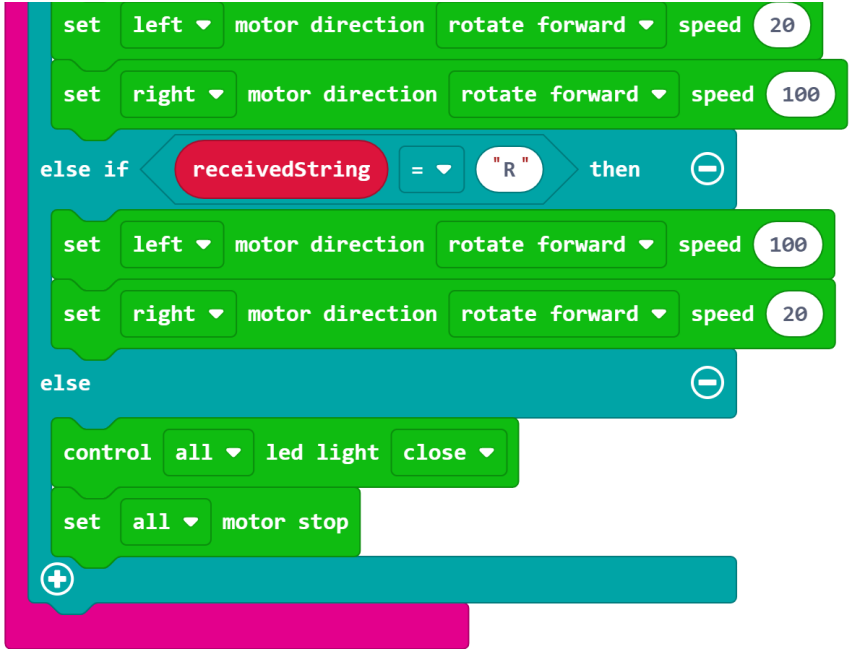
Another micro:bit motherboard needs to be installed on the remote control handle, and the two motherboards communicate through the wireless network to realize the remote control function. Compared with infrared, wireless communication has the characteristics of no directionality and long distance. Next we try to write a remote control car. This time you need to program two motherboards. The motherboard of the car terminal: write the receiving and executing program. Handle end: need to write a program to send instructions.

Remote Controller Program Link: https://makecode.microbit.org/_Ew9VCh414YW3 (https://makecode.microbit.org/_Ew9VCh414YW3)


on start
radio set group 1
set pull pin P13 to none
set pull pin P14 to none
set pull pin P15 to none
set pull pin P16 to none
forever
if digital read pin P15 = 0 then
radio send string "Open"
else if digital read pin P13 = 0 then
radio send string "Close"
else if digital read pin P16 = 0 then
radio send string "LEDL"
else if digital read pin P14 = 0 then
radio send string "LEDR"
else
if analog read pin P2 > 550 and analog read pin P1 > 400 and analog read pin P1 < 600 then
radio send string "F"
else if analog read pin P2 < 450 and analog read pin P1 > 400 and analog read pin P1 < 600 then
radio send string "B"
else if analog read pin P1 < 450 and analog read pin P2 > 400 and analog read pin P2 < 600 then
radio send string "L"
else if analog read pin P1 > 550 and analog read pin P2 > 400 and analog read pin P2 < 600 then
radio send string "R"
else
radio send string "S"

Maqueen Plus Car Program Link: https://makecode.microbit.org/_bTPhA5JDs6mA (https://makecode.microbit.org/_bTPhA5JDs6mA)



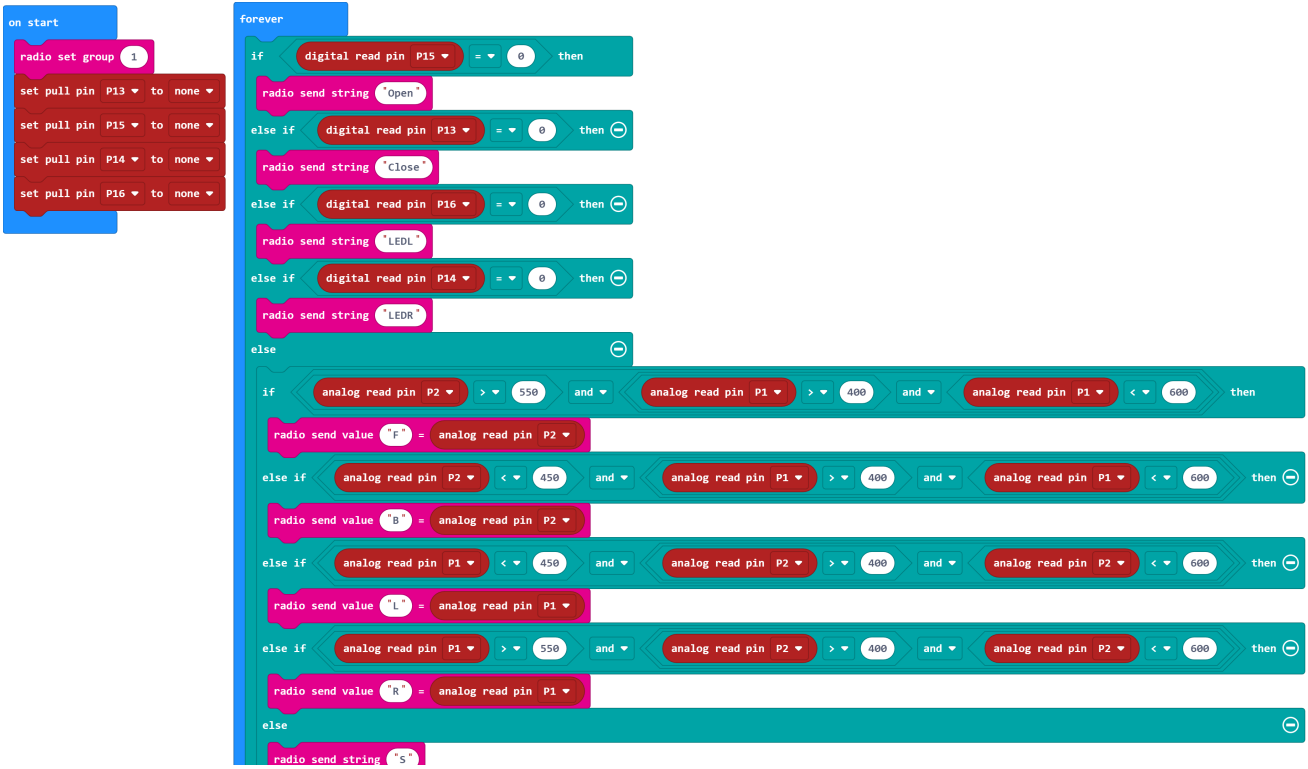


Download the corresponding programs to the microbit motherboard on the handle side and the microbit motherboard on the car side respectively. Turn on the power of the handle and the car. Move the joystick on the left side of the handle to control the forward, backward, left and right turn of the car.

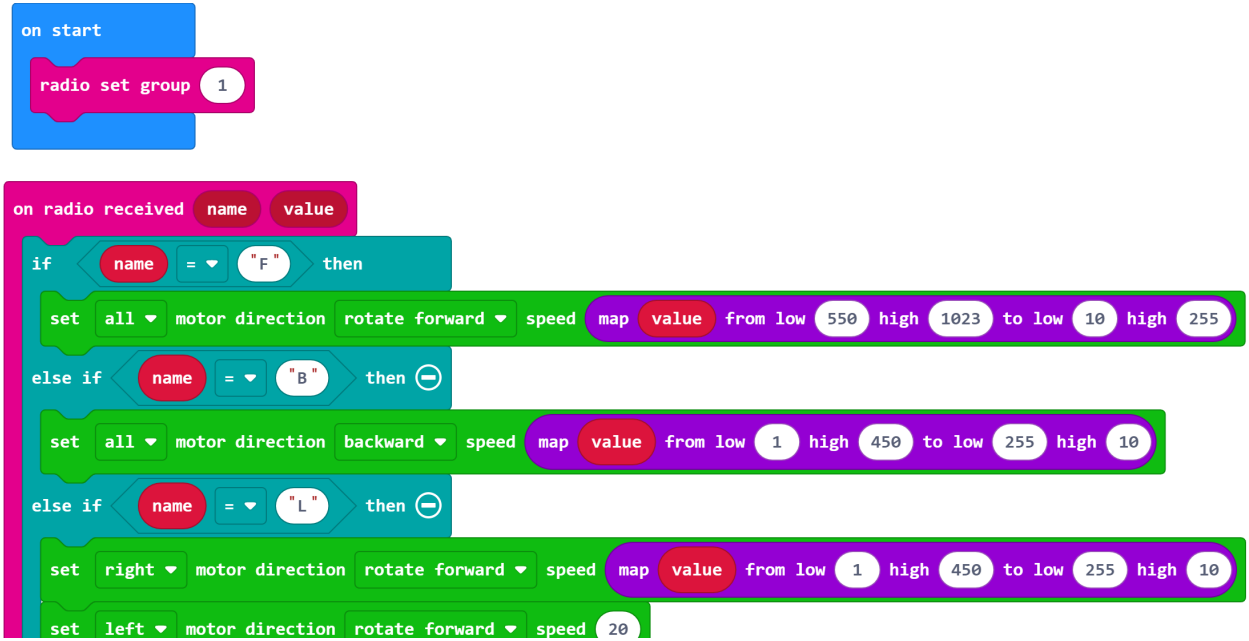
However, you may find a problem, we can only control the direction of the car, not the speed of

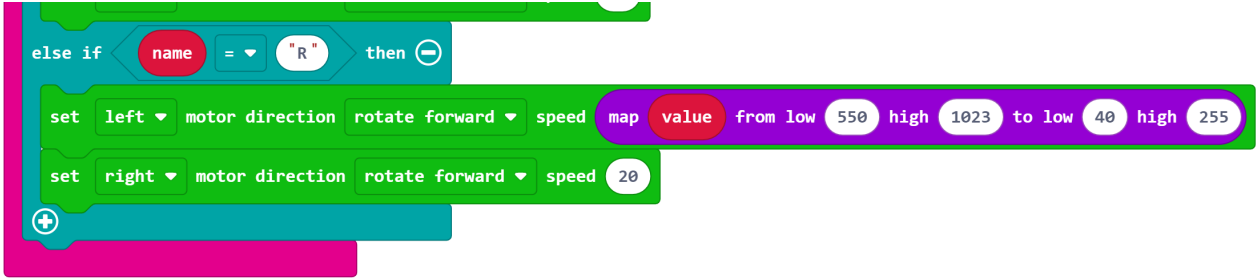
the car. Now let's update the program so that the remote control handle can control both the direction and the speed.

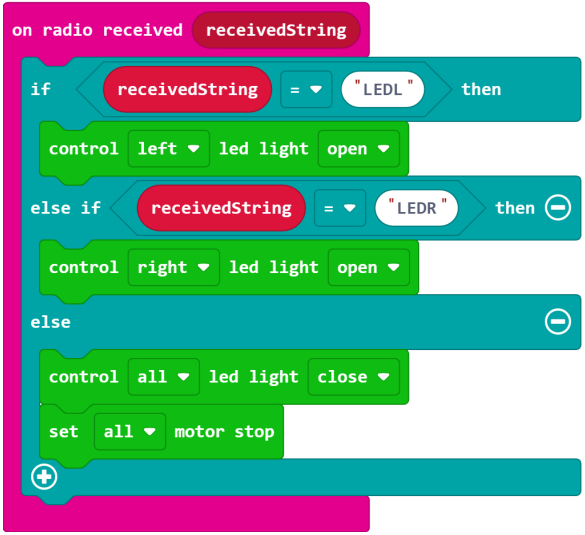
Remote Controller Program Link: https://makecode.microbit.org/_99UJEH9hahwF (https://makecode.microbit.org/_99UJEH9hahwF)


on start
radio set group 1
set pull pin P13 to none
set pull pin P15 to none
set pull pin P14 to none
set pull pin P16 to none
forever
if digital read pin P15 = 0 then
radio send string "Open"
else if digital read pin P13 = 0 then
radio send string "Close"
else if digital read pin P16 = 0 then
radio send string "LEDL"
else if digital read pin P14 = 0 then
radio send string "LEDR"
else
if analog read pin P2 > 550 and analog read pin P1 > 400 and analog read pin P1 < 600 then
radio send value "F" = analog read pin P2
else if analog read pin P2 < 450 and analog read pin P1 > 400 and analog read pin P1 < 600 then
radio send value "B" = analog read pin P2
else if analog read pin P1 < 450 and analog read pin P2 > 400 and analog read pin P2 < 600 then
radio send value "L" = analog read pin P1
else if analog read pin P1 > 550 and analog read pin P2 > 400 and analog read pin P2 < 600 then
radio send value "R" = analog read pin P1
else
radio send string "S"

Maqueen Plus Car Program Link: https://makecode.microbit.org/_evzJtbUqoLFt (https://makecode.microbit.org/_evzJtbUqoLFt)



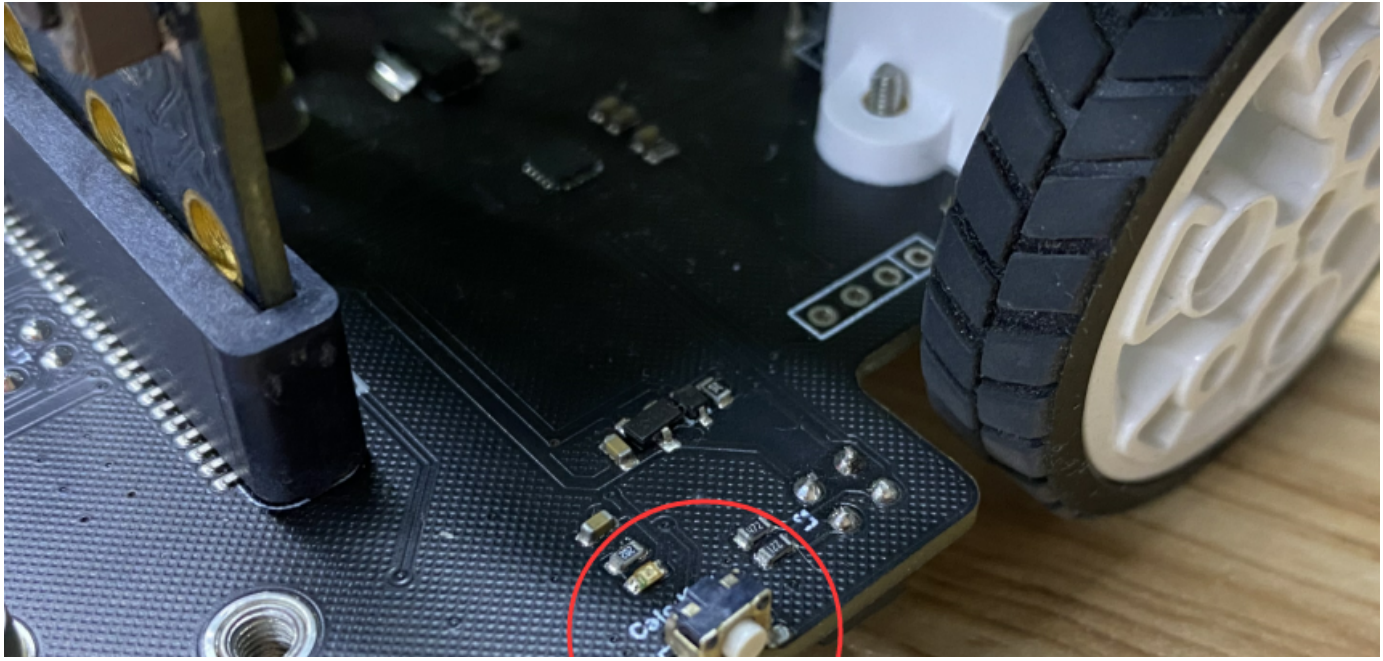




Download the corresponding programs to the micro:bit on the handle side and the microbit motherboard on the car side respectively. Turn on the power of the handle and the car. Gently

push the joystick forward, Maqueen Plus will gradually start and speed up. Try turning around and going backward. The speed in each direction is related to the angle of the joystick. You can not only control the direction of the car, but also the speed of the car.

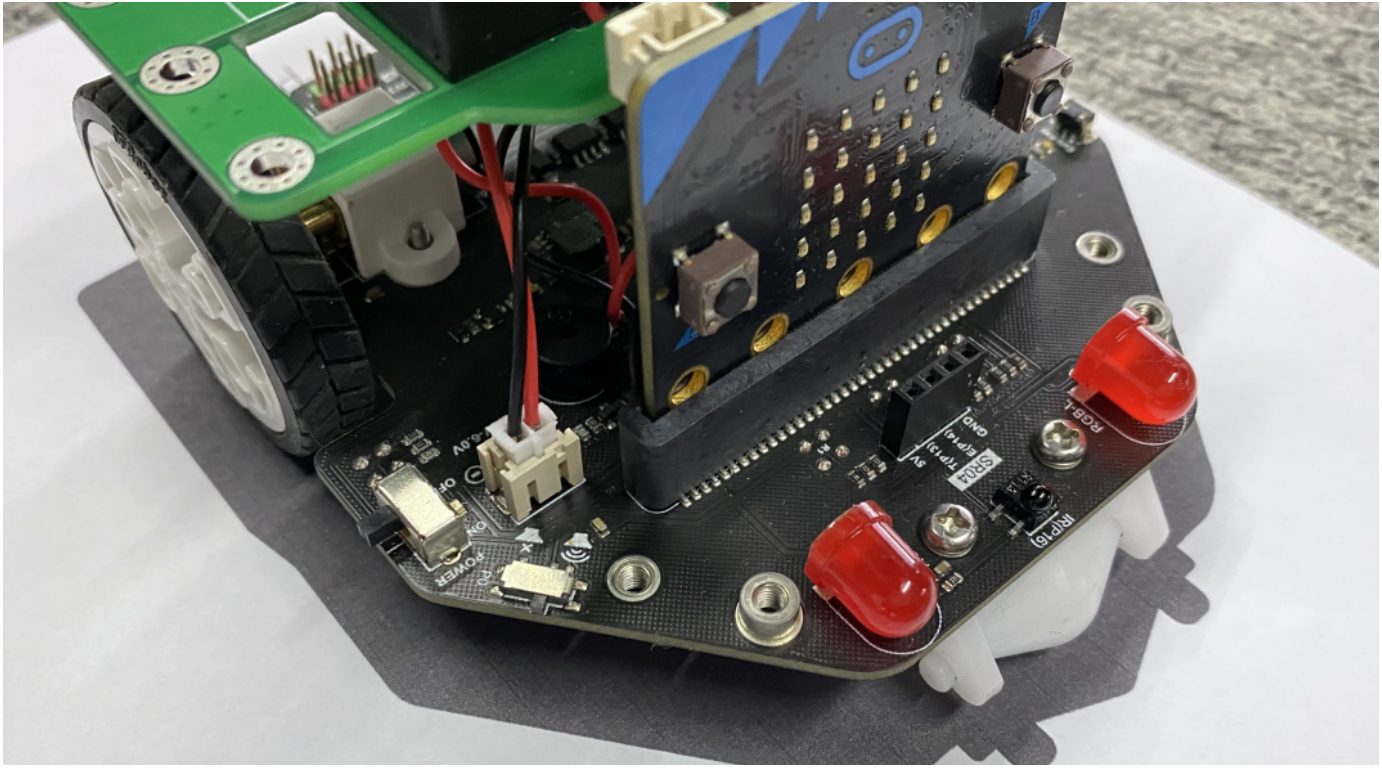
7. Link-tracking Sensor Calibration



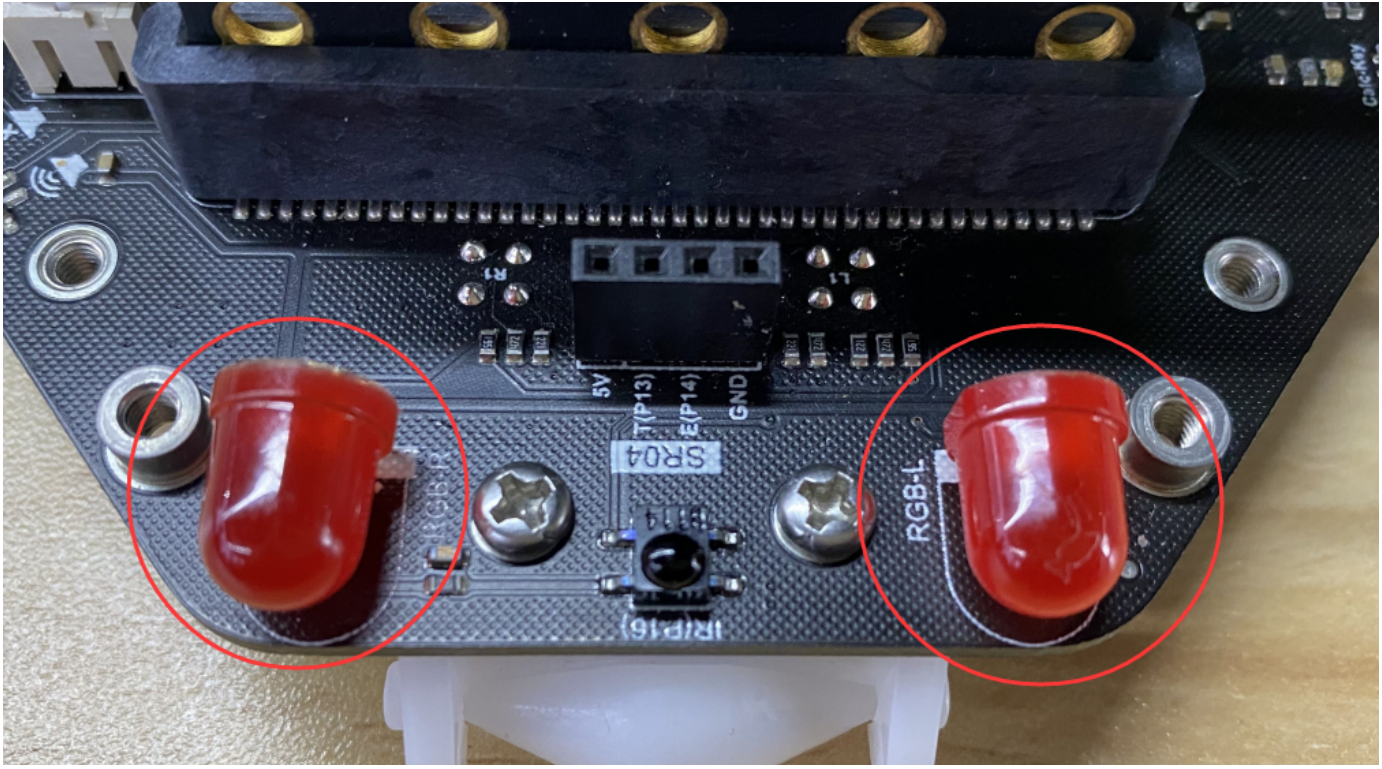
The line-tracking sensor has been calibrated before leaving the factory, and you do not need to calibrate it again. Just like the reset button of digital products, you don't need to use it in most cases. Incorrect calibration methods can also cause the line-tracking sensor to fail.

If you find that the line-tracking sensor does not recognize the black lines that can be recognized normally, then you can try to calibrate it. Methods as below:

1. Place the car on the black block of the map which delivered with the product, and make sure that the 5 sensors are in the black area. as the picture shows:


SR04
GND
5V
RGB-L
MP-16

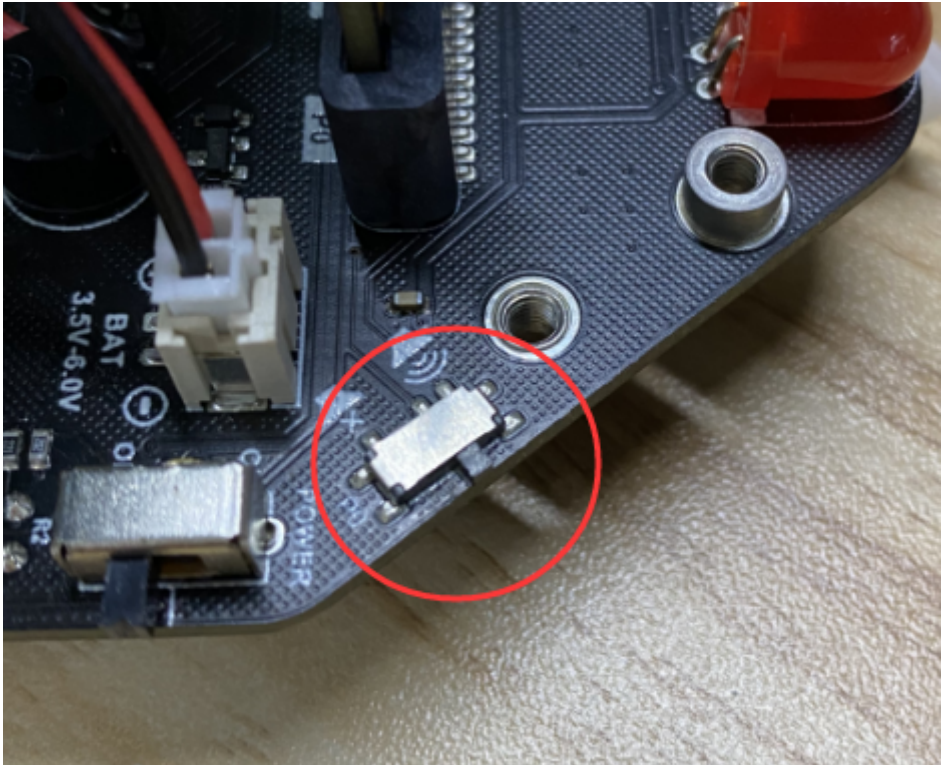
2. Long press the calibration button for about 1 second. At this time, the car lights flash and the calibration is completed.


RGB-R
RGB-L
5V
T(P13)
E(P14)
GND
SR04
IR(P16)

3. Check the calibration result: when the calibration done, place the line-tracking sensor in the black area, the line-tracking indicator is on, and the indicator is off in the white area, indicating that the calibration is correct.

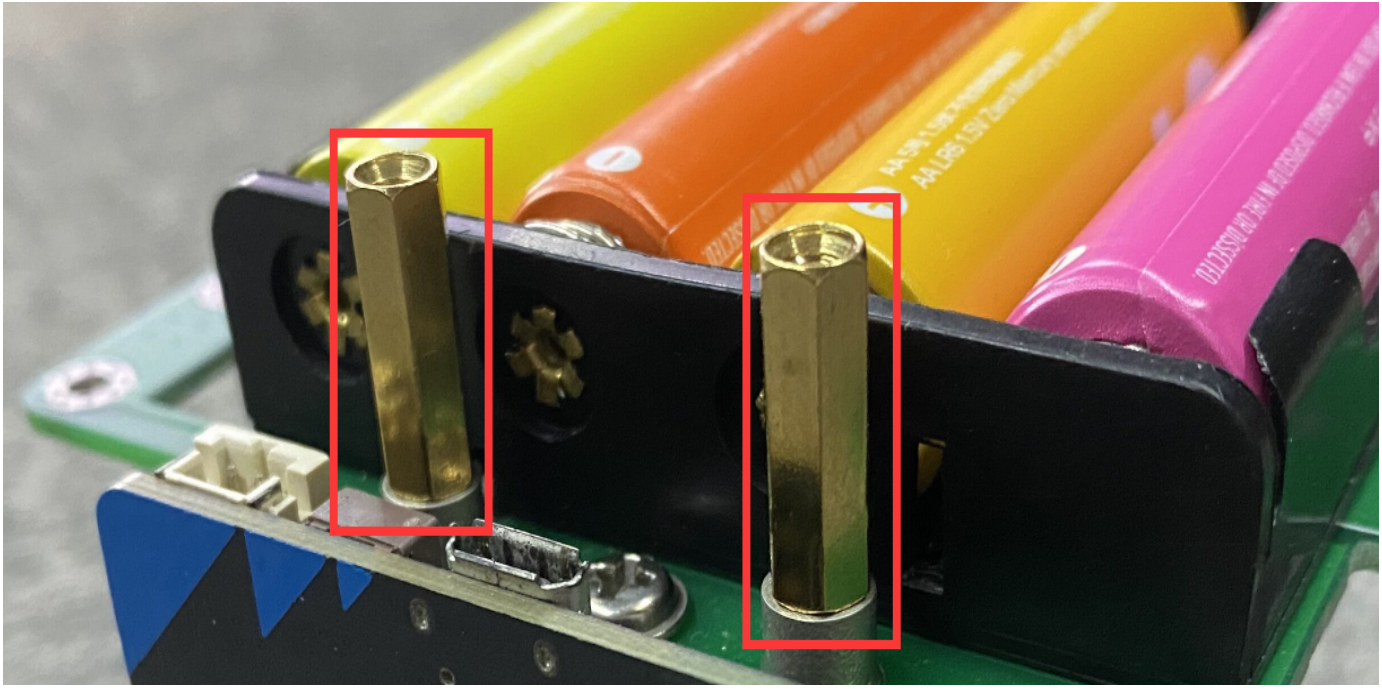
8. Buzzer Switch

When you don't need to use the buzzer, but you want to use the P0 port, or use the micro:bit V2 motherboard, you can use this switch to turn off the buzzer. The picture shows the state of turning on the buzzer. Flick to the other side to turn off the buzzer sound.

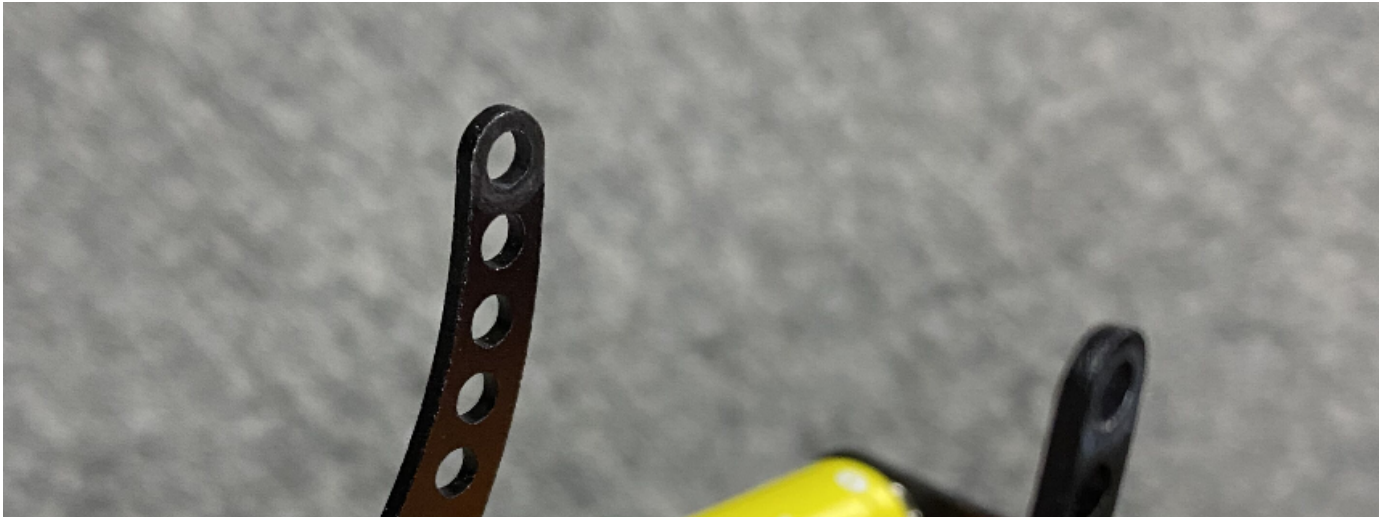

P0
BAT
3.5V-6.0V
R2
POWER

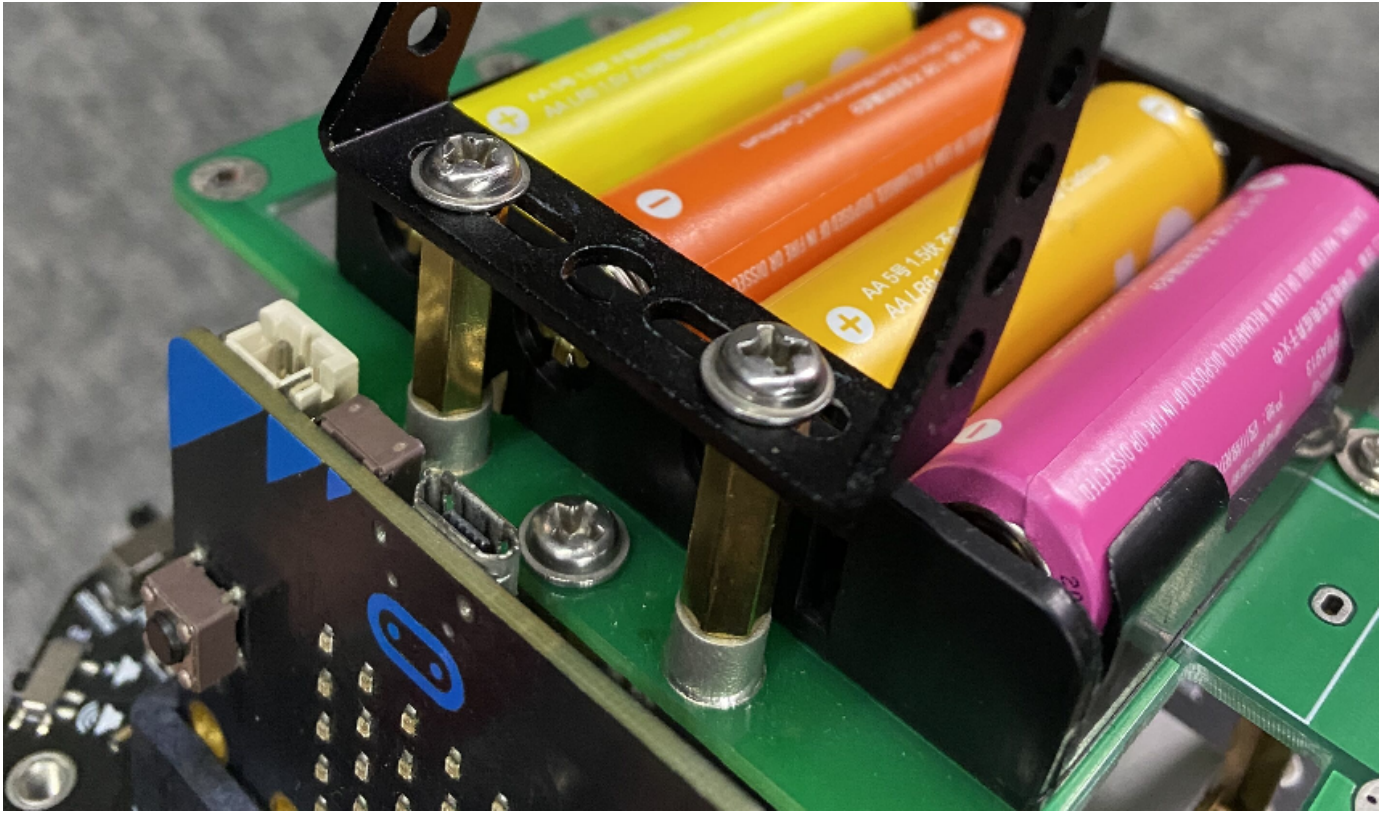
9. Install HuskyLens Camera

1. Install the two copper pillars delivered by the product in the position as shown in the figure.

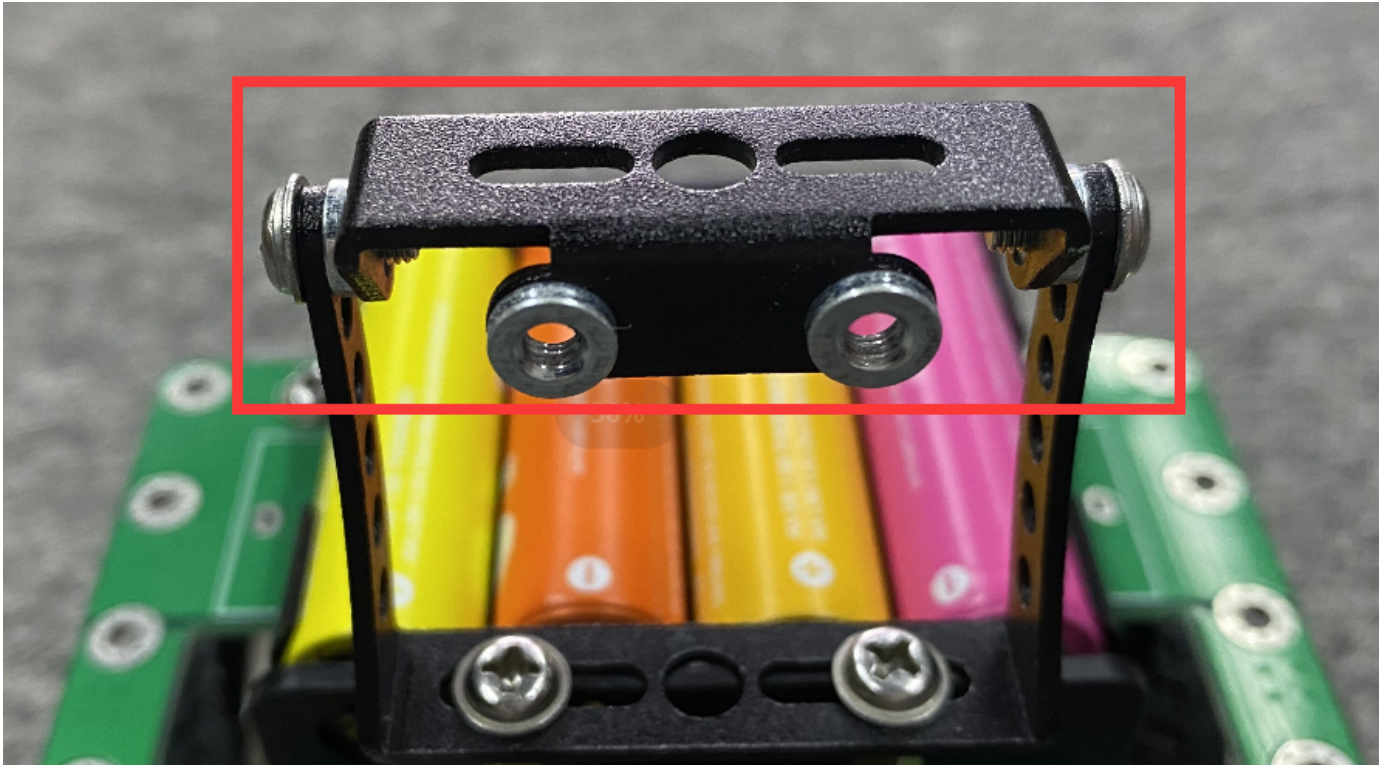


2. Fix the arc-shaped bracket (the bracket and mounting screws are provided with the HuskyLens product) on the copper column with screws.

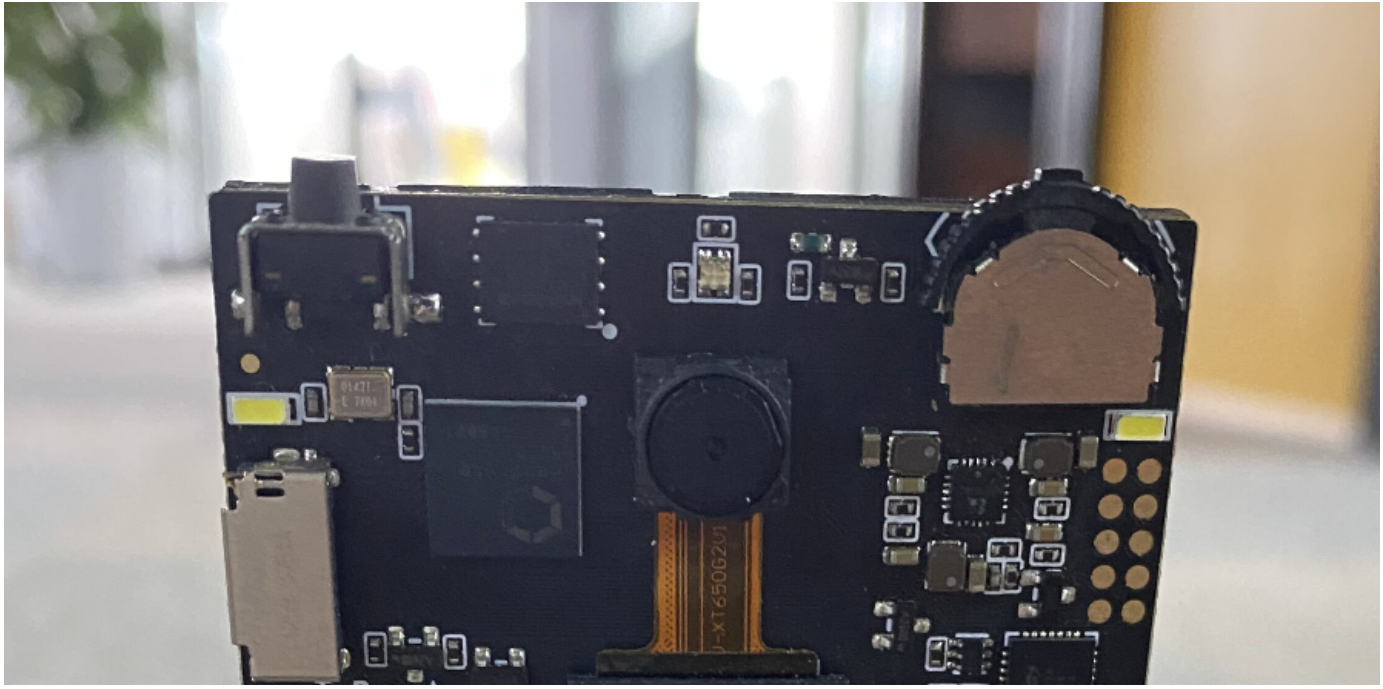


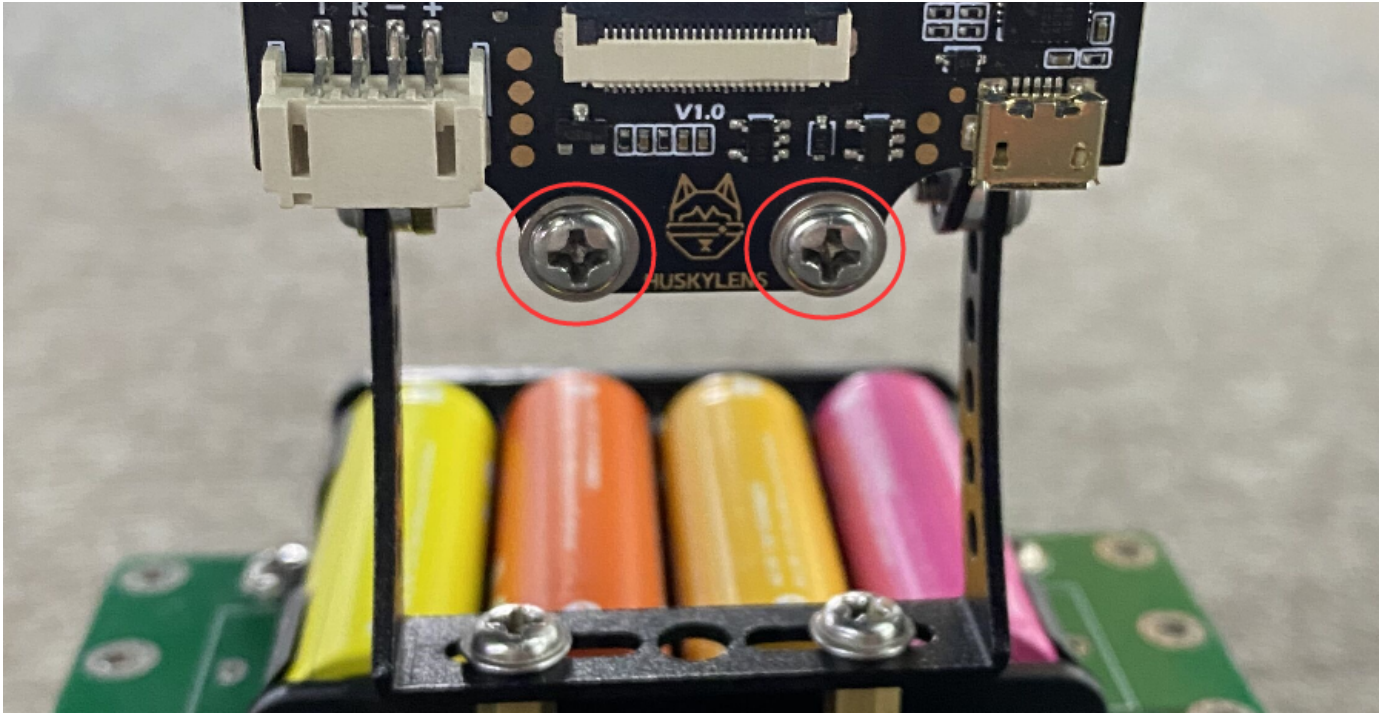


3. Install the other bracket (the bracket and mounting screws are provided with the HuskyLens product)

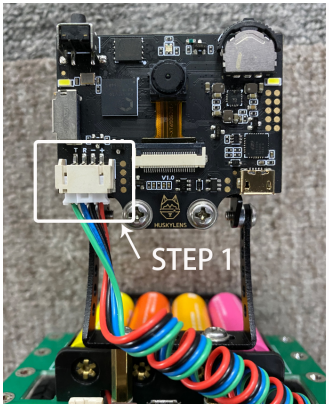


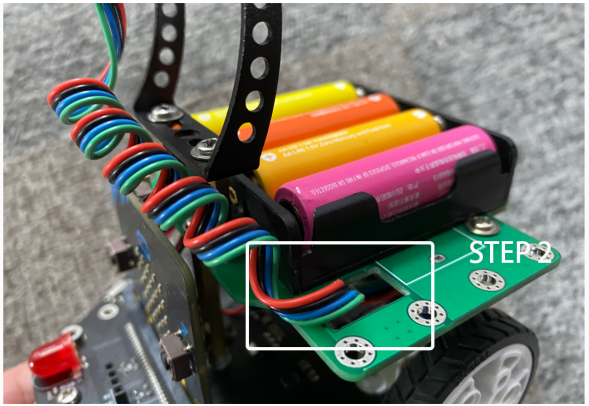
4. Install HuskyLens AI camera

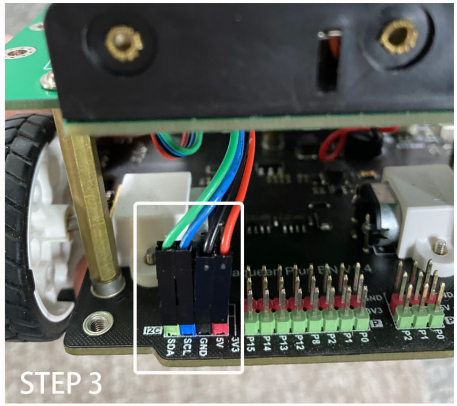




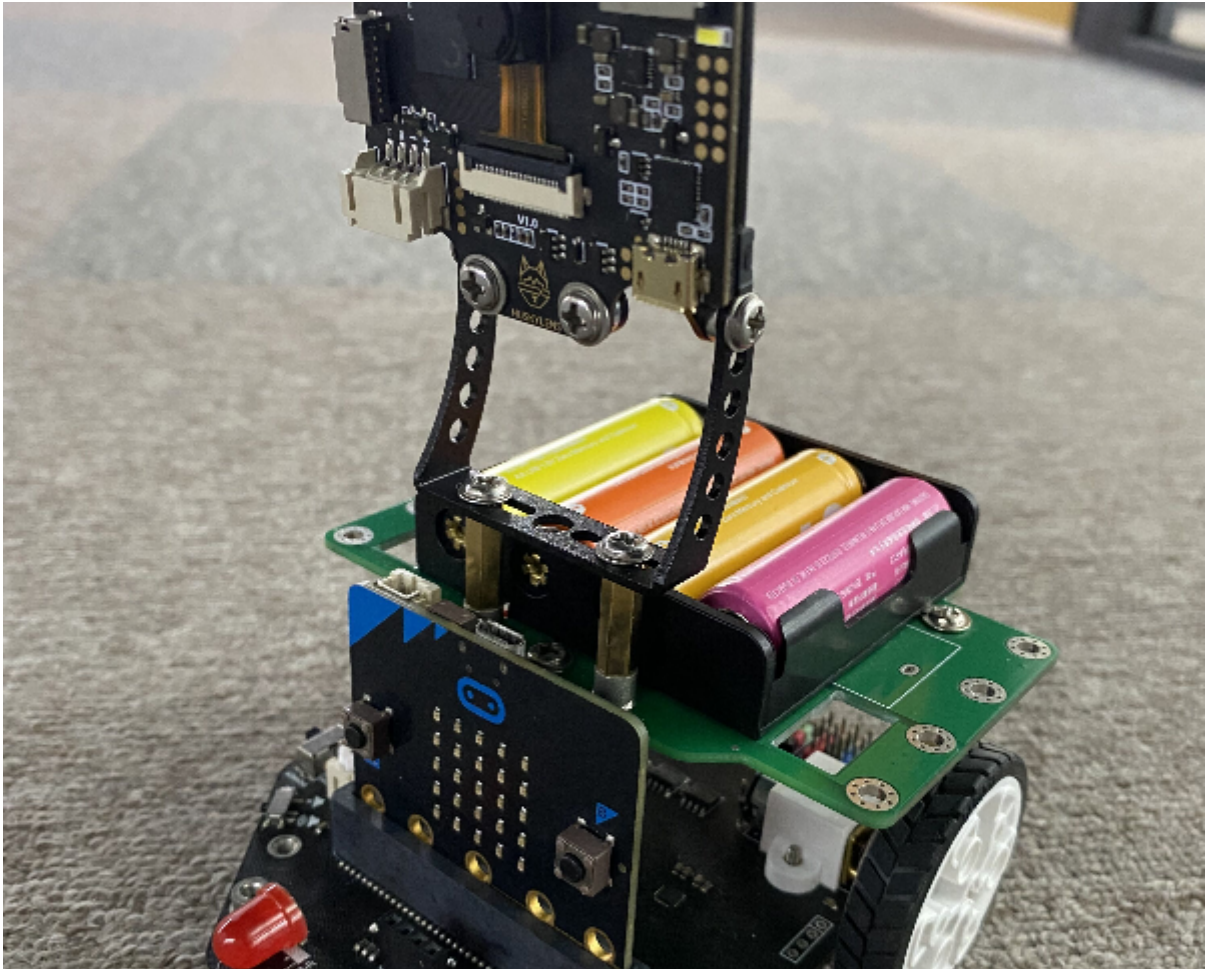
5. Plug in the AI camera connection





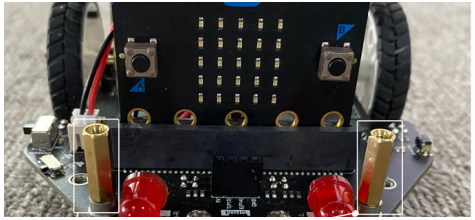
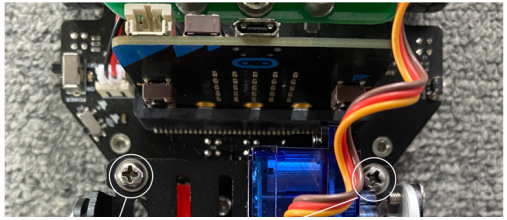
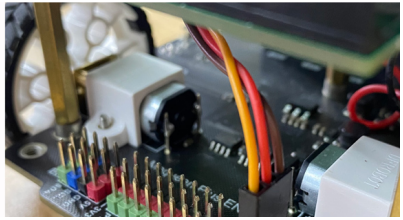


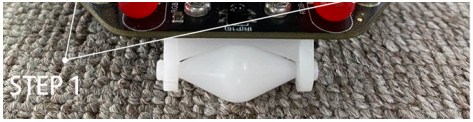
6. Installation done.

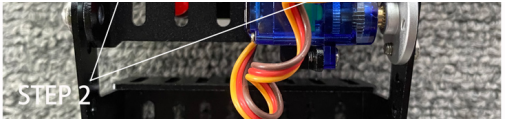


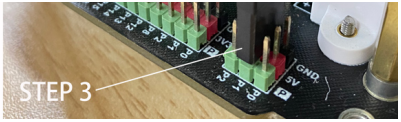
10. Install Maqueen Mechanic Kit

1. Install the delivered copper pillar at the position shown in the figure.
2. Install the assembled Maqueen Mechanic on the copper column with screws.
3. Plug the connecting wire of the servo into any 5V port of P0 or P1 or P2 on the back, and be careful not to plug it in the reverse direction.







FAQ

For any questions, advice or cool ideas to share, please visit the **DFRobot Forum** (https://www.dfrobot.com/forum/).

More Documents

Get **micro:Maqueen Plus V2** (https://www.dfrobot.com/product-2027.html) from DFRobot Store or **DFRobot Distributor**. (https://www.dfrobot.com/index.php?route=information/distributorslogo)

Turn to the Top