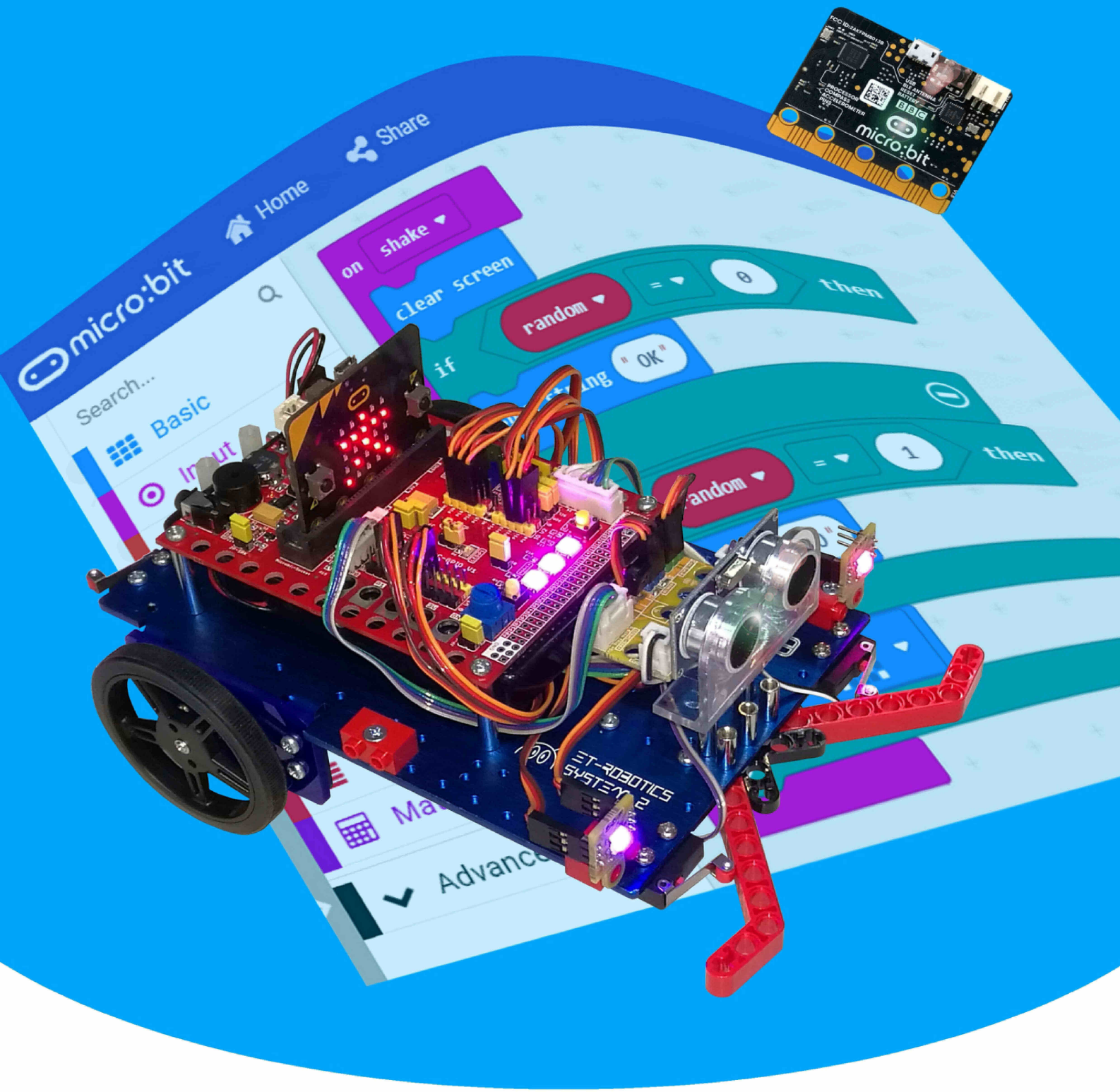
คู่มือเรียนรู้ และใช้งาน

ET-BBC ROBOT

Manual V1.0 : 25/FEB/2020

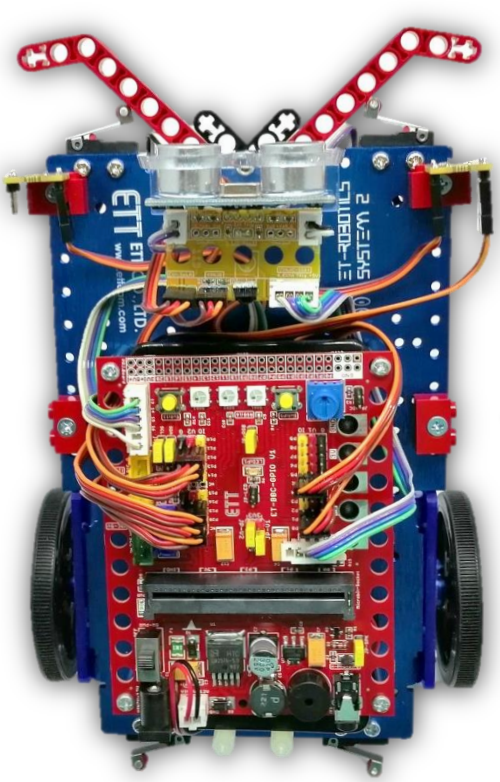


บริษัท อีทีที จำกัด
TEL : 02-7121120 FAX : 02-3917216
e-mail : sale@etteam.com

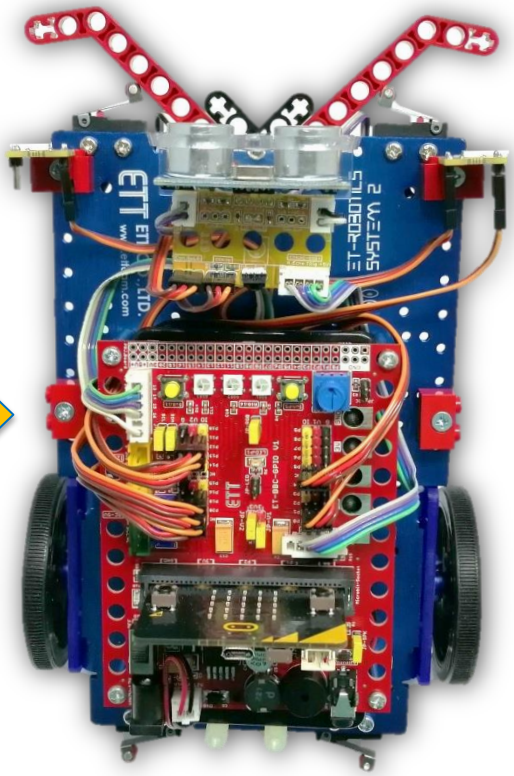
facebook.com/ett.co.th
ett.thai

www.etteam.com รายละเอียดการสั่งซื้อ LINE ID : etteam1 แผนที่ ETT MAP

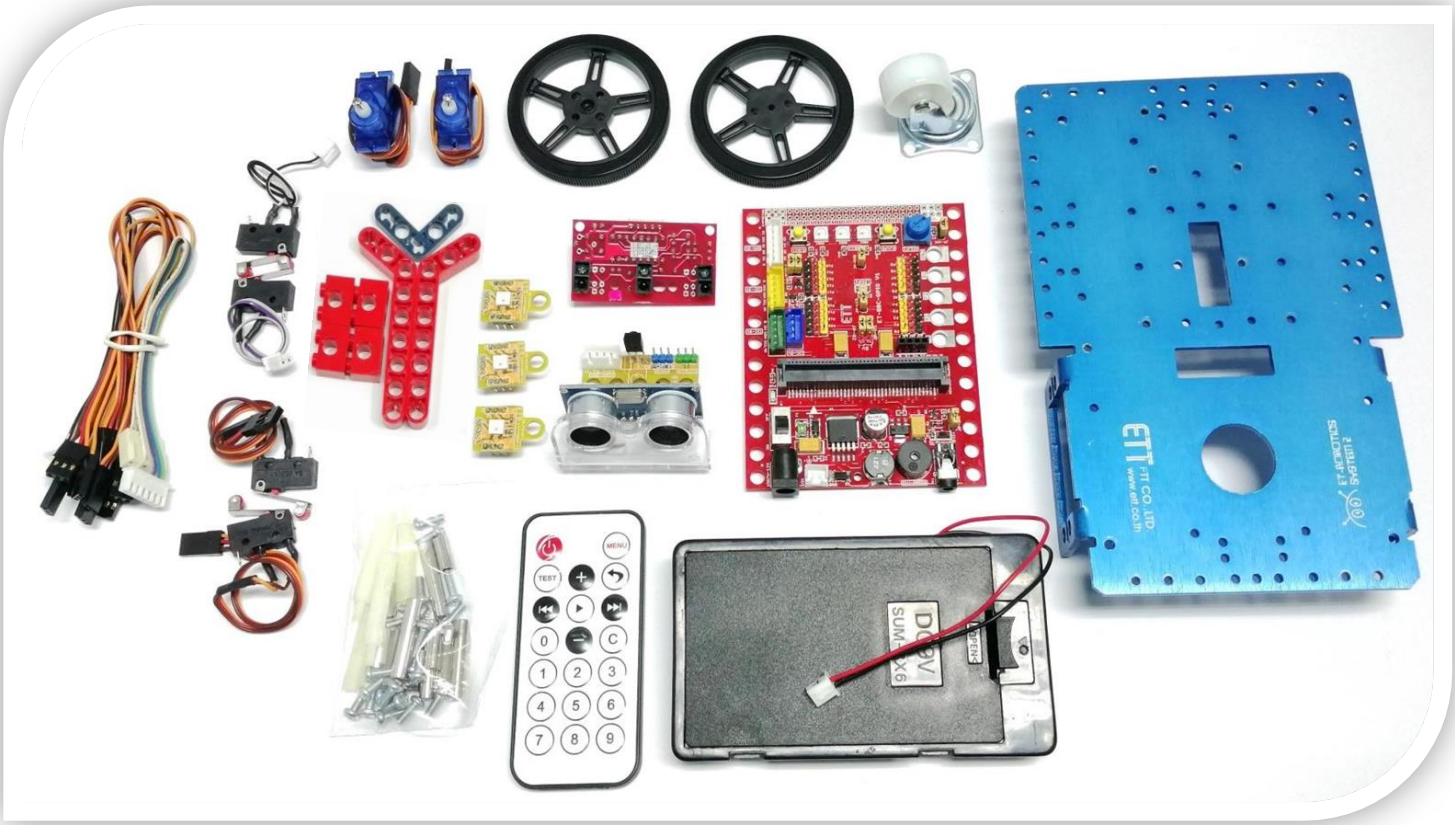
ส่วนประกอบ ET-BBC ROBOT



รุ่น ET-BBC ROBOT



รุ่น ET-BBC ROBOT + Micro:bit



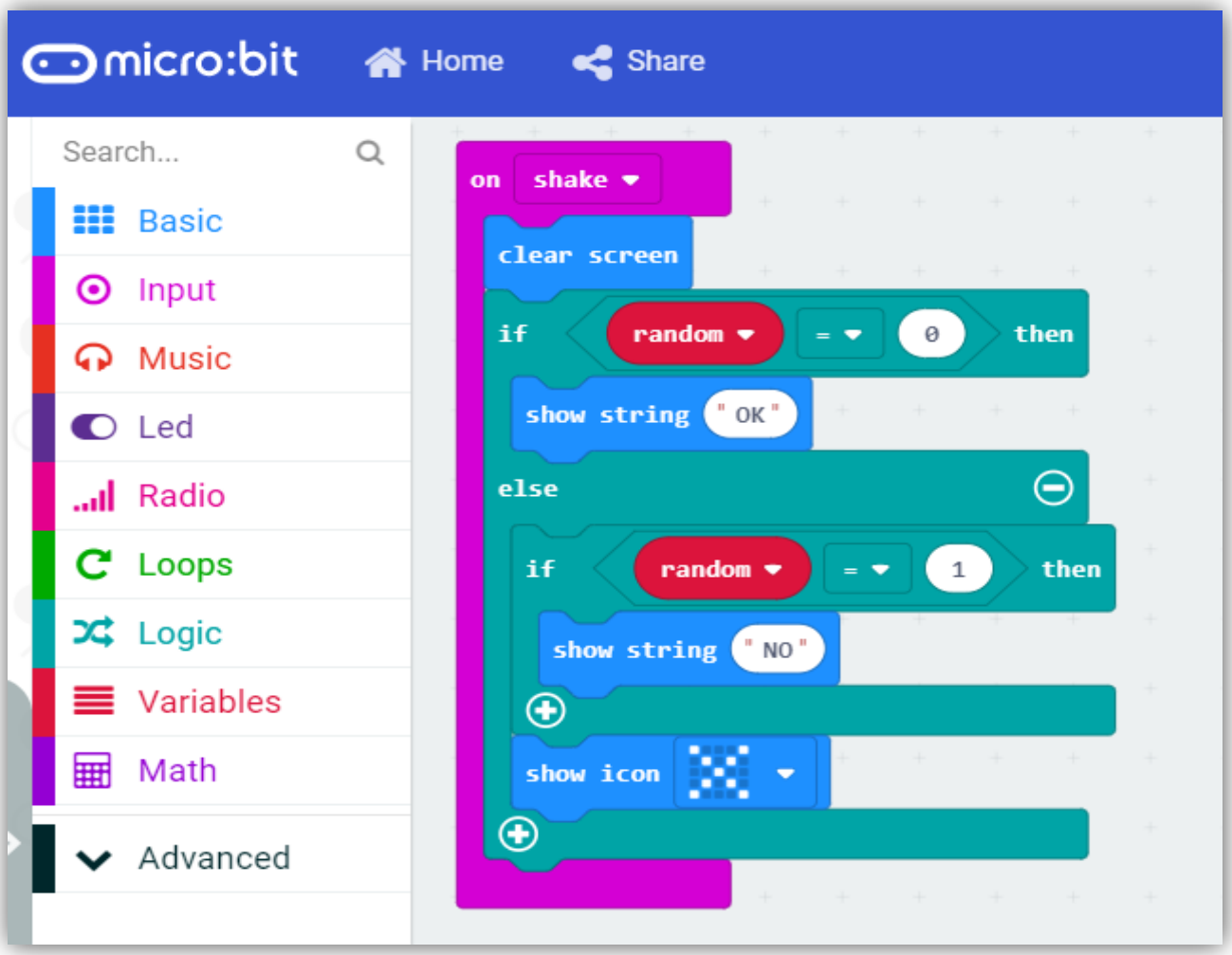
1.แนะนำ ET-BBC ROBOT

ET-BBC ROBOT เป็น Robot ประเภทรถสามล้อ สร้างขึ้นสำหรับการเรียนรู้ทดลองในเรื่องของการควบคุมหุ่นยนต์ แบบอัตโนมัติ เช่น เดินตามเส้น , หลบหลีกสิ่งกีดขวาง และ การควบคุมหุ่นยนต์ แบบ Manual ด้วย IR-Remote Control เป็นต้น ซึ่งจะใช้การเขียนโปรแกรม ควบคุมการทำงานของ Robot ด้วยโปรแกรมแบบ Block โดยใช้ Make Code Editor (On-Line) เป็นหลัก

สำหรับ Robot รุ่นนี้จะมีโครงสร้างตัวถังเป็นโครงอะลูมิเนียมแข็งแรง ประกอบยึดกับบอร์ด ET-BBC GPIO ซึ่งเป็นบอร์ดแปลง Pin IO สำหรับรองรับบอร์ดควบคุม Micro:bit ที่จะนำมาใช้กับ Robot เพื่อใช้เป็นจุดต่อสายสัญญาณของอุปกรณ์ IO และ Sensor ต่างๆ ที่อยู่รอบๆข้าง Robot เช่น Micro-Servo , LED-RGB , Micro-Switch , Ultrasonic , IR-Receive ไปยังบอร์ด Control Micro:bit

ลักษณะของ ET-BBC ROBOT

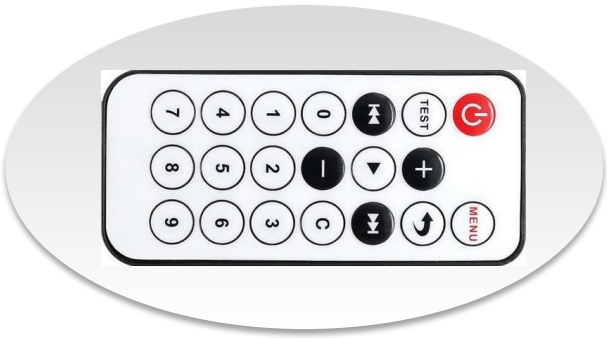
- รองรับบอร์ด Control Micro:Bit เพื่อใช้สำหรับเขียนโปรแกรมในการควบคุม Robot
- รองรับการพัฒนาโปรแกรมแบบ Block ด้วยโปรแกรม MakeCode Editor แบบ Online ลักษณะโปรแกรมแสดงดังรูปด้านล่าง หรือจะ พัฒนาด้วยโปรแกรมอื่นๆที่รองรับการใช้งานของบอร์ด Micro:bit ก็ได้ แต่ทาง ETT จะไม่ได้ทำตัวอย่างโปรแกรมไว้รองรับ ตัวอย่างโปรแกรมที่ทำไว้รองรับจะเป็นแบบ Block ที่เขียนบน MakeCode Editor ตามที่แสดงในรูปด้านล่างเท่านั้น



- อุปกรณ์ IO ที่มีให้เล่นและเรียนรู้ในการเขียนโปรแกรมควบคุมบน ET-BBC ROBOT ได้แก่

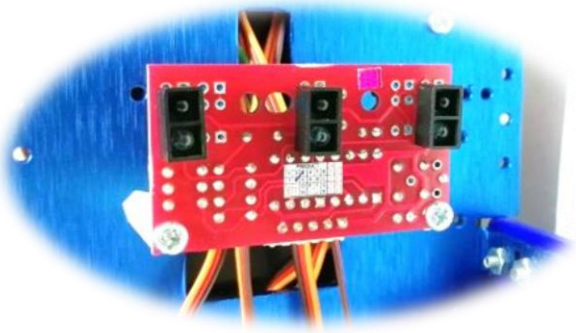
1) Sensor Ultrasonic (ตรวจจับวัตถุ)

2) IR-Remote Receive(IR-ตัวรับ)

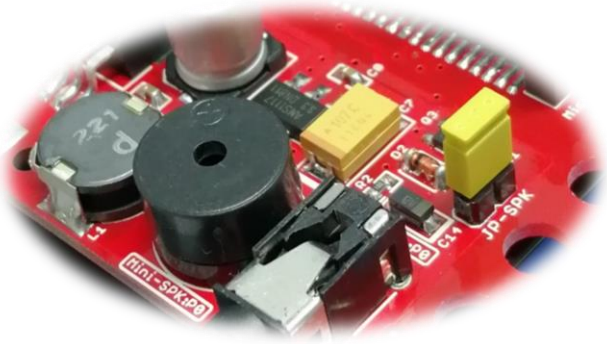


3) IR-Remote Key (IR-ตัวส่ง)

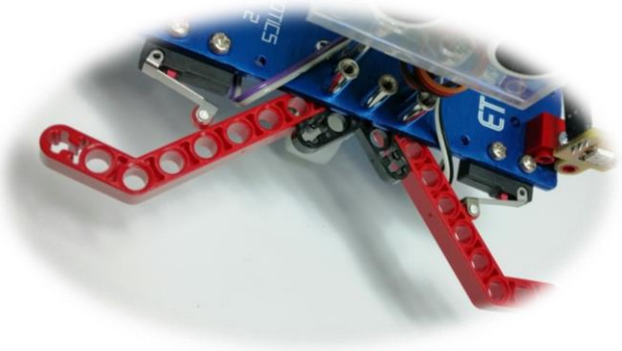
4) LED-RGB #WS2812B



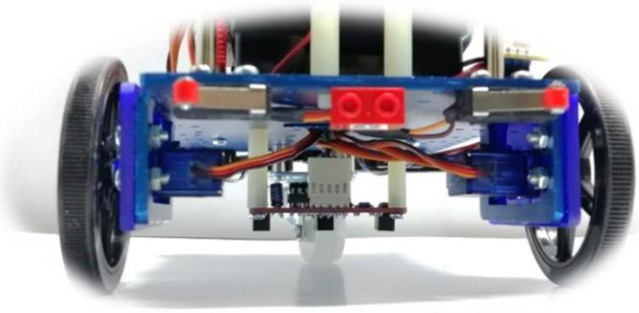
5. Infrared Sensor Tracker (ตรวจจับเส้น ขาว-ดำ)



6) Mini-Speaker



7) Micro-Switch (Switch กันชน)

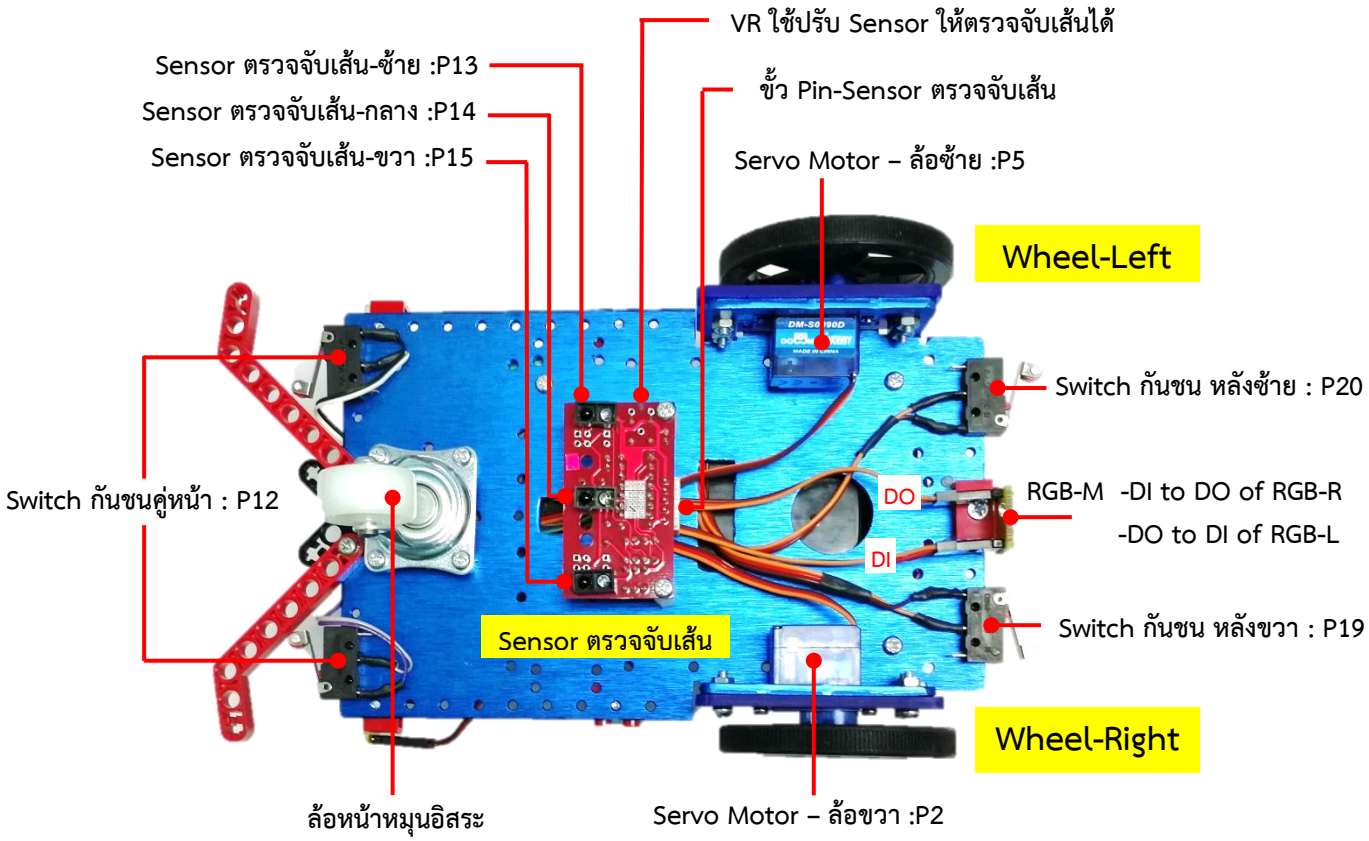


8) Wheel-Micro Servo Motor

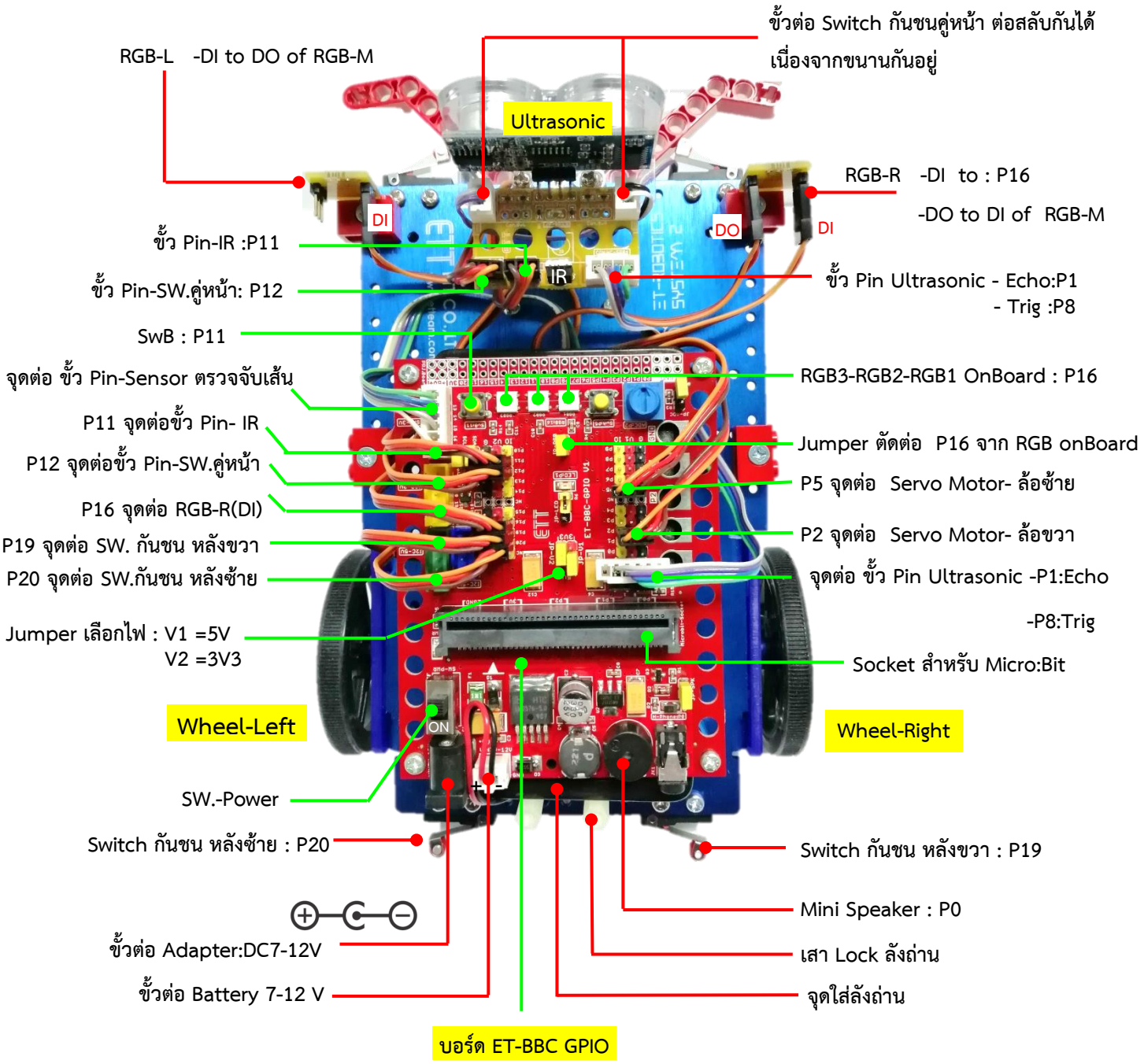
2. คุณสมบัติ ET-BBC ROBOT

- ใช้ไฟเลี้ยง Power DC 7-12V (รองรับขั้วแบบ Type-J สำหรับ Adapter ของ ETT และ Type-B สำหรับลังถ่าน Battery)
-มีลังถ่านเปล่าสำหรับ Battery AA 6 ก้อน =9V (ไม่มี Battery ให้) พร้อมขั้วต่อแบบ Type-B
-มี Switch Power แบบเลื่อน ON/OFF On Board ET-BBC GPIO
- มี Sensor ตรวจวัดระยะทางหรือสิ่งกีดขวาง Ultrasonic HC-SR04
-มี Sensor Infrared (R-Tracker3) ใช้ตรวจจับเส้นด้วยการสะท้อนสัญญาณ (High-Low Level)
-มี Mini Speaker On Board ET- BBC GPIO
-มี ตัวรับสัญญาณ Remote Control (IR-Receiver 4838)
-มีตัวส่งสัญญาณ IR-Remote ET-IR Remote Key
-มี Micro SW. ด้านหน้า 2 ตัว 1 Input และด้านหลัง 2 ตัว 2 Input
-มี LED RGB WS2812 On Board ET-BBC GBIO 3 ตัว และ External Board 3 ตัว (ET-RGB1x1 Kit)
- บอร์ด ET-BBC GPIO สำหรับแปลง Pin IO ของบอร์ดควบคุม Micro:bit และใช้เป็นจุดต่อสายสัญญาณควบคุมอุปกรณ์ IO ต่างๆ
- มีล้อหน้าเคลื่อนที่แบบอิสระ และ ล้อหลัง ขับเคลื่อนด้วย Micro Servo ควบคุมการหมุนด้วย PWM จำนวน 2 ตัว
- สามารถพัฒนาโปรแกรมแบบ Block ด้วย MakeCode Editor แบบ Online หรือ โปรแกรมอื่นๆที่รองรับการใช้งานของบอร์ด micro:bit
--มีตัวอย่างโปรแกรมแยกทดลองอุปกรณ์ IO แต่ละส่วน และตัวอย่างรวม IO พร้อมเล่น แบบ Block ที่พัฒนาบน MakeCode Editor
-ตัวถัง ET-BBC ROBOT ทำด้วย อะลูมิเนียม
- ขนาดของ ET-BBC ROBOT กว้างxยาวxสูง = 130 mm x 225 mm x 112 mm

3. โครงสร้างและ Port ต่อใช้งานของ ET-BBC ROBT



รูป3.1 โครงสร้าง ET-BBC ROBOT ด้านล่าง และ Pin Control



คำเตือน : ห้ามต่อ Adapter และ Battery พร้อมกัน ให้เลือกใช้งานอย่างใดอย่างหนึ่ง

หมายเหตุ : -รายละเอียดการ Control และ Logic การทำงานของอุปกรณ์ IO แต่ละตัวได้อธิบายไว้ในหัวตัวอย่างโปรแกรม

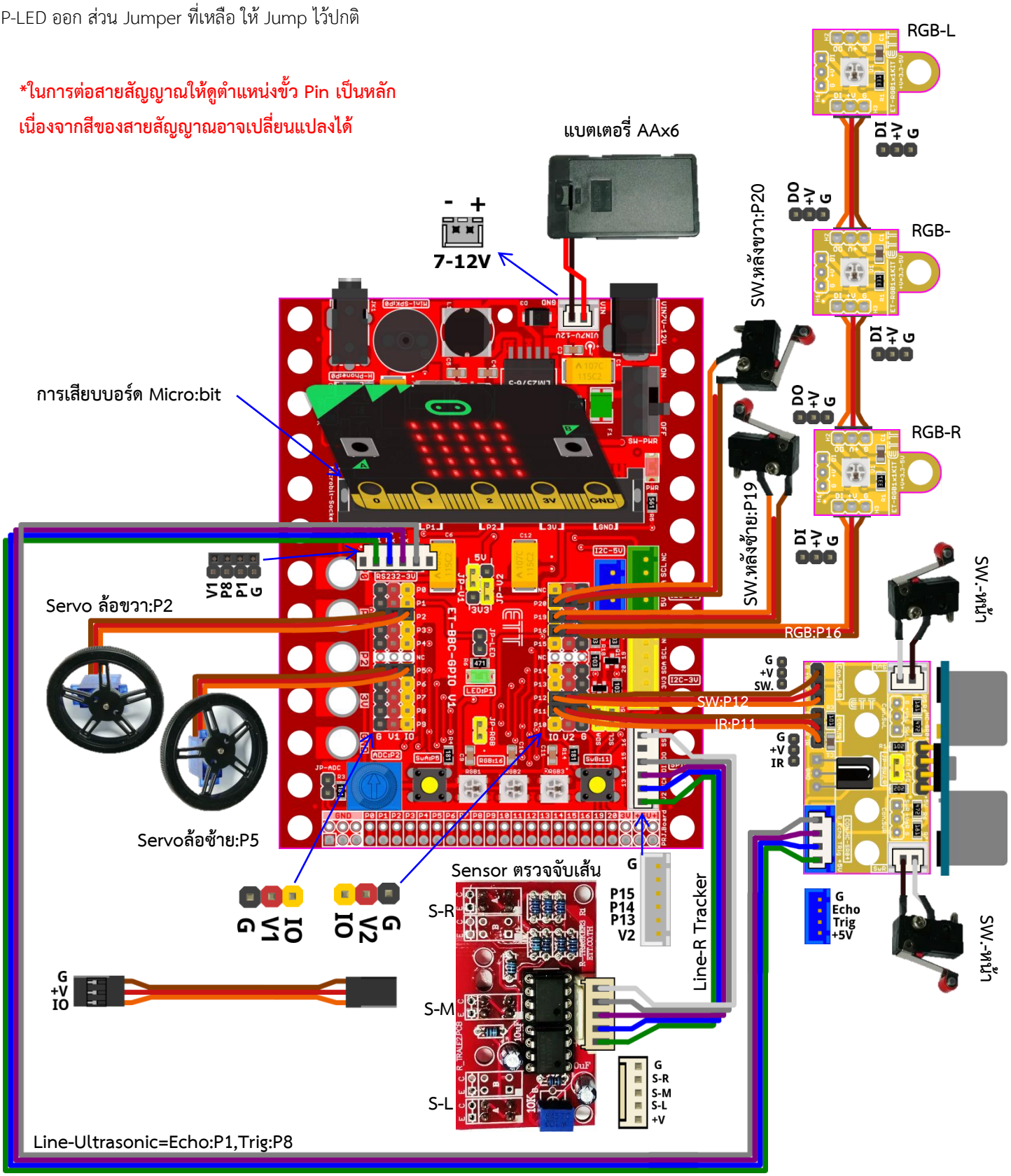
-รายละเอียดและตำแหน่ง IO Pin ของบอร์ด ET-BBC GPIO ดูได้ในท้ายคู่มือ

รูป3.2 โครงสร้าง ET-BBC ROBOT ด้านบน และ Pin Control

4. การต่ออุปกรณ์ IO เข้ากับ Port IO Control บนบอร์ด ET-BBC GPIO

เมื่อใช้งานกับตัวอย่างของ ETT ที่บอร์ด ET-BBC-GPIO ให้ Set Jumper JP-V1 =5V และ JP-V2 = 3V3 , ถอด Jumper JP-ADC และ JP-LED ออก ส่วน Jumper ที่เหลือ ให้ Jump ไว้ปกติ

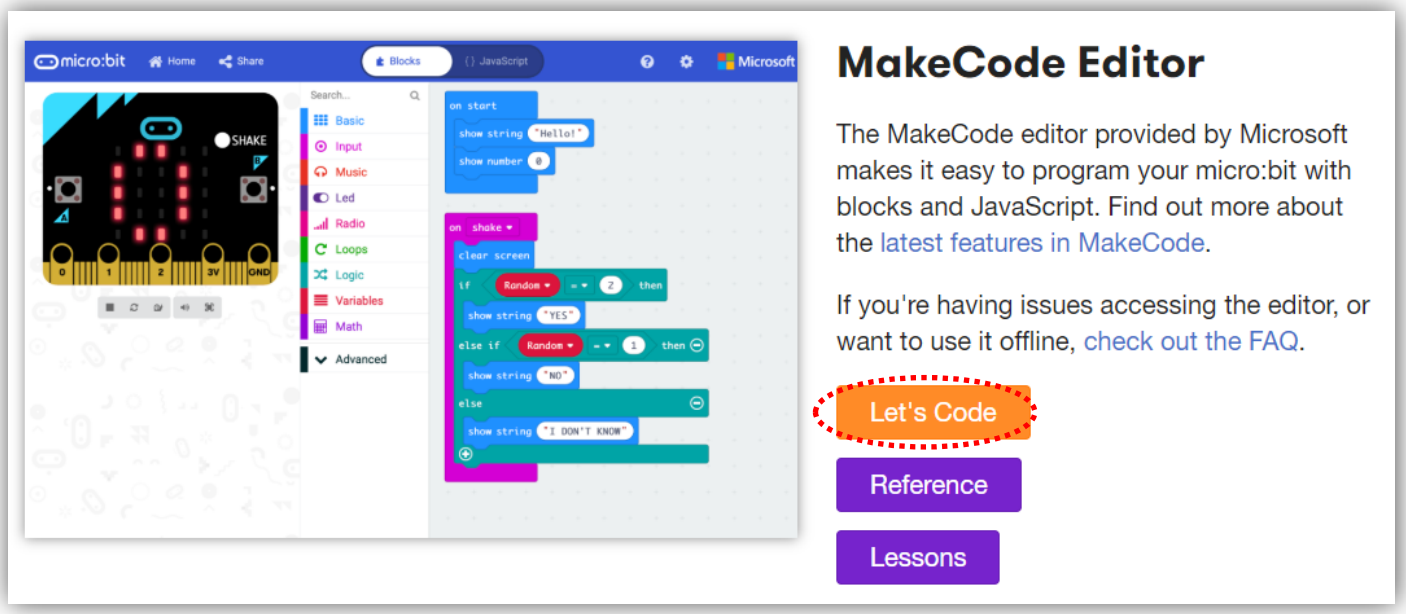
*ในการต่อสายสัญญาณให้ดูตำแหน่งขั้ว Pin เป็นหลัก
เนื่องจากสีของสายสัญญาณอาจเปลี่ยนแปลงได้



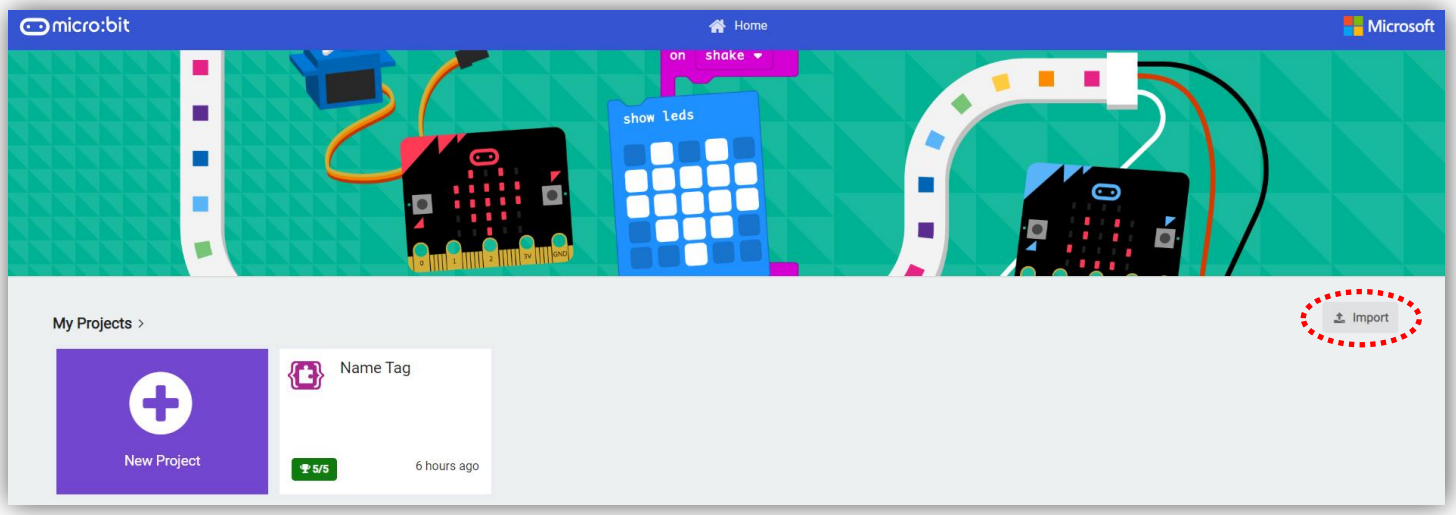
รูป4.1แสดงการเชื่อมต่ออุปกรณ์ IO ของ ET-BBC ROBOT เข้ากับบอร์ด ET-BBC GPIO

5. การใช้งานโปรแกรม MakeCode Editor

ในการเขียนโปรแกรมพัฒนา ET-BBC ROBOT ที่จะกล่าวในคู่มือนี้และตัวอย่างที่ให้ไป จะพัฒนาด้วยโปรแกรม MakeCode Editor ซึ่งจะเป็นการเขียนโปรแกรมแบบใช้ Blockต่อกัน โดยเวลาใช้งานโปรแกรมผู้ใช้จะต้องทำการเชื่อมต่อ Internet ด้วย สามารถเข้าไปใช้งานโปรแกรมได้ที่ Link: https://microbit.org/code/ จะได้หน้าต่างดังรูป5.1 แล้วเลือกที่ปุ่ม Let's Code [Let's Code] เพื่อเข้าสู่หน้าต่างสร้าง Project ดังรูป5.2 ก็เริ่มสร้าง Project สำหรับเขียนโปรแกรมตามขั้นตอนด้านล่างต่อไป



รูป5.1: เมื่อเข้าหน้าเว็บใช้งานโปรแกรม MakeCode Editor



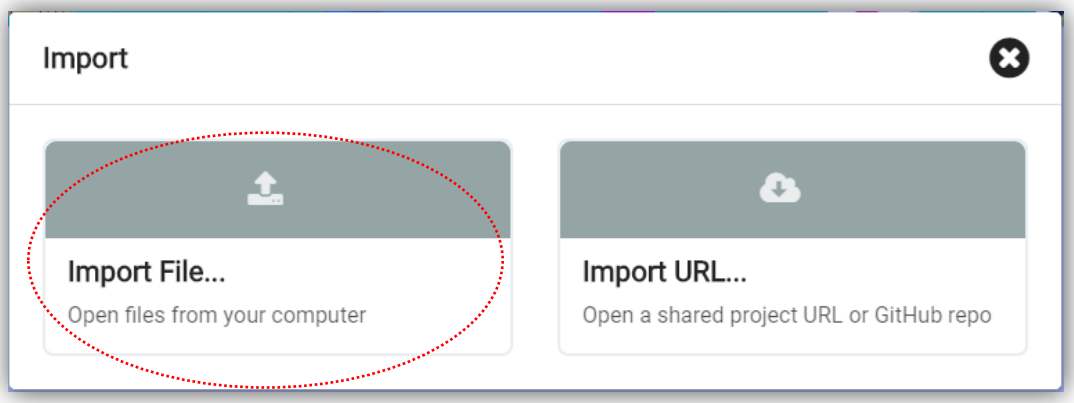
รูป5.2: หลังกดปุ่ม Let's Code เพื่อเข้าสู่หน้าต่างสร้าง Project

5.1 การสร้าง Project File และการเปิด File เก่ามาแก้ไข

ในการเขียนโปรแกรมในที่นี้เราจะไม่แนะนำให้ผู้ใช้สร้าง Project ด้วยปุ่ม New Project เนื่องจากทุกครั้งที่สร้าง Project ด้วยวิธีนี้ library ที่มีการ Add เข้ามาใช้งานจากภายนอกจะถูกลบทิ้งเสมอเหลือเฉพาะ Library มาตรฐานที่โปรแกรม MakeCode มีให้เท่านั้น ซึ่งถ้าผู้ใช้ต้องการใช้ Library จากภายนอกอยู่ก็ต้อง Add Library ใหม่เสมอ ดังนั้นเราจะใช้การเปิด File เก่าขึ้นมาแก้ไข มีขั้นตอนดังนี้

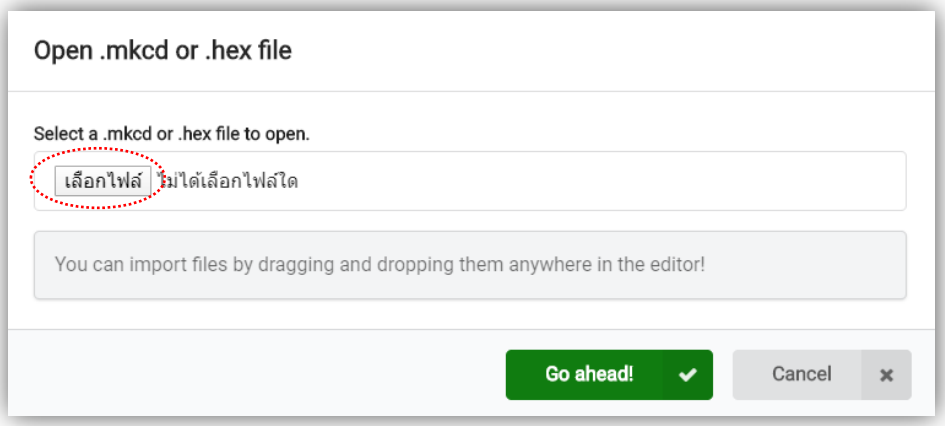
- จากรูป5.2 ให้เลือกที่ปุ่ม Import [Import] เพื่อนำ File เก่ามาแก้ไข

-เมื่อแสดงหน้าต่าง Import ดังรูป 5.3 เลือกที่ปุ่ม Import File...



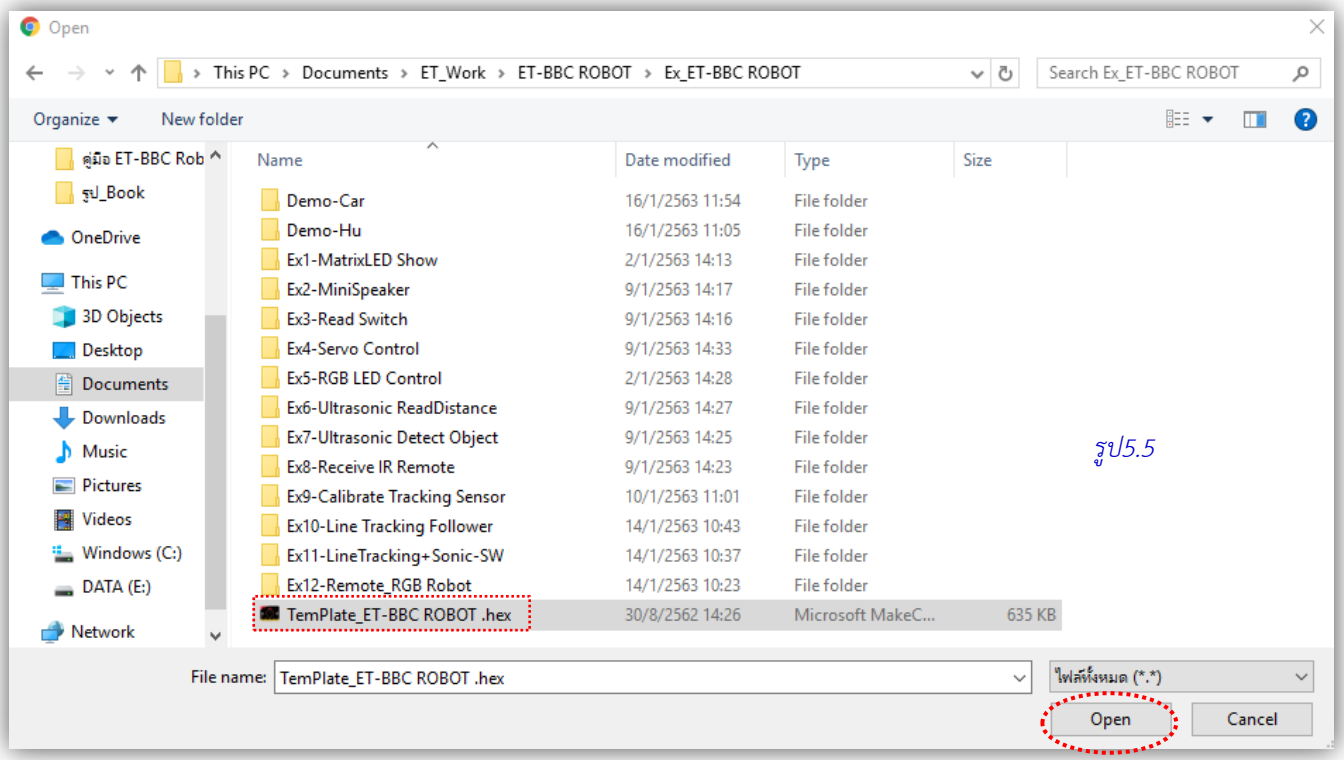
รูป5.3

- จะได้หน้าต่างดังรูป 5.4 ให้คลิกที่ปุ่ม เลือกไฟล์



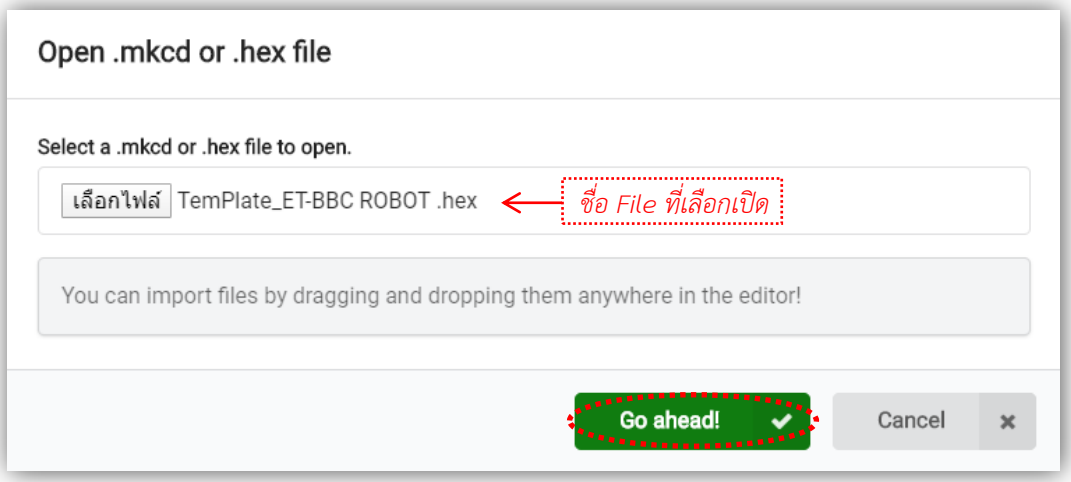
รูป5.4

- จากนั้นเข้าไปยังที่เก็บ File ตัวอย่างของ ETT และเลือกเปิด Filexxx.hex ที่จะนำมาแก้ไข หรือเลือกเปิด File Template_ET-BBC ROBOT.hex ที่ทาง ETT ทำไว้ให้ก็ได้ ซึ่ง File นี้จะเป็น File เปล่าที่ยังไม่มีการเขียนโปรแกรม แต่มีการ Add Library สำหรับใช้งานกับ ET-BBC ROBOT ไว้ให้เรียบร้อยแล้ว ดังรูป 5.5 แล้ว คลิก ปุ่ม Open



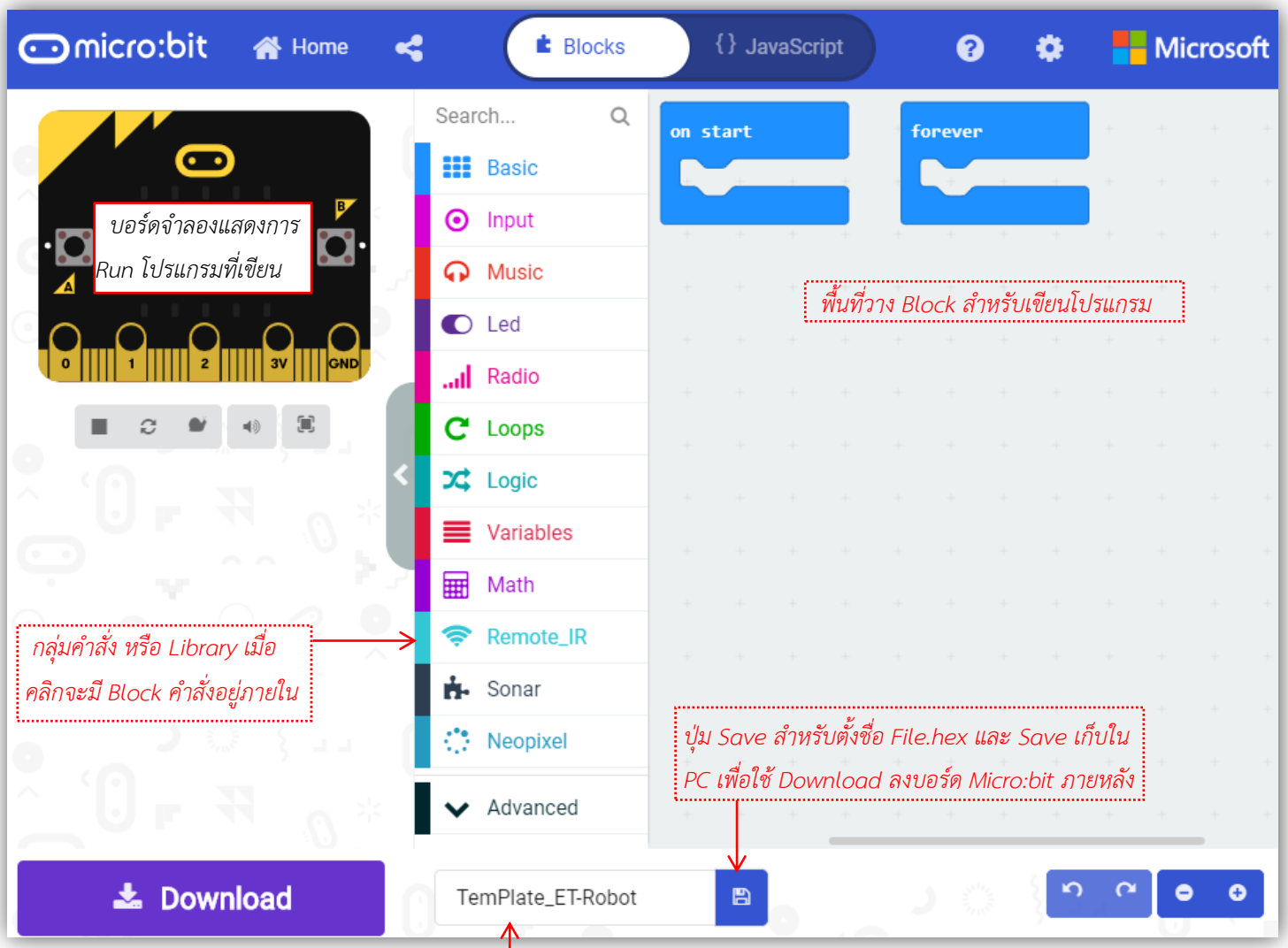
รูป5.5

- จะได้หน้าต่างดังรูป5.6 ชื่อไฟล์ที่ผู้ใช้เลือกเปิดก็จะปรากฏขึ้น แล้วคลิกที่ปุ่ม Go ahead! เพื่อเปิดไฟล์ Project ที่เลือก



รูป5.6

- จะได้หน้าต่างพร้อมเขียนโปรแกรมดังรูป5.7 ผู้ใช้สามารถนำบล็อกที่อยู่ในกลุ่มคำสั่งต่างๆมาใช้เขียนโปรแกรมได้

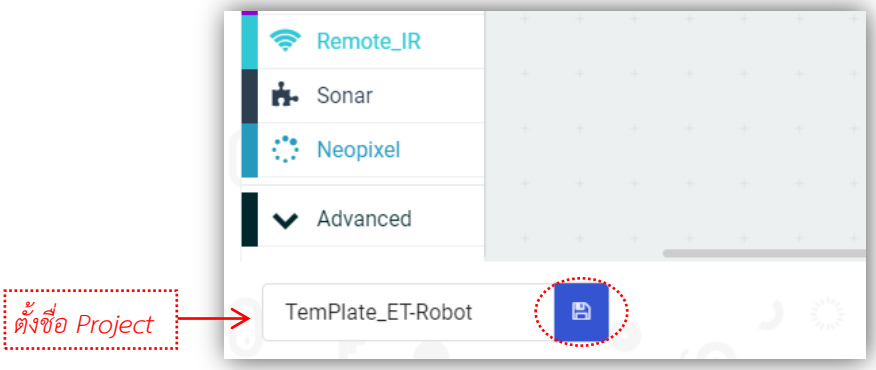


รูป5.7:หน้าต่างพร้อมใช้เขียนโปรแกรม

5.2 การ Save File Program เก็บใน PC

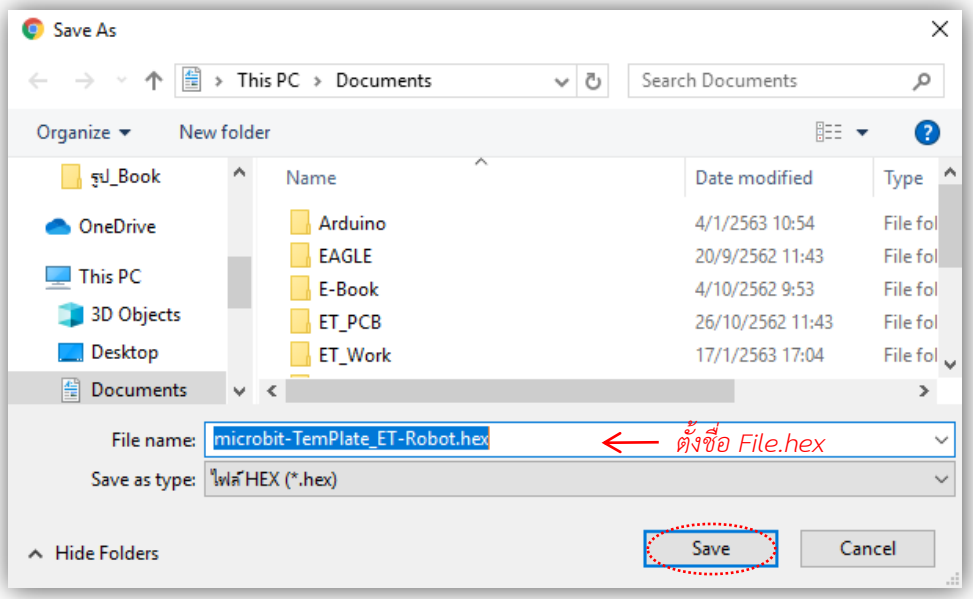
Program ที่ผู้ใช้เขียนขึ้นมานั้น ถ้าไม่มีการ Save เก็บไว้ในเครื่อง PC จะมีโอกาสสูญหายได้ถ้า Browser ที่ใช้เปิดโปรแกรมมีปัญหาขึ้น หรือ มีการลบ Project File บนหน้าเว็บทิ้งก่อนการ Save ซึ่งขั้นตอนการ Save มีดังนี้

- หลังจากเขียนโปรแกรมเสร็จหรือก่อนเขียนโปรแกรม จากรูป 5.8 ในช่องตั้งชื่อ Project ให้ผู้ใช้ใส่ชื่อ Project ที่ต้องการลงไป จากนั้นคลิกที่ปุ่ม save [ ]



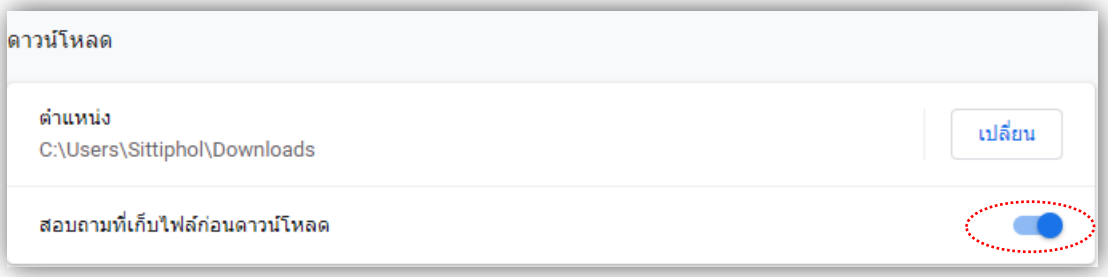
รูป5.8

- จากนั้นจะได้หน้าต่างดัง รูป5.9 เพื่อเลือกที่ตั้งสำหรับ Save File ผู้ใช้สามารถตั้งชื่อ File ใหม่ได้โดยมีนามสกุล xx.hex แล้วกด Save ซึ่ง File ที่ save เก็บไว้นี้จะนำไปใช้สำหรับ Download ลงบอร์ด Micro:bit และสามารถใช้โปรแกรม MakeCode Editor เปิดขึ้นมาทำการแก้ไขโปรแกรมได้ด้วย



รูป5.9

หมายเหตุ *ถ้ากดปุ่ม Save แล้วไม่มีหน้าต่างดังรูป5.9 ขึ้นมา จะเกิดจากการตั้งค่า Browser ที่ใช้งาน โดยถ้าเป็น Browser google chrome ตัว File ที่ Save จะถูกตั้งชื่อให้อัตโนมัติและ Save เก็บไว้ใน Folder Download สามารถแก้ให้ ปรากฏหน้าต่างดังรูป5.9 ได้โดย เปิด google chrome ขึ้นมา ที่มุมบนขวาเลือกปุ่ม ⋮ แล้วเลือก 'การตั้งค่า' เลือก 'ขั้นสูง' ไปที่ หัวข้อ 'ดาวน์โหลด' แล้วเลื่อนปุ่มตามรูป5.10 ไปทางขวา เสร็จสิ้นการตั้งค่า สำหรับ Browser อื่นผู้ใช้ต้องหาวิธีตั้งค่าเอง*

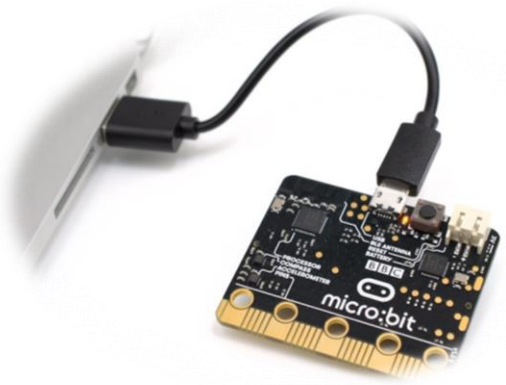


รูป5.10

5.3 การ Download Program File.hex ลงบอร์ด Micro:bit

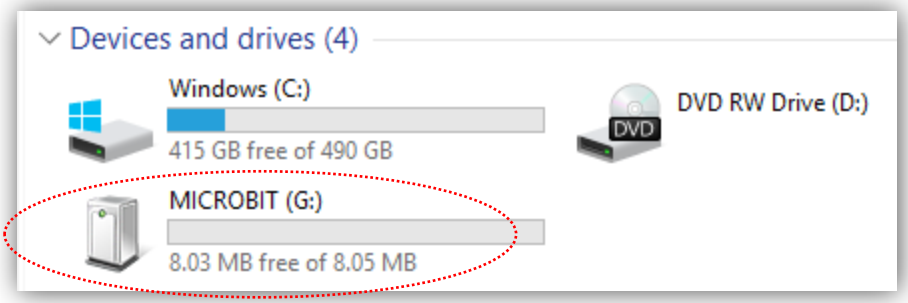
หลังจากทำการ Save File เรียบร้อย ผู้ใช้สามารถนำ file.hex ที่ Save ไว้มาทำการ Download ลงบอร์ด Micro:bit ได้ดังนี้

-เสียบสาย Micro USB เข้ากับบอร์ด Micro:bit และสายอีกด้านเสียบเข้าช่อง USB ของ PC ดังรูป



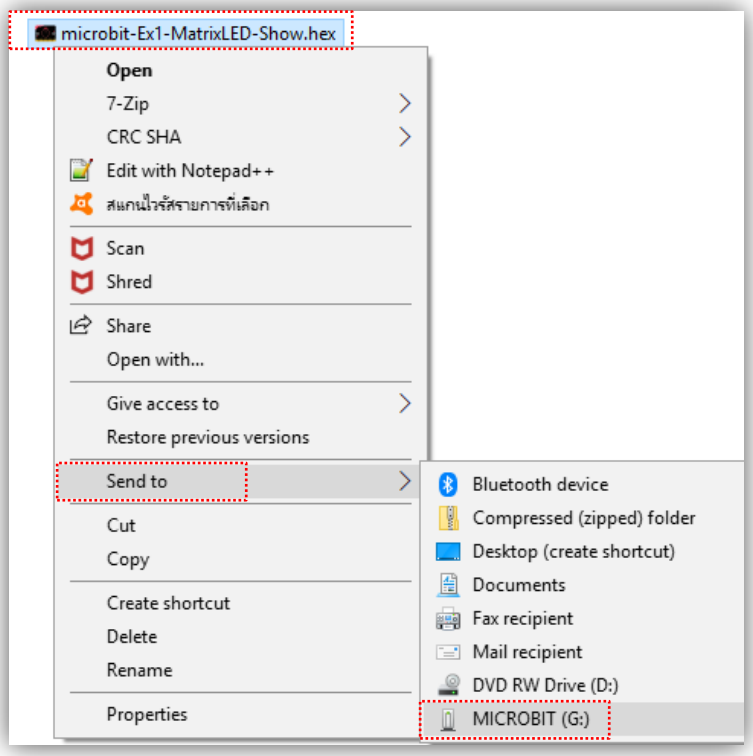
รูป5.11

-จากนั้นคลิกเข้าไปที่ This PC หรือ My Computer ก็จะปรากฏ Drive ขึ้นมา เหมือนเราเสียบ USB Flash Drive ดังรูป5.12 ซึ่งปกติจะมีหน้าต่างเด้งขึ้นมาให้อัตโนมัติ แสดงว่าติดต่อกับบอร์ด Micro:bit ได้แล้ว



รูป5.12

- เมื่อพบ Drive Microbit แล้ว ให้ไปที่ File.hex ที่จะใช้ทำการ Download ให้คลิกขวาที่ File แล้วเลือก Send to เลือก Drive MICOBIT ไฟล์ก็จะถูก Download ไปยังบอร์ด Micro:bit ดังรูป5.13

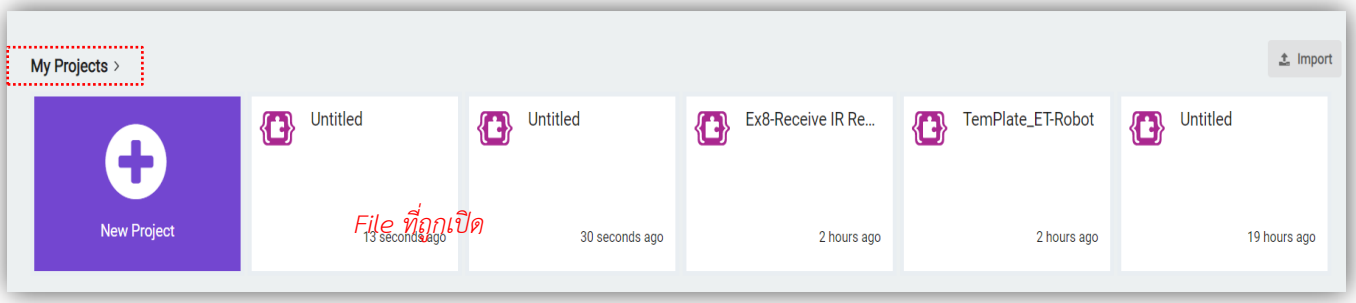


รูป5.13

5.4 การ ลบ File Project ที่สร้างขึ้นบนหน้า Web ของโปรแกรม MakeCode Editor

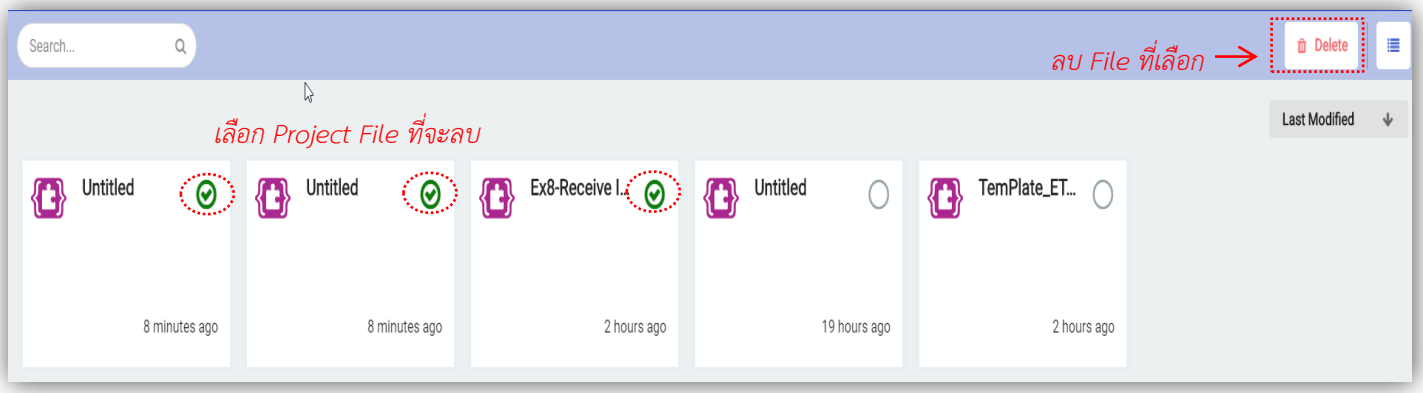
การลบ File Project บนหน้า Web จากโปรแกรม MakeCode Editor ที่เกิดจากการสร้างขึ้นของผู้ใช้ หรือ เปิดขึ้นมาจาก File ที่เก็บอยู่ในเครื่อง PC จะไม่ส่งผลต่อ File ที่ Save เก็บไว้ในเครื่องแต่อย่างใด เพราะเวลาเราเปิด File ขึ้นมาเยอะๆอาจทำให้ผู้ใช้สับสนว่า File ตัวใดเป็น File ล่าสุด ที่เปิดมาแก้ไขได้ ดังนั้นขอแนะนำวิธีลบ Project File ที่เปิดขึ้นมาดังนี้

- ไปที่หน้าสร้าง New Project ดังรูป5.14 จะเห็นว่ามี File ที่เปิดขึ้นมาหลาย File ค้างอยู่ เมื่อต้องการจะลบให้คลิกที่ My Projects>



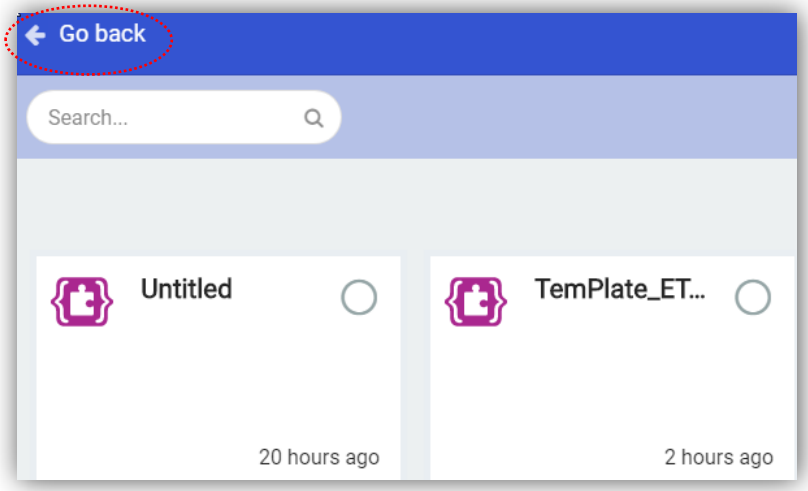
รูป5.14

- ทำเครื่องหมายถูกในช่องวงกลมสำหรับ File ที่ต้องการจะลบ จากนั้นคลิกที่คำสั่ง Delete เพื่อลบ File ดังรูป 5.15



รูป5.15

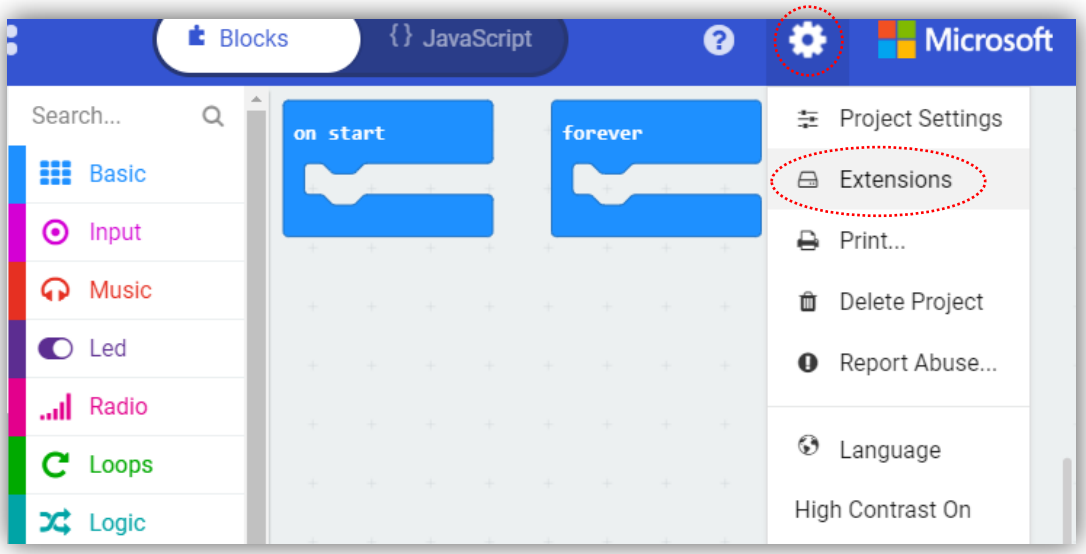
- คลิก Go Back เพื่อกลับหน้า New Project ดังรูป5.16



รูป5.16

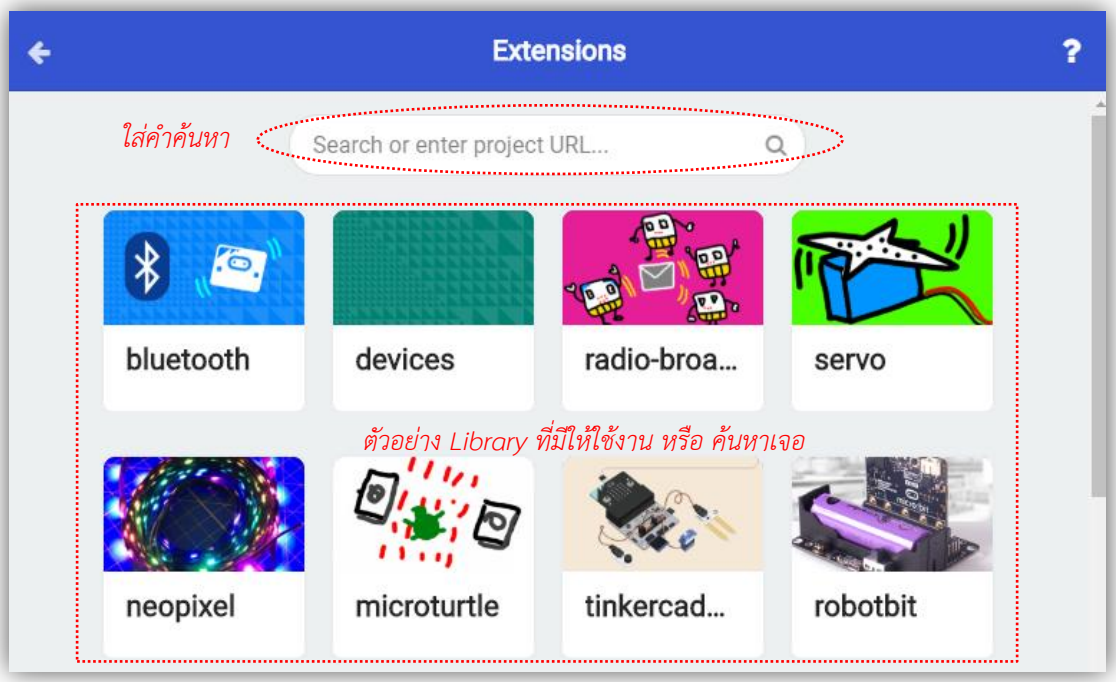
5.5 การเพิ่มกลุ่มคำสั่งหรือ Add Library จากภายนอกเข้ามาใช้ใน Project

- ไปที่หน้าสำหรับเขียนโปรแกรมดังรูป 5.17 แล้วคลิกที่รูปฟันเฟือง และเลือก Extensions

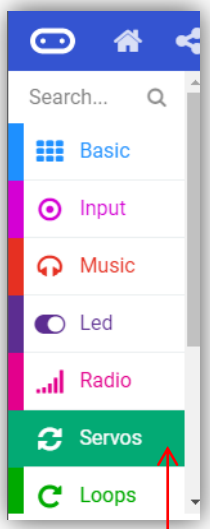


รูป5.17

- จะได้หน้าต่าง Extensions ขึ้นมาดังรูป5.18 ในช่องค้นหา ให้ใส่คำค้นหา หรือ link ของ Library ที่ต้องการเพิ่ม ถ้าค้นพบก็จะปรากฏ Library ใต้ช่องค้นหา ซึ่งปกติจะมี Library ส่วนหนึ่งไว้ให้เลือกใช้งานอยู่แล้ว ซึ่งสามารถ Add เข้าไปใน Project ได้ด้วยการ ดับเบิลคลิก ที่รูป Library นั้น Library ก็จะถูกเพิ่มเข้ามาใน Project File ให้อัตโนมัติ ดังรูป5.19



รูป5.18

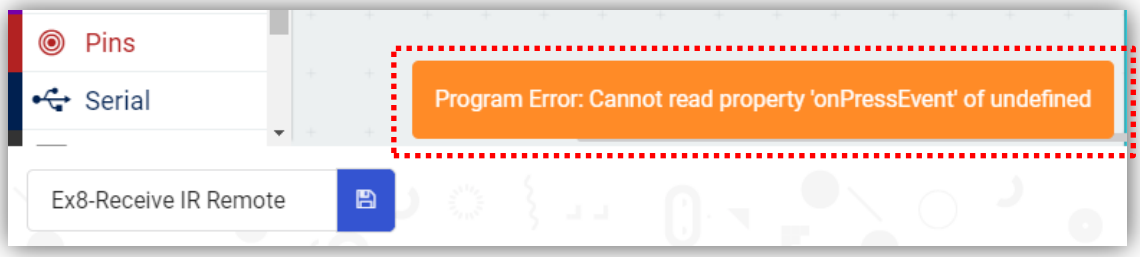


รูป5.19 ตัวอย่าง Library Servo ถูกเพิ่มเข้ามา

6. ตัวอย่างโปรแกรม ET-BBC ROBOT โดยใช้ MakeCode Editor (Folder:Ex_Standard)

สำหรับตัวอย่างที่จะกล่าวถึงจะอ้างอิงการเขียนโปรแกรมแบบ Block โดยใช้โปรแกรม MakCode Editor ซึ่งในตัวอย่าง จะมีรายละเอียดพื้นฐานในการ Control อุปกรณ์ IO ต่างๆ ที่ใช้กับ ET-BBC ROBOT และ อ้างอิงการต่ออุปกรณ์ IO ต่างๆ เข้ากับ Port IO ของ Micro:bit ตาม ET-BBC ROBOT ผ่านบอร์ด ET-BBC GPIO

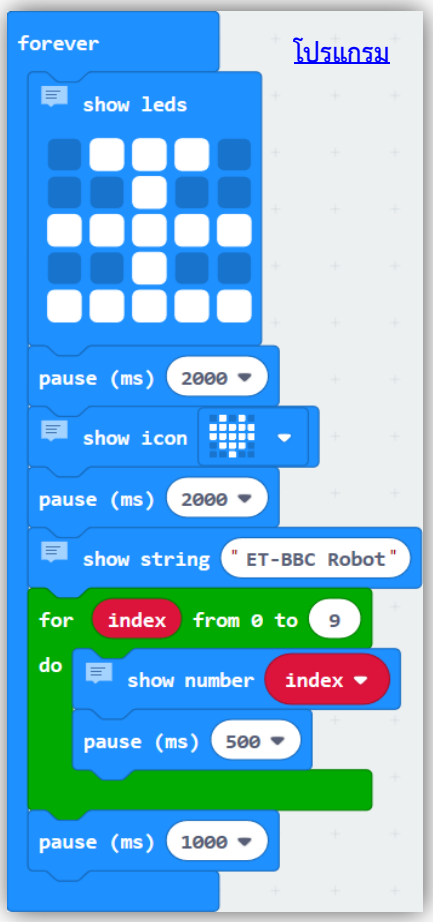
หมายเหตุ *เมื่อเปิด File ตัวอย่างหรือ Template ที่ ETT ทำไว้ขึ้นมาแก้ไข ถ้าตัวอย่างใด มีการเรียกใช้งาน Block ของกลุ่มคำสั่ง REMOTE_IR [Remote_IR] ระหว่างที่เขียนโปรแกรมจะปรากฏข้อความ Error ขึ้นมาเป็นระยะดังรูปที่ 6.1 ผู้ใช้ไม่ต้องสนใจให้ปล่อยผ่านไปจะไม่มีผลต่อโปรแกรมที่เขียนและการใช้งานของกลุ่มคำสั่ง REMOTE_IR แต่อย่างใด*

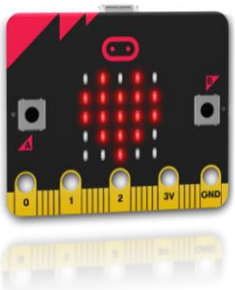


รูป6.1

Ex1-MatrixLED Show

ตัวอย่างนี้จะเป็นการศึกษาเพื่อทดสอบการเขียนโปรแกรมควบคุม Matrix-LED ที่อยู่บนบอร์ด Micro:bit โดยใช้ Block ในกลุ่มคำสั่ง Basic ซึ่งจะทำให้ผู้ใช้เห็นความแตกต่างกันในการใช้งาน Block แสดงผลในแต่ละแบบ และเลือกใช้งานได้เหมาะสม

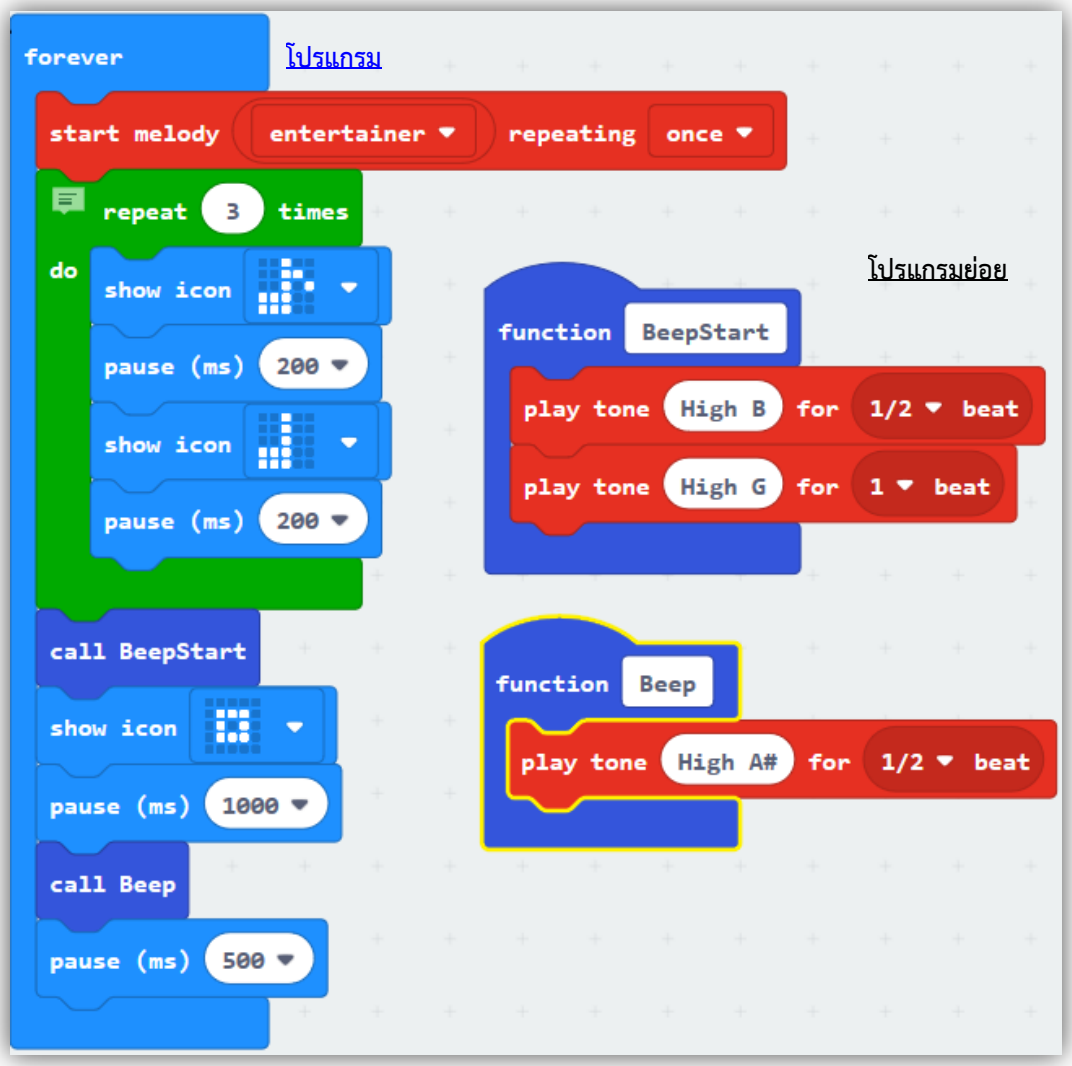




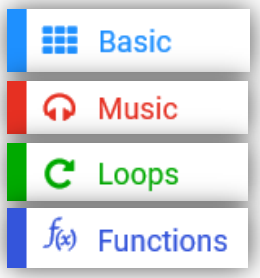
การทำงานของโปรแกรม: เริ่มต้นที่ Matrix-LED จะแสดงรูปคน 2 วินาที ตามด้วยรูปหัวใจ 2 วินาที และข้อความ ‘ET-BBC ROBOT’ จากนั้นก็จะแสดงการนับตัวเลข 0-9 และวนกลับมาเริ่มใหม่

Ex2-MiniSpeaker

ตัวอย่างนี้จะเป็นการศึกษาเพื่อทดสอบการเขียนโปรแกรมควบคุม Mini-Speaker ที่อยู่บนบอร์ด ET-BBC GPIO โดยใช้ Block ในกลุ่มคำสั่งของ Music และใช้ Pin-P0 ในการ Control ซึ่งในการกำเนิดเสียงของลำโพง จะใช้การสร้างพัลส์เป็นความถี่ป้อนให้กับลำโพง ในการทดลองให้ Short Jumper JP-SPK และเสียบบอร์ด Micro:bit เข้ากับบอร์ด ET-BBC GPIO



กลุ่มคำสั่งใช้งาน

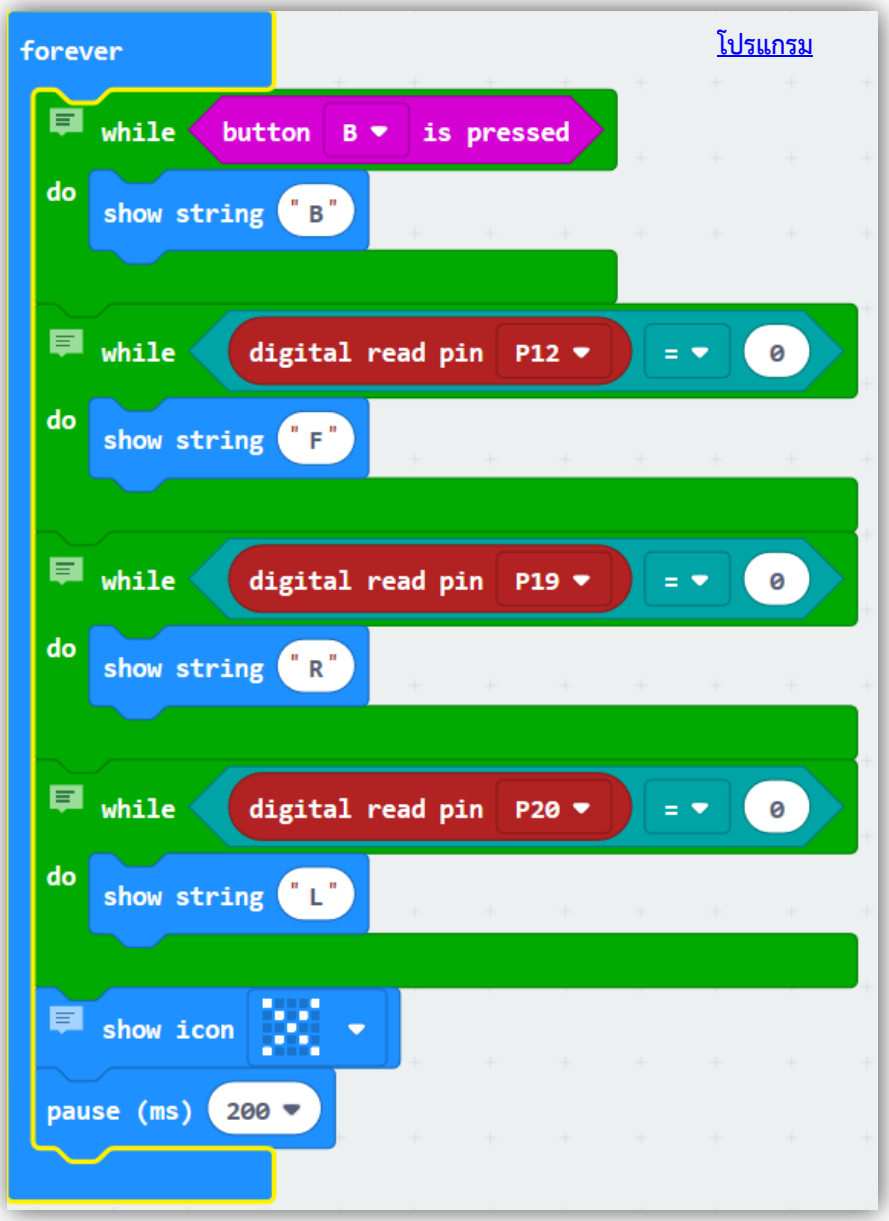


การทำงานของโปรแกรม : เริ่มต้นจะให้เล่นเสียงดนตรีหนึ่งครั้ง และ ที่ Matrix-LED แสดงตัวโน๊ต โดยให้มีเขบ็ตตัวโน๊ตกระพริบ 3 ครั้ง จากนั้นก็เล่นเสียงติ้งต่อง และ ที่ Matrix-LED แสดงรูปสี่เหลี่ยมเล็ก แล้วตามด้วยเสียง Beep แล้วจบโปรแกรม กลับไปเริ่มต้นเล่นสียงดนตรี ใหม่

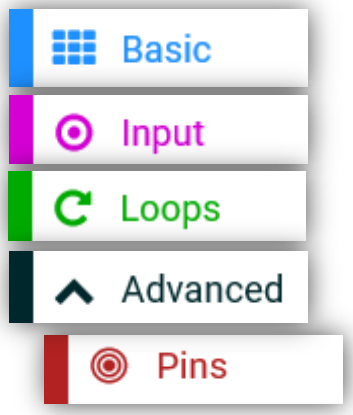
Ex3-Read Switch

ตัวอย่างนี้จะเป็นการศึกษาเพื่อทดสอบการเขียนโปรแกรมอ่านค่าสถานะของ Switch โดยใช้ Block ในกลุ่มคำสั่งของ Input และ Pins ซึ่งจะอ่านค่า SW.B จาก Pin-P11 , Micro SW. คู่หน้าที่ต่อขนานกันอยู่ จาก Pin-P12 , Micro SW. หลังขวา จาก Pin-P19 และ Micro SW.หลังซ้าย จาก Pin-P20 ในการทดลองให้ Set Jumper JP-V2 ไปทางด้าน 3V3 , การทำงานของ Switch แต่ละตัวในสภาวะปกติ(ปล่อย Switch) จะอ่านสถานะได้ Logic 1 แต่เมื่อมีการกด Switch จะอ่านสถานะได้ Logic 0 จนกว่าจะมีการปล่อย Switch ในการอ่านสถานะ Switch ควรจะมีการหน่วงเวลาหลังการอ่านเพื่อป้องกันการกด Switch ซ้ำโดยไม่ตั้งใจ

โปรแกรม



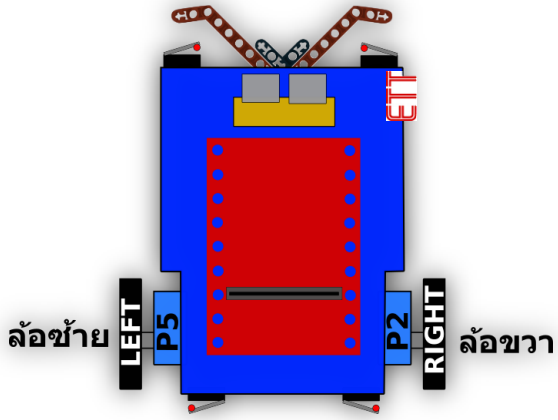
กลุ่มคำสั่งใช้งาน

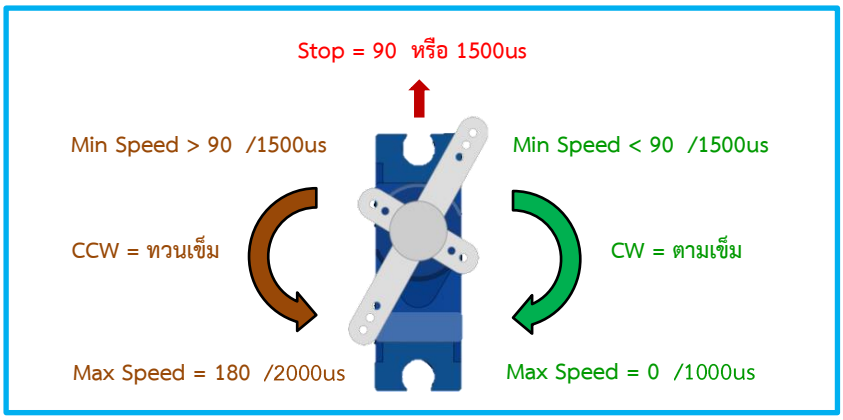


การทำงานของโปรแกรม : เริ่มต้นโปรแกรมจะวนรออ่านสถานะ Switch จาก 4-Pin ตามที่กล่าวข้างต้น ขณะยังไม่มีการกด Switch ใดๆ Matrix-LED จะแสดงเครื่องหมาย X เมื่อมีการกด Switch.B ที่ Matrix-LED จะแสดงตัว B จนกว่าจะมีการปล่อย Switch เป็นต้น สรุปก็คือถ้ามีการกด Switch ตัวใด ที่ Matrix-LED ก็จะแสดงตัวอักษรตามโปรแกรมที่เขียนไว้

Ex4-Servo Control

ตัวอย่างนี้จะเป็นการศึกษาเพื่อทดสอบการเขียนโปรแกรมควบคุม Servo Motor พื้นฐานให้ หยุด , หมุนทวนเข็มนาฬิกา , หมุนตามเข็มนาฬิกา เพื่อนำไปใช้ควบคุม Robot ให้ เดินหน้า , ถอยหลัง และ Stop รวมทั้งปรับความเร็ว Servo Motor ได้ โดยใช้ Block ในกลุ่มคำสั่งของ Pins และใช้ Pin P2 ควบคุม Servo Motor ล้อขวา และ Pin P5 ควบคุม Servo Motor ล้อซ้าย อ้างอิงตำแหน่งล้อ ซ้าย-ขวา และทิศทางการหมุน ตามรูปด้านล่าง ในการทดลองให้ Set Jumper JP-V1=5V และ JP-ADC= ไม่ jump





สำหรับ Servo Motor รุ่นที่ใช้เมื่ออ้างอิงการเขียนโปรแกรมด้วย MakeCode Editor โดยใช้ชุดคำสั่งที่มากับโปรแกรมผู้ใช้สามารถเลือกใช้ชุดคำสั่งควบคุมการหมุนได้ 2 แบบ คือ

- ป้อนค่าเป็นคาบเวลา PWM โดยช่วงที่ Servo ทำงาน หรือค่าที่ป้อน คือ 0 –2000 us ใช้ชุดคำสั่ง --- >

-ป้อนค่าเป็นมุม องศา โดยช่วงที่ Servo ทำงานหรือค่าที่ป้อน คือ 0-180 องศา ใช้ชุดคำสั่ง --- >

สำหรับค่า องศา และค่า PWM ที่ใช้ควบคุมการหมุนของ Servo Motor กับรุ่นที่ใช้งานสรุปได้ดังนี้

180 หรือ 2000 us = Servo หมุน Max-Speed ทวนเข็ม(CCW) ช่วงความเร็ว 91-180 องศา หรือ 1501-2000 us

0 หรือ 1000 us = Servo หมุน Max-Speed ตามเข็ม(CW) ช่วงความเร็ว 89-0 องศา หรือ 1499-1000 us

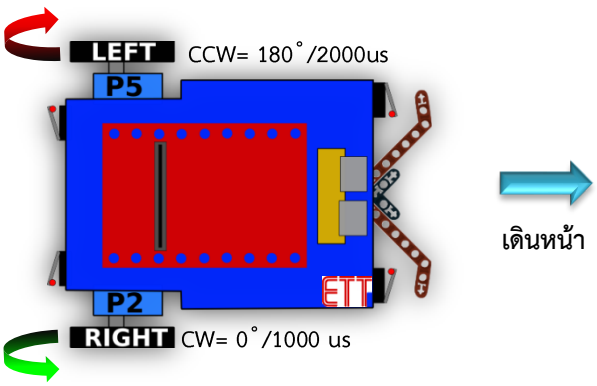
90 หรือ 1500 us = Servo Stop (ค่า Center)

<u>หมายเหตุ</u> ค่าที่กล่าวถึงข้างต้นเป็นค่าในทางทฤษฎีกับ Servo รุ่นที่ใช้ ในทางปฏิบัติอาจมีคลาดเคลื่อนได้ สังเกต ค่า Min-Speed จะเป็นค่าที่เข้าใกล้มุม 90 องศา หรือ 1500 us ตามทิศทางนั้นๆ

<u>การ Control Servo สำหรับ ROBOT</u>

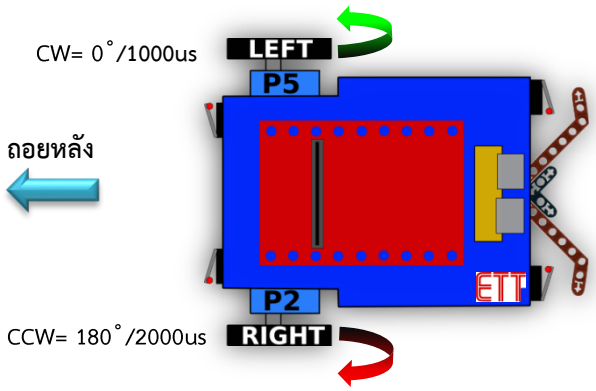
-Robot เดินหน้า (Forward) = Servo ล้อซ้าย (P5) : หมุนทวนเข็ม CCW (180 หรือ 2000us)

Servo ล้อขวา (P2) : หมุนตามเข็ม CW (0 หรือ 1000us)



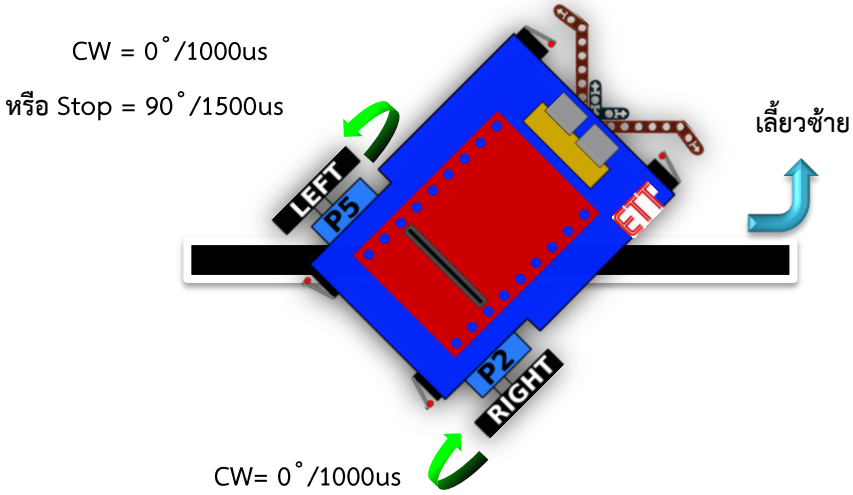
-Robot ถอยหลัง (Backward) = Servo ล้อซ้าย(P5) = หมุนตามเข็ม CW (0˚หรือ 1000us)

Servo ล้อขวา(P2) = หมุนทวนเข็ม CCW (180˚ หรือ 2000us)



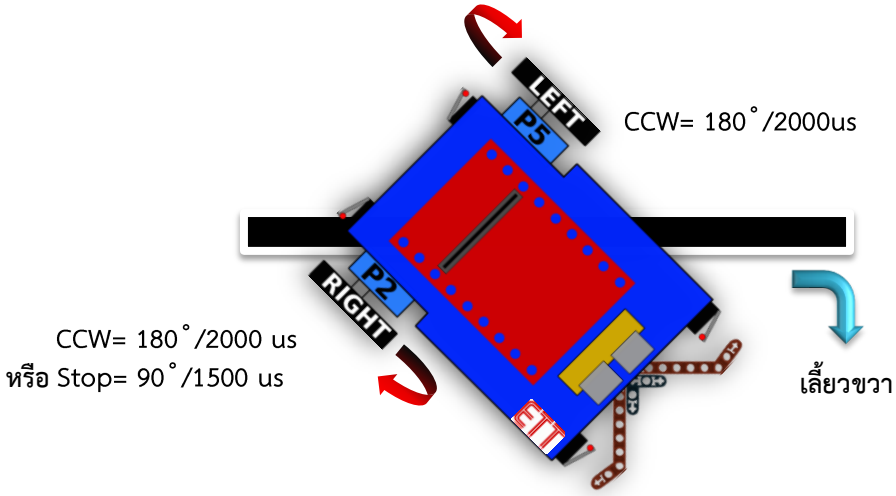
-Robot เลี้ยวซ้าย (Turn-Left) = Servo ล้อซ้าย (P5) = หมุนตามเข็ม CW (0 หรือ 1000 us) หรือ Stop (90 หรือ 1500us)

Servo ล้อขวา (P2) = หมุนตามเข็ม CW (0 หรือ 1000 us)

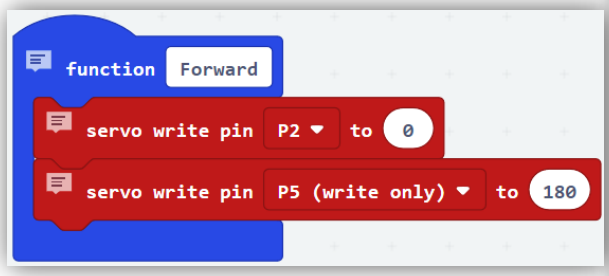


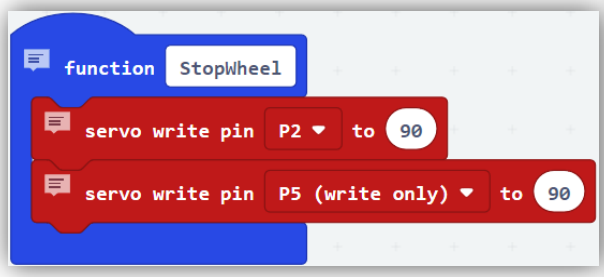
-Robot เลี้ยวขวา (Turn-Right) = Servo ล้อซ้าย (P5) = หมุนทวนเข็ม CCW (180 : 2000us)

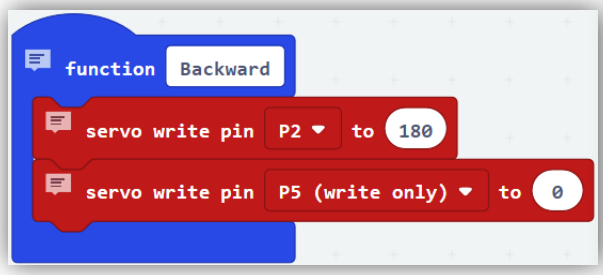
Servo ล้อขวา (P2) = หมุนทวนเข็ม CCW (180 : 2000us) หรือ Stop (90 : 1500us)

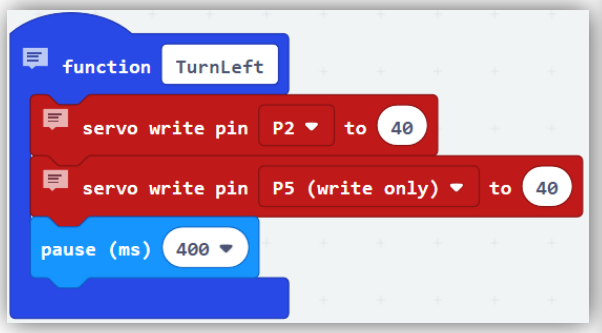


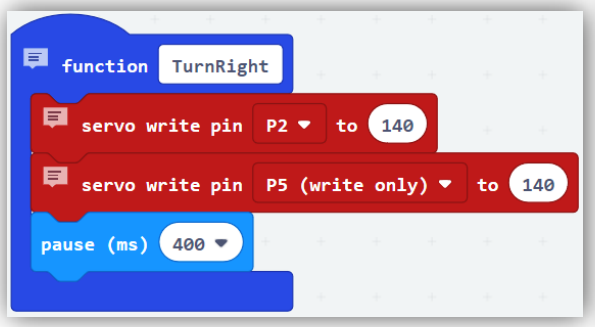
โปรแกรมย่อย-Ctr Wheel

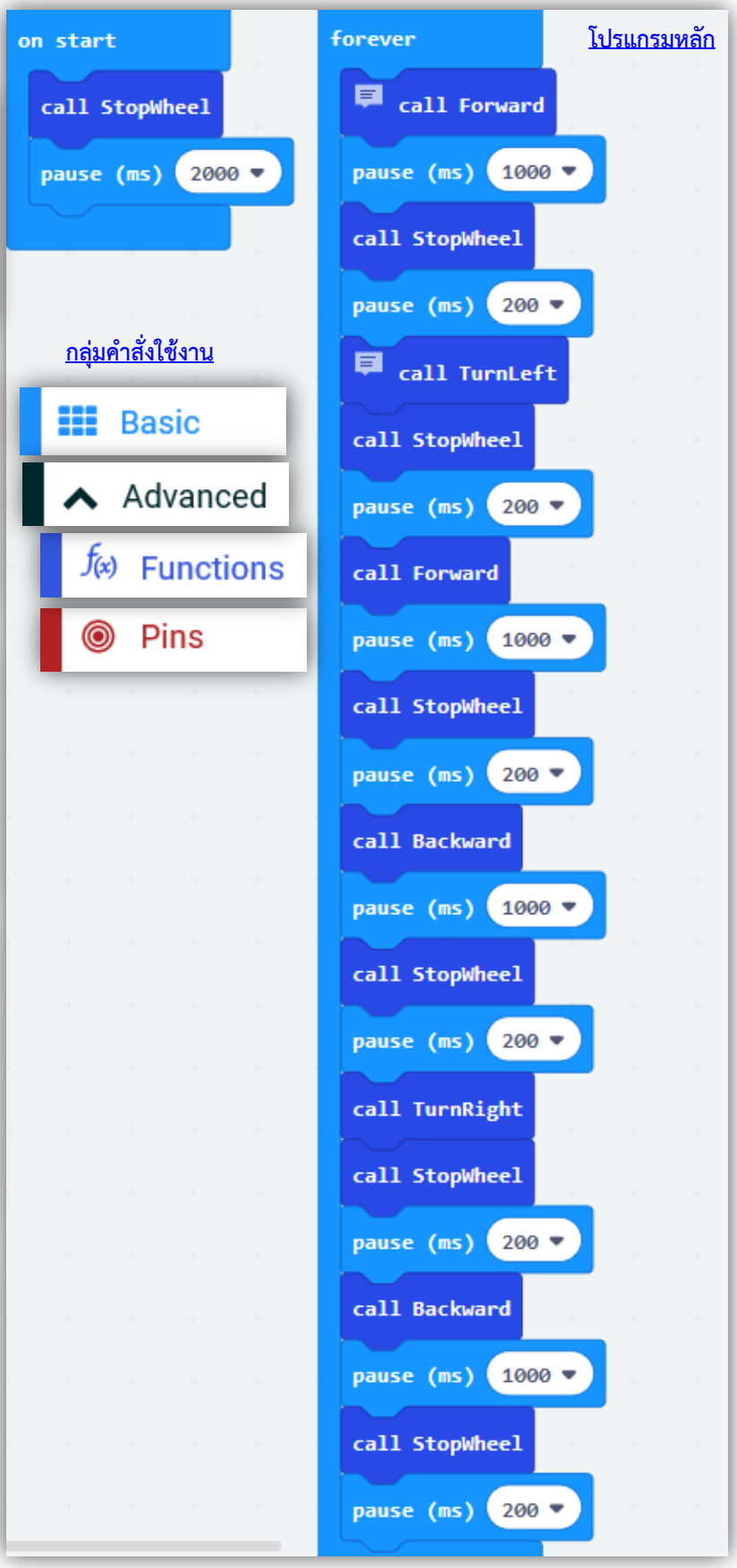










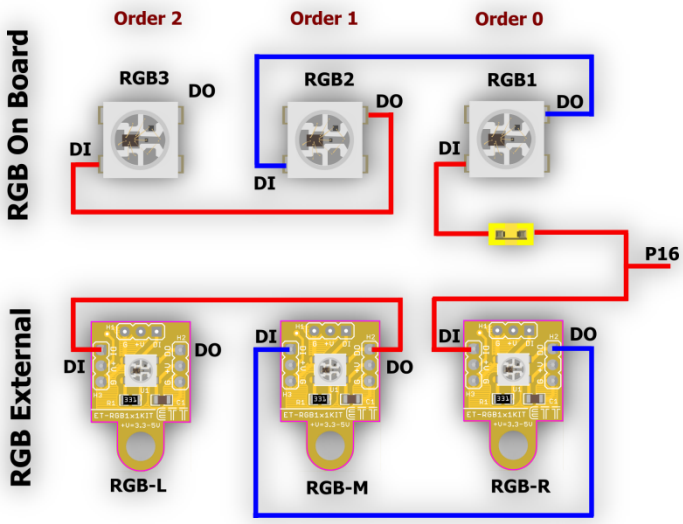


การทำงานของโปรแกรม : เริ่มต้น Robot เดินหน้า 1 วินาที แล้วเลี้ยวซ้าย แล้วเดินหน้า 1 วินาที ,จากนั้น ถอยหลัง 1 วินาที แล้วเลี้ยวขวา แล้วถอยหลังอีก 1 วินาที

Ex5-RGB LED Control

ตัวอย่างนี้จะเป็นการศึกษาเพื่อทดสอบการเขียนโปรแกรมควบคุม LED-RGB WS2812 ให้ติดในสีต่างๆโดยใช้ Block ในกลุ่มคำสั่ง Neopixel เพื่อให้ผู้ใช้เข้าใจการใช้งาน Block ต่างๆและสมารถ Control LED-RGB ให้ติดตามที่ผู้ใช้ต้องการได้

การ Control จะใช้ Pin P16 ในการ Control เพียง Pin เดียวเท่านั้น โดย LED-RGB ของ Robot จะมีด้วยกัน 6 ดวง แยกเป็น LED-RGB On Board 3 ดวง ได้แก่ RGB1,RGB2,RGB3 ซึ่งต่ออนุกรมกัน และ LED-RGB External 3ดวง ได้แก่ RGB-R,RGB-M,RGB-L ซึ่งต่ออนุกรมกัน โดย LED-RGB ทั้ง 2 ชุดจะต่อขนานกันอีกที ดังรูป

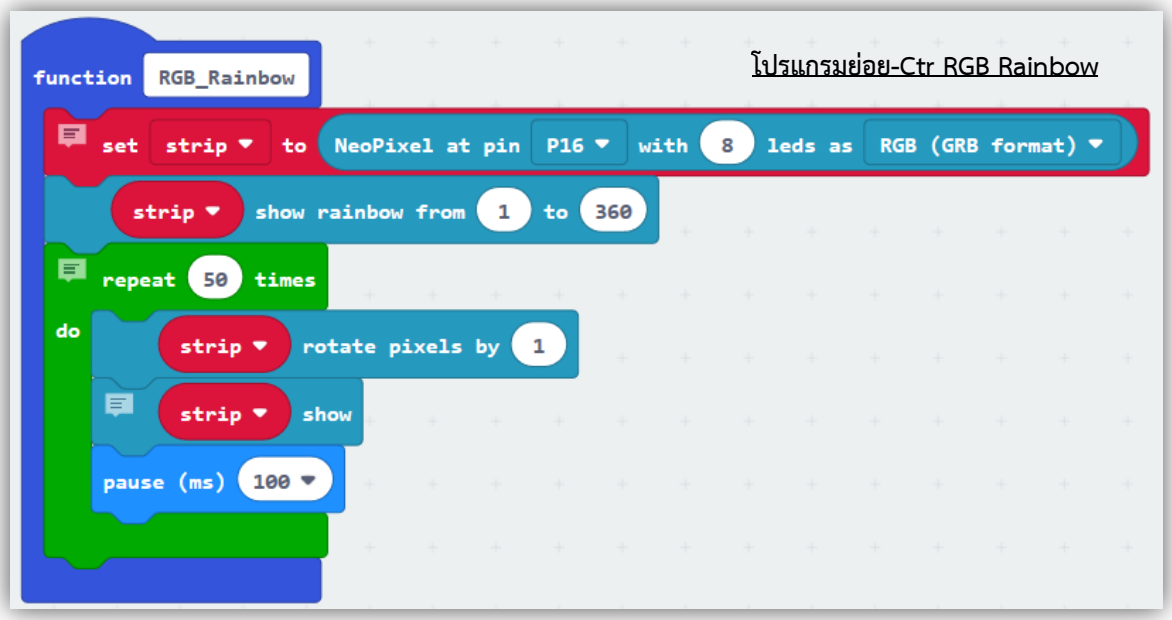


ดังนั้นในการเขียนโปรแกรมก็จะเหมือนกับ Control LED-RGB เพียง 3 ดวง ซึ่งจะทำให้ LED-RGB ทั้ง 2 ชุดแสดงผลเหมือนกันเป็นคู่ๆ ดังนี้

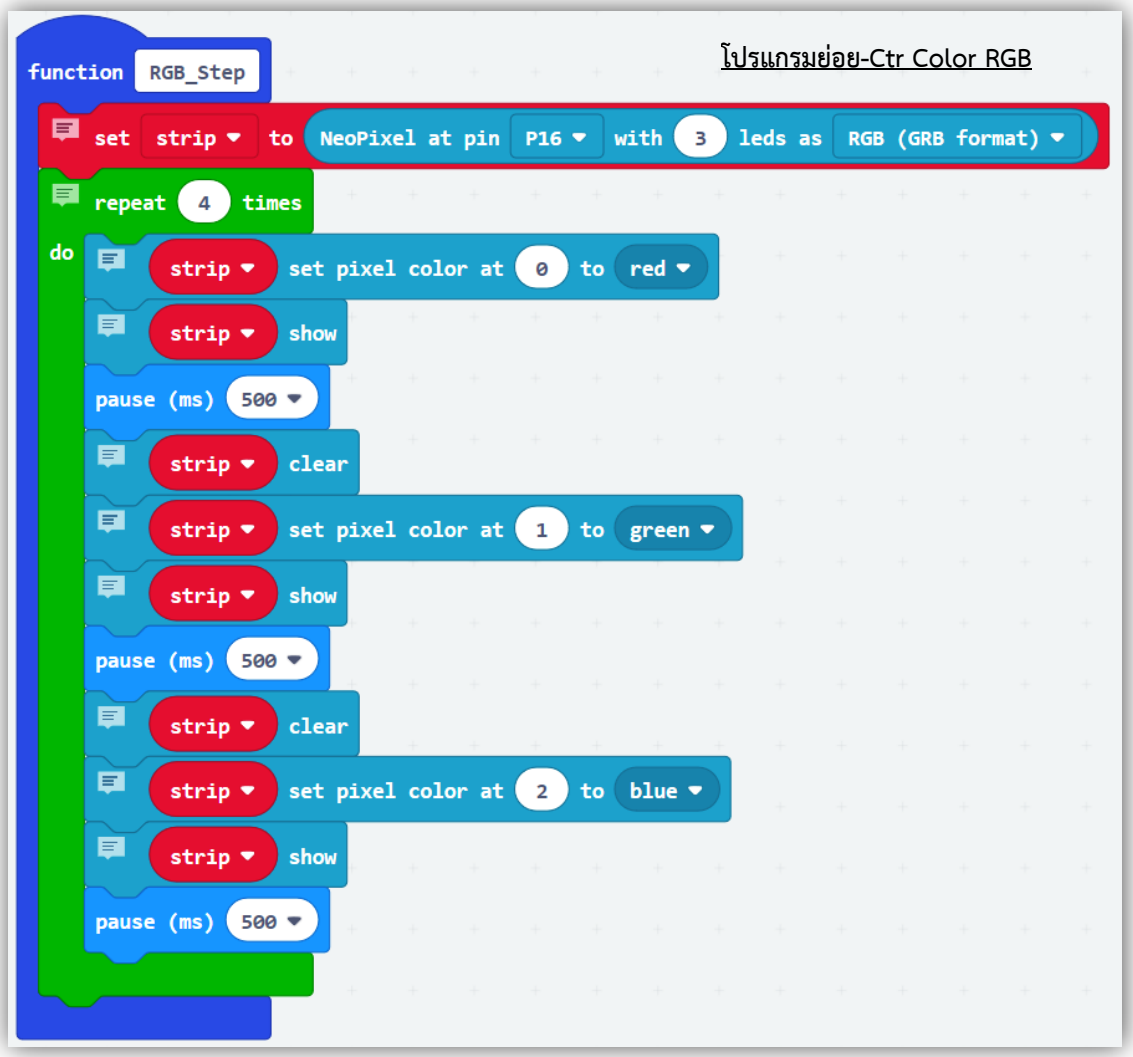
- RGB1=RGB-R ให้อ้างลำดับ LED-RGB ในการเขียนโปรแกรมเป็น ลำดับ 0
- RGB2=RGB-M ให้อ้างลำดับ LED-RGB ในการเขียนโปรแกรมเป็น ลำดับ 1
- RGB3=RGB-L ให้อ้างลำดับ LED-RGB ในการเขียนโปรแกรมเป็น ลำดับ 2

ในการทดลองให้ Set Jumper JP-V2 ไปทางด้าน 3V3 และ Short Jumper JP-RGB ถ้าต้องการให้ LED-RGB On Board ติด แต่ถ้าไม่ต้องการให้ติดก็ไม่ต้อง Short Jumper JP-RGB

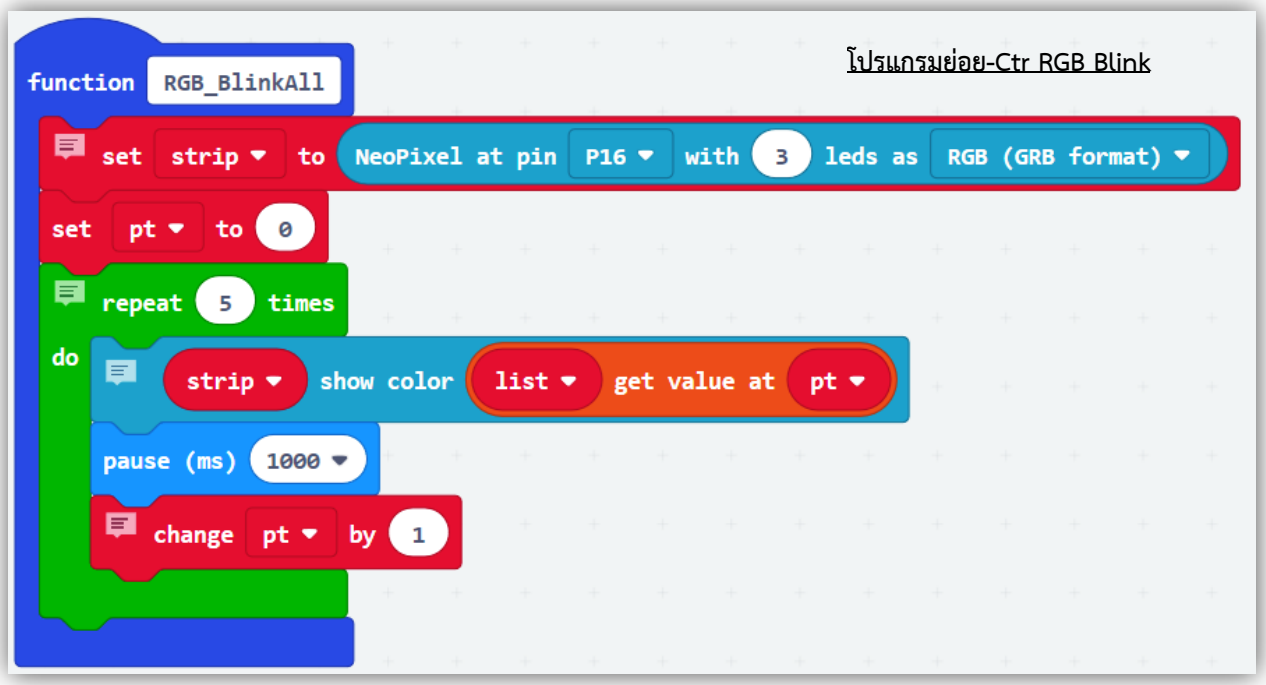
โปรแกรมย่อย-Ctr RGB Rainbow



โปรแกรมย่อย-Ctr Color RGB



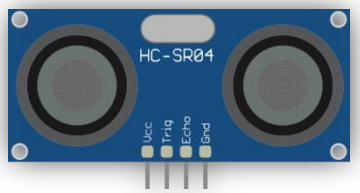
โปรแกรมย่อย-Ctr RGB Blink



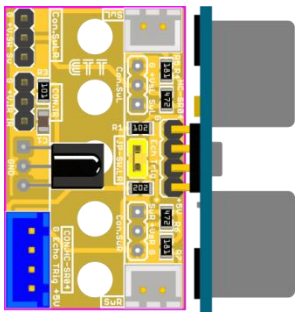
การทำงานของโปรแกรม : เริ่มต้น LED-RGB ทั้งหมดติดแบบ Rainbow ตามด้วยLED-RGB ติดทีละดวงตามลำดับของ RGB โดยจะวน 4 รอบ จากนั้น LED ทั้งหมดจะติดเป็นสีเดียวกันและมีการเปลี่ยนสีจนครบ 5 สี แล้วกลับไปเริ่มต้นโปรแกรมใหม่

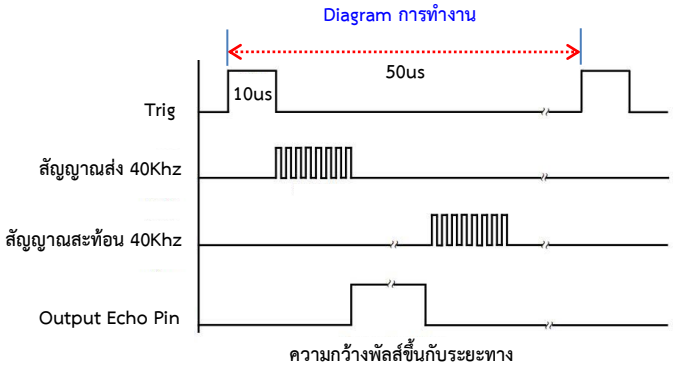
Ex6- Ultrasonic ReadDistance

ตัวอย่างนี้จะเป็นการศึกษาเพื่อทดสอบการเขียนโปรแกรมควบคุม Module Ultrasonic HC-SR04 ในการอ่านค่าระยะทาง หน่วย cm. มาแสดงผลที่ LED-Matrix ของบอร์ด Micro:bit โดยใช้ Block ในกลุ่มคำสั่ง Sonar ซึ่งหลักการทำงานของ Ultrasonic นั้นจะวัดระยะทางโดยการส่งคลื่นเสียงออกไปที่ความถี่หนึ่งและรับคลื่นเสียงที่สะท้อนกลับมา โดยจะบันทึกช่วงเวลาที่เริ่มต้นส่งคลื่นเสียงออกไปและคลื่นเสียงสะท้อนกลับมาแล้วนำค่าเวลาทั้งหมดมาคำนวณหาระยะของวัตถุ



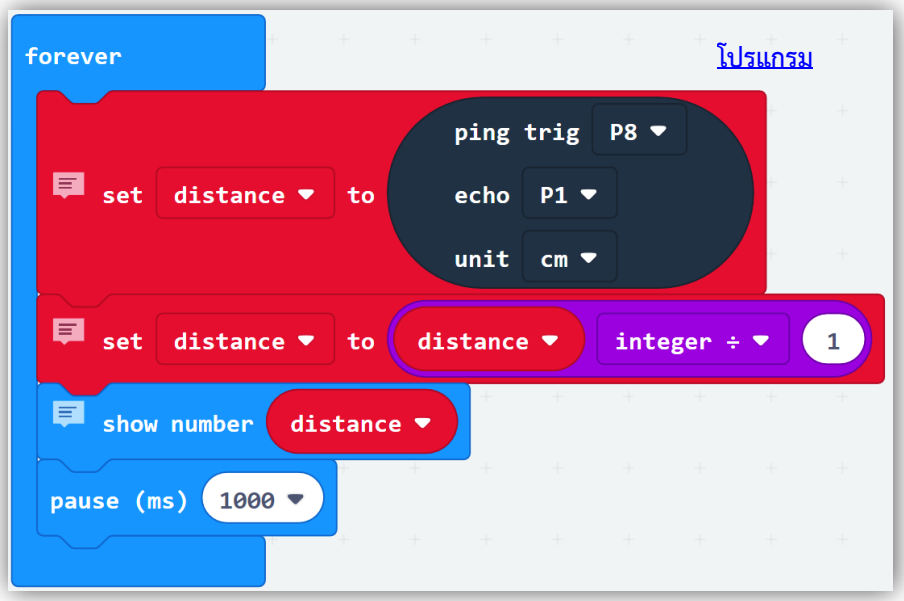
Module Ultrasonic


Module ใช้งานกับ ET-BBC ROBOT

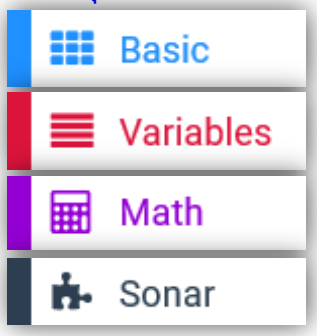


การ Control จะใช้ Pin P1 : Echo และ Pin P8 : Trig โดย P1 จะทำหน้าที่เป็น Input เพื่ออ่านคลื่นเสียงที่สะท้อนกลับจากวัตถุ ส่วน P8 จะใช้เป็น Output สำหรับส่งสัญญาณ Trig (Active 1 , Default 0) เพื่อเริ่มต้นส่งคลื่นเสียงออกไป ในการทดลองให้ Set Jumper JP-V1 = 5V ส่วน Jumper JP-LED = ไม่ Jump

โปรแกรม



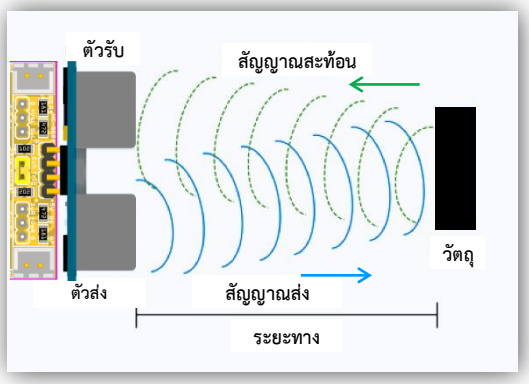
กลุ่มคำสั่งใช้งาน



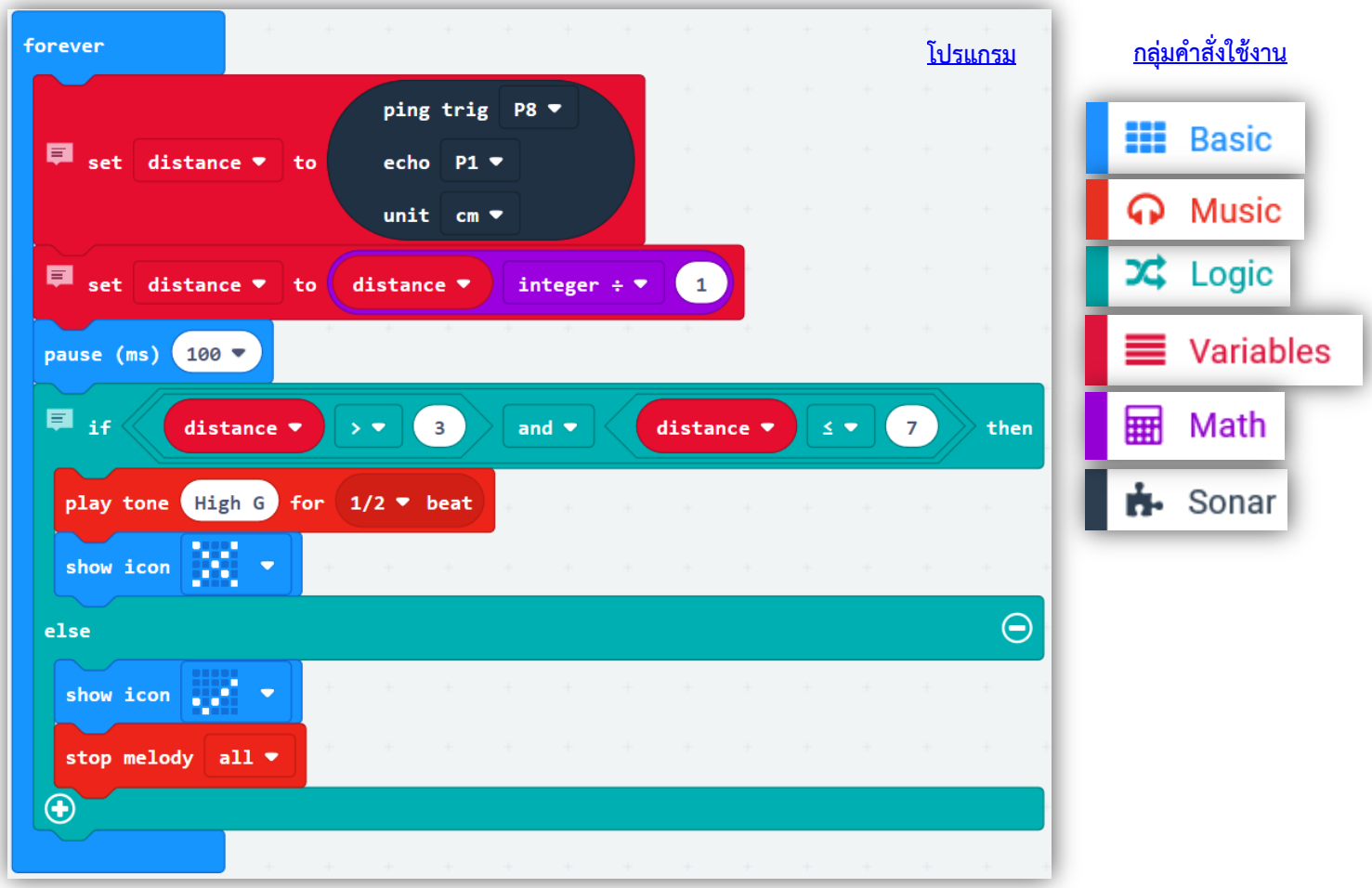
การทำงานของโปรแกรม : เนื่องจากในโปรแกรมเราใช้กลุ่มคำสั่งที่เป็น Block คำสั่งสำเร็จซึ่งเมื่อกำหนด Pin การสื่อสารเรียบร้อยก็จะคำนวณค่าระยะทางออกมาให้เป็น cm. เก็บไว้ในตัวแปรเราแค่นำค่าในตัวแปรมาใช้งาน ซึ่งในโปรแกรมจะทำการวนอ่านค่าระยะห่างที่ Ultrasonic ตรวจจับวัตถุ มาแสดงผลบน LED-Matrix ในรูปของเลขจำนวนเต็ม หน่วย Cm. สำหรับการใช้ Block เขียนโปรแกรมนั้นจากการทดลองพบว่าค่าระยะที่วัดได้จะน้อยกว่าความเป็นจริง 4-5 cm.

Ex7- Ultrasonic Detect Object

ตัวอย่างนี้จะเป็นการศึกษาเพื่อทดสอบการเขียนโปรแกรมควบคุม Module Ultrasonic HC-SR04 ในการตรวจจับวัตถุหรือสิ่งกีดขวางที่อยู่ด้านหน้าและนำผลที่ได้ไปสั่งให้ Robot ทำงานตามเงื่อนไขต่างๆที่ผู้ใช้ต้องการอัตโนมัติ โดยจะใช้การส่งคลื่นเสียงออกไปที่ความถี่หนึ่งและรับคลื่นเสียงที่สะท้อนวัตถุที่อยู่ด้านหน้ากลับมาแล้วคำนวณหาระยะของวัตถุ ถ้าอยู่ในระยะที่ผู้ใช้ตรวจจับ ก็สั่งให้ Robot ทำเงื่อนไขที่ต้องการ



การ Control จะใช้ Pin P1 : Echo และ Pin P8 : Trig โดย P1 จะทำหน้าที่เป็น Input เพื่ออ่านคลื่นเสียงที่สะท้อนกลับจากวัตถุ ส่วน P8 จะใช้เป็น Output สำหรับส่งสัญญาณ Trig (Active 1 , Default 0) เพื่อเริ่มต้นส่งคลื่นเสียงออกไป และ Pin P0 จะใช้สำหรับ Control Mini Speaker เพื่อกำเนิดเสียง Beep ในการทดลองให้ Set Jumper JP-V1 = 5V ส่วน Jumper JP-LED = ไม่ Jump , JP-SPK = Jump

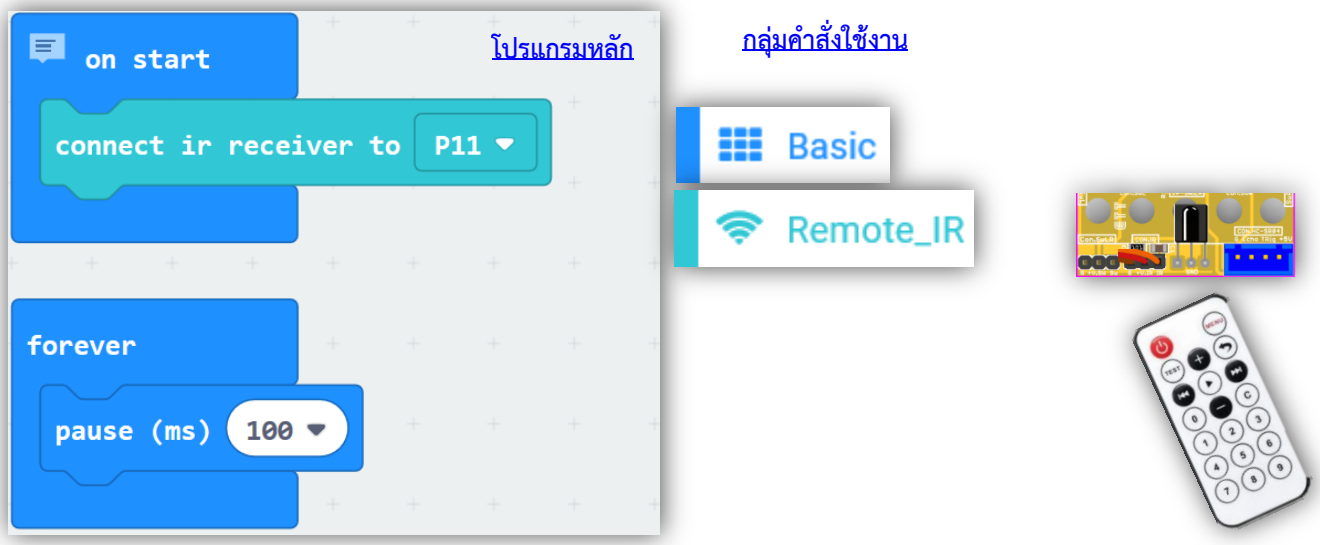


การทำงานของโปรแกรม : ให้ Ultrasonic วนอ่านค่าระยะวัตถุถ้าอยู่ในช่วง 3-7 cm ให้ LED-Matrix แสดงสัญลักษณ์ X และมีเสียง Beeb ดัง แต่ถ้าไม่อยู่ในช่วงที่กำหนดให้แสดงสัญลักษณ์เครื่องหมายถูก และเสียง Beep หยุดลง

Ex8- Receive IR Remote

ตัวอย่างนี้จะเป็นการศึกษาเพื่อทดสอบการเขียนโปรแกรมอ่านค่าจาก IR-Remote ตัวรับ ซึ่งรับค่าการกด Key จากผู้ใช้ด้วย Remote ตัวส่ง โดยใช้ Block ในกลุ่มคำสั่ง Remote_IR เป็นตัวถอดรหัส Key ที่กด และนำผล Key ที่ได้ไปควบคุมการทำงานต่างตามที่ผู้ใช้เขียนโปรแกรมการตอบสนองของ Key นั้นๆไว้

การ Control จะใช้ Pin P11 : IR สำหรับอ่านค่าความถี่จากตัว IR-Receive (4838) ที่รับสัญญาณการกด Key มาจาก Remote ตัวส่ง ในการทดลองให้ Set Jumper JP-V2 = 3V3 และห้ามใช้งาน SW.B เนื่องจาก Pin ทับซ้อนกันอยู่



โปรแกรมตอบสนองการกด Key บางส่วน

การทำงานของโปรแกรม : เมื่อกด ET-IR REMOTE Key จะแสดงอักษรย่อของ Key ที่ถูกกด บน Matrix-LED ของบอร์ด Micro:bit ในการทำงานของกลุ่มคำสั่ง Remote_IR นั้น เมื่อรับค่า Key มาถอดรหัสแล้วก็จะใช้การเปรียบเทียบ Key ว่าตรงกับโปรแกรมตอบสนอง Interrupt ของ Key ใดก็จะกระโดดไปทำงานในฟังก์ชั่นที่ตรงกับ Key นั้น

Ex9- Calibrate Tracking Sensor

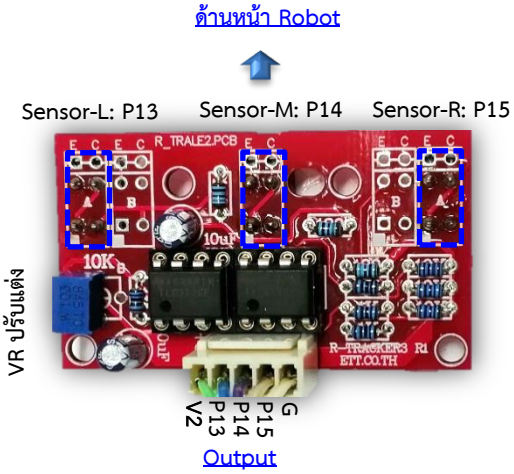
ตัวอย่างนี้จะเป็นการศึกษาเพื่อทดสอบการเขียนโปรแกรมอ่านค่า Sensor สำหรับตรวจจับเส้นเพื่อนำค่าการอ่าน ไปแสดงผลบน Matrix-LED และ RGB-LED เพื่อทำการตรวจสอบความพร้อมของตัว Sensor แต่ละตัวว่าทำงานถูกต้องตามสภาพแวดล้อมจริงหรือไม่ ถ้าไม่ถูกต้องจะได้ทำการปรับแต่งให้เหมาะสม

สำหรับ Sensor ตรวจจับเส้นที่ใช้จะเป็น Module R-Tracker แบบ Infrared ประกอบด้วย Sensor จำนวน 3 ตัว วางเรียงกัน 3 ตำแหน่ง ซึ่ง Module นี้ จะให้ Output ของ Sensor แต่ละตัวแบบ Logic โดยใช้หลักการสะท้อนกลับของสัญญาณ ในการตรวจจับเส้น ซึ่งสถานะของ Output จะมีการเปลี่ยนแปลงตามการสะท้อนของสัญญาณ Infrared ดังนี้

-เมื่อมีการสะท้อนสัญญาณ (พื้นผิวสีขาว) Output ของ sensor ตัวที่สะท้อนสัญญาณ จะให้ logic เป็น 0

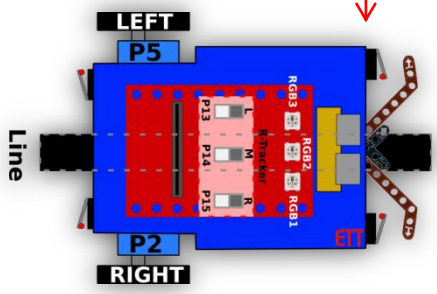
-เมื่อไม่มีการสะท้อนสัญญาณ (พื้นผิวสีดำ) Output ของ sensor ตัวที่ไม่สะท้อนสัญญาณ จะให้ logic เป็น 1

การอ่านค่า Output ของ R-Tracker มาใช้งานจะใช้ Pin P13 : อ่านค่า Sensor-ซ้าย , P14: อ่านค่า Sensor-กลาง และ P15 : อ่านค่า Sensor-ขวา โดยอ้างอิงตามรูปด้านล่าง ส่วน Pin P16 จะใช้ Control RGB-LED สำหรับแสดงสถานะ การทำงานของ Sensor ในการทดลองให้ Set Jumper JP-V2 = 3V3 และ JP-RGB = Jump

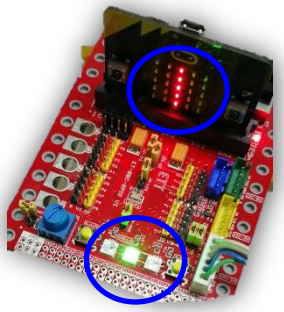


การทำงานของโปรแกรม : ในตัวอย่างนี้จะใช้ทดสอบและปรับแต่ง Sensor ตรวจจับเส้นให้ทำงานได้ถูกต้องเนื่องจากในแต่ละพื้นที่ที่มีการนำ Robot ไปใช้งานนั้นมีปริมาณความเข้มของแสงไม่เท่ากันทำให้ Sensor ตรวจจับเส้นทำงานผิดพลาดได้ ดังนั้นก่อนใช้งานผู้ใช้จะต้องทำการตรวจสอบและปรับแต่ง Sensor ก่อนเสมอ โดยเมื่อ Run โปรแกรมแล้วให้ทำดังนี้

- นำ Robot ไปวางไว้บนเส้นดำ โดยพื้นหลังเส้นดำต้องเป็นสีที่อ่อนกว่า ในการวางให้วาง Sensor ตัวใดตัวหนึ่งให้อยู่บนเส้นดำ ดังรูป จะให้ Sensor ตัวกลาง(P14) อยู่บนเส้นดำ



- จากนั้นปรับ VR ที่บอร์ด Sensor ดังรูป จนเห็น LED-RGB หรือ Matrix-LED ในตำแหน่งที่ตรงกับ Sensor ที่วางอยู่บนเส้นดำติดแล้วจึงหยุดปรับ เช่น เมื่อวาง Sensor-M บนเส้นดำ แล้วปรับ VR จนเห็น LED-RGB2 ติด หรือ Matrix-LED เส้นกลางติด ถือว่า Sensor-M ทำงานถูกต้อง ดังรูปด้านล่าง

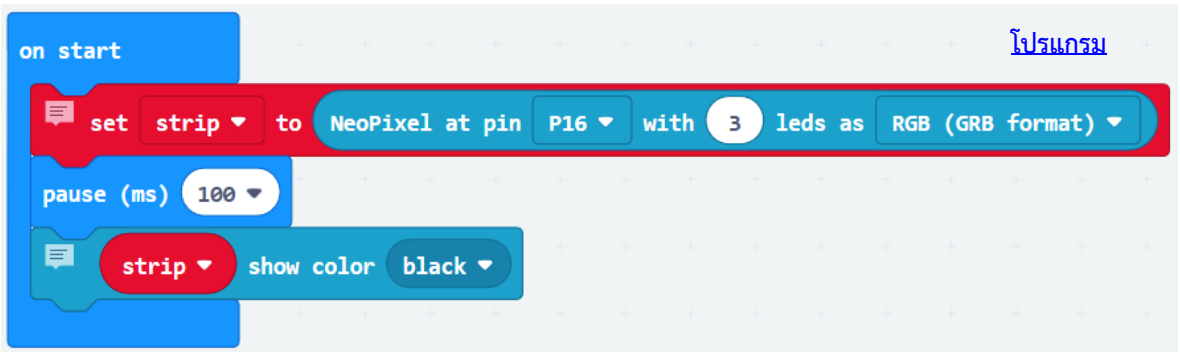


รูปแสดง LED-RGB2 และ Matrix-LED ตรงกลาง ติด เมื่อ Sensor-M อยู่บนเส้นดำและทำงานถูกต้อง

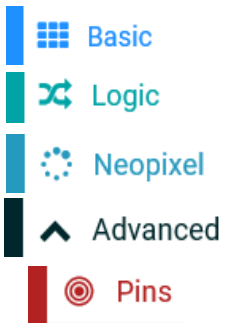
-หลังจากตั้ง Sensor ได้หนึ่งตัวแล้ว ลองเลื่อน Sensor ตำแหน่งอื่นให้ทับเส้นดำดูบ้าง จะเห็นว่า Led ในตำแหน่งที่ตรงกับ Sensor ที่ทับเส้นดำอยู่จะต้องติด ถ้ามี Sensor ทับเส้นดำ 2 ตัว Led ที่ตรงกับ Sensor ที่ทับเส้นดำก็จะติด 2 ดวงเช่นกัน ถ้าไม่มี Sensor ใดอยู่บนเส้นดำ LED ทั้งหมดจะต้องดับ

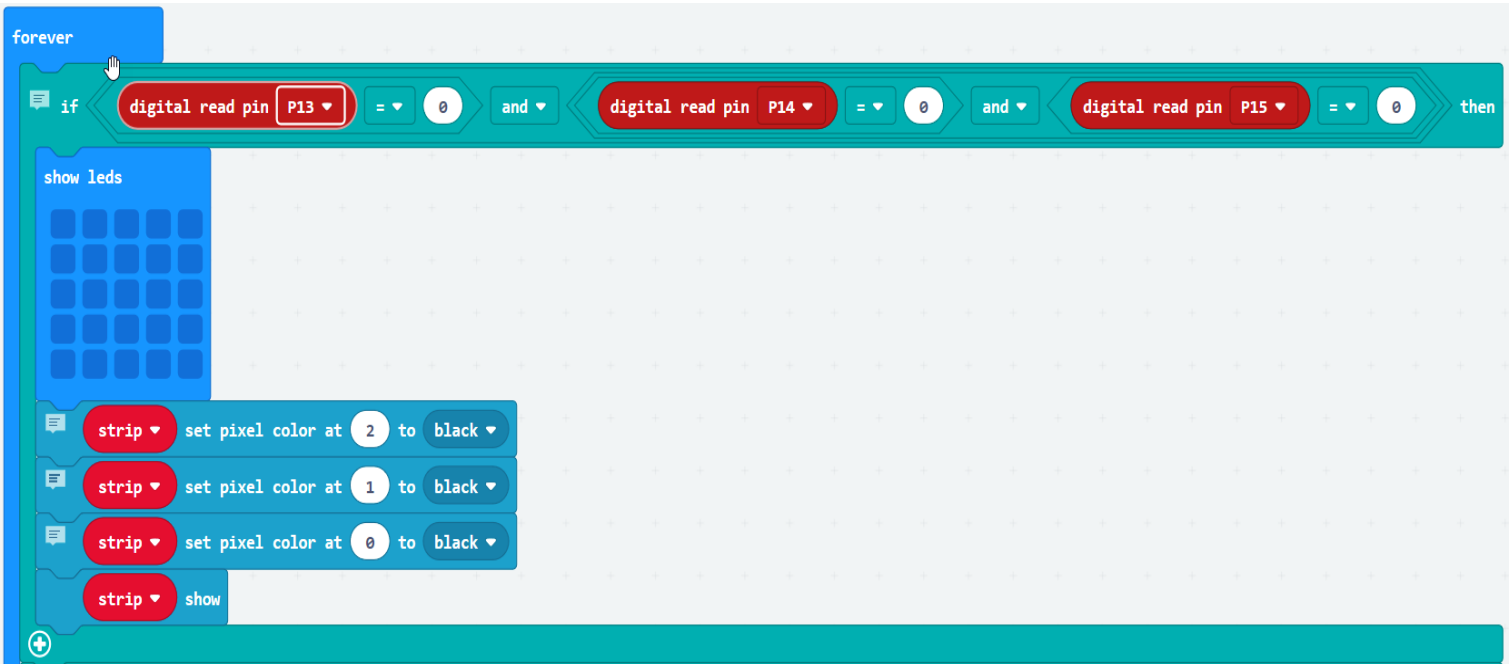
จากขั้นตอนการปรับแต่งจะเห็นว่าเมื่อ Sensor อยู่บนเส้นดำจะไม่มีการสะท้อนกลับของสัญญาณ จะได้ Output เป็น Logic 1 เราก็จะนำ Output ที่ได้ไปกำหนดเงื่อนไขให้ LED ในตำแหน่งเดียวกับ Sensor นั้นๆติดสว่าง แต่ถ้า Sensor ตัวใดไม่อยู่บนเส้นดำโดยที่เส้นดำมีพื้นหลังเป็นสีที่อ่อนกว่าจะทำให้มีการสะท้อนกลับของสัญญาณ จะได้ Output เป็น Logic 0 เราก็จะนำ Output ที่ได้ไปกำหนดเงื่อนไขให้ LED ในตำแหน่งเดียวกับ Sensor นั้นๆดับ เป็นต้น

โปรแกรม

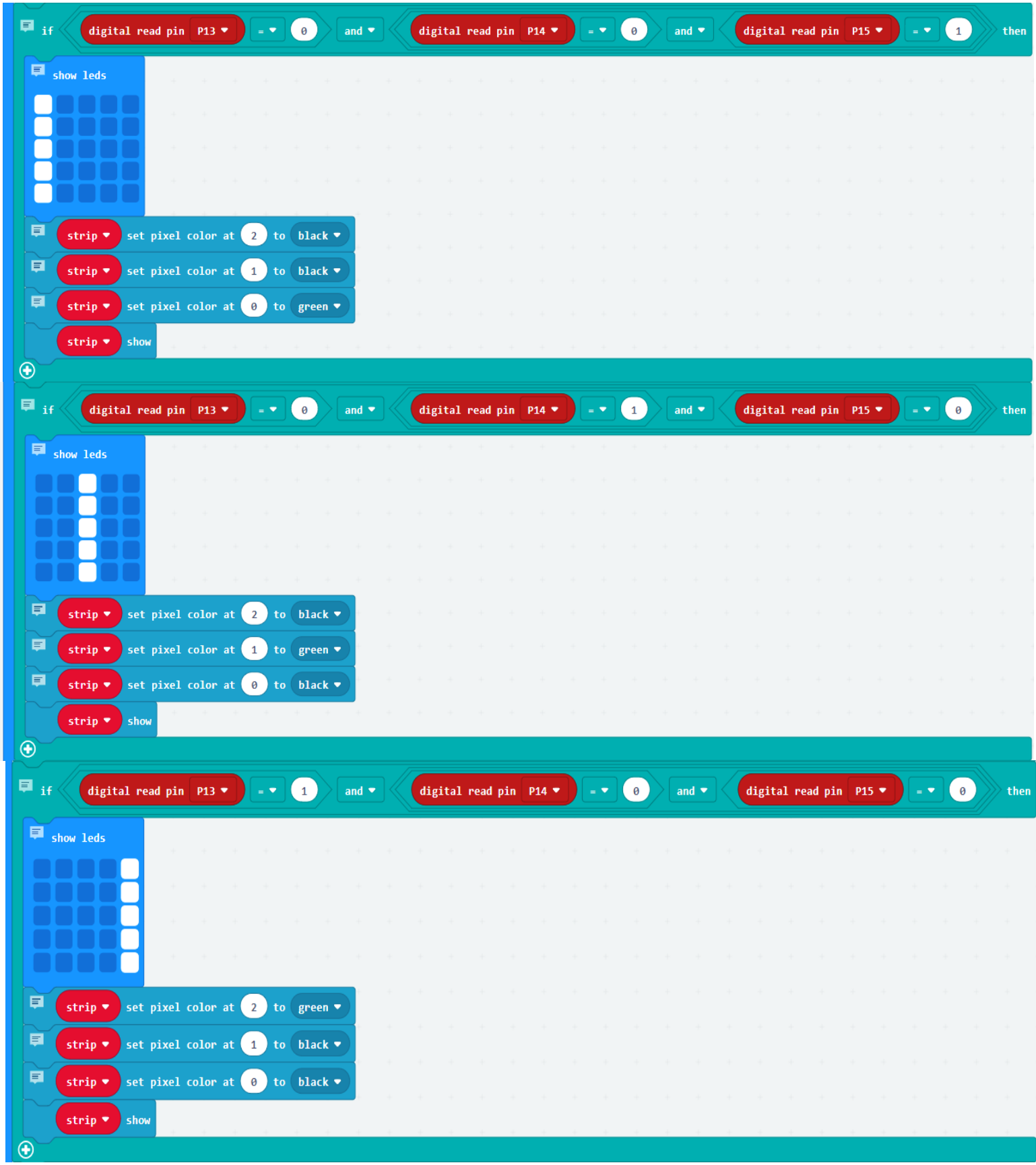


กลุ่มคำสั่งใช้งาน

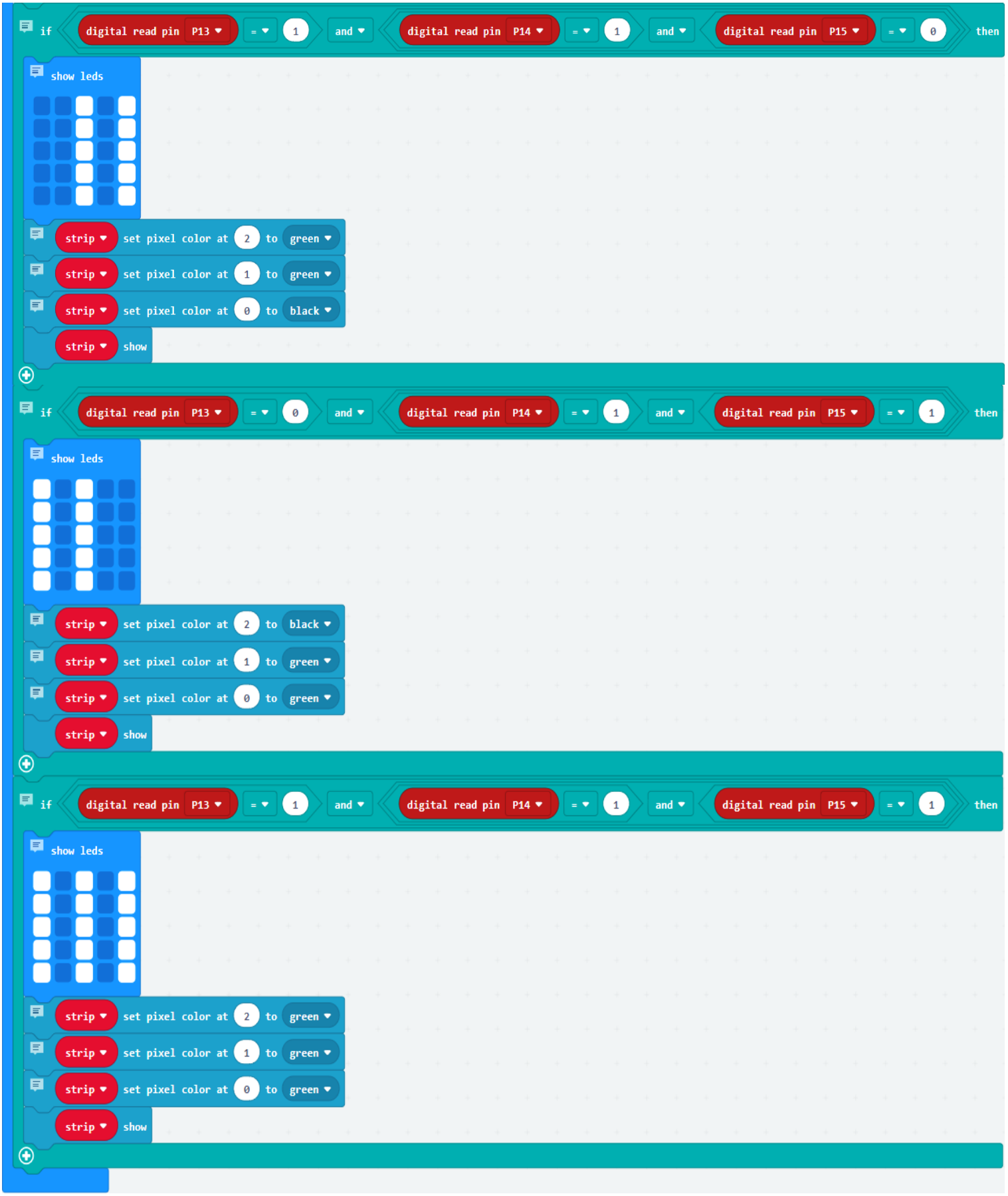




โปรแกรม(ต่อ)


if digital read pin P13 = 0 and digital read pin P14 = 0 and digital read pin P15 = 1 then
show leds
strip set pixel color at 2 to black
strip set pixel color at 1 to black
strip set pixel color at 0 to green
strip show
if digital read pin P13 = 0 and digital read pin P14 = 1 and digital read pin P15 = 0 then
show leds
strip set pixel color at 2 to black
strip set pixel color at 1 to green
strip set pixel color at 0 to black
strip show
if digital read pin P13 = 1 and digital read pin P14 = 0 and digital read pin P15 = 0 then
show leds
strip set pixel color at 2 to green
strip set pixel color at 1 to black
strip set pixel color at 0 to black
strip show

โปรแกรม(ต่อ)


if digital read pin P13 = 1 and digital read pin P14 = 1 and digital read pin P15 = 0 then
show leds
strip set pixel color at 2 to green
strip set pixel color at 1 to green
strip set pixel color at 0 to black
strip show
if digital read pin P13 = 0 and digital read pin P14 = 1 and digital read pin P15 = 1 then
show leds
strip set pixel color at 2 to black
strip set pixel color at 1 to green
strip set pixel color at 0 to green
strip show
if digital read pin P13 = 1 and digital read pin P14 = 1 and digital read pin P15 = 1 then
show leds
strip set pixel color at 2 to green
strip set pixel color at 1 to green
strip set pixel color at 0 to green
strip show

Ex10- Line Tracking Follower

ตัวอย่างนี้จะเป็นการศึกษาเพื่อทดสอบการเขียนโปรแกรมอ่านค่าสถานะของ Sensor ตรวจจับเส้น เพื่อนำค่าสถานะ Output ที่อ่านได้ไปควบคุมการเคลื่อนที่ของ Robot ให้วิ่งไปตามเส้นที่ผู้ใช้กำหนด

การ Control และ Pin ใช้งานสำหรับ Servo ให้อ้างอิงตามตัวอย่าง Ex4 ส่วนการตรวจจับเส้นให้อ้างอิงตามตัวอย่าง Ex9 และ Sw.B ใช้ Pin P11 ในการทดลองให้ Set Jumper JP-V1 = 5V , JP-V2 = 3V3 , JP-ADC = ไม่ Jump

หลักการควบคุม Robot ให้วิ่งตามเส้น

ในการควบคุมทิศทางหุ่นยนต์ให้วิ่งไปตามเส้นนั้นจะต้องเขียนโปรแกรมให้มีการตรวจสอบเงื่อนไขตามสถานะ output ของ Sensor ที่ใช้ตรวจจับเส้น ซึ่ง Module ที่นำมาใช้งานกับ Robot นั้น จะมี Sensor อยู่ด้วยกัน 3 จุด คือ ซ้าย(P13) , กลาง(P14) และ ขวา (P15) ทำให้เกิดเงื่อนไขขึ้นได้ทั้งหมด 8 เงื่อนไข (2^n เมื่อ n= จำนวน Sensor ที่ใช้) โดยรูปแบบของเงื่อนไขต่างๆในการตรวจจับเส้นจะพิจารณาเงื่อนไขตาม Output ที่ได้จากการทำงานของ Sensor 3 ตัว โดยใช้เส้นสีดำที่มีพื้นหลังเป็นสีขาวหรือสีอ่อนกว่า ดังนี้

-เมื่อ Sensor อยู่บนเส้นดำ ไม่มีการสะท้อนสัญญาณ จะได้ Output = 1

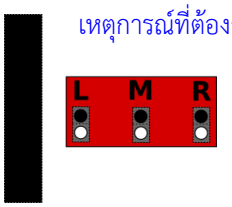
-เมื่อ Sensor อยู่นอกเส้นดำ มีการสะท้อนสัญญาณ จะได้ Output = 0

และรูปแบบเหตุการณ์ที่จะเกิดขึ้นได้ในแต่ละเงื่อนไขสรุปได้ดังนี้

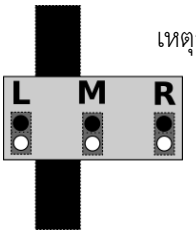
เมื่อ P13 = Output Sensor-L , P14 = Output Sensor-M , P15 = Output Sensor-R

เงื่อนไข 1 Output : P13=0 , P14=0 , P15=0 // ออกนอกเส้นทั้งหมด(000)

ตัวอย่างเหตุการณ์ที่เกิดได้ในเงื่อนไขนี้ : เงื่อนไขนี้ไม่ควรใช้เนื่องจากใช้แยกแยะการหลุดจากเส้นได้ไม่แน่นอน



เหตุการณ์ที่ต้องการตามเงื่อนไข

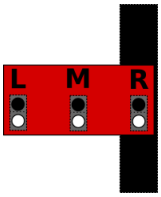


เหตุการณ์ที่ไม่ต้องการแต่เกิดได้

การ Control เช่น :ให้รถหยุดหรือหมุนกลับ หรือ แล้วแต่ผู้ใช้ต้องการ หรือ อื่นๆ

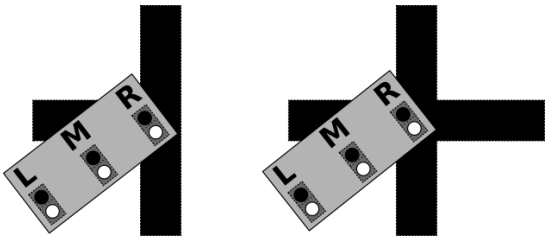
เงื่อนไข 2 Output : P13=0 , P14=0 , P15=1 //ออกนอกเส้นไปทางซ้าย(001)

ตัวอย่างเหตุการณ์ที่เกิดได้ในเงื่อนไขนี้ :



เหตุการณ์ที่ต้องการตามเงื่อนไข

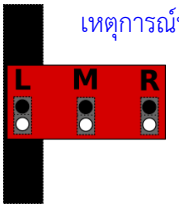
เหตุการณ์ที่ไม่ต้องการแต่เกิดได้



การ Control เช่น :ให้ปรับรถมาทางขวา หรือ อื่นๆ

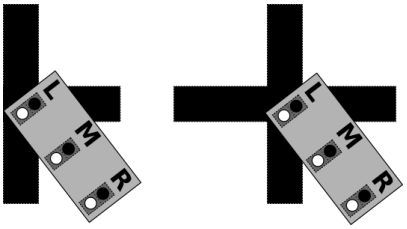
เงื่อนไข 3 Output : P13=1 , P14=0 , P15=0 // ออกนอกเส้นไปทางขวา (100)

ตัวอย่างเหตุการณ์ที่เกิดได้ในเงื่อนไขนี้ :



เหตุการณ์ที่ต้องการตามเงื่อนไข

เหตุการณ์ที่ไม่ต้องการแต่เกิดได้

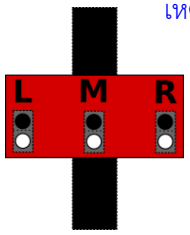


การ Control เช่น: ให้ปรับรถมาทางซ้าย หรือ อื่นๆ

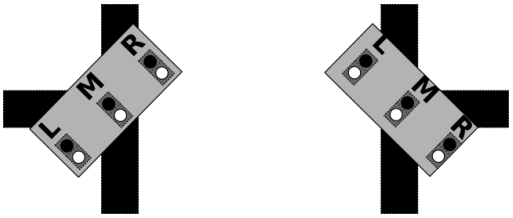
เงื่อนไข 4 Output : P13=0 ,P14=1 , P15=0 // อยู่กลางเส้น (010)

ตัวอย่างเหตุการณ์ที่เกิดได้ในเงื่อนไขนี้ :

เหตุการณ์ที่ต้องการตามเงื่อนไข



เหตุการณ์ที่ไม่ต้องการแต่เกิดได้

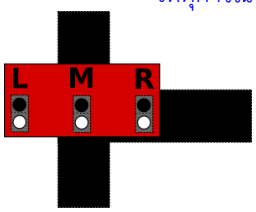


การ Control เช่น :ให้รถวิ่งตรงไป หรือ ถอยหลัง หรือ อื่นๆ

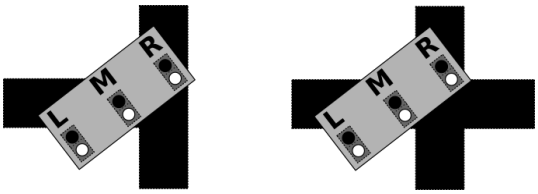
เงื่อนไข 5 Output : P13=0 , P14=1 , P15=1 // เจอทางแยกด้านขวา (011)

ตัวอย่างเหตุการณ์ที่เกิดได้ในเงื่อนไขนี้ :

เหตุการณ์ที่ต้องการตามเงื่อนไข



เหตุการณ์ที่ไม่ต้องการแต่เกิดได้

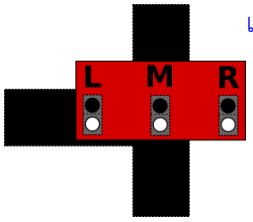


การ Control เช่น : ให้รถเลี้ยวขวา หรือ ตรงไปไม่สนใจแยก หรือ อื่นๆ

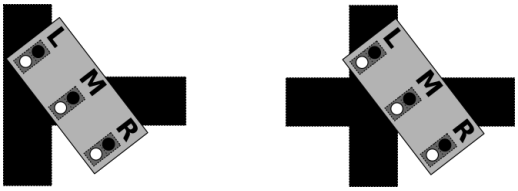
เงื่อนไข 6 Output : P13=1 , P14=1 , P15=0 // เจอทางแยกด้านซ้าย (110)

ตัวอย่างเหตุการณ์ที่เกิดได้ในเงื่อนไขนี้ :

เหตุการณ์ที่ต้องการตามเงื่อนไข



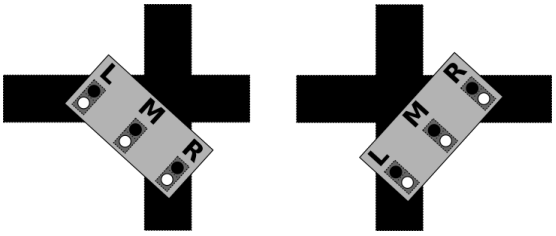
เหตุการณ์ที่ไม่ต้องการแต่เกิดได้



การ Control เช่น : ให้รถเลี้ยวซ้าย หรือ ตรงไปไม่สนใจแยก หรือ อื่น

เงื่อนไข 7 Output : P13=1 , P14=0 , P15=1 // เจอ4แยก หรือ ทางแยกซ้าย หรือ ทางแยกขวา (101)

ตัวอย่างเหตุการณ์ที่เกิดได้ในเงื่อนไขนี้ : เงื่อนไขนี้ไม่ควรใช้เนื่องจากแยกแยะทางแยกไม่ได้ว่าแยกซ้ายหรือขวาหรือ4แยก



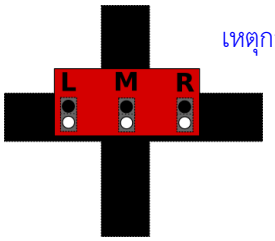
เหตุการณ์ที่ไม่ต้องการแต่เกิดได้

การ Control เช่น :ให้รถเลี้ยวขวา หรือ ให้รถเลี้ยวซ้าย หรือ ตรงไปไม่สนใจแยก หรือ อื่นๆ

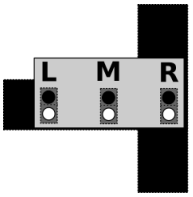
เงื่อนไข 8 Output : P13=1 , P14=1 , P15=1 // เจอ4แยก (111)

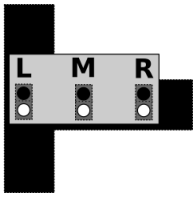
ตัวอย่างเหตุการณ์ที่เกิดได้ในเงื่อนไขนี้ :

เหตุการณ์ที่ต้องการตามเงื่อนไข



เหตุการณ์ที่ไม่ต้องการแต่เกิดได้





การ Control เช่น : ให้รถเลี้ยวขวา หรือ ให้รถเลี้ยวซ้าย หรือ ตรงไปไม่สนใจแยก หรือ อื่นๆ

จากเงื่อนไขทั้งหมดข้างต้นสามารถพิจารณาเลือกเงื่อนไขมาใช้ในการเขียนโปรแกรมได้ดังนี้ (ไม่จำเป็นต้องนำมาใช้ทุกเงื่อนไข)

กรณีที่1 ถ้าเส้นทางที่รถใช้วิ่งไม่มีทางแยก(ทางเดี่ยว) ผู้ใช้สามารถนำเงื่อนไขที่ 2 (001), เงื่อนไขที่3 (100) และ เงื่อนไขที่4 (010) มาใช้ก็พอ หรือใช้เฉพาะเงื่อนไขที่2 กับ 3 ก็ได้ เนื่องจากกรณีนี้เมื่อพิจารณาจาก Module Sensor ที่ใช้งาน โอกาสที่ Output Sensor จะเป็น 1 ทั้งสอง Output พร้อมกันไม่น่าจะเกิด ส่วนการวิ่งของรถอาจจะมีการส่ายมากบ้างน้อยบ้าง หรือหลุดเส้น ก็ขึ้นอยู่กับความเร็วในการวิ่งและการปรับรถเข้าหาเส้นต้องกำหนดให้เหมาะสม เนื่องจาก Sensor ที่น้อยตัว โอกาสที่จะค่อยๆปรับรถเข้าหาเส้นที่ละนิดทำได้ยาก

กรณีที่2 ถ้าเส้นทางที่รถใช้วิ่งมีทางแยก ผู้ใช้จะต้องนำเงื่อนไขที่ 2 (001), เงื่อนไขที่3 (100) และ เงื่อนไขที่4 (010) มาใช้ร่วมกับเงื่อนไขที่สอดคล้องกับทางแยกได้แก่ เงื่อนไขที่5 (011) , เงื่อนไขที่6 (110) ,เงื่อนไขที่7 (101) และ เงื่อนไขที่8 (111) แต่ต้องบอกก่อนว่าเมื่อมาถึงทางแยกโอกาสที่รถจะทำตามเงื่อนไขตามที่ผู้ใช้เขียนกำหนดไว้ในโปรแกรมทุกครั้งที่เจอทางแยกนั้นเป็นไปได้ยาก เนื่องจากจังหวะที่รถวิ่งมาถึงทางแยก อาจจะไปตรงกับเงื่อนไขทางเดี่ยว (เงื่อนไข 2,3,4) ก็จะทำให้การ Control รถผิดพลาด อย่าลืมว่าในความเป็นจริงรถจะวิ่งส่ายไปมาโอกาสที่รถจะวิ่งมาเจอเงื่อนไขที่สอดคล้องกับทางแยกก็เป็นไปได้แต่จะไม่ทุกครั้ง การแก้จะต้องเพิ่มจำนวน Sensor ให้มากขึ้นเพื่อจะได้เงื่อนไขเพิ่มขึ้นไม่ไปชนกับเงื่อนไขทางเดี่ยว

หมายเหตุ : สำหรับเงื่อนไขที่1 ออกนอกเส้นทั้งหมด(0,0,0) ไม่แนะนำให้ใช้งานเพราะจะทำให้ Robot ทำงานผิดพลาด เนื่องจากระยะห่างของ Sensor มีขนาดกว้างกว่าเส้นดำ(อ้างอิงความกว้างตามเทปพันสายไฟ) ทำให้สามารถเกิดเหตุการณ์ขึ้นได้ 2 แบบ คือ Robot ออกนอกเส้นจริง กับ Robot อยู่บนเส้นโดยเส้นนั้นอยู่ระหว่างช่อง Sensor ถ้าเรานำเงื่อนไขนี้มาใช้ Robot จะแยกไม่ได้ว่า เงื่อนไขนี้เกิดจากเหตุการณ์ไหนกันแน่ เช่น ถ้าเราเลือกให้ Robot Stop เมื่อออกจากเส้นทั้งหมด Robot ก็อาจจะ Stop ทั้งที่ยังไม่ออกจากเส้นทั้งหมดก็ได้ ถ้าเงื่อนไขนี้เกิดขึ้นในจังหวะเส้นดำอยู่ระหว่าง Sensor เป็นต้น

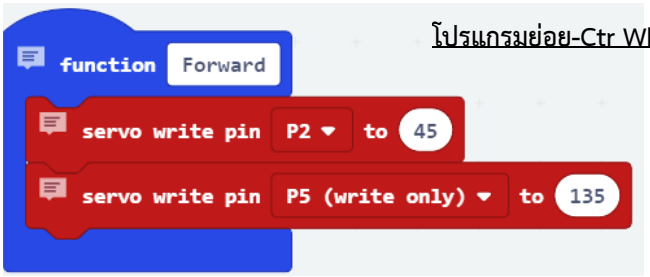
สำหรับการตรวจจับเส้นที่กล่าวมาทั้งหมดข้างต้นจะอ้างอิงกับ Module Sensor ตรวจจับเส้นที่ให้ไปกับ Robot เท่านั้น ถ้ามีการปรับเปลี่ยน Sensor ตรวจจับเส้นนอกเหนือจากที่ให้ไปผู้ใช้จะต้องวิเคราะห์เหตุการณ์ที่จะเกิดขึ้นในแต่ละเงื่อนไขเอาใหม่ตามความเป็นจริงที่จะเกิดกับ Sensor ที่นำมาใช้งาน โดยจำนวนเงื่อนไขที่ได้จะขึ้นอยู่กับจำนวน Sensor ที่นำมาใช้งาน 2^n เสมอ แต่จำนวนเหตุการณ์ที่เกิดขึ้นในแต่ละเงื่อนไขอาจมากน้อยไม่เท่ากันยิ่ง Sensor มากยิ่งดี จะทำให้รถมีความผิดพลาดในการตรวจจับเส้นน้อยลงโดยเฉพาะทางแยก และช่วยลดการส่ายของรถได้ในระดับหนึ่ง เนื่องจากถ้า Sensor มาก จะทำให้ได้จำนวนเงื่อนไขเพิ่มขึ้น ทำให้จำนวนเหตุการณ์ที่เกิดขึ้นหลายๆเหตุการณ์ในเงื่อนไขเดียวกันลดลงเพราะถูกแยกออกไปในเงื่อนไขที่เพิ่มเข้ามาทำให้การควบคุมรถทำได้แม่นยำขึ้น ยิ่งถ้าได้หนึ่งเงื่อนไขมีจำนวนเหตุการณ์เดียวจะดีที่สุด

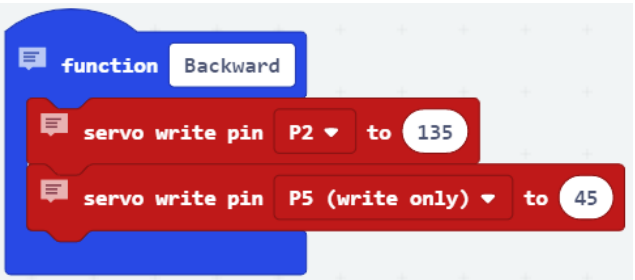
แนะนำ -เพื่อรองรับการทำงานของโปรแกรม เส้นที่ใช้ควรเป็นเส้นสีดำบนพื้นหลังสีขาวหรือสีที่อ่อนกว่า โดยตัวเส้นควรใช้เทปพันสายไฟสีดำ ไม่แนะนำให้ใช้เส้นจากการ Print เพราะจะทำให้ Sensor อ่านสถานะ Logic ผิดพลาด ส่งผลให้ Robot หลุดเส้นได้ง่าย

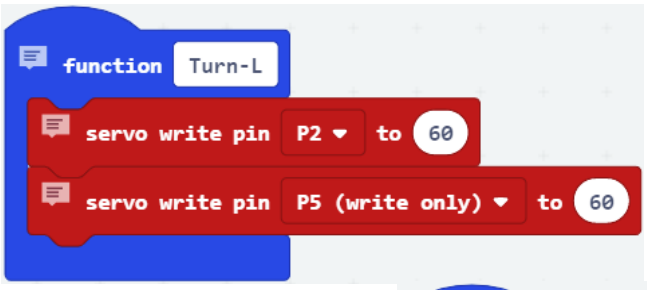
-เส้นทางที่ใช้ทดสอบ Robot ไม่ควรเป็นทางแยก หรือ มุมฉาก เนื่องจาก Sensor มีจำนวนน้อยทำให้การตรวจจับอาจผิดพลาดเวลาเจอทางแยกหรือมุมฉาก Robot ก็อาจหลุดเส้น หรือ ไม่วิ่งไปในทิศทางที่ผู้ใช้กำหนดไว้ในโปรแกรม

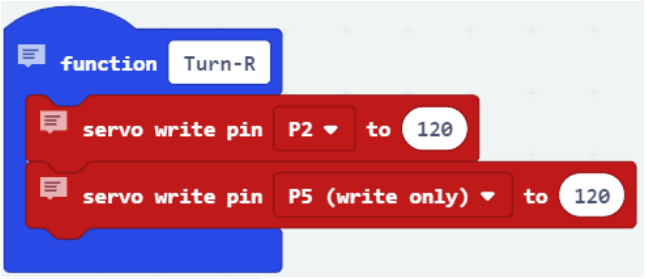
-การลดความเร็วการวิ่งของ Robot จะช่วยลดการหลุดจากเส้นของ Robot ได้

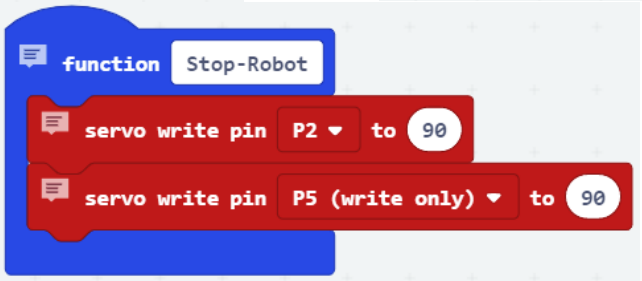
โปรแกรมย่อย-Ctr Wheel



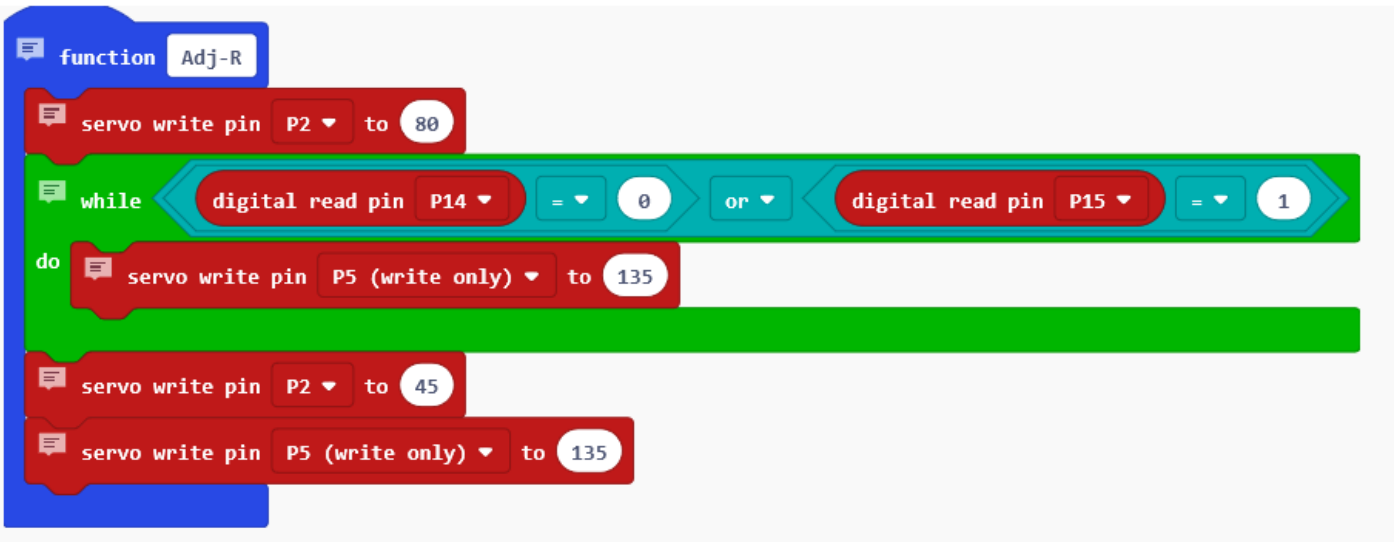




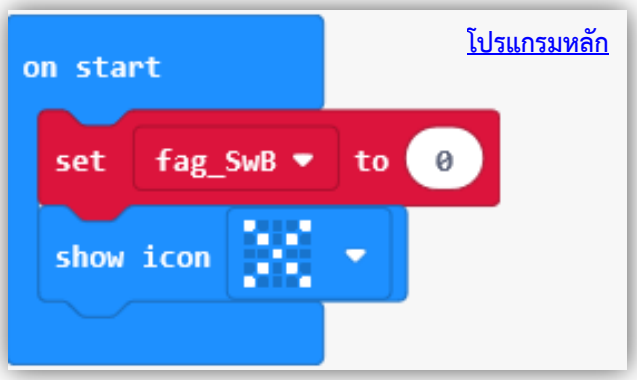




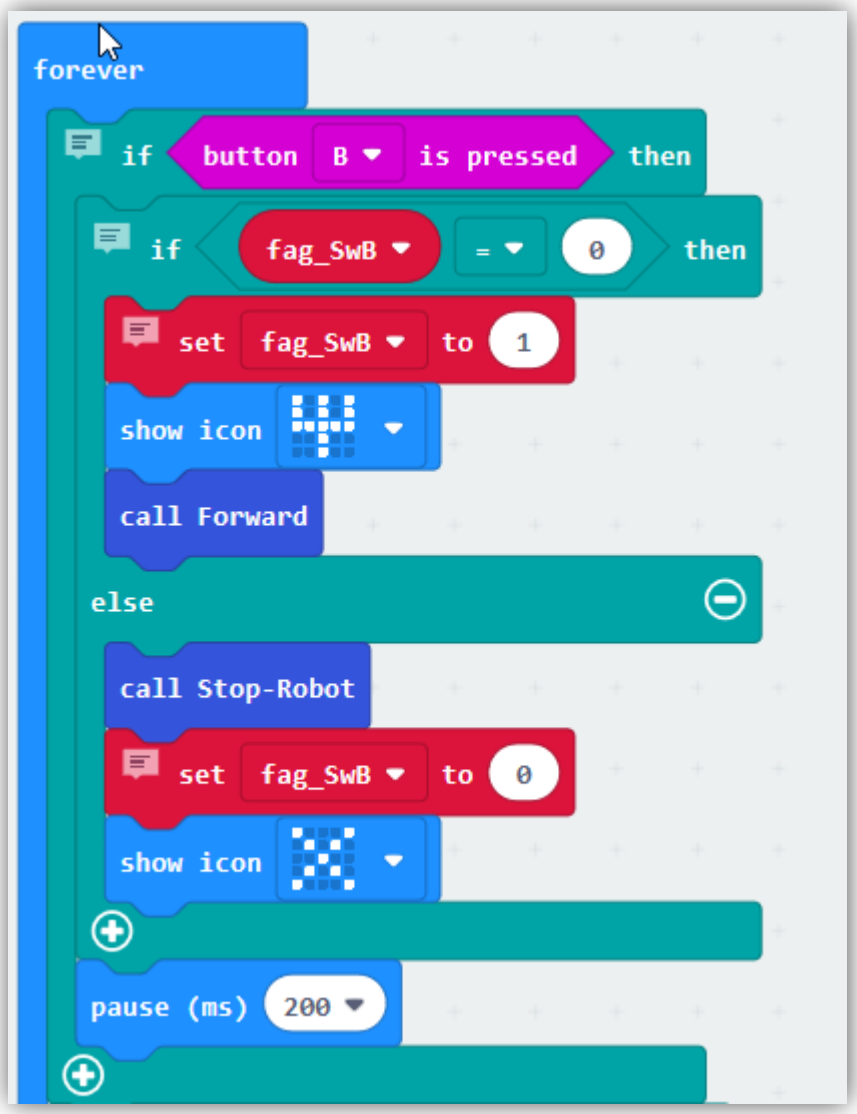
โปรแกรมย่อย- สำหรับปรับล้อเข้าหาเส้น

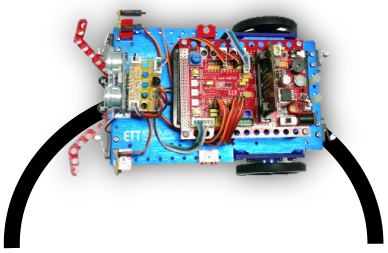


โปรแกรมหลัก

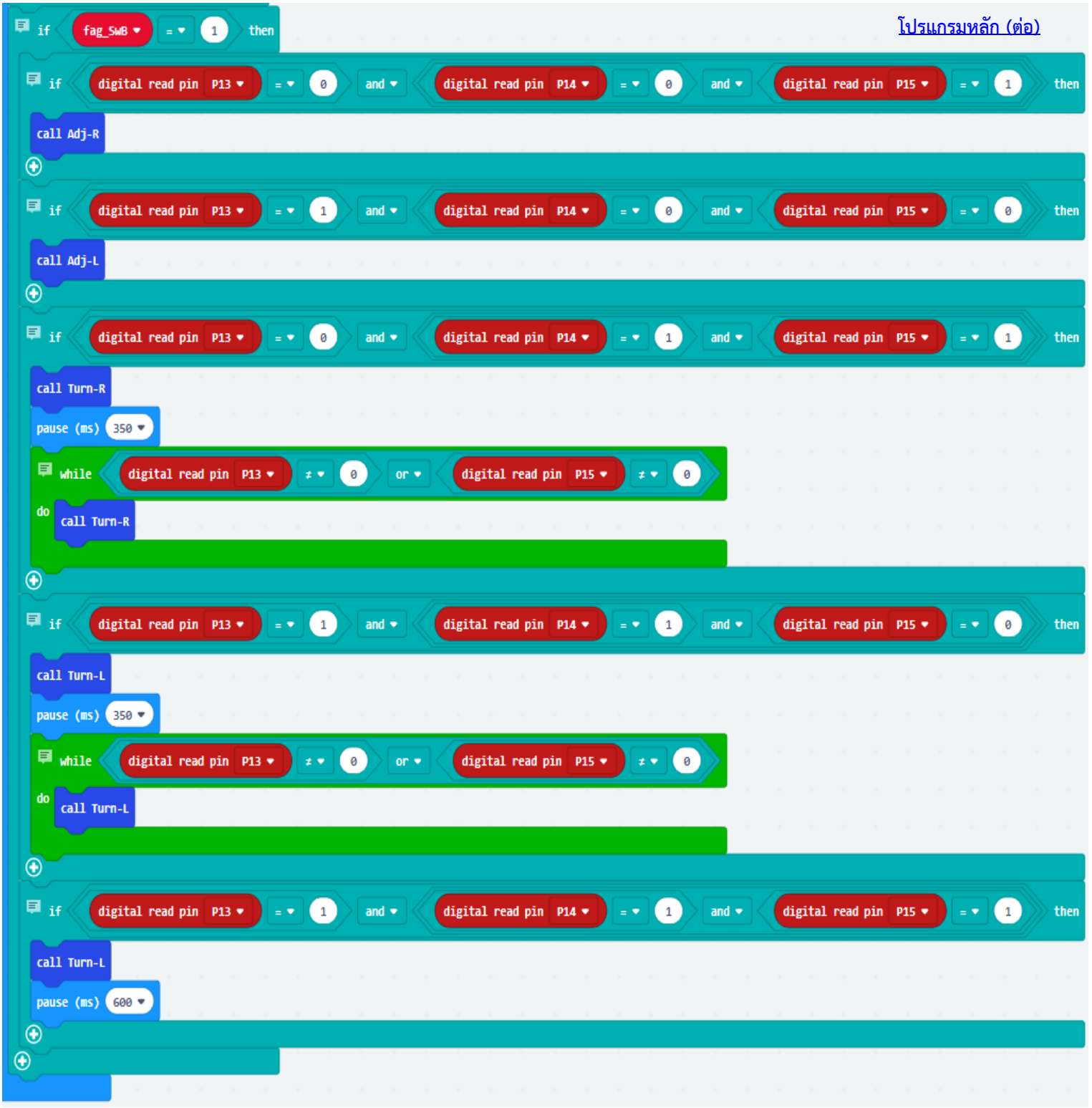


กลุ่มคำสั่งใช้งาน





โปรแกรมหลัก (ต่อ)



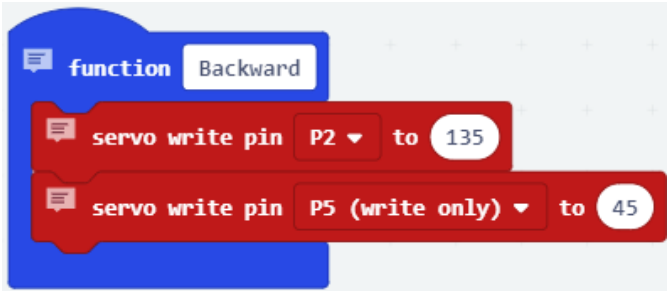
การทำงานของโปรแกรม : เมื่อ Power-ON ให้วาง Robot ไว้บนเส้นดำ แล้วกด SW.B เพื่อ Start Robot ตัว Robot ก็จะเริ่มเดินไปข้างหน้า ขณะเดียวกันโปรแกรมก็จะวนอ่านสถานะ Sensor ตรวจจับเส้น มาตรวจสอบเงื่อนไข เพื่อใช้ปรับ Servo บังคับให้ Robot วิ่งไปตามเส้นได้ถูกต้อง เมื่อต้องการ Stop Robot ให้กด Sw.B อีกครั้ง

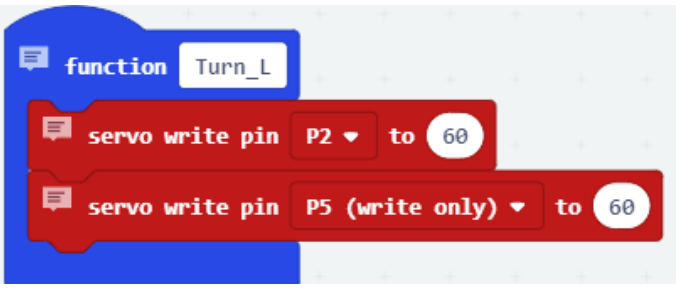
Ex11- LineTracking+Sonic-SW

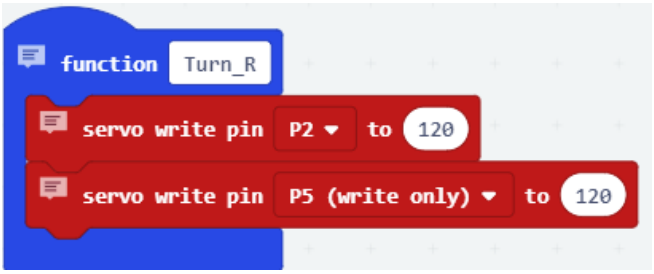
ตัวอย่างนี้จะเป็นการศึกษาเพื่อทดสอบการเขียนโปรแกรมควบคุมให้ Robot ตรวจสอบสิ่งกีดขวางที่อยู่ด้านหน้าด้วย Sensor Ultrasonic และ Sw.กันชนหน้าและหลังในขณะที่มีการเดินตามเส้นอยู่ เมื่อพบสิ่งกีดขวางก็ให้ Robot หมุนกลับทางเดิม หรือผู้ใช้สามารถเปลี่ยนให้ทำอย่างอื่นได้ตามต้องการ

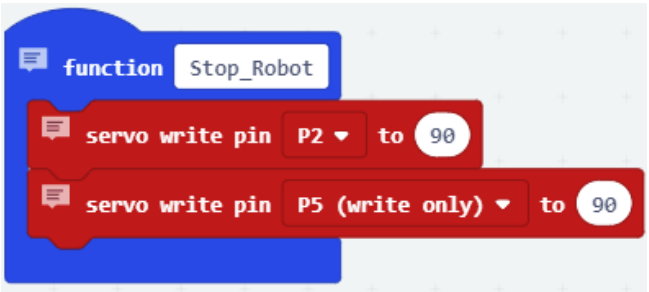
การ Control และ Pin ใช้งานสำหรับ Servo ให้อ้างอิงตามตัวอย่าง Ex4 ส่วนการตรวจจับเส้นให้อ้างอิงตามตัวอย่าง Ex9,Ex10 และ การตรวจจับสิ่งกีดขวางให้อ้างอิงในตัวอย่าง Ex7 ส่วน Sw.B ใช้ P11 , Sw.กันชน คู่หน้าใช้ P12 , Sw.กันชนหลังขวา P19 และ Sw.กันชนหลังซ้าย P20 ในการทดลองให้ Set Jumper JP-V1 = 5V , JP-V2 = 3V3 , JP-LED กับ JP-ADC = ไม่ Jump

โปรแกรมย่อย-Ctr Wheel



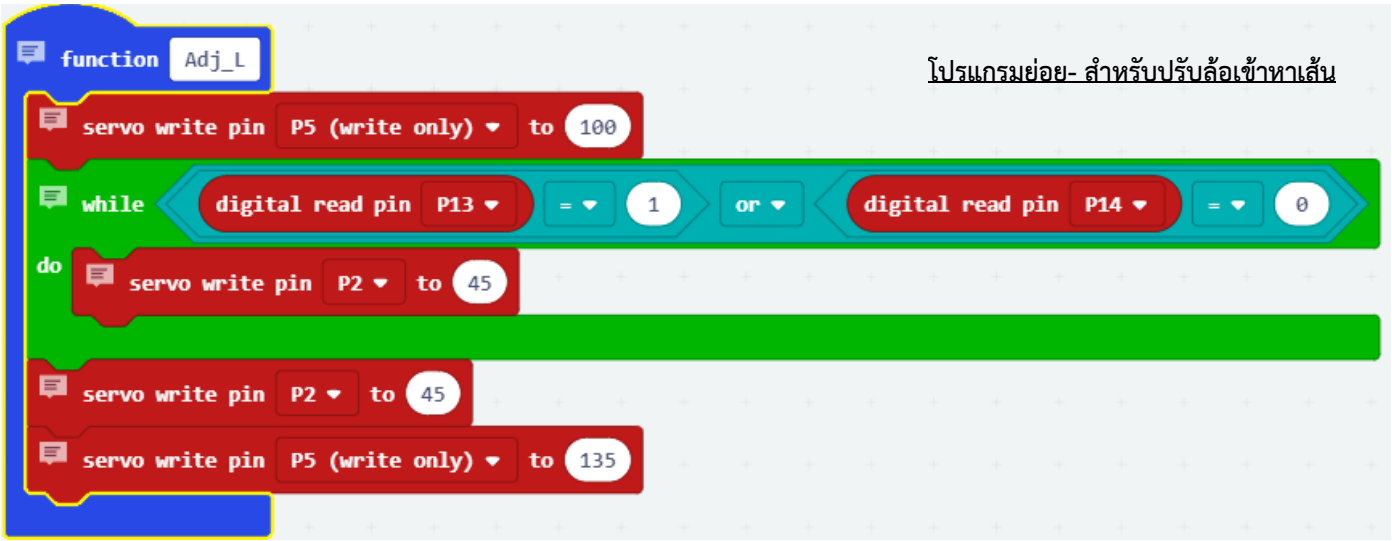




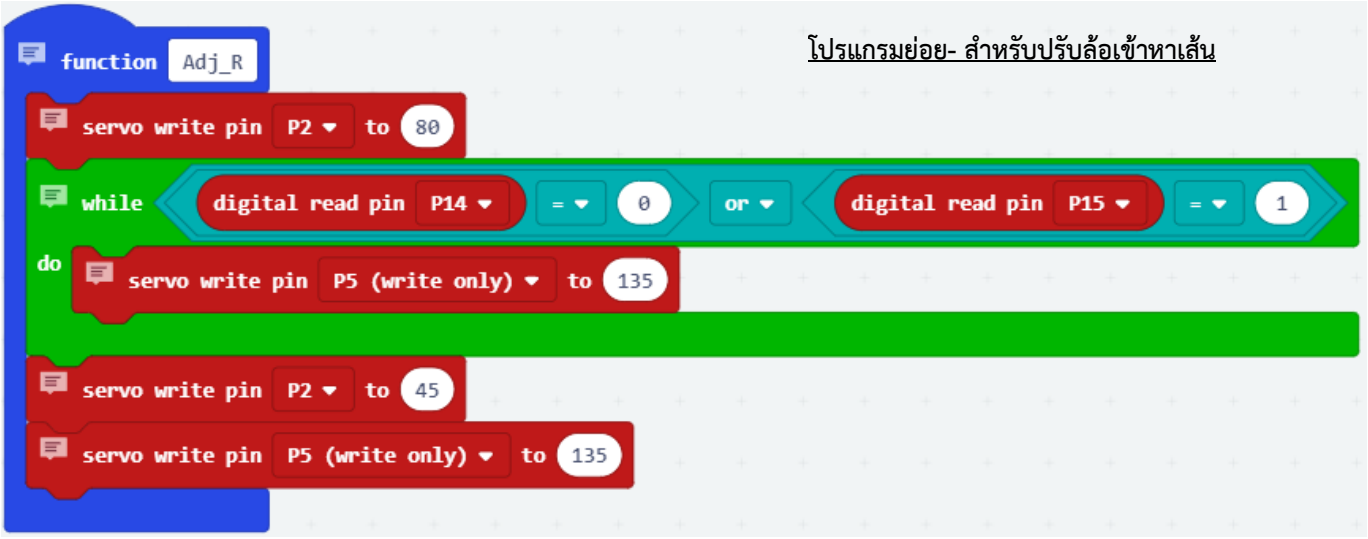


โปรแกรมย่อย-อ่านระยะทาง

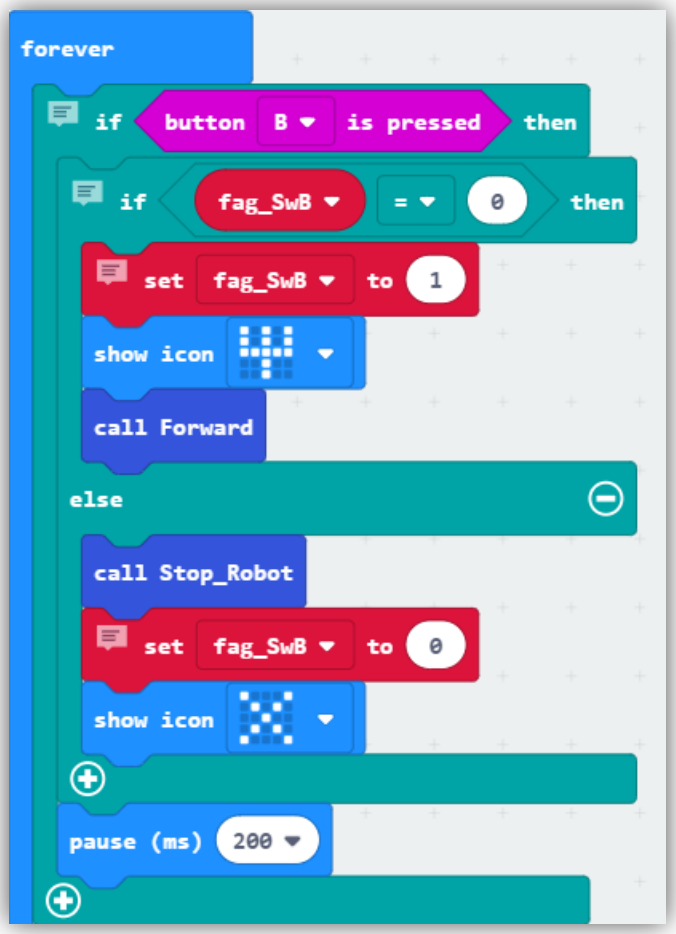
โปรแกรมย่อย- สำหรับปรับล้อเข้าหาเส้น



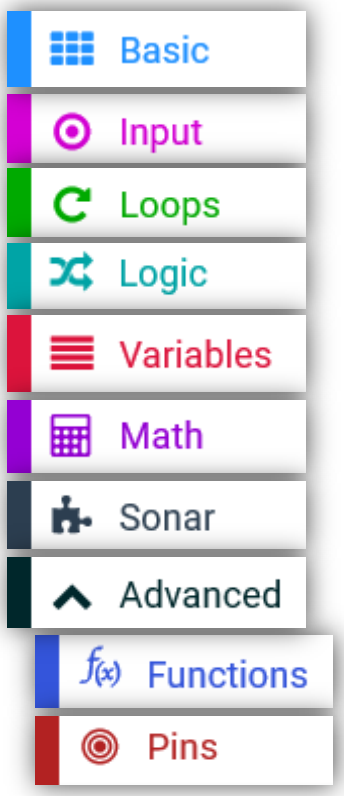
โปรแกรมย่อย- สำหรับปรับล้อเข้าหาเส้น



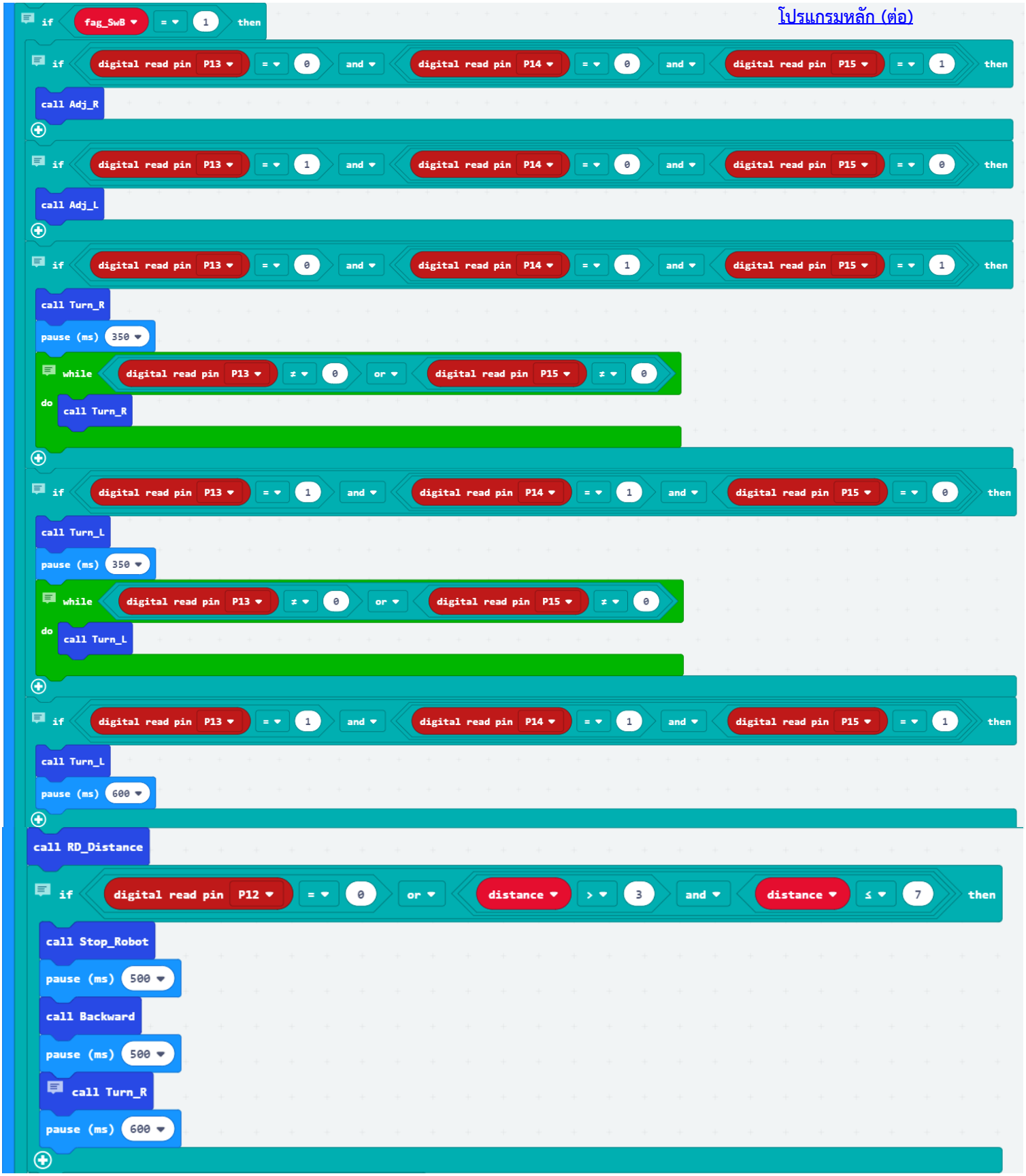
โปรแกรมหลัก



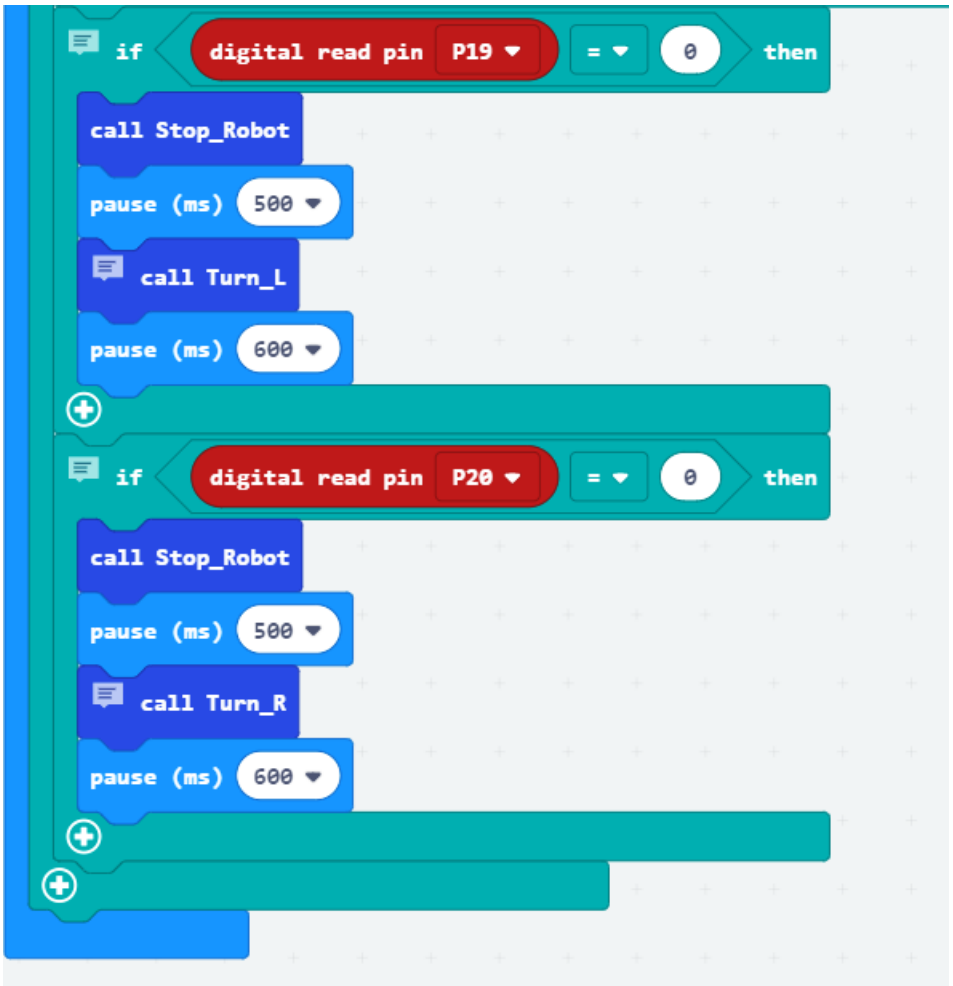
กลุ่มคำสั่งใช้งาน

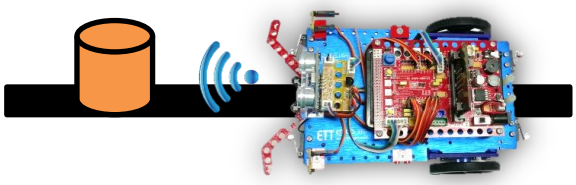


โปรแกรมหลัก (ต่อ)



โปรแกรมหลัก (ต่อ)





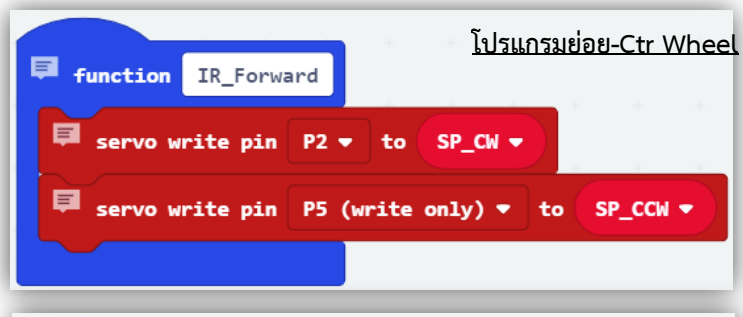
การทำงานของโปรแกรม : เมื่อ Power ON กด Sw.B เพื่อ Start Robot สำหรับเดินตามเส้น ขณะเดินตามเส้น ถ้า Sw.กันชนคู่หน้าถูกกด หรือ มีวัตถุมากั้นด้านหน้าในระยะที่กำหนดให้ Ultasonic ตรวจจับ(10 cm) Robot จะถอยหลังและหมุนตัวกลับ180 องศา เพื่อเปลี่ยนทางวิ่งและถ้า Sw. กันชนหลังถูกกดขณะที่ Robot เดินตามเส้นอยู่ Robot จะหมุนตัวกลับ 180 องศา เพื่อเปลี่ยนทางวิ่ง เมื่อต้องการ Stop Robot ให้กด Sw.B อีกครั้ง

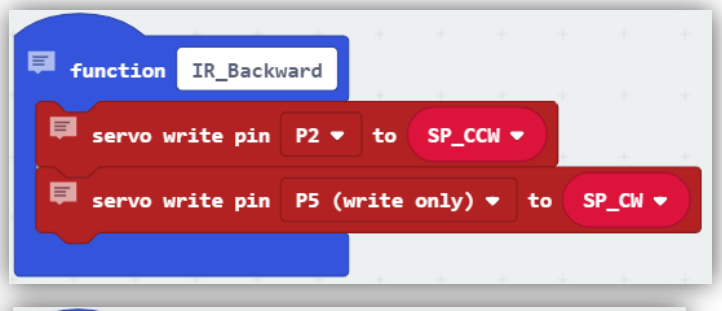
Ex12- Remote_RGB Robot

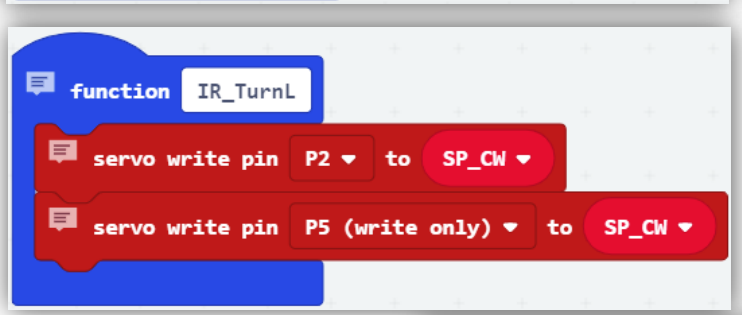
ตัวอย่างนี้จะเป็นการศึกษาเพื่อทดสอบการเขียนโปรแกรมให้ Robot รับคำสั่งจาก ET-IR Remote Key โดยให้ทำงานตามคำสั่งของ Key ที่ผู้ใช้กด เช่น สั่งให้ Robot เดินหน้าด้วย Key+ , สั่งให้ Robot ถอยหลังด้วย Key – หรือ สั่งให้ Robot เปิด-ปิด ไฟ LED-RGB ด้วย Key ใดๆ แล้วแต่ผู้ใช้จะกำหนดเองซึ่งสามารถปรับเปลี่ยนได้ในโปรแกรมตามที่ผู้ใช้ต้องการ

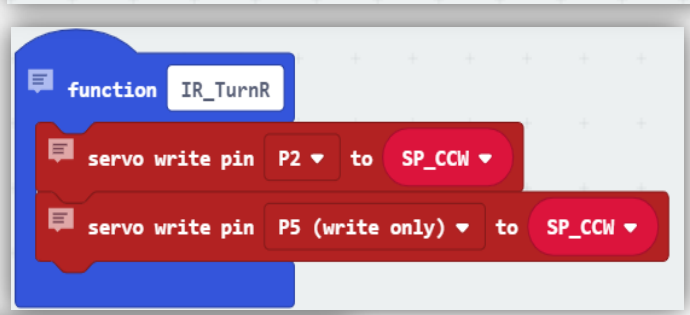
การ Control และ Pin ใช้งานสำหรับ Servo ให้อ้างอิงตามตัวอย่าง Ex4 ส่วน การ Control LED-RGB ทั้งหมดใช้ Pin P16 , การอ่านค่าสัญญาณความถี่จาก Remote ใช้ Pin P11 และ Mini Speaker ใช้ Pin P0 ในการทดลองให้ Set Jumper JP-V1 = 5V , JP-V2 = 3V3 , JP-LED กับ JP-ADC = ไม่ Jump , JP-RGB และ JP-SPK = Jump

โปรแกรมย่อย-Ctr Wheel

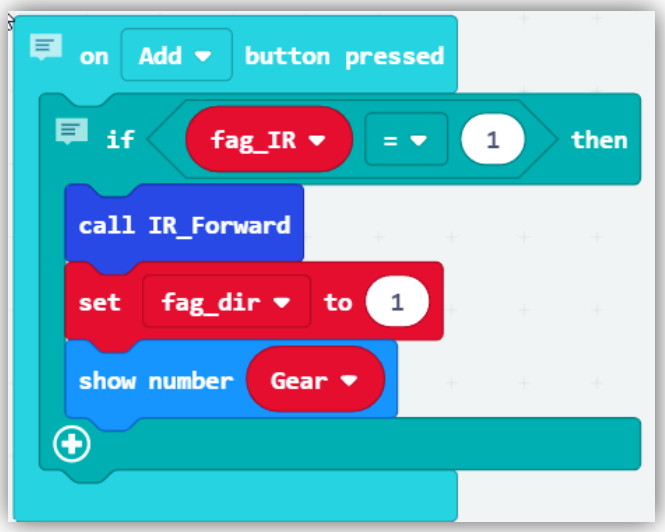


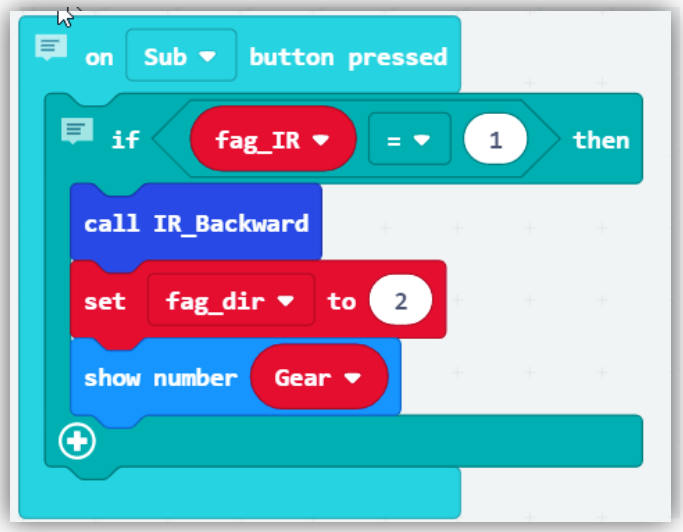




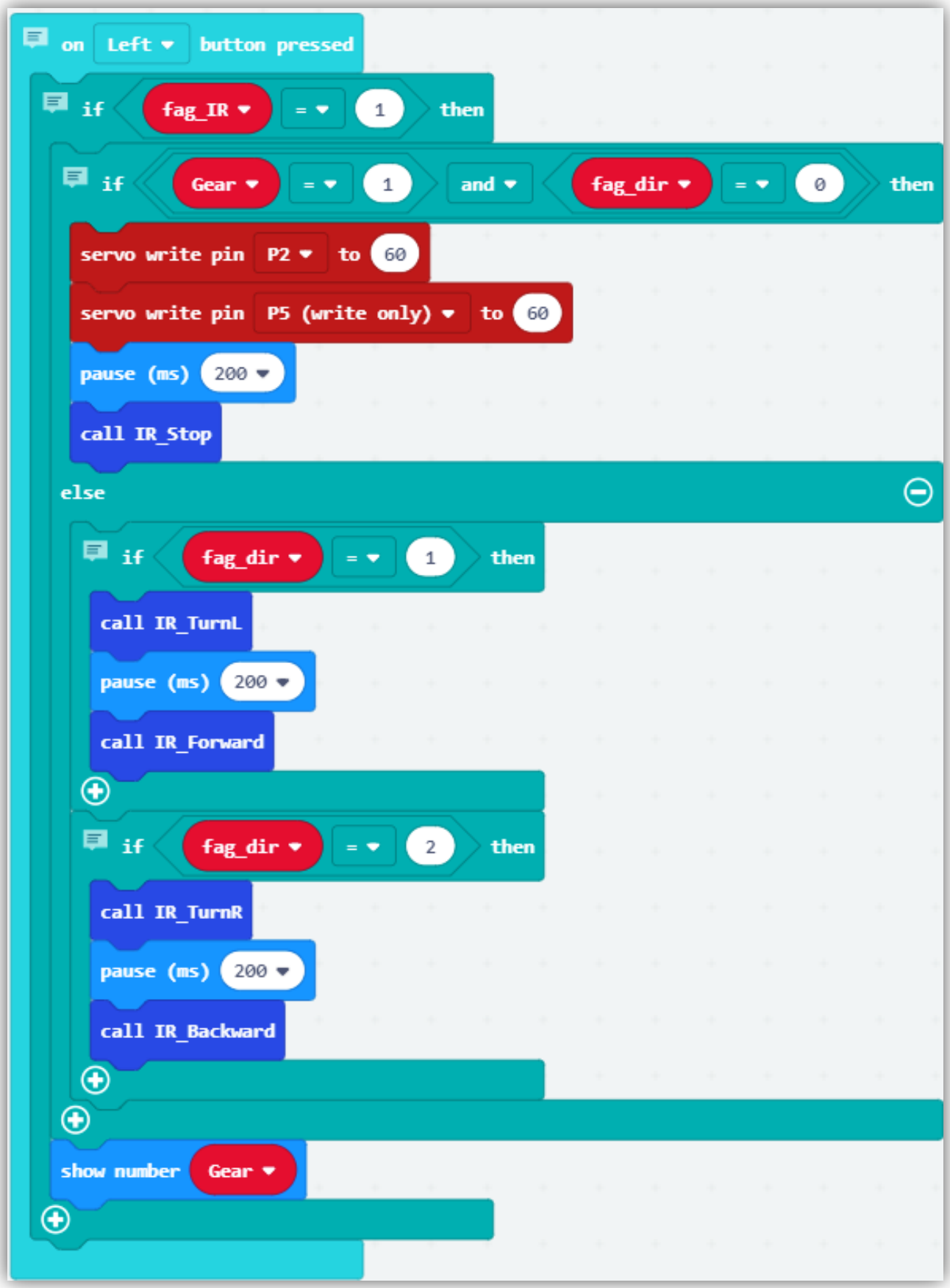


โปรแกรมย่อย-ตอบสนอง IR Key –for Ctr Robot

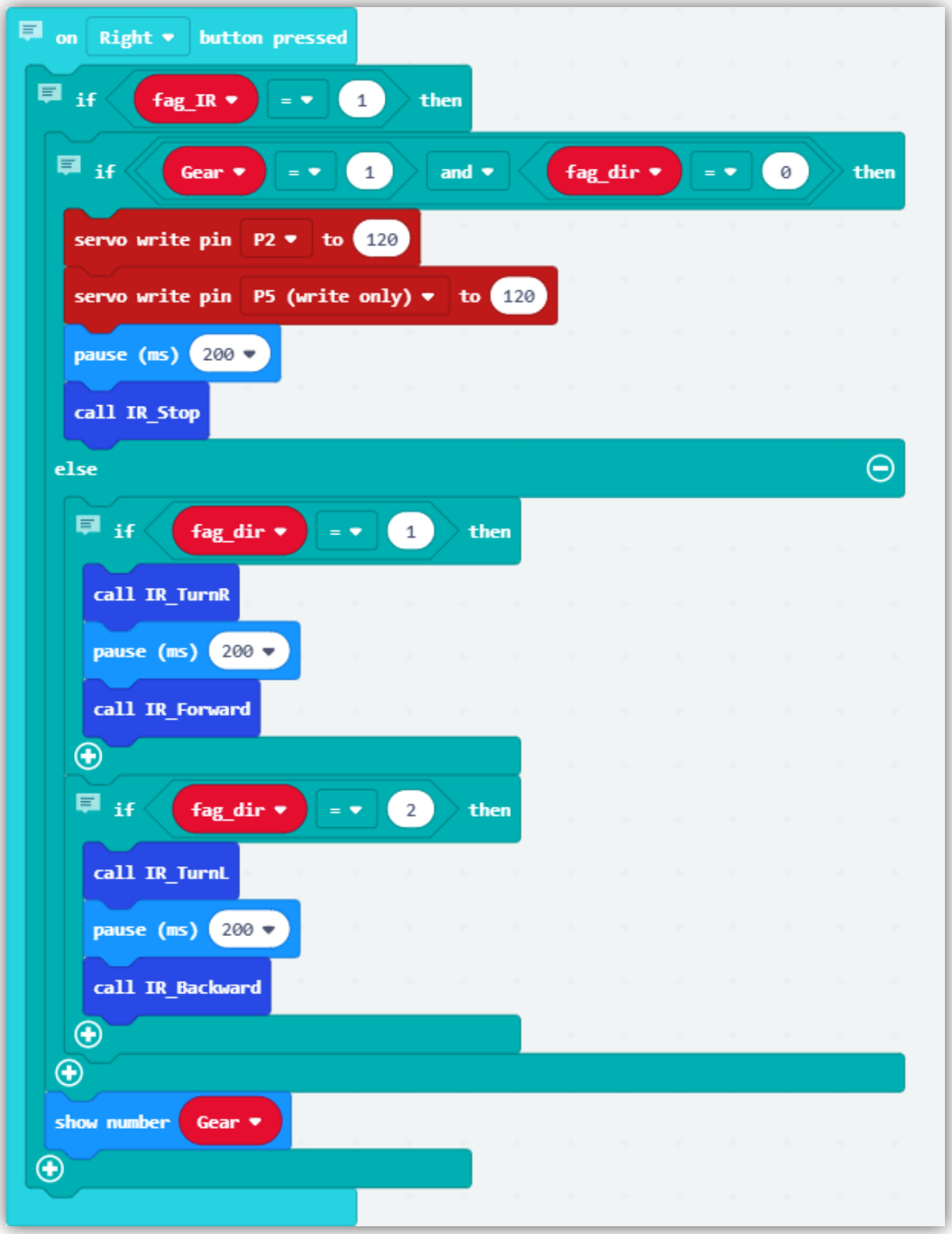




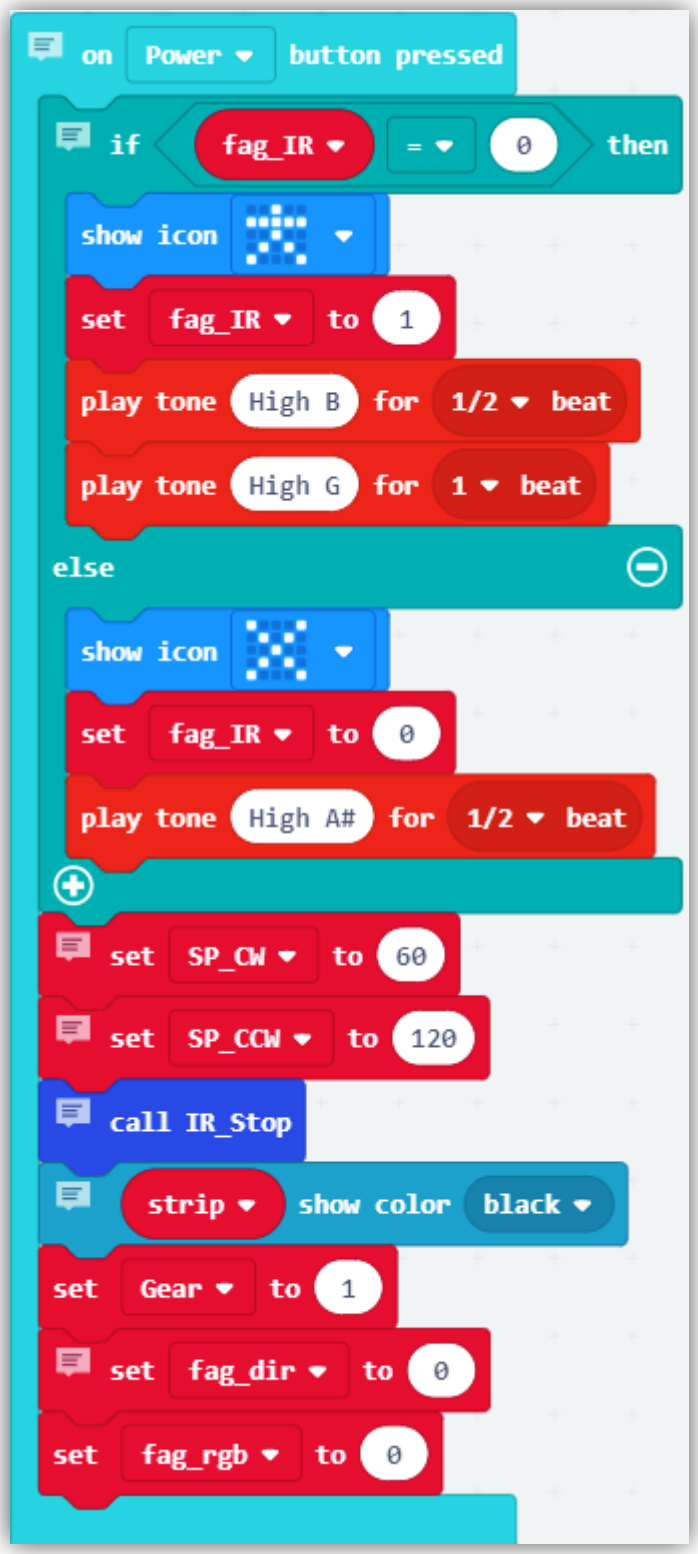
โปรแกรมย่อย-ตอบสนอง IR Key for Ctr Robot (ต่อ)

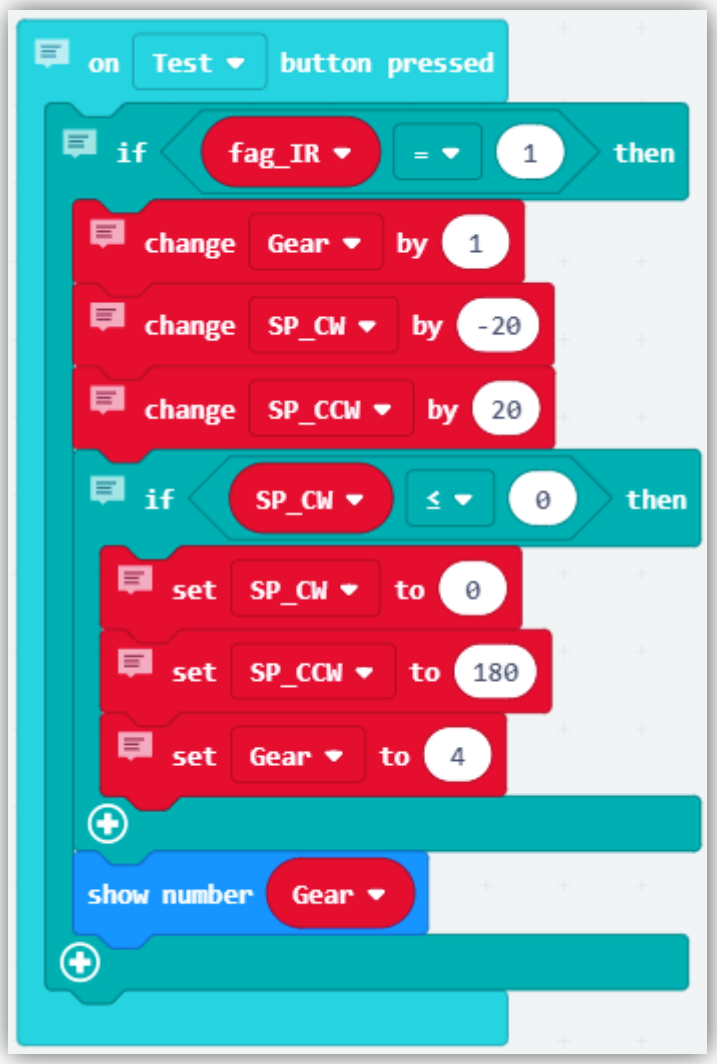


โปรแกรมย่อย-ตอบสนอง IR Key-for Ctr Robot (ต่อ)

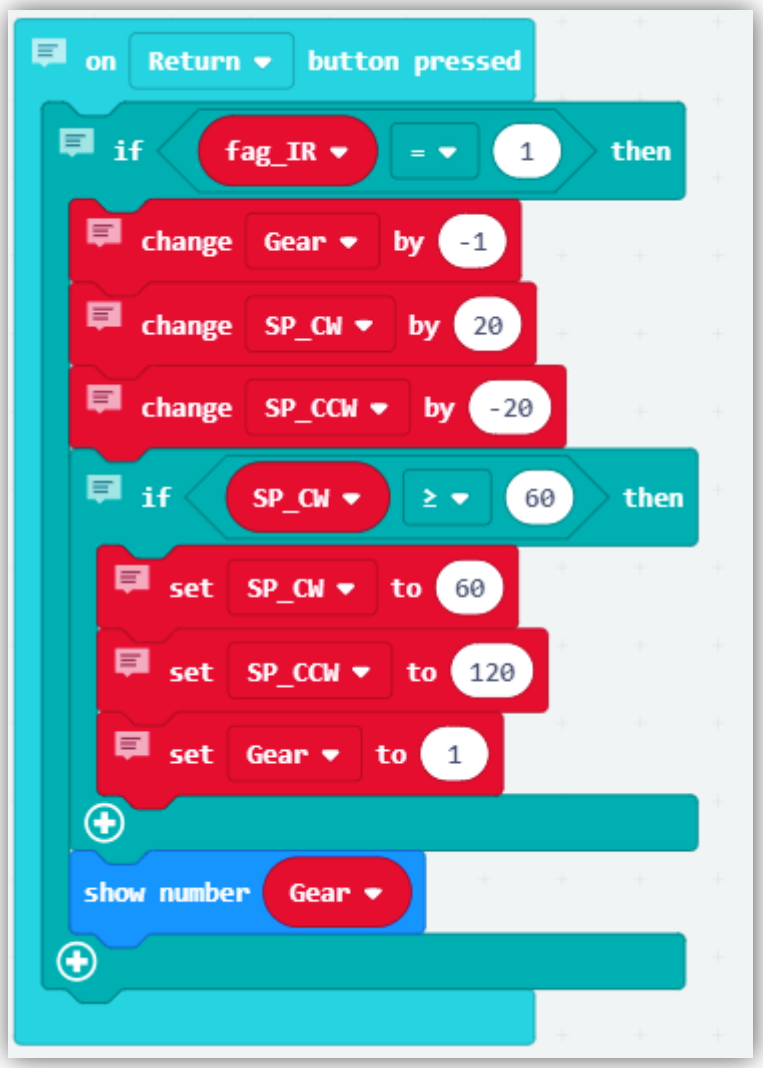


โปรแกรมย่อย-ตอบสนอง IR Key-for Ctr Robot (ต่อ)

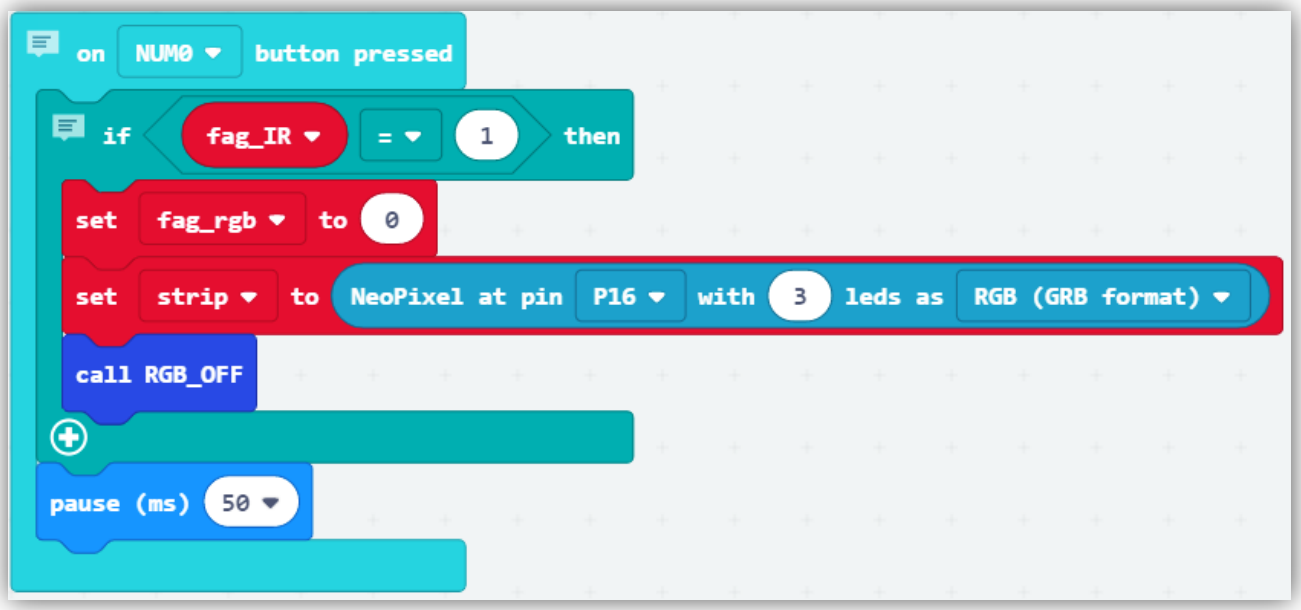




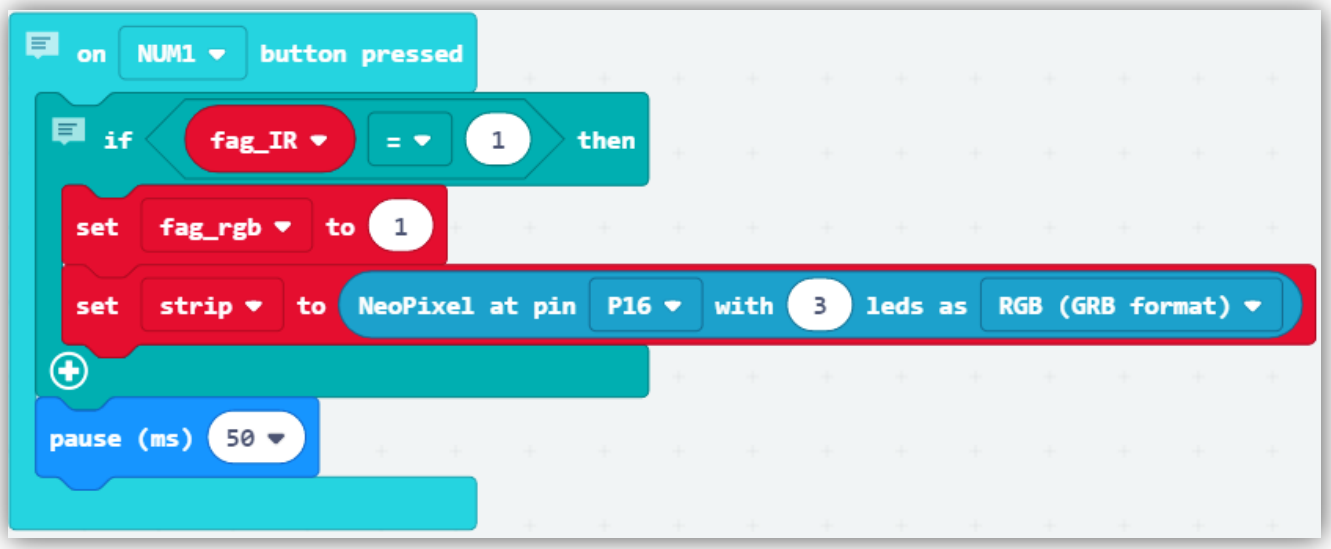
โปรแกรมย่อย-ตอบสนอง IR Key-for Ctr Robot (ต่อ)

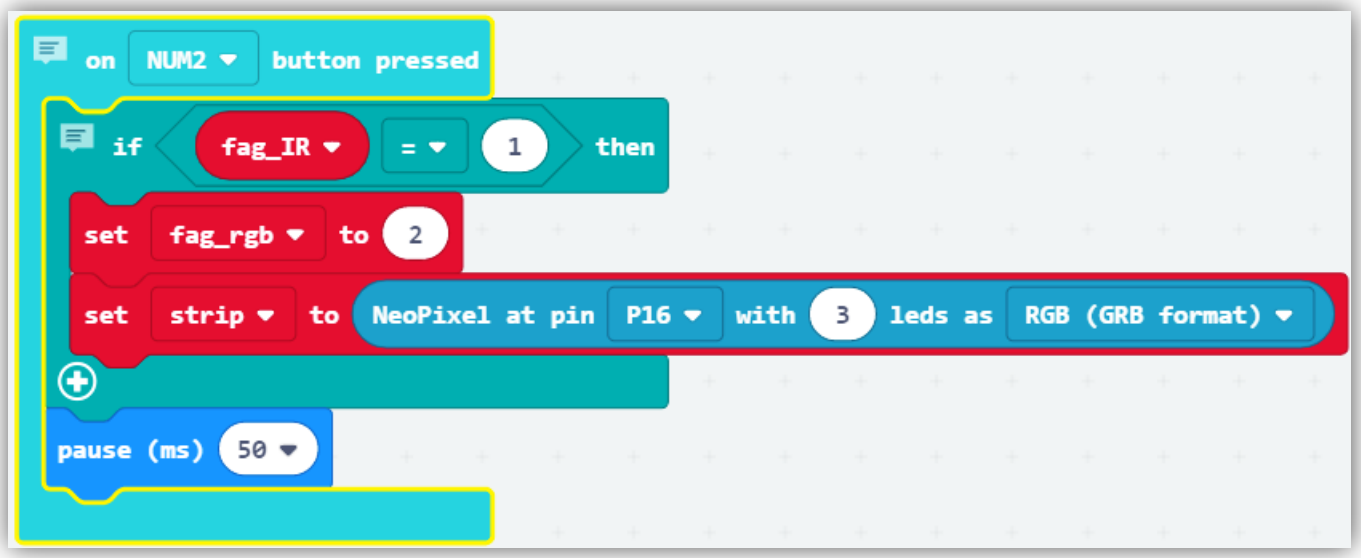


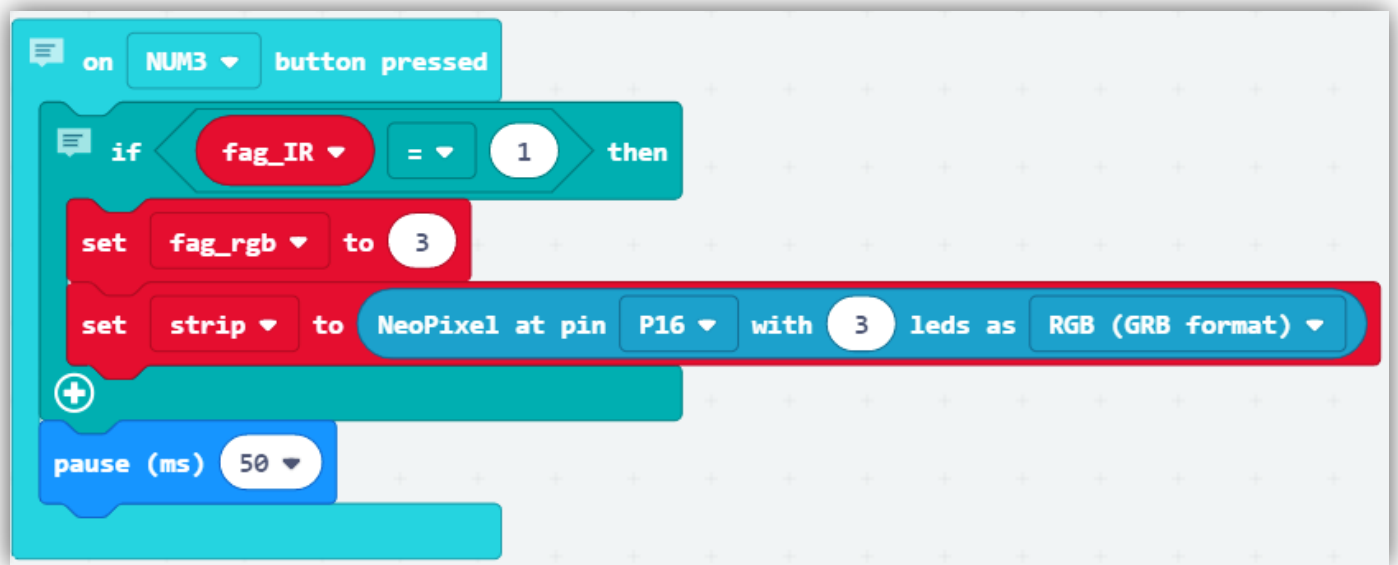
โปรแกรมย่อย-ตอบสนอง IR Key- for Ctr RGB



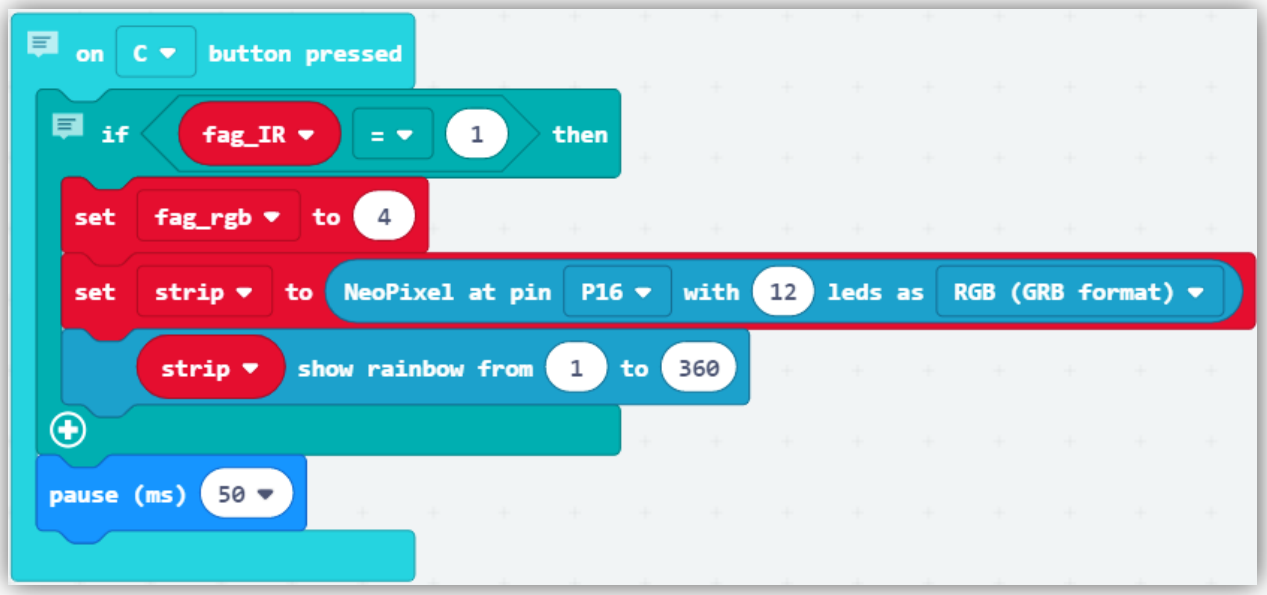
โปรแกรมย่อย-ตอบสนอง IR Key- for Ctr RGB (ต่อ)



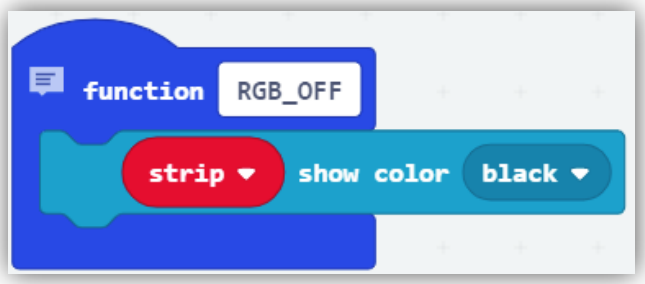


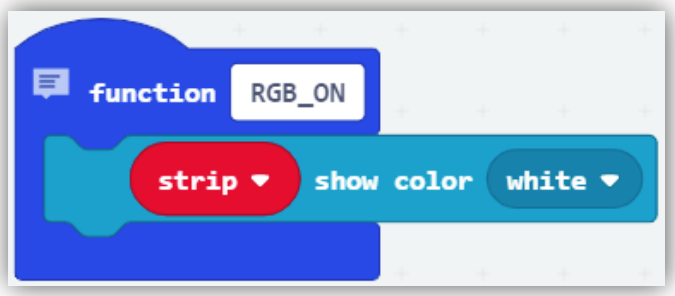


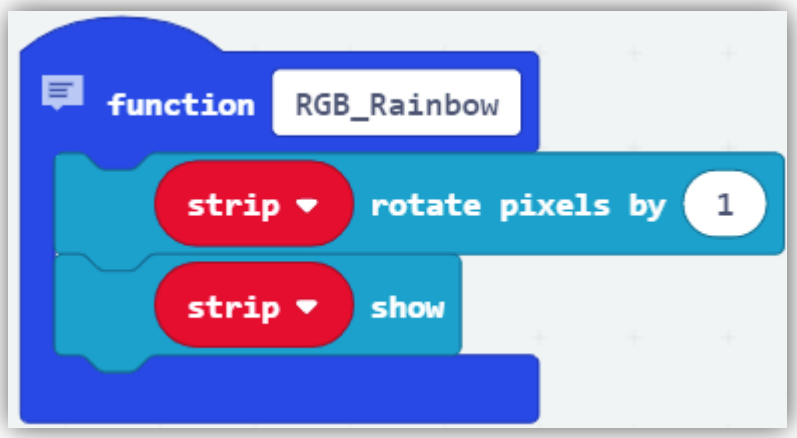
โปรแกรมย่อย-ตอบสนอง IR Key-for Ctr RGB (ต่อ)



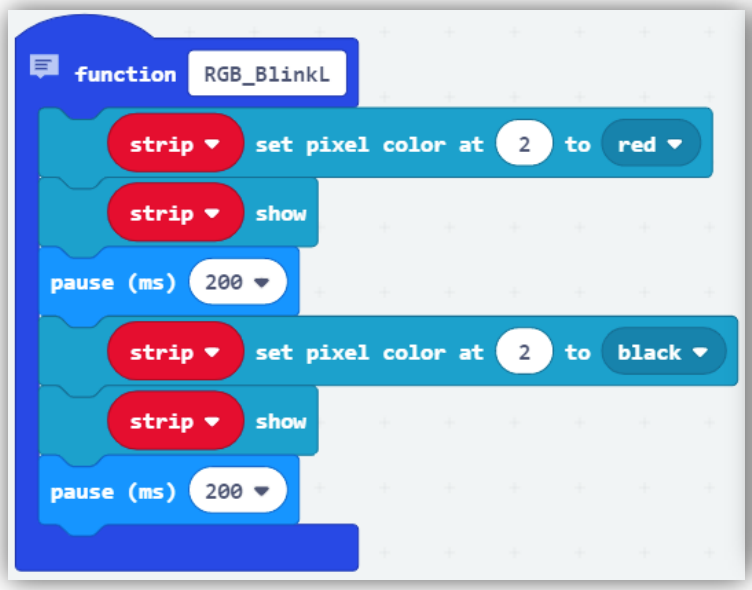
โปรแกรมย่อย-Control RGB

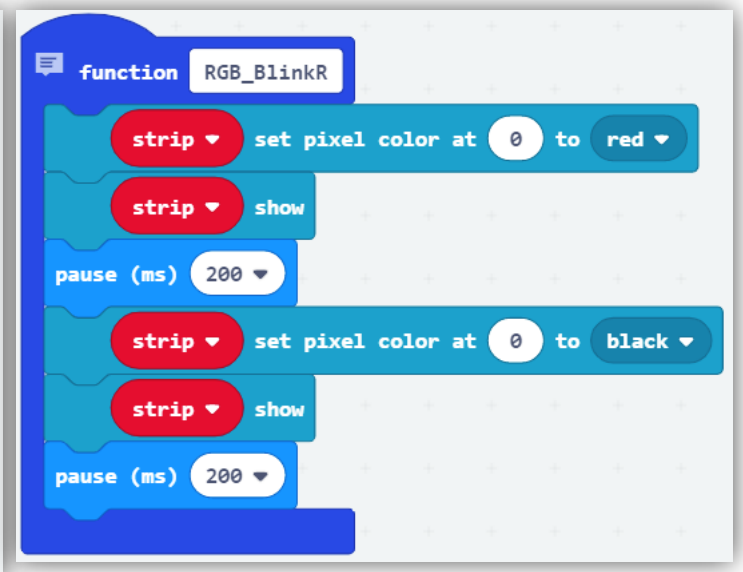




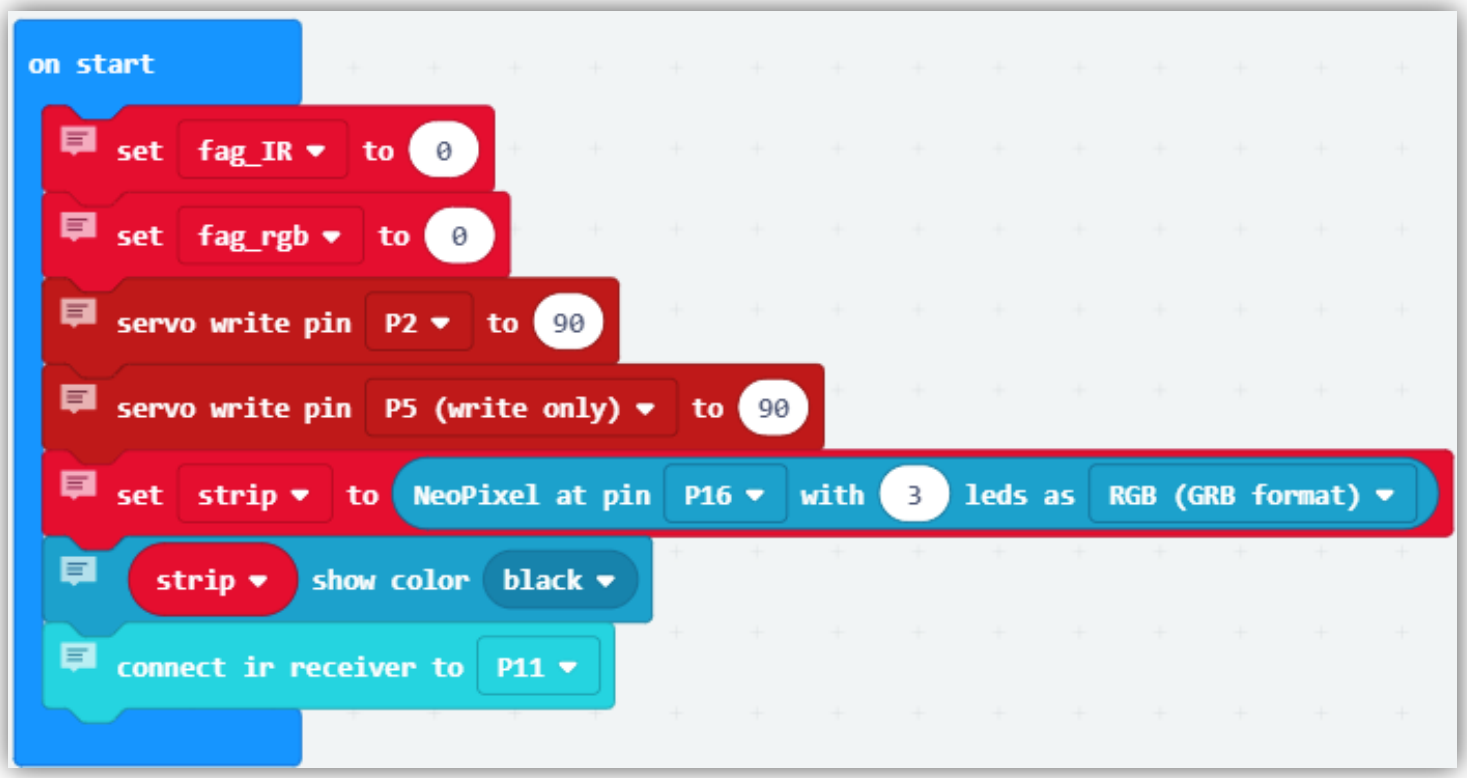


โปรแกรมย่อย-Control RGB (ต่อ)

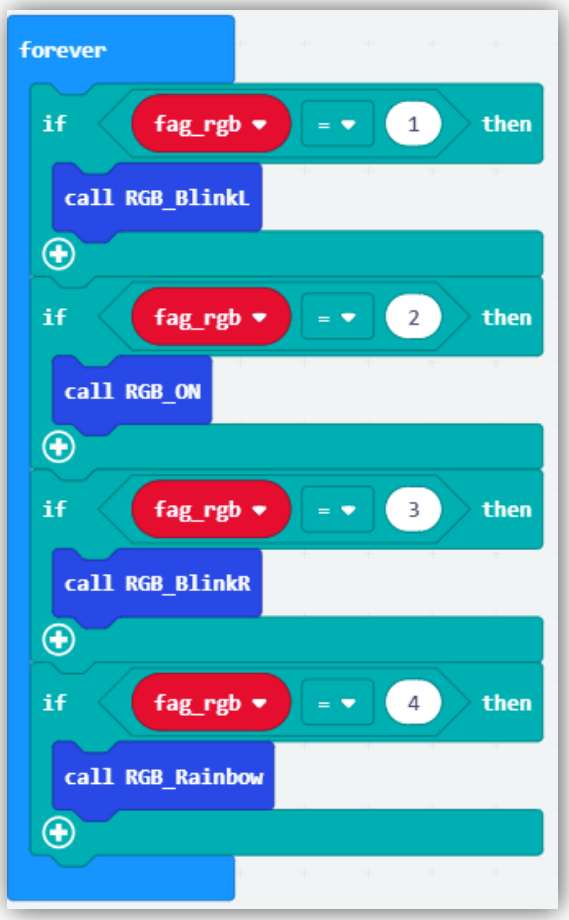




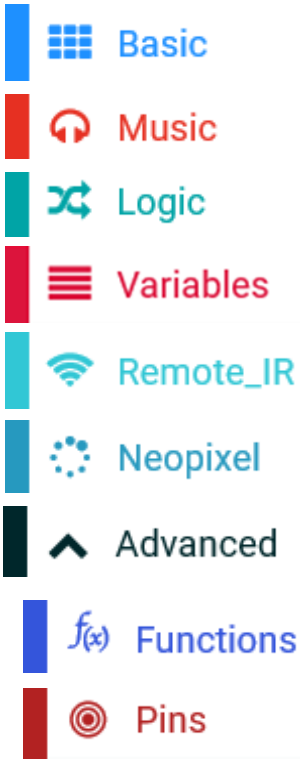
โปรแกรมหลัก

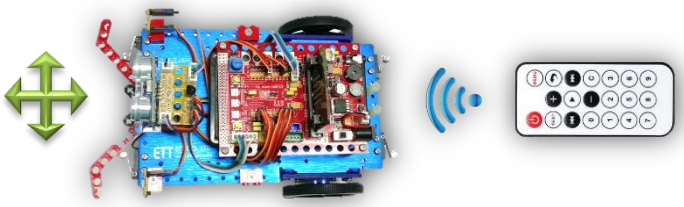


โปรแกรมหลัก (ต่อ)



กลุ่มคำสั่งใช้งาน





การทำงานของโปรแกรม : เมื่อ Power ON

1) ให้กด Key Power ของ Remote เพื่อ Enable Remote Key ทั้งหมด

2) กด Key Test เพิ่ม Gear หรือ Key return เพื่อลด Gear (0-4 เมื่อ Stop:0 , Speed Min(1) ---> Max(4))

3) กด Key ทิศทาง + , - , |<< , >>| , |> เพื่อ Control Robot ให้วิ่งตามคำสั่ง Key โดยที่ Gear ต้องไม่เป็น 0 เมื่อ

+ = Robot เดินหน้า , − = Robot ถอยหลัง , ⏮ = Robot เลี้ยวซ้าย , ⏭ = Robot เลี้ยวขวา , ▶ = Stop

4) กด Key 0, C 1 , 2 , 3 ควบคุม LED-RGB โดย

0 = LED-RGB ดับหมด , C = เล่น RGB Rainbow , 1 = Blink LED Left , 3 = Blink LED Right

2 = RGB ALL ติดสีขาว

7. ตัวอย่าง Demo ET-BBC ROBOT โดยใช้ MakeCode Editor (Folder: Ex_DemoFull)

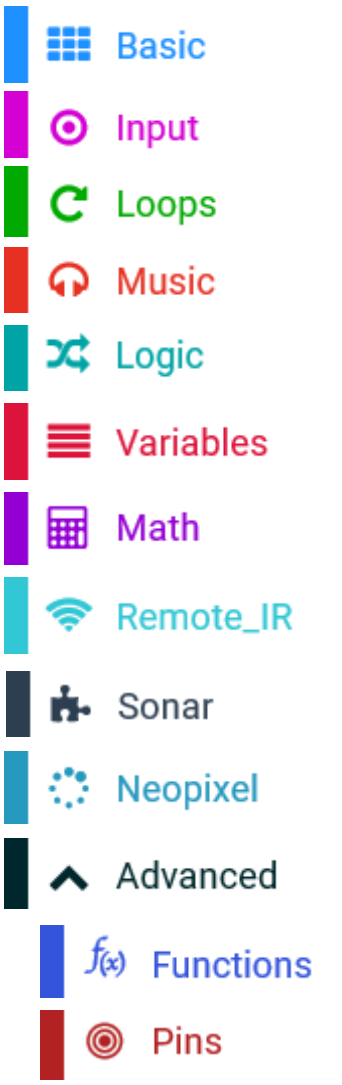
ตัวอย่าง Demo ET-BBC ROBOT จะเป็นตัวอย่าง Control Robot ที่สามารถนำ Robot ไปใช้เล่นในเบื้องต้นได้เลย โดยผู้ใช้ไม่ต้องเสียเวลาในการเขียนโปรแกรม แค่ Download File Demo ลงบอร์ด Micro:bit เท่านั้นแล้วนำไปเสียบเข้ากับ ET-BBC ROBOT ก็ใช้งานได้เลย

การ Control และ Pin ใช้งาน อ้างอิงตามรูปที่ 4.1 หรือตามตัวอย่างทั้งหมดที่กล่าวมาข้างต้น ส่วนการ Set Jumper สรุปดังนี้

JP-V1 = 5V , JP-V2 = 3V3 , JP-ADC = Not Jump , JP-LED = Not Jump , JP-SPK = Jump , JP-RGB = Jump , JP-I2C5V = SDA/SCL =Jump และตัวอย่างโปรแกรม DEMO กับตัวอย่างอื่นๆ สามารถเข้าไปโหลดได้ที่ Link :

http://etteam.com/prodUBit/ET-BBC_ROBOT/index.html

กลุ่มคำสั่งใช้งาน



การทำงานของโปรแกรม : ซึ่งในตัวอย่างนี้จะมีการทำงานอยู่ 3 Mode คือ Mode Setup R-Tracker , Mode Auto เดินตามเส้น และ Mode Manual Control Robot ด้วย IR-Remote เหมือนรถบังคับ มีการใช้งานดังนี้

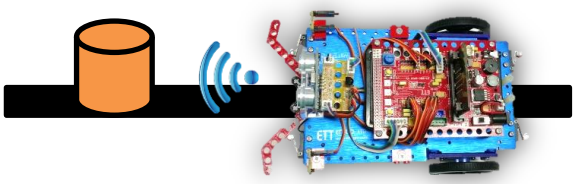
7.1 การ Setup R-Tracker (Sensor ตรวจจับเส้น)

1) Download ตัวอย่าง DEMO ลงบอร์ด Micro:bit ให้เรียบร้อย
2) เสียบบอร์ด Micro:bit เข้ากับบอร์ด ET-BBC ROBTO และ ตรวจสอบสายต่อต่างๆ Jumper และ แบตเตอรี่ให้เรียบร้อย
3) กด SW.B ที่บอร์ด Micro:bit หรือบอร์ด ET-BBC GPIO ค้างไว้
4) เลื่อน SW. Power บนบอร์ด ET-BBC GPIO เพื่อจ่ายไฟเลี้ยงให้ Robot
5) ที่ Matrix-LED ของบอร์ด Micro:bit จะแสดงตัว C และเสียง Beep แสดงว่าเข้าสู่ Mode Setup R-Tracker
6) นำ Robot ไปวางบนเส้นดำและเลื่อนให้ Sensor ตัวใดตัวหนึ่งใน 3ตัวที่อยู่ใต้ Robot ให้อยู่บนเส้นดำ(อ้างอิงตามตัวอย่าง Ex9) จากนั้นให้สังเกตุ RGB-LED บนบอร์ด ET-BBC GPIO หรือที่ Matrix-LED บนบอร์ด Micro:bit ในตำแหน่งที่ตรงกับ Sensor ที่อยู่บนเส้นดำว่าติดหรือไม่ (RGB3=Sensor-L ,RGB2=Sensor-M ,RGB1=Sensor-R) ถ้าไม่ติดให้ปรับ VR ที่บอร์ด R-Tracker จนเห็น RGB-LED หรือ Matrix-LED ในตำแหน่งที่ตรงกับ Sensor ที่อยู่บนเส้นดำติด (Output Sensor=1)
7) ลองเลื่อน Sensor ตำแหน่งอื่นให้ทับเส้นดำดูบ้าง LED ในตำแหน่งที่ตรงกับ Sensor นั้นๆจะต้องติด (LED จะต้องติดตรงกับ Sensor ที่ทับเส้นดำ เช่น ถ้า Sensor ทับเส้นดำ 2 ตัว LED ก็จะต้องติด 2 ตัว ในตำแหน่งที่ตรงกับ Sensor เป็นต้น)
8) กด Sw.B อีกครั้ง เพื่อออกจาก Mode Setup R-Tracker เมื่อออกแล้ว Matrix-LED จะแสดงตัว X เพื่อรอรับการควบคุมจากผู้ใช้

7.2 การใช้งาน Mode Auto เดินตามเส้น

1) Download ตัวอย่าง DEMO ลงบอร์ด Micro:bit ให้เรียบร้อย
2) เสียบบอร์ด Micro:bit เข้ากับบอร์ด ET-BBC ROBTO และ ตรวจสอบสายต่อต่างๆ Jumper และ แบตเตอรี่ให้เรียบร้อย
3) เลื่อน SW. Power บนบอร์ด ET-BBC GPIO เพื่อจ่ายไฟเลี้ยงให้ Robot (Power ON)
4) กด SW.B เพื่อ Start Robot เข้า Mode เดินตามเส้น โดยมีเงื่อนไขตรวจสอบเส้นดังนี้
   -ขณะ Robot เดินตามเส้น ถ้าเจอแยกขวา (Sensor-L=0,Sensor-M=1,Sensor-R=1) ให้ Robot เลี้ยวขวา
   -ขณะ Robot เดินตามเส้น ถ้าเจอแยกซ้าย (Sensor-L=1,Sensor-M=1,Sensor-R=0) ให้ Robot เลี้ยวซ้าย
   -ขณะ Robot เดินตามเส้น ถ้าเจอ3แยก (Sensor-L=1,Sensor-M=1,Sensor-R=1) ให้ Robot เลี้ยวซ้าย
   -ขณะ Robot เดินตามเส้นถ้า Micro.SW หน้า ตัวใดตัวหนึ่งหรือทั้ง 2ตัว ถูกกด หรือ Ultasonic ตรวจจับวัตถุด้านหน้า ได้ว่าอยู่ในระยะที่กำหนดไว้(5-10 cm) ให้ Robot หมุนตัวกลับ 180 องศา และวิ่งไปข้างหน้าต่อ
   -ขณะ Robot เดินตามเส้นถ้า Micro.SW หลังขวา ถูกกด Robot Stop และ Turn-L 180 และวิ่งไปข้างหน้าต่อ
   -ขณะ Robot เดินตามเส้นถ้า Micro.SW หลังซ้าย ถูกกด Robot Stop และ Turn-R 180 และวิ่งไปข้างหน้าต่อ
5) ถ้ามีการกด SW.B ขณะ Robot วิ่งตามเส้นอยู่ เป็นการ Stop Robot

หมายเหตุ: ถ้าเส้นทางที่ใช้วิ่งมีแยกหรือเป็นมุมฉาก การตัดสินใจทำคำสั่งของ Robot อาจไม่เป็นไปตามที่กล่าวข้างต้นทุกครั้งที่พบแยก ขึ้นอยู่กับจังหวะที่ Robot เข้าแยกว่า Sensor อยู่ในตำแหน่งตรงกับเงื่อนไขที่กำหนดหรือไม่ถ้าไม่ตรงอาจวิ่งผ่านแยกไป



7.3 การใช้งาน Mode Manual บังคับโดยใช้ ET-Remote Key(IR)

1) Download ตัวอย่าง DEMO ลงบอร์ด Micro:bit ให้เรียบร้อย

2) เสียบบอร์ด Micro:bit เข้ากับบอร์ด ET-BBC ROBTO และ ตรวจสอบสายต่อต่างๆ Jumper และ แบตเตอรี่ให้เรียบร้อย

3) เลื่อน SW. Power บนบอร์ด ET-BBC GPIO เพื่อจ่ายไฟเลี้ยงให้ Robot (Power ON)

4) กด Key Power ของ Remote เพื่อ Enable Key Remote ทั้งหมด

5) กด Key Test เพิ่ม Gear หรือ Key return เพื่อลด Gear (0-4 เมื่อ Stop:0 , Speed Min(1) ---> Max(4))

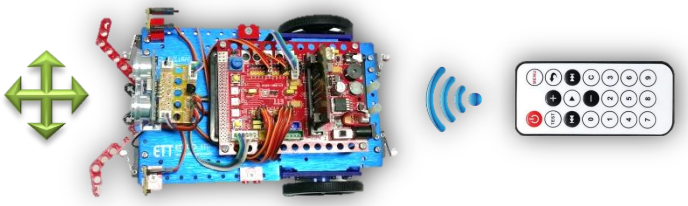
6) กด Key ทิศทาง + , - , |<< , >>| , |> เพื่อ Control Robot ให้วิ่งตามคำสั่ง Key โดยที่ Gear ต้องไม่เป็น 0 เมื่อ

= Robot เดินหน้า , = Robot ถอยหลัง , = Robot เลี้ยวซ้าย , = Robot เลี้ยวขวา , = Stop

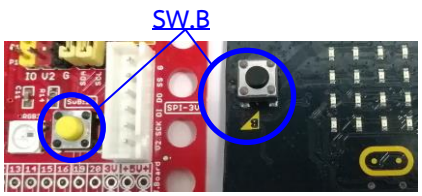
7) กด Key 0, C 1 , 2 , 3 ควบคุม LED-RGB โดย

0 = LED-RGB ดับหมด , C = เล่น RGB Rainbow , 1 = Blink LED Left , 2 = RGB ALL ติดสีขาว

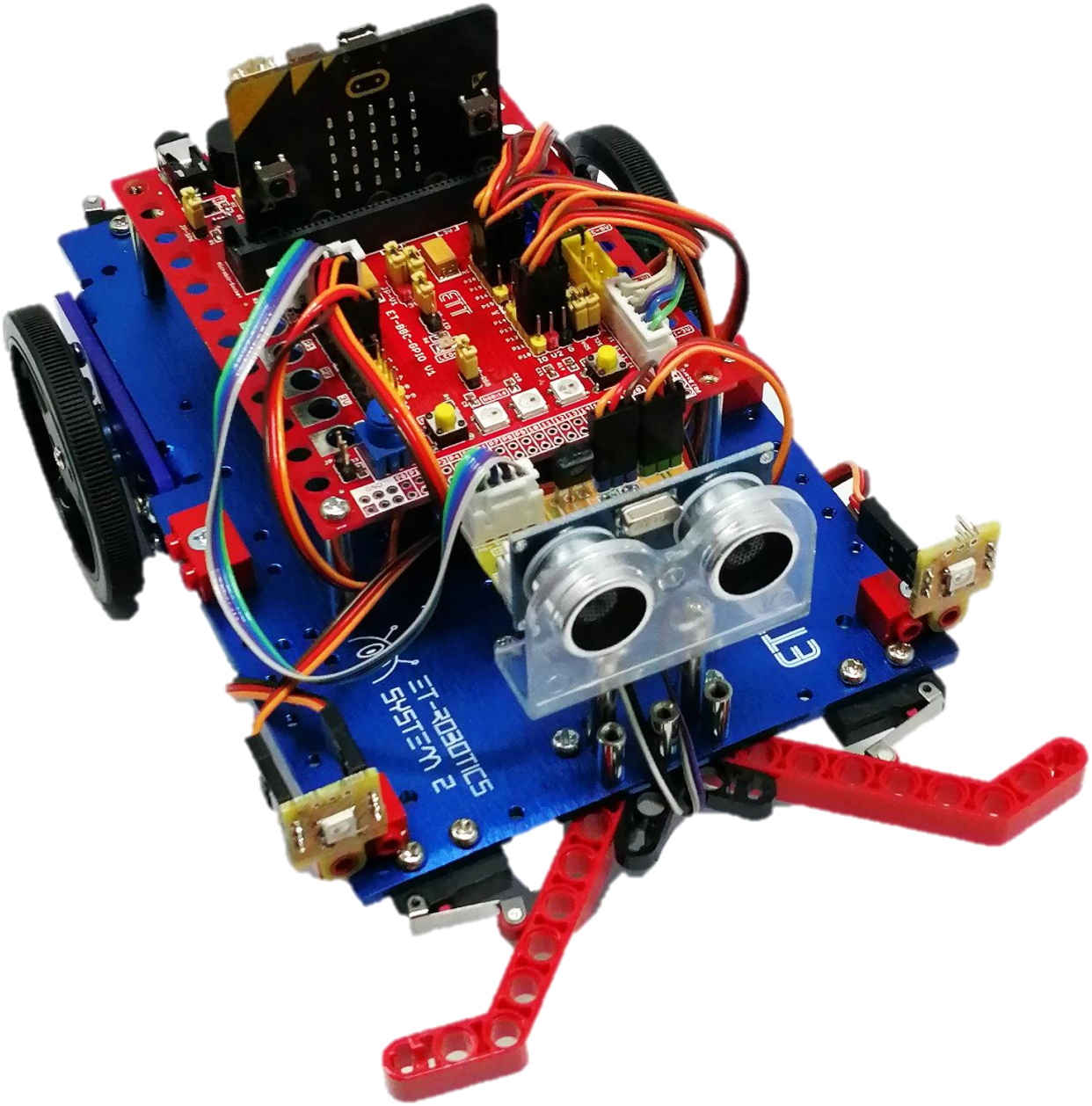
3 = Blink LED Right



ระหว่างที่อยู่ในโหมด Auto สามารถกด Remote Key Power เพื่อเปลี่ยนโหมดมาเล่นในโหมด Manual ได้ หรือ ระหว่างที่อยู่ในโหมด Manual สามารถกด SW.B เพื่อเปลี่ยนโหมดมาเล่นในโหมด Auto เดินตามเส้น ได้ เช่นกัน

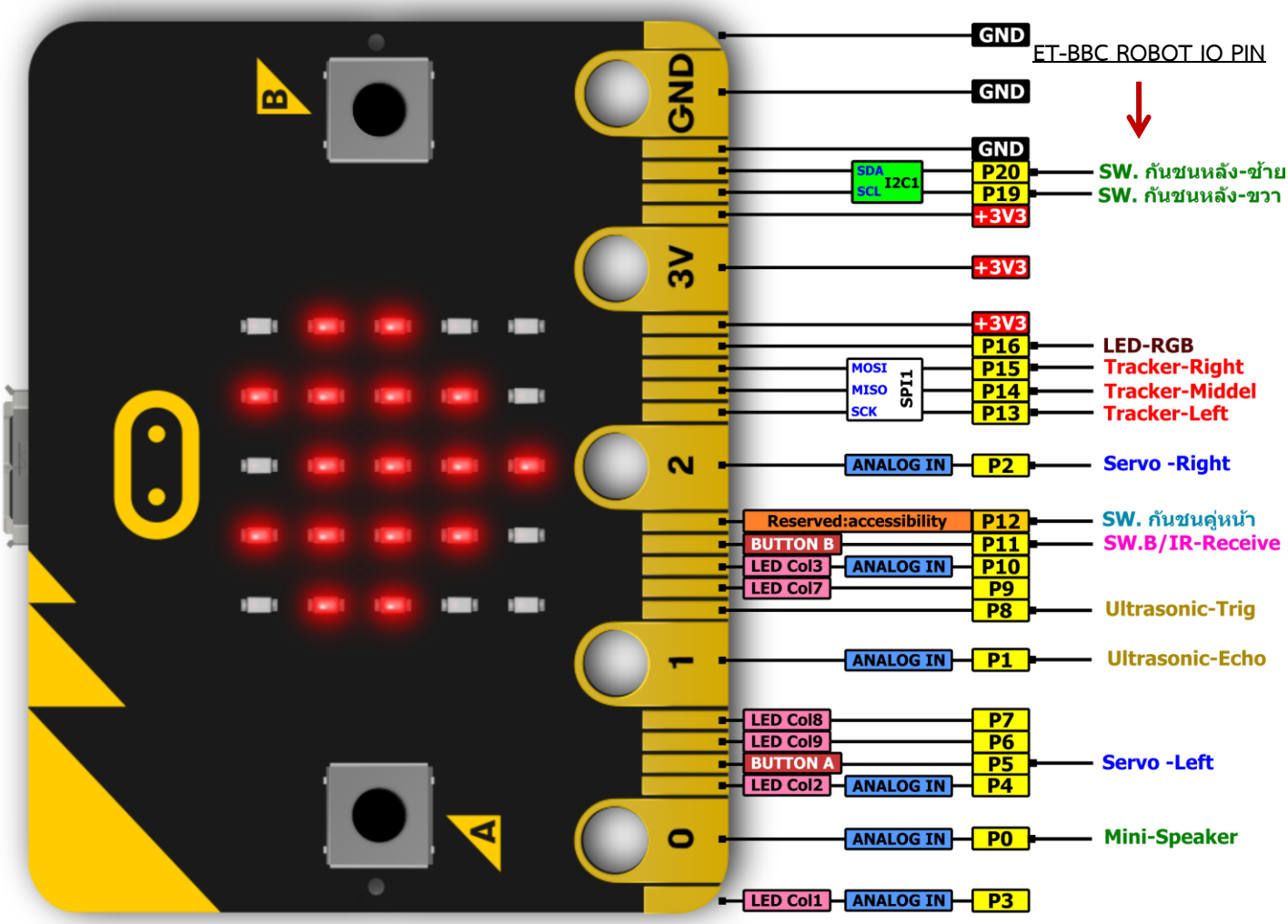


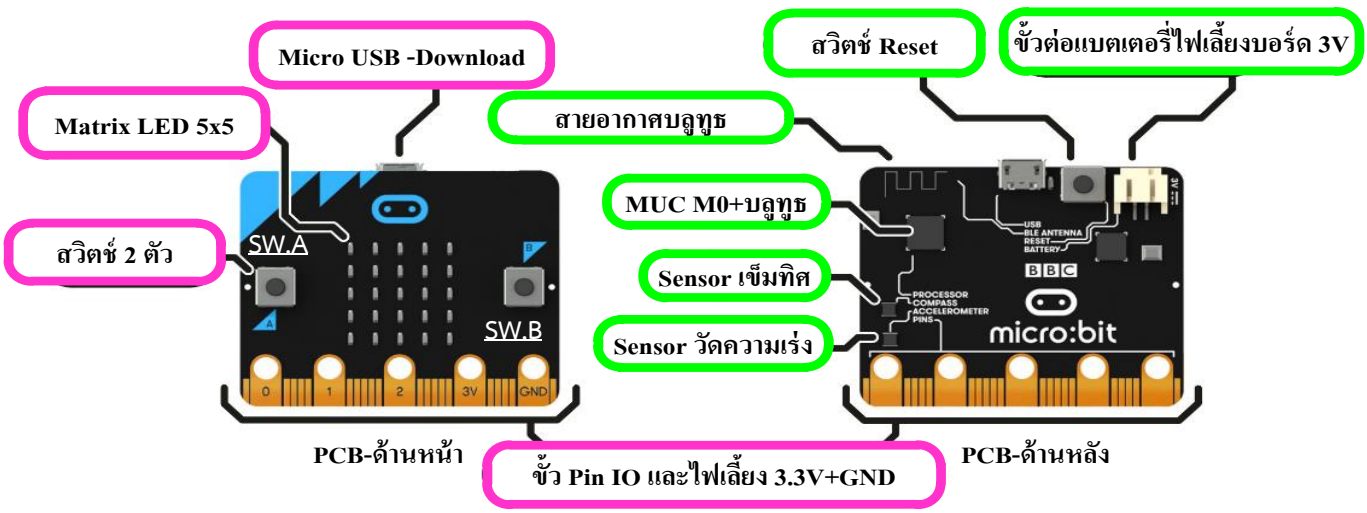
รูปแสดงตำแหน่ง SW.B บนบอร์ด Micro:bit และ ET-BBC GPIO



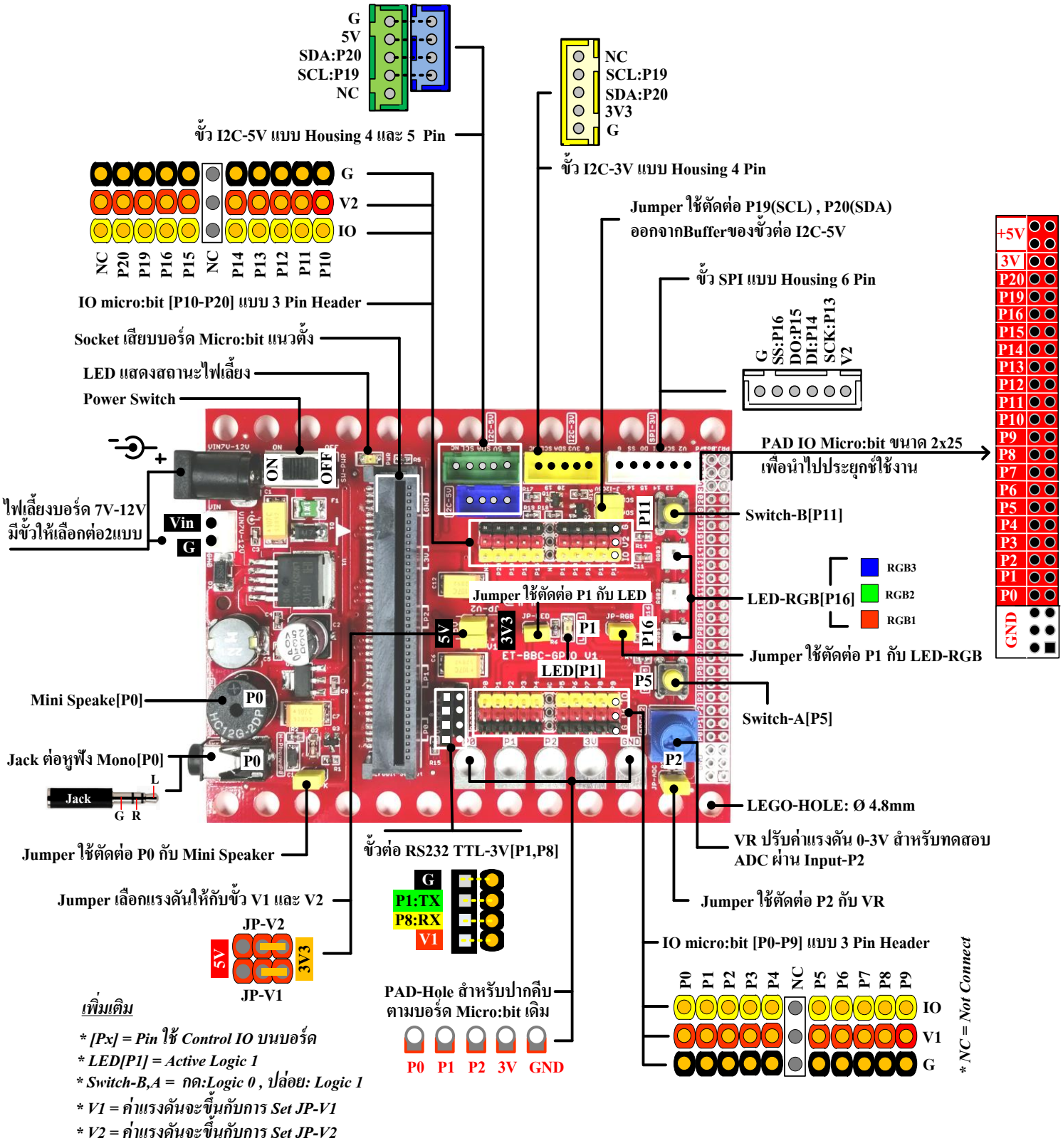
ภาคผนวก

บอร์ด Micro:bit และ IO PIN



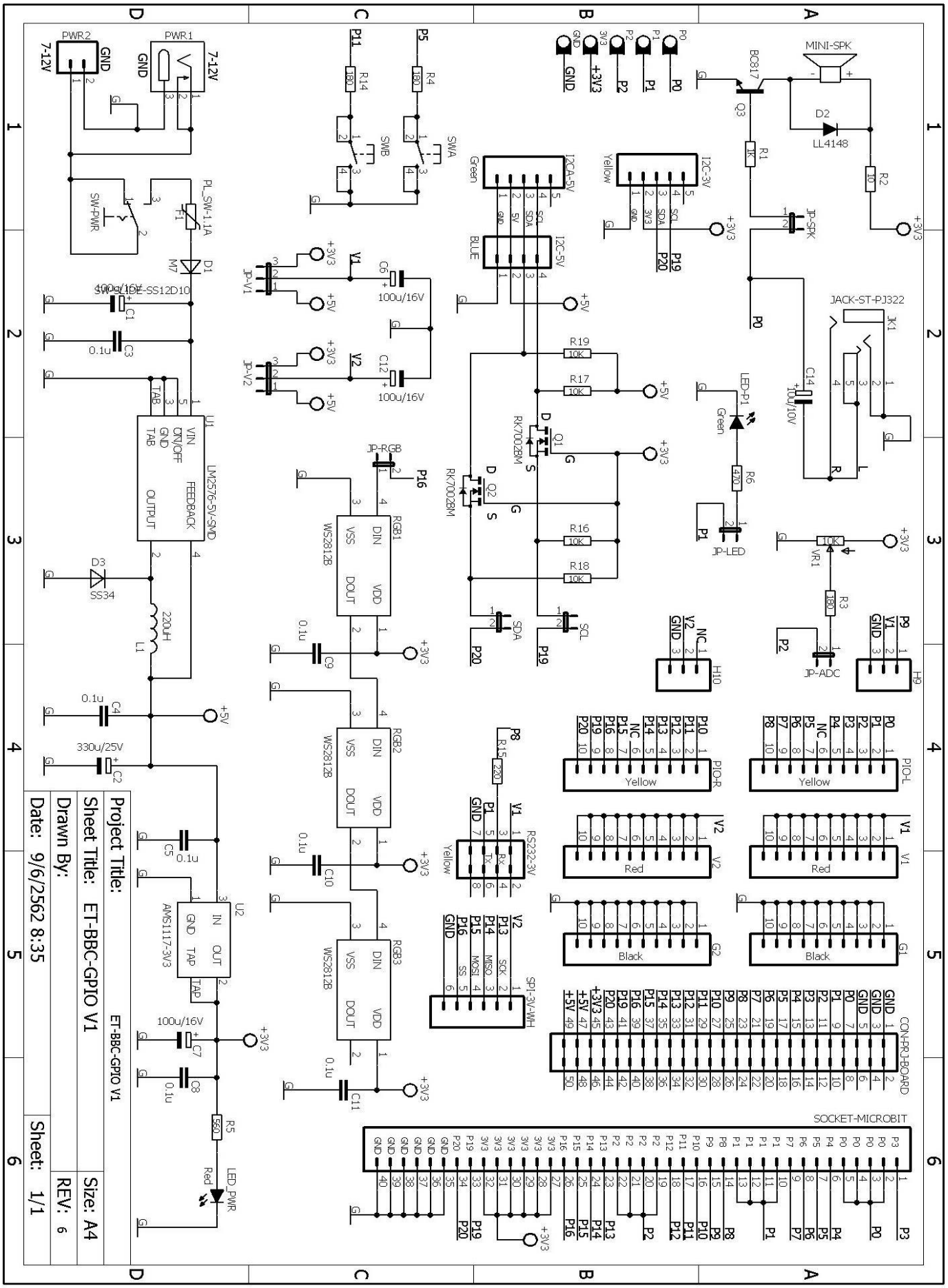


บอร์ด ET-BBC GPIO V1



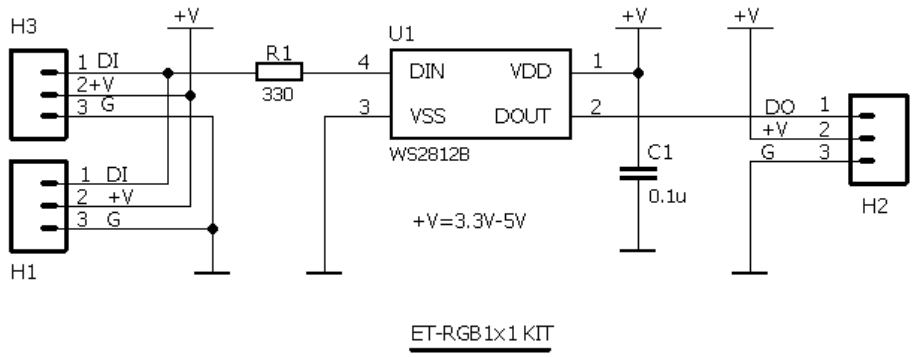
เพิ่มเติม

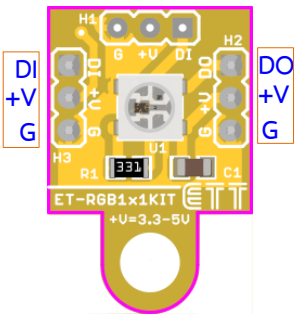
* *[Px] = Pin ใช้ Control IO บนบอร์ด*
* *LED[P1] = Active Logic 1*
* *Switch-B,A = กด:Logic 0 , ปล่อย: Logic 1*
* *V1 = ค่าแรงดันจะขึ้นกับการ Set JP-V1*
* *V2 = ค่าแรงดันจะขึ้นกับการ Set JP-V2*



วงจร ET-BBC GPIO V1

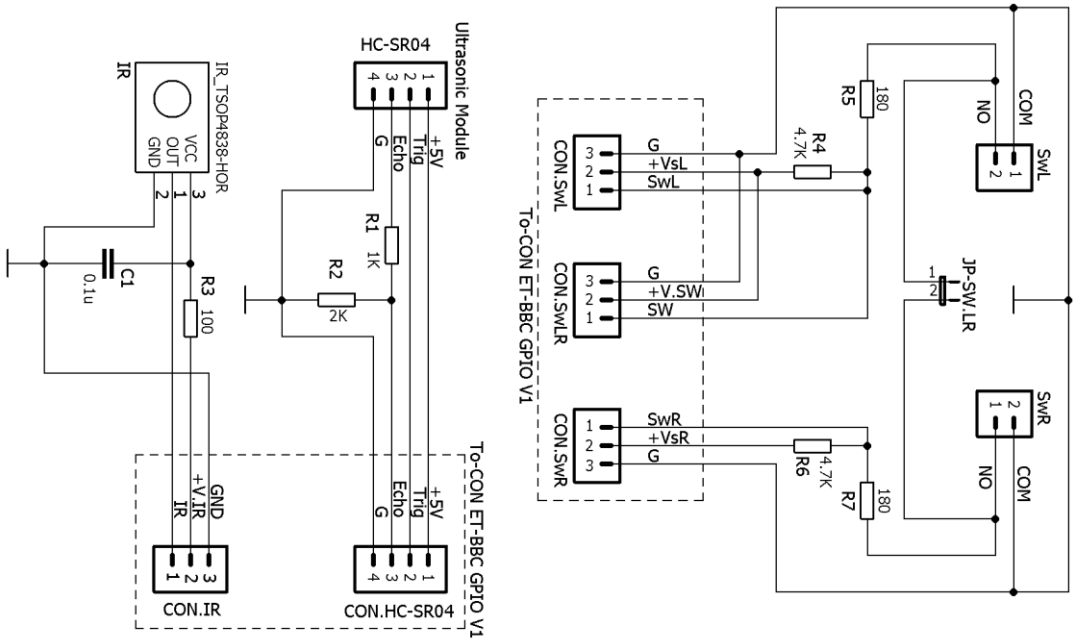
Module และ วงจร ET-RGB1x1 KIT

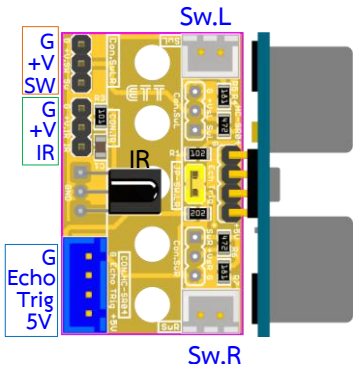




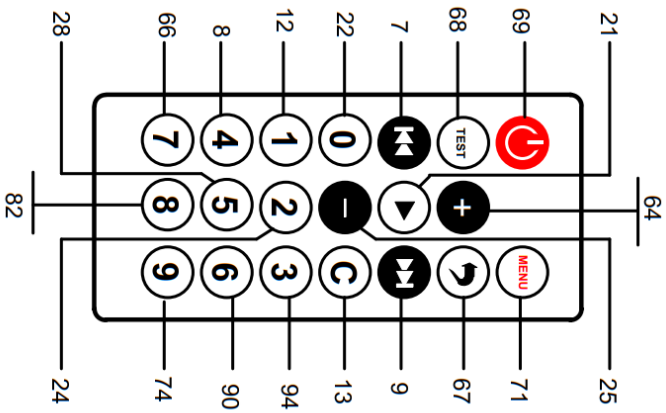
ET-RGB1x1 KIT

Module และ วงจร IR+Micro-SW.+Ultrasonic Logic Level



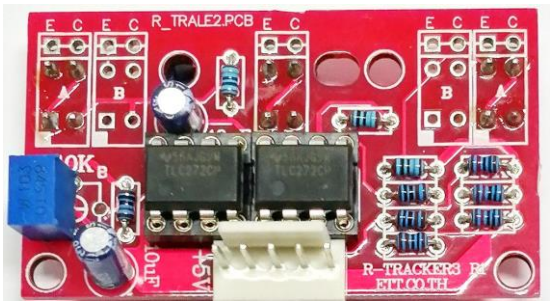


Key Code ET-IR Remote Key (Dec 1 Byte)

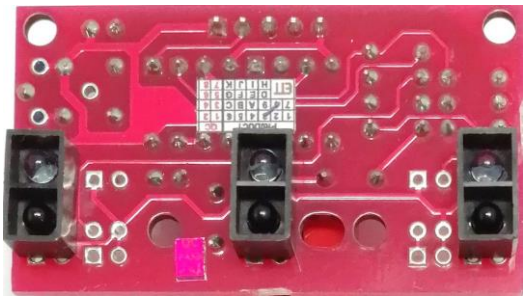


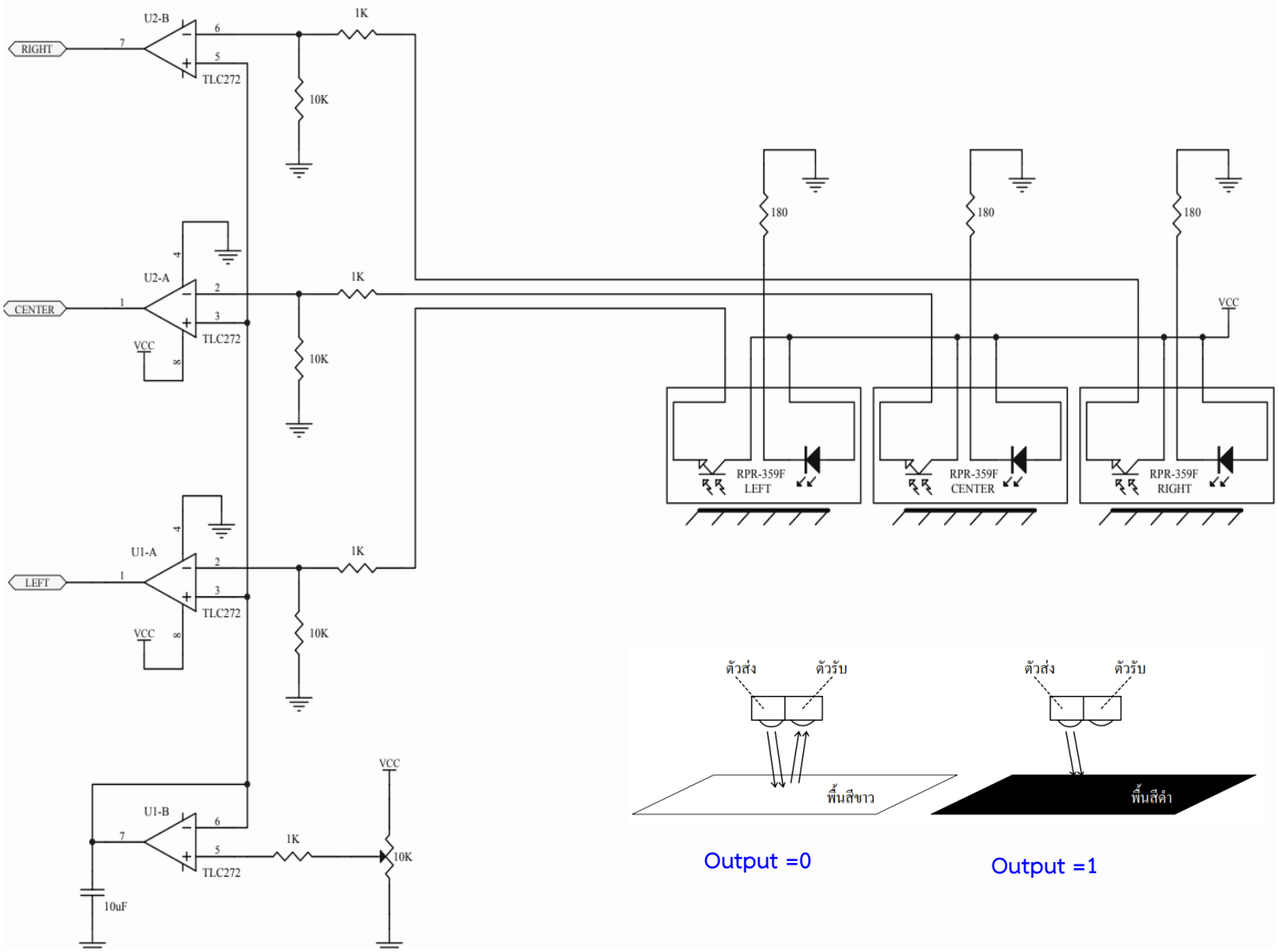
Module และ วงจร R-Tracker

ด้านบน

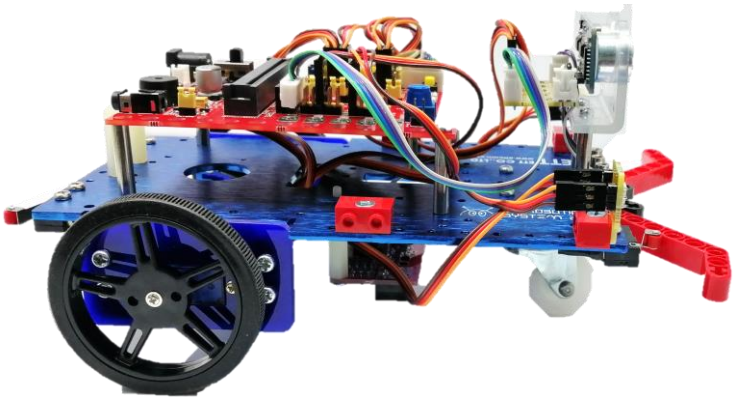
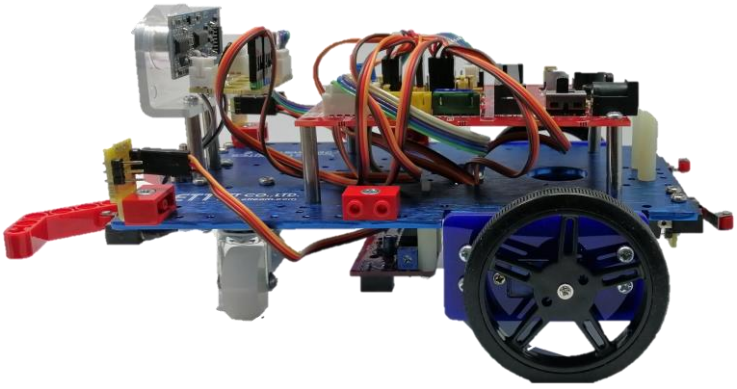
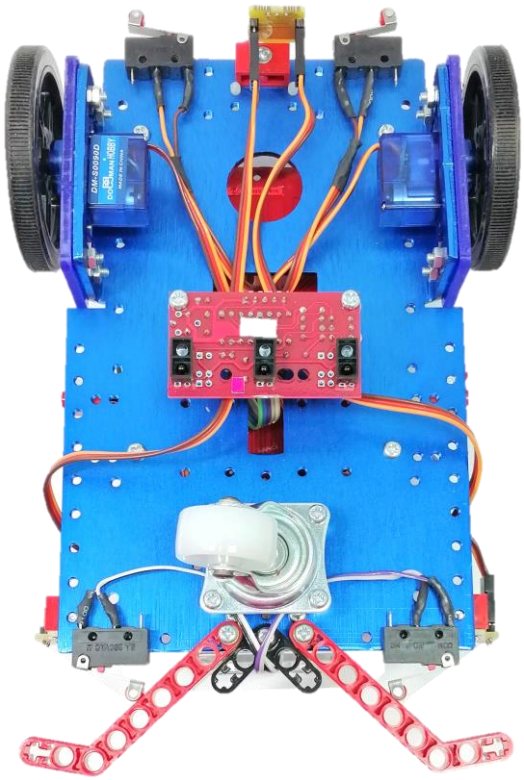
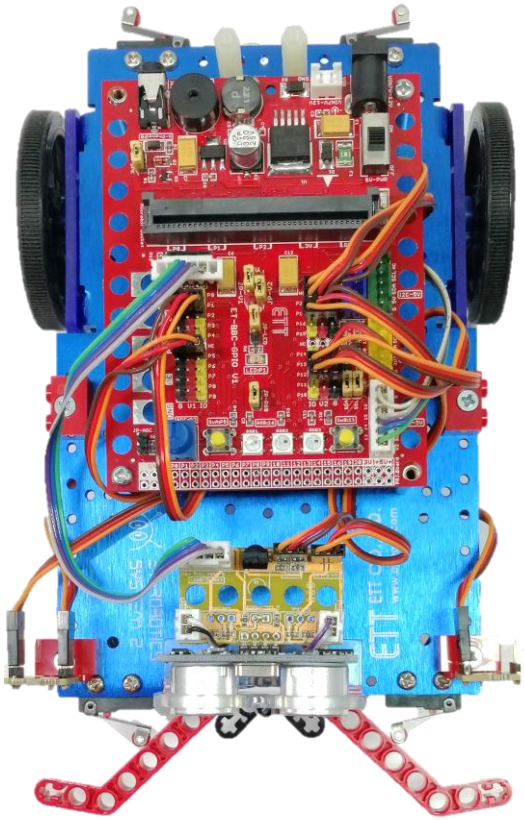
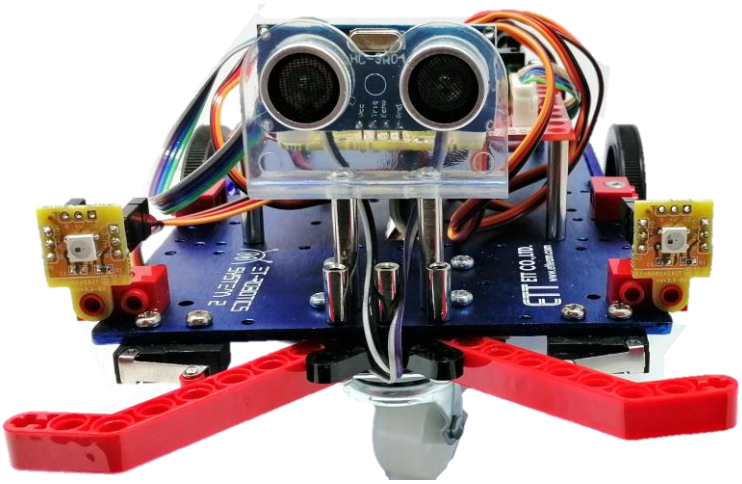
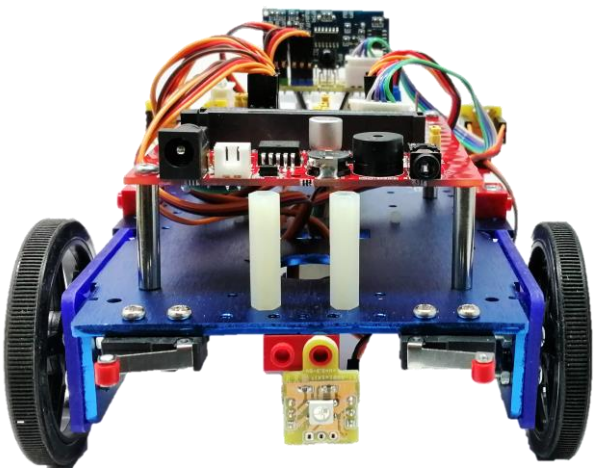


ด้านล่าง





Output =0

Output =1

มุมมอง ET-BBC ROBOT