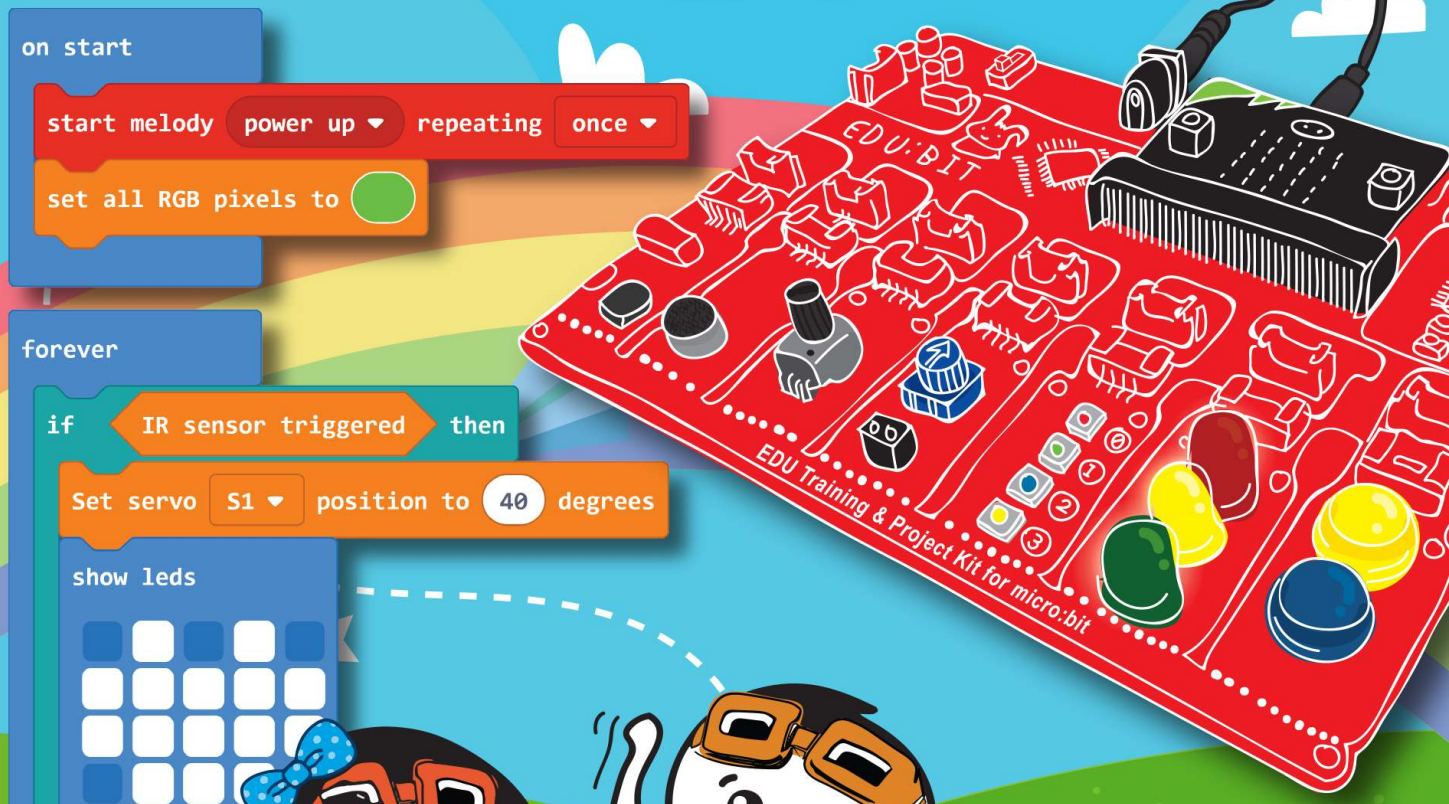
เวอร์ชันภาษาไทย v1.0

Cytron Technologies

CLASSROOM STEAM EDUCATION

Explore STEM & Coding with

EDU:BIT



จดหมายจาก
RERO EDUTEAM @ CYTRON

ถึง _____
(ชื่อเด็กๆ)

คุณเคยได้ยินคำว่า micro:bit บ้างรึเปล่า? มันคือบอร์ดที่สามารถเขียนโปรแกรมเพื่อควบคุมการทำงานได้ซึ่งถูกออกแบบขึ้นที่ประเทศอังกฤษ และมีการจัดจำหน่ายในหลากหลายประเทศทั่วโลกเพื่อที่จะสนับสนุนให้เด็กๆเรียนรู้การเขียนโปรแกรมด้วยความสนุกสนานและง่ายดาย

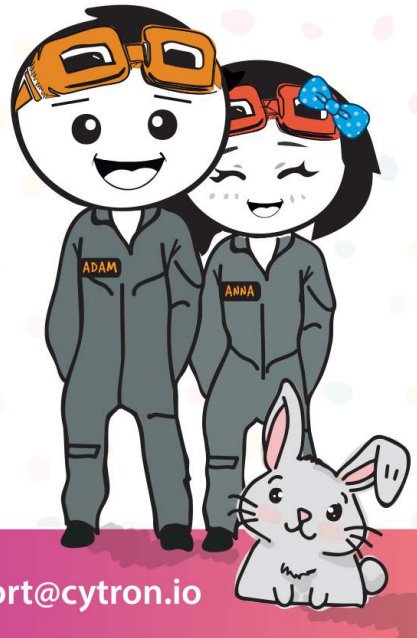
วิศวกรของ Cytron ได้ยกระดับขึ้นอีกขั้นด้วยการสร้างบอร์ด EDU:BIT คุณจะสามารถเรียนรู้การเขียนโปรแกรมได้อย่างเป็นระบบ ซึ่งบนบอร์ดคุณจะได้รับ Music Bit ซึ่งจะประกอบด้วยบัซเซอร์และพอร์ทต่อสำหรับลำโพงภายนอก, Sound Bit สำหรับตรวจจับเสียง, Potentio Bit สำหรับการควบคุมแบบอนาล็อก, IR Bit สำหรับตรวจจับวัตถุ, RGB Bit สำหรับการแสดงแสงสีต่างๆ, Traffic Light Bit ซึ่งประกอบไปด้วย LED สีแดง สีเหลือง และสีเขียว และสุดท้าย Button Bit ปุ่มสำหรับกดบน micro:bit ที่มีขนาดใหญ่ขึ้น อีกทั้งภายในชุดนี้คุณจะยังได้รับ DC Motor และ Servo Motor ไปอีกด้วย เจ๋งมั้ยล่ะ!

มาเริ่มกันเลยดีมั้ย? ซึ่งภายในหนังสือเล่มนี้ คุณจะได้เรียนรู้และสร้างเกมคลาสิกในวัยเด็กที่คุณคุ้นเคย ยกตัวอย่างเช่นเกม เป่า ยิ้ง ฉุบ, เกมบันไดงู, เกมไล่จับ, เสียงปรบมือของใครดังกว่ากันนะ?, ทวิสเตอร์, ไซม่อนสั่งให้.... และเกมสนุกๆอื่นๆอีกมากมาย โดยคุณสามารถทำตามขั้นตอนแบบ step-by-step เพื่อสร้างเกมแสนสนุกไว้เล่นกับเพื่อนของคุณเอง อีกทั้งคุณยังสามารถดัดแปลงส่วนต่างๆของโปรแกรมได้อย่างอิสระ เพื่อสร้างเกมเวอร์ชั่นอัพเกรดด้วยตัวของคุณเองอีกด้วย

ที่หน้าสุดท้ายของทุกบท จะมีแบบทดสอบสนุกๆ ซึ่งจะต้องอาศัยความรู้ที่คุณได้เรียนรู้มาเพื่อที่จะสร้างโปรแกรมสำหรับการเรียนรู้ของคุณ ลองทำดูนะ หากคุณมีปัญหาอะไร เราอยู่ตรงนี้พร้อมช่วยคุณเสมอ

พร้อมรึยังล่ะ? มาเริ่มต้นการเดินทางอันน่าตื่นเต้นนี้กันเถอะ
ขอให้สนุกกับการเรียนรู้และทดลองสิ่งใหม่ๆนะ

เชียร์
อดัมและแอนนา



เรียนรู้ STEM และการเขียนโปรแกรม ด้วย EDU:BIT ผ่านแบบฝึกหัดและชุดโปรเจ็กต์

เขียน
Cheryl Ng, SC Lim & Adrian Teo

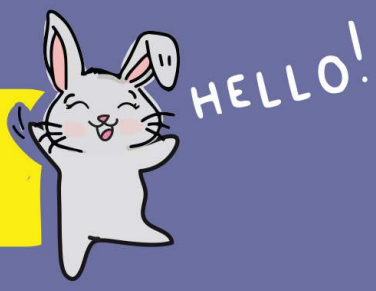
ทดลองใช้งานและรับรอง
Joshayne D. Lim (อายุ 7 ปี)

ภาพประกอบโดย
Suhana Oazmi

พิมพ์ครั้งที่ 2 กันยายน 2563

เผยแพร่โดย Cytron Technologies

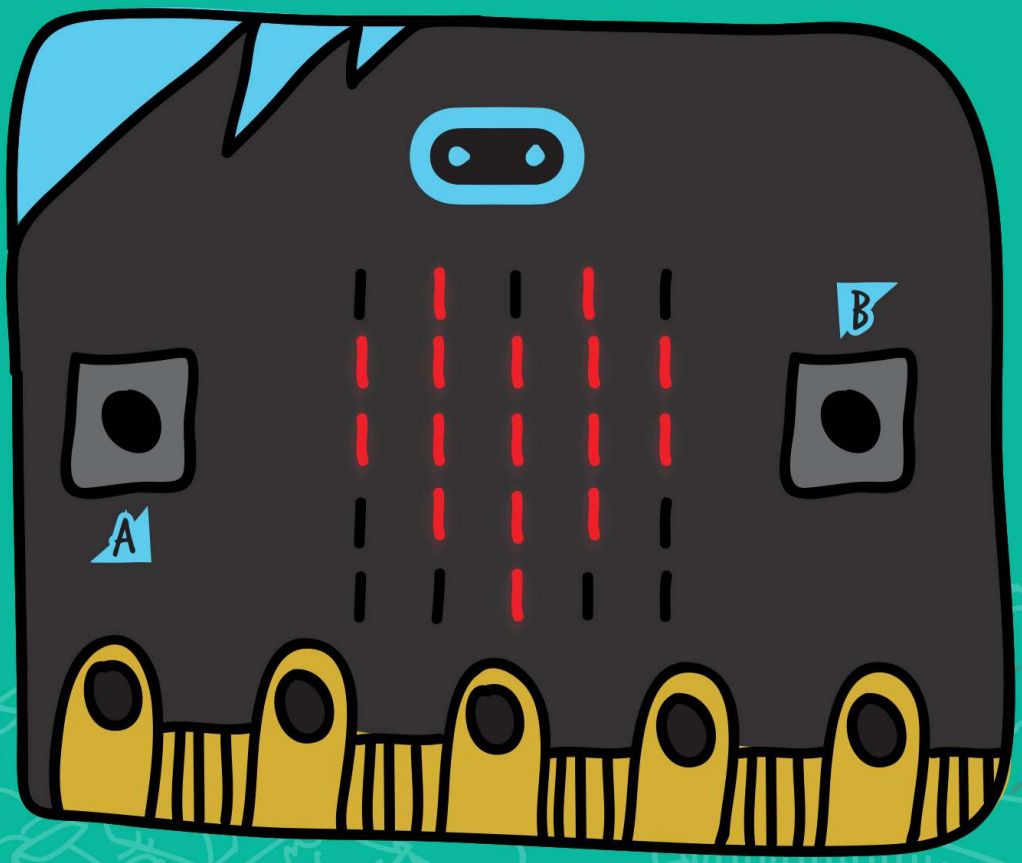
สารบัญ



บทที่ 1

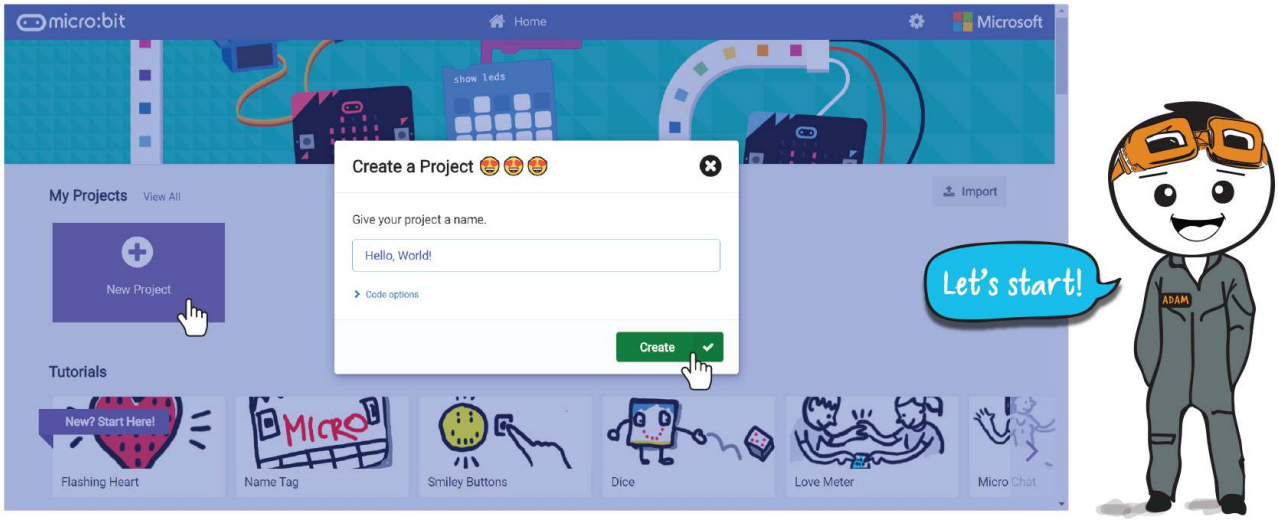
>Hello, World!

LED Matrix บน micro:bit สวัสดีชาวโลก!

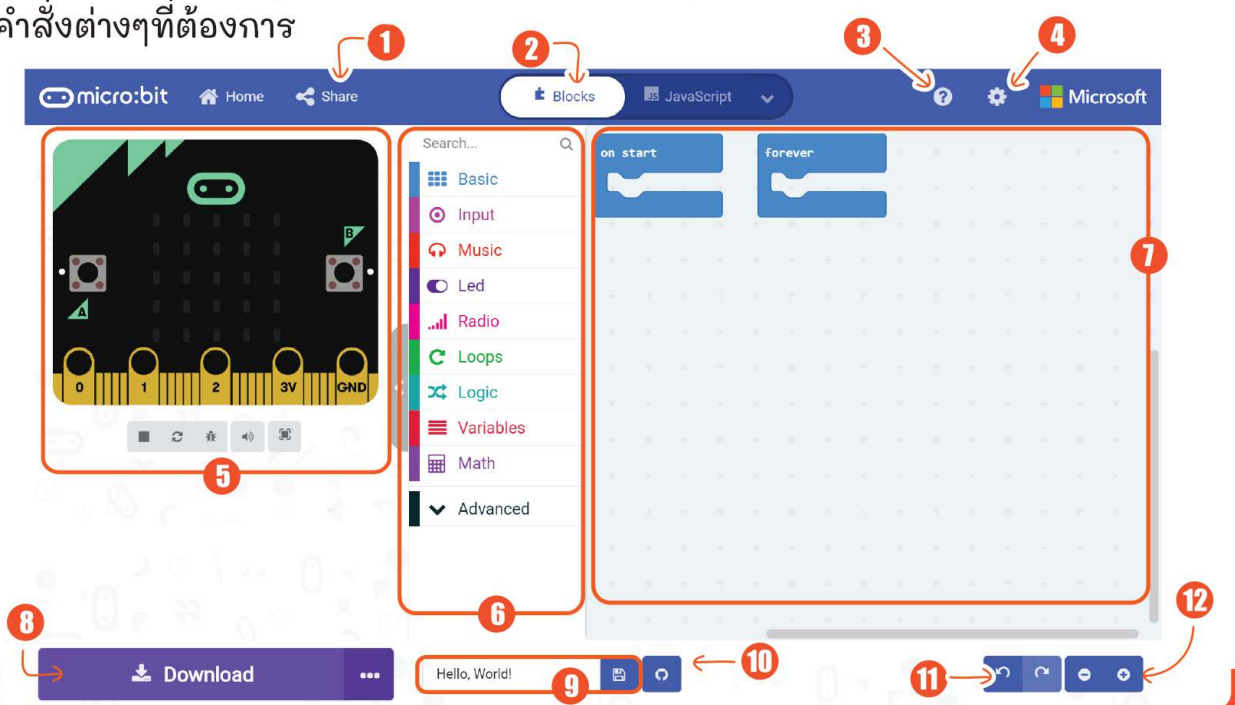


มาเขียนโปรแกรมกัน!

ขั้นตอนที่ 1 เปิดเว็บเบราเซอร์ของคุณขึ้นมาแล้วไปที่ https://makercode.microbit.org/
คลิกที่ “New Project” ตั้งชื่อโปรเจ็กต์ของคุณแล้วกด “Create”

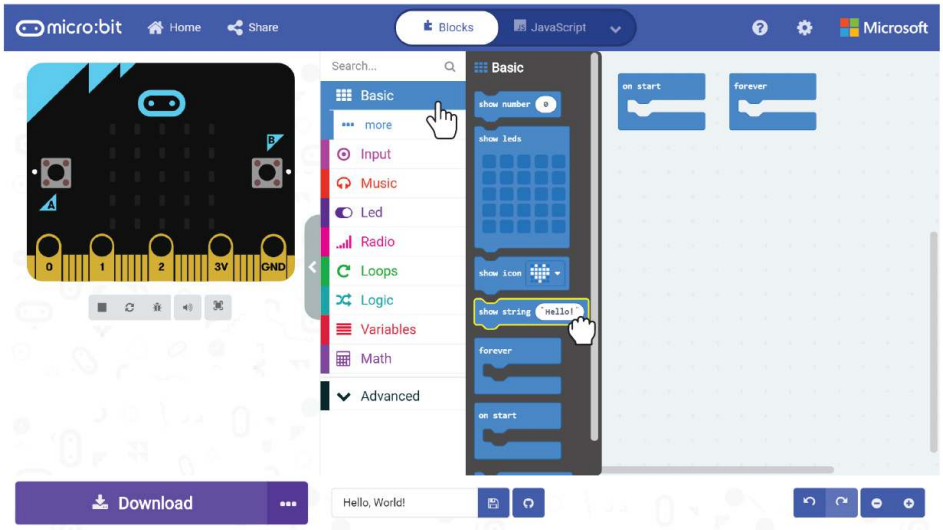


คุณจะเห็นหน้าต่าง Microsoft MakeCode Editor ซึ่งจะช่วยให้คุณสามารถเขียนโปรแกรมเพื่อควบคุมการทำงาน EDU:BIT ของคุณได้อย่างง่ายดาย โดยการลากและวางคำสั่งต่างๆที่ต้องการ

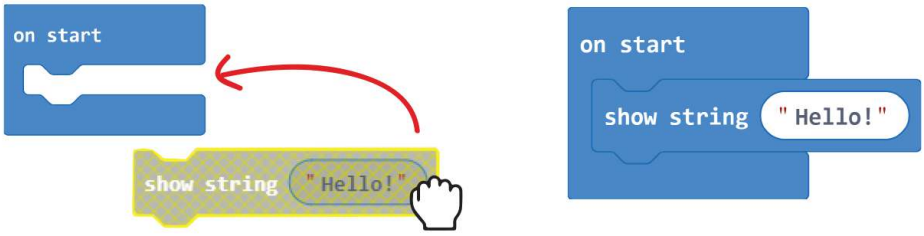


1.) เผยแพร่และแชร์โปรเจ็กต์ของคุณ
2.) ตัวเลือกสำหรับรูปแบบการเขียนโปรแกรมแบบ Block, JavaScript หรือ Python
3.) เปิดเมนูสำหรับช่วยเหลือ
4.) เปลี่ยนแปลงการตั้งค่า, เพิ่มเติมส่วนขยาย และจับคู่อุปกรณ์
5.) **SIMULATOR** - จำลองลักษณะการทำงานของโปรแกรมของคุณบน micro:bit จำลอง
6.) **TOOLBOX / CATEGORY DRAWER** - ส่วนที่รวบรวม Block สำหรับเขียนโปรแกรมประเภทต่างๆ โดยที่ Block ที่เป็นประเภทเดียวกันจะมีสีเหมือนกัน
7.) **PROGRAMMING WORKSPACE** - โปรแกรมถูกเขียนขึ้นโดยการนำ Block มาต่อเข้าด้วยกันบริเวณนี้
8.) ดาวน์โหลดโปรแกรมของคุณลง micro:bit
9.) ตั้งชื่อและบันทึกโปรเจ็กต์ของคุณลงบนคอมพิวเตอร์
10.) สร้าง Github repository
11.) ยกเลิกคำสั่ง,ทำซ้ำ
12.) ซูมเข้า/ซูมออก

ขั้นตอนที่ 2 ที่ Toolbox ด้านซ้าย เลือก [Basic] และเลือก [show string] block

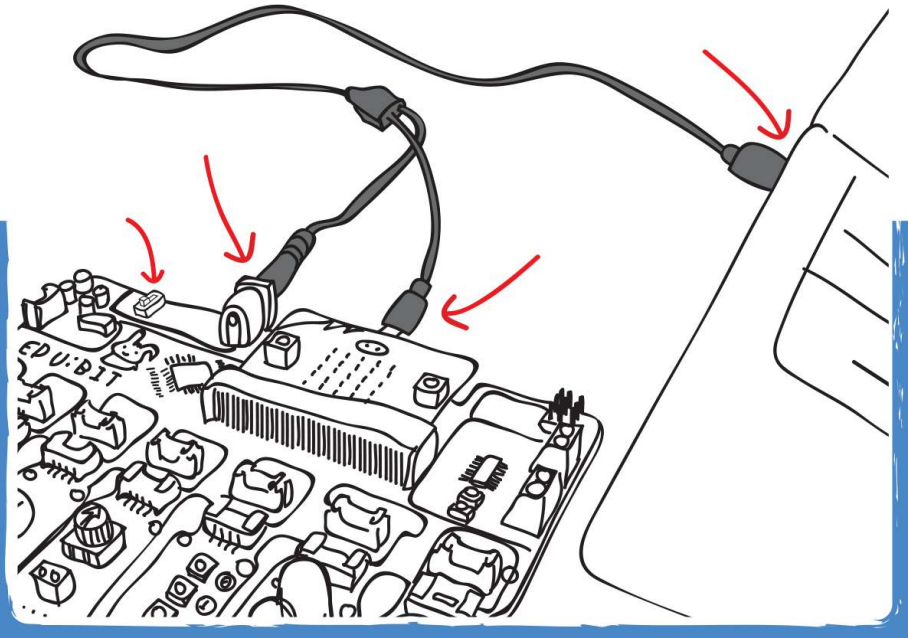


ขั้นตอนที่ 3 เลือก [show string] block แล้วนำไปวางที่ slot [on start]

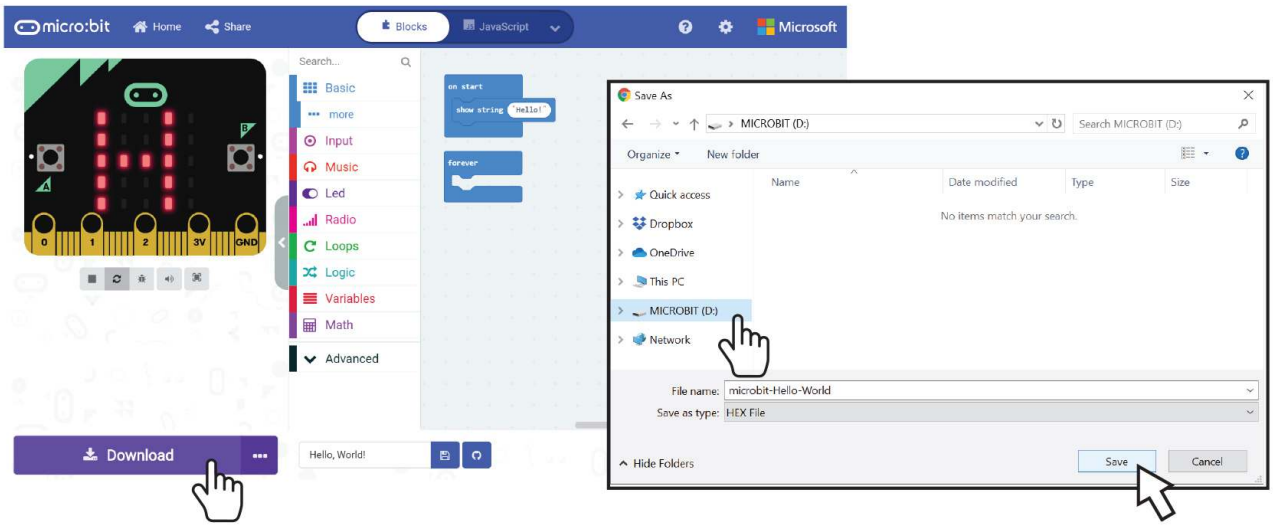


ขั้นตอนที่ 4 เชื่อมต่อสาย USB ระหว่าง EDU:BIT กับคอมพิวเตอร์ ดังภาพ

อย่าลืม! เปิดการทำงานของ EDU:BIT โดยการสไลด์สวิตซ์บน EDU:BIT ไปที่ ON



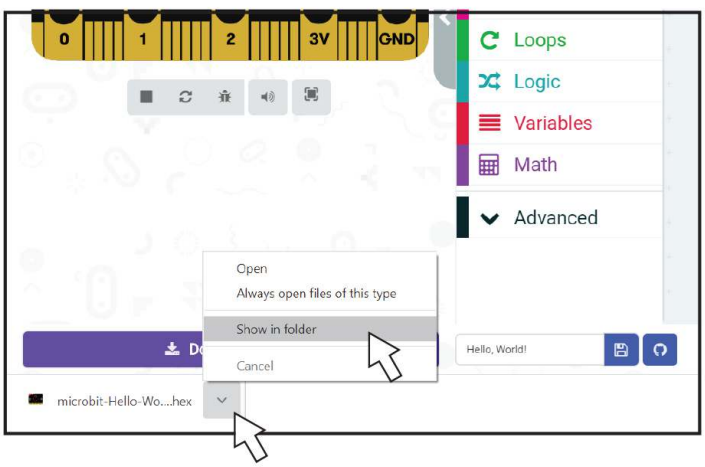
ขั้นตอนที่ 5 กดปุ่ม **Download** เมื่อหน้าต่างป๊อป-อัพปรากฏขึ้นมา ทำการเลือกดาวน์โหลดไฟล์ไว้ที่ไดรฟ์ MICROBIT ปิดหน้าต่างเมื่อการดาวน์โหลดเสร็จสิ้น

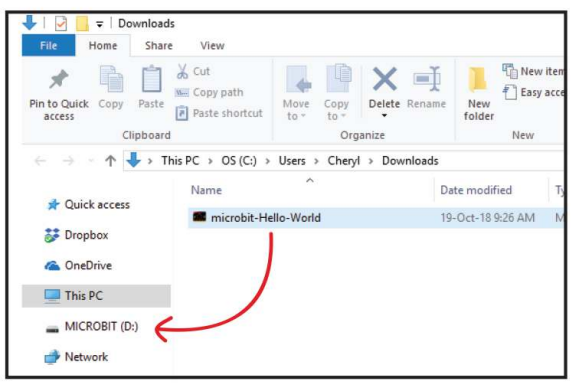


ขั้นตอนการอัพโหลดโปรแกรมเข้าสู่อุปกรณ์ เรียกว่า **"การแฟลช"** โดยหลอด LED สีส้มด้านหลังของ micro:bit จะกระพริบระหว่างการโอนถ่ายโปรแกรมเข้าสู่ตัวบอร์ด โดยโปรแกรมจะทำงานโดยอัตโนมัติเมื่อโอนถ่ายโปรแกรมเสร็จสิ้น

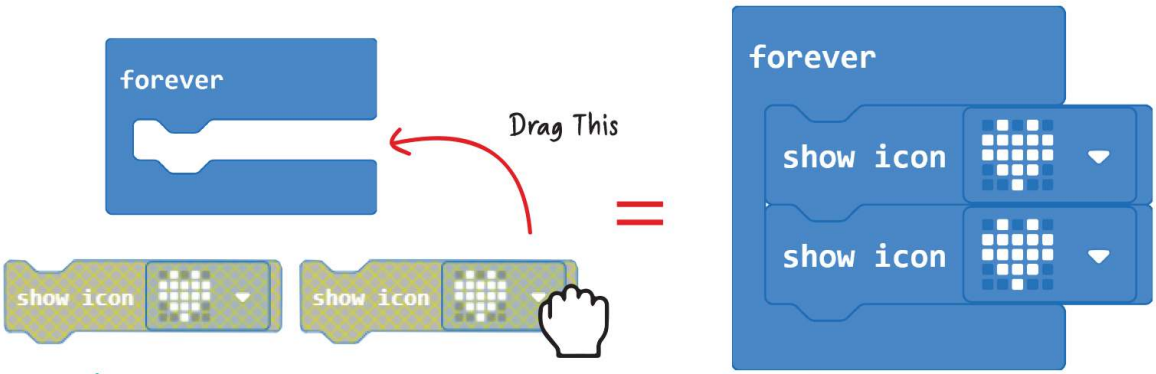
NOTE!

ถ้าหากหน้าต่าง ป๊อป-อัพไม่ปรากฏขึ้นมา หมายความว่าไฟล์ถูกดาวน์โหลดโดยอัตโนมัติไปที่ตำแหน่งที่เบราเซอร์ได้ตั้งค่าไว้ ทำการคลิกขวาที่ไฟล์โปรแกรม .hex ที่เราได้ทำการดาวน์โหลดมาที่ด้านล่างของหน้าต่างเบราเซอร์ เลือก "Show in folder" คลิกและลากไฟล์โปรแกรม "microbit-xxxx.hex" แล้วนำไปวางที่ไดร์ฟ MICROBIT เหมือนกับการคัดลอกไฟล์ลงแฟลชไดร์ฟตามปกติ

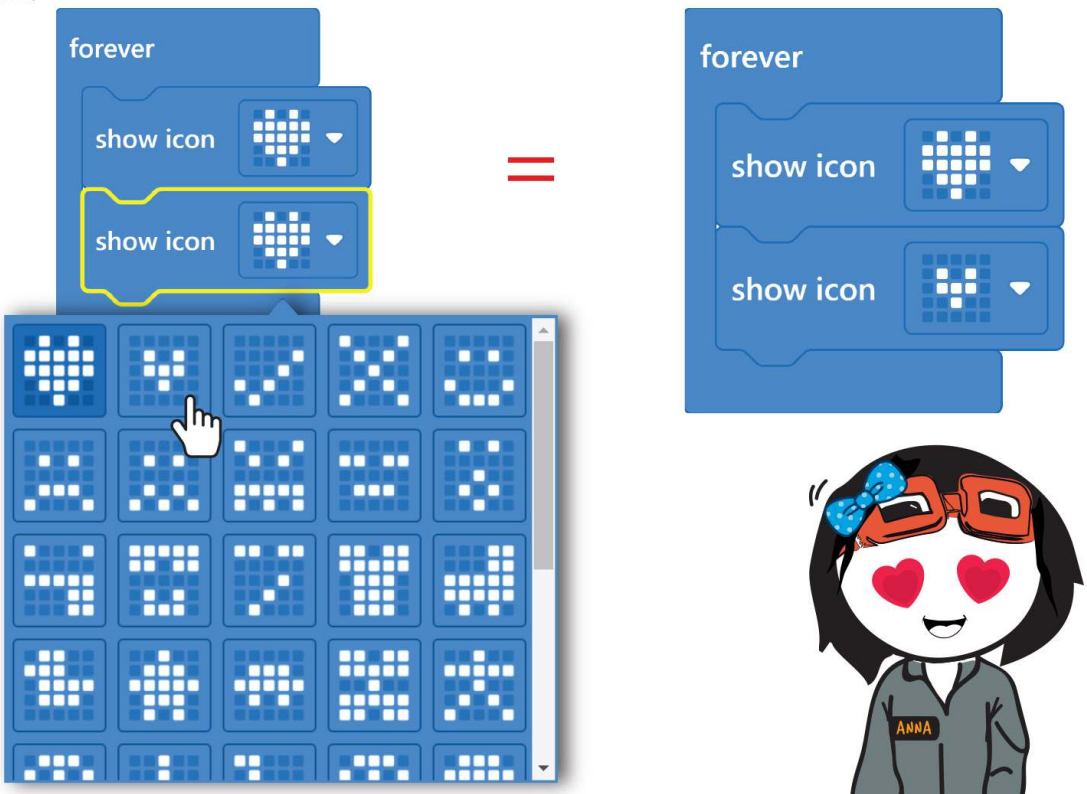




ขั้นตอนที่ 6 คลิก [Basic] ที่ Toolbox และเลือก [show icon] block ทำซ้ำจนได้จำนวน block [show icon] ที่ต้องการแล้วจึงนำไปใส่ใน [forever] slot

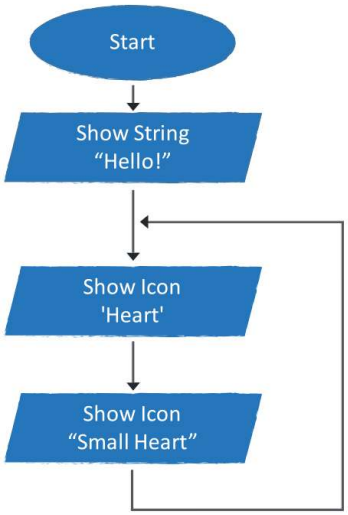


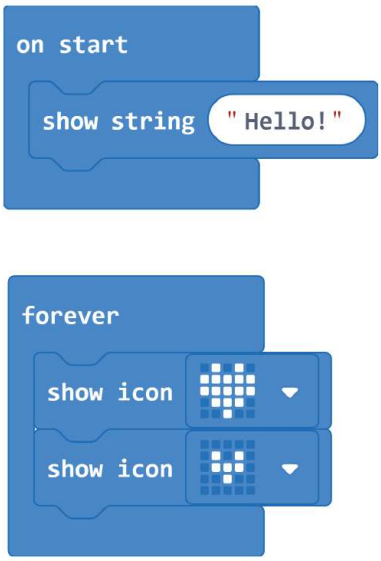
ขั้นตอนที่ 7 คลิกซ้ายที่ไอคอนของ [show icon] block ที่ 2 แล้วทำการเลือกรูป "หัวใจดวงเล็ก" จาก หน้าต่างป๊อป-อัพที่ปรากฏขึ้นมา หลังจากนั้นทำการแฟลชโปรแกรมลง EDU:BIT



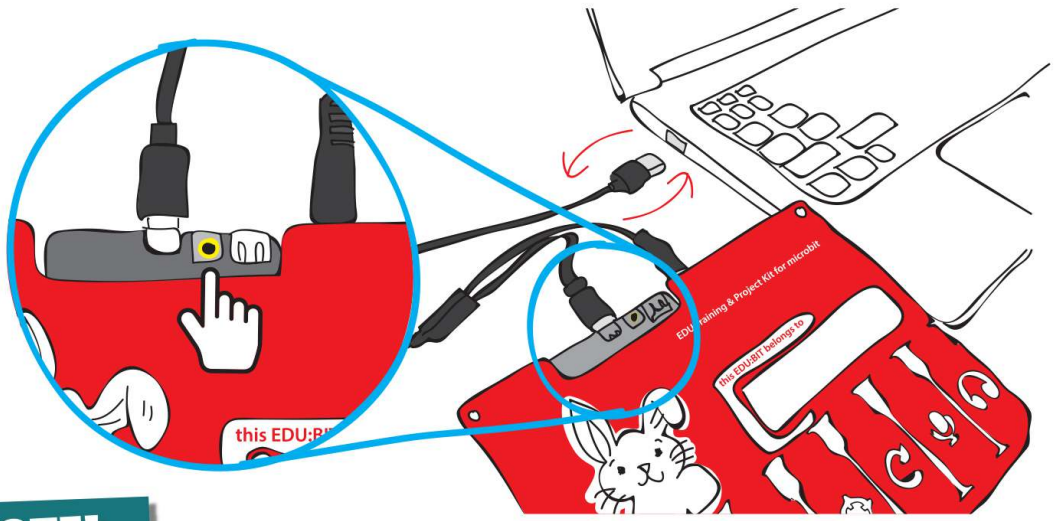
BREAK THE CODE

คุณสังเกตเห็นไหมว่า
คำว่า “Hello” จะปรากฏบนจอเพียงแค่ครั้งเดียว
แต่ภาพหัวใจเต้นจะแสดงวนไปเรื่อยๆ ทำไมล่ะ?





block คำสั่งที่อยู่ใน [on start] slot จะทำงานเพียงครั้งแรกที่บอร์ดเริ่มทำงาน และ blockคำสั่งที่อยู่ [forever] slot จะทำงานซ้ำไปเรื่อยๆ

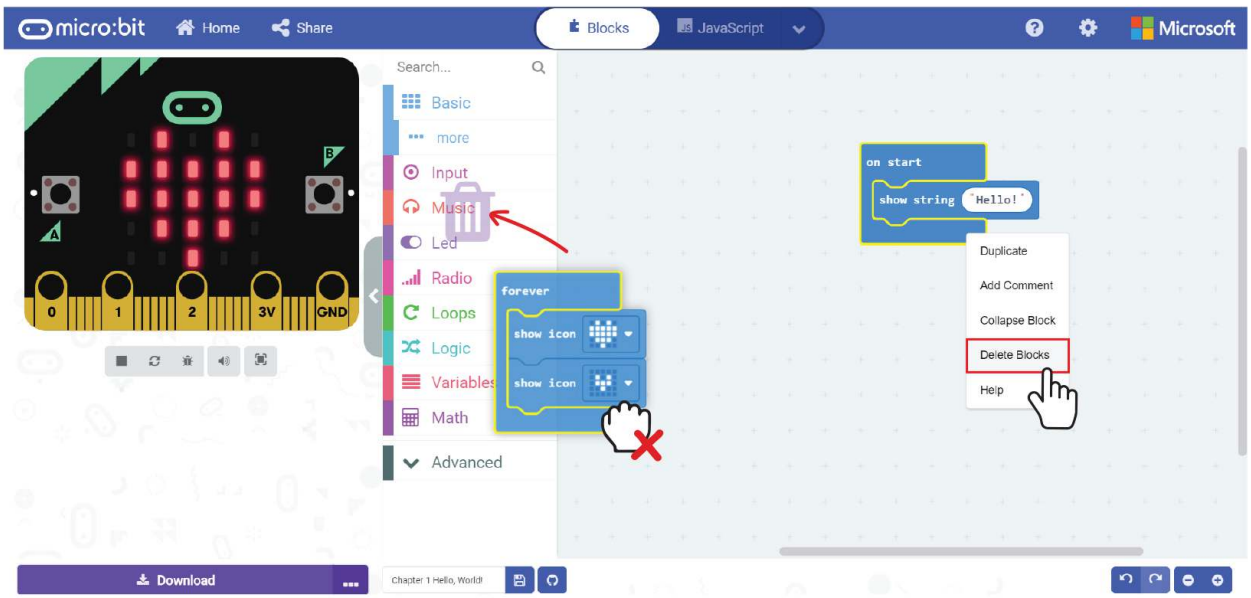


NOTE!

ถ้าคุณต้องการที่จะเริ่มต้นโปรแกรมทั้งหมดใหม่ สามารถทำได้โดยการกดปุ่ม Reset หรือทำการทำการถอดสาย USB ออกแล้วเสียบใหม่เพื่อเป็นการรีเซ็ตบอร์ด

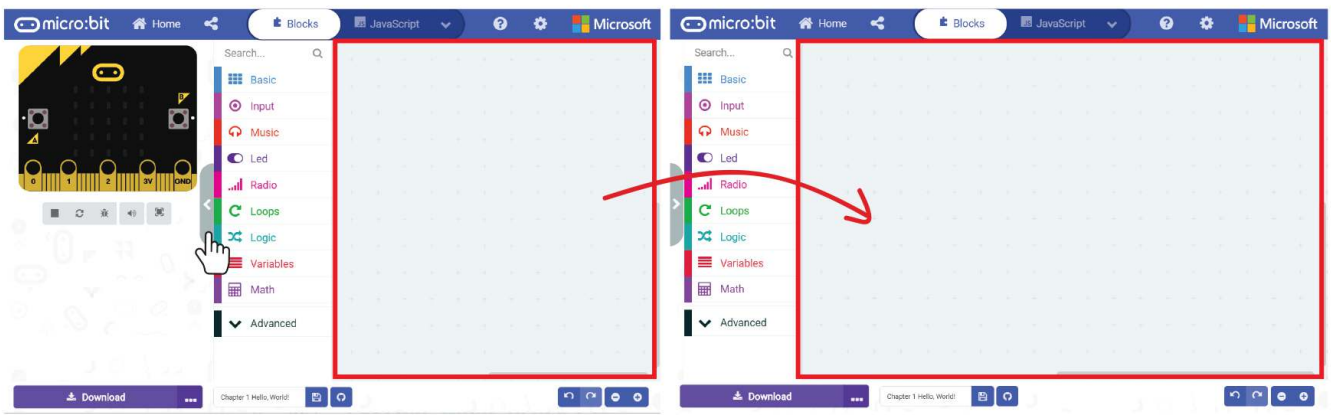
เคล็ดลับ #1!

คุณสามารถลบ block ที่ไม่ต้องการได้โดยการคลิกที่ block ที่ต้องการลบและลากไปวางบริเวณ Toolbox ซึ่งจะปรากฏไอคอนรูปถังขยะ หรืออีกวิธีสามารถทำได้โดยการคลิกขวาที่ block ที่ต้องการลบและเลือก “Delete Block”



เคล็ดลับ #2!

ถ้าคุณไม่ได้ใช้ micro:bit simulator ทางด้านซ้าย สามารถปิด tab simulator เพื่อเพิ่มพื้นที่ในการเขียน block โปรแกรมได้

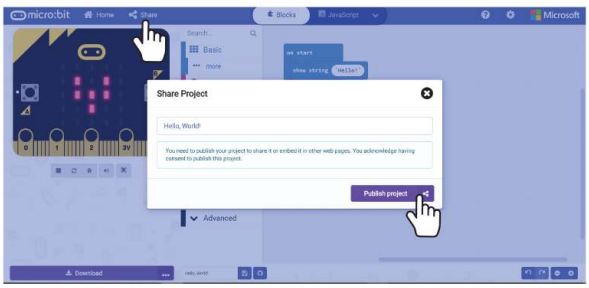


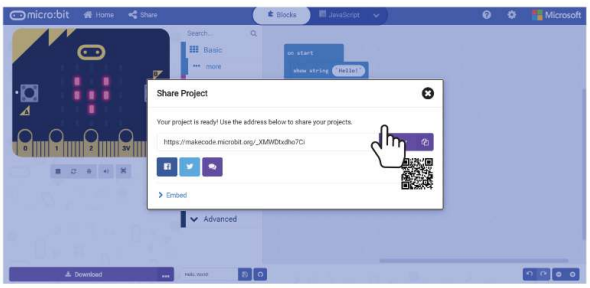
เคล็ดลับ #3!

คุณสามารถรวมกลุ่ม block คำสั่งได้โดยการคลิกขวาที่กลุ่ม block คำสั่งที่ต้องการรวมกลุ่ม และเลือก “Collapse Block” สำหรับการแตกกลุ่ม block คำสั่งสามารถทำได้เพียงกดที่สัญลักษณ์

เคล็ดลับ #4!

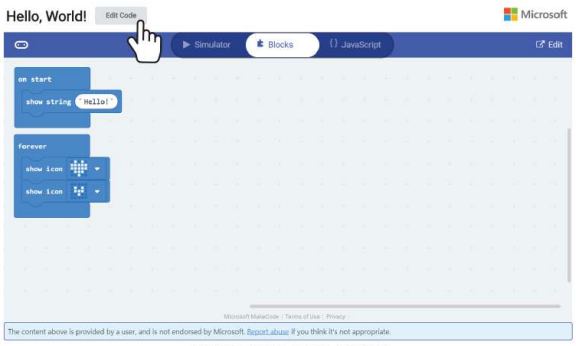
คุณสามารถแบ่งปันโปรแกรมของคุณกับเพื่อนหรือครูได้โดยการคลิก [**Share**] จะปรากฏหน้าต่างป๊อป-อัพขึ้นมาให้เลือก [**Publish project**] หลังจากนั้นจะปรากฏอีกหน้าต่างป๊อป-อัพพร้อมลิงค์โปรเจ็กต์ของคุณ คลิก [**Copy**] เพื่อคัดลอกลิงก์สำหรับแบ่งปันกับเพื่อนของคุณ





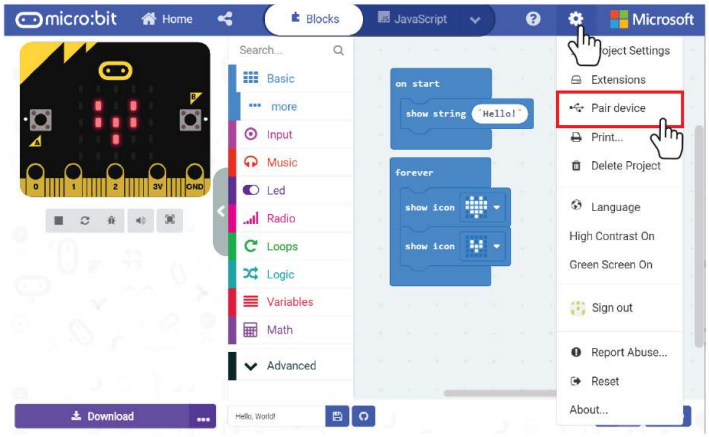
เคล็ดลับ #5!

คุณครูหรือเพื่อนของคุณจะเห็นโปรแกรมของคุณเมื่อเปิดลิงค์โปรเจ็กต์ในเบราเซอร์โดยที่จะสามารถดูและแก้ไขโปรแกรมได้โดยคลิกที่ปุ่ม [**Edit Code**]



เคล็ดลับ #6!

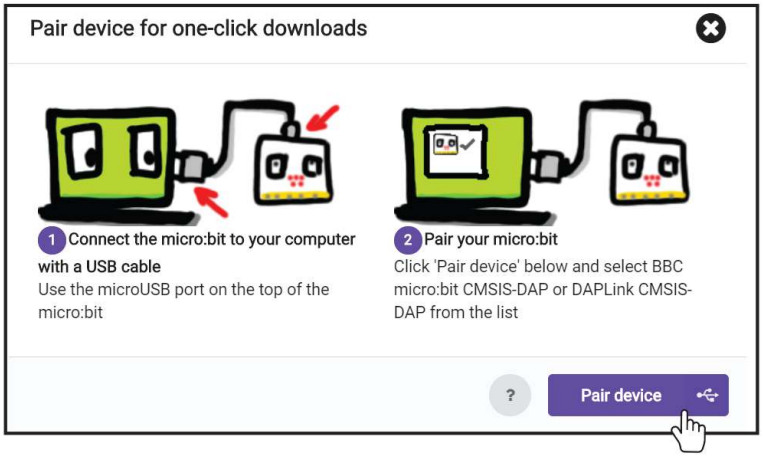
รู้หรือไม่ คุณสามารถจับคู่อุปกรณ์ได้โดยการคลิกเพียงหนึ่งครั้ง โดยการกดที่ไอคอนรูปเฟืองทางด้านขวาบนและเลือก “Pair device”



NOTE!

micro:bit ของคุณต้องเป็น firmware เวอร์ชันล่าสุด และต้องใช้เบราเซอร์ Microsoft Edge หรือ Chrome เวอร์ชันล่าสุด หากคุณต้องการอัพเดต firmware ของ micro:bit ของคุณให้เป็นเวอร์ชันล่าสุด สามารถทำได้ตามขั้นตอนในเว็บไซต์ https://microbit.org/get-started/user-guide/firmware/

ทำการเชื่อมต่อ micro:bit ของคุณกับคอมพิวเตอร์แล้วทำการคลิกปุ่ม [**Pair device**] ในหน้าต่างป๊อป-อัพที่ปรากฏขึ้นมา และเลือกอุปกรณ์ที่มีชื่อว่า “BBC micro:bit CMSIS-DAP หรือ “DAPLink CMSIS-DAP” จากรายชื่ออุปกรณ์ที่ปรากฏขึ้นมาแล้วคลิก [**Connect**]

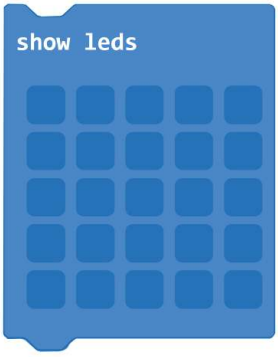


หลังจากทำการเชื่อมต่อกับอุปกรณ์เรียบร้อยแล้ว คุณสามารถแฟลชโปรแกรมเข้าสู่ EDU:BIT ได้โดยตรง โดยการคลิกปุ่ม [**Download**] ลองทำดูสิ!

หากคุณมีปัญหาหากับการจับคู่เชื่อมต่ออุปกรณ์ของคุณ สามารถดูวิธีแก้ปัญหาเพิ่มเติมได้ที่ https://makecode.microbit.org/device/usb/webusb/troubleshoot

มาลองดู Block รูปแบบอื่นกัน

1# คุณสามารถใช้ [show leds] block เพื่อออกแบบไอคอนของคุณเอง และ [show number] block เพื่อแสดงตัวเลข

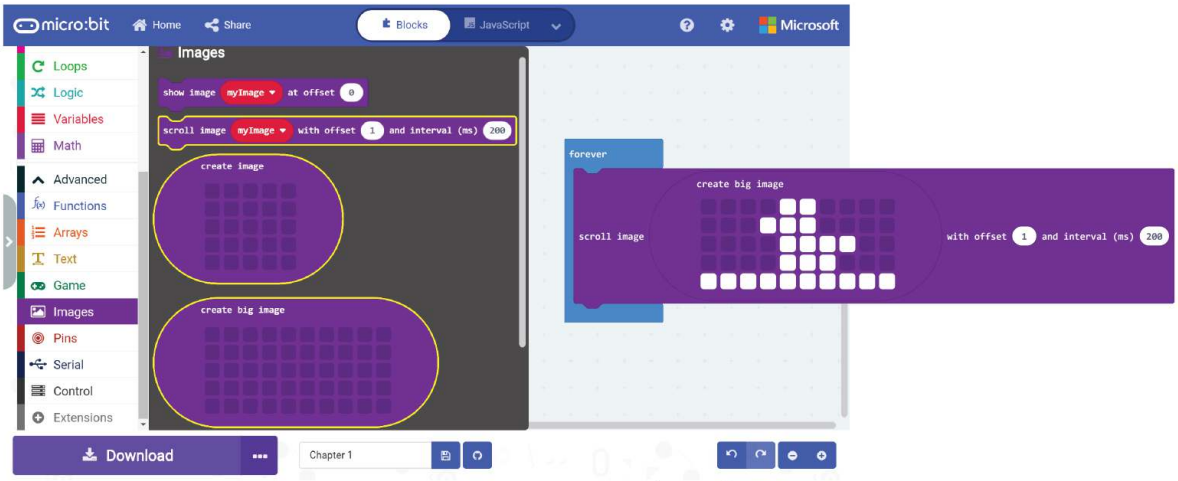


รู้หรือไม่?

1000 มิลลิวินาที (ms) = 1 วินาที

2# คุณสามารถเพิ่ม [pause] block เข้าไปในโปรแกรมเพื่อหน่วงเวลาการทำงานของโปรแกรมได้ ซึ่งฟังก์ชันนี้จะหน่วงเวลาการทำงานของโปรแกรมตามที่คุณตั้งไว้ในหน่วยมิลลิวินาที (ms)

3# การทำให้รูปภาพเลื่อนผ่าน LED matrix สามารถทำได้โดยใช้ [scroll image _ with offset _ and interval (ms) _] block ร่วมกับ [crete image] block หรือ [create big image] block อย่างใดอย่างหนึ่ง จาก [Images] ใน Toolbox (ถัดจาก **Advanced**)



เมื่อคุณเริ่มโปรแกรมจะเห็นเป็ดตัวเล็กๆเคลื่อนที่ผ่านจอ LED matrix ทีละตัว

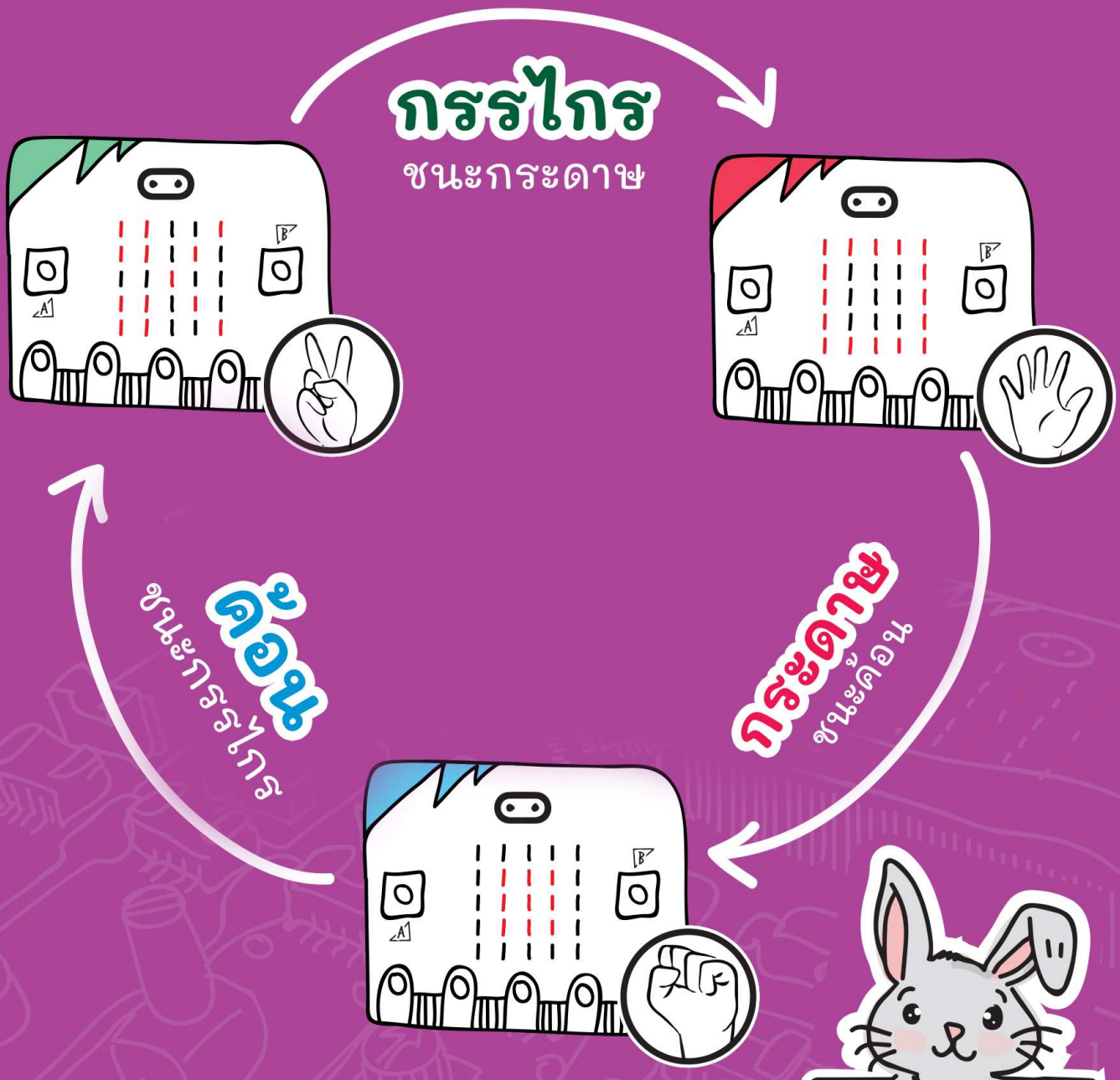
APPLICATION CHALLENGE

เขียนโปรแกรมให้ EDUːBIT ทำงานเป็นป้ายประกาศ	
On start	แสดงภาพเคลื่อนไหวง่ายๆเพื่อดึงดูดความสนใจ และแสดงชื่อห้องเรียนของคุณ
Forever	แสดงวันที่ปัจจุบันและข้อมูลสำคัญอื่นๆของห้องเรียน

บทที่ 2

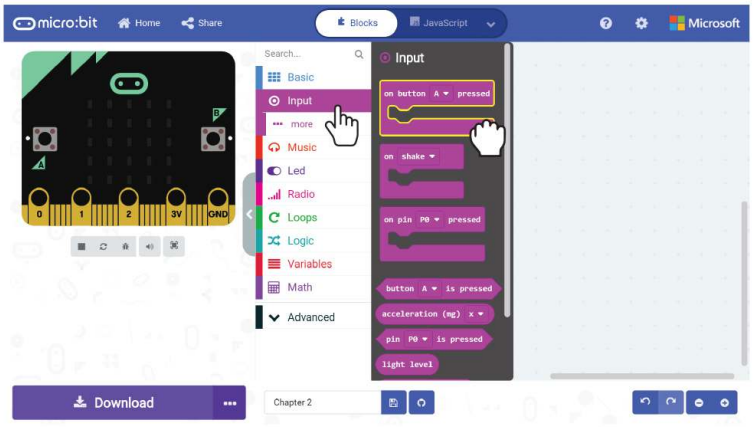
เป่า ยิ้งงงง ฉุบ~เธอออกอะไรหน่ะ

ปุ่มกดบน micro:bit และ Button Bit

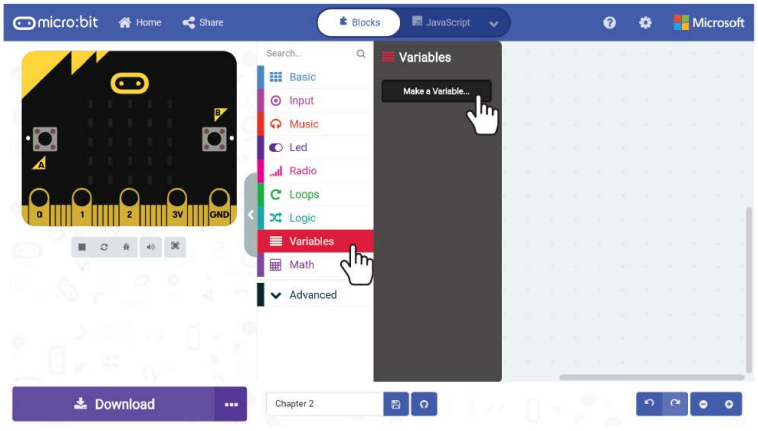


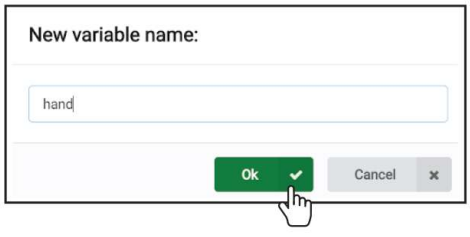
มาเขียนโปรแกรมกัน!

ขั้นตอนที่ 1 ไปที่ https://makecode.microbit.org/ (หรือ คลิก Home หากคุณอยู่ใน MakeCode Editor อยู่แล้ว) สร้างโปรเจ็กต์ใหม่ คลิกเลือก [Input] ใน Toolbox และเลือก [on button _ pressed] block

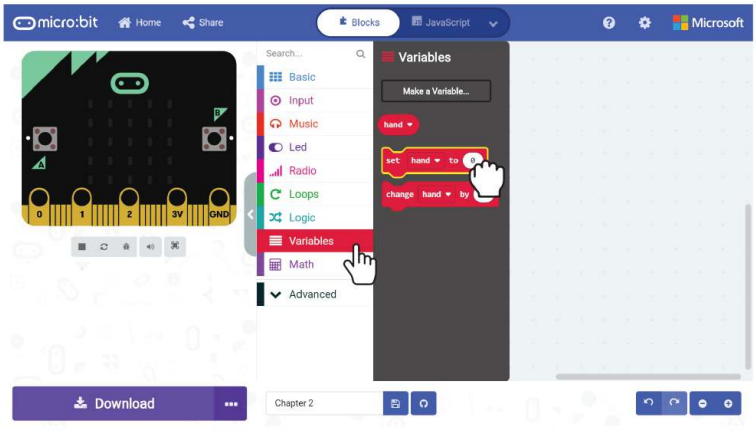


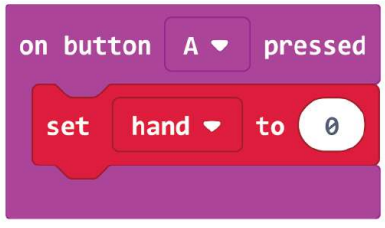
ขั้นตอนที่ 2 ทำการสร้างตัวแปรโดยการ คลิกเลือก [Variable] ใน Toolbox และเลือก [Make a Variable] ตั้งชื่อตัวแปรในหน้าต่างป๊อป-อัพที่ปรากฏขึ้นมาว่า 'hand' และคลิก OK



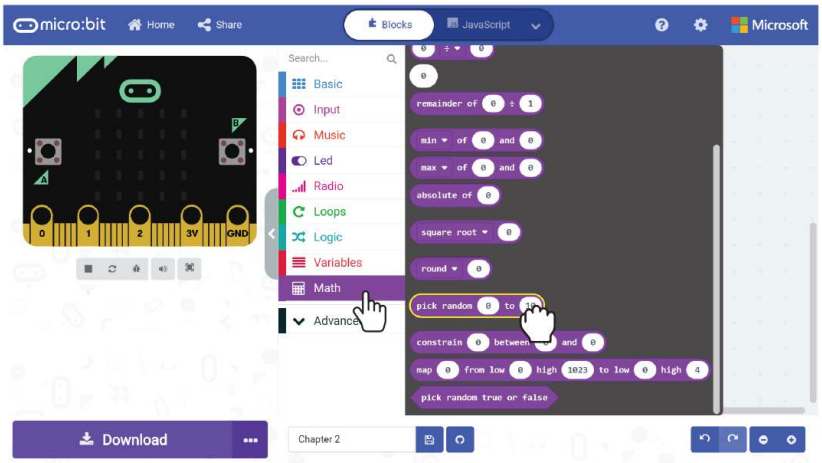


ขั้นตอนที่ 3 ทำการกำหนดค่าตัวแปรโดยการ คลิกเลือก [Variable] ใน Toolbox และเลือก [set _ to _] block แล้วนำไปใส่ใน [on button A pressed] block

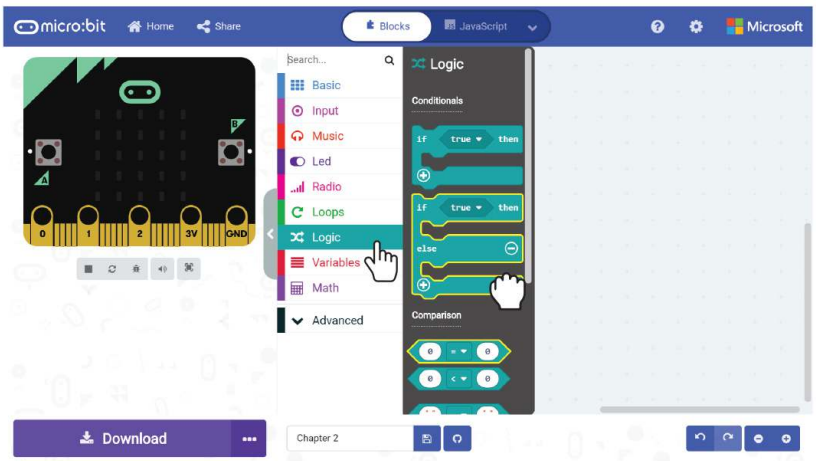


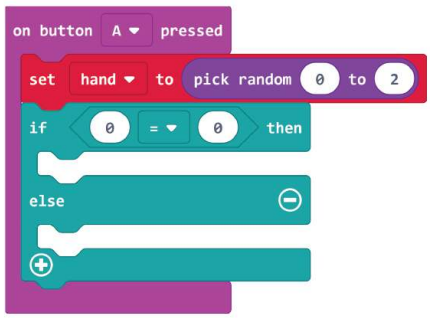


ขั้นตอนที่ 4 คลิกเลือก [Math] ใน Toolbox และเลือก [pick random _ to _] block แล้วทำการเปลี่ยนเลขภายใน block เป็น “10 to 2”

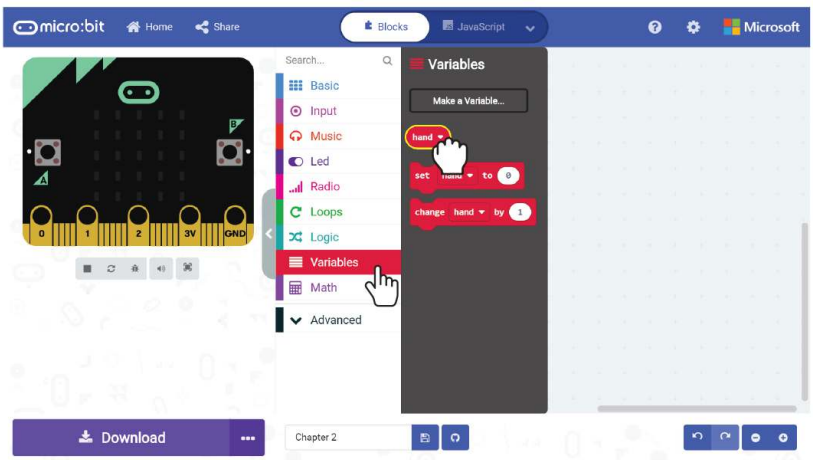


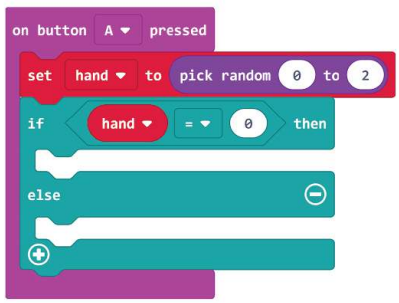
ขั้นตอนที่ 5 คลิกเลือก [Logic] ใน Toolbox และเลือก [if-then-else] block และ [_ = _] block โดยนำ [_ = _] block ไปวางใน ‘if’ slot



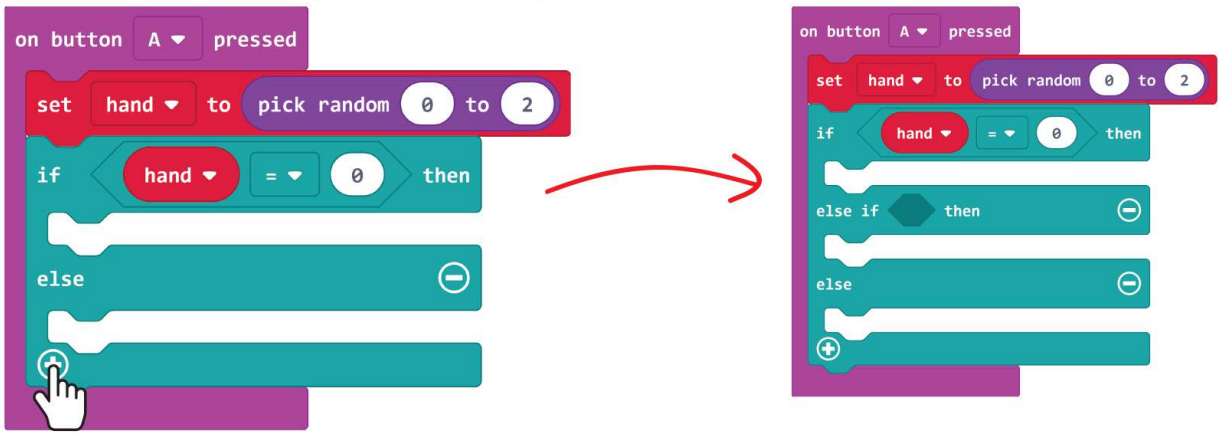


ขั้นตอนที่ 6 คลิกเลือก [Variable] ใน Toolbox และเลือก [hand] block แล้วนำไปวางไว้ใน [_ = _] block ดังภาพ

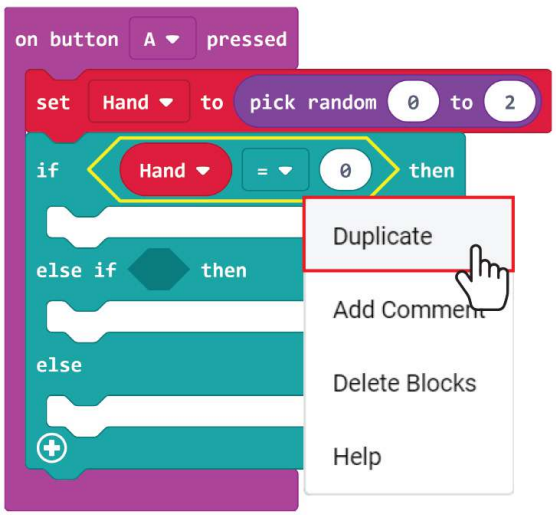




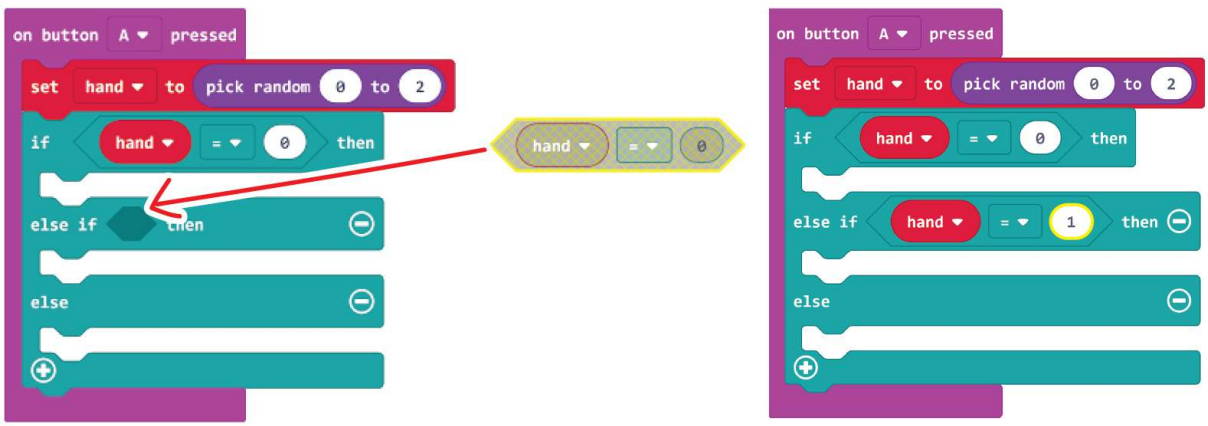
ขั้นตอนที่ 7 คลิกที่เครื่องหมาย ⊕ บน [if-then-else] slot เพื่อทำการเพิ่มเงื่อนไข [else if] เข้าไปใน [if-then-else] slot



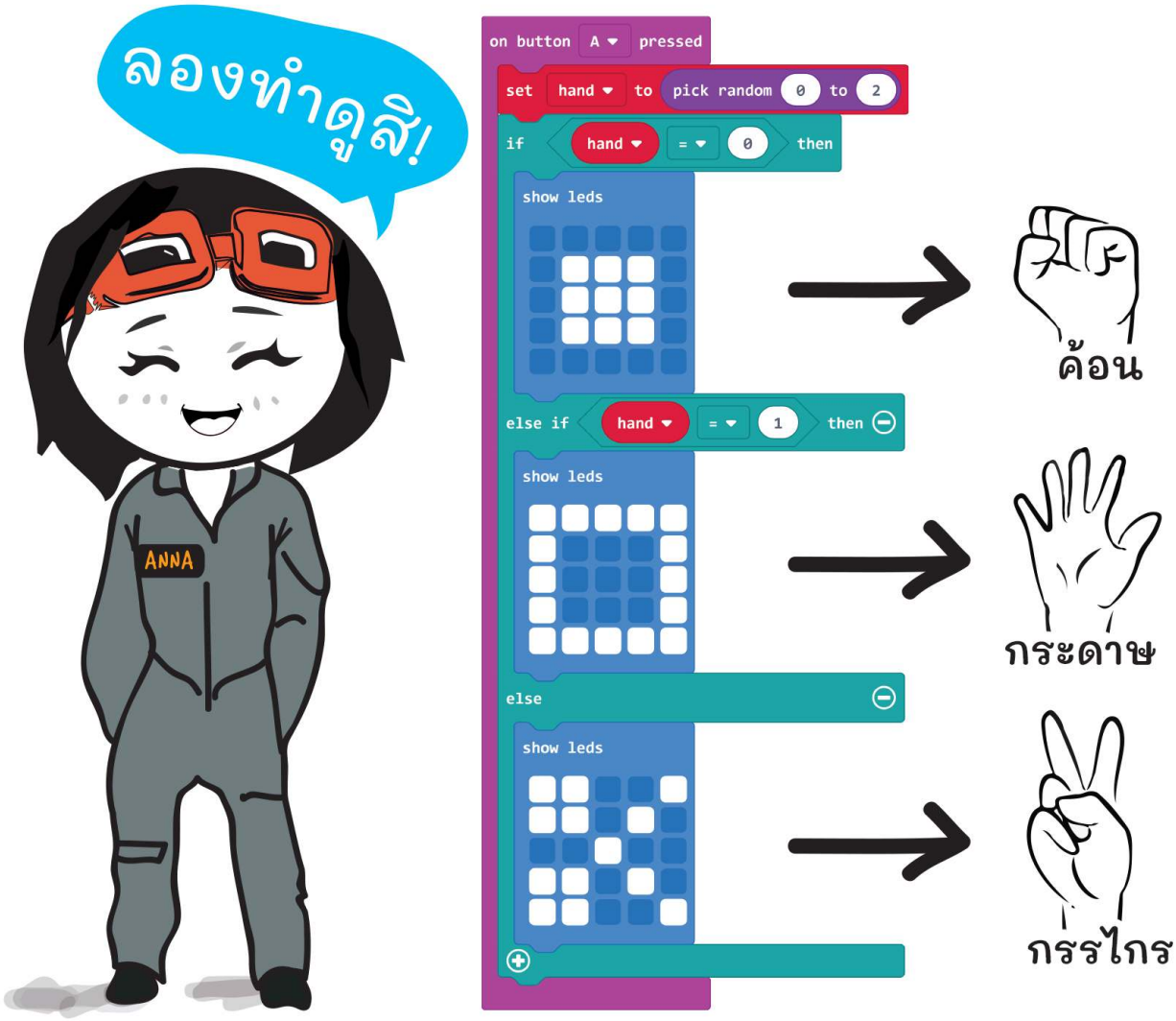
ขั้นตอนที่ 8 คลิกขวาที่ [_ = _] block และเลือก “Duplicate”



ขั้นตอนที่ 9 นำ [_ = _] block ที่ได้จากการ duplicate ไปวางไว้ใน [else if] slot แล้วเปลี่ยนตัวเลขเป็น “0 to 1”

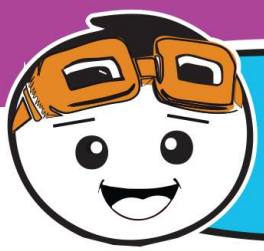


ขั้นตอนที่ 10 เพิ่ม [show leds] block ที่อยู่ใน [Basic] Toolbox ใน ‘if’, ‘else if’ และ ‘else’ slot สร้างรูปด้วย LED โดยการกดช่องสี่เหลี่ยมใน [show leds] block เพื่อสร้างสัญลักษณ์ที่ต้องการตามตัวอย่างด้านล่าง



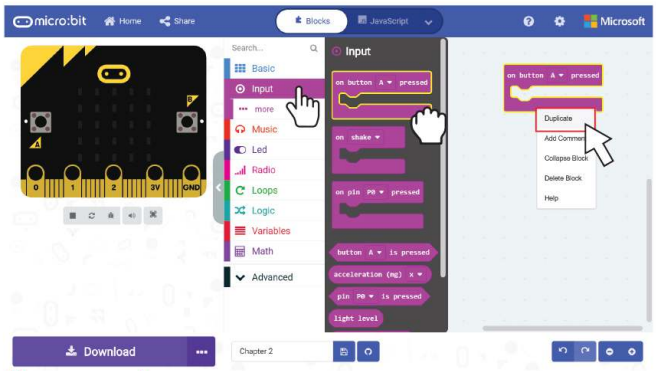
แฟลชโค๊ดใส่ EDU:BIT แล้วเริ่มเล่น เป่า ยิ๊ง ฉุบ กับเพื่อของคุณได้เลย เมื่อไหร่ก็ตามที่คุณกดปุ่ม A บน micro:bit หรือ ปุ่มสีเหลือง LED matrix จะทำการสุ่มว่าคุณจะออก “ค้อน” , “กรรไกร” หรือ “กระดาษ”

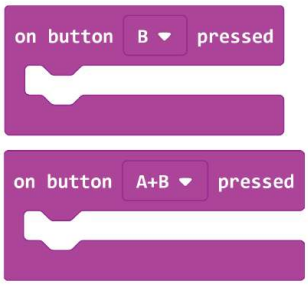
เตือนความจำ : ถ้าคุณต้องการจะบันทึกโปรเจ็กต์นี้เก็บไว้ อย่าลืม! บันทึกลงโฟลเดอร์บนเครื่องของคุณโดยการกดปุ่ม “Save”



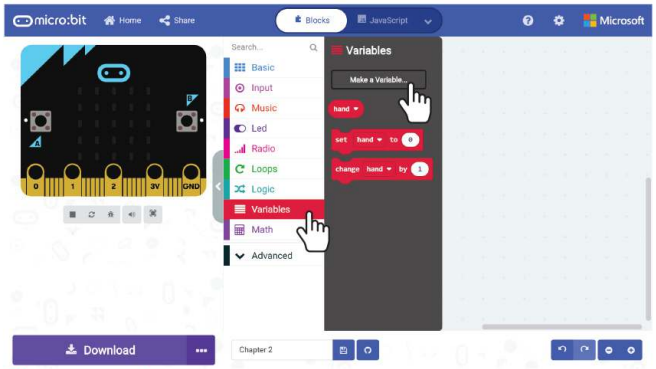
เพิ่มความท้าทายของเกมนี้ขึ้นอีกโดยการกำหนดให้แต่ละคนมี 3 ชีวิตสามารถทำได้โดยการสร้างตัวแปรใหม่ที่มีชื่อว่า 'lives' และเพิ่มใส่ใน block ของโปรแกรม

ขั้นตอนที่ 11 คลิกเลือก [Input] ใน Toolbox และเลือก [on button _ pressed] block หลังจากนั้นทำการ Duplicate block และทำการเปลี่ยนการตั้งค่าเป็น '**button B**' และ '**button A+B**' ตามลำดับ

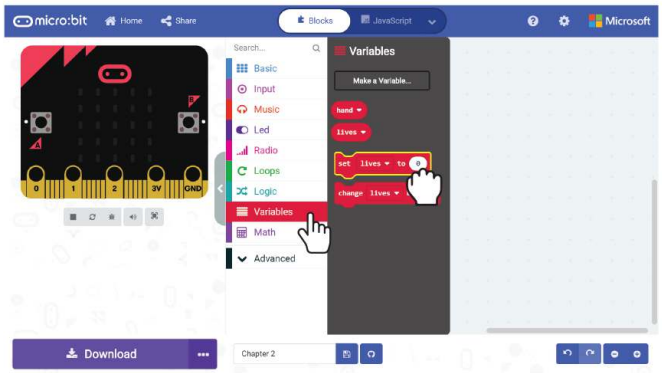


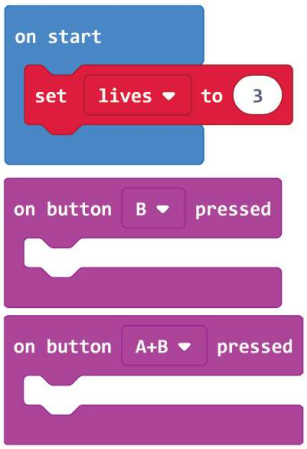


ขั้นตอนที่ 12 คลิกเลือก [Variable] ใน Toolbox และเลือก [Make a Variable] และตั้งชื่อตัวแปรว่า '**lives**' ในหน้าต่างป๊อป-อัพที่ปรากฏขึ้นมา และคลิก OK

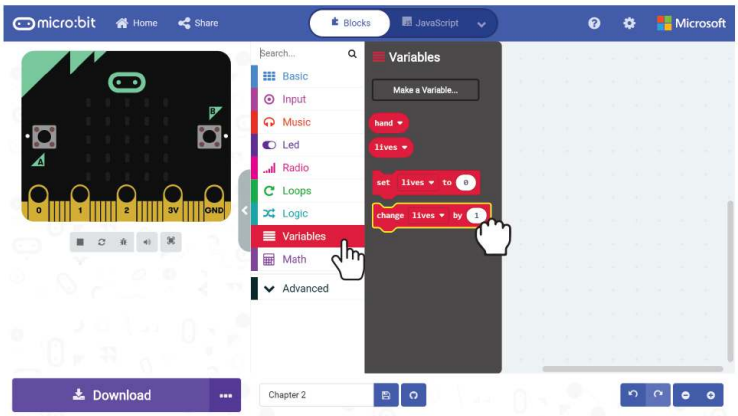


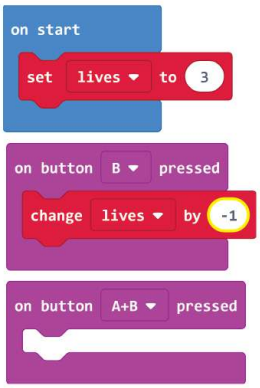
ขั้นตอนที่ 13 คลิกเลือก [Variable] ใน Toolbox และเลือก [set _ to _] block แล้วนำไปใส่ใน [Basic] : [on start] block หลังจากนั้นทำการเลือกค่าตัวแปรเป็น '**lives**' และกำหนดค่าของตัวแปรให้เท่ากับ **3**



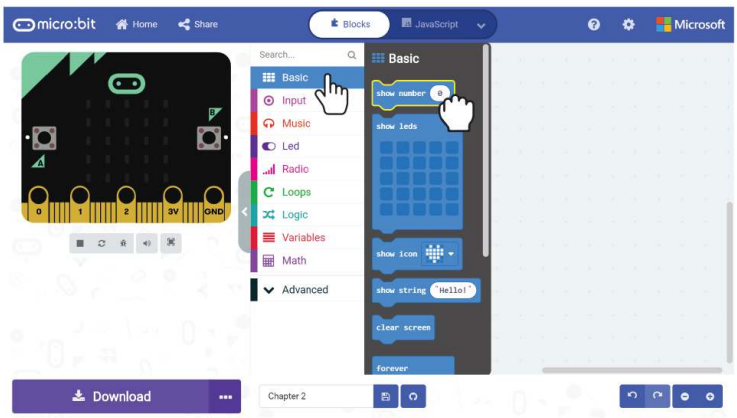


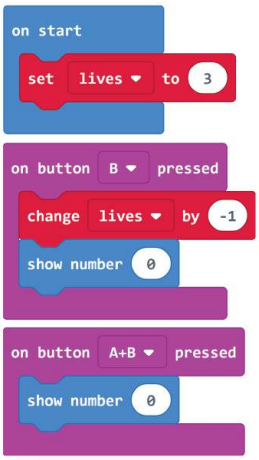
ขั้นตอนที่ 14 คลิกเลือก [Variable] ใน Toolbox และเลือก [change _ by _] block แล้วนำไปไว้ใน [on button B pressed] block แล้วทำการกำหนดค่าของตัวแปร 'lives' เป็น -1



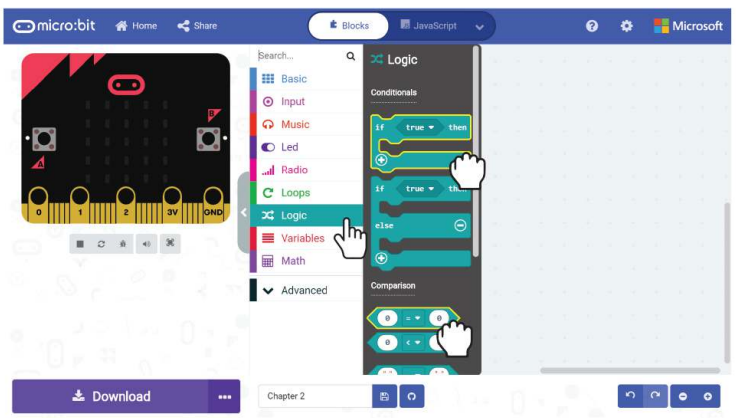


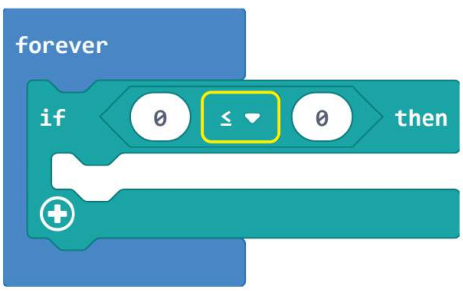
ขั้นตอนที่ 15 คลิกเลือก [Basic] ใน Toolbox และเลือก [show number] block แล้วทำการ Duplicate แล้วนำไปวางใน [on button A+B pressed] block และ [on button B pressed] block



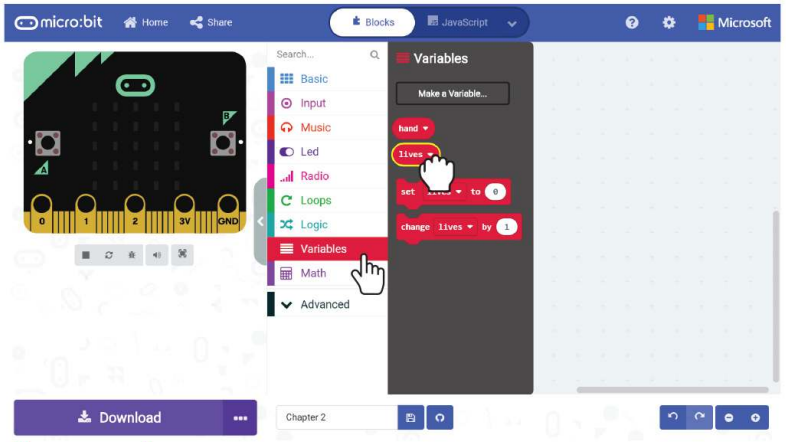


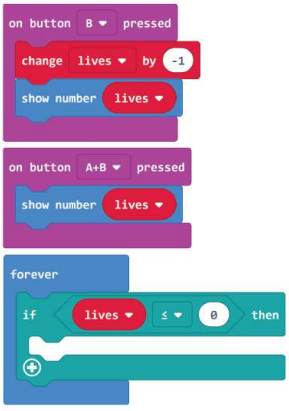
ขั้นตอนที่ 16 คลิกเลือก [Logic] ใน Toolbox และเลือก [if-then] block และ [_ = _] block แล้วนำไปเพิ่มใน [Basic] : [Forever] block และทำการตั้งค่าเครื่องหมายเป็น '≤'



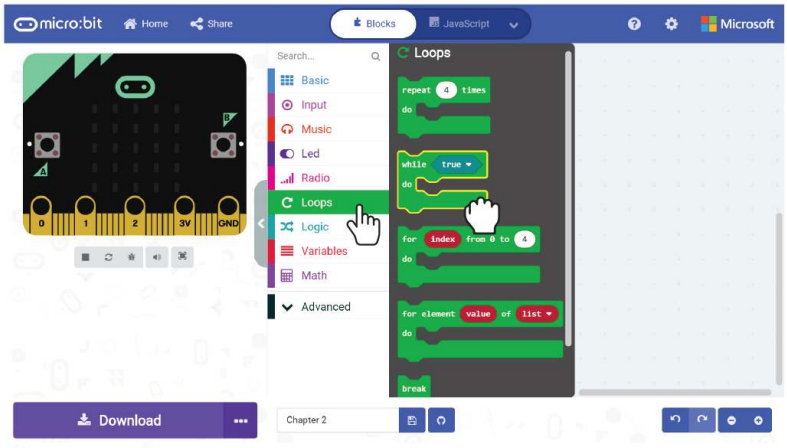


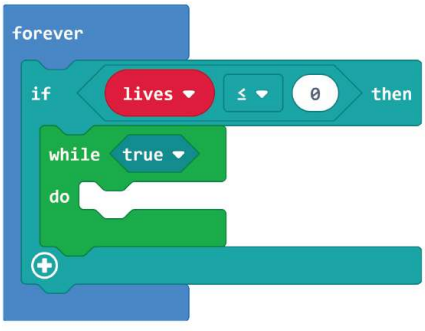
ขั้นตอนที่ 17 คลิกเลือก [Variable] ใน Toolbox และเลือก [lives] block ทำการ Duplicate แล้วนำไปไว้ใน [show number] block ทั้งสอง block และ slot ด้านซ้ายของ [_ = _] block



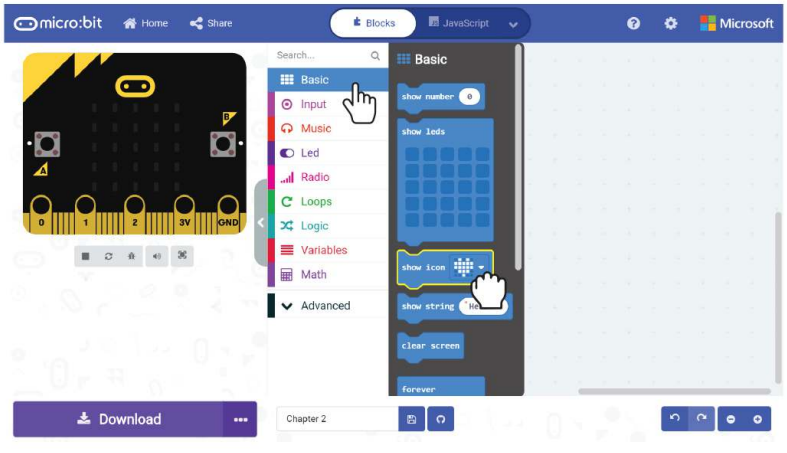


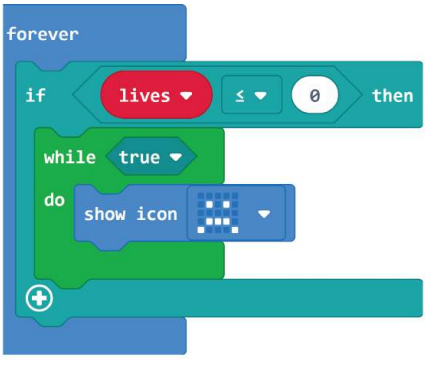
ขั้นตอนที่ 18 คลิกเลือก [Loop] ใน Toolbox และเลือก [while _ do] block แล้วนำไปวางใน [if-then] block



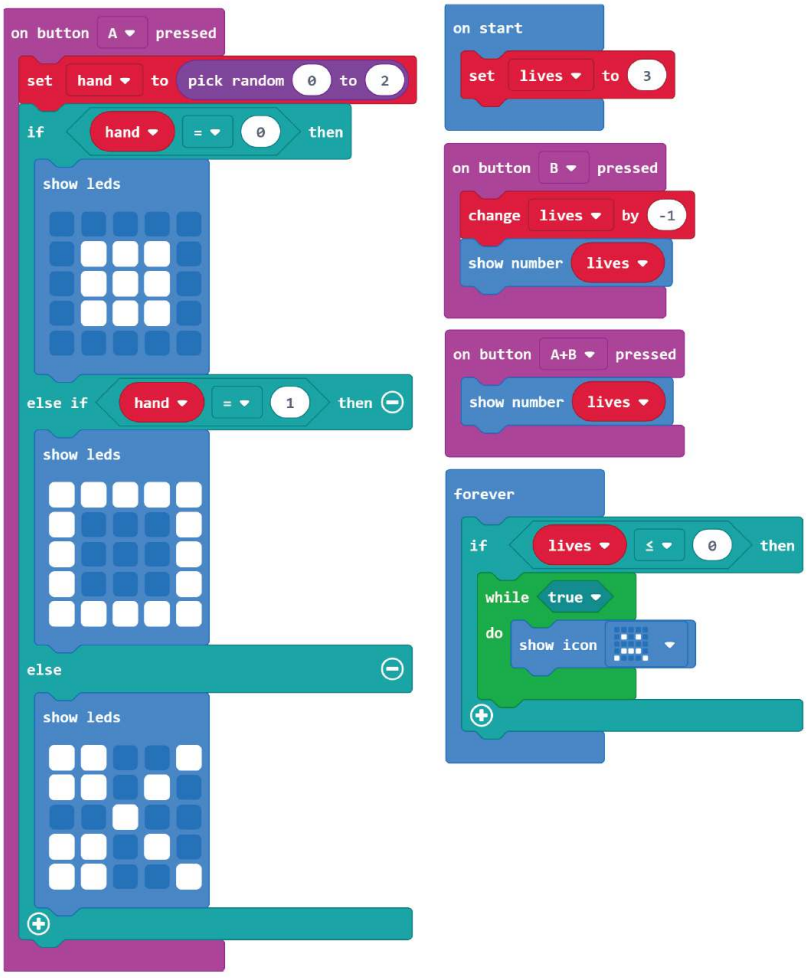


ขั้นตอนที่ 19 คลิกเลือก [Basic] ใน Toolbox และเลือก [show icon] block แล้วนำไปวางใน slot ของ [while _ do] block และตั้งค่าไอคอนที่จะแสดงเป็น **'หน้าเศร้า (sad face)'**



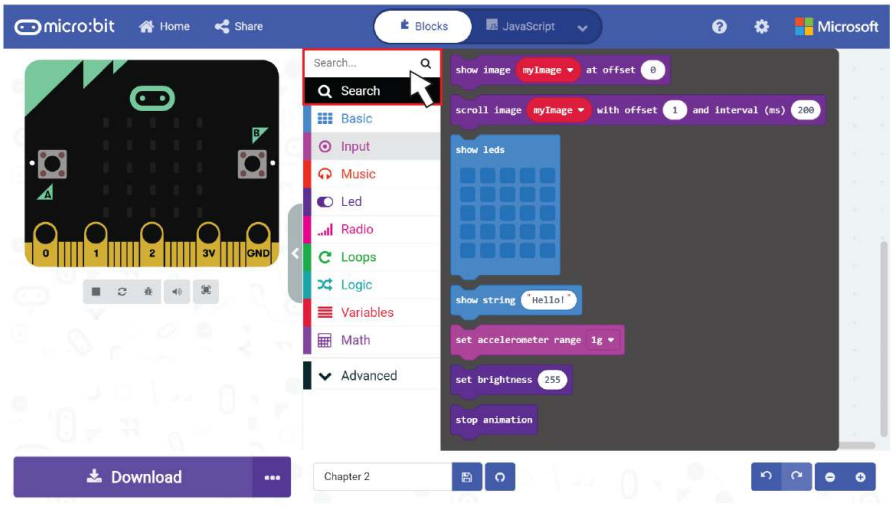


ขั้นตอนที่ 20 นี่คือโปรแกรมแบบเสร็จสมบูรณ์แล้ว ทำการแฟลชโปรแกรมเข้าสู่ EDU:BIT ของคุณ แล้วเตรียมสนุกกับการแข่งขันเพื่อตามหา ราชา/ราชินี แห่งการเป่า ยิ้ง ฉุบ ได้เลย!



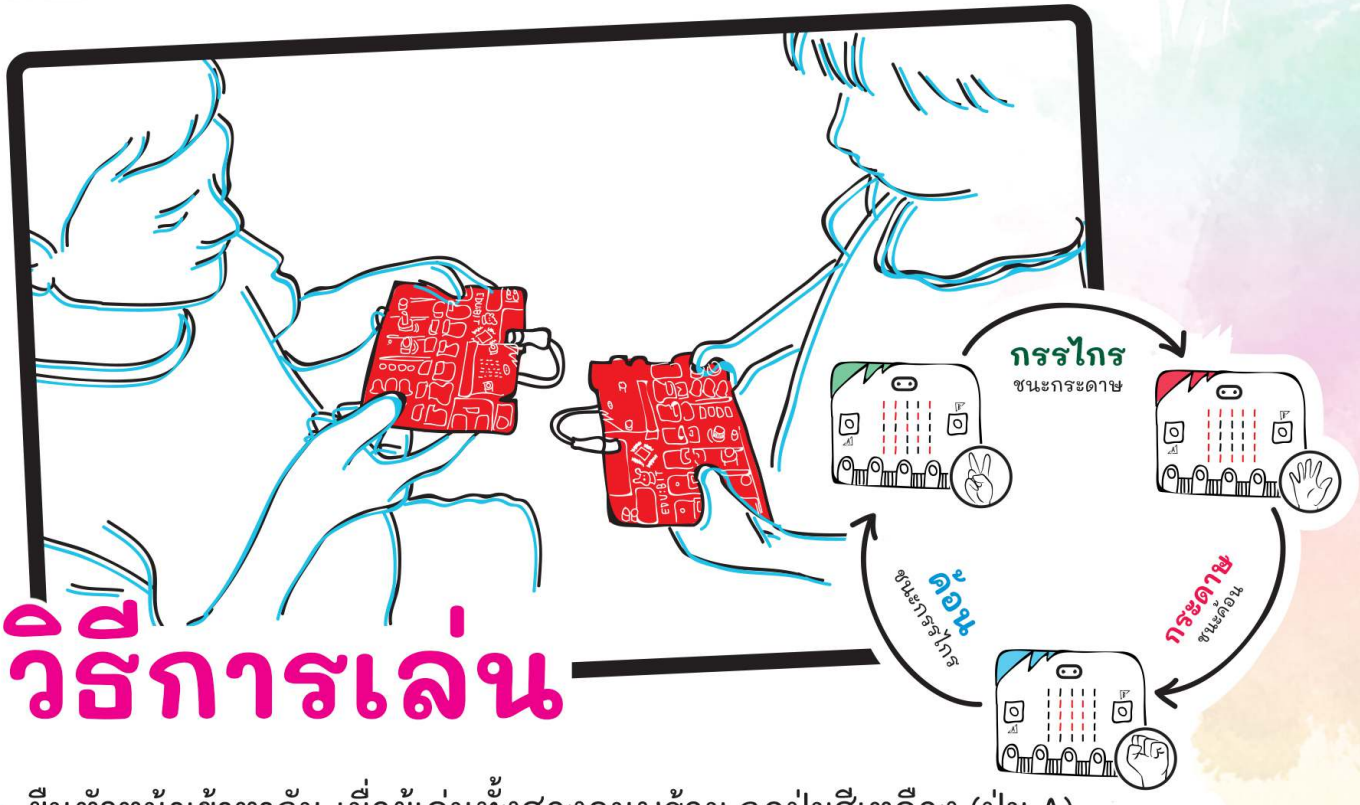
เคล็ดลับ!

block สำหรับเขียนโปรแกรมทุก block จะมีสีเป็นของตนเอง คุณสามารถหา block ประเภทเดียวกันที่คุณต้องการได้โดยการหา block ที่มีสีเหมือนกัน หรืออีกทางหนึ่งสามารถทำได้โดยการพิมพ์ keyword ของ block ที่คุณต้องการในช่องค้นหา



มาลองเล่นกัน!

เป่า ยิ้ง ฉุบ เวอร์ชัน Advance



วิธีการเล่น

- ยืนหัวหน้าเข้าหากัน เมื่อผู้เล่นทั้งสองคนพร้อม กดปุ่มสีเหลือง (ปุ่ม A) เพื่อสุ่มว่าออก ค้อน กระดาษ หรือ กรรไกร
- เช็คว่าใครแพ้ใครชนะ
- หากคุณเป็นผู้แพ้ ให้ทำการกดปุ่มสีน้ำเงิน (ปุ่ม B) บน EDU:BIT ของคุณ 1 ครั้งเพื่อ -1 แต้มชีวิตของคุณ
- สามารถแต้มเช็คชีวิตที่เหลืออยู่ของคุณได้โดยการกดปุ่มสีเหลืองและสีน้ำเงินพร้อมกัน
- หากผู้เล่นคนไหนแพ้ครบ 3 ครั้ง แต้มชีวิตหมด EDU:BIT ของเขาจะแสดงหน้าเศร้า (sad face)

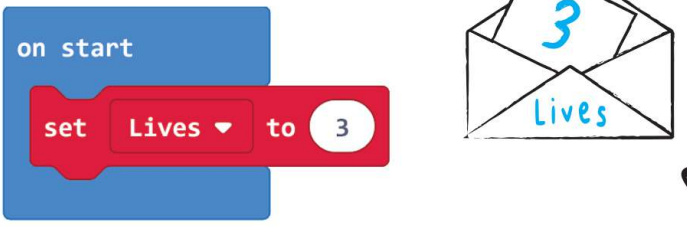
NOTE!

- หากต้องการเล่นอีกครั้ง คุณจำเป็นที่จะต้องรีเซทบอร์ดของคุณเพื่อเริ่มเกมใหม่
- หากอยู่คนเดียวไม่มีเพื่อนเล่น คุณสามารถเล่นบน Simulator ใน MakeCode Editor ได้

BREAK THE CODE

ในการเขียนโปรแกรมคอมพิวเตอร์ เราจะใช้ ตัวแปร (variables) เพื่อเก็บข้อมูลหรือค่าต่างๆที่สามารถเปลี่ยนแปลงได้ขณะ runtime (ขณะที่โปรแกรมทำงานอยู่) เหมือนกับซองจดหมายที่ภายในมีกระดาษที่เขียนตัวเลขหรือข้อมูลอยู่หนึ่งแผ่น ซึ่งกระดาษแผ่นนั้นสามารถหยิบออกมาและใส่แผ่นใหม่ที่มีตัวเลขหรือข้อมูลใหม่เข้าไปแทนได้

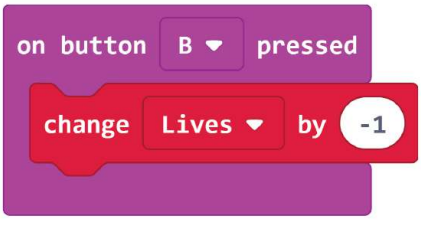
ในโค้ดที่ผ่านมา เราสร้างตัวแปรที่มีชื่อว่า “lives” และกำหนดค่าให้เท่ากับ 3

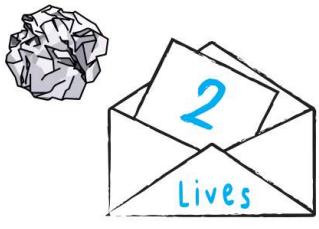


เลข 3 ที่ถูกเขียนบนกระดาษคือข้อมูล

จ่าหน้าซองด้วยชื่อว่า “Lives” โดยกำหนดให้เป็นตัวแปร

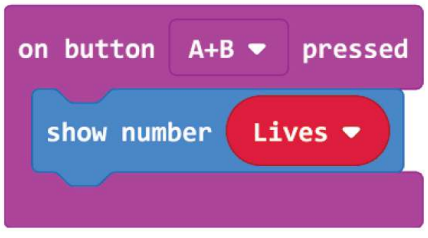
หลังจากนั้น เมื่อปุ่ม B ถูกกด ตัวเลขในกระดาษจะถูกลบด้วย 1

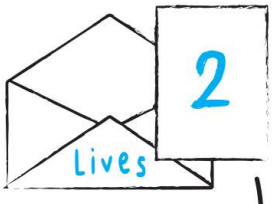


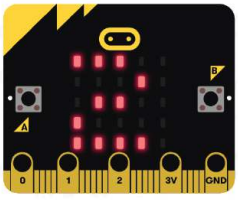


เมื่อปุ่ม B ถูกกด กระดาษที่เขียนเลข 3 ไว้จะถูกนำออกจากซองจดหมายและแทนที่ด้วยกระดาษแผ่นใหม่ที่เขียนเลข 2 (เนื่องจาก 3-1 = 2)

เมื่อกดปุ่ม A+B พร้อมๆกันจะทำให้ LED matrix แสดงค่าในตัวแปร “lives” ที่เหลืออยู่





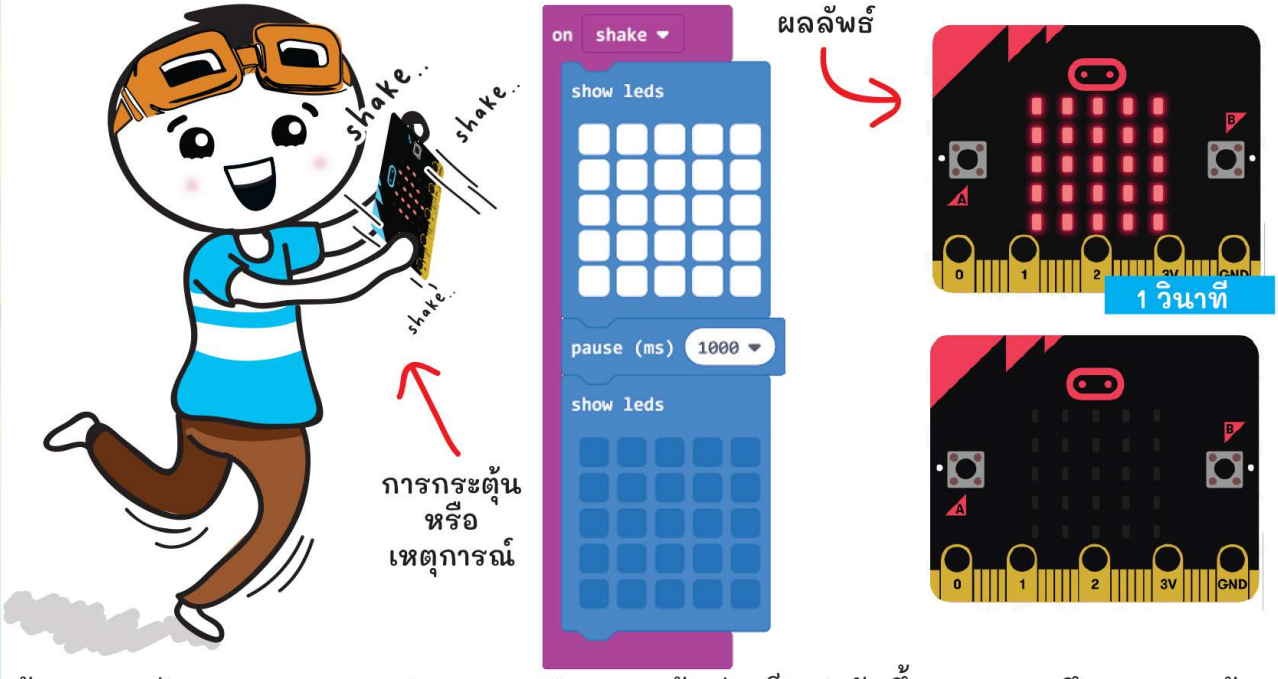


เมื่อปุ่ม A และ B ถูกกดพร้อมๆกันก็เหมือนกับการหยิบกระดาษที่มีตัวเลขในซองจดหมายออกมาเพื่ออ่านข้อมูลบนกระดาษใบนั้น

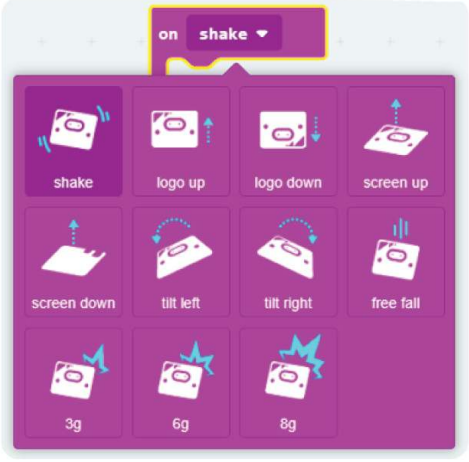
มาลองดู Block รูปแบบอื่นกัน

นอกเหนือจาก [on button _ pressed] block คุณยังสามารถใช้ block อื่นๆใน [Input] Toolbox สำหรับการเขียนโปรแกรมเชิงเหตุการณ์ (event-based programming) ซึ่งสิ่งที่จะเกิดขึ้นเกิดจากการที่ผู้ใช้ทำอะไรซักอย่าง เช่นการกดปุ่มหรือการสั่นบอร์ด เราจะเรียกว่า 'event'

ยกตัวอย่างโค้ดจากภาพ โค้ดนี้จะทำให้ LED matrix ติดเป็นเวลา 1 วินาทีเมื่อไหร่ก็ตามที่บอร์ดมีการสั่น ไม่เชื่อก็ลองดูสิ!

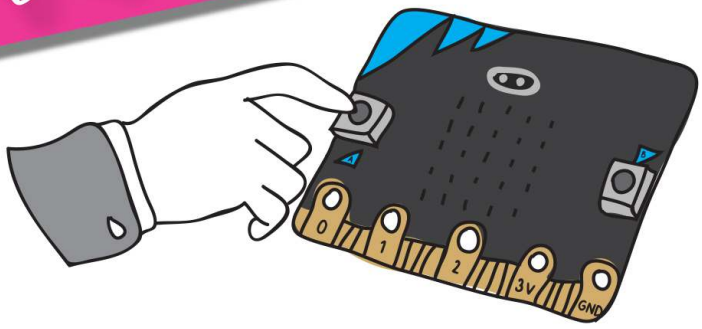


ถ้าคุณกดปุ่ม [shake] บน block จะปรากฏหน้าต่างป๊อป-อัพขึ้นมาแสดงถึงการกระตุ้นหรือเหตุการณ์ประเภทต่างๆ ลองเขียนโปรแกรมให้ EDU:BIT แสดงไอคอนรูปแบบที่แตกต่างกันสำหรับการกระตุ้นหรือเหตุการณ์ที่แตกต่างกัน ลองทำดูสิ ขอให้สนุกนะ!

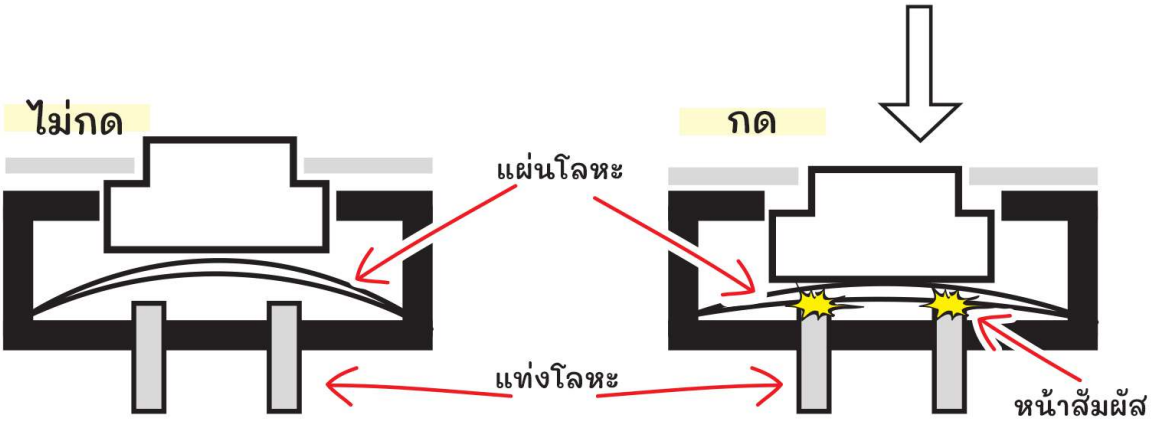


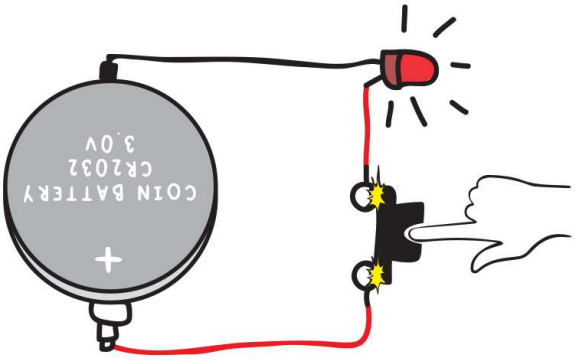
EDU:BIT สามารถตรวจจับการสั่นและความเอียงของตัวเองได้เพราะว่ามีเซนเซอร์ตรวจจับการเคลื่อนไหว (motion sensor) อยู่บน microbit

FUN FACT!



ปุ่มกด เป็นอุปกรณ์รับ input
โดยมี 2 รูปแบบ
คือ “ถูกกด” และ “ไม่ถูกกด”





เมื่อปุ่มถูกกด วงจรในภาพจะครบวงจรทำให้หลอด LED ติด! เดาซิว่าจะเกิดอะไรขึ้นเมื่อเลิกกดปุ่ม?

HELP

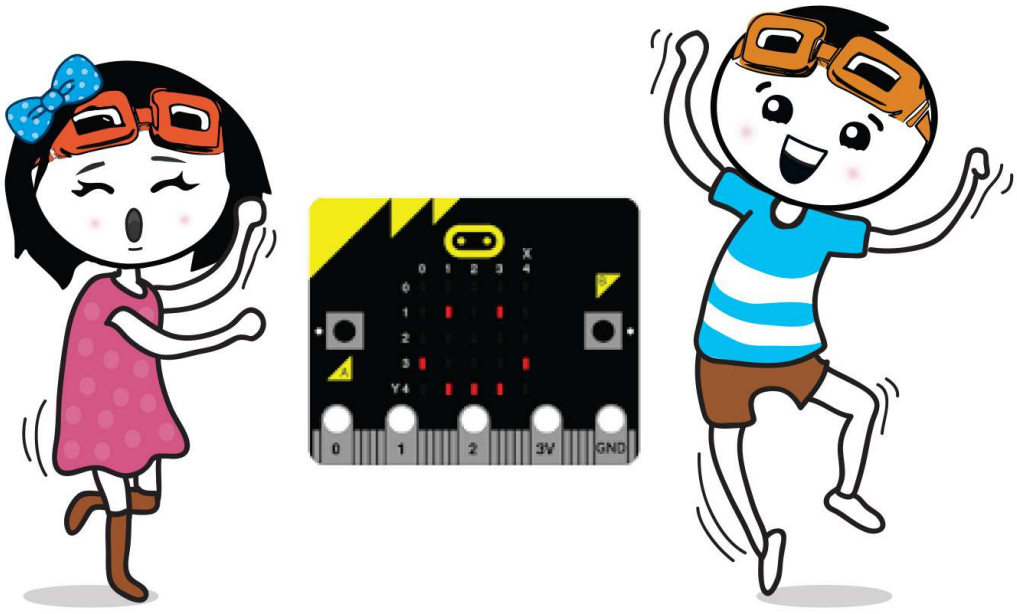
ปุ่มฉุกเฉิน

กดปุ่มนี้เมื่อ
เกิดเหตุฉุกเฉิน

ปุ่มสีดำ สีเขียว สีเทา และสีขาวใช้สำหรับการเปิด-ปิดตามปกติ แต่ปุ่มสีแดงใช้สำหรับเหตุฉุกเฉินหรือหยุดการทำงานของเครื่องจักร

youtu.be/t_Qujjd_38o

APPLICATION CHALLENGE

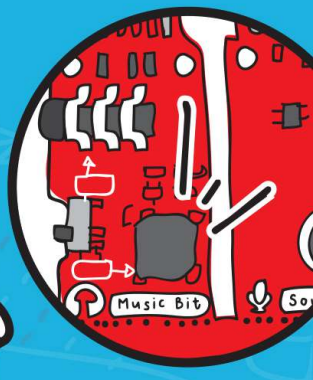


เขียนโปรแกรมให้ EDU:BIT ทำหน้าที่เป็นเครื่องนับจำนวนนักเรียน โดยเด็กผู้หญิงต้องกดปุ่ม A และผู้ชายต้องกดปุ่ม B เมื่อเข้าห้องเรียน	
On start	แสดงหน้ายิ้มบน LED matrix กำหนดให้ตัวแปร Girl = 0 และ Boys = 0
On Button A pressed (ปุ่มสีเหลือง)	บวกค่าในตัวแปร Girl ทีละ 1
On Button B pressed (ปุ่มสีน้ำเงิน)	บวกค่าในตัวแปร Boy ทีละ 1
On Button A+B pressed	แสดงข้อมูลเลื่อนผ่านจอ LED matrix ซึ่งได้แก่ Total = (Girl+Boy) ; Girl = (Girl) ; Boy = (Boy)

บทที่ 3

มาฟังเพลงกัน~

Music Bit (Piezo Buzzer + Audio Jack)

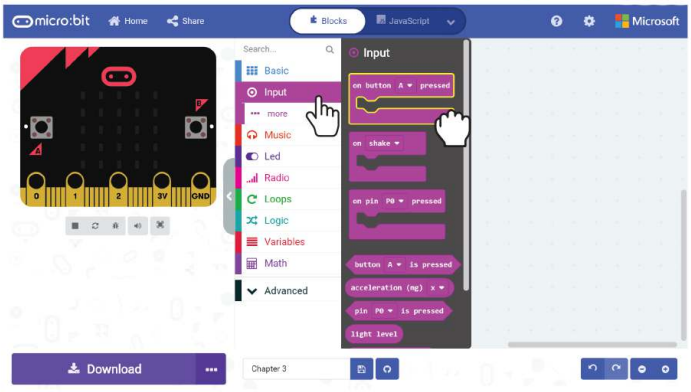


สแกนเลย!

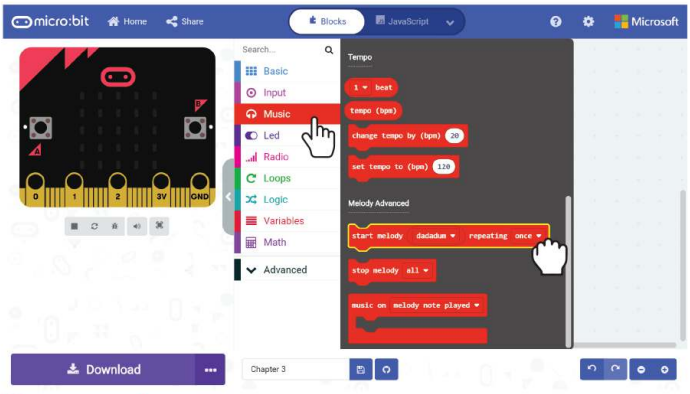
link.cytron.io/edubit-chapter-3

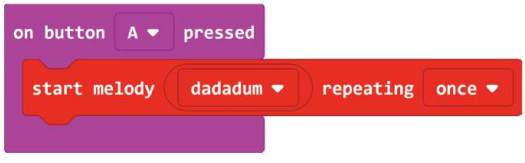
มาเขียนโปรแกรมกัน!

ขั้นตอนที่ 1 สร้างโปรเจ็กต์ของคุณใหม่บน MakeCode Editor คลิก [Input] ใน Toolbox และเลือก [on button _ pressed] block

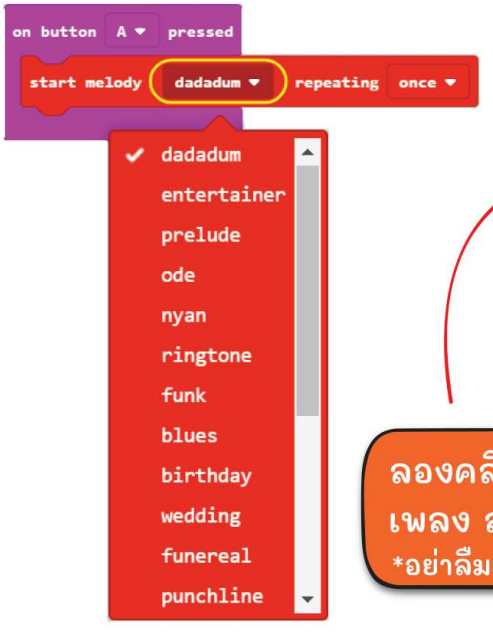


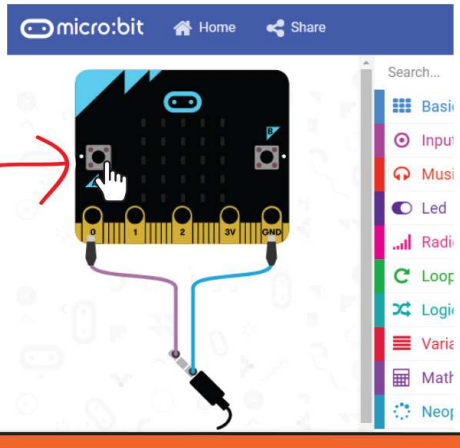
ขั้นตอนที่ 2 คลิกเลือก [Music] ใน Toolbox และเลือก [start melody _ repeating] block





ขั้นตอนที่ 3 คลิกที่ [dadadum] ใน [start melody _ repeating] block และเลือกเพลง 'birthday' จากตัวเลือก



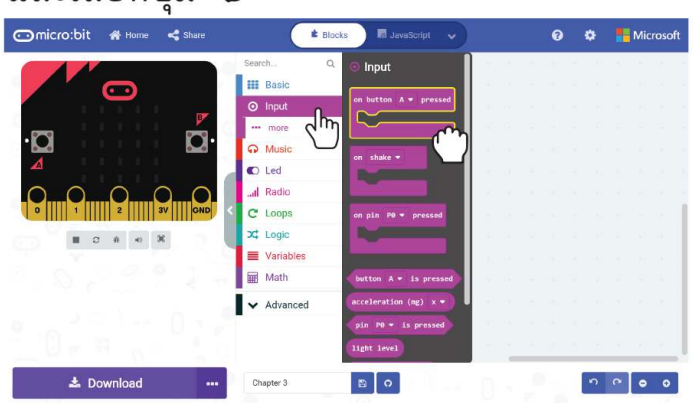


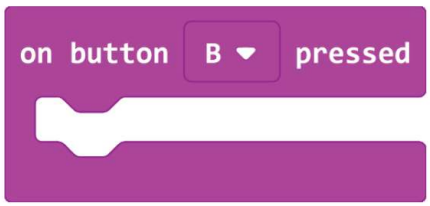
ลองคลิกที่ปุ่ม A บน micro:bit simulator คุณจะได้ยินเสียงเพลง ลองเลือกเพลงอื่นดูสิ

*อย่าลืมเปิดเสียงของคอมพิวเตอร์นะ

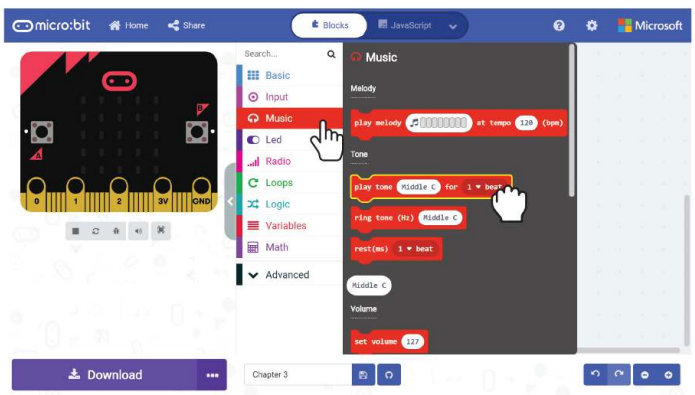
นอกเหนือจากเพลงใน List คุณสามารถโปรแกรม EDU:BIT ให้เล่นเพลงที่คุณต้องการได้ ไม่เชื่อลองเล่นเพลงนี้ดู

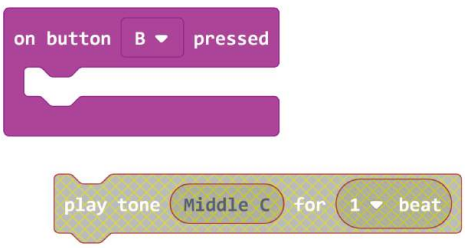
ขั้นตอนที่ 4 คลิก [Input] ใน Toolbox และเลือก [on button _ pressed] block และเลือกปุ่ม “B”



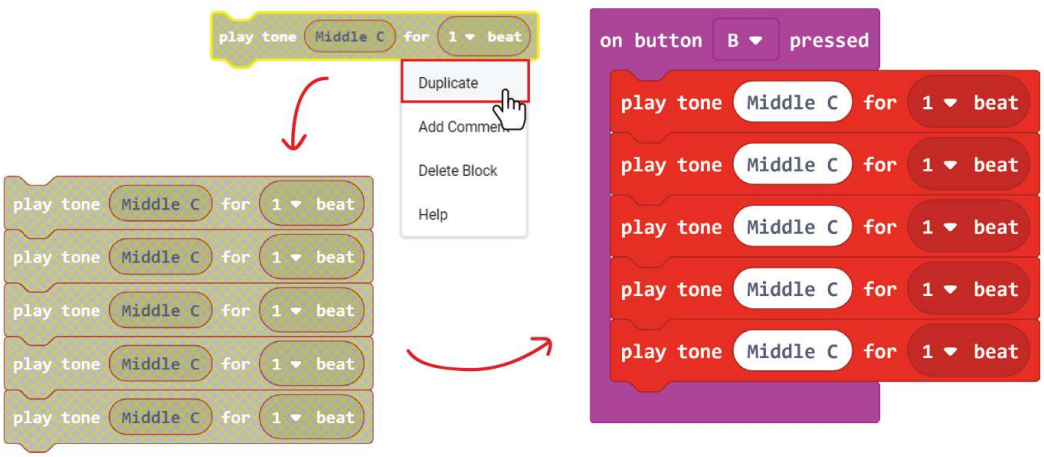


ขั้นตอนที่ 5 คลิกเลือก [Music] ใน Toolbox และเลือก [play tone _ for _ beat] block

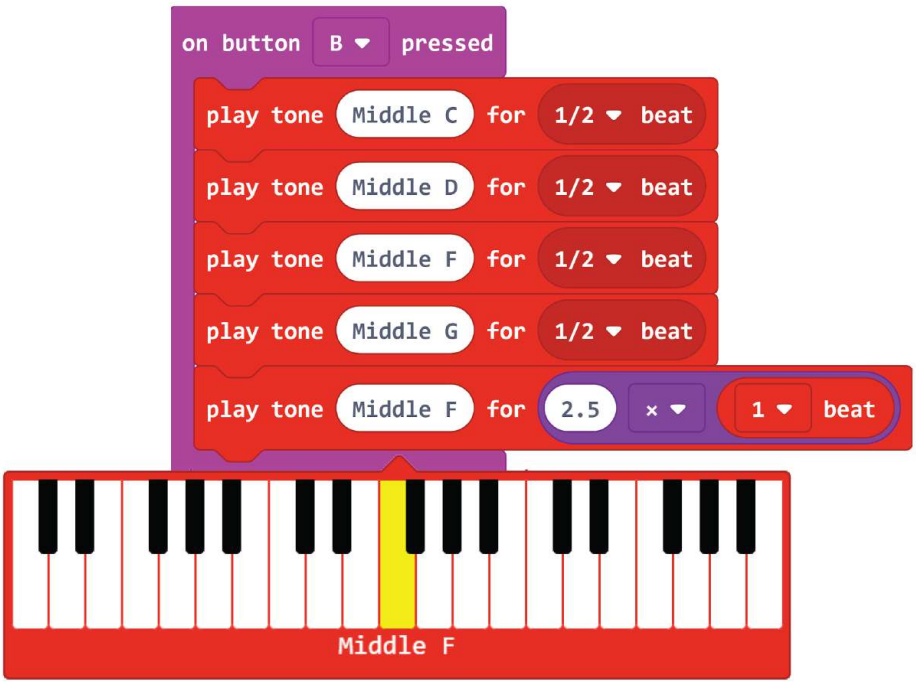




ขั้นตอนที่ 6 บน Workspace คลิกขวาที่ [play tone _ for _ beat] block และทำซ้ำให้มี [play tone _ for _ beat] block ครบ 5 block แล้ววาง block ทั้งหมดใน [on button B pressed] block

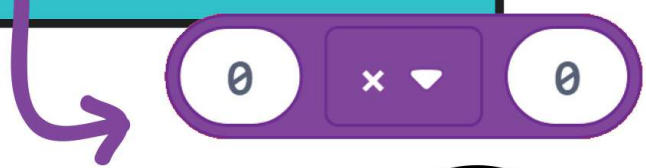


ขั้นตอนที่ 7 เลือก 'tone' และ 'beat' ใน [play tone_for_beat] blocks ตามภาพตัวอย่างด้านล่าง



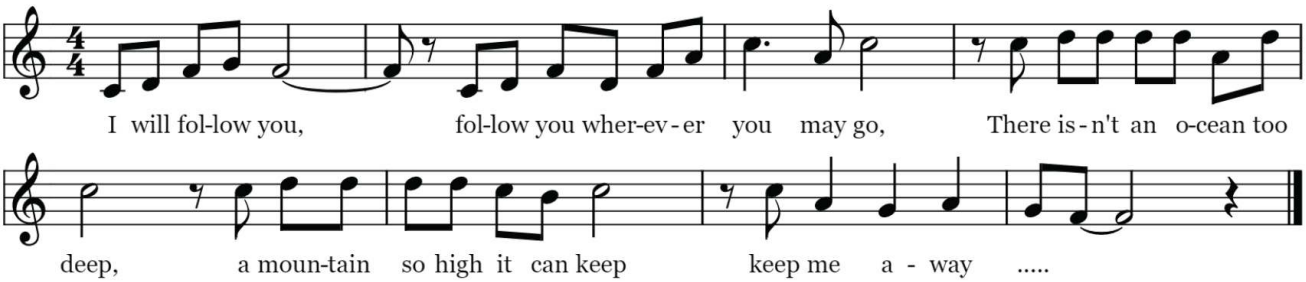
NOTE!

block สีม่วงนี้มาจาก [Math] ใน Toolbox



ขั้นตอนที่ 8 เขียนโค๊ดเพื่อเล่นเพลงทั้งเพลงโดยการใช้ [play tone _ for _ beat] block โดยเปลี่ยน 'tone' และ 'beat' ให้ตรงกับเพลงที่จะเล่น โดยคุณสามารถดู tones และ beats ของเพลงได้ที่หน้าถัดไป

I Will Follow You



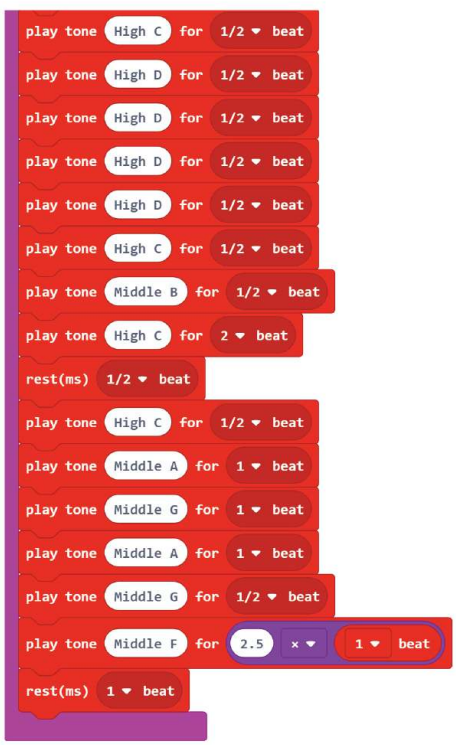
I will fol-low you,

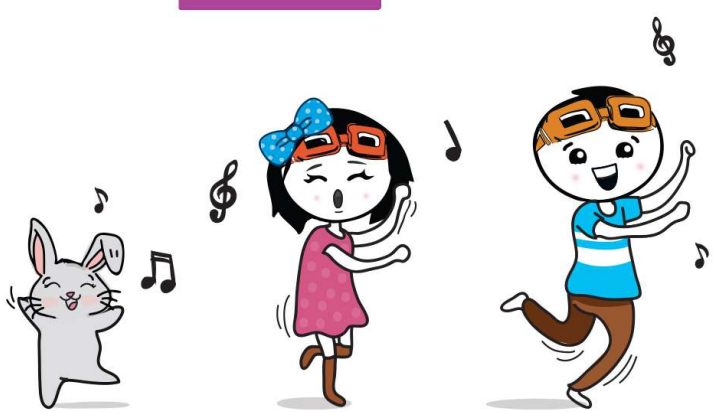
Fol-low you wher-ev-er you may go,

There is n't an o-cean too deep

A moun-tain so high it can keep,

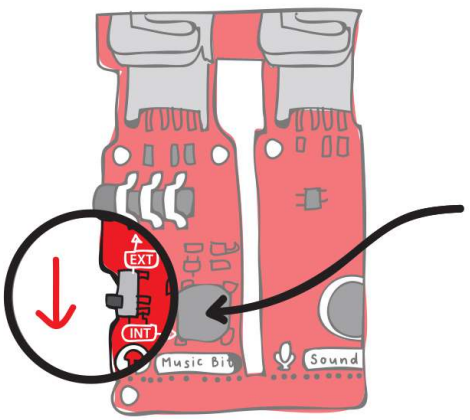
Keep me a-way ... ...

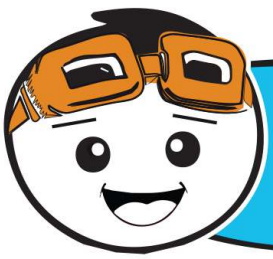




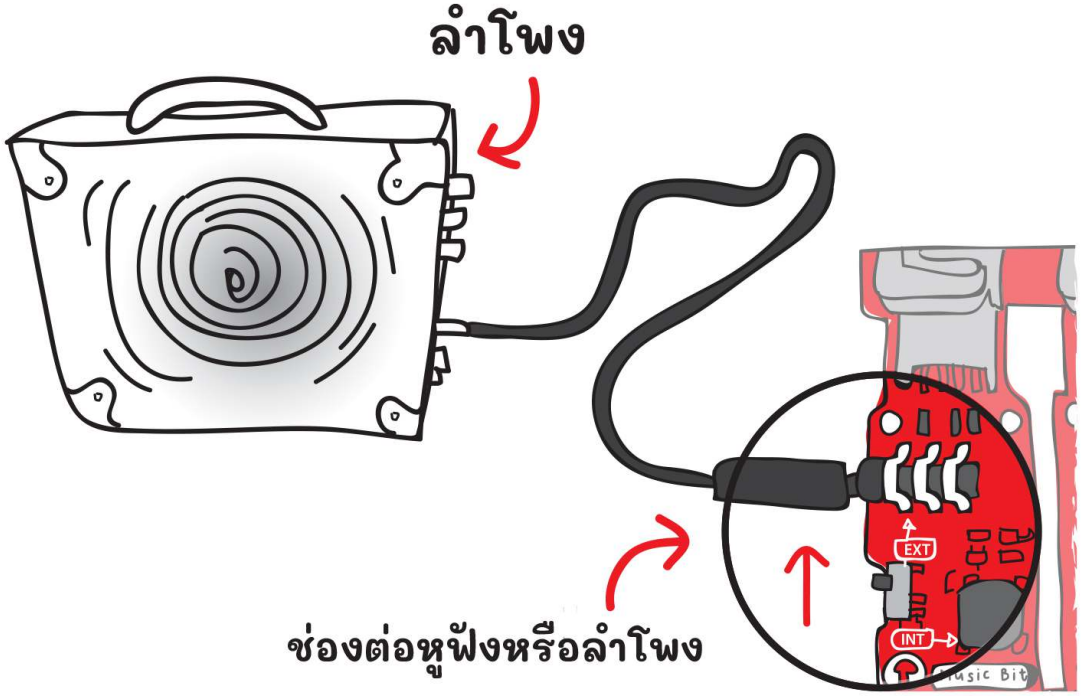
ขั้นตอนที่ 9 แฟลชโค๊ดที่เสร็จสมบูรณ์ลง EDU:BIT

EDU:BIT จะเล่นเพลง “I Will Follow You” เมื่อคุณกดปุ่มสีน้ำเงิน (ปุ่ม B) บน EDU:BIT ของคุณ อย่าลืม! เปิด EDU:BIT ของคุณและ เปิด piezo buzzer โดยการสไลด์สวิตซ์ไปที่ “INT” (internal)



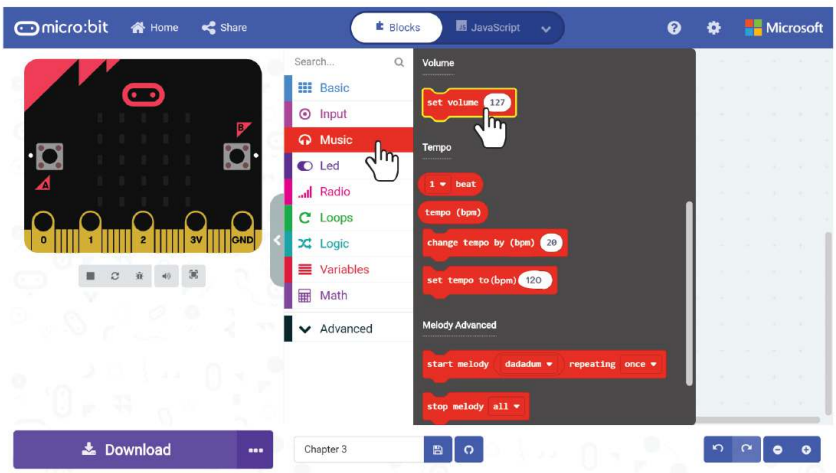


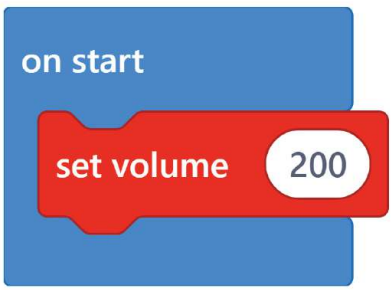
คุณสามารถต่อลำโพงภายนอกหรือหูฟังกับช่องต่อลำโพงบน EDU:BIT และสไลด์สวิตซ์ไปที่ “EXT” (external)



เสียงเพลงเบาไปหรือดังไปรึเปล่า? คุณสามารถใช้ [set volume_] block เพื่อเพิ่ม-ลดเสียง ซึ่งมีช่วงอยู่ที่ 0 (เบาที่สุด) ถึง 255 (ดังที่สุด)

ขั้นตอนที่ 10 คลิกเลือก [Music] ใน Toolbox และเลือก [set volume _] block นำไปใส่ใน [on start] block และตั้งค่า [set volume _] block ไว้ที่ค่าเท่ากับ 200

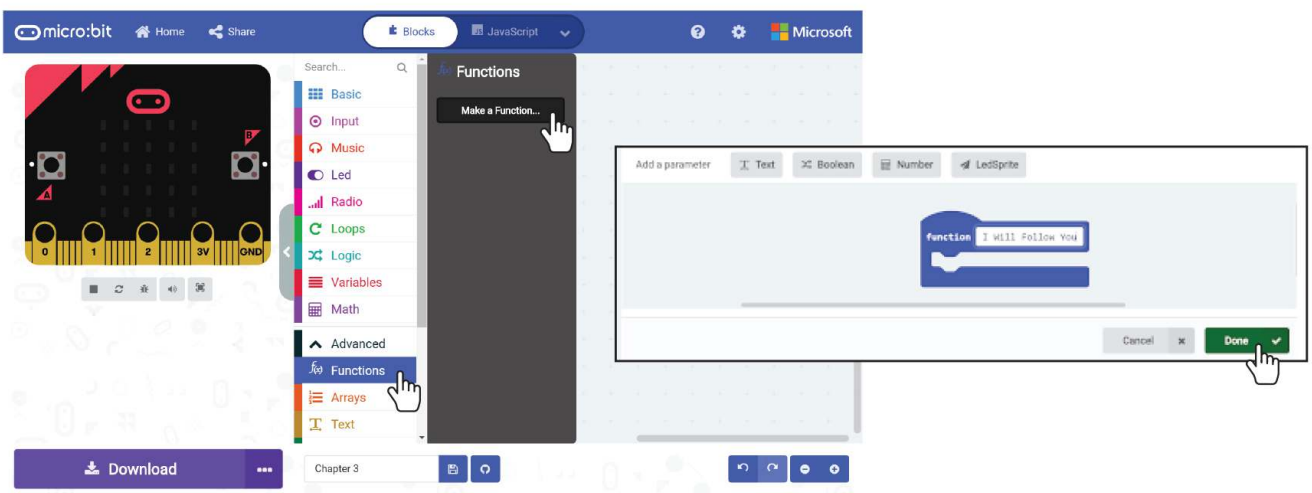




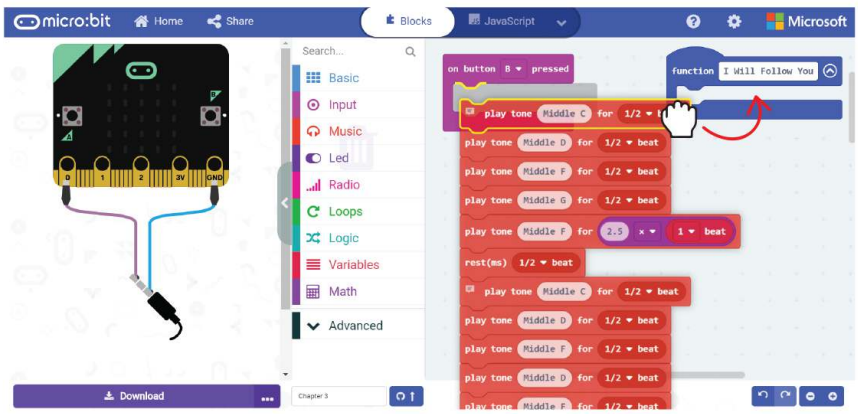
NOTE!

เราสามารถสร้าง block โปรแกรมที่ทำงานเฉพาะทางด้วย function ยกตัวอย่างเช่น คุณต้องการให้โค๊ดของคุณเล่นเพลง “I Will Follow You” ในการเขียนโปรแกรม ฟังก์ชันคืองานหรือกลุ่มของขั้นตอนการทำงาน เมื่อฟังก์ชันถูกกำหนดขึ้นในโปรแกรม จะสามารถถูกเรียกใช้ได้หลายๆตำแหน่งภายในโปรแกรม โดยที่คุณไม่ต้องสร้าง block ซ้ำๆ

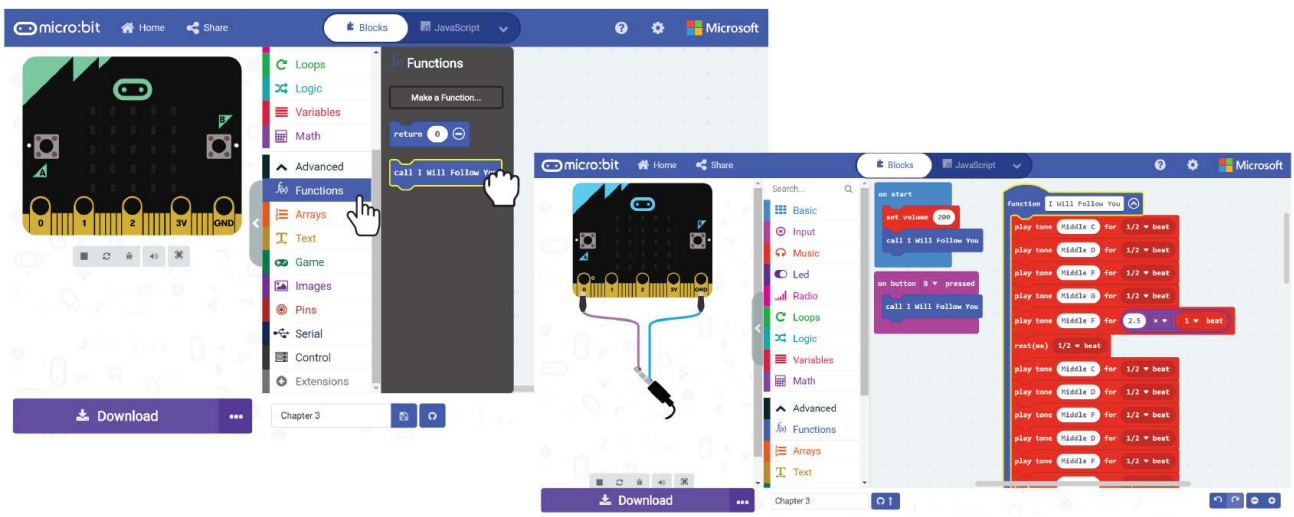
ขั้นตอนที่ 11 คลิกเลือก [Advanced] ใน Toolbox และเลือก [Functions] คลิกเลือก [Make a Function] และตั้งชื่อเป็น ‘I Will Follow You’ บนหน้าต่างป๊อป-อัพที่ปรากฏขึ้น และคลิก ‘Done’



ขั้นตอนที่ 12 [function I Will Follow You] block จะปรากฏขึ้นบนหน้าจอของคุณ คลิกที่ block บนสุดของ [play tune] block ที่อยู่ใน [on Button B pressed] block แล้วลากไปวางใน [function I Will Follow You] slot

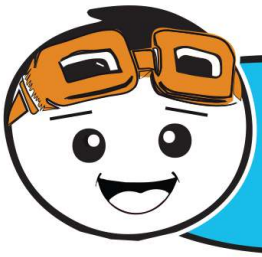


ขั้นตอนที่ 13 คลิกเลือก [Functions] ใน Toolbox และเลือก [call I Will Follow You] block ทำการทำซ้ำ block แล้วนำไปวางที่ [on start] block และ [on button B pressed] block ดังตัวอย่างด้านล่าง

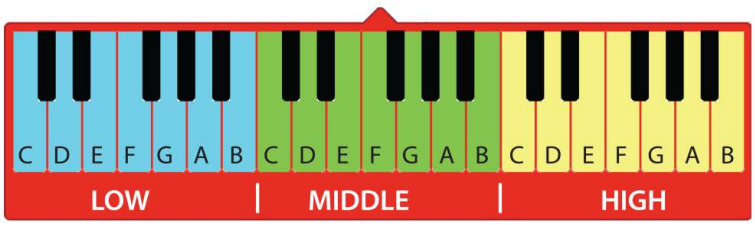


ขั้นตอนที่ 14 แฟลชโค๊ดที่เสร็จสมบูรณ์ลงใน EDU:BIT แล้วเตรียมสนุกกับเพลงที่คุณสร้างขึ้นมาๆได้เลย

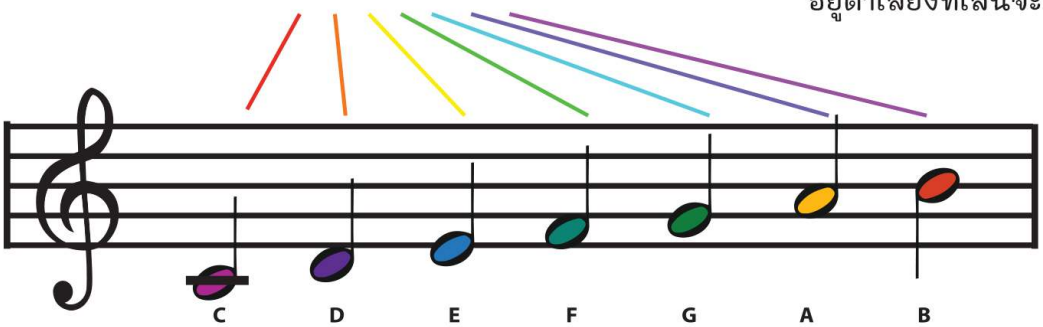
BREAK THE CODE



คุณสามารถโปรแกรม EDU:BIT ให้เล่นเพลงอะไรก็ได้ หากคุณรู้วิธีการอ่านโน๊ตเพลง นี่คือหลักการง่ายๆในการ “แกะเพลง (decode)” จากตัวโน๊ต



ตำแหน่งของโน๊ตเพลงบนบรรทัด 5 เส้น บ่งบอกถึง tone ของเสียง หากตำแหน่งของโน๊ตบนบรรทัด 5 เส้นอยู่สูงเสียงที่เล่นจะเป็นเสียงสูง และในทางกลับกัน หากตำแหน่งของโน๊ตบนบรรทัด 5 เส้นอยู่ต่ำเสียงที่เล่นจะเป็นเสียงต่ำ



Sign	Rest	Relative Length	Duration
𝅝	𝄻	Whole Note	4 beats
𝅗𝅥	𝄼	Half Note	2 beats
♩	𝄽 or 𝄾	Quarter Note	1 beat
♪	𝄿	Eighth Note	1/2 beat
𝅘𝅥𝅯	𝅀	Sixteenth Note	1/4 beat

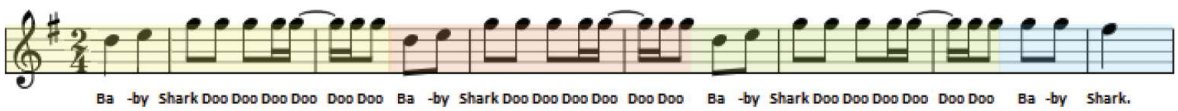
ลักษณะของตัวโน๊ตที่แตกต่างกัน บ่งบอกถึงความยาวเสียงของโน๊ตนั้นที่ต้องเล่น

BREAK THE CODE

จากหลักการในหน้าที่แล้ว คุณ “แกะเพลง” นี้ได้รึเปล่า?

Baby Shark

♩ = 115



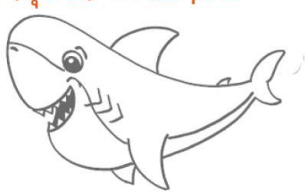
Line 1	Ba	-by	Shark	doo	doo,	doo	doo	doo	doo
Note	High D	High E	High G		High G	High G	High G		High G
Beat	1		1/2	1/2	1/2		1/2	1/4	1/2

Line 2	Ba	-by	Shark	doo	doo,	doo	doo	doo	doo
Note	High D		High G	High G	High G		High G	High G	High G
Beat	1/2	1/2	1/2		1/2	1/4	1/2		1/2

เล่นซ้ำ 2 รอบ
(เป็นทั้งบรรทัดที่ 2
และ บรรทัดที่ 3)

Line 4	Ba	-by	Shark
Note		High G	High F#
Beat	1/2	1/2	

โปรมแกรม EDU:BIT ให้เล่นเพลง Baby Shark
เมื่อกดปุ่มสีเหลือง (ปุ่ม A) และ
ปุ่มสีน้ำเงิน (ปุ่ม B) พร้อมๆกัน



NOTE!

ใช้ [set volume _] block สำหรับเพิ่ม-ลดความดังของเสียง

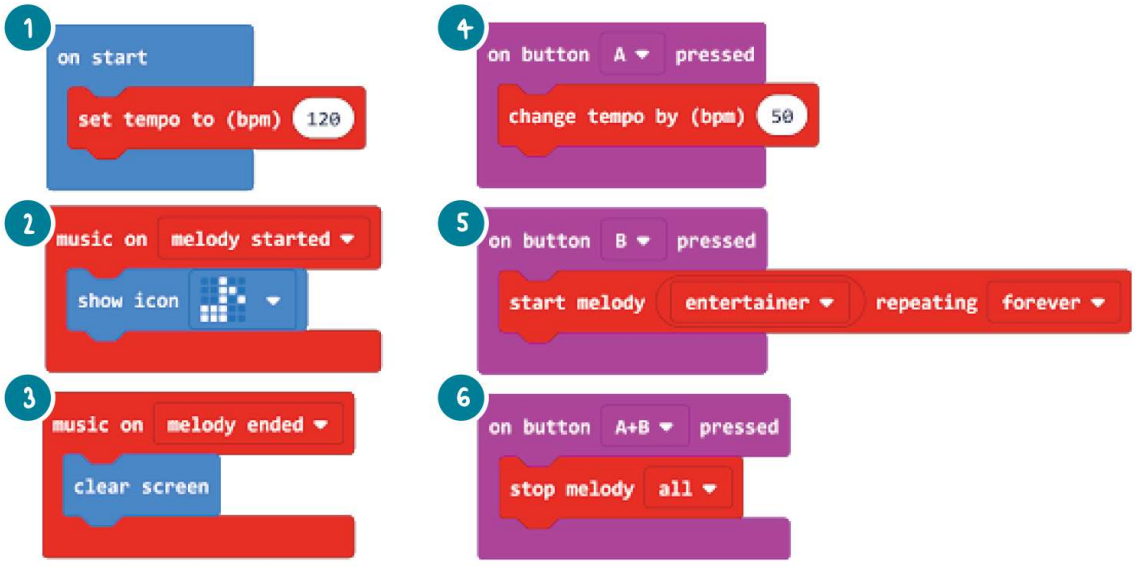
มาลองดู Block รูปแบบอื่นกัน

#1 คุณสามารถตั้งค่า “tempo” (จังหวะความเร็วของเพลง) โดยใช้ [set tempo to (bpm) _] block ยิ่ง bpm (beats per minute) เท่าไหร่ เพลงก็จะยิ่งเร็วขึ้นเท่านั้น และคุณยังสามารถใช้ [change tempo by (bpm) _] block ในการเปลี่ยน tempo ได้ด้วย

#2 คุณสามารถใช้ [stop melody _] block ในการหยุดเพลงขณะที่เพลงกำลังเล่นอยู่ได้

#3 คุณสามารถใช้ [music on _] และเงื่อนไขภายใน block ที่คุณเลือกไว้ เช่น เมื่อเพลงเริ่ม หรือเมื่อเพลงจบ เป็นตัวกระตุ้น (trigger) ในโค้ดของคุณได้

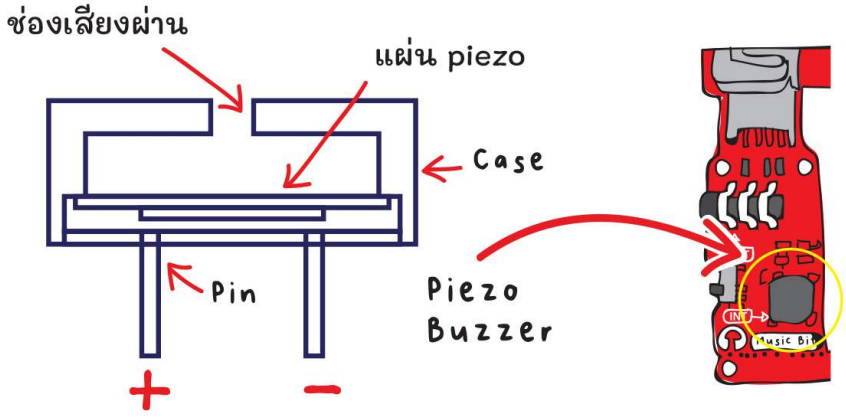
ตัวอย่างโค้ด



1. ในโปรแกรมนี้ tempo ปกติของเพลงถูกตั้งไว้ที่ 120 bpm
2. เมื่อเพลงเริ่มขึ้น LED matrix จะแสดงสัญลักษณ์รูปโน๊ตดนตรี
3. เมื่อเพลงจบ LED matrix ทุกดวงจะดับลง
4. เมื่อไหร่ก็ตามที่ปุ่ม A ถูกกด tempo ของเพลงจะถูกเพิ่มขึ้นครั้งละ 50 bpm
5. เพลงชื่อ “entertainer” จะถูกเล่นเมื่อไหร่ก็ตามที่ปุ่ม B ถูกกด
6. เพลงจะหยุดเมื่อปุ่ม A+B ถูกกดพร้อมๆกัน

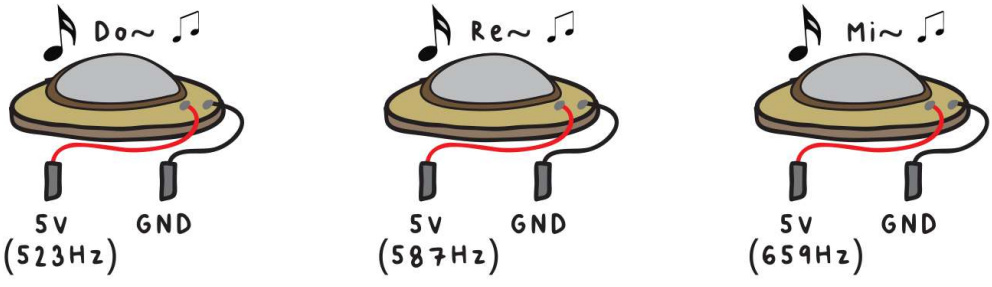
FUN FACT!

piezo buzzer โดยปกติแล้วจะใช้ในการสร้างเสียงด้วยการสั่นของแผ่น piezo เมื่อได้รับสัญญาณไฟฟ้า



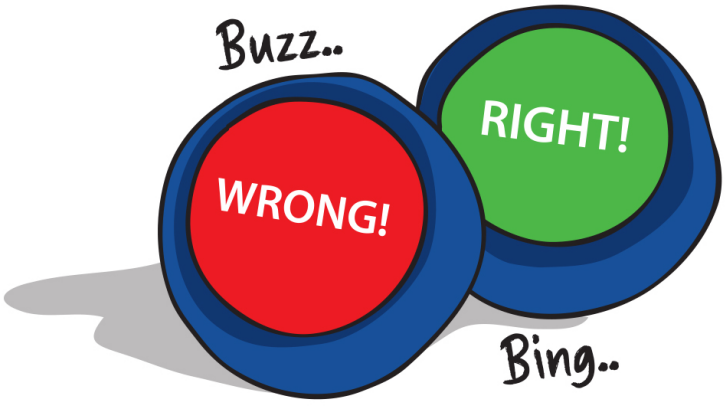
โดยการเปลี่ยนความถี่ของสัญญาณไฟฟ้า ความเร็วของการสั่นจะเปลี่ยนไป และด้วยเหตุนี้ piezo buzzer จึงสามารถสร้างเสียงได้อย่างหลากหลาย

แผ่น Piezo



หูของมนุษย์มีช่วงการได้ยินอยู่ที่ 20Hz ถึง 20,000Hz โดยเสียงอะไรก็ตามที่ต่ำกว่า 20Hz เราจะเรียกว่า อินฟราโซนิค (infrasonic) และเสียงอะไรก็ตามที่สูงกว่า 20,000Hz เราจะเรียกว่า อัลตร้าโซนิค (Ultrasonic)

APPLICATION CHALLENGE



โปรแกรม EDU:BIT ใช้ทำงานเป็น Buzzer ใน Game Show เพื่อส่งสัญญาณว่าถูกหรือผิด	
On start	แสดงหน้ายิ้ม
เมื่อปุ่ม A ถูกกด (ปุ่มสีเหลือง)	แสดงเครื่องหมาย ✓ และเล่นเพลง “power up” 1 ครั้ง
เมื่อปุ่ม B ถูกกด (ปุ่มสีน้ำเงิน)	แสดงเครื่องหมาย ✗ และเล่นเพลง “wawawawaa” 1 ครั้ง
เมื่อปุ่ม A+B ถูกกด	เคลียร์หน้าจอ

บทที่ 4

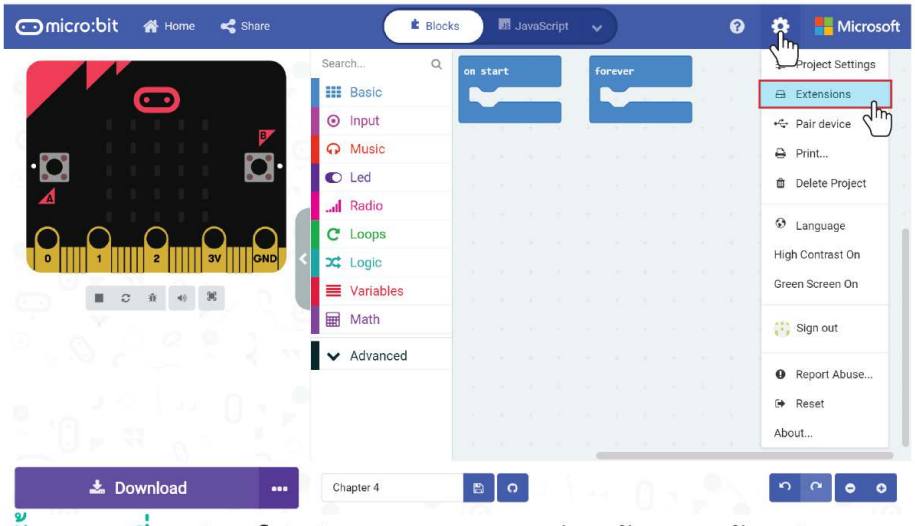
ทายซิฉันวาดอะไร

Traffic Light Bit

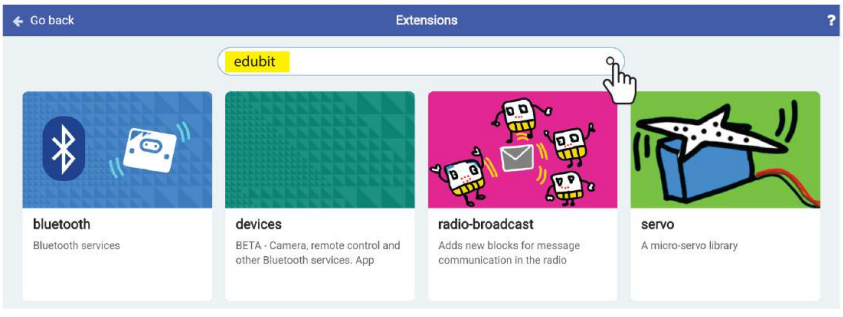
คุณเคยสังเกตุเห็น LED สีแดง สีเหลืองและสีเขียวบน EDU:BIT ของคุณไหม มันคือ Traffic Light Bit โดยคุณจะต้องเพิ่มส่วนขยาย (EDU:BIT extension) ใน MakeCode Editor ของคุณก่อนเพื่อที่จะโปรแกรมการทำงาน โดยส่วนขยาย (extensions) คือ กลุ่มของ block เฉพาะที่เพิ่มเข้ามาใน editor เพื่อง่ายต่อการเขียนโปรแกรมเพื่อควบคุมการทำงานของอุปกรณ์เสริมต่างๆของ micro:bit อย่างเช่น บอร์ด EDU:BIT

มาเขียนโปรแกรมกัน!

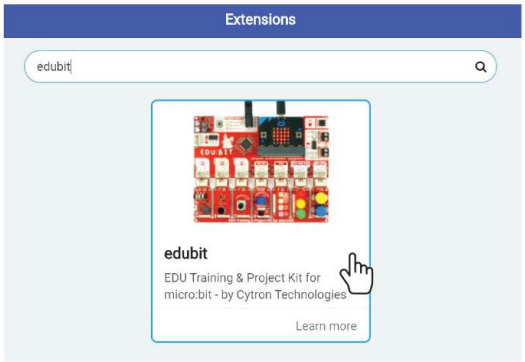
ขั้นตอนที่ 1 สร้างโปรเจ็กต์ใหม่ใน MakeCode Editor คลิกที่ไอคอน ⚙ ด้านขวาบนแล้วเลือก 'Extensions' (จำเป็นจะต้องมีอินเตอร์เน็ตสำหรับการติดตั้ง extensions)

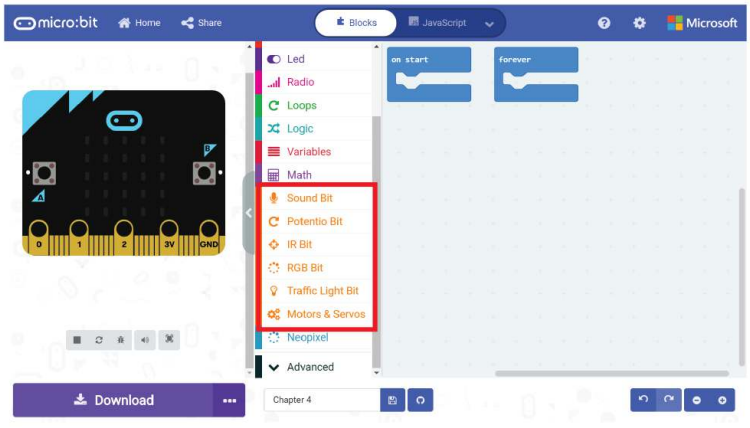


ขั้นตอนที่ 2 พิมพ์คำว่า "edubit" ในช่องค้นหาแล้วคลิก Enter

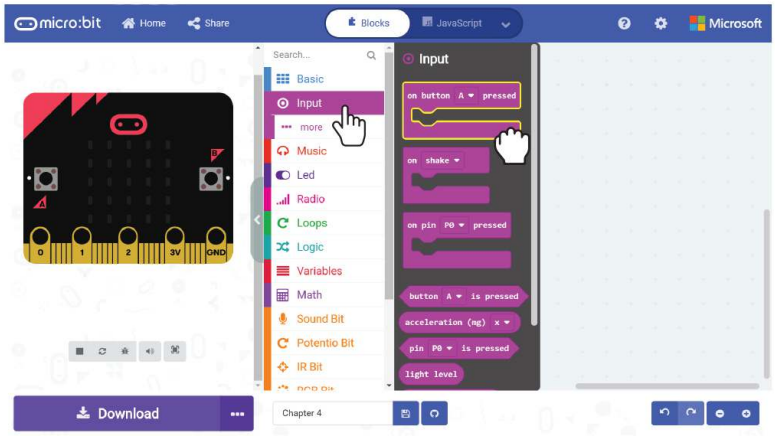


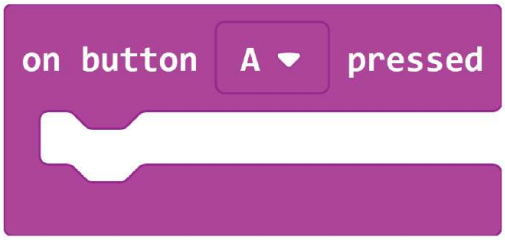
ขั้นตอนที่ 3 คลิกที่ส่วนขยาย 'edubit' รอจนกระทั่งดาว์นโหลดและติดตั้งเสร็จสิ้น tและคุณจะสังเกตุเห็นหมวดหมู่ใหม่ปรากฏขึ้นที่ Toolbox ใน MakeCode Editor ของคุณ



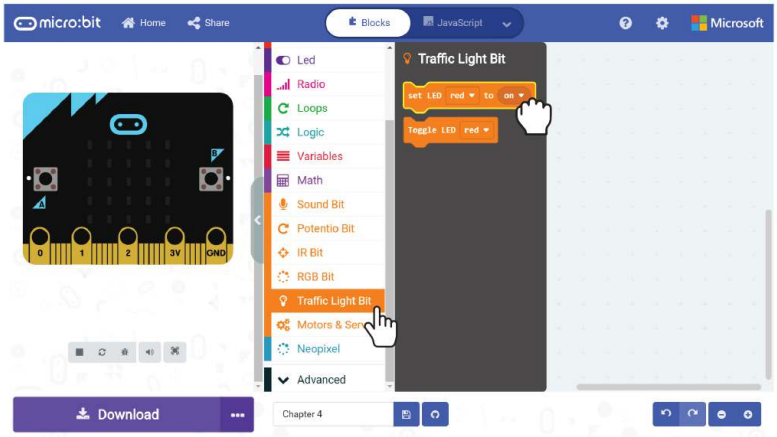


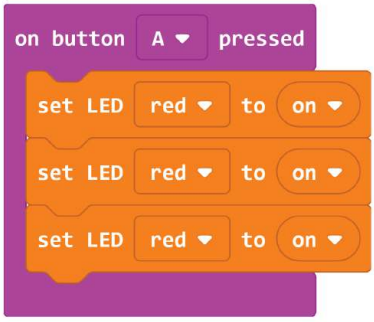
ขั้นตอนที่ 4 คลิก [Input] ใน Toolbox และเลือก [on button _ pressed] block



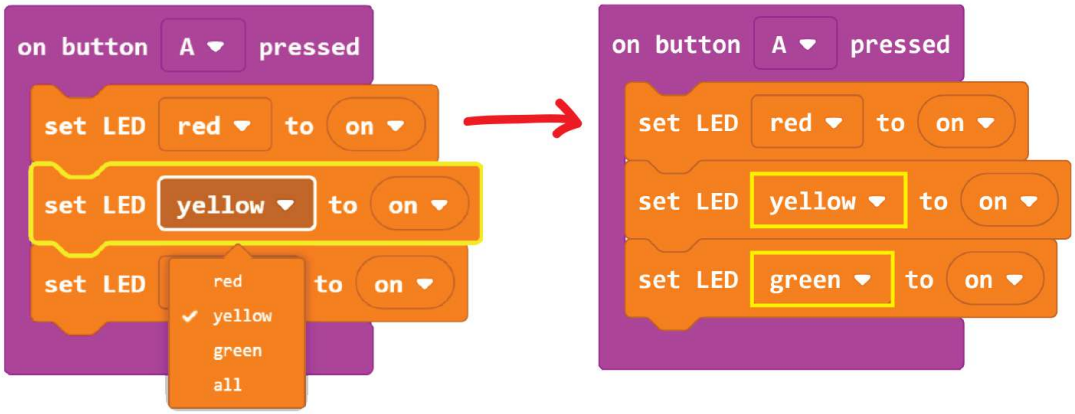


ขั้นตอนที่ 5 คลิก [Traffic Light Bit] ใน Toolbox แล้วเลือก [set LED _ to _] block และใน Workspace คลิกขวาที่ [set LED _ to _] block และทำการ 'Duplicate' จนได้ [set LED _ to _] block จำนวน 3 blocks แล้วนำทั้ง 3 blocks เข้าไปวางไว้ใน [on button A pressed] slot

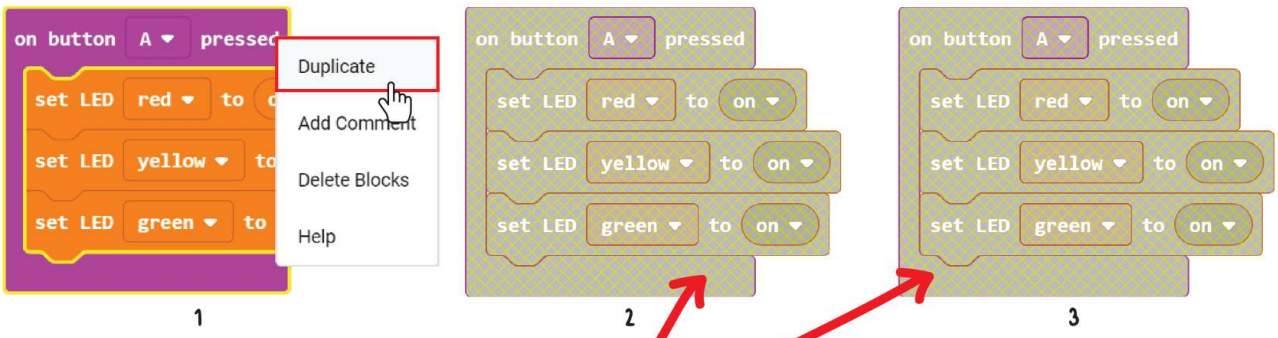




ขั้นตอนที่ 6 คลิกที่ตัวเลือกสีใน [set LED _ to _] block ของ block ที่ 2 และ 3 แล้วเปลี่ยนเป็น "Yellow" และ "Green" ตามลำดับ



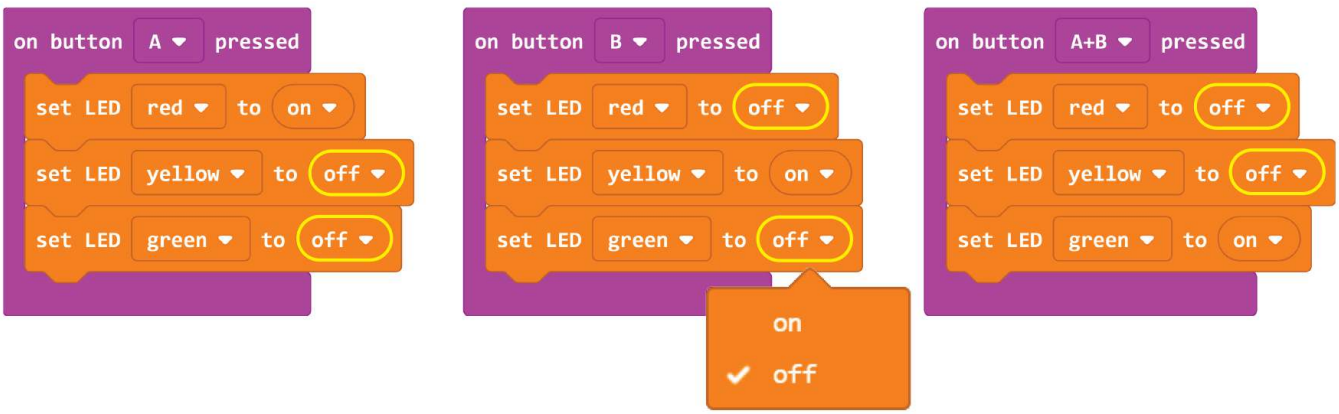
ขั้นตอนที่ 7 คลิกขวาที่ [on button _ pressed] block และเลือก 'Duplicate' จนได้กลุ่ม block ที่เหมือนกัน 3 blocks



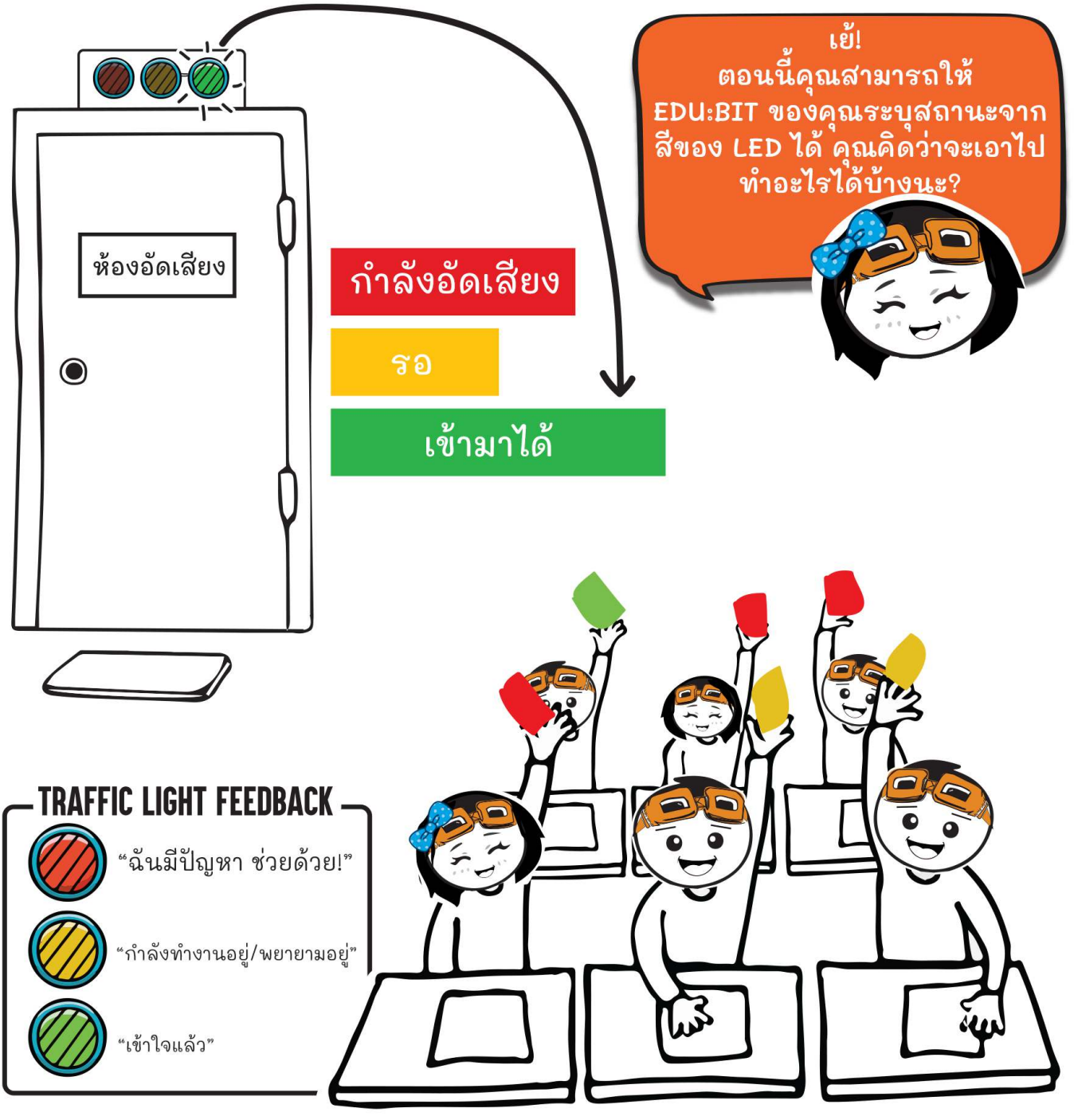
**กลุ่ม block ที่ได้จากการ 'Duplicate' อีก 2 block จะไม่ทำงานเพราะว่าเป็นเงื่อนไขที่กดปุ่ม A เหมือนกลุ่ม block แรก

ขั้นตอนที่ 8 เปลี่ยนเงื่อนไขของกลุ่ม blocks ที่ 2 และ 3 ที่ [on button _ pressed] block จาก "**A**" เป็น "**B**" และ "**A+B**" ตามลำดับ

ขั้นตอนที่ 9 เปลี่ยนสถานะการเปิด - ปิดของหลอดไฟดังตัวอย่างด้านล่าง



ขั้นตอนที่ 10 แฟลชโค้ดลง EDU:BIT ของคุณ และสังเกตุว่าเกิดอะไรขึ้นเมื่อกดปุ่ม A, ปุ่ม B และปุ่ม A+B พร้อมกัน



LED หรือ light emitting diode เป็นตัวอย่างของอุปกรณ์ Digital Output โดยมีแค่ 2 สถานะ คือ เปิด และ ปิด โดยที่ Output แบบ Digital เปิด เท่ากับ 1 และ ปิด เท่ากับ 0

คุณสามารถโปรแกรม EDU:BIT ของคุณเป็นอุปกรณ์ timer indicator ได้โดยมีตัวอย่างโค้ด ดังนี้

timer ถูกกระตุ้นให้เริ่มนับถอยหลัง โดยการสั่นของ EDU:BIT

เล่นเพลงเพื่อเป็นสัญญาณว่าเริ่มนับถอยหลังแล้ว

หลอด LED สีเขียวติดขึ้น

หลอด LED สีเหลืองติดขึ้น

หลอด LED สีแดงติดขึ้น

เล่นเพลง “wawawawaa” เพื่อเป็นสัญญาณว่าเวลาหมดแล้ว

หลอด LED สีแดงกระพริบ 10 ครั้ง

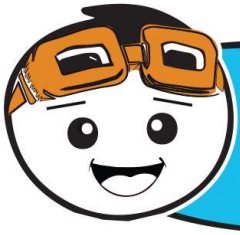
```
on shake
play tone Middle C for 1 beat
set LED red to off
set LED yellow to off
set LED green to on
pause (ms) 2000
set LED red to off
set LED yellow to on
set LED green to off
pause (ms) 2000
set LED red to on
set LED yellow to off
set LED green to off
pause (ms) 2000
start melody wawawawaa repeating once
repeat 10 times
do
    Toggle LED red
    pause (ms) 500
```

ในโค้ดตัวอย่างนี้ LED แต่ละดวง จะติดหลอดละ 2000 ms (2 วินาที)

หากคุณต้องการให้ LED แต่ละดวง ติดหลอดละ 1 นาที ควรกำหนดค่าใน [pause _] block เป็นเท่าไหร่

1 นาที = 60 วินาที

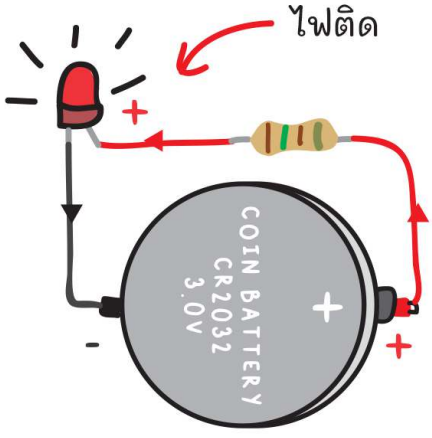
```
Toggle LED red
```

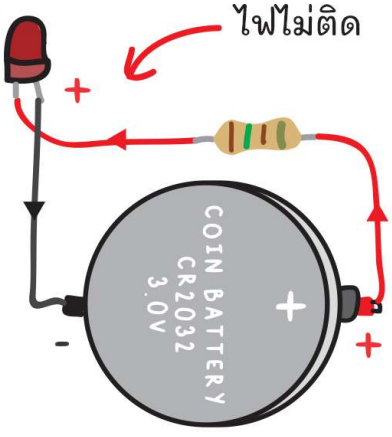


“Toggle” คือการเปลี่ยนแปลงสถานะหนึ่งเป็นอีกสถานะหนึ่ง ยกตัวอย่าง เช่น ถ้าสถานะปัจจุบันคือ ON เมื่อ toggle สถานะก็จะกลายเป็น OFF ในทางกลับกัน หากเรา toggle สถานะ LED อย่างรวดเร็ว ก็จะปรากฏลักษณะ LED ที่กระพริบอย่างรวดเร็ว

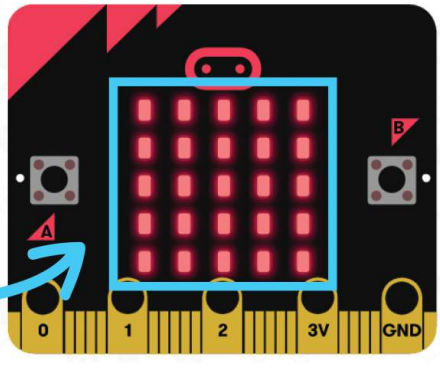
FUN FACT!

light-emitting diode (LED) คือ อุปกรณ์กึ่งตัวนำที่สามารถสร้างแสงสว่างได้จากไฟฟ้า โดยประกอบไปด้วย 2 ขั้ว คือ ขั้วบวก และ ขั้วลบ โดยเมื่อต่อ LED กับแบตเตอรี่ถูกขั้ว จะเกิดกระแสไหลผ่าน LED จนเกิดแสงสว่างขึ้น





LEDs ที่ใช้บน microbit เป็นประเภท surface-mount technology (SMT) จึงมีขนาดที่เล็กมากๆ



นอกเหนือจาก LED พวกนี้แล้ว ยังมี SMT LEDs อีก 41 หลอด บน EDU:BIT อีกนะ ลองหาดูสิ!

youtu.be/qqBmvHD5bCw

APPLICATION CHALLENGE

โปรแกรม EDU:BIT ให้ทำงานเป็นเครื่องนับคะแนนและเครื่องจับเวลาไปพร้อมๆกันเพื่อใช้ในการเล่นเกม “ทายซิฉันวาดอะไร. หรือเกม “ใบ้คำ”	
On start	กำหนดตัวแปร Team A = 0 กำหนดตัวแปร TEAM B = 0
On Button A pressed (ปุ่มสีเหลือง)	บวกคะแนน TEAM A คะแนนและแสดงคะแนนปัจจุบันของ TEAM A
On Button B pressed (ปุ่มสีน้ำเงิน)	บวกคะแนน TEAM B คะแนนและแสดงคะแนนปัจจุบันของ TEAM B
On Button A+B pressed (ปุ่มสีน้ำเงิน+ปุ่มสีเหลือง)	แสดงคะแนนปัจจุบันของ TEAM A และ TEAM B
On shake	เริ่มนับถอยหลัง 1 นาที โดยเมื่อเหลือ 30 วินาที LED สีเขียวจะติด ,เหลือ 20 วินาที LED สีเหลืองจะติด และ สุดท้ายเหลือ 10 วินาทีสุดท้าย LED สีแดงจะติดและจะเล่นเพลง “wawawawaa” เมื่อเวลาหมดลงพร้อมกับ LED สีแดงกระพริบอีก 10 ครั้ง

เคล็ดลับ
คุณต้องสร้างตัวแปรขึ้นมาใหม่ 2 ตัวแปร
โดยตั้งชื่อว่า Team A และ Team B ตามลำดับ

Let's Play

Win, Lose or Draw~

HOW TO PLAY:

- แบ่งทีมออกเป็น 2 ทีม คือ Team A และ Team B
- สมาชิกจากทีม A 1 คนจะออกมาสุ่มหยิบการ์ดที่มีชื่อของสิ่งที่จะวาดเขียนอยู่ ให้อ่านจนเข้าใจว่าต้องวาดอะไร แล้วทำการเขย่า EDU:BIT เพื่อเริ่มจับเวลา (นับถอยหลัง 1 นาที)
- เริ่มวาดรูปบนกระดานเพื่อให้สมาชิกในทีมทายได้เลย โดยห้ามพูดหรือบอกใบ้โดยใช้ท่าทางต่างๆ
- หากมีสมาชิกใน Team A สามารถทายได้ถูกว่าวาดรูปอะไรก่อนหมดเวลา Team A ก็จะได้รับคะแนน 1 คะแนน (โดยการกดปุ่ม A หรือ ปุ่มสีเหลือง)
- แต่หาก Team A ไม่มีคนตอบได้ในระยะเวลาที่กำหนด Team B สามารถ “steal” เพื่อตอบได้โดยหากตอบถูกก็จะได้รับคะแนน 1 คะแนน
- ทั้งสองทีมสลับกันเล่นจนกระทั่งจบเกม
- ทีมที่ได้คะแนนมากที่สุดเป็นฝ่ายชนะ

NOTE!

Scan Here

สแกนที่ QR Code เพื่อดาวน์โหลดการ์ดคำศัพท์ที่ใช้ในเกม หรือหากคุณไม่อยากวาดรูป ลองเปลี่ยนเป็นเกมใบ้คำดูสิ กฎกติกาเหมือนกับเกมทายรูปเลย

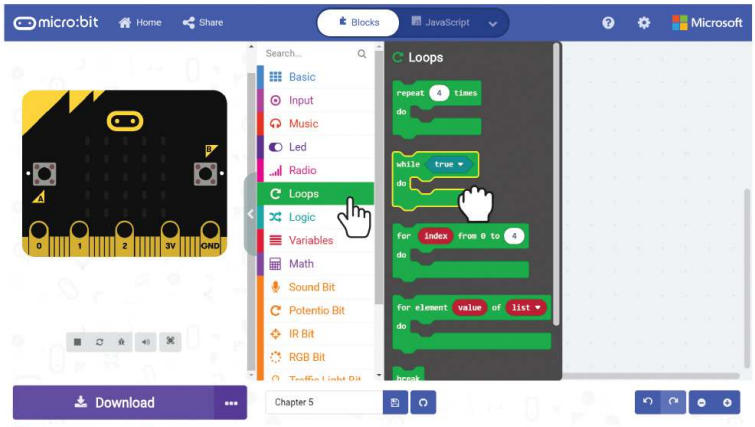
บทที่ 5

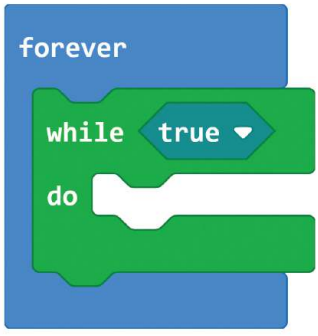
ลูกเต๋าอินฟาเรดดิจิตอล

IR Bit

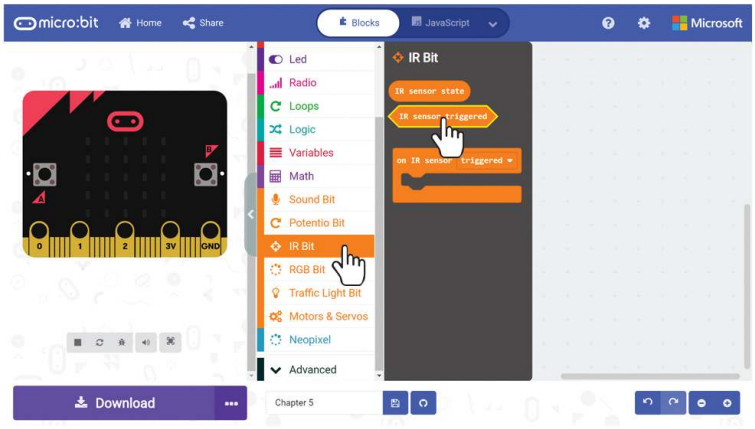
มาเขียนโปรแกรมกัน!

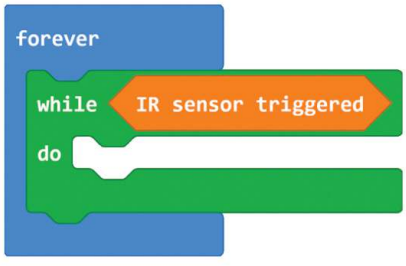
ขั้นตอนที่ 1 สร้างโปรเจ็กต์ใหม่ใน MakeCode Editor ของคุณ และเพิ่มส่วนขยาย EDU:BIT (ทำตามขั้นตอนในหน้าที่ 40) คลิกที่ [Loops] ใน Toolbox และเลือก [while _ do] block แล้วนำไปวางใน [forever] slot



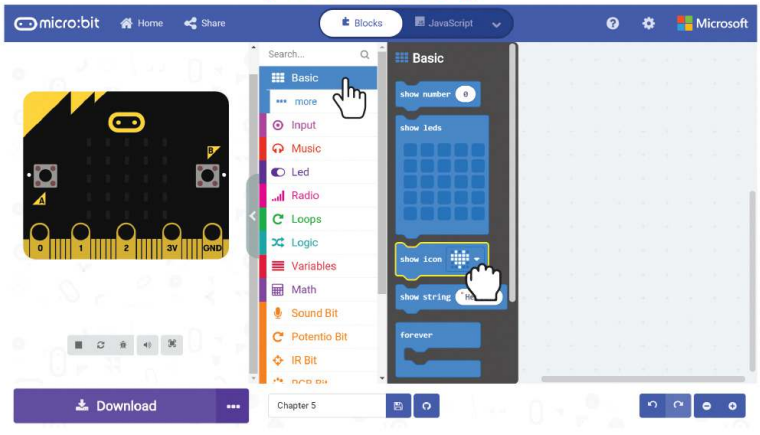


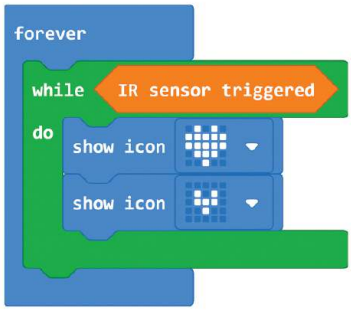
ขั้นตอนที่ 2 คลิก [IR Bit] ใน Toolbox และเลือก [IR sensor triggered] block แล้วนำไปวางใน slot เงื่อนไขของ [while _ do] block



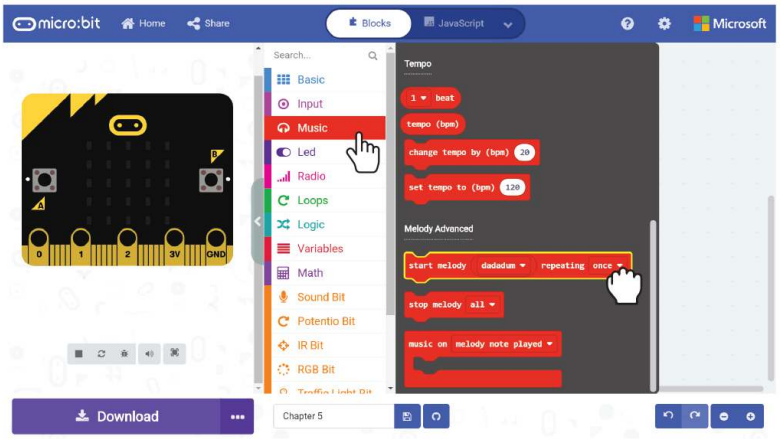


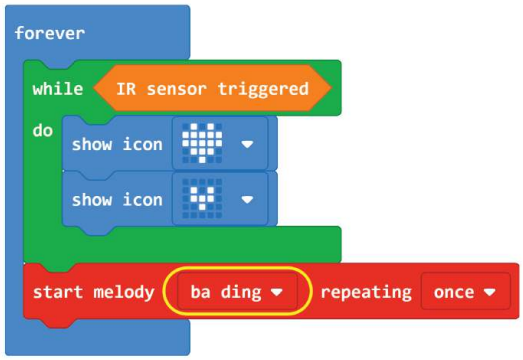
ขั้นตอนที่ 3 คลิก [Basic] ใน Toolbox และเลือก [show icon] block ออกมา 2 block แล้วเปลี่ยนรูปไอคอนที่จะแสดงของ [show icon] block ที่ 2 ให้เป็นรูปหัวใจดวงเล็ก แล้วนำไปวางใน [while _ do] block



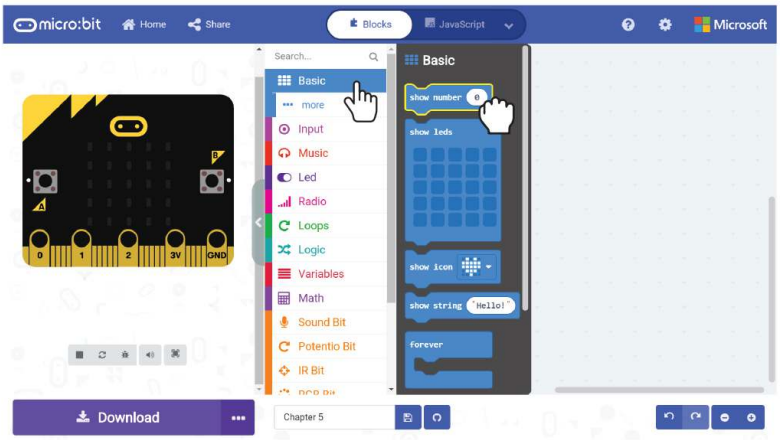


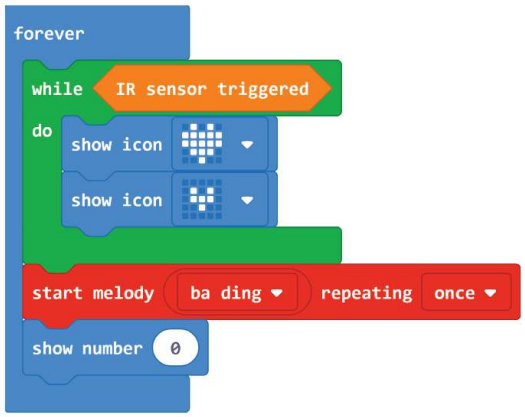
ขั้นตอนที่ 4 คลิก [Music] ใน Toolbox และเลือก [start melody _ repeating _] block แล้วเปลี่ยนเพลงจาก “dadadum” ให้เป็นเพลง “ba ding”



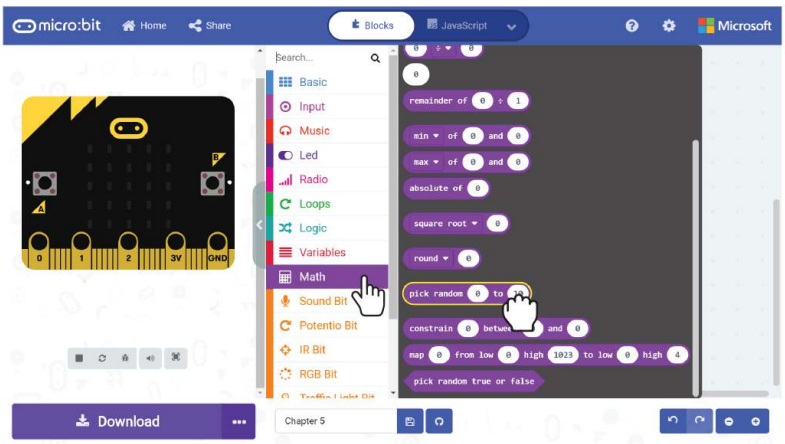


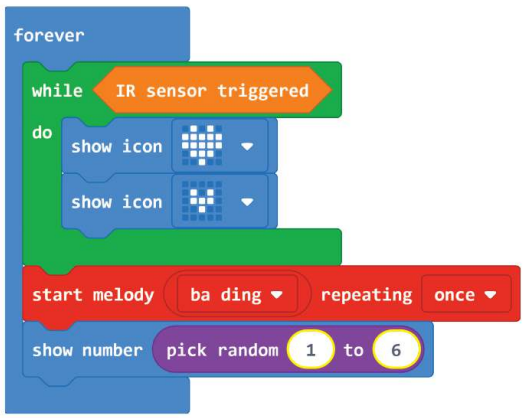
ขั้นตอนที่ 5 คลิก [Basic] ใน Toolbox และเลือก [show number] block



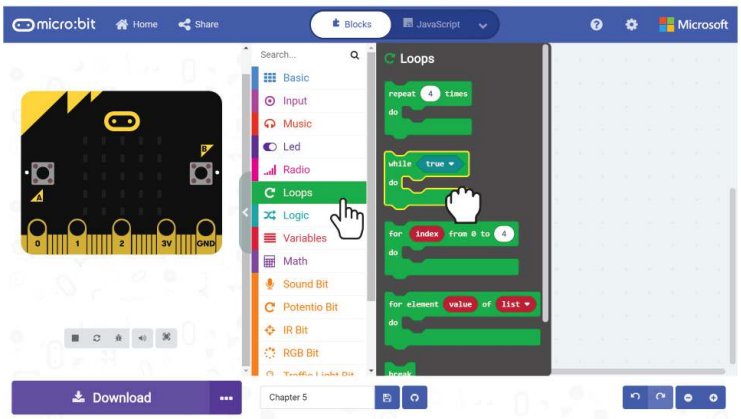


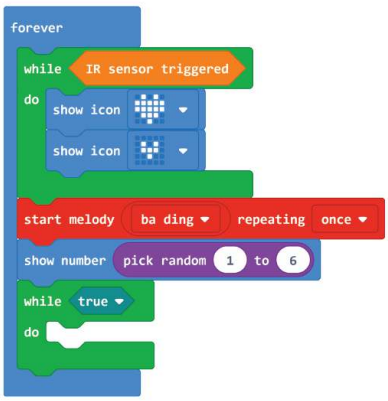
ขั้นตอนที่ 6 คลิก [Math] ใน Toolbox แบะเลือก [pick random _ to _] block แล้วตั้งค่าตัวเลขให้เป็น “1” to “6”



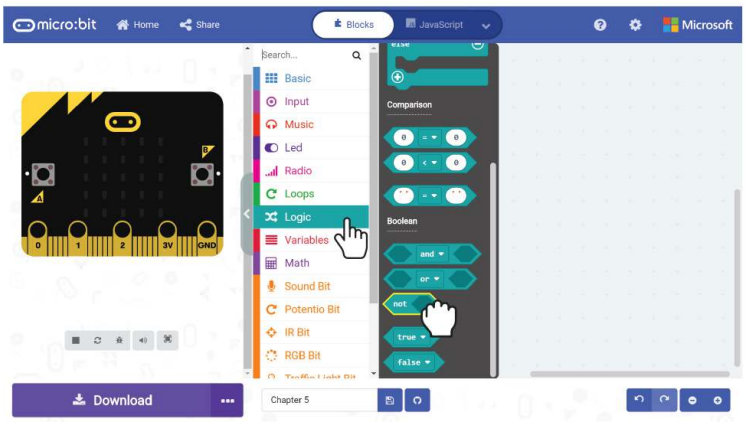


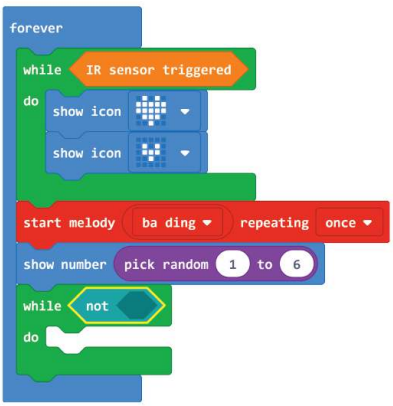
ขั้นตอนที่ 7 คลิก [Loops] ใน Toolbox และเลือก [while _ do _] block



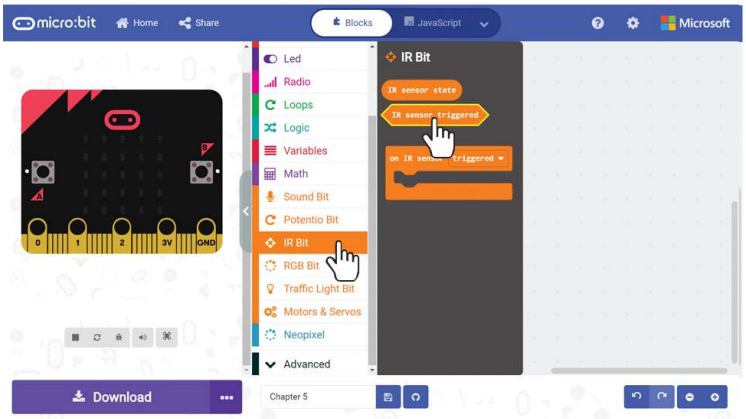


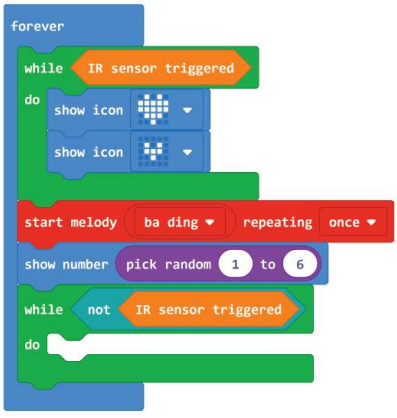
ขั้นตอนที่ 8 คลิก [Logic] ใน Toolbox และเลือก Boolean [not] block แล้วนำไปวางในเงื่อนไขของ [while _ do] block





ขั้นตอนที่ 9 คลิก [IR Bit] ใน Toolbox และเลือก [IR sensor triggered] block แล้วนำไปวางใน slot ว่างของ [not] block

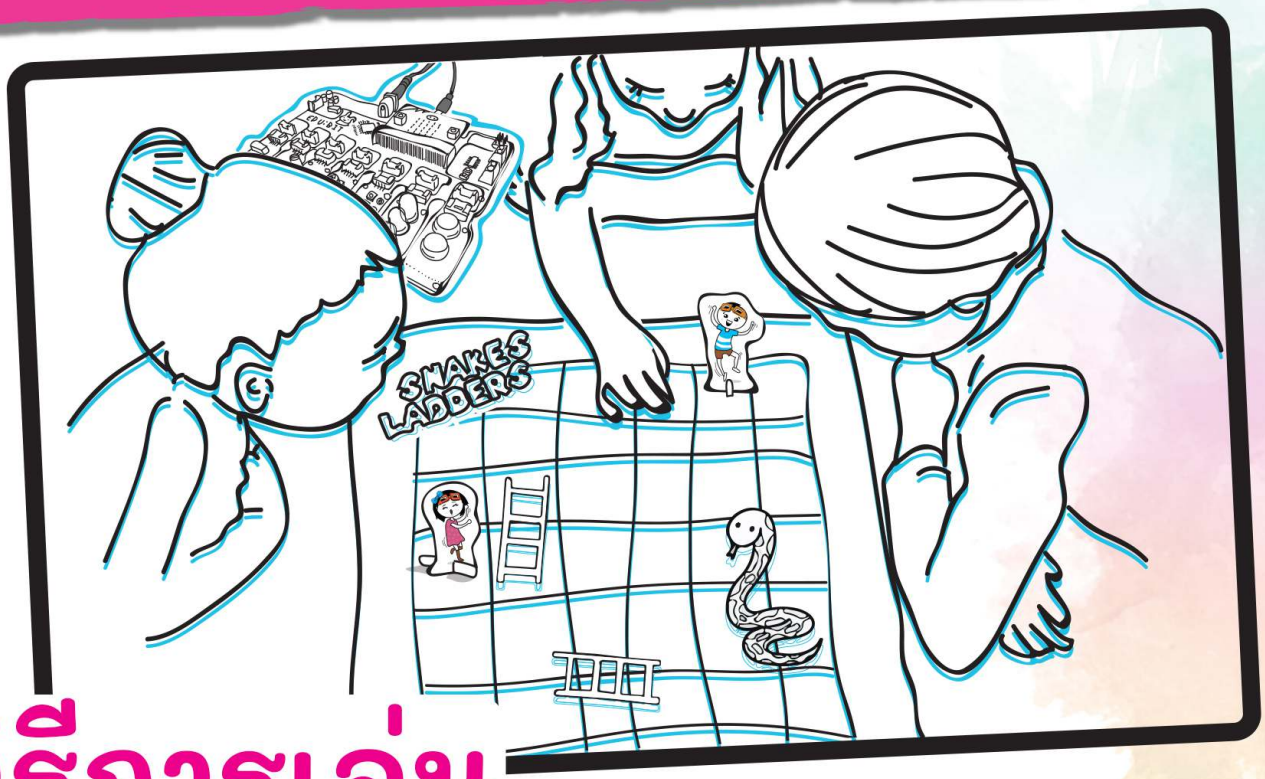




ขั้นตอนที่ 10 แฟลชโค้ดลง EDU:BIT ของคุณ

Let's Play

เกมบันไดงู



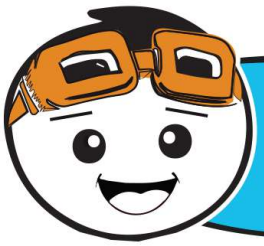
วิธีการเล่น

- ผู้เล่นแต่ละคนเลือกหมากตัวละครแล้วนำหมากวางไว้ที่จุดเริ่มต้น
- ผู้เล่นแต่ละคนเริ่มทอยลูกเต๋าโดยการวางฝ่ามือไว้ด้านบนเหนือ IR Bit จนกระทั่งเห็นไอคอนรูปหัวใจเต้นให้นำมือออก
- เดินหมากของคุณตามตัวเลขที่ปรากฏบน LED matrix (ระหว่าง 1 ถึง 6)
- ถ้าตัวหมากของคุณเดินไปหยุดบริเวณช่องที่เป็นด้านล่างของบรรไดให้ทำการเดินหมากของคุณข้ามไปยังบริเวณด้านบนของบรรไดได้ทันที แต่หากหมากของคุณเดินไปบริเวณช่องที่เป็นหัวของงู คุณจะต้องเดินหมากของคุณลงมาบริเวณหางของงูทันที
- ผู้เล่นคนแรกที่เดินหมากถึงช่องที่ 100 ก่อนเป็นผู้ชนะ

NOTE!

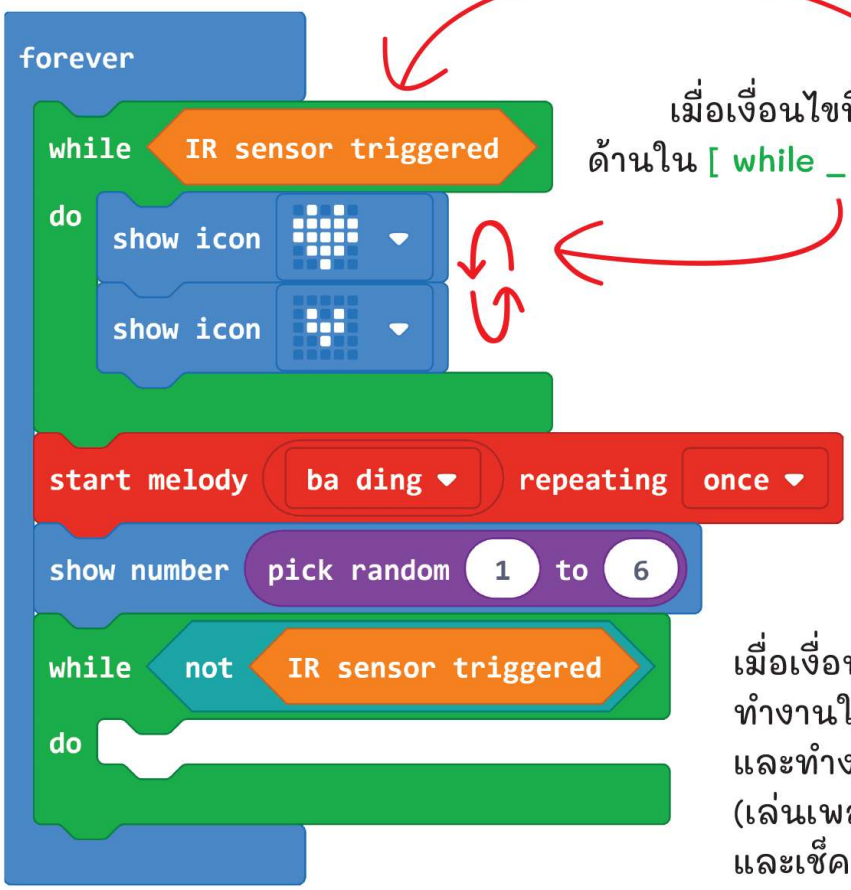
กระดานและหมากของเกมบรรไดงูมีให้อยู่ในกล่อง โดยแกะตัวละครและแท่นออกจากแผ่นแล้วนำมาประกอบกันให้เป็นหมากดังภาพ

BREAK THE CODE



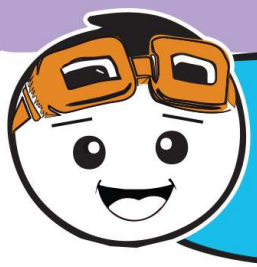
เราใช้ [while _ do] block จาก [Loops] Toolbox ในโค้ดก่อนหน้านี้ คุณรู้รึเปล่าว่า while loop ทำงานอย่างไร?

เมื่อโปรแกรมทำงานถึง [while _ do] block โปรแกรมจะทำการตรวจสอบเงื่อนไขใน [while _ do] block (เงื่อนไขต้องเป็น TRUE) โปรแกรมถึงจะทำงานในส่วน block ที่อยู่ใน [while _ do] block โดยจะทำงานวนซ้ำไปเรื่อยๆ จนกระทั่งเมื่อเงื่อนไขที่ตรวจสอบนั้นกลายเป็นเท็จ (FALSE) โปรแกรมจะออกจากการทำงานของ block ที่อยู่ใน [while _ do] block และทำงานในส่วน block ถัดไป



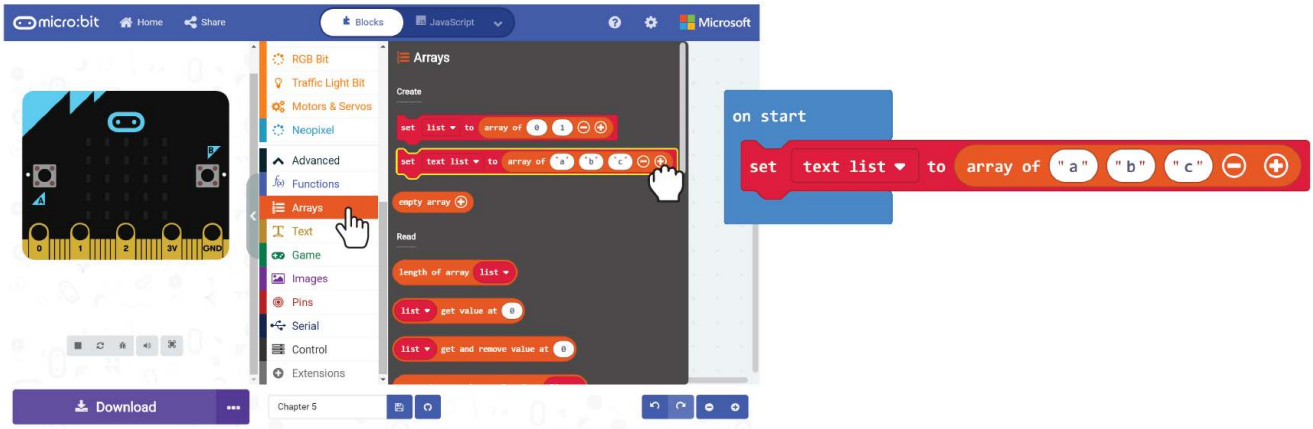
เมื่อเงื่อนไขนี้เป็น true จะทำงานคำสั่งด้านใน [while _ do] block (แสดงรูปหัวใจเต้น)

เมื่อเงื่อนไขเป็น false จะออกจากการทำงานใน [while _ do] block ด้านบน และทำงานใน block ที่ถัดลงมาแทน (เล่นเพลง, สุ่มตัวเลขและแสดงตัวเลข และเช็ค IR sensor)

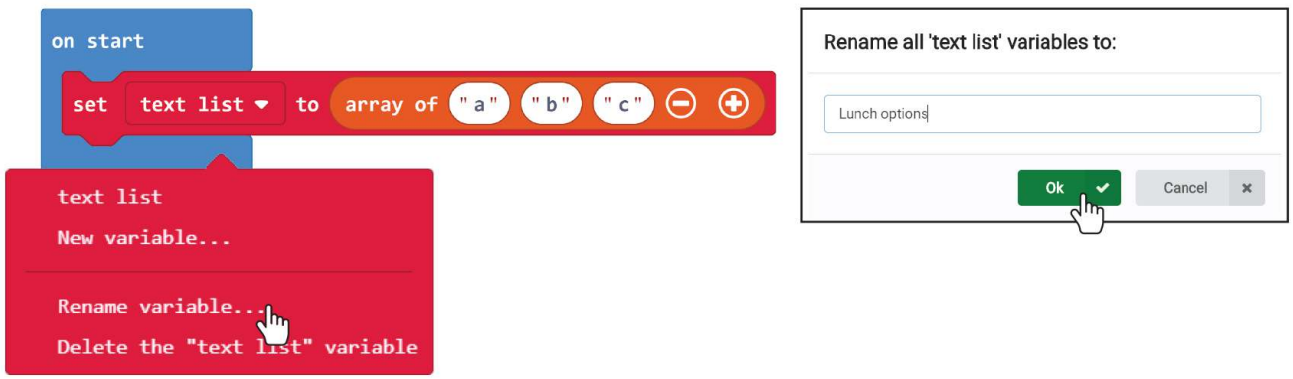


นอกจากการใช้ EDU:BIT เป็นลูกเต๋าดิจิตอลแล้ว คุณสามารถปรับแต่งโค้ดก่อนหน้านี้เพื่อให้ EDU:BIT ช่วยคุณตัดสินใจตัวเลือกที่ตัดสินใจได้ยาก ยกตัวอย่างเช่น กลางวันนี้จะกินอะไรดีน้า?

ขั้นตอนที่ 11 คลิก [Advance] : [Arrays] ใน Toolbox และเลือก [set text list to array of _ _ _] block แล้วนำไปวางที่ [Basic] : [on start] slot

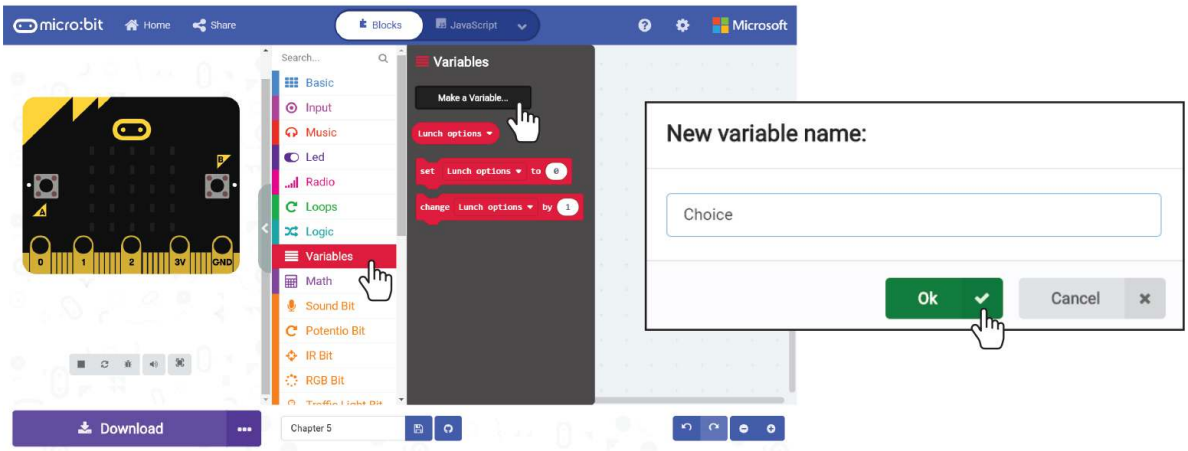


ขั้นตอนที่ 12 คลิก [text list] block และเลือก “Rename Variable” และตั้งชื่อใหม่ในหน้าต่างป๊อป-อัพที่ปรากฏขึ้นมาว่า ‘Lunch options’ แล้วคลิก OK

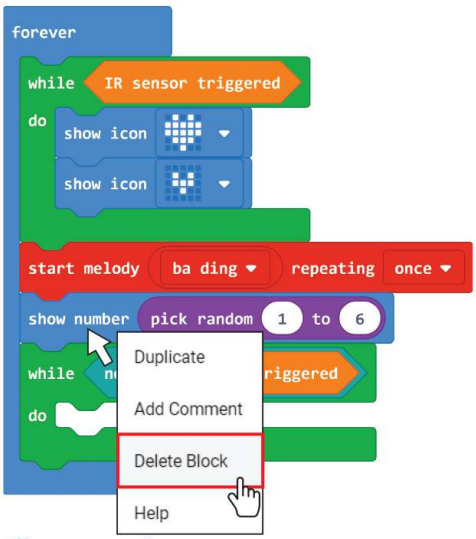


ขั้นตอนที่ 13 คลิกบน [array of _ _ _] block ทีละช่อง แล้วใส่เมนูที่ต้องการให้เลือกในแต่ละ block

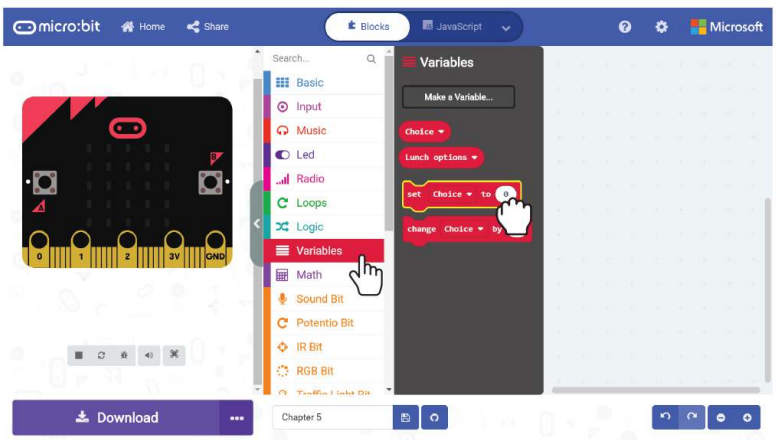
ขั้นตอนที่ 14 คลิก [Variable] ใน Toolbox และทำการสร้างตัวแปรใหม่ขึ้นมาโดยให้มีชื่อว่า 'Choice'

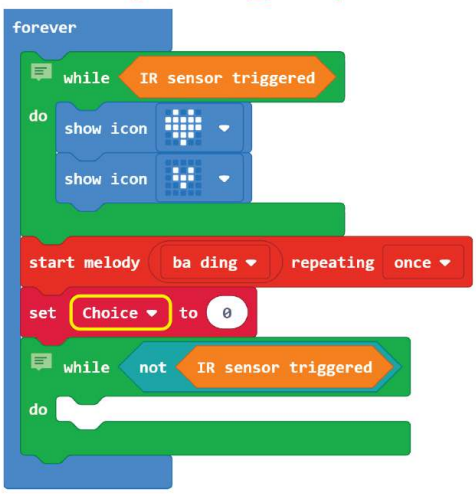


ขั้นตอนที่ 15 คลิกขวาที่ [show number [pick random _ to _]] block และเลือก 'Delete Blocks'

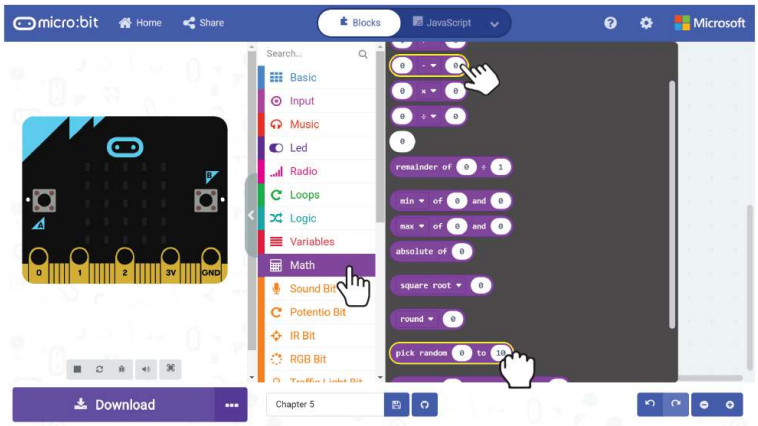


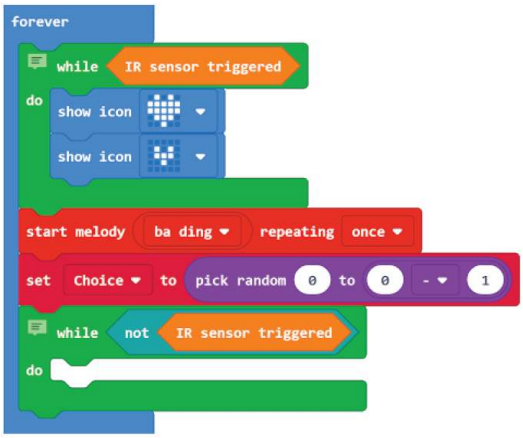
ขั้นตอนที่ 16 คลิก [Variable] ใน Toolbox และเลือก [set _ to _] block แล้วนำไปวางไว้ระหว่าง [start melody _ repeating _] block และ [while _ do] block แล้วทำการเปลี่ยนตัวแปรเป็น 'Choice'



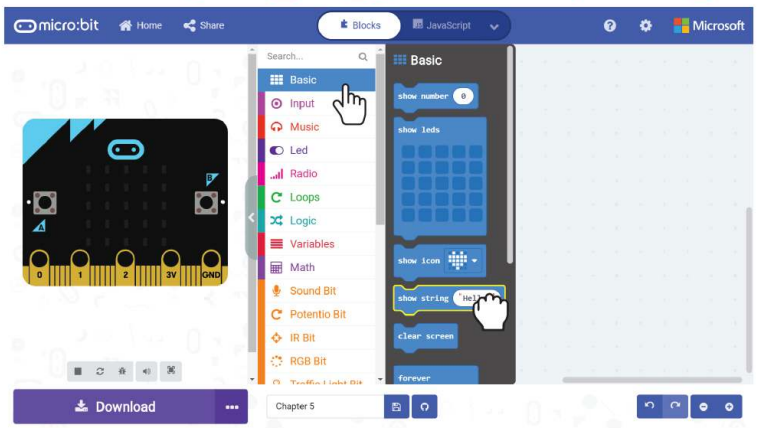


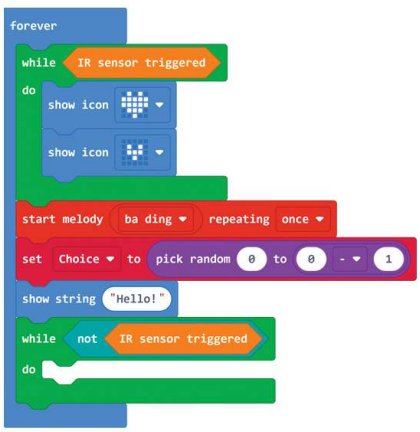
ขั้นตอนที่ 17 คลิก [Math] ใน Toolbox และเลือก [pick random _ to _] และ [_ - _] block แล้วนำไปวางใน [set _ to _] block แล้วทำการเปลี่ยนค่าสุดท้ายให้เป็น 1



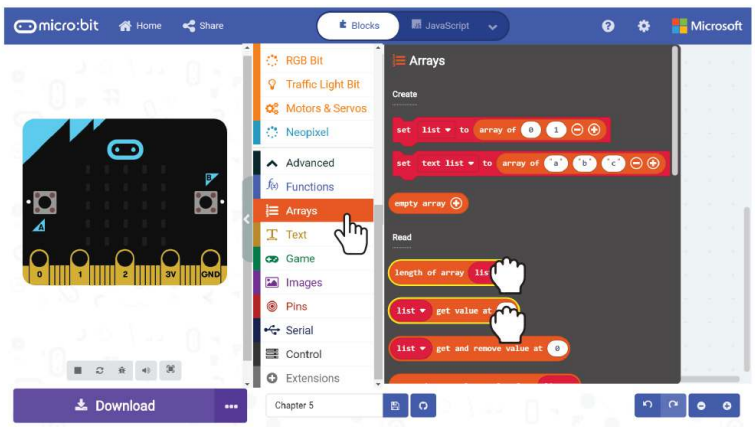


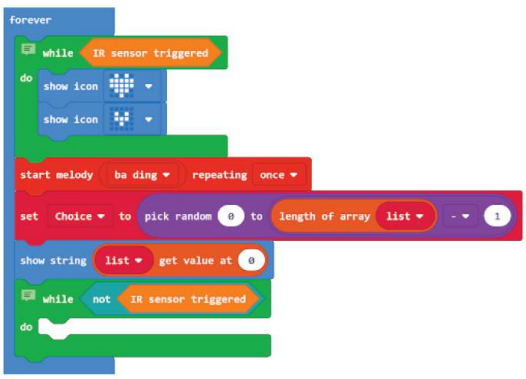
ขั้นตอนที่ 18 คลิก [Basic] ใน Toolbox และเลือก [show string] block แล้วนำไปวางระหว่าง [set _ to _] block และ [while _ do] block ดังภาพ



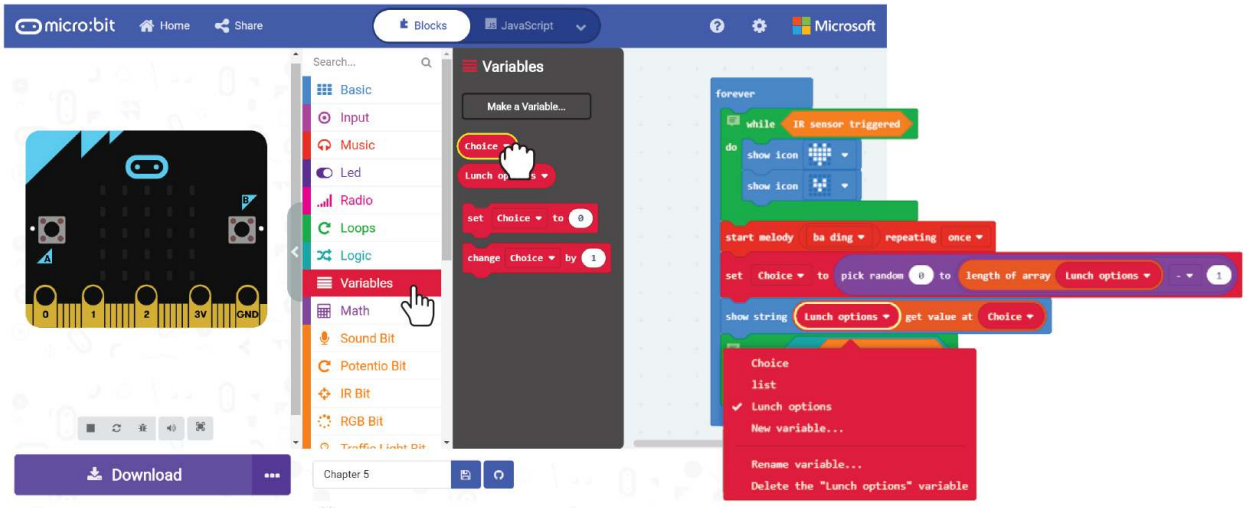


ขั้นตอนที่ 19 คลิก [Advanced] : [Arrays] ใน Toolbox และเลือก [length of array_] นำไปใส่ใน [pick random _ to _] และ [_ get value at _] block นำไปใส่ใน [show string] block ดังภาพ

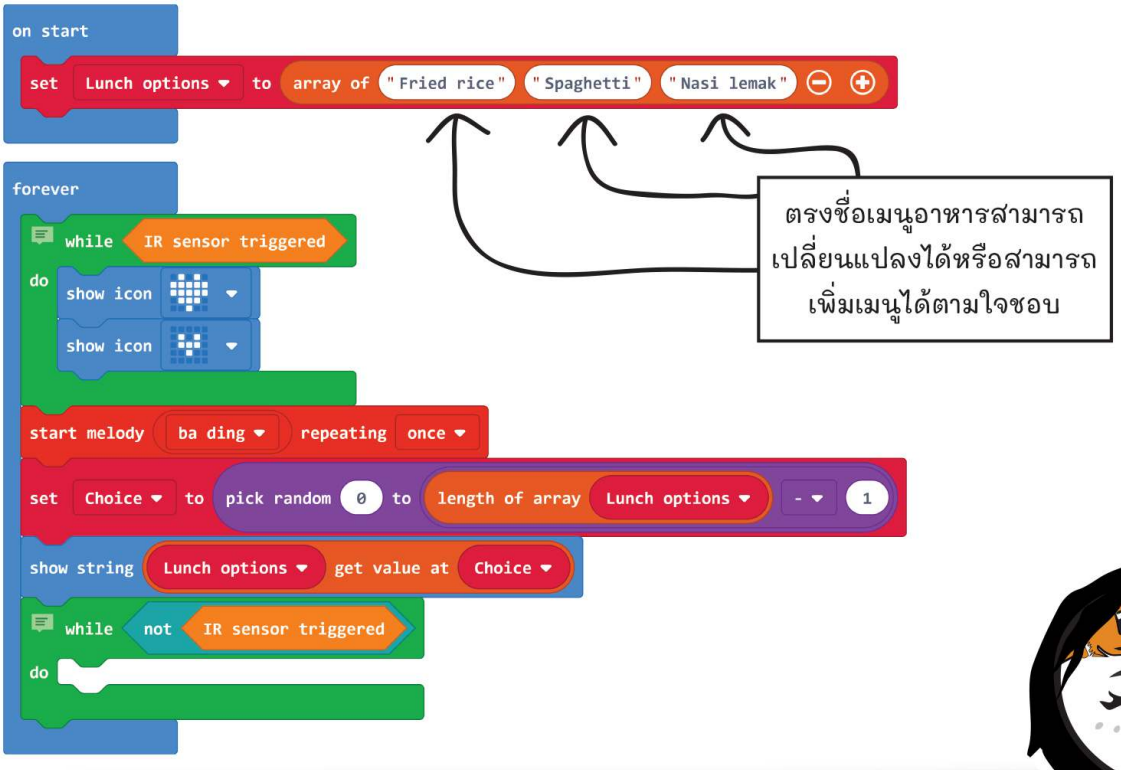




ขั้นตอนที่ 20 คลิก [list] และเลือกตัวแปรที่จะแสดงเป็น 'Lunch options' สำหรับทั้งสอง block และสุดท้าย คลิก [Variable] ใน Toolbox และเลือก [Choice] block แล้วนำไปวางใน slot ที่ว่างของ [_ get value at _] block



นี่คือโค้ด "กลางวันนี้กินอะไรดีน้า" ที่เสร็จสมบูรณ์แล้ว

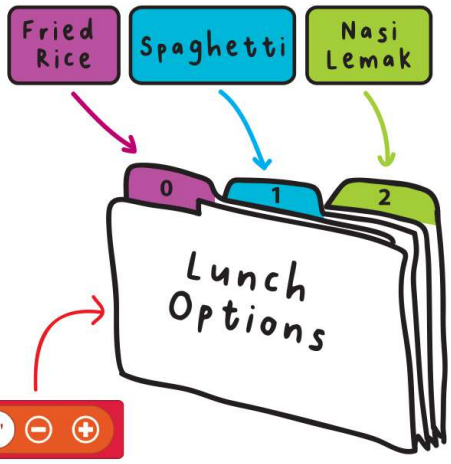


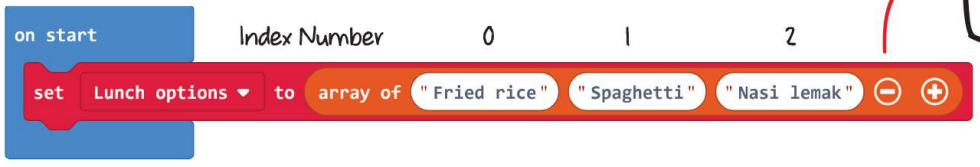
ต่อไปหากคุณไม่แน่ใจว่าจะกินอะไรดี ปล่อยให้ EDU:BIT ตัดสินใจให้คุณโดยการนำมือไปวางเหนือ IR Bit แล้วนำมือออก นอกจากนี้คุณยังสามารถดัดแปลงโค้ดนี้ให้ช่วยตัดสินใจว่าคุณจะเล่นเกมอะไรกับเพื่อนๆดี รู้มั้ยว่าต้องเปลี่ยนส่วนไหนของโค้ดเพื่อทำแบบนั้น??

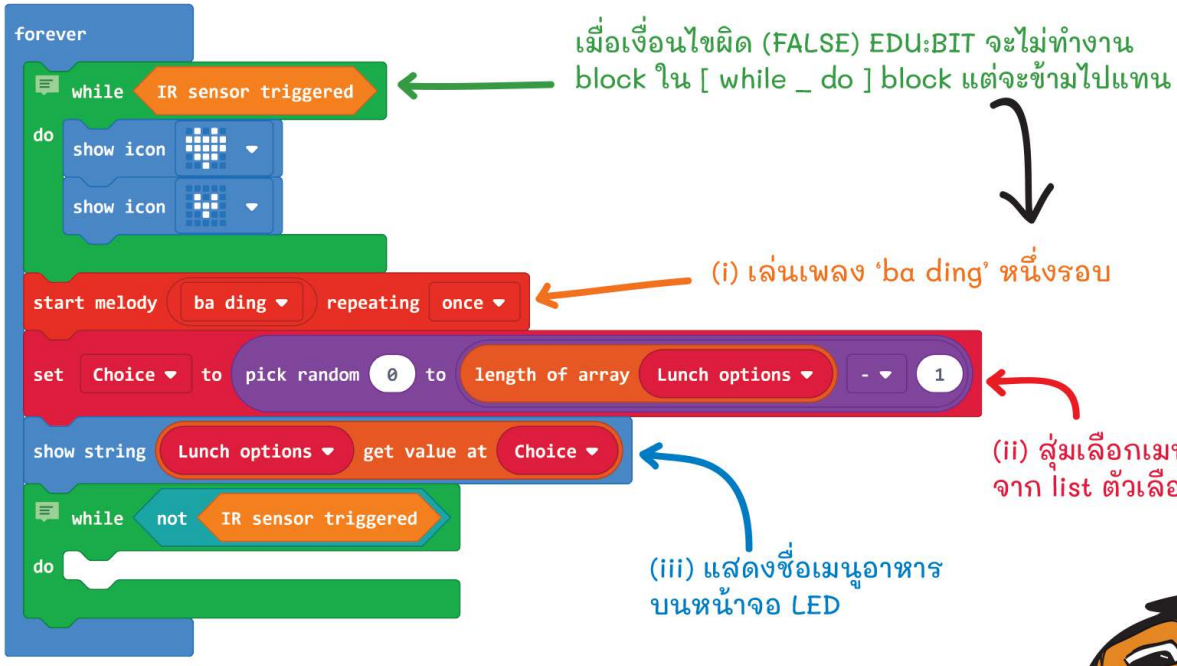
BREAK THE CODE

อาเรย์ (array) คือ list หรือคอลเลคชั่นกลุ่มของตัวแปรที่เกี่ยวข้องกัน เปรียบได้กับแฟ้มเอกสารที่แบ่งออกเป็นส่วนต่างๆ และแต่ละส่วนจะมีข้อมูลอยู่ เหตุผลของการใช้อาเรย์ก็เพราะว่าเราสามารถแก้ไขโค้ดของเราได้ง่ายเมื่อเราต้องการที่จะเพิ่มหรือลดองค์ประกอบภายใน list

ดังโค้ดด้านล่าง ยกตัวอย่าง เราสร้างอาเรย์ขึ้นมาพร้อมกับ 3 element ข้างใน และตั้งชื่ออาเรย์นั้นว่า “Lunch options” เราสามารถแก้ไขรายชื่ออาหารได้โดยการบ่งชี้ตำแหน่งของแต่ละ element อีกทั้งยังสามารถเพิ่ม หรือ ลด จำนวนของ element ใน list ได้โดยการกดที่ปุ่ม ⊕ หรือ ⊖







เมื่อเงื่อนไขผิด (FALSE) EDU:BIT จะไม่ทำงาน block ใน [while _ do] block แต่จะข้ามไปแทน

(i) เล่นเพลง ‘ba ding’ หนึ่งรอบ

(ii) สุ่มเลือกเมนูอาหารจาก list ตัวเลือก

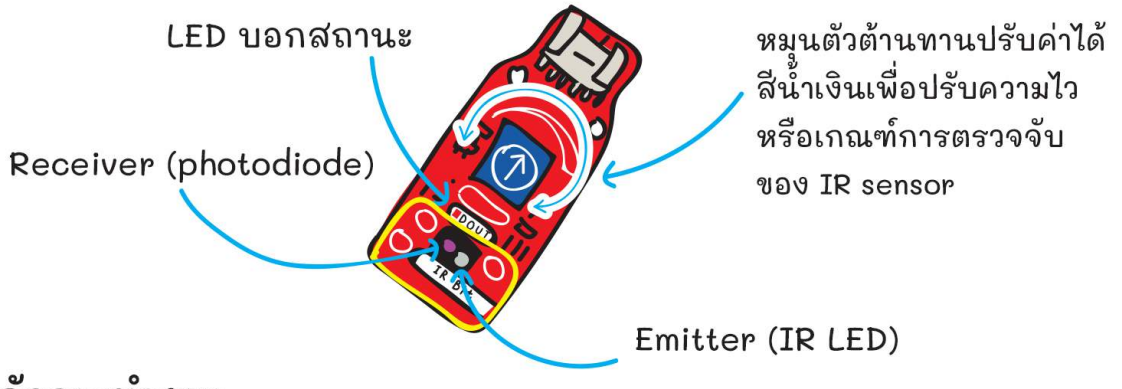
(iii) แสดงชื่อเมนูอาหารบนหน้าจอ LED

ในการเขียนโปรแกรม เราจะเริ่มนับเลขจาก 0 ไม่ใช่จาก 1 จากตัวอย่าง “Fried rice” มีเลข index เท่ากับ 0 ใน list และ “Nasi lemak” ที่เป็นเมนูสุดท้ายใน list มีเลข index เท่ากับ 2 ถึงแม้ว่าจะเป็นเมนูที่ 3 ใน list ก็ตาม

FUN FACT!

infrared (IR) sensor

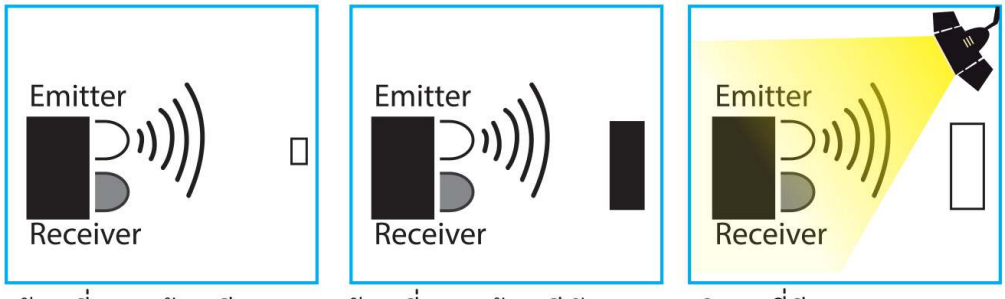
เป็นอุปกรณ์อิเล็กทรอนิกส์ที่ใช้ในการตรวจจับสิ่งกีดขวางโดย IR sensor จะมีส่วนประกอบแบ่งออกเป็น 2 ส่วน ได้แก่ ส่วน emitter (IR LED) และ ส่วน receiver (photodiode)



หลักการทำงาน

IR LED จะเปล่งแสงอินฟาเรดซึ่งจะไปสะท้อนกับวัตถุแล้วกลับมายัง receiver หากมีวัตถุอยู่ด้านหน้าเซ็นเซอร์ IR Bit จะถูก “triggered” โดยปริมาณแสงที่สะท้อนกลับมามีค่าเกินเกณฑ์ที่ได้ทำการตั้งไว้ โดยเมื่อ triggered แล้ว LED บอกสถานะบน IR Bit จะสว่างขึ้น

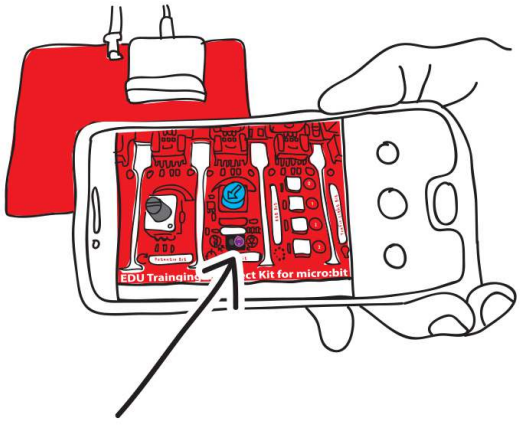
ถ้าไม่มีวัตถุอยู่ด้านหน้าเซ็นเซอร์หรืออยู่ห่างออกไปมาก แสงอินฟาเรดที่สะท้อนกลับมาสู่ receiver ก็จะน้อยมากหรือไม่มีเลย ด้วยเหตุนี้ IR Bit ก็จะไม่ถูก triggered อย่างไรก็ตาม IR sensor อาจทำงานไม่ปกติหรือไม่ทำงานหากอยู่ในสภาพแวดล้อมดังต่อไปนี้



วัตถุที่จะสะท้อนมีขนาดเล็กเกินไป | วัตถุที่จะสะท้อนมีลักษณะพื้นผิวเป็นสีดำหรือสีมืด | บริเวณที่มีแสงรบกวนมาก

รู้หรือไม่?

แสงอินฟาเรดเป็นแสงที่ไม่สามารถมองเห็นได้ด้วยตาเปล่า อย่างไรก็ตามคุณสามารถมองเห็นแสงอินฟาเรดได้ง่ายๆด้วยการ ใช้กล้องของโทรศัพท์ถ่ายบริเวณ IR LED

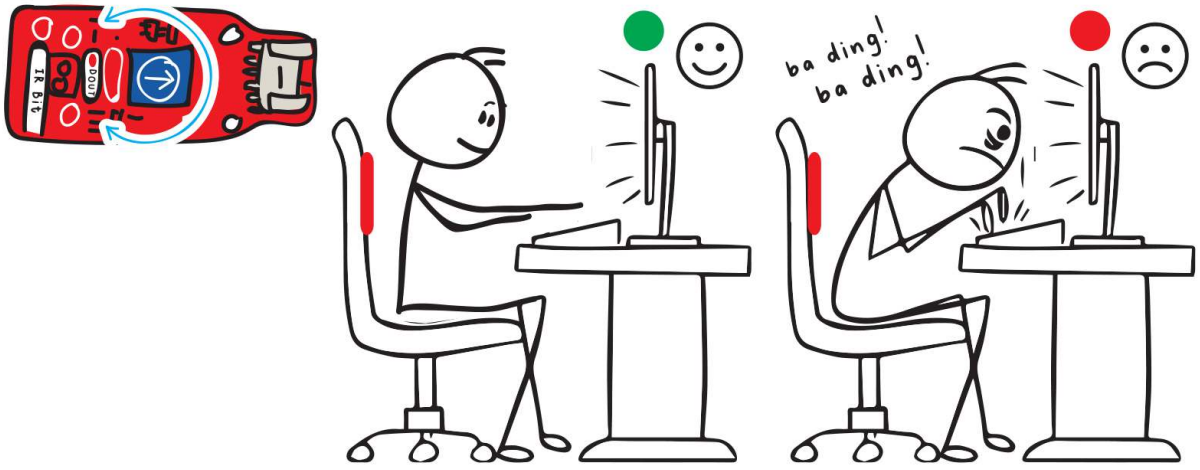


APPLICATION CHALLENGE

โปรแกรม EDU:BIT ให้ทำหน้าที่เป็นอุปกรณ์แจ้งเตือนการนั่งที่ไม่ถูกต้อง	
On start	แสดงคำว่า “Mind your posture”
เมื่อ IR ถูก triggered	แสดงหน้ายิ้มบน LED matrix และเปิด LED สีเขียว
เมื่อ IR ไม่ถูก triggered	เล่นเพลง ‘ba ding’ และแสดงหน้าเศร้าบน LED matrix และกระพริบ LED สีแดง

หลักการทำงาน

ติดตั้ง EDU:BIT ไว้บริเวณพนักพิงเก้าอี้ ดังภาพ นั่งบนเก้าอี้ด้วยท่าทางที่ถูกต้อง ปรับตัวต้านทานปรับค่าได้สีน้ำเงินบน IR Bit จนกว่า LED แสดงสถานะจะติด (IR Bit ตรวจพบหลังของคนนั่งเก้าอี้แล้ว) ขั้นตอนนี้เรียกว่าการ **Calibration**

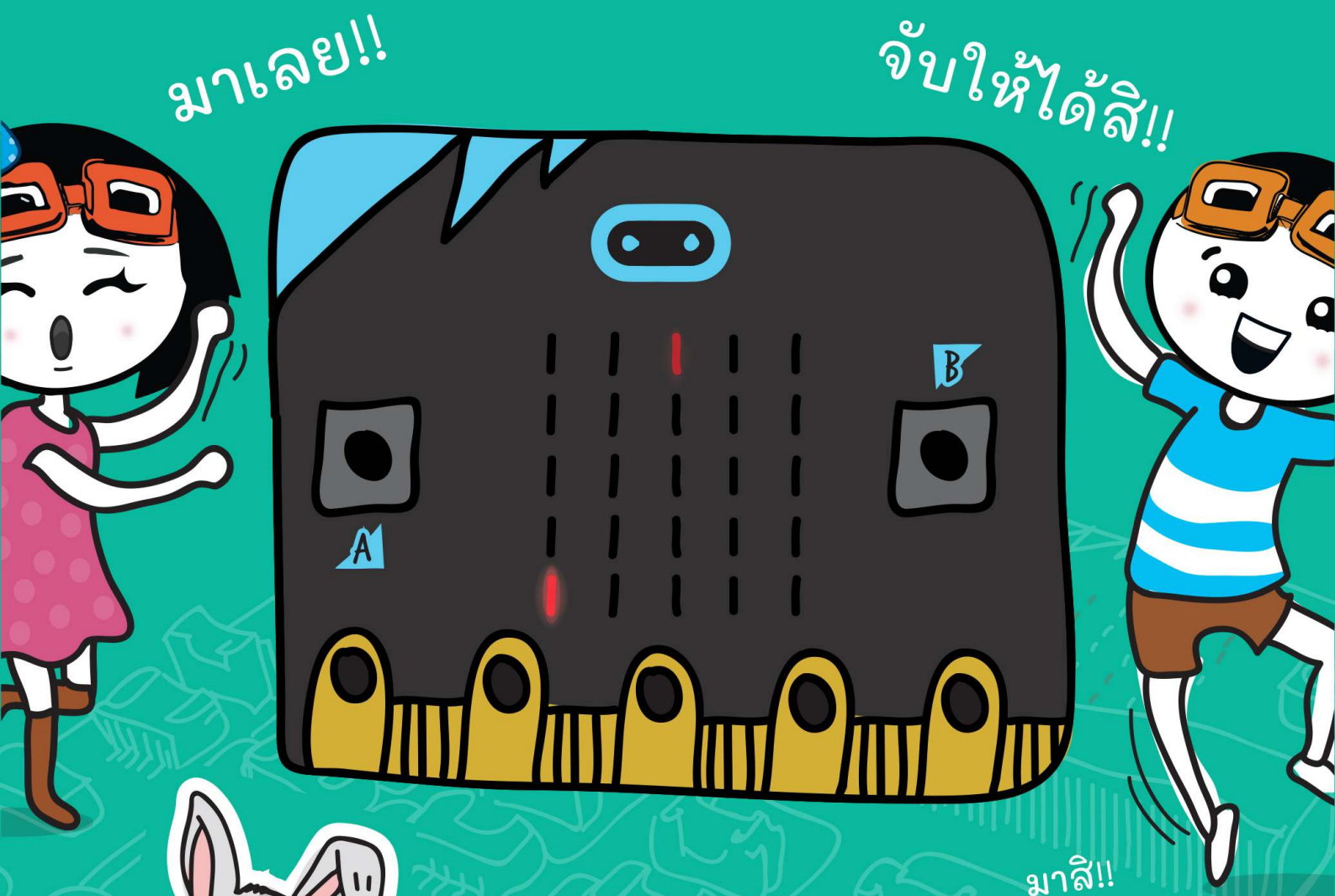


คุณต้องทำการ calibrate ใหม่ทุกครั้งที่คุณใส่เสื้อสีต่างๆกัน รู้ไหมว่าเพราะอะไร?

บทที่ 6

มาไล่จับกันเถอะ

Potentio Bit

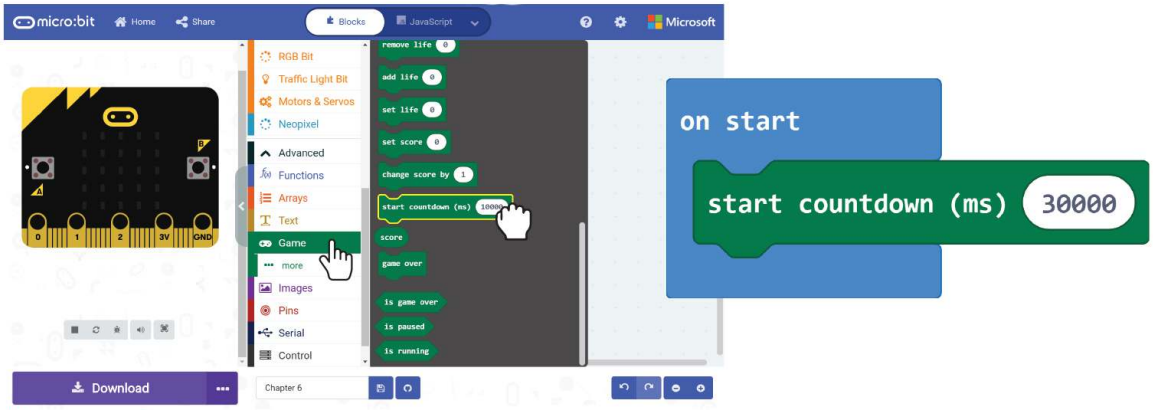


สแกนเลย!

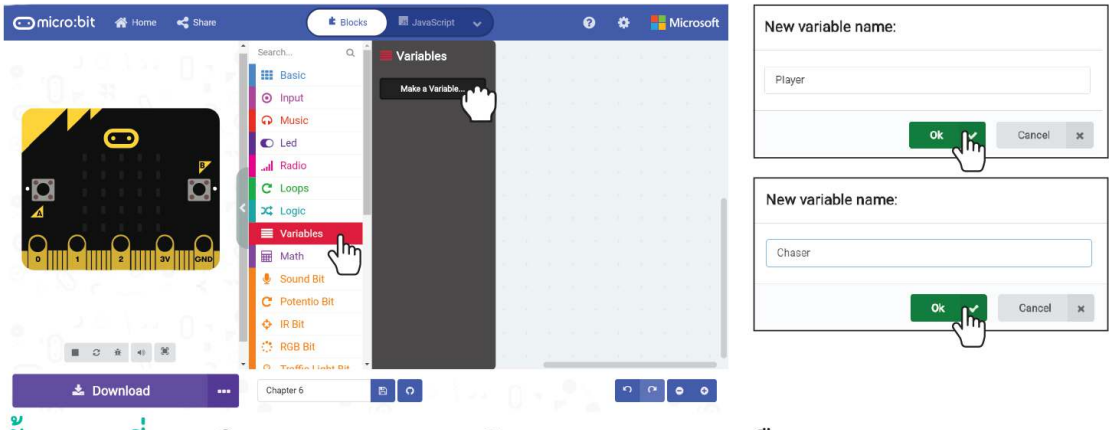
link.cytron.io/edubit-chapter-6

มาเขียนโปรแกรมกัน!

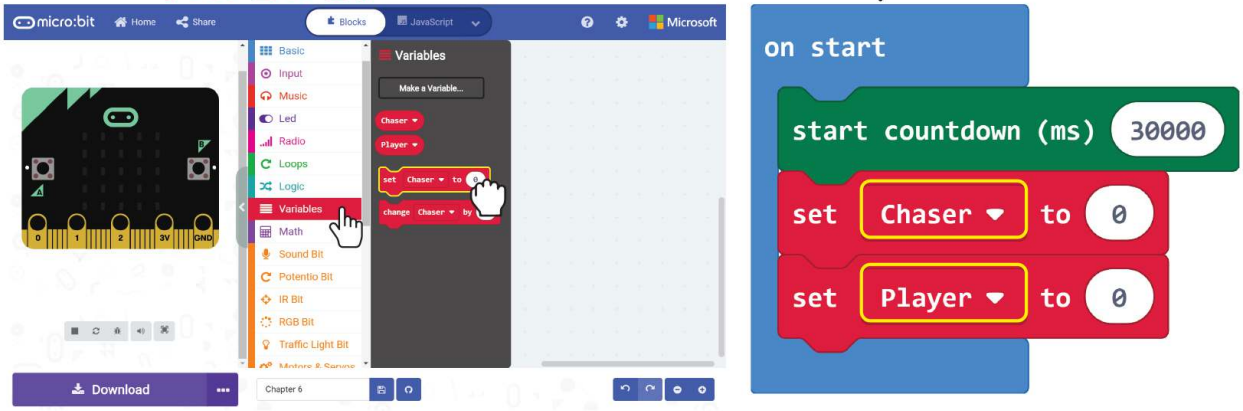
ขั้นตอนที่ 1 สร้างโปรเจ็กต์ใหม่ใน MakeCode Editor ของคุณ และเพิ่มส่วนขยาย EDU:BIT (ทำตอนขั้นตอนในหน้าที่ 40) คลิก [**Advance**] : [**Game**] ใน Toolbox เลือก [**start countdown (ms) _**] block แล้วนำไปวางใน [**on start**] block และเปลี่ยนค่าภายในให้เป็น 30000 ms



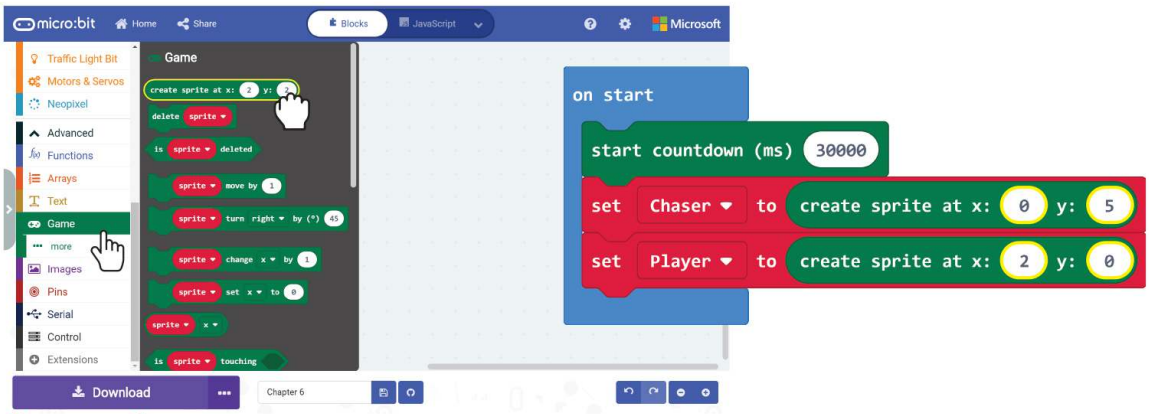
ขั้นตอนที่ 2 คลิก [**Variables**] ใน Toolbox และคลิก [**Make a Variable**] ตั้งชื่อว่า '**Player**' ในหน้าต่างป๊อป-อัพแล้วคลิก OK และสร้างอีกตัวแปรหนึ่งให้มีชื่อว่า '**Chaser**'



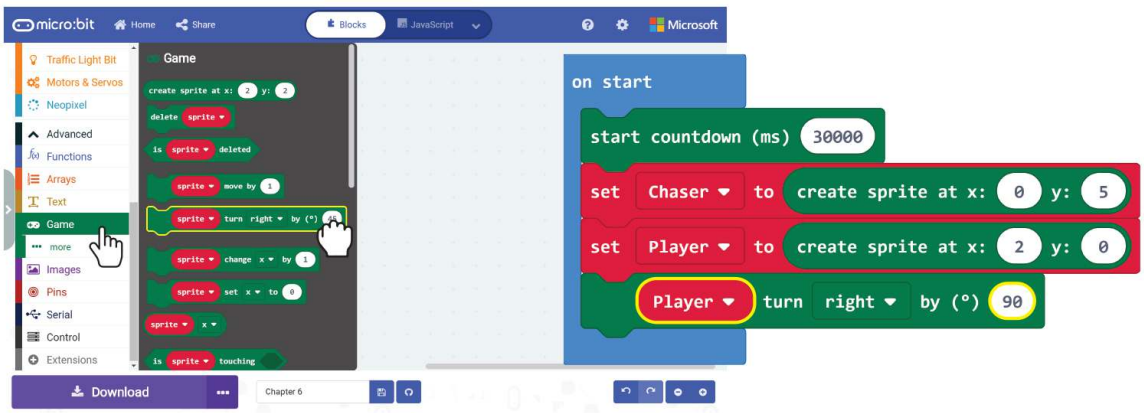
ขั้นตอนที่ 3 คลิก [**Variables**] ใน Toolbox และเลือก [**set _ to _**] block แล้วทำการ Duplicate [**set _ to _**] block แล้วนำไปวางใน [**on start**] block แล้วทำการตั้งค่าตัวแปรใน [**set _ to _**] block ให้เป็น '**Chaser**' และ '**Player**' ตามลำดับ



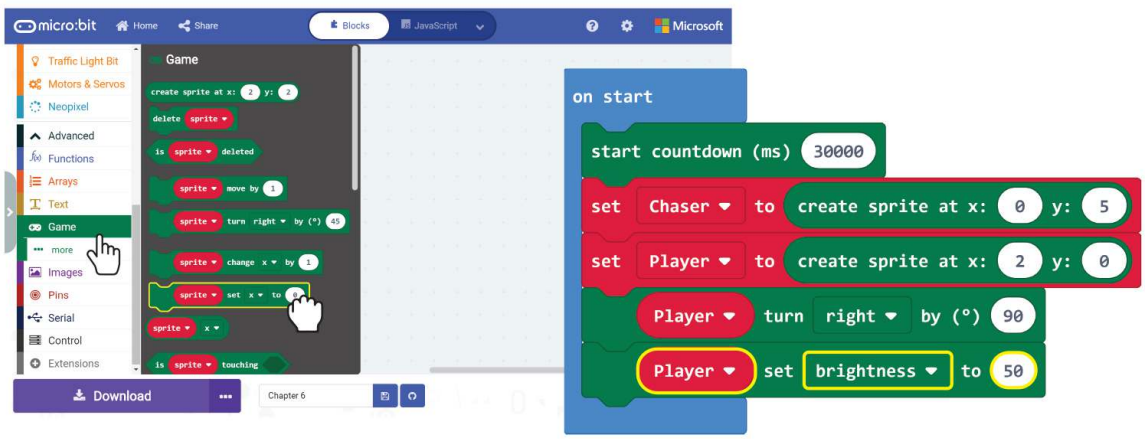
ขั้นตอนที่ 4 คลิก [**Advance**] ใน Toolbox และเลือก [**Game**] เลือก [**create sprite at x: _ y: _**] block แล้วทำการ Duplicate แล้วนำไปวางใน [**set _ to _**] blocks เปลี่ยนค่าภายใน [**create sprite at x: _ y: _**] block ให้เป็น **x: 0** และ **y: 5** สำหรับ **'Chaser'** และ **x: 2** และ **y: 0** สำหรับ **'Player'**



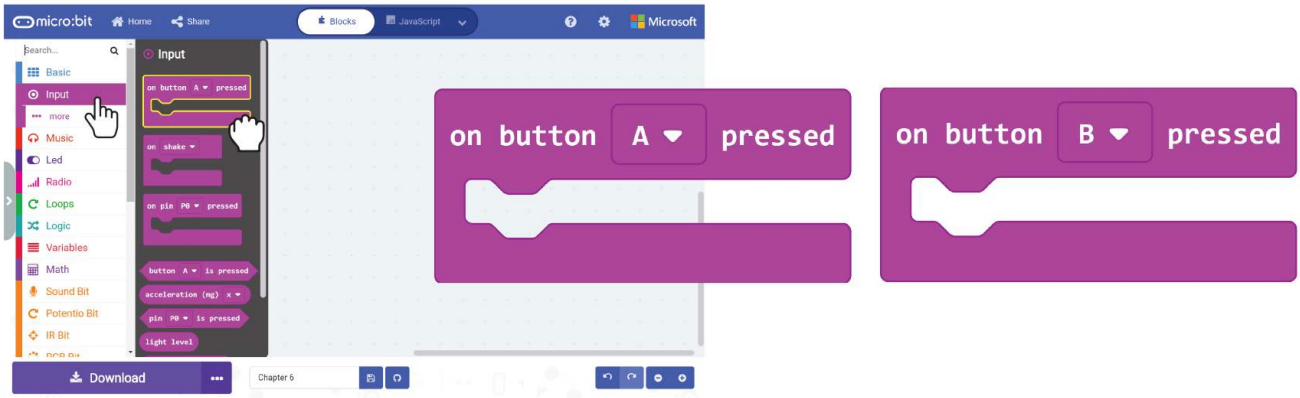
ขั้นตอนที่ 5 คลิก [**Advanced**] : [**Game**] ใน Toolbox และเลือก [**_ turn _ by(°) _**] block แล้วนำไปวางใน [**on start**] block แล้วทำการตั้งค่าตัวแปรเป็น **'Player'** และตั้งค่ามุมให้เป็น **90°**



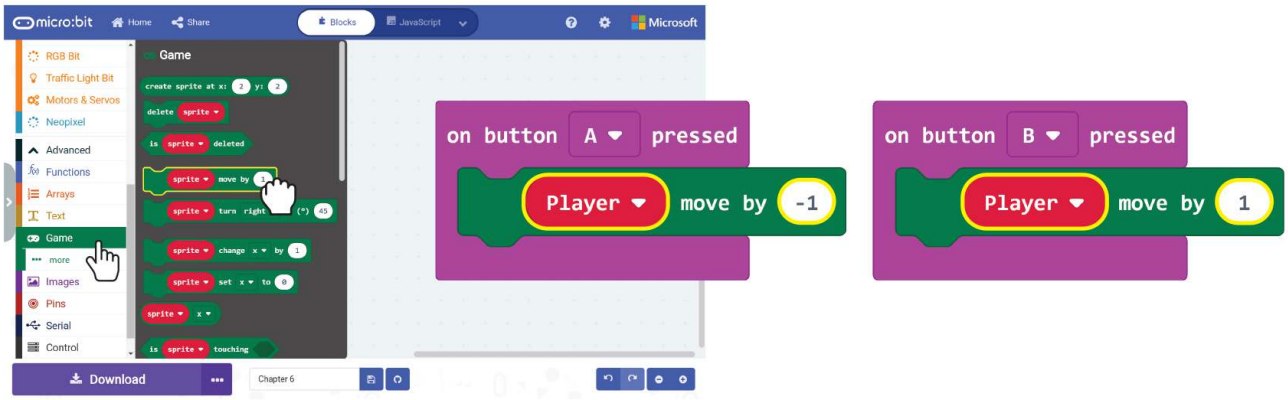
ขั้นตอนที่ 6 คลิก [**Advance**] : [**Game**] ใน Toolbox และเลือก [**_ set _ to _**] block เลือกตัวแปร **'Player'** และตั้งค่าภายใน block โดยเปลี่ยน **'x'** เป็น **'brightness'** และตั้งค่า **'brightness'** ให้เท่ากับ **50**

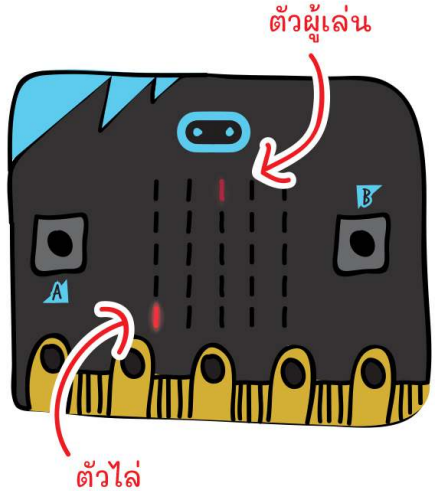


ขั้นตอนที่ 7 คลิก [Input] ใน Toolbox และเลือก [on button _ pressed] block ทำการ Duplicate block และเลือกปุ่ม 'B' บน [on button _ pressed] block ที่ 2



ขั้นตอนที่ 8 คลิก [Game] ใน Toolbox และเลือก [_ move by _] block ทำการ Duplicate แล้วนำไปวางใน [on button _ pressed] slot เลือกตัวแปร 'Player' ให้กับ block ทั้งสองและเปลี่ยนตั้งค่าตัวแปรเป็น 1 (เมื่อปุ่ม A ถูกกด) และ -1 (เมื่อปุ่ม B ถูกกด)

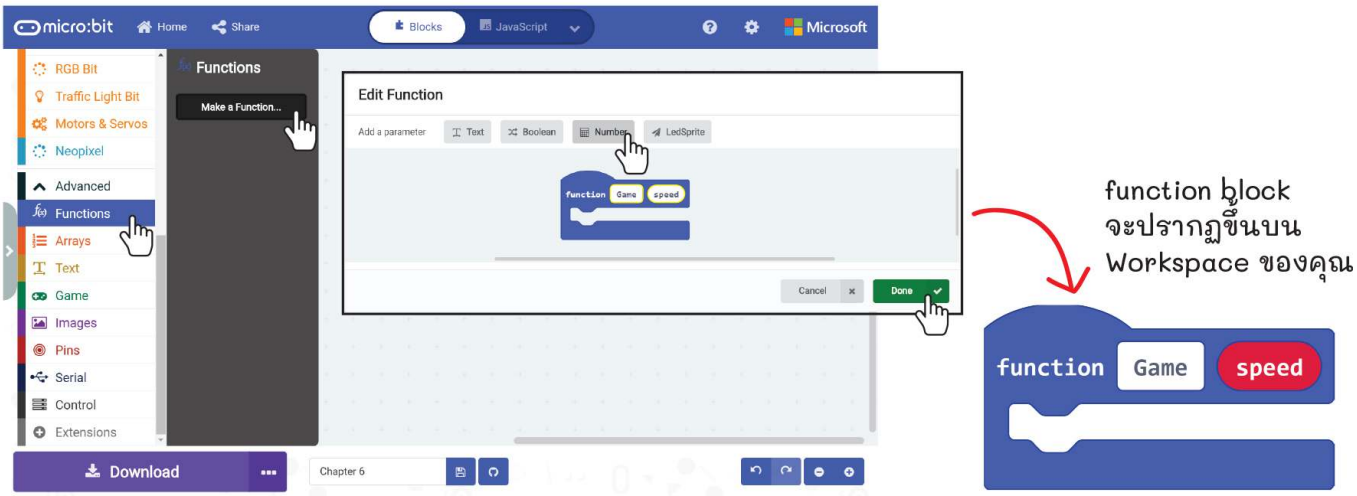




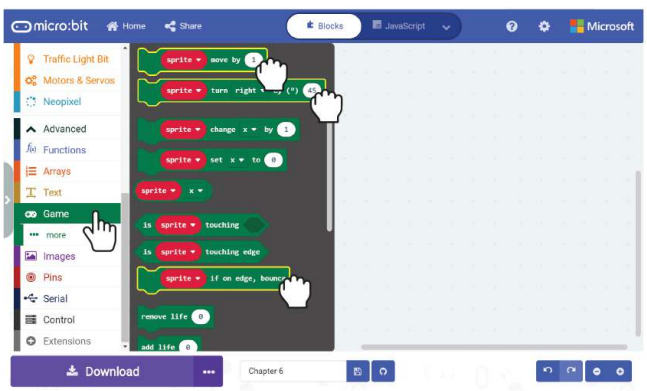
แฟลชโค๊ดลงสู่ EDU:BIT ของคุณเมื่อคุณกดปุ่มสีน้ำเงิน (ปุ่ม B) คุณสังเกตเห็นหรือไม่ว่า LED จะขยับลงมา นั่นคือตัวละครของคุณ

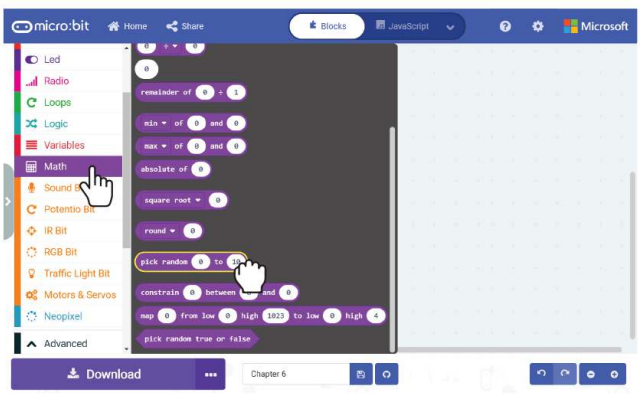
แล้วเกิดอะไรขึ้นตอนคุณกดปุ่มสีเหลืองล่ะ?

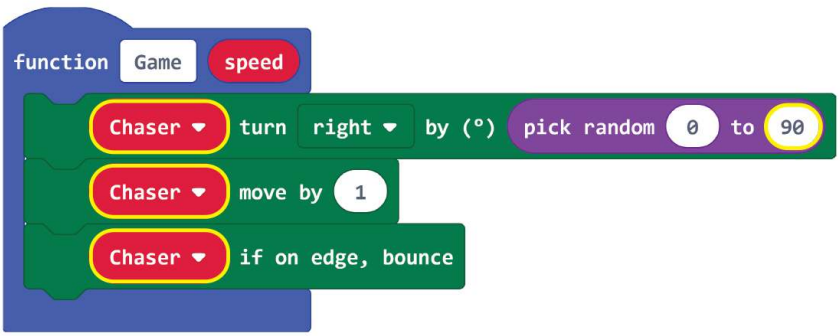
ขั้นตอนที่ 9 คลิก [**Advanced**] ใน Toolbox และเลือก [**Functions**] คลิก [**Make a Function...**] ในหน้าต่าง Edit Function ทำการเปลี่ยนชื่อ 'doSomething' ให้เป็น '**Game**' ต่อมา คลิก [**Number**] เพื่อทำการ add พารามิเตอร์และตั้งชื่อใหม่จาก '**num**' ให้เป็น '**speed**' ใน function block และคลิก 'Done'



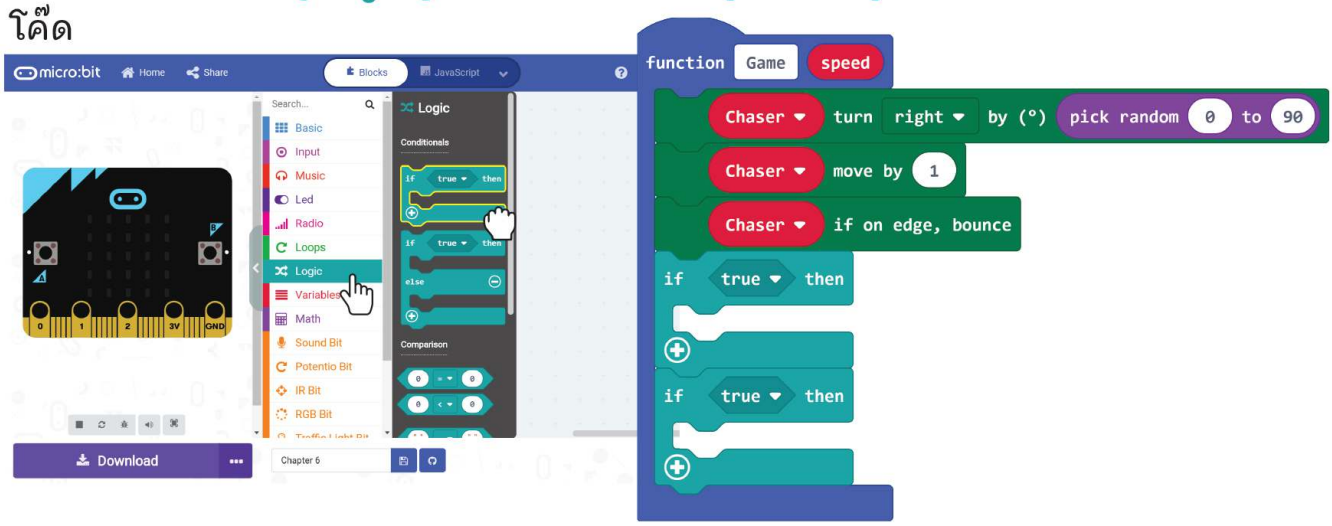
ขั้นตอนที่ 10 สร้าง code โดยการเพิ่ม block จาก [**Advances**] : [**Game**] และ [**Math**] ใน Toolbox ดังภาพด้านล่าง อย่าลืมเปลี่ยนตัวแปรให้เป็น '**Chaser**' และค่าให้เป็น **90**



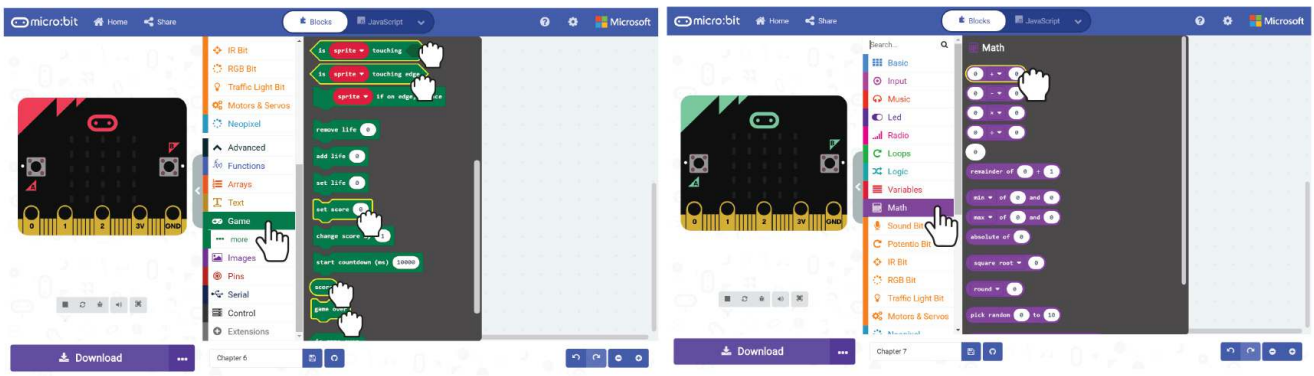


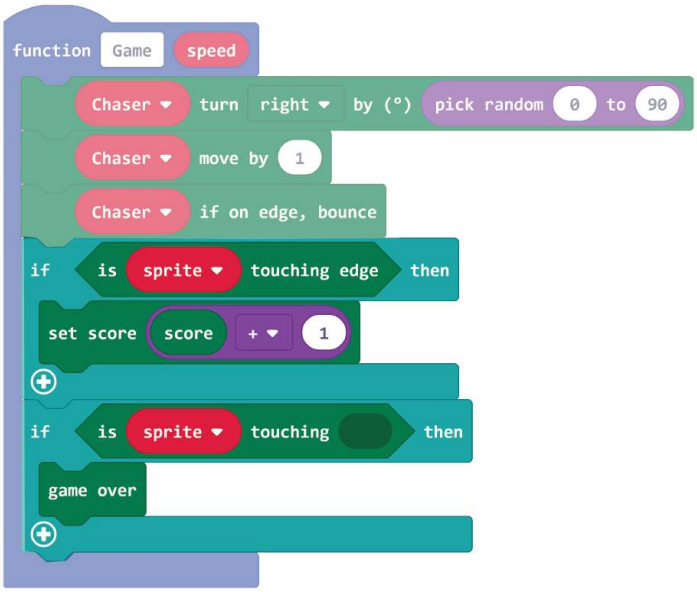


ขั้นตอนที่ 11 เลือก [Logic] ใน Toolbox เลือก [if-then] block จำนวน 2 blocks ลงในโค๊ด

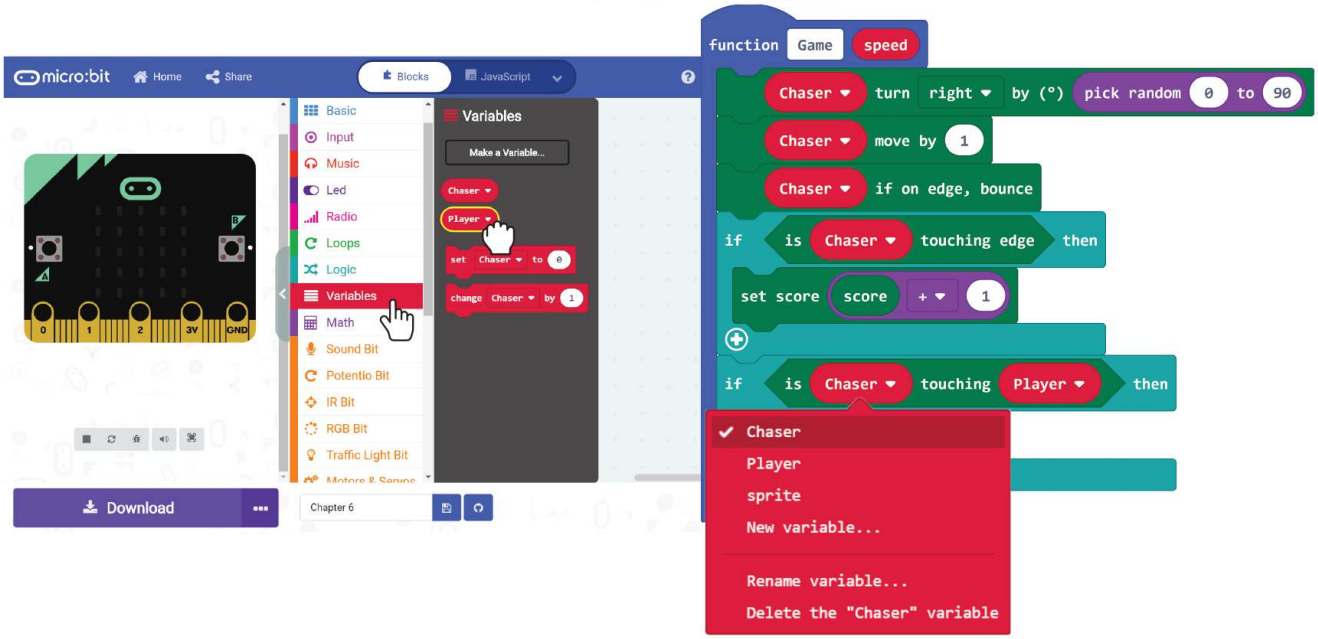


ขั้นตอนที่ 12 เลือก block จาก [**Advance**] : [**Game**] และ [**Math**] ใน Toolbox ดังภาพด้านล่าง

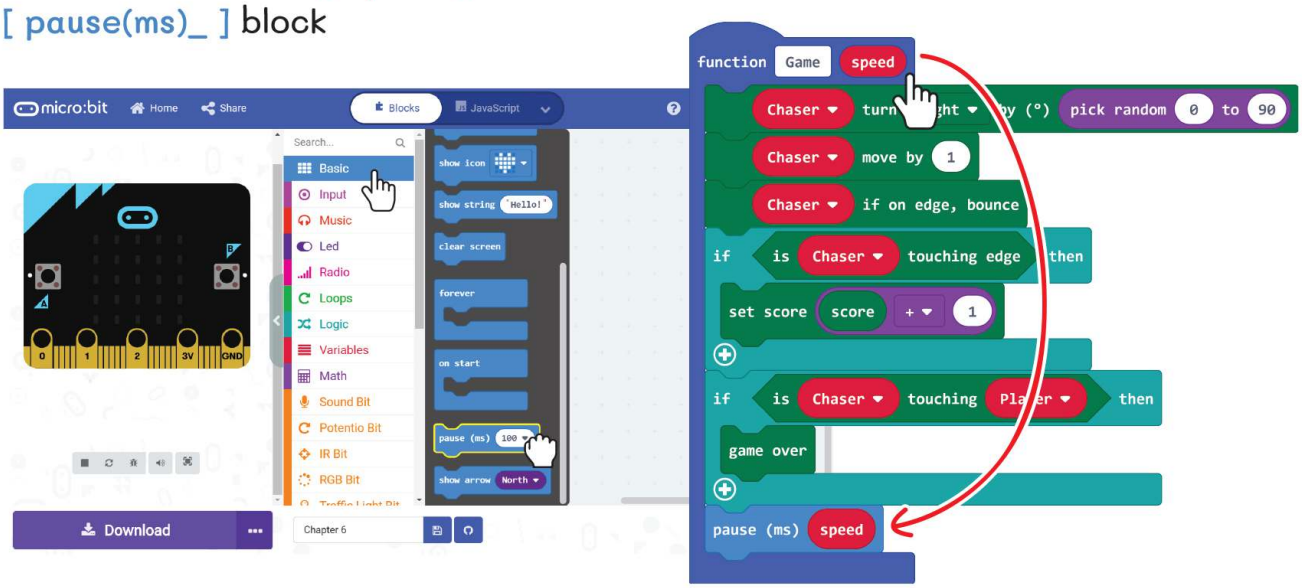




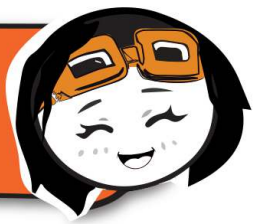
ขั้นตอนที่ 13 เปลี่ยนตัวแปรจาก [Sprite] เป็น 'Chaser' โดยการคลิกที่ตัวแปรใน block แล้วเลือกตัวแปรที่ชื่อ 'Chaser' แทน คลิก [Variables] ใน Toolbox และเลือก [Player] block แล้วนำไปวางใน slot ที่ว่างของ [is_touching_] block



ขั้นตอนที่ 14 คลิก [Basic] ใน Toolbox และเลือก [pause(ms)_] block แล้วนำไปวางในโค๊ด ดังภาพ คลิก [speed] บน function block ทำการลากนำมาวางใน slot ของ [pause(ms)_] block

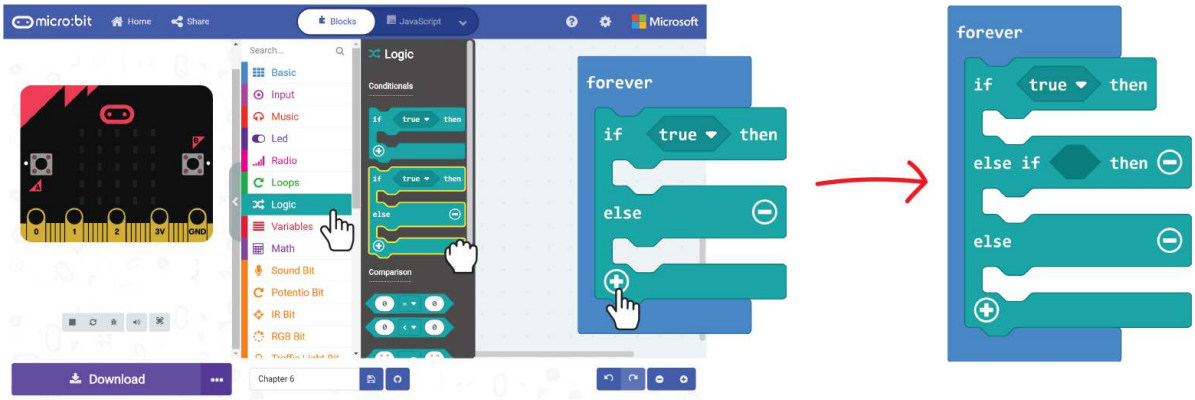


คุณสามารถเพิ่มความสนุกตื่นเต้นของเกมโดยการเล่นเพลงเมื่อ Chaser จับ Player ได้ รู้ไหมว่าต้อง block ไหน แล้วนำไปวางตรงไหนเพื่อทำแบบนั้น?

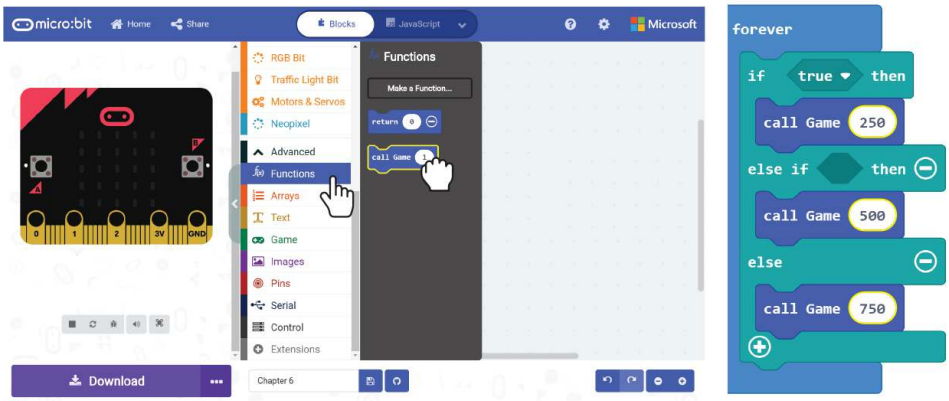


มาเพิ่มระดับความยาก ให้กับเกมนี้กัน!

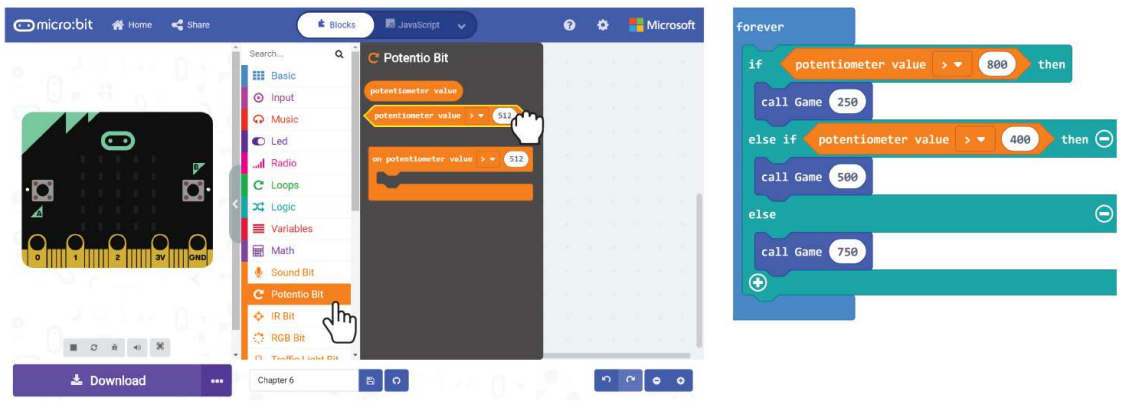
ขั้นตอนที่ 15 คลิก [Logic] ใน Toolbox และเลือก [if-then-else] block แล้วนำไปวาง ใน [forever] slot คลิก (+) เพื่อเพิ่มเงื่อนไขของ [if-then-else] block



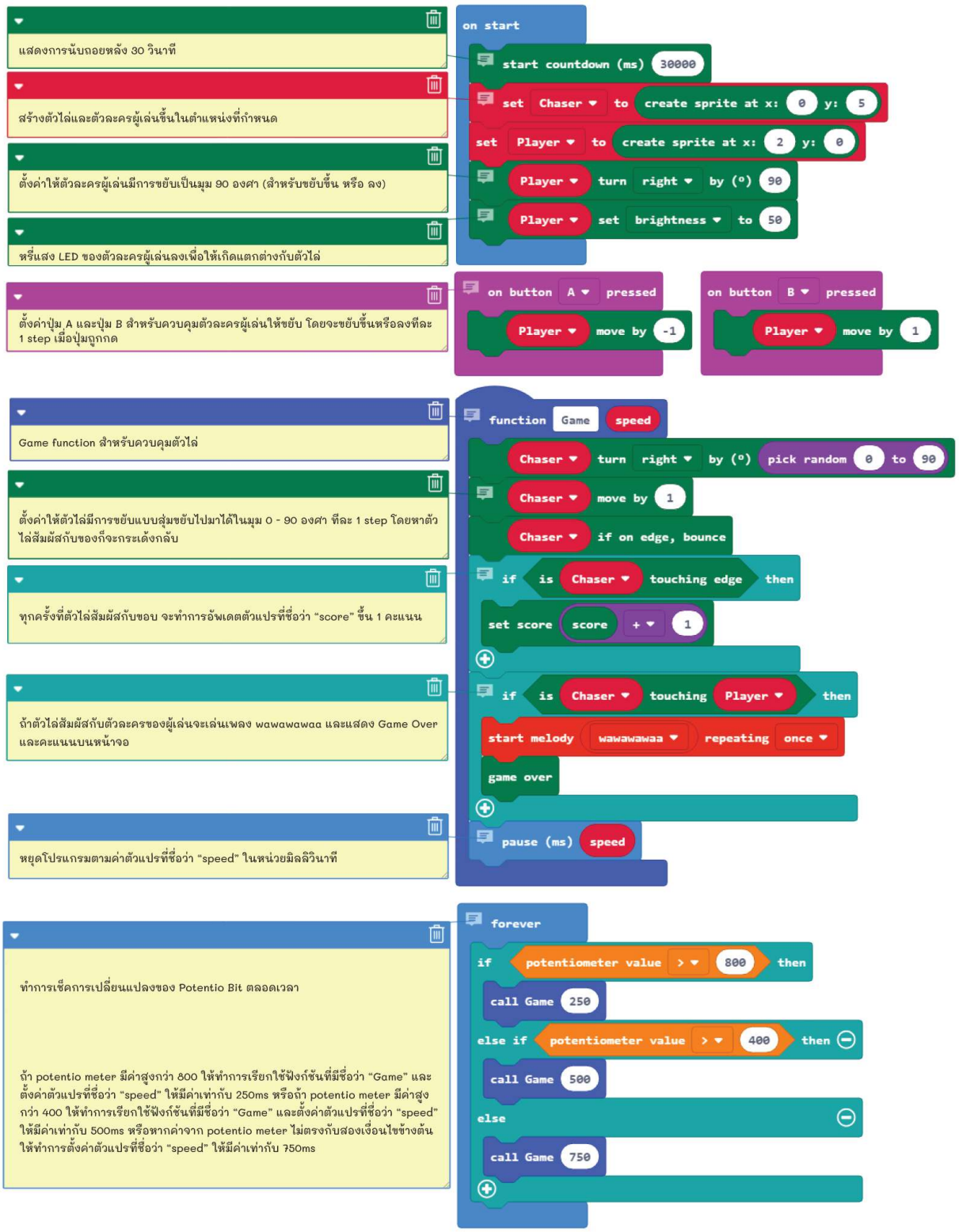
ขั้นตอนที่ 16 คลิก [Functions] ใน Toolbox และเลือก [call Game_] block ทำการ Duplicate แล้วนำไปวางใน [if-then-else] block ทุก slot ทำการเปลี่ยนค่าใน [call Game_] block ให้เป็น 250, 500 และ 750 ตามลำดับ



ขั้นตอนที่ 17 คลิก [Potentio Bit] ใน Toolbox และเลือก [potentiometer value >_] block ทำการ Duplicate แล้วนำไปวางในเงื่อนไขของ [if-then-else] block เปลี่ยนค่าของ [potentiometer value >_] block ให้เป็น 800 สำหรับ block แรก และ 400 สำหรับ block ที่สอง



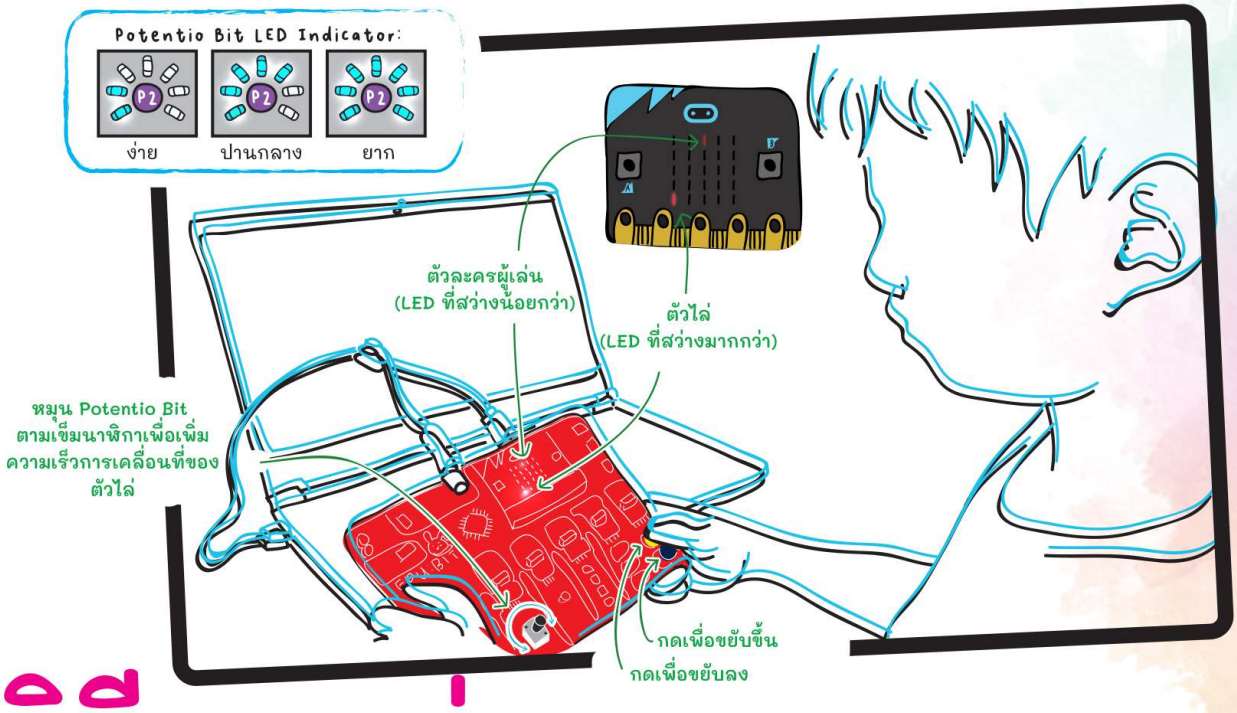
นี่คือโค้ดที่เสร็จสมบูรณ์:



ขั้นตอนที่ 18 อัพโหลดโค้ดที่เสร็จสมบูรณ์แล้วเข้าสู่ EDU:BIT ของคุณและเตรียมตัวสนุกไปกับเกมไล่จับไปกับเพื่อนของคุณนะ!

Let's Play

เกมไล่จับ



วิธีการเล่น

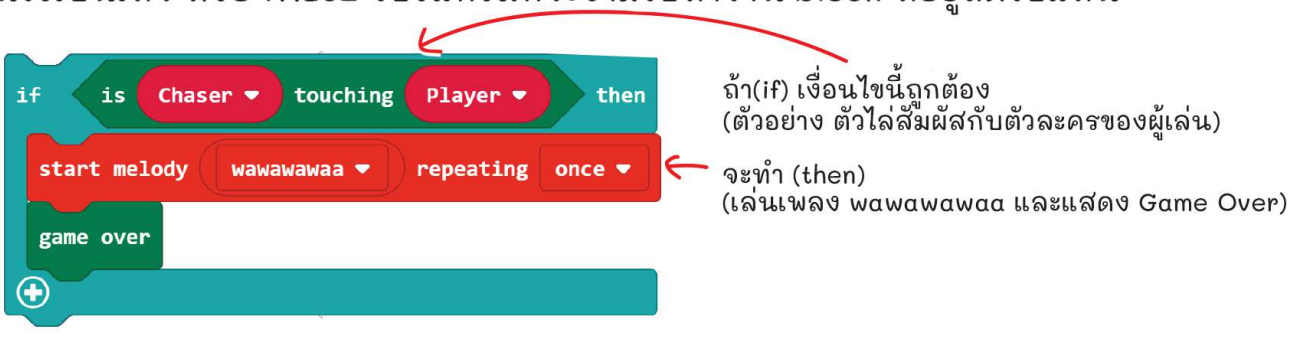
- เมื่อเปิดสวิตซ์ ตัวไล่จะขยับไปมาแบบสุ่มทิศทาง
- ทำการขยับตัวละครของผู้เล่นเพื่อหลบหลีกตัวไล่ โดยการกดปุ่มสีเหลือง (ปุ่ม A) สำหรับขยับขึ้น และปุ่มสีน้ำเงิน (ปุ่ม B) สำหรับขยับลง
- เกมจะจบลงหากตัวละครของผู้เล่นสัมผัสกับตัวไล่ หรือครบเวลา 30 วินาที
- ทุกครั้งที่ตัวไล่สัมผัสกับขอบของจอ คุณจะได้ 1 คะแนน โดยผู้เล่นที่มีคะแนนสูงที่สุดเป็นฝ่ายชนะ

TIPS!

#1 ในการทำคะแนนให้ได้สูงที่สุดใน 30 วินาที สามารถทำได้โดยการเพิ่มความเร็วการขยับของตัวไล่ เพราะจะทำให้ตัวไล่สัมผัสกับขอบของจอได้ถี่กว่าปกติ
#2 หลังจาก Game Over คุณสามารถเริ่มเกมใหม่ได้ด้วยการกดปุ่ม A และ B พร้อมๆกันเพื่อเริ่มเกม นี่เป็น built-in function ของ [Game] block

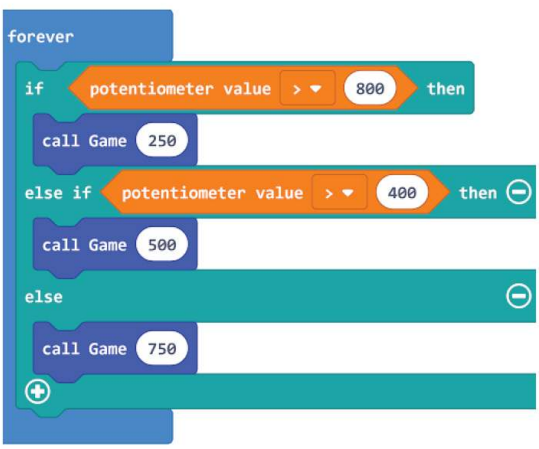
BREAK THE CODE

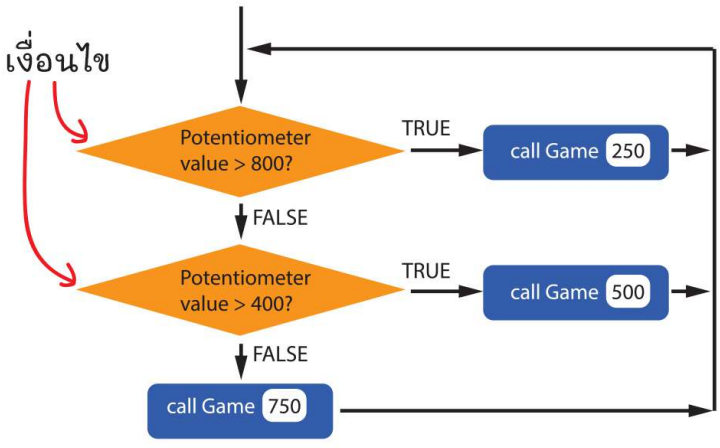
ในการเขียนโปรแกรม เราจะใช้ **เงื่อนไข if statements** ในการตัดสินใจ ใน MakeCode เราจะใช้ [if-then] block หรือ [if-then-else] block จาก [Logic] ใน Toolbox เพื่อสร้างเงื่อนไขการตัดสินใจ โดยโปรแกรมจะทำการเช็คเงื่อนไข โดยหากเงื่อนไขเป็นจริง หรือ TRUE โปรแกรมก็จะทำการทำงานคำสั่งที่อยู่ภายใน block เงื่อนไข แต่หากเงื่อนไขเป็นเท็จ หรือ FALSE โปรแกรมก็จะข้ามไปทำงาน block ที่อยู่ถัดไปแทน



เมื่อเรามีเงื่อนไขหลายเงื่อนไข โปรแกรมก็จะทำการประเมินเงื่อนไขตามลำดับ จากบนลงล่าง และทำงานโค๊ดที่อยู่ในเงื่อนไขที่ถูกต้องเป็นอันดับแรก ด้วยเหตุนี้ เงื่อนไขที่อยู่ด้านบนจะมีความสำคัญมากกว่าเงื่อนไขที่อยู่ด้านล่าง

ตัวอย่างเช่น โค้ดของเกมจะทำการกำหนดความเร็วการเคลื่อนที่ของตัวไล่โดยเทียบกับค่าจาก potentiometer เทียบกับเกณฑ์ที่กำหนดไว้ล่วงหน้า

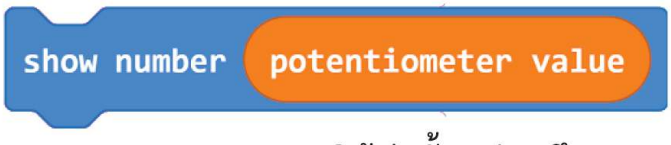




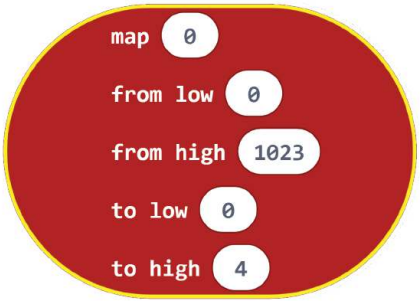
ถ้า (if) ค่าจาก potentiometer > 800 ทำการเรียกใช้ฟังก์ชัน Game (ด้วยตัวแปร “speed” ที่ถูกตั้งค่าไว้ที่ 250 ms)
หรือถ้า (else if) ค่าจาก potentiometer > 400 ทำการเรียกใช้ฟังก์ชัน Game (ด้วยตัวแปร “speed” ที่ถูกตั้งค่าไว้ที่ 500 ms)
หรือหากนอกเหนือจากเงื่อนไขข้างต้น (else) ทำการเรียกใช้ฟังก์ชัน Game (ด้วยตัวแปร “speed” ที่ถูกตั้งค่าไว้ที่ 750 ms)

EXPLORE MORE BLOCKS

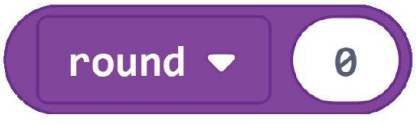
#1 ใช้ [Basic] : [show number] block ร่วมกับ [Potentio Bit] : [potentiometer value] block เพื่ออ่านค่าและแสดงค่าจาก potentiometer



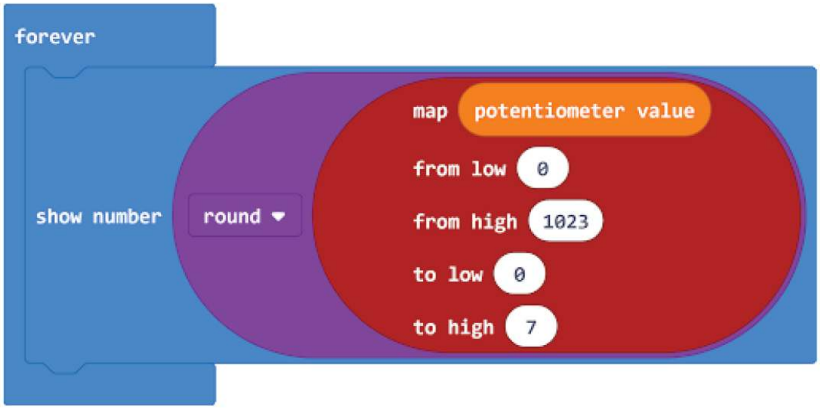
#2 potentiometer จะให้ค่าตั้งแต่ 0 ถึง 1023 คุณสามารถใช้ [map _ from low _ from high _ to low _ to high _] block จาก [Advaced] : [Pins] ใน Toolbox เพื่อ map ค่าจากช่วงที่มีเป็นช่วงที่ต้องการ



#3 mapping block จะให้ค่าออกมาเป็นเลขที่มีทศนิยม (เช่น 1.68, 3.998) โดยการทำให้เลขเป็นจำนวนเต็มจะใช้ [round _] block จาก [Math] ใน Toolbox



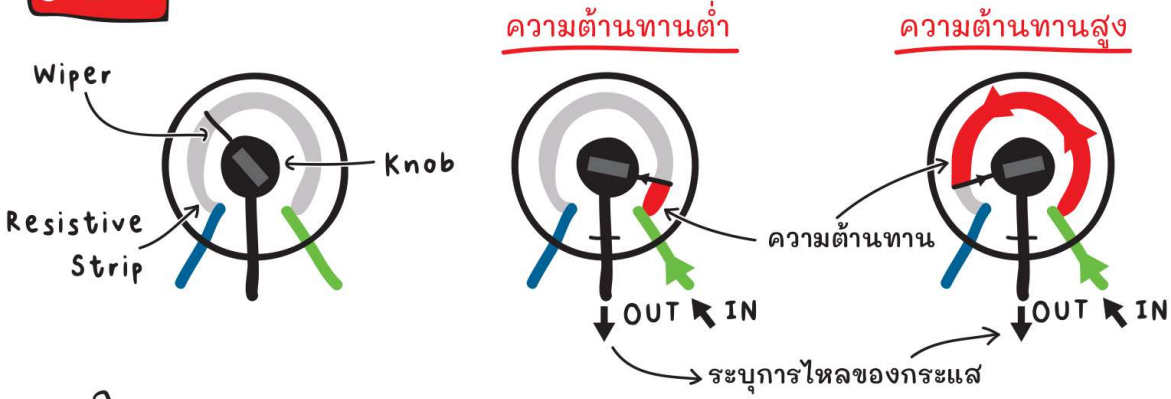
นี่เป็นโค้ดตัวอย่างในการ map ค่าที่ได้จาก Potentio Bit ให้ได้ค่าที่อยู่ในช่วง 0 ถึง 7 โดยค่าที่ได้จะเป็นจำนวนเดต็มและแสดงอยู่บน LED matrix



FUN FACT!

Potentiometers หรือ pots หมายถึงตัวต้านทานที่สามารถปรับค่าความต้านทานได้ง่ายโดยใช้ knob หรือ slider

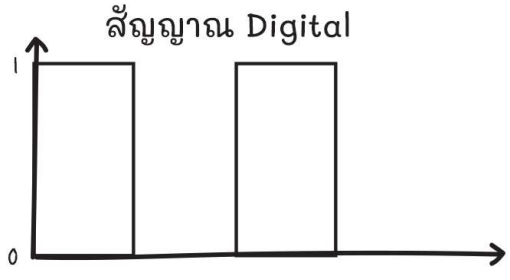
ถ้าคุณมี potentiometer ที่มีความต้านทาน 10,000Ω คุณสามารถปรับค่าความต้านทานได้ระหว่าง 0Ω ถึง 10,000Ω โดยการหมุน wiper ตามตำแหน่งดังภาพ



การใช้งานทั่วไปในชีวิตประจำวัน :

- การหมุนเพื่อปรับเปลี่ยนความดังของลำโพง
- การหมุนเพื่อปรับเปลี่ยนความถี่ของวิทยุ
- การหมุนเพื่อปรับเปลี่ยนอุณหภูมิของฮีตเตอร์

potentiometer บน EDU:BIT เป็น input ประเภท analog โดยหลักการเป็นการวัดศักย์ไฟฟ้าและแปลงแรงดันไฟฟ้าที่วัดได้ (ระหว่าง 0V ถึง 3.3V) ให้เป็นเลขจำนวนเต็มระหว่าง 0 ถึง 1023



APPLICATION CHALLENGE

โปรแกรม EDU:BIT ให้เป็นนาฬิกาจับเวลา โดยใช้ Potentio Bit ในการปรับระยะเวลา (ระหว่าง 0 ถึง 60 วินาที) โดยกดปุ่ม A เพื่อเริ่มนับถอยหลัง และกดปุ่ม B เพื่อ reset	
On Start	ตั้งค่าการทำงานเป็น Mode 0
เมื่อปุ่ม A ถูกกด (ปุ่มสีเหลือง)	ตั้งค่าการทำงานเป็น Mode 1 ตั้ง Start Time ให้เป็น running time แสดงหน้ายิ้มบนจอ
เมื่อปุ่ม B ถูกกด (ปุ่มน้ำเงิน)	ตั้งค่าการทำงานเป็น Mode 0
Forever	เช็ค Mode การทำงานตลอดเวลา • ถ้า Mode = 0 ตั้งค่าระยะเวลานับถอยหลังจากค่าจำนวนเต็มที่ถูก map ให้อยู่ในช่วง 0 ถึง 60 แล้ว และแสดงระยะเวลาที่จะนับถอยหลังบนหน้าจอ • หรือถ้าเป็น Mode = 1 ทำการเช็คว่า (running time - Start Time) > (Duration x 10000) เป็นจริงหรือไม่ โดยหากเป็นจริง (TRUE) ให้ทำการเล่นเพลง wawawawaa และตั้งค่าการทำงานเป็น Mode 2 • หรือหากไม่ตรงกับเงื่อนไขใดๆข้างต้นให้แสดงหน้าเศร้า

เคล็ดลับสำหรับคุณ

ข้อที่ 1 คุณจำเป็นที่จะต้องสร้าง 3 ตัวแปร ได้แก่ Mode, Start Time และ Duration

ข้อที่ 2 Running time (ms) block หาได้จาก [Input] ใน Toolbox

ข้อที่ 3 ใช้เงื่อนไขดังภาพด้านล่างในการเช็คว่าหมดเวลารึยัง

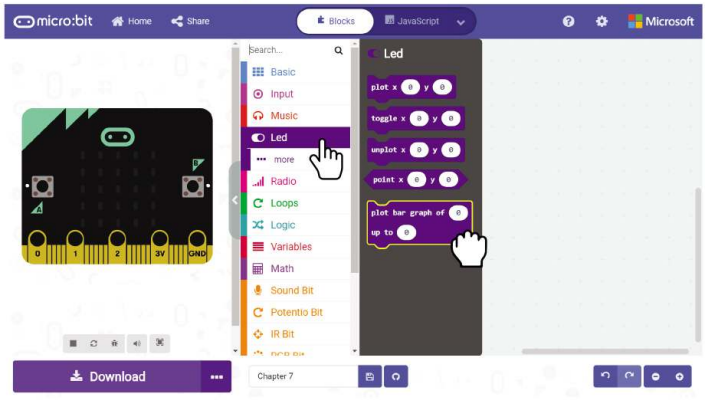
บทที่ 7

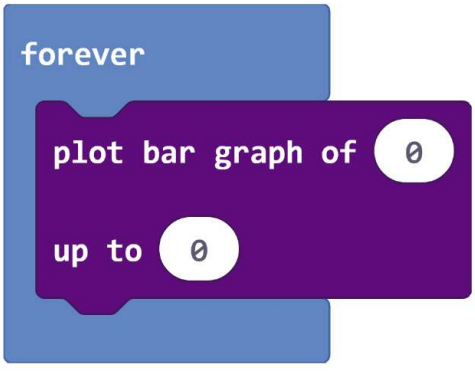
เสียงปรบมือของใครดังกว่ากันนะ?

(Sound Bit)

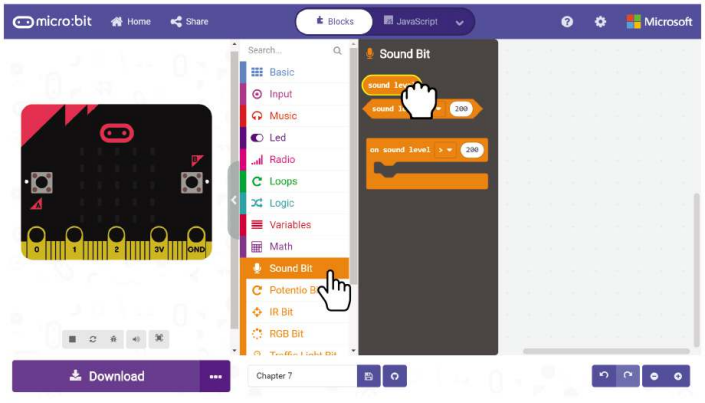
วิธีการเขียน

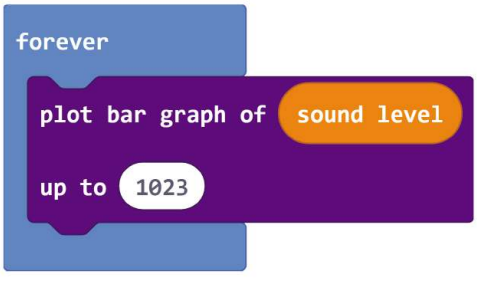
ขั้นตอนที่ 1 ในหน้า Makecode editor กดเริ่มโปรเจ็คใหม่และเพิ่มส่วนเสริม EDU:BIT (สามารถย้อนกลับไปดูได้ที่หน้า 40) คลิกที่หมวดหมู่ [Led] และลากบล็อกคำสั่ง [plot bar graph of _ up to _] มาวางไว้ในบล็อก [forever]



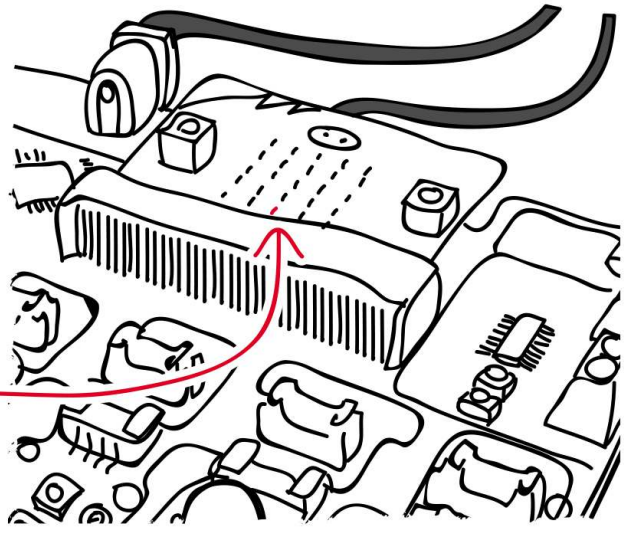


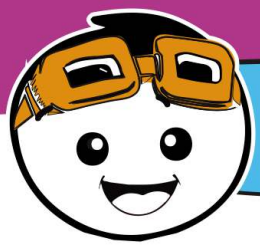
ขั้นตอนที่ 2 คลิกที่หมวดหมู่ [Sound bit] และเลือกบล็อกคำสั่ง [sound level] ไปวางในบล็อก [plot bar graph of _ up to _] และเปลี่ยนค่าอันที่สองจากเลข 0 เป็น 1023





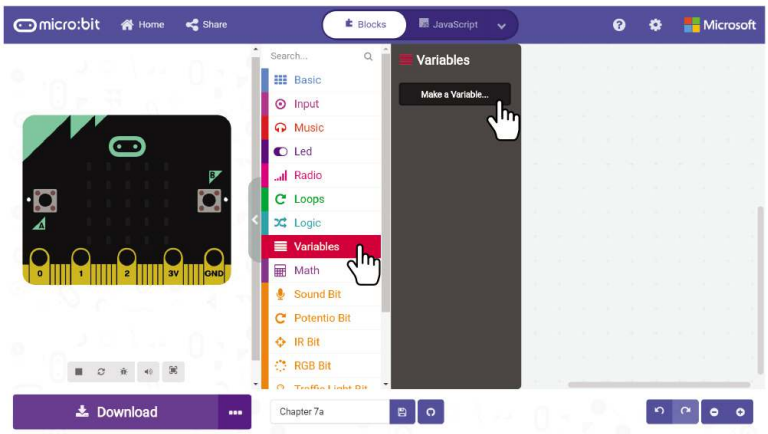
ขั้นตอนที่ 3 แฟลชโค้ดลงบอร์ด EDU:BIT น้องๆจะสังเกตเห็นจอ LED Matrix แสดงผลเมื่อน้องๆปรบมือหรือเคาะโต๊ะ

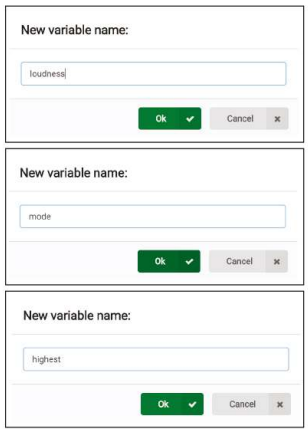




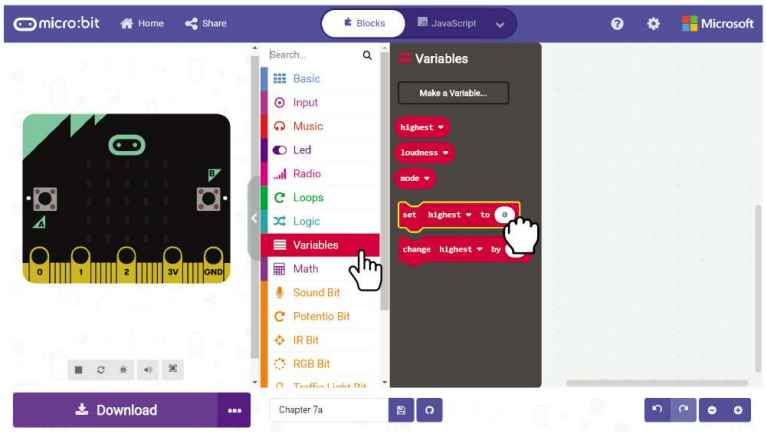
ลองทำให้EDU:BITวัดเสียงปรบมือซิ

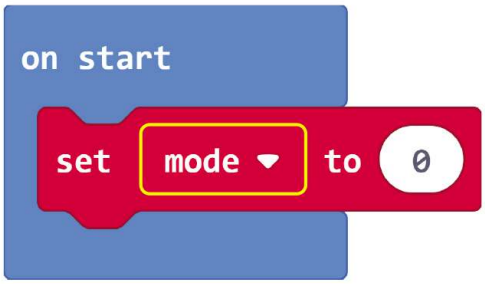
ขั้นตอนที่ 4 เริ่มโปรเจ็คใหม่ และเพิ่มส่วนเสริม EDU:BIT เข้าไป คลิกที่หมวดหมู่ [Variables] จากนั้นเลือกคำสั่ง [Make a Variable] พิมพ์คำว่า “mode” ลงไปในหน้าต่างที่แสดงขึ้นมาจากนั้นคลิก OK และสร้างตัวแปรอีก 2 ค่า ชื่อว่า “loudness” และ “highest” โดยใช้แบบก่อนหน้า



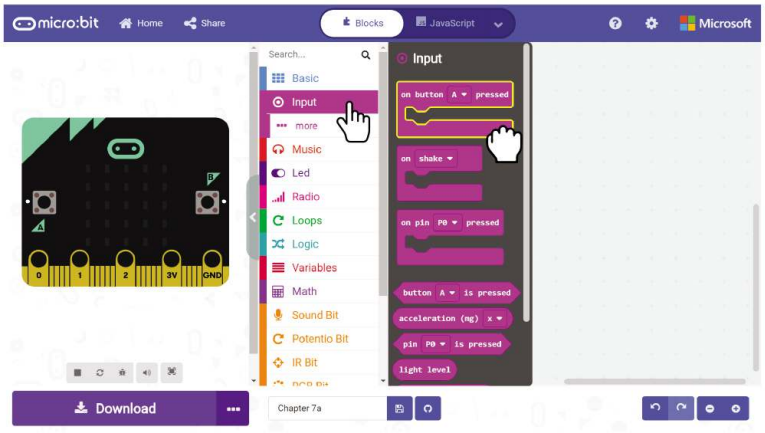


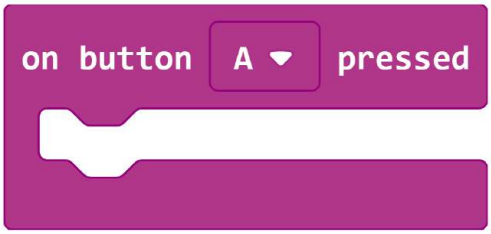
ขั้นตอนที่ 5 เลือกบล็อกคำสั่ง [set _ to _] จากหมวดหมู่ [Variables] แล้วลากมาวางไว้ที่บล็อก [on start] จากนั้นเปลี่ยนค่าตัวแปรเป็น “mode”





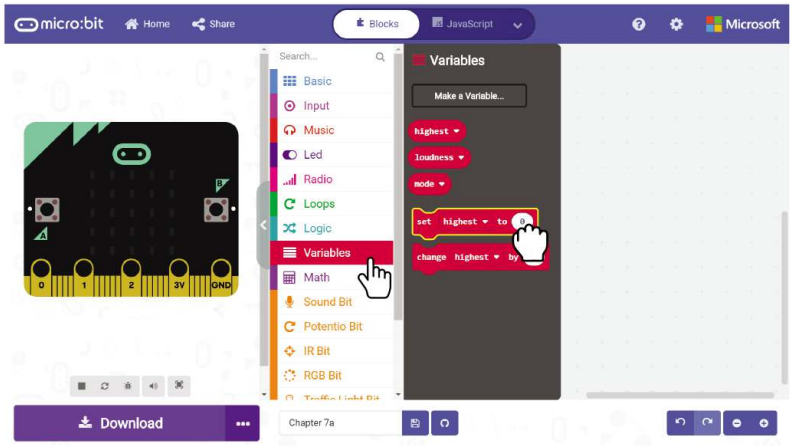
ขั้นตอนที่ 6 เลือกหมวดหมู่ [Input] และเลือกบล็อกคำสั่ง [on button _pressed] ออกมาวาง

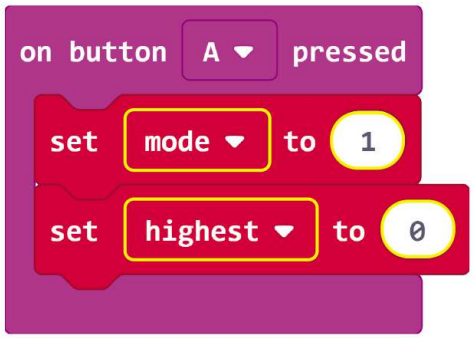




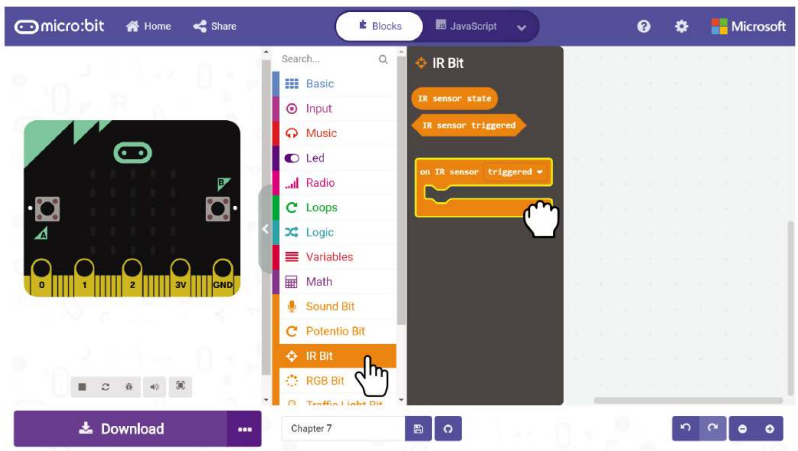
บทที่ 7 : เสียงปรบมือของใครดังกว่ากันนะ?

ขั้นตอนที่ 7 เลือกบล็อกคำสั่ง [set _ to _] จากหมวดหมู่ [Variables] และวางบล็อกคำสั่งไว้ใน [on button A pressed] เปลี่ยนค่าในกล่องแรกเป็น “mode” และเปลี่ยนค่าในกล่องที่สองเป็น 1 ส่วนอีกบล็อกเปลี่ยนค่าในกล่องแรกเป็น “highest” และเปลี่ยนค่าในกล่องที่สองเป็น 0

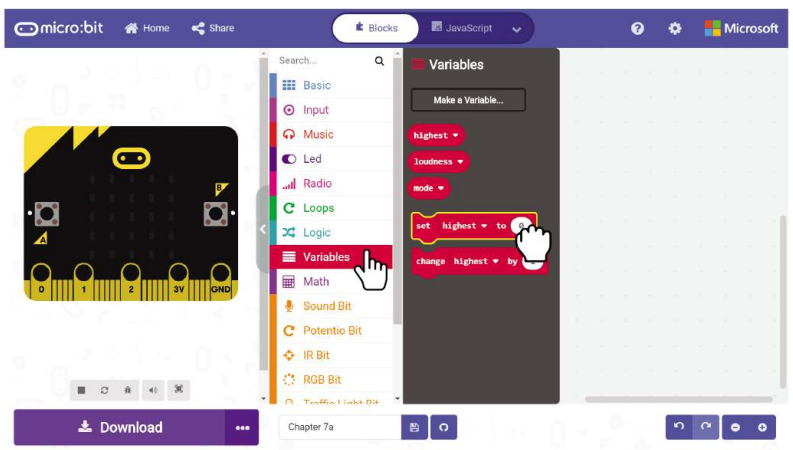


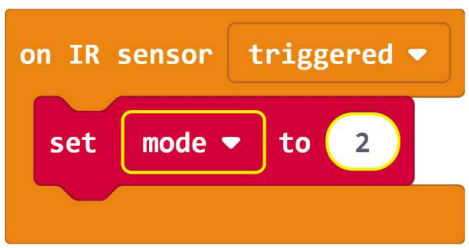


ขั้นตอนที่ 8 เลือกหมวดหมู่ [IR bit] และเลือกบล็อกคำสั่ง [on IR sensor triggered] ออกมาวาง



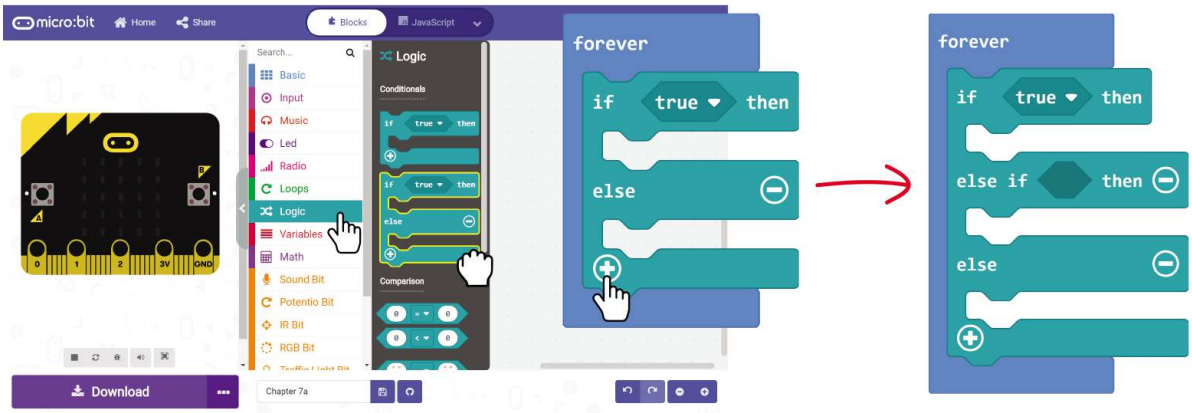
ขั้นตอนที่ 9 เลือกหมวดหมู่ [Variables] และเลือกบล็อกคำสั่ง [set _ to _] วางเอาไว้ที่บล็อก [on IR sensor triggered] เปลี่ยนค่าในกล่องแรกเป็น “mode” และเปลี่ยนค่าในกล่องที่สองเป็น 2



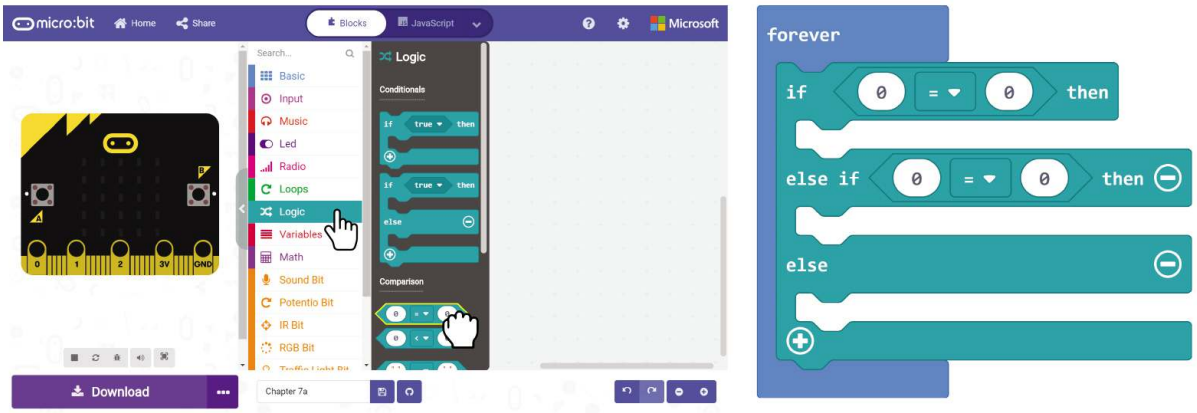


บทที่ 7 : เสียงปรบมือของใครดังกว่ากันนะ?

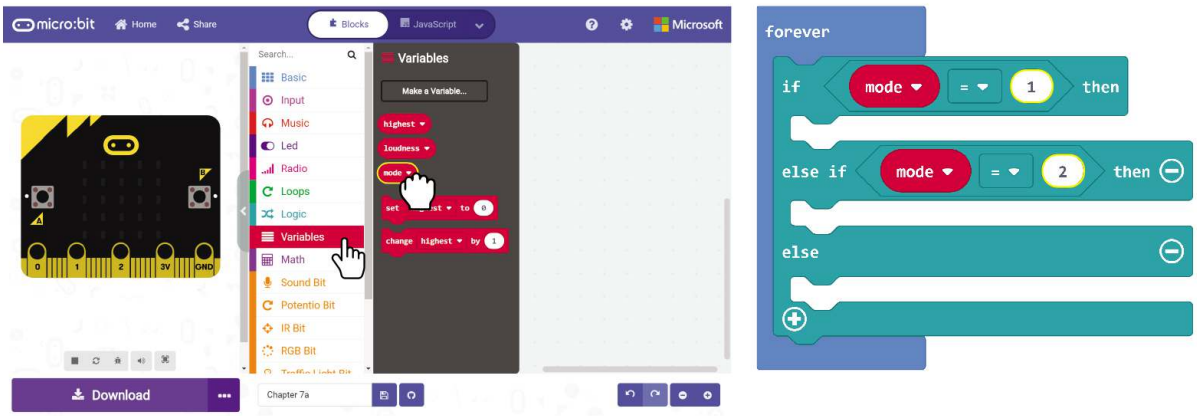
ขั้นตอนที่ 10 เลือกหมวดหมู่ [Logic] จากนั้นเลือกบล็อกคำสั่ง [if-then-else] มาวางไว้ที่บล็อก [forever] จากนั้นคลิกที่เครื่องหมายบวก เพื่อเพิ่มเงื่อนไข else-if ลงไปในบล็อกคำสั่ง



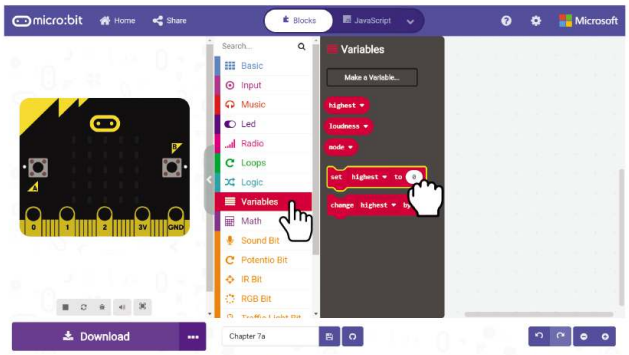
ขั้นตอนที่ 11 เลือกหมวดหมู่ [Logic] และเลือกบล็อกเปรียบเทียบ [_=_] จากนั้นคัดลอกบล็อกเปรียบเทียบ [_=_] ออกมาและวางบล็อกเปรียบเทียบ [_=_] เอาไว้ในช่องของ [if-then-else]

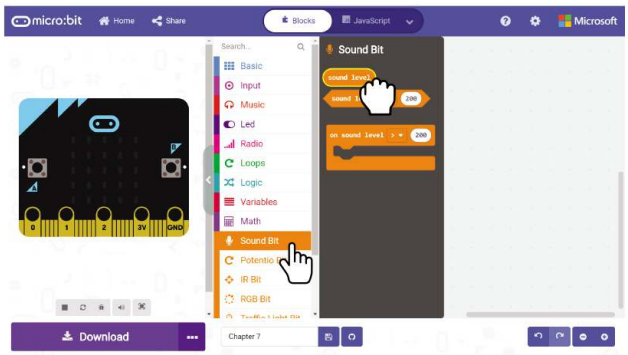


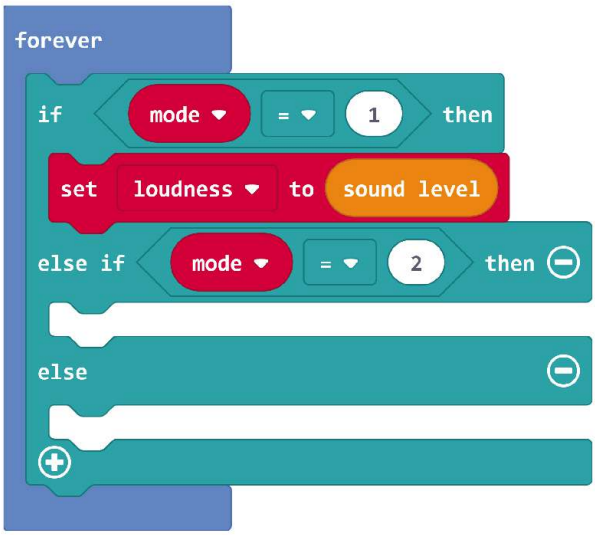
ขั้นตอนที่ 12 เลือกบล็อก [mode] จากหมวดหมู่ [Variable] และวางเอาไว้ที่กล่องข้อมูลด้านซ้ายของบล็อกเปรียบเทียบ [_=_] จากนั้นเปลี่ยนค่าในกล่องด้านขวาเป็น 1 และ 2 ตามลำดับ



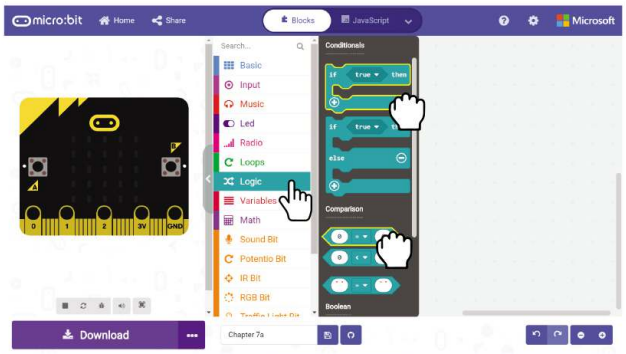
ขั้นตอนที่ 13 เลือกบล็อกคำสั่ง [set _ to _] จากหมวดหมู่ [Variables] แล้วลากไปวางในช่องแรกของบล็อกคำสั่ง [if-then-else] ตั้งค่าตัวแปรเป็น 'loudness' และลากบล็อก [sound level] จากหมวดหมู่ [Sound Bit] ลงในกล่องเก็บค่า

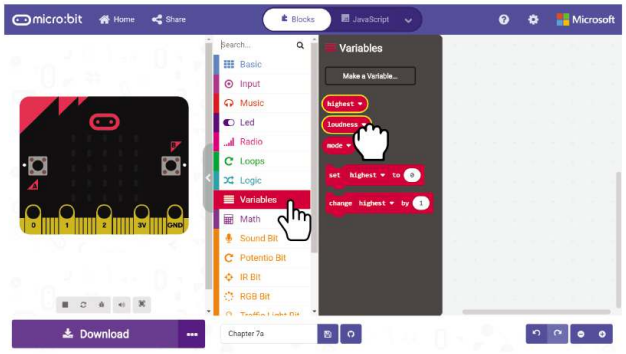


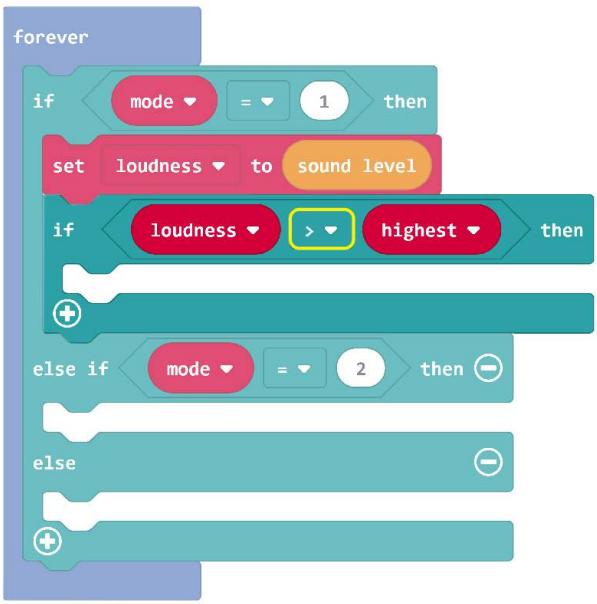




ขั้นตอนที่ 14 เลือกบล็อกคำสั่ง [if-then] และ บล็อกเปรียบเทียบ [_=_] จากหมวดหมู่ [Logic] จากนั้นเปลี่ยนสัญลักษณ์จาก = เป็น > แล้วนำบล็อก [loudness] และ [highest] จากหมวดหมู่ [Variables] วางลงในช่องของบล็อกเปรียบเทียบ

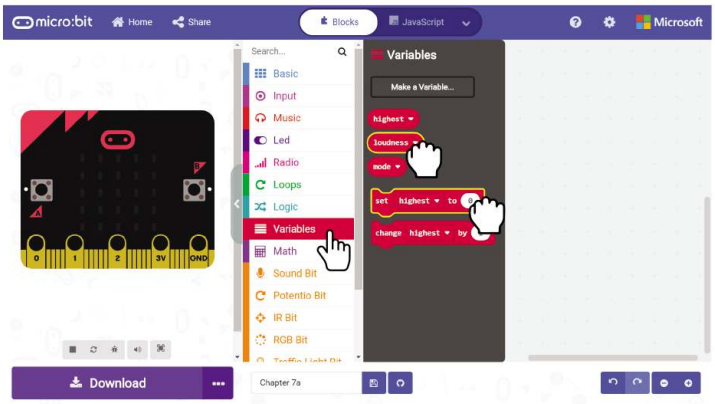


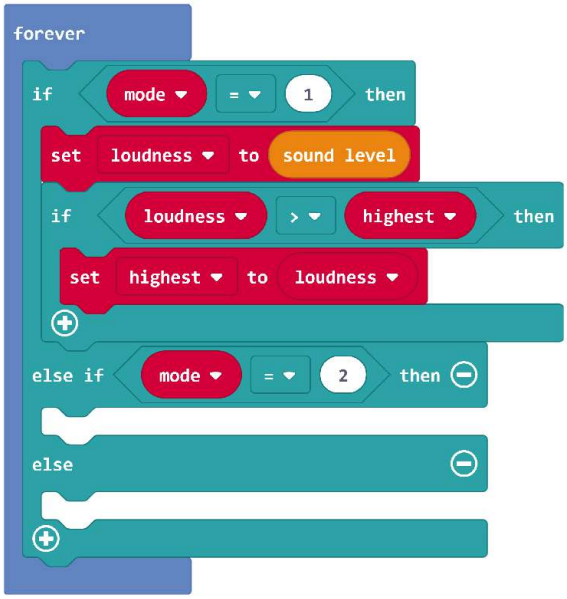




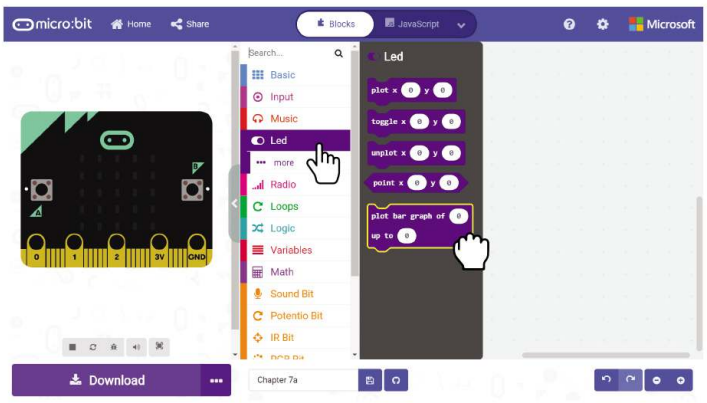
บทที่ 7 : เสียงปรบมือของใครดังกว่ากันนะ?

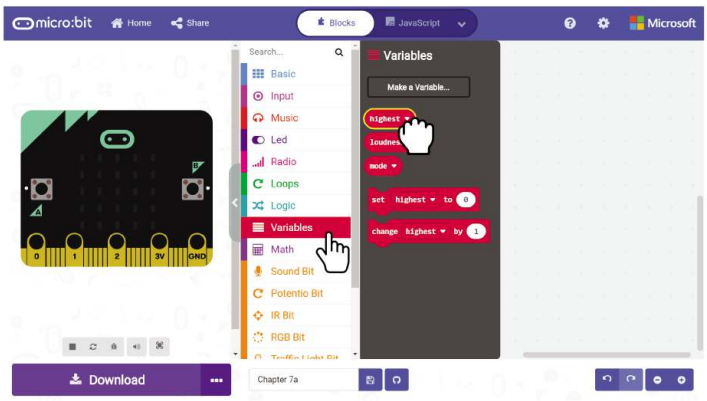
ขั้นตอนที่ 15 เลือกบล็อกคำสั่ง [set _ to _] จากหมวดหมู่ [Variables] จากนั้นลากไปวางในช่องของบล็อก [if-then] และนำบล็อก [loudness] จากหมวดหมู่ [Variables] มาวางลงในกล่องเก็บค่า

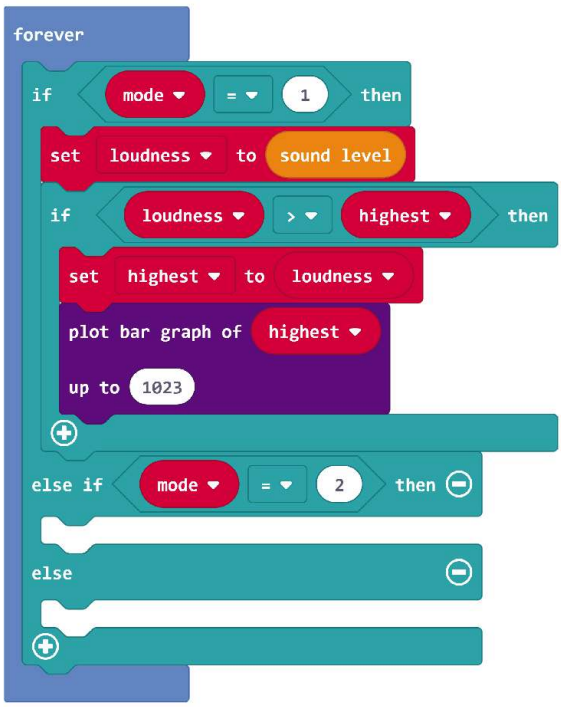


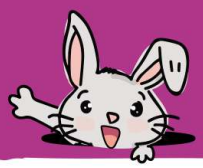


ขั้นตอนที่ 16 เลือกบล็อกคำสั่ง [plot bar graph of _ up to _] จากหมวดหมู่ [Led] และเลือกบล็อก [loudness] จากหมวดหมู่ [Variable] แล้วนำไปวางในบล็อก [plot bar graph of _ up to_] และเปลี่ยนค่าเป็น 1023

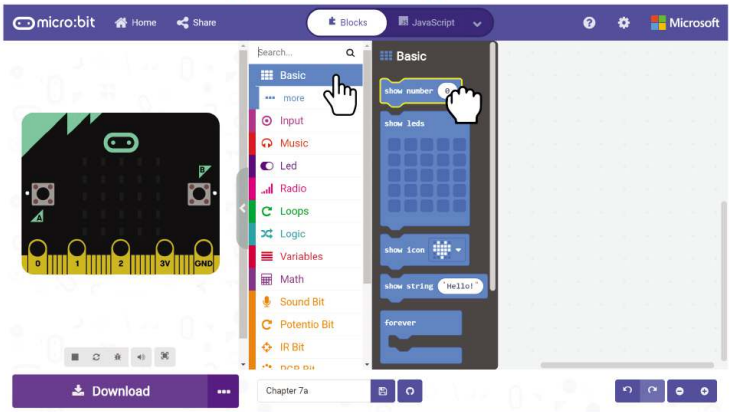


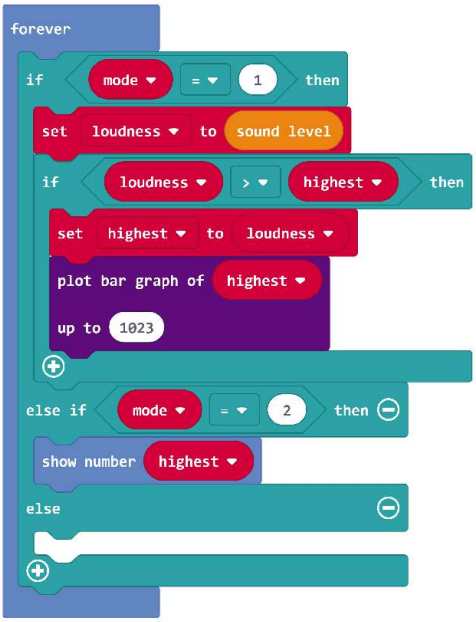




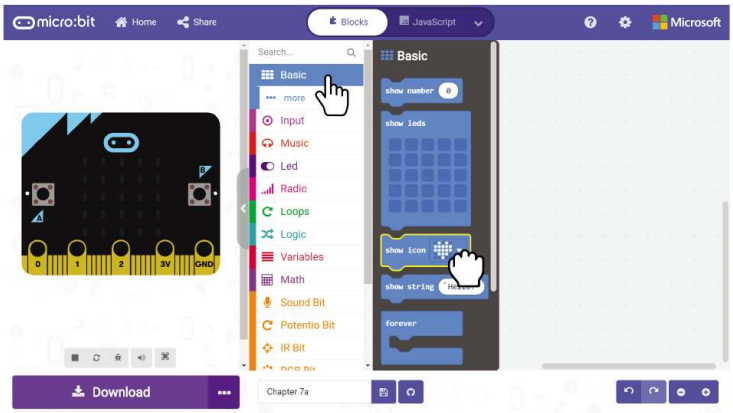


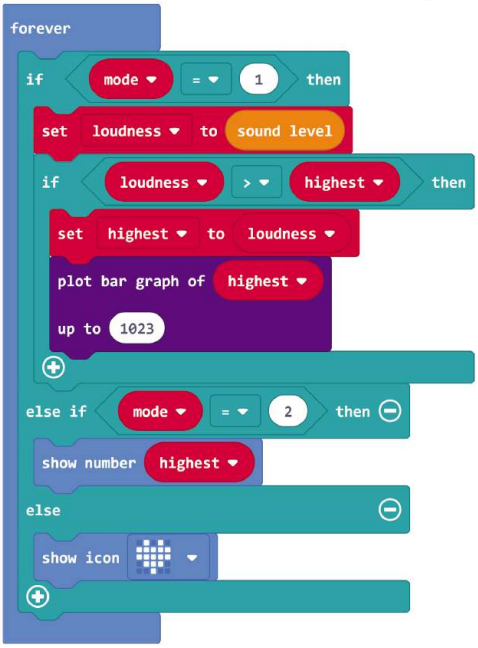
ขั้นตอนที่ 17 เลือกบล็อกคำสั่ง [show number] จากหมวดหมู่ [Basic] แล้วลากไปวางในช่องที่สองของบล็อกคำสั่ง [if-then-else] และเลือกบล็อก [highest] จากหมวดหมู่ [Variables] วางลงในกล่องเก็บค่าของบล็อก [show number]



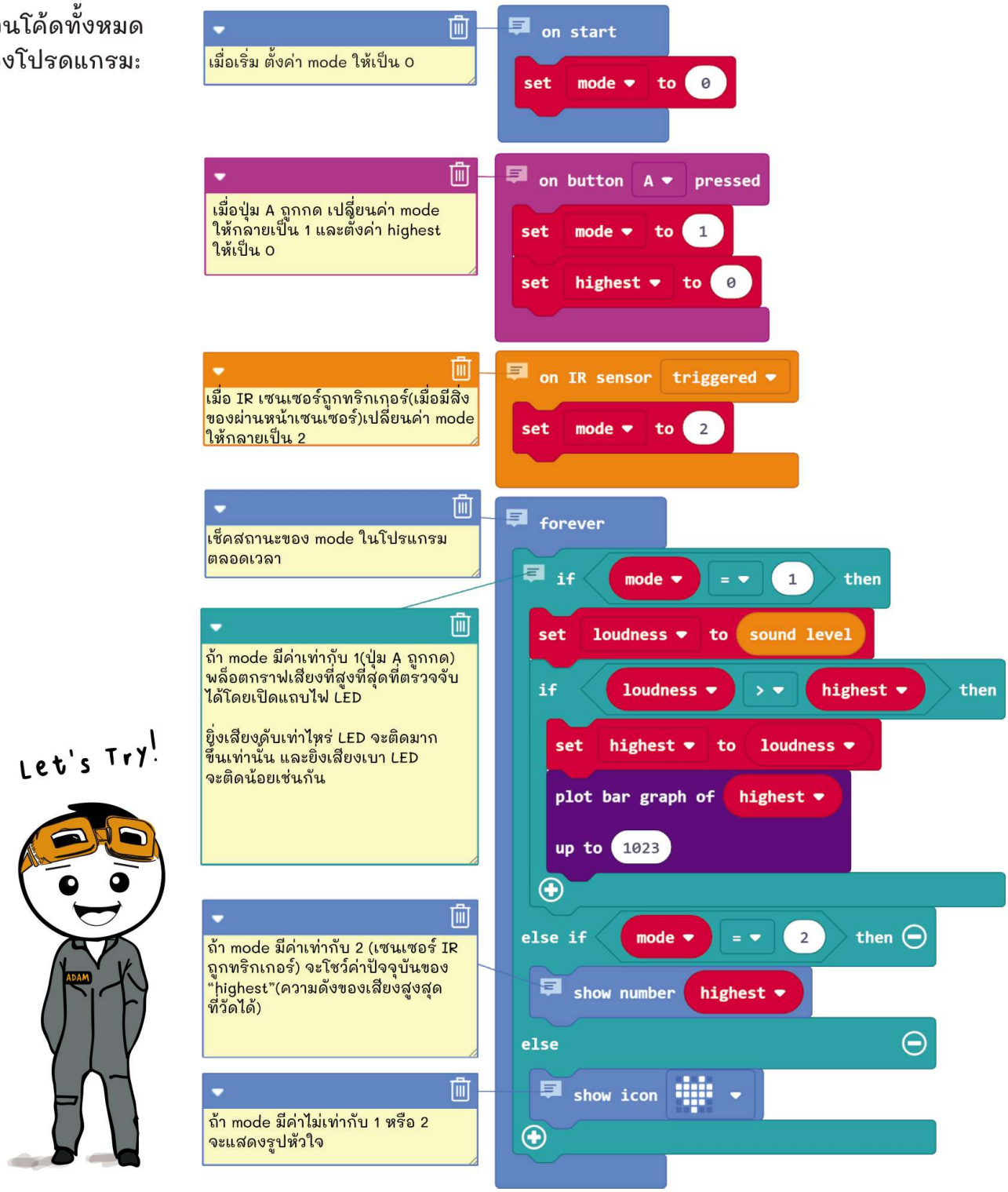


ขั้นตอนที่ 18 เลือกบล็อกคำสั่ง [show icon] จากหมวดหมู่ [Basic] แล้วลากไปวางในช่องสุดท้ายของบล็อกคำสั่ง [if-then-else]





ส่วนโค้ดทั้งหมดของโปรดแกรมะ



ขั้นตอนที่ 19 แฟลชโค้ดลงบอร์ด EDU:BIT และเมื่อมีเครื่องวัดระดับเสียงปรบมือก็ถึงเวลาแสดงของคนเก่งแล้วววววว

Let's Play

Let's Hear the Applause!

วิธีการเล่นะ

- "ผู้เข้าแข่งขัน" จะได้เวลาเตรียมตัวสำหรับการแสดงสั้น ๆ โดยจะแบ่งกันเป็นเดี่ยว ,คู่ หรือกลุ่มก็ได้ แต่ละทีมสามารถเลือกได้ว่าจะแสดงการร้องเพลง ,เต้น หรือเล่าเรื่องตลกก็ได้
- เมื่อทุกคนพร้อมแล้ว ให้แต่ละคนเริ่มการแสดงของตัวเอง หลังจากแต่ละการแสดง "ผู้ชม" จะปรบมือให้ในแต่ละการแสดง ยอ่งดังเท่าไหร่ก็แสดงว่าเขาชอบโชว์นั้นมากเท่านั้น

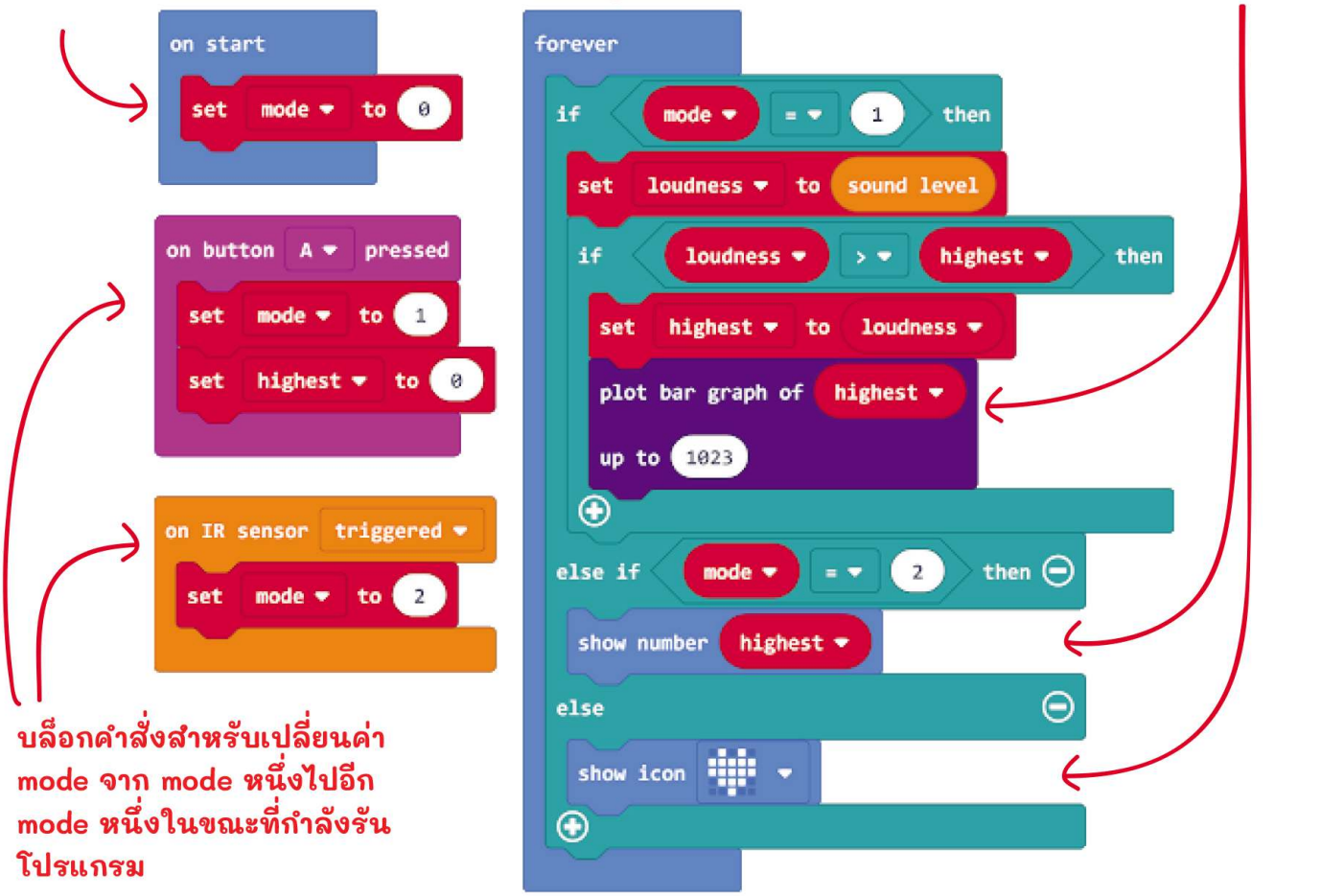
- เมื่อเสียงปรบมือเงียบลง ให้ทริก IR เซนเซอร์ที่บอร์ด(โดยน้อง ๆ สามารถใช้มือเลื่อนผ่านหน้าเซนเซอร์ได้) เพื่อให้หน้าจอแสดงค่าระดับเสียงสูงสุดที่บันทึกไว้
- อย่าลืมกดปุ่ม A เพื่อรีเซ็ตคะแนนที่ตัวบอร์ด ก่อนเริ่มการแสดงถัดไป
- ผู้ชนะคือคนที่ได้คะแนนสูงที่สุด ขอให้สนุกหล่ะ!

BREAK THE CODE

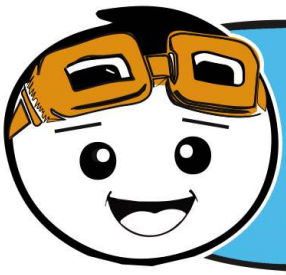
เมื่อน้องๆมีโค้ดหลายๆส่วนในโปรแกรม น้องๆสามารถใช้บล็อกคำสั่งรอเหตุการณ์ที่เกิดได้(ยกตัวอย่างเช่น เมื่อปุ่ม A ถูกกด หรือเมื่อ IR sensors มีวัตถุเคลื่อนที่ผ่าน) และสับเปลี่ยนโปรแกรมไปทำงานในส่วนอื่นได้

เซ็ตค่า mode ให้เป็น 0
(เมื่อเริ่มเปิดโปรแกรม)

เช็คค่า mode ปัจจุบันว่ามีสถานะเป็น
อย่างไรจากนั้นรัน คำสั่งที่เข้าเงื่อนไข mode นั้นๆ



บล็อกคำสั่งสำหรับเปลี่ยนค่า mode จาก mode หนึ่งไปอีก mode หนึ่งในขณะที่กำลังรันโปรแกรม

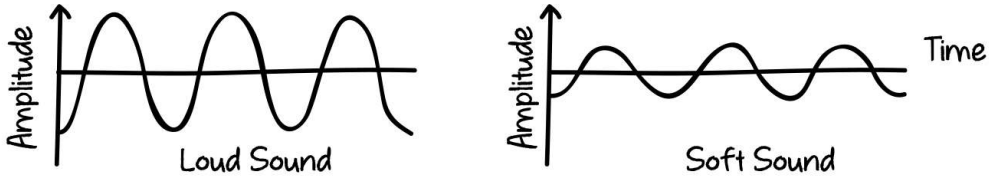


ถ้าน้องๆมี mode ที่จะเพิ่มมากกว่านี้ น้องๆสามารถเพิ่มเงื่อนไขในการเกิดได้ ยกตัวอย่างเช่นบล็อกคำสั่ง [on shake] และ บล็อกคำสั่ง [on sound level > _] เข้าไปในโปรแกรมของน้องๆ และกดที่เครื่องหมายบวกบนบล็อกคำสั่ง [if-then-else] เพื่อเพิ่มเงื่อนไขในการเกิด

FUN FACT!

เสียงถูกสร้างขึ้นเมื่อวัตถุมีการสั่นสะเทือน ยกตัวอย่างเช่น เมื่อกลองถูกตี จะทำให้โมเลกุลของอากาศบริเวณรอบ ๆ นั้นเกิดการสั่นสะเทือน และนั่นจึงทำให้เกิดเสียงขึ้น

เซนเซอร์เสียงคืออุปกรณ์ที่ตรวจวัดความเข้มข้นของคลื่นเสียง(ความดังของเสียง) และแปลงเป็นสัญญาณไฟฟ้า



รู้ไหม? เซนเซอร์เสียงทำงานคล้ายกับหูของเรา เพราะเมื่ออากาศรอบ ๆ หูของเรามีการสั่นสะเทือน หูจะเปลี่ยนการสั่นสะเทือนที่เกิดขึ้นในอากาศเป็นสัญญาณไฟฟ้าเคมี ที่สมองสามารถแปลให้เรารู้ได้ว่าเสียงนั้นคืออะไร

ตัวอย่างการใช้งานจริง

- สัญญาณกันขโมย
- เปิดปิดแสงไฟโดยใช้เสียง
- เครื่องตรวจวัดเสียงเด็กทารก

น้อง ๆ คิดว่าเราสามารถใช้เซนเซอร์วัดเสียงต่าง ๆ ในอวกาศได้หรือไม่ และเพราะอะไร

APPLICATION CHALLENGE

เขียนโปรแกรมให้บอร์ด EDU:BIT สามารถเป็นเครื่องแสดงความดังในห้องเรียนได้ โดยให้แสดงผลเป็นสีตามไฟจราจรบนตามระดับของความดังของเสียงที่เกิดขึ้น

ระดับความดัง	ช่วงระดับเสียงที่วัดได้	สีของไฟ LED ที่แสดงผล
เสียงดังมาก; โปรดลดความดังลงหน่อยนะ	( ) ถึง 1023	ไฟ LED สีแดง
เสียงเริ่มดัง; ระวังการใช้เสียงหน่อยนะ	( ) ถึง ( )	ไฟ LED สี เหลือง
ดีมาก เสียงประมาณนี้เลย	0 ถึง ( )	ไฟ LED สีเขียว

เคล็ดลับเล็ก ๆ น้อย ๆ
เคล็ดลับ#1: น้อง ๆ ต้องเริ่มด้วยการกำหนดช่วงของเสียงในแต่ละระดับความดัง
เคล็ดลับ#2: เพื่อให้ทำงานได้ดี น้อง ๆ ควรหาค่าเฉลี่ยของระดับเสียงที่วัดได้ในช่วงเวลาปกติด้วยนะ

ง่ายไปใช้ไหม? ลองอะไรที่ท้าทายกว่านี้หน่อย เปลี่ยนโค้ดของน้อง ๆ ให้ช่วงของค่าที่กำหนดสัมพันธ์กับค่าที่อยู่กับตัวต้านทานปรับค่าได้ (Potentiometer)

บทที่ 8

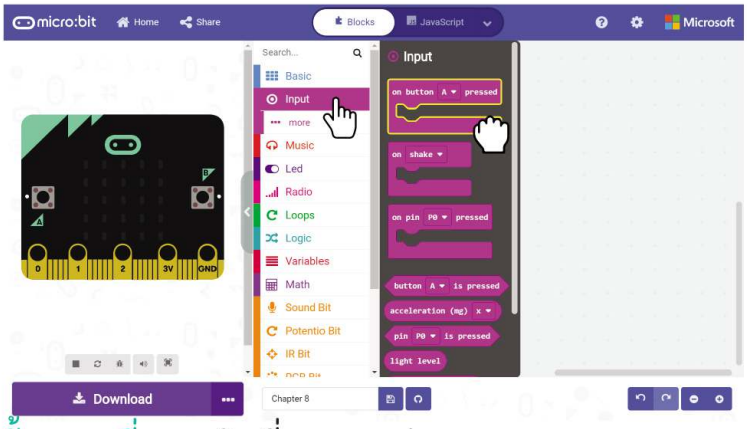
มาลองหมุนมอเตอร์กัน!

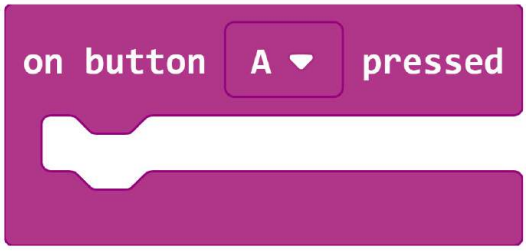
DC Motor

link.cytron.io/edubit-chapter-8

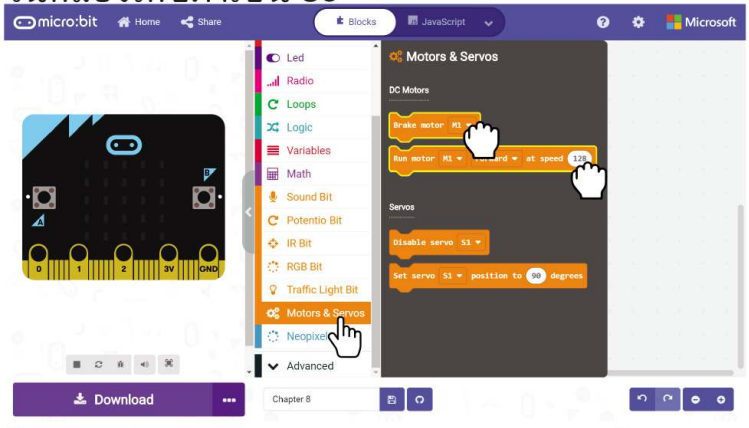
เริ่มเขียนโปรแกรม

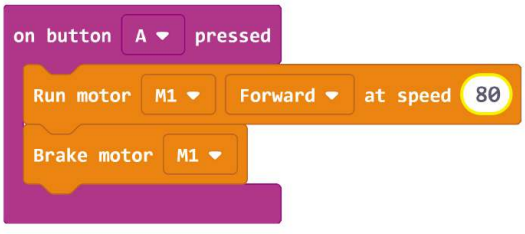
ขั้นตอนที่ 1 ในหน้า Makecode editor กดเริ่มโปรเจ็คใหม่และเพิ่มส่วนเสริม EDU:BIT (สามารถกลับไปดูได้ที่หน้า 40) คลิกหมวดหมู่ [Input] และเลือกบล็อกคำสั่ง [on button pressed]



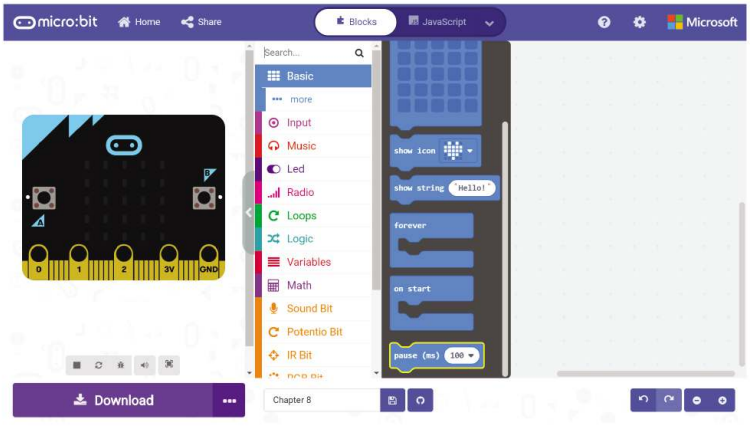


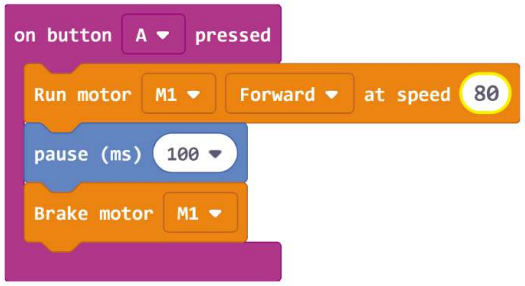
ขั้นตอนที่ 2 คลิกที่หมวดหมู่ [Motor & Servos] จากนั้นลากบล็อกคำสั่ง [Run motor __ at speed __] และ [Brake motor __] มาที่หน้าจอเขียนโปรแกรม จากนั้นเปลี่ยนค่าความเร็วในกล่องเก็บค่าเป็น 80



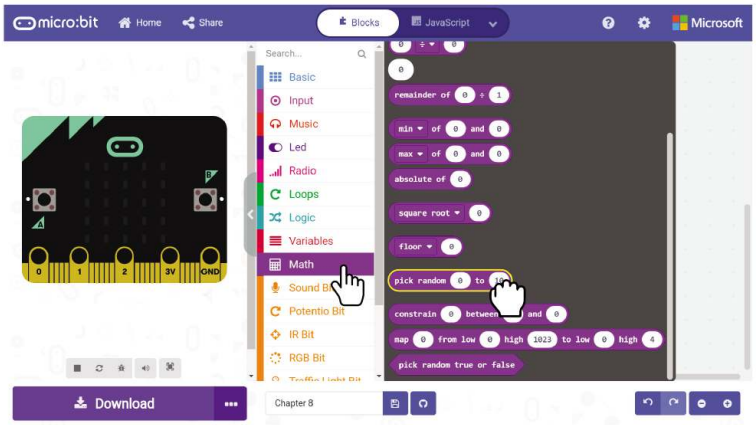


ขั้นตอนที่ 3 คลิกที่หมวดหมู่ [Basic] จากนั้นลากบล็อกคำสั่ง [pause__] ไปวางคั่นกลางระหว่างบล็อกคำสั่ง [Run motor __ at speed __] และบล็อกคำสั่ง [Brake motor__]

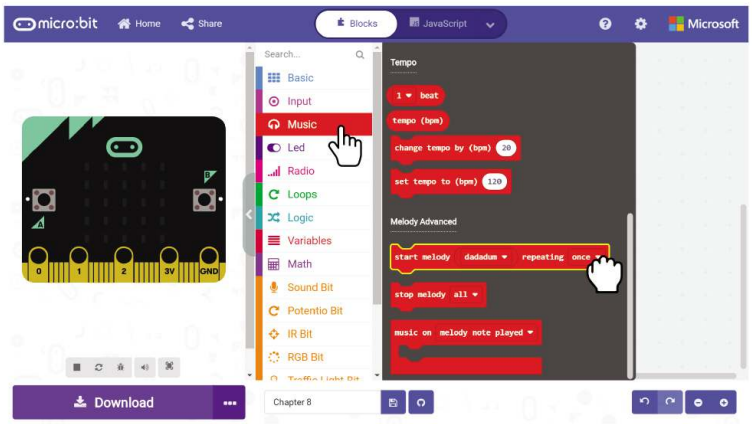


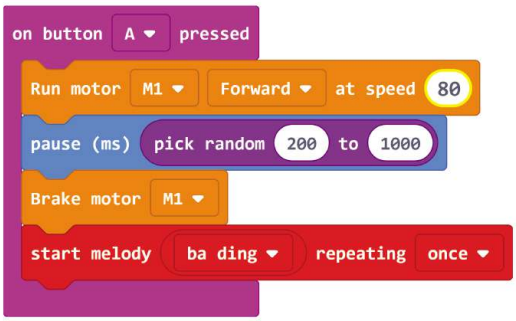


ขั้นตอนที่ 4 คลิกที่หมวดหมู่ [Math] และเลือกบล็อกคำสั่ง [pick random __ to __] วางไว้ในบล็อก [pause__] และเปลี่ยนค่าในกล่องเก็บค่าเป็น 200 และ 1000 ตามลำดับ



ขั้นตอนที่ 5 คลิกที่หมวดหมู่ [Music] และเลือกบล็อกคำสั่ง [start melody __ repeating __] เปลี่ยนเสียงเพลงเป็น “ba ding” (หรือเลือกเสียงดนตรีที่น้อง ๆ ชอบ)

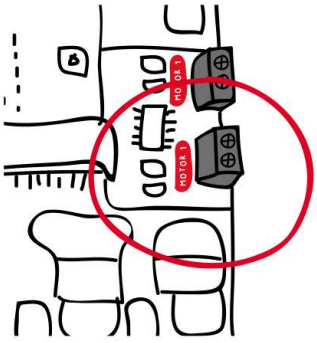


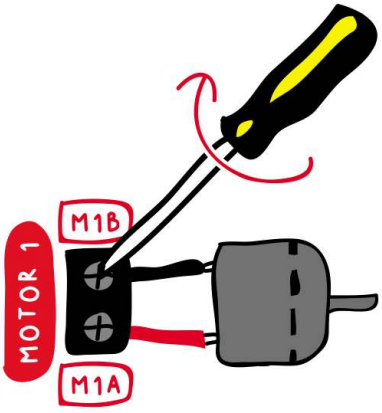


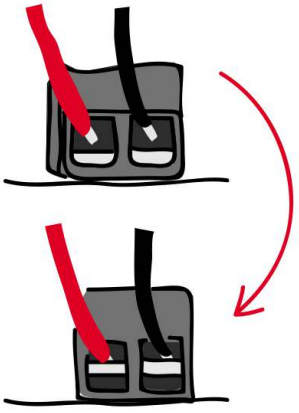
ขั้นตอนที่ 6 แฟลชโค้ดลงบอร์ด EDU:BIT

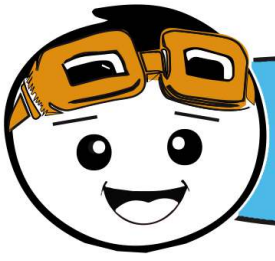
น้องๆ สามารถใช้โปรแกรมนี้ ในการเล่นกิจกรรมที่ใช้เครื่องหมุนได้ โดยโปรแกรมนี้ เมื่อเปิดมอเตอร์จะเริ่มหมุนไปด้วยเวลาที่สุ่มขึ้นมา จากนั้นจึงหยุด

ขั้นตอนที่ 7 ต่อ DC motor(มอเตอร์ไฟฟ้ากระแสตรง) ที่ตำแหน่ง motor 1 บนบอร์ด เสียบสายเข้าไปในช่องและใช้ไขควงที่ได้มา ไขยึดไว้ให้แน่น



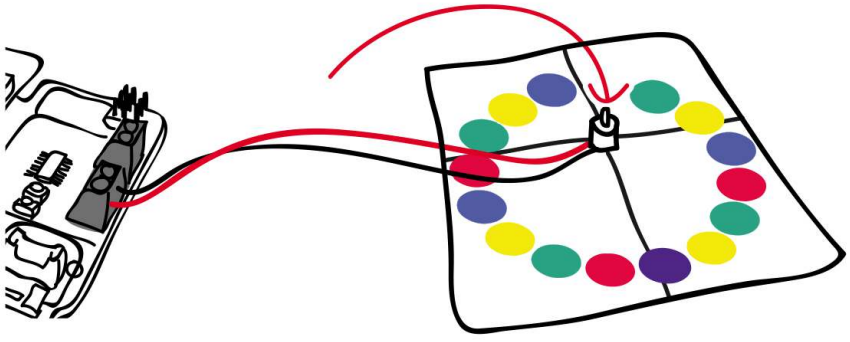




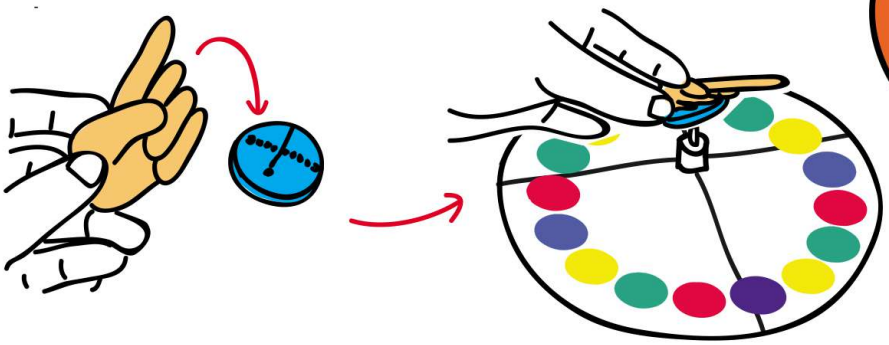


กดที่ปุ่มสีเหลือง (ปุ่ม A) เพื่อทดสอบมอเตอร์ ถ้ามอเตอร์ไม่หมุน ให้น้องๆลองตรวจดูตรงสายไฟที่ต่อระหว่างมอเตอร์กับบอร์ดว่าแน่นหรือไม่ และตรวจดูว่าบอร์ด EDU:BIT เปิดแล้วหรือไม่

ขั้นตอนที่ 8 ใช้กาวสองหน้าหรือว่ากาวร้อนติดมอเตอร์เอาไว้ที่ศูนย์กลางของแท่นหมุน ตามรูปภาพ



ขั้นตอนที่ 9 นำเลเซอร์ ไปติดไว้ที่จานสีพลาสติกด้วยกาว จากนั้นติดจานพลาสติกไว้บนแกนมอเตอร์



จานสีพลาสติก, เลเซอร์ และแผ่นรองสนาม อยู่ในกล่องแล้วว

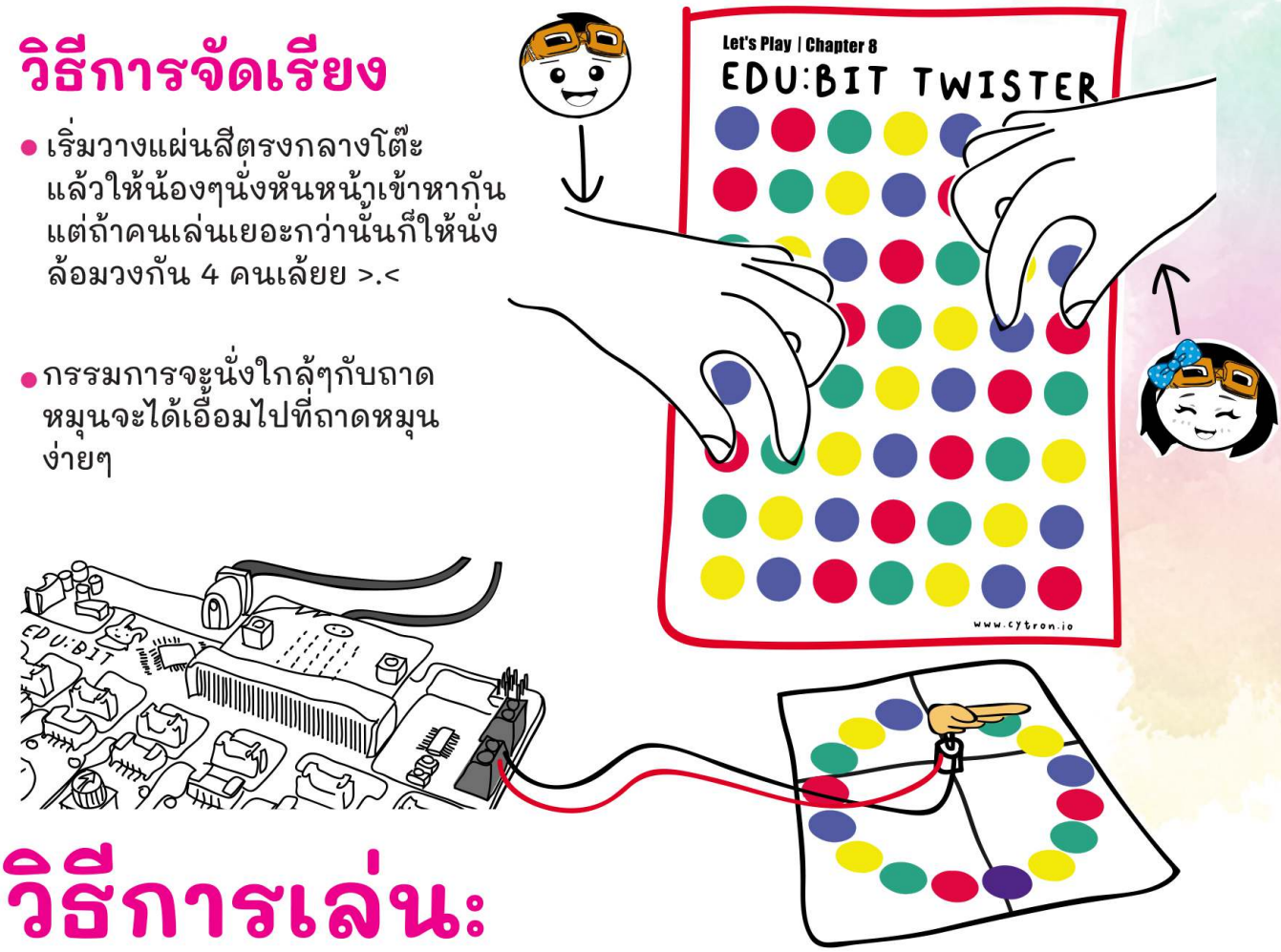
Let's Play

Let's Take a Spin!

วิธีการจัดเรียง

- เริ่มวางแผ่นสีตรงกลางโต๊ะ แล้วให้น้องๆนั่งหันหน้าเข้าหากัน แต่ถ้าคนเล่นเยอะกว่านั้นก็ให้นั่งล้อมวงกัน 4 คนเลยย >.<

- กรรมการจะนั่งใกล้ๆกับถาดหมุนจะได้เอื้อมไปที่ถาดหมุนง่ายๆ



วิธีการเล่นะ

ผู้เล่นจะทำการสลับกันวางนิ้วบนถาดสีตรงกลางตามที่กรรมการบอก

กรรมการจะกดปุ่มสีเหลือง(หรือปุ่ม A) เพื่อเริ่มหมุนวงล้อสีและบอกผู้เล่นว่าต้องวางนิ้วไหนไปวางที่สีอะไร ยกตัวอย่างเช่น ใช้นิ้วชี้ วางที่สีแดง เป็นต้น

เมื่อถึงรอบของน้องๆ น้องๆจะต้องรอฟังกรรมการว่ากรรมการบอกนิ้วและสีซึ่งน้องๆจะต้องวางนิ้วให้ถูกนิ้วลงบนสีที่กรรมการบอก ถ้าสมมุติว่านิ้วที่กรรมการบอกวางไว้ที่สีที่กำหนดอยู่แล้วให้ลองย้ายนิ้วนั้นไปไว้ที่วงกลมอื่นที่เป็นสีเดียวกัน

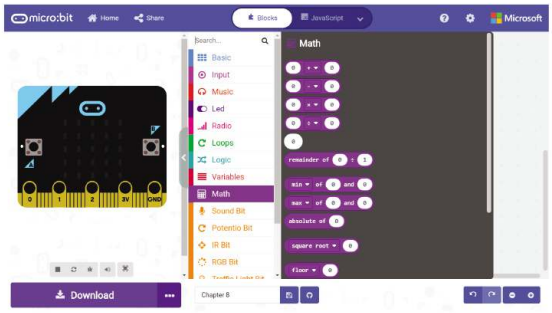
ถ้าน้องๆไม่สามารถทำตามที่กรรมการบอกได้ น้องคนนั้นๆจะแพ้ออกจากเกม

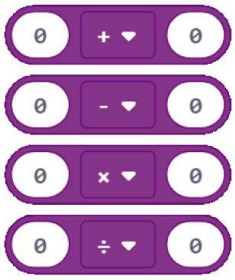
น้องที่เหลืออยู่คนสุดท้ายเป็นผู้ชนะ!

EXPLORE MORE BLOCKS

น้องๆสามารถใช้บล็อกคำสั่งในหมวดหมู่ [Math] มาใช้สำหรับการคำนวณทางคณิตศาสตร์ได้

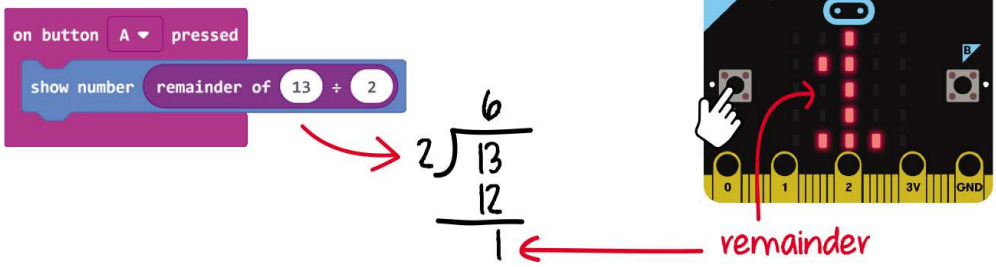
#1 น้องๆสามารถใช้บล็อกดังต่อไปนี้เพื่อ บวก ลบ คูณ หรือ หารได้



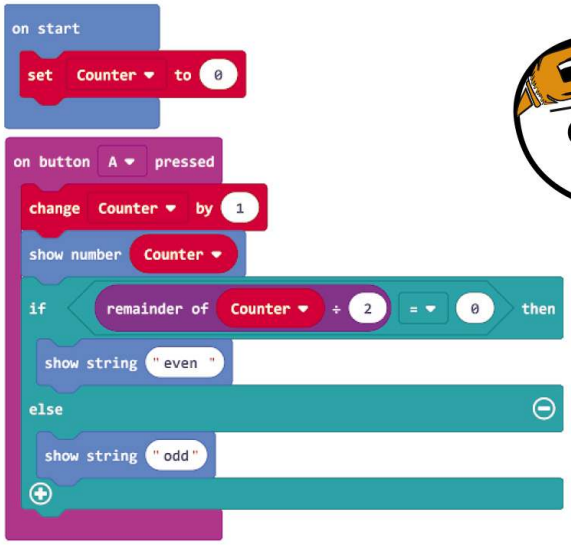


#2 น้องๆสามารถใช้บล็อกคำสั่ง [remainder of __ ÷ __] เพื่อหาว่าเลขที่นำมาหารเหลือเศษเท่าไหร่

For example:



#3 น้องๆ สามารถใช้บล็อกคำสั่ง [remainder of __ ÷ __] เพื่อแยกว่าเลขไหนเป็นเลขคู่หรือเลขคี่โดยการนำไปหาร 2 ถ้าเหลือเศษ 1 ก็จะเป็นเลขคี่ ถ้าเหลือเศษ 0 ก็จะเป็นเลขคู่ มาลองกันเถอะ

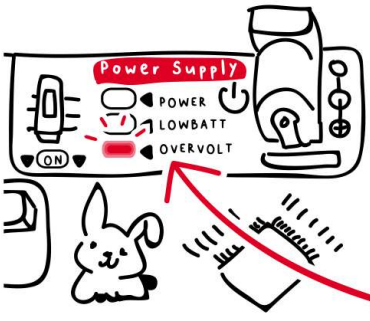


น้องๆเคยสังเกตุหรือเปล่าว่าบอร์ด EDU:BIT จะมีคำว่า “odd” โผล่ขึ้นมาเวลากดปุ่ม A ครั้งแรก (1 = เลขคี่) และขึ้นว่า “even” เมื่อกดปุ่มซ้ำอีกครั้งนึง (2 = เลขคู่) ลองดูซิว่าจะเกิดอะไรขึ้นถ้าน้องๆกดปุ่ม A ทั้งหมด 99 ครั้ง?

FUN FACT!

มอเตอร์กระแสตรง หรือที่รู้จักกันคือมอเตอร์ดีซี (DC motor) คืออุปกรณ์ไฟฟ้าที่เปลี่ยนพลังงานไฟฟ้าให้กลายเป็นการหมุน

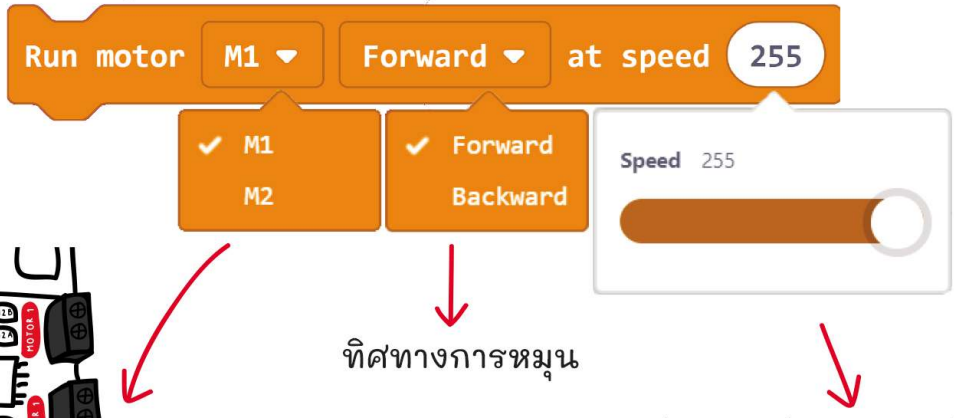
น้อง ๆ ต้องจ่ายไฟเข้าไปเพื่อให้มอเตอร์ ดีซีหมุน โดยเราสามารถควบคุมความเร็วในการหมุนได้โดยการปรับค่าไฟที่จ่ายเข้าไปให้กับมอเตอร์ ยิ่งค่าไฟสูงมอเตอร์จะยิ่งหมุนเร็ว โดยค่าไฟที่เหมาะสมกับมอเตอร์บน EDU:BIT จะอยู่ที่ 3.6 ถึง 6 โวลต์
EDU:BIT kit is 3.6V - 6V.

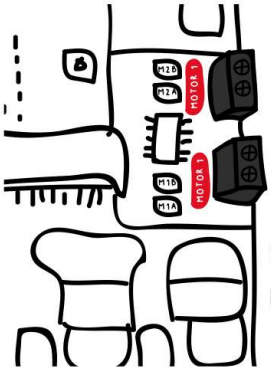


ข้อควรระวัง!
การจ่ายปริมาณไฟมากกว่าที่แนะนำจะทำให้มอเตอร์พังได้เร็วขึ้น

ไฟบ่งบอกว่าจ่ายไฟฟ้ามากเกิน

น้องๆสามารถควบคุมทิศทางการหมุนและความเร็วของมอเตอร์ได้โดยการใช้บล็อกคำสั่งตามรูป





มีช่องสำหรับติดมอเตอร์อยู่ 2 ช่องบนบอร์ด EDU:BIT เลือกให้ถูกช่องล่ะ!

ทิศทางการหมุน

นี่เป็นค่าที่อยู่ในช่วงที่เหมาะจาก 0 ถึง 255 โดยยิ่งค่าสูงมอเตอร์จะยิ่งหมุนเร็ว ยิ่งค่าน้อยมอเตอร์จะยิ่งหมุนช้า

รู้หรือเปล่าว่าบอร์ด EDU:BIT มีวงจรสำหรับทดสอบมอเตอร์ด้วยนะ ลองกดที่ปุ่มสีขาว(ปุ่มที่มีเขียนว่า M1A, M1B, M2A, M2B) เพื่อลองทดสอบดูว่ามอเตอร์ใช้งานได้

APPLICATION CHALLENGE

เขียนโปรแกรม EDU:BIT ซึ่งใช้เสียงในการควบคุมการหมุนของมอเตอร์และกำหนดความเร็วโดยใช้ตัวต้านทานปรับค่าได้	
บล็อกคำสั่ง [on start]	แสดงรูปหัวใจ(หรือรูปที่น้องๆชอบ) และตั้งตัวแปร “mode” ให้มีค่าเป็น 0
บล็อกคำสั่ง [on sound level > __]	เพิ่มค่าของ “mode” 1
บล็อกคำสั่ง [forever]	ตั้งค่าตัวแปร “speed” ให้มีค่าตามตัวต้านทาน ปรับค่าได้จาก 0-1023 เป็น 0-255 และ เช็ค “mode” ตลอดเวลา • ถ้า “mode” เป็นเลขคู่ให้มอเตอร์หยุดวิ่ง • ถ้า “mode” เป็นเลขคีให้มอเตอร์วิ่งด้วยความเร็วเท่ากับตัวแปร “speed” (ตัวแปร “speed” มีค่าเท่ากับตัวต้านทานปรับค่าได้)

เคล็ดลับสำหรับน้องๆ

เคล็ดลับ 1 น้องต้องลองสังเกตุว่าจะตั้งค่าบล็อกคำสั่ง [on sound level >__] ไว้ที่ค่าเท่าไหร่ถึงจะสามารถสั่งให้มอเตอร์หมุน/หยุดหมุนได้

เคล็ดลับ 2 น้องๆต้องสร้างตัวแปร “mode” กับ “speed”

เคล็ดลับ 3 ลองติดใบพัดลมเอาไว้ที่มอเตอร์แล้วสั่งให้โปรแกรมทำงาน ถ้าน้องๆไม่รู้สึกว่าไม่มีลมพัดให้ลองสลับฝั่งที่ขั้วมอเตอร์หมุน จากหมุนไปข้างหน้าให้หมุนถอยหลัง หรือจากหมุนถอยหลัง ให้หมุนไปข้างหน้า

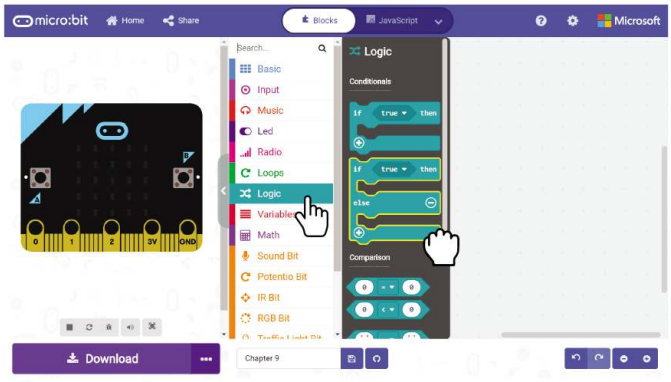
บทที่ 9

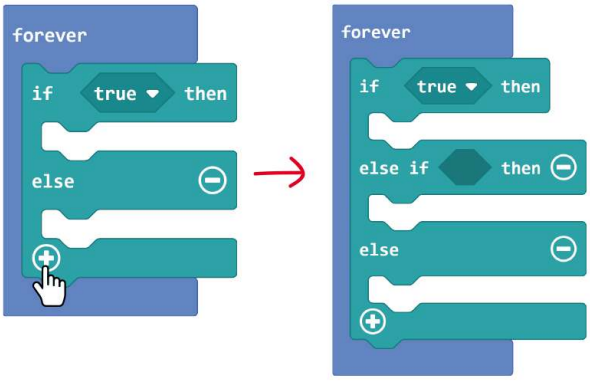
ยิงลูกโทษ...เข้าไปแล้ว!!!

Servo Motor

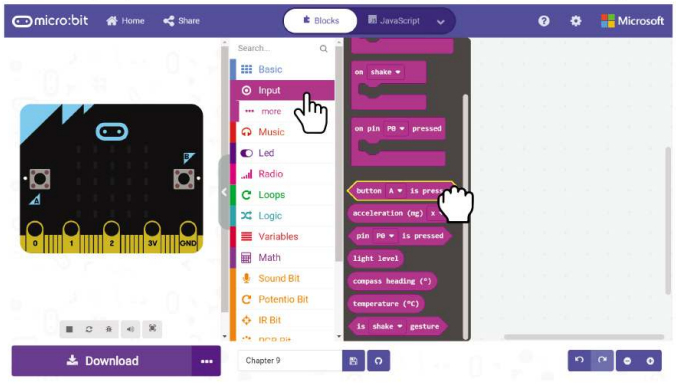
เริ่มเขียนโปรแกรม

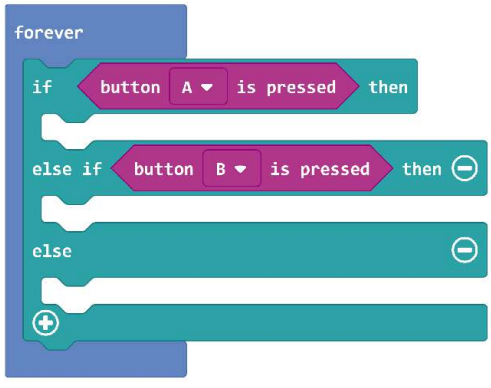
ขั้นตอนที่ 1 สร้างโปรเจกต์ใหม่ใน MakerCode Editor และเพิ่มส่วนเสริม EDU:BIT (สามารถกลับไปดูได้ที่หน้า 40) คลิกที่หมวดหมู่ [Logic] แล้วเลือกบล็อกคำสั่ง [if-then-else] จากนั้นลากบล็อกมาวางที่ช่อง [forever] ทำการเลือกเครื่องหมาย + เพื่อเพิ่มเงื่อนไข else — if ลงในบล็อก



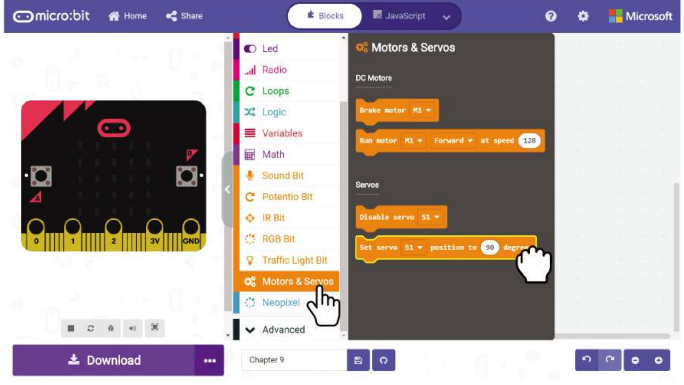


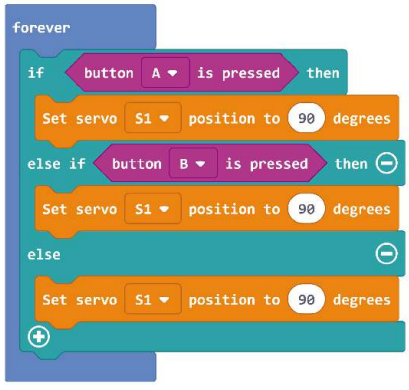
ขั้นตอนที่ 2 คลิกที่หมวดหมู่ [Input] แล้วเลือกบล็อก [button_ is pressed] จากนั้นลากบล็อก[button_ is pressed] มาวางที่บล็อกเงื่อนไข [if-then-else] ทำแบบนี้อีกครั้ง แต่ในบล็อกที่สองให้เปลี่ยนเป็น 'button B'



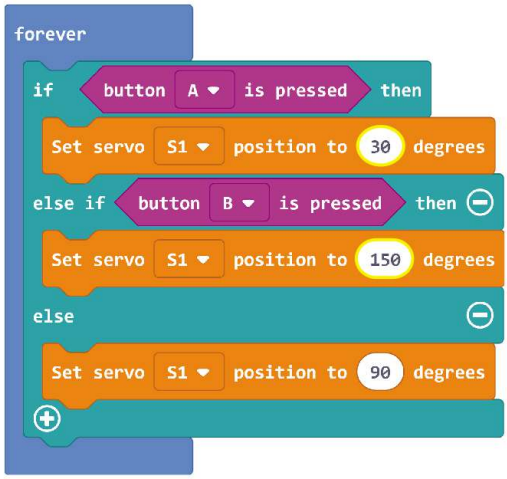


ขั้นตอนที่ 3 คลิกที่หมวดหมู่ [Motors & Servos]แล้วเลือกบล็อก [set servo_position to_ drgrees] จากนั้นลากบล็อก[set servo_position to_drgrees] ไปวางที่บล็อกเงื่อนไข [if-then-else] และทำแบบนี้กับทุกช่อง

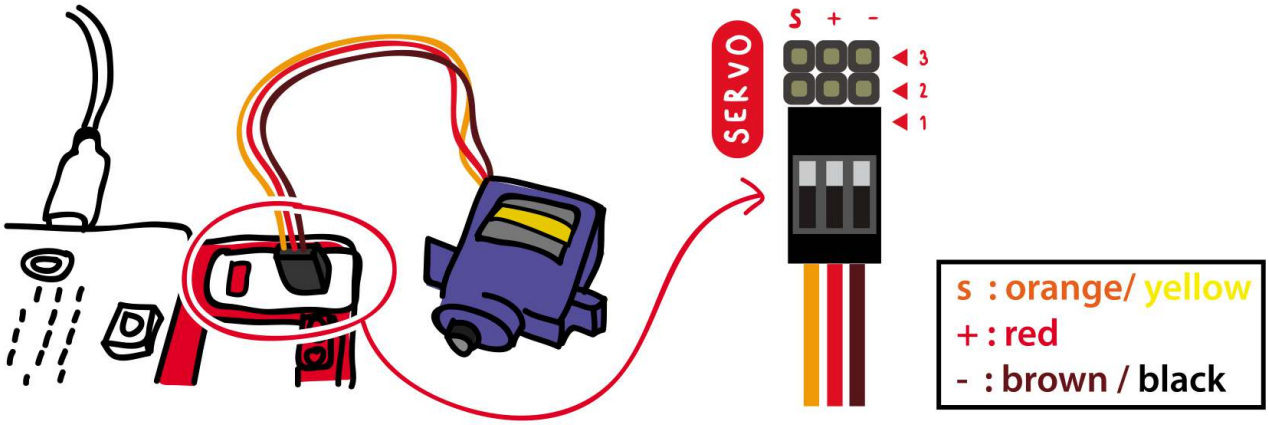




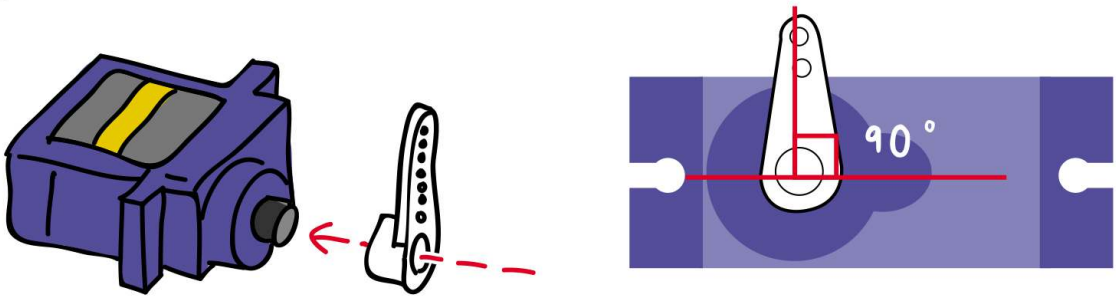
ขั้นตอนที่ 4 เปลี่ยนค่าในบล็อกแรกเป็น 30 และบล็อกที่สองเป็น 150 จากนั้นแฟลชโค้ดลงบอร์ด EDU:BIT



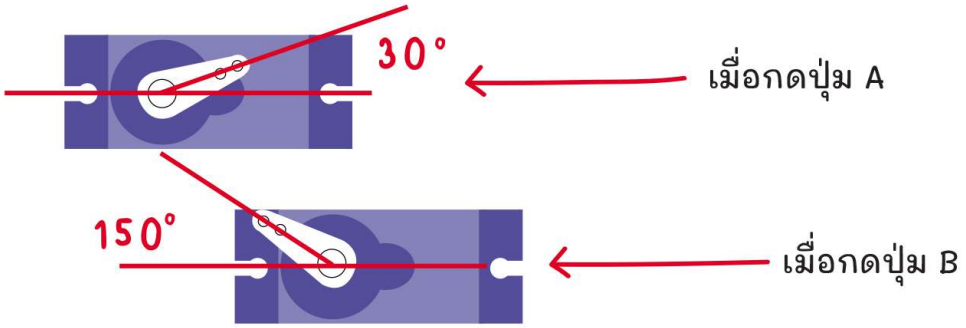
ขั้นตอนที่ 5 เสียบสาย servo motor และต่อเข้ากับ Servo port 1 บนบอร์ด EDU:BIT ตามรูปด้านล่าง



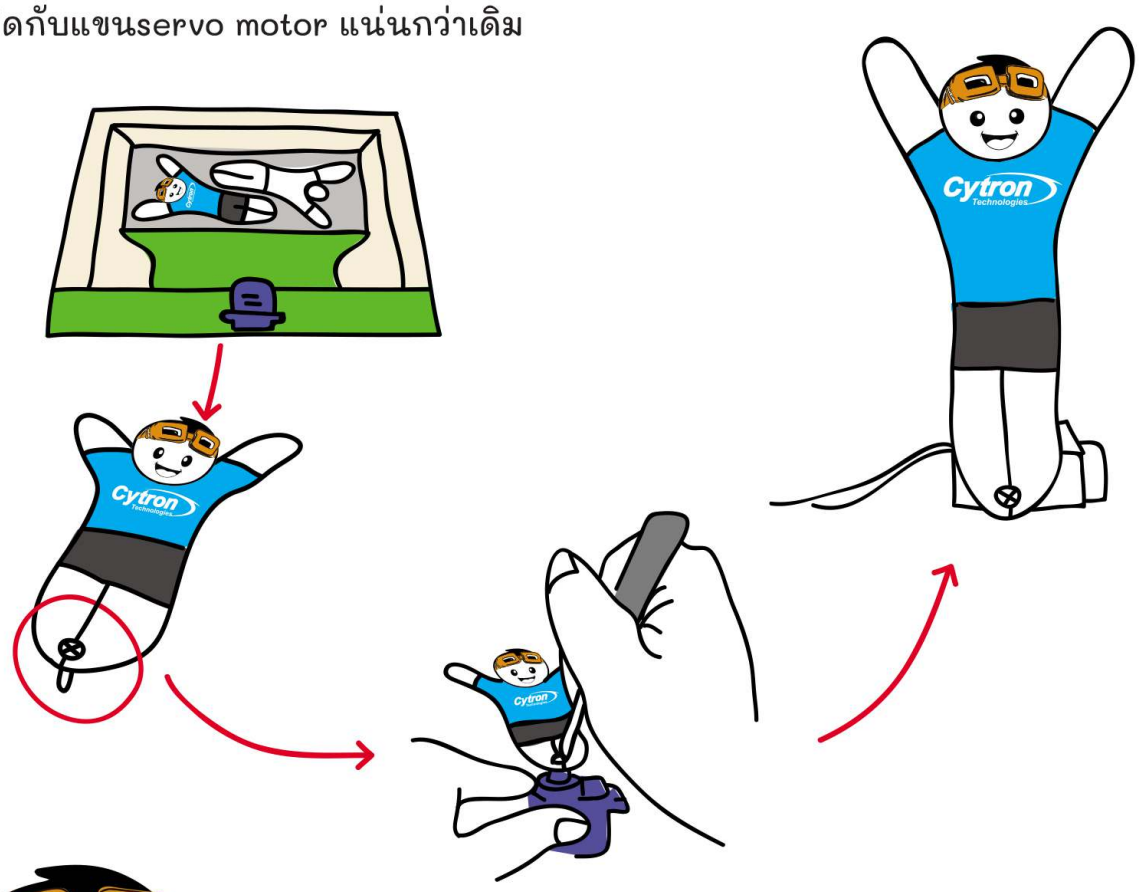
ขั้นตอนที่ 6 จ่ายไฟเข้าบอร์ดEDU:BIT จากนั้นติดแขนไปที่ Servo motor 90 องศา ตามรูปด้านล่าง

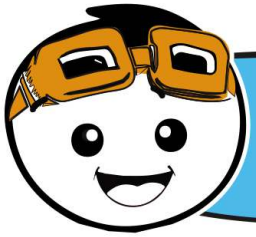


ขั้นตอนที่ 7 กดปุ่ม A จากนั้นกดปุ่ม B เพื่อทดสอบ



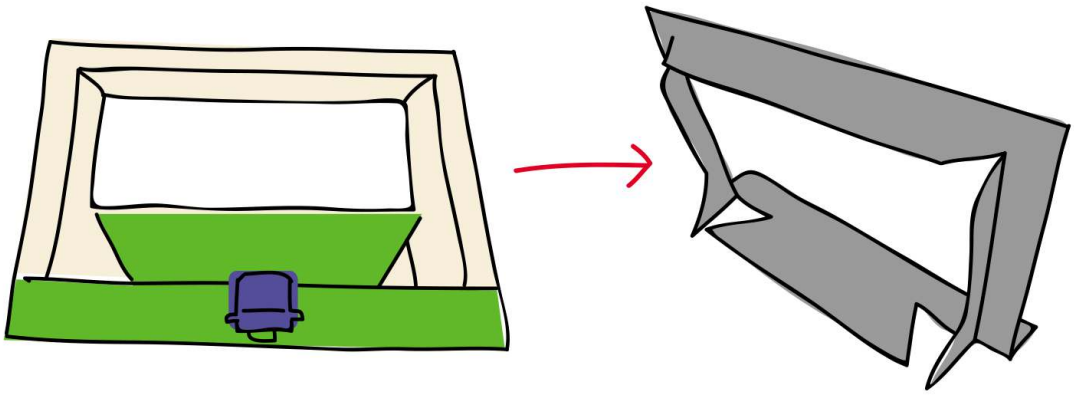
ขั้นตอนที่ 8 นำป็อปอัพผู้รักษาประตูจากการ์ดที่ให้ มาไขเข้ากับแขนของ Servo motor หรือน้องๆจะใช้กาวร้อนหรือเทปสองหน้ามาติดเพิ่มเพื่อให้มั่นใจกว่าเดิมว่าผู้รักษาประตูจะติดกับแขนservo motor แน่นกว่าเดิม



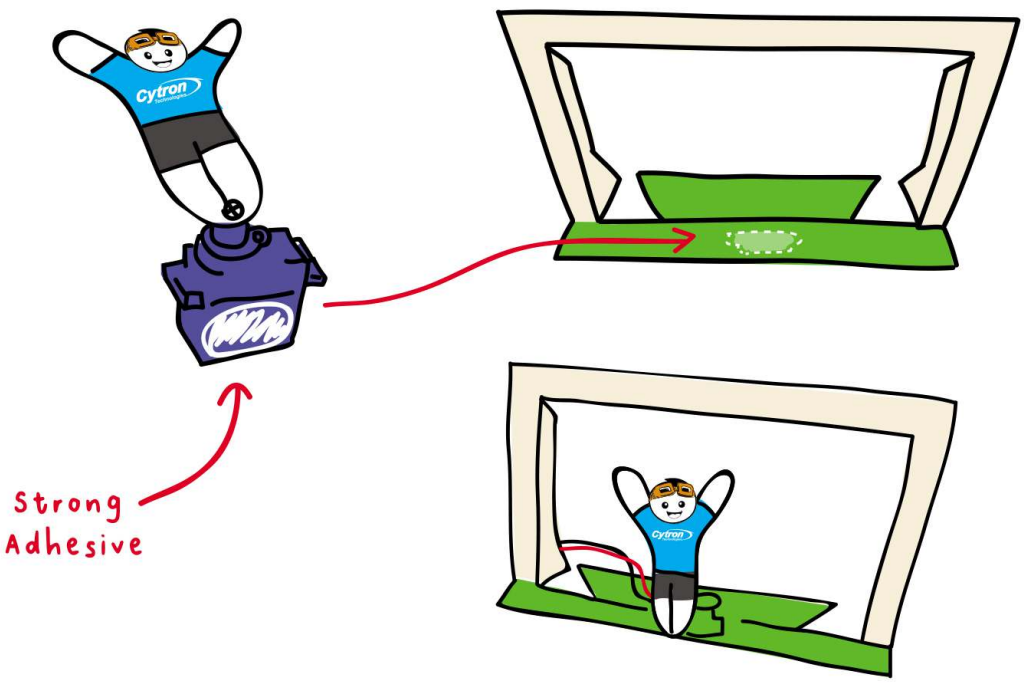


น้องๆ สามารถนำป็อปอัพอื่นๆหรือออกแบบตัวการ์ตูนของตัวเองเพื่อมาติดเป็นผู้รักษาประตูก็ได้นะ!!!

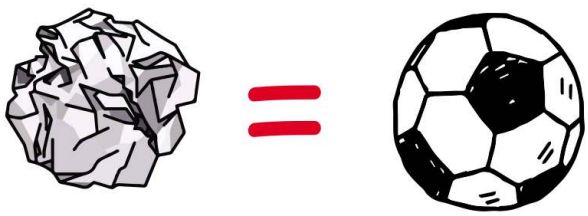
ขั้นตอนที่ 9 นำป๊อปอัพโกลออกมาจัดตั้งให้เรียบร้อย



ขั้นตอนที่ 10 นำกาวที่ติดแน่น เช่น เทปสองหน้า หรือกาวร้อน เพื่อนำ Servo motor ติดเข้ากับโกล ตามตำแหน่งที่ระบุไว้

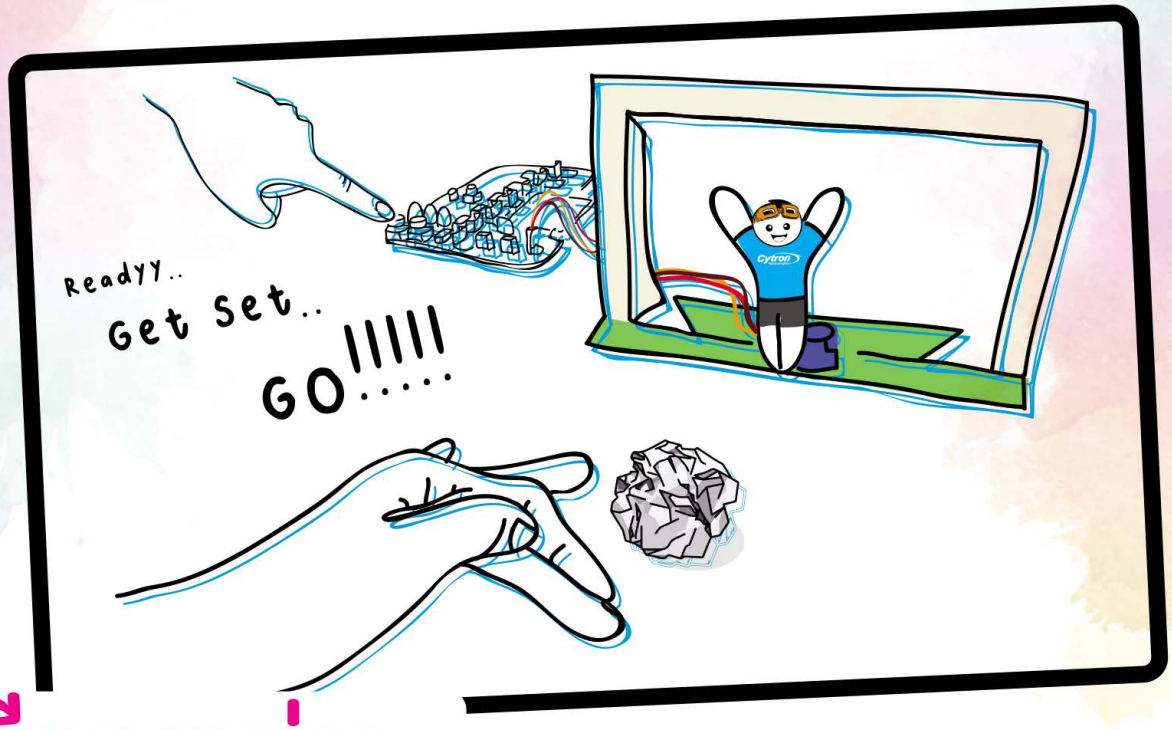


ขั้นตอนที่ 11 นำเศษกระดาษมาปั้นให้เป็น ลูกบอล พวกเราพร้อมที่จะสนุกไปกับเกมส์ยิงลูกโทษแล้ว! น้องๆพร้อมรึยังงง?!!



Let's Play

Penalty Shoot-Out... Goal!!!



วิธีการเล่น :

จัดเตรียมตำแหน่งของโกผู้เล่นดีดลูกบอลเข้าใส่ประตูและทำเครื่องหมายจุดยิงลูกโทษ (ประมาณ 1 เมตร จากโกล หรือปรับระยะทางสำหรับผู้เล่นที่อายุน้อย)

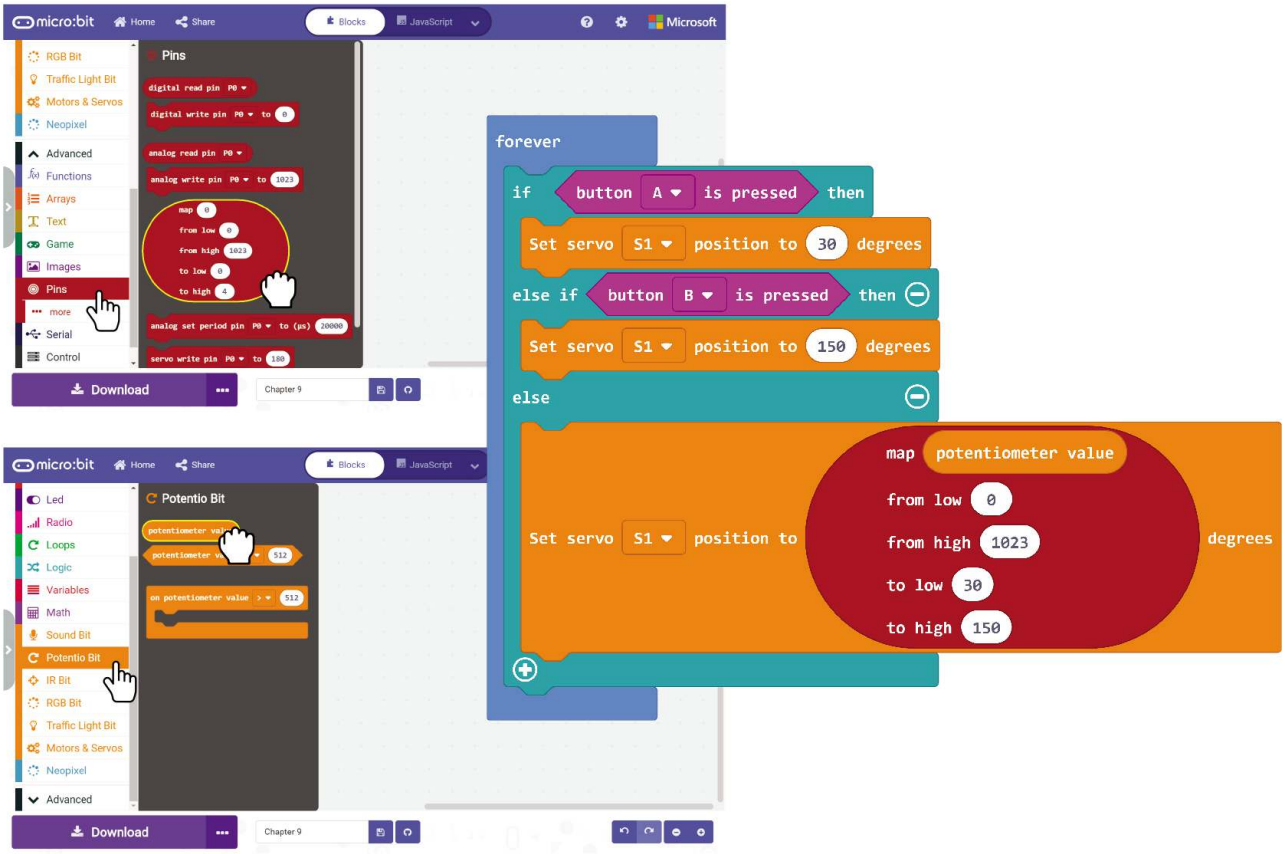
- ผู้เล่นผลัดกันเป็นผู้ยิงประตูและผู้รักษาประตู
- ผู้เล่นดีดลูกบอลเข้าใส่ประตู
- ผู้รักษาประตูป้องกันประตูโดย กดปุ่มสีเหลือง(A)เพื่อป้องกันทางซ้าย และ กดปุ่มสีฟ้า (B)เพื่อย้ายไปทางขวา
- ในหนึ่งรอบ ผู้เล่นแต่ละคนจะได้รับโอกาสในการยิง 5 รอบ ผู้เล่นคนไหนที่มีคะแนนสูงสุด ก็จะเป็นผู้ชนะ !

น้องๆรู้หรือไม่ การยิงจุดโทษมีไว้เพื่อตัดสินผู้ชนะ ในการแข่งขันฟุตบอลเมื่อทีมทั้งสองทีมมีคะแนนเท่ากันในตอนท้ายการแข่งขันจะมีการเพิ่มเวลาพิเศษให้ ในการยิงจุดโทษแต่ละทีมจะมีโอกาสยิงทีมละ 5 นัด ผู้เล่นจะยิงลูกโทษจากจุดที่กำหนดไว้ไปยังประตู ซึ่งป้องกันโดยผู้รักษาประตูของทีมตรงข้ามเท่านั้น ชัยชนะจะตกเป็นของทีมที่ยิงประตูได้ !!!!

โค้ดก่อนหน้าที่เขียน เราทำให้ผู้รักษาประตูหมุนไปทางซ้ายและขวาได้ แต่เราสามารถเขียนโค้ดเพื่อกำหนดตำแหน่งของผู้รักษาประตูได้ด้วยนะ โดยการใช้ Potentio Bit

ขั้นตอนที่ 12 คลิกที่หมวดหมู่ [Advanced] แล้วเลือกหมวดหมู่ [Pins] เพื่อบล็อก [map_ from low _from high_to low_to high_]ลงในโค้ด

ขั้นตอนที่ 13 คลิกที่หมดหมู่ [Potentio Bit] แล้วเลือกบล็อก [potentiometer value] จากนั้นลากบล็อก [potentiometer value] มาวางที่บล็อก [map_from low _from high_to low_to high_] และเปลี่ยนค่า สองค่าสุดท้ายเป็น 30 และ 150 ตามลำดับ



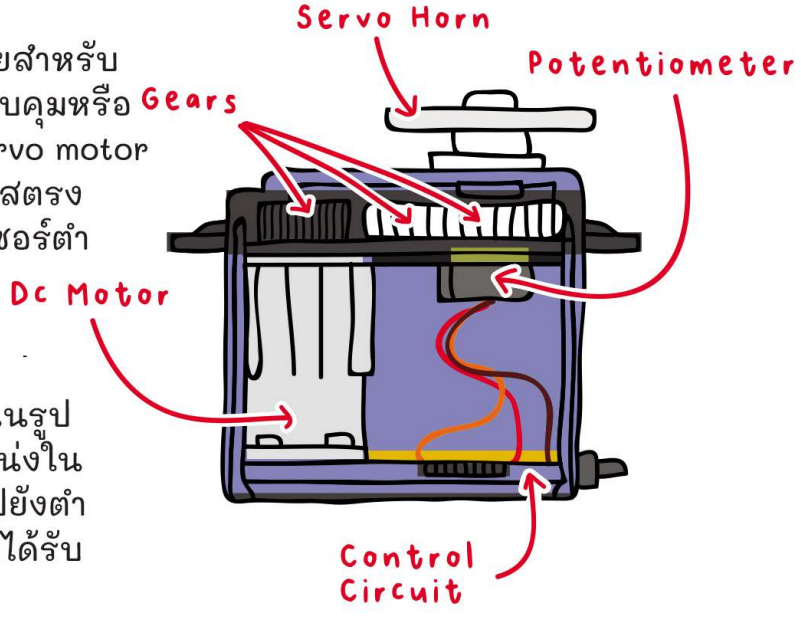
ขั้นตอนที่ 14 แฟลชโค้ดไปยัง บอร์ด EDU:BIT ตอนนี้น้องๆสามารถควบคุมตำแหล่งของผู้รักษาประตูได้แล้ว ไปสนุกกันนนน

หากน้องๆต้องการฝึกซ้อมการยิงลูกโทษด้วยตัวเอง น้องๆสามารถแก้ไขโค้ดเป็น โหมดฝึกซ้อม (practice mode) โดยทำให้ผู้รักษาประตูหมุนไปทางซ้ายและขวาอย่างต่อเนื่อง ลองดู!

FUN FACT!

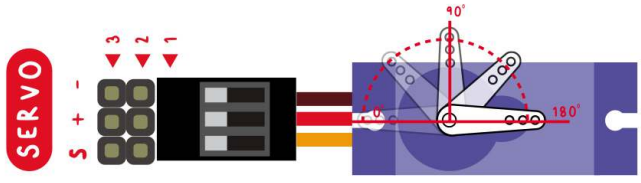
Servo motor ใน EDU:BIT kit เป็นที่รู้จักกัน
ในชื่อ RC(radio control) servo มีการใช้งานกันอย่างแพร่หลาย
ในของเล่น RC และหุ่นยนต์ขนาดเล็กเพื่อควบคุมการเคลื่อนไหว

Servo motor ใช้ระบบสามสายสำหรับ power(+) ground(-) และ ควาบคุมหรือสัญญาณ(s) แล้วโดยทั่วไป Servo motor จะประกอบด้วย มอเตอร์กระแสตรง เกียร์ โพเทนชิโอมิเตอร์(เซนเซอร์ตำแหน่ง)และวงจรควบคุม



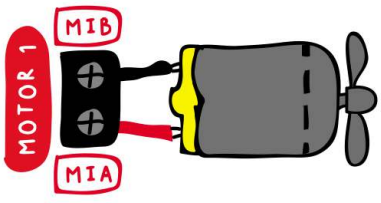
ตัวควบคุมในตัวจะแปลคำสั่งในรูปแบบของ จังหวะ ไปเป็นตำแหน่งในองศา Servo motor จะหมุนไปยังตำแหน่งที่สอดคล้องกับจังหวะที่ได้รับ

ไม่เหมือนกับมอเตอร์กระแสตรง ที่จะหมุนไปเรื่อยๆ เราสามารถควบคุมการหมุนของ servo motor ไปยังมุมที่เราต้องการได้ ในช่วง 0 ถึง 180 องศา



0-180° rotation

Servo Motor VS DC Motor



360° rotation

APPLICATION CHALLENGE

โปรแกรม EDU:BIT เพื่อทำหน้าที่เป็นเครื่องให้จังหวะ (metronome) เมื่อเปิดเครื่องตัวชี้จะแกว่ง(ติดกับแขนของservo motor) ไปทางซ้าย ทางขวาไปเรื่อยๆในจังหวะที่คงที่ Potentio Bit จะเป็นตัวควบคุมจังหวะ เมื่อกดปุ่มสีเหลือง (ปุ่ม A)จะแสดงจังหวะปัจจุบัน (เช่น 120 bpm)

พี่ๆมีเคล็ดลับมาฝากกก

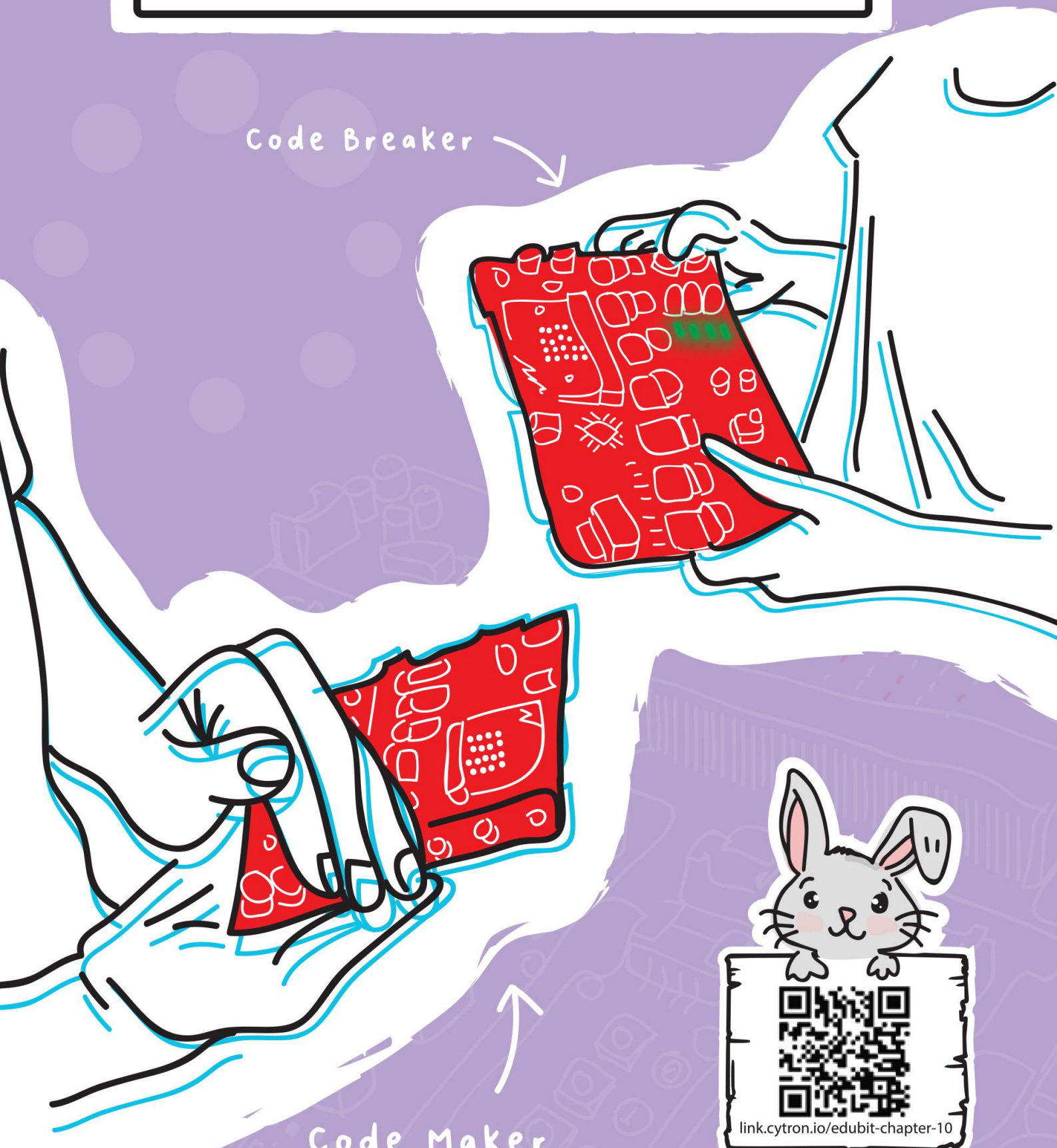
เคล็ดลับ#1 น้องๆต้องสร้างตัวแปร2ตัวแปร คือ Tempo และ Delay
เคล็ดลับ#2 ในทั่วไปเครื่องให้จังหวะจะมีช่วงอยู่ที่ 40 ถึง 200 bpm
เคล็ดลับ#3 จังหวะ 60 bpm (หรือ ครั้งที่แกว่งต่อ 1 นาที) หมายถึง ตัวชี้จะแกว่งจากจุดหนึ่งไปถึงอีกจุดหนึ่ง 60 ครั้ง ต่อ 1 นาที นั่นคือ ตั้งชี้จะแกว่ง 1 ครั้ง ต่อ 1 วินาที
เคล็ดลับ#4 ยิ่งมีจังหวะเร็วเท่าไร ยิ่งมีความล่าช้าน้อย

เครื่องให้จังหวะ คือ เครื่องที่สร้างเสียงคลิก หรือเสียงอื่นๆ ในปกติผู้ใช้สามารถกำหนดจังหวะต่อนาที (bpm) ได้ นักดนตรีใช้เครื่องนี้ฝึกซ้อมเพื่อให้ทราบจังหวะที่คงที่ เครื่องให้จังหวะโดยทั่วไปจะใช้การเคลื่อนไหวแบบ Synchronized

-วิกิพีเดีย-

บทที่ 10

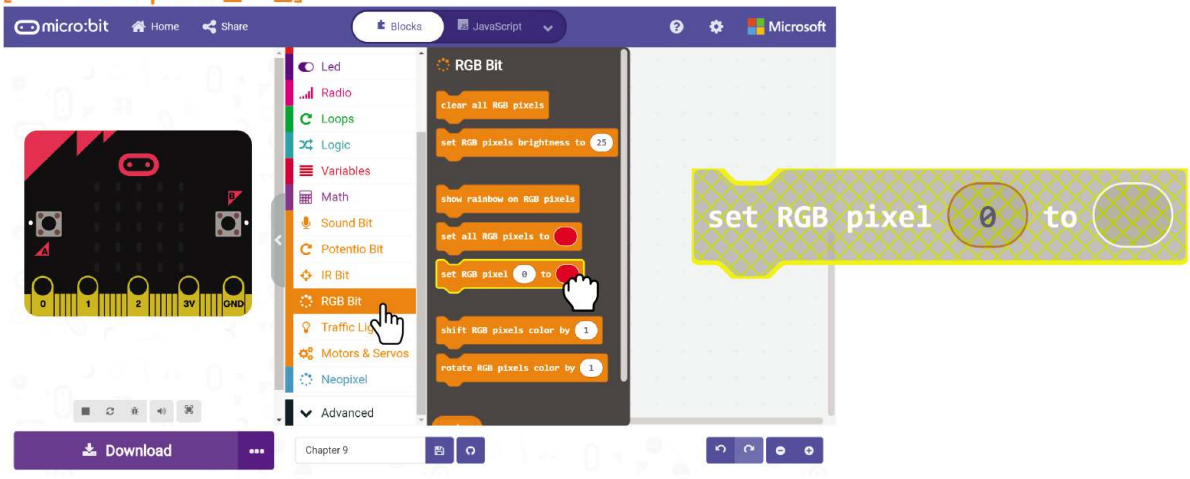
เกมทายใจ, คุณแก้ปริศนาได้รึเปล่า (RGB Bit)



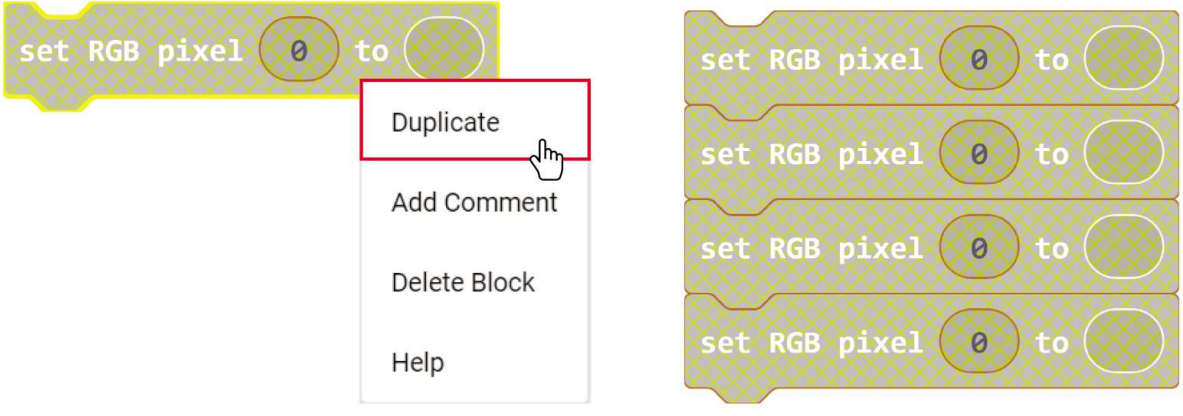
link.cytron.io/edubit-chapter-10

เริ่มเขียนโปรแกรม

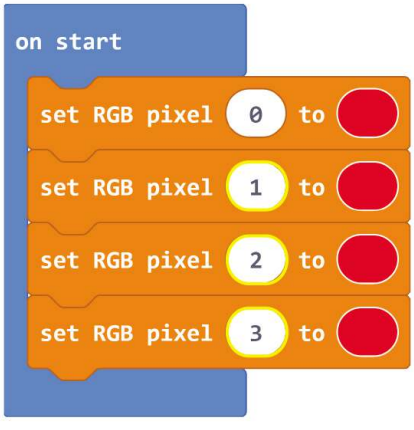
ขั้นตอนที่ 1 สร้างโปรเจคใหม่ใน MakeCode Editor จากนั้นเพิ่มส่วนเสริม EDU:BIT (สามารถกลับไปดูได้ที่หน้า 40) คลิกที่หมวดหมู่ [RGB Bit] จากนั้นเลือกบล็อกคำสั่ง [set RGB pixel_to_]

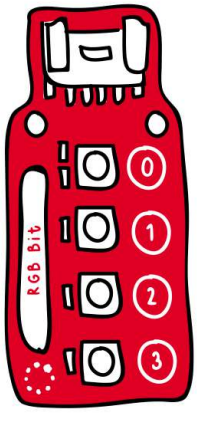


ขั้นตอนที่ 2 ใน Workspace คลิกขวาที่บล็อกคำสั่ง [set RGB pixel_to_] จากนั้นคลิก Duplicate ทำซ้ำจนกระทั่งมีบล็อกคำสั่ง [set RGB pixel_to_] ทั้งหมดสี่บล็อก



ขั้นตอนที่ 3 วางบล็อกคำสั่งลงในบล็อก [on start] เปลี่ยนเลข RGB pixel จาก 0 เป็น 1, 2, 3 ในบล็อกที่สอง สาม และสี่

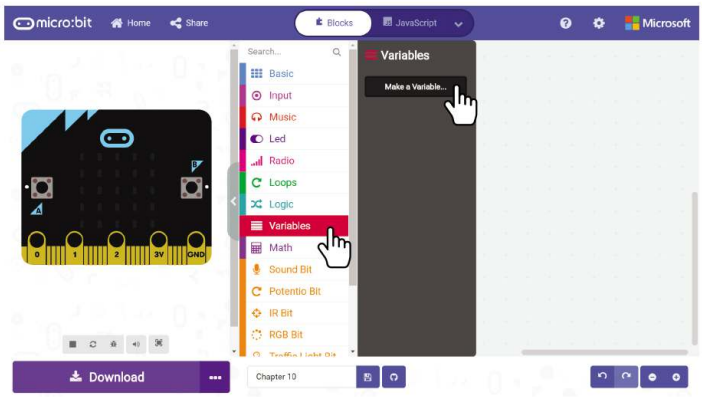




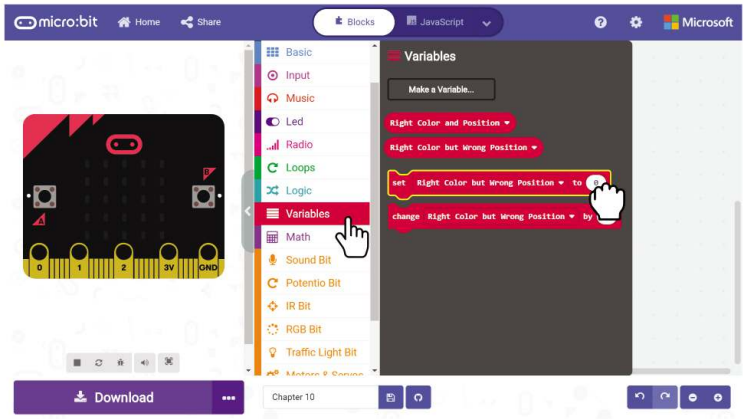
บน RGB Bit มี RGB LEDs 4 ตัว โดยแต่ละตัวถูกกำหนดไว้ด้วยตัวเลข (0-3) ใช้เลขนี้เพื่อเขียนโปรแกรมของ LED แต่ละตัว

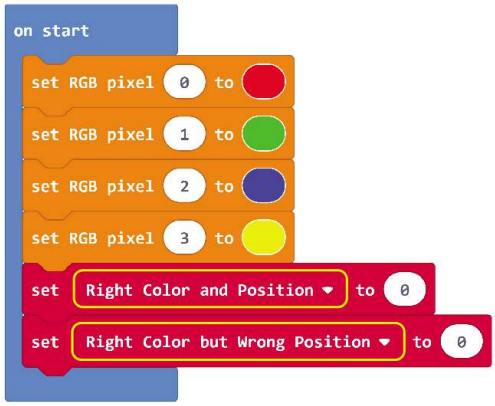
ขั้นตอนที่ 4 แฟลชโค้ดไปยัง EDU:BIT

ขั้นตอนที่ 5 เพิ่มตัวแปรใหม่สองตัว ตั้งชื่อว่า “Right Color and Position” และ “Right Color but Wrong Position”

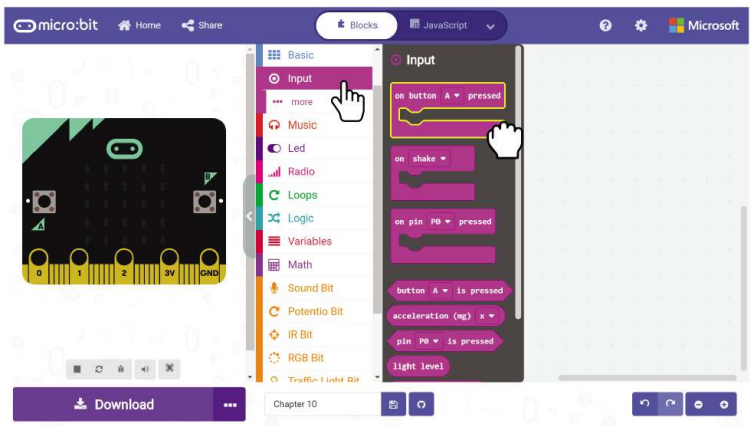


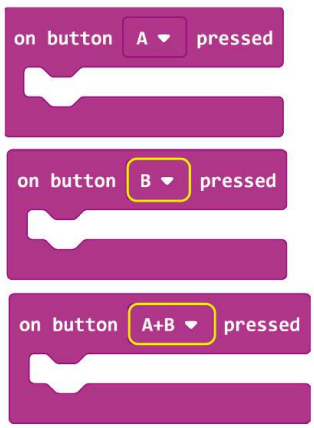
ขั้นตอนที่ 6 คลิกที่หมวดหมู่ [Variables] และเลือกบล็อกคำสั่ง [set_to_] คัดลอกบล็อกคำสั่งออกมาและลากไปยังบล็อก [on start] ตั้งค่าตัวแปรหนึ่งเป็น “Right Color and Position” และอีกตัวแปรเป็น “Right Color but Wrong Position”



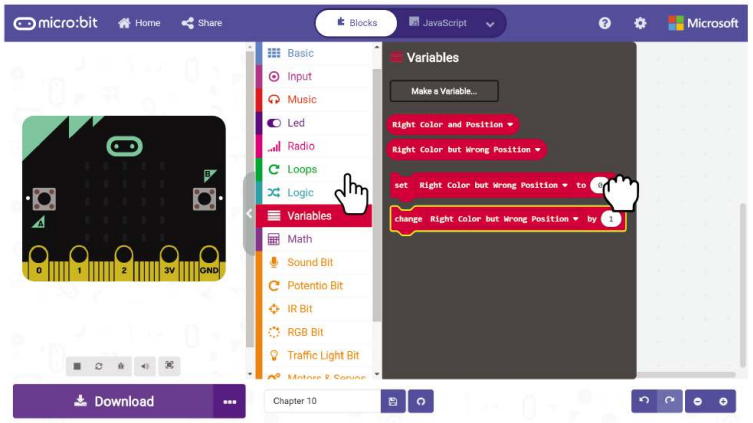


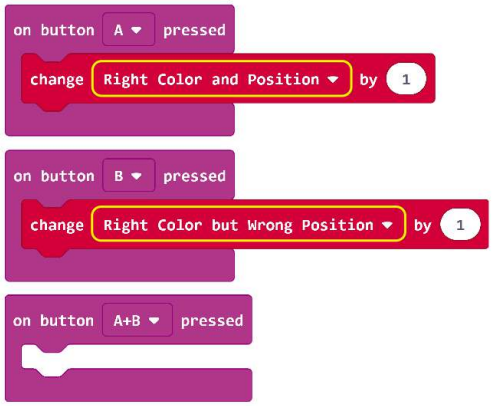
ขั้นตอนที่ 7 คลิกที่หมวดหมู่ [Input] และเลือกบล็อกคำสั่ง [on button_pressed] คัดลอกบล็อกคำสั่งออกมาเปลี่ยนชื่อบล็อกคำสั่งที่สองเป็น B และบล็อกคำสั่งที่สามเป็น A+B



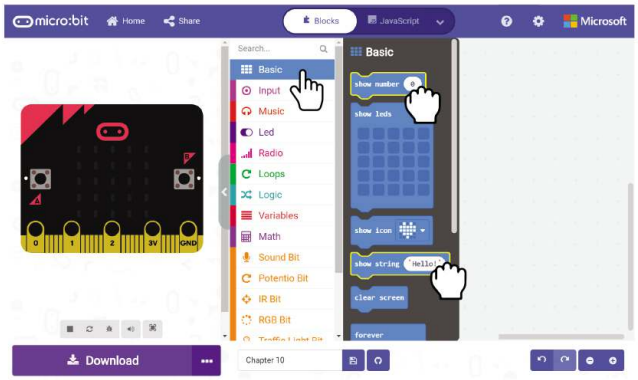


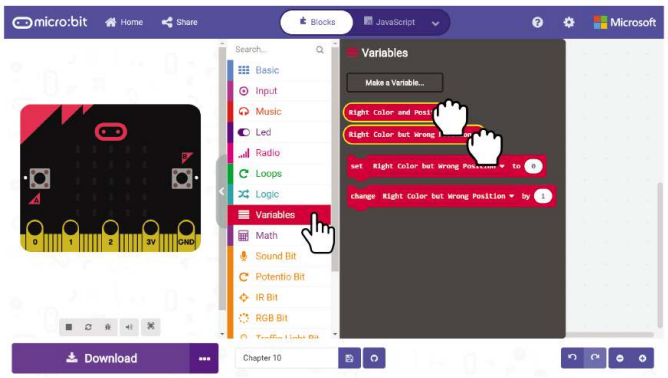
ขั้นตอนที่ 8 คลิกที่หมวดหมู่ [Variables] อีกครั้งและเลือกบล็อกคำสั่ง [change_by_] คัดลอกบล็อกออกมา และนำไปต่อกับบล็อกคำสั่ง [on button A pressed] และ [on button B pressed] ตั้งค่าตัวแปรหนึ่งเป็น “Right Color and Position” และอีกตัวแปรเป็น “Right Color but Wrong Position”



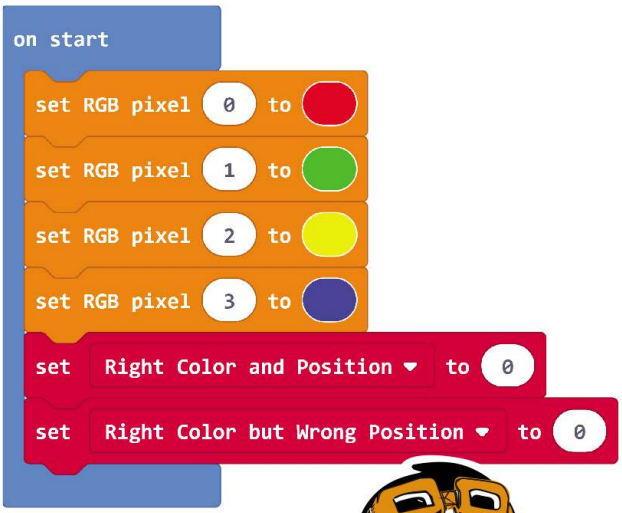


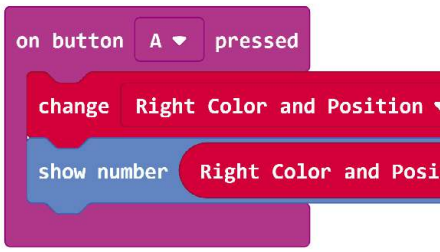
ขั้นตอนที่ 9 เพิ่มบล็อกคำสั่ง [show number] และ [show string] ในหมวดหมู่ [Basic] รวมถึงบล็อกคำสั่ง [Right Color and Position] และ [Right Color but Wrong Position] ในหมวดหมู่ [Variables]

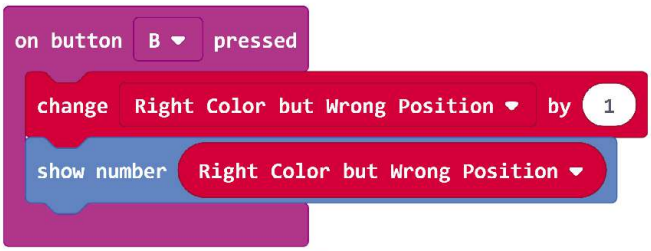


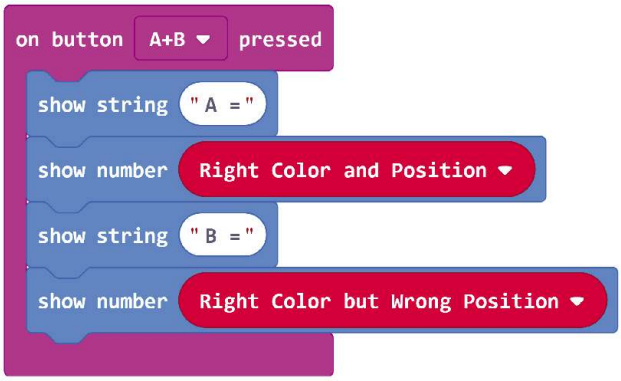


ขั้นตอนที่ 10 เปลี่ยน “Hello!” ในบล็อกคำสั่ง [show string] เป็น “A=” และ “B=” ตามลำดับ นี่คือโค้ดที่เสร็จเรียบร้อย





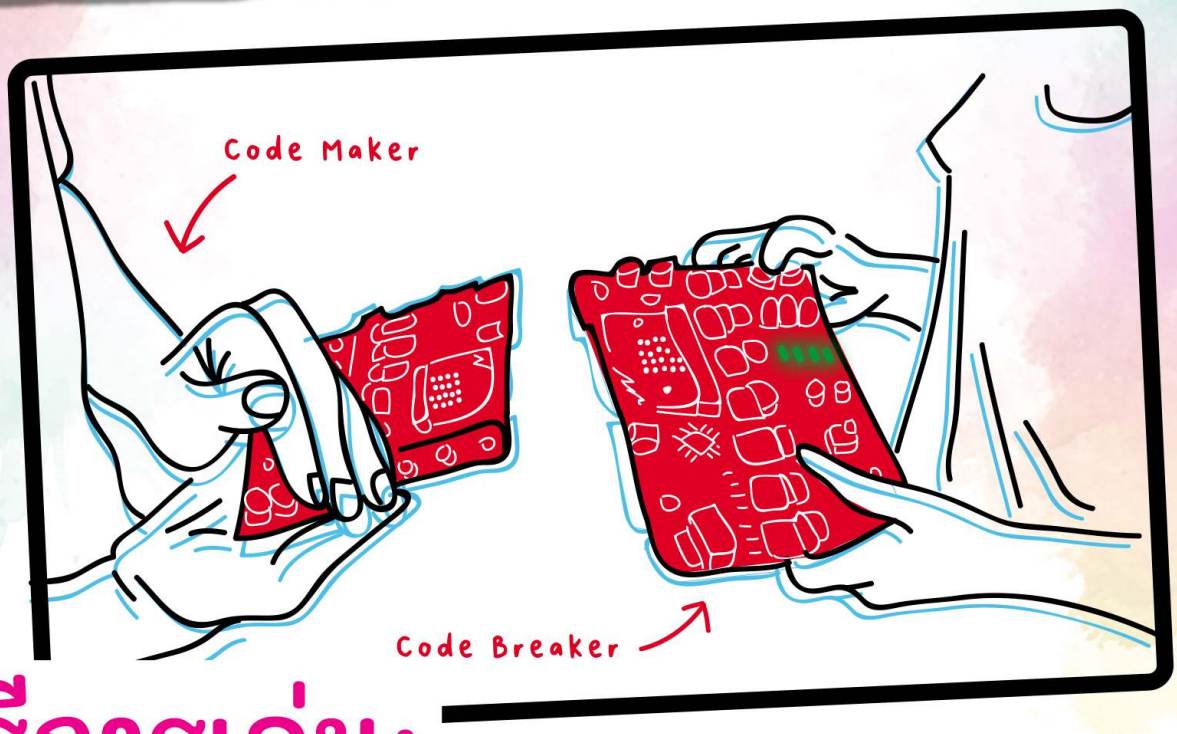




ขั้นตอนที่ 9 แฟลชโค้ดลงบอร์ด EDU:BIT ทีนี้ก็ถึงเวลาที่น้องๆจะได้เล่นแล้ว!

Let's Play

Mastermind, Can You Break The Secret Code?



วิธีการเล่น:

- ผู้เล่นที่ 1 Code Master จะต้องตั้งรหัสลับโดยการทำให้ไฟ LED ทั้งสี่ดวงสว่างขึ้นแบบสุ่มลำดับ กำหนดให้สีเริ่มต้นเป็นสีแดง และสีเขียว อย่าลืมปิดบอร์ดของคุณไว้เพื่อไม่ให้ผู้เล่นคนอื่นเห็นรูปแบบสี
- ผู้เล่นที่ 2 Code Breaker ต้องทายรหัสลับโดยที่ผู้เล่นที่ 2 จะต้องแสดงไฟ RGB LED บน EDU:BIT ของตัวเองให้ Code Maker ดู
- Code Maker เช็กความถูกต้องหลังจากนั้นจึงกดปุ่มสีเหลือง (ปุ่ม A) และ/หรือ ปุ่มสีน้ำเงิน (ปุ่ม B) บน EDU:BIT ของตนเอง เพื่อระบุจำนวน LED ที่แสดงได้ "ถูกสีและ ถูกตำแหน่ง" และ "ถูกสีแต่ผิดตำแหน่ง"
- Code Breaker สามารถกดปุ่มสีเหลืองและสีน้ำเงินพร้อมกันเพื่ออ่านค่าที่ถูกส่งกลับมา
- ทำขั้นตอนที่ 2 และ 3 จนกว่า Code Breaker จะทายลำดับสีได้ถูกต้อง (ทายได้ไม่เกิน 10 ครั้งต่อรอบการเล่น)
- เปลี่ยนบทบาทผู้เล่นเพื่อเล่นในรอบถัดไป

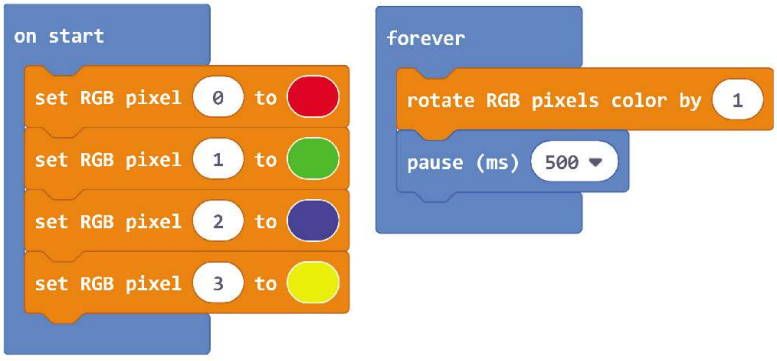
How to Win:

สามารถชนะได้ด้วยการ ทายลำดับสีถูกต้อง หากไม่สำเร็จชัยชนะจะกลายเป็นของ Code Maker

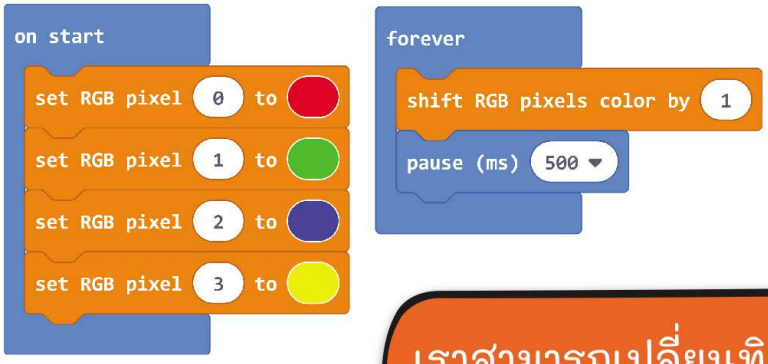
EXPLORE MORE BLOCKS

#1 สามารถตั้งค่า LED บน RGB Bit เพื่อที่จะแสดงผลสีของสายรุ้งโดยการแทนที่บล็อกคำสั่ง [set RGB pixel_to_] ด้วยบล็อกคำสั่ง [show rainbow on RGB pixels]

#2 สามารถสร้างเอฟเฟกต์ไฟวิ่งโดยการวางบล็อกคำสั่ง [rotate RGB pixels color by_] ลงในบล็อกคำสั่ง [forever] อย่าลืมเพิ่มบล็อกคำสั่ง [pause] เพื่อลดความเร็วของโปรแกรมสำหรับดูการทำงานของเอฟเฟกต์ นี่คือโค้ดตัวอย่าง



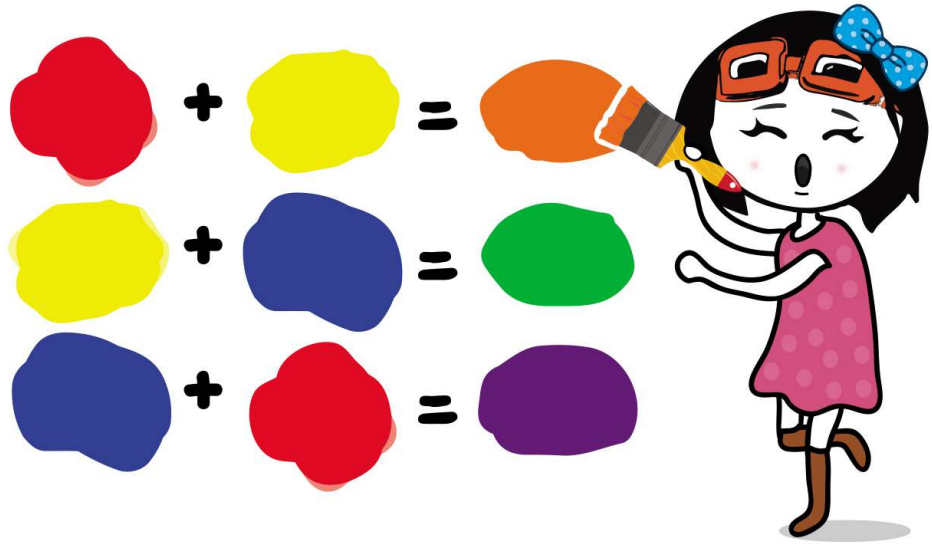
#3 นอกจากนี้ยังสามารถย้ายพิกเซลทีละตัวโดยใช้บล็อกคำสั่ง [shift RGB pixels color by_] ในบล็อก [forever] และเพิ่มบล็อกคำสั่ง [pause] เพื่อลดความเร็วของโปรแกรมสำหรับดูการทำงานของเอฟเฟกต์ ซึ่งพิกเซลจะดับลงทีละตัว นี่คือโค้ดตัวอย่าง



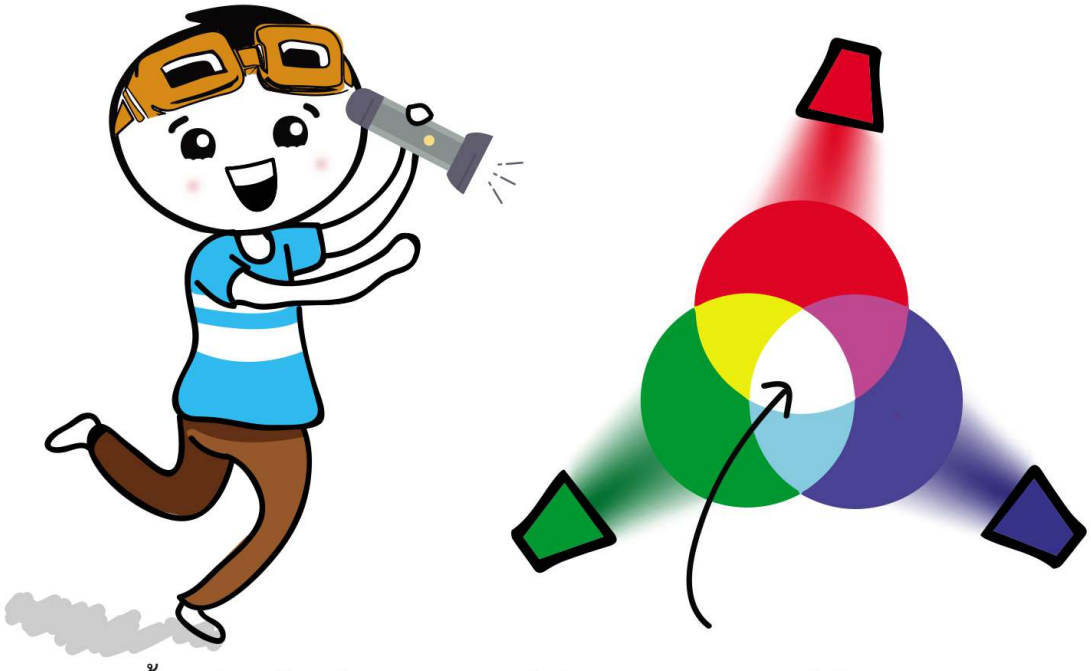
เราสามารถเปลี่ยนทิศทางของเอฟเฟกต์ #2 และ #3 ด้านบนโดยการเปลี่ยนค่าตัวแปรจากจำนวนบวกเป็นจำนวนติดลบ

FUN FACT!

ในวิชาศิลปะ เราอาจเคยเรียนเรื่องแม่สี 3 สี มีสีแดง เหลือง และน้ำเงิน และเมื่อผสมมันเข้าด้วยกันจะได้สีตามนี้



อย่างไรก็ตามอุปกณ์ที่ใช้ไฟสำหรับการแสดงผลค่าสี เช่น โทรทัศน์และหน้าจอคอมพิวเตอร์ จะแสดงผลโดย โมเดลสี RGB.



ในโมเดลนี้แม่สีจะเป็นสีแดง (Red) สีเขียว(Green) และสีน้ำเงิน(Blue) แทน เมื่อสีทั้งหมดรวมกันจะกลายเป็นสีขาว

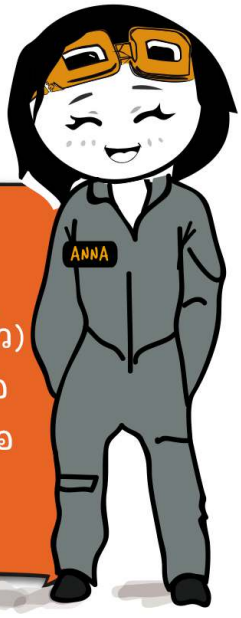
APPLICATION CHALLENGE

เขียนโปรแกรม EDU:BIT ให้กลายเป็นเครื่องมือสำหรับเกมทดสอบความจำ

มันเป็นอย่างไร

- เอียง EDU:BIT ไปทางซ้ายเพื่อเปิด LED บน RGB Bit ให้ขึ้นแบบสุ่มรูปแบบสีประมาณสองถึงสามวินาทีเพื่อเริ่ม
- สังเกตและพูดลำดับสีออกมาหลังจากไฟ LED ดับลง
- กดปุ่มสีน้ำเงิน (ปุ่ม B) เพื่อให้ไฟ LED ติดในรูปแบบเดิมอีกครั้ง เพื่อเช็กคำตอบ
- ถ้าหากตอบถูกต้องให้กดปุ่มสีเหลือง (ปุ่ม A) เพื่ออัพเดตและแสดงผลคะแนน เกมจะจบลงเมื่อตอบผิด
- สามารถปรับระดับความยากของเกมได้โดยการหมุน knob of Potentio Bit เพื่อเพิ่มหรือลดเวลาการสว่างของไฟ LED
- ผู้เล่นที่ได้คะแนนมากที่สุดจะเป็นผู้ชนะ

พี่ๆมีเคล็ดลับ
เคล็ดลับ #1 ต้องสร้างตัวแปรสองตัว - คะแนน และ รูปแบบสี
เคล็ดลับ #2 ต้องตั้งค่าลำดับสีล่วงหน้า (สีของ RGB LED แต่ละตัว) สำหรับแต่ละรูปแบบ ใช้สีที่มากขึ้นหรือตั้งค่ารูปแบบล่วงหน้าเพื่อทำให้เกมน่าลุ้นมากขึ้น และในทางกลับกันอาจจะต้องจำกัดสีหรือรูปแบบสำหรับผู้เล่นที่อายุน้อย

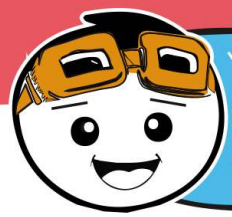


โบนัสพิเศษ

Simon Says with LEDs

การสื่อสารผ่านสัญญาณวิทยุ

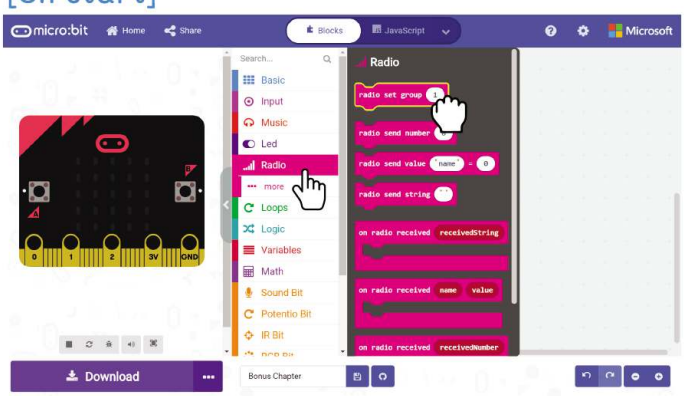
link.cytron.io/edubit-bonus-chapter

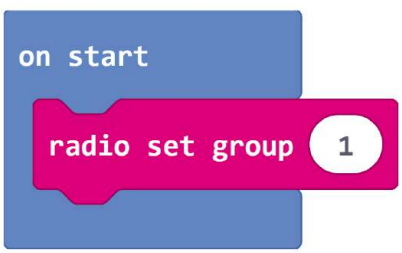


ไม่ว่าการสื่อสารชนิดไหน ก็ต้องมีองค์ประกอบอย่างน้อย 2 องค์ประกอบก็คือ ผู้รับ และ ผู้ส่ง ในเกมนี้ต้องใช้บอร์ด EDU:BITs 2 บอร์ด เพื่อทำการสื่อสารกันด้วยการรับและส่งสัญญาณผ่านการกระจายสัญญาณวิทยุ

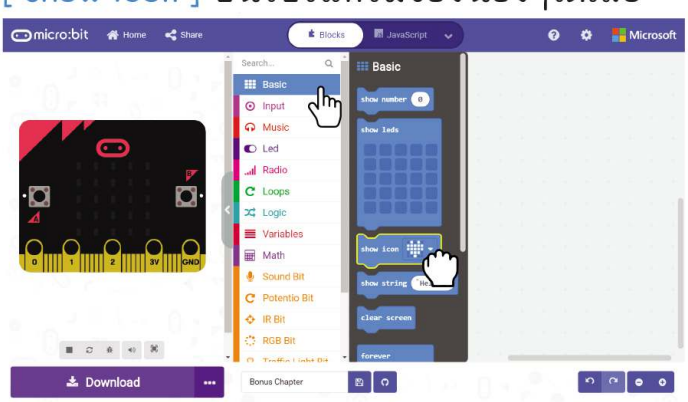
เริ่มเขียนโปรแกรม

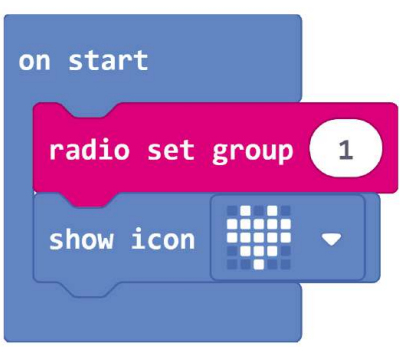
ขั้นตอนที่ 1 การสร้างโปรเจ็คใหม่ใน MakeCode Editor และเพิ่มส่วนเสริม EDU:BIT คลิกที่หมวดหมู่ [Radio] จากนั้นเลือก [radio set group] แล้วลากบล็อกไปยังบล็อกที่เขียนว่า [on start]



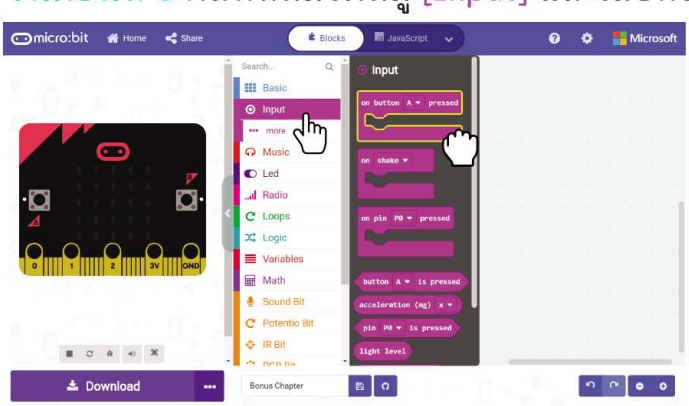


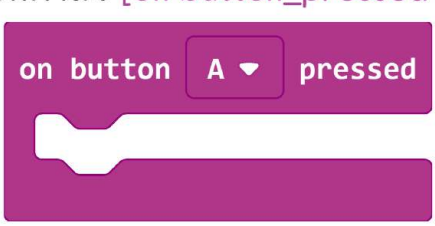
ขั้นตอนที่ 2 คลิกที่หมวดหมู่ [Basic] ใน MakeCode Editor จากนั้นคลิกที่บล็อก [show icon] บนโปรแกรมของน้องๆได้เลย



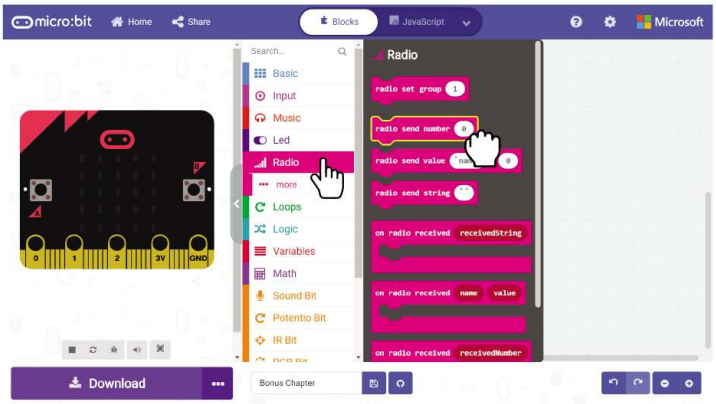


ขั้นตอนที่ 3 คลิกที่หมวดหมู่ [Input] และเลือกบล็อกคำสั่ง [on button_pressed]

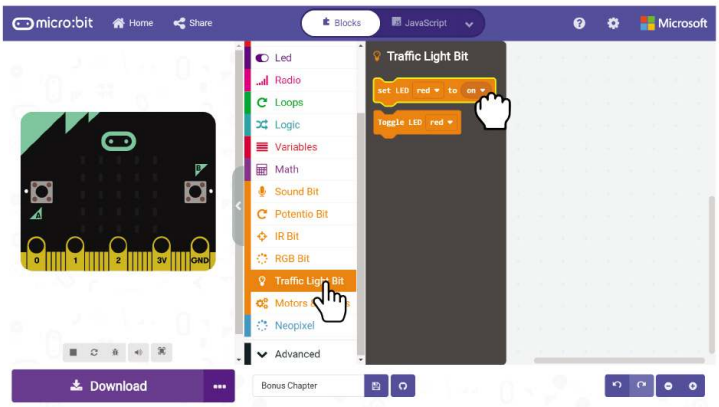


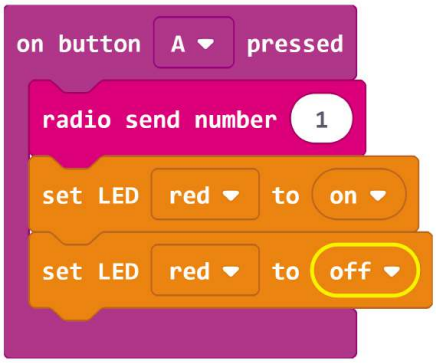


ขั้นตอนที่ 4 คลิกที่หมวดหมู่ [Radio] และเลือกบล็อกคำสั่ง [radio send number] จากนั้นเปลี่ยนค่าให้เป็น 1

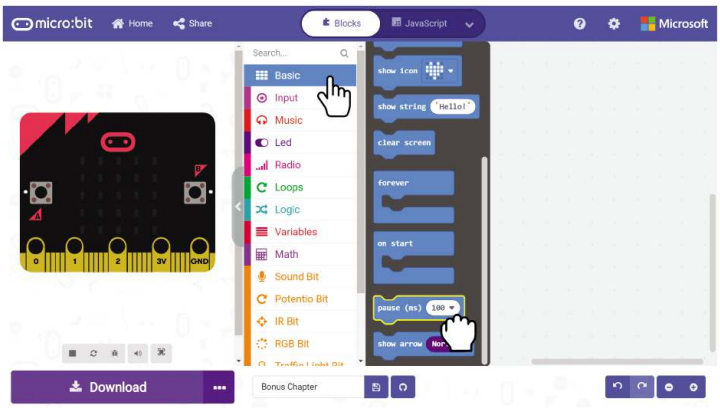


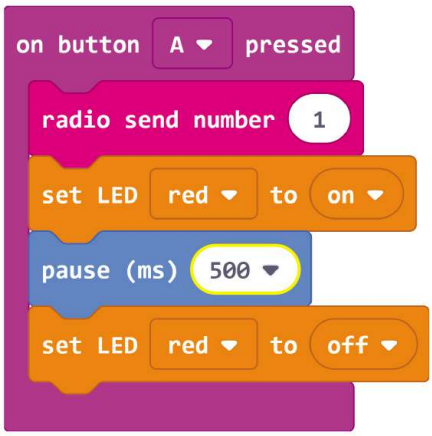
ขั้นตอนที่ 5 คลิกที่หมวด [Traffic Light Bit] และเลือกบล็อกคำว่า [set LED_to_] ทำซ้ำอีกรอบและลากทั้งสองบล็อกไปยังบล็อกที่เขียนว่า [on button A pressed] จากนั้นเปลี่ยนการตั้งค่าบล็อกที่สองเป็น “off”



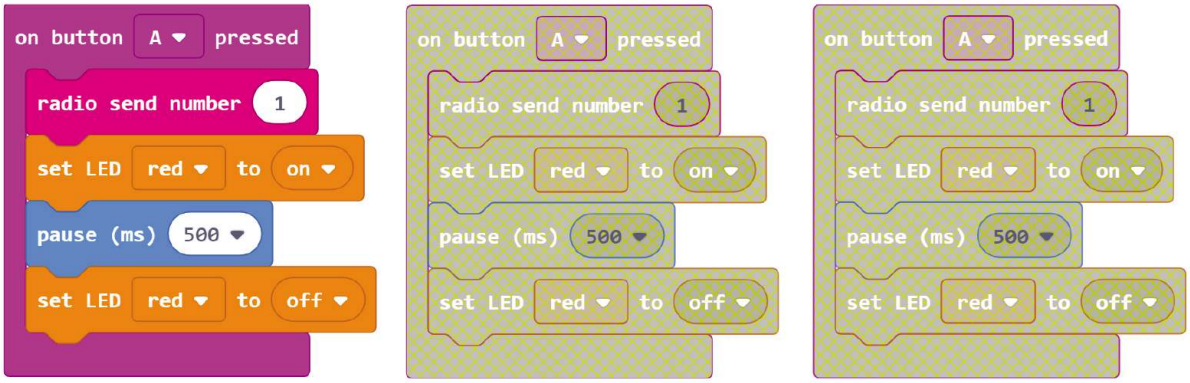


ขั้นตอนที่ 6 คลิกที่หมวด [Basic] และเลือก [pause(ms)_] จากนั้นลากบล็อกไปแทรกระหว่างช่องบล็อกที่เขียนว่า [set LED_to_] และเปลี่ยนค่าให้เป็น 500

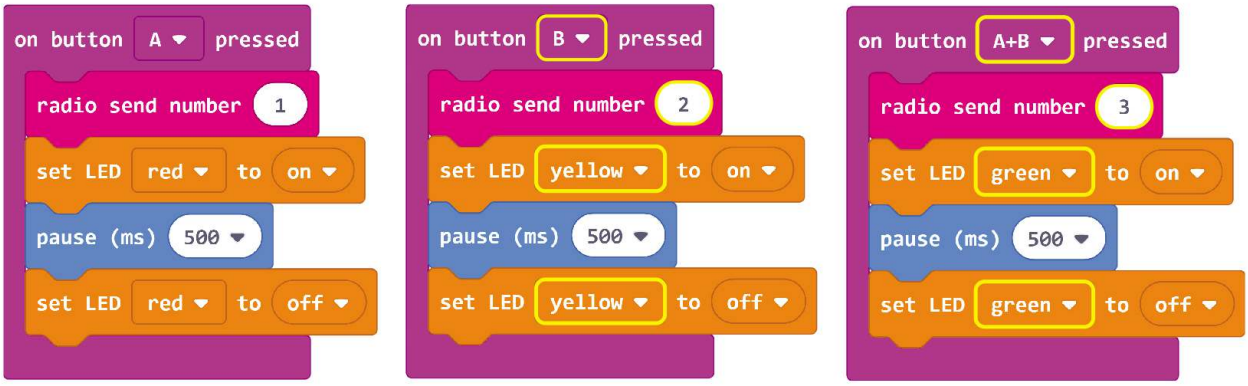




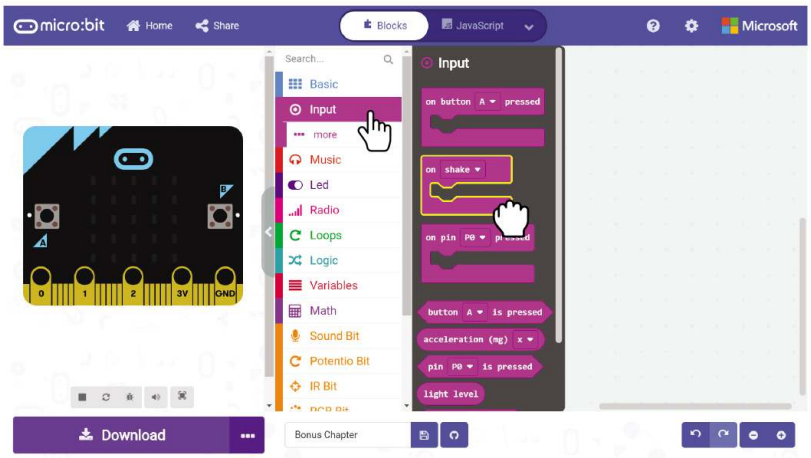
ขั้นตอนที่ 7 ให้คลิกขวาบนบล็อกที่เขียนว่า [on button A pressed] จากนั้นให้เลือก “Duplicate’การทำซ้ำจะทำให้ได้บล็อกเดียวกัน 3 ชุด

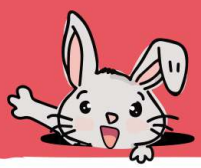


ขั้นตอนที่ 8 เปลี่ยนการตั้งค่าสำหรับปุ่ม,ค่าตัวเลขและสีของ LED ของบล็อกโค้ดที่ซ้ำกันดังรูปภาพที่แสดงข้างล่างต่อไปนี้:

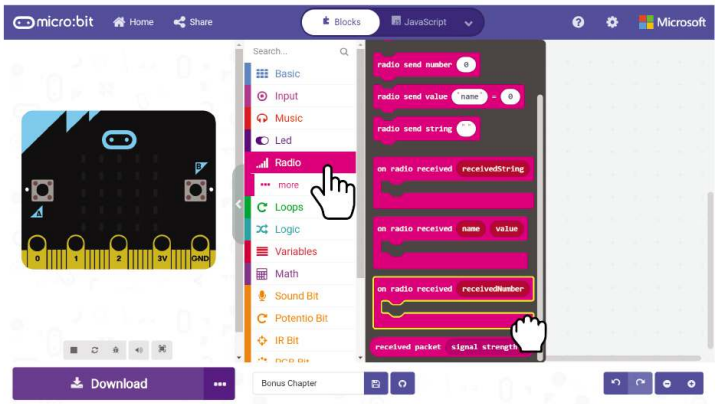


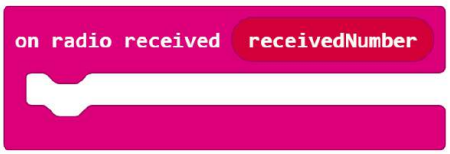
ขั้นตอนที่ 9 คลิกหมวด [Input] และเลือก [on] ที่บล็อก จากนั้นเปลี่ยนการตั้งค่าเป็น "tilt left" และให้คลิกที่หมวด [Radio] และเลือกบล็อกที่เขียนว่า [radio send number] จากนั้นเปลี่ยนค่าเป็น 4



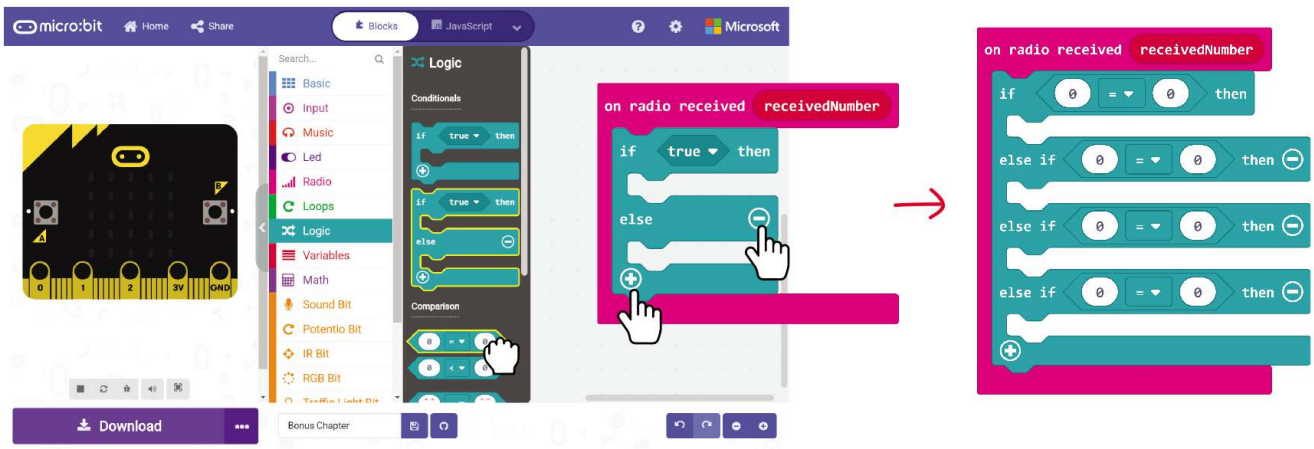


ขั้นตอนที่ 10 คลิกหมวด [Radio] และเลือกบล็อก [on radio received receivedNumber]

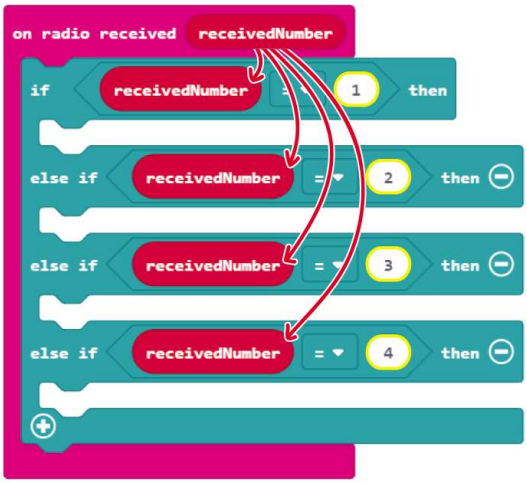




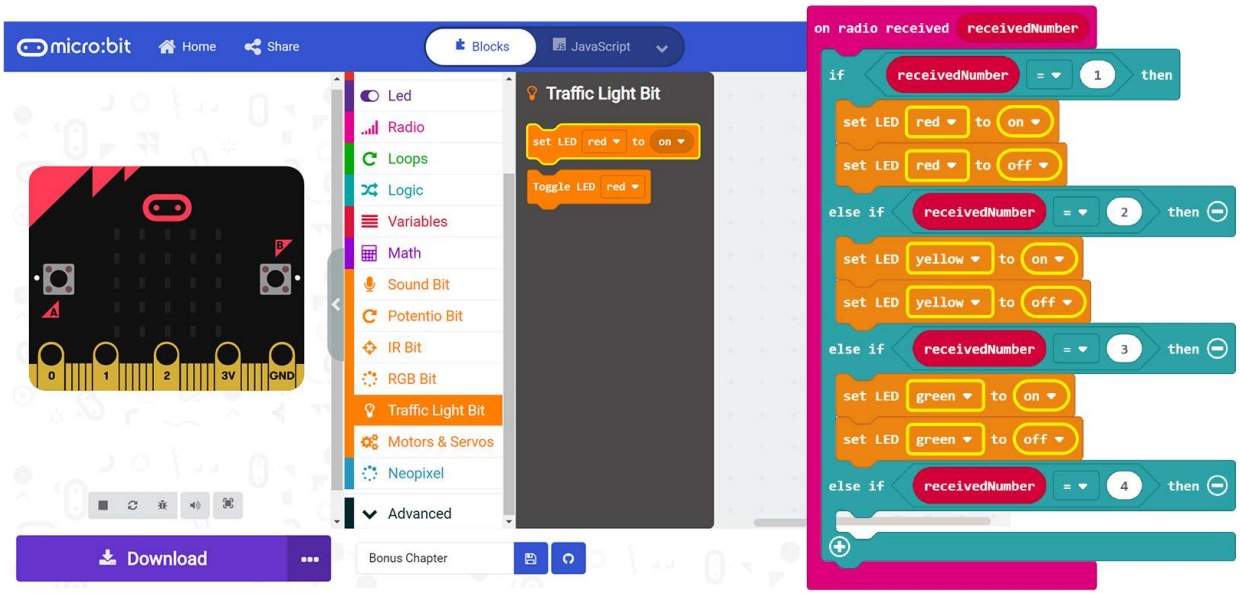
ขั้นตอนที่ 11 คลิกหมวด [Logic] และเลือกบล็อก [if-then-else] คลิกปุ่ม + เพื่อเพิ่มเงื่อนไขอื่น ๆ อีกสามเงื่อนไข[else if] และคลิกปุ่ม - เพื่อลบเงื่อนไข [else] แนบ [Logic]: [__ = _] ลงในบล็อกของการเปรียบเทียบของแต่ละช่อง "if"และ "else if" ในช่องที่กำหนด



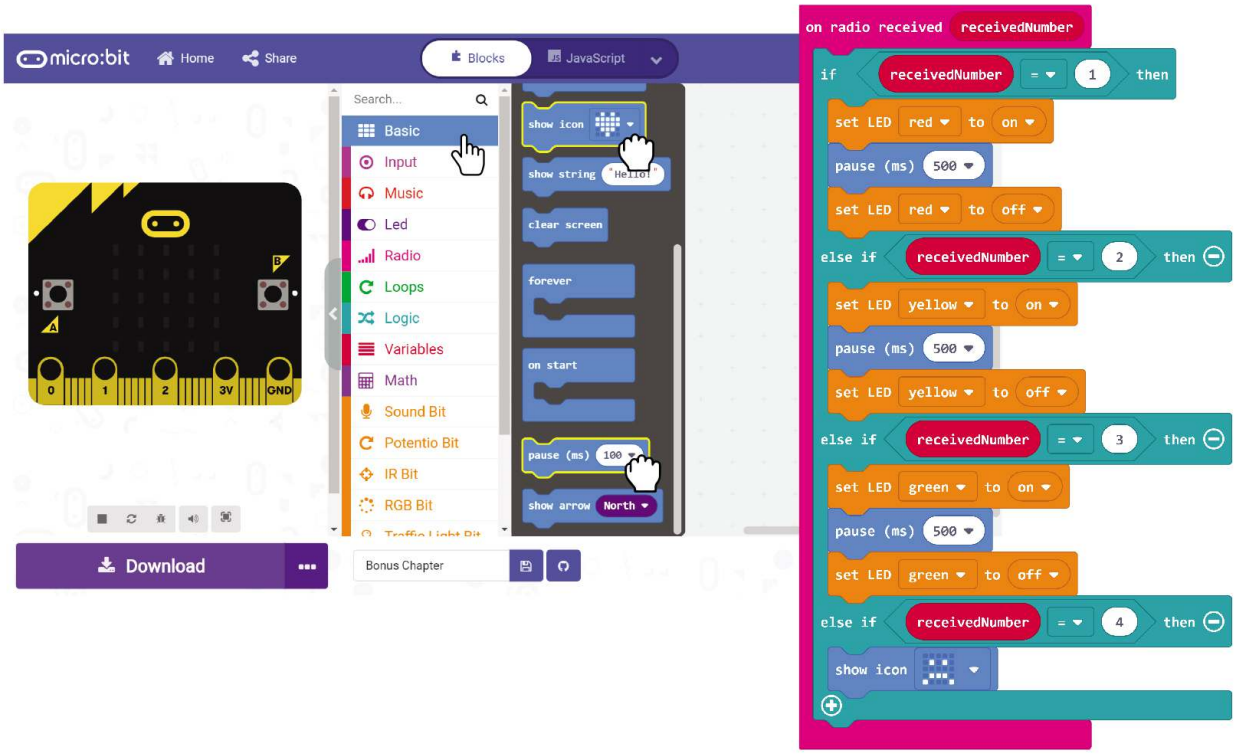
ขั้นตอนที่ 12 คลิกและลากตัวแปร 'receivedNumber' ลงในบล็อกเปรียบเทียบดังที่แสดงไว้ และเปลี่ยนค่าเป็น 1, 2, 3 และ 4 ตามลำดับ



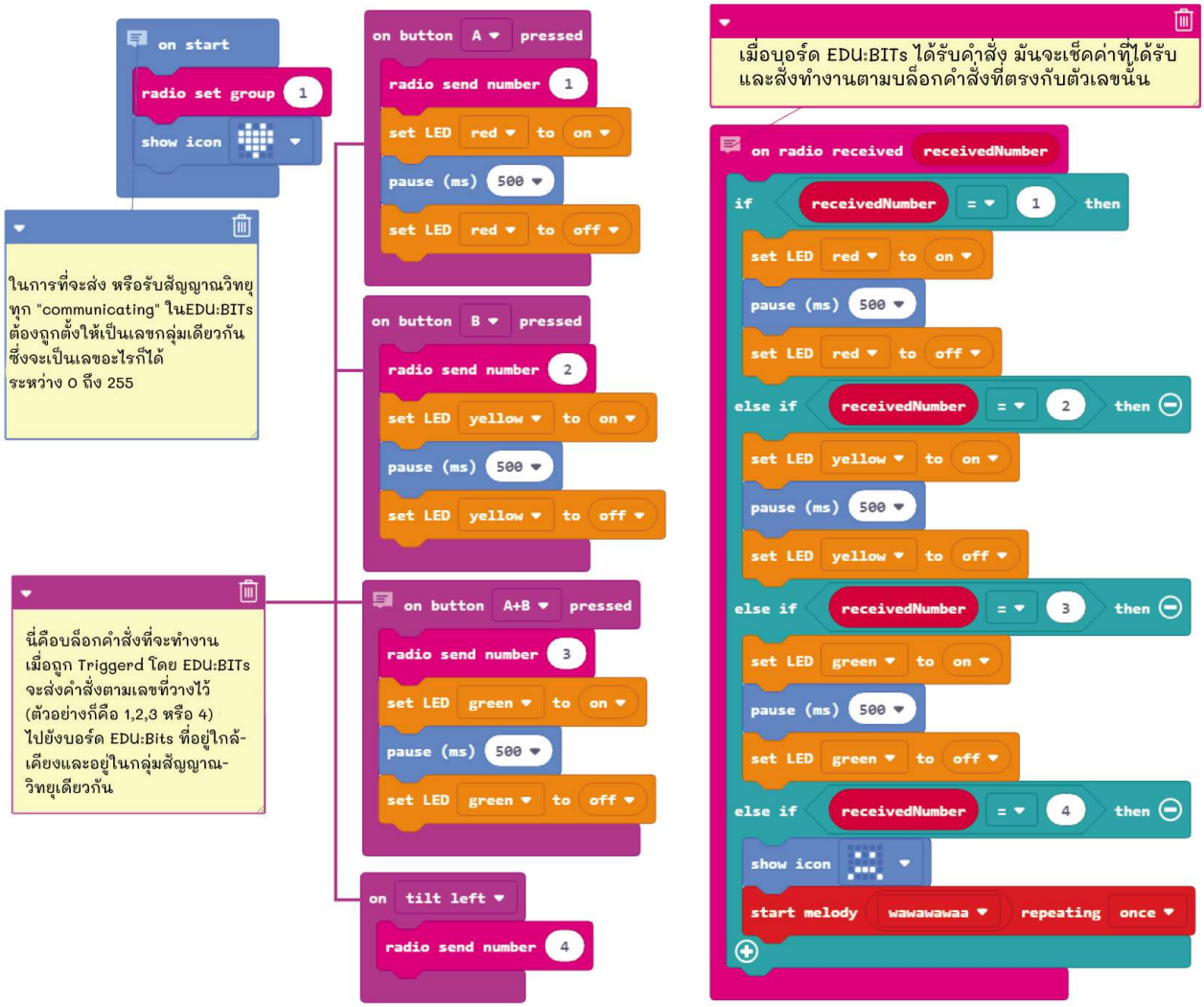
ขั้นตอนที่ 13 คลิกหมวด [Traffic Light Bit] และเลือก [set LED to_ block] ให้ทำบล็อกซ้ำกันและลากให้เข้ากับสามช่องแรกจากนั้นเปลี่ยนสีและการตั้งค่าเปิด / ปิดดังที่แสดงตัวอย่างไว้



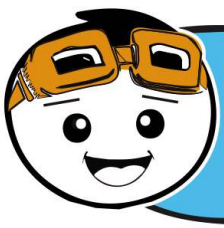
ขั้นตอนที่ 14 คลิกหมวด [Basic] และเลือก [pause(ms)_] จากนั้นทำซ้ำอีกรอบและลากบล็อกแทรกไปยังระหว่างบล็อกที่เขียนว่า [set LED_to_] และเปลี่ยนค่าเป็น 500 จากนั้นเพิ่มบล็อก [Basic]: [show icon] ในช่อง "else if" และเปลี่ยนไอคอนเป็น "sad face"



ขั้นตอนที่ 15 คลิกหมวด [Music] และเพิ่ม [start melody_repeating_] เพื่อกรอกรหัสของน้องๆ จากนั้นเปลี่ยนทำนองเป็น "wawawawaa" (หรือทำนองใดก็ได้ที่น้องๆชอบ)



ขั้นตอนที่ 16 แฟลชโค้ดที่เสร็จเสร็จแล้วไปยังบอร์ด EDU: BIT ของน้องๆแบบเดียวกับเพื่อนของน้องๆได้เลย



เมื่อบอร์ด EDUBITS ของทั้งสองเปิดใช้งานน้องๆสามารถส่งสัญญาณวิทยุเพื่อเปิดไฟ LED บนบอร์ด EDUBIT ของเพื่อนน้องๆได้ และในทางกลับกันเพื่อนของน้องๆยังสามารถกดปุ่มบนบอร์ด EDUBIT ของเพื่อนๆเพื่อเปิดไฟ LED บนบอร์ดของน้องๆ

Let's Play

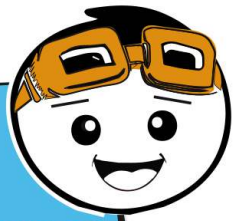
Simon Says with LEDs

หลังจากที่น้องๆได้แฟลชโค้ดไปที่บอร์ด EDU: BIT ของทั้งสองแล้วน้องๆสามารถท้ากับเพื่อนของน้องๆให้เล่นเกมของไซม่อนพูดว่า ในเวอร์ชันตอบโต้ไปมาได้

วิธีการเล่น:

ผู้เล่นทั้งสองฝ่ายต้องผลัดกันเป็น "ไซม่อน" กดปุ่มเพื่อให้สีแดง, สีเหลือง และสีเขียวของ LEDs สว่างขึ้นเมื่อถึงตาของน้องๆ

- ในการเริ่มเกมผู้เล่น 1 จะเปิดไฟ LED หนึ่งดวงบน Traffic Light Bit
- ให้ผู้เล่น 2 สังเกตการ จากนั้นไฟของ LEDเดียวกันจะสว่างตามไฟ LED อีกดวง
- เกมยังคงดำเนินต่อไปโดยการที่ผู้เล่นแต่ละคนต้องผลัดกันทำซ้ำตามลำดับล่าสุดจากนั้นจึงเปิดไฟ LED อีกดวงเพื่อเพิ่มในลำดับ
- หากผู้เล่นคนใดไฟของ LED สว่างผิด (หรือผิดลำดับ) ฝ่ายตรงข้ามจะเอียง EDU: BIT ของตัวเองเพื่อจบเกม
- ผู้แพ้จะต้องรีเซ็ต EDU: BIT ของตัวเองเพื่อเริ่มเกมใหม่

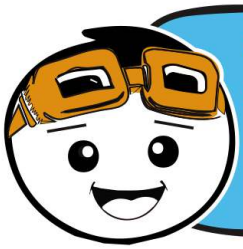


เกมนี้เริ่มต้นง่าย แต่จะยาวขึ้นเรื่อย ๆ และย้ายที่ซับซ้อนหลังจากแต่ละครั้งที่น้องๆเล่น เพื่อที่จะชนะน้องๆต้องสังเกตลำดับอย่างรอบคอบ และนี้เป็นเกมที่สนุกมาปในการฝึกพลังความจำของตัวน้องๆเอง

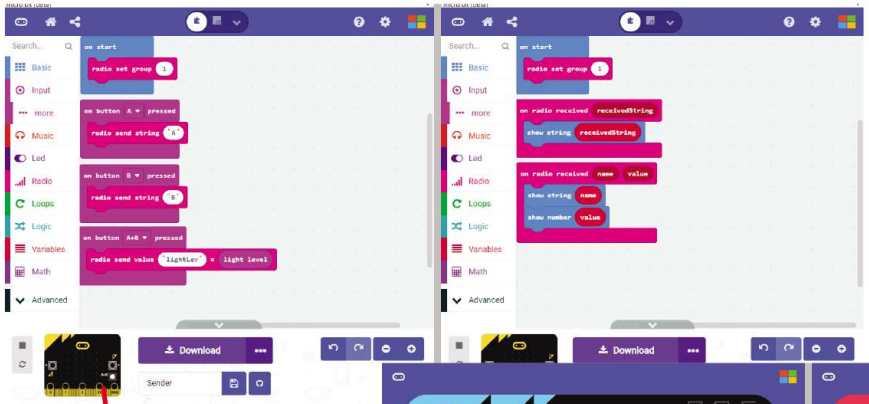
EXPLORE MORE BLOCKS

1 นอกจากการสู่งตัวแล้วน้องๆยังสามารถส่งข้อความโดยใช้บล็อก [radio send string "_"] จากนั้นน้องๆจะต้องใช้บล็อก [on radio received receivedString] เพื่อรับสัญญาณออกอากาศแบบสตริง สตริงสูงสุดจะมีความยาว 19 ตัวอักษร

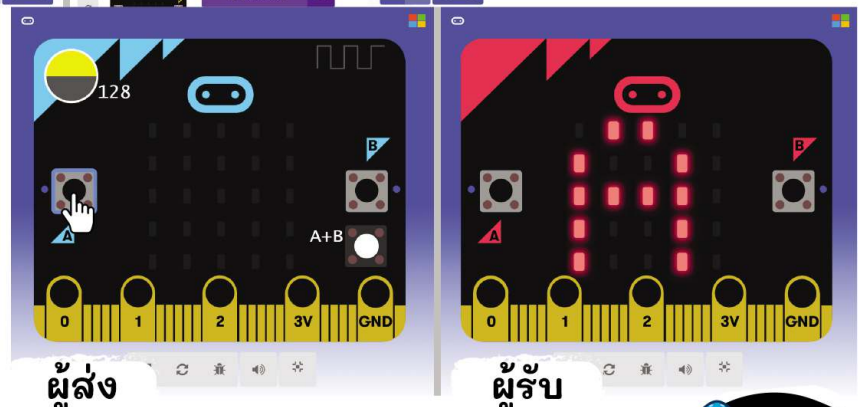
2 ใช้บล็อก [radio send value "_" = _] และบล็อก [on radio received name value] หากน้องๆต้องการส่ง และ / หรือรับข้อความและตัวเลขร่วมกัน ความยาวสตริงสูงสุดคือ 8 อักษร



หากน้องๆมีปัญหาไม่สามารถเข้าถึงบอร์ด EDU:BITs ในหลายๆเครื่อง น้องๆสามารถทดสอบฟังก์ชั่นการสื่อสารของวิทยุได้เพียงไปที่ makecode.com/multi เพื่อเขียนโค้ดของน้องๆสำหรับให้ "ผู้ส่ง" และ "ผู้รับ" น้องๆสามารถดูผลลัพธ์บนหน้าแบบจำลองได้เลย



คลิกที่บอร์ด micro:bit
เพื่อที่จะเปิดแบบเต็มหน้าจอ



ผู้ส่ง ผู้รับ

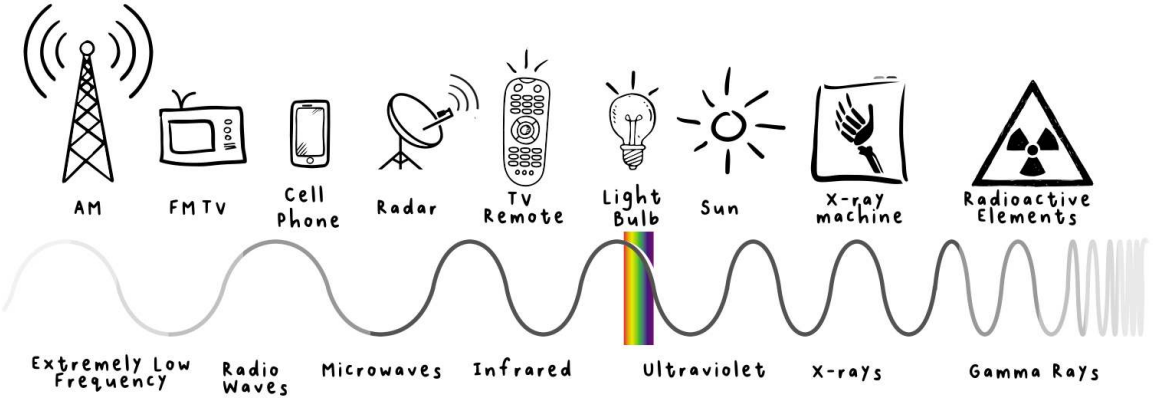
เมื่อน้องๆลองกดปุ่ม A ของ “ผู้ส่ง”(seder) บน micro: bit จากนั้นตัวผู้รับบน micro: bit จะรับสัญญาณวิทยุและแสดงสตริงที่ได้รับ เช่น A. คำถามคือจะเกิดอะไรขึ้นถ้าน้องๆกดปุ่ม A + B?

FUN FACT!

หากน้องๆเปิดบอร์ดEDU: BIT ของตัวเอง
จากล่างขึ้นบน น้องๆจะเห็นเสาร์อากาศ
ของวิทยุและบลูทูธในตัวได้ที่นี่

เสาร์อากาศส่งสัญญาณในรูปแบบของคลื่นวิทยุแม่เหล็กไฟฟ้าซึ่งมักใช้
สำหรับโทรทัศน์และวิทยุในการกระจายเสียงรวมทั้งการส่งสัญญาณดาว
เทียมอีกด้วย

สเปกตรัมของแม่เหล็กไฟฟ้า



NOTE!

เพื่อให้บอร์ด EDU: BIT ของน้องๆ ส่งและรับสัญญาณวิทยุจากบอร์ด EDU: BITS อื่น ๆ น้องๆ
จะต้องตั้งค่าทั้งหมดเป็นกลุ่มวิทยุเดียวกัน

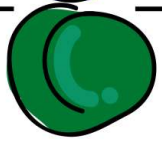
APPLICATION CHALLENGE

ตั้งค่าระบบการสื่อสารเครือข่ายสำหรับชั้นเรียนของคุณ

มันทำงานอย่างไร?

ตั้งค่าบอร์ด EDU: BIT ทุกรายการในชั้นเรียนเป็นกลุ่มวิทยุเดียวกัน

บอร์ด EDU: BIT ของคุณครูจะถูกตั้งค่าให้เลื่อนข้อความเมื่อได้รับสัญญาณ "string" และไฟ LED บน Traffic Light Bit จะสว่างขึ้นเมื่อได้รับสัญญาณวิทยุจาก"number" โดย ...

Number Received	Light up LED	What it means?
1	แดง 	A/ไม่/ผิด
2	เหลือง 	B/อาจจะ/ไม่แน่ใจ
3	เขียว 	C/ใช่/ถูกต้อง

บอร์ดEDU: BITs ของนักเรียนจะถูกตั้งค่าให้ส่งสตริงที่มีชื่อของนักเรียนจากนั้นส่งตัวเลข (1 หรือ 2 หรือ 3) เพื่อทำให้ไฟ LED บนบอร์ด EDU: BIT ของคุณครูสว่างขึ้นเมื่อได้รับการถูกกระตุ้น

กดปุ่ม A เพื่อส่งหมายเลข 1
กดปุ่ม B เพื่อส่งหมายเลข 2
กดปุ่ม A + B เพื่อส่งหมายเลข 3

และนี่คือเคล็ดลับสำหรับคุณครูคือการกำหนดชื่อเล่นสั้นๆ (มีเพียงสองหรือสามตัวอักษร) ให้นักเรียนแต่ละคนเพื่อลดเวลาในการเลื่อนข้อความ

สิ่งที่ฉันได้เรียนรู้

- ☐ โปรแกรม EDU: BIT จะแสดงข้อความและภาพเคลื่อนไหวบนเมทริกซ์ LED
- ☐ ดาวน์โหลด, บันทึก, เผยแพร่และแก้ไขไฟล์ที่ MakeCode.hex

- ☐ ใช้บล็อก input สำหรับการเขียนโปรแกรมตามเหตุการณ์
- ☐ สร้างและใช้ตัวแปร

- ☐ ใช้ Piezo buzzer บน Music Bit เพื่อเล่นทำนองง่ายๆ
- ☐ สร้างและใช้ฟังก์ชัน
- ☐ อ่านโน้ตเพลง

- ☐ โปรแกรม EDU: BIT เพื่อควบคุม LEDs บน Traffic Light Bit เปิดปิดและสลับ
- ☐ เพิ่มส่วนขยายได้ใน MakeCode Editor

- ☐ โปรแกรม EDU: BIT เพื่อตรวจจับวัตถุโดยใช้ IR Bit
- ☐ ใช้ในขณะที่ลูป
- ☐ สร้างและใช้อาร์เรย์

- ☐ โปรแกรม EDU: BIT เพื่ออ่านค่าอนาล็อกจาก Potentio Bit
- ☐ การอ่านอินพุตแบบอนาล็อกของแผนที่
- ☐ ใช้ลอจิกบล็อกสำหรับการเขียนโปรแกรมตามเงื่อนไข

- ☐ โปรแกรม EDU: BIT ตรวจจับเสียงรบกวนด้วย Sound Bit
- ☐ ใช้เหตุการณ์ทริกเกอร์เพื่อสลับระหว่างโหมดต่างๆ

- ☐ โปรแกรม EDU: BIT ควบ DC คุมมอเตอร์ทิศทางการปั่น และควบคุมความเร็ว
- ☐ ใช้บล็อกคณิตศาสตร์เพื่อคำนวนการทางคณิตศาสตร์

- ☐ โปรแกรม EDU: BIT ควบคุมเซอร์โวมอเตอร์ - ตำแหน่งเชิงมุม

- ☐ โปรแกรม EDU: BIT เปิดไฟ LED RGB บน RGB Bit ในรูปแบบต่างๆ สี / รูปแบบ

- ☐ โปรแกรม EDU: BIT ส่งและรับสัญญาณวิทยุ

*ติ๊กที่ช่องทำเครื่องหมายหากน้องๆเชี่ยวชาญทักษณะแล้ว จากนั้นให้กลับไปที่บทที่เกี่ยวข้องเพื่อปรับปรุงใหม่

NOTE FROM RERO EDUTEAM @ CYTRON

CONGRATULATIONS!!!

น้องๆได้ทำมันในทุกบทแล้วและเรียนรู้การเขียนโค้ดด้วย MakeCode Editor พี่ทั้งสองหวังว่าน้องๆ จะสนุกกับการเล่นเกมสุดฮิตในวัยเด็กของพวกพี่ และ ยกนิ้วให้กับการพัฒนาแอปพลิเคชันที่มีประโยชน์สำหรับห้องเรียนของน้องๆ

ตอนนี้น้องๆควรเข้าใจในสิ่งที่น้องๆสามารถทำได้บน micro: bit และ Bits ทั้งหมดนี่ที่จะมาพร้อมกับความพิเศษบนบอร์ด EDU: BIT ของน้องๆ นี่ๆ...น้องๆรู้ไหมว่าน้องสามารถแยกแต่ละ Bits ออกจากกันได้?

เอาเลยไป "ทำลาย" พวกเขาถ้าคุณต้องการ เมื่อหักจากบอร์ดหลักแล้วน้องๆสามารถทำได้โดยสร้างโปรเจ็คใหม่ด้วย Bits ต่างๆ และน้องๆจะต้องเชื่อมต่อโดยการใช้สายเคเบิลของ plug and play

และตอนนี้ก็ถึงเวลาที่น้องๆจะต้องใช้ความคิดและระดมความคิดสำหรับเกมและแอปพลิเคชันใหม่ ๆ พวกพี่สองคนรอแทบอดใจรอไม่ไหวที่จะได้เห็นโปรเจ็กต์สุดเจ๋งที่น้องๆกำลังจะทำขึ้น

อย่าลืมแบ่งปันผลงานของน้องๆกับพวกพี่ด้วยนะ ส่งอีเมลหรือฝากข้อความถึงพี่ๆบน FB ของพี่ๆ พวกพี่จะต้องดีใจแน่ๆถ้าได้เห็นข้อความจากน้องๆ

Cheers,
Adam & Anna


Bit..
Bit...