

Bluguard Maqueen Plus

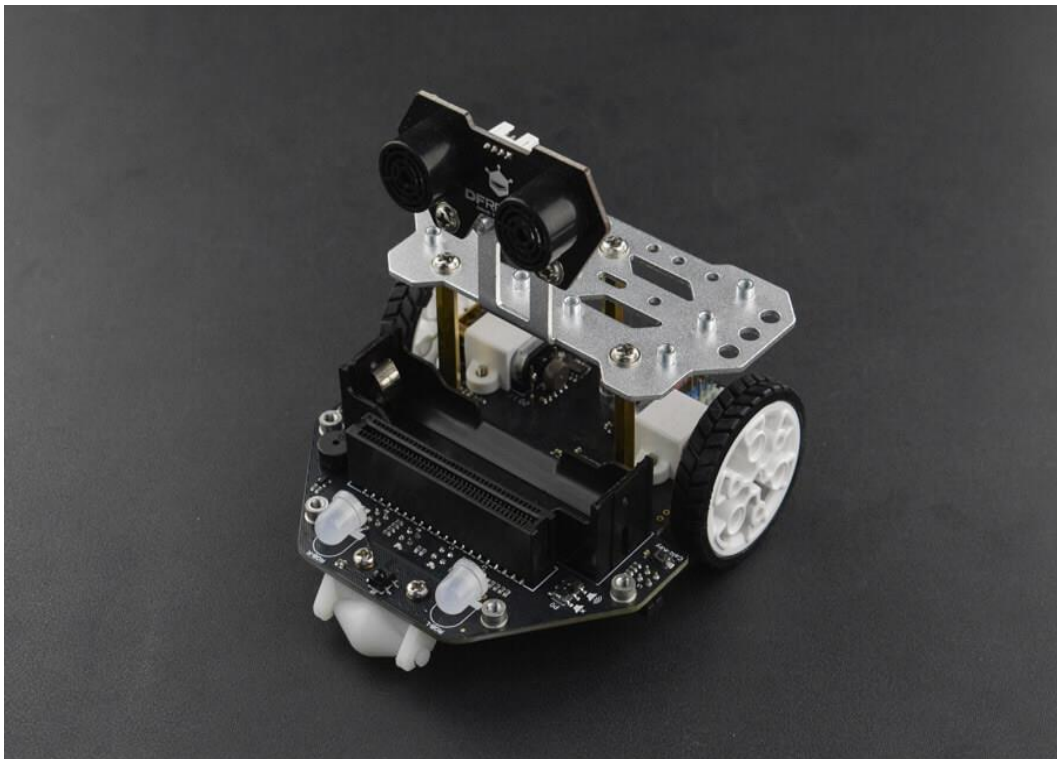


TABLE OF CONTENTS

Chapter 1: Introduction to Bluguard Maqueen Plus.....	3
Chapter 2: Step to Calibrate Line Tracking Sensor	7
Chapter 3: Import the MakeCode Graphical Library	9
Chapter 4: Line Follower Moving Along Circle	11
Chapter 5: Line Follower Moving Along Cross Line.....	15
Chapter 6: Obstacle Avoidance Robot.....	20
Chapter 7: Car Park Helper	26
Chapter 8: PID Control for Bluguard Maqueen Plus	31
Chapter 9: Play Music with RGB Light Blinking	32
Chapter 10: Light Chaser	34
Chapter 11: Motion Sensing Racing Car	36

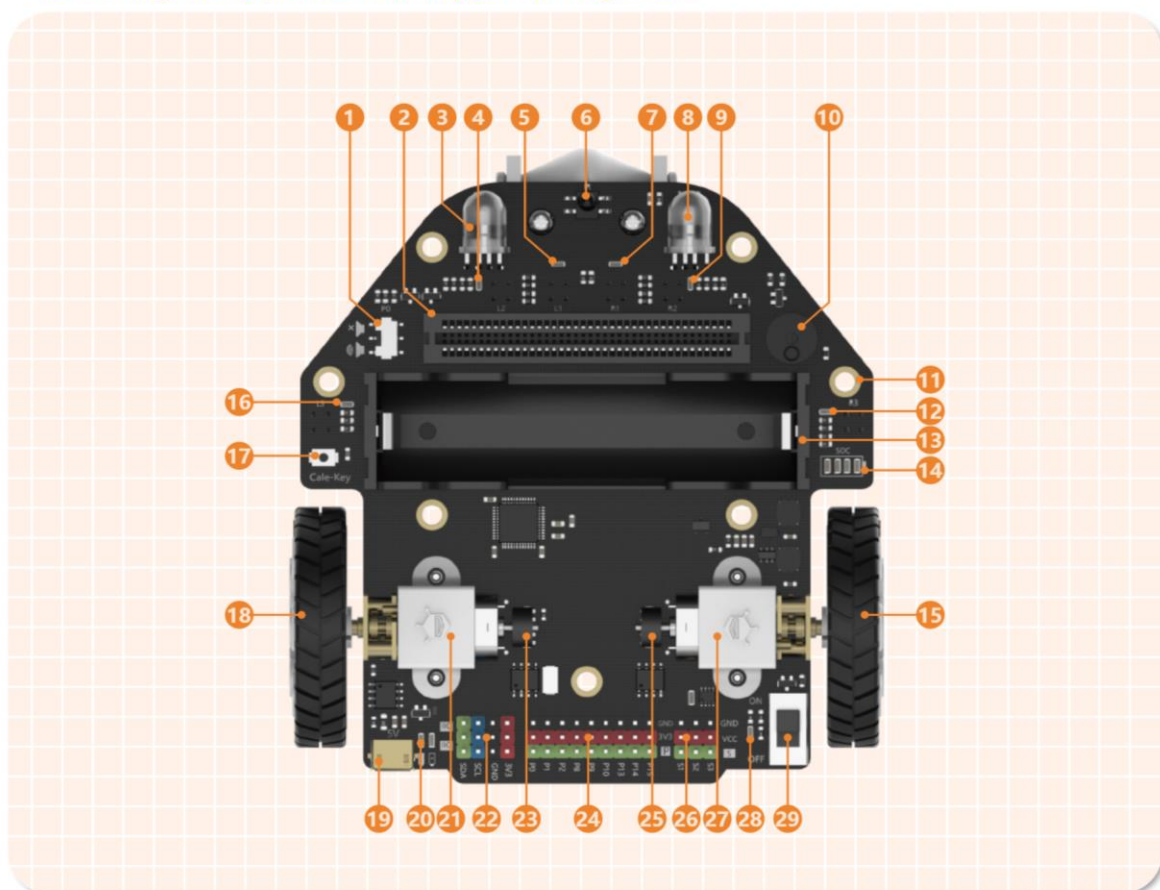
Chapter 1: Introduction to Bluguard Maqueen Plus

Bluguard Maqueen Plus is a wonderful educational robot customized for young children. It can be programmable via Mind+ and MakeCode platforms, allowing users to execute awesome functions by simply dragging and placing the graphical blocks. It is equipped with a rechargeable Lithium battery 18650.

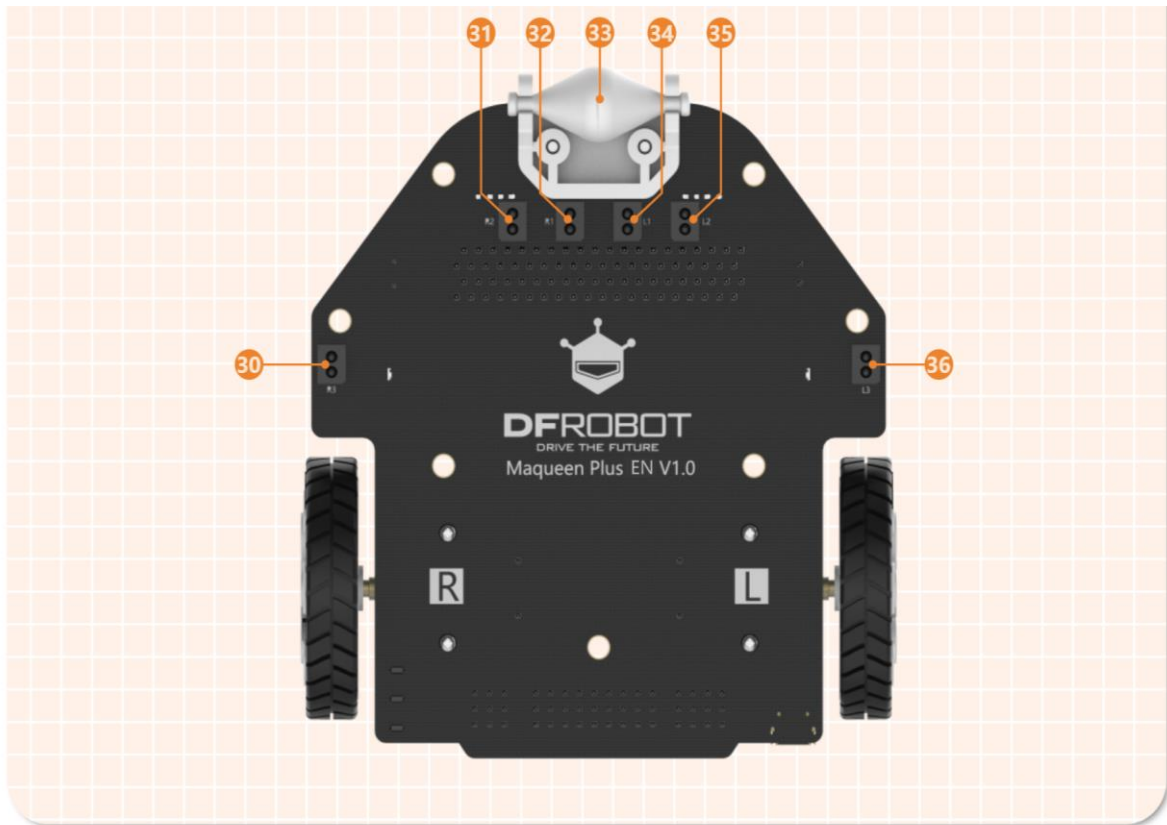
Suggest Age: 8 +

Adult supervision is recommended for children under 8 years old.

Before we get started, let's see what Maqueen Plus has got there.



- | | | | | |
|----------------------|----------------------------------|-------------------------------|------------------------------|---------------------------|
| 1 Buzzer switch | 2 micro:bit socket | 3 RGB-LED-L | 4 L2 indicator LED | 5 L1 indicator LED |
| 6 Infrared receiver | 7 R1 indicator LED | 8 RGB-LED-R | 9 R2 indicator LED | 10 Buzzer |
| 11 M3 Mounting holes | 12 R3 indicator LED | 13 Battery case | 14 Electricity indicator LED | 15 Right wheel |
| 16 L3 indicator LED | 17 Line-tracking Calibration Key | 18 Left wheel | 19 Charging port | 20 Charging indicator LED |
| 21 Motor-L | 22 IIC expansion port | 23 Encoder-L | 24 GPIO port | 25 Encoder-R |
| 26 Servo port | 27 Motor-R | 28 Power supply indicator LED | 29 Power supply switch | |



30 R3 line-tracking sensor

31 R2 line-tracking sensor

32 R1 line-tracking sensor

33 Support wheel

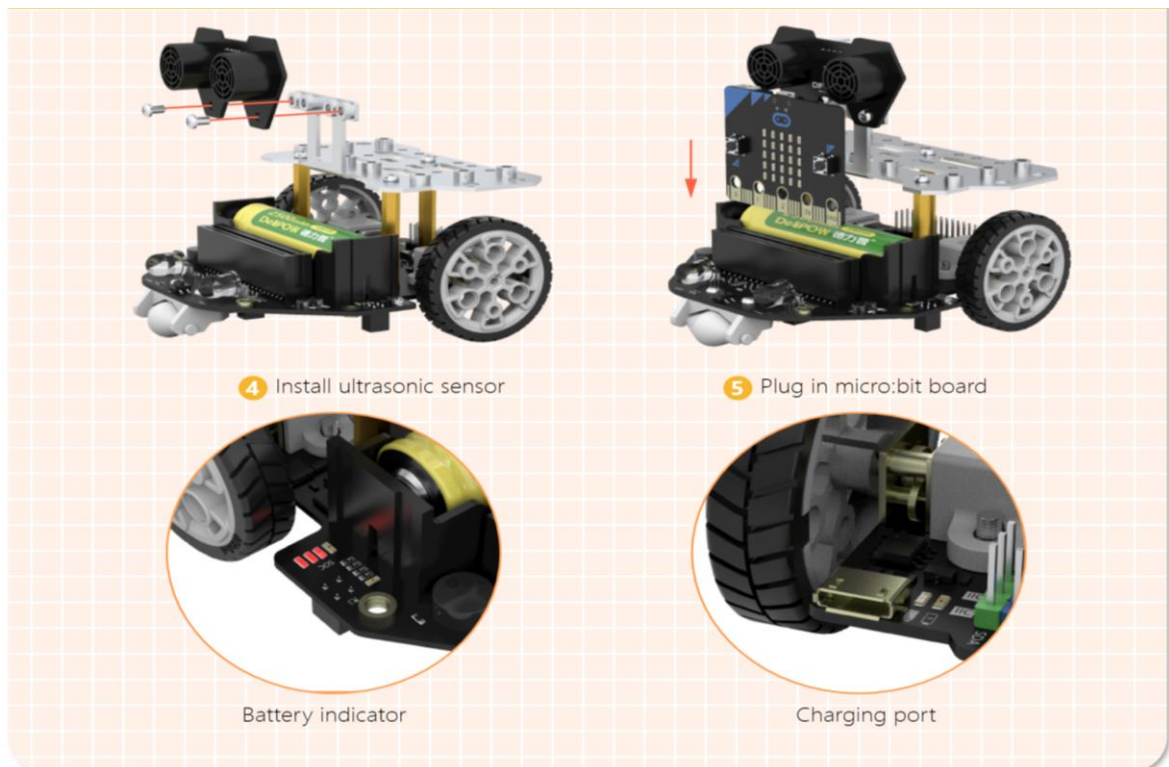
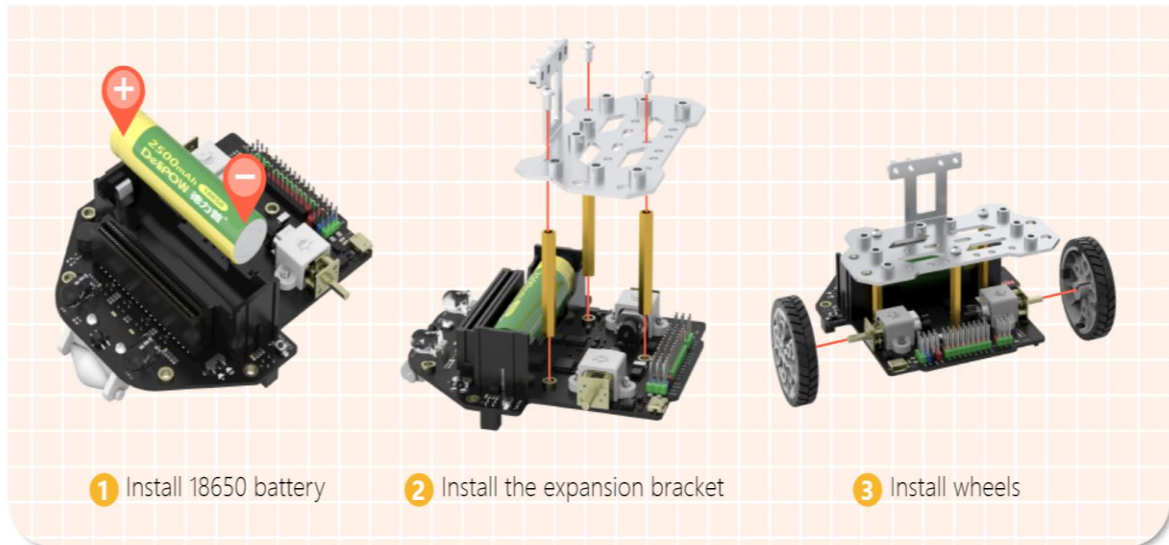
34 L1 line-tracking sensor

35 L2 line-tracking sensor

36 L3 line-tracking sensor

Bluguard Maqueen Plus Assembly Guide

Note: Bluguard Maqueen Plus is equipped with a lithium battery. Make sure the correct polarity when installing a battery and avoid short circuit the battery's terminals.



Note: when the battery is fully charged, all LEDs will be on. The LEDs will be off one by one as the power gradually decreases. If all lights go out, the battery needs to be recharged.

Specification:

1. Power Supply: 3.7V 18650 lithium battery
2. Charging Voltage: 5V Charging Current: 900mA
3. Charging Time: about 4hours
4. Power Indicator: 4 LEDs
5. Motor Specification: N20 motor 260 R/M
6. Buzzer x1
7. RGB Light x2
8. GPIO Expansion Port: P0 P1 P2 P8 P12 P13 P14 P15 P16
9. I2C Port: x3
10. Servo Expansion Port: x3
11. Line-tracking Sensor x6
12. Line-tracking Sensor Output: digital +analog
13. Support Calibration for Line-tracking Sensor
14. IR Receiving Sensor x1 Ultrasonic Sensor: URM10
15. Top Metal Plate: x1
16. M3 Threaded Connections x12
17. Map Size: 50cmx50cm
18. Product Dimension: 107x100mm/4.21 x3.94"

Packing List:

- Maqueen Plus x 1
- Ultrasonic Sensor x 1
- Top Metal Plate x 1
- Wheel x 2
- Copper Pillar x 3
- Instructions x 1
- Line-tracking Map x 1

Note: micro:bit and the 18650 lithium battery are not included

Chapter 2: Step to Calibrate Line Tracking Sensor

Bluguard Maqueen Plus is equipped with 6 line-tracking sensors and come with dedicated indicators. The designated indicator will light up in the event of a line-tracking sensor detecting a black line. You may perform the following calibration if your line-tracking sensor is not sensitive to a black line.

1. Place the Bluguard Maqueen Plus into the calibration area of the line-tracking map, and switch on its power.



2. Press "Calc-key" for about 1 second, the 2 front large LEDs will flash in green. Upon release of the key, the calibration is completed.



3. If all the line-tracking sensor indicators are turned on in the black area and turned off in the white area, the calibration is completed.

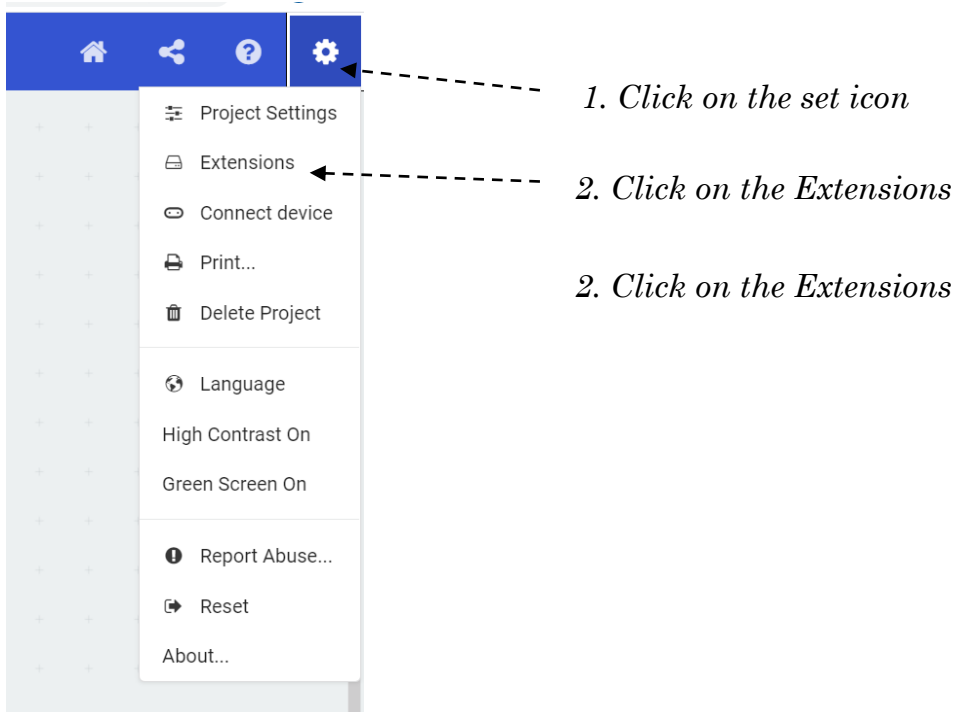
Note:

1. The internal chip will automatically save the calibration configuration. Thus, users do not need to calibrate it every time.

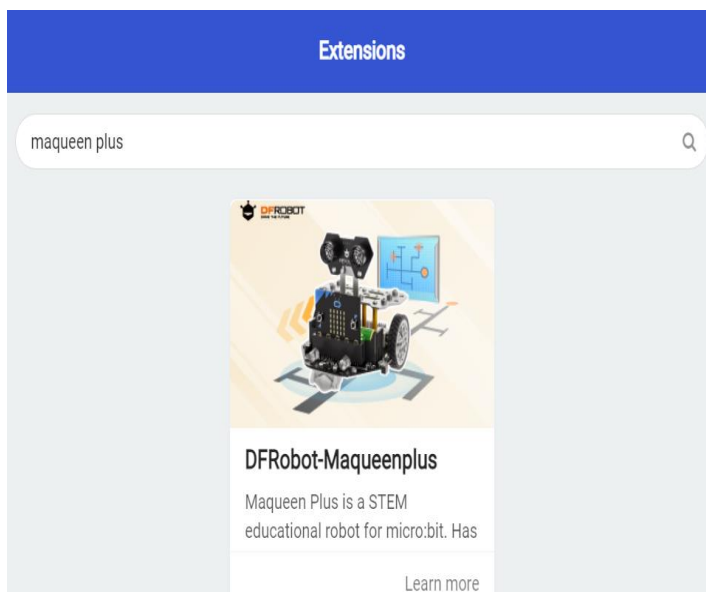
2. Bluguard Maqueen Plus has been factory calibrated, and it can be used directly normally.

Chapter 3: Import the MakeCode Graphical Library

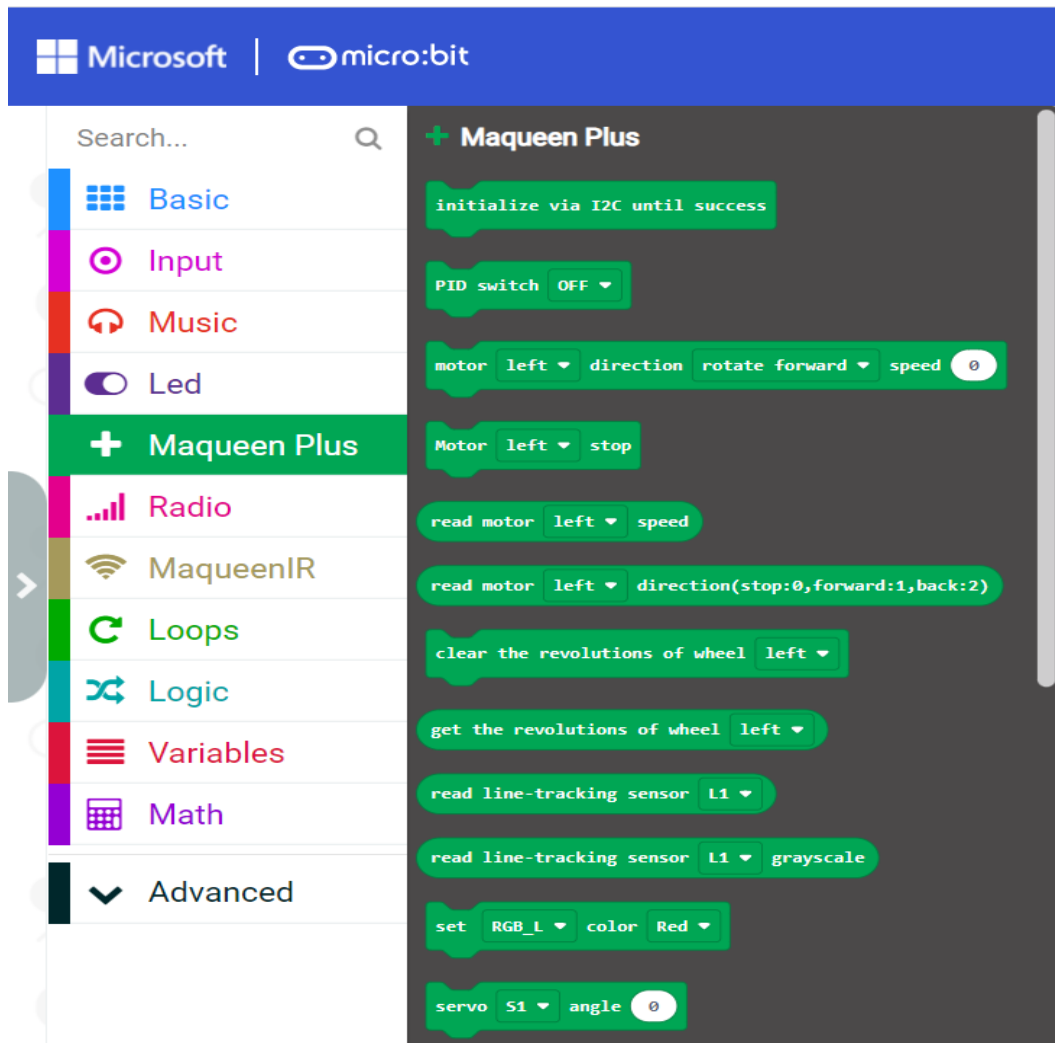
1. Click the link <https://makecode.microbit.org>, enter the makecode graphical online programming platform and create **New Project**. (Note: Loading will be slow the first time, please wait patiently)
2. Import the extensions.



3. Click on Search and type the 'Maqueen plus'. Click on the Maqueen plus's library after searching out.

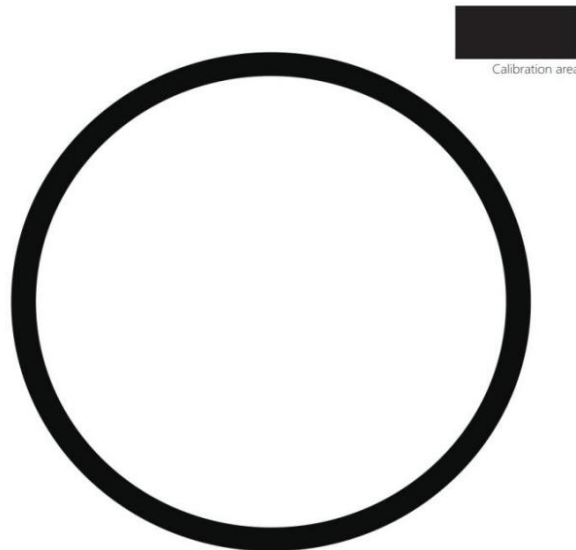


4. Import completed.



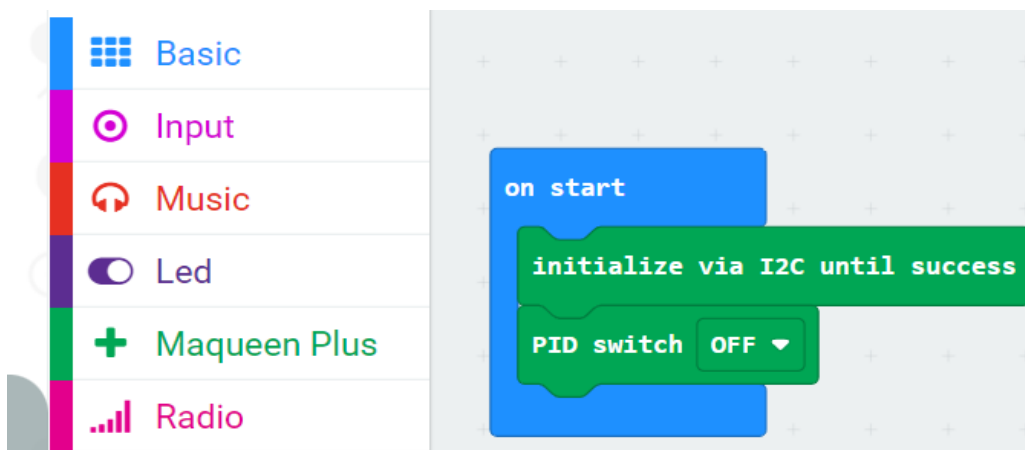
Chapter 4: Line Follower Moving Along Circle

You may operate and turn on the Bluguard Maqueen Plus into a line follower and program it to move along the circle track on the map.

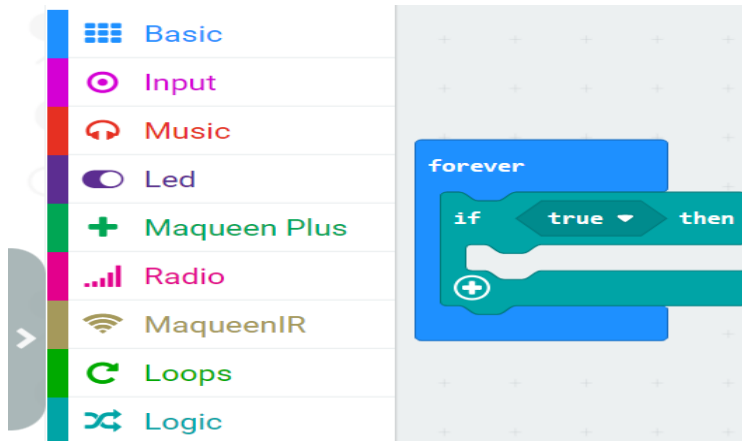


Step of Makecode Graphical Program:

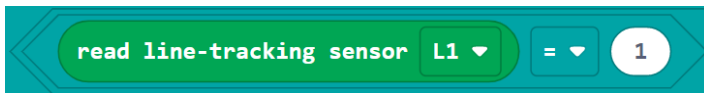
1. Add and drag the block 'initialize via I2C until success' (Maqueen Plus) and 'PID switch [OFF]' (Maqueen Plus) to the block 'on start' (Basic).



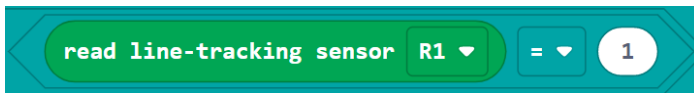
2. Add the block 'forever' (Basic) and block 'if...then' (Logic).



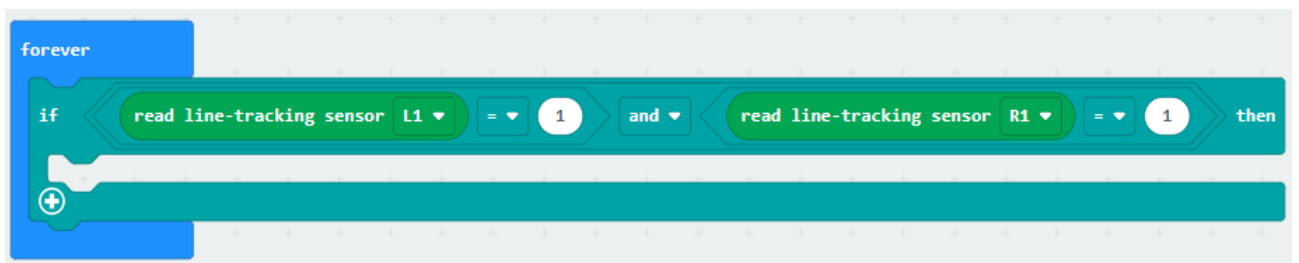
3. Add the block '[0] = [0]' (Logic) and 'read line-tracking sensor [L1]' (Maqueen Plus). Replace the value '[0]' with 'read line-tracking sensor [L1]' and change another value '[0]' to '[1]'.



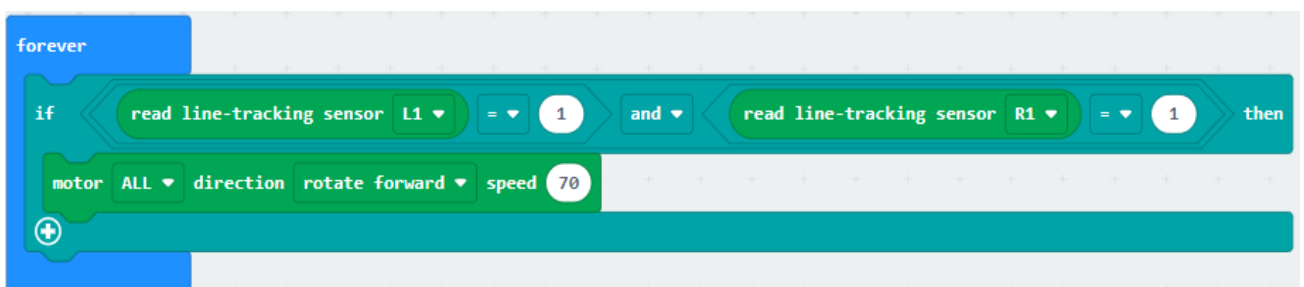
4. Duplicate the step 3 block and change '[L1]' to '[R1]'.



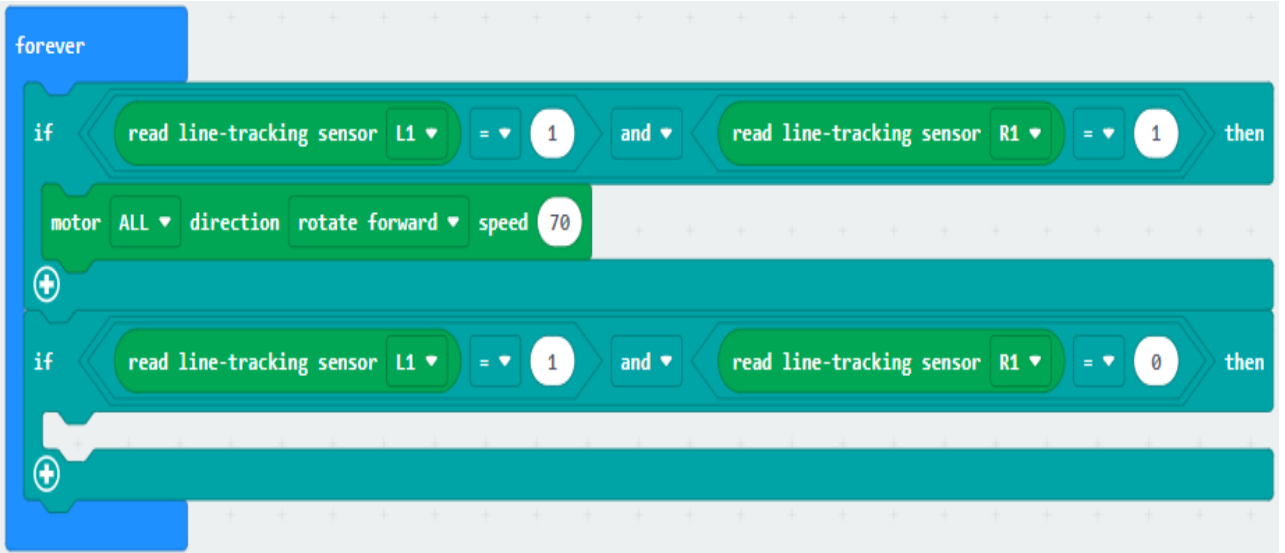
5. For condition 'if', add the block '[] and []' (Logic). Slot in '[block step 3] and [block step 4]'.



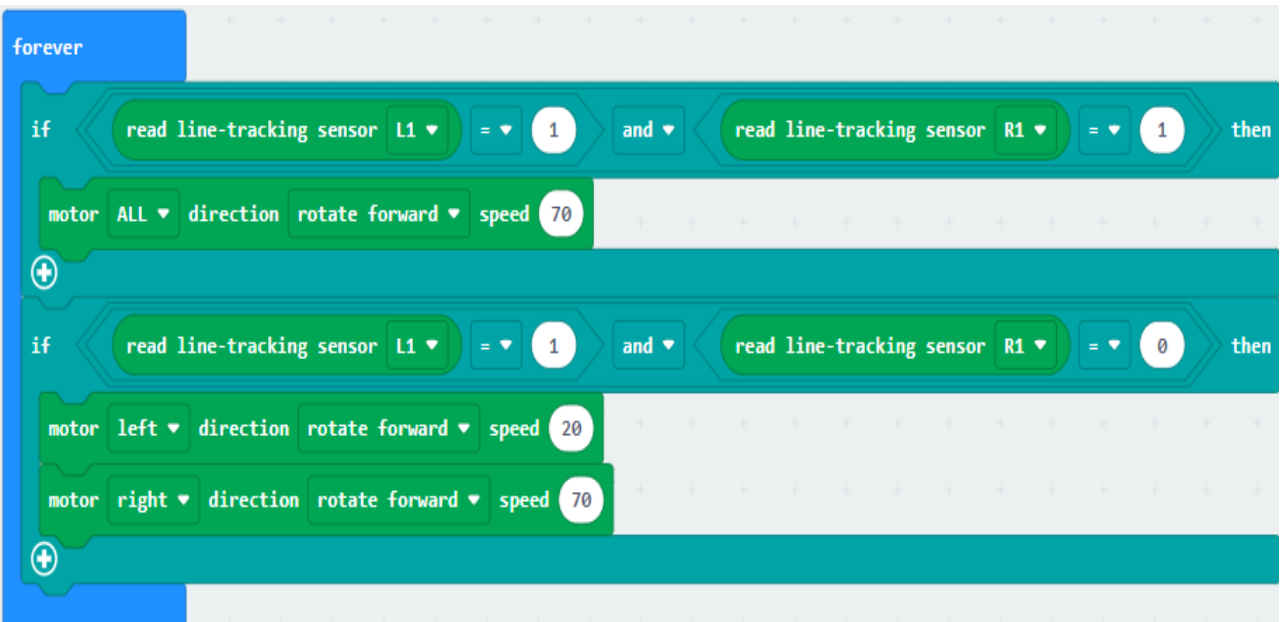
6. For condition 'then', add the block 'motor [ALL] direction [rotate forward] speed [70]' (Maqueen Plus).



7. Add the 2nd block 'if...then' (Logic). Duplicate block of 1st condition 'if' '[----] and [-- --]' and change 'read line-tracking sensor [R1] = [0]'



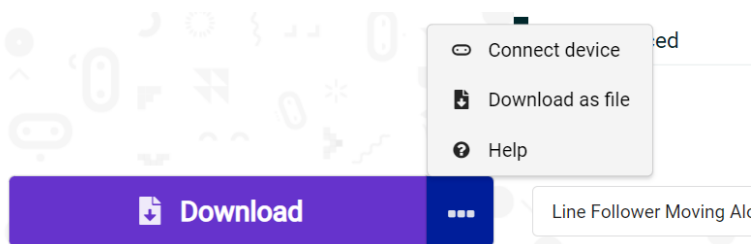
8. For condition 'then', add the block 'motor [left] direction [rotate forward] speed [20]' (Maqueen Plus) and 'motor [right] direction [rotate forward] speed [70]' (Maqueen Plus).



- Duplicate block of 2nd condition 'if...then'. For condition 'if', change value 'read...[L1]=[0]' and 'read...[R1]=[1]'. For condition 'then', change 'motor [left]...[rotate forward]...[70]' and 'motor [right]...[rotate forward]...[20]'.



- Go to 'connect device' after connecting micro:bit with cable. Just follow instructions and this step is just a one-time setup.



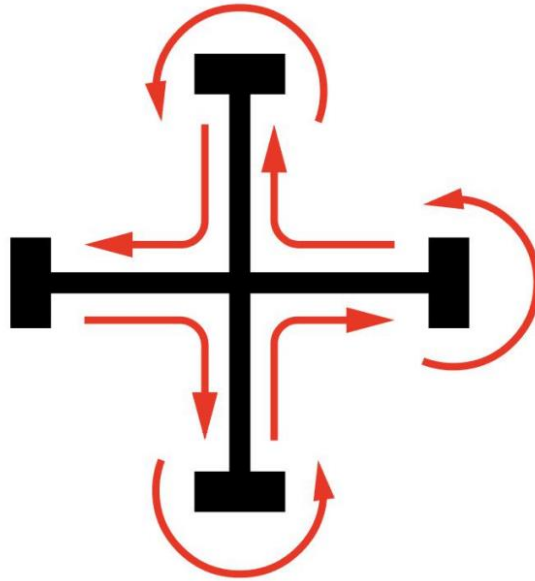
- Click **Download** to transfer your code to micro:bit.

Program Link :

https://makecode.microbit.org/_g2f9aUAmU6sw

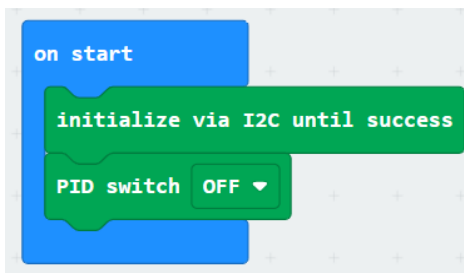
Chapter 5: Line Follower Moving Along Cross Line

In this chapter, you may program Maqueen Plus to drive along the cross line on the track map. 4 line-tracking sensors will be used in this chapter.

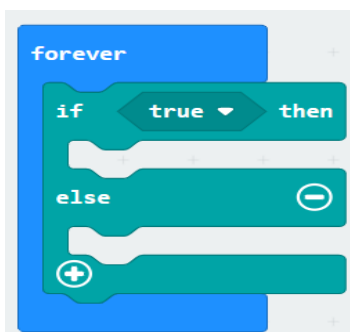


Step of Makecode Graphical Program:

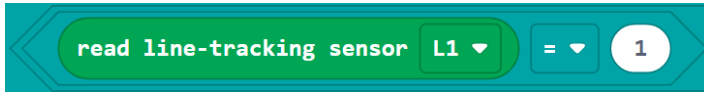
1. Add and drag the block 'initialize via I2C until success' (Maqueen Plus) and 'PID switch [OFF]' (Maqueen Plus) to the block 'on start' (Basic).



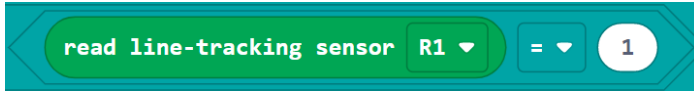
2. Add and slot the block 'if...then...else' (Logic) into block 'forever' (Basic).



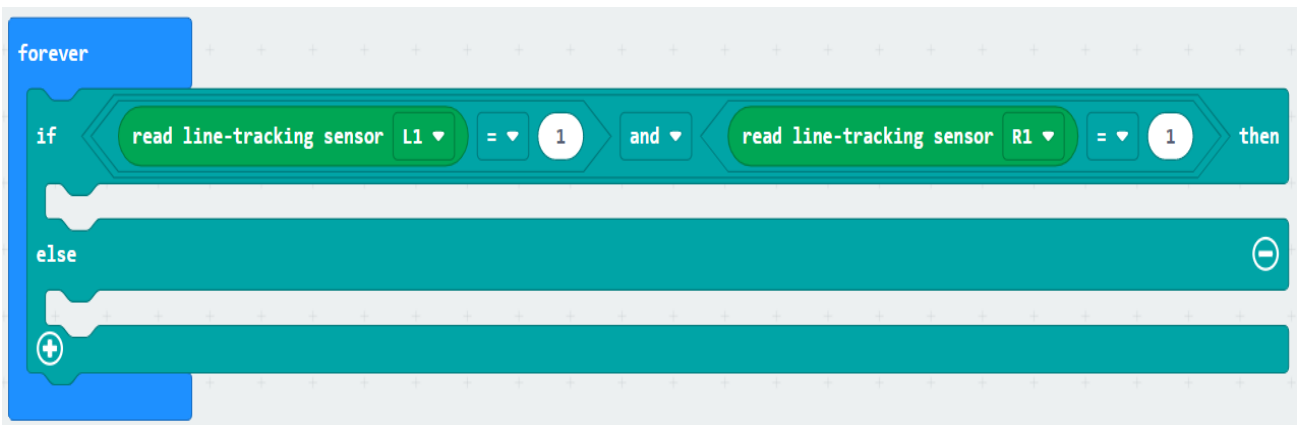
3. Add the block '`[0] = [0]`' (Logic) and '`read line-tracking sensor [L1]`' (Maqueen Plus). Replace the value '`[0]`' with '`read line-tracking sensor [L1]`' and change another value '`[0]`' to '`[1]`'.



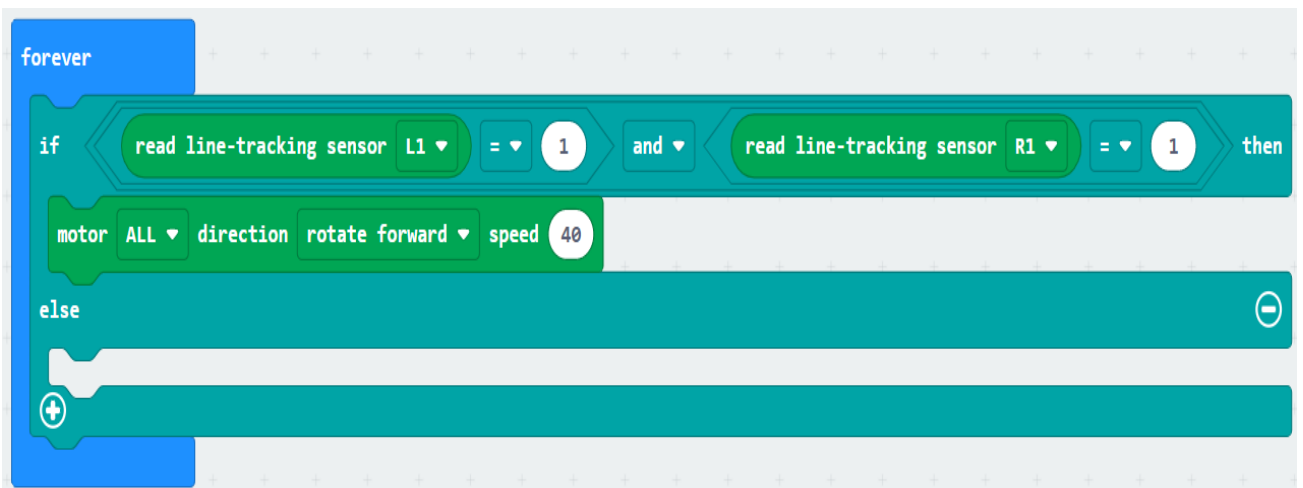
4. Duplicate the step 3 block and change '`[L1]`' to '`[R1]`'.



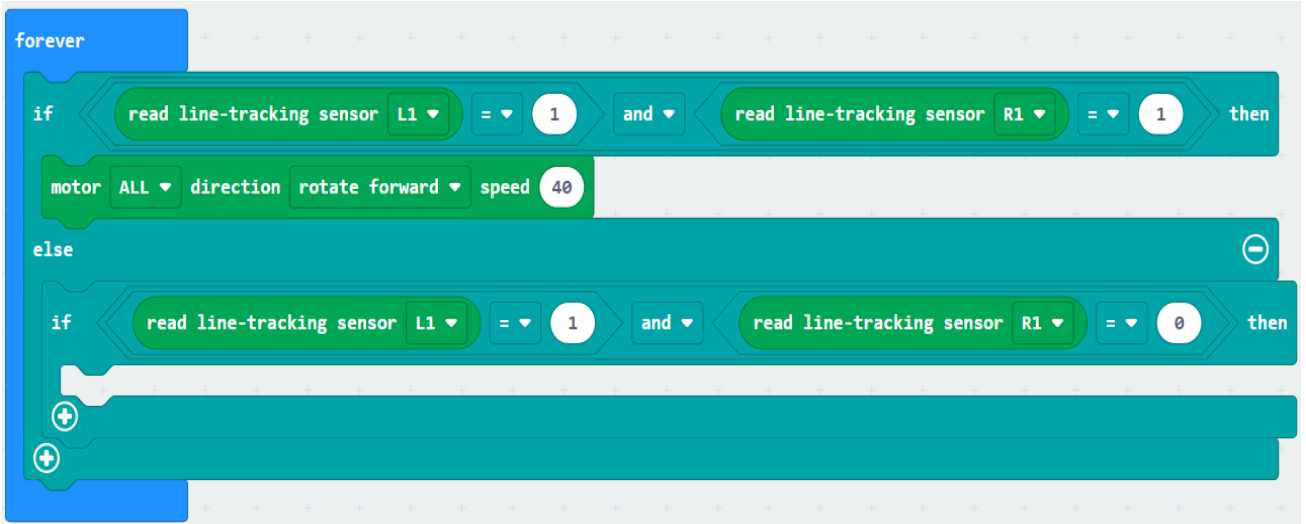
5. For condition 'if', add the block '`[] and []`' (Logic). Slot in '`[block step 3]` and '`[block step 4]`'.



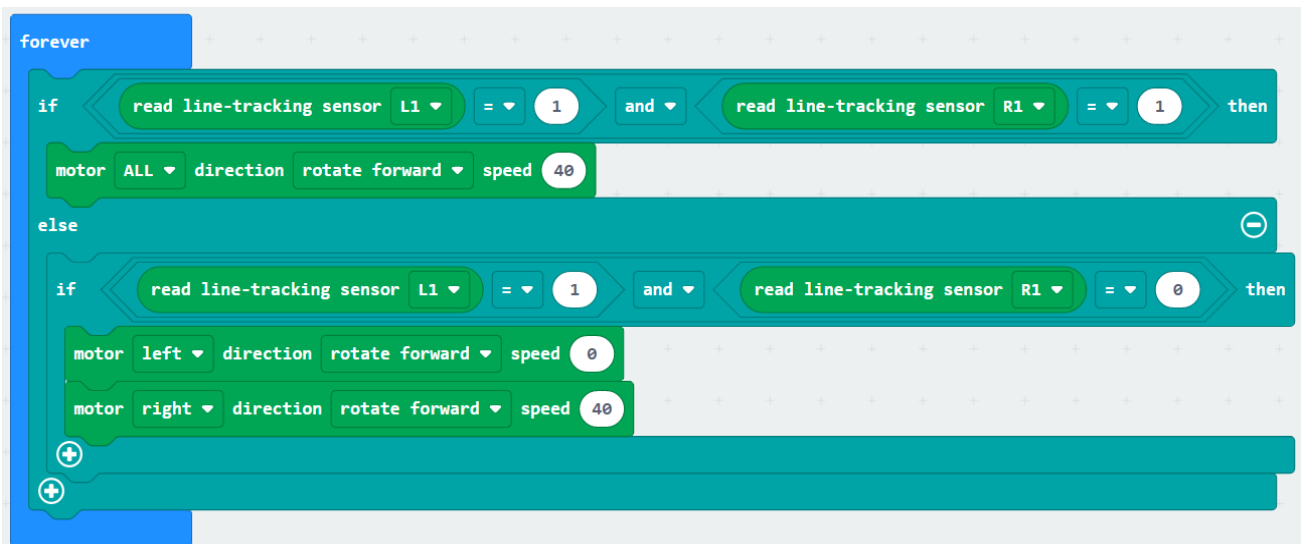
6. For condition 'then', add the block '`motor [ALL] direction [rotate forward] speed [40]`' (Maqueen Plus).



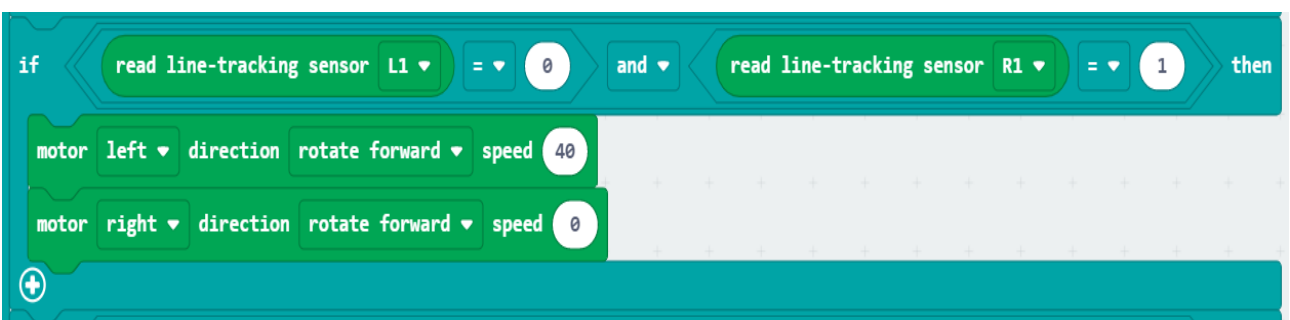
- For condition 'else', add the block 'if...then' (Logic). Duplicate block of 1st condition 'if'[---] and [---]' and change 'read line-tracking sensor [R1] = [0]'.



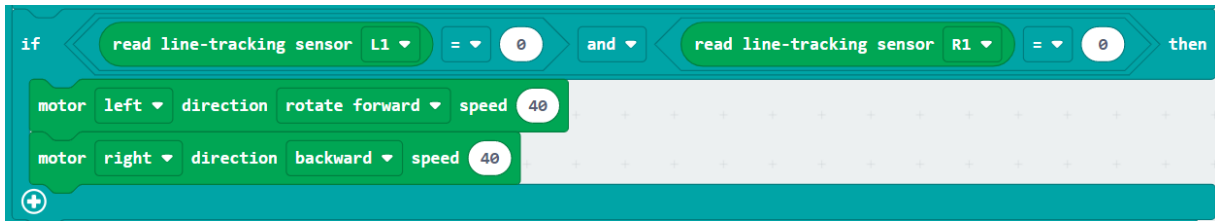
- For condition 'then', add the block 'motor [left] direction [rotate forward] speed [0]' (Maqueen Plus) and 'motor [right] direction [rotate forward] speed [40]'.



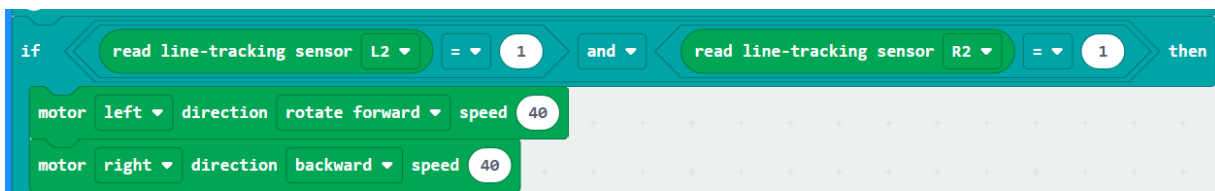
- Duplicate block of condition 'if...then'. For condition 'if', change value 'read...[L1]=[0]' and 'read...[R1]=[1]'. For condition 'then', change 'motor [left]...[rotate forward]...[40]' and 'motor [right]...[rotate forward]...[0]'.



- Duplicate block of condition 'if...then'. For condition 'if', change value 'read...[L1]=[0]' and 'read...[R1]=[0]'. For condition 'then', change 'motor [left]...[rotate forward]...[40]' and 'motor [right]...[backward]...[40]'.



- Duplicate block of condition 'if...then'. For condition 'if', change value 'read...[L2]=[1]' and 'read...[R2]=[1]'. For condition 'then', change 'motor [left]...[rotate forward]...[40]' and 'motor [right]...[backward]...[40]'.



- Slot in steps 9, 10 and 11 as shown below.



13. Click **Download** to transfer your code to micro:bit.

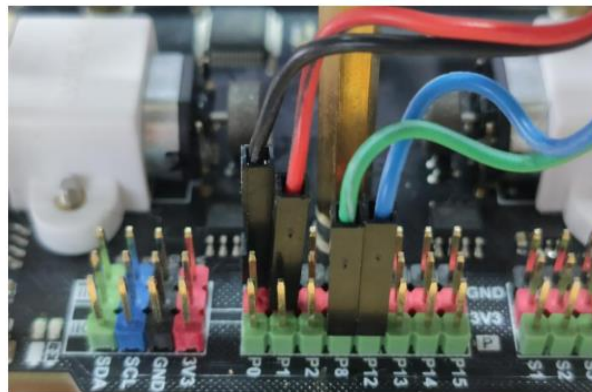
Program Link :

https://makecode.microbit.org/_37qXCL1oLX1s

Chapter 6: Obstacle Avoidance Robot

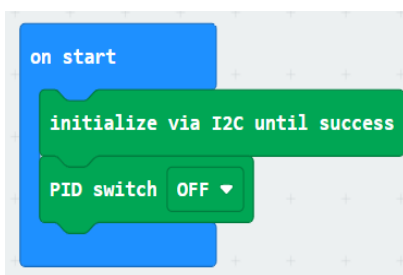
The ultrasonic sensor continuously measures and detects the distance between the Bluguard Maqueen Plus and the obstacle ahead in moving, when the distance is smaller than 20cm, Bluguard Maqueen Plus would turn left or right to avoid the obstacle.

You may connect the ultrasonic sensor to P8(green wire) and P12(blue wire), the designated port setting in the program. The red wire should be connected to a 3.3V port and the black one to a GND port.

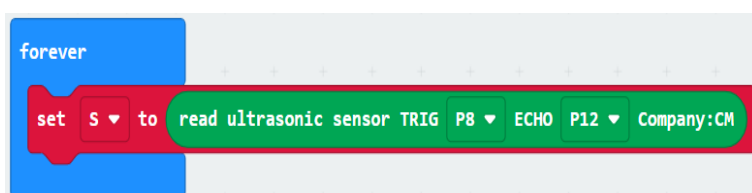


Step of Makecode Graphical Program:

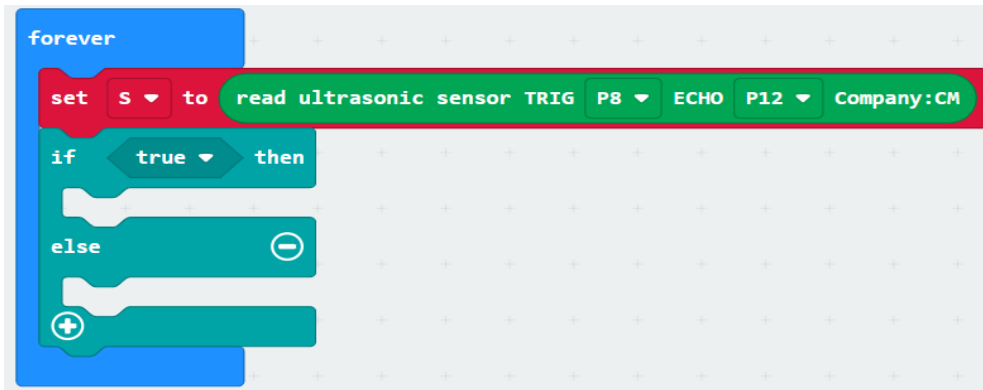
1. Add and drag the block 'initialize via I2C until success' (Maqueen Plus) and 'PID switch [OFF]' (Maqueen Plus) to the block 'on start' (Basic).



2. Go to (Variables) and make a variable 'S'. Add the block 'set [S] to [0]' (Variables). Replace the '[0]' with 'read ultrasonic sensor TRIG [P8] ECHO [P12] Company:CM' (Maqueen Plus) into block 'forever' (Basic).



3. Add the block 'if...then...else' (Logic).



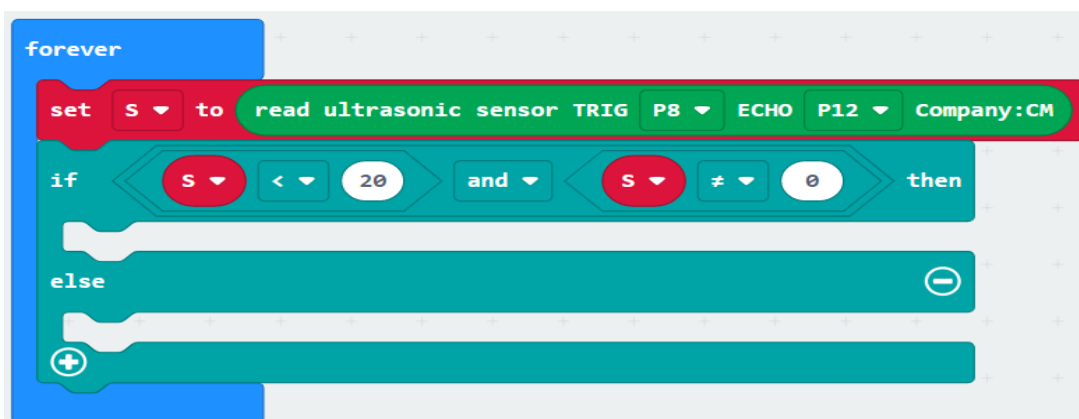
4. Add the block '[0] < [0]' (Logic) and 'S' (Variables). Replace the value '[0]' with 'S' and change another value '[0]' to '[20]'.



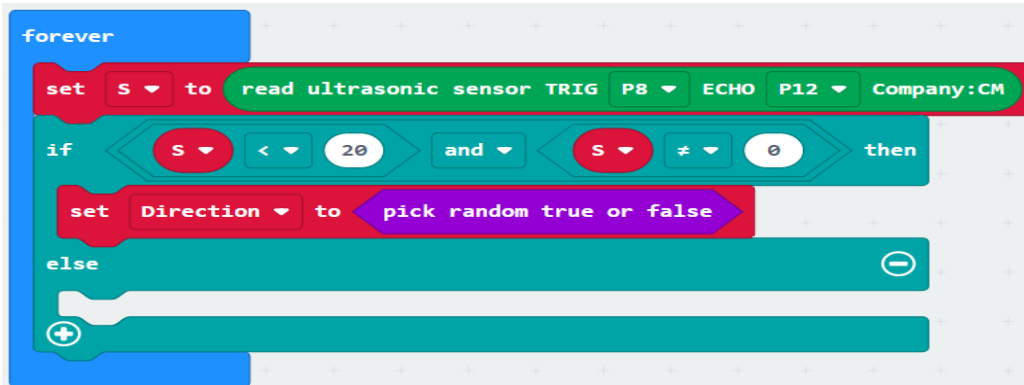
5. Duplicate the step 4 block and change '[<]' to '[≠]' and '[20]' to '[0]'.



6. For condition 'if', add the block '[] and []' (Logic). Slot in '[block step 4] and [block step 5]'.

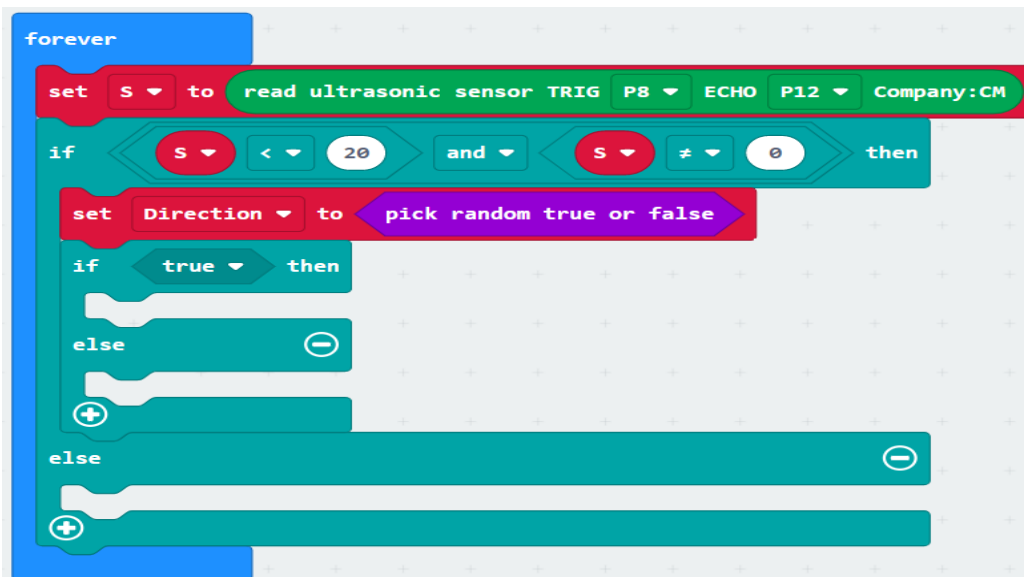


7. Create a variable 'Direction' (Variables). For condition 'then', add the block 'set [Direction] to [0]' (Variables). Replace the '[0]' with 'pick random true or false' (Math).



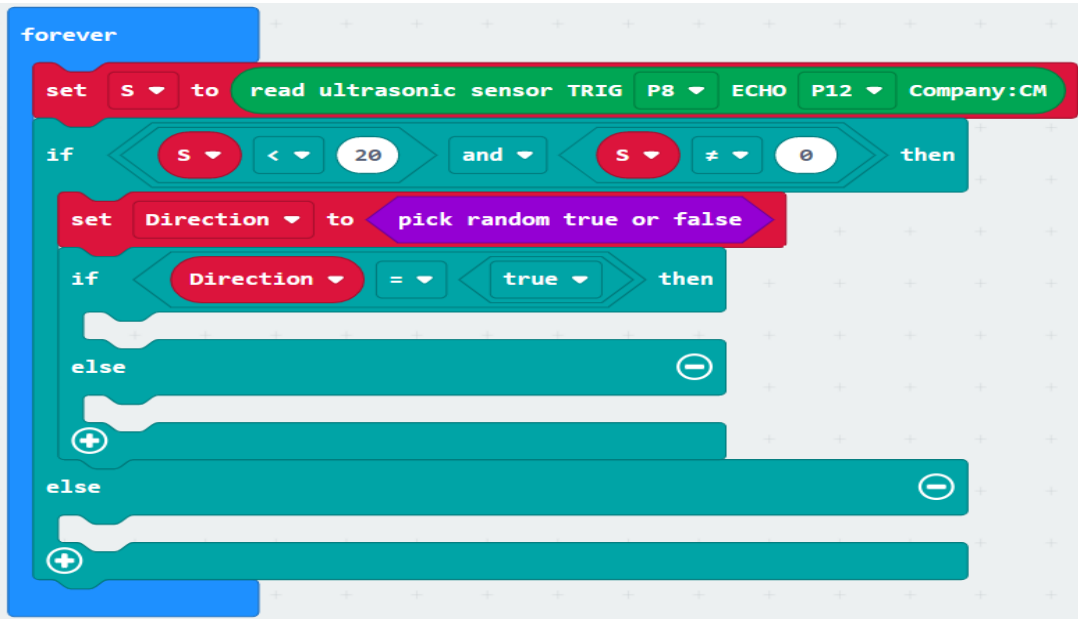
The image shows a Scratch code block for step 7. It is a 'forever' loop containing a 'set S to read ultrasonic sensor TRIG P8 ECHO P12 Company:CM' block. Below this is an 'if' block with the condition 'S < 20 and S ≠ 0'. Inside the 'if' block, there is a 'set Direction to pick random true or false' block. The 'else' part of the 'if' block is currently empty.

8. Add the block 'if...then...else' (Logic).



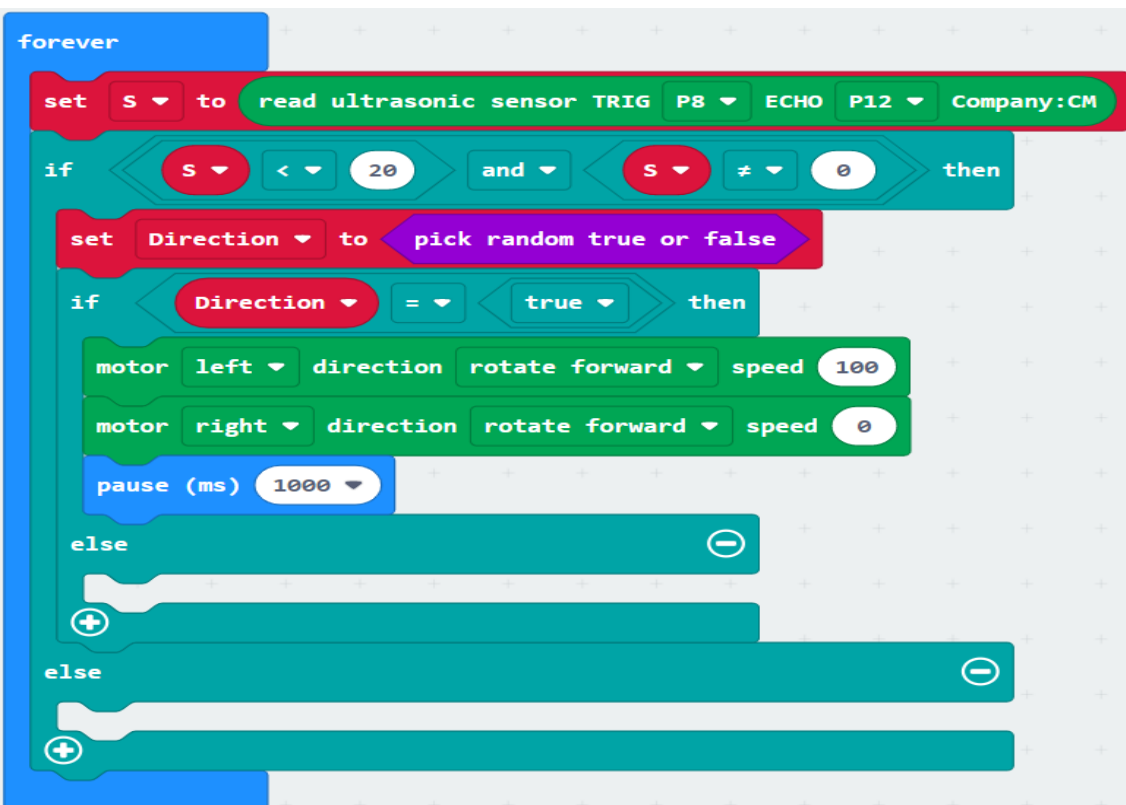
The image shows a Scratch code block for step 8. It is a 'forever' loop containing a 'set S to read ultrasonic sensor TRIG P8 ECHO P12 Company:CM' block. Below this is an 'if' block with the condition 'S < 20 and S ≠ 0'. Inside the 'if' block, there is a 'set Direction to pick random true or false' block. Below that is another 'if' block with the condition 'true'. This second 'if' block has an empty 'then' part and an empty 'else' part. The 'else' part of the first 'if' block is also empty.

9. For condition 'if', add the block '[0] = [0]' (Logic) and change it to '[Direction] = [true]' (Logic).



The image shows a Scratch code block for step 9. It is a 'forever' loop containing the following blocks: a 'set S to read ultrasonic sensor TRIG P8 ECHO P12 Company:CM' block; an 'if' block with conditions 'S < 20' and 'S ≠ 0'; a 'set Direction to pick random true or false' block; an 'if' block with condition 'Direction = true'; and two 'else' blocks, each with a '+' or '-' sign.

10. For condition 'then', add the block 'motor [left] direction [rotate forward] speed [100]' (Maqueen Plus) and 'motor [right] direction [rotate forward] speed [0]'. After that, add the block 'pause (ms) [1000]' (Basic).



The image shows a Scratch code block for step 10. It is a 'forever' loop containing the following blocks: a 'set S to read ultrasonic sensor TRIG P8 ECHO P12 Company:CM' block; an 'if' block with conditions 'S < 20' and 'S ≠ 0'; a 'set Direction to pick random true or false' block; an 'if' block with condition 'Direction = true'; a 'motor left direction rotate forward speed 100' block; a 'motor right direction rotate forward speed 0' block; a 'pause (ms) 1000' block; and two 'else' blocks, each with a '+' or '-' sign.

11. For condition 'else', add the block 'motor [left] direction [rotate forward] speed [0]' (Maqueen Plus) and 'motor [right] direction [rotate forward] speed [100]'. After that add the block 'pause (ms) [1000]' (Basic).

```
forever
  set S to read ultrasonic sensor TRIG P8 ECHO P12 Company:CM
  if S < 20 and S ≠ 0 then
    set Direction to pick random true or false
    if Direction = true then
      motor left direction rotate forward speed 100
      motor right direction rotate forward speed 0
      pause (ms) 1000
    else
      motor left direction rotate forward speed 0
      motor right direction rotate forward speed 100
      pause (ms) 1000
  else
    [ ]
  [ ]
```

The image shows a Scratch script for a robot's ultrasonic sensor. It starts with a 'forever' loop. Inside the loop, a variable 'S' is set to the value of an ultrasonic sensor (TRIG P8, ECHO P12, Company:CM). An 'if' block checks if 'S' is less than 20 and not equal to 0. If true, a 'Direction' variable is set to a random choice of true or false. Another 'if' block checks if 'Direction' is true. If true, the left motor is set to rotate forward at speed 100, and the right motor is set to rotate forward at speed 0. A 'pause (ms) 1000' block follows. If 'Direction' is false, the left motor is set to rotate forward at speed 0, and the right motor is set to rotate forward at speed 100. Another 'pause (ms) 1000' block follows. There are also empty 'else' and 'else' blocks at the bottom of the script.

- For condition 'else', add the block 'motor [ALL] direction [rotate forward] speed [100]' (Maqueen Plus).

```
forever
  set S to read ultrasonic sensor TRIG P8 ECHO P12 Company:CM
  if S < 20 and S != 0 then
    set Direction to pick random true or false
    if Direction = true then
      motor left direction rotate forward speed 100
      motor right direction rotate forward speed 0
      pause (ms) 1000
    else
      motor left direction rotate forward speed 0
      motor right direction rotate forward speed 100
      pause (ms) 1000
  else
    motor ALL direction rotate forward speed 100
```

- Click **Download** to transfer your code to micro:bit.

Program Link :

https://makecode.microbit.org/_7R0icURoX6Ed

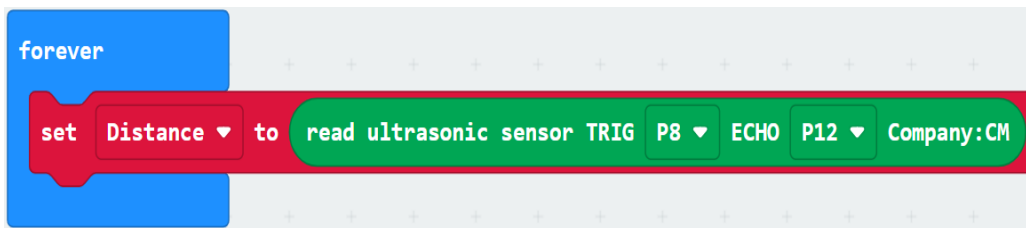
Chapter 7: Car Park Helper

Nowadays, reversing radar systems are commonly installed in the car to assist the driver to help the driver reverse and park the car more safely and easily. The reversing radar system can detect the distance between the car and obstacle via ultrasonic sensors, then alert users by beeps or the dashboard display.

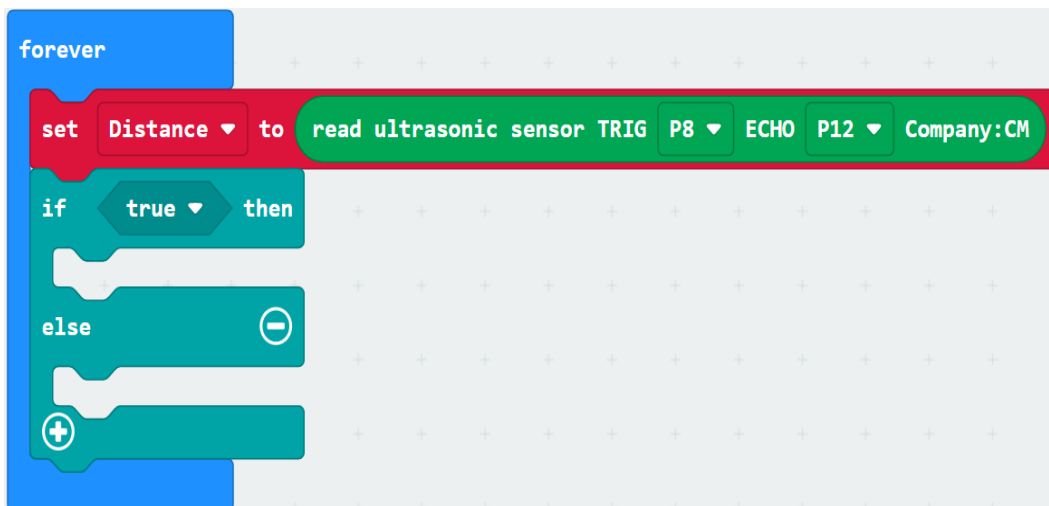
Here, you may learn how to measure the distance back of the Blugard Maqueen Plus and obstacles. First, we need to install an ultrasonic sensor facing the back direction.

Step of Makecode Graphical Program:

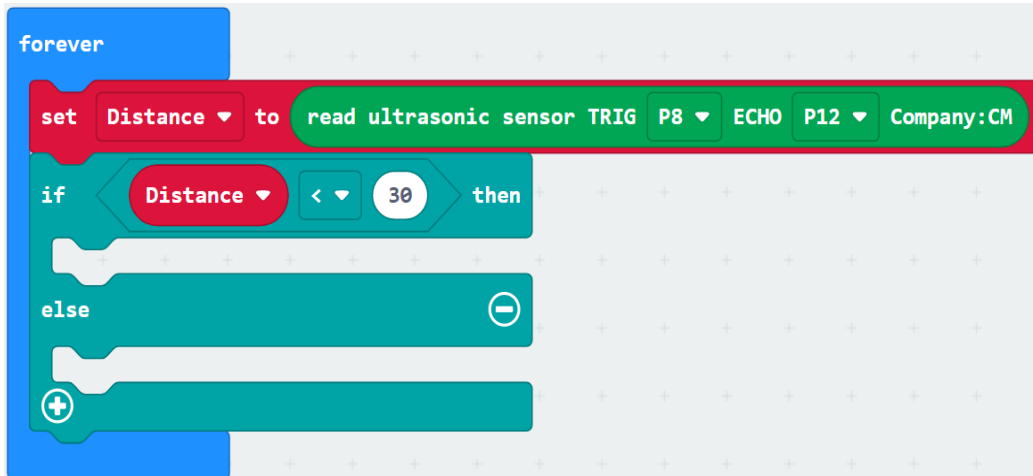
1. Create a variable 'Distance'(Variables). Add the block 'set [Distance] to [0]' (Variables). Replace the '[0]' with 'read ultrasonic sensor TRIG [P8] ECHO [P12] Company:CM' (Maqueen Plus) into block 'forever' (Basic).



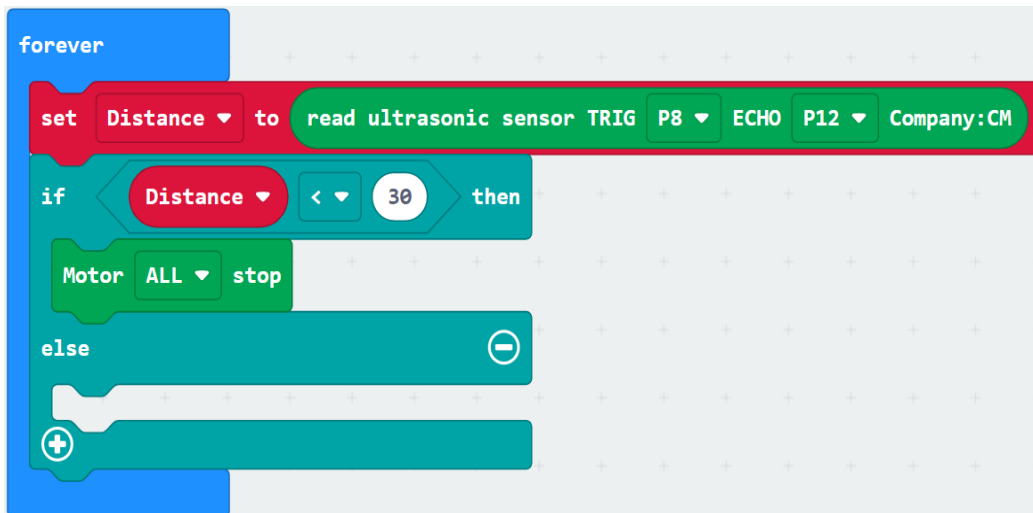
2. Add the block 'if...then...else' (Logic).



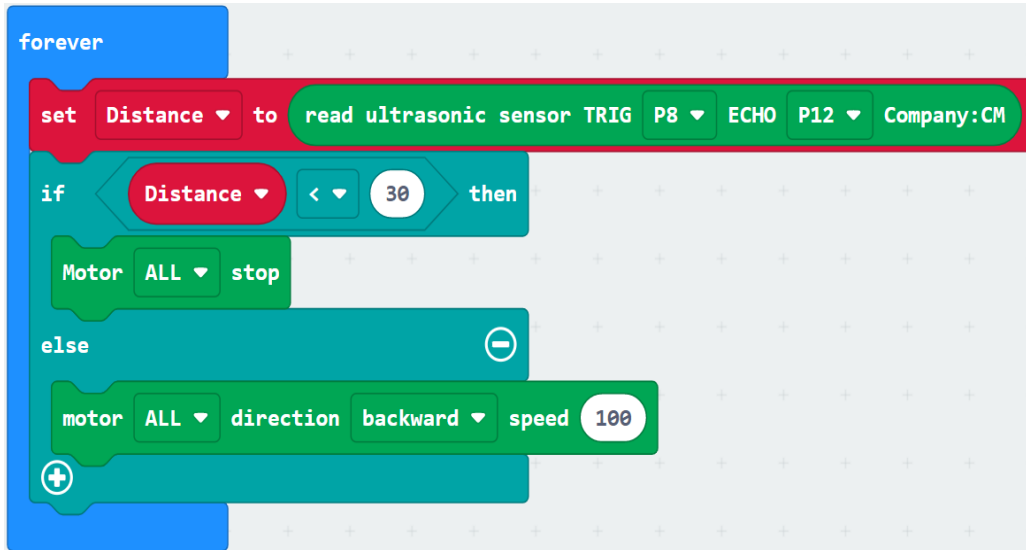
- For condition 'if', add the block '[0] < [0]' (Logic) and change it to '[Distance] < [30]' (Logic).



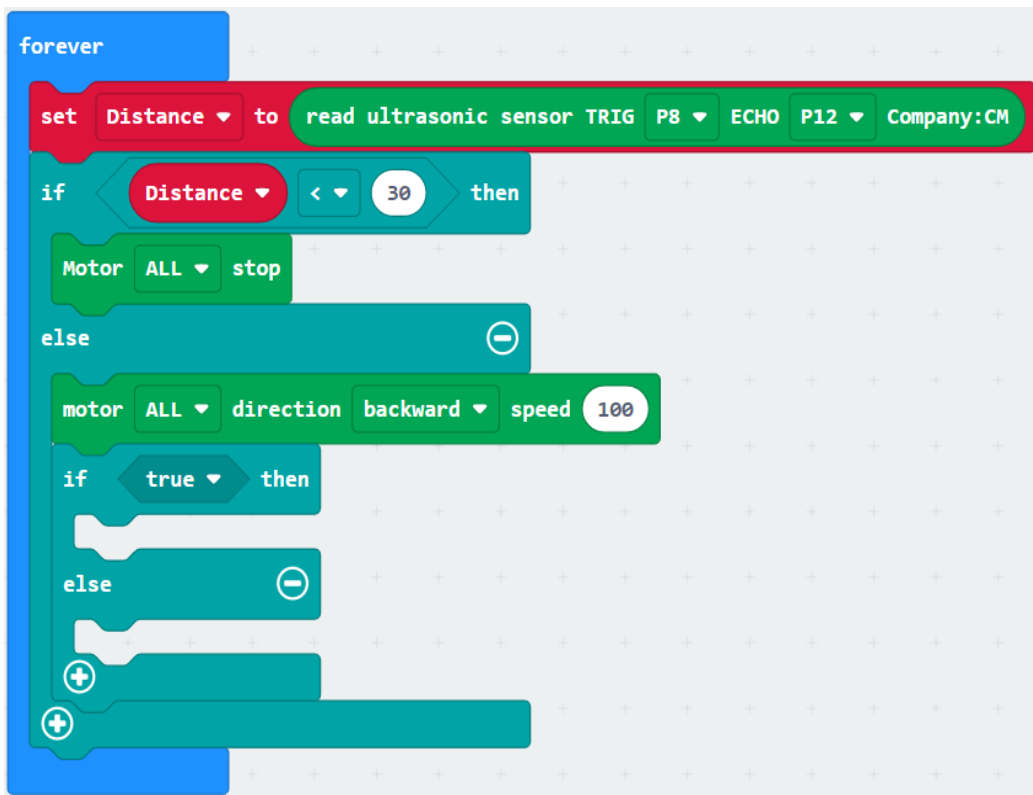
- For condition 'then', add the block 'Motor [ALL] stop' (Maqueen Plus).



- For condition 'else', add the block 'Motor [ALL] direction [backward] speed [100]' (Maqueen Plus).



- Add the block 'if...then...else' (Logic).



- For condition 'if', add the block '[0] < [0]' (Logic) and change it to '[Distance] < [200]' (Logic).

```
forever
  set Distance to read ultrasonic sensor TRIG P8 ECHO P12 Company:CM
  if Distance < 30 then
    Motor ALL stop
  else
    motor ALL direction backward speed 100
    if Distance < 200 then
      
```

- For condition 'then', add the block 'play tone [Middle C] for [1 beat]' (Music).

```
forever
  set Distance to read ultrasonic sensor TRIG P8 ECHO P12 Company:CM
  if Distance < 30 then
    Motor ALL stop
  else
    motor ALL direction backward speed 100
    if Distance < 200 then
      play tone Middle C for 1 beat
    else
      
```


9. For condition 'else', add the block 'stop melody [all]' (Music).

```
forever
  set Distance to read ultrasonic sensor TRIG P8 ECHO P12 Company:CM
  if Distance < 30 then
    Motor ALL stop
  else
    motor ALL direction backward speed 100
    if Distance < 200 then
      play tone Middle c for 1 beat
    else
      stop melody all
```

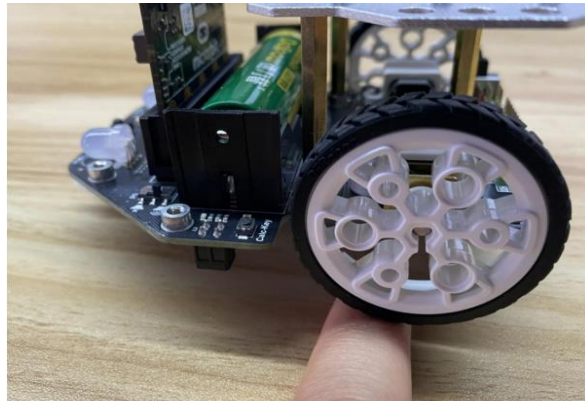
10. Click **Download** to transfer your code to micro:bit.

Program Link :

https://makecode.microbit.org/_Kh8a0tcEzXTy

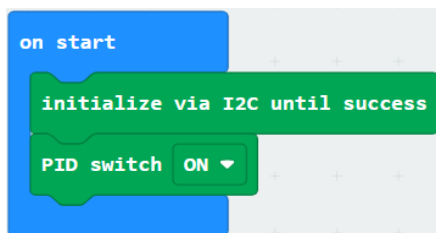
Chapter 8: PID Control for Bluguard Maqueen Plus

PID is used to adjust the speed of the two motors and provide sufficient torque at various speeds. Bluguard Maqueen Plus is equipped with an onboard encoder and PID control function, allowing a user to adjust the torque and speed of a motor in real-time. You may download the program, and try letting Maqueen Plus climb across some small obstacles like a finger, eraser, etc.

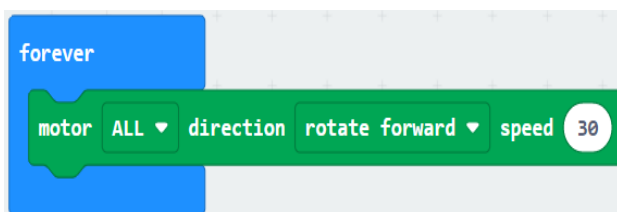


Step of Makecode Graphical Program:

1. Add and drag the block 'initialize via I2C until success' (Maqueen Plus) and 'PID switch [ON]' (Maqueen Plus) to the block 'on start' (Basic).



2. Add the block 'motor [ALL] direction [rotate forward] speed [30]' (Maqueen Plus) to the block 'forever' (Basic).



3. Click **Download** to transfer your code to micro:bit.

Program Link :

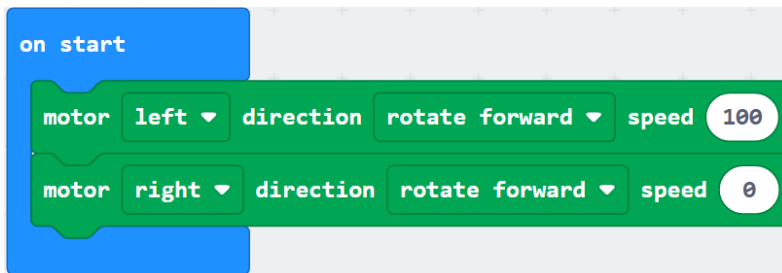
https://makecode.microbit.org/_5ejfYY6TpXef

Chapter 9: Play Music with RGB Light Blinking

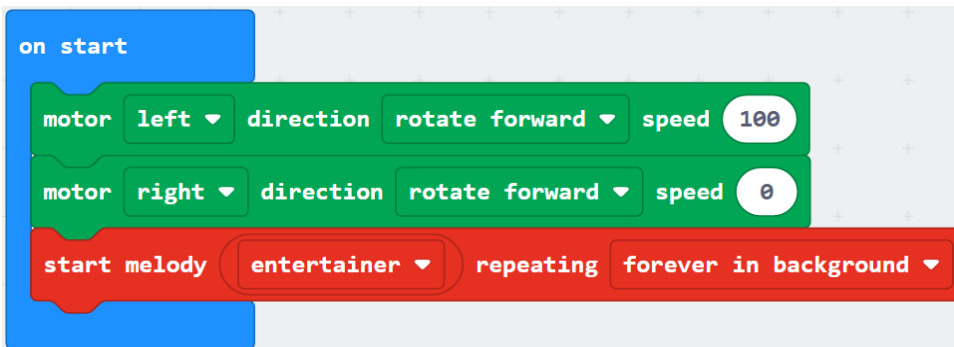
This demo will show you how to program the buzzer module and the RGB light module.

Step of Makecode Graphical Program:

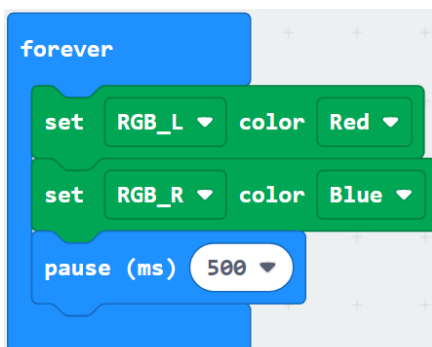
1. Add the block 'motor [left] direction [rotate forward] speed [100]' (Maqueen Plus) and 'motor [right] direction [rotate forward] speed [0]' to the block 'on start' (Basic).



2. Add the block 'start melody [entertainer] repeating [forever in background]' (Music).



3. Add the block 'set [RGB_L] color [Red]' (Maqueen Plus), 'set [RGB_R] color [Blue]' and block 'pause (ms) [500]' (Basic) to block 'forever' (Basic).



4. Add the block 'set [RGB_L] color [Blue]' (Maqueen Plus), 'set [RGB_R] color [Red]' and block 'pause (ms) [500]' (Basic).



5. Click **Download** to transfer your code to micro:bit.

Program Link :

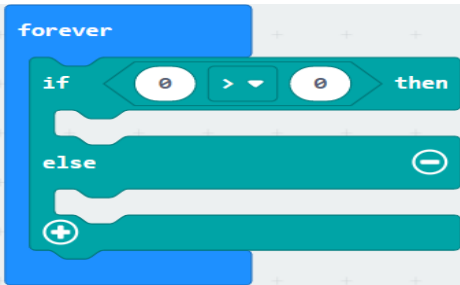
https://makecode.microbit.org/_LPD6mF9bDhXF

Chapter 10: Light Chaser

When the light level is more than the given value (eg: 100), Bluguard Maqueen Plus will move towards the light source, otherwise, it will revolve around its centre.

Step of Makecode Graphical Program:

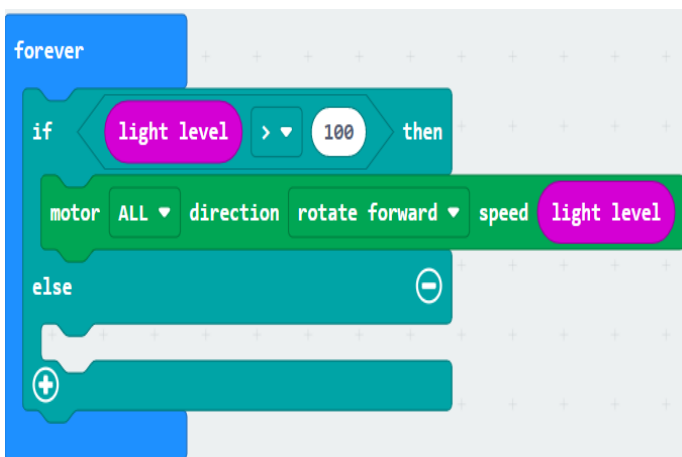
1. Add the block 'forever' (Basic), 'if...then...else' (Logic)' and '[0] > [0]' (Logic).



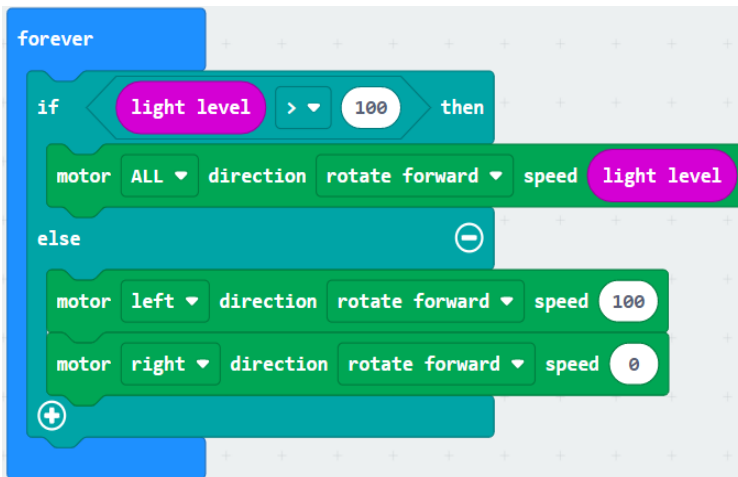
2. For condition 'if', replace left '[0]' with '[light level]' (Input). Change the value another '[0]' to '[100]'.



3. For condition 'then', add the block 'motor [ALL] direction [rotate forward] speed [0]' (Maqueen Plus) and replace '[0]' with '[light level]' (Input).



4. For condition 'else', add the block 'motor [left] direction [rotate forward] speed [100]' (Maqueen Plus) and 'motor [right] direction [rotate forward] speed [0]'.



5. Click **Download** to transfer your code to micro:bit.

Program Link :

https://makecode.microbit.org/_5wDX5sco2He4

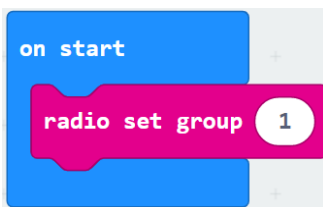
Chapter 11: Motion Sensing Racing Car

This chapter consists of lessons on how to control the movement of Bluguard Maqueen Plus by changing the direction of the remote controller. Bluguard Maqueen Plus can execute the motion-sensing since the accelerometer on the micro:bit can simultaneously detect the orientation of the board.

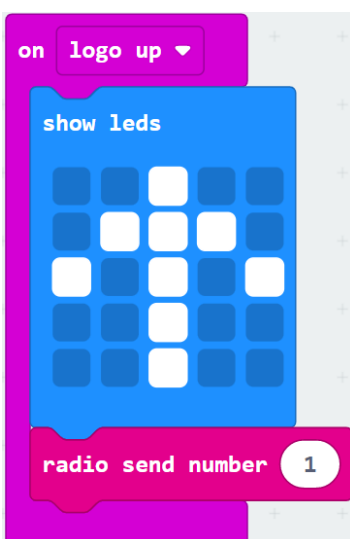
We need two micro:bits to perform radio communication. One for sending out signal (transmitting end), one for receiving a signal (receiving end).

Step of Makecode Graphical Program:

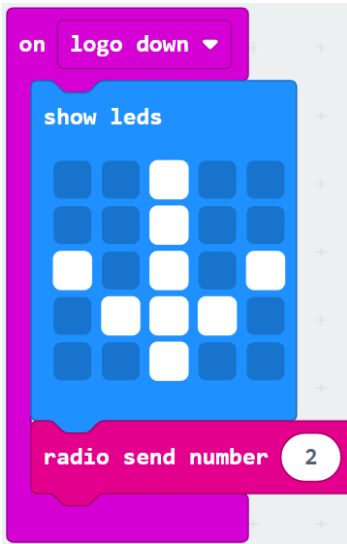
1. Create a new project name 'Motion Sensing Racing Car – Transmitting end'.
2. To enable the radio communication, add the block 'radio set group [1]' (Radio) in the block 'on start' (Basic).



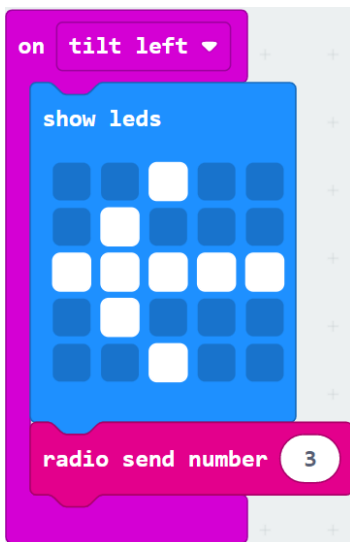
3. Add the block 'on [logo up]' (Input). Add the block 'show leds [↑]' (Basic) and the block 'radio send number [1]' (Radio).



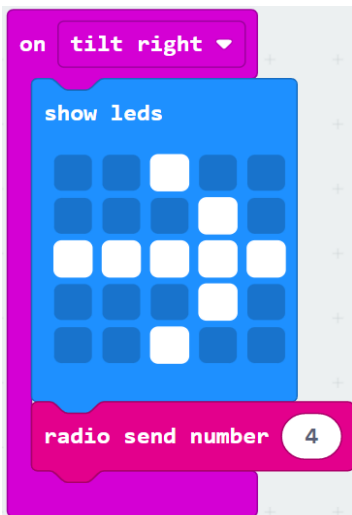
4. Add the block 'on [logo down]' (Input). Add the block 'show leds [↓]' (Basic) and the block 'radio send number [2]' (Radio).



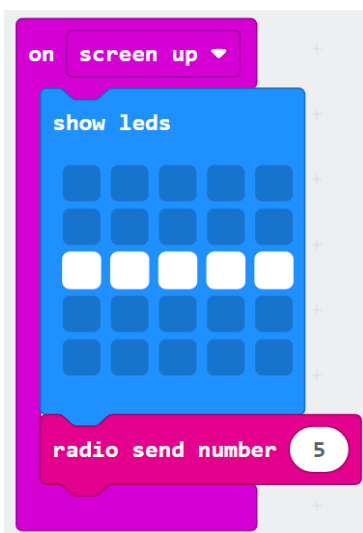
5. Add the block 'on [tilt left]' (Input). Add the block 'show leds [←]' (Basic) and the block 'radio send number [3]' (Radio).



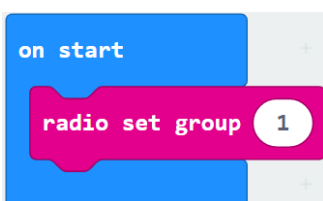
6. Add the block 'on [tilt right]' (Input). Add the block 'show leds [→]' (Basic) and the block 'radio send number [4]' (Radio).



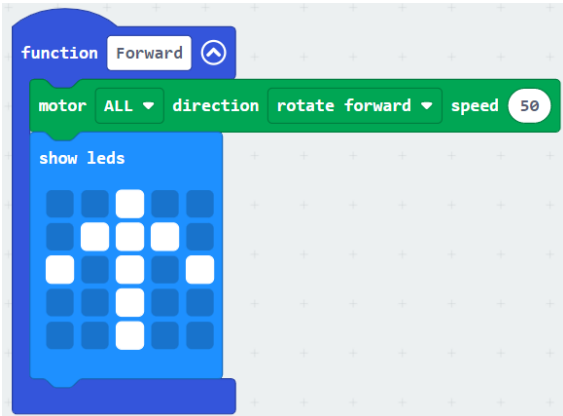
7. Add the block 'on [screen up]' (Input). Add the block 'show leds [---]' (Basic) and the block 'radio sends number [5]' (Radio).



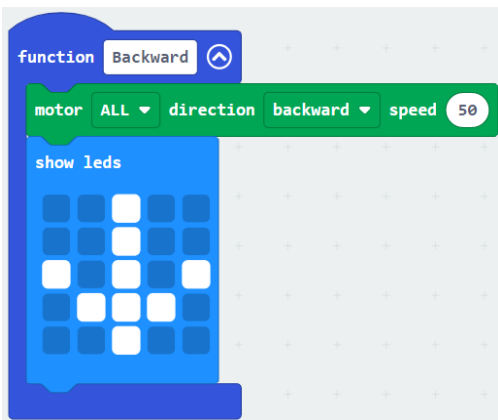
8. Click **Download** to transfer your code to 1st unit micro:bit.
9. Create a new project name 'Motion Sensing Racing Car – Receiving end'.
10. To enable the radio communication, add the block 'radio set group [1]' (Radio) in the block 'on start' (Basic).



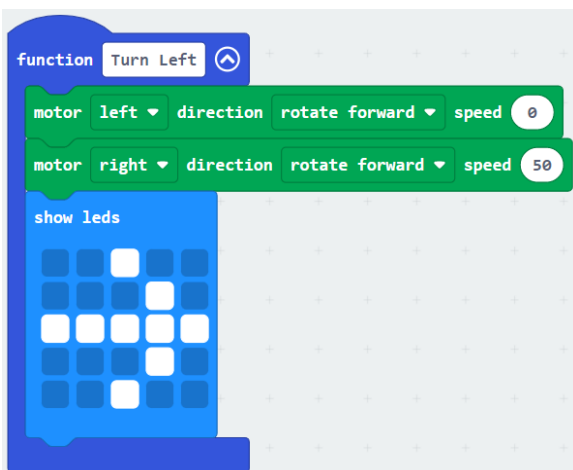
11. Add the block 'function [Forward]' (Function) and the block 'motor [ALL] direction [rotate forward] speed [50]' (Maqueen Plus). Add the block 'show LEDs [↑]' (Basic).



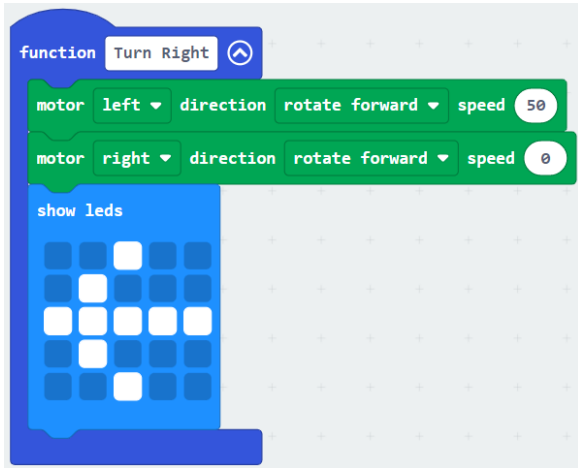
12. Add the block 'function [Backward]' (Function) and the block 'motor [ALL] direction [backward] speed [50]' (Maqueen Plus). Add the block 'show leds [↓]' (Basic).



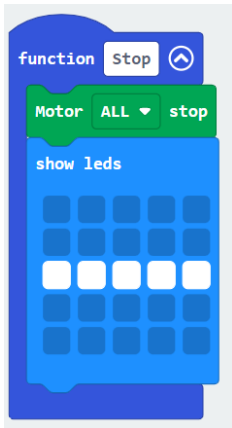
13. Add the block 'function [Turn Left]' (Function). Add the block 'motor [left] direction [rotate forward] speed [0]' (Maqueen Plus) and 'motor [right] direction [rotate forward] speed [50]'. Add the block 'show leds [→]' (Basic).



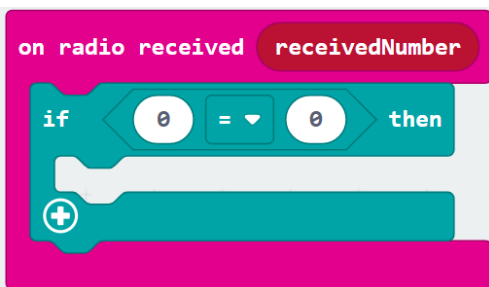
14. Add the block 'function [Turn Left]' (Function). Add the block 'motor [left] direction [rotate forward] speed [0]' (Maqueen Plus) and 'motor [right] direction [rotate forward] speed [50]'. Add the block 'show leds [←]' (Basic).



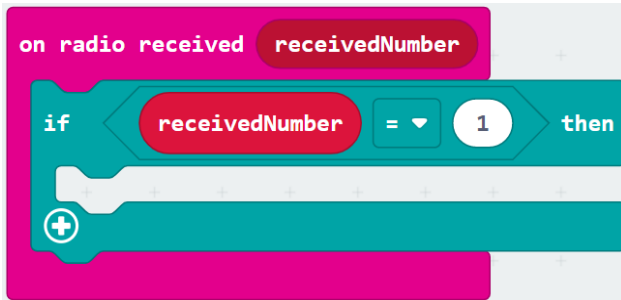
15. Add the block 'function [Stop]' (Function). Add the block 'motor [All] stop' (Maqueen Plus). Add the block 'show leds [---]' (Basic).



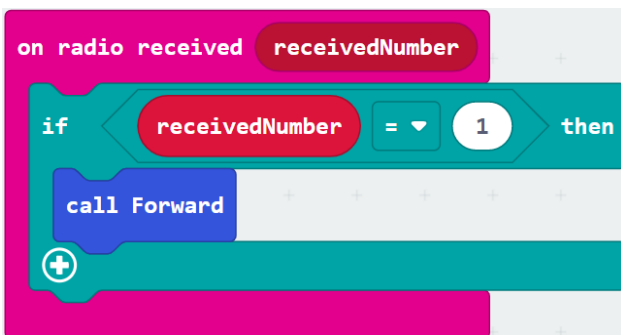
16. Add the block 'on radio received [receivedNumber]' (Radio) and the block 'if...then' (Logic) and '[0] = [0]' (Logic).



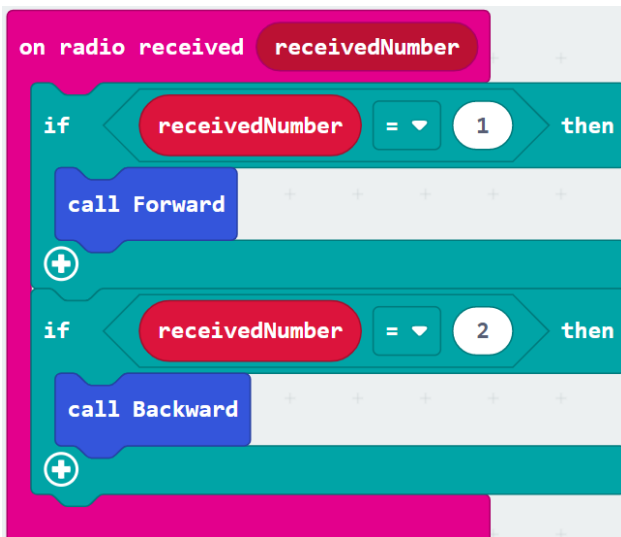
- For condition 'if', drag the '[receivedNumber]' to replace left '[0]' and replace the value another '[0]' to '[1]'.



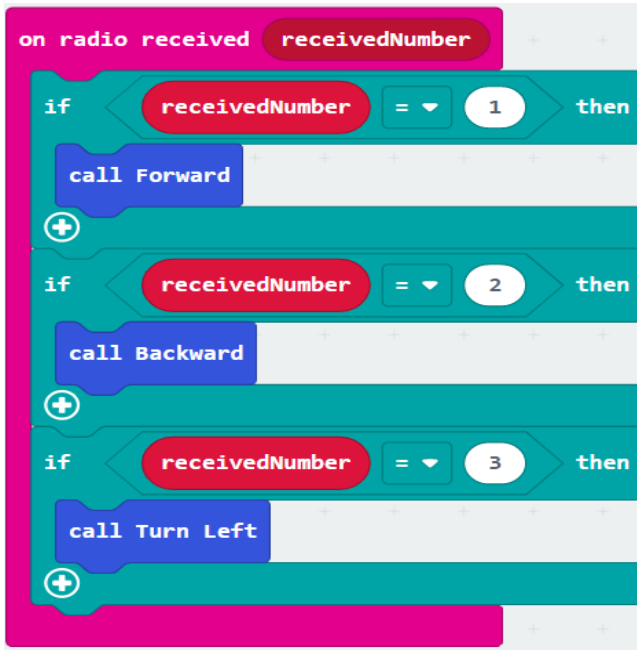
- For condition 'then', add the block '[call Forward]' (Function).



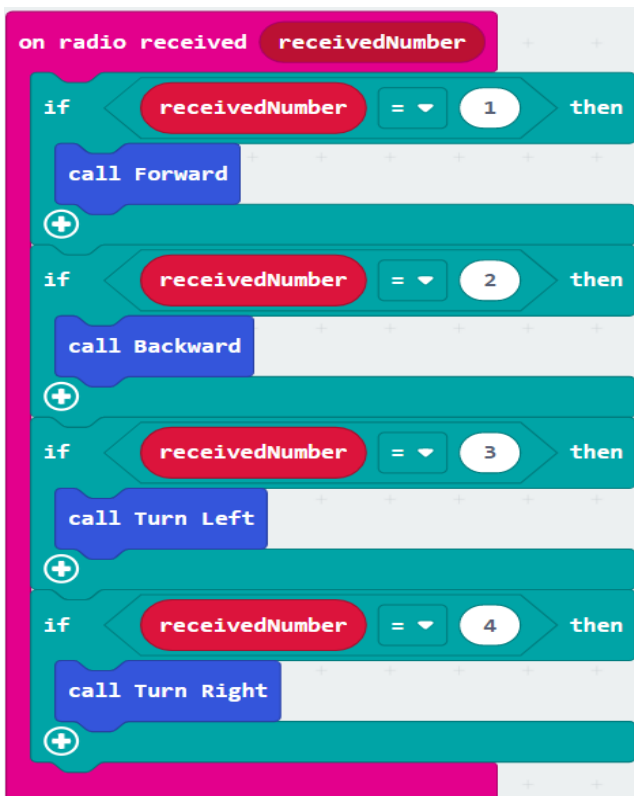
- Add the block 'if...then' (Logic). Duplicate the 1st block conditional 'if' '[----]=[----]' and replace value '[1]' to '[2]'. For condition 'then', add the block '[call Backward]' (Function).



20. Add the block 'if...then' (Logic). Duplicate the 2nd block conditional 'if' '----]=[----]' and replace value '[2]' to '[3]'. For condition 'then', add the block '[call Turn Left]' (Function).



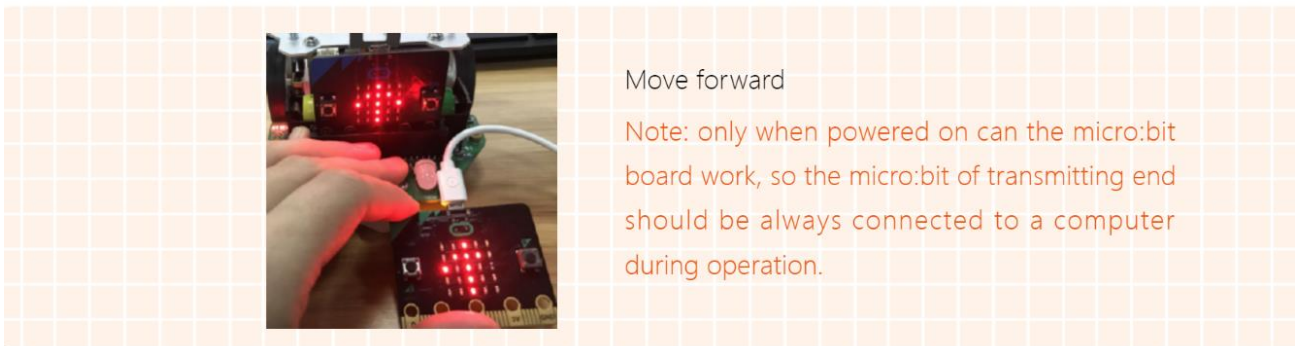
21. Add the block 'if...then' (Logic). Duplicate the 3rd block conditional 'if' '----]=[----]' and replace value '[3]' to '[4]'. For condition 'then', add the block '[call Turn Right]' (Function).



22. Add the block 'if...then' (Logic). Duplicate the 4th block conditional 'if' '[----]=[----]' and replace value '[4]' to '[5]'. For condition 'then', add the block '[call Stop]' (Function).

```
on radio received receivedNumber
if receivedNumber = 1 then
  call Forward
+
if receivedNumber = 2 then
  call Backward
+
if receivedNumber = 3 then
  call Turn Left
+
if receivedNumber = 4 then
  call Turn Right
+
if receivedNumber = 5 then
  call Stop
+
```

23. Click **Download** to transfer your code to 2nd unit micro:bit.
24. When we complete all the above steps, turn Maqueen Plus's power switch on, then we can use the micro:bit of the transmitting end to control the car.



Program Link :

Motion Sensing Racing Car - Transmitting End :

https://makecode.microbit.org/_Tpdgrt4yaKEh

Motion Sensing Racing Car – Receiving End :

https://makecode.microbit.org/_WLM6WXhPLLLH