

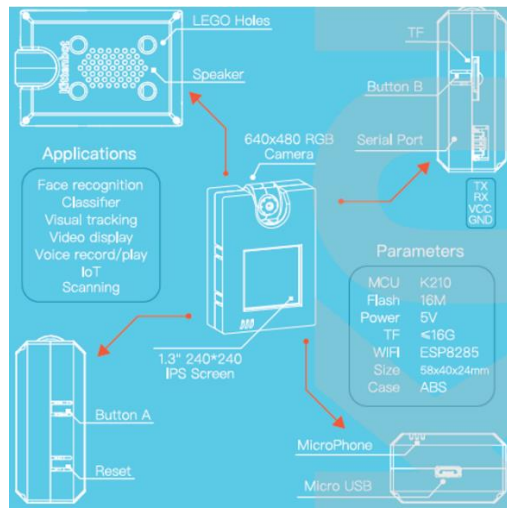
KOI Kittenbot คืออะไร?



KittenBot KOI



AI Camera KOI เป็นโมดูลปัญญาประดิษฐ์ออฟไลน์ที่พัฒนาโดย Kittybot มีวัตถุประสงค์เพื่อให้เป็นแพลตฟอร์มที่ใช้งานง่ายสำหรับนักเรียนและผู้ที่ยังชื่นชอบ KOI เป็นโมดูลแบบรวมที่รวมชิป AI, กล้อง, ไมโครโฟน, ลำโพง, จอแสดงผล IPS และชิป Wi-Fi ไว้ด้วยกัน โดยมีความสามารถมากมาย ทั้งการจดจำภาพ การตรวจจับและการจดจำใบหน้า การจัดประเภทวัตถุ การรู้จำเสียงและข้อความเป็นคำพูด เป็นต้นส่วนประกอบของ KOI



1. เลนส์แบบปรับหมุนได้ (Rotate the lens) เลนส์หมุนได้ตั้งแต่ 0 ถึง 180 องศา

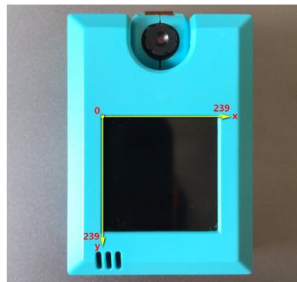
2. กล้องหน้า (Camera front) กล้องอยู่หน้ากล้องในทิศทางเดียวกับหน้าจอ โดยทั่วไปจะใช้ส่วนหน้าในการโต้ตอบการติดตามใบหน้า การโต้ตอบการติดตามวัตถุ ข้อมูลกล้องสามารถดูได้โดยตรงจากหน้าจอ



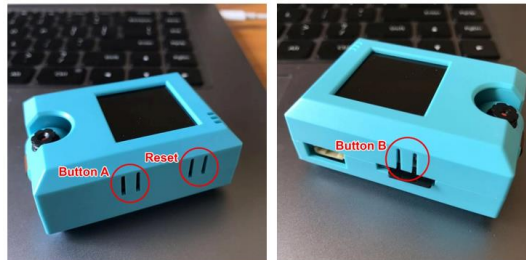
3. กล้องหลัง (Camera rear) กล้องและหน้าจออยู่ในทิศทางตรงกันข้าม ดังแสดงในรูป โดยทั่วไปจะใช้สำหรับสแกนบาร์โค้ด, QR Code, การติดตามเส้น ฯลฯ



4. หน้าจอแสดงผล (Screen and screen coordinate system) หน้าจอเป็นหน้าจอ IPS ความละเอียดสูง 240 x 240 พิกเซล โดยมีระบบพิกัดเริ่มต้นที่มุมซ้ายบนของหน้าจอ ใช้เพื่อแสดงสตรีมข้อมูลแบบเรียลไทม์ของกล้องและข้อมูลเชิงโต้ตอบ



5. ปุ่ม A, B และ ปุ่ม Reset มีปุ่มตั้งโปรแกรมได้ที่ด้านซ้ายและด้านขวาของโมดูล คล้ายกับปุ่ม Micro:bit A และ B



6. ไมโครโฟน (microphone) มีไมโครโฟนอยู่ที่ด้านล่างซ้ายของหน้าจอ ระยะเวลาวิทยุประมาณ 50 ชม.



7. พอร์ตข้อมูล USB (USB data port) พอร์ตข้อมูลนี้ใช้เพื่ออัปเดตเฟิร์มแวร์ของโมดูลหรือจ่ายไฟให้กับโมดูล



8. พอร์ตอนุกรม PH 2.0 4 PIN (PH2.0 4PIN serial port) ใช้สำหรับการสื่อสารแบบอนุกรมกับบอร์ดไมโครคอนโทรลเลอร์ Microbit, Arduino หรือบอร์ดอื่น ๆ ที่เป็นอนุกรม



9. ช่องเสียบการ์ด (TF card slot) เป็นช่องเสียบการ์ดป้อนข้อและรองรับหน่วยความจำสูงสุดคือ 16GB. ซึ่งสามารถจัดเก็บเสียง รูปภาพ โมเดลการเรียนรู้ของเครื่อง ฯลฯ

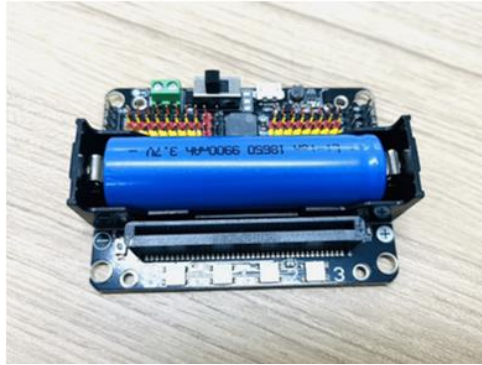


10. ลำโพง (Speaker) สำหรับเล่นไฟล์เสียงในการ์ดหน่วยความจำ



การต่ออุปกรณ์

1. ใส่ถ่านชาร์จบนบอร์ดขยาย Robotbit โดยใส่ขั้ว + และขั้ว - ให้ถูกต้อง



2. เสียบบอร์ด Micro: bit เข้ากับบอร์ดขยาย Robotbit

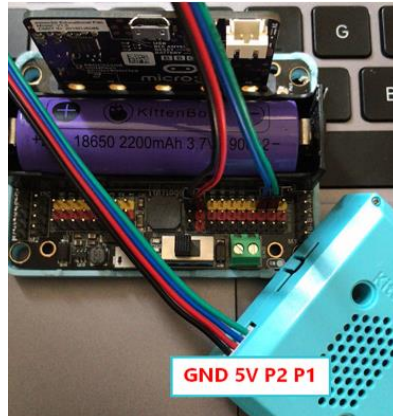


3. เสียบสาย Micro USB เชื่อมต่อบอร์ด Micro:bit กับคอมพิวเตอร์

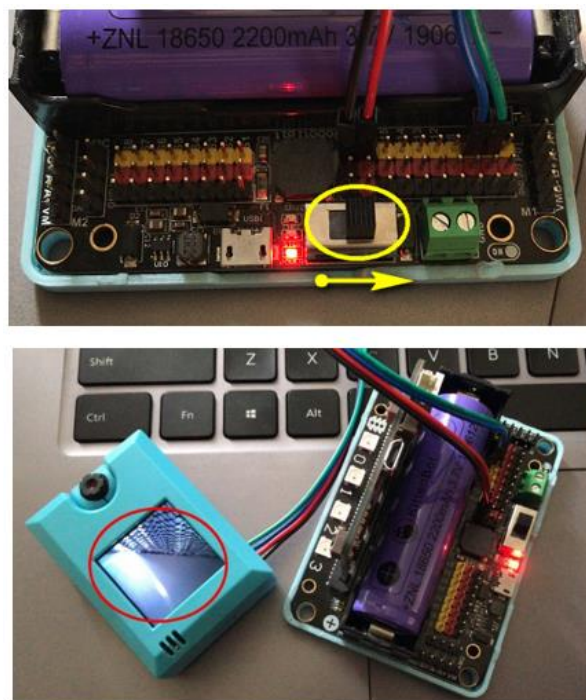


4. เชื่อมต่อ KOI Kittenbot กับ Robotbit

- GND ของ KOI (สายสีดำ) ต่อเข้ากับขา GND ของ Robotbit
- VIN ของ KOI (สายสีแดง) ต่อเข้ากับขา 5V ของ Robotbit
- Tx ของ KOI (สายสีน้ำเงิน) ต่อเข้ากับขา P2 ของ Robotbit
- Rx ของ KOI (สายสีเขียว) ต่อเข้ากับขา P1 ของ Robotbit
- เสียบสาย Micro USB เพื่อจ่ายไฟเลี้ยงให้ KOI

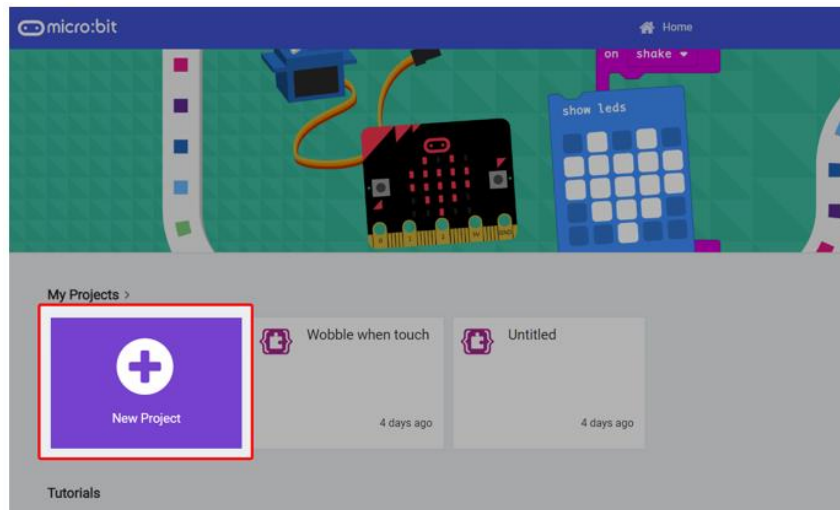


5. เปิดสวิตช์ Robotbit ไฟที่บอร์ดขยาย Robotbit จะติด และหน้าจอ KOI จะสว่างขึ้น

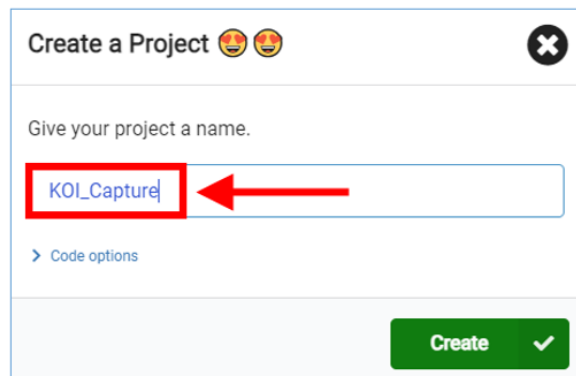


เขียนโปรแกรมด้วย Makecode Online

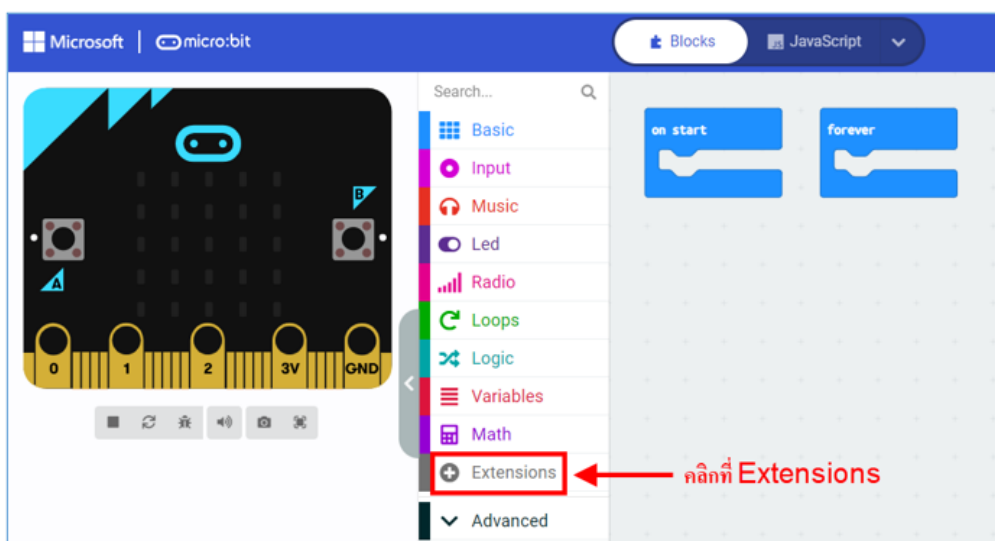
1. เข้าเว็บไซต์ <https://makecode.microbit.org/> แล้วคลิกเมนู New Project



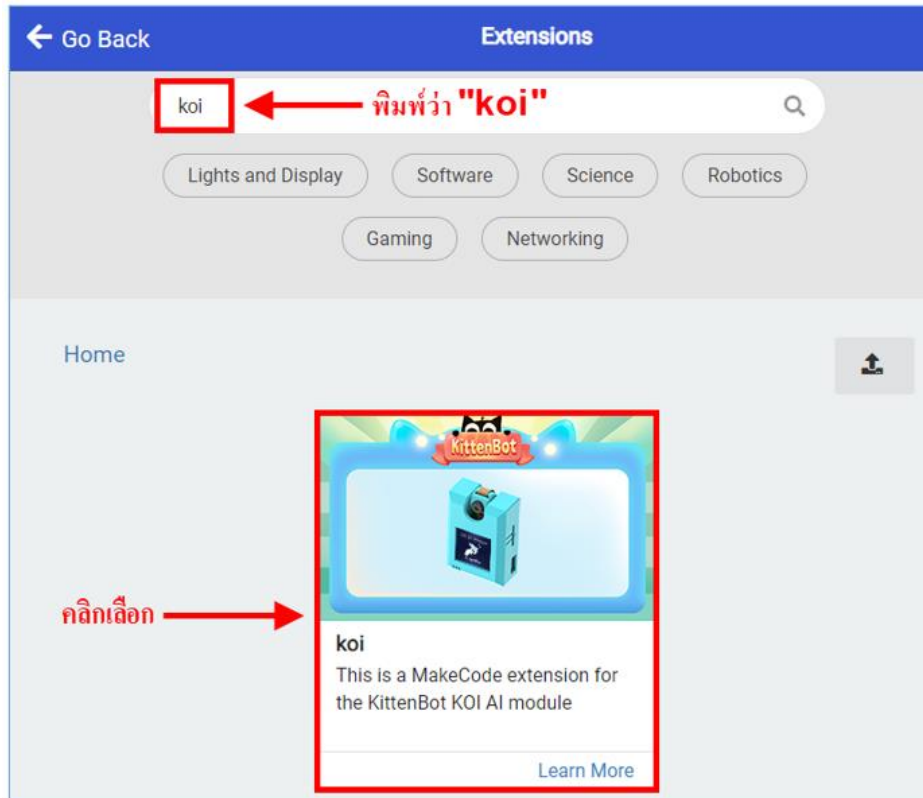
2. ตั้งชื่อโปรเจกต์ จากนั้นคลิกปุ่ม Create



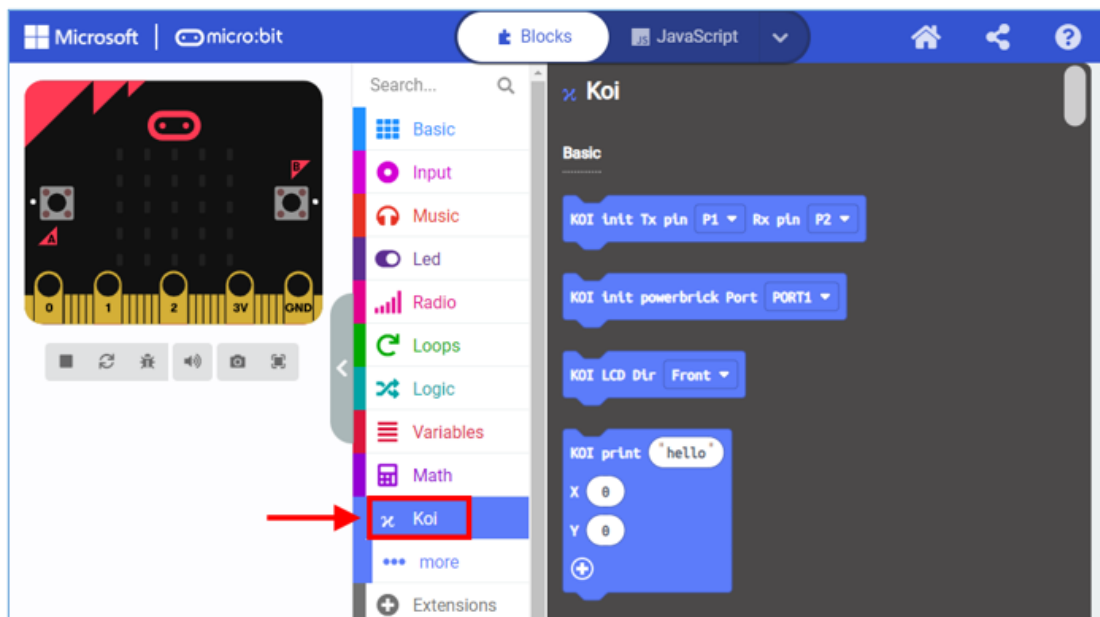
3. ดาวน์โหลด Extensions ที่ชื่อว่า “koi” โดยคลิกที่ Extensions



4. พิมพ์คำว่า “koi” จากนั้นคลิกเลือก koi เพื่อทำการเพิ่ม Extension ในการใช้งานกล้อง KOI

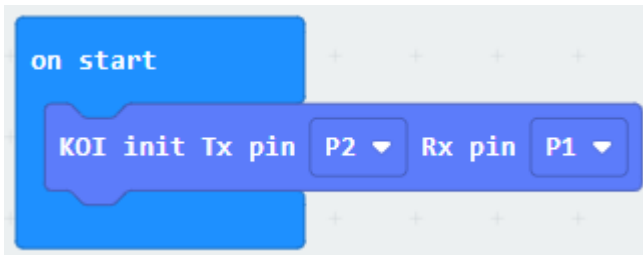


5. จะปรากฏเมนู Koi เข้ามาในหน้าจออินเทอร์เน็ตเฟส ดังรูป



เริ่มต้นการใช้งานต้องทำให้บอร์ด Micro:bit รู้จักกับอุปกรณ์ ก่อน เพื่อจะได้สั่งงานได้ถูกต้อง

ใส่บล็อกคำสั่ง



-กำหนดค่า Tx = P2 , Rx = P1

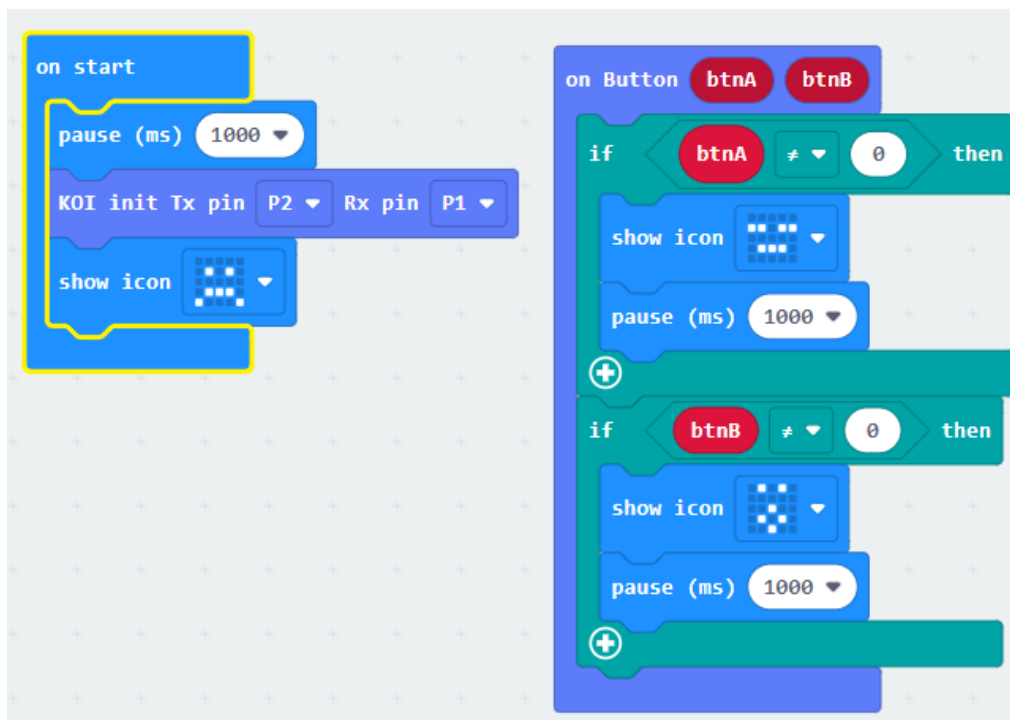
การทำงานของปุ่ม A และ B บน Koi



สถานะของปุ่มของปุ่ม A และปุ่ม B ตามลำดับ และปุ่ม Reset

สถานะคือ 1 เมื่อกดปุ่ม ปรกติปุ่มจะมีสถานะเป็น 0 ถ้าเมื่อใดที่มีการกดปุ่ม จะมีสถานะเป็น 1

ตัวอย่างของการเขียนโค้ดคำสั่ง



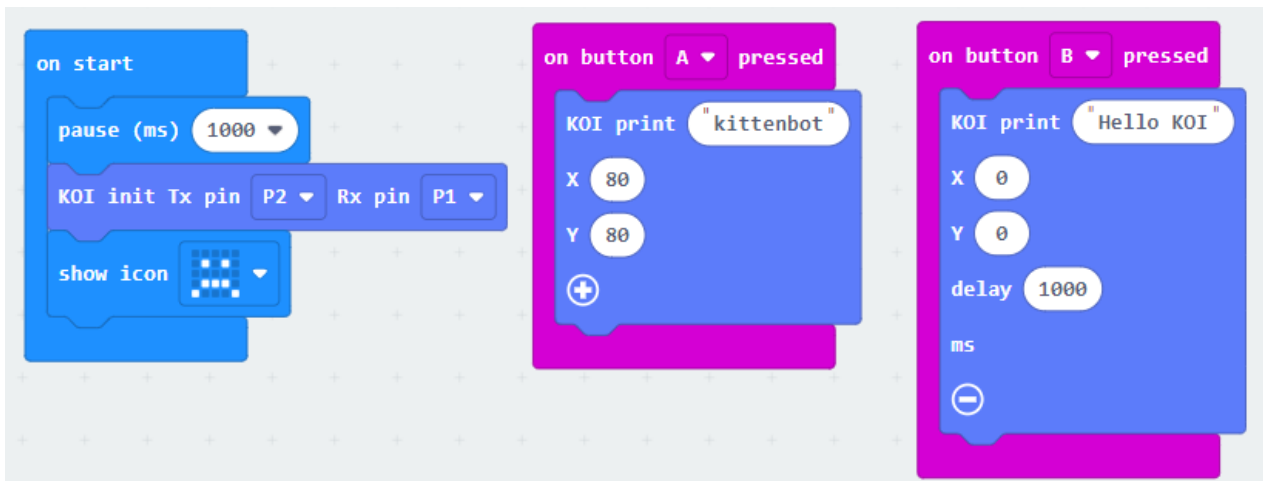
ภาพหน้าจอและการแสดงผล

กรอกช่องว่าง X และ Y ด้วย 0-239 (ดูภาพด้านล่าง) เพื่อกำหนดตำแหน่งเริ่มต้นของข้อมูลที่แสดง



บล็อกการพิมพ์จะแสดงสตริงที่พิกัดเฉพาะ โดยจะแสดงข้อความเป็นเวลา 1 วินาทีตามค่าเริ่มต้น ระยะเวลาสามารถเปลี่ยนแปลงได้ด้วยการหน่วงเวลา

ตัวอย่างของการเขียนโค้ดคำสั่ง



ดาวน์โหลดโปรแกรมลงใน Microbit กดปุ่ม A ของ Microbit จากนั้นหน้าจอ KOI จะปรากฏข้อความ kittenbot กดปุ่ม B ของ Microbit จากนั้น หน้าจอ KOI จะปรากฏข้อความ Hello KOI เป็นเวลา 1 วินาที

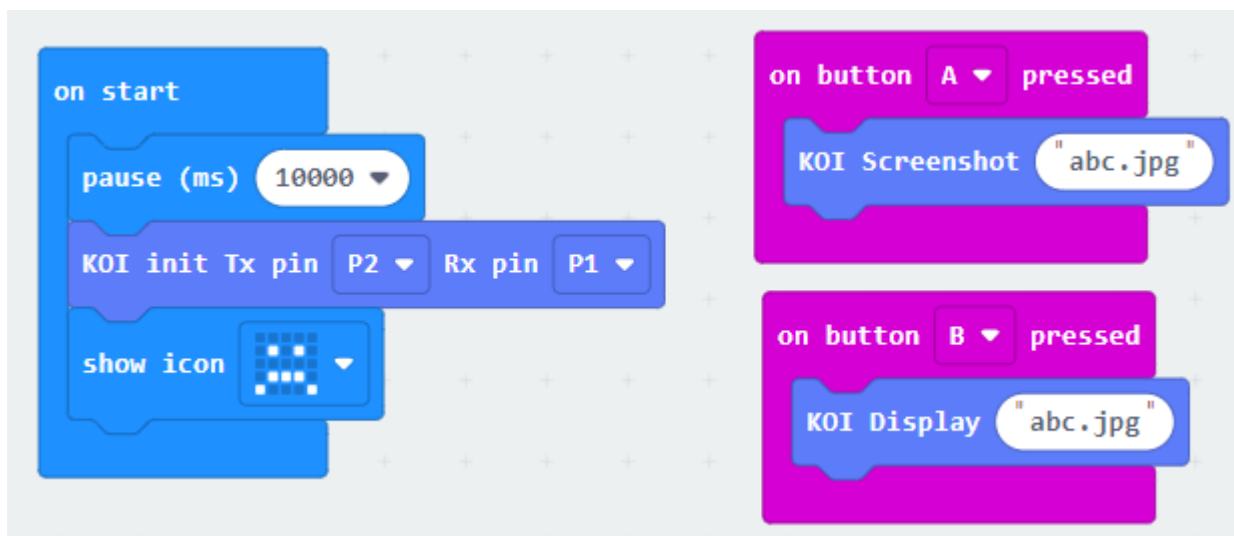
การวางแนวการแสดงผล

เนื่องจาก KOI ไม่สามารถตรวจจับทิศทางของกล้องได้ จึงต้องตั้งค่าการวางแนวการแสดงผลด้วยตนเองเพื่อให้ภาพสอดคล้องกับทิศทางของกล้อง



การถ่ายภาพและการแสดงผลภาพ

ตัวอย่างของการเขียนโค้ดคำสั่ง

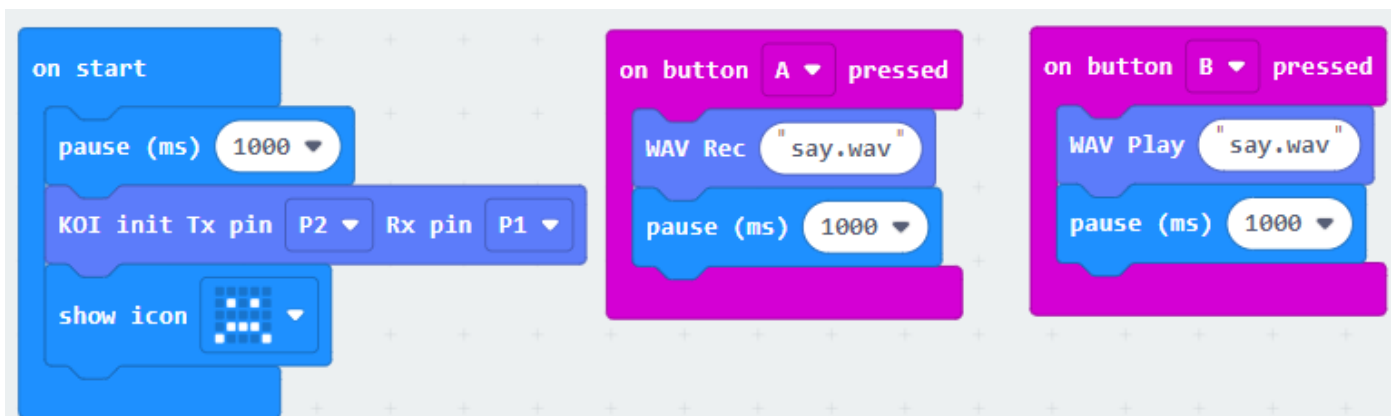


กดปุ่ม A ของ Micro:bit เพื่อ "ถ่ายภาพ"

กดปุ่ม B ของ Micro:bit เพื่อแสดงภาพที่คุณเพิ่งถ่าย

การบันทึกและการเล่นเสียง

ตัวอย่างของการเขียนโค้ดคำสั่ง

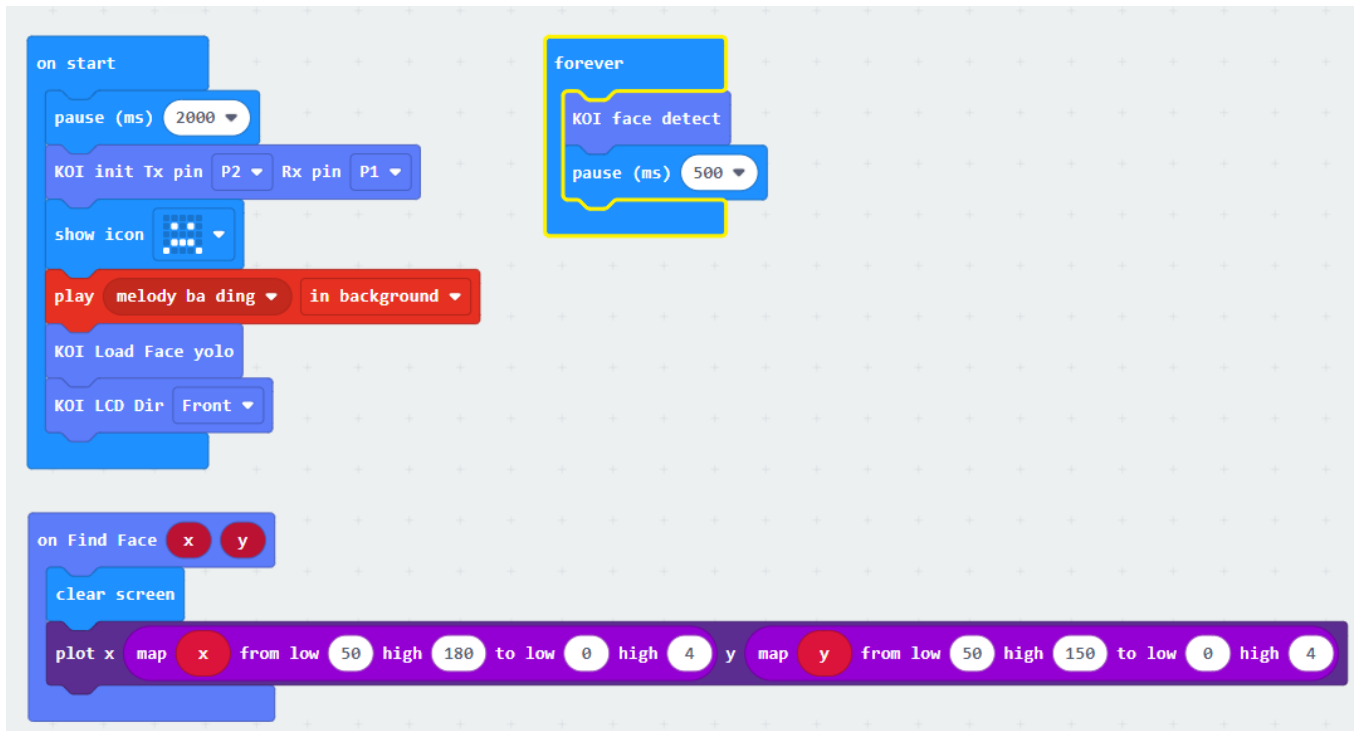


กดปุ่ม A ของ Micro:bit เพื่อบันทึกคำพูดของคุณลงในไมโครโฟน

เมื่อคุณกดปุ่ม B ของ Micro:bit ลำโพงจะเล่นเสียงที่คุณเพิ่งบันทึก

การติดตามใบหน้าว่าอยู่ตำแหน่งไหน

โค้ดคำสั่ง



```
on start
  pause (ms) 2000
  KOI init Tx pin P2 Rx pin P1
  show icon
  play melody ba ding in background
  KOI Load Face yolo
  KOI LCD Dir Front

forever
  KOI face detect
  pause (ms) 500

on Find Face x y
  clear screen
  plot x map x from low 50 high 180 to low 0 high 4 y map y from low 50 high 150 to low 0 high 4
```

รอนจนกว่าจะมีกรอบสีเขียวขึ้นที่หน้าจอ KOI แล้วให้เราเคลื่อนที่ใบหน้าของเราไปมา แล้วดูที่บอร์ด Micro:bit จะมีจุดหลอดไฟเคลื่อนที่ไปยังตำแหน่งตรงกับหน้าจอของเรา

การอ่านรหัส QR Code



```
on start
  pause (ms) 2000
  KOI init Tx pin P2 Rx pin P1
  show icon
  play tone Middle C for 1 beat until done
  KOI LCD Dir Back

on button A pressed
  KOI QR code

on QR code link
  show string link
  KOI print link
  X 0
  Y 0
```

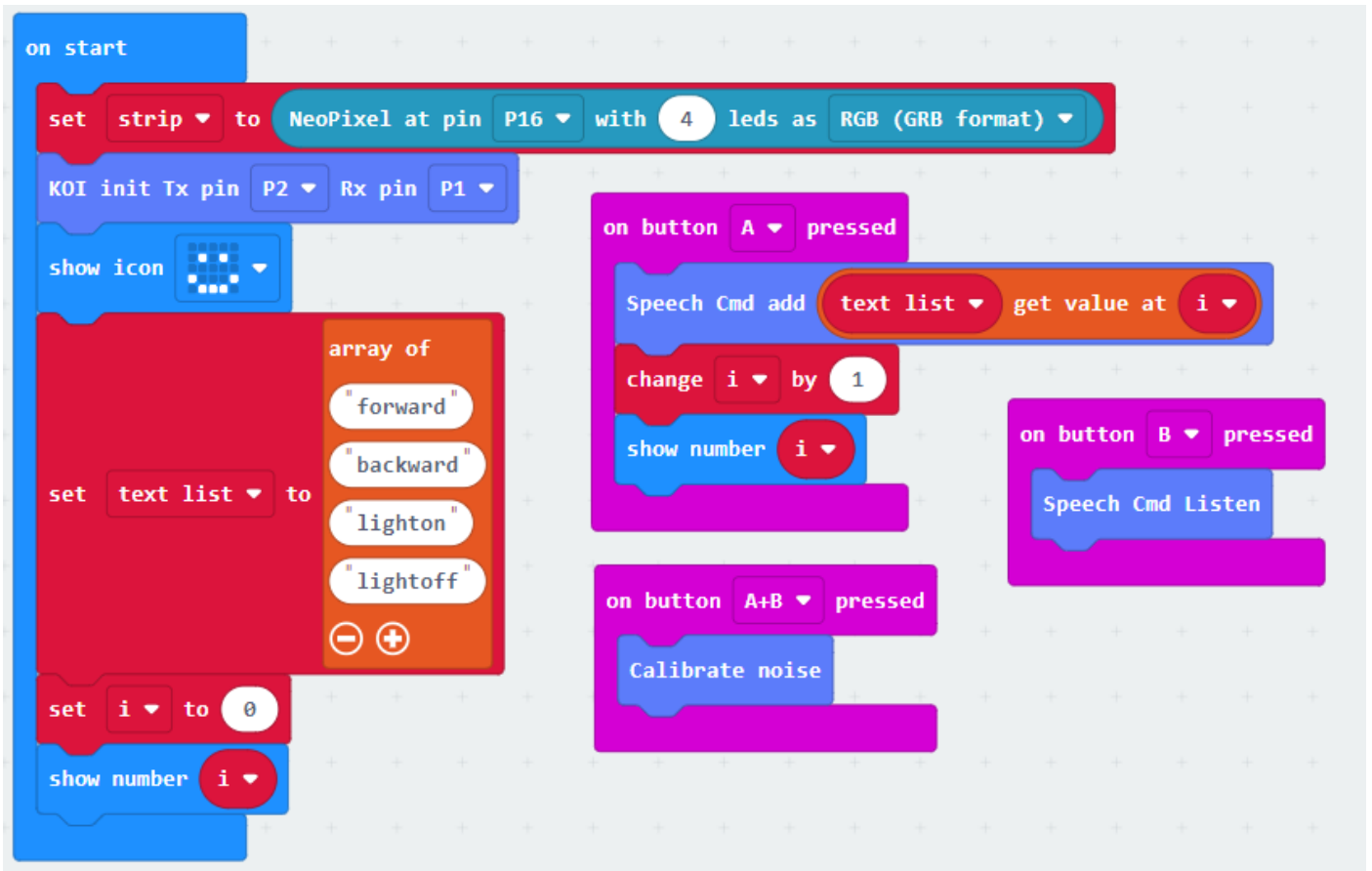
ต้องทำการสร้าง QR Code จากโปรแกรมอะไรก็ได้เช่น Canva
เมื่อต้องการจะส่ง QR Code ให้กดปุ่ม A อันนี้ให้แสดงข้อความที่เราใส่ใน QR Code

การประยุกต์ใช้ QR Code ทำงาน

การเรียนรู้ของเครื่อง – การจดจำวัตถุแบบกำหนดเอง

การจดจำเสียงและสั่งการด้วยเสียง

ติดตั้ง Extensions KOI , Robotbit

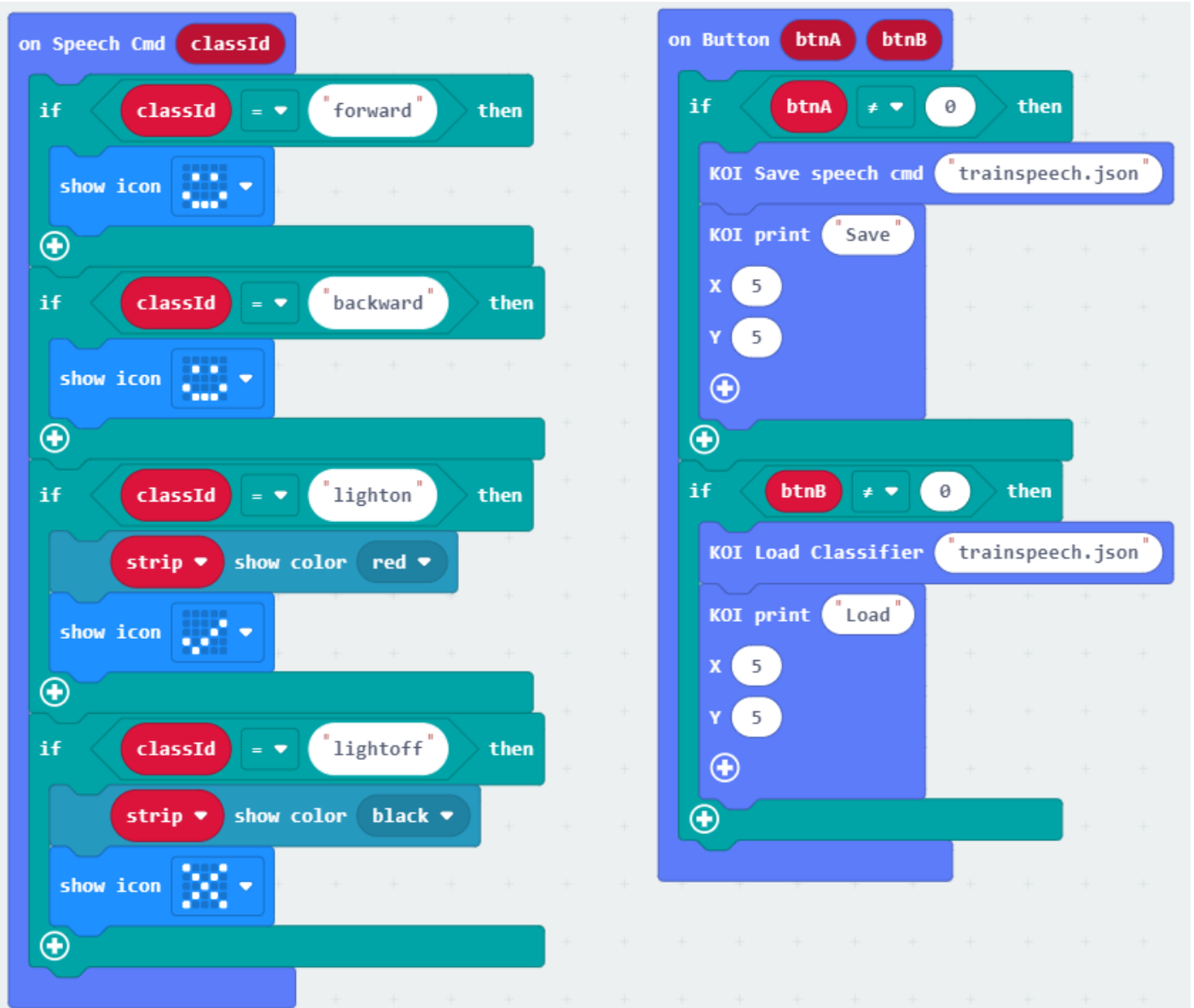


ตั้งค่าที่เราต้องการเทรน ในส list ตั้งค่าตัวแปรเพื่อทำการเปลี่ยนทีละคำ

กดปุ่ม A เป็นการเทรน คำไปเรื่อยๆ

กดปุ่ม B เป็นการฟังคำสั่งที่เราเทรนไป

กดปุ่ม A+B เพื่อเริ่มต้นการ Calibrate noise



จากนั้นนำคำสั่งที่ได้จากการเทรน มาใส่คำสั่งว่าให้แต่ละคำทำงานอะไร
กลุ่ม A ที่ KOI เป็นการบันทึกสิ่งที่เราเทรนไว้
กลุ่ม B ที่ KOI เป็นการโหลดเสียงที่เราทำการเทรนไว้

การเริ่มต้นใช้ Micro:bit V2

การใช้ Mic ตรวจสอบความดังที่เข้ามา

```
on start
  show icon [Micro:bit icon]
  set strip to NeoPixel at pin P16 with 4 leds as RGB (GRB format)

forever
  show number [sound level]
  if [sound level] > 100 then
    show icon [Micro:bit icon]
    pause (ms) 2000
    strip show color green
  else
    show icon [Micro:bit icon]
    pause (ms) 2000
    strip show color black
```

The image shows a Scratch script for a Micro:bit V2. The script starts with an 'on start' block containing a 'show icon' block and a 'set strip to NeoPixel at pin P16 with 4 leds as RGB (GRB format)' block. Below this is a 'forever' loop. Inside the loop, there is a 'show number' block with 'sound level' as the value. This is followed by an 'if' block: 'if [sound level] > 100 then'. The 'then' branch contains a 'show icon' block, a 'pause (ms) 2000' block, and a 'strip show color green' block. The 'else' branch contains a 'show icon' block, a 'pause (ms) 2000' block, and a 'strip show color black' block.

การใช้คำสั่งเสียงให้รถหุ่นยนต์ Maqueen

ฝั่ง Microbit V2 + Robotbit

```
on start
  radio set group 13
  KOI init Tx pin P2 Rx pin P1
  show icon [robot]
  array of ["forward", "backward", "left", "right"]
  set text list to [array]
  set i to 0
  show number i

on button A pressed
  Speech Cmd add [text list] [get value at i]
  change i by 1
  show number i

on button B pressed
  Speech Cmd Listen

on button A+B pressed
  Calibrate noise

on Button btnA btnB
  if btnA != 0 then
    KOI Save Classifier "car.json"
    KOI print "save"
    X 0
    Y 0
  if btnB != 0 then
    KOI Load Classifier "car.json"
    KOI print "load"
    X 0
    Y 0
```

```
on Speech Cmd classId
  radio set group 13
  if classId = "forward" then
    radio send string "forward"
    show string "f"
  +
  if classId = "backward" then
    radio send string "backward"
    show string "b"
  +
  if classId = "left" then
    radio send string "left"
    show string "l"
  +
  if classId = "right" then
    radio send string "right"
    show string "r"
  +
```

ผังรหัสนยนต์ Microbit v1 + Maqueen

```
on start
  radio set group 13
  show icon [grid icon]

on radio received receivedString
  set command to receivedString
```

```
function backward
  motor all move Backward at speed 50
  pause (ms) 1000

function forward
  motor all move Forward at speed 50
  pause (ms) 1000

function left
  motor left move Forward at speed 0
  motor right move Forward at speed 50
  pause (ms) 1000
  motor all move Forward at speed 50

function right
  motor left move Forward at speed 50
  motor right move Forward at speed 0
  pause (ms) 1000
  motor all move Forward at speed 50

function stop
  motor all stop
```

การเขียนคำสั่งให้รถหุ่นยนต์ทำงาน

```
forever
  if (command = "forward") then
    call forward
    pause (ms) 2000
    set command to "stop"
  else if (command = "backward") then
    call backward
    pause (ms) 2000
    set command to "stop"
  else if (command = "left") then
    call left
    pause (ms) 2000
    set command to "stop"
  else if (command = "right") then
    call right
    pause (ms) 2000
    set command to "stop"
  else if (command = "stop") then
    call stop
```

The image shows a Scratch script for a robot's movement logic. It is enclosed in a 'forever' loop. The script uses a series of 'if' and 'else if' blocks to check the value of a variable named 'command'. If the command is 'forward', it calls a 'forward' block, pauses for 2000 ms, and then sets 'command' to 'stop'. Similar logic is applied for 'backward', 'left', and 'right' commands. Finally, an 'else if' block checks for the 'stop' command and calls a 'stop' block. The script ends with a plus sign icon, indicating it can be expanded.