



愛動智教育系統

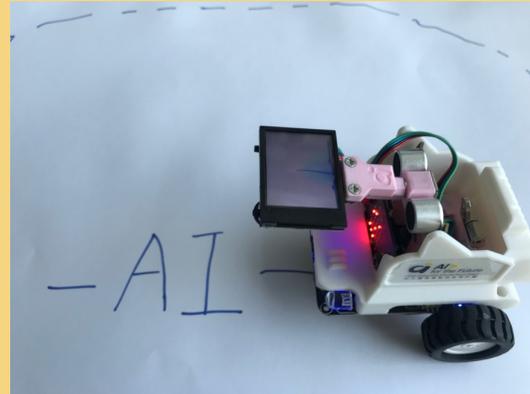
CUHK iCar Experiment Manual
Experiment 2: Line Tracking Experiment

Write Your Own Code

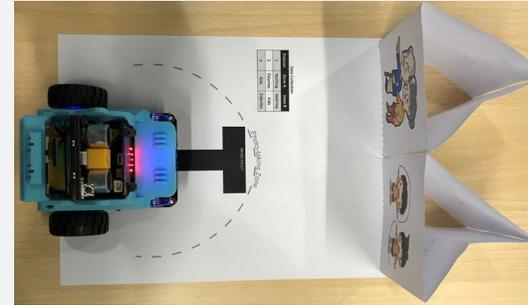
CUHK iCar



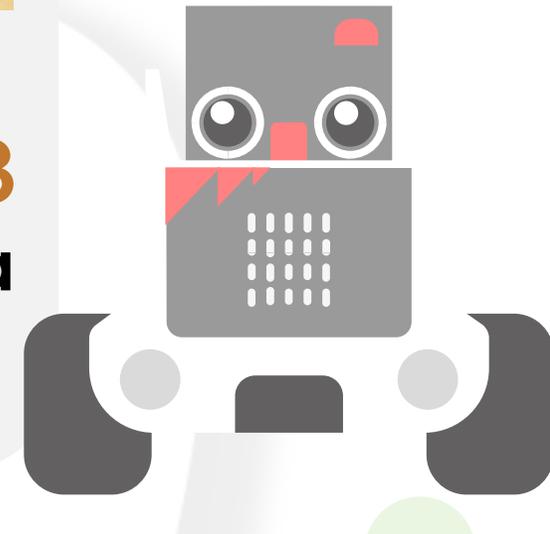
Experiment 1
Face Following



Experiment 2
Line Tracking



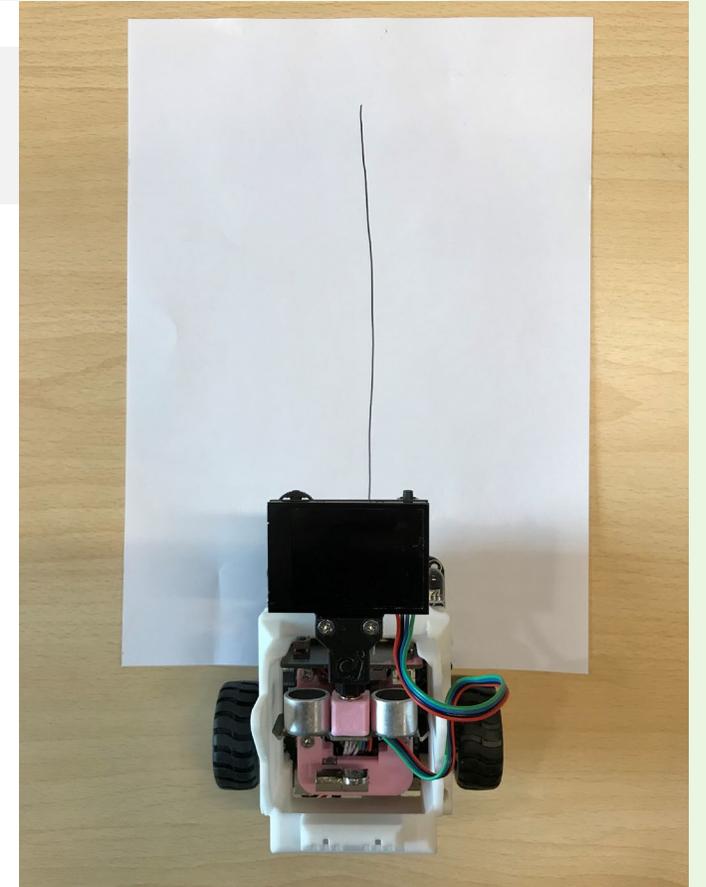
Experiment 3
Moral Dilemma



Line Tracking Experiment

Introduction Of The Experiment

This experiment showcases how the self-driving car tracks a black line.



Download Program To micro:bit

Line_tracking.hex

```
on start
  Hookless initialize I2C until success
  Hookless switch algorithm to Line Tracking
  show icon

function Move_Forward
  iCar Move Forward at speed 50 %

function Turn_Left
  iCar Turn Left at speed 50 %

function Turn_Right
  iCar Turn Right at speed 50 %

function Line_Tracking_Mode
  Hookless request data once and save into the result
  if Hookless check if 2D array is on screen from the result then
    set scanner to Hookless get X endpoint of 2D array from the result
    if scanner < 100 then
      call Turn_Left
    if scanner > 200 and scanner < 220 then
      call Move_Forward
    if scanner > 220 then
      call Turn_Right
    else
      iCar Stop
```

Method 1

Clone the .hex to micro:bit directly

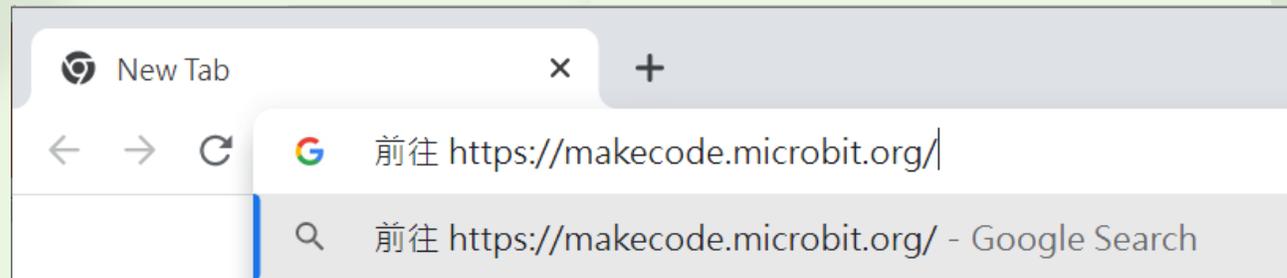
Method 2

Write your own code on MakeCode

Code On MakeCode

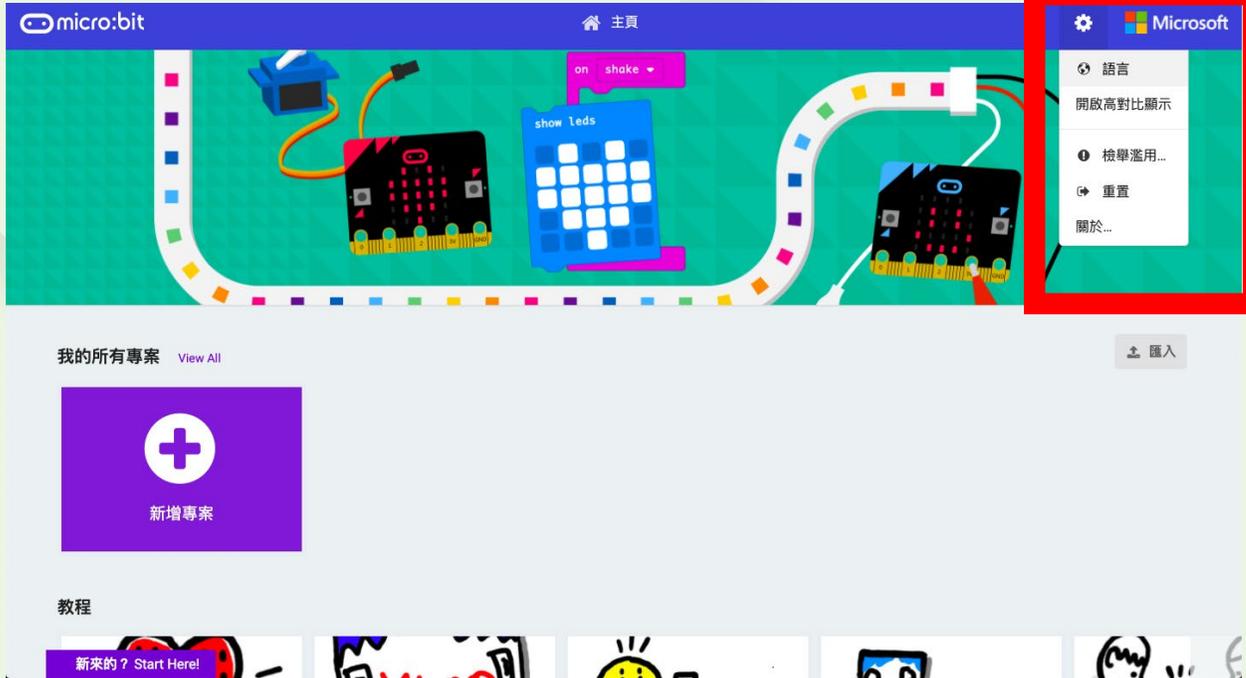


Browse <https://makecode.microbit.org/>





Caution
Please Set **English** As The Language!



1. Click
2. Click

English must be set as the language, otherwise the program may fail.

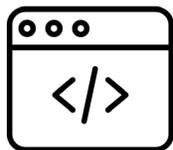
micro:bit 主頁 Microsoft

選擇語言

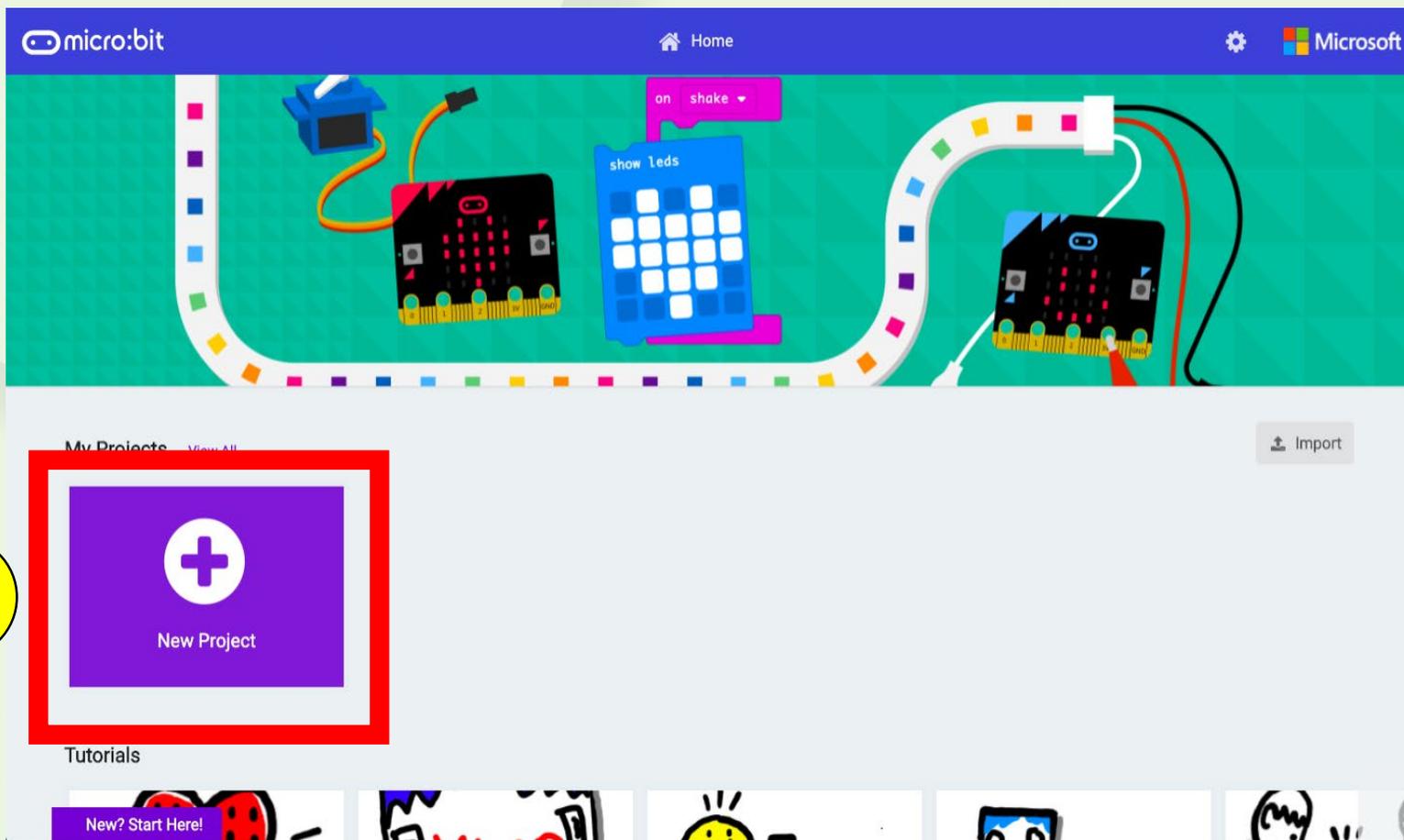
English English	العربية Arabic	български Bulgarian
Čeština Czech	Dansk Danish	Deutsch German
Ελληνικά Greek	Español (España) Spanish (Spain)	Suomi Finnish
Français French	עברית Hebrew	Magyar Hungarian

我的所有專案 匯入

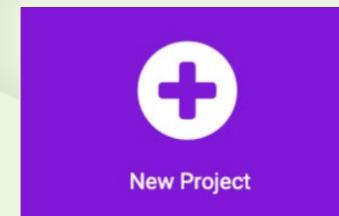
3. Click English



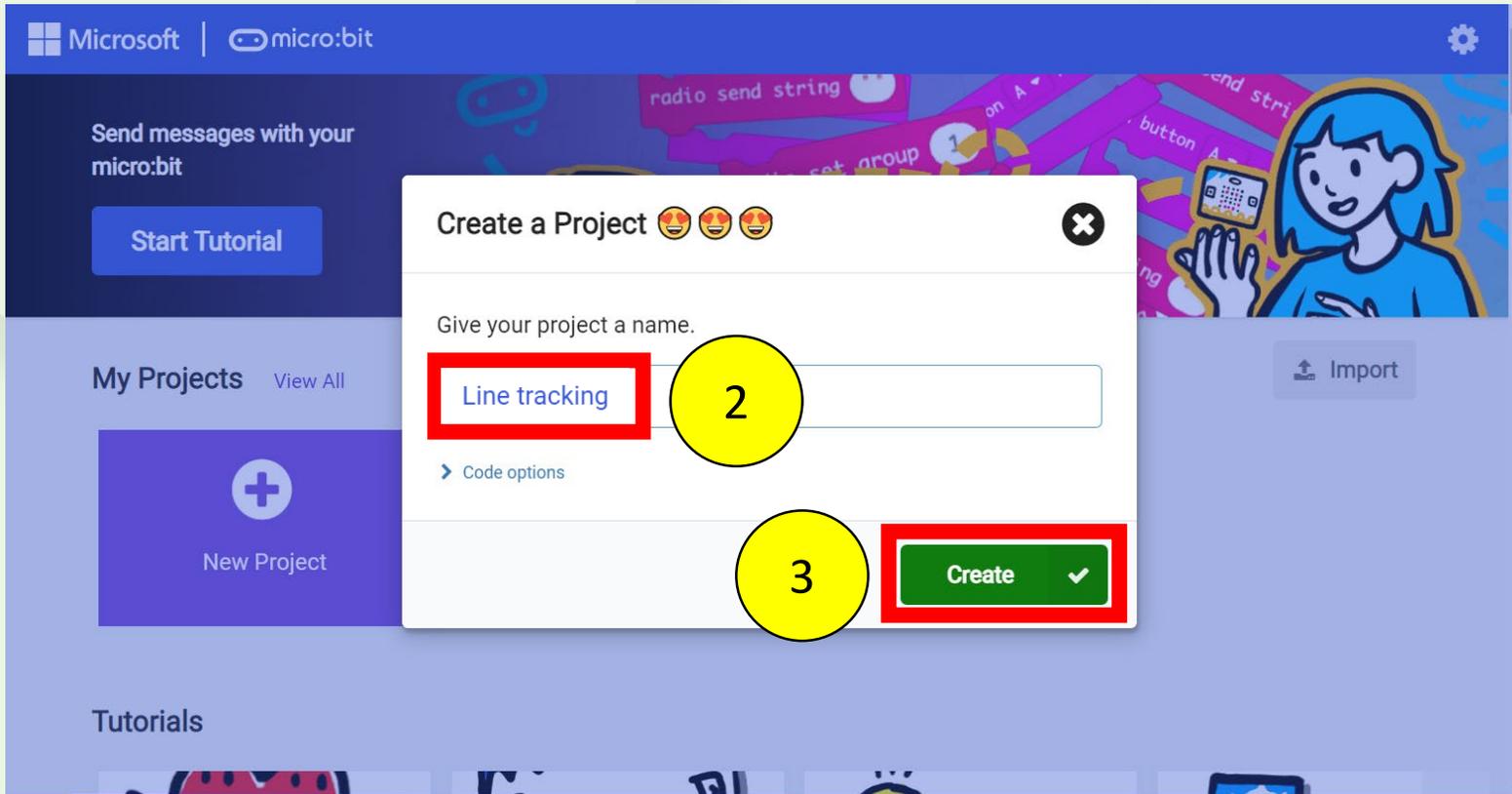
New Project



1. Click New Project

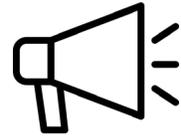


1

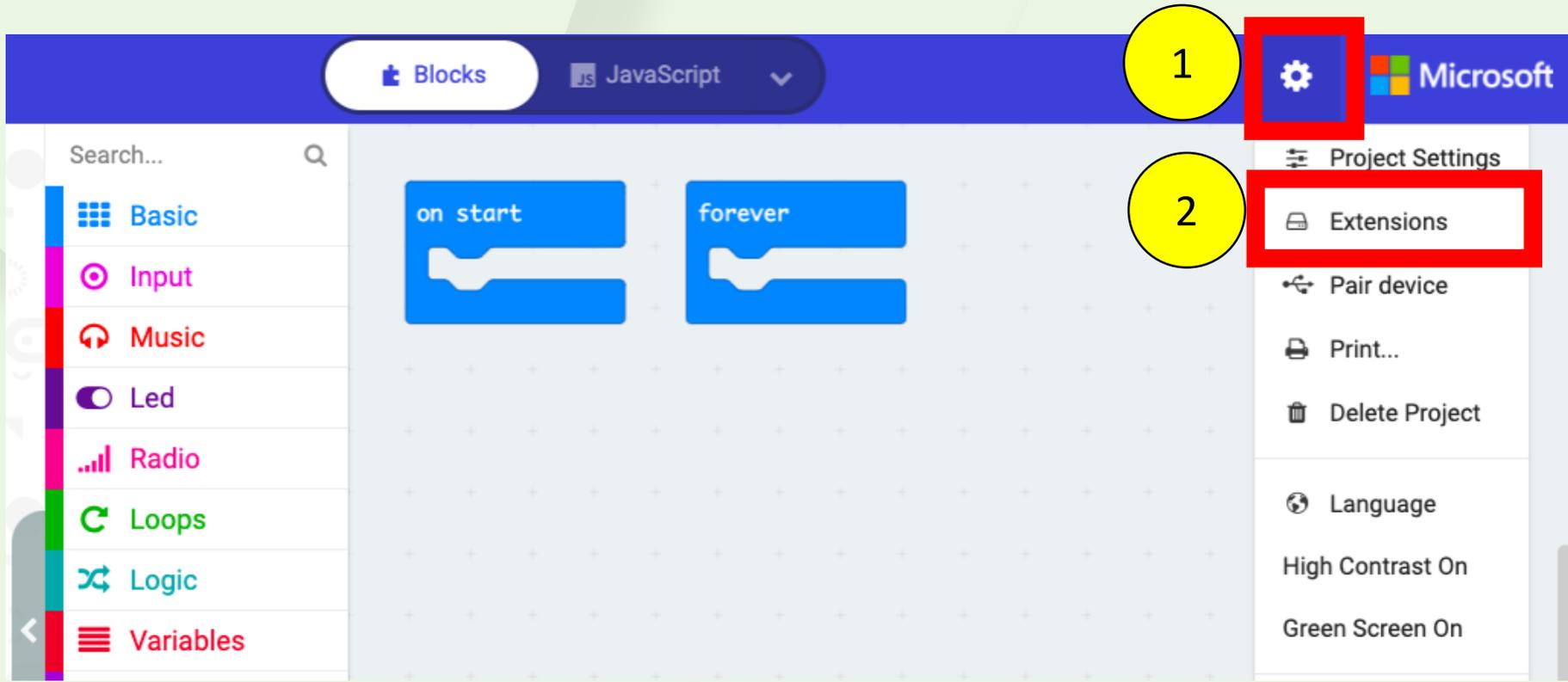


2. Enter “Line tracking”

3. Click 



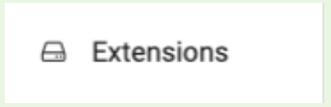
Prepare CUHK-JC-iCar Extension



1. Click

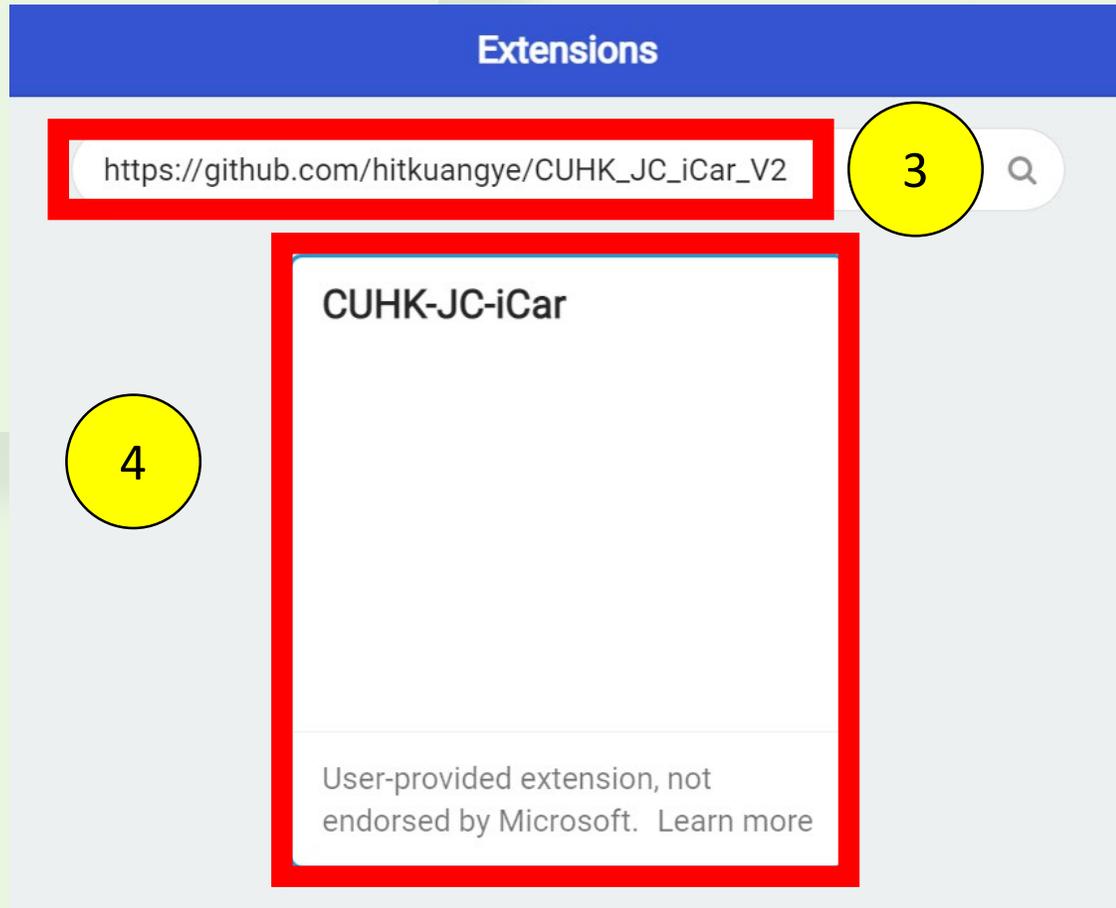


2. Click

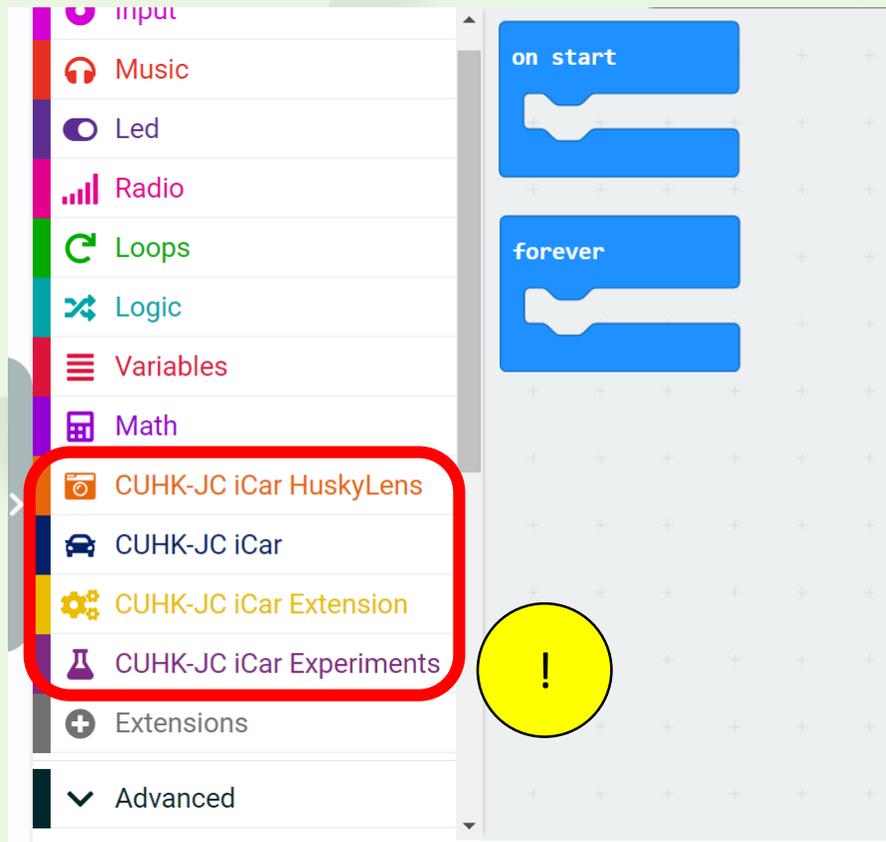


1

2



3. Paste the following link
https://github.com/hitkuangye/CUHK_JC_iCar_V2
4. Click CUHK-JC-iCar extension



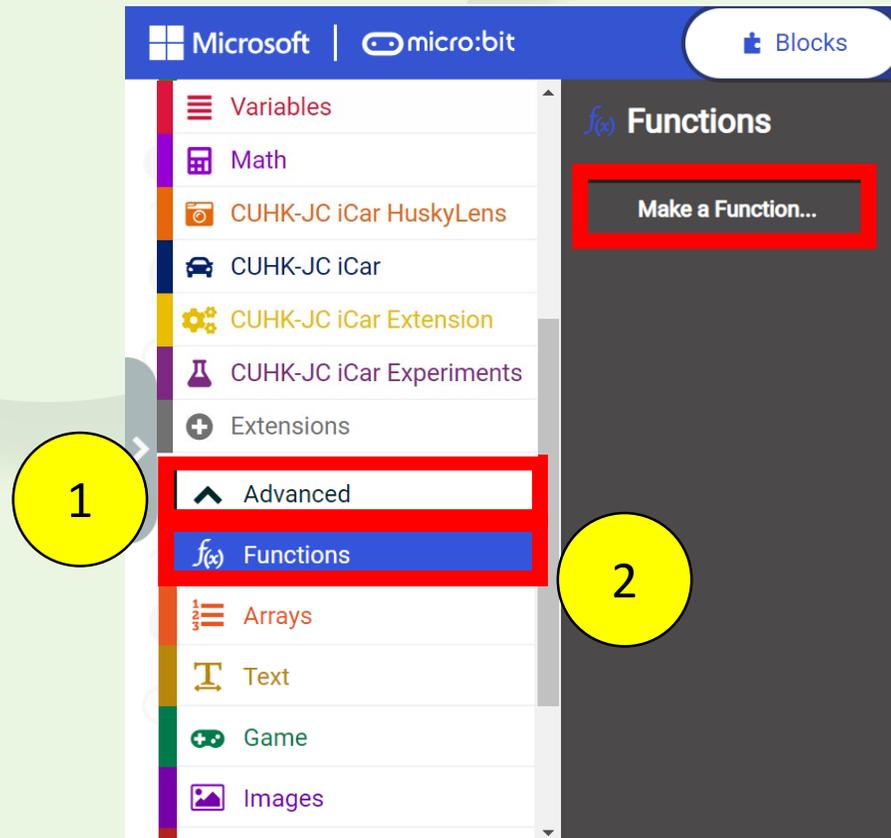
Extensions are included!

Introduction To micro:bit



Before we start programming, let's learn more about micro:bit!

Functions Modules



1. Scroll to the bottom of module list and click 

2. Click 

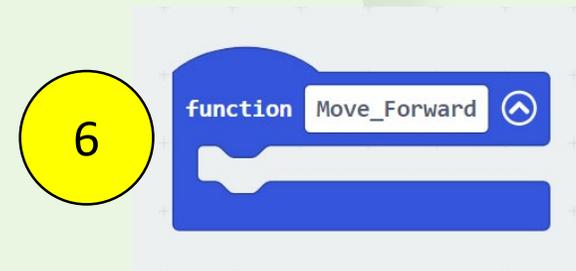
3. Click 



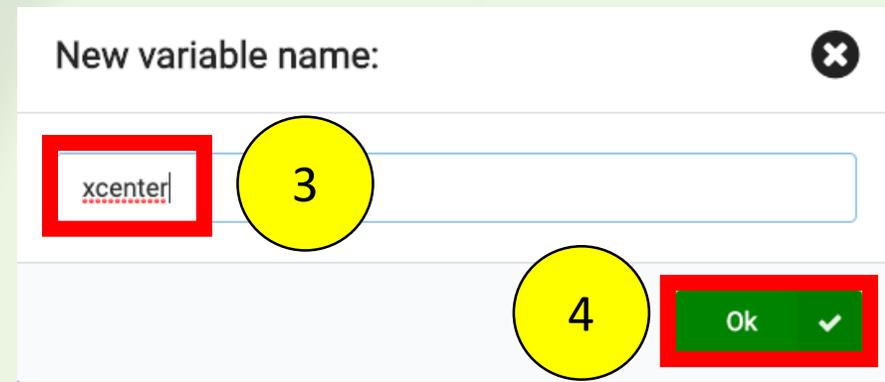
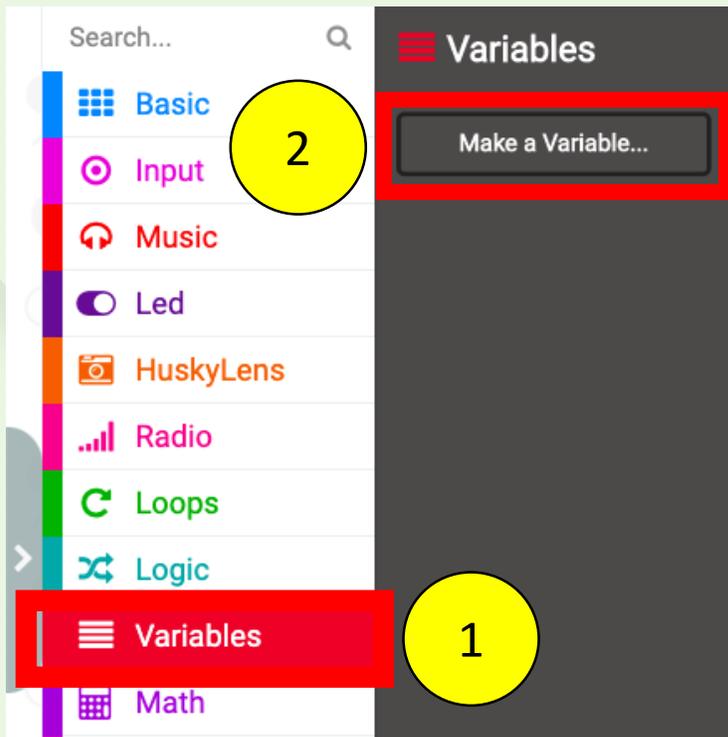
4. Name the function
"Move_Forward"

5. Click 

6. "Move_Forward"
function will appear
on the screen



Variables Modules



1. Click **Variables**
2. Click **Make a Variable...**
3. Name the variable "xcenter"
4. Click **Ok**
5. Programming blocks related to "xcenter" will appear on the list

The Remaining Modules

CUHK-JC iCar HuskyLens

HuskyLens initialize I2C until success

HuskyLens switch algorithm to Face Recognition

HuskyLens request data once and save into the result

HuskyLens check if ID 1 frame is on screen from the result

HuskyLens get X beginning of ID 1 arrow from the result

Basic

show icon

CUHK-JC iCar

iCar Move Forward at speed 1 %

iCar Stop

f(x) Functions

call Move_Forward

Variables

xcenter

set xcenter to 0

Logic

if true then

if true then
else

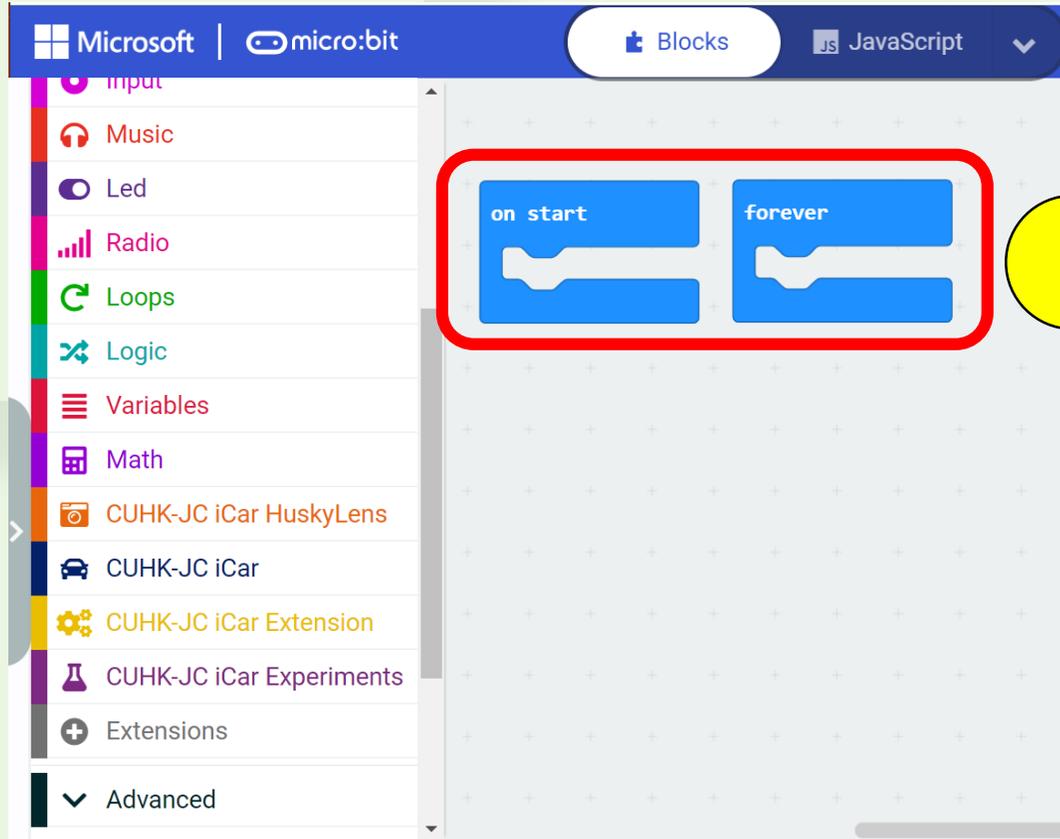
+

0 < 0

and



Let's start coding!

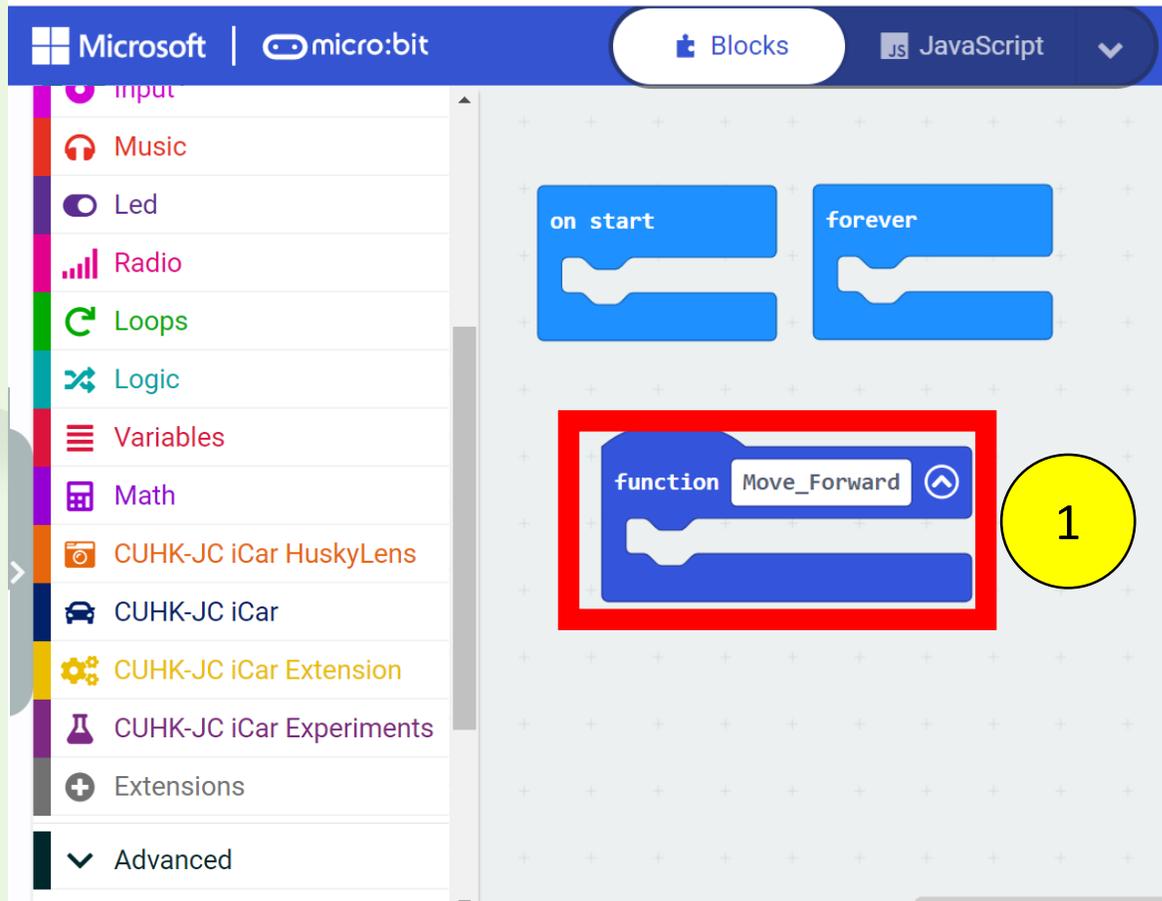


The program will pre-set “on start” and “forever” at the beginning.

We can ignore them for now.
Please don't delete!

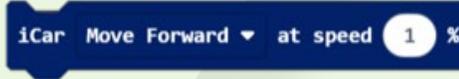


Step 1: Adding The “Move_Forward” Function



1. Add “Move_Forward” function



2. Drag  from  into "Move_Forward" function

3. Inside  set to "Move Forward"

4. Inside  set the speed to "25"

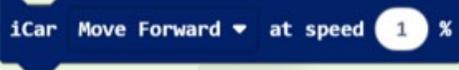


Step 2: Adding The “Turn_Left” Function

1. Add “Turn_Left” function

1



2. Drag  from  into "Turn_Left" function

3. Inside  set to "Turn Left"

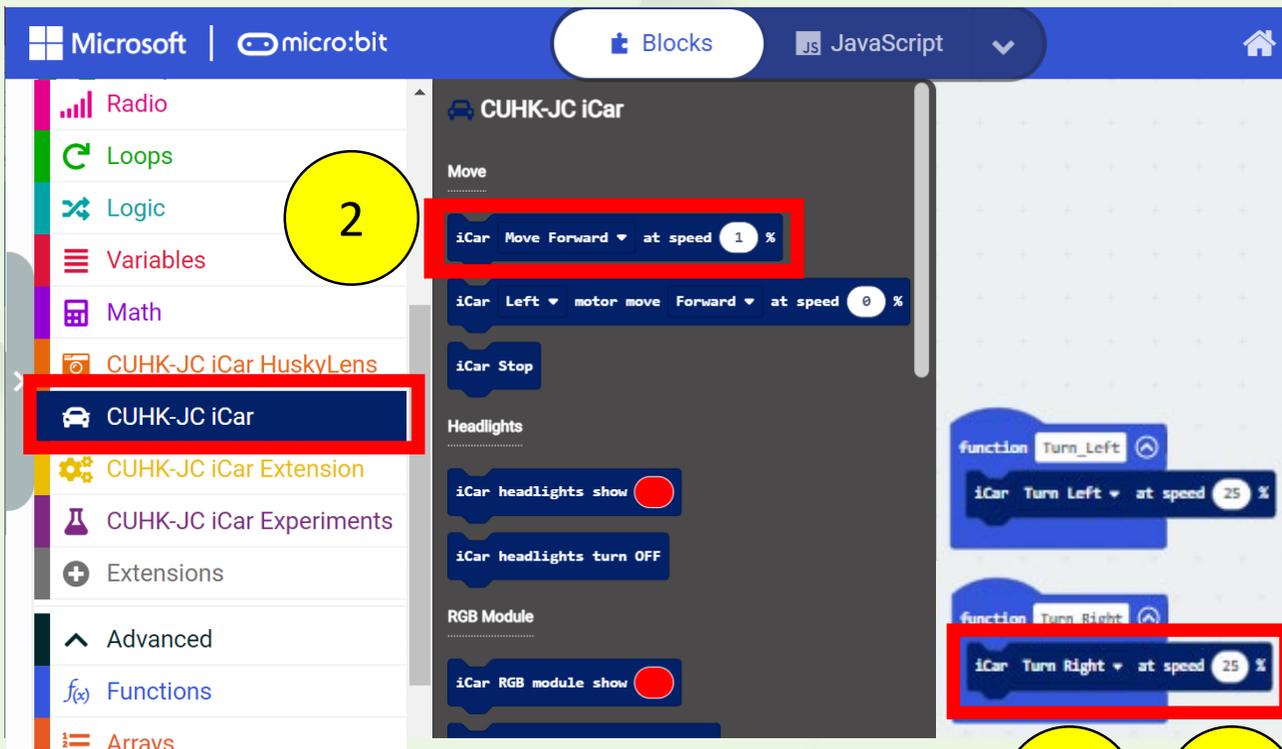
4. Inside  set the speed to "25"



Step 3: Adding The “Turn_Right” Function

The screenshot shows the Microsoft micro:bit IDE interface. The top bar includes the Microsoft logo, the 'micro:bit' text, and tabs for 'Blocks' and 'JavaScript'. A left-hand sidebar lists various categories: Input, Music, Led, Radio, Loops, Logic, Variables, Math, CUHK-JC iCar HuskyLens, CUHK-JC iCar, CUHK-JC iCar Extension, CUHK-JC iCar Experiments, Extensions, and Advanced. The main workspace contains a script starting with 'on start' and 'forever' blocks. Below these are three function blocks: 'function Move_Forward' (iCar Move Forward at speed 25%), 'function Turn_Left' (iCar Turn Left at speed 25%), and 'function Turn_Right' (iCar Turn Right at speed 25%). The 'Turn_Right' block is highlighted with a red rectangular border, and a yellow circle with the number '1' is placed next to it.

1. Add “Turn_Right” function



2. Drag 

from 

into "Turn_Right" function

3. Inside 

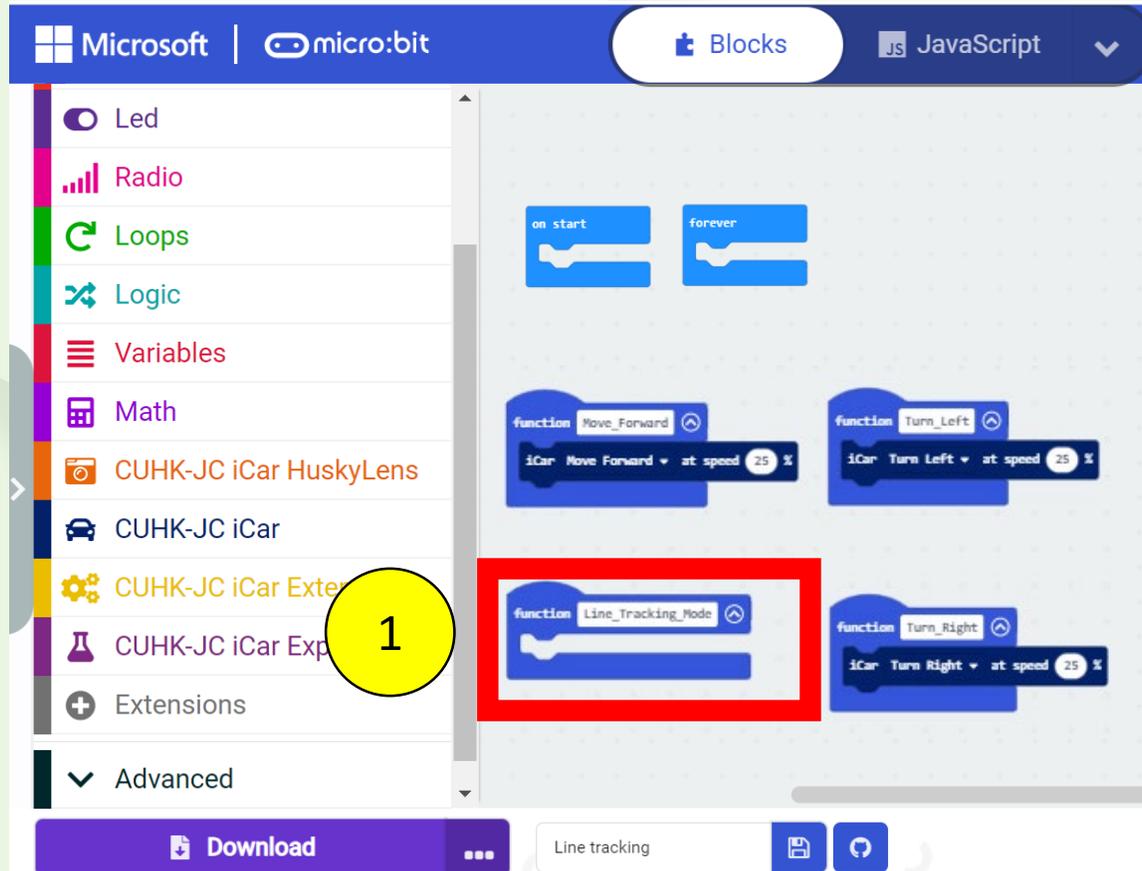
set to "Turn Right"

4. Inside 

set the speed to "25"



Step 4: Adding The “Line_Tracking_Mode” Function



1. Add “Line_Tracking_Mode” function

Microsoft | micro:bit

Blocks JavaScript

Led

Radio

Loops

Logic

Variables

Math

CUHK-JC iCar HuskyLens

CUHK-JC iCar

CUHK-JC iCar Extension

CUHK-JC iCar Exp

Extensions

Advanced

on start

forever

function Move_Forward

iCar Move Forward at speed 25 %

function Turn_Left

iCar Turn Left at speed 25 %

function Line_Tracking_Mode

HuskyLens request data once and save into the result

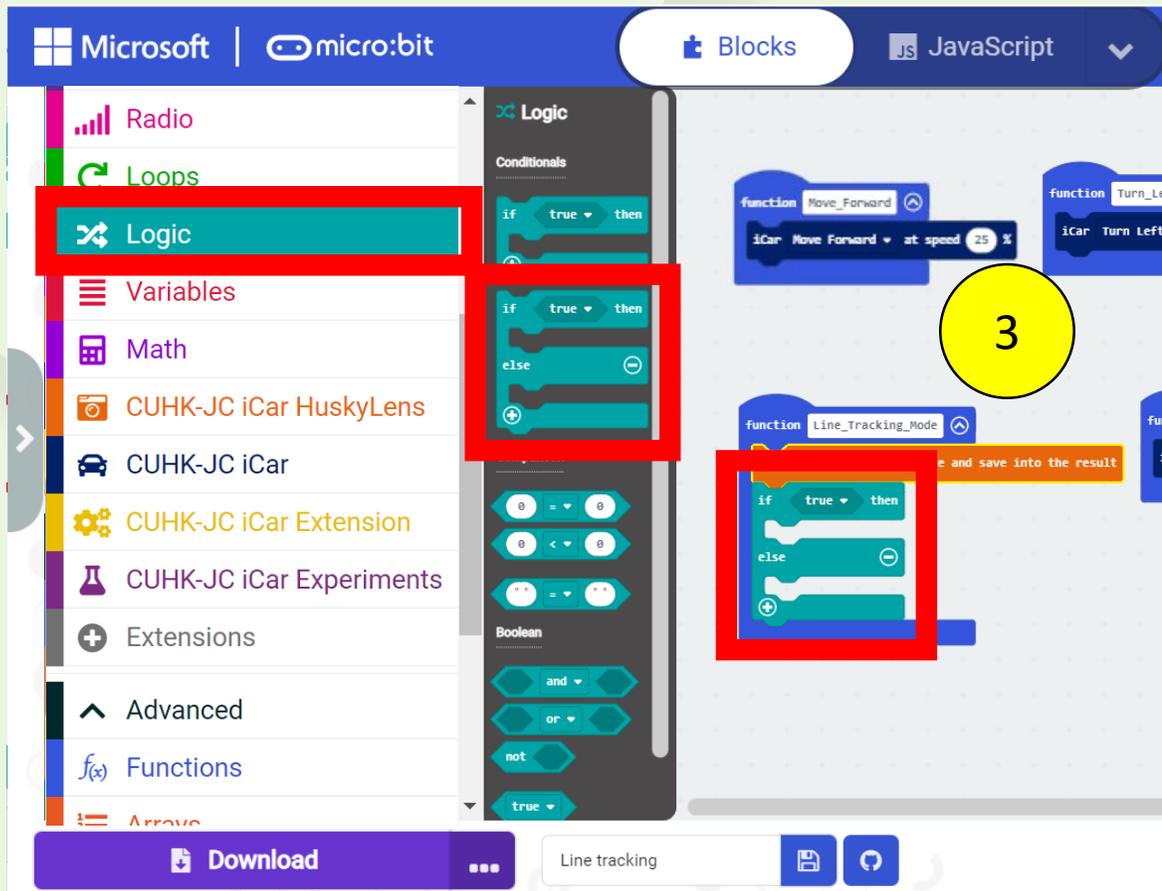
function Turn_Right

iCar Turn Right at speed 25 %

Download

Line tracking

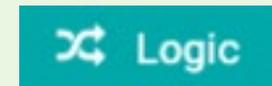
2. Drag **HuskyLens request data once and save into the result**
- from **CUHK-JC iCar HuskyLens**
- into “Line_Tracking_Mode” function



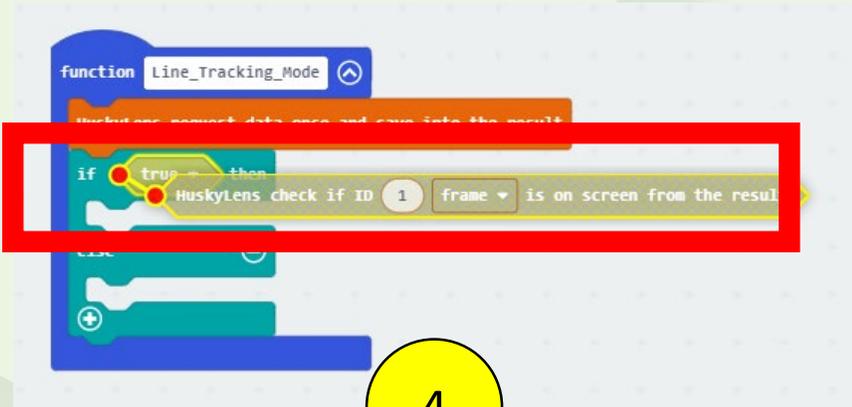
3. Drag



from

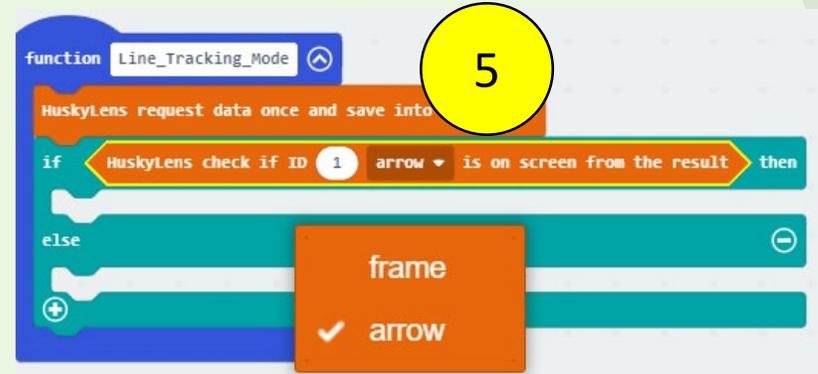
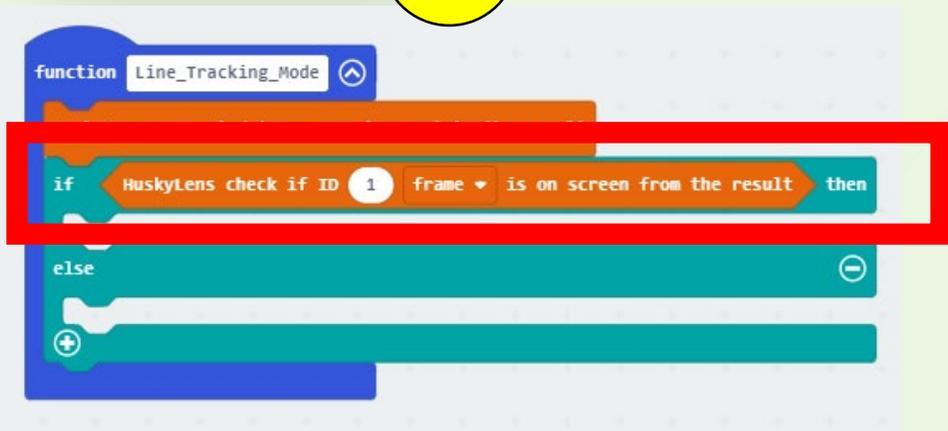


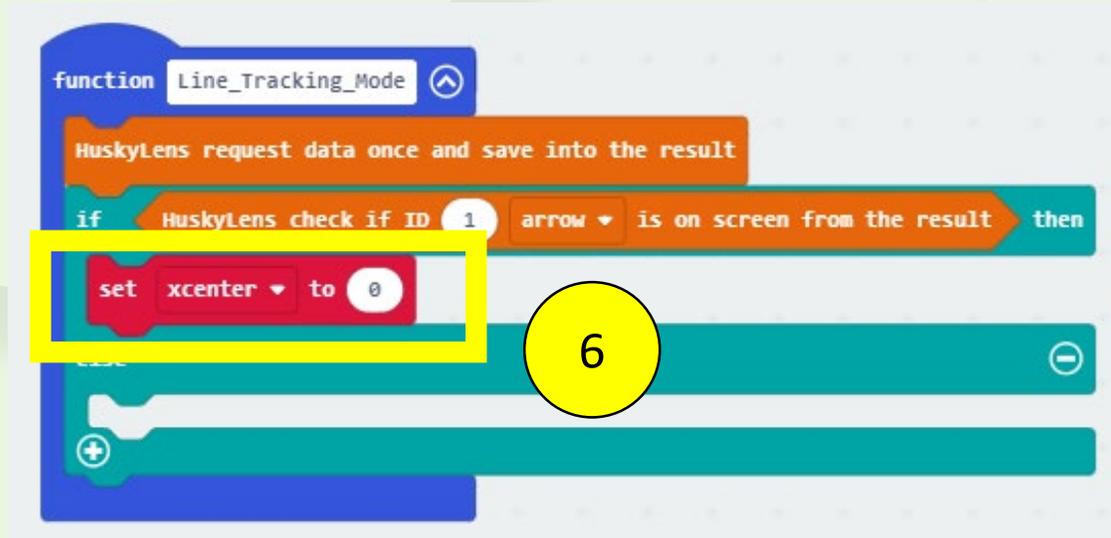
into "Line_Tracking_Mode" function

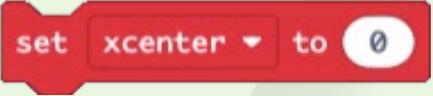


4. Drag **HuskyLens check if ID 1 frame is on screen from the result** from **CUHK-JC iCar HuskyLens** into "if then else"

5. Inside **HuskyLens check if ID 1 frame is on screen from the result** click "arrow"





6. Drag  from  into "if then else"

```
function Line_Tracking_Mode  
  HuskyLens request data once and save into the result  
  if HuskyLens check if ID 1 arrow is on screen from the result then  
    set xcenter to HuskyLens get X beginning of ID 1 arrow from the result  
  else  
  end  
end
```

7

7. Drag HuskyLens get X endpoint of ID 1 arrow from the result from CUHK-JC iCar HuskyLens into set xcenter to 0

```
function Line_Tracking_Mode  
  HuskyLens request data once and save into the result  
  if HuskyLens check if ID 1 arrow is on screen from the result then  
    set xcenter to HuskyLens get X endpoint of ID 1 arrow from the result  
  else  
    HuskyLens get X beginning of ID 1 arrow from the result  
    HuskyLens get Y beginning of ID 1 arrow from the result  
    HuskyLens get X endpoint of ID 1 arrow from the result  
    HuskyLens get Y endpoint of ID 1 arrow from the result  
  end  
end
```

```
function Line_Tracking_Mode  
  HuskyLens request data once and save into the result  
  if HuskyLens check if ID 1 arrow is on screen from the result then  
    set xcenter to HuskyLens get X endpoint of ID 1 arrow from the result  
    if true then  
        
    else  
        
    
endfunction
```

8

8. Drag



from



into "if then else"

```
function Line_Tracking_Mode  
  HuskyLens request data once and save into the result  
  if HuskyLens check if ID 1 arrow is on screen from the result then  
    set xcenter to HuskyLens get X endpoint of ID 1 arrow from the result  
    if xcenter < 100 then  
    else  
  end  
end
```

9

10

11

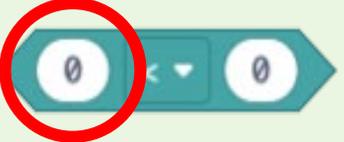
9. Drag 

from 

into "if then"

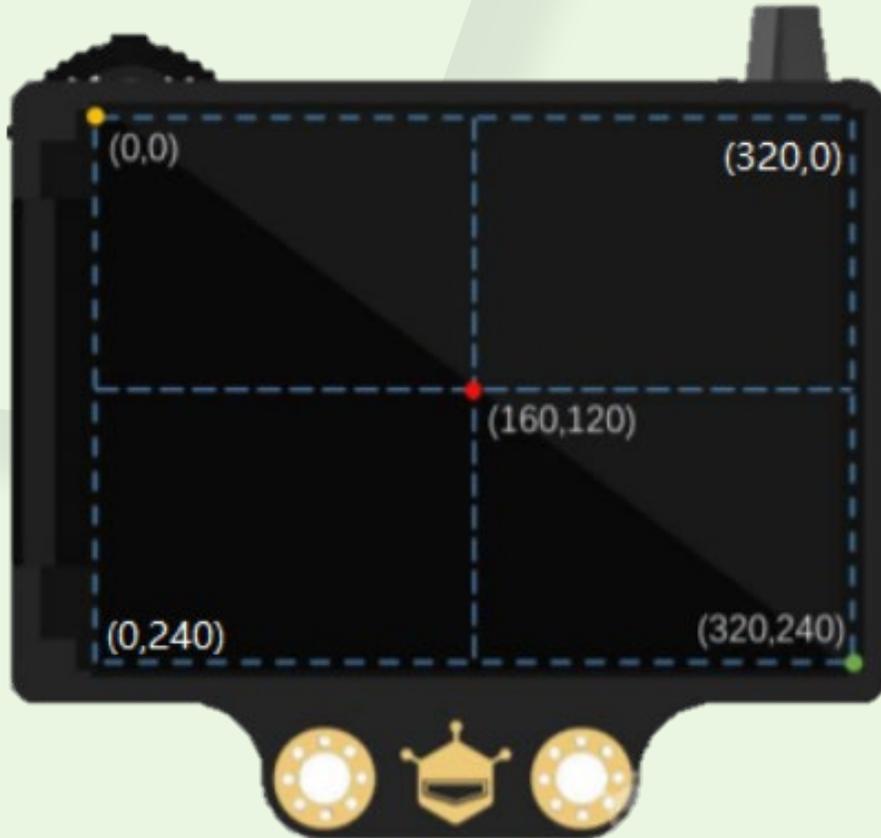
10. Drag 

from 

into 

11. Inside  enter "100"

Remark: HuskyLens' coordinates



- The object's coordinates will be displayed when HuskyLens detects an object
- Format: (x, y)

```
function Line_Tracking_Mode
  HuskyLens request data once and save into the result
  if HuskyLens check if ID 1 arrow is on screen from the result then
    set xcenter to HuskyLens get X endpoint of ID 1 arrow from the result
    if xcenter < 100 then
      call Turn_Left
    else
  
```

12. Drag **call Turn_Left**

from **f(x) Functions**

into “if then”

```
function Line_Tracking_Mode
  HuskyLens request data once and save into the result
  if HuskyLens check if ID 1 arrow is on screen from the result then
    set xcenter to HuskyLens get X endpoint of ID 1 arrow from the result
    if xcenter < 100 then
      call Turn_Left
    if true then
      +
      -
      +
```

13. Drag



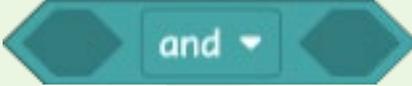
from



into “if then else”

```
function Line_Tracking_Mode  
  HuskyLens request data once and save into the result  
  if HuskyLens check if ID 1 arrow is on screen from the result then  
    set xcenter to HuskyLens get X endpoint of ID 1 arrow from the result  
    if xcenter < 100 then  
      call Turn_Left  
    if and then  
  else  
  +  
  +
```

14

14. Drag 
from 
into "if then"

```
function Line_Tracking_Mode  
  HuskyLens request data once and save into the result  
  if HuskyLens check if ID 1 arrow is on screen from the result then  
    set xcenter to HuskyLens get X endpoint of ID 1 arrow from the result  
    if xcenter < 100 then  
      call Turn_Left  
    if xcenter ≥ 100 and xcenter ≤ 220 then  
      call Move_Forward  
    if xcenter > 220 then  
      call Turn_Right  
    if xcenter < 0 then  
      call Turn_Left  
  end  
end
```

15

15. Set as follows

```
function Line_Tracking_Mode
  HuskyLens request data once and save into the result
  if HuskyLens check if ID 1 arrow is on screen from the result then
    set xcenter to HuskyLens get X endpoint of ID 1 arrow from the result
    if xcenter < 100 then
      call Turn_Left
    +
    if xcenter ≥ 100 and xcenter ≤ 220 then
      call Move_Forward
    +
    if xcenter > 220 then
      call Turn_Right
    +
  +
  +
  +
```

16

16. Set as follows

```
function Line_Tracking_Mode
  HuskyLens request data once and save into the result
  if HuskyLens check if ID 1 arrow is on screen from the result then
    set xcenter to HuskyLens get X endpoint of ID 1 arrow from the result
    if xcenter < 100 then
      call Turn_Left
    +
    if xcenter ≥ 100 and xcenter ≤ 220 then
      call Move_Forward
    +
    if xcenter > 220 then
      call Turn_Right
    +
    iCar Stop
  -
```

17. Drag **iCar Stop**

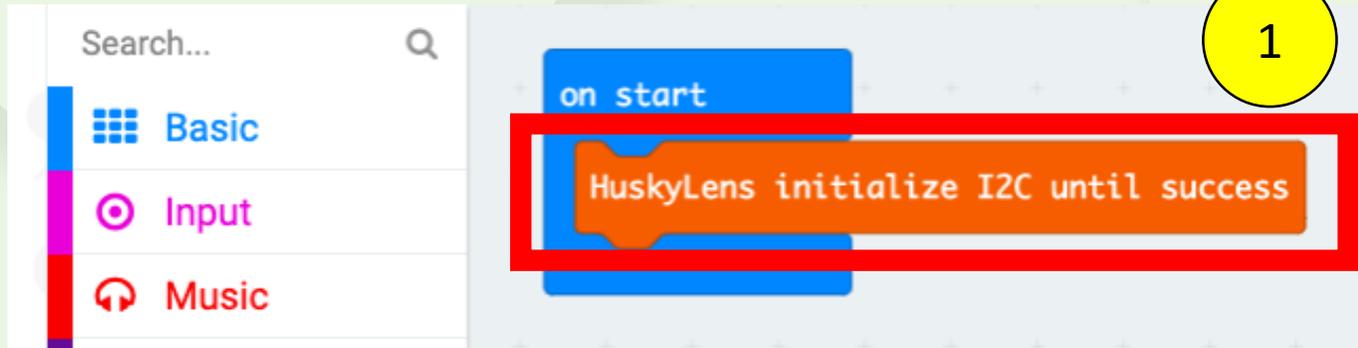
from **CUHK-JC iCar**

into "if then else"

17



Step 5: Setting Up The “on start”



1. Drag

HuskyLens initialize I2C until success

from

CUHK-JC iCar HuskyLens

into “on start”

```
on start
  HuskyLens initialize I2C until success
  HuskyLens switch algorithm to Face Recognition
```

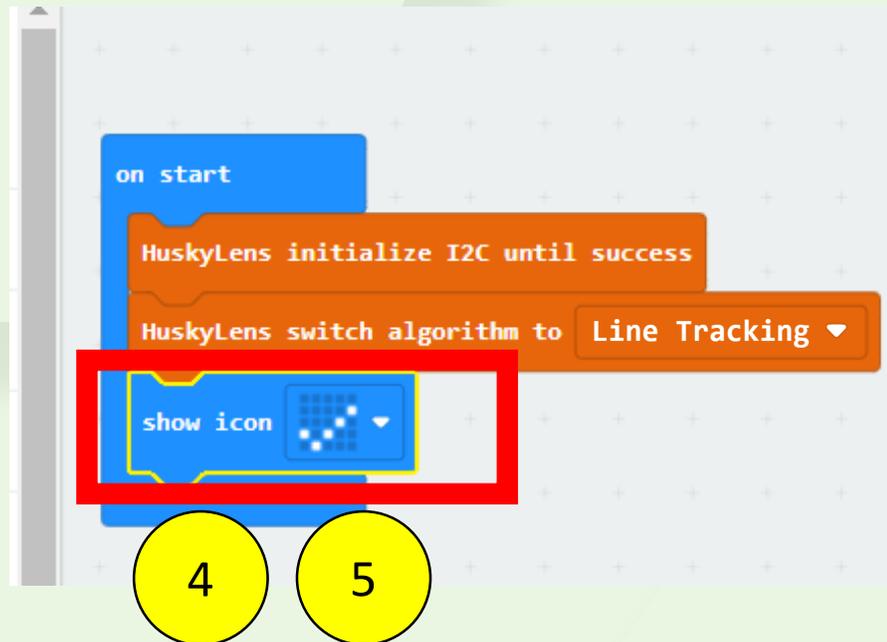
2

```
on start
  HuskyLens initialize I2C until success
  HuskyLens switch algorithm to Line Tracking
```

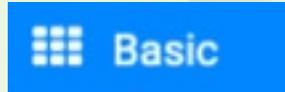
- Face Recognition
- Object Tracking
- Object Recognition
- ✓ Line Tracking
- Color Recognition
- Tag Recognition
- Object Classification
- QR Recognition (EDU only)
- Barcode Recognition (EDU only)

3

2. Drag HuskyLens switch algorithm to Face Recognition from CUHK-JC iCar HuskyLens into "on start"
3. Inside HuskyLens switch algorithm to Face Recognition click "Line Tracking"



4. Drag 

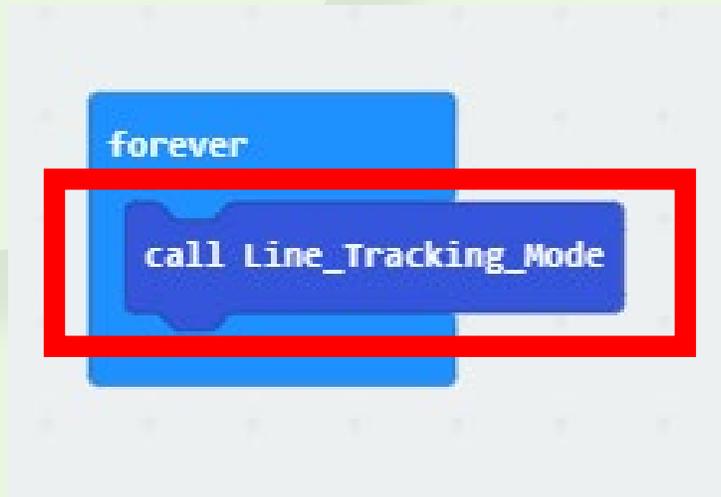
from 

into "on start"

5. Click 



Step 6: Setting Up The “forever”



1

1. Drag `call Line_Tracking_Mode` from `f(x) Functions` into “forever”



Finished!

```
on start
  HuskyLens initialize I2C until success
  HuskyLens switch algorithm to Line Tracking
  show icon ✓

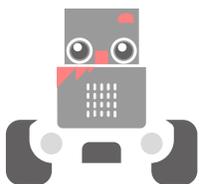
forever
  call Line_Tracking_Mode

function Move_Forward
  iCar Move Forward at speed 25 %

function Turn_Left
  iCar Turn Left at speed 25 %

function Turn_Right
  iCar Turn Right at speed 25 %

function Line_Tracking_Mode
  HuskyLens request data once and save into the result
  if HuskyLens check if ID 1 arrow is on screen from the result then
    set xcenter to HuskyLens get X endpoint of ID 1 arrow from the result
    if xcenter <= 180 then
      call Turn_Left
    if xcenter >= 180 and xcenter <= 220 then
      call Move_Forward
    if xcenter > 220 then
      call Turn_Right
  else
    iCar Stop
```



Step 7: Download The Program To CUHK iCar

Microsoft | micro:bit

Search...

Basic

Input

Music

Led

Radio

Loops

Logic

Variables

Math

CUHK-JC iCar HuskyLens

CUHK-JC iCar

CUHK-JC iCar Extension

CUHK-JC iCar Experiments

Extensions

Advanced

on start

HuskyLens initialize 22C until success

HuskyLens switch algorithm to Line Tracking

show icon

forever

call Line_Tracking_Mode

function Move_Forward

iCar Move Forward at speed 25 %

function Turn_Left

iCar Turn Left at speed 25 %

function Turn_Right

iCar Turn Right at speed 25 %

function Line_Tracking_Mode

HuskyLens request data once and save into the result

if HuskyLens check if ID 1 arrow is on screen from the result then

set xcenter to HuskyLens got X endpoint of ID 1 arrow from the result

if xcenter <= 100 then

call Turn_Left

if xcenter >= 100 and xcenter <= 220 then

call Move_Forward

if xcenter >= 220 then

call Turn_Right

else

iCar Stop

Download

1

1. Click

 Download

Microsoft | micro:bit

Blocks JavaScript

CUHK-JC iCar HuskyLe

CUHK-JC iCar

CUHK-JC iCar Extensio

Advanced

Functions

Arrays

Text

Game

Images

Pins

Serial

Download completed...

Your code is being downloaded as a .hex file. You can drag this file to your micro:bit using your computer's file explorer.

New! Download your code faster by pairing with web usb!
Pair now

Don't show this again

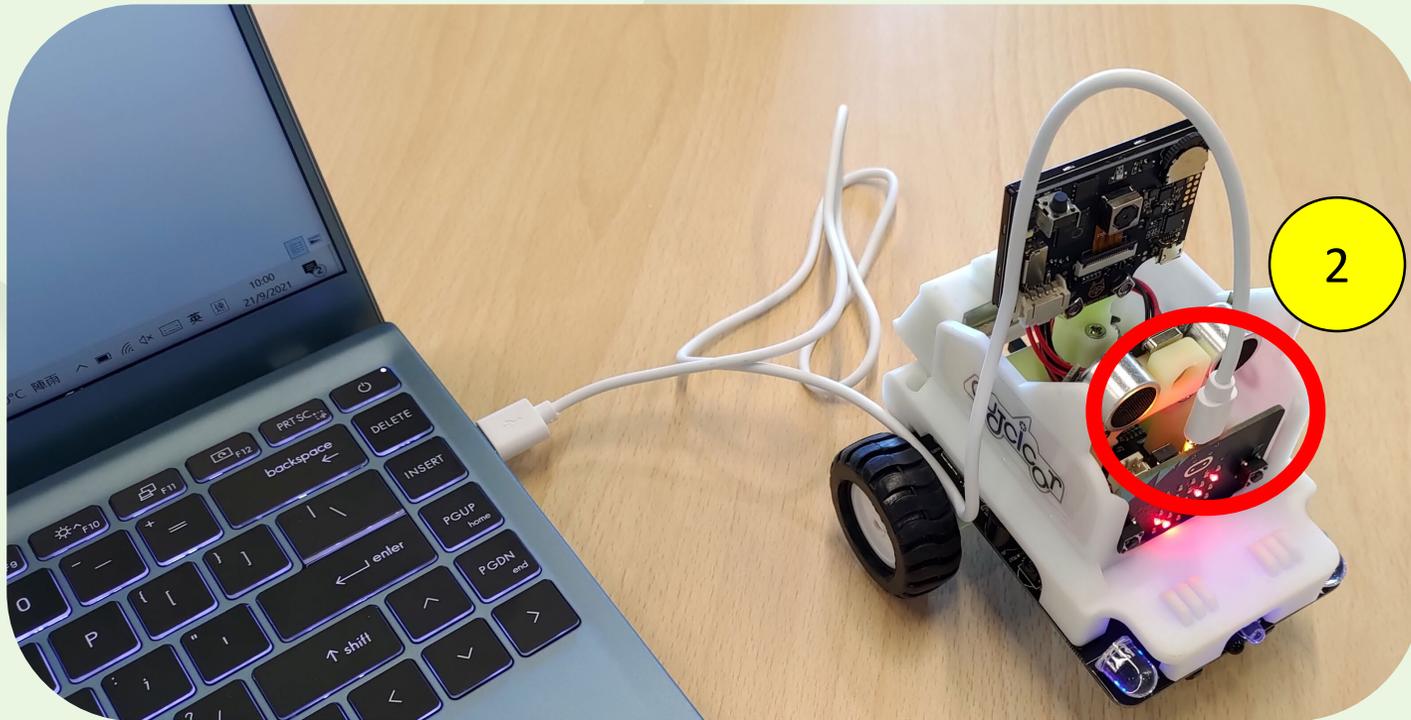
Help Download again Done

Line tracking

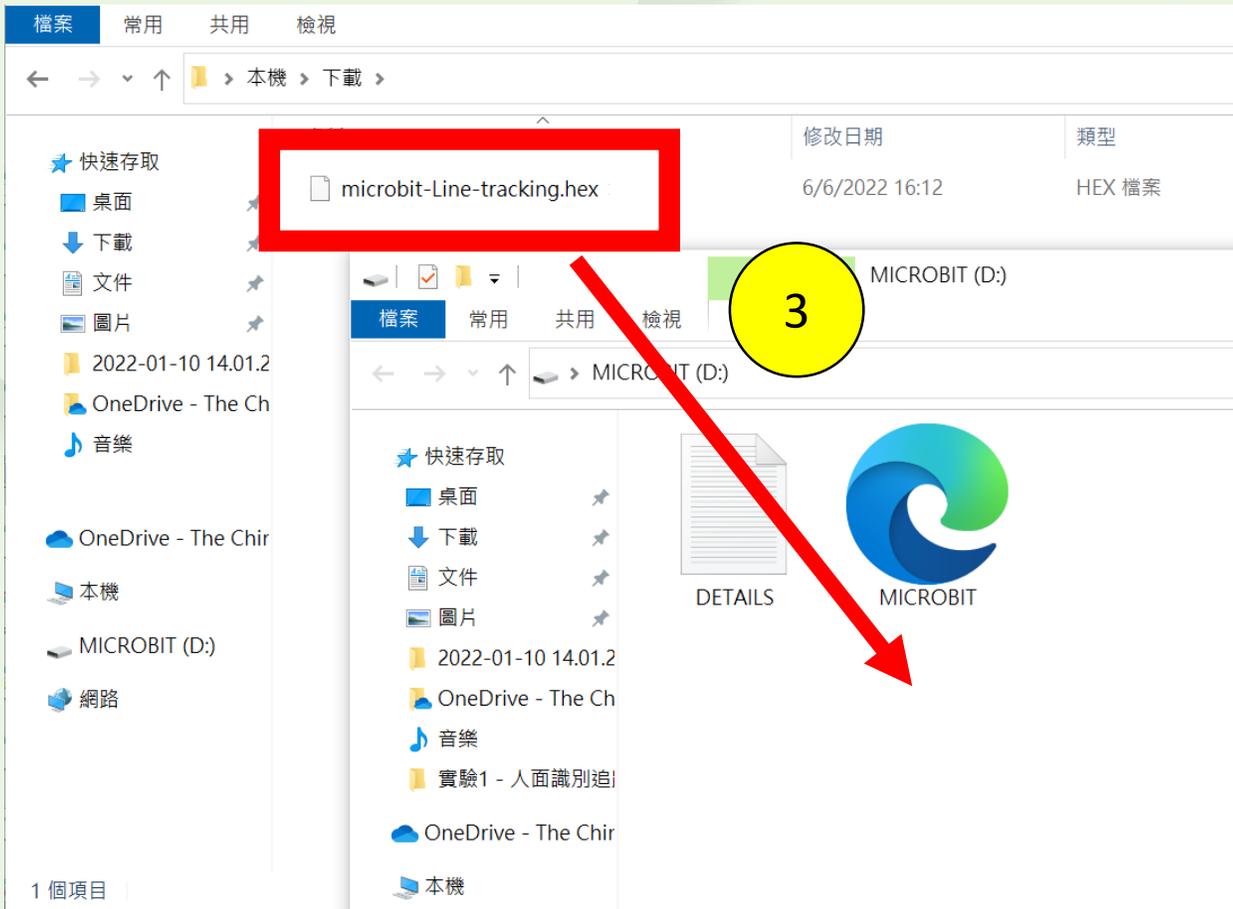
microbit-Line-tra...hex

The hex file is downloaded!





2. Connect the micro:bit to computer by a micro USB cable



3. Drag the downloaded hex file into the micro:bit window



4. Wait for the completion of cloning process

Caution:

- The micro:bit window will potentially disappear after the completion
- After the completion of cloning process, the hex file will not be displayed in the micro:bit window

5. Disconnect the micro:bit from your computer

If the connection between CUHK iCar and the computer is failed:

- Restart the computer
- Try another USB port
- Change the micro USB cable

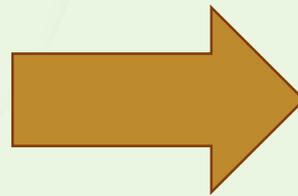


You have finished programming!

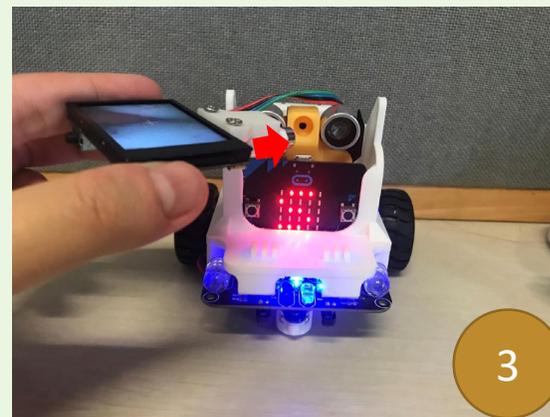
Let's start the experiment!



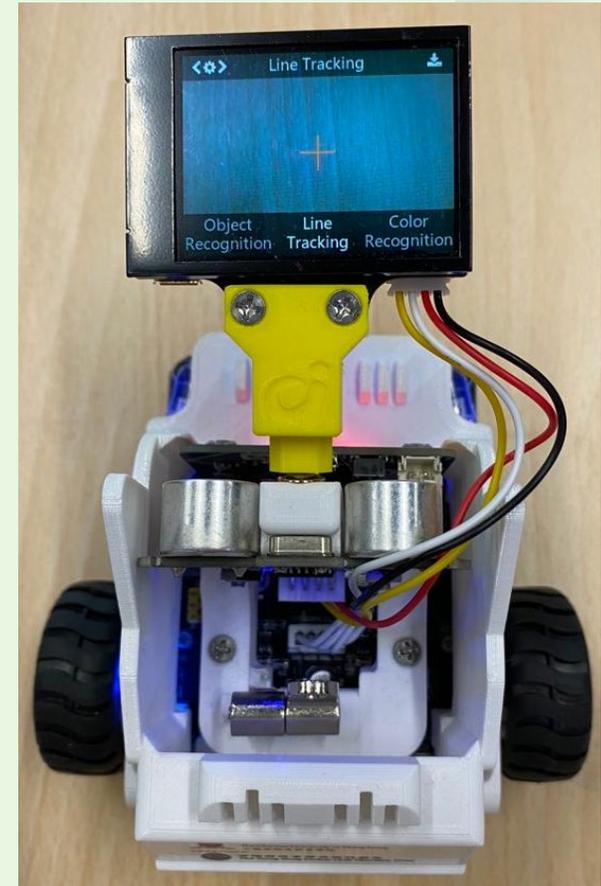
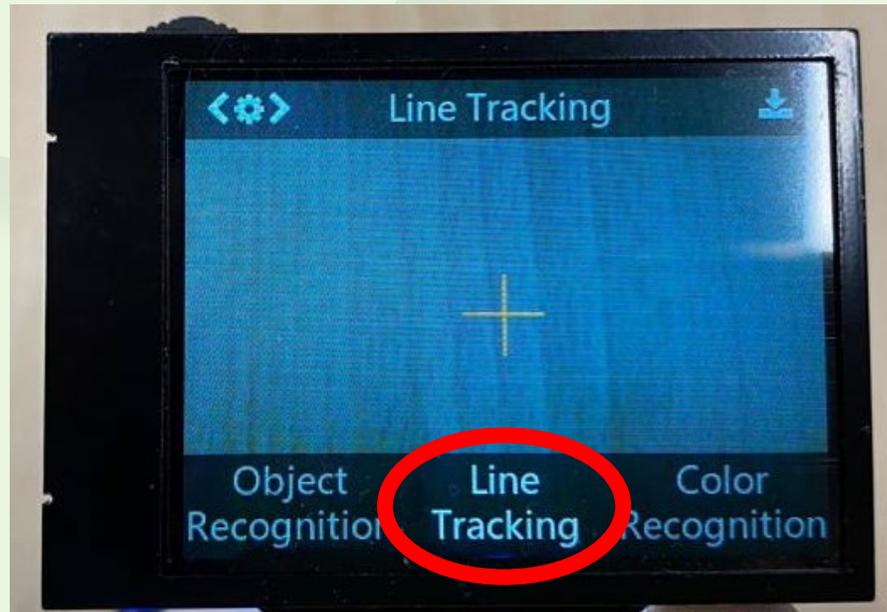
Switch On Your CUHK iCar



Change the position of HuskyLens to face the ground.

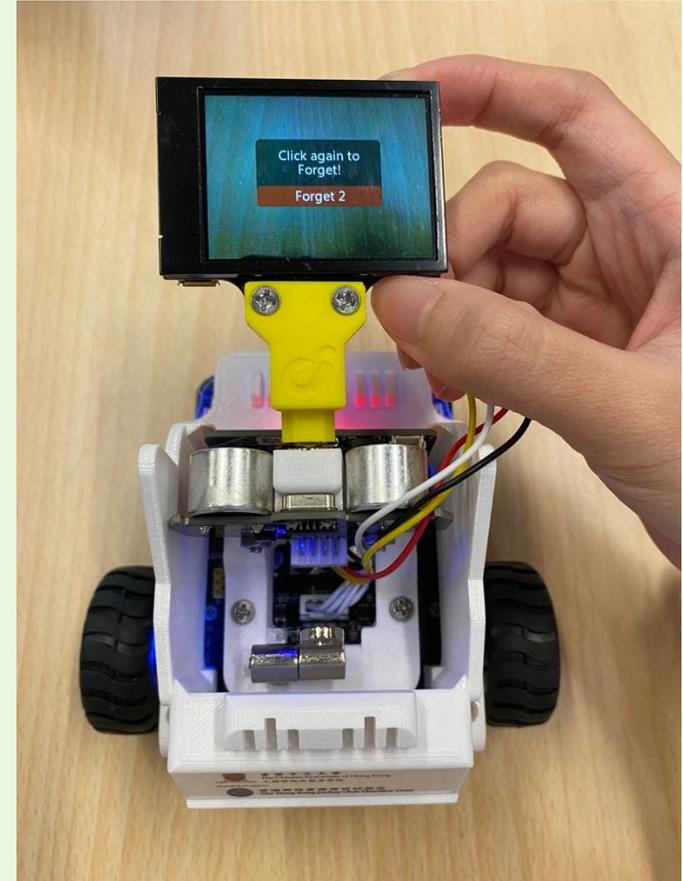


Once it is switched on, the HuskyLens will automatically adjust to Line Tracking Mode.

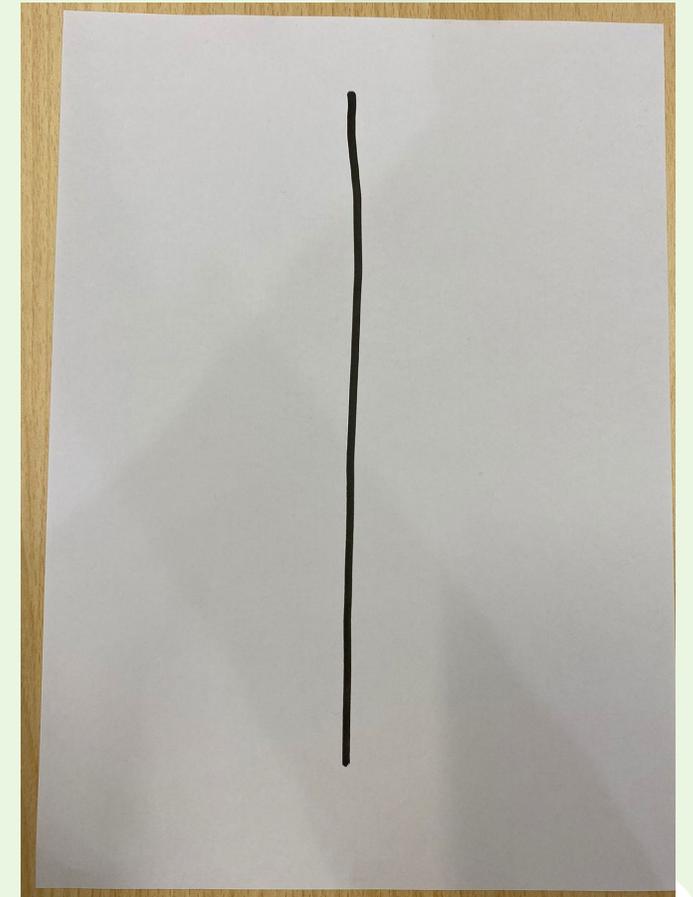
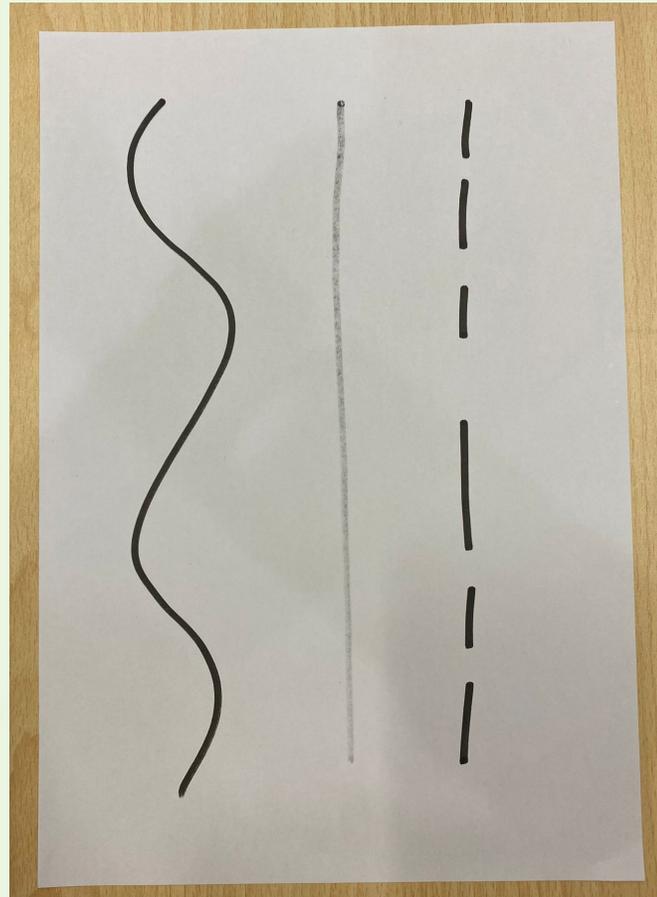


Clear The Previous Data

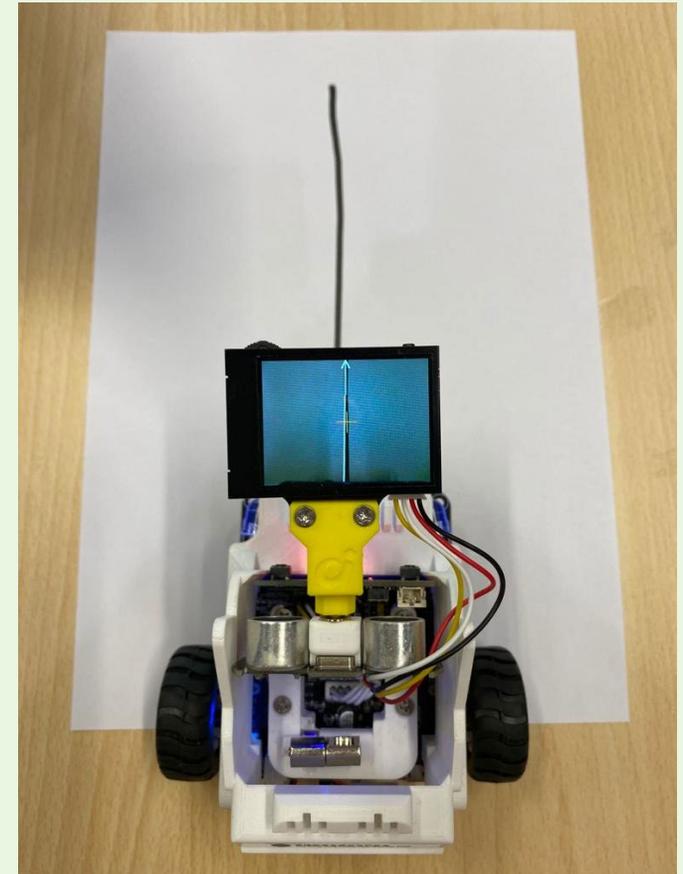
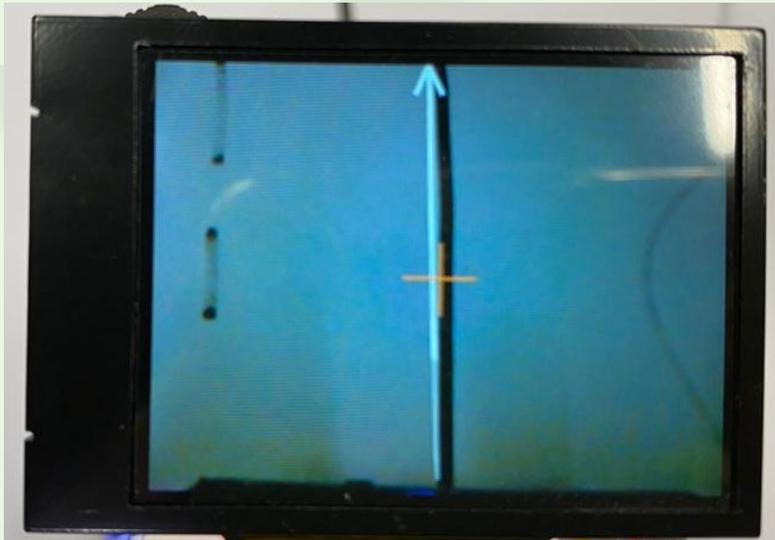
Press the learning button.
Then, press the button again when the confirmation box is appeared to “Forget” the data.



Draw black lines on the white paper.

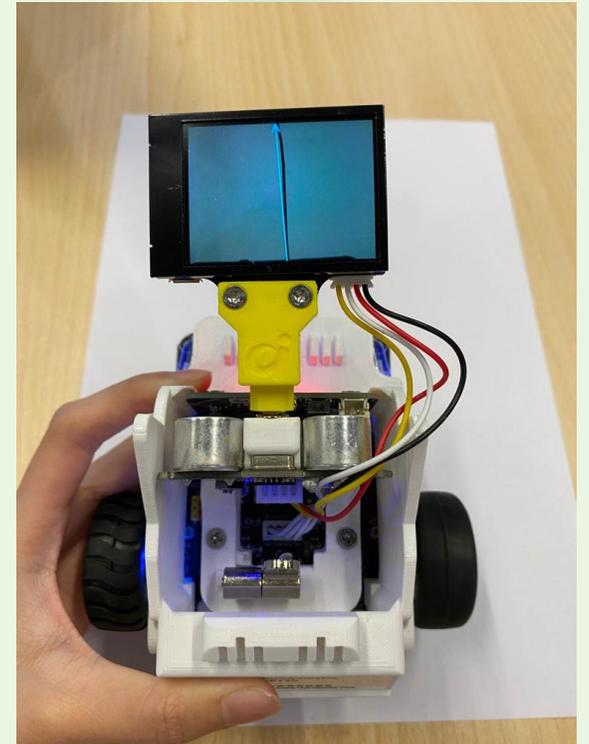
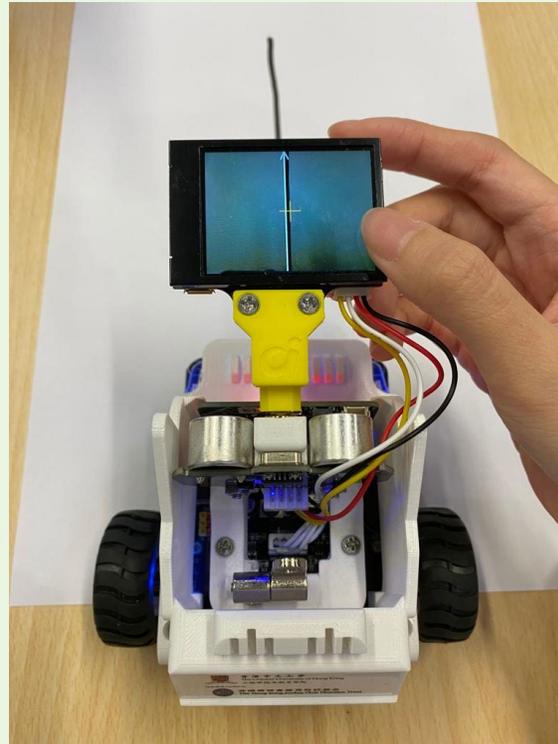


The line should be spotted by the camera of HuskyLens.
Make sure the white arrow on the monitor is aligned with the black line.



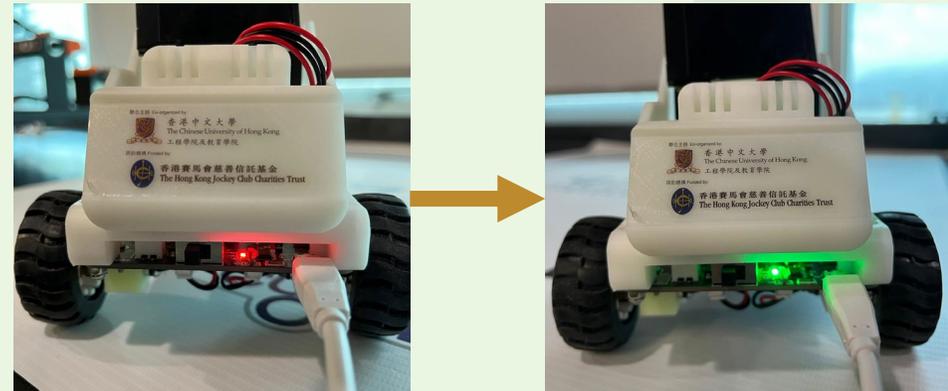
Once the line and the camera are aligned, press the learning button.

The white arrow will then change to blue arrow, and the CUHK iCar will track the line automatically.



If CUHK iCar does not work as expected:

- Try to fully charge the CUHK iCar



- If the CUHK iCar still does not work as expected, then you can try to revise the program by yourself. For details, please refer to the next slide

```

on start
  HuskyLens initialize I2C until success
  HuskyLens switch algorithm to Line Tracking
  show icon

forever
  call Line_Tracking_Mode

function Move_Forward
  iCar Move Forward at speed 25 %

function Turn_Left
  iCar Turn Left at speed 25 %

function Turn_Right
  iCar Turn Right at speed 25 %

function Line_Tracking_Mode
  HuskyLens request data once and save into the result
  if HuskyLens check if ID 1 arrow is on screen from the result then
    set xcenter to HuskyLens get X endpoint of ID 1 arrow from the result
    if xcenter < 100 then
      call Turn_Left
    if xcenter >= 100 and xcenter <= 220 then
      call Move_Forward
    if xcenter > 220 then
      call Turn_Right
    else
      iCar Stop
  
```

The numbers circled in red are the recommended speeds when the battery is fully charged

- Please adjust the speed gradually by +/- 5 according to the battery capacity or battery age, then re-enter the adjusted value to the position marked by the red circle
- After the adjustment, download the program to the micro:bit again. For details, please refer to slide 45