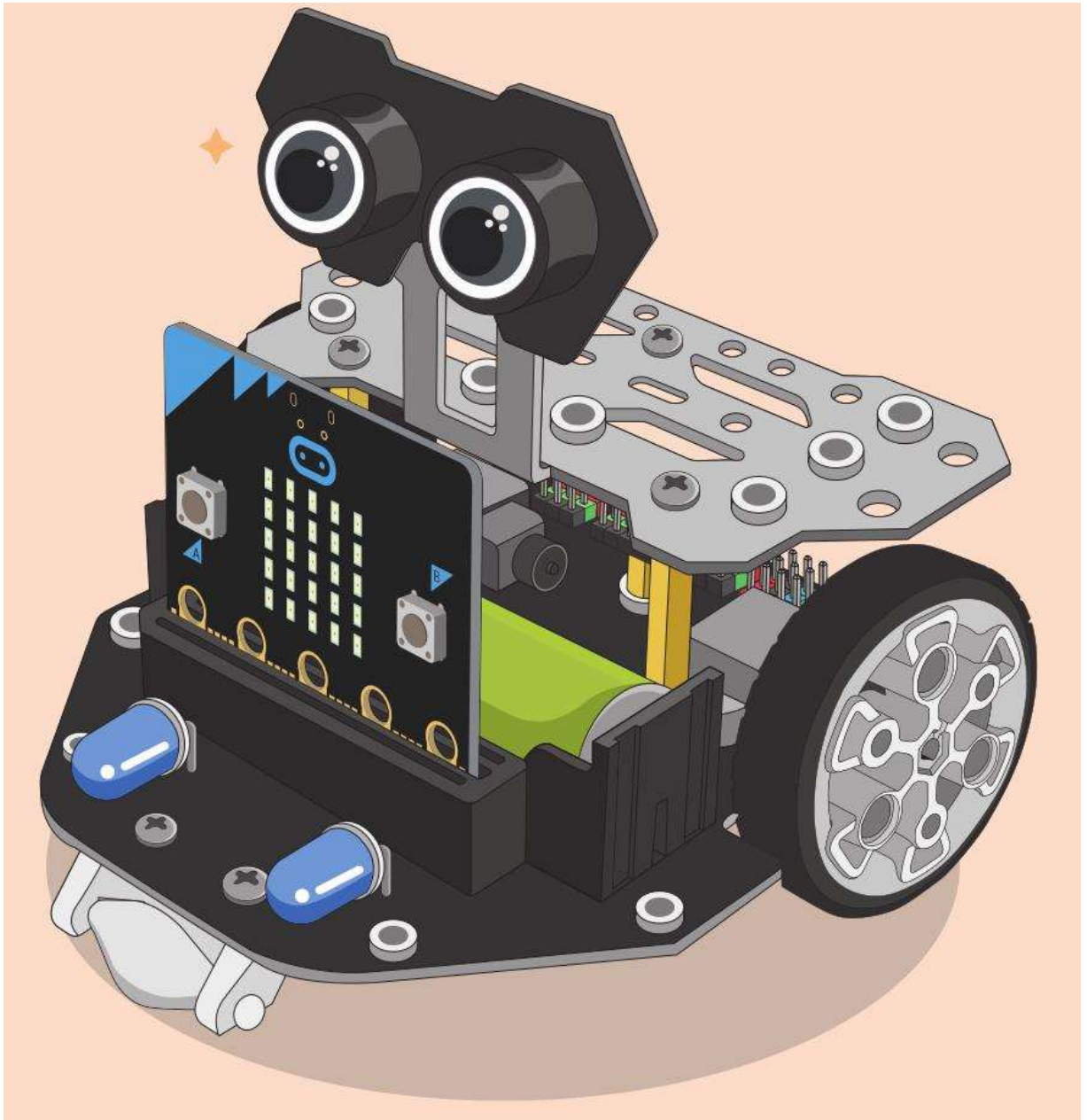


Getting Started with Maqueen Plus



www.DFRobot.com

Content

Chapter 1 Introduction to Maqueen Plus.....	3
Introduction.....	3
Overview.....	4
Specification.....	5
Maqueen Plus vs Maqueen Lite.....	6
Assembly Guide.....	7
How to calibrate line-tracking sensor?.....	8
Chapter 2 Programming Maqueen Plus on MakeCode.....	10
MakeCode address and program library.....	10
Maqueen Plus Functions.....	10
HuskyLens AI Camera Block Description.....	12
Project 1: Line follower moving along a circle.....	16
Project 2: Line follower moving along a cross line.....	17
Project 3: Obstacle Avoidance Robot.....	18
Project 4: IR-controlled Maqueen Plus.....	20
Project 5: PID Control for Maqueen Plus.....	21
Project 6: Speed up and Slow down.....	22
Project 7: HuskyLens AI Camera - Line Tracking.....	23
Project 8: HuskyLens AI Camera - Tail After.....	24
Project 9: HuskyLens AI Camera - Passing a Traffic Light.....	25
Project 10: Maqueen Mechanic - Loader.....	25

Chapter 1 Introduction to Maqueen Plus

Introduction

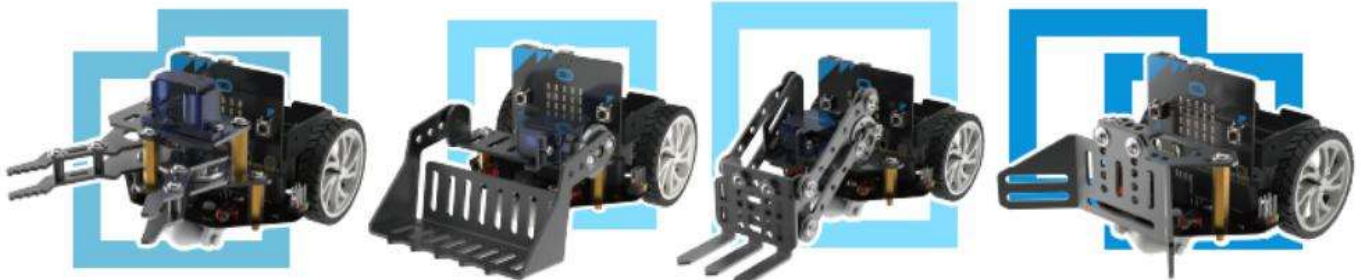
Micro:Maqueen launched by DFRobot is a series of educational robot products for primary and secondary school programming. It includes a cost-effective “Lite” version, a powerful “Plus” version and rich peripherals such as “Mechanic”. You can choose different versions and peripherals based on your needs.



micro:Maqueen Lite

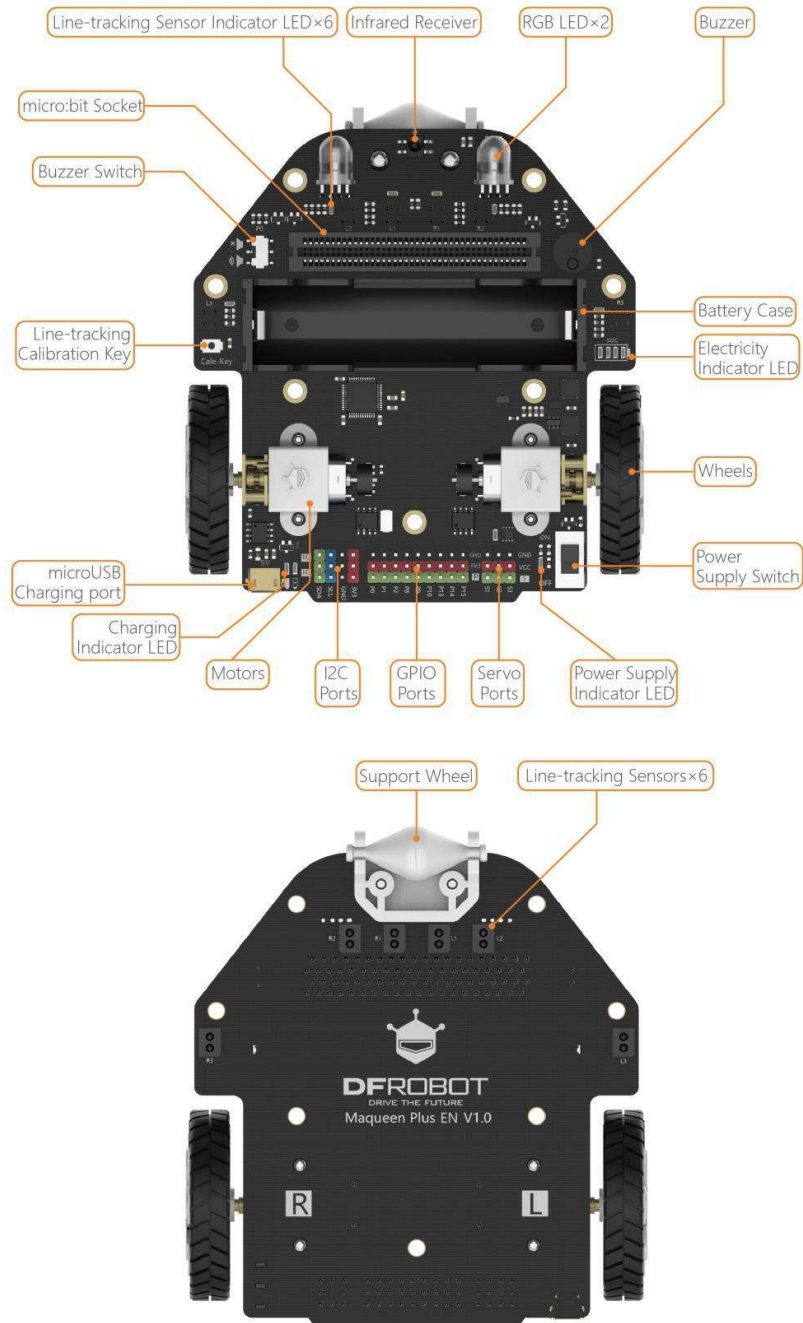


micro:Maqueen Plus



micro:Maqueen Mechanic

Overview



Specification

Power Supply: 3.7V 18650 lithium battery

Charging Voltage: 5V

Charging Current: 900mA

Charging Time: about 4hours

Power Indicator: 4 LEDs

Motor Specification: N20 motor 260 R/M

Buzzer x1

RGB Light x2

GPIO Expansion Port: P0 P1 P2 P8 P12 P13 P14 P15 P16

I2C Port: x3

Servo Expansion Port: x3

Line-tracking Sensor x6

Line-tracking Sensor Output: digital +analog

Support Calibration for Line-tracking Sensor

IR Receiving Sensor x1

Ultrasonic Sensor: URM10

Top Metal Plate: x1

M3 Threaded Connections x12

Map Size: 50cmx50cm

Product Dimension: 107x100mm/4.21 x3.94"

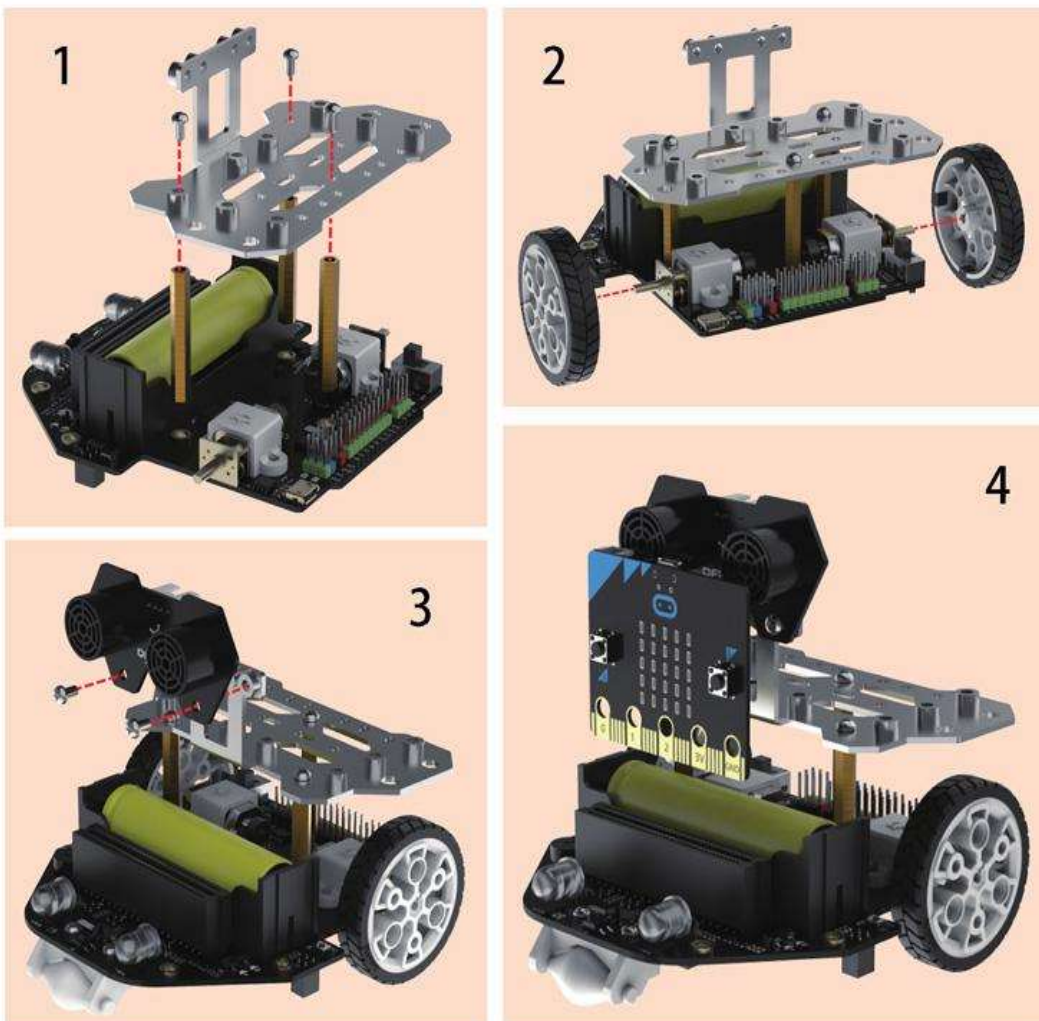
Maqueen Plus vs Maqueen Lite

Name	Maqueen Lite	Maqueen Plus
Power Supply	3 AAA Batteries	18650 Li-ion battery (2300mA~2500mA)
Charging circuit	x	√
Power display	x	√
Encoder and PID control	x	√
Support for installing Huskylens AI camera	x	√
Support for line-tracking sensor calibration	√	√
Support for analog reading of line-tracking sensor	x	√
Number of line-tracking sensor	2	6
Number of IO expansion port	4	12
Number of servo port	2	3
Number of mechanic expansion thread	2	16
Motor rated rotation speed	133	260
LED color	Red LED	Large size RGB LED with 7colors
Ultrasonic Model	H-SR04	DFRobot high-quality URM10 ultrasonic sensor
Continuous usage time with Huskylens	30min	180min

Standard continuous usage time	8h	24h
Come with line-tracking map	×	√
Onboard IR receiver and buzzer	√	√
Onboard WS2812 RGB LED	√	×

Assembly Guide

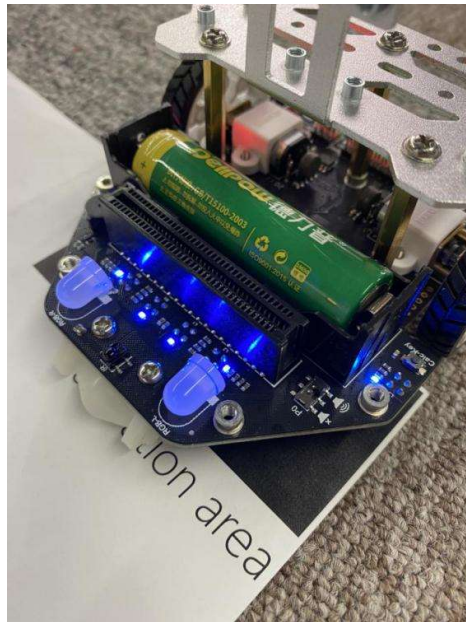
Note: power Maqueen Plus with 18650 chargeable lithium battery. Pay attention to polarity when installing battery, and it is prohibited to short circuit the battery's positive and negative poles.



How to calibrate line-tracking sensor?

There are 6 line-tracking sensors on Maqueen Plus and each of them has an indicator. When a line-tracking sensor detects a black line, the corresponding indicator will light up. If you found that any line-tracking sensor is not sensitive to a black line, calibrate it as follows:

1. Put Maqueen Plus into the calibration area of the line-tracking map, turn on its power.



2. Press "Calc-key" for about 1 second, the 2 front large LEDs will flash in green. Release the key, then calibration is done.



If all the line-tracking sensor indicators turn on in the black area and turn off in the white area, the calibration is successful.

Note:

1. The internal chip will automatically save the calibration, so you do not need to calibrate it every time you use it.
2. Maqueen Plus has been factory calibrated, and it can be used directly normally.

Chapter 2 Programming Maqueen Plus on MakeCode

The basic usage of MakeCode will be omitted here. This chapter will mainly introduce the function of Maqueen Plus and how to program it on MakeCode.

MakeCode address and program library

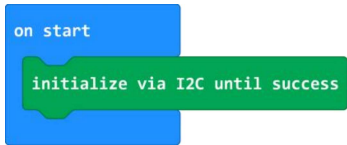



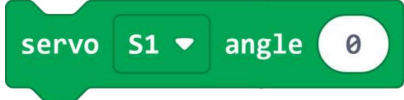
MakeCode programming platform address: <https://makecode.microbit.org>



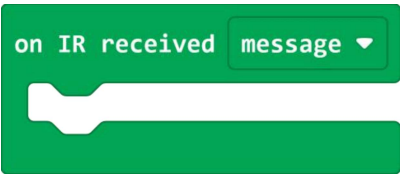


Maqueen Plus library: <https://github.com/DFRobot/pxt-DFRobot-MaqueenPlus>


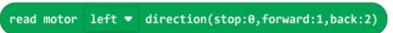


Huskylens AI Camera library: https://github.com/DFRobot/pxt-DFRobot_HuskyLens

OLED Screen library: <https://github.com/DFRobot/pxt-OLEDV1>



Maqueen Plus Functions






	<p>I2C Init Function: a necessary block for initializing I2C communication. This block only needs to run once at the start of main program. If the communication fails, micro:bit LED matrix will display "× " otherwise, it displays "√" .</p>
	<p>Motor Control Function: control the direction and speed of Maqueen Plus. Motor: left, right, all Direction: CW, CCW Speed: 0~255</p>
	<p>Motor Stop Function: stop motor, same as adjusting motor speed to 0. Motor: left, right, all</p>
	<p>RGB LED Control Function: control the two LEDs on Maqueen Plus. LED: RGB_L, RGB_R Color: red, green, blue, yellow, purple, cyan, white, turn off.</p>
	<p>Servo Drive Function: set the angle of servo connected to S1,S2, S3. Port: S1, S2, S3 Angle: 0~180° (Recommend not exceed 170°)</p>






	<p>Read Line-tracking Sensor</p> <p>Function: read the value of the six line-tracking sensors on Maqueen Plus.</p> <p>When a black line is detected, the line-tracking sensor indicator will be on, and the sensor outputs 1. Otherwise, the indicator turns off, output 0.</p> <p>Sensor: L1, L2, L3, R1, R2, R3</p> <p>Return: black 1, white 0</p>
	<p>Read the Received IR Value</p> <p>Function: read the value received by onboard IR sensor. It uses the NEC IR protocol, and the returned value has been converted into decimal data type.</p> <p>Return: decimal integer(Read the last two digits of the hexadecimal key value of the remote control, and convert it into a decimal number.)</p> <p>Protocol: NEC</p>
	<p>On IR Received Block(Triggered by an event)</p> <p>Function: when an IR data received, save it into the variable message, and run the codes inside this block.</p> <p>Data Type: decimal integer(Read the last two digits of the hexadecimal key value of the remote control, and convert it into a decimal number.)</p> <p>Protocol: NEC</p>
	<p>PID Switch</p> <p>Function: set PID for motor driving. Turn on PID to adjust the speed and torque of the motor in real time. When PID is enabled, the motor offers accurate speed and large torque even at low speed. But there is about 50ms delay for PID adjustment, so it may not be suitable for high real-time control.</p> <p>Can be set: on, off</p>
	<p>Motor Speed Compensation</p> <p>Function: adjust speed difference caused by driving roads, wheels and motor parameters in PID mode.</p> <p>Motor: left, right</p> <p>Speed Range: 0~255 (This value is not the actual speed, it corresponds to 0 ~ 1 revolution, and the maximum compensation value is 1 RPM)</p>






	<p>Read Motor Actual Speed Function: the hall sensor installed on the end part of Maqueen Plus that can detect motor speed in real-time. Motor: left, right Return Value Range: 0~255</p>
	<p>Read Motor Direction Function: the hall sensor installed on the end part of Maqueen Plus that can detect motor speed in real-time. Motor: left, right Return Value Range: 0 stop; 1 forward; 2 back</p>
	<p>Read Grayscale of Line-tracking Sensor Function: detect the grayscale of a black line. Set different gray segments on a routine to make Maqueen Plus execute various instruction, like slowing down, stopping, etc. Sensor: L1 L2 L3 R1 R2 R3 Return Value Range: 0~4095</p>
	<p>Read Distance from Ultrasonic Sensor Function: Maqueen Plus is equipped with URM10 ultrasonic sensor for detecting distance. It offers 5cm~300cm detection range , and 1cm~3cm error. It will be more accurate when detection distance is in 20cm~80cm. The return value will be 0 when over 300cm. Option: connect T and E of the sensor as the same with the software setting. Detection Range: 5cm~300cm</p>

Huskylens AI Camera Block Description

	<p>I2C Init Function: a necessary for initializing I2C communication protocol. This block only needs to run once at the start of main program. If the communication fails, the micro:bit Matrix will show “×”, otherwise, it displays “√” .</p>
	<p>Functions Switch Function: set the working mode of Huskylens. It only needs to run once at the start of main program most of time. The function selection should be the same as the hardware connection. There are 6 working modes: 1. Face Recognition</p>

	<ol style="list-style-type: none"> 2. Object Tracking 3. Object Recognition 4. Line Tracking 5. Color Recognition 6. Tag Recognition
	<p>Request data once from Huskylens</p> <p>Function: a necessary block for HuskyLens. It is usually used in a loop and can be called multiple times. Execute once to read data from Huskylens, such as value of X, Y or Z and ID data.</p>
	<p>Read total number of IDs that Huskylens have learned</p> <p>Function: read the total number of objects Huskylens learned and use it as a variable, return data of unsigned integer.</p>
	<p>Judge if HuskyLens detected object and box or arrow appears on the screen</p> <p>Function: a judgment statement to determine whether HuskyLens detects a object, there will be a box or arrow appearing on the screen if it detects.</p> <p>Object detected, return: true</p> <p>Object not detected, return: false</p> <p>Option: box, arrow</p>
	<p>Read the value of box near the centre of screen</p> <p>Function: HuskyLens can detect multiple objects at the same time, and this block can be used to read the value of box near the centre of screen with providing several kinds of readings:</p> <p>ID: read the ID number of the box</p> <p>X center: read the x-axis of the centre point of box.</p> <p>Y center: read the y-axis of the centre point of box.</p> <p>Width: read the width of box in pixel.</p> <p>Height: read the height of box in pixel.</p>
	<p>Read the value of arrow near the centre of screen</p> <p>Function: HuskyLens can detect multiple lines in Line-tracking mode, and this block can be used</p>

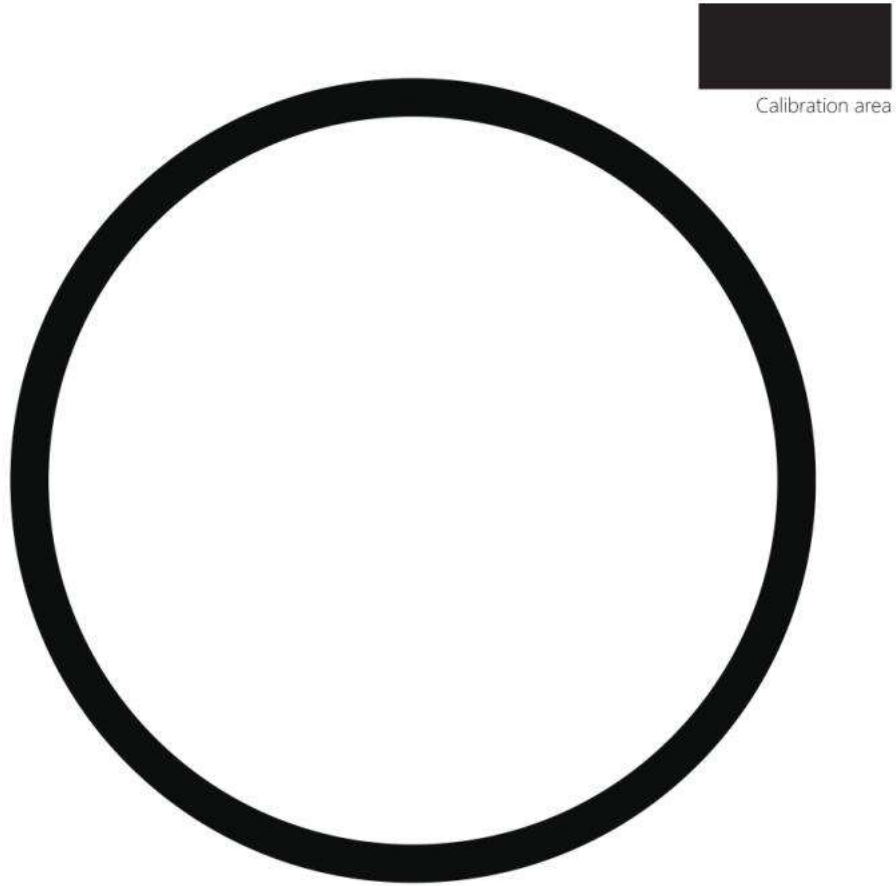
	<p>to read the value of arrow near the centre of screen with providing several readings:</p> <p>ID: read the ID number of arrow</p> <p>X Start: read the X-axis of the starting point of arrow.</p> <p>Y Start: read the Y-axis of the starting point of arrow.</p> <p>X End: read the X-axis of the endpoint of arrow.</p> <p>Y End: read the Y-axis of the endpoint of arrow.</p>
	<p>Judge if the detected object has been learned</p> <p>Function: determine whether an object has been learned when HuskyLens detected multiple objects so as to avoid causing chaos during data calling.</p> <p>Object learned, return: true</p> <p>Object not learned, return: false</p>
	<p>Judge if a specific learned ID appears on the screen.</p> <p>Function: HuskyLens can detect and learn multiple object IDs, and record them with ID numbers. This block is used to determine if a learned object ID appears on the screen.</p> <p>Appeared in the screen, return: true</p> <p>Not appeared on the screen, return: false</p>
	<p>Read box parameter of a specific ID</p> <p>Function: Huskylens can store different objects with ID numbers when it detected multiple objects. This block is used to read box parameter of a specific ID.</p> <p>Parameter option: X center, Y center, Width, Height (Unit: pixel)</p>
	<p>Read arrow parameter of a specific ID</p> <p>Function: Huskylens can store different objects with ID number when it detected multiple objects. This block is used to read arrow parameter of a specific ID.</p> <p>Parameter option: X start, Y start, X end, Y end (Unit: pixel)</p>
	<p>Read total number of recognized arrow or box on the screen</p> <p>Function: read the total number of recognized objects on the screen, usually used in object recognition mode.</p>

	Option: arrow, box
	<p>Read box parameter of a specific serial number on the screen.</p> <p>Function: objects are recognized in order, and this block can be used to read box parameter of a specific serial number, for instance, read the box parameter of the second recognized object.</p> <p>Option: ID, X center, Y center, width, Height (Unit: pixel)</p>
	<p>Read arrow parameter of a specific serial number on the screen</p> <p>Function: objects are recognized in order, and this can be used to read arrow parameter of a specific serial number, for instance, read the arrow parameter of the second recognized object.</p> <p>Option: ID, X start, Y start, X end, Y end(Unit: pixel)</p>
	<p>Read total number of box or arrow of a specific ID on the screen</p> <p>Function: read the total number of the recognized objects on the screen. For example, count how many cars are in the screen when it learned the car.</p> <p>Option: arrow, box</p>
	<p>Read box parameter of a specific serial number range on the screen</p> <p>Function: objects are recognized in order, and this block can be used to box parameter of a specific serial number range, for instance, to read box parameter of the second to the fifth recognized objects.</p> <p>Option: ID, X center, Y center, width, height (Unit: pixel)</p>
	<p>Read arrow parameter of a specific serial number range on the screen</p> <p>Function: objects are recognized in order, and this block can be used to arrow parameter of a specific serial number range, for instance, to read arrow parameter of the second to the fifth recognized objects.</p> <p>Option: ID, X start, Y start, X end, Y end (Unit: pixel)</p>

Project 1: Line follower moving along a circle

1-1 Introduction

Turn Maqueen Plus into a line follower and program it to move along a circle.



1-2 Program Link

https://makecode.microbit.org/_Mz5aDj3dp92w

1-3 Example Code

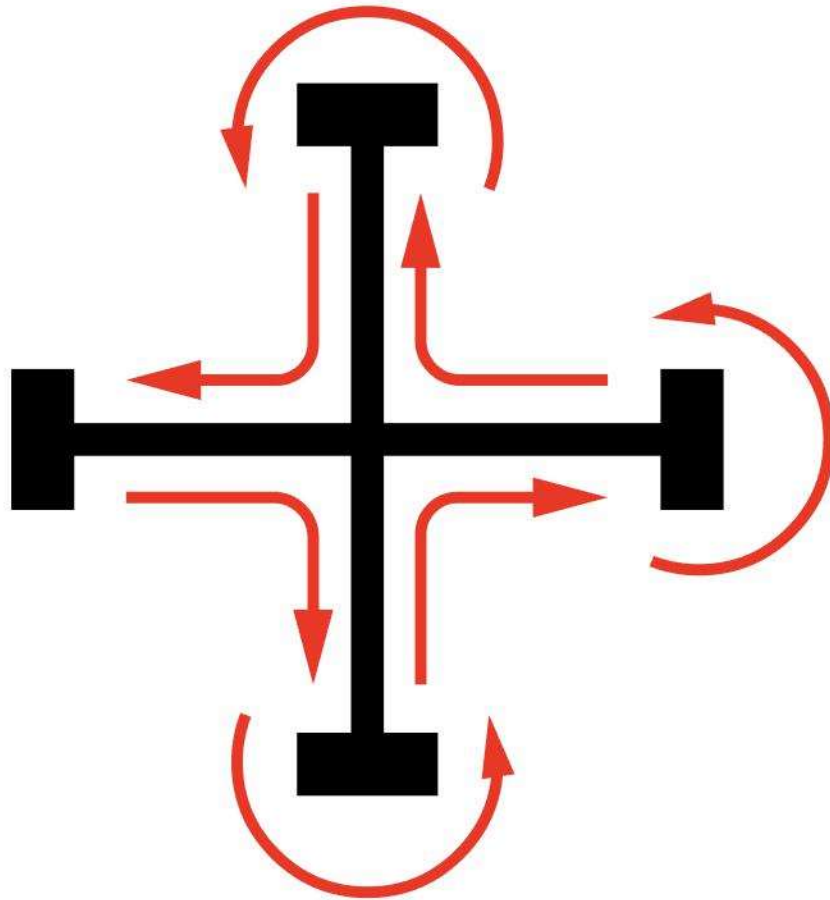
```
on start
  initialize via I2C until success
  PID switch OFF

forever
  if read patrol sensor L1 = 1 and read patrol sensor R1 = 1 then
    Motor ALL direction CW speed 70
  if read patrol sensor L1 = 1 and read patrol sensor R1 = 0 then
    Motor left direction CW speed 20
    Motor right direction CW speed 70
  if read patrol sensor L1 = 0 and read patrol sensor R1 = 1 then
    Motor left direction CW speed 70
    Motor right direction CW speed 20
```


Project 2: Line follower moving along a cross line

2-1. Introduction

Program Maqueen Plus drive along the cross line on the map. 4 line-tracking sensors will be used in this project.



2-2 Program Link: https://makecode.microbit.org/_Kfw1qqUXeVj4

2-3 Example Code:

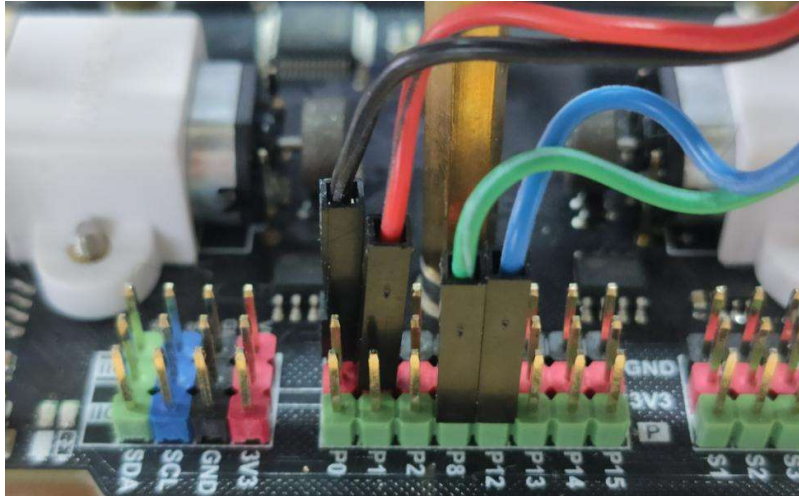
```

forever
  if << read patrol sensor L1 >> = << 1 >> and << read patrol sensor R1 >> = << 1 >> then
    Motor ALL direction CW speed 40
  else
    if << read patrol sensor L1 >> = << 1 >> and << read patrol sensor R1 >> = << 0 >> then
      Motor left direction CW speed 0
      Motor right direction CW speed 40
    +
    if << read patrol sensor L1 >> = << 0 >> and << read patrol sensor R1 >> = << 1 >> then
      Motor left direction CW speed 40
      Motor right direction CW speed 0
    +
    if << read patrol sensor L1 >> = << 0 >> and << read patrol sensor R1 >> = << 0 >> then
      Motor right direction CCW speed 40
      Motor left direction CW speed 40
    +
    if << read patrol sensor L2 >> = << 1 >> and << read patrol sensor R2 >> = << 1 >> then
      Motor right direction CCW speed 40
      Motor left direction CW speed 40
    +
  
```

Project 3: Obstacle Avoidance Robot

3-1 Introduction

The ultrasonic sensor constantly detects the distance between the Maqueen Plus and obstacle ahead in moving, when the distance is smaller than 20cm, Maqueen Plus randomly turns left or right to avoid the obstacle. Connect the ultrasonic sensor to P8(green wire) and P12(blue wire), just corresponding to the port setting in the program. The red wire should be connected to a 3.3V port, and the black one to a GND port.



3-2 Program Link

https://makecode.microbit.org/_bD150m79X8w2

3-3 Example Code

```
on start
  initialize via I2C until success
  PID switch OFF

forever
  set U to read ultrasonic sensor T P8 E P12 cm
  if U < 20 and U ≠ 0 then
    set direction to pick random true or false
    if direction = true then
      Motor left direction CW speed 100
      Motor right direction CW speed 0
      pause (ms) 1000
    else
      Motor left direction CW speed 0
      Motor right direction CW speed 100
      pause (ms) 1000
  else
    Motor ALL direction CW speed 100
```

Project 4: IR-controlled Maqueen Plus

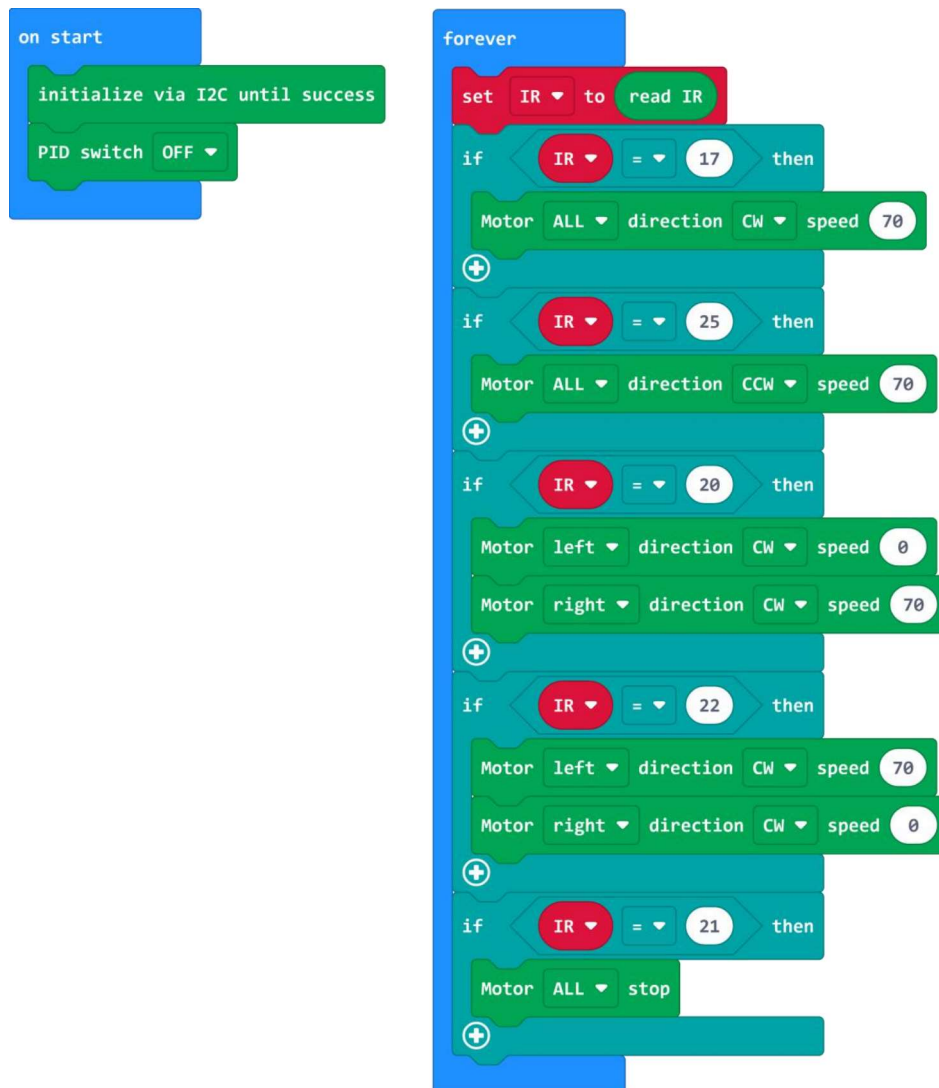
4-1 Introduction

Use the keys 2, 4, 6, 8 and 5 on the remote controller to operate Maqueen Plus.

4-2 Program Link

https://makecode.microbit.org/_ccr5CCg62VbC

4-3 Example Code

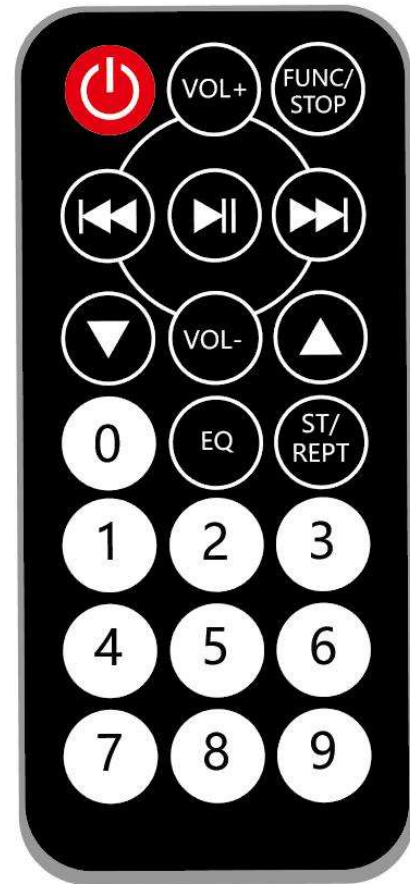


```
on start
  initialize via I2C until success
  PID switch OFF

forever
  set IR to read IR
  if IR = 17 then
    Motor ALL direction CW speed 70
  +
  if IR = 25 then
    Motor ALL direction CCW speed 70
  +
  if IR = 20 then
    Motor left direction CW speed 0
    Motor right direction CW speed 70
  +
  if IR = 22 then
    Motor left direction CW speed 70
    Motor right direction CW speed 0
  +
  if IR = 21 then
    Motor ALL stop
  +
```

4-4 Remote Controller Key Value List

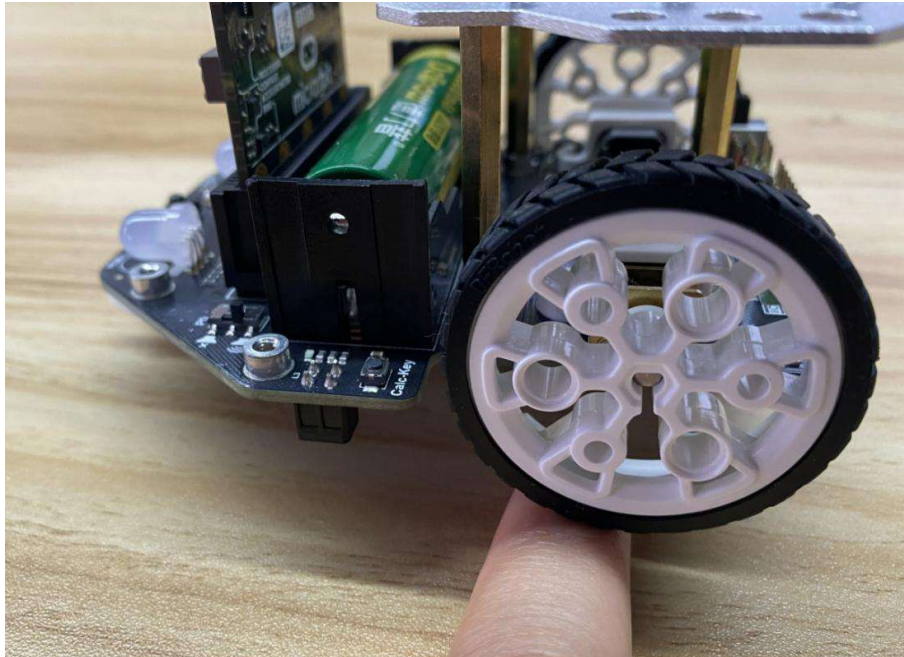
Key	Value(In hexadecimal)	Value(In decimal)
Red Key	0xff00	0
VOL+	0xfe01	1
FUNC/STOP	0xfd02	2
Left Arrow	0xfb04	4
Pause	0xfa05	5
Right Arrow	0xf906	6
Down Arrow	0xf708	8
VOL-	0xf609	9
Up Arrow	0xf50a	10
0	0xf30c	12
EQ	0xf20d	13
ST/REPT	0xf10e	14
1	0xef10	16
2	0xee11	17
3	0xed12	18
4	0xeb14	20
5	0xea15	21
6	0xe916	22
7	0xe718	24
8	0xe619	25
9	0xe51a	26



Project 5: PID Control for Maqueen Plus

5-1 Introduction

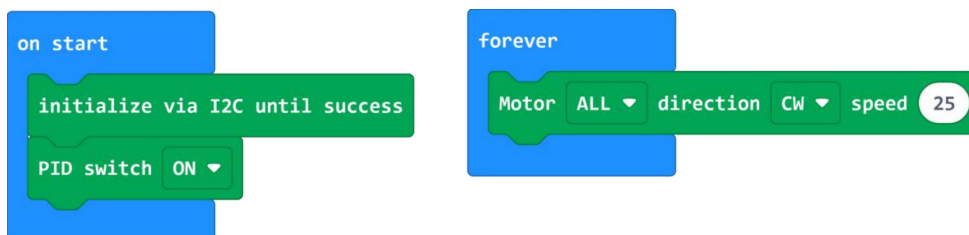
PID can accurately adjust the speed of the two motors and guarantee enough torque at different speeds. Maqueen Plus comes with an on-board encoder and PID control function, which can adjust the torque and speed of a motor in real-time. Download the program, and try letting Maqueen Plus climb across some small obstacles like finger, eraser, etc.



5-2 Program Link

https://makecode.microbit.org/_YxpKywbJxakH

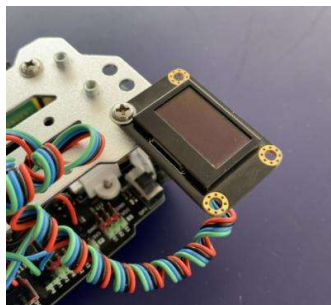
5-3 Example Code



Project 6: Speed up and Slow down

6-1 Introduction

Maqueen Plus constantly goes faster until the speed reaches 150, then gradually slowing down. When its speed is small than 20, stop moving. Meanwhile, the current speed will be displayed on the OLED screen. Enable PID function to control the speed accurately.



6-2 Program Link

https://makecode.microbit.org/_6bDYxchJk9Lk

6-3 Example Code

```
on start
  initialize via I2C until success
  PID switch ON
  INIT_oled
  set speed to 5
  set label to 1

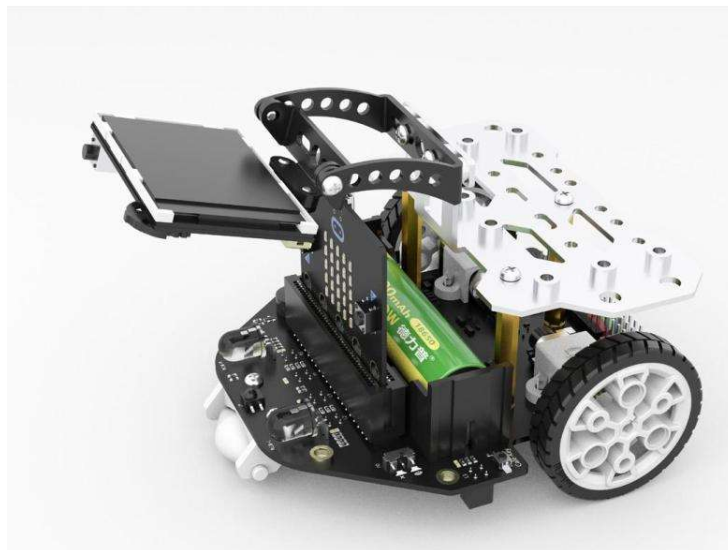
forever
  OLED show line 0 number read Motor left speed
  OLED show line 1 number read Motor left speed

forever
  if speed < 150 and label = 1 then
    Motor ALL direction CW speed speed
    change speed by 5
    pause (ms) 200
  else
    set label to 0
    if speed > 20 and label = 0 then
      change speed by -5
      Motor ALL direction CCW speed speed
      pause (ms) 200
    if speed ≤ 20 then
      Motor ALL stop
```

Project 7: Huskylens AI Camera - Line Tracking

7-1 Introduction

Let Maqueen Plus work with Huskylens camera. The camera recognizes the black line, then Maqueen Plus drives along that road. Download program into micro:bit, adjust the angle of the camera, put the Maqueen Plus on the line, and power on.



7-2 Program Link

https://makecode.microbit.org/_W5fdWb8xea15

7-3 Example Code

```
on start
  initialize via I2C until success
  HuskyLens initialize via I2C until success
  HuskyLens change Line Tracking algorithm until success

forever
  HuskyLens request once enter the result
  if HuskyLens get from result ID 1 arrow in picture? then
    if HuskyLens get from result ID 1 arrow parameter xTarget > 140 and HuskyLens get from result ID 1 arrow parameter xTarget < 180 then
      Motor ALL direction CW speed 60
    if HuskyLens get from result ID 1 arrow parameter xTarget < 140 then
      Motor left direction CW speed 30
      Motor right direction CW speed 90
    if HuskyLens get from result ID 1 arrow parameter xTarget > 180 then
      Motor left direction CW speed 90
      Motor right direction CW speed 30
    else
      Motor ALL stop
```

Project 8: HuskyLens AI Camera - Tail After

8-1 Introduction

Two Maqueen Plus cars will be used here. Let the first Maqueen car move forward freely, the second one tails after it using a HuskyLens AI camera.

8-2 Program Link

https://makecode.microbit.org/_Y4ai3y2jvdEh

8-3 Example Code

```
on start
  initialize via I2C until success
  HuskyLens initialize via I2C until success
  HuskyLens change Color Recognition algorithm until success

forever
  HuskyLens request once enter the result
  if HuskyLens get from result ID 1 box in picture? then
    if HuskyLens get from result ID 1 box parameter object width < 120 then
      if HuskyLens get from result ID 1 box parameter x coordinates > 140 and HuskyLens get from result ID 1 box parameter x coordinates < 180 then
        Motor ALL direction CW speed 60
      if HuskyLens get from result ID 1 box parameter x coordinates < 140 then
        Motor left direction CW speed 30
        Motor right direction CW speed 90
      if HuskyLens get from result ID 1 box parameter x coordinates > 180 then
        Motor left direction CW speed 90
        Motor right direction CW speed 30
      else
        Motor ALL stop
```


Project 9: Huskylens AI Camera - Passing a Traffic Light

9-1 Introduction

Let Huskylens AI learn red and green cards. When it recognizes the green card, the Maqueen Plus car moves forward. When the red card is recognized, the car stops. At the same time, the color recognized is displayed with the RGB LEDs on the Maqueen Plus.

Note: the surrounding environment should not be too complex in case causing misrecognitions.

9-2 Program Link

https://makecode.microbit.org/_e0Y86Pgc8gFg

9-3 Example Code

```
on start
  initialize via I2C until success
  Huskylens initialize via I2C until success
  Huskylens change Color Recognition algorithm until success
  PID switch ON

forever
  Huskylens request once enter the result
  if Huskylens get from result ID 1 box in picture? then
    set ALL color Red
    Motor ALL stop
  if Huskylens get from result ID 2 box in picture? then
    set ALL color Green
    Motor ALL direction CW speed 50
```

Project 10: Maqueen Mechanic - Loader

10-1 Introduction

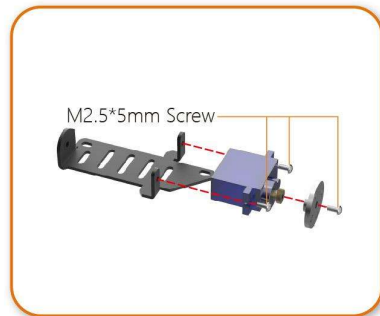
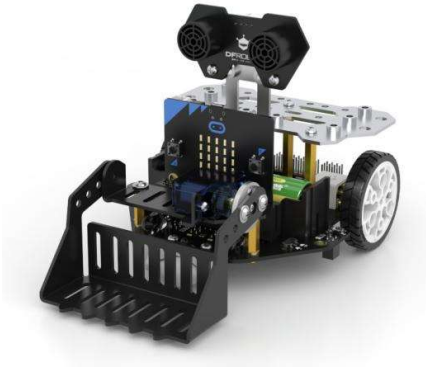
Try to install the Loader accessories on Maqueen Plus and use the remote control handle to operate it.

Necessary accessories: 1. Maqueen remote control handle 2. Maqueen loader accessories 3. Prepare one more micro:bit main board

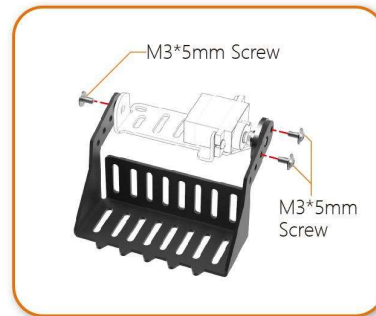
As shown in the figure below:



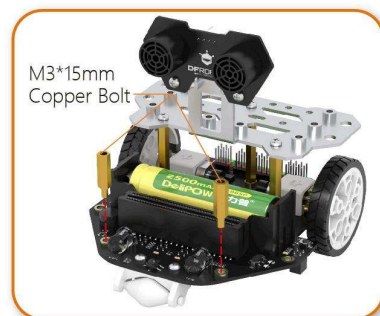
10-2 Assembly



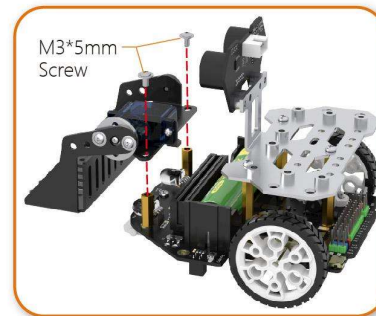
● Step1



● Step2



● Step3



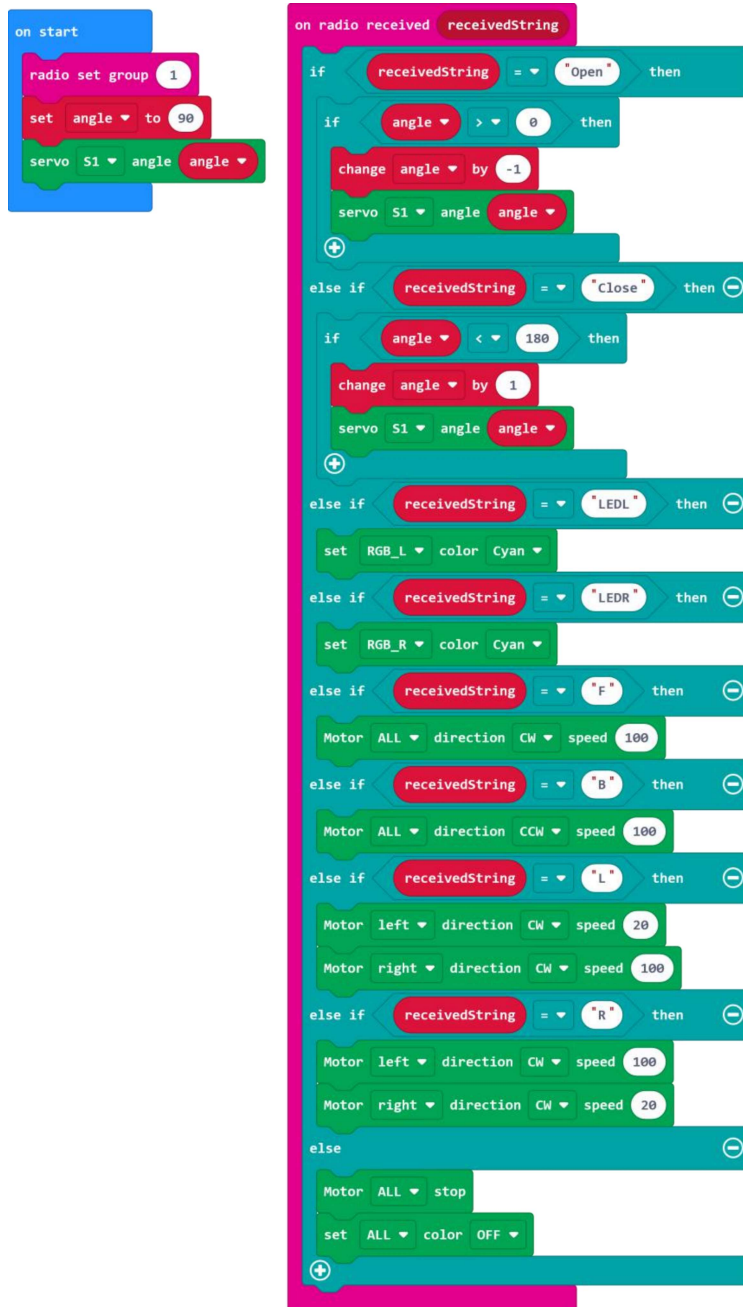
● Step4

10-3 Program Link and Example Code

Set switch quantity for remote-control handle

In this project, the handle is set as the switch quantity, which is used to control the car to move forward, backward, left turn and right. But it can't control the speed. The up and down buttons on the right of the handle control the movement of Maqueen loader, and the left and right buttons control the RGB LEDs to be on and off.

Programs for Maqueen Plus: https://makecode.microbit.org/_MyscR05Vc2tz



```
on start
  radio set group 1
  set angle to 90
  servo S1 angle angle

on radio received receivedString
  if receivedString = "Open" then
    if angle > 0 then
      change angle by -1
      servo S1 angle angle
    else if receivedString = "Close" then
      if angle < 180 then
        change angle by 1
        servo S1 angle angle
      else if receivedString = "LEDL" then
        set RGB_L color Cyan
      else if receivedString = "LEDR" then
        set RGB_R color Cyan
      else if receivedString = "F" then
        Motor ALL direction CW speed 100
      else if receivedString = "B" then
        Motor ALL direction CCW speed 100
      else if receivedString = "L" then
        Motor left direction CW speed 20
        Motor right direction CW speed 100
      else if receivedString = "R" then
        Motor left direction CW speed 100
        Motor right direction CW speed 20
      else
        Motor ALL stop
        set ALL color OFF
```

Program for Remote-control Handle:

https://makecode.microbit.org/_HRWfzpg02Mrv

```

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 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"
  
```

Set analog quantity for remote-control handle

The remote-control handle is set as analog quantity, and then the speed and direction of Maqueen Plus can be controlled at the same time. The more the handle button is pressed, the faster Maqueen Plus will go. The up and down buttons on the right of the handle control the movement of Maqueen loader, and the left and right buttons control the RGB LEDs to be on and off.

Programs for Maqueen Plus: https://makecode.microbit.org/_daYbLRyUtI7

```

当无线接收到数据时运行 receivedString
  如果为 receivedString = "Open" 则
    如果为 angle > 0 则
      以 -1 为幅度更改 angle
      舵机 S1 角度 angle
    否则如果为 receivedString = "Close" 则
      如果为 angle < 180 则
        以 1 为幅度更改 angle
        舵机 S1 角度 angle
      否则如果为 receivedString = "LEDL" 则
        设置 左RGB灯 颜色 红色
      否则如果为 receivedString = "LEDR" 则
        设置 右RGB灯 颜色 红色
      否则
        电机 全部 停止
        设置 全部RGB灯 颜色 关闭

写开机时
  无线设置组 1
  将 angle 设为 90
  舵机 S1 角度 angle

当无线接收到数据时运行 name value
  如果为 name = "F" 则
    电机 全部 方向 正转 速度 映射 value 从低 550 从低 1024 至低 10 至高 255
  否则如果为 name = "B" 则
    电机 全部 方向 反转 速度 映射 value 从低 1 从低 450 至低 255 至高 10
  否则如果为 name = "L" 则
    电机 右侧 方向 正转 速度 映射 value 从低 1 从低 450 至低 255 至高 40
    电机 左侧 方向 正转 速度 20
  否则如果为 name = "R" 则
    电机 左侧 方向 正转 速度 映射 value 从低 550 从低 1024 至低 40 至高 255
    电机 右侧 方向 正转 速度 20
  
```

Program for Remote-control Handle:

https://makecode.microbit.org/_Wmxd6k2Era7z

```
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 < 500 then
      radio send value "1" = analog read pin P2
    else if analog read pin P2 < 450 and analog read pin P1 > 400 and analog read pin P1 < 500 then
      radio send value "2" = analog read pin P2
    else if analog read pin P1 < 450 and analog read pin P2 > 400 and analog read pin P2 < 500 then
      radio send value "3" = analog read pin P1
    else if analog read pin P1 > 550 and analog read pin P2 > 400 and analog read pin P2 < 500 then
      radio send value "4" = analog read pin P1
    else
      radio send string "5"
```