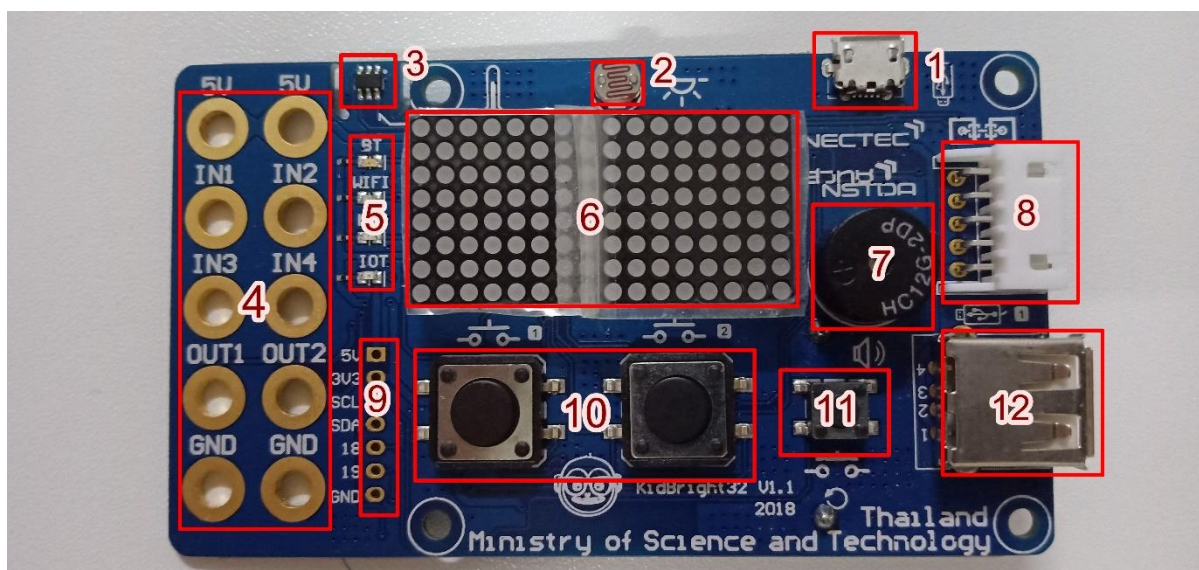


คู่มืออบรมการใช้บอร์ดKidBright





บอร์ดKidBrightด้านหน้า

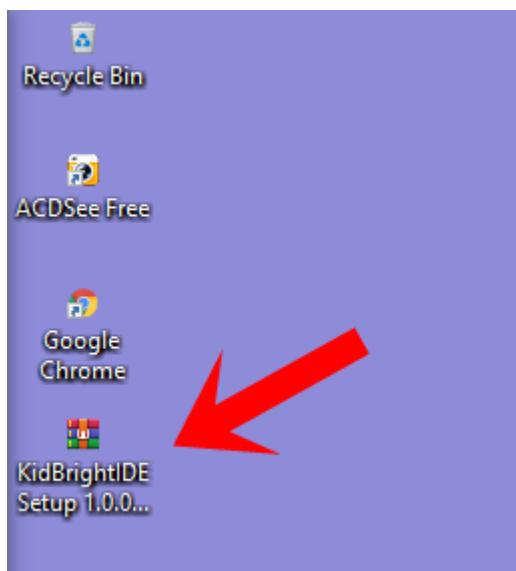


บอร์ดKidBrightด้านหลัง

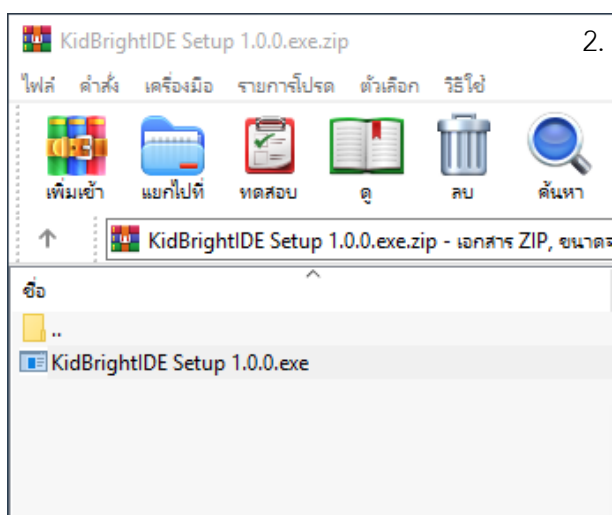
ตำแหน่งอุปกรณ์บนบอร์ดKidBright

- | | | |
|-----------------------------|----------------------------|------------------------|
| 1. ช่องเสียบ USB | 2. เซนเซอร์วัดแสง(LDR) | 3. เซนเซอร์วัดอุณหภูมิ |
| 4. พอร์ต I/O | 5. ไฟแสดงสถานะการเชื่อมต่อ | 6. Dot matrix |
| 7. Buzzer | 8. KidBright chain | 9. พอร์ต I/O(I2C) |
| 10. ปุ่มกด1 และ ปุ่มกด2 | 11. ปุ่มReset | 12. พอร์ตUSB |
| 13. Real Time Clock battery | 14. Processor | |

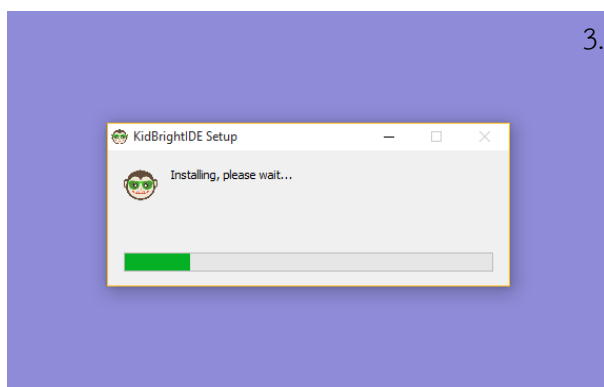
วิธีการติดตั้งโปรแกรม



1. ดาวน์โหลดโปรแกรม KidBright IDE ที่ <https://sites.google.com/view/wukidbright/downloads> จะได้เป็นไฟล์ .rar



2. เปิดไฟล์และดับเบิลคลิกเพื่อทำการติดตั้ง



3. หลังจากดับเบิลคลิกเพื่อทำการติดตั้ง โปรแกรมจะเลือกที่ตั้งอัตโนมัติ



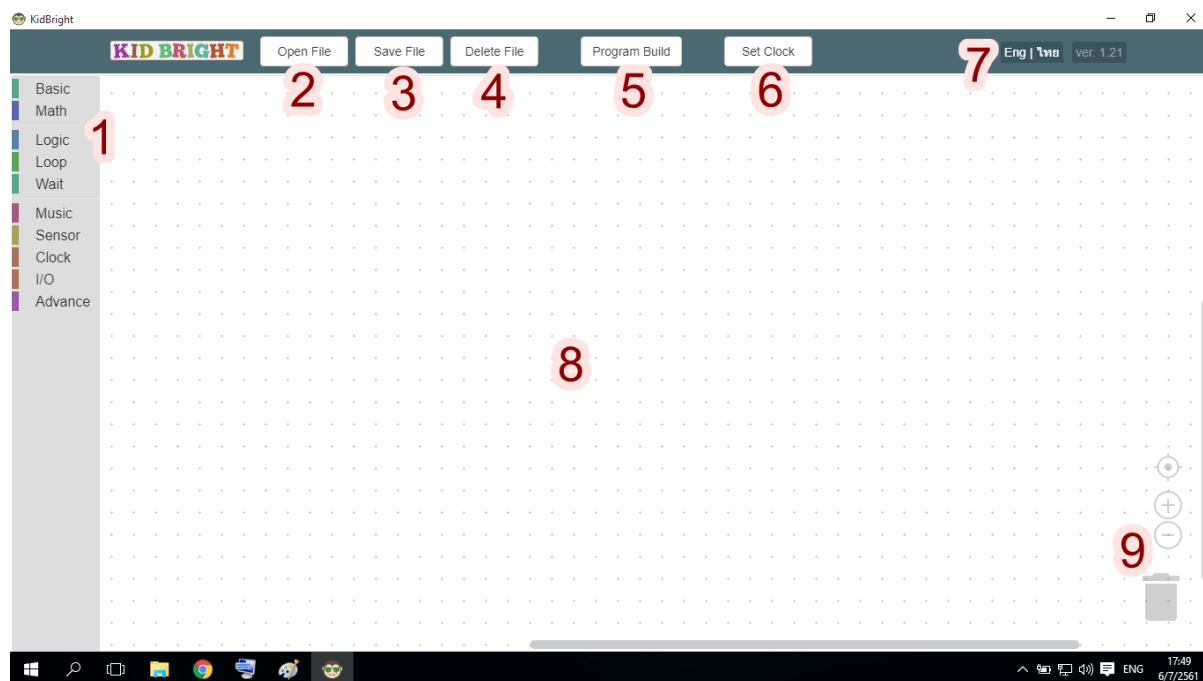
4. เมื่อติดตั้งเสร็จสิ้นโปรแกรมจะมีไอคอนโปรแกรมที่ Desktop สามารถตรวจสอบที่ติดตั้งโปรแกรมโดยการคลิกขวา -> Properties -> Target

วิธีแก้ปัญหาเบื้องต้นกรณี Port Failed



1. ลองปิดเปิดโปรแกรม KidBright ใหม่ หลังจากเปิดให้ลอง Build Program อีกครั้ง
 - 1.1. ถ้ายังไม่สามารถ Build Program ได้ให้ลอง Uninstall แล้วติดตั้งใหม่อีกครั้ง
2. ในกรณีข้อ(1) ยังไม่สามารถแก้ไขได้ให้ทำตามขั้นตอนดังนี้
 - 2.1. ลองใช้งานบอร์ดกับเครื่องอื่น ถ้าสามารถใช้งานได้ให้ทำข้อ(2.2)
 - 2.2. ดาวน์โหลดไดรเวอร์ Cypress USB UART Driver ที่
[ลิงค์http://www.cypress.com/file/135701/download](http://www.cypress.com/file/135701/download)

หน้าต่างโปรแกรม KidBright IDE



หน้าต่างโปรแกรม KidBright IDE

- | | | |
|-----------------------|-----------------------|------------------------|
| 1. แถบบล็อกเครื่องมือ | 2. เปิดโปรแกรมที่เซฟ | 3. เซฟโปรแกรมที่เขียน |
| 4. ลบโปรแกรมที่เซฟ | 5. แฟลชโปรแกรมลงบอร์ด | 6. ตั้งเวลาให้บอร์ด |
| 7. เปลี่ยนภาษา | 8. พื้นที่วางบล็อก | 9. ย่อ-ขยาย และลบบล็อก |

ความหมายของเครื่องมือแต่ละชนิด

1. แถบ Basic: แถบการแสดงผลโดยใช้ Dot matrix

LED 16x8

Clear LED 16x8

LED 16x8 2-chars

LED 16x8 Scroll

LED 16x8 Scroll When Ready

Delay 0.5

Forever

“ Hello World! ”

แสดงผล Dot matrix แบบกำหนดเอง

เคลียร์สถานะ Dot matrix

แสดงผล 2 ตัวอักษร

แสดงผลแบบเลื่อน

แสดงผลแบบเลื่อนเมื่อ Dot matrix พร้อมทำงาน

หน่วยเวลาทำงาน

วนทำงานซ้ำตลอดเวลา

กำหนดข้อความ

2. แถบ Math: แถบเครื่องมือดำเนินการทางคณิตศาสตร์ การประกาศตัวแปร และการกำหนดค่าตัวแปร

0

+

set x to

x

กำหนดค่าคงที่

ดำเนินการทางคณิตศาสตร์

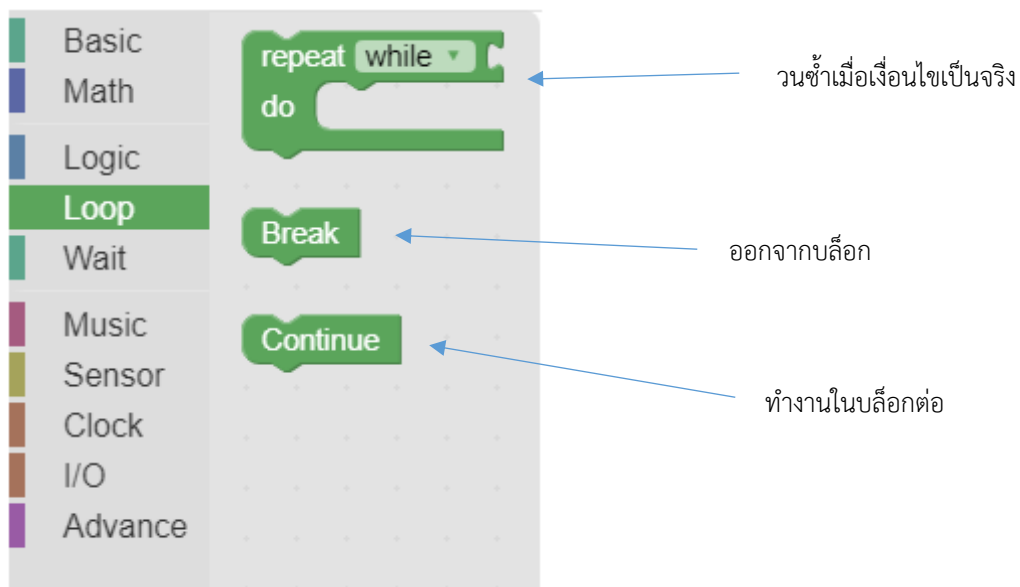
กำหนดค่าตัวแปร

ประกาศตัวแปร

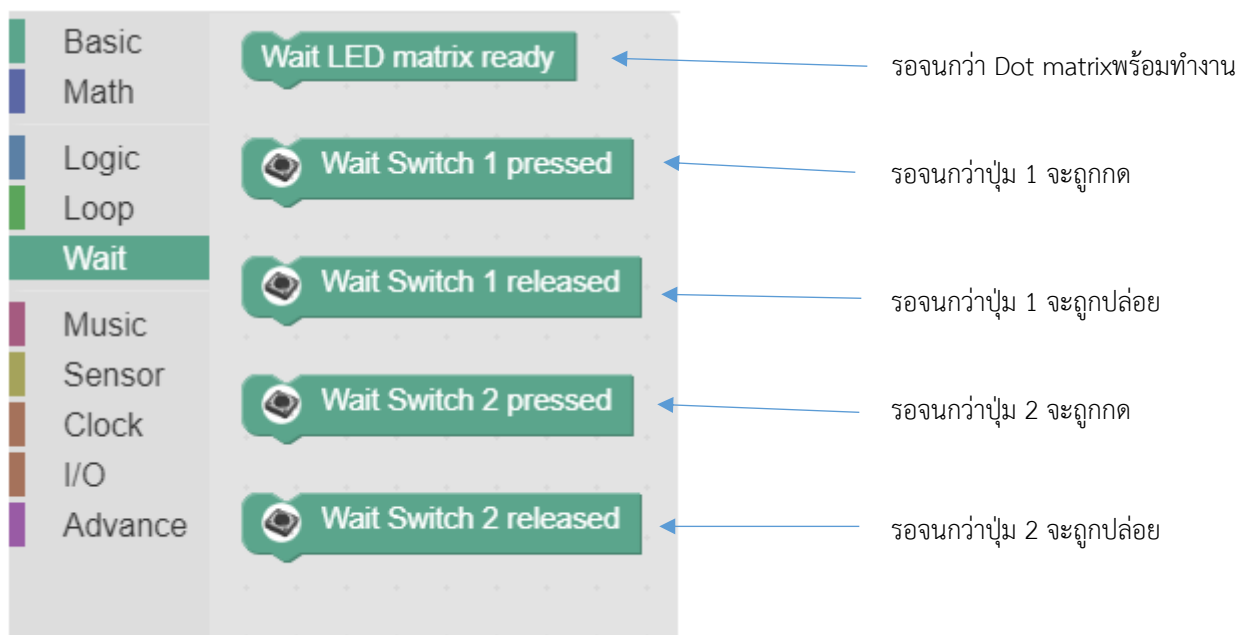
3. แถบ Logic: แถบเครื่องมือการตัดสินใจและการเปรียบเทียบ

Basic	if	←	ถ้า (เงื่อนไข) เป็นจริง ให้ทำ....
Math	do		
Logic	if	←	ถ้า (เงื่อนไข) เป็นจริง ให้ทำ....
Loop	do		
Wait	else	←	ถ้าเป็นเท็จ ให้ทำ....
Music			
Sensor	=	←	เปรียบเทียบค่าแบบคณิตศาสตร์ (ให้คำตอบจริง-เท็จ)
Clock	and	←	เปรียบเทียบค่าแบบตรรกศาสตร์ (ให้คำตอบจริง-เท็จ)
I/O	not	←	อินเวอร์ส(จริง->เท็จ, เท็จ->จริง)
Advance	true	←	ค่าคงที่แบบตรรกะ
	LED matrix ready to scroll	←	ตั้งสถานะให้ Dot matrix
	Switch 1 pressed	←	เมื่อปุ่ม 1 ถูกกดจะให้ค่าจริง
	Switch 1 released	←	เมื่อปุ่ม 1 ปล่อยจะให้ค่าจริง
	Switch 2 pressed	←	เมื่อปุ่ม 2 ถูกกดจะให้ค่าจริง
	Switch 2 released	←	เมื่อปุ่ม 2 ปล่อยจะให้ค่าจริง

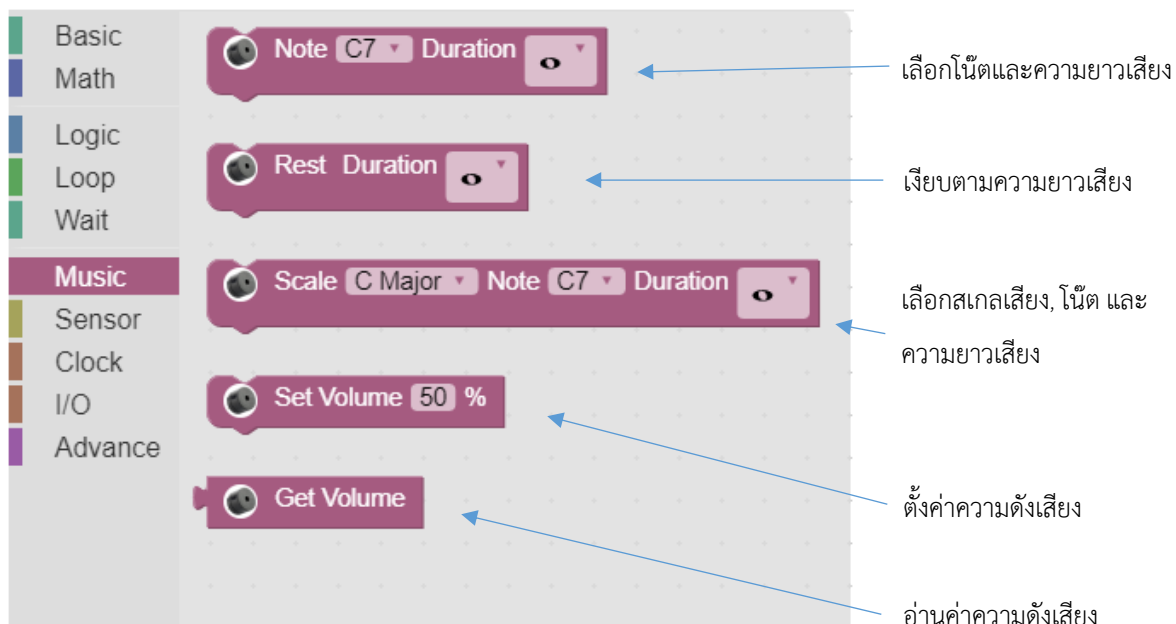
4. แถบ Loop: แถบเครื่องมือใช้ในการทำงานแบบวนซ้ำ



5. แถบ Wait: แถบที่รอสถานะของอุปกรณ์



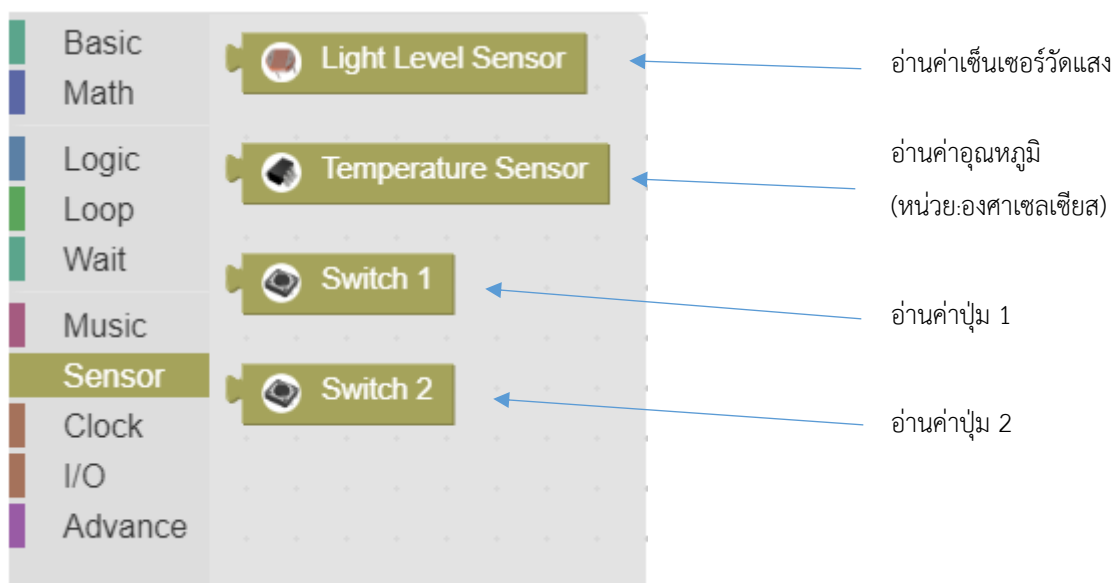
6. แถบ Music: แถบที่ใช้ควบคุมการทำงานของ Buzzer เพื่อให้เกิดเสียง



The image shows the Scratch Music block palette with five blocks: Note, Rest, Scale, Set Volume, and Get Volume. Blue arrows point from Thai text labels to specific parts of the blocks.

- Note** block:
 - เลือกโน้ตและความยาวเสียง (Select note and sound duration)
- Rest** block:
 - เงียบตามความยาวเสียง (Silence for the duration of the sound)
- Scale** block:
 - เลือกสเกลเสียง, โน้ต และความยาวเสียง (Select sound scale, note, and duration)
- Set Volume** block:
 - ตั้งค่าความดังเสียง (Set sound volume)
- Get Volume** block:
 - อ่านค่าความดังเสียง (Read sound volume)

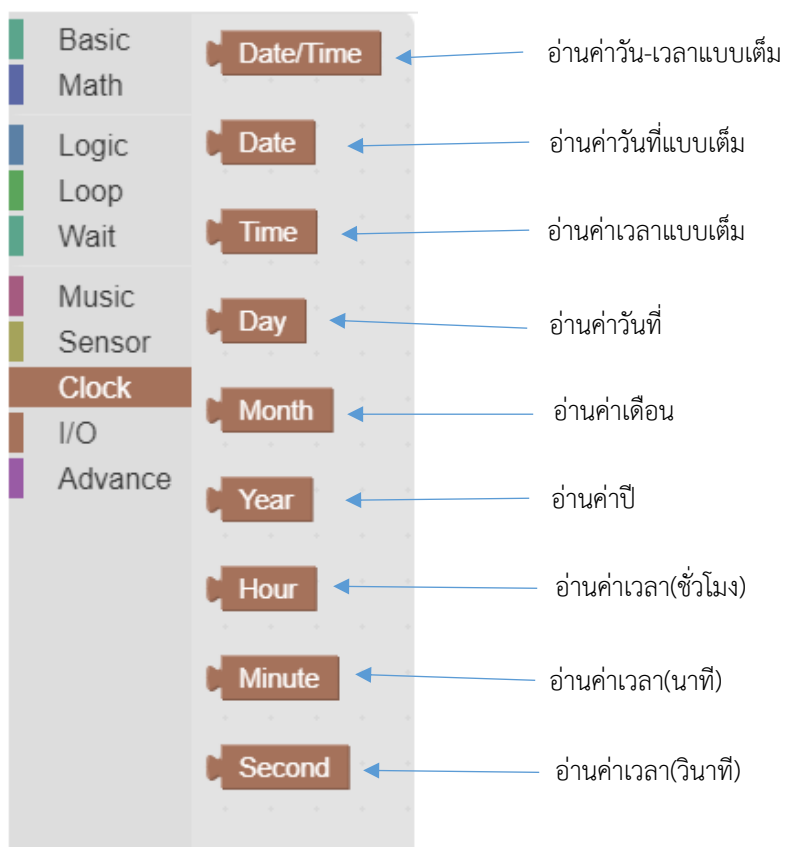
7. แถบ Sensor: แถบที่ใช้ในการอ่านค่าเซ็นเซอร์ที่อยู่บนบอร์ด



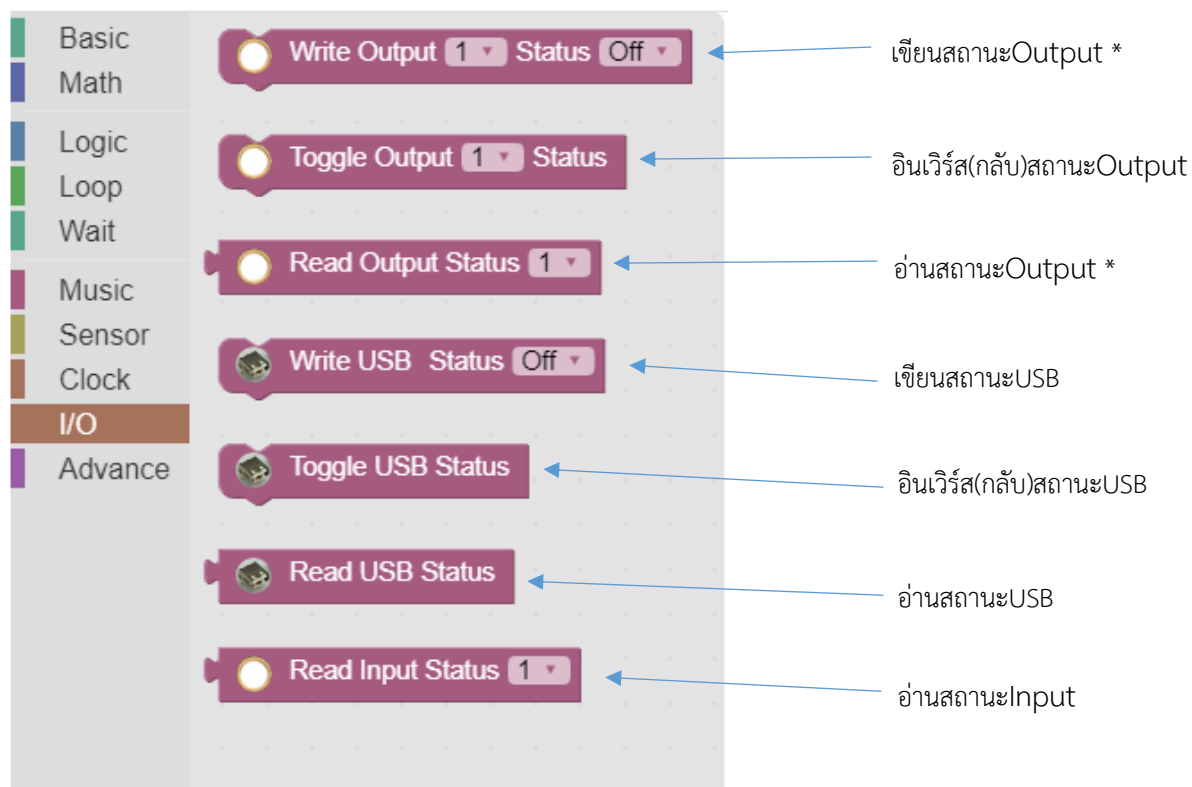
The image shows the Scratch Sensor block palette with four blocks: Light Level Sensor, Temperature Sensor, Switch 1, and Switch 2. Blue arrows point from Thai text labels to each block.

- Light Level Sensor** block:
 - อ่านค่าเซ็นเซอร์วัดแสง (Read light sensor value)
- Temperature Sensor** block:
 - อ่านค่าอุณหภูมิ (หน่วย: องศาเซลเซียส) (Read temperature (unit: degrees Celsius))
- Switch 1** block:
 - อ่านค่าปุ่ม 1 (Read button 1)
- Switch 2** block:
 - อ่านค่าปุ่ม 2 (Read button 2)

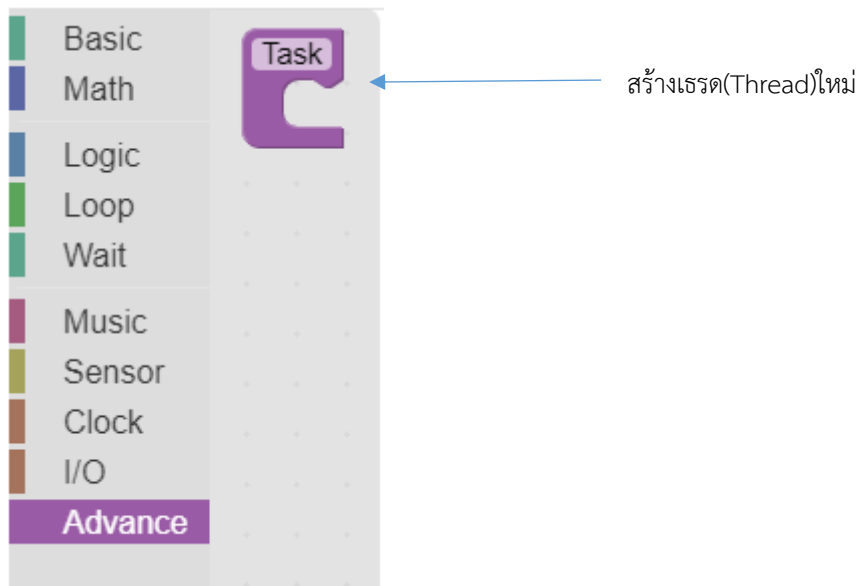
8. แถบ Clock: เป็นแถบที่ใช้อ่านค่าเวลาจาก RTC(Real Time Clock)



9. แถบ I/O: เป็นแถบใช้ในการควบคุม Input/Output *



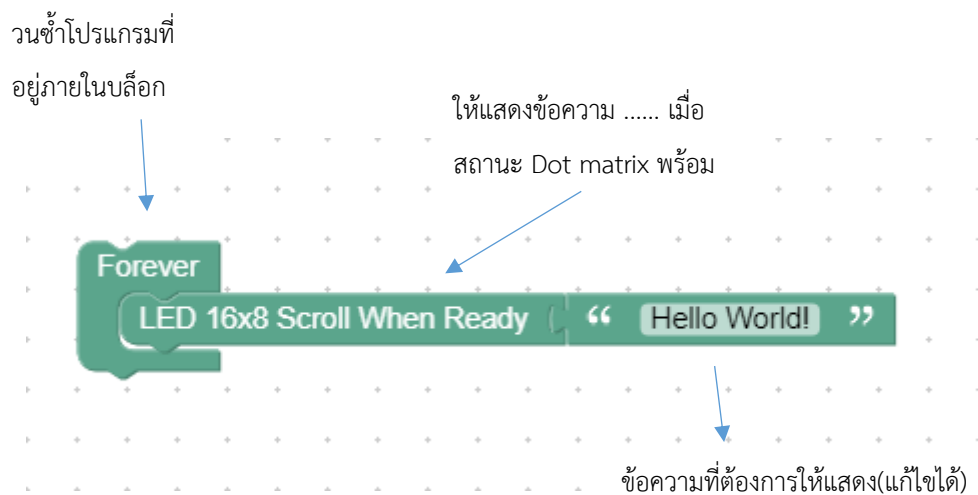
10. แถบ Task: เป็นแถบที่ใช้เขียน Multi-tasking



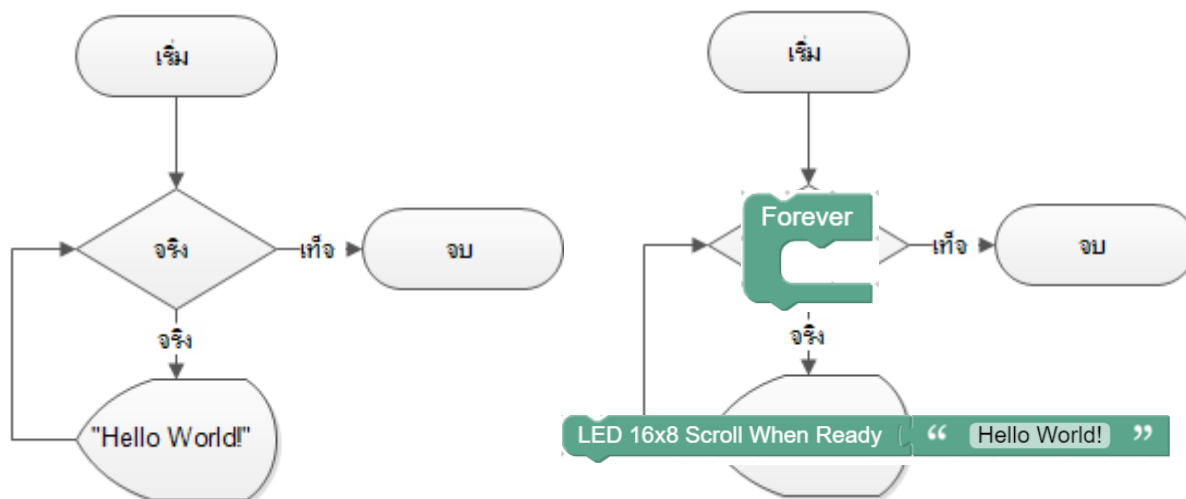
* วงจร Output เป็นวงจรแบบ Open-drain แนะนำให้ดู Schematic ของวงจบบอร์ดเพื่อประกอบความเข้าใจในการต่ออุปกรณ์ภายนอก

โปรแกรมที่ 1: Hello World!

โปรแกรม Hello World! เป็นโปรแกรมพื้นฐานไม่ว่าจะเขียนด้วยภาษาใดก็ตาม ในโปรแกรม KidBright IDE เราสามารถเขียนได้ตามรูปด้านล่าง โดยแต่ละบล็อกที่ประกอบเป็นโปรแกรมมีการทำงานดังนี้



มุมมอง Flowchart เทียบกับบล็อก



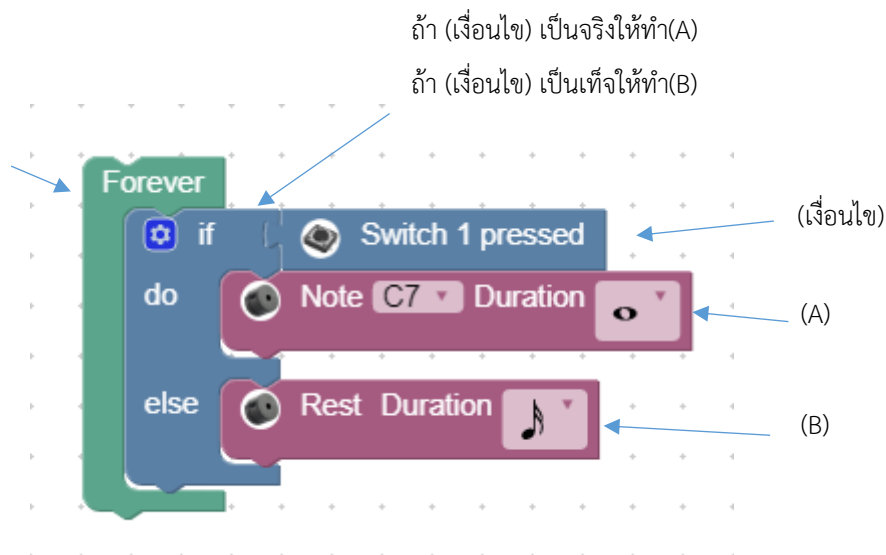
- ☐ ลองเอา Forever ออกแล้วสังเกตการเปลี่ยนแปลง
- ☐ ลองเปลี่ยนไปใช้ LED 16x8 Scroll และ LED 16x8 2-Char โดยลองสังเกตความแตกต่างระหว่างที่ใช้ LED 16x8 Scroll When Ready
- ☐ ให้สังเกตผลที่เกิดขึ้นถ้าหากเปลี่ยนจากบล็อก "Hello world!" เป็นปุ่ม 1 ดังภาพ



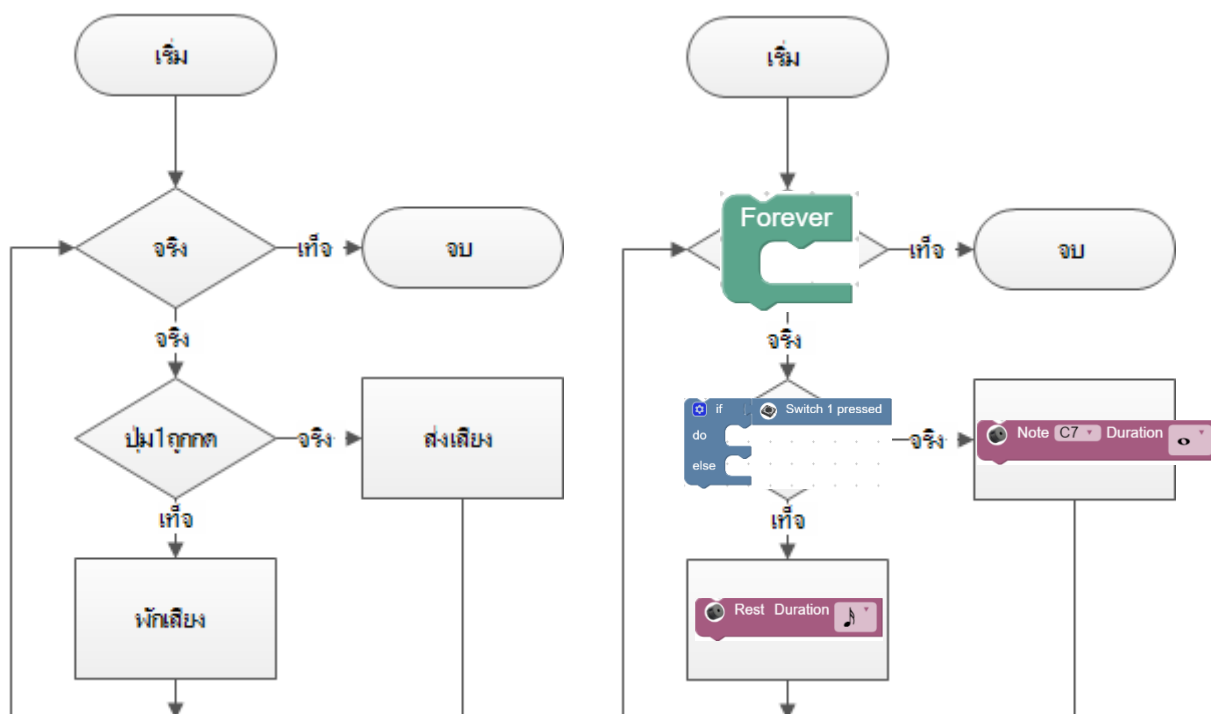
โปรแกรมที่ 2: กดปุ่มมีเสียง

โปรแกรมนี้เป็นตัวอย่างของการเขียนโปรแกรมแบบมีการตัดสินใจเมื่อกดปุ่ม 1 Buzzer จะมีเสียงและเมื่อปล่อยปุ่ม 1 จะไม่มีเสียง

วนซ้ำโปรแกรมที่
อยู่ภายในบล็อก








มุมมอง Flowchart เทียบกับบล็อก



ในบล็อก Music จะเห็นได้ว่าเราสามารถปรับได้สองส่วนคือ Note กับ Duration เราสามารถปรับได้ดังนี้

Symbol	C	D	E	F	G	A	B
Note	โด	เร	มี	ฟา	ซอล	ลา	ที

ตัวเลขข้างหลังโน้ตเป็นตัวบอกว่าโน้ตตัวนี้อยู่ในออกเตฟไหนเช่น C5 กับ C7 เป็นโน้ตโดทั้งคู่ แต่ว่า C7 จะมีเสียงสูงกว่า C5 เพราะว่ายู่ออกเตฟที่สูงกว่านั่นเอง

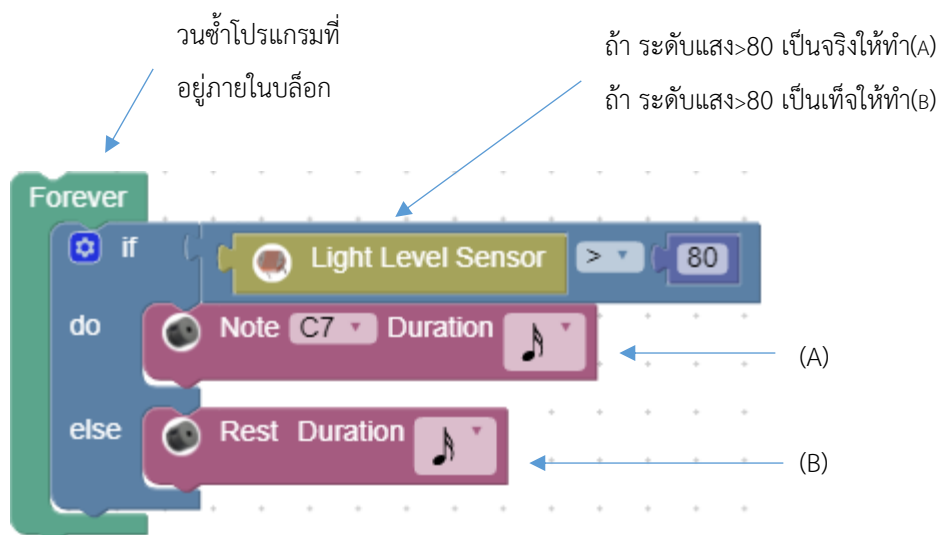
Symbol					
Duration	4	2	1	1/2	1/4

ตัวเลข Duration จะเป็นตัวบอกว่าโน้ตตัวนี้มีค่าความยาวเป็นกี่จังหวะของห้องเพลง (1ห้องเพลง = 4จังหวะ)

☐ จากรายละเอียดด้านบน ให้ลองแต่งเพลงแบบง่ายๆจากเพลงที่เคยได้ยิน เช่น เพลงช้าง

โปรแกรมที่ 3: นาฬิกาปลุกตามแสง

ปกติแล้วนาฬิกาปลุกจะปลุกจากเวลาที่ตั้งไว้ล่วงหน้า แต่โปรแกรมนีเป็นการปลุกโดยมีแสงเป็นตัวกระตุ้นการปลุก



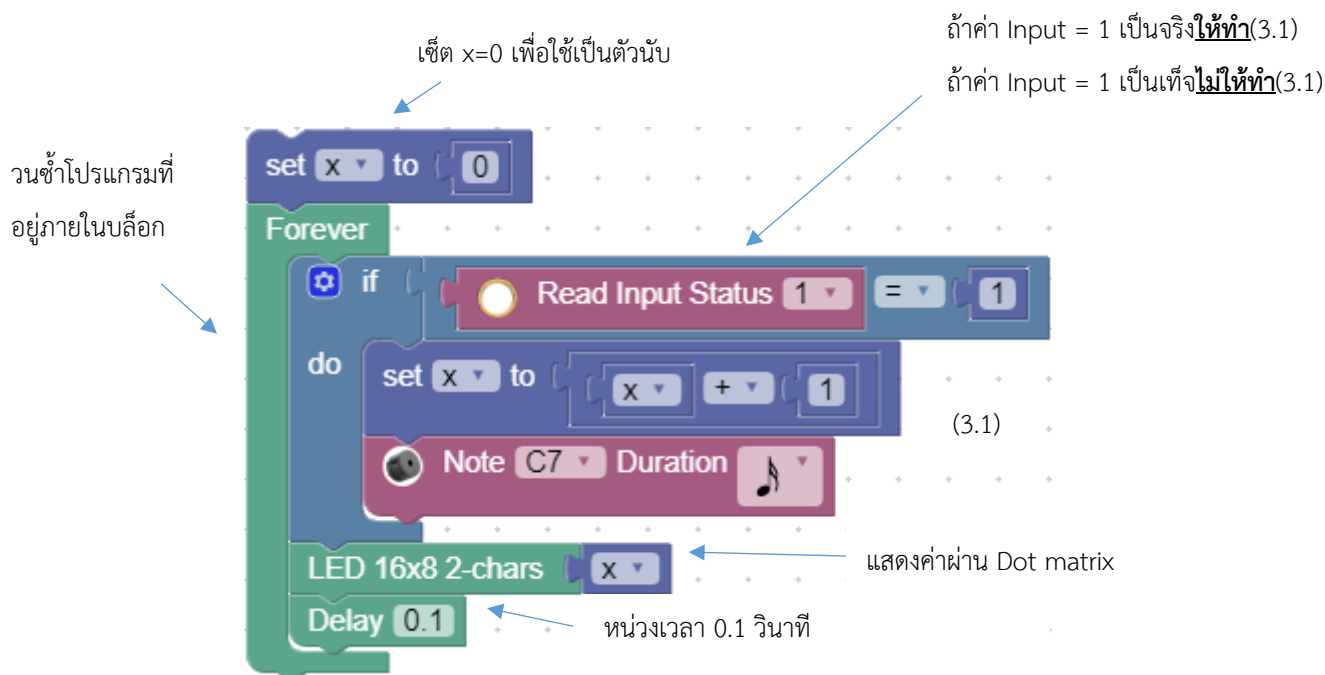
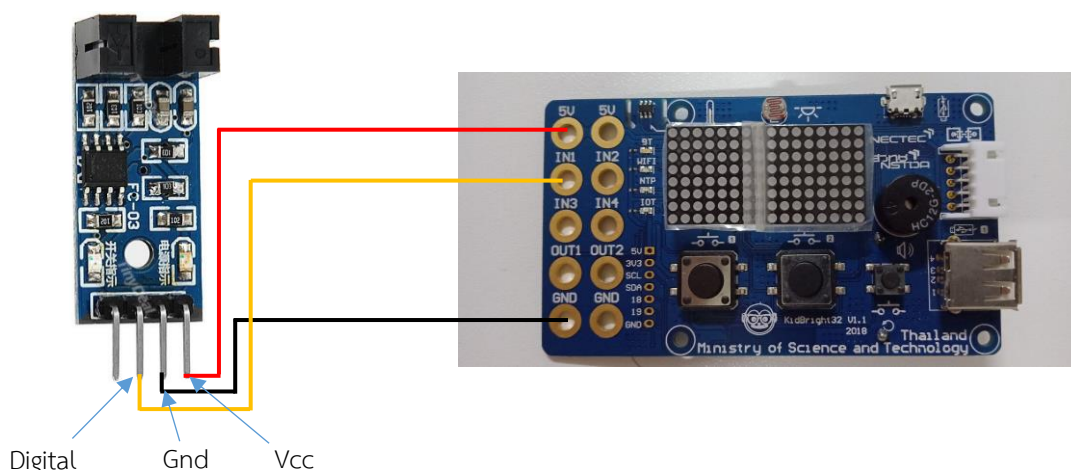
โปรแกรมนีจะไม่มีตัวอย่าง Flowchart ให้ดูเพราะว่าโครงสร้างเป็นแบบเดียวกับโปรแกรมข้อ(2) จะมีจุดต่างกันที่เงื่อนไขการตัดสินใจ โปรแกรมข้อ(2) เป็นการตัดสินใจโดยใช้สวิตช์ ซึ่งสวิตช์จะให้ค่าเป็น Digital คือ 0 กับ 1 แต่เซนเซอร์แสงจะให้ค่าเป็น Analog ระหว่าง 0-100 ดังนั้นเราจึงสามารถปรับค่าให้ทำงานได้ที่ระดับความเข้มแสงต่างกันได้

- ☐ จากตัวอย่างด้านบน จงเขียนโปรแกรมแจ้งเตือนเมื่ออุณหภูมิเกินกำหนด(เงื่อนไข: แจ้งเตือนเมื่ออุณหภูมิสูงกว่า 86 องศาฟาเรนไฮต์) และเขียน Flowchart ของโปรแกรมแจ้งเตือน

ตัวอย่างโปรแกรมที่ผ่านมาเป็นการใช้อุปกรณ์ที่อยู่บนบอร์ดทั้งหมด แต่ในส่วนหลังจากนี้จะมีการต่ออุปกรณ์ภายนอกเพิ่มเติม ไม่ว่าจะเป็นอุปกรณ์หรือเซนเซอร์ต่างๆ ดังนั้นหลังจากนี้ควรใช้ความระมัดระวังในการต่อวงจรไฟฟ้าอุปกรณ์หรือเซนเซอร์ เพื่อป้องกันอุปกรณ์เสียหาย

โปรแกรมที่ 4: โปรแกรมเคาน์เตอร์

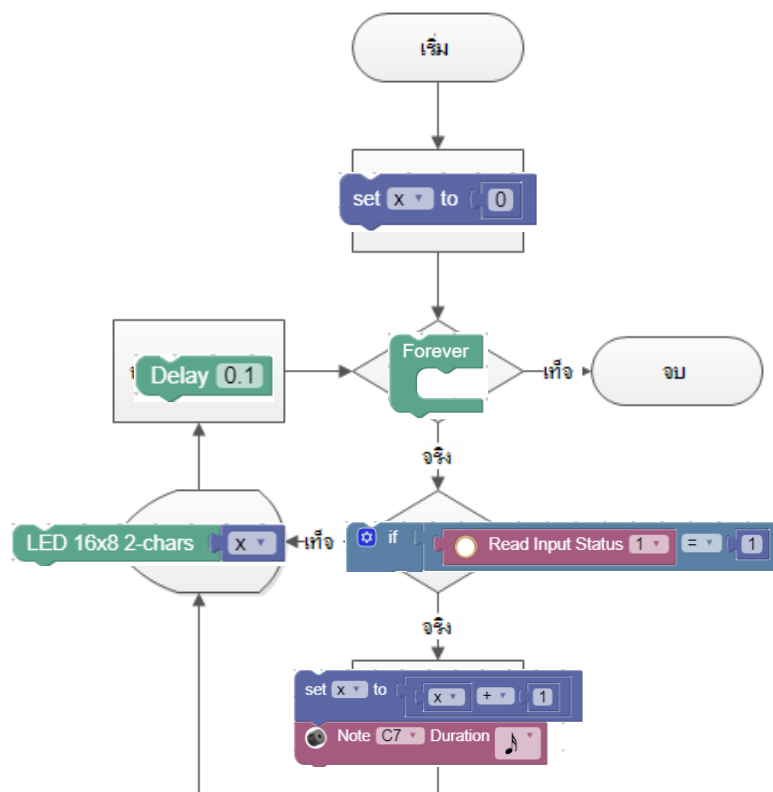
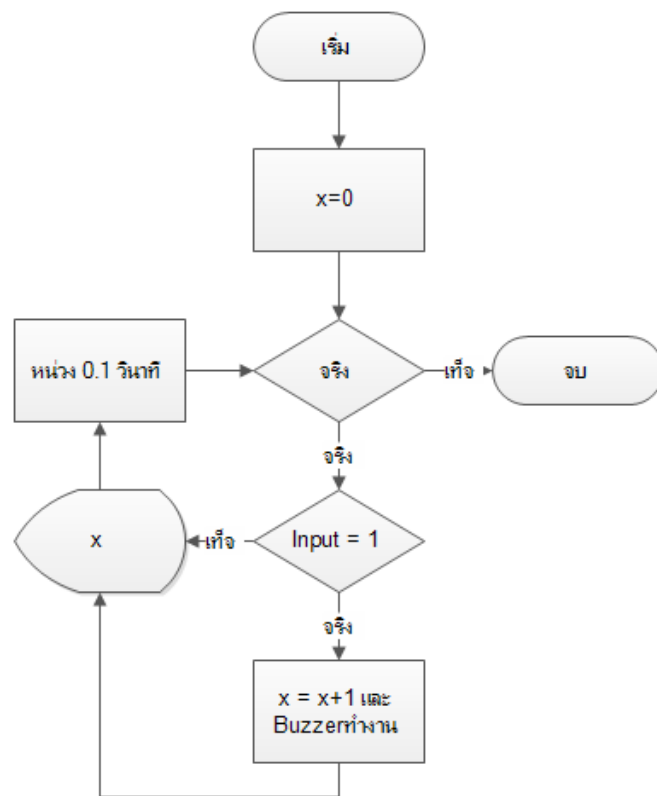
โปรแกรมเคาน์เตอร์เป็นโปรแกรมที่ใช้นับ ถ้าเอาแบบที่เห็นใกล้ๆตัวมากที่สุดก็น่าจะเป็นตู้ขายของอัตโนมัติ(นับเหรียญ) หรือ ประตูทางขึ้นคนเข้า-ออกได้



อธิบายการทำงาน

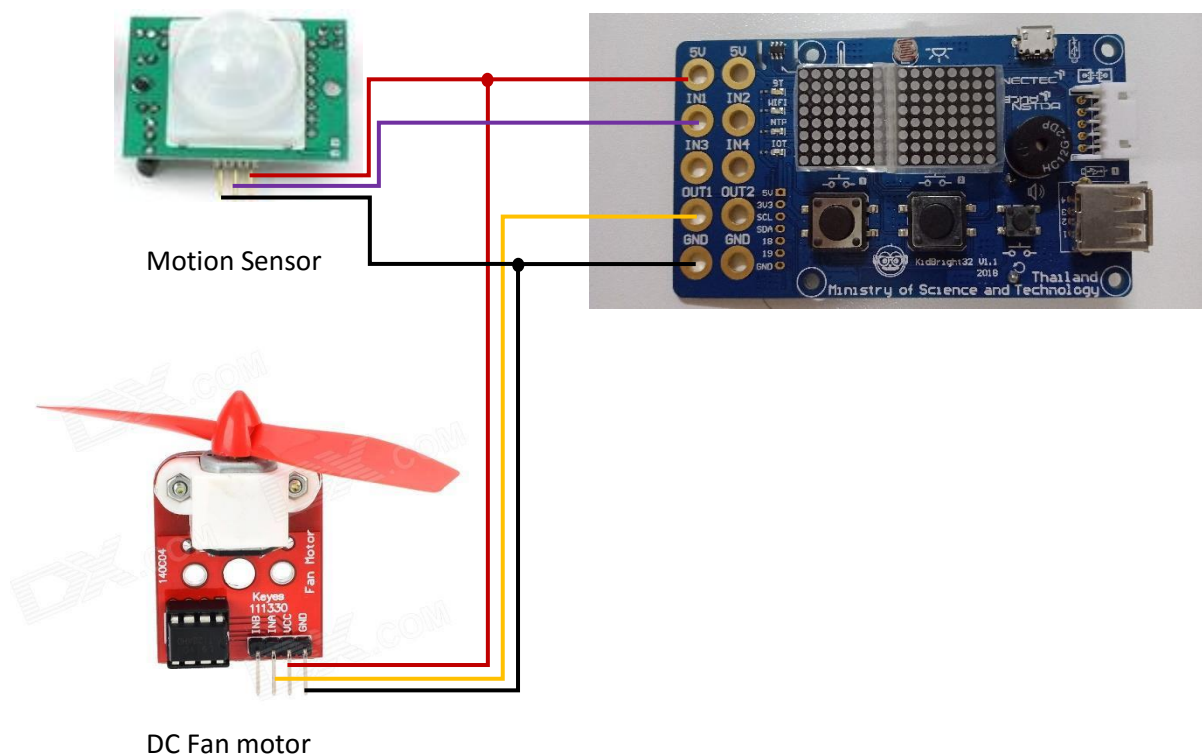
ปกติแล้วสัญญาณที่ Digital pin จะเท่ากับ 0 ทำให้IN1เป็น 0 เมื่อเข้าไปเช็คเงื่อนไขในifจะทำให้เป็นเท็จจึงไม่ทำงานภายในบล็อก แต่เมื่อมีวัตถุผ่านระหว่างช่องเซนเซอร์จะทำให้สัญญาณเปลี่ยนเป็น 1 เมื่อเข้าไปเช็คเงื่อนไขในifจะทำให้เป็นจริงจึงทำงานต่ภายในบล็อก ให้เพิ่มค่าตัวแปรขึ้นครั้งละ 1 พร้อมกับBuzzer ทำงาน หลังจากออกนอกบล็อกก็ทำการแสดงค่าตัวแปรกับหน่วงเวลาก่อนที่จะวนกลับไปทำงานซ้ำอีกรอบ

มุมมอง Flowchart เทียบกับบล็อก



โปรแกรมที่ 5: Motion Sence

Motion Sence หมายถึงการตรวจจับความเคลื่อนไหว โปรแกรมนี้จะเป็นตัวอย่างการใช้งาน Output โดยจะมี Pir Motion sensor เป็นตัวป้อนสัญญาณ Digital input ส่วนอุปกรณ์ที่เชื่อมต่อOutput นั้นจะเป็น DC Fan motor เพื่อใช้แสดงผลการตรวจจับ

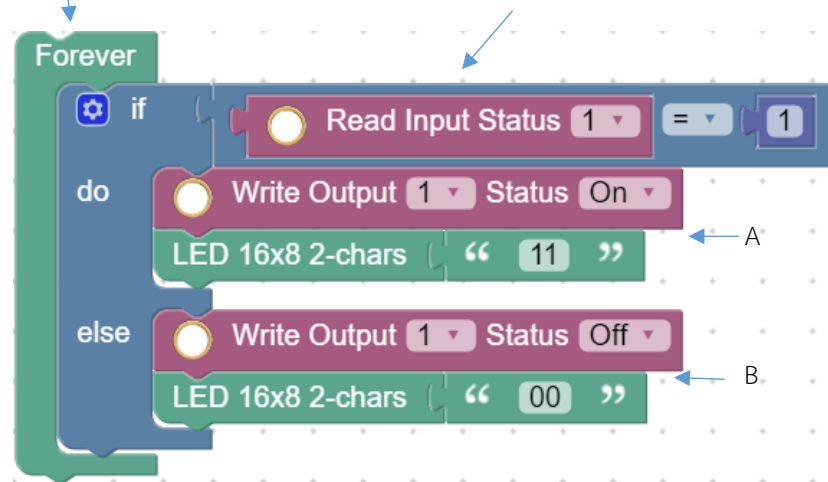


ด้วยความที่เป็นวงจรแบบ Open-Drain ทำให้ต้องใช้ความรู้ด้านไฟฟ้ามาประกอบในการต่อวงจรด้วย เนื่องจากวงจรแบบนี้จะทำการดึงกระแสจากแหล่งจ่ายโดยโดยไม่มีการจำกัดการไหล ดังนั้นอาจจะทำให้แหล่งจ่ายเกิดความเสียหายได้แนะนำให้ดู Schematic ของวงจรบนบอร์ดเพื่อประกอบความเข้าใจในการต่ออุปกรณ์ภายนอก(อยู่น้าสุดท้าย)

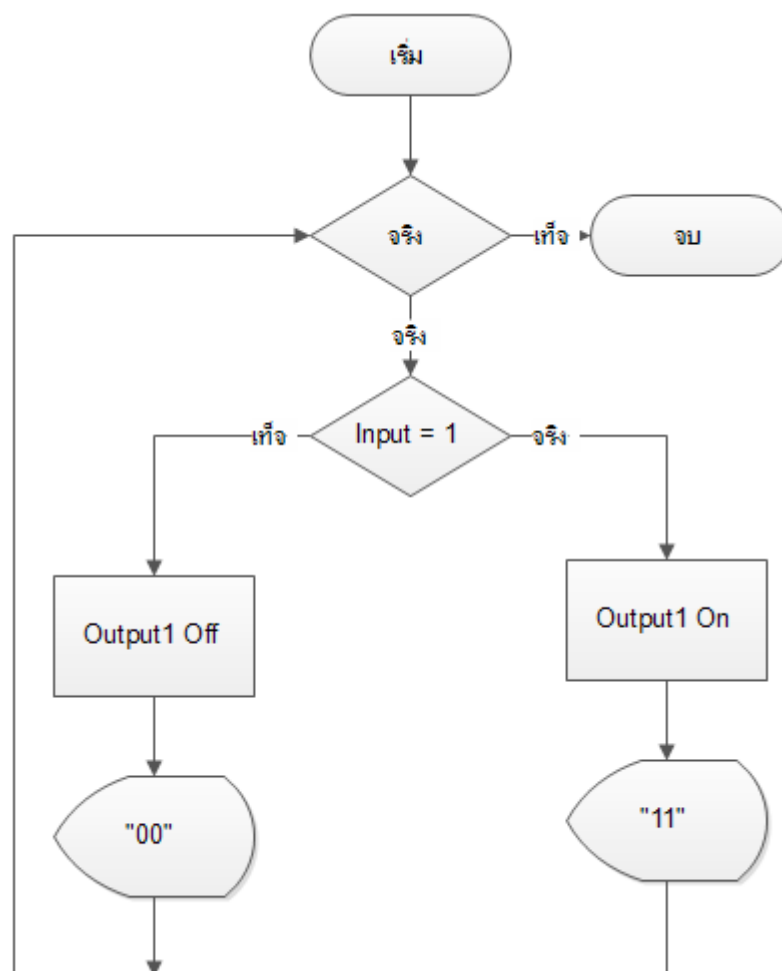
วนซ้ำโปรแกรมที่
อยู่ภายในบล็อก

ถ้าค่า Input = 1 เป็นจริงให้ทำ(A)

ถ้าค่า Input = 1 เป็นเท็จไม่ทำ(B)



Flowchart

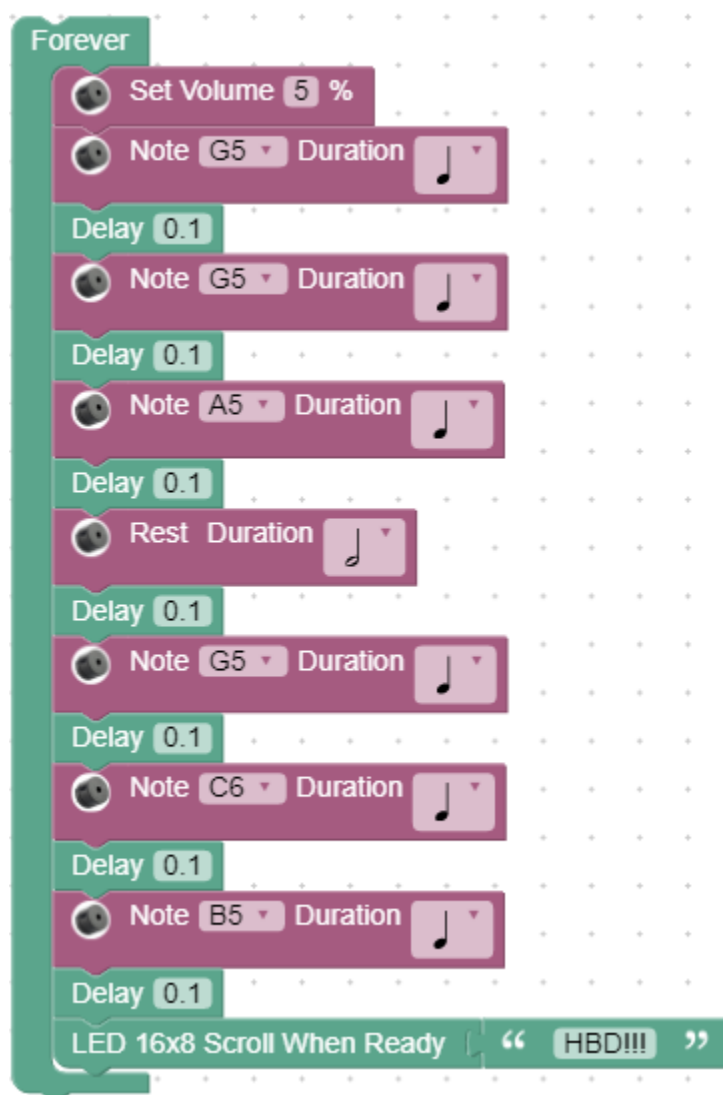


- ☐ ลองสลับตำแหน่งสาย IN A กับ IN B ที่ Module DC Fan motor แล้วสังเกตผลที่เปลี่ยนแปลง
- ☐ จากโปรแกรมตัวอย่าง จงเขียนโปรแกรมควบคุมทิศทางการหมุนพัดลมโดยใช้ปุ่ม1และ2

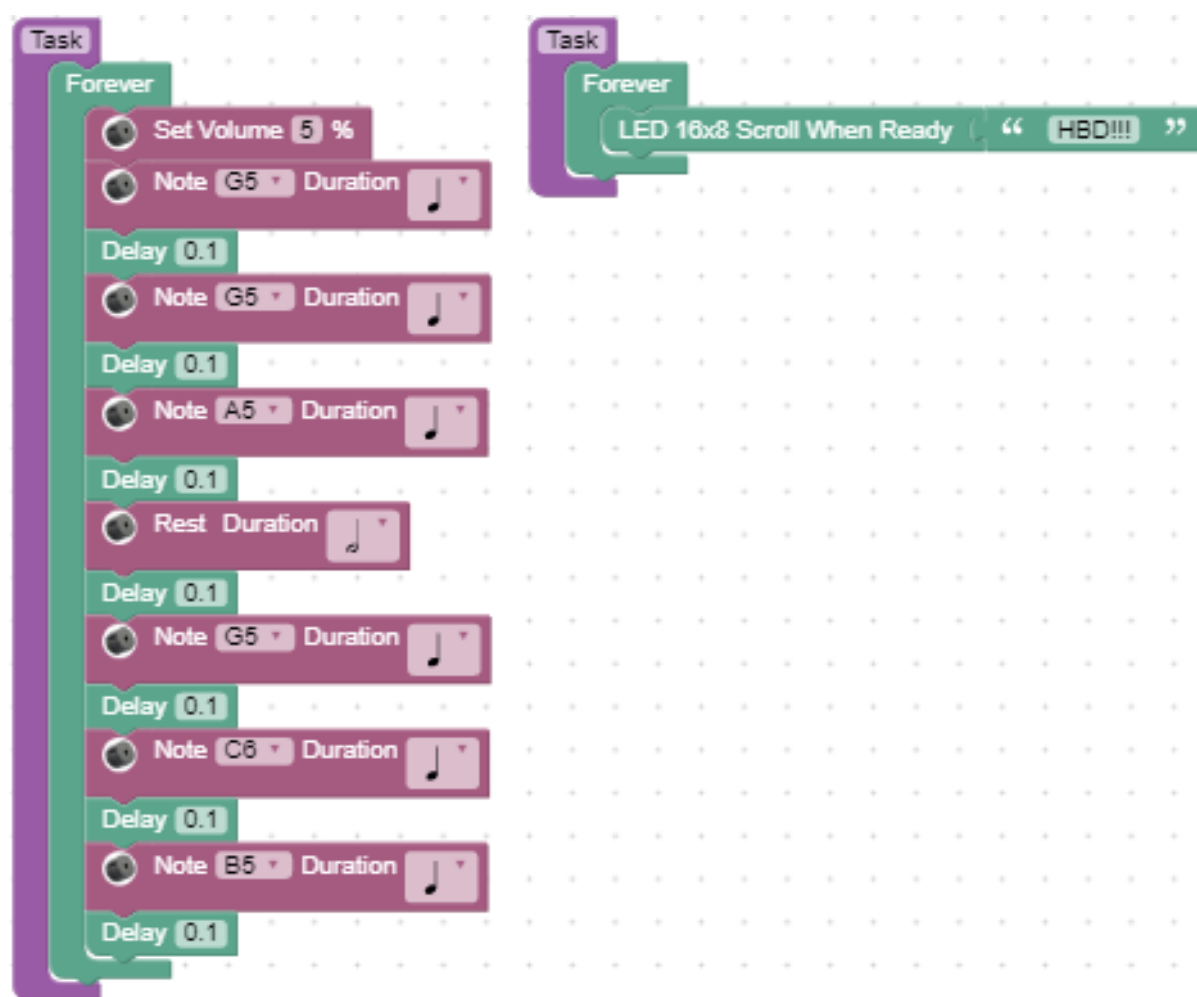
โปรแกรมที่ 6: จงทำงานหลายงานพร้อมๆกัน

ถ้าสังเกต Flowchart ให้ดีๆจะพบว่าในโปรแกรมหนึ่งๆจะมี Flow การไหลไปทิศทางเดียว โดย 1 Flowchart จะเท่ากับ 1 โปรแกรม แต่ถ้าเกิดเหตุการณ์ที่ต้องให้ทำงานสองอย่างพร้อมๆกัน การเขียนโปรแกรมแบบปกติจะไม่สามารถทำได้ เพราะต้องทำคำสั่งด้านบนก่อน แต่ด้วยการใช้ Task จะทำให้การทำงานหลายอย่างเกิดขึ้นพร้อมกันได้ เพื่อความเข้าใจลองเขียนโปรแกรมตามที่ให้ไปแล้วสังเกตผลเทียบกัน

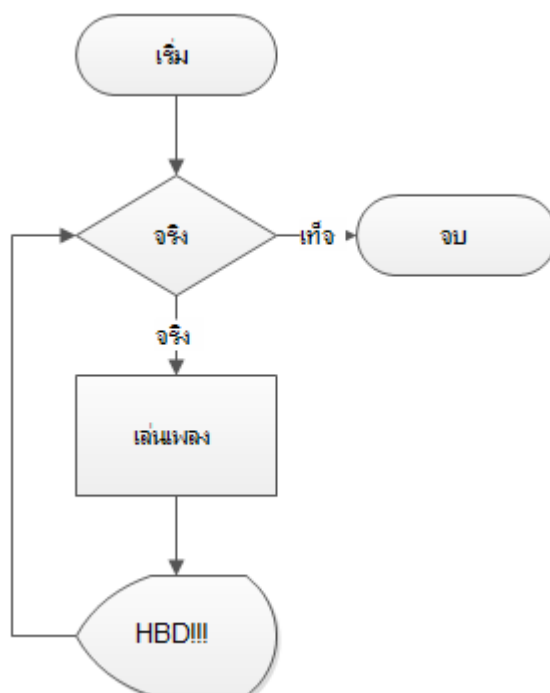
แบบที่1



แบบที่ 2



ถ้าได้ลองเขียนแล้วก็จะพบว่าแบบที่(1)จะแสดงคำว่า “HBD!!!” ซ้ำกว่าแบบที่(2)อย่างเห็นได้ชัด เป็นเพราะว่าแบบที่(1)นั้นจะต้องรอให้เพลงเล่นจบก่อนจึงจะแสดงข้อความออกมา



แต่ในแบบที่(2)จะเห็นว่าการแสดงผลทั้งคู่พร้อมๆกัน เมื่อสังเกตจาก Flowchart จะพบว่ามีสองฝั่งอยู่ในรูปเดียว การทำงานในลักษณะนี้เรียกว่า Multi-Tasking คือการประมวลผลหรือทำอะไรพร้อมๆกันโดยไม่ต้องรอให้งานอื่นเสร็จก่อนในแต่ละฝั่งจะเรียกว่าทาส์(Task) สามารถมาทำงานของฝั่งตัวเองต่อได้เลยในโปรแกรมตัวอย่างนี้จะประกอบไปด้วย 2 Task คือ (1)เล่นเพลงและ(2)แสดงผลDot matrix

