อัพเดทล่าสุด 16 มกราคม 2565

ARTRONSHOP
www.artronshop.co.th

คู่มือการใช้งาน Rapbit32XA เบื้องต้น

หุ่นยนต์พร้อมแขนจับ เขียนโปรแกรมไพทอน

จัดทำโดย บริษัท อาร์ทรอน ชอป จำกัด

แจกฟรี ห้ามจำหน่าย

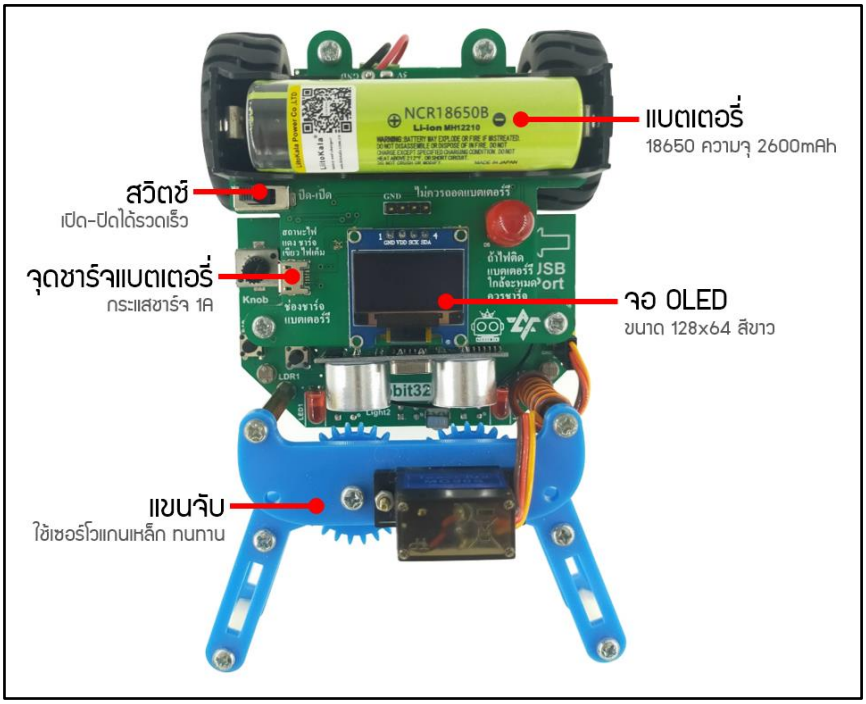
สารบัญ

Rapbit32XA หุ่นยนต์พร้อมแขนจับ เขียนโปรแกรมไพทอน เหมาะสำหรับนำไปจัดการเรียนรู้แบบสะเต็มศึกษา พัฒนาแนวคิดเชิงตรรกะไปพร้อม ๆ กับการเรียนรู้แบบสนุกสนาน ด้วยหุ่นยนต์ที่มีล้อเขียนโปรแกรมควบคุมได้อิสระให้เดินหน้า-ถอยหลัง เลี้ยวซ้าย เลี้ยวขวา หรือหมุนตัวได้อิสระ มาพร้อมเซ็นเซอร์ตรวจจับเส้น 3 ตัว, แขนจับ (Gripper), เซ็นเซอร์วัดระยะวัตถุด้วยคลื่นอัลตร้าโซนิค, เซ็นเซอร์แสงซ้าย-ขวา (2 ตัว), สวิตช์กดติด-ปล่อยดับโปรแกรมได้, ตัวต้านทานปรับค่าได้, ไฟหน้าซ้าย-ขวา (2 ตัว) ตัวรับรีโมทแบบ IR และหลอดแอลอีดีแบบ RGB 4 ดวง ใช้หน่วยประมวลหลัก ESP32 ECO V3 รองรับการเชื่อมต่อ WiFi และบลูทูธ แบตเตอรี่ 18650 ความจุ 2600mAh ชาร์จแบตด้วยการเสียบสาย MicroUSB หน้าจอ OLED ขนาด 0.96 นิ้ว ความละเอียด 128x64 พิกเซล มีสวิตช์ปิดเปิด เขียนโปรแกรมสั่งงานภาษาบล็อกหรือโค้ดไพทอนด้วย microBlock IDE หรือเขียนโปรแกรมภาษา C/C++ ด้วย Arduino IDE หรือ KBIDE

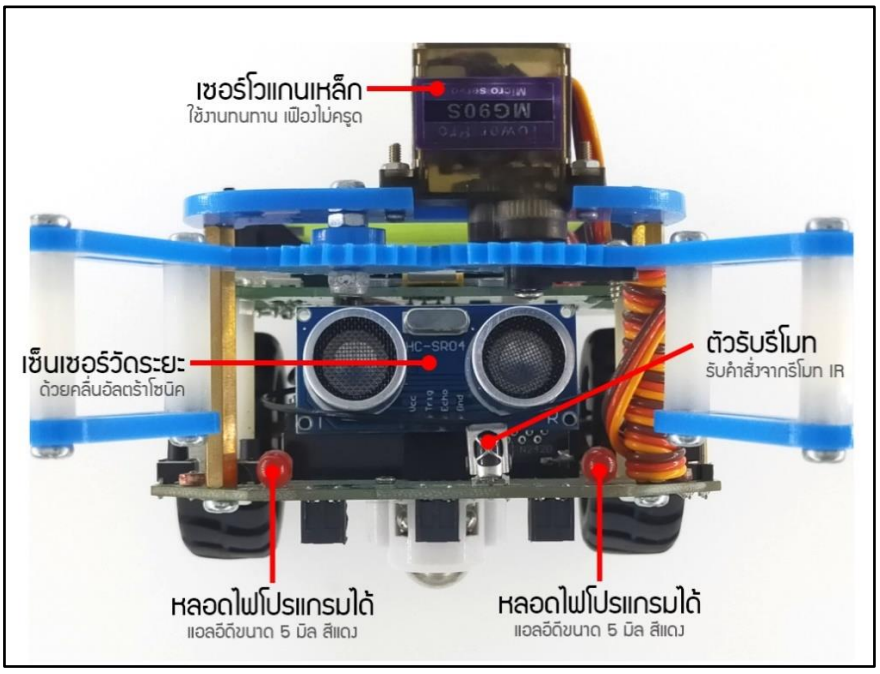
สามารถประยุกต์ทำหุ่นยนต์ได้หลากหลาย เช่น หุ่นยนต์เดินตามเส้น, หุ่นยนต์เดินตามแสง, หุ่นยนต์ขนย้ายสิ่งของ, หุ่นยนต์วิ่งตามโปรแกรม, รถบังคับด้วยสมาร์ทโฟน หากเชื่อมต่อกับ HuskyLens สามารถทำหุ่นยนต์ AI วิ่งตามสิ่งของ, หุ่นยนต์วิ่งตามมนุษย์ด้วยการตรวจจับใบหน้า ได้

ArtronShop ร่วมมือกับ PrinceBot ร่วมออกแบบและจัดทำ Rapbit32XA ขึ้นมา ลิขสิทธิ์ชื่อ Rapbit32 เป็นของ PrinceBot กรณีอ้างถึงหุ่นยนต์ Rapbit32XA ขอให้ใช้ชื่อเต็ม คือ Rapbit32XA (XA มาจาก x Atron)

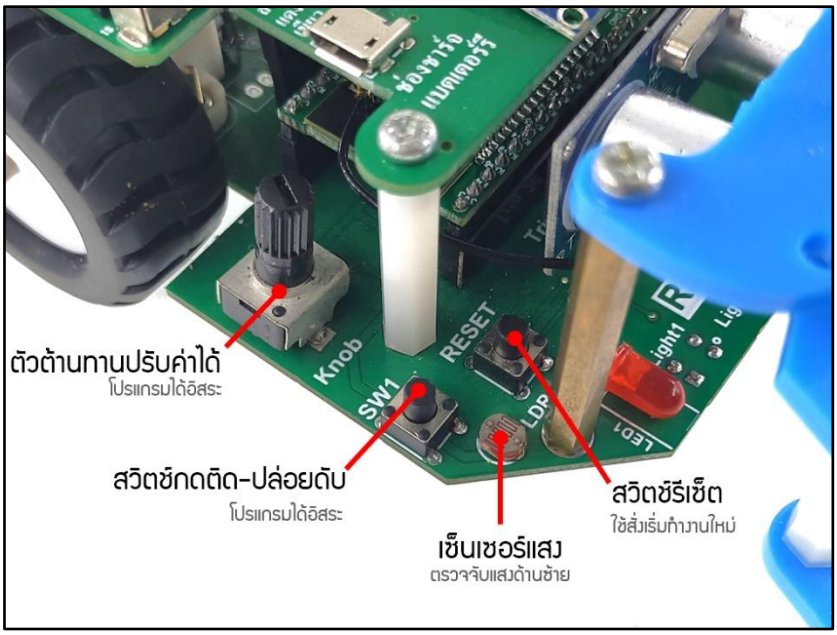
ส่วนประกอบ Rapbit32XA



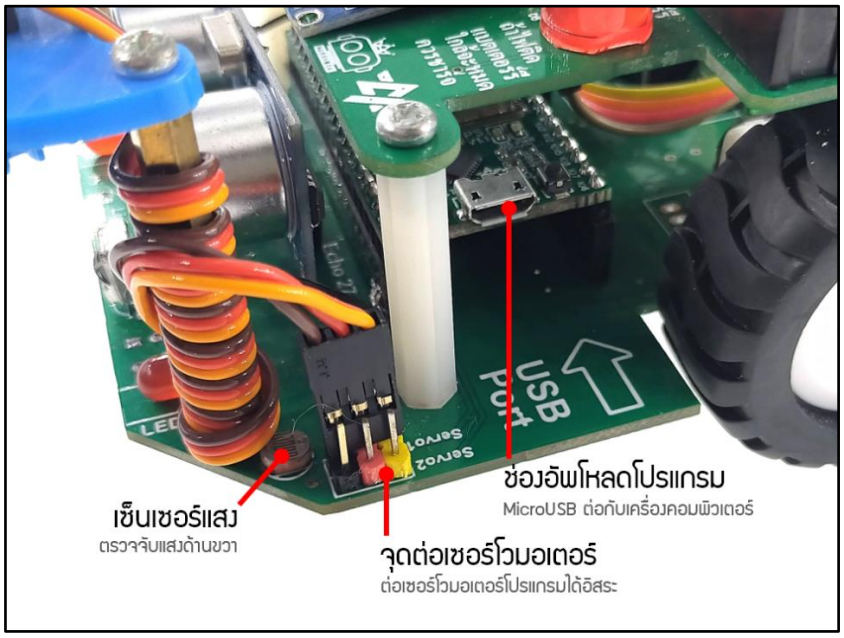
รูปที่ 1 ส่วนประกอบ Rapbit32XA มุมบน



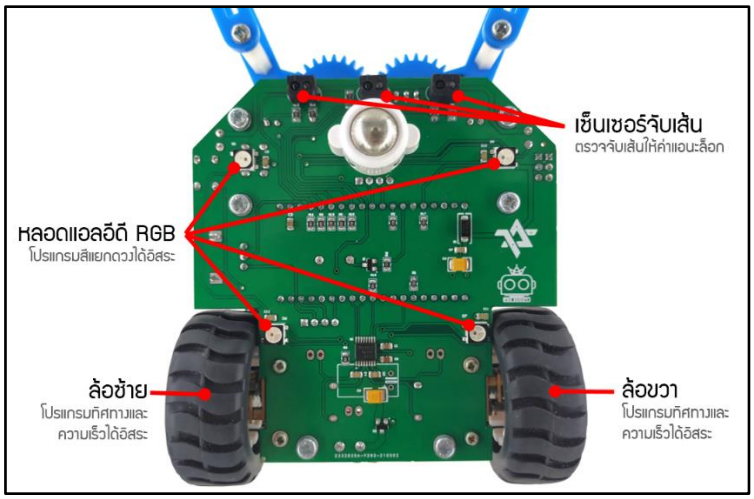
รูปที่ 2 ส่วนประกอบ Rapbit32XA ด้านหน้า



รูปที่ 3 ส่วนประกอบ Rapbit32XA ด้านซ้าย



รูปที่ 4 ส่วนประกอบ Rapbit32XA ด้านขวา



รูปที่ 5 ส่วนประกอบ Rapbit32XA ด้านล่าง

เขียนโปรแกรมสั่งงาน Rapbit32XA ด้วยภาษาบล็อก

Rapbit32XA รองรับการเขียนโปรแกรมด้วยภาษาบล็อกทั้งโปรแกรม microBlock IDE และ KBIDE ในเอกสารฉบับนี้จะแนะนำให้ใช้ร่วมกับโปรแกรม microBlock IDE สามารถเขียนได้ทั้งภาษาบล็อกและโค้ดไพทอน เมื่อเรียนรู้การเขียนโปรแกรมภาษาบล็อกไประดับหนึ่งแล้ว สามารถสลับไปเขียนภาษาไพทอนได้ทันที

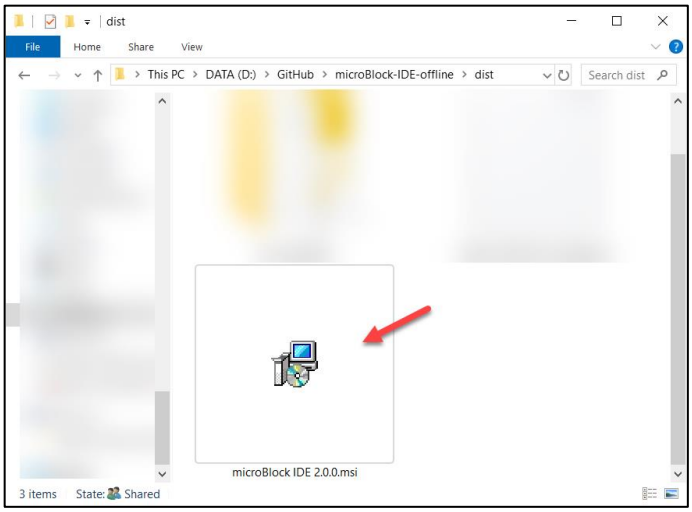
การติดตั้งโปรแกรม microBlock IDE

ดาวน์โหลดโปรแกรม microBlock IDE ซึ่งสามารถดาวน์โหลดได้โดยเข้าไปที่

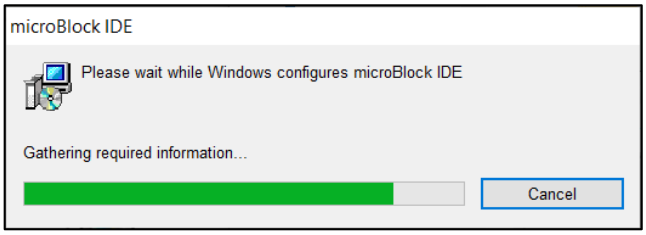
https://microblock.app/download

ในหน้าดาวน์โหลดโปรแกรม microBlock IDE สำหรับระบบปฏิบัติการ Windows จะมีให้เลือกดาวน์โหลด 4 ตัวเลือก โดยมีทั้งแบบไฟล์ติดตั้ง และแบบไม่ต้องติดตั้ง เพื่อความสะดวกในการใช้งานครั้งต่อไป แนะนำให้เลือกดาวน์โหลดแบบติดตั้ง (หมายเลข 1 หรือหมายเลข 3)

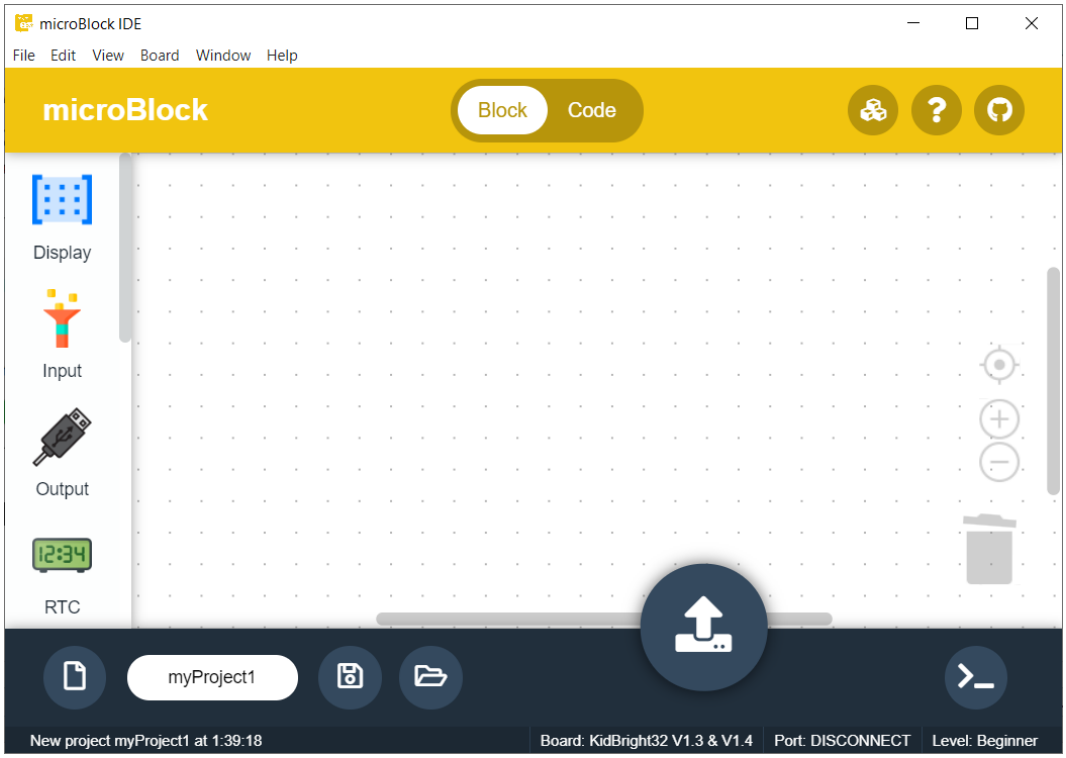
เมื่อดาวน์โหลดเรียบร้อยแล้ว ให้เปิดไฟล์ติดตั้งที่ดาวน์โหลดมา



หน้าต่างติดตั้งโปรแกรมจะแสดงขึ้นมา ให้รอจนกว่าจะติดตั้งเสร็จ



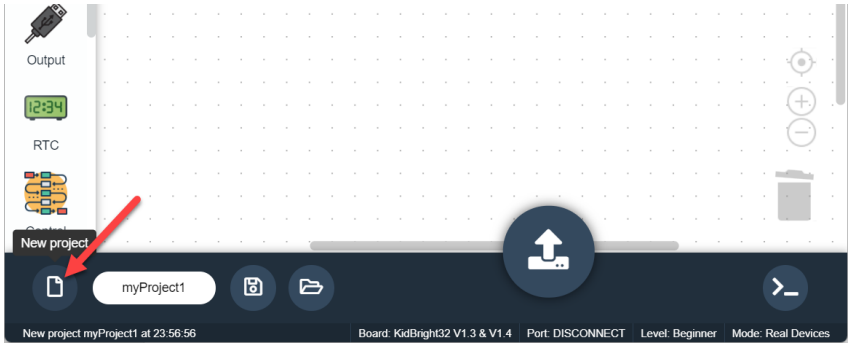
เมื่อติดตั้งเสร็จ โปรแกรม microBlock IDE จะเปิดขึ้นมาอัตโนมัติ



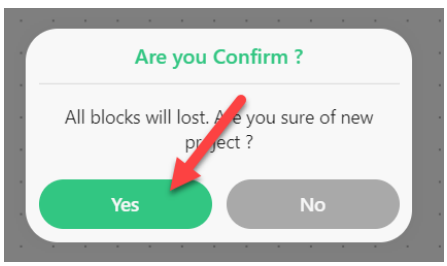
★ ที่มา [เริ่มต้นใช้งาน microBlock IDE เวอร์ชั่นโปรแกรม - microBlock IDE]

การเลือกบอร์ด Rapbit32XA

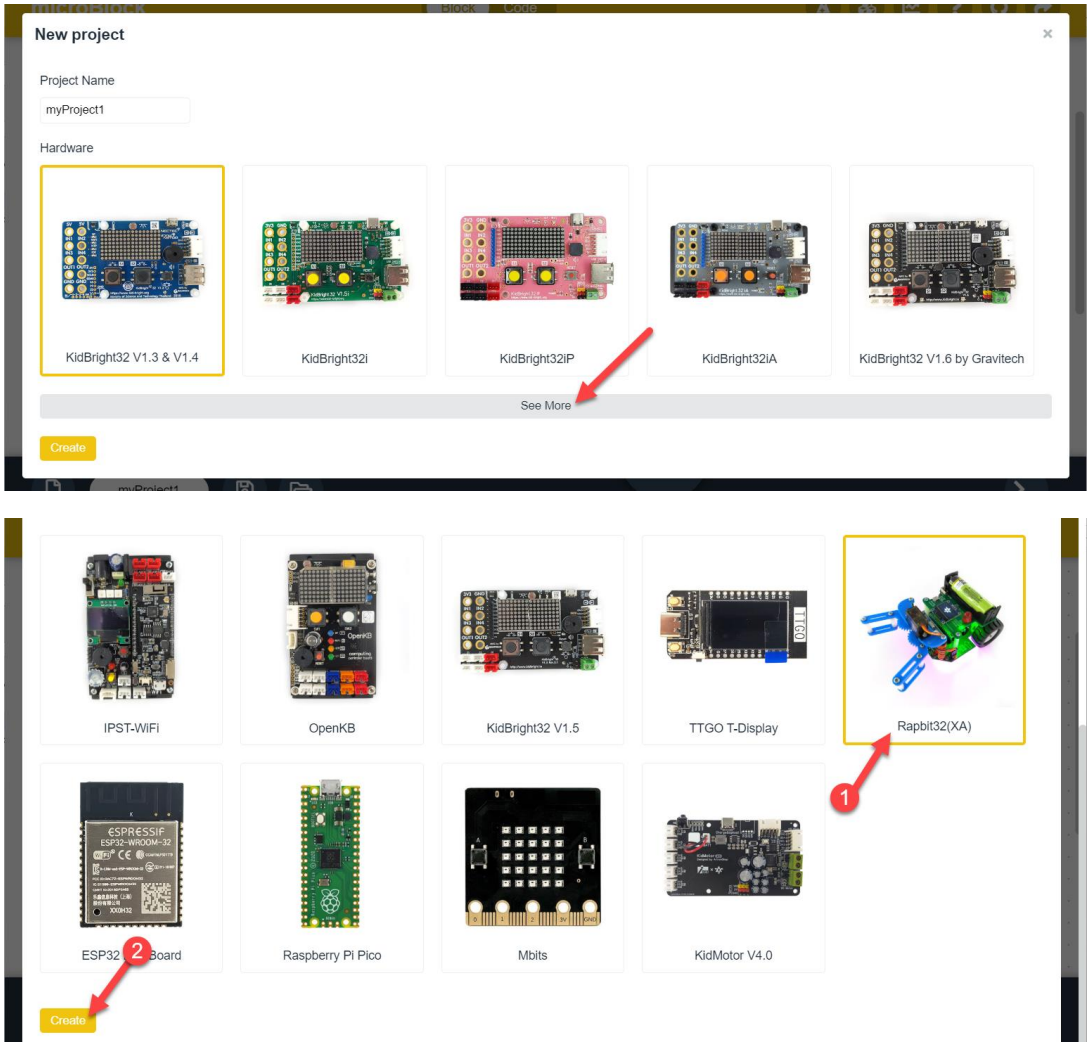
กดปุ่มสร้างโปรเจคใหม่มุมซ้ายล่าง



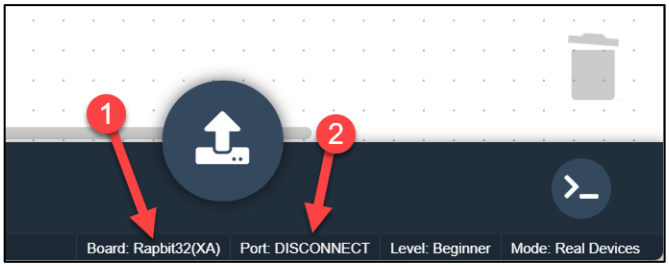
กดปุ่ม Yes เพื่อยืนยันสร้างโปรเจคใหม่ (สร้างโปรเจคใหม่บล็อกที่เคยลากไว้จะหายไป)



ในหน้าต่าง New project ให้กดปุ่ม See More จากนั้นเลือก Rapbit32XA แล้วกดปุ่ม Create

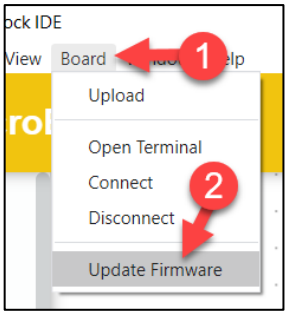


จากนั้นเมนูให้เลือกบล็อกด้านซ้ายจะเปลี่ยนเป็นของ Rapbit32XA ให้ตรวจสอบชื่อบอร์ดที่เลือก (1) และตรวจสอบสถานการณ์เชื่อมต่อ (2) ที่แถบด้านล่าง

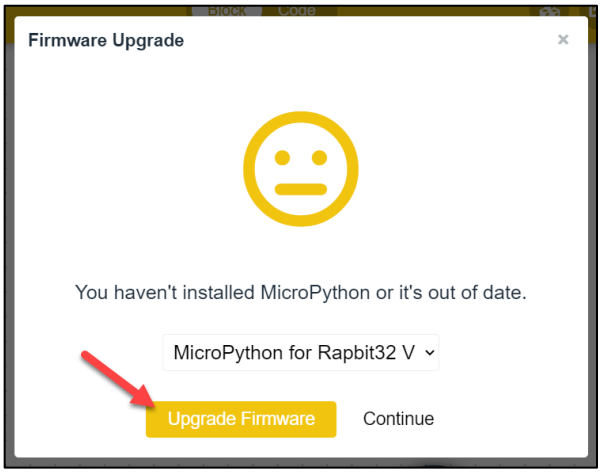


การติดตั้งเฟิร์มแวร์ MicroPython for Rapbit32XA

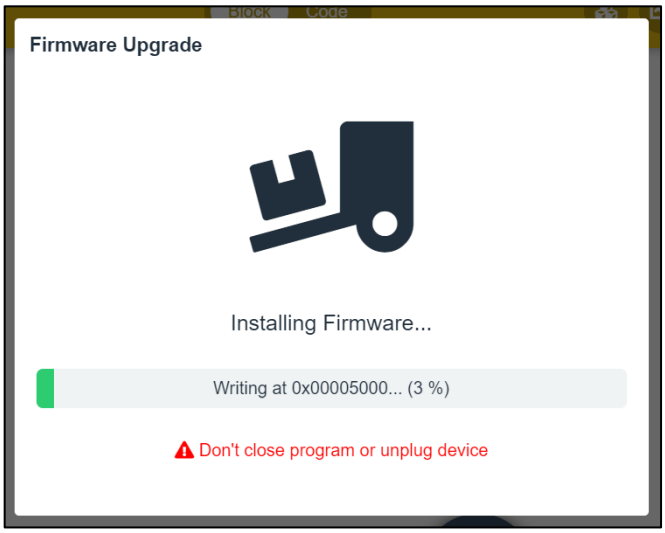
การใช้งานครั้งแรกจำเป็นต้องติดตั้งเฟิร์มแวร์ MicroPython ก่อน โดยเชื่อมต่อตัวรถเข้ากับเครื่องคอมพิวเตอร์ จากนั้นกดไปที่ Board เลือก Update Firmware



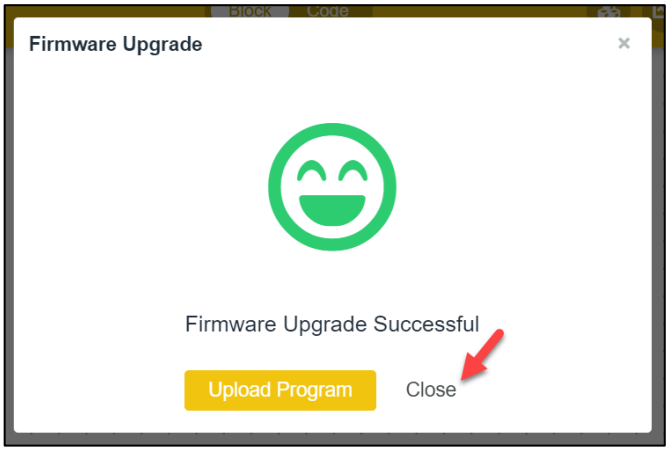
ในหน้าต่าง Firmware Upgrade ให้กดปุ่ม Upgrade Firmware



โปรแกรมจะเริ่มติดตั้ง MicroPython ลง Rapbit32XA ให้รอจนกว่าการติดตั้งจะเสร็จสิ้น

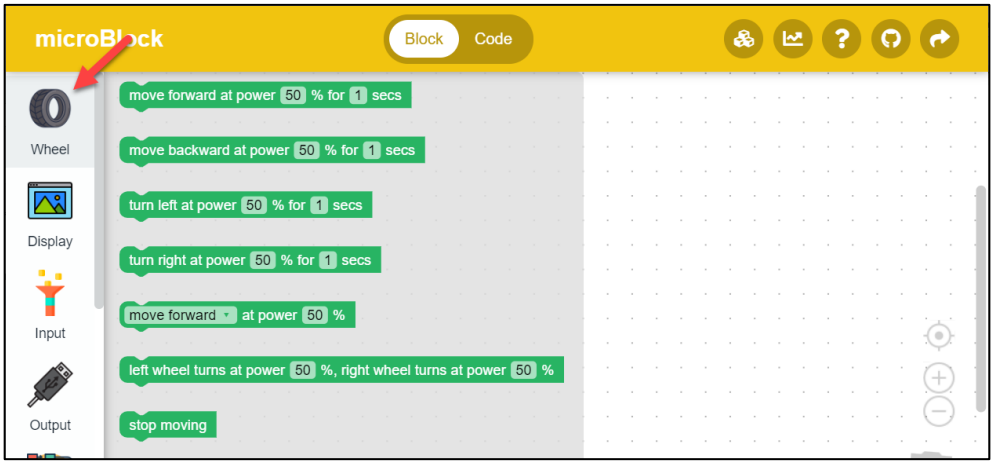


เมื่อเสร็จแล้วให้กดปุ่ม Close เพื่อปิดหน้าต่างไป

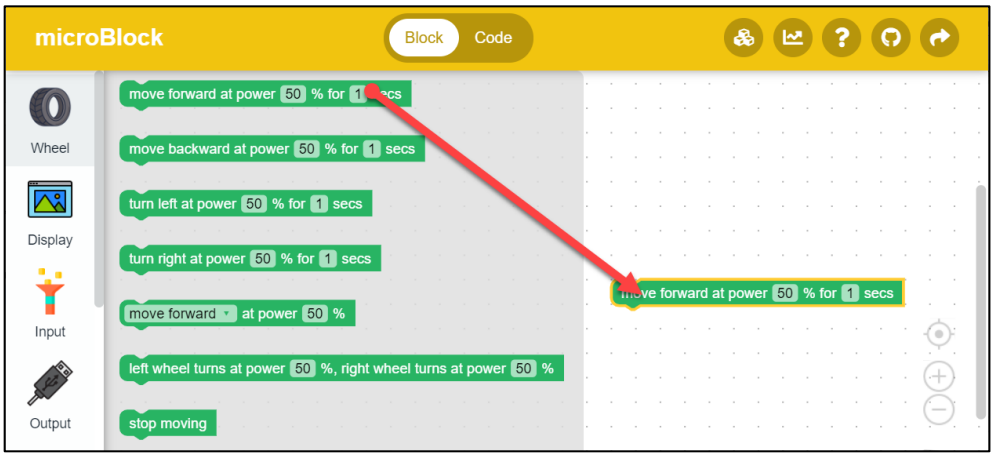


การเขียนโปรแกรมให้รถวิ่ง

คลิกที่เมนู Wheel จะมีบล็อกขึ้นมาให้เลือกใช้งานดังนี้



การใช้งานให้ลากบล็อกที่ต้องการใช้งานมาใส่ในพื้นที่ทำงาน ตัวอย่างต้องการให้รถเดินหน้าเป็นเวลา 1 วินาที จึงลากบล็อก move forward at power ... % for ... secs ออกมา



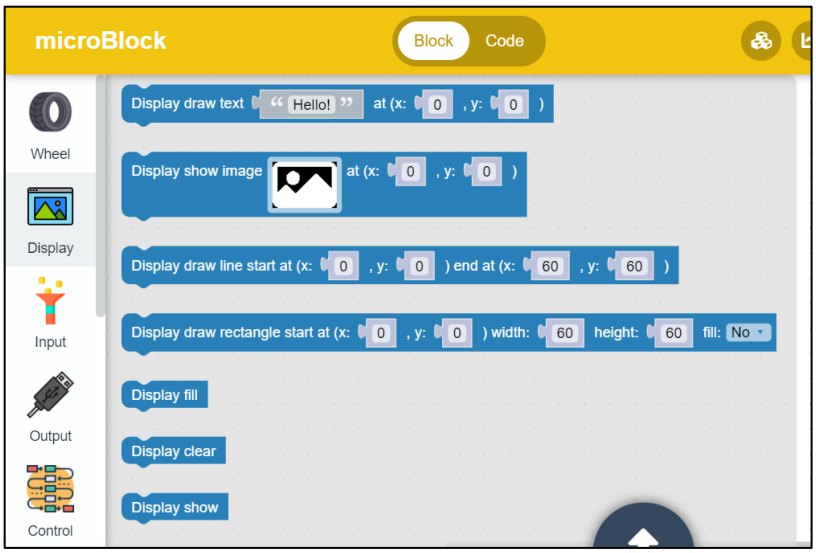
กดปุ่มอัพโหลดโปรแกรม แล้วรอซักครู่

โปรแกรมจะเริ่มทำงานทันที รถจะวิ่งเดินหน้าเป็นเวลา 1 วินาทีด้วยความเร็ว 50%

หมายเหตุ. ก่อนอัพโหลดโปรแกรมแนะนำให้ปิดสวิตช์บนตัวรถไว้ก่อนเพื่อปิดการทำงานของมอเตอร์ ป้องกันรถวิ่งตกจากโต๊ะหรือสร้างความเสียหายกับสิ่งของโดยรอบ เมื่อต้องการดูผลการทำงานของโปรแกรมค่อยกดปุ่ม RESET เพื่อให้โปรแกรมเริ่มทำงานใหม่

การใช้งานหน้าจอแสดงผล

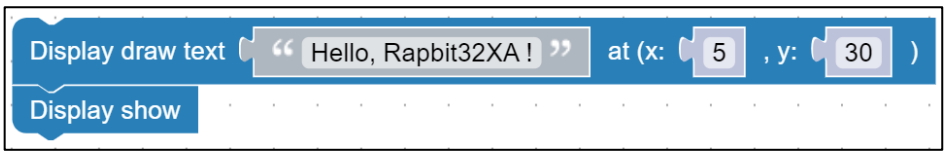
หน้าจอแสดงผลบน Rapbit32XA เป็นหน้าจอแบบ OLED สีขาว-ดำ ขนาด 0.96 นิ้ว ความละเอียด 128x64 พิกเซล รองรับการแสดงผลรูปภาพและข้อความ การสั่งงานหน้าจอใช้บล็อกในเมนู Display โดยมีบล็อกให้เลือกใช้ดังนี้



ตัวอย่าง ต้องการแสดงผลคำว่า Hello, Rapbit32XA ! เยื้องซ้ายจอ (x) 5 พิกเซล และเยื้องบน (y) ที่ 30 พิกเซล จึงลากบล็อก Display draw text ออกมา แล้วแก้ไขข้อมูลบนบล็อกตามรูป

คำสั้ง Display draw ยังไม่มีผลให้หน้าจอทำงาน (หน้าจอยังไม่แสดงข้อความ) จำเป็นต้องเรียกใช้คำสั่ง Display show เพื่อให้หน้าจอแสดงผลตามบล็อกก่อนหน้านี้

ลากบล็อก Display show มาต่อด้านล่างบล็อก Display draw ดังรูป

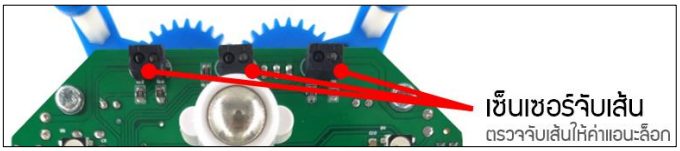


อัพโหลดโปรแกรม หน้าจอแสดงคำว่า Hello, Rapbit32XA ! ดังรูปที่ 6

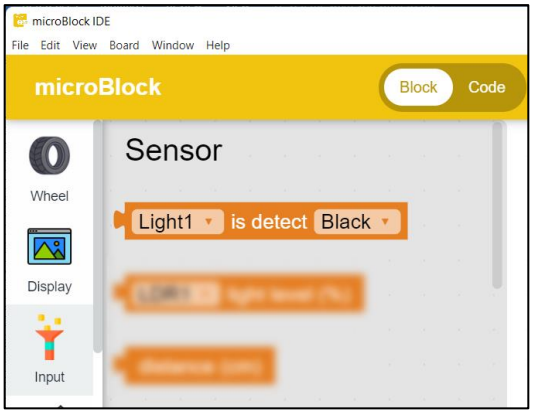
รูปที่ 6 หน้าจอ OLED แสดงคำว่า Hello, Rapbit32XA !

การใช้งานเซ็นเซอร์ตรวจจับเส้น

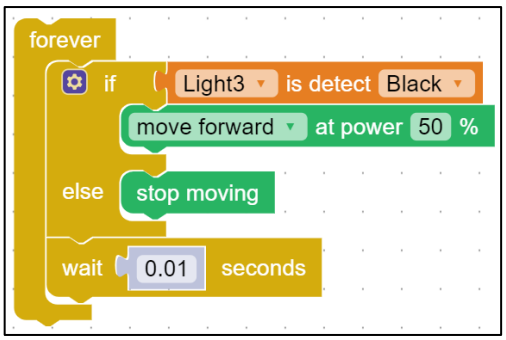
Rapbit32XA มีเซ็นเซอร์ตรวจจับเส้นจำนวน 3 ตัว ประกอบด้วย Light1, 2 และ 3 อยู่ด้านใต้



การใช้งานให้ใช้บล็อก light... is detect ... ในเมนู Input หัวข้อ Sensor ร่วมกับบล็อก if



ตัวอย่าง เขียนโปรแกรมเมื่อเซ็นเซอตรวจจับเส้นซ้ายสุด (Light3) ตรวจเจอสีดำ ให้รถเดินหน้า แต่ถ้าเจอสีขาว ให้หยุด เขียนโปรแกรมได้ดังนี้



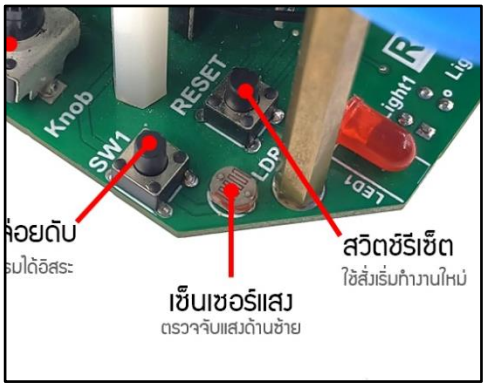
หลักการทำงานมีดังนี้

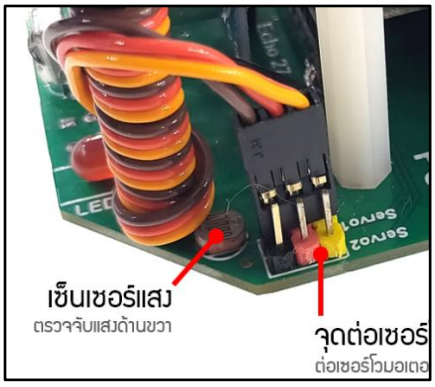
- บล็อก forever – ให้บล็อกภายใต้บล็อกนี้ทำงานไม่สิ้นสุด
  - บล็อก if และบล็อก Light3 is detect Block – ถ้า Light3 เจอสีดำ ให้
    - รถวิ่งเดินหน้าด้วยความเร็ว 50%
  - else – ถ้าเงื่อนไขไม่เป็นจริง (เจอสีขาว) ให้
    - รถหยุดวิ่ง
  - บล็อก wait ... seconds – หน่วงเวลา 0.01 วินาที

เมื่ออัพโหลดโปรแกรมลงบอร์ด ผลที่ได้ เมื่อเซ็นเซอตรวจจับเส้นซ้ายสุด (Light3) ยังเจอสีดำอยู่ รถเดินหน้าไปเรื่อย ๆ แต่เมื่อ Light3 เจอสีขาว รถหยุดวิ่ง

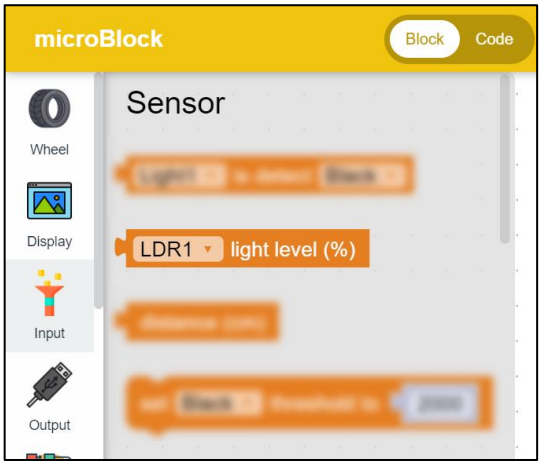
การใช้งานเซ็นเซอร์แสง

เซ็นเซอร์แสง (LDR) สามารถประยุกต์ทำรถวิ่งตามแสง หรือวิ่งหนีแสงได้ มีด้วยกัน 2 ตัวด้านหน้าทางซ้าย และขวา (LDR1 และ LDR2)

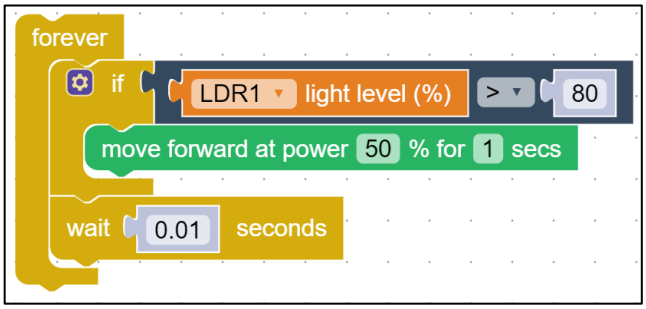




การใช้งานให้ใช้บล็อก ... light level (%) ในเมนู Input หัวข้อ Sensor ร่วมกับบล็อก if



ตัวอย่าง เขียนโปรแกรมทำรถวิ่งหนีแสง เมื่อเจอแสงรถจะวิ่งไปข้างหน้าเพื่อหนีแสง เขียนโปรแกรมได้ดังนี้



หลักการทำงานมีดังนี้

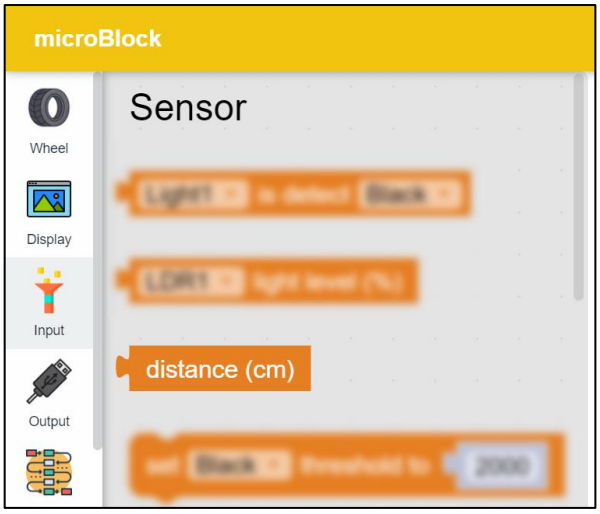
- บล็อก forever – ให้บล็อกภายใต้บล็อกนี้ทำงานไม่สิ้นสุด
    - บล็อก if บล็อกมากกว่า และบล็อก LDR1 light level (%) – ถ้า LDR2 ได้รับแสงมากกว่า 80% ให้
        - รถวิ่งเดินหน้าด้วยความเร็ว 50% เป็นเวลา 1 วินาที
    - บล็อก wait ... seconds – หน่วงเวลา 0.01 วินาที

เมื่ออัพโหลดโปรแกรมลงบอร์ด ผลที่ได้ นำไฟฉายส่องไปที่ LDR1 รถวิ่งจะไปข้างหน้าเพื่อหนีแสง

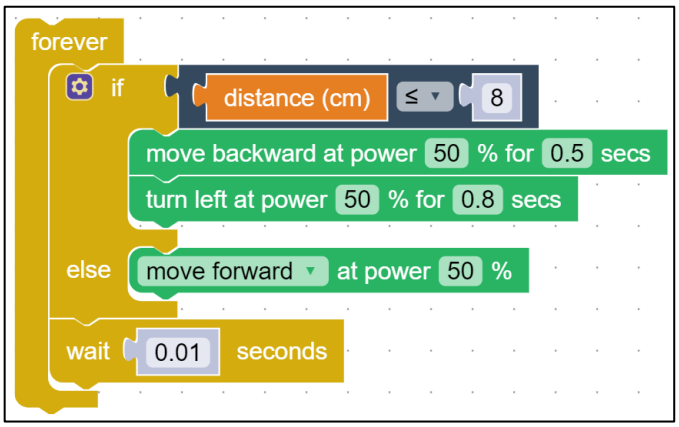
การใช้งานเซ็นเซอร์วัดระยะ

เซ็นเซอร์วัดระยะใช้ตรวจสอบวัตถุกีดขวาง หรือใช้ตรวจจับวัตถุที่อยู่ด้านหน้า เพื่อใช้แขนจับหยิบ-เลื่อนวัตถุ เซ็นเซอร์วัดระยะได้ 2 เซนติเมตร ถึง 4 เมตร

การใช้งานให้ใช้บล็อก distance (cm) ในเมนู Input หัวข้อ Sensor ร่วมกับบล็อก if



ตัวอย่าง เขียนโปรแกรมหุ่นยนต์ดูดฝุ่น มีหลักการทำงานที่สำคัญคือ หุ่นยนต์จะวิ่งไปด้านหน้าเรื่อย ๆ เมื่อเจอกำแพง หรือเจอกำแพง หุ่นยนต์จะถอยหลังแล้วเลี้ยวไปทางอื่นเพื่อไม่ให้ชนกำแพง เขียนโปรแกรมได้ดังนี้



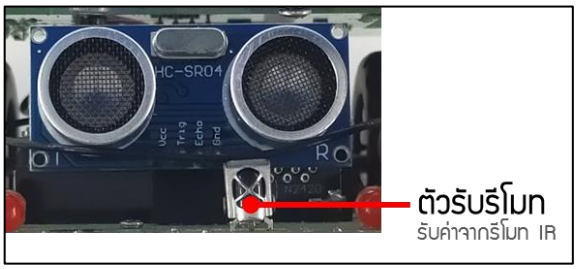
หลักการทำงาน มีดังนี้

- บล็อก forever – ให้บล็อกภายใต้บล็อกนี้ทำงานไม่สิ้นสุด
    - บล็อก if บล็อกน้อยกว่าหรือเท่ากับ และบล็อก distance (cm) – ถ้าวัดระยะได้น้อยกว่าหรือเท่ากับ 8 เซ็นติเมตรให้
        - รถวิ่งถอยหลังด้วยความเร็ว 50% เป็นเวลา 0.5 วินาที
        - รถเลี้ยวซ้ายด้วยความเร็ว 50% เป็นเวลา 0.8 วินาที
    - else – ถ้าเงื่อนไขไม่เป็นจริง (ระยะมากกว่า 8 เซ็นฯ) ให้
        - รถวิ่งไปข้างหน้า
    - บล็อก wait ... seconds – หน่วงเวลา 0.01 วินาที

เมื่ออัพโหลดโปรแกรมลงบอร์ด ผลที่ได้ รถจะวิ่งเดินหน้าไปเรื่อย ๆ เมื่อเจอกำแพงหรือสิ่งกีดขวาง รถจะถอยหลังแล้วเลี้ยวเพื่อเลี่ยงการชน

การใช้งานตัวรับรีโมท

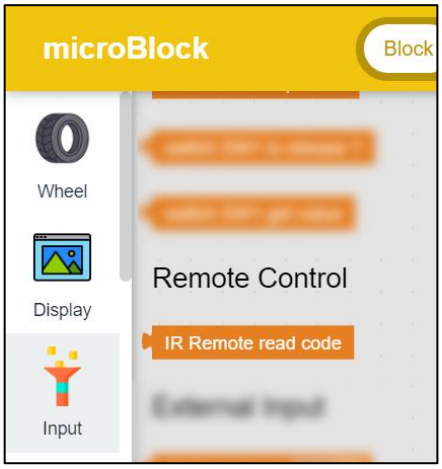
ตัวรับรีโมทรองรับการใช้งานร่วมกับรีโมทที่แถมไปให้เท่านั้น โดยตัวรับรีโมทจะอยู่ด้านหน้าตัวรถ การใช้งานต้องกดรีโมทบริเวณด้านหน้าตัวรถเท่านั้น



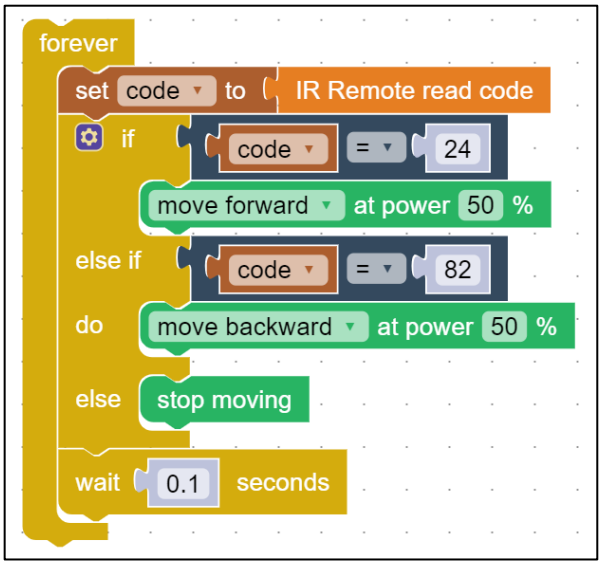
ตัวรับรีโมทจะรับค่าเป็นตัวเลขรหัสของปุ่มที่กด เช่น กดปุ่ม ▲ เลขรหัสจะเป็น 24 เลขรหัสแต่ละปุ่มบนรีโมท สามารถดูได้จากตามตารางด้านล่างนี้

ปุ่ม	เลขรหัส	ปุ่ม	เลขรหัส	ปุ่ม	เลขรหัส
❶	69	❼	7	▲	24
❷	70	❽	21	▼	82
❸	71	❾	9	◀	8
❹	68	✱	22	▶	90
❺	64	❿	25	OK	28
❻	67	#	13		

การใช้งานให้ใช้บล็อก IR Remote read code ในเมนู Input หัวข้อ Remote Control ร่วมกับ if



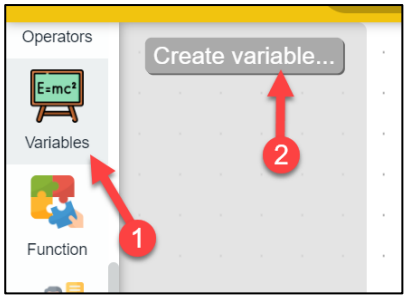
ตัวอย่าง ทำรถบังคับด้วยรีโมท กดปุ่ม ▲ ให้รถเดินหน้า กดปุ่ม ▼ ให้รถถอยหลัง เขียนโปรแกรมได้ดังนี้



บล็อก set code to ... และบล็อก code เกิดจากการสร้างตัวแปรขึ้นมา ขั้นตอนการสร้างตัวแปรมีดังนี้

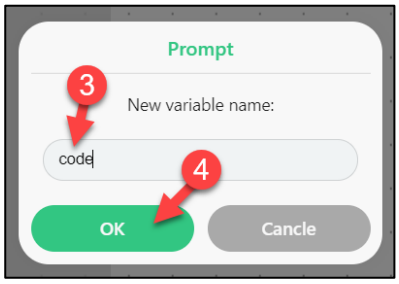
(1) คลิกที่เมนู Variables

(2) กดปุ่ม Create variable...

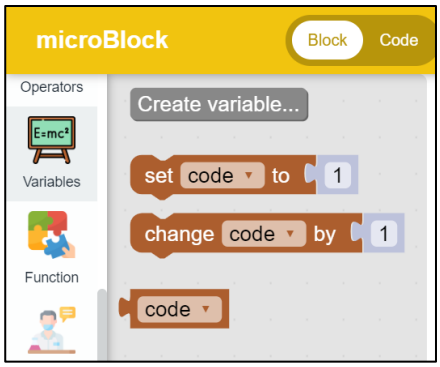


(3) ใส่ชื่อตัวแปรที่ต้องการสร้าง

(4) กดปุ่ม OK



จะมีบล็อกเกี่ยวกับตัวแปรแสดงขึ้นมาแล้ว



หลักการทำงาน มีดังนี้

- บล็อก forever – ให้บล็อกภายใต้บล็อกนี้ทำงานไม่สิ้นสุด
  - บล็อก set code to และบล็อก IR Remote read code – อ่านค่าจากตัวรับรีโมท เก็บเลขโค้ดที่อ่านได้ลงตัวแปร code
  - บล็อก if บล็อกเท่ากับ และบล็อก code – ถ้าในตัวแปร code เท่ากับ 24 (ปุ่ม ▲) ให้
    - รถวิ่งเดินหน้าด้วยความเร็ว 50%
  - บล็อก else if บล็อกเท่ากับ และบล็อก code – ถ้าเงื่อนไขก่อนหน้าไม่เป็นจริง ให้ตรวจสอบว่า code เท่ากับ 28 (ปุ่ม ▼) หรือไม่ ถ้าใช่ ให้
    - รถวิ่งถอยหลังด้วยความเร็ว 50%
  - บล็อก else – ถ้าเงื่อนไขก่อนหน้าไม่ใช่ทั้งหมด ให้
    - รถหยุดวิ่ง
  - บล็อก wait ... seconds – หน่วงเวลา 0.01 วินาที

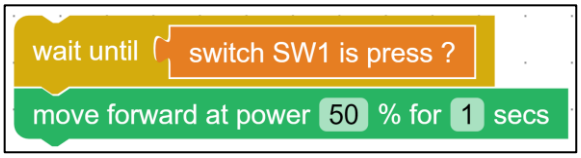
เมื่ออัพโหลดโปรแกรมลงบอร์ด ผลที่ได้ กดปุ่ม ▲ บนรีโมท รถเดินหน้า กดปุ่ม ▼ รถถอยหลัง

การใช้งานสวิตช์

สวิตช์ SW1 สามารถนำมาใช้เขียนโปรแกรมอ่านค่าและใช้งานได้อิสระ เหมาะสำหรับทำปุ่ม Start กดปุ่ม SW1 แล้วโปรแกรมหลักจึงจะทำงาน

การใช้งานสวิตช์ SW1 ให้ใช้บล็อกในเมนู Input หัวข้อ Switch

ตัวอย่าง เขียนโปรแกรมให้กดปุ่ม SW1 แล้วรถค่อยเดินหน้า เขียนโปรแกรมได้ดังนี้



หลักการทำงาน มีดังนี้

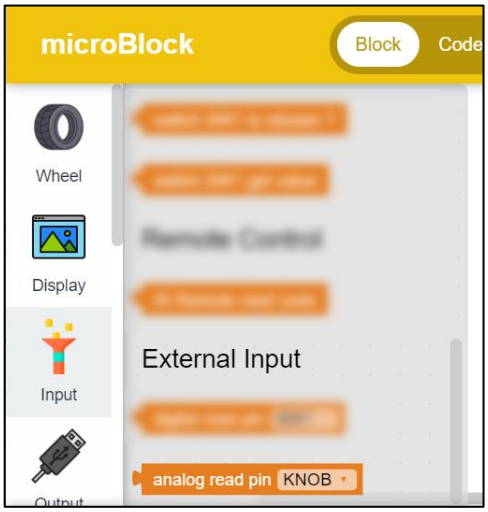
- บล็อก wait until และ switch SW1 is press ? – ให้รอจนกว่าสวิตช์ SW1 จะถูกกด
- รถวิ่งไปข้างหน้าด้วยความเร็ว 50% เป็นเวลา 1 วินาที

เมื่ออัพโหลดโปรแกรมลงบอร์ด ผลที่ได้ รถยังนิ่ง ล้อยังไม่หมุน เมื่อกดสวิตช์ SW1 ล้อจะหมุนทันที

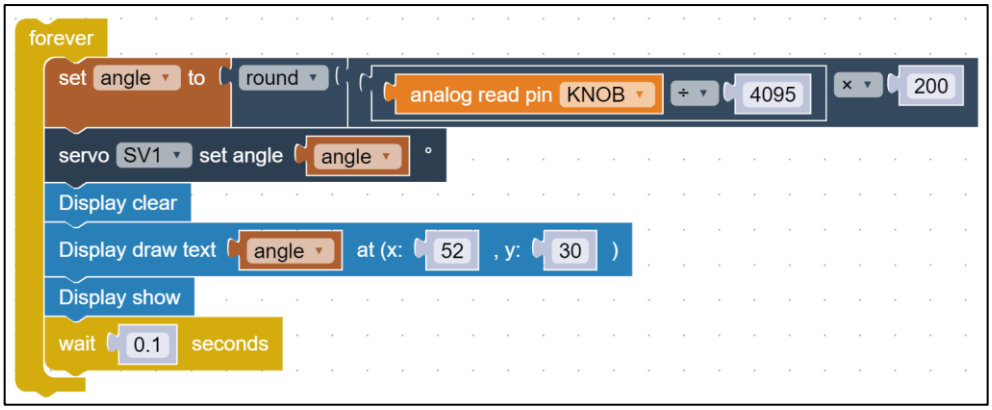
การใช้งานตัวต้านทานปรับค่าได้

ตัวต้านทานปรับค่าได้ อยู่ด้านหน้าตัวรถ เมื่อใช้มือหมุนปรับเพื่อเปลี่ยนค่าได้

การใช้งานตัวต้านทานปรับค่าได้ ให้ใช้บล็อก analog read ในเมนู Input หัวข้อ External Input อ่านค่าออกมาได้เลย โดยให้ค่า 0 ถึง 4095 (ปรับซ้ายสุด-หมุนทวนเข็มได้ 0, ปรับขวาสุด-หมุนตามเข็มได้ 4095)



ตัวอย่าง เขียนโปรแกรมให้เซอร์โวมอเตอร์แขนจับหมุนตามองศาที่ปรับตัวต้านทานปรับค่าได้ พร้อมแสดงผลองศาที่หมุนบนหน้าจอ เขียนโปรแกรมได้ดังนี้



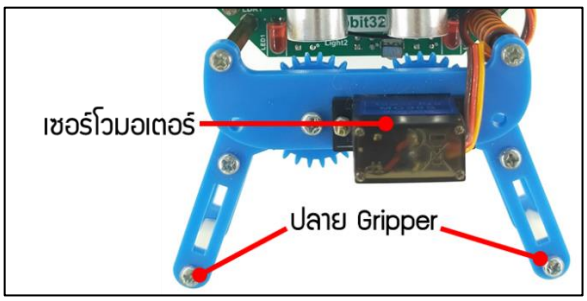
หลักการทำงาน มีดังนี้

- บล็อก forever – ให้บล็อกภายใต้บล็อกนี้ทำงานไม่สิ้นสุด
    - บล็อก set angle to, round, □, ÷ – อ่านค่าจากตัวต้านทานปรับค่าได้ นำมาหาร 4095 คูณ 200 เพื่อแปลงตัวเลขช่วง 0 ถึง 4095 เป็น 0 ถึง 200 (เทียบบัญญัติไตรยาง) แล้วปัดเศษด้วย round นำค่าที่ได้เก็บลงตัวแปร angle
    - servo SV1 set angle ... – กำหนดให้เซอร์โว SV1 หมุนตามองศาในตัวแปร angle
    - บล็อก Display clear – ล้างหน้าจอ OLED (ลบข้อความก่อนหน้าที่เคย draw ไป)
    - บล็อก Display draw text – เขียนข้อความในตัวแปร angle ลงบนหน้าจอที่พิกัด x: 52, y: 30
    - บล็อก Display show – สั่งให้หน้าจอแสดงผล
    - บล็อก wait ... seconds – หน่วงเวลา 0.1 วินาที

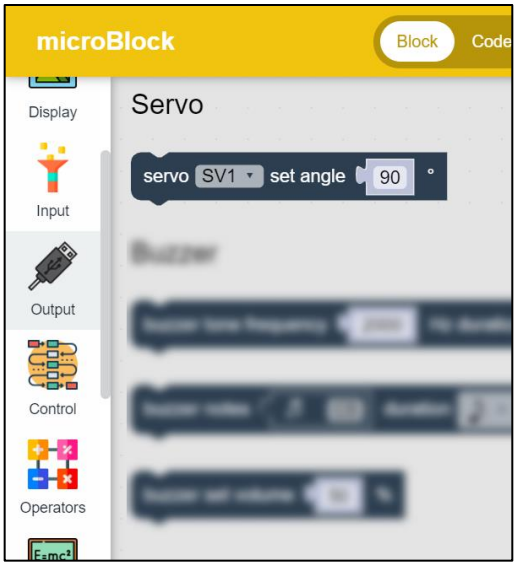
เมื่ออัพโหลดโปรแกรมลงบอร์ด ผลที่ได้ หน้าจอจะแสดงค่าองศาที่เซอร์โวมอเตอร์แขนจับหมุน เมื่อปรับหมุนตัวต้านทานปรับค่าได้ หน้าจอจะแสดงตัวเลขของศาเปลี่ยนไป รวมทั้งแขนจับ หุบ-กาง ตามทิศทางการหมุนวต้านทานปรับค่าได้

การใช้งานแขนจับ

แขนจับ (Gripper) ใช้จับสิ่งของเพื่อย้ายตำแหน่ง ส่วนประกอบที่สำคัญของ Gripper คือเซอร์โวมอเตอร์ เมื่อสั่งให้เซอร์โวมอเตอร์หมุนในองศาที่แตกต่างกัน จะส่งผลให้ Gripper หุบหรือกาง ตามองศาที่สั่งเซอร์โวมอเตอร์

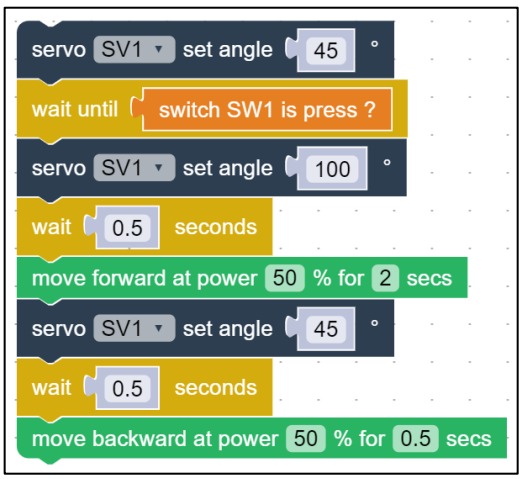


การเขียนโปรแกรมสั่งงาน Gripper ให้ใช้บล็อก servo SV1 set angle ... ในเมนู Output หัวข้อ Servo สังให้เซอร์โวมอเตอร์หมุนเป็นองศา เพื่อให้ Gripper หุบหรือกาง



การหาองศาที่ Gripper กางได้สุด และองศาที่ Gripper หยิบสิ่งของได้แน่น แนะนำให้ใช้โปรแกรมตัวอย่างในหัวข้อ การใช้งานตัวต้านทานปรับค่าได้ ในการหา

ตัวอย่างโปรแกรม ใช้ Gripper ย้ายวัตถุ เขียนโปรแกรมได้ดังนี้



หลักการทำงาน มีดังนี้

- บล็อก servo SV1 set angle 45 – สั่งให้ Gripper อยู่ตำแหน่งกาง
- บล็อก wait until switch SW1 is press ? – รอจนกว่าจะกดปุ่ม SW1
- บล็อก servo SV1 set angle 100 – Gripper จับวัตถุ
- บล็อก wait 0.5 seconds – หน่วงเวลา 0.5 วินาที (รอให้จับวัตถุนิ่ง)
- บล็อก move forward – ให้รถเดินหน้าด้วยความเร็ว 50% เป็นเวลา 2 วินาที
- บล็อก servo SV1 set angle 45 – Gripper ปล่อยวัตถุ

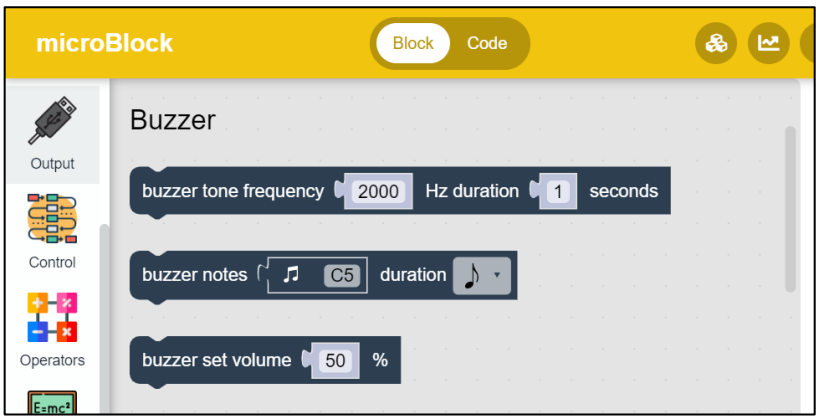
- บล็อก wait 0.5 seconds – หน่วงเวลา 0.5 วินาที (รอให้วัตถุนิ่ง)
- บล็อก move backward – ให้รถเดินหน้าด้วยความเร็ว 50% เป็นเวลา 1 วินาที

เมื่ออัพโหลดโปรแกรมลงบอร์ด ผลที่ได้ หุ่นยนต์อยู่นิ่ง ๆ นำของมาวางระหว่าง Gripper กดปุ่ม SW1 แล้ว Gripper หนีบวัตถุไว้ วิ่งไปข้างหน้า ปล่อยวัตถุ แล้วถอยหลัง

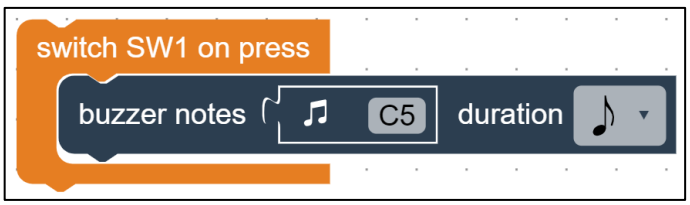
การใช้งานบัสเซอร์

บัสเซอร์อยู่ใต้บอร์ด IOXESP32PS ไม่สามารถมองเห็นได้ถ้าไม่ถอดบอร์ดแบตเตอรี่และบอร์ด IOXESP32PS ออก บัสเซอร์ใช้ส่งเสียงโน๊ตต่าง ๆ เพื่อบอกสถานะ เช่น ส่งเสียงเพื่อบอกสถานะจบการทำงาน ส่งเสียงเพื่อบอกสถานะเริ่มทำงาน ส่งเสียงเมื่อทำงานผิดพลาด เป็นต้น

การเขียนโปรแกรมสั่งงานบัสเซอร์ ให้ใช้บล็อกในเมนู Output หัวข้อ Buzzer



ตัวอย่างโปรแกรม ให้ส่งเสียงโน๊ต C5 เมื่อกดปุ่ม SW1 เขียนโปรแกรมได้ดังนี้



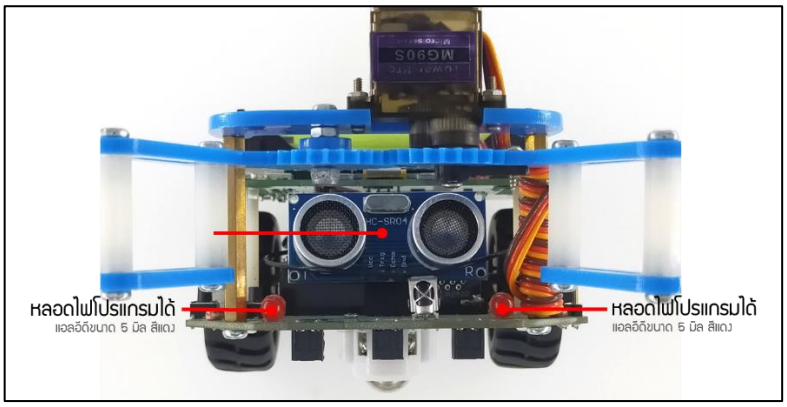
หลักการทำงาน มีดังนี้

- บล็อก switch SW1 on press – โปรแกรมภายใต้บล็อกนี้ถูกเรียกใช้เมื่อสวิตช์ SW1 ถูกกด
    - บล็อก buzzer notes ... duration ... – สั่งให้บัสเซอร์ส่งเสียงออกมา

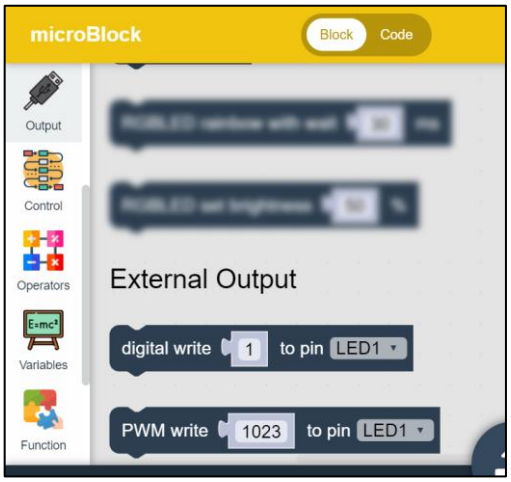
เมื่ออัพโหลดโปรแกรมลงบอร์ด ผลที่ได้ กดสวิตช์ SW1 จะมีเสียงดังออกมาจากบัสเซอร์

การสั่งงานหลอดแอลอีดีไฟหน้า

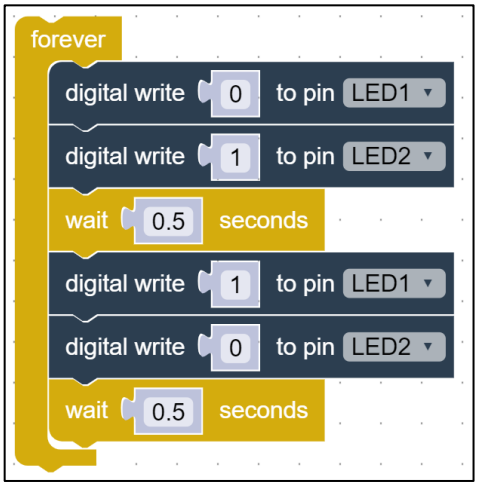
ไฟหน้า Rapbit32XA เป็นหลอดแอลอีดี 2 ดวง ขนาด 5 มิลลิเมตร สามารถสั่งให้เปิด-ปิดได้อิสระ



การเขียนโปรแกรมสั่งงานไฟหน้า ให้ใช้บล็อก digital write ในเมนู Output หัวข้อ External Output โดยสั่งให้เป็น 1 ไฟจะติด และสั่งให้เป็น 0 ไฟจะดับ



ตัวอย่างโปรแกรม สั่งให้ไฟหน้าซ้ายและไฟหน้าขวากระพริบสลับกัน เขียนโปรแกรมได้ดังนี้



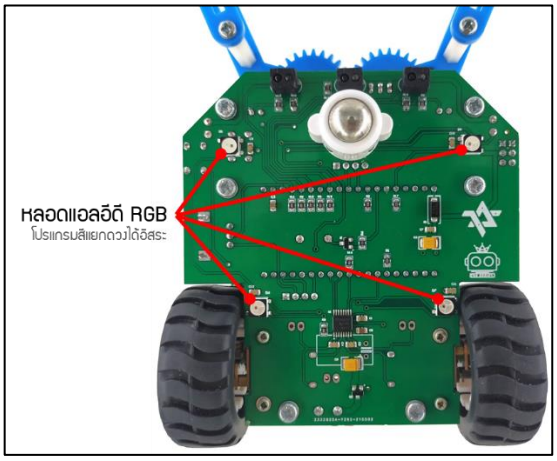
หลักการทำงาน มีดังนี้

- บล็อก forever – ให้บล็อกภายใต้บล็อกนี้ทำงานไม่สิ้นสุด
    - digital write 0 to pin LED1 – สั่งให้ไฟหน้าด้านซ้ายดับ
    - digital write 1 to pin LED2 – สั่งให้ไฟหน้าด้านขวาติด
    - บล็อก wait ... seconds – หน่วงเวลา 0.5 วินาที
    - digital write 1 to pin LED1 – สั่งให้ไฟหน้าด้านซ้ายติด
    - digital write 0 to pin LED2 – สั่งให้ไฟหน้าด้านขวาดับ
    - บล็อก wait ... seconds – หน่วงเวลา 0.5 วินาที

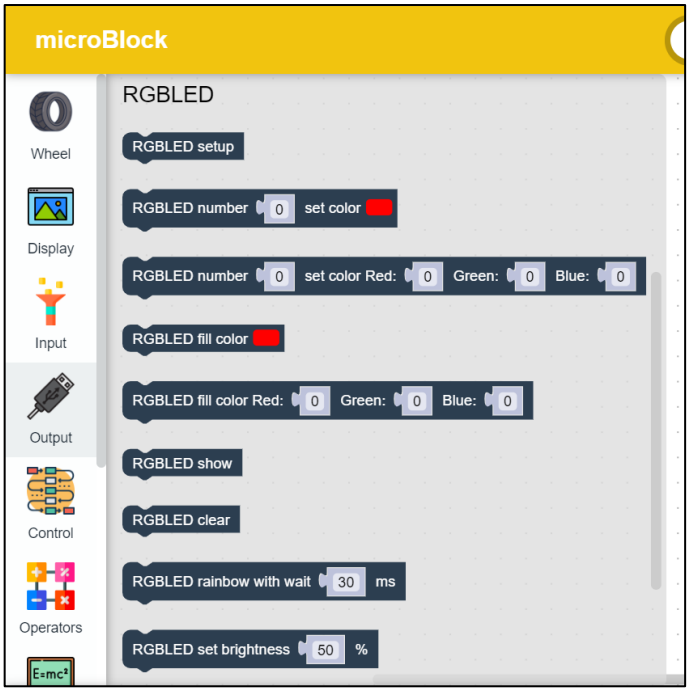
เมื่ออัพโหลดโปรแกรมลงบอร์ด ผลที่ได้ ไฟหน้าซ้ายและไฟหน้าขวากระพริบสลับกันทุก ๆ 0.5 วินาที

การสั่งงานหลอดแอลอีดี RGB

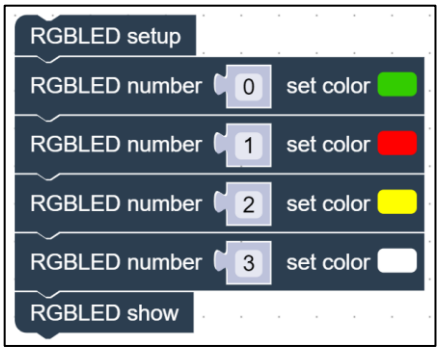
หลอดแอลอีดี RGB ใต้รถ 4 ดวง สามารถสั่งงานแต่ละดวงให้ติดเป็นสีที่ต้องการได้อิสระ เหมาะสำหรับใช้แสดงผลการทำงานของโปรแกรม หรือแสดงแสงสีเพื่อความสวยงาม



การเขียนโปรแกรมสั่งงานหลอด LEDRGB ให้ใช้บล็อกในเมนู Output หัวข้อ RGBLED โดยการใช้งานจำเป็นต้องเรียกใช้บล็อก RGBLED setup ก่อนเสมอ และหลังใช้บล็อก fill หรือบล็อก set ต้องเรียกใช้บล็อก show เพื่อให้การกำหนดสีมีผล



ตัวอย่างโปรแกรม ให้หลอดแอลอีดี RGB แสดงผลเป็นสี เขียว แดง เหลือง ขาว เขียนโปรแกรมได้ดังนี้



หลักการทำงาน มีดังนี้

- บล็อก RGBLED setup – เริ่มต้นใช้งานหลอดแอลอีดี RGB
- RGBLED number 0 set color ... – สั่งให้ไฟดวงที่ 0 ติดเป็นสีเขียว (ดวงขวาหน้า)
- RGBLED number 1 set color ... – สั่งให้ไฟดวงที่ 1 ติดเป็นสีแดง (ดวงซ้ายหน้า)
- RGBLED number 2 set color ... – สั่งให้ไฟดวงที่ 2 ติดเป็นสีเหลือง (ดวงซ้ายหลัง)
- RGBLED number 3 set color ... – สั่งให้ไฟดวงที่ 3 ติดเป็นสีขาว (ดวงขวาหลัง)
- RGBLED show – สั่งให้แสดงผลสีตามที่กำหนดก่อนหน้านี้

ผลที่ได้ หลอดแอลอีดี RGB ติดสว่างตามสีที่กำหนด

บริษัท อาร์ทรอน ชอป จำกัด รับพัฒนาสินค้าอิเล็กทรอนิกส์ บอร์ดอิเล็กทรอนิกส์ ด้านระบบ IoT พัฒนาเว็บไซต์ระบบ IoT ด้วย ReactJS / Next.js รับทำระบบหลังบ้านให้อุปกรณ์ IoT รับเขียนเฟิร์มแวร์อุปกรณ์ IoT ด้วย Arduino IDE / PlatformIO / Atmel Studio / MPLAB IDE / STM32CubeIDE / IAR / ESP-IDF / Arm Keil รับจัดหาชิ้นส่วนอิเล็กทรอนิกส์ ออกใบเสนอราคา และใบกำกับภาษีได้

37/145 ซ.โรงเรียนสวนกุหลาบนนทบุรี ถ.ติวานนท์ ตำบลปากเกร็ด อำเภอปากเกร็ด จังหวัดนนทบุรี 11120

ติดต่อสอบถามข้อมูลเพิ่มเติม


www.artronshop.co.th


02 003 3688 (สำนักงาน)


contact@atronshop.co.th

@artronshop


ArtronShop


ArtronShop