



愛動智教育系統

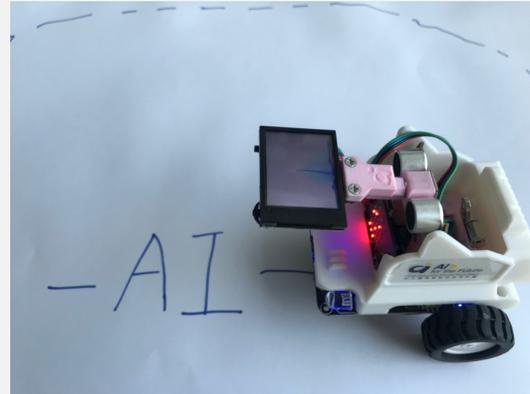
CUHK iCar Experiment Manual
Experiment 1: Face Following Experiment

Write Your Own Code

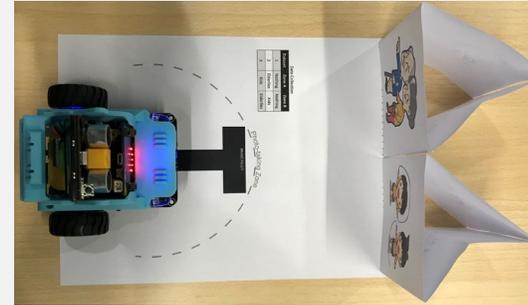
CUHK iCar



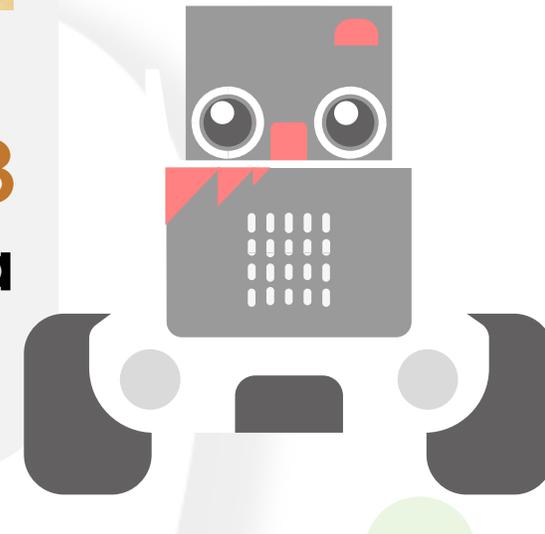
Experiment 1
Face Following



Experiment 2
Line Tracking



Experiment 3
Moral Dilemma



Face Following Experiment

Introduction Of The Experiment

This experiment showcases how the self-driving car follows a specific face.



Download Program To micro:bit

Face_following.hex

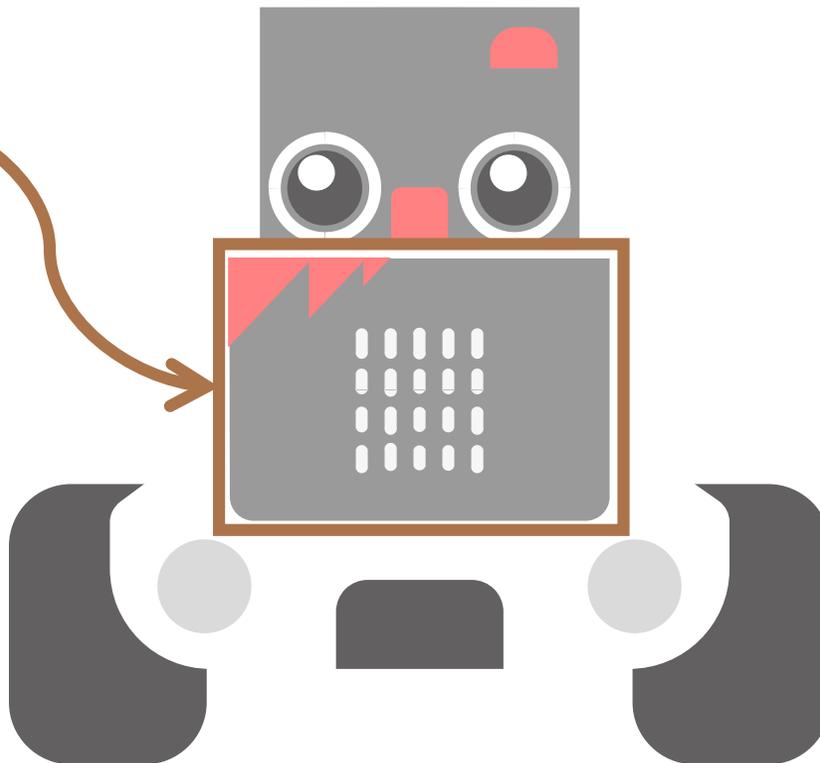
```
on start
  Huskylens initialize I2C until success
  Huskylens switch algorithm to Face Recognition
  show icon

function Move_Forward
  iCar Move Forward at speed 30 %

function Turn_Left
  iCar Turn Left at speed 30 %

function Turn_Right
  iCar Turn Right at speed 30 %

function Face_Following_Mode
  Huskylens request data once and save into the result
  if Huskylens check if ID 1 frame is on screen from the result then
    set xcenter to Huskylens get X center of ID 1 frame from the result
    if xcenter < 50 then
      call Turn_Left
    if xcenter >= 50 and xcenter <= 240 then
      call Move_Forward
    if xcenter > 240 then
      call Turn_Right
    else
      iCar Stop
```



Method 1

Clone the .hex to micro:bit directly

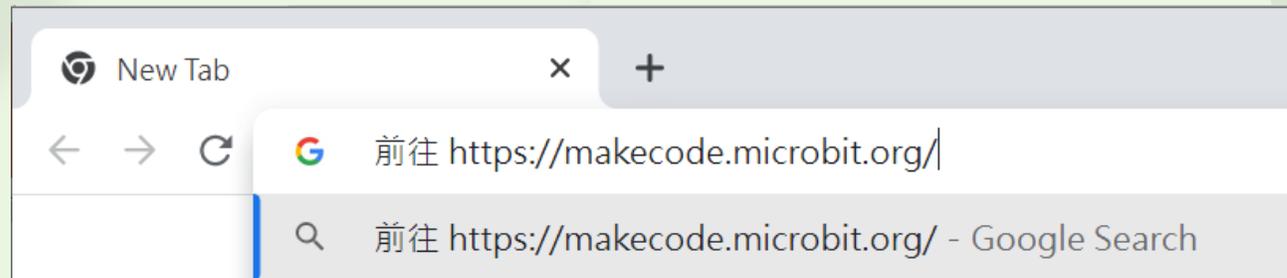
Method 2

Write your own code on MakeCode

Code On MakeCode

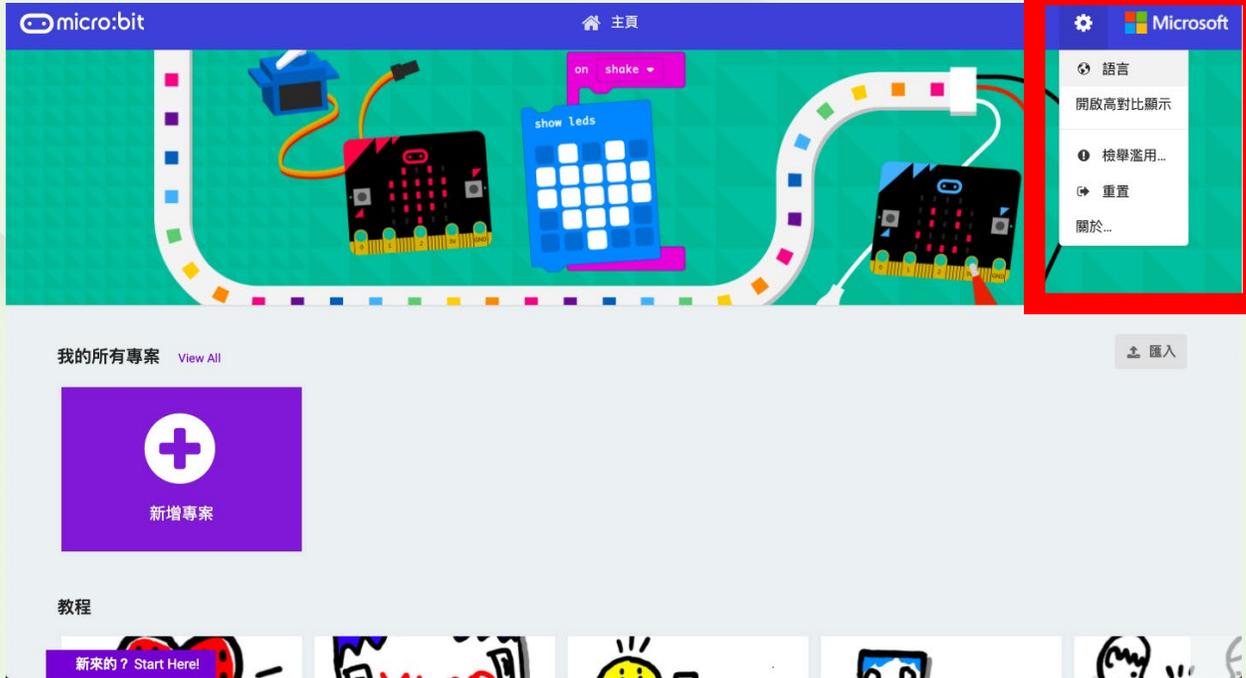


Browse <https://makecode.microbit.org/>



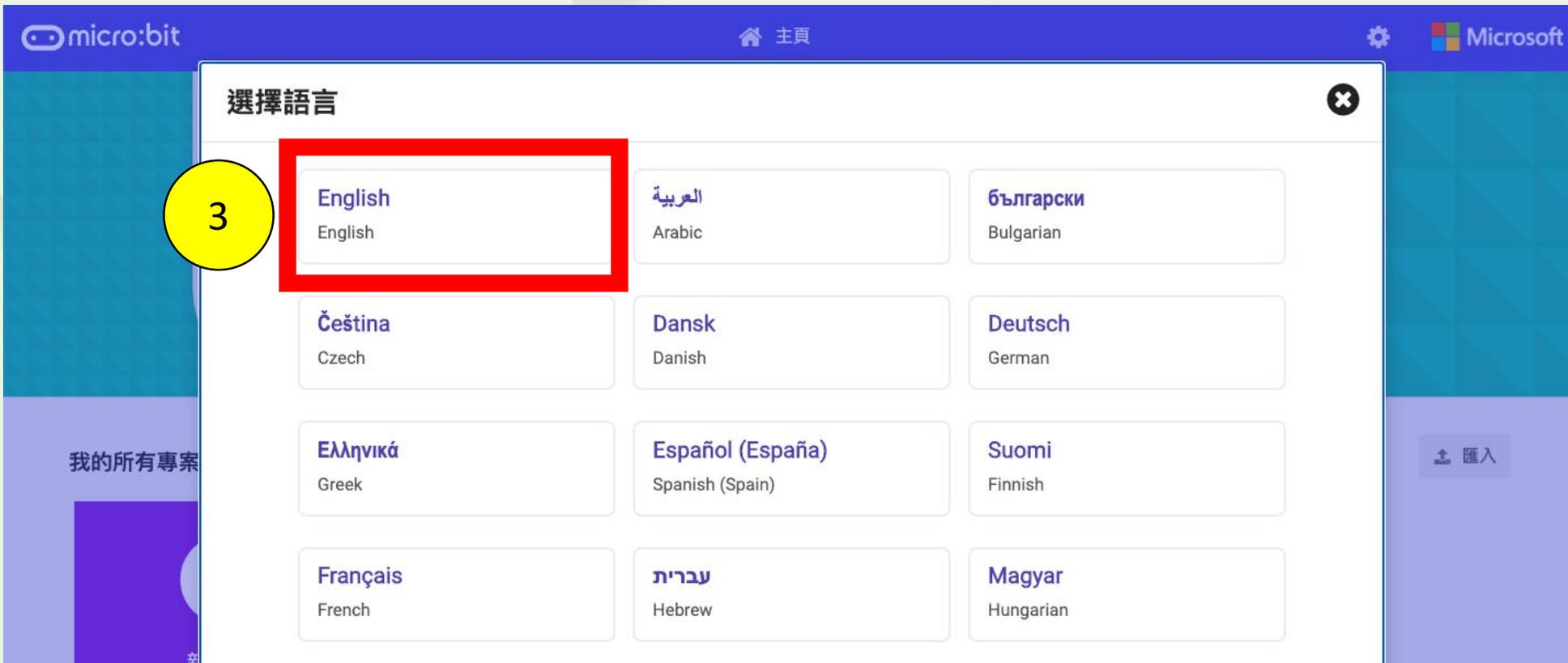


Caution
Please Set **English** As The Language!



1. Click
2. Click

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



micro:bit 主頁 Microsoft

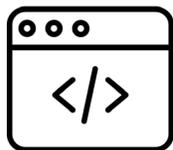
選擇語言

3

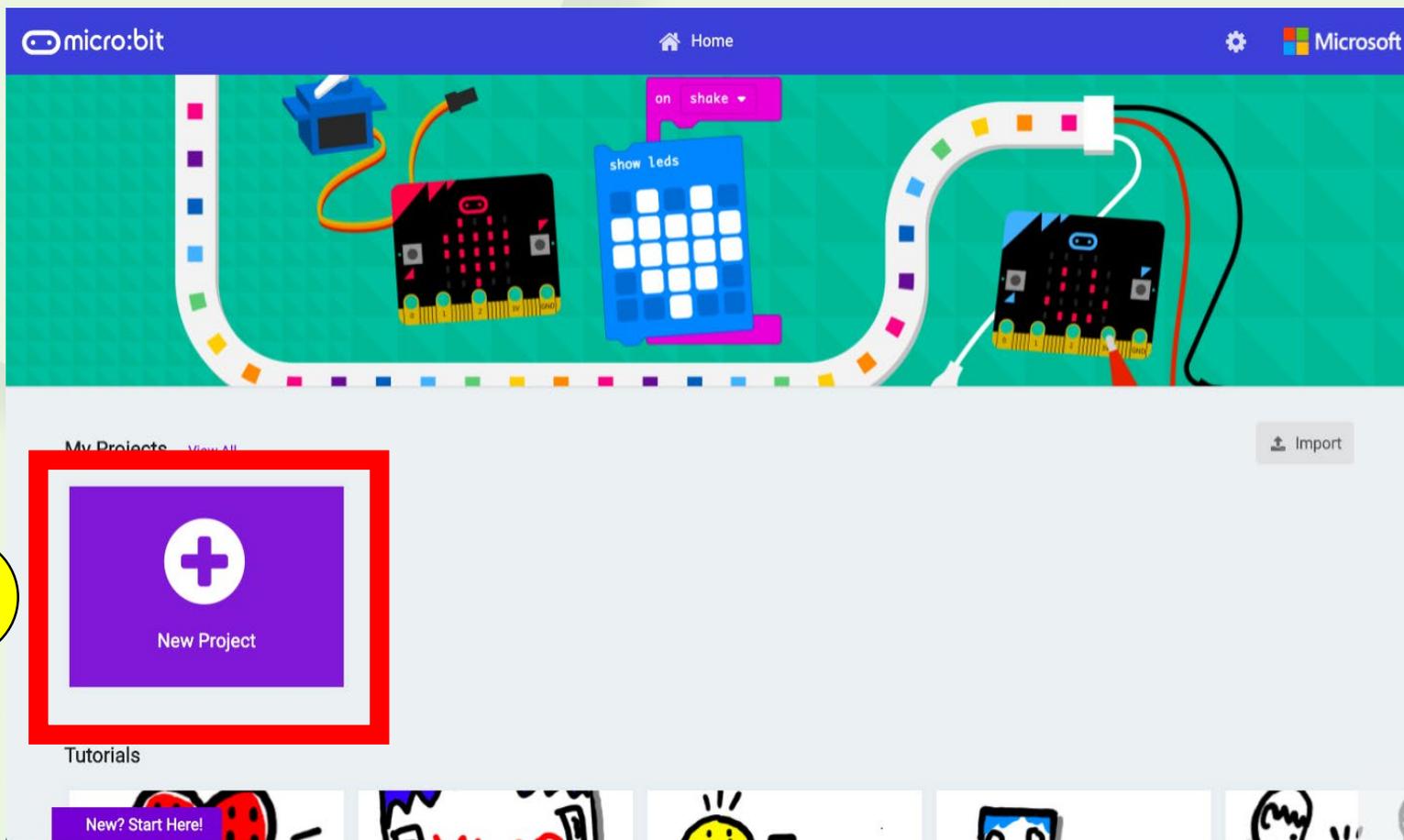
English English	العربية Arabic	български Bulgarian
Čeština Czech	Dansk Danish	Deutsch German
Ελληνικά Greek	Español (España) Spanish (Spain)	Suomi Finnish
Français French	עברית Hebrew	Magyar Hungarian

我的所有專案 匯入

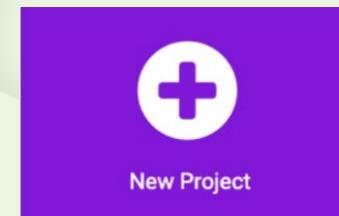
3. Click English



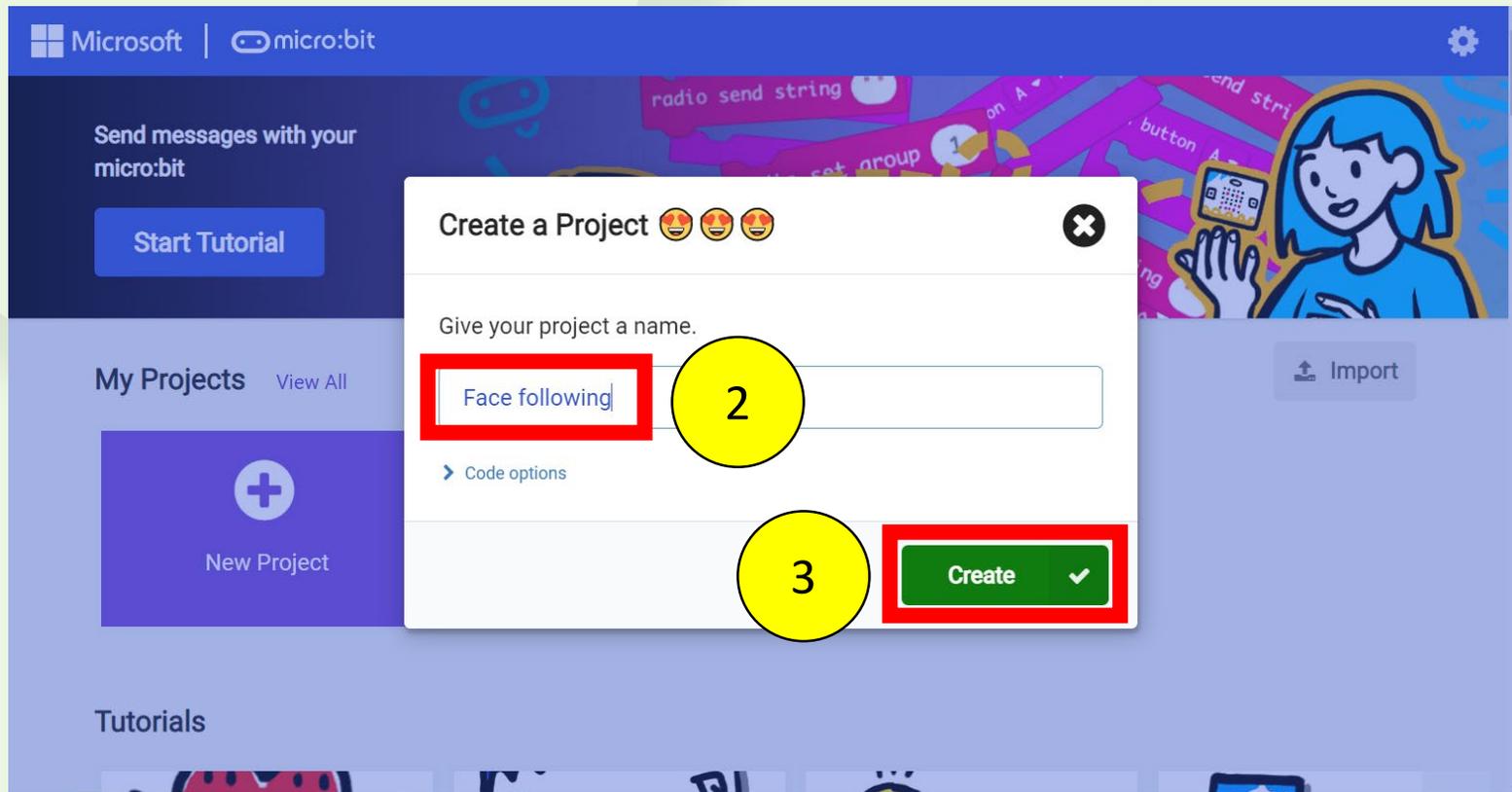
New Project



1. Click New Project



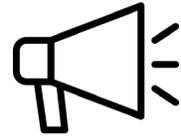
1



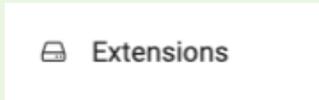
2. Enter “Face following”

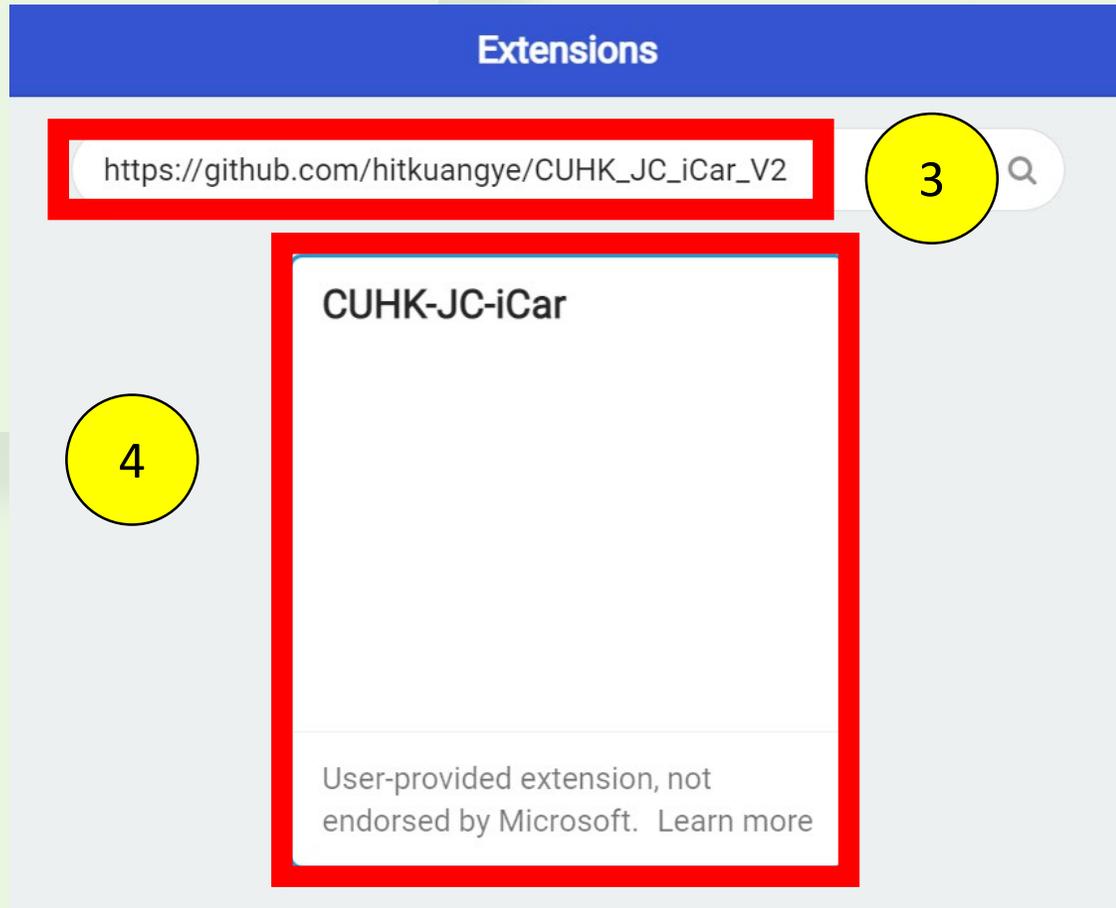
3. Click

Create

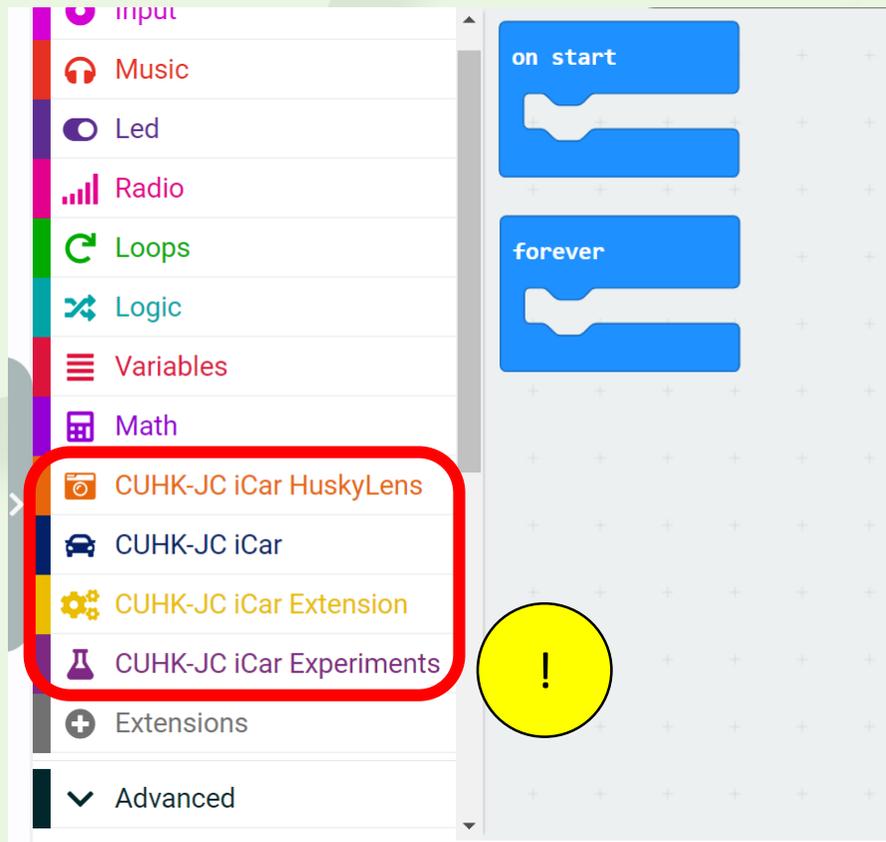


Prepare CUHK-JC-iCar Extension

1. Click 
2. Click 



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



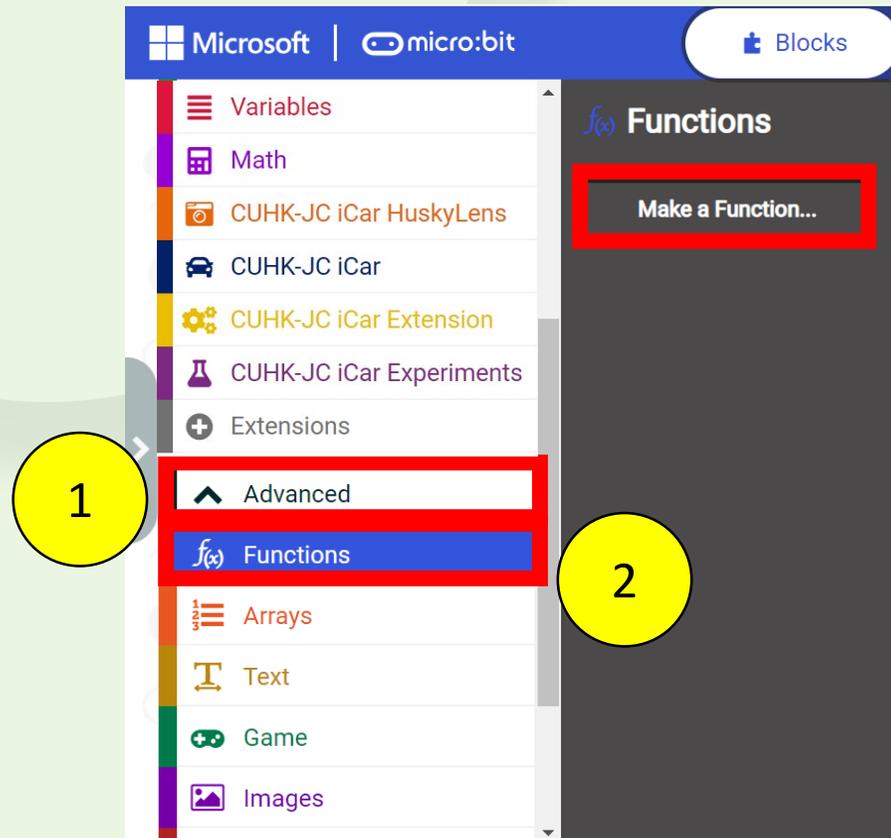
Extensions are included!

Introduction To micro:bit



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

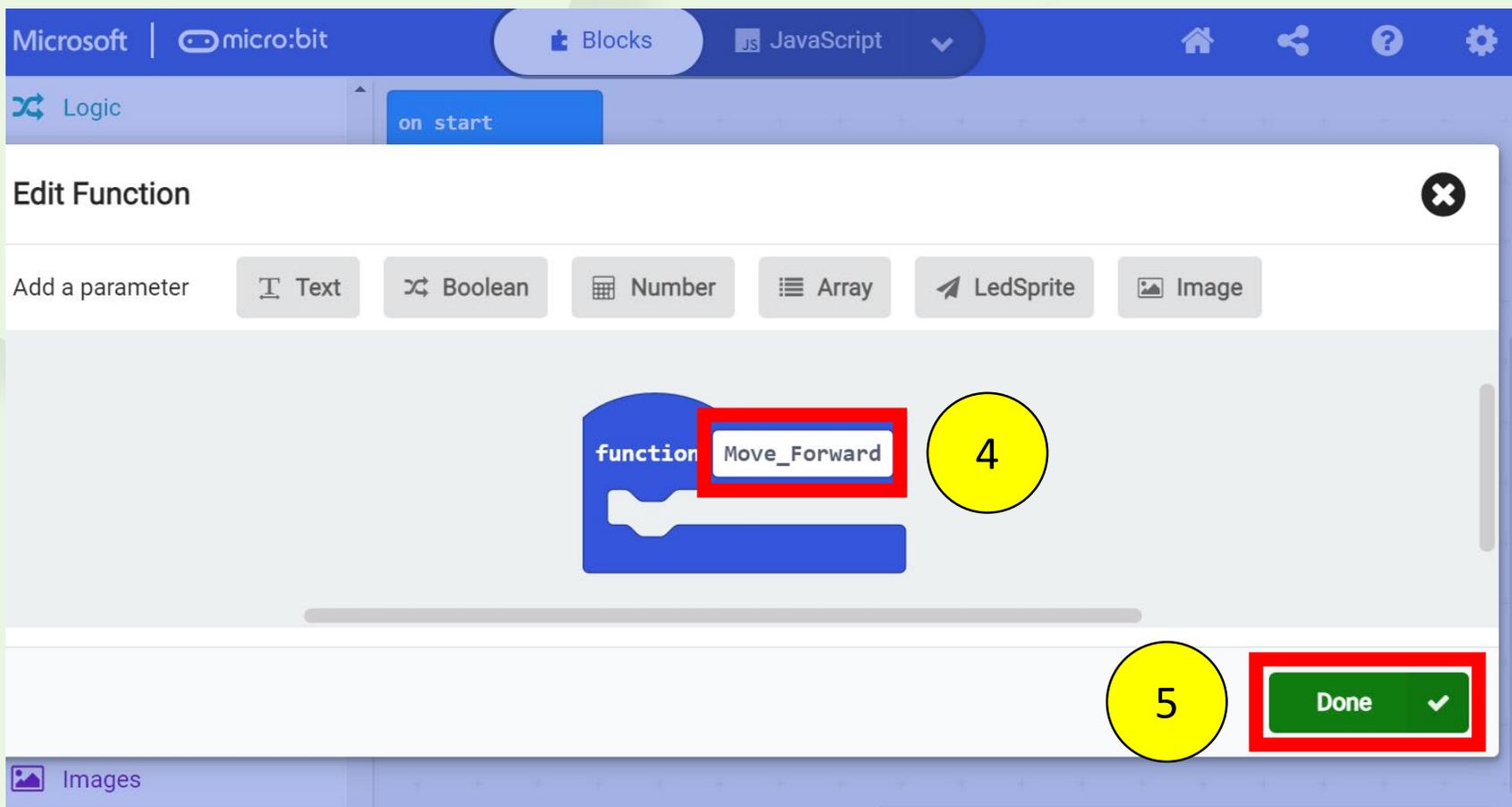
Functions Modules



1. Scroll to the bottom of module list and click 

2. Click 

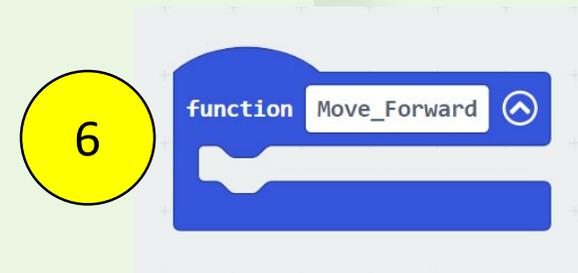
3. Click 



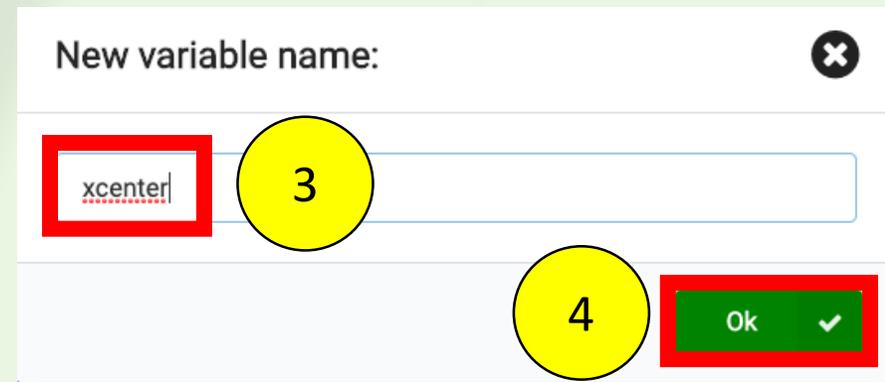
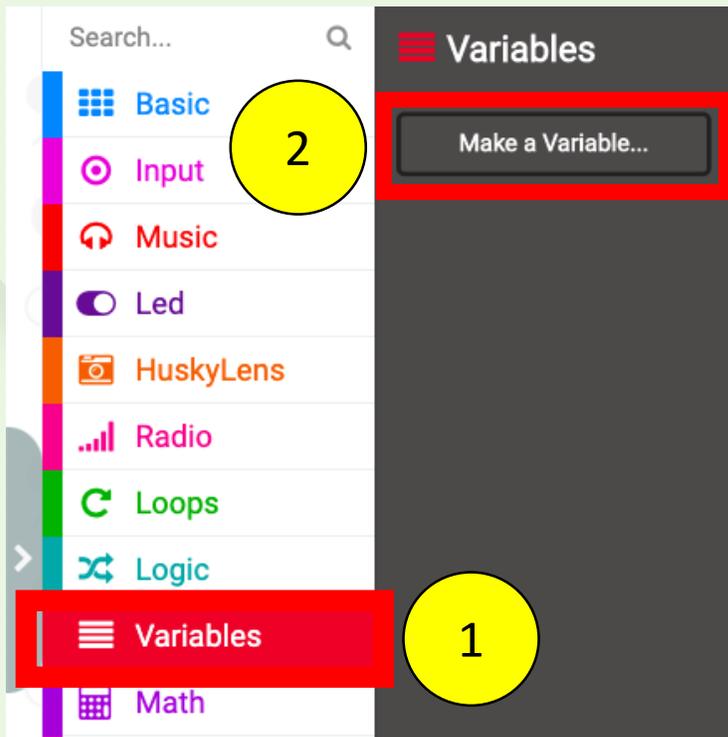
4. Name the function
"Move_Forward"

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 center of ID 1 frame 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

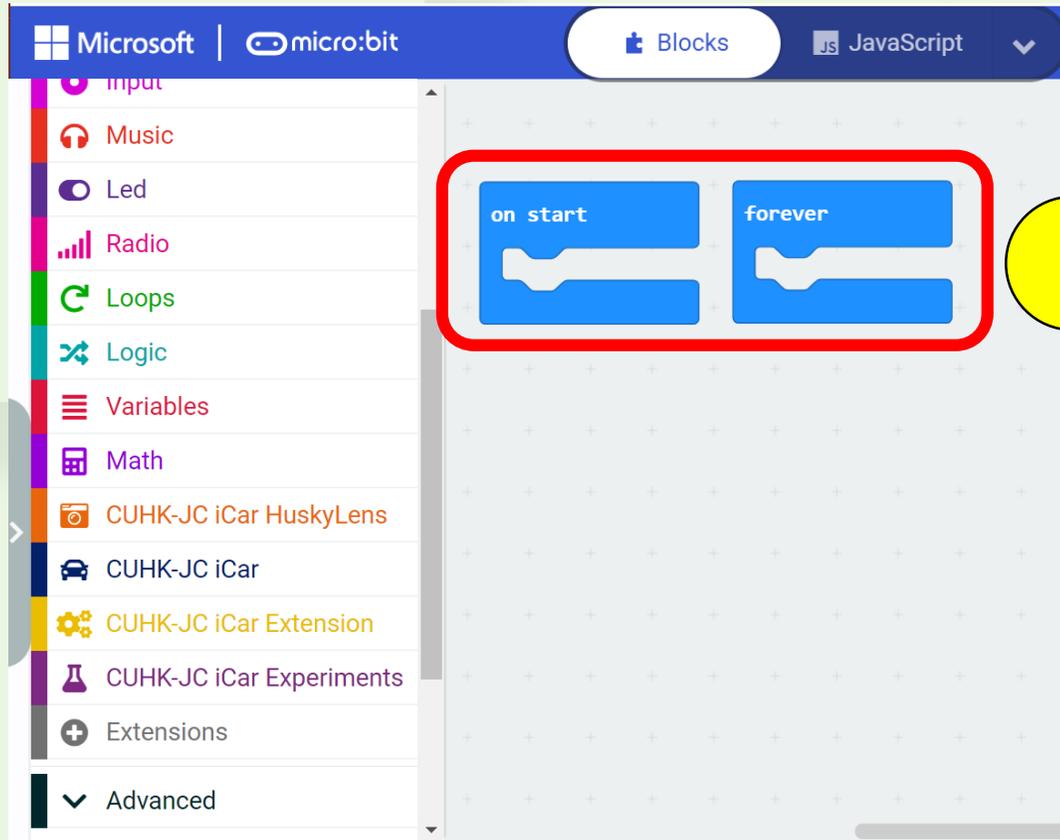
+

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!

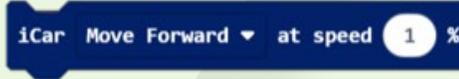


Step 1: Adding The “Move_Forward” Function

The screenshot shows the Microsoft micro:bit IDE interface. The top bar includes the Microsoft logo, the micro:bit logo, and tabs for 'Blocks' and 'JavaScript'. The left 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 an 'on start' block and a 'forever' loop block. A new 'function Move_Forward' block is being added to the workspace, highlighted with a red border and a yellow circle with the number 1 next to it.

1. Add “Move_Forward” function



2. Drag  from  into "Move_Forward" function

3. Inside  set to "Move Forward"

4. Inside  set the speed to "30"



Step 2: Adding The “Turn_Left” Function

The screenshot shows the Microsoft micro:bit IDE interface. The top bar includes the Microsoft logo, the micro:bit logo, and tabs for 'Blocks' and 'JavaScript'. The left sidebar contains a list of 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 shows a script starting with 'on start' and 'forever' blocks. A 'function Move_Forward' block is already present, containing an 'iCar Move Forward' block with a speed of 30%. A new 'function Turn_Left' block is being added to the script, highlighted with a red border. A yellow circle with the number '1' is positioned to the right of the 'Turn_Left' block.

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 "30"



Step 3: Adding The “Turn_Right” Function

Microsoft | micro:bit

Blocks JavaScript

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

forever

function Move_Forward

iCar Move Forward at speed 30 %

function Turn_Left

iCar Turn Left at speed 30 %

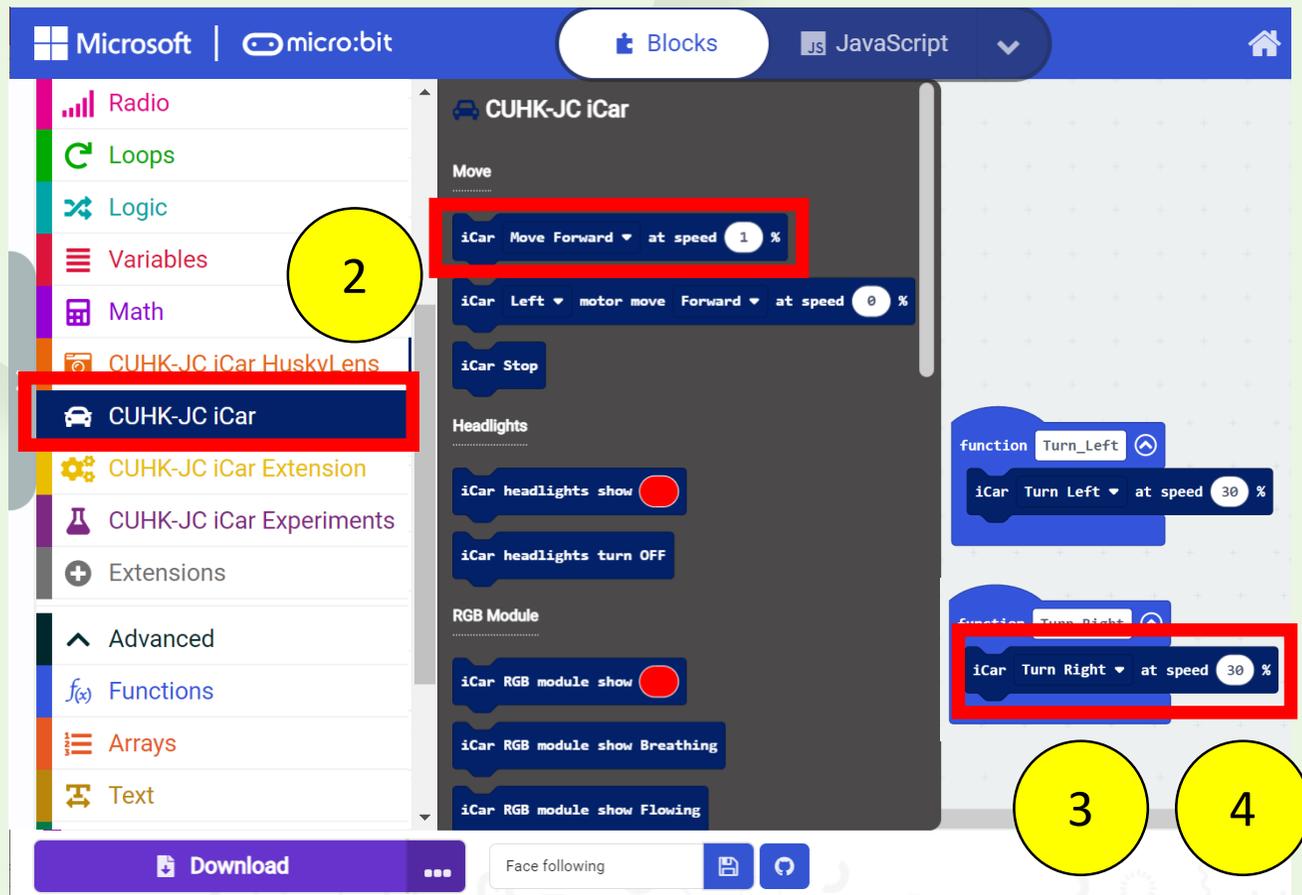
function Turn_Right

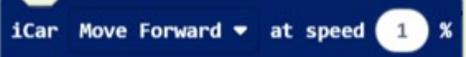
1

Download

Face following

1. Add “Turn_Right” function



2. Drag 

from 

into "Turn_Right" function

3. Inside 

set to "Turn Right"

4. Inside 

set the speed to "30"



Step 4: Adding The “Face_Following_Mode” Function

Microsoft | micro:bit

Blocks JavaScript

Input

Music

Led

Radio

Loops

Logic

Variables

Math

CUHK-JC iCar HuskyLens

CUHK-JC iCar

CUHK-JC iCar Extension

CUHK-JC iCar Experiment

Extensions

Advanced

function Move_Forward

iCar Move Forward at speed 30 %

function Turn_Left

iCar Turn Left at speed 30 %

function Face_Following_Mode

function Turn_Right

iCar Turn Right at speed 30 %

Download

Face following

1. Add “Face_Following_Mode” function

Microsoft | micro:bit

Blocks JavaScript

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 forever

function Move_Forward
iCar Move Forward at speed 30 %

function Turn_Left
iCar Turn Left at speed 30 %

function Face_Following_Mode
HuskyLens request data once and save into the result

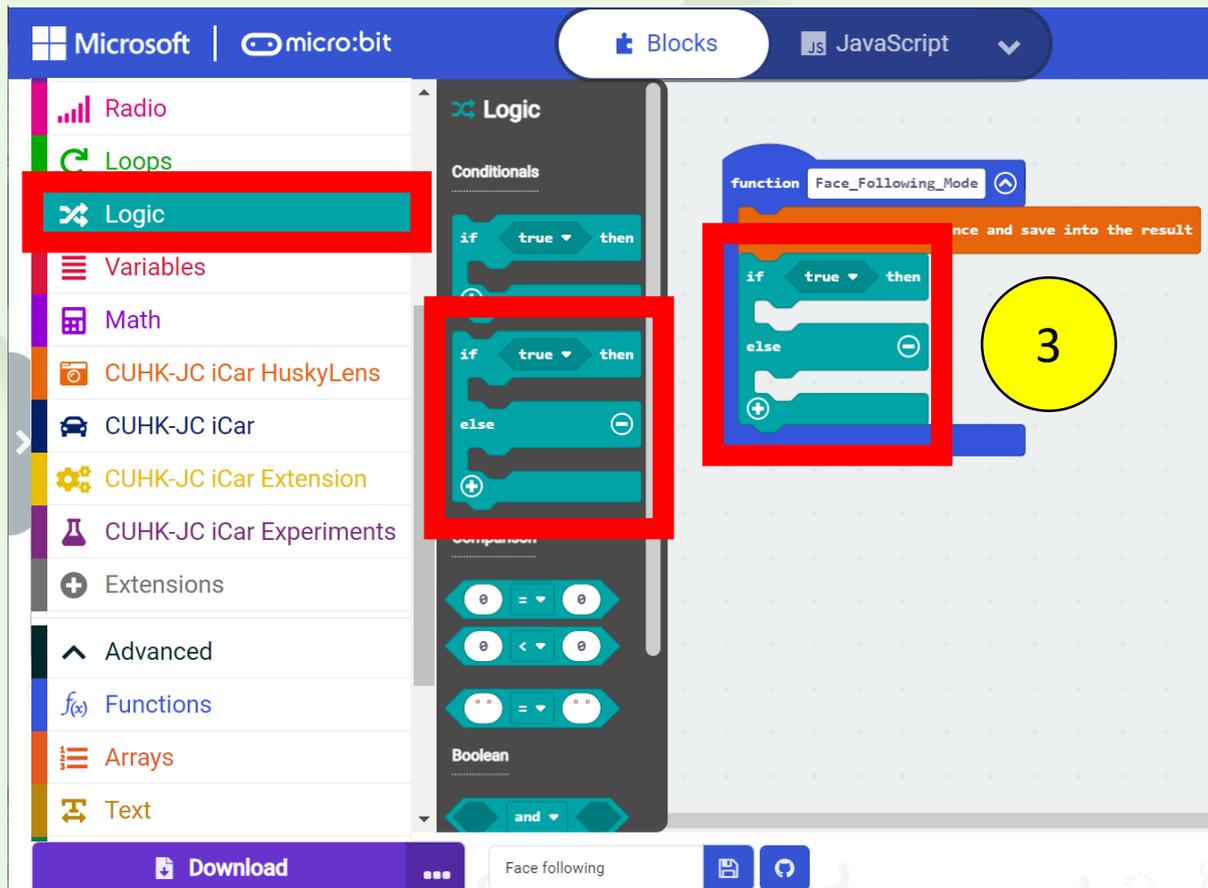
function Turn_Right
iCar Turn Right at speed 30 %

Download

Face following

2

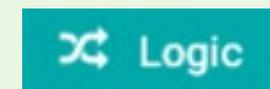
2. Drag **HuskyLens request data once and save into the result**
- from **CUHK-JC iCar HuskyLens**
- into “Face_Following_Mode ” function



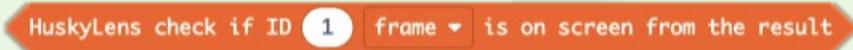
3. Drag



from



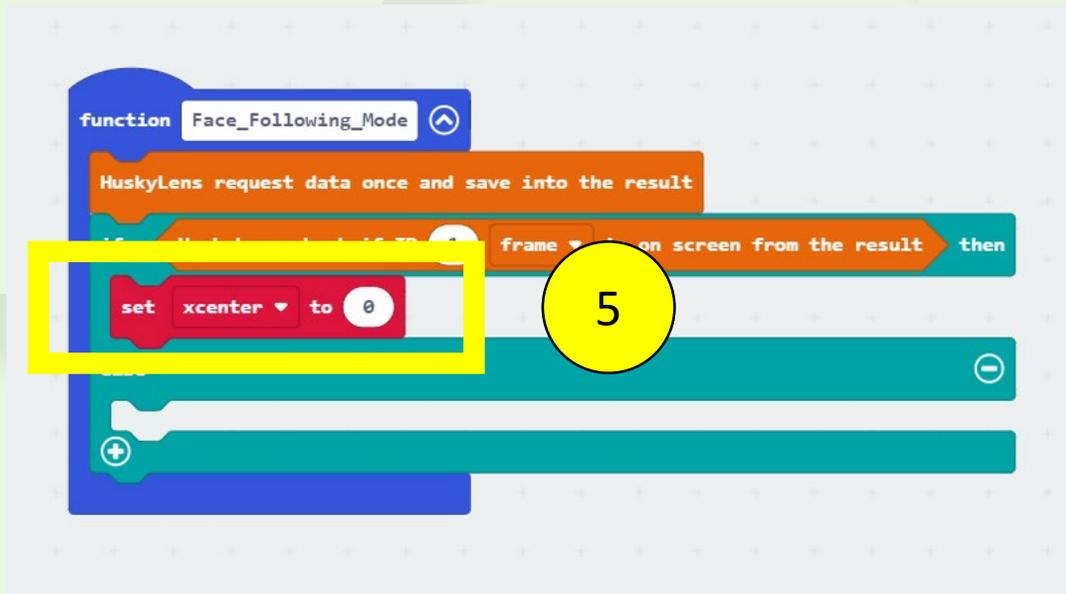
into "Face_Following_Mode" function

4. Drag  from  into "if then else"

4

```
function Face_Following_Mode  
  if true then  
    HuskyLens check if ID 1 frame is on screen from the result
```

```
function Face_Following_Mode  
  if HuskyLens check if ID 1 frame is on screen from the result then  
  else
```



5. Drag **set xcenter to 0**
from **Variables**
into "if then else"

```
function Face_Following_Mode  
  HuskyLens request data once and save into the result  
  if HuskyLens  
    set xcenter to HuskyLens get X center of ID 1 frame from the result  
  else  
  +
```

6. Drag HuskyLens get X center of ID 1 frame from the result from CUHK-JC iCar HuskyLens into set xcenter to 0

6

```
function Face_Following_Mode  
  HuskyLens request data once and save into the result  
  if HuskyLens  
    set xcenter to HuskyLens get X center of ID 1 frame from the result  
  else  
  +
```

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

7. Drag



from



into "if then else"

```
function Face_Following_Mode  
  HuskyLens request data once and save into the result  
  if HuskyLens check if ID 1 frame is on screen from the result then  
    set xcenter to HuskyLens get X center of ID 1 frame from the result  
    if xcenter < 80 then  
    else  
  end  
end
```

8

9

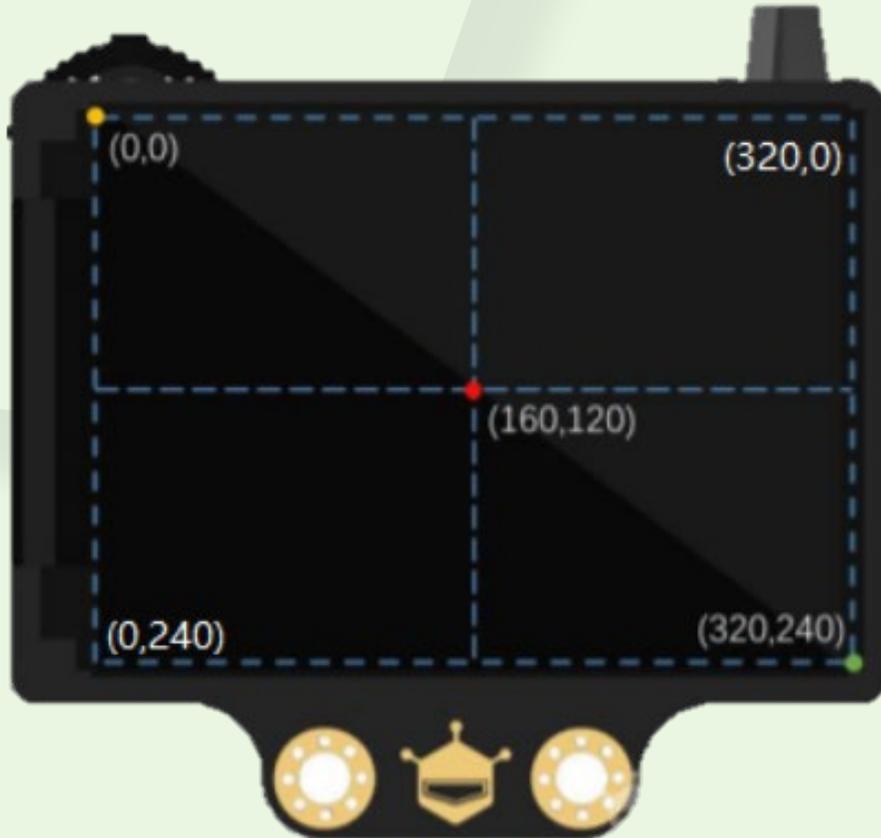
10

8. Drag 
from 
into "if then"

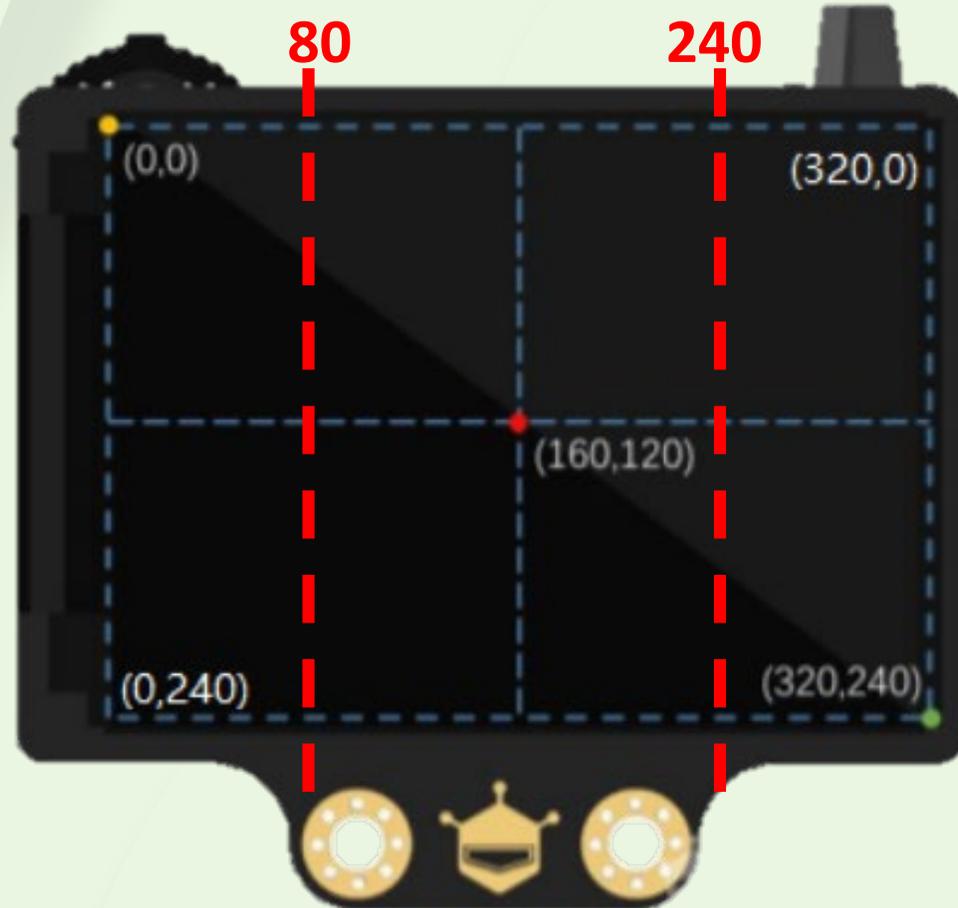
9. Drag 
from 
into 

10. Inside  enter "80"

Remark: HuskyLens' coordinates



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



```
function Face_Following_Mode
  HuskyLens request data once and save into the result
  if HuskyLens check if ID 1 frame is on screen from the result then
    set xcenter to HuskyLens get X center of ID 1 frame from the result
    if xcenter < 80 then
      call Turn_Left
    else
  +
```

11. Drag **call Turn_Left**
from **f(x) Functions**
into "if then"

11

```
function Face_Following_Mode
  HuskyLens request data once and save into the result
  if HuskyLens check if ID 1 frame is on screen from the result then
    set xcenter to HuskyLens get X center of ID 1 frame from the result
    if xcenter < 80 then
      call Turn_Left
    if true then
  
```

12. Drag



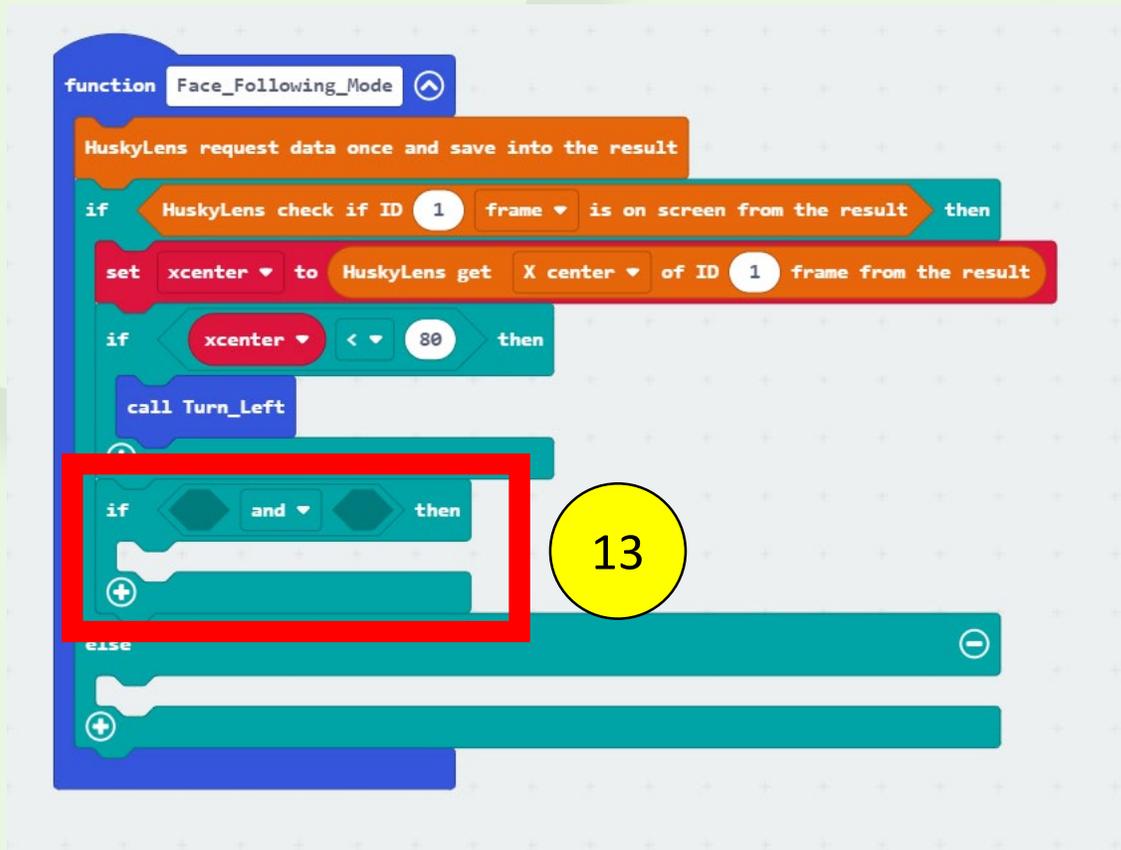
from

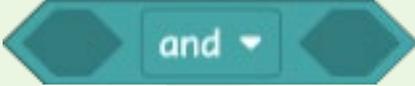


into "if then else"

```
function Face_Following_Mode
  HuskyLens request data once and save into the result
  if HuskyLens check if ID 1 frame is on screen from the result then
    set xcenter to HuskyLens get X center of ID 1 frame from the result
    if xcenter < 80 then
      call Turn_Left
    if and then
  else

```



13. Drag  from  into “if then”

```
function Face_Following_Mode  
  HuskyLens request data once and save into the result  
  if HuskyLens check if ID 1 frame is on screen from the result then  
    set xcenter to HuskyLens get X center of ID 1 frame from the result  
    if xcenter < 80 then  
      call Turn_Left  
    if xcenter ≥ 80 and xcenter ≤ 240 then  
      call Move_Forward  
    if xcenter > 240 then  
      call Turn_Right  
  end  
end
```

14

14. Set as follows

```
function Face_Following_Mode
  HuskyLens request data once and save into the result
  if HuskyLens check if ID 1 frame is on screen from the result then
    set xcenter to HuskyLens get X center of ID 1 frame from the result
    if xcenter < 80 then
      call Turn_Left
    if xcenter ≥ 80 and xcenter ≤ 240 then
      call Move_Forward
    if xcenter > 240 then
      call Turn_Right
  else
    call Move_Backward
  end if
end function
```

15

15. Set as follows

```
function Face_Following_Mode
  HuskyLens request data once and save into the result
  if HuskyLens check if ID 1 frame is on screen from the result then
    set xcenter to HuskyLens get X center of ID 1 frame from the result
    if xcenter < 80 then
      call Turn_Left
    +
    if xcenter ≥ 80 and xcenter ≤ 240 then
      call Move_Forward
    +
    if xcenter > 240 then
      call Turn_Right
    +
    iCar Stop
  +
  stop
```



16

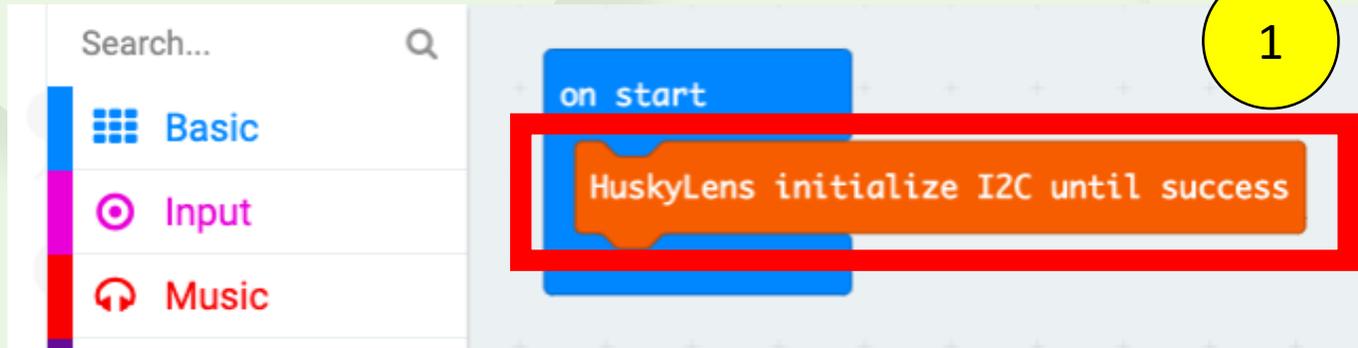
16. Drag **iCar Stop**

from **CUHK-JC iCar**

into "if then else"



Step 5: Setting Up The “on start”



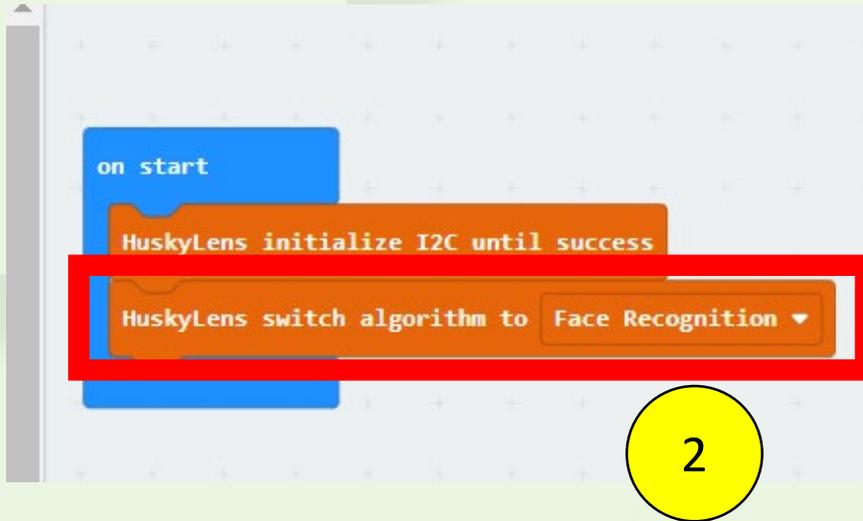
1. Drag

HuskyLens initialize I2C until success

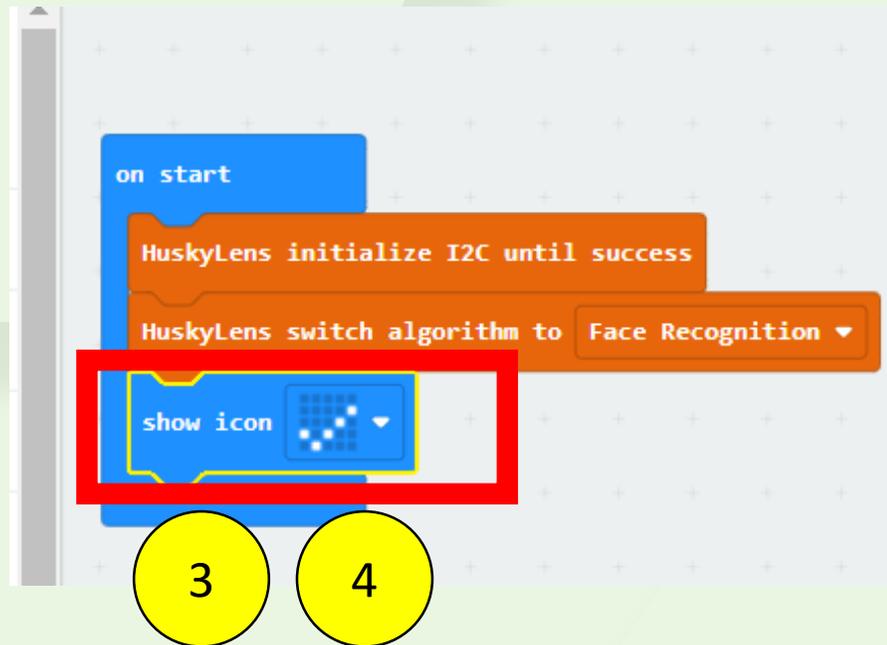
from

CUHK-JC iCar HuskyLens

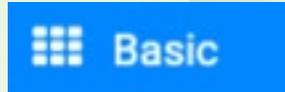
into “on start”



2. Drag  from  into "on start"



3. Drag 

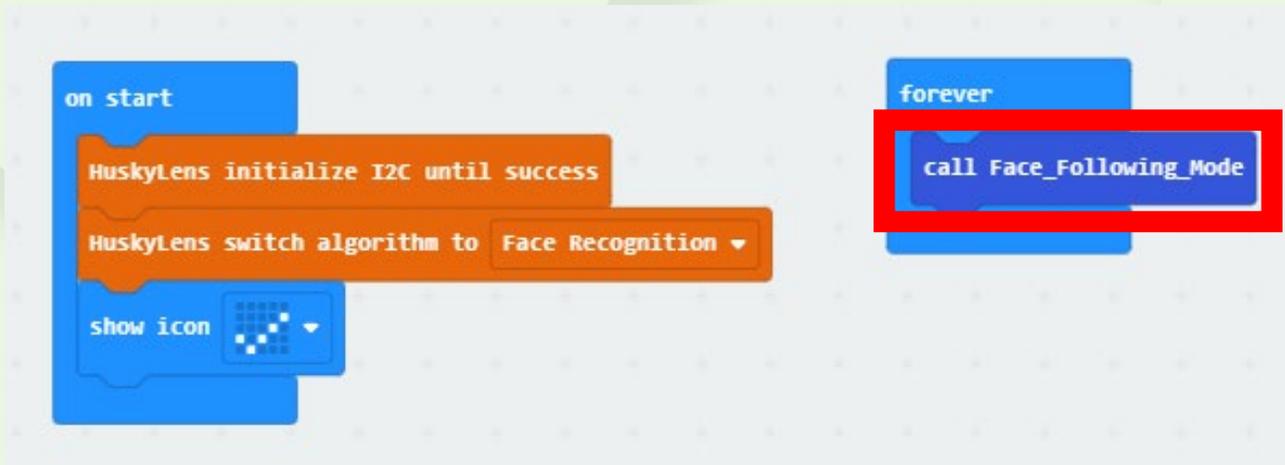
from 

into "on start"

4. Click 



Step 6: Setting Up The “forever”



1. Drag

call Face_Following_Mode

from

$f(x)$ Functions

into “forever”

1



Finished!

```

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

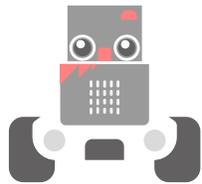
forever
  call Face_Following_Mode

function Move_Forward
  iCar Move Forward at speed 30 %

function Turn_Left
  iCar Turn Left at speed 30 %

function Turn_Right
  iCar Turn Right at speed 30 %

function Face_Following_Mode
  HuskyLens request data once and save into the result
  if HuskyLens check if ID 1 frame is on screen from the result then
    set xcenter to HuskyLens get X center of ID 1 frame from the result
    if xcenter < 80 then
      call Turn_Left
    if xcenter >= 80 and xcenter <= 240 then
      call Move_Forward
    if xcenter > 240 then
      call Turn_Right
  else
    iCar Stop
  
```



Step 7: Download The Program To CUHK iCar

Microsoft | micro:bit

Blocks JavaScript

Search...

Basic

Input

Music

Led

Radio

Loops

Logic

Variables

Math

CUHK-JC iCar HuskyLens

CUHK-JC iCar

on start

HuskyLens initialise I2C until success

HuskyLens switch algorithms to Face Recognition

show icon

forever

call Face_Following_Mode

function Move_Forward

iCar Move Forward at speed 30

function Turn_Left

iCar Turn Left at speed 30

function Turn_Right

iCar Turn Right at speed 30

function Face_Following_Mode

HuskyLens request data once and save into the result

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

set xcenter to HuskyLens get X center of ID 1 frame from the result

if xcenter < 80 then

call Turn_Left

if xcenter >= 80 and xcenter <= 240 then

call Move_Forward

if xcenter > 240 then

call Turn_Right

else

iCar Stop

Download

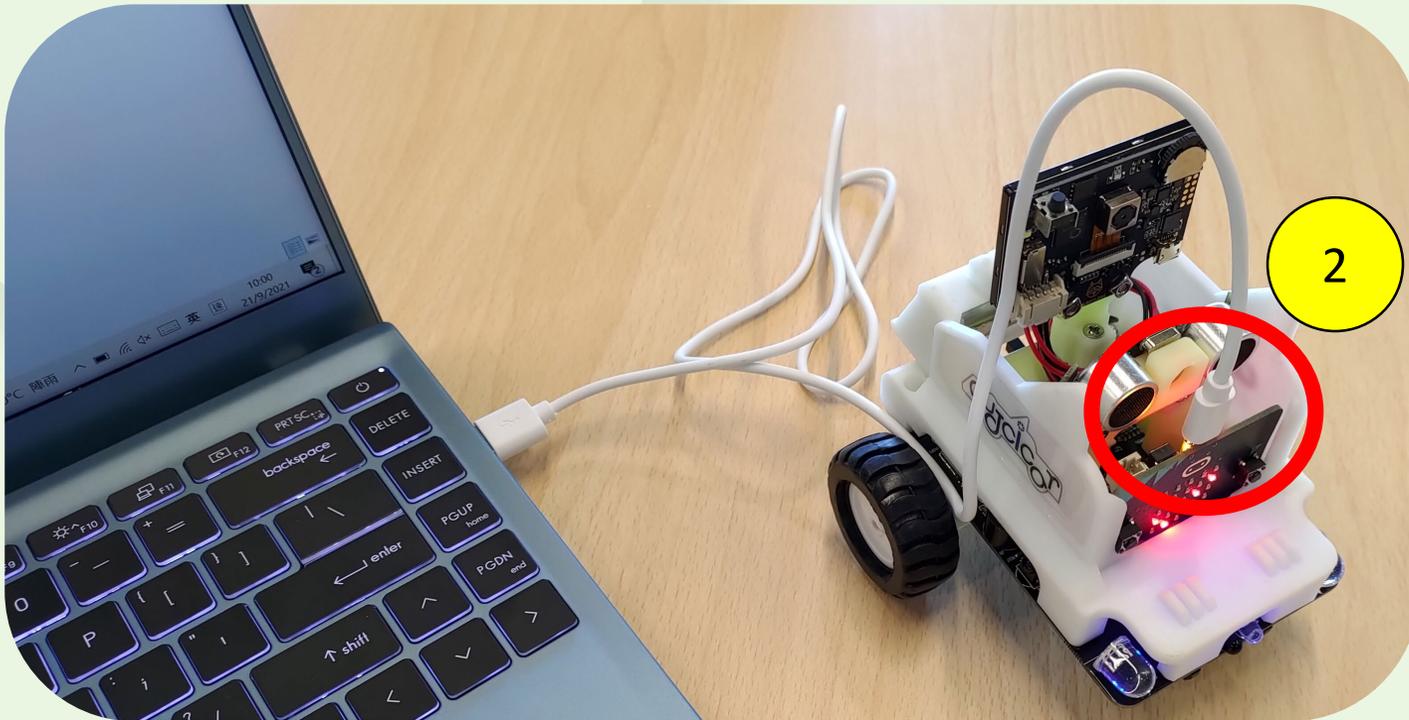
Face following

1. Click

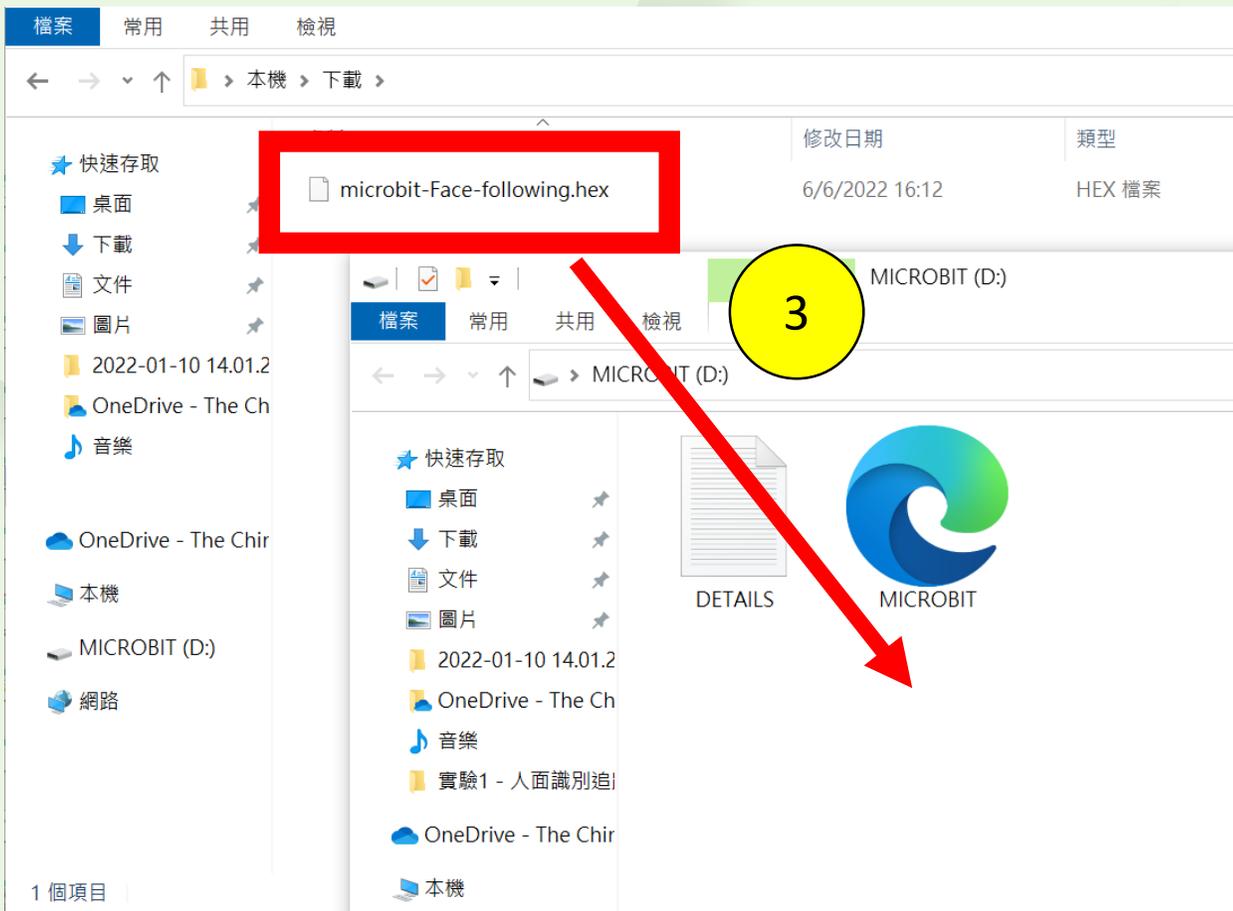
Download

1

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

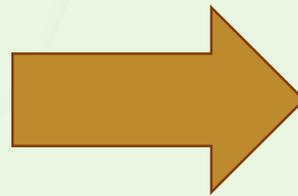


You have finished programming!

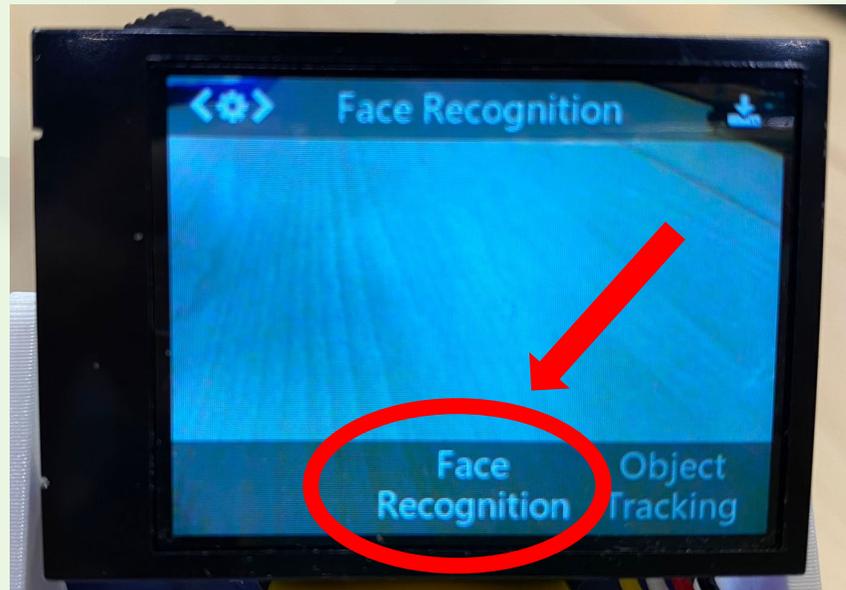
Let's start the experiment!



Switch On Your CUHK iCar



Once it is switched on, the HuskyLens will automatically adjust to Face Recognition Mode.

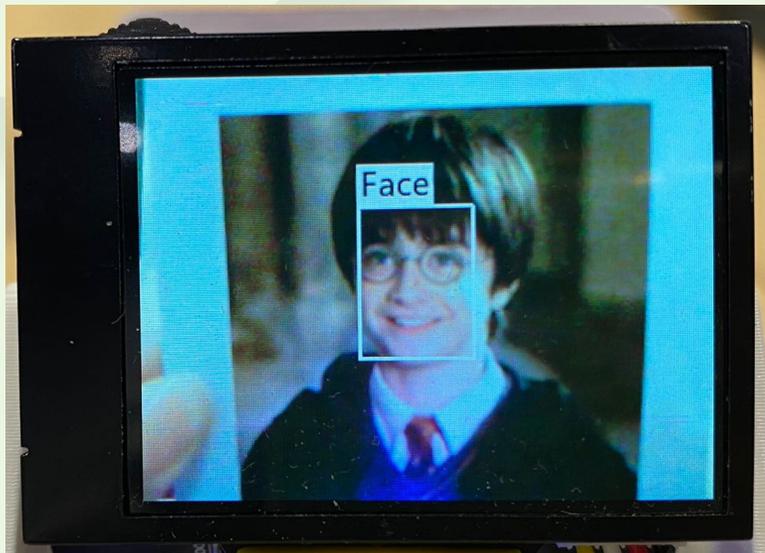


Clear The Previous Data

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



When a portrait photo is placed before the HuskyLens, a white frame will be displayed to indicate the position of the face.



Press the learning button to record the face. The frame will turn from white to blue, adding a label of "ID1".

Then, CUHK iCar will track and follow the face labelled as "ID1".



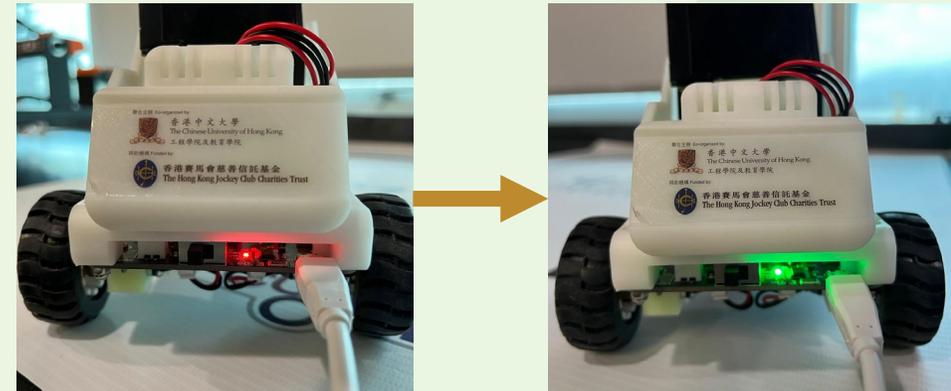
If another face is shown, a white frame will be displayed, as HuskyLens has not yet “learned” such person’s face.

If you would like the CUHK iCar to track and follow another face, you are required to clear the previous recorded face by pressing the learning button twice.



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 provided program by yourself. For details, please refer to the next slide

```

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

forever
  call Face_Following_Mode

function Move_Forward
  iCar Move Forward at speed 30 %

function Turn_Left
  iCar Turn Left at speed 30 %

function Turn_Right
  iCar Turn Right at speed 30 %

function Face_Following_Mode
  HuskyLens request data once and save into the result
  if HuskyLens check if ID 1 frame is on screen from the result then
    set xcenter to HuskyLens get X center of ID 1 frame from the result
    if xcenter < 30 then
      call Turn_Left
    if xcenter ≥ 80 and xcenter ≤ 240 then
      call Move_Forward
    if xcenter > 240 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 46