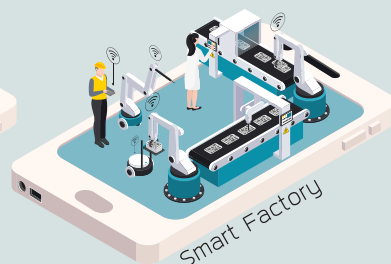
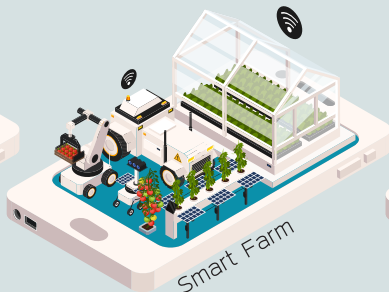
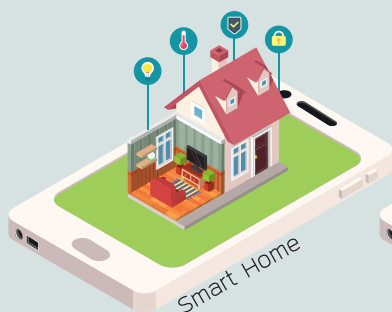


โครงการ
ค่ายอิเล็กทรอนิกส์
และสารสนเทศ

Cheewa
Inno Camp 2020

คู่มือประกอบการอบรม



คู่มือประกอบการอบรม

โครงการค่ายอิเล็กทรอนิกส์และสารสนเทศ

พิมพ์ครั้งที่ 1 สิงหาคม 2563

จำนวน 100 เล่ม

สงวนสิทธิ์ ตาม พ.ร.บ. ลิขสิทธิ์ (ฉบับเพิ่มเติม) พ.ศ. 2558 ไม่อนุญาตให้คัดลอก
ทำซ้ำ และดัดแปลง ส่วนหนึ่งส่วนใด ของหนังสือฉบับนี้ นอกจากได้รับอนุญาต
เป็นลายลักษณ์อักษรจากเจ้าของลิขสิทธิ์เท่านั้น



จัดทำโดย

งานพัฒนากำลังคนด้านอิเล็กทรอนิกส์และสารสนเทศ (HRDS)

ฝ่ายบริหารโครงสร้างพื้นฐาน (IMD)

ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ

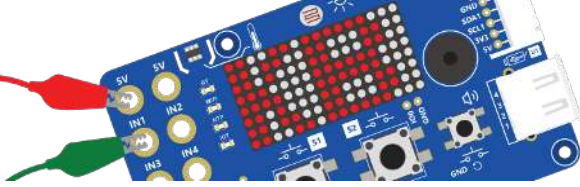
112 อุทยานวิทยาศาสตร์ประเทศไทย ตำบลคลองหนึ่ง

อำเภอคลองหลวง จังหวัดปทุมธานี 12120

โทรศัพท์ 02-564-6900 โทรสาร 02-564-6901-3

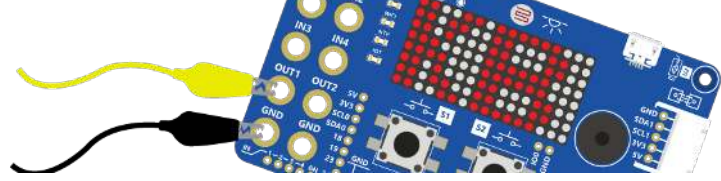
www.nectec.or.th

www.nectec.or.th/inno-hrd/th/embeddediot

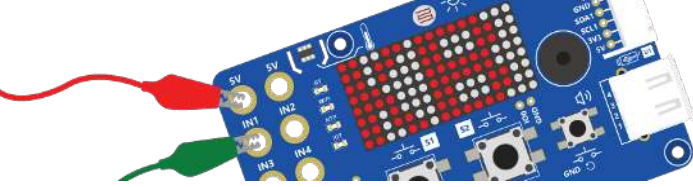


สารบัญ

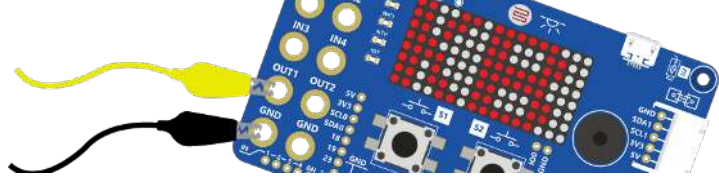
บทที่ 1 รู้จักกับระบบสมองกลฝังตัว	5
1.1 ความหมายของระบบสมองกลฝังตัว	5
1.2 ส่วนประกอบของระบบสมองกลฝังตัว	6
1.2.1 ตัวอย่างอุปกรณ์รับเข้า	6
1.2.2 ตัวอย่างอุปกรณ์ส่งออก	10
1.2.3 อุปกรณ์ประมวลผล	12
1.2.4 แหล่งพลังงาน	14
บทที่ 2 รู้จักกับอินเทอร์เน็ตของสรรพสิ่ง IOT	17
2.1 แนวคิดเรื่องอินเทอร์เน็ตของสรรพสิ่ง	17
2.2 องค์ประกอบของระบบอินเทอร์เน็ตของสรรพสิ่ง	18
2.3 ตัวอย่างผลิตภัณฑ์ในท้องตลาดภายใต้แนวคิดอินเทอร์เน็ตของสรรพสิ่ง	19
บทที่ 3 รู้จักกับบอร์ดสมองกลฝังตัว KIDBRIGHT และอุปกรณ์บนบอร์ด	21
3.1 ประวัติโดยสังเขป	21
3.2 ส่วนประกอบของบอร์ดรุ่น 1.3	22
3.3 ความแตกต่างระหว่างบอร์ดรุ่น 1.3 และ 1.5	24
3.4 ข้อมูลจำเพาะ (Specification)	25
บทที่ 4 รู้จักกับ KIDBRIGHT IDE	29
4.1 KidBright IDE คืออะไร?	29
4.2 การติดตั้ง KidBright IDE	30
4.3 การควบคุมบอร์ด KidBright และอุปกรณ์บนบอร์ดด้วย KidBright IDE	31
4.3.1 ส่วนประกอบในหน้าต่างหลักของโปรแกรม	31
4.3.2 ตัวอย่างการควบคุม Dot Matrix LED	32
4.3.3 ตัวอย่างการใช้ปุ่ม Button และ เล่นเสียงโน้ตดนตรีด้วย Buzzer	33

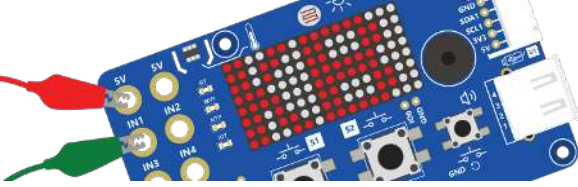


4.3.4 ตัวอย่างการอ่านค่าจาก Temperature Sensor และ LDR	34
4.4 แบบฝึกหัดและโจทย์	35
บทที่ 5 การเชื่อมต่อภายนอกเข้ากับบอร์ด KIDBRIGHT	37
5.1 การเชื่อมต่ออุปกรณ์ดิจิทัล	41
5.2 การควบคุมอุปกรณ์ไฟฟ้ากระแสตรงขนาดใหญ่	46
5.3 การควบคุมอุปกรณ์ไฟฟ้ากระแสสลับ	47
5.4 การเชื่อมต่ออุปกรณ์อนาล็อก	48
5.5 การเชื่อมต่ออุปกรณ์ I2C	49
บทที่ 6 การส่งรับข้อมูลระหว่างบอร์ด KIDBRIGHT และแอปพลิเคชัน KIDBRIGHT IOT	51
6.1 แนะนำแอปพลิเคชัน KidBright IoT	51
6.2 การติดตั้งแอปพลิเคชัน KidBright IoT ในมือถือ	52
6.3 ตัวอย่างการใช้งานเพื่อส่ง-รับข้อมูลระหว่างบอร์ด และ แอปพลิเคชัน	52
6.3.1 ตัวอย่างการส่งค่าอุณหภูมิที่วัดได้จากบอร์ด KidBright ไปที่แอปพลิเคชันมือถือ	53
6.3.2 ตัวอย่างการควบคุม Buzzer และ Dot Matrix LED จากแอปพลิเคชันมือถือ	54
6.4 แบบฝึกหัดและโจทย์	55
บทที่ 7 รู้จักกับ NETPIE 2020	57
7.1 ประวัติโดยสังเขป	57
7.2 การสมัครเป็นผู้ใช้งาน NETPIE 2020	58
7.3 แนวคิดของ NETPIE 2020 และการเลือกรูปแบบการใช้งานสำหรับนักพัฒนา	59
7.4 การสร้าง Project และ Device เบื้องต้นใน NETPIE	60
7.5 รู้จักกับ NETPIE Freeboard	63
บทที่ 8 การใช้งาน NETPIE 2020 ร่วมกับบอร์ด KIDBRIGHT	65
8.1 การใช้งานเชื่อมต่อ NETPIE ด้วย Kidbright IDE	65
8.2 วิธีการใช้งาน NETPIE 2020 ในรูปแบบการส่ง-รับข้อความ	67
8.2.1 การนำค่า sensor ไปแสดงบน NETPIE Freeboard	67



8.2.2 การควบคุมไฟ USB โดยใช้ NETPIE Freeboard	71
8.3 วิธีการใช้งาน NETPIE 2020 ในรูปแบบ Shadow	75
บทที่ 9 การใช้งาน LINE NOTIFY ร่วมกับบอร์ด KIDBRIGHT	81
9.1 รู้จักกับ Line Notify	81
9.2 การเพิ่ม Line Notify เป็นเพื่อน	81
9.3 การติดตั้ง Plugin ใน KidBright IDE เพื่อใช้งาน Line Notify	82
9.4 วิธีการใช้งาน Line Notify ร่วมกับบอร์ด KidBright	83
บทที่ 10 เปรียบเทียบการเขียนโปรแกรมด้วย KIDBRIGHT IDE และ ARDUINO	87
10.1 การเขียนโปรแกรมไมโครคอนโทรลเลอร์ด้วย Arduino	87
10.2 การติดตั้งโปรแกรม Arduino IDE	88
10.3 การตั้งค่าในโปรแกรม Arduino IDE เพื่อเขียนโปรแกรมบอร์ด KidBright	89
10.3.1 ติดตั้งบอร์ด ESP32 ใน Arduino IDE	89
10.3.2 ตรวจสอบการเชื่อมต่อกับบอร์ด KidBright	91
10.4 การติดตั้ง Library ใน Arduino IDE สำหรับใช้งานบอร์ด KidBright	94
10.4.1 การติดตั้ง Library ด้วย Library Manager ในโปรแกรม Arduino IDE	96
10.4.2 การติดตั้ง Library โดย Zip File ในโปรแกรม Arduino IDE	96
10.5 เปรียบเทียบการใช้งาน KidBright IDE และ Arduino	97
10.6 เปรียบเทียบตัวอย่างโปรแกรมที่สร้างด้วย KidBright IDE และ Arduino IDE	97
10.6.1 การแสดงข้อความผ่าน Dot Matrix LED	97
10.6.2 การอ่านค่า Light Sensor เพื่อนำมาแสดงบน Dot Matrix LED	99
10.6.3 การอ่านค่า Temperature Sensor เพื่อนำมาแสดงบน Dot Matrix LED	101
10.6.4 การควบคุม Buzzer เพื่อเล่นเสียงโน้ตดนตรี	103
10.6.5 การใช้งานปุ่มกดบนบอร์ด KidBright เพื่อเล่นเสียงดนตรี	106





บทที่ 1

รู้จักกับระบบสมองกลฝังตัว

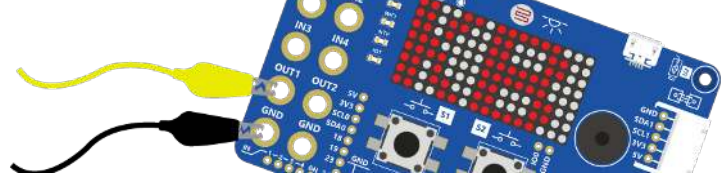
1.1 ความหมายของระบบสมองกลฝังตัว

ระบบสมองกลฝังตัวหมายถึงระบบหรืออุปกรณ์ที่สามารถทำงานได้อย่างอัตโนมัติตามคำสั่ง (โปรแกรม) ที่ได้กำหนดไว้ล่วงหน้า โดยอาจมีส่วนประกอบของอุปกรณ์ประมวลผล เช่น ไมโครคอนโทรลเลอร์ซึ่งทำหน้าที่เปรียบเสมือนกับสมองของระบบที่คอยรับข้อมูลจากอุปกรณ์รับเข้า (เซนเซอร์) ซึ่งทำหน้าที่เปรียบเสมือนกับประสาทสัมผัสเพื่อนำมาประมวลผลและตัดสินใจตามคำสั่งที่ได้กำหนดไว้ จากนั้นจึงสั่งการอุปกรณ์ส่งออกอาทิมอเตอร์ หลอดไฟหรือลำโพง เพื่อปฏิสัมพันธ์กับโลกภายนอกต่อไป

ในปัจจุบันระบบสมองกลฝังตัวถูกนำมาใช้อย่างแพร่หลายในชีวิตประจำวันของมนุษย์ ตัวอย่างเช่น ตุ๊กตาที่สามารถส่งเสียงร้องได้เมื่อถูกสัมผัส เต้าไมโครเวฟที่สามารถตั้งเวลาหรืออุ่นอาหารได้โดยอัตโนมัติ ระบบกันชนด้านหลังในรถยนต์ที่สามารถส่งเสียงเตือนเมื่อผู้ขับขี่กำลังจะถอยรถชนสิ่งกีดขวางด้านหลัง ประตูของร้านสะดวกซื้อที่สามารถเปิดได้เองเมื่อมีคนเดินผ่าน หลอดไฟที่สามารถเปิด-ปิดได้แบบอัตโนมัติเมื่อมีคนเดินผ่าน เป็นต้น



รูปภาพ 1-1 ตัวอย่างระบบสมองกลฝังตัวในชีวิตประจำวัน

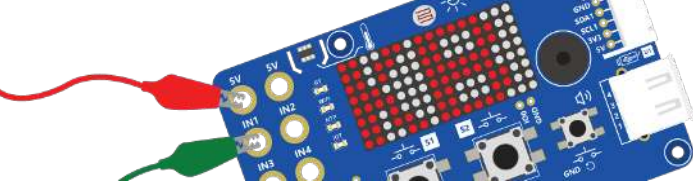



1.2 ส่วนประกอบของระบบสมองกลฝังตัว

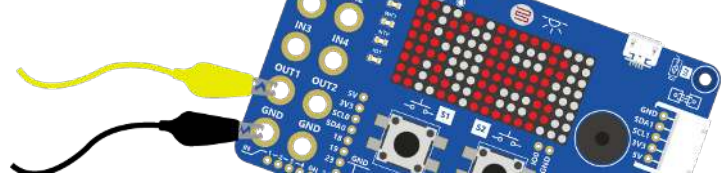
ระบบสมองกลฝังตัวมีส่วนประกอบพื้นฐาน 4 ส่วนดังนี้ 1. อุปกรณ์รับเข้า (Sensor) ทำหน้าที่เป็นประสาทสัมผัสของระบบ ตัวอย่างเช่น ปุ่มกด เซนเซอร์วัดอุณหภูมิ เป็นต้น 2. อุปกรณ์ส่งออก (Actuator) ทำหน้าที่ปฏิสัมพันธ์กับภายนอก เช่น หลอดไฟ จอภาพ มอเตอร์ ลำโพง เป็นต้น 3. อุปกรณ์ประมวลผลทำหน้าที่เป็นเหมือนสมองของอุปกรณ์เช่นไมโครคอนโทรลเลอร์หรือคอมพิวเตอร์ 4. แหล่งพลังงาน เช่น แบตเตอรี่ ถ่านไฟฉาย อัดแดปเตอร์ เป็นต้น

1.2.1 ตัวอย่างอุปกรณ์รับเข้า

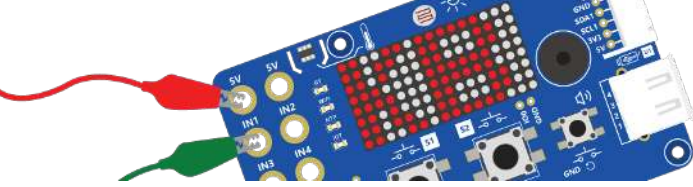
ประเภท/ชื่อ	ภาพตัวอย่าง	คำอธิบาย
สวิตช์/ ปุ่มกด	 <p>ที่มา: https://learn.sparkfun.com/tutorials/switch-basics/momentary-switches</p>	ปุ่มกดเป็นอุปกรณ์พื้นฐานในการควบคุมการทำงานของระบบผ่านการกดของผู้ใช้ ปุ่มกดที่มีจำหน่ายในท้องตลาดมีหลากหลายขนาด สี สัน และรูปทรงสำหรับการยึดติดแบบต่าง ๆ และสามารถแบ่งได้เป็น 2 ประเภทหลักคือ ปุ่มกดแบบกดติดปล่อยดับและปุ่มกดแบบกดติดกดดับ
สวิตช์/สวิตช์ เลื่อน	 <p>ที่มา: https://www.sparkfun.com/products/9609</p>	ปุ่มเลื่อนเป็นปุ่มอีกประเภทหนึ่งที่ใช้ในการตัดกระแสไฟฟ้าเพื่อเปิด-ปิดการทำงานของวงจร
สวิตช์/ลิมิต สวิตช์	 <p>ที่มา: https://www.sparkfun.com/products/13013</p>	ลิมิตสวิตช์เป็นปุ่มกดที่มักใช้ในการตรวจจับการเคลื่อนที่ของกลไกหรือวัตถุต่าง ๆ เช่น ใช้ตรวจสอบรางม้วนว่าเลื่อนมาถึงขอบแล้วหรือไม่เพื่อยุติการทำงานของมอเตอร์


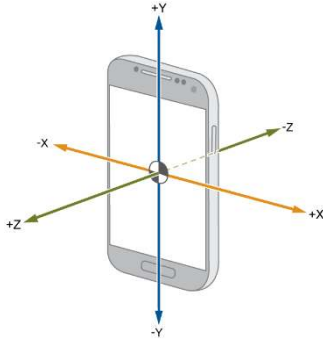

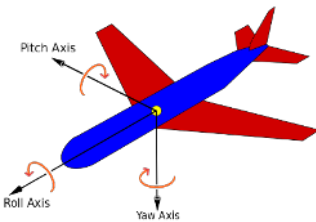


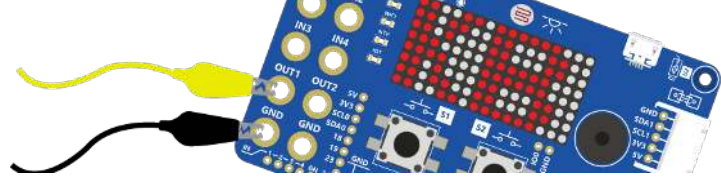
<p>สวิตช์/ ปุ่มกดแบบ คาปาซิ ทีฟทัช</p>	 <p>ที่มา: https://www.sparkfun.com/products/14520</p>	<p>ปุ่มกดแบบคาปาซิทีฟทัชเป็นปุ่มกดที่นิยมในอุปกรณ์ที่ต้องการการกันน้ำหรือฝุ่นละอองเนื่องจากสามารถซ่อนอยู่ด้านหลังวัสดุเช่นพลาสติก อะคริลิกหรือกระดาษที่มีความหนาไม่มากได้</p>
<p>สวิตช์/ปุ่ม หมุน</p>	 <p>ที่มา: https://www.sparkfun.com/products/9806</p>	<p>ปุ่มหมุนเป็นอุปกรณ์พื้นฐานในการควบคุมการทำงานของระบบเช่นการปรับเพิ่มลดความสว่างของหลอดไฟ ความดังของเสียง เป็นต้น</p>
<p>แสง/ตัว ต้านทาน ปรับค่าตาม แสง</p>	 <p>ที่มา: https://www.sparkfun.com/products/9088</p>	<p>ตัวต้านทานปรับค่าตามแสงเป็นอุปกรณ์ที่ใช้ในการวัดความสว่างของแสงโดยรอบ เช่น ใช้ในไฟสนามหรือไฟถนนแบบอัตโนมัติที่สามารถเปิด-ปิดได้เองเมื่อถึงเวลากลางคืน เป็นต้น</p>
<p>อุณหภูมิ/ เซนเซอร์วัด อุณหภูมิ</p>	 <p>ที่มา: https://www.sparkfun.com/products/13314</p>	<p>เซนเซอร์วัดอุณหภูมิแบบไมโครเทคโนโลยีเป็นเซนเซอร์วัดอุณหภูมิที่มีขนาดเล็กมาก มักใช้ในอุปกรณ์เช่น โทรศัพท์มือถือ เครื่องคอมพิวเตอร์และอุปกรณ์พกพาต่าง ๆ เซนเซอร์ประเภทนี้ส่วนใหญ่จะวัดอุณหภูมิได้ในช่วง -40 ถึง 125 องศาเซลเซียส</p>



<p>อุณหภูมิ/ เซนเซอร์วัด อุณหภูมิ แบบกันน้ำ</p>	 ที่มา: https://www.sparkfun.com/products/11050	<p>เซนเซอร์ประเภทนี้มีการทำงานคล้ายกับเซนเซอร์วัดอุณหภูมิแบบไมโครฯ ด้านบนแต่สามารถจุ่มลงในน้ำได้จึงเหมาะกับการที่ต้องติดตั้งใกล้ของเหลว</p>
<p>ความชื้น/ เซนเซอร์วัด ความชื้น</p>	 ที่มา: https://www.sparkfun.com/products/13763	<p>เซนเซอร์วัดความชื้นสามารถวัดค่าความชื้นสัมพัทธ์ได้ตั้งแต่ 0-100% เซนเซอร์ประเภทนี้มักถูกนำไปใช้สร้างโครงการเช่นโรงเรือนปลูกเห็ดอัญหริยะ เป็นต้น</p>
<p>ระยะทาง/ เซนเซอร์วัด ระยะ ทางผ่าน คลื่นเสียง อัลตราโซนิก</p>	 ที่มา: https://www.sparkfun.com/products/13959	<p>เซนเซอร์วัดระยะทางผ่านคลื่นเสียงอัลตราโซนิกใช้หลักการการวัดระยะเวลาที่คลื่นเสียงใช้ในการสะท้อนกลับมาที่ตัวเซนเซอร์เพื่อคำนวณเป็นระยะทาง โดยมีระยะการวัดได้ตั้งแต่ 2 - 400 ซม. แต่อาจมีความคลาดเคลื่อนสูงเมื่อวัตถุที่ต้องการวัดมีพื้นผิวนุ่มหรืออยู่ในระยะใกล้</p>
<p>ระยะทาง/ เซนเซอร์วัด ระยะ ทางผ่านแสง</p>	 ที่มา: https://www.sparkfun.com/products/12784	<p>เซนเซอร์วัดระยะทางผ่านแสงส่วนใหญ่จะมีขนาดเล็กกว่าเซนเซอร์ที่ใช้คลื่นเสียงอัลตราโซนิกและมักใช้แสงอินฟราเรดในย่านที่ตามนุษย์ไม่สามารถมองเห็นได้ ระยะการวัดสูงสุดไม่เกิน 10-25 ซม. อย่างไรก็ตามในปัจจุบันมีเซนเซอร์ที่ใช้แสงเลเซอร์แทน ทำให้สามารถวัดได้ระยะสูงสุดถึง 4 เมตร</p>



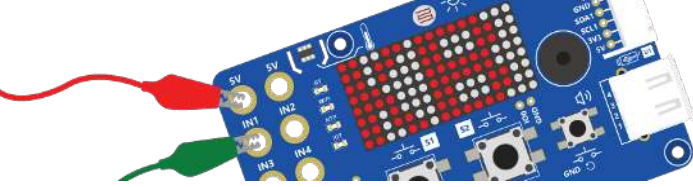
<p>ความเร่ง/ มาตร ความเร่ง</p>	 <p>ที่มา: https://www.sparkfun.com/products/9269</p>	<p>มาตรวัดความเร่งเป็นอุปกรณ์ที่ใช้ในการวัดความเร่งโดยสามารถวัดได้ทั้งความเร่งคงที่ (จากแรงดึงดูดของโลก) และความเร่งแบบฉับพลัน (จากการเคลื่อนที่และการสั่นสะเทือน) เซนเซอร์ประเภทนี้มักถูกนำไปประยุกต์ใช้ในการตรวจจับการเคลื่อนไหว ตรวจจับการหล่น หรือตรวจจับความเอียง เป็นต้น</p>  <p>ที่มา: https://www.mathworks.com/help/supportpkg/android/ref/accelerometer.html?requestedDomain=</p>
<p>การหมุน/ไจโรสโคป</p>	 <p>ที่มา: https://www.sparkfun.com/products/13284</p>	<p>ไจโรสโคปเป็นเซนเซอร์ที่ใช้ในการวัดความเร็วการหมุนของอุปกรณ์ เซนเซอร์ประเภทนี้มักถูกนำไปประยุกต์สร้างเป็นเกมเช่นใน Wiimote อุปกรณ์ควบคุมหลักสำหรับเครื่องเล่นเกม Wii ของบริษัท Nintendo</p>  <p>ที่มา: http://samselectronicsprojects.blogspot.com/2014/07/getting-roll-pitch-and-yaw-from-mpu-6050.html</p>



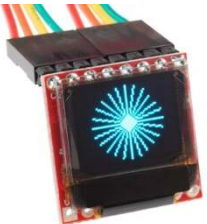



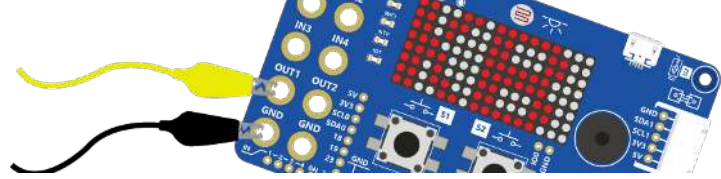
เสียง/ ไมโครโฟน	 <p>ที่มา: https://www.sparkfun.com/products/12758</p>	ไมโครโฟนเป็นอุปกรณ์ที่เปลี่ยนจากเสียงให้เป็นสัญญาณไฟฟ้า โดยสามารถนำไปประยุกต์ใช้ในเครื่องอัดเสียง อุปกรณ์ที่ส่งงานผ่านเสียง เป็นต้น
สิ่งมีชีวิต/ เซนเซอร์ ตรวจจับ คลื่นรังสีอิน ฟราเรด	 <p>ที่มา: https://www.sparkfun.com/products/13968</p>	เซนเซอร์ตรวจจับคลื่นรังสีอินฟราเรดเป็นเซนเซอร์ที่สามารถตรวจจับคลื่นรังสีอินฟราเรดที่แผ่ออกมาระหว่างการเคลื่อนที่ของสิ่งมีชีวิต เซนเซอร์ประเภทนี้มักถูกใช้ในระบบรักษาความปลอดภัยและหลอดไฟเปิด-ปิดอัตโนมัติ เป็นต้น

1.2.2 ตัวอย่างอุปกรณ์ส่งออก

ประเภท/ชื่อ	ภาพตัวอย่าง	คำอธิบาย
หลอดไฟ/ หลอดไฟ แอลอีดี	 <p>ที่มา: https://www.sparkfun.com/products/9590</p>	หลอดไฟแอลอีดีเป็นหลอดไฟประเภทหนึ่งที่มีขนาดเล็ก ใช้พลังงานน้อย และให้ความสว่างสูงเมื่อเทียบกับหลอดไฟประเภทอื่น ๆ หลอดไฟแอลอีดีขนาดเล็กดังรูปมักใช้ในการแสดงสถานะการทำงานของอุปกรณ์ให้กับผู้ใช้
หลอดไฟ/ หลอดไฟ แอลอีดีแบบ RGB	 <p>ที่มา: https://www.sparkfun.com/products/9264</p>	หลอดไฟแอลอีดีแบบ RGB เป็นหลอดไฟที่มีหลอดไฟสีแดง เขียวและน้ำเงินรวมอยู่ภายใน ทำให้สามารถเปลี่ยนสีได้ตามที่ต้องการโดยปรับความสว่างของแสงสีแดง เขียวและน้ำเงิน




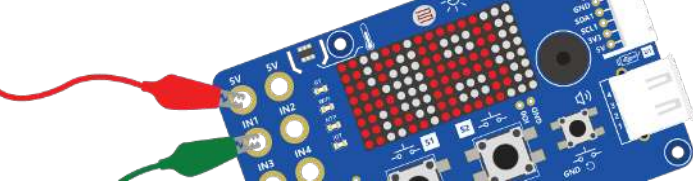
<p>จอแสดงผล/ จอแสดงผล ชนิดเซเว่น เซกเมนต์</p>	 <p>ที่มา: https://www.sparkfun.com/products/9483</p>	<p>จอแสดงผลชนิดเซเว่นเซกเมนต์เป็นจอแสดงผลที่นิยมใช้ ในการแสดงตัวเลขเช่นในลิฟต์โดยสาร ป้ายนับเวลาถอย หลังตามท้องถนนหรือนาฬิกาดิจิตอล เป็นต้น ทั้งนี้ เนื่องจากจอประเภทนี้มีราคาไม่แพง ความสว่างสูงและ สามารถมองเห็นได้ชัดเจนจากทุกมุมมองแม้ในบริเวณที่มี แดดจ้า</p>
<p>จอแสดงผล/ แอลซีดีชนิด แสดง ตัวอักษร แบบ 16 ตัวอักษร 2 บรรทัด</p>	 <p>ที่มา: https://www.sparkfun.com/products/9053</p>	<p>จอแอลซีดีชนิดนี้นิยมใช้ในการแสดงสถานะของอุปกรณ์ ต่าง ๆ เช่น เครื่องพิมพ์สามมิติ หรือเครื่องจักรในโรงงาน อุตสาหกรรม เป็นต้น เนื่องจากราคาไม่แพงและตัวอักษร มีขนาดใหญ่ แต่อาจไม่เหมาะกับงานที่ต้องการโชว์ ภาพกราฟิกต่าง ๆ</p>
<p>จอแสดงผล/ โอแอลอีดี ขนาดเล็ก</p>	 <p>ที่มา: https://www.sparkfun.com/products/13003</p>	<p>จอภาพแบบโอแอลอีดีเป็นจอภาพที่ได้รับความนิยม เนื่องจากใช้พลังงานน้อย มีความละเอียดสูงและสามารถ แสดงภาพกราฟิกได้ดี แต่มีข้อเสียคือราคาที่ค่อนข้างสูง กว่าจอภาพประเภทอื่น ๆ</p>
<p>มอเตอร์/ มอเตอร์ กระแสตรง</p>	 <p>ที่มา: https://www.sparkfun.com/products/11696</p>	<p>มอเตอร์กระแสตรงเป็นมอเตอร์ประเภทหนึ่งที่จะหมุน เมื่อจ่ายไฟไปยังตัวมอเตอร์และจะหมุนกลับทิศทางเมื่อ สลับด้านของสายไฟ</p>


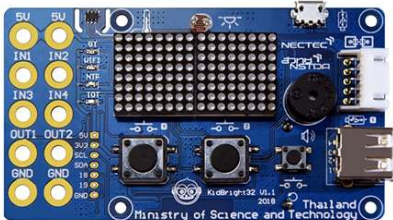


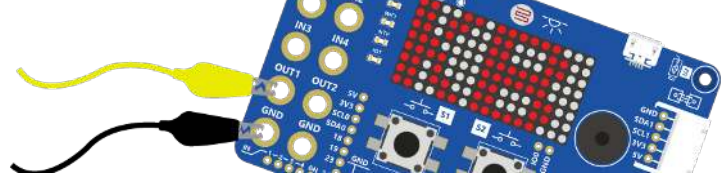
มอเตอร์/ เซอร์โว มอเตอร์	 <p>ที่มา: https://www.sparkfun.com/products/14760</p>	เซอร์โวมอเตอร์เป็นมอเตอร์ประเภทหนึ่งที่ใช้สามารถควบคุมทิศทางและองศาการหมุนได้อย่างแม่นยำ
ลำโพง/บัส เซอร์	 <p>ที่มา: https://www.sparkfun.com/products/7950</p>	บัสเซอร์เป็นอุปกรณ์ที่ใช้ในการบอกสถานะหรือส่งเสียงร้องเตือน
ลำโพง/ ลำโพงฟูล เรนจ์	 <p>ที่มา: https://www.sparkfun.com/products/9151</p>	ลำโพงฟูลเรนจ์เป็นลำโพงประเภทหนึ่งที่สามารถตอบสนองย่านความถี่เสียงได้ครอบคลุมทั้งสูง กลาง และต่ำ ลำโพงประเภทนี้นิยมใช้ในอุปกรณ์ที่ต้องการคุณภาพเสียงสูง เช่น ตู้กดขายได้

1.2.3 อุปกรณ์ประมวลผล

ประเภท	ภาพตัวอย่าง	คำอธิบาย
บอร์ด Arduino UNO	 <p>ที่มา: https://store.arduino.cc/usa/arduino-uno-rev3</p>	บอร์ด Arduino UNO เป็นบอร์ดไมโครคอนโทรลเลอร์แบบโอเพ่นซอร์สที่พัฒนาขึ้นโดย Arduino.cc โดยใช้ไมโครคอนโทรลเลอร์รุ่น ATmega328P จากบริษัท ไมโครชิพ เทคโนโลยี Arduino UNO เป็นบอร์ดไมโครคอนโทรลเลอร์ที่

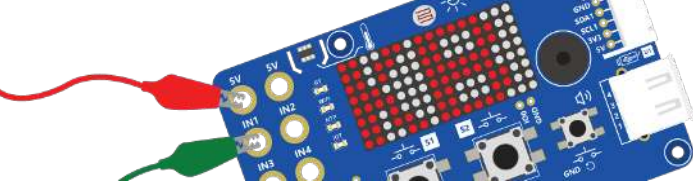




		ได้รับความนิยมอย่างมาก จึงทำให้มีไลบรารีรองรับมากมายและมีการนำไปดัดแปลงและผลิตบอร์ดเทียบเท่าออกมาหลากหลายรุ่น
บอร์ด NodeMCU V2.0	 <p>ที่มา: https://netpie.gitbooks.io/nodemcu-esp8266-on-netpie/content/chapter1.html</p>	บอร์ด NodeMCU V2.0 เป็นบอร์ดไมโครคอนโทรลเลอร์ที่ใช้โมดูล ESP-12E ภาย ใน โม ดูล ประกอบด้วยชิป ESP8266 ซึ่งเป็นไมโครคอนโทรลเลอร์พร้อมไวไฟในตัว ทำให้บอร์ดรุ่นนี้เป็นบอร์ดที่มีราคาถูกที่สุดในโลกรุ่นหนึ่งที่สามารถเชื่อมต่ออินเทอร์เน็ตผ่านไวไฟได้ในตัว
บอร์ด Kid Bright	 <p>ที่มา: https://www.kid-bright.org</p>	KidBright เป็นบอร์ดไมโครคอนโทรลเลอร์ที่พัฒนาขึ้นโดยทีมนักวิจัยจากเนคเทคและสวทช. ภาย ใน ป ระ ก อ บ ด์ วั ย ไมโครคอนโทรลเลอร์ ESP32 อุปกรณ์ส่งออกและตัวตรวจจับพื้นฐาน เพื่อให้เหมาะกับเด็กและเยาวชนในการเรียนการสอนโปรแกรมมิ่งในโรงเรียนทั่วประเทศ

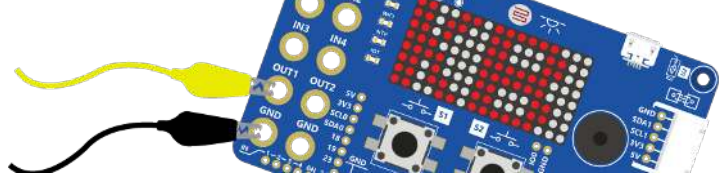


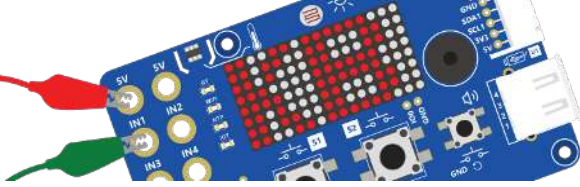
1.2.4 แหล่งพลังงาน

ประเภท	ภาพตัวอย่าง	คำอธิบาย
ถ่านคา บอนซิงค์/ ถ่านอัล คาไลน์		ถ่านคาร์บอนซิงค์/ถ่านอัลคาไลน์ หรือที่เรียกกันโดยทั่วไปว่า ถ่านไฟฉาย ถ่าน AA หรือถ่าน AAA เป็นแหล่งพลังงานที่ใช้ แพร่หลายในอุปกรณ์ เครื่องใช้ไฟฟ้าเช่นไฟฉายรีโมททีวี นาฬิกาแขวน รถบังคับวิทยุ เป็นต้น แหล่งพลังงานประเภทนี้มีความต่างศักย์ 1.5 โวลต์ต่อเซลล์ และมีจำหน่ายในหลากหลาย ขนาดตามประมาณประจุไฟฟ้า ตั้งแต่ AAAA (300 mAh), AAA, AA, C, D (10,000 mAh)
ถ่าน นิเกิล เมทัลไฮ ไดรด์ (Ni- MH)	 ที่มา: Ashley Pomeroy / CC BY-SA	ถ่านนิเกิลเมทัลไฮไดรด์ หรือ ที่เรียกกันโดยทั่วไปว่าถ่านชาร์จ เป็นแบตเตอรี่ที่ผลิตขึ้นเพื่อทดแทนการใช้ถ่านคาร์บอนซิงค์ หรือถ่านอัลคาไลน์มีข้อดีที่สามารถชาร์จซ้ำได้หลายร้อยครั้ง และมีประจุไฟฟ้าใกล้เคียงถ่านอัลคาไลน์ แต่มีความต่างศักย์ต่ำกว่าที่ 1.2 โวลต์



<p>แบตเตอรี่ ลิเทียม ไอออน</p>	 <p>ที่มา: https://www.sparkfun.com/products/13854</p>	<p>แบตเตอรี่ ลิเทียมไอออนเป็นแบตเตอรี่ที่ได้รับความนิยมในอุปกรณ์อิเล็กทรอนิกส์ในปัจจุบัน เช่น โทรศัพท์มือถือ หูฟังไร้สาย เครื่องเล่นเกม เป็นต้น เนื่องจากมีความหนาแน่นของพลังงานสูงทำให้มีน้ำหนักเบา ขนาดเล็ก และมีความจุสูงเมื่อเทียบกับแบตเตอรี่รูปแบบอื่น ๆ รวมถึงสามารถชาร์จซ้ำได้ตั้งแต่หลักร้อยจนถึงหนึ่งพันรอบ แหล่งพลังงานประเภทนี้มีความต่างศักย์ 3.7 โวลต์ต่อเซลล์ แต่ก็มีราคาสูงและอาจไหม้หรือระเบิดได้หากถูกกระแทกหรือถูกชาร์จอย่างไม่ถูกต้อง</p>
<p>แบตเตอรี่ แบบ ตะกั่ว-กรด</p>	 <p>ที่มา: https://www.bhphotovideo.com/c/product/1009424-REG/profoto_100222_lead_acid_battery_for.html</p>	<p>แบตเตอรี่แบบตะกั่ว-กรด เป็นแบตเตอรี่ที่ชาร์จซ้ำได้แบบแรกในโลก มีข้อดีที่ราคาถูกและสามารถจ่ายกระแสได้สูงในระยะเวลานานๆ ทำให้เหมาะอย่างมากในการใช้เป็นแบตเตอรี่รถยนต์ แบตเตอรี่ประเภทนี้มีข้อเสียที่มีความหนาแน่นของพลังงานต่ำ ทำให้มีขนาดใหญ่และหนักเมื่อเทียบกับแบตเตอรี่รูปแบบอื่น ๆ</p>





บทที่ 2

รู้จักกับอินเทอร์เน็ตของสรรพสิ่ง IoT

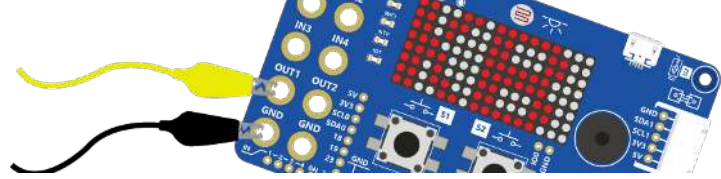
2.1 แนวคิดเรื่องอินเทอร์เน็ตของสรรพสิ่ง

อินเทอร์เน็ตของสรรพสิ่ง (Internet of Things) หรือ ไอโอที (IoT) หมายถึงการที่อุปกรณ์ต่าง ๆ ซึ่งหมายรวมถึงสิ่งของ พาหนะ หรือสิ่งปลูกสร้างใด ๆ ที่มีส่วนประกอบของระบบสมองกลฝังตัวและความสามารถในการเชื่อมต่อกับเครือข่าย สามารถเชื่อมโยงและแลกเปลี่ยนข้อมูลกันได้ผ่านเครือข่ายอินเทอร์เน็ต ทำให้เราสามารถสั่งการควบคุมและรับ-ส่งข้อมูลจากอุปกรณ์เหล่านั้นได้จากระยะไกล ในปัจจุบันมีการนำเทคโนโลยีนี้ไปประยุกต์ใช้ในด้านต่าง ๆ มากมายเช่น เครื่องซักผ้า-อบผ้าที่สามารถควบคุมและดูสถานะได้ผ่านโทรศัพท์มือถือจากระยะไกล ระบบบ้านอัจฉริยะ (Smart Home) ที่เจ้าของบ้านสามารถดูสถานะการเปิด-ปิด ประตู หน้าต่าง การใช้น้ำ ไฟ และสั่งเปิด-ปิดเครื่องใช้ไฟฟ้า ก๊อกน้ำ ฯลฯ ได้จากระยะไกล หรือระบบกริดไฟฟ้าและมิเตอร์ไฟฟ้าอัจฉริยะ (Smart Grid / Smart Electric Meter) ที่สามารถวัดปริมาณกระแสไฟฟ้าที่ใช้ในแต่ละบ้านและส่งค่าที่วัดได้ไปยังการไฟฟ้านครหลวงเพื่อออกใบเรียกเก็บค่าใช้ไฟฟ้า ทำให้สามารถลดค่าใช้จ่ายและลดการใช้แรงงานคนในการเดินจดปริมาณการใช้กระแสไฟฟ้าของบ้านแต่ละหลัง เป็นต้น



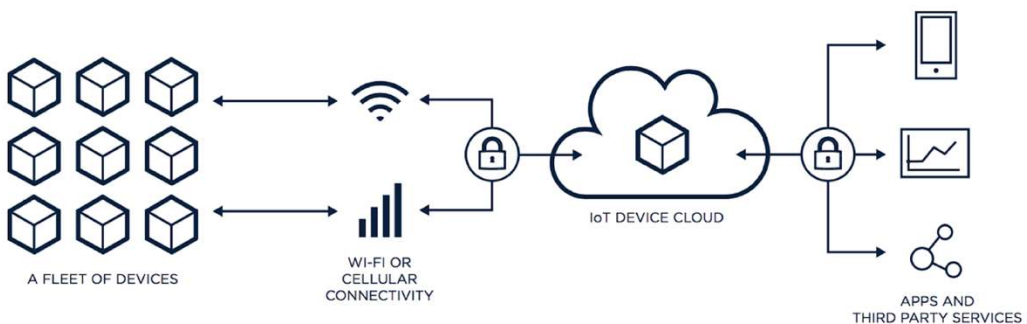
รูปภาพ 2-1 ตัวอย่างของระบบ Smart Home ที่มีจำหน่ายในท้องตลาดในปัจจุบัน

(ที่มา: <https://gadgets.ndtv.com/smart-home/news/samsung-smartthings-cam-smart-bulb-wi-fi-plug-launched-us-price-features-2058818>)



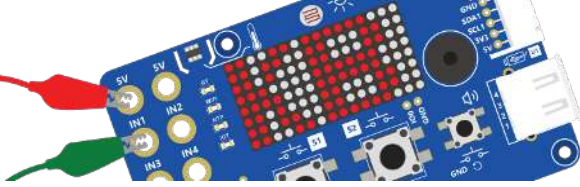
2.2 องค์ประกอบของระบบอินเทอร์เน็ตของสรรพสิ่ง

1. ตัวอุปกรณ์ (Devices หรือ Things) หมายถึงอุปกรณ์ต่าง ๆ เช่นระบบสมองกลฝังตัว บอร์ดไมโครคอนโทรลเลอร์ เซนเซอร์ คอมพิวเตอร์ โทรศัพท์มือถือ เป็นต้น ที่ทำหน้าที่ตรวจวัดรวบรวม แสดงผลข้อมูล และอาจมีการปฏิสัมพันธ์กับมนุษย์หรือสิ่งแวดล้อมรอบข้าง
2. การเชื่อมต่อ (Connectivity) หมายถึงระบบเครือข่ายและรูปแบบการส่งสัญญาณที่ทำให้ตัวอุปกรณ์สามารถเชื่อมต่อกันและเชื่อมต่อกับระบบคลาวด์ได้ตัวอย่างเช่น ระบบอินเทอร์เน็ต 3G และ 4G, เครือข่าย NB-IOT, Wi-Fi หรือ Bluetooth เป็นต้น
3. ระบบคลาวด์ (Cloud) หมายถึงระบบเครื่องแม่ข่ายที่มีความยืดหยุ่น รองรับการขยายตัว และมีรูปแบบการจ่ายเงินตามการใช้งานจริง ซึ่งช่วยให้ผู้พัฒนาสามารถออกแบบระบบที่รองรับการใช้งานจากผู้ใช้ที่มีจำนวนผันแปรได้โดยไม่ต้องลงทุนวางระบบเซิร์ฟเวอร์ขนาดใหญ่ ในกรณีของระบบอินเทอร์เน็ตของสรรพสิ่งมักใช้ระบบคลาวด์เป็นศูนย์กลางจัดเก็บและประมวลผลข้อมูล
4. แผงควบคุมและแสดงผล (Dashboard) หมายถึงหน้าจอสำหรับผู้ใช้ ซึ่งทำหน้าที่ในการแสดงผลข้อมูลที่เก็บรวบรวมจากแต่ละอุปกรณ์และอาจมีความสามารถในการสั่งการและควบคุมการทำงานของอุปกรณ์ต่าง ๆ ในระบบ โดยอาจอยู่ในรูปแบบของเว็บไซต์หรือแอปพลิเคชันในโทรศัพท์สมาร์ทโฟน เป็นต้น



รูปภาพ 2-2 ส่วนประกอบโดยภาพรวมของระบบอินเทอร์เน็ตของสรรพสิ่ง

(ที่มา: <https://www.quora.com/What-are-some-concerns-of-the-Internet-of-Things-IoT>)



2.3 ตัวอย่างผลิตภัณฑ์ในท้องตลาดภายใต้แนวคิดอินเทอร์เน็ตของสรรพสิ่ง

1. Samsung SmartThings เป็นระบบที่เชื่อมต่อเครื่องใช้ไฟฟ้าในบ้าน รวมถึงอุปกรณ์ในชุด SmartThings เช่น กล้องวงจรปิด ปลั๊กอัจฉริยะ ตัวตรวจจับการเคลื่อนไหว ตัวตรวจจับน้ำรั่ว ฯลฯ เพื่อบริหาร จัดการ และดูแลบ้านได้ผ่านแอปพลิเคชันบนสมาร์ตโฟนและสมาร์ตวอตช์ ตัวอย่างการใช้งานเช่นการสั่งเปิดไฟโดยอัตโนมัติเมื่อเปิดประตูบ้าน การตรวจจับน้ำรั่วซึม การตรวจสอบการลืมเปิดเครื่องใช้ไฟฟ้าทิ้งไว้ในขณะที่ไม่มีคนอยู่บ้าน

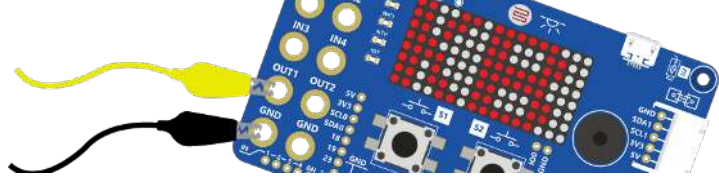


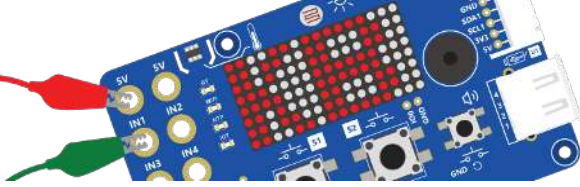
รูปภาพ 2-3 ชุดอุปกรณ์ Smart Home ยี่ห้อ Samsung Smart Things

2. Tile Bluetooth Tracker เป็นอุปกรณ์สำหรับค้นหาและติดตามสิ่งของโดยใช้สัญญาณ Bluetooth ร่วมกับเครือข่ายของผู้ติดตั้งแอปพลิเคชัน Tile เพื่อช่วยค้นหาสิ่งของที่อยู่นอกระยะการทำงานของเครือข่าย Bluetooth ในปัจจุบันมีผู้ใช้ Tile ใน 195 ประเทศ และมีจำนวนอุปกรณ์จำหน่ายไปแล้วกว่า 10 ล้านชิ้น



รูปภาพ 2-4 ชุดอุปกรณ์ติดตามสิ่งของรุ่น Tile Slim, Tile Mate และ Tile Sticker (จากซ้ายไปขวา)





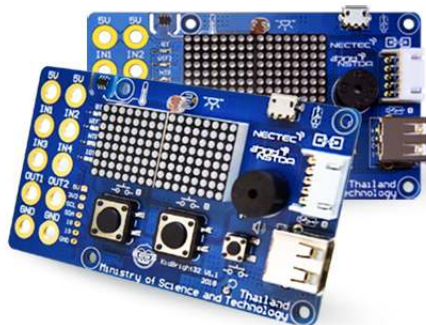
บทที่ 3

รู้จักกับบอร์ดสมองกลฝังตัว KidBright และอุปกรณ์บนบอร์ด

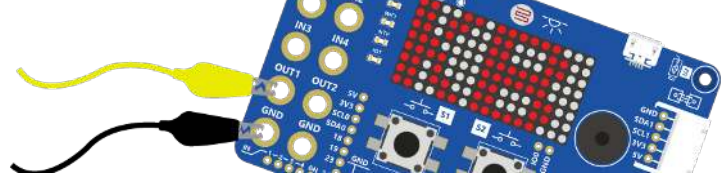
3.1 ประวัติโดยสังเขป

บอร์ดสมองกลฝังตัว KidBright เป็นบอร์ดไมโครคอนโทรลเลอร์ที่พัฒนาขึ้นโดยทีมนักวิจัยจากเนคเทคและสวทช. เพื่อให้เหมาะกับเด็กและเยาวชนในการเรียนการสอนโปรแกรมมิ่งในโรงเรียนทั่วประเทศ โดยมีจุดเริ่มต้นจากโครงการสื่อการสอนโปรแกรมมิ่งในโรงเรียน (Coding at School) นำโดยกระทรวงวิทยาศาสตร์และเทคโนโลยี และศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ (เนคเทค-สวทช.) ซึ่ง ได้จัดทำบอร์ดสมองกลฝังตัว KidBright รุ่น 1.3 จำนวน 200,000 ชุด แจกจ่ายให้โรงเรียนระดับมัธยมศึกษาในสังกัดของรัฐประมาณ 1,000 แห่งทั่วประเทศ

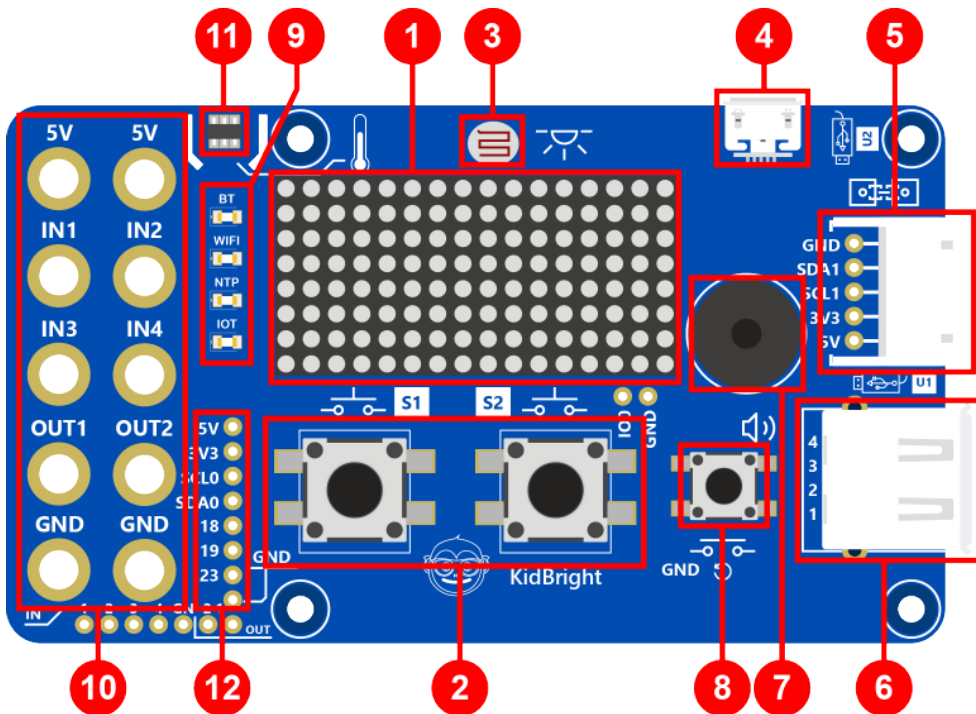
ภายในบอร์ด KidBright ประกอบด้วยไมโครคอนโทรลเลอร์รุ่น ESP32 ซึ่งรองรับการเชื่อมต่อไวไฟและบลูทูธเพื่อการสร้างโครงงานที่มีการเชื่อมต่อแบบไร้สาย รวมทั้งอุปกรณ์ส่งออกและตัวตรวจจับพื้นฐานซึ่งเพียงพอแก่การเรียนรู้และทดลองด้านระบบสมองกลและอินเทอร์เนตแห่งสรรพสิ่งเบื้องต้น ตัวบอร์ดได้ถูกออกแบบให้ใช้ร่วมกับโปรแกรม KidBright IDE ซึ่งเป็นโปรแกรมสำหรับสร้างชุดคำสั่งโดยการลากวาง (Drag and Drop) ซึ่งช่วยให้ผู้เรียนมุ่งเน้นไปที่กระบวนการคิดและการแก้ปัญหา มากกว่าการเรียนรู้เรื่องการพิมพ์ชุดคำสั่งด้วยตนเอง ในปัจจุบันบอร์ด KidBright ได้พัฒนามาถึงรุ่น 1.5 (ข้ามรุ่น 1.4) ซึ่งได้เพิ่มความสามารถในการเชื่อมต่อกับอุปกรณ์ภายนอกโดยยังคงความเข้ากันทั้งในด้านซอฟต์แวร์และฮาร์ดแวร์ได้กับบอร์ดรุ่นก่อนหน้า



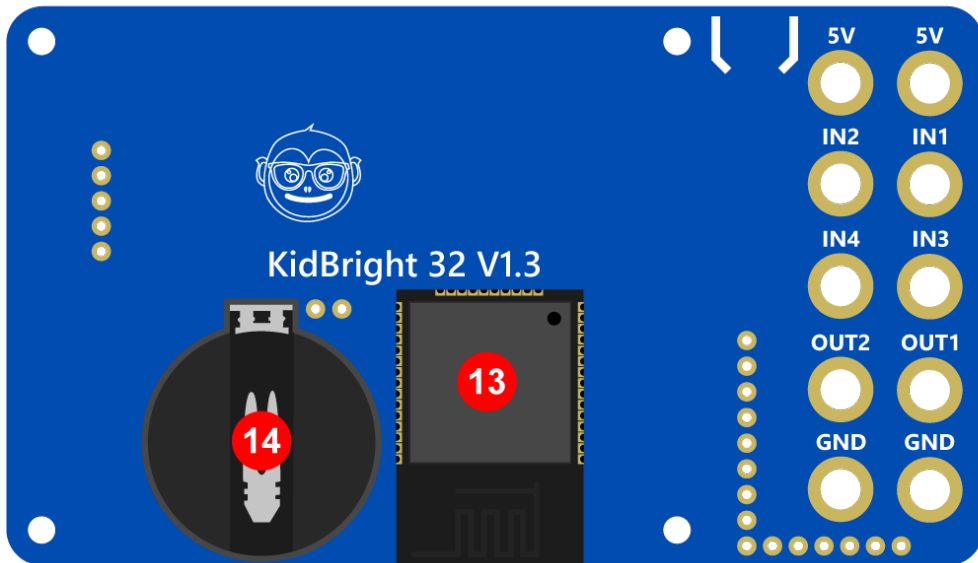
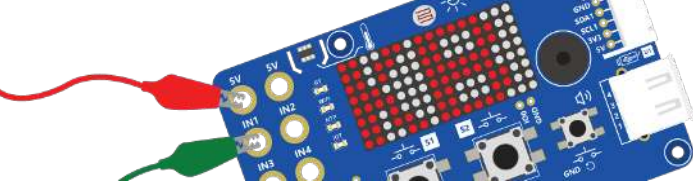
รูปภาพ 3-1 บอร์ดสมองกลฝังตัว KidBright



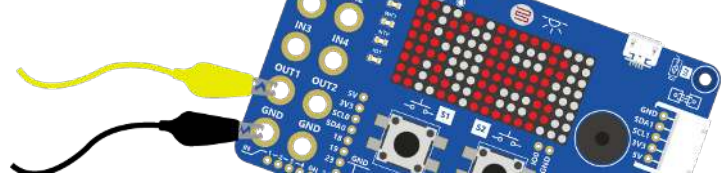
3.2 ส่วนประกอบของบอร์ดรุ่น 1.3



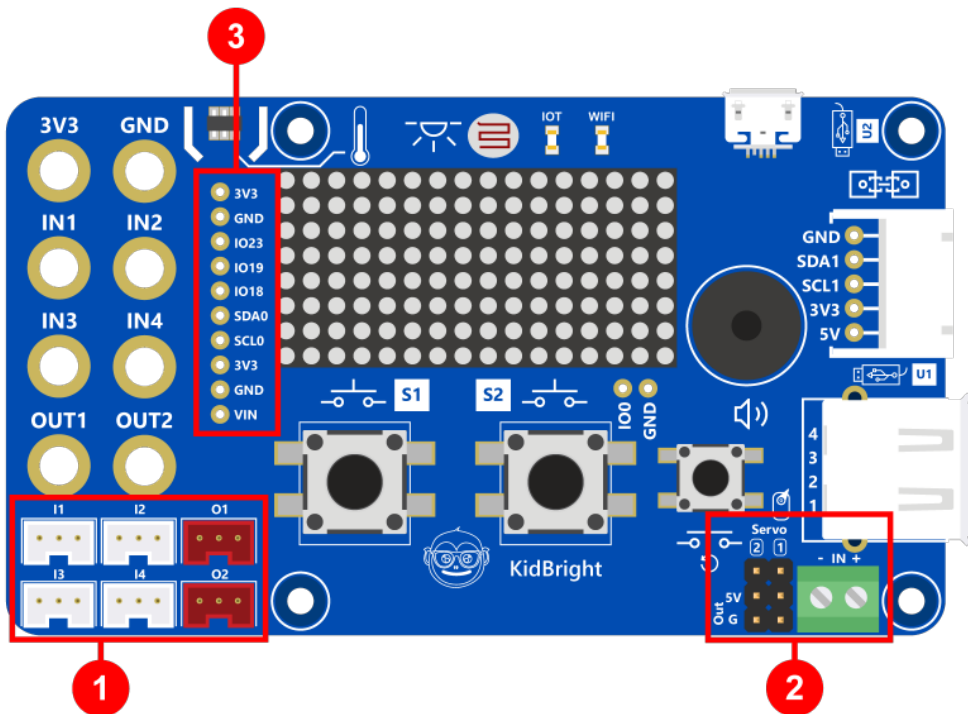
1. หน้าจอแสดงผล LED แบบ Dot Matrix ขนาด 16x8 ดวง
2. ปุ่มกดประเภทกดติด-ปล่อยดับ จำนวน 2 ตัว
3. เซนเซอร์ตรวจจับแสง สำหรับวัดความสว่างโดยรอบ
4. ช่องไมโครยูเอสบี สำหรับอัปโหลดโปรแกรมและจ่ายไฟให้กับตัวบอร์ด
5. ช่องเชื่อมต่อ KB Chain สำหรับเชื่อมต่อกับอุปกรณ์ภายนอกในรูปแบบ I²C
6. ช่องยูเอสบีชนิด A สำหรับเชื่อมต่อกับอุปกรณ์ที่ใช้ไฟเลี้ยงแบบ USB เพื่อควบคุมผ่านบอร์ด KidBright
7. ลำโพงเปียโซ สำหรับสร้างเสียงร้องหรือเสียงดนตรี
8. ปุ่มรีเซ็ต สำหรับเริ่มต้นการทำงานของโปรแกรมใหม่
9. หลอดไฟ LED สำหรับแสดงสถานะการทำงานของบอร์ด
10. ช่องเชื่อมต่ออุปกรณ์ภายนอก โดยใช้หัวปลั๊กแบบกล้วย
11. ตัวตรวจจับอุณหภูมิ สำหรับวัดอุณหภูมิปัจจุบัน
12. ช่องเชื่อมต่ออุปกรณ์ภายนอกแบบบัดกรี ระยะห่าง 2.54 มม.



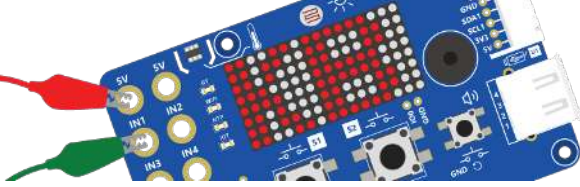
13. โมดูล ESP-WROOM-32 ซึ่งเป็นโมดูลไมโครคอนโทรลเลอร์ไร้สายซึ่งประกอบด้วยชิป ESP32
14. กระบะถ่านขนาด 3 โวลต์ชนิด CR2032 เพื่อรักษาเวลาในวงจรฐานเวลาจริงในขณะที่บอร์ดไม่ได้เชื่อมต่อไฟเลี้ยงจากภายนอก



3.3 ความแตกต่างระหว่างบอร์ดรุ่น 1.3 และ 1.5

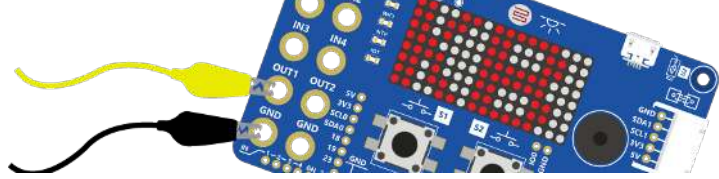


1. เพิ่มช่องเชื่อมต่อแบบ JST แบบ 3 พิน ระยะห่าง 2 มม. จำนวน 6 ช่องเชื่อมต่อกับช่องเชื่อมต่อแบบปลั๊กกล้วยเดิม ทั้งนี้ช่อง O1 และ O2 ไม่ได้เชื่อมต่อวงจรในรูปแบบ Open Drain เหมือนกับช่องเชื่อมต่อ OUT1 และ OUT2 และตัดช่องเชื่อมต่อไฟ 5V แบบปลั๊กกล้วยออกและเปลี่ยนเป็นช่องเชื่อมต่อไฟ 3.3V
2. เพิ่มช่องเชื่อมต่อ Servo โดยใช้หัวเชื่อมต่อ 3 พินระยะห่าง 2.54 มม. แบบมาตรฐานพร้อมช่องต่อไฟเลี้ยงเซอร์โวมอเตอร์นอก ทำให้สามารถเชื่อมต่อกับเซอร์โวมอเตอร์ได้สะดวกยิ่งขึ้น
3. ปรับตำแหน่งของจุดต่อพอร์ตแบบบัดกรีระยะห่าง 2.54 มม. เพื่อให้ง่ายต่อการใช้งาน

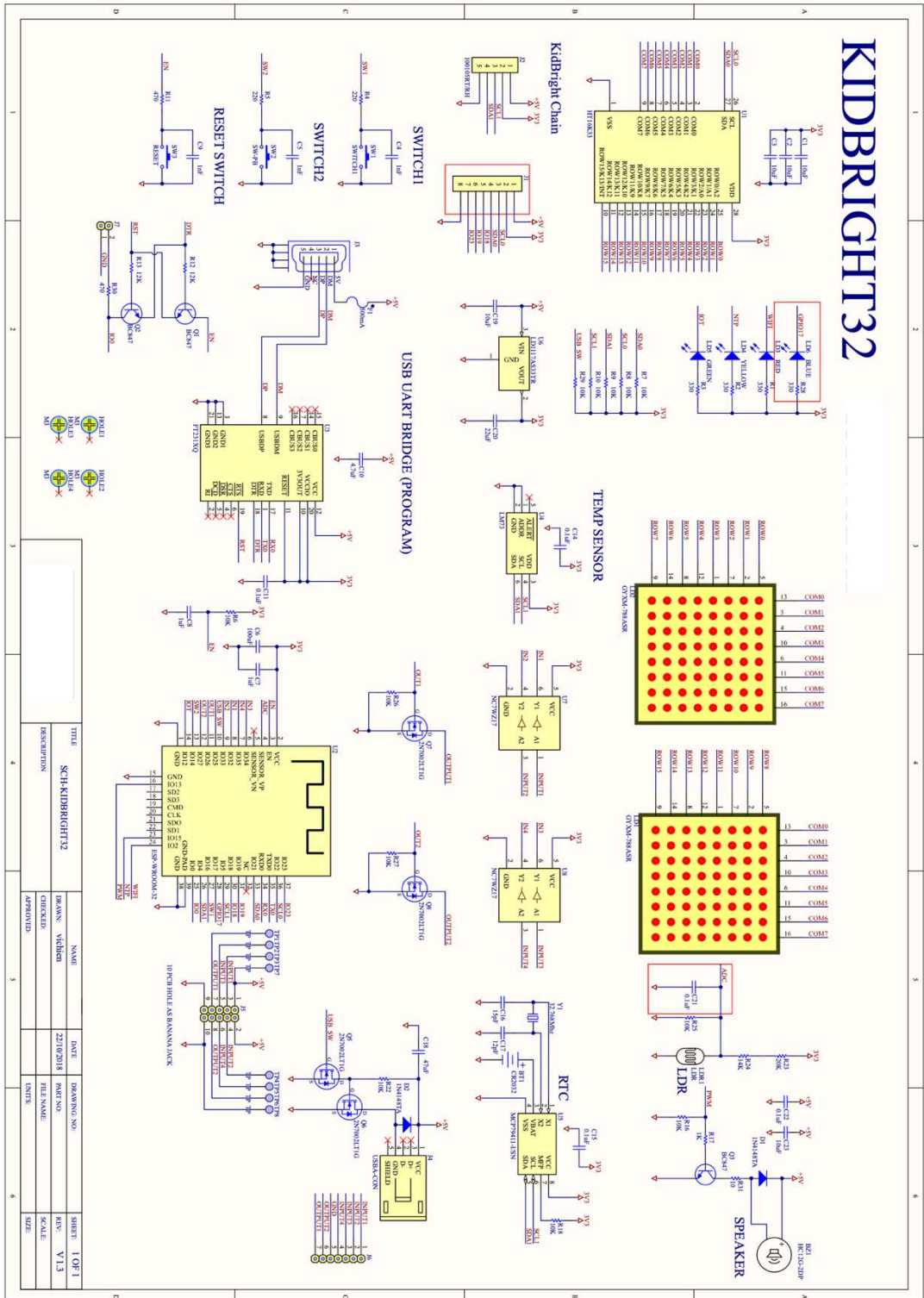


3.4 ข้อมูลจำเพาะ (Specification)

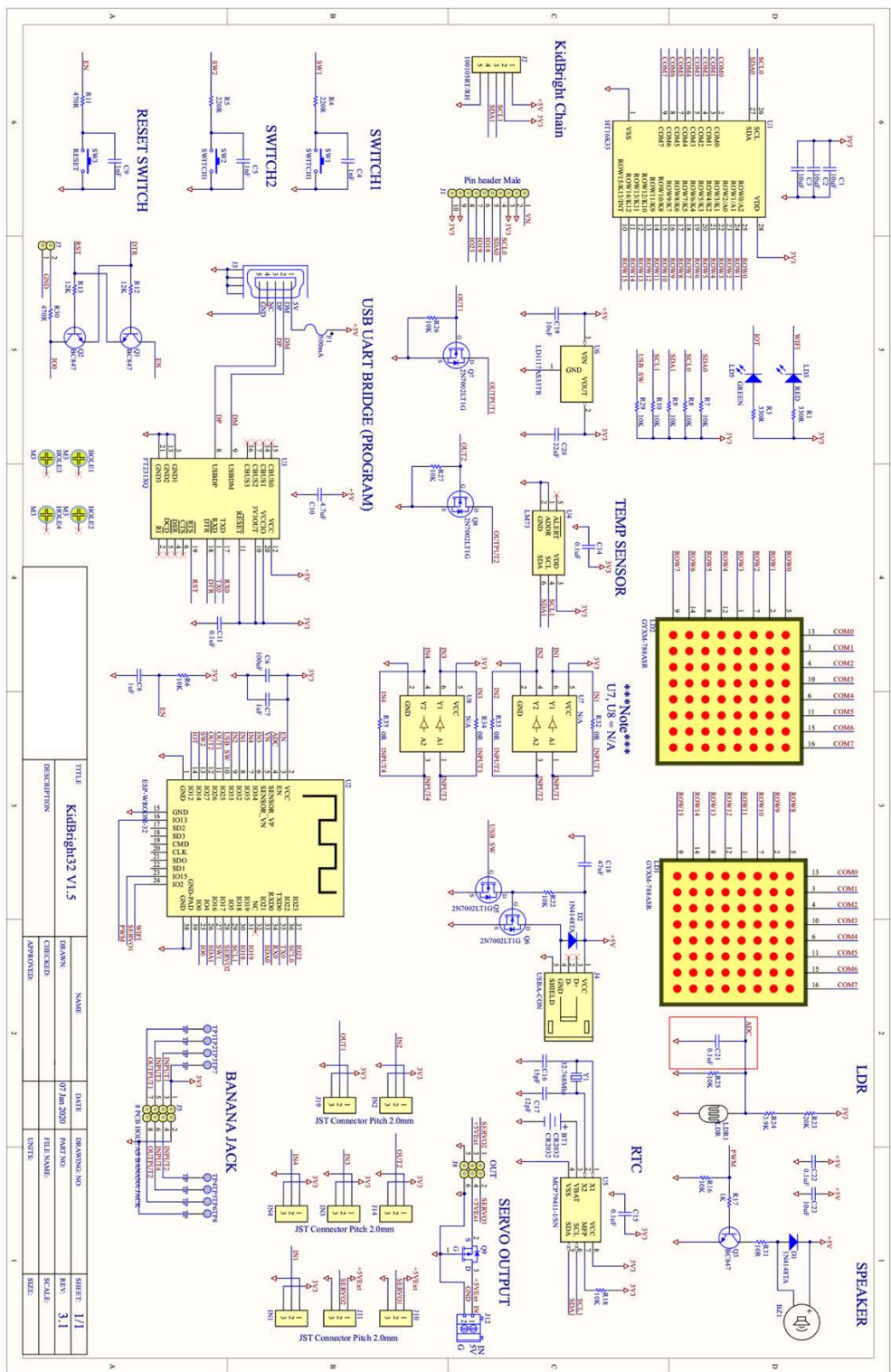
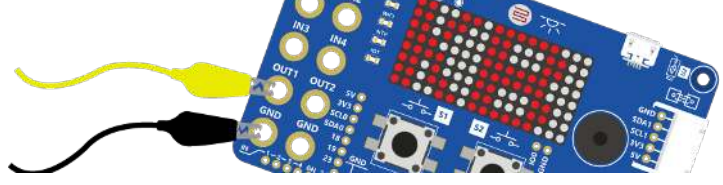
- ใช้โมดูล ESP-WROOM-32 ซึ่งภายในเป็นชิป ESP32 ซึ่งเป็นไมโครคอนโทรลเลอร์ที่ผลิตโดยบริษัท Espressif จากประเทศจีน มีความสามารถรองรับการเชื่อมต่อ Wi-Fi และ Bluetooth ทำงานที่สัญญาณนาฬิกา 260 MHz และมีหน่วยความจำภายใน 512 KB
- (เฉพาะรุ่น 1.3) มีช่องเชื่อมต่อไฟเลี้ยง 5V และ GND ในรูปแบบ Banana Plug จำนวนอย่างละ 2 ช่อง
- (เฉพาะรุ่น 1.5) มีช่องเชื่อมต่อไฟเลี้ยง 3.3V และ GND ในรูปแบบ Banana Plug จำนวนอย่างละ 1 ช่อง
- มีช่องเชื่อมต่ออินพุตพร้อมวงจรป้องกันแรงดันไฟเกิน (วงจรป้องกันมีเฉพาะในรุ่น 1.3) ในรูปแบบ Banana Plug จำนวน 4 ขาคือ IN1 IN2 IN3 IN4 เชื่อมต่อกับขา IO32-35 ของโมดูล ESP-WROOM-32 ตามลำดับ
- มีช่องเชื่อมต่อเอาต์พุตพร้อมมอสเฟตเพื่อขับโหลดภายนอกแบบ Open Drain ในรูปแบบ Banana Plug จำนวน 2 ขาคือ OUT1 OUT2 เชื่อมต่อกับ IO26 และ IO27 ของโมดูล ESP-WROOM-32 ตามลำดับ
- มีช่องเชื่อมต่อชนิด I2C จำนวน 2 ชุดได้แก่ SCL0/SDA0 ซึ่งใช้เชื่อมต่อกับไอซีควบคุมหลอดไฟ LED แบบ Dot Matrix และขาเชื่อมต่อแบบระยะ 2.54 มม. และ SCL1/SDA1 ซึ่งใช้สำหรับช่องเชื่อมต่อ KB Chain
- มีช่องเชื่อมต่ออินพุตและ/หรือเอาต์พุต 3 ช่องที่ขาเชื่อมต่อแบบระยะ 2.54 มม. คือขา 18 19 และ 23 ซึ่งเชื่อมต่อกับ IO18 IO19 และ IO23 ของโมดูล ESP-WROOM-32 ตามลำดับ
- มีวงจรควบคุมการจ่ายไฟเลี้ยงไปยังพอร์ต USB Type A บนบอร์ดเพื่อทำการเปิด-ปิดอุปกรณ์ภายนอกที่รองรับการเชื่อมต่อและใช้ไฟเลี้ยงจากพอร์ต USB
- มีไอซีแปลงสัญญาณ USB to Serial เบอร์ FT231 สำหรับติดต่อสื่อสารกับคอมพิวเตอร์
- มีไอซีวัดอุณหภูมิเบอร์ LM73 สามารถวัดอุณหภูมิได้ระหว่าง -40 ถึง +150 องศาเซลเซียส ด้วยความแม่นยำ ± 1 องศา ในย่านการวัด -10 ถึง +80 องศาเซลเซียส โดยเชื่อมต่อผ่านบัส SCL1/SDA1
- มีไอซีฐานเวลานาฬิกาจริงเบอร์ MCP7940N โดยเชื่อมต่อผ่านบัส SCL1/SDA1



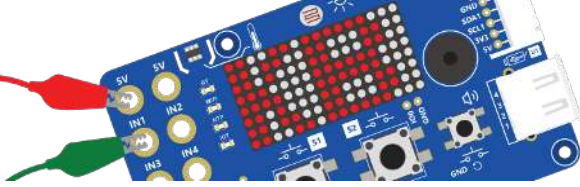
- มีตัวต้านทานปรับค่าตามแสง (LDR) เชื่อมต่อกับโมดูล ADC ของโมดูล ESP-WROOM-32 ผ่านขา IO36 (SENSOR_VP)
- มีลำโพงบัสเซอร์เบอร์ HC12G เชื่อมต่อผ่านขา IO13
- (เฉพาะรุ่น 1.5) มีช่องเชื่อมต่อมอเตอร์เซอร์โวจำนวน 2 ช่องพร้อมช่องต่อไฟเลี้ยงเซอร์โวภายนอก
- (เฉพาะรุ่น 1.5) มีช่องเชื่อมต่อแบบ JST 3 พิน ระยะห่าง 2 มม. จำนวน 6 ช่องเชื่อมต่อกับปลั๊กกล้วยเดิม



รูปภาพ 3-2 แผนผังวงจรบอร์ดสมองกลฝังตัว KidBright รุ่น 1.3



รูปภาพ 3-3 แผนผังวงจรบอร์ดสมองกลฝังตัว KidBright รุ่น 1.5



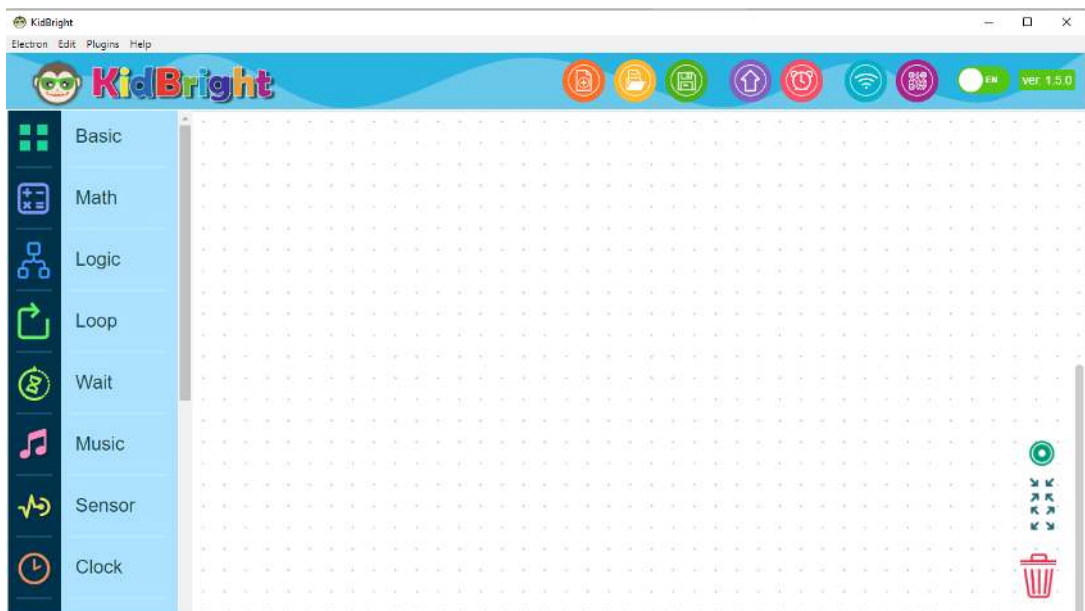
บทที่ 4

รู้จักกับ KidBright IDE

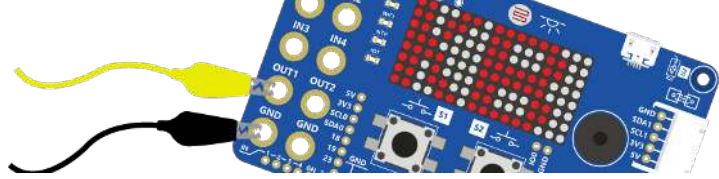
4.1 KidBright IDE คืออะไร?

KidBright IDE คือโปรแกรมที่ออกแบบมาโดยเฉพาะ สำหรับการใช้สร้างชุดคำสั่ง ให้กับบอร์ด KidBright ในรูปแบบของ block-structured programming ซึ่งเป็นโปรแกรมที่ถูกพัฒนาโดยศูนย์เทคโนโลยีอิเล็กทรอนิกส์ และคอมพิวเตอร์แห่งชาติ (NECTEC) เป็นการเขียนโปรแกรมโดยการลากกล่องคำสั่งพื้นฐานมาวางต่อกันเพื่อทำการเชื่อมโยงเป็นชุดคำสั่งหรือโปรแกรม จากนั้นโปรแกรมจะทำการแปลงภาษาที่เรียกว่าการ compile ให้อยู่ในรูปแบบของภาษาเครื่อง เพื่อให้ไมโครโพรเซสเซอร์ ESP32 บนบอร์ด KidBright สามารถทำงานได้ตามชุดคำสั่ง

การเขียนชุดคำสั่งในโปรแกรม KidBright จะเป็นการเขียนโปรแกรมที่ถูกออกแบบมาให้เข้าใจง่าย ไม่จำเป็นต้องเรียนรู้ภาษาเฉพาะทาง คำสั่งต่าง ๆ อยู่ในรูปแบบของกล่องข้อความ ซึ่งใช้ภาษาที่สามารถเข้าใจได้ง่ายไม่ซับซ้อน จึงไม่ทำให้เป็นอุปสรรคต่อการเรียนรู้ และเหมาะสำหรับการเรียนรู้ของผู้เริ่มต้น หรือผู้ที่ไม่มีพื้นฐานการเขียนโปรแกรมมาก่อน

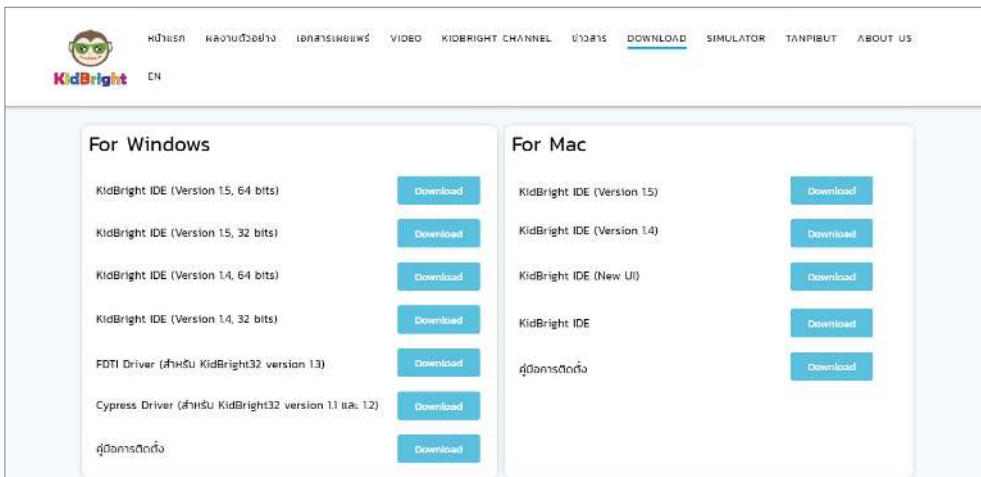


รูปภาพ 4-1 ภาพหน้าจอแสดงโปรแกรม KidBright IDE



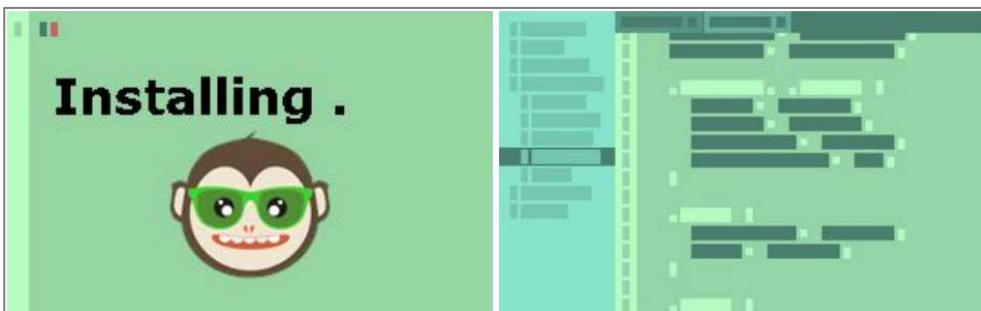
4.2 การติดตั้ง KidBright IDE

โปรแกรม KidBright IDE รองรับทั้งระบบปฏิบัติการ Windows และ Mac OS ซึ่งสามารถดาวน์โหลดได้จาก <https://www.kid-bright.org/kidbright/downloads> (ตัวอย่างในหนังสือเล่มนี้จะใช้ Version 1.6)

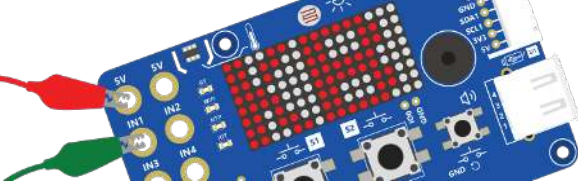


รูปภาพ 4-2 ภาพเว็บไซต์ดาวน์โหลด KidBright IDE

หลังจากทำการดาวน์โหลดเสร็จสิ้นเราจะได้ไฟล์ .zip ให้ทำการแตกไฟล์ และทำการกดที่ไฟล์ KidBright-Installer.exe เพื่อทำการ Install โปรแกรม KidBright IDE ระหว่างทำการ install จะเห็นหน้าต่างดังรูปภาพ 4-3 รอจนหน้าต่างหายไป และระบบทำการเปิดตัวโปรแกรมขึ้นมาอัตโนมัติ ถือว่าขั้นตอนการ install เสร็จสมบูรณ์



รูปภาพ 4-3 หน้าต่างระหว่างติดตั้งโปรแกรม KidBright IDE

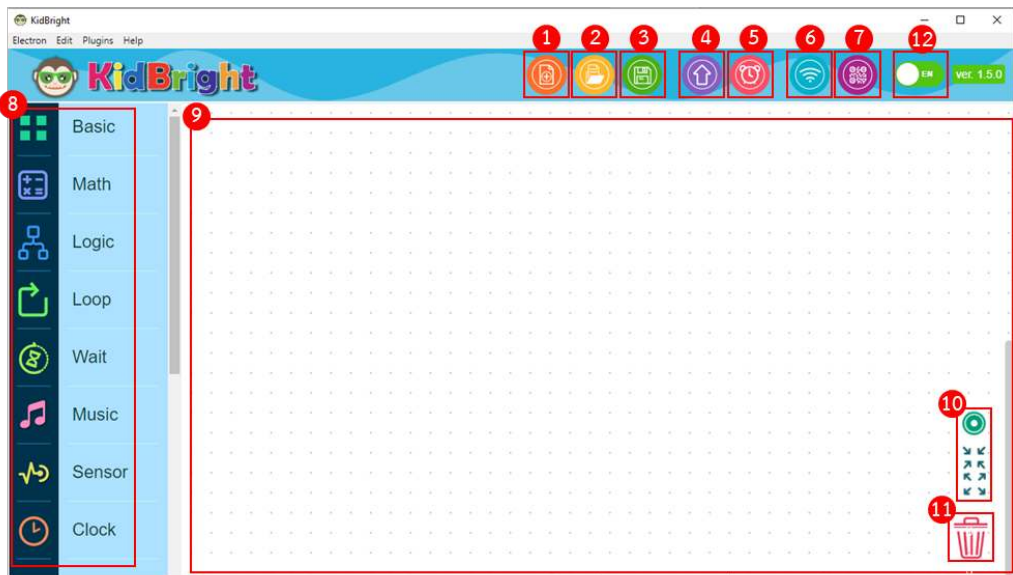


4.3 การควบคุมบอร์ด KidBright และอุปกรณ์บนบอร์ดด้วย KidBright IDE

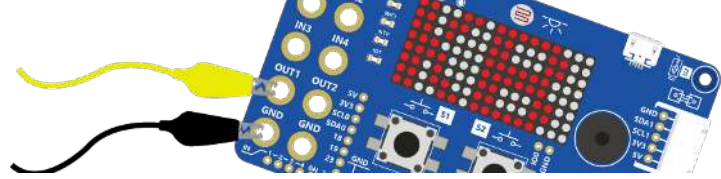
ในส่วนนี้จะสอนฟังก์ชันการใช้งาน block-structured ต่าง ๆ บนโปรแกรม KidBright IDE และวิธีการเบื้องต้นในการเขียนชุดคำสั่งควบคุมบอร์ด KidBright และอุปกรณ์ต่าง ๆ บนบอร์ด

4.3.1 ส่วนประกอบในหน้าต่างหลักของโปรแกรม

เมื่อเปิดโปรแกรมจะพบกับหน้าต่างหลักของโปรแกรม ซึ่งมีส่วนประกอบดังนี้



1. ปุ่มสำหรับสร้างไฟล์ใหม่ในการเขียนโปรแกรม
2. ปุ่มสำหรับเปิดไฟล์โปรแกรมที่บันทึกไว้ในคอมพิวเตอร์
3. ปุ่มสำหรับบันทึกโปรแกรมลงบนคอมพิวเตอร์
4. ปุ่มสำหรับ Upload โปรแกรมลงบนบอร์ด KidBright
5. ปุ่มสำหรับตั้งเวลาปัจจุบันให้กับชิป RTC บนบอร์ด KidBright
6. ปุ่มสำหรับตั้งค่า Wi-Fi ให้กับบอร์ด KidBright
7. ปุ่มสำหรับสร้าง QR Code Mac Address ของบอร์ด KidBright
8. หมวดหมู่ block ต่าง ๆ สำหรับใช้เขียนโปรแกรมควบคุมบอร์ด KidBright
9. พื้นที่สำหรับวาง block เพื่อสร้างชุดคำสั่ง
10. ปุ่มปรับมุมมองภาพบนพื้นที่สร้างชุดคำสั่ง
11. ปุ่มสำหรับลบ block ที่ไม่ต้องการใช้งานออกจากพื้นที่สร้างชุดคำสั่ง
12. ปุ่มสำหรับเปลี่ยนภาษา



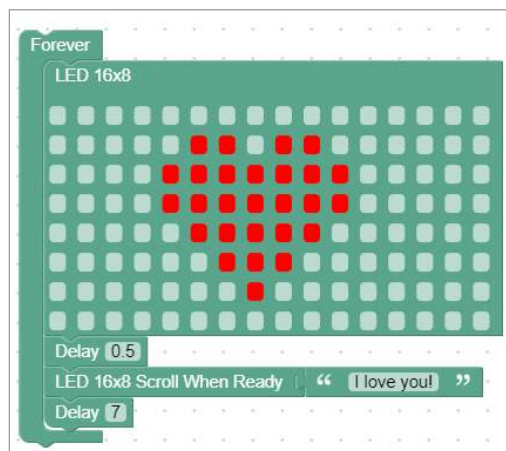
สำหรับหมวดหมู่ block-structured ต่าง ๆ ในข้อที่ 8 สามารถดูคำอธิบาย Block คำสั่งเพิ่มเติมได้จากเอกสารประกอบโครงการค่ายนักอิเล็กทรอนิกส์ด้านเทคโนโลยี IOT 2019 หน้าที 9 – 15 ซึ่งสามารถดาวน์โหลดได้จาก QR Code ด้านล่างนี้



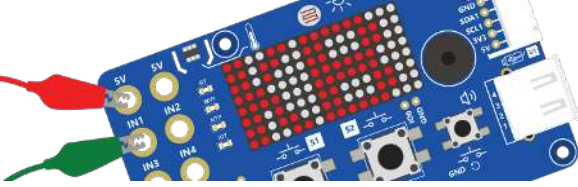
4.3.2 ตัวอย่างการควบคุม Dot Matrix LED

ในตัวอย่างนี้จะเป็นการสอนใช้งานแสดงผล OUTPUT บน Dot Matrix LED โดยในที่นี้เราจะทำการวาดรูปหัวใจ และแสดงคำว่า “I Love You!” ออกทางหน้าจอ Dot Matrix LED

1. ต่อบล็อกคำสั่ง Forever จาก Basic เพื่อให้บอร์ดวนรอบการทำงานซ้ำคำสั่งต่าง ๆ ที่อยู่ภายในคำสั่ง Forever
2. ต่อบล็อกคำสั่ง LED 16x8 ในการวาดรูปหัวใจ
3. ต่อบล็อกคำสั่ง Delay เพื่อหน่วงเวลาแสดงผลภาพรูปหัวใจ
4. ต่อบล็อกคำสั่ง LED 16x8 Scroll When Ready เพื่อแสดงข้อความเลื่อนบน Dot Matrix LED คำว่า “I Love You!”
5. ต่อบล็อกคำสั่ง Delay เพื่อหน่วงเวลาแสดงข้อความ



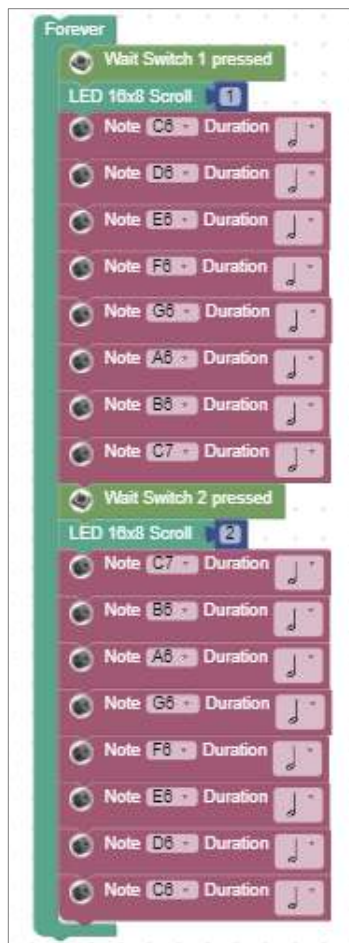
รูปภาพ 4-4 บล็อกคำสั่งการแสดงผล OUTPUT บน Dot Matrix LED



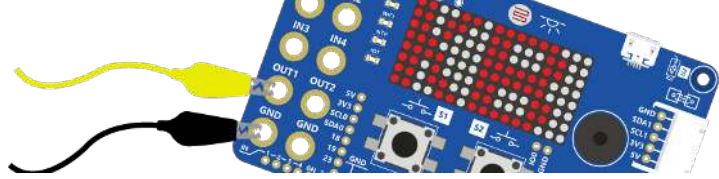
4.3.3 ตัวอย่างการใช้ปุ่ม Button และ เล่นเสียงโน้ตดนตรีด้วย Buzzer

ในตัวอย่างนี้เราจะทำการ กดปุ่ม S1 เพื่อให้บอร์ดแสดงเลข “1” บน Dot Matrix LED และเล่นโน้ตตั้งแต่ โดต่ำ ถึง โดสูง ด้วย Buzzer และเมื่อกดปุ่ม S2 จะทำการแสดงเลข “2” บน Dot Matrix LED และเล่นโน้ตตั้งแต่โดสูงถึงโดต่ำ ด้วย Buzzer

1. ต่อบล็อกคำสั่ง Forever จาก Basic เพื่อให้บอร์ดวนรอบการทำงานซ้ำคำสั่งต่าง ๆ ที่อยู่ภายในคำสั่ง Forever
2. ต่อบล็อกคำสั่ง Wait Switch 1, 2 pressed เพื่อรอให้มีการกดปุ่ม S1, S2 จึงทำงานต่อบล็อกในคำสั่งต่อไป
3. ต่อบล็อกคำสั่ง LED 16x8 Scroll เพื่อแสดงเลข “1”, “2” บน Dot Matrix LED
4. ต่อบล็อกคำสั่ง Note / Duration เพื่อเล่นเสียงโน้ตดนตรีที่ต้องการ



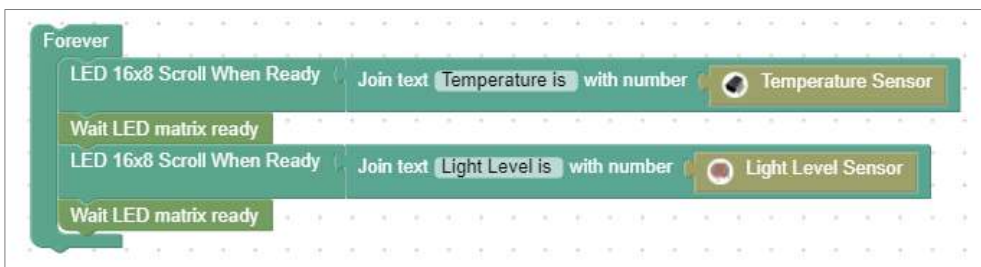
รูปภาพ 4-5 ภาพบล็อกคำสั่งการใช้ปุ่ม Button และ เล่นเสียงโน้ตดนตรีด้วย Buzzer



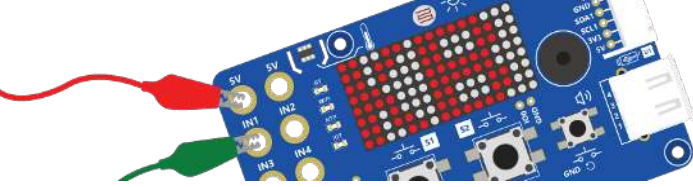
4.3.4 ตัวอย่างการอ่านค่าจาก Temperature Sensor และ LDR

ในตัวอย่างนี้เราจะทำการ แสดงค่า อุณหภูมิ และความสว่างของแสง ขึ้นเป็นข้อความเลื่อนบน Dot Matrix LED

1. ต่อบล็อกคำสั่ง Forever จาก Basic เพื่อให้บอร์ดวนรอบการทำงานซ้ำคำสั่งต่าง ๆ ที่อยู่ภายในคำสั่ง Forever
2. ต่อบล็อกคำสั่ง LED 16x8 Scroll When Ready เพื่อแสดงค่าอุณหภูมิ และความสว่างบน Dot Matrix LED โดยเราสามารถอ่านค่าอุณหภูมิได้จาก บล็อกคำสั่ง Temperature Sensor และค่าความสว่างได้จาก Light Level Sensor นอกจากนี้ยังสามารถเขียนข้อความร่วมกับค่าจาก Sensor ได้ด้วยบล็อกคำสั่ง Join text...with number
3. ต่อบล็อกคำสั่ง Wait LED matrix ready เพื่อรอให้แสดงข้อความ และค่าจาก Sensor จนครบถ้วนจึงจะเริ่มแสดงข้อความถัดไป



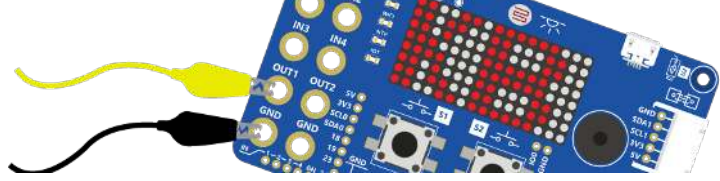
รูปภาพ 4-6 ภาพโปรแกรมสั่งการแสดงผลค่าอุณหภูมิ และ ค่าความสว่างบน Dot Matrix LED

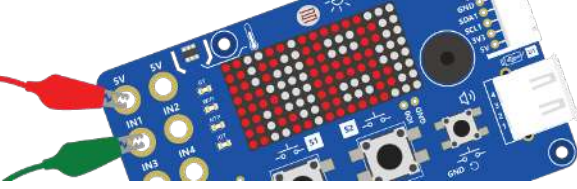


4.4 แบบฝึกหัดและโจทย

ในส่วนนี้จะจะเป็นแบบฝึกหัดเพื่อทำความเข้าใจกับโปรแกรม KidBright IDE ซึ่งจะเป็นแบบฝึกหัดและโจทยที่มีความง่ายไปจนยาก

1. สร้างโปรแกรมที่นำค่าจากเซนเซอร์วัดอุณหภูมิมาแสดงผลบนหน้าจอ Dot Matrix LED และหากอุณหภูมิเกิน 35 องศาเซลเซียส ให้ทำการส่งเสียงเตือนผ่านบัสเซอร์
2. สร้างโปรแกรมที่เมื่อกดปุ่ม S1 และจะเล่นเสียงตัวโน้ต ตามค่าความเข้มแสงที่เปลี่ยนแปลงไป โดยยังมีความสว่างมากให้เล่นโน้ตเสียงสูงและเมื่อความสว่างลดลงให้เล่นโน้ตเสียงต่ำ
3. สร้างเกมทดสอบความเร็วเมื่อเริ่มเกมใหม่ ให้นำมือไปวางปิด LDR ไว้เพื่อเป็นการรีเซ็ตค่า จากนั้นให้ยกมือขึ้นเพื่อกลับมากดปุ่ม S1 หรือ S2 ให้เร็วที่สุดตามผังหน้าจอ Dot Matrix LED ที่แสดงสีไฟขึ้นแบบสุ่ม โดยเกมจะจบลงก็ต่อเมื่อผู้เล่นกดผิดปุ่มไม่ตรงกับผังของสีไฟที่แสดงบนจอหรือ กดไม่ทันภายในเวลาครึ่งวินาที และเมื่อเกมจบลงให้ทำการส่งเสียงเตือนออกทางลำโพงบัสเซอร์เมื่อจบเกม
4. สร้างเกมกตโดยมีผู้เล่น 2 คนโดยให้ผู้เล่นกดปุ่ม S1 และ S2 แข่งกันผู้เล่นคนใดกดถึง 20 ครั้งก่อนจะเป็นฝ่ายชนะโดยก่อนเริ่มเกมหรือเมื่อเล่นจบแล้วจะต้องทำการรีเซ็ตเกมใหม่ โดยการกด S1 และ S2 ค้างไว้พร้อมกันจนมีเสียงแจ้งเตือนจากบัสเซอร์และหากฝ่ายใดเป็นผู้ชนะให้ขึ้นเครื่องหมายถูกบนหน้าจอ Dot Matrix LED ของฝ่ายนั้นและมีเสียงร้องแจ้งเตือนเมื่อจบเกม
5. สร้างนาฬิกาปลุกที่สามารถตั้งเวลาปลุกได้และจะไม่หยุดส่งเสียงหากไม่มีการเปิดไฟในห้องนอนและทำการกดปุ่ม S1 เพื่อปิดเสียง





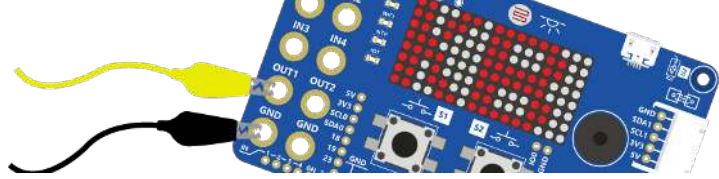
บทที่ 5

การเชื่อมต่อภายนอกเข้ากับบอร์ด KidBright

การเชื่อมต่ออุปกรณ์ภายนอกเข้ากับบอร์ด KidBright ผู้พัฒนาจำเป็นต้องทราบประเภทของสัญญาณและโปรโตคอลที่อุปกรณ์นั้น ๆ รองรับว่าเป็นรูปแบบใด อาทิ ดิจิตอล อนาล็อก I²C SPI หรือ UART โดยสามารถศึกษาเพิ่มเติมได้จากคู่มือหรือ Datasheet ของอุปกรณ์รุ่นนั้น ๆ ที่ได้รับมาจากผู้ผลิต ทั้งนี้ผู้พัฒนาสามารถสังเกตได้ด้วยหลักการเบื้องต้นดังนี้

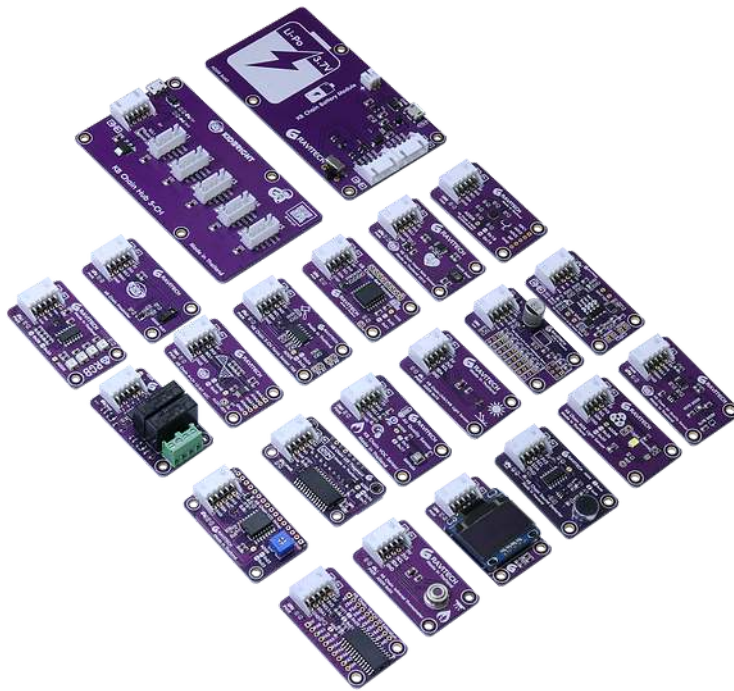
1. ในกรณีที่อุปกรณ์ใช้สายสัญญาณเพียงเส้นเดียวและมีการทำงานเป็นสองสถานะคือทำงานหรือไม่งาน ตรวจพบหรือตรวจไม่พบ สามารถอนุมานได้ว่าเป็นอุปกรณ์ประเภทดิจิตอลซึ่งสามารถเชื่อมต่อใช้งานกับบอร์ด KidBright และ KidBright IDE ได้โดยไม่ต้องติดตั้งปลั๊กอินเพิ่มเติม
2. ในกรณีที่อุปกรณ์ใช้สายสัญญาณเพียงเส้นเดียวและรูปแบบข้อมูลมีความต่อเนื่องเช่นค่าอุณหภูมิเป็นตัวเลข ระยะทางในหน่วยซม. หรือความดังเสียง สามารถอนุมานได้ว่าเป็นอุปกรณ์อนาล็อก ซึ่งสามารถเชื่อมต่อกับบอร์ด KidBright รุ่น 1.5 ขึ้นไปได้เท่านั้น
3. ในกรณีที่ใช้สายสัญญาณ 2 เส้นชื่อ SDA และ SCL จะถือว่าเป็นอุปกรณ์ประเภท I2C สามารถเชื่อมต่อกับบอร์ด KidBright ได้โดยไม่ต้องติดตั้งปลั๊กอินเพิ่มเติมใน KidBright IDE สำหรับแต่ละอุปกรณ์
4. ในกรณีที่ใช้สายสัญญาณ 4 เส้นชื่อ MISO, MOSI, SCK และ CS จะถือว่าเป็นอุปกรณ์ประเภท SPI ซึ่งต้องบัดกรีเพื่อเชื่อมต่อและยังไม่มีบล็อกรองรับในโปรแกรม KidBright IDE
5. ในกรณีที่ใช้สายสัญญาณ 2 เส้นชื่อ TX และ RX จะถือว่าเป็นอุปกรณ์ประเภท UART ซึ่งในปัจจุบันยังไม่มีบล็อกรองรับใน KidBright IDE จึงสามารถใช้งานได้โดยการเขียนโปรแกรมผ่าน Arduino IDE เท่านั้น

ทั้งนี้บางอุปกรณ์อาจรองรับการเชื่อมต่อมากกว่า 1 รูปแบบ หรืออาจใช้สายสัญญาณมากกว่า 1 เส้นก็ได้ตัวอย่างเช่น เซนเซอร์วัดระยะทางแบบอัลตราโซนิกรุ่น HC-SR04 และเซนเซอร์วัดอุณหภูมิรุ่น DHT11 ซึ่งสามารถวัดระยะทางและอุณหภูมิได้โดยใช้สายสัญญาณดิจิตอล 2 เส้นด้วยโปรโตคอลเฉพาะซึ่งจำเป็นต้องติดตั้งบล็อกเพิ่มเติมเพื่อใช้งานกับ KidBright IDE

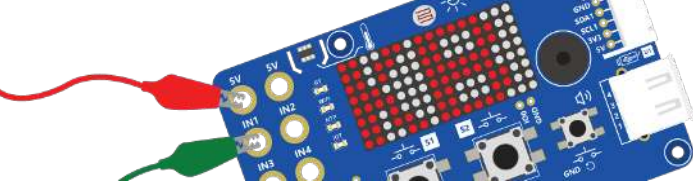


สำหรับบอร์ด KidBright รุ่น 1.3 จะรองรับการเชื่อมต่ออุปกรณ์ภายนอกได้ 4 รูปแบบคือ

1. ใช้สายเชื่อมต่อแบบปลั๊กกล้วย (Banana Plug) เชื่อมกับช่องเชื่อมต่อ IN1-4 หรือ OUT1-2
2. ใช้สายเชื่อมต่อที่มีปลายสายเป็นคลิปจระเข้เชื่อมต่อกับช่องเชื่อมต่อ IN1-4 หรือ OUT1-2
3. ใช้สายจัมเปอร์ขนาด 2.54 มม. เชื่อมต่อกับช่องเชื่อมต่อหมายเลข 18, 19 และ 23 บริเวณด้านซ้ายของปุ่มกด S1
4. ใช้สายเชื่อมต่อ KB Chain (JST ระยะห่าง 2 มม. ชนิด 5 พิน) เชื่อมต่อกับช่องเชื่อมต่อ KB Chain โดยในขณะนี้ทางบริษัท กราวีเทคโนโลยี (ไทยแลนด์) จำกัด ได้พัฒนาบอร์ดเชื่อมต่อมากกว่า 22 แบบสามารถศึกษารายละเอียดเพิ่มเติมได้จาก www.kidbright.io/shop



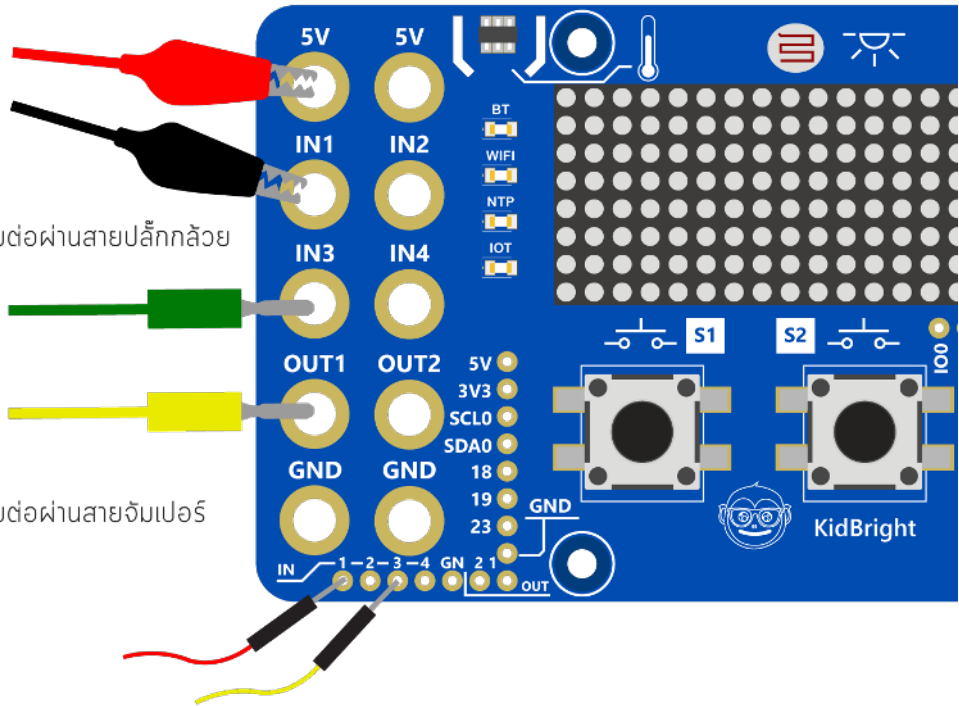
รูปภาพ 5-1 ภาพตัวอย่างอุปกรณ์ต่อพ่วงในรูปแบบ KB Chain
(ที่มา: <https://www.kidbright.io/product-page/master-chain-kit>)



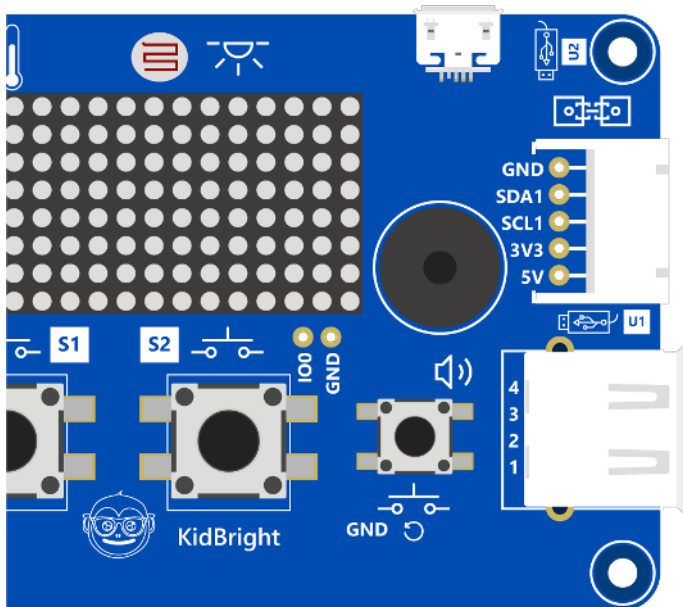
1 เชื่อมต่อผ่านคลิปหนีบปากจระเข้

2 เชื่อมต่อผ่านสายปลั๊กกล้วย

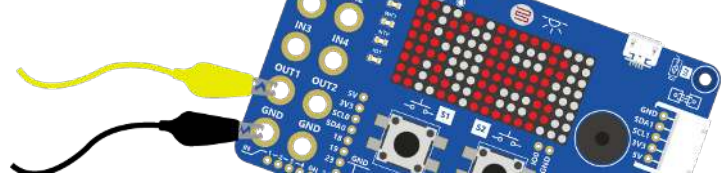
3 เชื่อมต่อผ่านสายจัมเปอร์



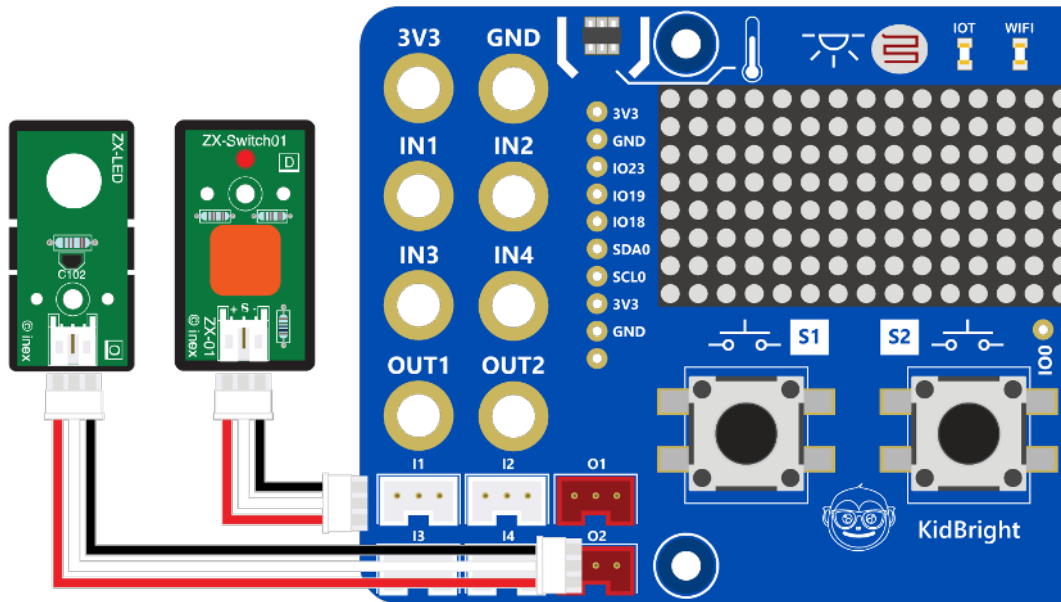
4 เชื่อมต่อผ่านพอร์ต KB Chain



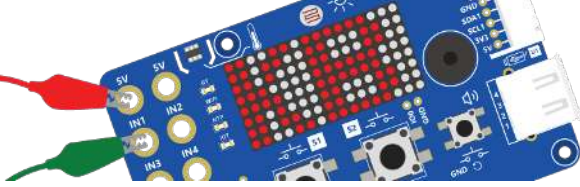
รูปภาพ 5-2 ภาพตัวอย่างการเชื่อมต่ออุปกรณ์ภายนอกเข้ากับบอร์ด Kid Bright รุ่น 1.3



สำหรับบอร์ด KidBright รุ่น 1.5 ได้เพิ่มช่องเชื่อมต่อในรูปแบบ JST ระยะ 2.0 มม. จำนวน 6 ช่องเพื่อช่วยให้สามารถเชื่อมต่อกับอุปกรณ์ภายนอกที่ใช้ช่องเชื่อมต่อแบบ JST ได้ง่ายยิ่งขึ้น โดยใช้สาย JST แบบ 3 พินเพียงเส้นเดียว โดยในขณะนี้ผู้พัฒนาสามารถหาซื้ออุปกรณ์ที่ใช้พอร์ตเชื่อมต่อ JST ได้จากบริษัท อินโนเวטיפ เอ็กเพอริเมนต์ จำกัดหรือ www.inex.co.th



รูปภาพ 5-3 ภาพตัวอย่างการเชื่อมต่ออุปกรณ์ภายนอกเข้ากับบอร์ด Kid Bright รุ่น 1.5

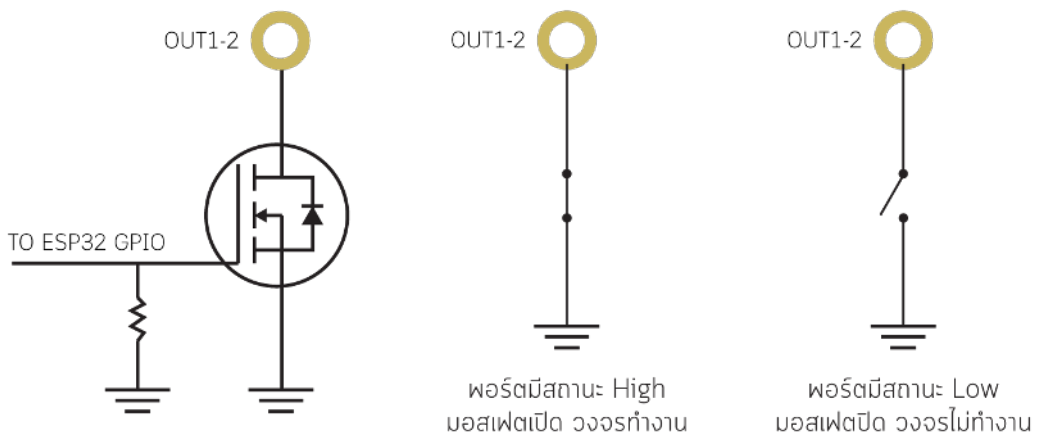


5.1 การเชื่อมต่ออุปกรณ์ดิจิทัล

ผู้พัฒนาสามารถเชื่อมต่ออุปกรณ์ดิจิทัลประเภทเอาต์พุตเช่นหลอดไฟ LED, ลำโพงบัซเซอร์, มอเตอร์ขนาดเล็ก ฯลฯ เข้าสู่ช่องเชื่อมต่อ OUT1, OUT2 หรือขา 18, 19 และ 23 ได้

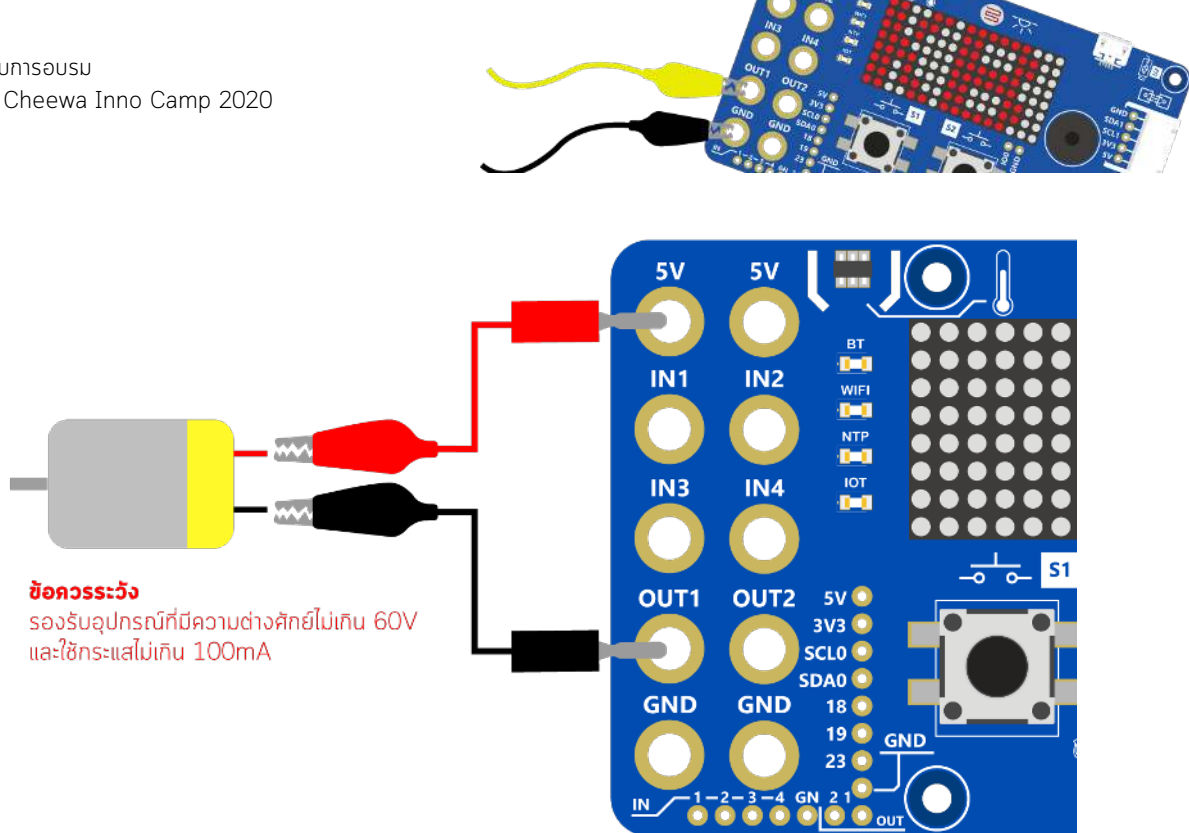
ทั้งนี้ช่องเชื่อมต่อ OUT1 และ OUT2 ในรูปแบบปลั๊กกล้วยได้มีการออกแบบวงจรในรูปแบบ Open Drain เพื่อป้องกันความเสียหายในกรณีที่ต่อวงจรไม่ถูกต้องและเพื่อให้รองรับอุปกรณ์ที่มีความต่างศักย์ไม่เกิน 60 โวลต์และใช้กระแสสูงสุดไม่เกิน 100 มิลลิแอมป์ได้โดยตรง การทำงานของวงจรสามารถอธิบายอย่างง่ายคือสามารถคุมให้อยู่ใน 2 สถานะ ดังนี้

1. เมื่อโปรแกรมให้พอร์ตมีสถานะเป็น High มอสเฟตจะเปิด ทำให้วงจรปิดและกระแสไหลลงกราวด์ได้ครบวงจรและอุปกรณ์ไฟฟ้าทำงาน
2. เมื่อโปรแกรมให้พอร์ตมีสถานะเป็น Low มอสเฟตจะปิด ทำให้วงจรเปิดและไม่มีกระแสไหลลงกราวด์ ทำให้อุปกรณ์ไฟฟ้าไม่ทำงาน

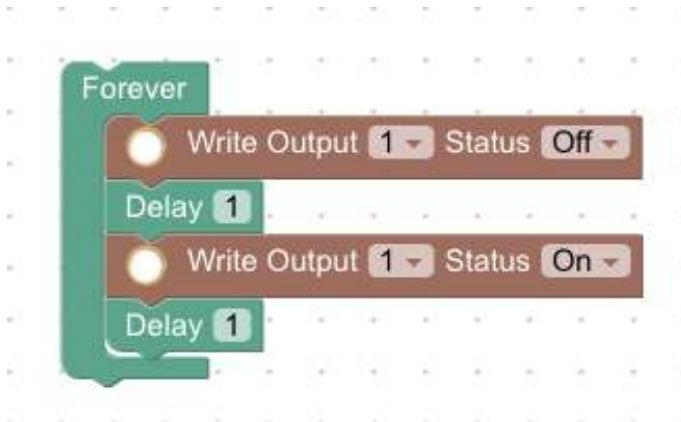


รูปภาพ 5-4 ภาพอธิบายการเชื่อมต่อวงจรของขาเอาต์พุตแบบ Open Drain

การออกแบบเช่นนี้ไม่สามารถจ่ายกระแสออกจากพอร์ตเพื่อเลี้ยงอุปกรณ์ภายนอกได้เหมือนพอร์ตทั่วไป ในการเชื่อมต่อวงจรจึงต้องทำการเชื่อมต่อขั้วบวกของอุปกรณ์เข้ากับแหล่งจ่ายไฟและขั้วลบเข้ากับช่องเชื่อมต่อ OUT1-2 ดังรูป

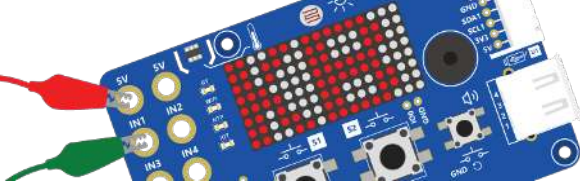


รูปภาพ 5-5 ตัวอย่างการเชื่อมต่อมอเตอร์ขนาดเล็กเข้ากับบอร์ด Kid Bright



รูปภาพ 5-6 ตัวอย่างโปรแกรมเพื่อเปิดและปิดอุปกรณ์ที่ต่อพ่วงกับพอร์ต OUT1 ทุก ๆ 1 วินาที

สำหรับช่องเชื่อมต่อหมายเลข 18, 19 และ 23 ทั้งในบอร์ดรุ่น 1.3 และ 1.5 รวมถึงช่องเชื่อมต่อ JST O1 และ O2 บนบอร์ด Kid Bright รุ่น 1.5 สามารถควบคุมให้เปิด (จ่ายไฟหรือลอจิก 1) หรือปิด (เชื่อมลงกราวด์หรือลอจิก 0) ได้ตามปกติ โดยสามารถจ่ายหรือรับกระแสได้ไม่เกิน 20 มิลลิแอมป์ (โดยประมาณ) เท่านั้น



ทั้งนี้การใช้งานพอร์ต 18, 19 และ 23 ผู้พัฒนาจะต้องทำการบัดกรีหัวเชื่อมต่อด้วยตนเอง โดยสามารถศึกษาวิธีการบัดกรีหัวเชื่อมต่อได้จากช่องทาง Youtube โดยคุณ rungsee jar (<https://youtu.be/KIOatjMtlcg>) และติดตั้งปลั๊กอินเพิ่มเติมเพื่อใช้งานบล็อกควบคุมพอร์ต ดังกล่าวตามขั้นตอนดังนี้

1. ดาวน์โหลดปลั๊กอิน OpenKB เวอร์ชันล่าสุดโดย INEX จากเว็บไซต์

<https://store.kidbright.info/plugin/20/OpenKB>

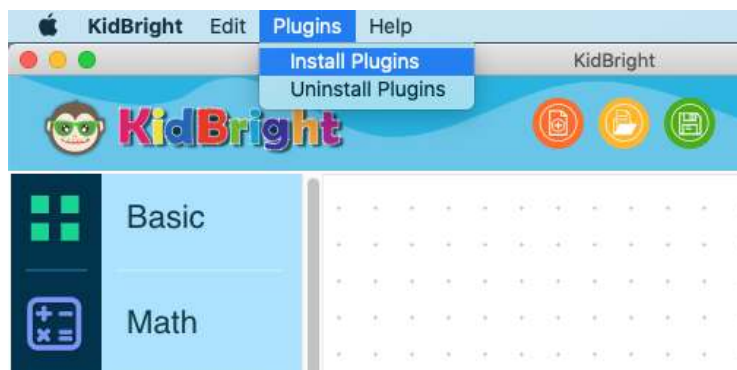
เวอร์ชัน (Version)

OpenKB V1.2.1

- ชื่อไฟล์: openkb_V1.2.1.zip
- MD5: 417498a40e6270139c075ed14872784b
- ขนาด: 12.15 kB
- ดาวน์โหลดแล้ว: 382 ครั้ง
- เพิ่มเมื่อ: วันจันทร์ที่ 20 มกราคม พ.ศ. 2563 20:41 น.
- ดาวน์โหลดเวอร์ชันนี้ หรือ ดูบล็อกตัวอย่าง

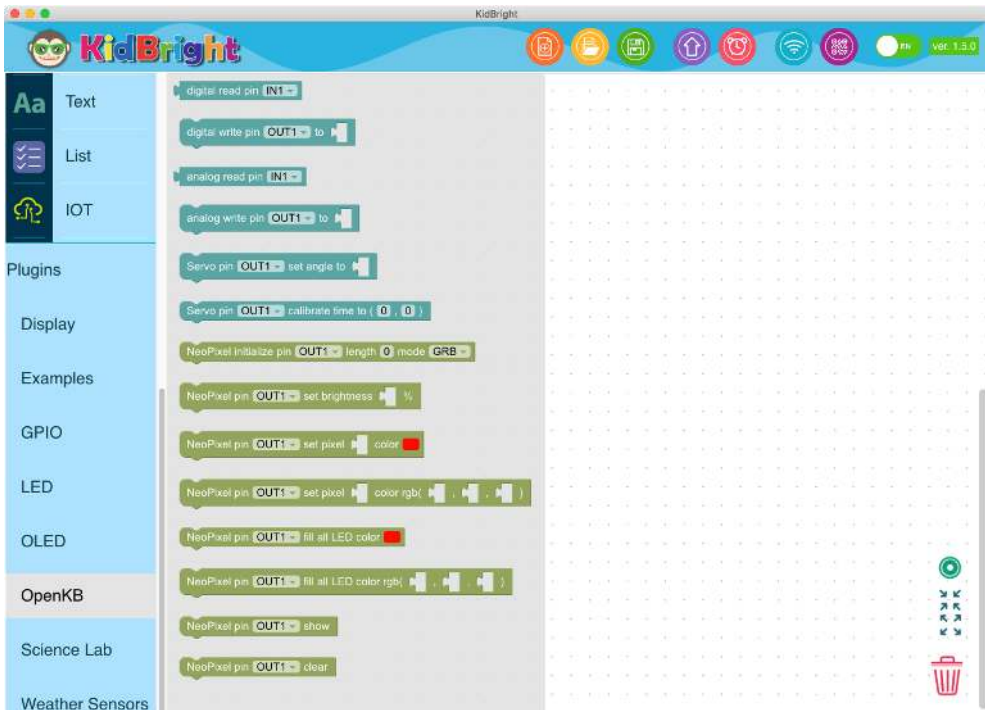
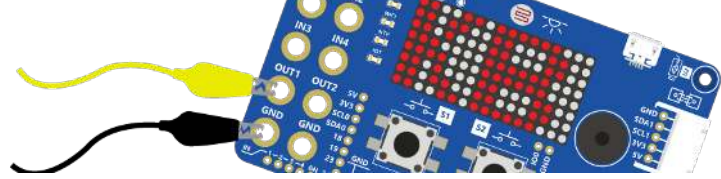
รูปภาพ 5-7 หน้าเว็บไซต์สำหรับดาวน์โหลดปลั๊กอิน OpenKB

2. เลือกเมนู Plugins > Install Plugins จากนั้นจึงเลือกไฟล์ปลั๊กอินที่ได้ดาวน์โหลดมาในขั้นตอนที่ 1



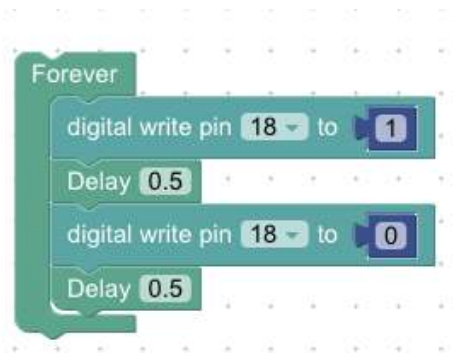
รูปภาพ 5-8 เมนูสำหรับติดตั้งปลั๊กอินลงใน KidBright IDE

3. เมื่อติดตั้งสำเร็จจะพบหมวด OpenKB ขึ้นในแถบ Plugins ของโปรแกรม KidBright IDE ดังภาพ



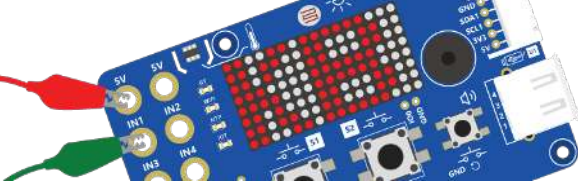
รูปภาพ 5-9 เมนูหมวด OpenKB ในแถบปลั๊กอิน

4. ตัวอย่างบล็อกด้านล่างจะทำการเปิด-ปิดอุปกรณ์ที่เชื่อมต่อกับพอร์ต 18 ทุก ๆ 0.5 วินาที

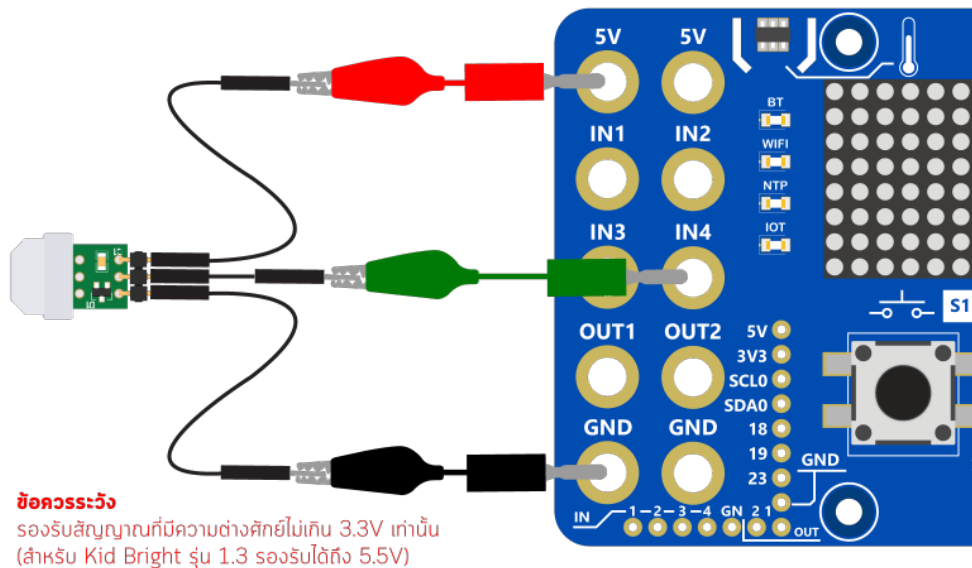


รูปภาพ 5-10 ตัวอย่างการควบคุมพอร์ต 18 ด้วยบล็อก digital write ในหมวด OpenKB

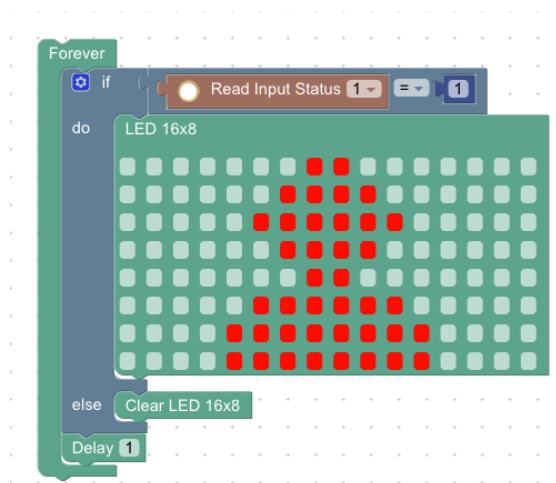
หมายเหตุ OpenKB เป็นบอร์ดพัฒนาที่ต่อยอดจากบอร์ด KidBright โดยมีความเข้ากันได้
ในเชิงซอฟต์แวร์และรองรับการพัฒนาโปรแกรมด้วย KidBright IDE ผู้พัฒนาที่สนใจสามารถศึกษา
ข้อมูลเพิ่มเติมได้ทาง <https://inex.co.th/shop/openkb.html>



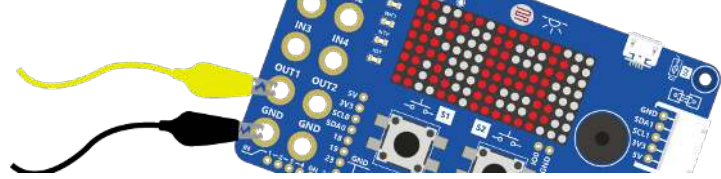
สำหรับอุปกรณ์อินพุตเช่นเซนเซอร์ PIR, เซนเซอร์จับเสียงดัง ฯลฯ สามารถเชื่อมต่อเข้าสู่ช่องเชื่อมต่อปลั๊กกล้วย IN1-4 ช่องเชื่อมต่อ 18, 19, 23 หรือช่องเชื่อมต่อ I1-4 บนบอร์ด KidBright รุ่น 1.5 ได้ ทั้งนี้ในบอร์ด Kid Bright รุ่น 1.3 ความต่างศักย์ของสัญญาณที่นำมาเชื่อมต่อกับช่องปลั๊กกล้วยจะต้องไม่เกิน 5.5 โวลต์ (แนะนำที่ 3.3 โวลต์) สำหรับช่องเชื่อมต่ออื่น ๆ และทุกช่องเชื่อมต่อบน Kid Bright รุ่น 1.5 จะต้องไม่เกิน 3.3 โวลต์เท่านั้น



รูปภาพ 5-11 ตัวอย่างเชื่อมต่อเซนเซอร์ PIR ซึ่งใช้การสื่อสารผ่านพอร์ตดิจิทัล 1 พอร์ตเข้ากับบอร์ด KidBright

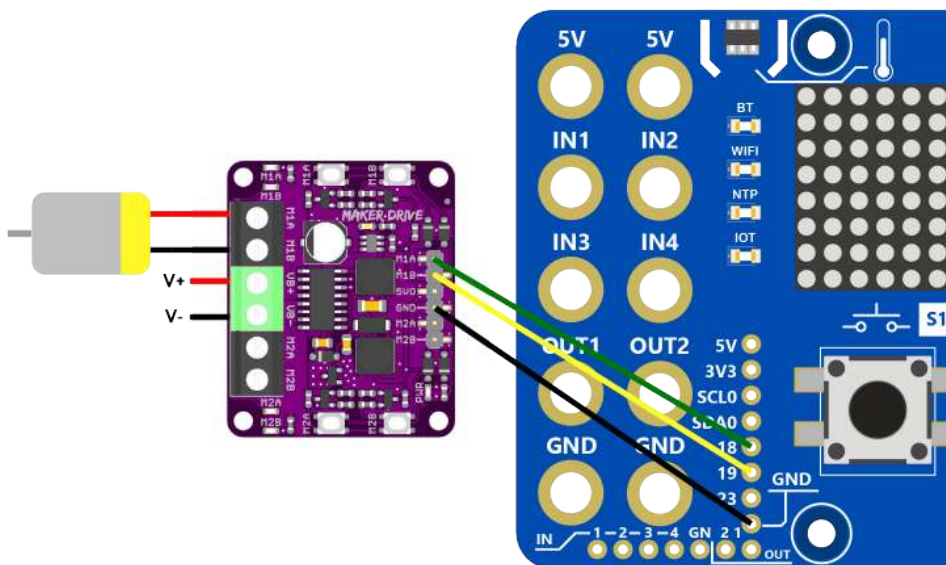


รูปภาพ 5-12 ตัวอย่างโปรแกรมเพื่อแสดงภาพคนเป็นเวลา 1 วินาทีเมื่อเซนเซอร์ PIR ตรวจจับคนได้

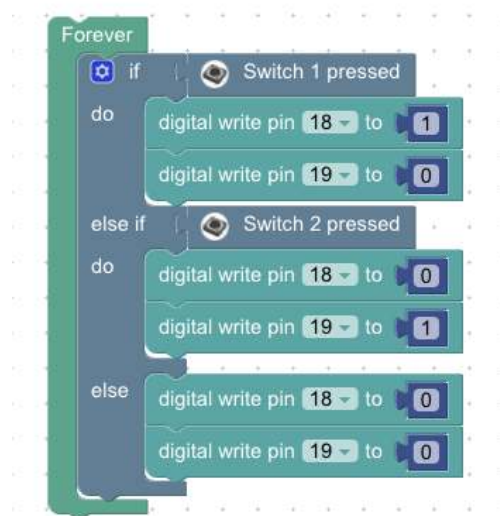


5.2 การควบคุมอุปกรณ์ไฟฟ้ากระแสตรงขนาดใหญ่

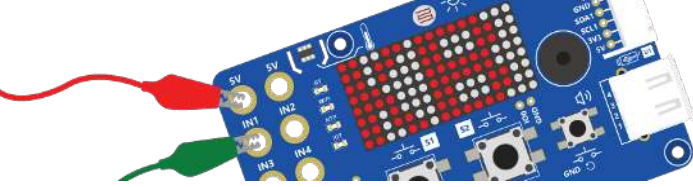
ถึงแม้ว่าบอร์ด KidBright จะได้รับการออกแบบให้มีช่องเชื่อมต่อแบบ Open Drain เพื่อควบคุมอุปกรณ์ไฟฟ้าขนาดเล็กได้โดยตรง การควบคุมอุปกรณ์ไฟฟ้ากระแสตรงที่ใช้กระแสสูงกว่า 100 มิลลิแอมป์สามารถทำได้โดยเชื่อมต่อวงจรเพิ่มเติมด้วยทรานซิสเตอร์หรือมอสเฟตหรือใช้บอร์ดสำเร็จรูปดังรูป



รูปภาพ 5-13 ตัวอย่างการเชื่อมต่อบอร์ดขับเคลื่อนมอเตอร์ Cytron Maker Drive เข้ากับบอร์ด KidBright



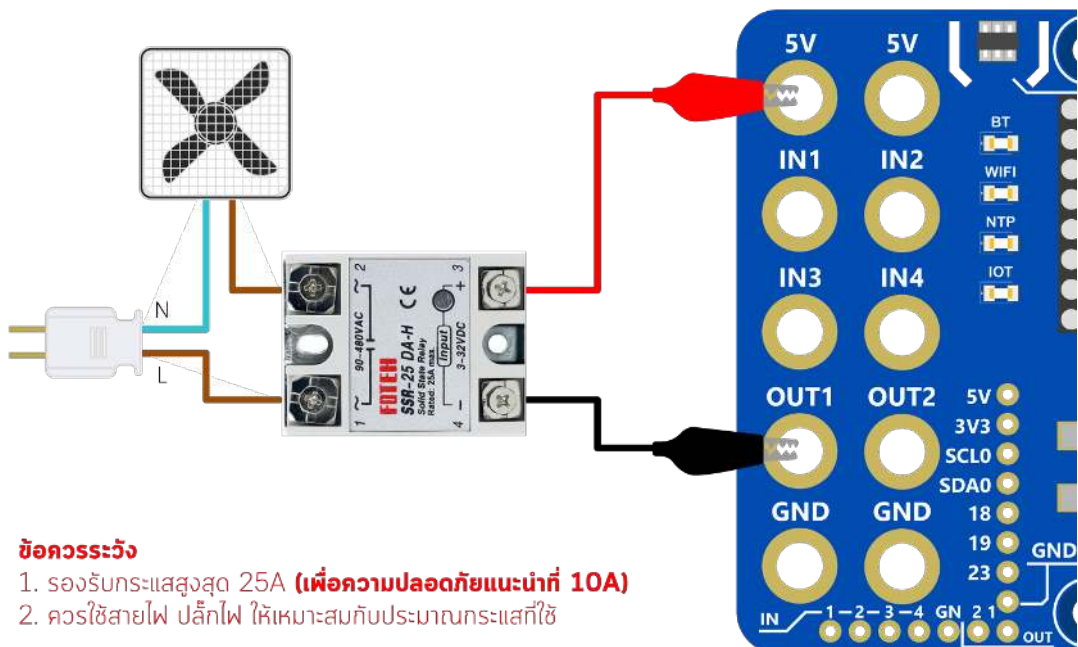
รูปภาพ 5-14 ตัวอย่างโปรแกรมหมุนมอเตอร์ตามเข็มเมื่อกดปุ่ม 1 และหมุนทวนเข็มเมื่อกดปุ่ม 2



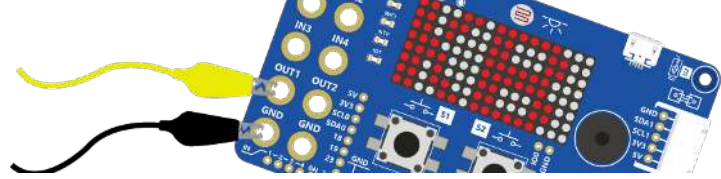
หมายเหตุ ในตัวอย่างข้างต้นผู้เขียนได้ทำการเชื่อมต่อบอร์ดขับเคลื่อนมอเตอร์รุ่น Cytron Maker Drive เข้ากับพอร์ตเชื่อมต่อหมายเลข 18 และ 19 จึงต้องทำการติดตั้งปลั๊กอินก่อนตามวิธีที่ได้กล่าวไปข้างต้น

5.3 การควบคุมอุปกรณ์ไฟฟ้ากระแสสลับ

บอร์ด KidBright สามารถนำไปประยุกต์ใช้ในการควบคุมเปิด-ปิดอุปกรณ์ไฟฟ้ากระแสสลับ อาทิหลอดไฟ 220V หรือเครื่องใช้ไฟฟ้าในบ้านได้ โดยเชื่อมต่อกับรีเลย์ ซึ่งเป็นอุปกรณ์ไฟฟ้าที่ทำหน้าเป็นเหมือนสวิตช์ที่สามารถสั่งงานเปิดปิดได้ด้วยสัญญาณไฟฟ้า โดยในตัวอย่างนี้จะแนะนำการเชื่อมต่อกับ Solid State Relay รุ่น SSR-25DA ซึ่งรองรับกระแสสูงสุด 25A ทั้งนี้ต้องใช้สายไฟ ปลั๊กไฟและขั้วต่อต่าง ๆ ให้เหมาะสม ในการใช้งานจริงแนะนำให้ใช้กระแสสูงสุดไม่เกิน 10A

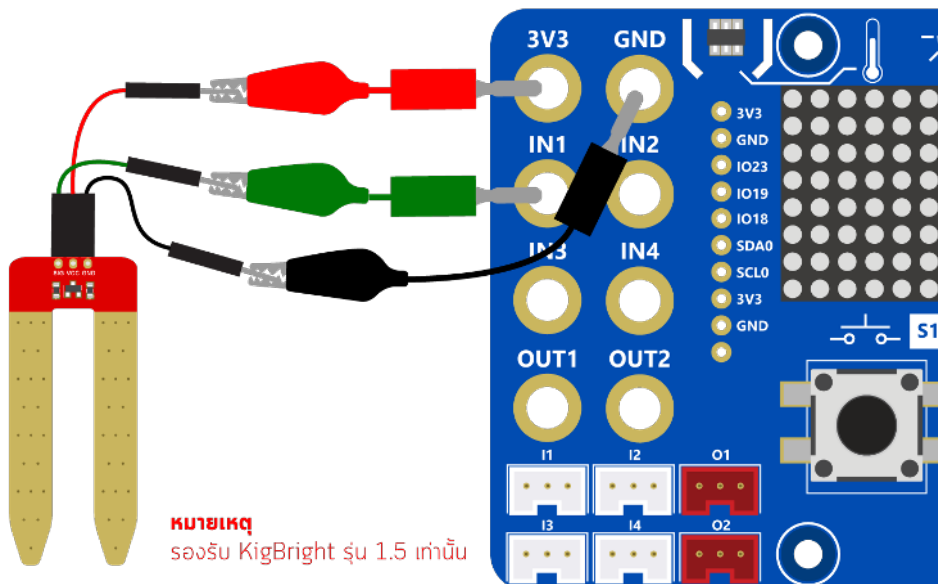


รูปภาพ 5-15 ตัวอย่างการเชื่อมต่อ Solid State Relay เข้ากับบอร์ด KidBright เพื่อควบคุมอุปกรณ์ไฟฟ้ากระแสสลับ



5.4 การเชื่อมต่ออุปกรณ์นอก

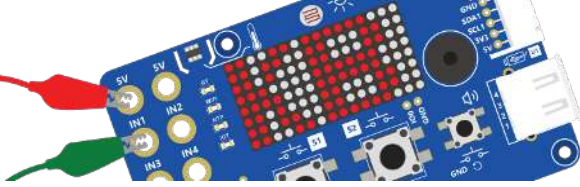
บอร์ด KidBright ตั้งแต่รุ่น 1.5 เป็นต้นไปรองรับการเชื่อมต่ออุปกรณ์ในรูปแบบอนาล็อกผ่านช่องเชื่อมต่อปลั๊กกล้วย IN1-4 และช่องเชื่อมต่อ JST 3 พิน I1-4 ดังรูป ทั้งนี้ผู้พัฒนาต้องลงปลั๊กอิน OpenKB เพิ่มเติมตามวิธีที่ได้กล่าวไปข้างต้น



รูปภาพ 5-16 ตัวอย่างการเชื่อมต่อเซนเซอร์อนาล็อกเข้ากับบอร์ด KidBright รุ่น 1.5

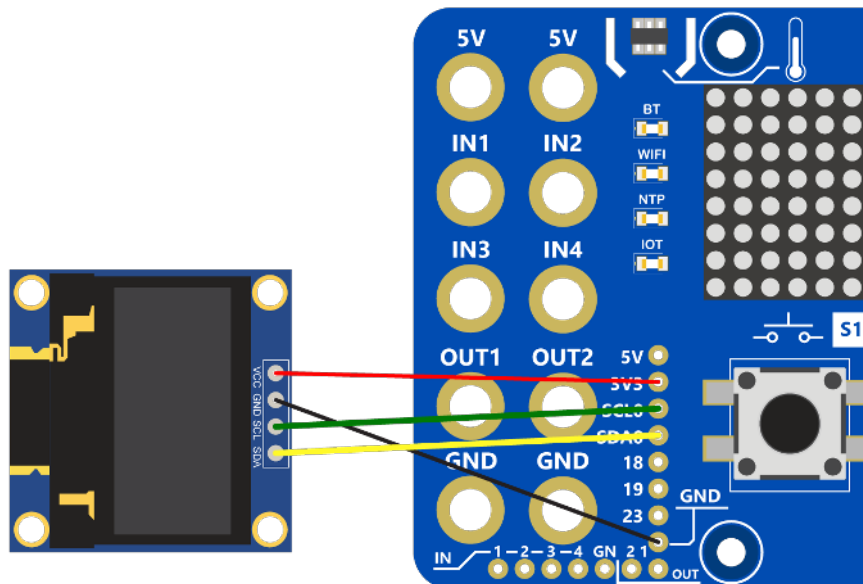


รูปภาพ 5-17 ตัวอย่างโปรแกรมเพื่ออ่านค่าความชื้นในดิน (0-1023) และแสดงผลทางหน้าจอ Dot Matrix LED



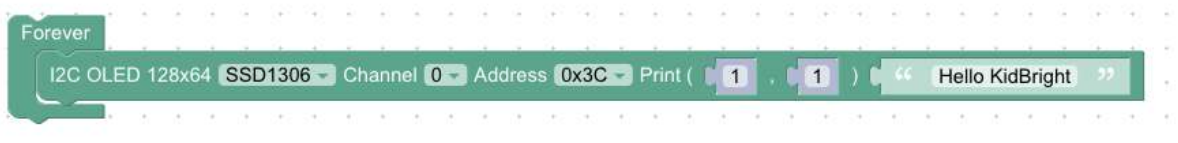
5.5 การเชื่อมต่ออุปกรณ์ I²C

การเชื่อมต่ออุปกรณ์ในรูปแบบ I²C สามารถทำได้ใน 2 ช่องทางคือ 1. ผ่านช่องเชื่อมต่อ KB-Chain โดยรองรับอุปกรณ์ในชุด KB Chain สามารถศึกษาข้อมูลเพิ่มเติมได้ที่ <https://www.kidbright.io/shop> และ 2. ผ่านช่องเชื่อมต่อระยะ 2.54 มม. ที่ขา SDA0 และ SCL0 ในที่นี้จะแนะนำการเชื่อมต่อหน้าจอ OLED ขนาด 128x64 px ด้วยวิธีที่ 2 ซึ่งสามารถใช้ได้กับเซนเซอร์ทั่วไปที่หาซื้อได้ตามท้องตลาด

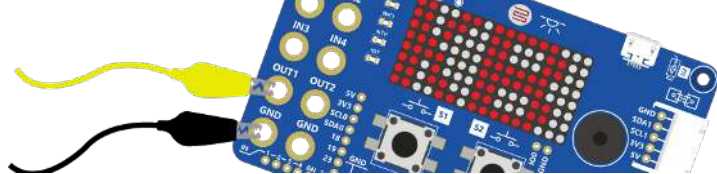


รูปภาพ 5-18 ตัวอย่างการเชื่อมต่อหน้าจอ OLED ในรูปแบบ I²C เข้ากับบอร์ด KidBright

ในการโปรแกรมผู้พัฒนาสามารถใช้บล็อกในหมวด OLED ซึ่งเป็น Plugin ที่ได้ติดตั้งมาแล้วพร้อมกับโปรแกรม KidBright IDE ได้ในทันที ดังรูป



รูปภาพ 5-19 ตัวอย่างโปรแกรมเพื่อควบคุมหน้าจอ OLED



บทที่ 6

การส่งรับข้อมูลระหว่างบอร์ด KidBright และแอปพลิเคชัน KidBright IoT

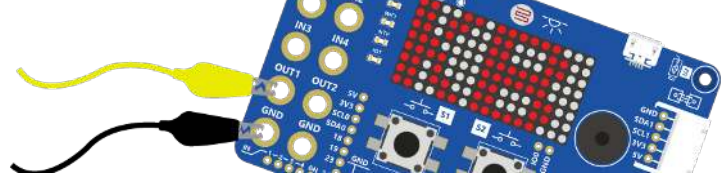
6.1 แนะนำแอปพลิเคชัน KidBright IoT

KidBright IoT คือชื่อของแอปพลิเคชันบนมือถือที่ถูกสร้างขึ้นเพื่อใช้งานร่วมกับบอร์ด KidBright เพื่อเพิ่มความสามารถให้บอร์ด KidBright ซึ่งทำให้เราสามารถส่งข้อมูลไปแสดงผลหรือรับข้อมูลเข้าสู่บอร์ดผ่านระบบอินเทอร์เน็ตเข้าสู่โทรศัพท์มือถือได้ การส่ง-รับข้อมูลผ่านอินเทอร์เน็ตมีประโยชน์ในการเฝ้าดูหรือสั่งการบอร์ด KidBright จากระยะไกลด้วยโทรศัพท์มือถือ ซึ่งสามารถนำไปใช้ทำโครงการที่ใช้แนวคิด Internet of Things เบื้องต้นได้ ตัวอย่างเช่น ระบบควบคุมอุณหภูมิระยะไกล ระบบเปิด-ปิดไฟอัตโนมัติ เป็นต้น

KidBright IoT นับเป็นแอปพลิเคชันบนมือถือสัญชาติไทย ที่ถูกพัฒนาขึ้นโดยบริษัท เน็กซ์พาย จำกัด ซึ่งให้บริการเกี่ยวกับ IoT Platform รวมทั้งให้คำปรึกษาเกี่ยวกับระบบ IoT และพัฒนาซอฟต์แวร์เพื่อเชื่อมต่ออุปกรณ์เข้ากับระบบ IoT ปัจจุบัน KidBright IoT มีให้ดาวน์โหลดสำหรับระบบปฏิบัติการ Android และ iOS โดยรุ่นปัจจุบันคือ 1.1.4 สำหรับ Android และ 1.1.2 สำหรับ iOS (ข้อมูลวันที่ 8 มิถุนายน 2563)



รูปภาพ 6-1 ภาพหน้าจอแสดงแอปพลิเคชัน KidBright IoT



แอปพลิเคชัน KidBright IoT เป็นแอปพลิเคชันที่ใช้งานง่าย และมีการตั้งค่าเพียงไม่กี่ขั้นตอน จึงทำให้ทุกคนสามารถใช้งานแอปพลิเคชันนี้ได้ ในการใช้งาน KidBright IoT จะใช้การตั้งค่าในโปรแกรม KidBright IDE เป็นหลัก แล้วจึงจะผูกการเชื่อมต่อเข้ากับแอปพลิเคชันในภายหลัง ทั้งนี้ รายละเอียดการตั้งค่าจะอยู่ในส่วนต่อไป

6.2 การติดตั้งแอปพลิเคชัน KidBright IoT ในมือถือ

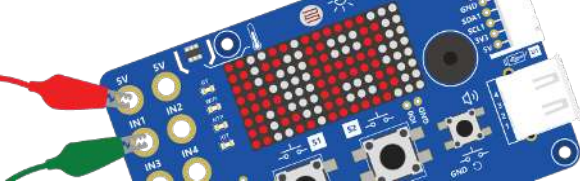
สำหรับ Android สามารถทำได้ผ่าน Google Play และสำหรับ IOS สามารถทำได้ผ่าน App Store โดยค้นหาคำว่า KidBright IoT จะเจอโลโก้ของแอปพลิเคชันดังรูปภาพ 6-2 จากนั้นกดติดตั้ง ทั้งนี้ KidBright IoT เป็นแอปพลิเคชันที่ใช้งานได้ฟรี



รูปภาพ 6-2 ภาพแสดงโลโก้ของแอปพลิเคชัน KidBright IoT

6.3 ตัวอย่างการใช้งานเพื่อส่ง-รับข้อมูลระหว่างบอร์ด และ แอปพลิเคชัน

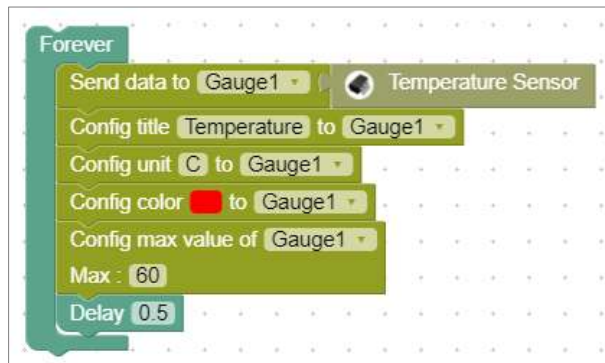
การตั้งค่าการใช้งาน KidBright IoT สามารถทำได้ไม่ยาก และสามารถศึกษาได้จากตัวอย่าง ทั้งนี้ ตัวอย่างที่เรานำมาใช้จะมีทั้งสิ้น 2 ตัวอย่าง คือ ตัวอย่างการส่งค่าอุณหภูมิที่วัดได้จากบอร์ด KidBright ไปที่แอปพลิเคชันมือถือ และ ตัวอย่างการควบคุม Buzzer และ OLED จากแอปพลิเคชันมือถือ



6.3.1 ตัวอย่างการส่งค่าอุณหภูมิที่วัดได้จากบอร์ด KidBright ไปที่แอปพลิเคชันมือถือ

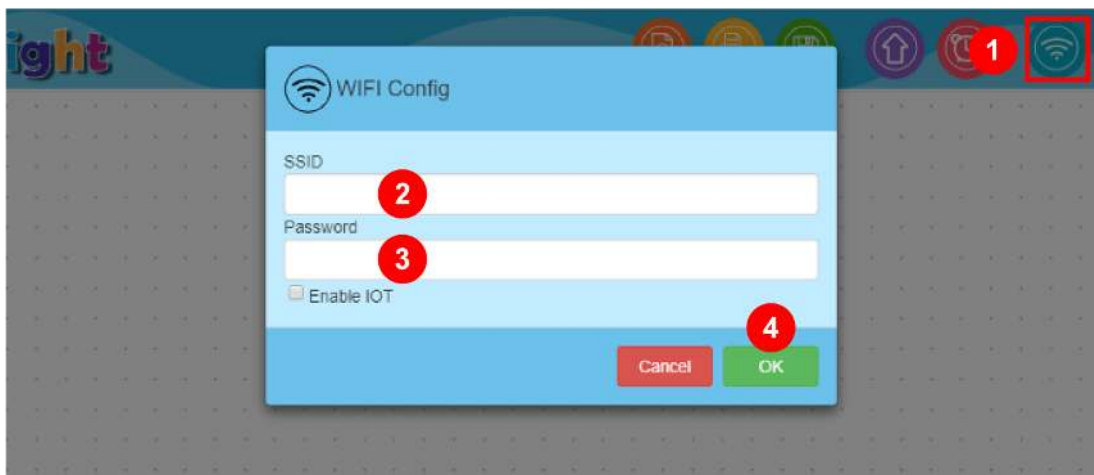
การส่งค่าอุณหภูมิไปที่แอปพลิเคชันมือถือ สามารถทำได้ตามขั้นตอนต่อไปนี้

1. ใน KidBright IDE ให้ทำการต่อบล็อกดังรูปภาพ 6-3 ทั้งนี้ บล็อก Forever และ Delay อยู่ในกลุ่ม Basic ส่วนบล็อก Temperature Sensor อยู่ในกลุ่ม Sensor และบล็อกอื่น ๆ อยู่ในกลุ่ม KB IOT



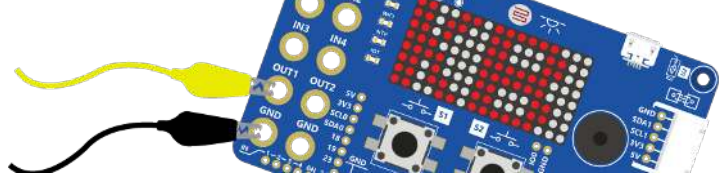
รูปภาพ 6-3 ภาพแสดงบล็อกเพื่อส่งข้อมูลอุณหภูมิไปยังแอปพลิเคชัน KidBright IoT

2. กดที่ปุ่ม Wi-Fi เพื่อทำการตั้งค่า SSID / Password และให้เลือก Enable IOT ทั้งนี้ ต้องมี Access Point ที่ปล่อยสัญญาณอินเทอร์เน็ต หากไม่มี สามารถใช้โทรศัพท์มือถือเพื่อตั้งค่าให้แชร์อินเทอร์เน็ตให้กับบอร์ด KidBright ได้



รูปภาพ 6-4 ภาพแสดงการตั้งค่า Wi-Fi's SSID และ Password สำหรับการเชื่อมต่ออินเทอร์เน็ต

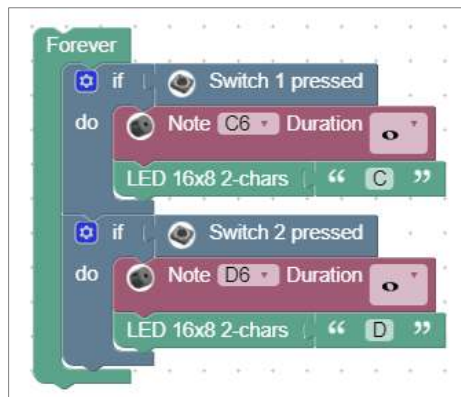
3. เสียบบอร์ด KidBright เข้าสู่คอมพิวเตอร์ด้วยสาย USB จากนั้นกดปุ่ม Upload



4. เมื่ออัปโหลดลงบอร์ดเสร็จสิ้น ให้กดที่ปุ่ม QR Code Mac Address
5. หลังจากนั้นให้เปิดแอปพลิเคชัน KidBright IoT แล้วทำการ Scan QR Code
6. จะได้หน้า Dashboard ซึ่งมีการแสดงอุณหภูมิที่มาตรวัด (gauge)

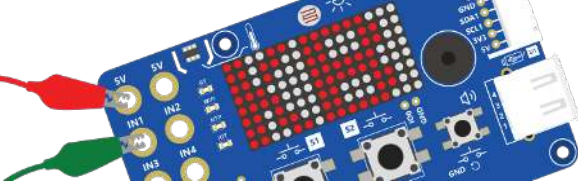
6.3.2 ตัวอย่างการควบคุม Buzzer และ Dot Matrix LED จากแอปพลิเคชันมือถือ

1. ใน KidBright IDE ให้ทำการต่อบล็อกดังรูปภาพ 6-5 โดยบล็อก Forever และ LED อยู่ใน
กลุ่ม Basic ส่วนบล็อก If และ Switch อยู่ในกลุ่ม Logic และบล็อก Note อยู่ในกลุ่ม
Music



รูปภาพ 6-5 ภาพแสดงบล็อกการเล่นเสียงดนตรีผ่าน Buzzer และแสดงข้อความผ่าน Matrix LED เมื่อกดที่ปุ่ม

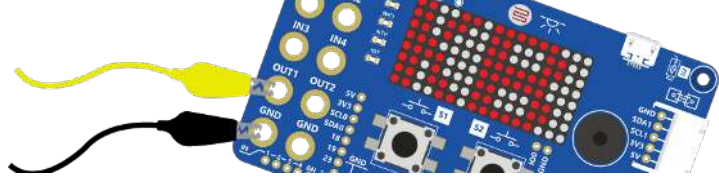
2. กดที่ปุ่ม Wi-Fi เพื่อทำการตั้งค่า SSID / Password และให้เลือก Enable IOT ทั้งนี้ ต้องมี
Access Point ที่ปล่อยสัญญาณอินเทอร์เน็ต หากไม่มี สามารถใช้โทรศัพท์มือถือเพื่อตั้งค่า
ให้แชร์อินเทอร์เน็ตให้กับบอร์ด KidBright ได้ ดังรูปภาพ 6-4
3. เสียบบอร์ด KidBright เข้าสู่คอมพิวเตอร์ด้วยสาย USB จากนั้นกดปุ่ม Upload
4. เมื่ออัปโหลดลงบอร์ดเสร็จสิ้น ให้กดที่ปุ่ม QR Code Mac Address
5. หลังจากนั้นให้เปิดแอปพลิเคชัน KidBright IoT แล้วทำการ Scan QR Code
6. จะได้หน้า Dashboard ซึ่งมีปุ่มกดให้ควบคุมการเปิด-ปิดเสียงดนตรีผ่าน Buzzer และ Dot
Matrix LED

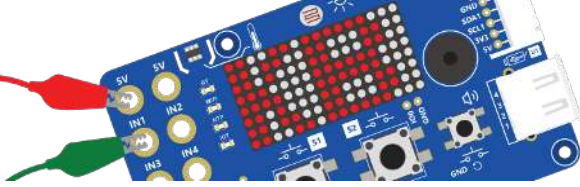


6.4 แบบฝึกหัดและโจทย์

แบบฝึกหัดและโจทย์สำหรับฝึกหัดสร้างโครงงานอย่างง่ายที่ใช้แนวคิด IoT ผ่านแอปพลิเคชัน KidBright IoT

1. สร้างระบบควบคุมแสงสว่างเปิด-ปิดระยะไกลผ่าน KidBright IoT โดยใช้หลอดไฟ USB
2. สร้างระบบเก็บค่าอุณหภูมิ แสดงผลออกหน้าจอ Dashboard ในรูปแบบ Graph ผ่านแอปพลิเคชัน KidBright IoT
3. สร้างระบบวัดแสงสว่างของห้อง หากแสงสว่างมีค่าต่ำกว่า 30% ให้ทำการแจ้งเตือนผ่าน Notification ไปที่โทรศัพท์มือถือทันที
4. สร้างระบบวัดแสงสว่างของห้อง หากแสงสว่างมีค่าต่ำกว่า 30% ให้ทำการแจ้งเตือนผ่าน Notification ไปที่โทรศัพท์มือถือทันที โดยมีเงื่อนไขเพิ่มเติมคือกำหนดให้ใน 3 นาที สามารถแจ้งเตือนได้แค่ครั้งเดียวเท่านั้น
5. สร้างระบบควบคุมกึ่งอัตโนมัติเพื่อควบคุมการเปิด-ปิดวาล์วน้ำสำหรับรดน้ำต้นไม้ ซึ่งการทำงานจะแบ่งเป็นสองส่วนคือส่วนอัตโนมัติและส่วนควบคุมด้วยมือ ในส่วนอัตโนมัติระบบจะเปิดวาล์วน้ำทุก ๆ วันช่วงเช้าและเย็นเป็นเวลา 1 นาที ได้แก่ เวลา 6.00 – 6.01 น. และ 18.00 – 18.01 น. และในส่วนควบคุมด้วยมือระบบจะสามารถถูกสั่งเปิดหรือปิดได้ผ่านโทรศัพท์มือถือ ซึ่งหากทั้งสองส่วนมีการสั่งการที่แตกต่างกัน ให้ระบบถือคำสั่งของส่วนควบคุมด้วยมือเป็นสำคัญ





บทที่ 7

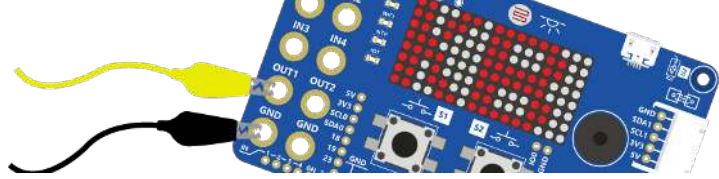
รู้จักกับ NETPIE 2020

7.1 ประวัติโดยสังเขป

NETPIE คือ ระบบ Cloud Platform ที่ถูกออกแบบมาให้เหมาะสมกับงานด้าน IOT ซึ่งกระทรวงวิทยาศาสตร์และเทคโนโลยี โดยศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (NECTEC) ได้เปิดตัว สำหรับนักพัฒนา โดยคำว่า NETPIE ย่อมาจาก Network Platform for Internet of Everything โดยเป็น คลาวด์เซิร์ฟเวอร์ที่ให้บริการในรูปแบบ Platform as a Service สำหรับติดต่อสื่อสาร แลกเปลี่ยนข้อมูลระหว่างอุปกรณ์ต่าง ๆ เหมาะสำหรับการพัฒนา IoT ในปัจจุบัน โดยรองรับการเชื่อมต่อผ่านอุปกรณ์ต่าง ๆ อาทิเช่น Raspberry PI, ESP8266, ESP32, HTML5 และ NodeJS เป็นต้น ทั้งนี้ NETPIE ยังช่วยลดภาระด้านความปลอดภัยโดยผู้พัฒนาสามารถกำหนดหรือออกแบบได้เองทั้งหมด ดังนั้น NETPIE จึงเป็นอีกทางเลือกหนึ่งสำหรับนักพัฒนา IoT ในยุคปัจจุบัน



รูปภาพ 7-1 ภาพแสดงแนวคิด NETPIE Cloud ที่เป็นศูนย์กลางการเชื่อมต่อกับอุปกรณ์ทุกประเภท

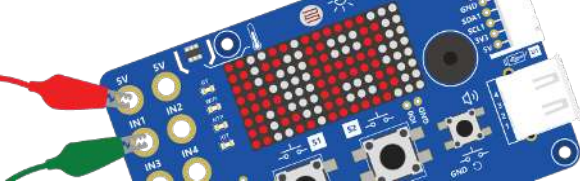


7.2 การสมัครเป็นผู้ใช้งาน NETPIE 2020

1. เข้าไปที่ <https://auth.netpie.io/signup> ทำการกรอกข้อมูลให้ครบถ้วนจากนั้นระบบจะทำการส่ง email เพื่อให้ทำการสมัครและตั้งรหัสผ่านครั้งแรก

รูปภาพ 7-2 แบบฟอร์มกรอกข้อมูลสำหรับการสมัครสมาชิก NETPIE

2. ระบบทำการส่ง email เพื่อให้ทำการตั้งรหัสผ่านครั้งแรก
3. ทำการ login เข้าสู่ระบบให้เรียบร้อยหากสามารถ login ได้ถือว่าการลงทะเบียนเสร็จสมบูรณ์

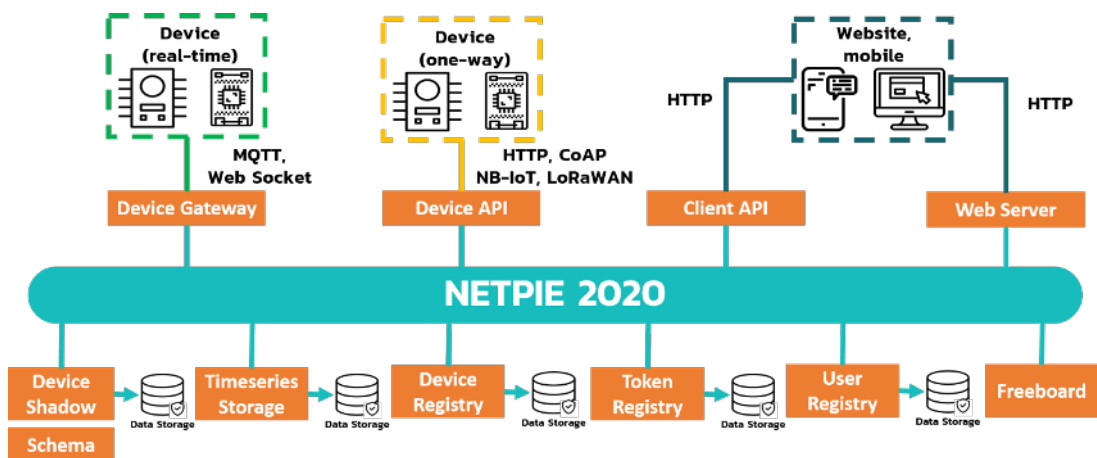


7.3 แนวคิดของ NETPIE 2020 และการเลือกรูปแบบการใช้งานสำหรับนักพัฒนา

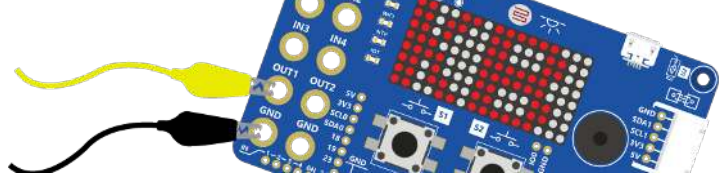
NETPIE 2020 เป็น Cloud Platform ที่ทำงานอยู่บนการสื่อสารหลากหลายรูปแบบไม่ว่าจะเป็น Native MQTT, Native WebSocket, HTTP, CoAP จึงทำให้ NETPIE 2020 มีความสามารถหลากหลายรูปแบบ รับส่งข้อมูลได้แบบ Real-time และทำงานร่วมได้กับหลากหลายอุปกรณ์ นอกจากนี้ NETPIE รุ่นปี 2020 ได้เพิ่มความสามารถในการเก็บข้อมูลอุปกรณ์ และทำการเชื่อมต่อ server ภายนอกหรือ web hook ได้อีกด้วย จึงถูกจัดให้เป็นแพลตฟอร์มที่ถูกพัฒนาขึ้นเพื่อสามารถตอบสนองผู้ใช้งานเชิงพาณิชย์ เช่น ผู้ผลิตอุปกรณ์ IoT, อุตสาหกรรม, โรงงาน และองค์กรที่พัฒนาสู่ยุค Digital Transformation 4.0 ซึ่งจะช่วยธุรกิจให้มีประสิทธิภาพยิ่งขึ้น ด้วยเทคโนโลยีการเชื่อมต่อทุกสรรพสิ่ง หรือ Internet of Things (IoT)

คุณสมบัติหลักๆของ NETPIE 2020 Platform ประกอบไปด้วย

1. การแสดงค่าข้อมูลจากเซ็นเซอร์หรืออุปกรณ์แบบ Real-time (Monitoring)
2. การควบคุมการทำงานของอุปกรณ์ต่าง ๆ ผ่าน Cloud Platform (Controlling)
3. การเก็บค่าข้อมูลที่ได้จากเซ็นเซอร์หรืออุปกรณ์ (Data Storage)
4. การแจ้งเตือนความผิดปกติของเซ็นเซอร์หรืออุปกรณ์จากที่ได้กำหนดไว้ (Notification)
5. การแสดงผลและควบคุมการทำงานของอุปกรณ์ผ่าน Dashboard (Dashboard for monitor & control)

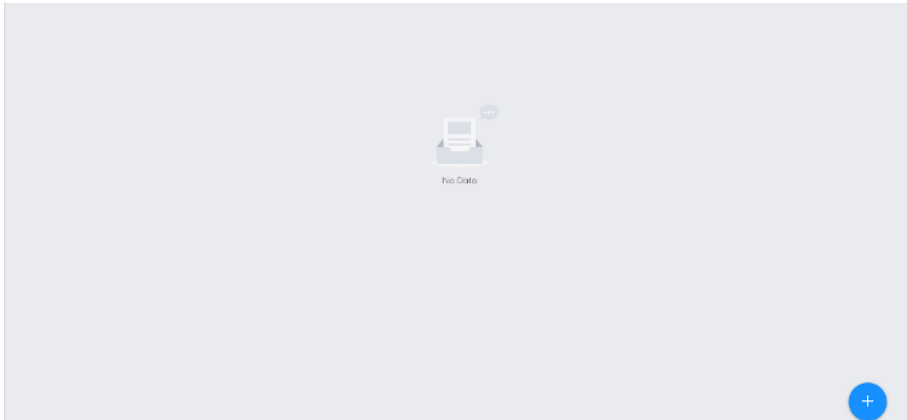


รูปภาพ 7-3 ภาพแสดงระดับชั้นของการทำงานของ NETPIE2020 อุปกรณ์ต่าง ๆ และคลังข้อมูล



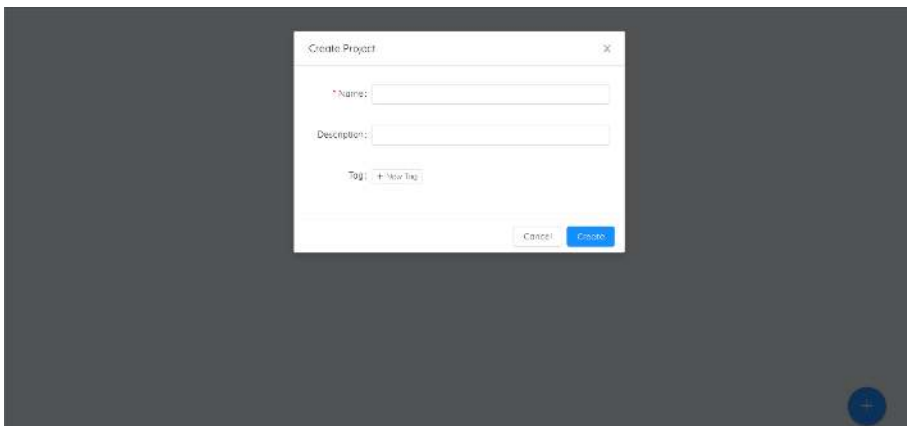
7.4 การสร้าง Project และ Device เบื้องต้นใน NETPIE

1. เมื่อล็อกอินเข้าสู่ <https://portal.netpie.io> เรียบร้อยแล้ว หน้าจอที่ปรากฏจะแสดงรายการ Project ทั้งหมดที่เคยสร้างไว้แล้วดังภาพ



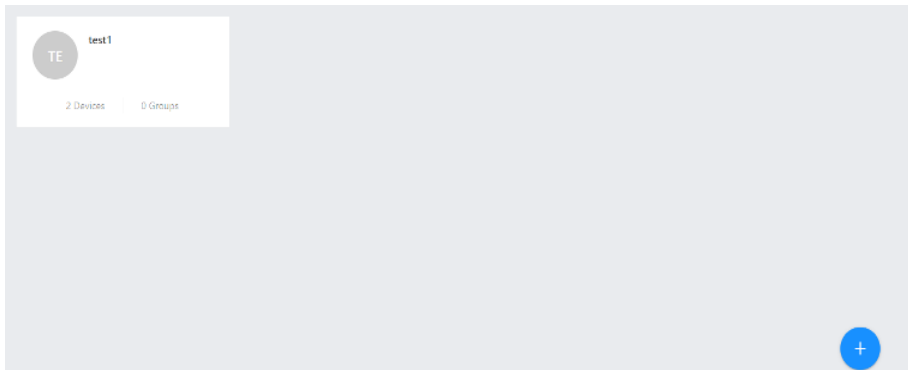
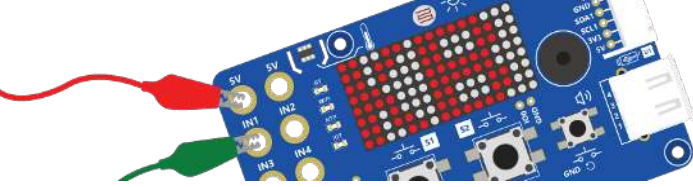
รูปภาพ 7-4 หน้าจอ NETPIE Portal

2. ทำการสร้าง Project ใหม่โดยการกดปุ่มบวกที่มุมขวาล่างของ NETPIE Portal จากนั้นทำการตั้งชื่อ Project และคลิกที่ปุ่ม Create



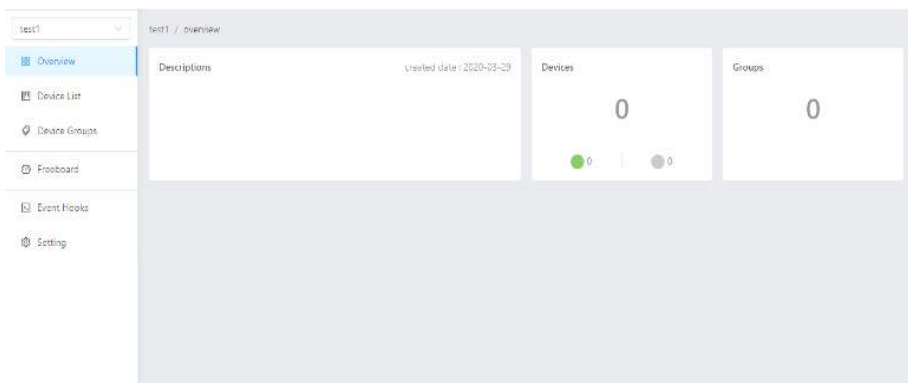
รูปภาพ 7-5 หน้าจอ New Project

3. เมื่อทำการสร้าง Project เสร็จแล้วให้ทำการคลิกเข้าไปใน Project เพื่อทำการเพิ่ม Device ใน Project ของเรา



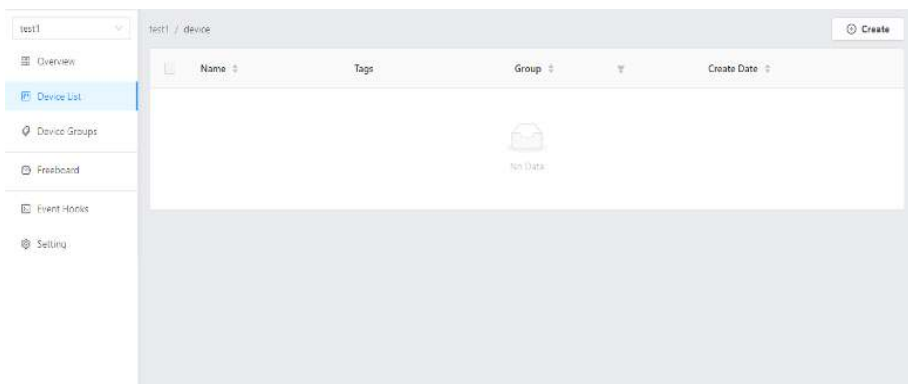
รูปภาพ 7-6 หน้า NETPIE Portal ที่มี Project ชื่อ test1

4. เมื่อคลิกเข้ามาแล้ว เราจะพบกับหน้าสรุปรวม Device ภายใน Project ของเราทั้งหมดโดยเราสามารถจัดการ Device ต่าง ๆ ได้ด้วยการกดที่ Device List

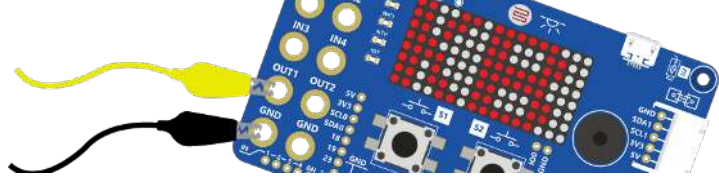


รูปภาพ 7-7 หน้า Overview ของ Project

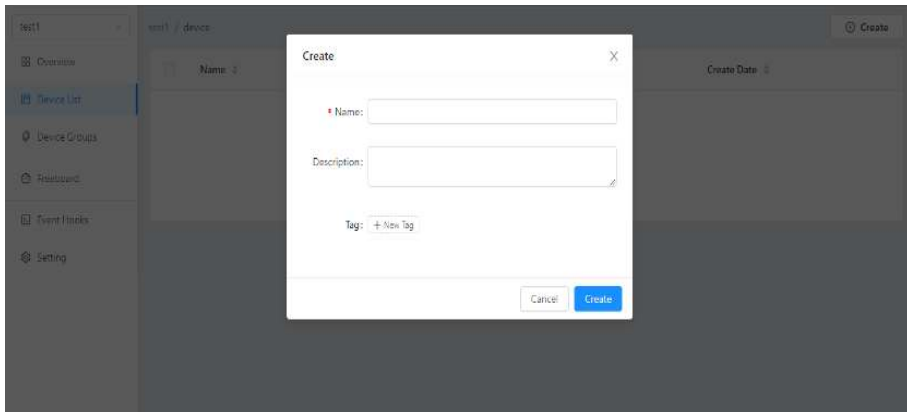
5. เมื่อทำการคลิกเข้าสู่หน้า Device list แล้วจะมี Device ทั้งหมดที่อยู่ใน Project ของเรา โดยเราสามารถจัดการตั้งค่า Device ต่าง ๆ ได้จากหน้านี้



รูปภาพ 7-8 หน้า Device List

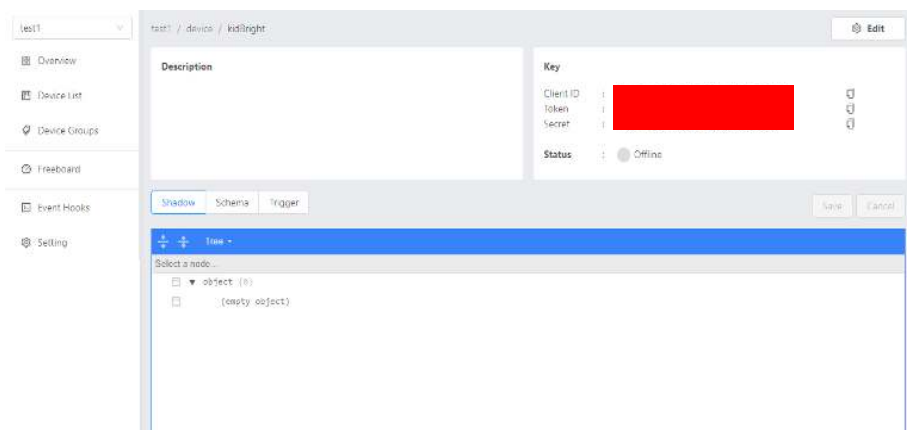


6. ทำการสร้าง Device ใหม่ด้วยการกดที่ปุ่ม Create ด้านบนขวามือและทำการตั้งชื่อ Device และคำอธิบายให้เรียบร้อย



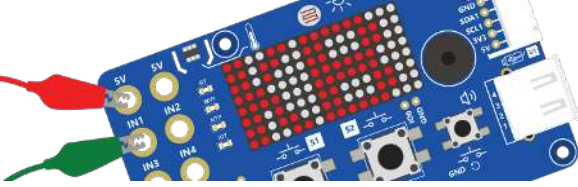
รูปภาพ 7-9 หน้า Create Device

7. หลังจากทำการสร้าง Device เสร็จเป็นที่เรียบร้อยแล้วให้ทำการคลิกเข้าไปภายในตัวอุปกรณ์ จะปรากฏหน้าจอที่บอกค่าพื้นฐานต่าง ๆ ของอุปกรณ์ที่ใช้ในการติดต่อกับเซิร์ฟเวอร์และสามารถตั้งค่าความสามารถของอุปกรณ์ได้จากหน้าจอดังกล่าว (อธิบายเพิ่มเติมในบทถัดไป)



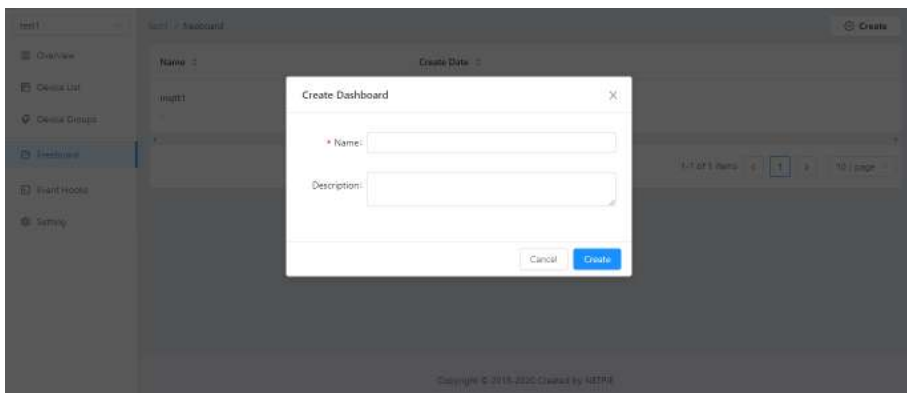
รูปภาพ 7-10 หน้าตั้งค่า Device

8. สามารถหาข้อมูลการใช้งานเบื้องต้นเพิ่มเติมได้จาก <https://docs.netpie.io/getting-started.html>



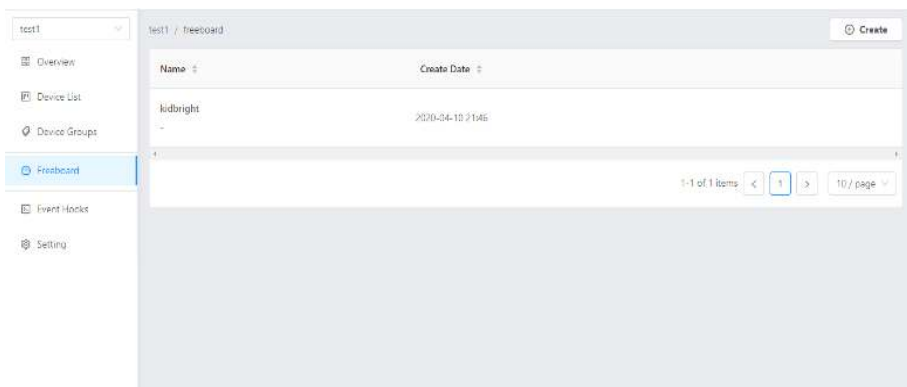
7.5 รู้จักกับ NETPIE Freeboard

NETPIE Freeboard เป็นเครื่องมือหนึ่งบนแพลตฟอร์มของ NETPIE IOT ซึ่งเป็นหน้าเว็บไซต์ที่ใช้สำหรับในการสร้าง interface ควบคุมอุปกรณ์ หรือรับค่าจากอุปกรณ์ไปแสดงบนหน้าเว็บ โดยมีฟังก์ชันและอุปกรณ์การแสดงผลและควบคุมหลายรูปแบบซึ่งผู้ใช้สามารถจัดการออกแบบหน้าต่างนี้ได้ด้วยตัวเองเพื่อทำการสื่อสารไปยังอุปกรณ์สื่อสารต่าง ๆ ที่เชื่อมต่ออยู่กับ NETPIE ผู้ใช้งานสามารถทำการสร้างหน้าต่าง NETPIE Freeboard ได้ด้วยการกดที่ Freeboard จากนั้นทำการกดปุ่ม create และทำการตั้งชื่อ Freeboard ของเราให้เรียบร้อย เพื่อสร้าง Freeboard

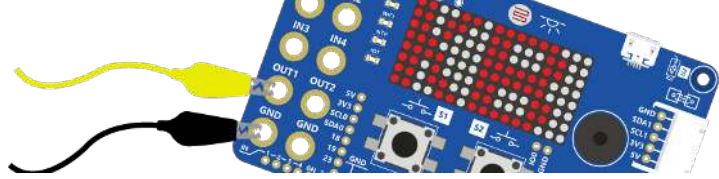


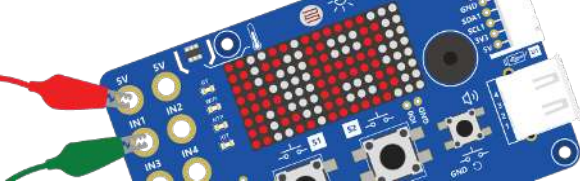
รูปภาพ 7-11 รูปแสดงการสร้างหน้าต่างสร้างหน้า Dashboard

หลังจากทำการสร้าง Dashboard เรียบร้อยแล้วจะเห็นหน้าเว็บที่แสดงรายชื่อของ Freeboard ของเราทั้งหมด ซึ่งเราสามารถคลิกเข้าไปเพื่อทำการตั้งค่าหน้าต่าง Freeboard ของเราทั้งหมดได้



รูปภาพ 7-12 รูปแสดงรายชื่อหน้าต่าง Freeboard



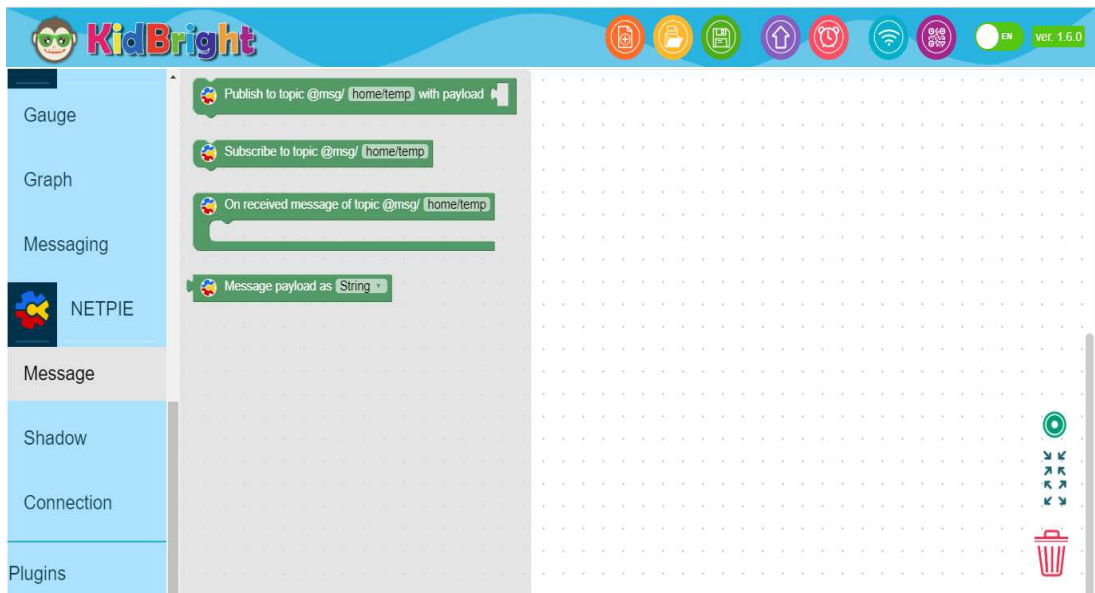


บทที่ 8

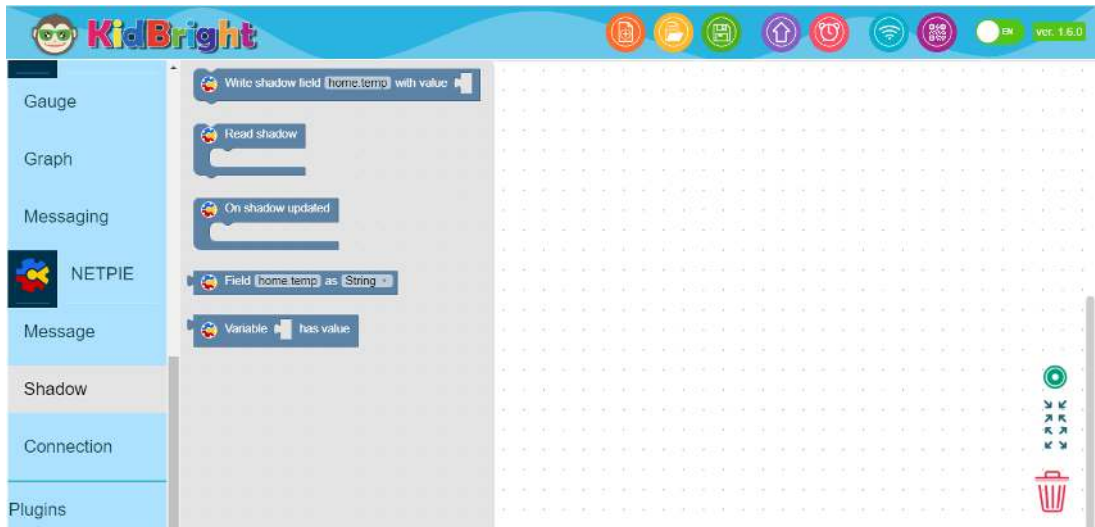
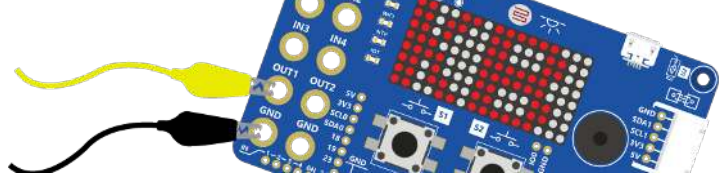
การใช้งาน NETPIE 2020 ร่วมกับบอร์ด KidBright

8.1 การใช้งานเชื่อมต่อ NETPIE ด้วย Kidbright IDE

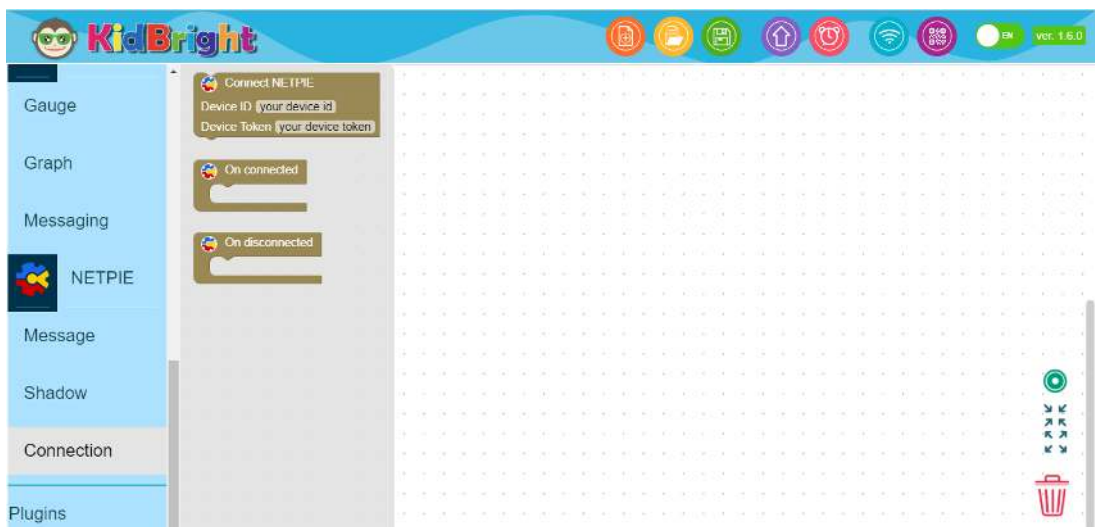
การเชื่อมต่อกับ NETPIE นั้นจำเป็นต้องใช้ร่วมกับ Kidbright IDE เวอร์ชัน 1.6 หรือสูงกว่าขึ้นไป โดยภายในโปรแกรมจะมีชุดคำสั่งของ NETPIE มาให้เลือกใช้งานทั้งหมด 3 หัวข้อด้วยกันซึ่งได้แก่ Message (ใช้สำหรับการสื่อสารแบบข้อความหรือ @msg) Shadow (ใช้สำหรับการสื่อสารและเก็บข้อมูลลงฐานข้อมูลหรือ @shadow) และ Connection (ใช้สำหรับเชื่อมต่อกับ NETPIE และ ตรวจสอบสถานะการเชื่อมต่อ) ดังรูปภาพ 8-1 รูปภาพ 8-2 และ รูปภาพ 8-3 ตามลำดับ



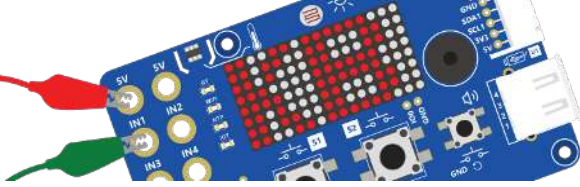
รูปภาพ 8-1 คำสั่งที่ใช้ในการสื่อสารแบบ Message กับ NETPIE



รูปภาพ 8-2 คำสั่งที่ใช้ในการสื่อสารแบบ Shadow กับ NETPIE



รูปภาพ 8-3 คำสั่งที่ใช้เชื่อมต่อ และตรวจสอบการเชื่อมต่อกับ NETPIE



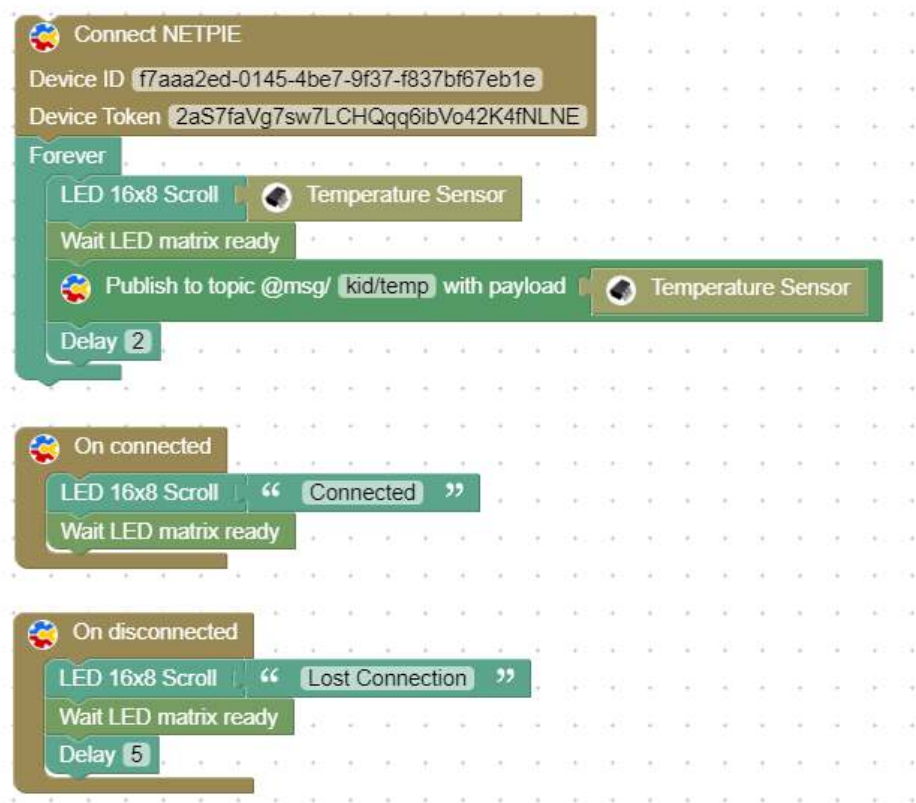
8.2 วิธีการใช้งาน NETPIE 2020 ในรูปแบบการส่ง-รับข้อความ

การสื่อสารในรูปแบบการส่ง-รับข้อความ @msg ของ NETPIE เป็นการสื่อสาร MQTT แบบพื้นฐานระหว่างอุปกรณ์ด้วยการส่งข้อความหากันบน topic เดียวกันตามที่ได้ตั้งไว้ โดยอุปกรณ์ที่จะได้รับข้อความนั้นจะต้องทำการ subscribe บน topic เดียวกันกับอุปกรณ์ที่ส่งข้อความออกมา

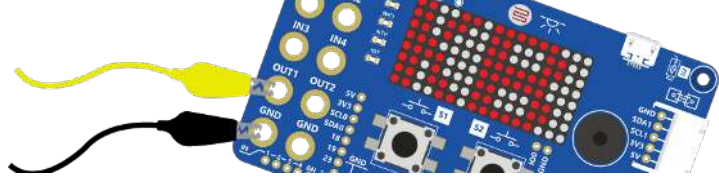
ในการสื่อสารในรูปแบบ @msg จะมีด้วยกันทั้งหมด 2 ตัวอย่างได้แก่ การส่งค่า sensor จากบอร์ดไปแสดงบนหน้าต่าง NETPIE Freeboard และการส่งข้อความ “on” และ “off” ระหว่างอุปกรณ์และ NETPIE Freeboard เพื่อทำการควบคุมหลอดไฟที่ต่อผ่านช่อง USB บนบอร์ด KidBright

8.2.1 การนำค่า sensor ไปแสดงบน NETPIE Freeboard

ส่วนที่ 1 การสร้างบล็อกโปรแกรมในฝั่งของ KidBright



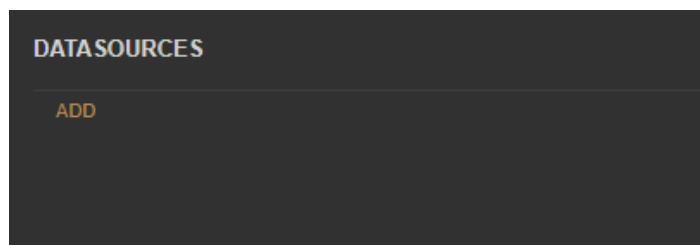
รูปภาพ 8-4 ภาพโปรแกรมเพื่อส่งค่าอุณหภูมิไปที่ NETPIE



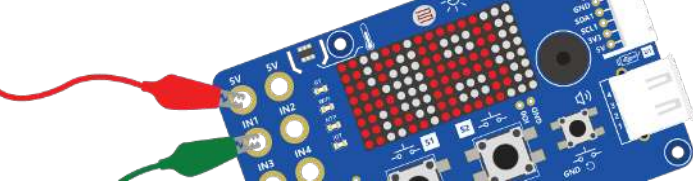
1. ในช่องของ Device ID นำ Device ID จากเว็บไซต์ของ NETPIE มาใส่ให้ถูกต้อง
2. ในช่องของ Device Token นำ Device Token จากเว็บไซต์ของ NETPIE มาใส่ให้ถูกต้อง
3. ภายในบล็อก Forever เราได้ทำการเขียนชุดคำสั่งให้แสดงค่าอุณหภูมิจากเซนเซอร์ออกทาง Dot Matrix LED และส่งค่าอุณหภูมิไปยัง NETPIE ในรูปแบบ @msg ด้วย Tag Publish to @msg/... with payload ... ซึ่งเป็น Tag ที่ใช้ในการส่งข้อความออกไปหาอุปกรณ์สื่อสาร ทุกๆ 2 วินาที
4. ภายในบล็อกคำสั่ง On connected จะทำการเชื่อมต่อกับ NETPIE และทำการตรวจสอบสถานะการเชื่อมต่อของอุปกรณ์หากเชื่อมต่อสำเร็จ จึงจะทำงานตามคำสั่งที่อยู่ภายในบล็อก ในที่นี้เป็นชุดคำสั่งให้แสดงข้อความคำว่า Connected ออกทาง Dot Matrix LED
5. ภายในบล็อกคำสั่ง On disconnected จะเป็นการเช็คการสูญเสียการเชื่อมต่อหากบอร์สูญเสียการเชื่อมต่อกับ NETPIE แล้วจึงทำตามคำสั่งที่อยู่ภายใน ในที่นี้เป็นแสดงข้อความคำว่า Lost Connection ออกทาง dot matrix led และรอ 5 วินาทีเพื่อทำการ connect กับ NETPIE ใหม่อีกครั้ง

ส่วนที่ 2 การตั้งค่าบน NETPIE Free Board

6. ทำการเพิ่มอุปกรณ์ที่ต้องการให้ Freeboard ทำการสื่อสารด้วยโดยทำการกดที่ปุ่ม ADD และทำการใส่ Device ID และ Device Token ของ Device ของอุปกรณ์ที่ต้องการสื่อสารด้วย



รูปภาพ 8-5 ภาพแสดงปุ่ม ADD สำหรับเพิ่มอุปกรณ์ในหน้า Freeboard



DATASOURCE

NAME:

DEVICE ID:
Client ID ของ Device ที่ต้องการอ่านข้อมูล

DEVICE TOKEN:
Token ของ Device ที่ต้องการอ่านข้อมูล

SUBSCRIBED TOPICS:
Topic ที่ต้องการ Subscribe

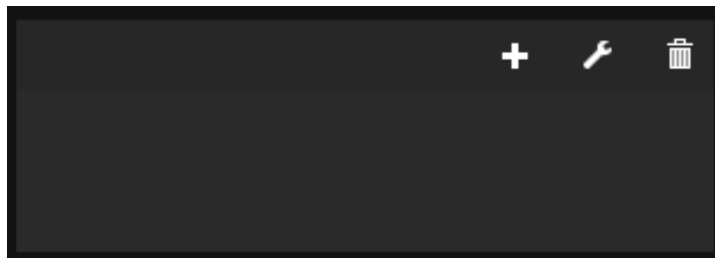
FEED: ☐ YES ☒ NO

SINCE:
Display data points since ... ago.

DOWN SAMPLING:
Resolution of the data points.

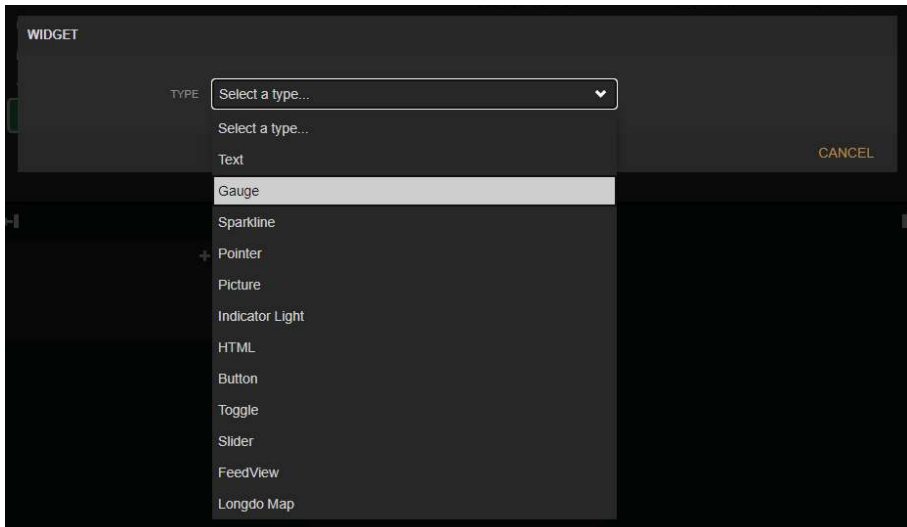
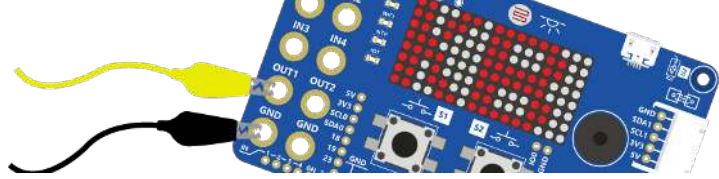
รูปภาพ 8-6 ภาพแสดงการตั้งค่าเพื่อให้ Freeboard สื่อสารกับอุปกรณ์ที่ต้องการ

- จากนั้นให้ทำการ Add Pane เพื่อสร้างพื้นที่สำหรับการเพิ่มอุปกรณ์แสดงผล และควบคุมต่าง ๆ ของ NETPIE Freeboard โดยกดที่เครื่องหมายบวก

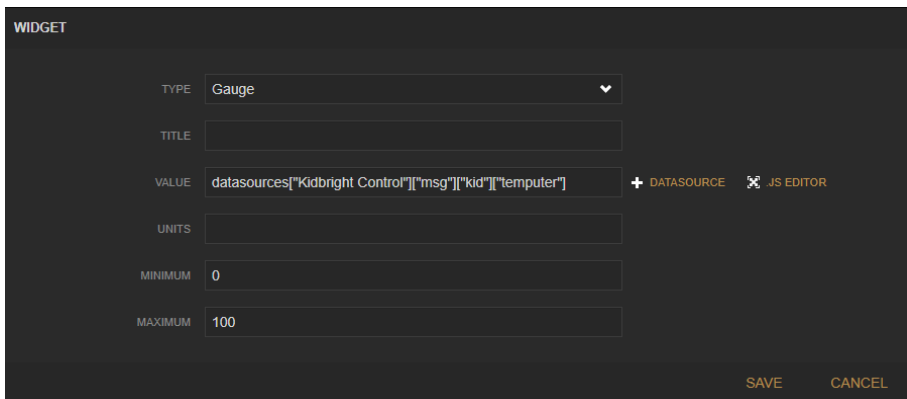


รูปภาพ 8-7 รูปแสดง PANE พื้นที่สำหรับเพิ่มอุปกรณ์แสดงผล และควบคุมต่าง ๆ

- ทำการเลือกอุปกรณ์ Gauge เพื่อสร้างหน้าปัดแสดงผลค่าอุณหภูมิที่ส่งมาจากบอร์ด KidBright และทำการตั้งค่าในช่อง Value ตามหัวข้อ Topic ที่เราทำการส่งมา

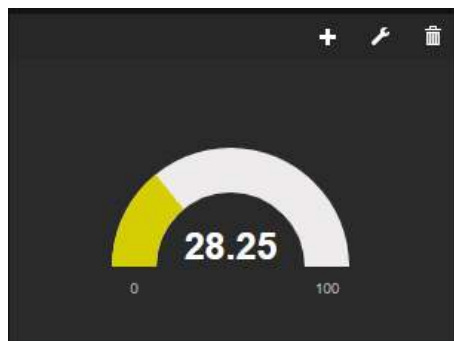


รูปภาพ 8-8 รูปแสดงการเลือก Widget ใน Freeboard

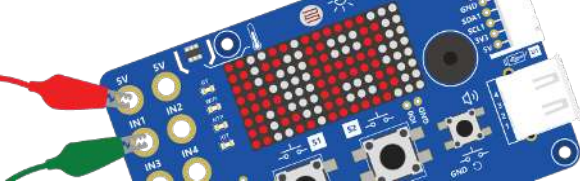


รูปภาพ 8-9 รูปแสดงการตั้งค่า Widget

9. เมื่อทำการตั้งค่าเสร็จเป็นที่เรียบร้อยแล้วให้ทำการ Save และจะมีหน้าต่างแสดงค่าอุณหภูมิที่ส่งมาจากบอร์ด KidBright

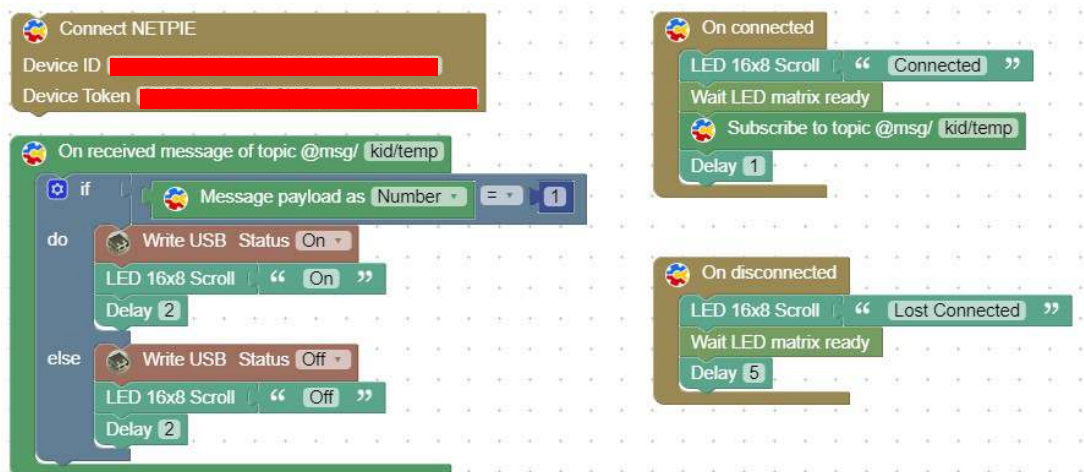


รูปภาพ 8-10 รูปภาพ Gauge แสดงอุณหภูมิ



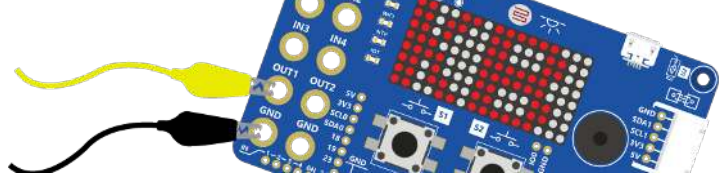
8.2.2 การควบคุมไฟ USB โดยใช้ NETPIE Freeboard

ส่วนที่ 1 การสร้างบล็อกโปรแกรมในฝั่งของ KidBright



รูปภาพ 8-11 รูปภาพโปรแกรมควบคุมไฟ LED จาก Freeboard

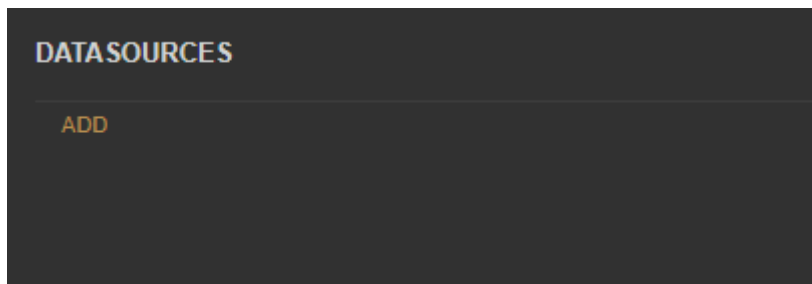
1. ในช่องของ Device ID นำ Device ID จากเว็บไซต์ของ NETPIE มาใส่ให้ถูกต้อง
2. ในช่องของ Device Token นำ Device Token จากเว็บไซต์ของ NETPIE มาใส่ให้ถูกต้อง
3. ในบล็อก On received message of topic @msg/... เป็นบล็อกที่จะทำงานเมื่ออุปกรณ์ได้รับข้อความที่ส่งมาจาก topic ที่กำหนด โดยผู้พัฒนาสามารถใส่คำสั่งต่าง ๆ ไว้ภายในบล็อกเพื่อให้อุปกรณ์ทำงานตามที่ต้องการได้ โดยสามารถอ่านข้อความในรูปแบบข้อความหรือตัวเลขได้จากบล็อก Message payload as ... และในที่นี้เราได้เขียนชุดคำสั่งให้แสดงข้อความ On ที่จอ LED และทำการกำหนดค่าพอร์ต USB ให้อยู่ในสถานะ On เพื่อทำการเปิดไฟ led ที่ได้ทำการเชื่อมต่อไว้ที่ช่อง USB เมื่อได้รับตัวเลข 1 จากอุปกรณ์คู่สื่อสาร
4. ภายในบล็อกคำสั่ง On connected จะทำการเชื่อมต่อกับ NETPIE และทำการตรวจสอบสถานะการเชื่อมต่อของอุปกรณ์หากเชื่อมต่อสำเร็จ จึงจะทำงานตามคำสั่งที่อยู่ภายในบล็อก ในที่นี้เป็นชุดคำสั่งให้แสดงข้อความคำว่า Connected ออกทาง Dot Matrix LED และได้ทำการ Subscribe Topic ที่ได้ทำการตั้งค่าไว้คอยติดตามเหตุการณ์ที่เกิดขึ้นระหว่างอุปกรณ์ เพื่อให้อุปกรณ์สามารถสื่อสารกันได้
5. ภายในบล็อกคำสั่ง On disconnected จะเป็นการตรวจสอบสถานการณ์เชื่อมต่อ โดยระบบจะทำงานตามคำสั่งที่อยู่ภายในเมื่ออุปกรณ์ขาดการเชื่อมต่อกับ NETPIE ในที่นี้เป็น



การแสดงความผิดพลาด Lost Connection ออกทาง Dot Matrix LED และรอ 5 วินาที
เพื่อทำการเชื่อมต่อกับ NETPIE ใหม่อีกครั้ง

ส่วนที่ 2 การตั้งค่าบน NETPIE Freeboard

1. ทำการเพิ่มอุปกรณ์ที่ต้องการให้ Freeboard ทำการสื่อสารด้วยโดยทำการกดที่ปุ่ม ADD
และทำการใส่ Client ID และ Token ของ Device ของอุปกรณ์ที่ต้องการสื่อสารด้วย



รูปภาพ 8-12 ภาพแสดงปุ่ม ADD เพิ่มอุปกรณ์ในหน้า Freeboard

DATASOURCE

NAME: Kidbright Control

DEVICE ID: [Redacted]

Client ID ของ Device ที่ต้องการอ่านข้อมูล

DEVICE TOKEN: [Redacted]

Token ของ Device ที่ต้องการอ่านข้อมูล

SUBSCRIBED TOPICS: @msg/#

Topic ที่ต้องการ Subscribe

FEED: ☐ YES ☒ NO

SINCE: 6

Hour

Display data points since ... ago.

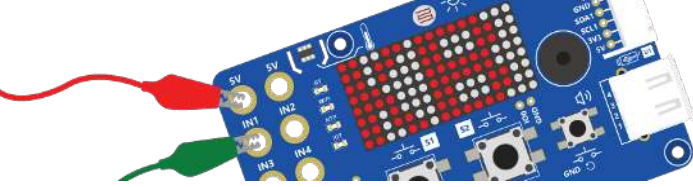
DOWN SAMPLING: 1

Minute

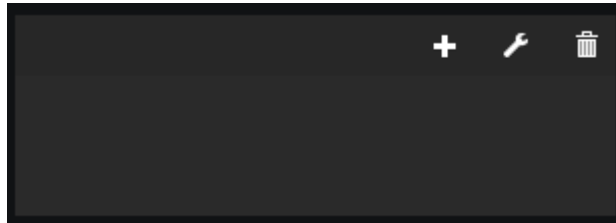
Resolution of the data points.

SAVE CANCEL

รูปภาพ 8-13 ภาพแสดงการตั้งค่าเพื่อให้ Freeboard สื่อสารกับอุปกรณ์ที่เราต้องการ

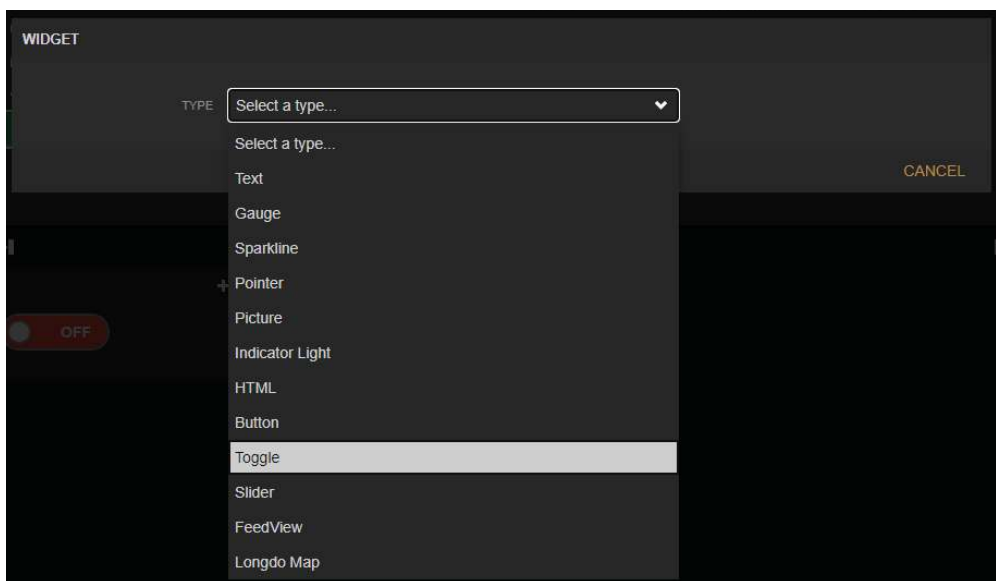


2. จากนั้นให้ทำการ ADD PANE เพื่อสร้างพื้นที่สำหรับการเพิ่มอุปกรณ์แสดงผล และควบคุมต่าง ๆ ของ NETPIE Freeboard

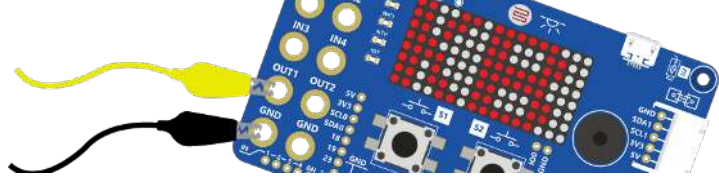


รูปภาพ 8-14 รูปแสดง PANE พื้นที่สำหรับเพิ่มอุปกรณ์แสดงผล และควบคุมต่าง ๆ

3. ทำการเลือกอุปกรณ์ Toggle เพื่อสร้างปุ่มควบคุมการเปิด-ปิด ไฟ LED บนบอร์ด KidBright และทำการตั้งค่า Toggle ต่าง ๆ



รูปภาพ 8-15 รูปแสดงการเลือก Widget Toggle Button



WIDGET

A simple toggle widget that can perform Javascript action.

TYPE: **Toggle** ▼

TOGGLE CAPTION:

TOGGLE STATE: + DATASOURCE JS EDITOR

Add a condition to switch a toggle state here. Otherwise it just toggle by click.

ON TEXT: **ON**

OFF TEXT: **OFF**

ONTOGGLEON ACTION: **netpie["TEST111"].publish("@msg/kidbright/led","1")**
JS code to run when a toggle is switched to ON

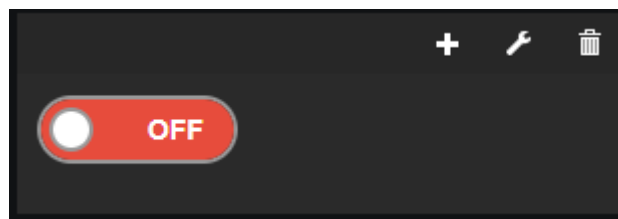
ONTOGGLOFF ACTION: **netpie["TEST111"].publish("@msg/kidbright/led","0")**
JS code to run when a toggle is switched to OFF

ONCREATED ACTION:
JS code to run after a toggle is created

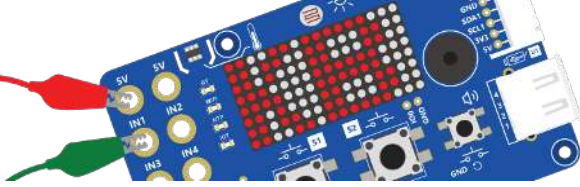
SAVE CANCEL

รูปภาพ 8-16 รูปแสดงการตั้งค่า Toggle

4. เมื่อทำการตั้งค่าเสร็จเป็นที่เรียบร้อยแล้วให้ทำการ Save และจะมีปุ่ม On-Off ขึ้นมาเพื่อควบคุมสถานะไฟ LED บนบอร์ด KidBright



รูปภาพ 8-17 รูปแสดงการตั้งค่าใช้งาน Toggle



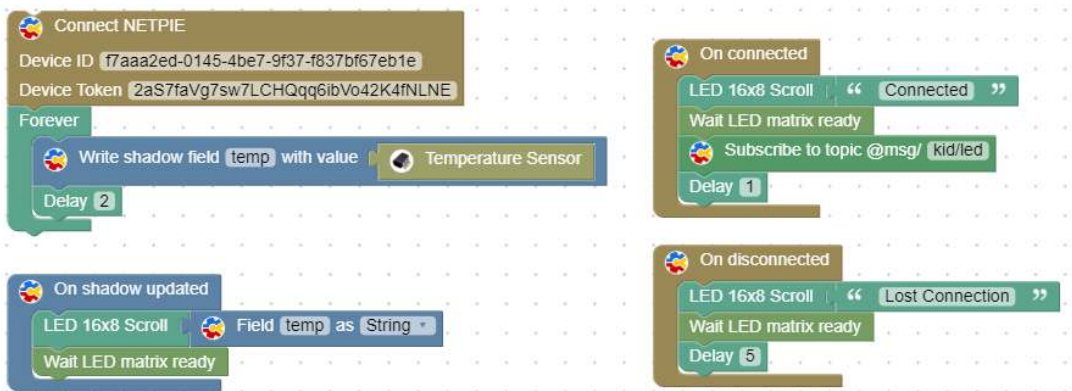
8.3 วิธีการใช้งาน NETPIE 2020 ในรูปแบบ Shadow

ในการใช้งานในรูปแบบ shadow นั้นเราต้องมาทำความรู้จักกันก่อนว่า shadow ใน NETPIE นั้นคืออะไร shadow คือ ฐานข้อมูลเสมือนของอุปกรณ์ เป็นฐานข้อมูลเล็ก ๆ ที่มีคู่อยู่กับอุปกรณ์ (Device) ทุกตัว ใช้สำหรับเก็บข้อมูลต่าง ๆ เกี่ยวกับอุปกรณ์นั้น ๆ (Device Shadow Data) เช่น ข้อมูลที่เกิดจากเซนเซอร์ ข้อมูลการกำหนดองค์ประกอบต่าง ๆ (Device Configuration) เป็นต้น ในการรับส่งข้อมูลในหมวด shadow นั้นจะต้องทำการ subscribe หัวข้อการสื่อสารไว้ที่ @shadow แล้วตามด้วยหัวข้อย่อยที่ต้องการทำการสื่อสารด้วย ซึ่งข้อมูลทุกอย่างที่ทำการสื่อสารระหว่างกันจะถูกบันทึกลงในฐานข้อมูลที่เราเรียกว่า shadow

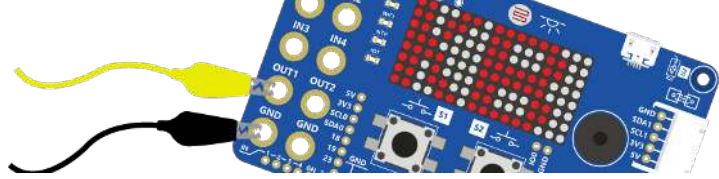
จากนั้นข้อมูลจะถูกจัดระเบียบโดยแผนผังข้อมูลที่เราเรียกว่า Device Schema ซึ่งทำให้เราสามารถจัดการข้อมูลและเข้าถึงได้ง่ายขึ้นอย่างเป็นระเบียบ เช่น การตรวจสอบชนิดข้อมูล (Data Validation) การแปลงข้อมูล (Data Transformation) เช่น เปลี่ยนหน่วยของข้อมูล การเก็บข้อมูลลงใน Time Series Database ทำให้สามารถเข้าถึงข้อมูลได้ในวันและเวลาที่ต้องการ โดย Device Schema จะประกาศในและกำหนดค่าในรูปแบบ JSON

ในตัวอย่างการใช้งาน NETPIE Shadow เราจะทำการส่งค่าอุณหภูมิขึ้นสู่ NETPIE Shadow และเมื่อมีการอัปเดตค่าอุณหภูมิจาก NETPIE เราจะทำการอ่านและดึงกลับมาแสดงบนหน้าจอ Dot Matrix LED

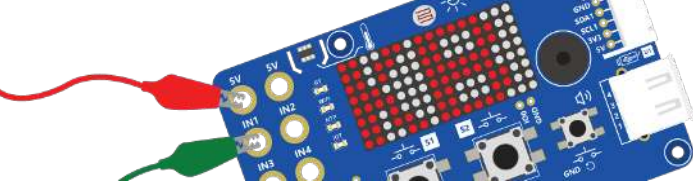
ส่วนที่ 1 การสร้างบล็อกโปรแกรมในฝั่งของ KidBright



รูปภาพ 8-18 รูปโปรแกรมส่งค่าอุณหภูมิแบบ shadow



1. ในช่องของ Device ID นำ Device ID จากเว็บไซต์ของ NETPIE มาใส่ให้ถูกต้อง
2. ในช่องของ Device Token นำ Device Token จากเว็บไซต์ของ NETPIE มาใส่ให้ถูกต้อง
3. ภายในบล็อก Forever เราได้ทำการส่งค่าอุณหภูมิขึ้นสู่ NETPIE ในรูปแบบ shadow เพื่อทำการเก็บค่าลงฐานข้อมูลด้วยคำสั่ง Write shadow field temp with value ... โดยคำสั่งนี้จะทำการส่งค่าขึ้นสู่ฐานข้อมูลในหัวข้อที่ได้ทำการตั้งค่าไว้ในรูปแบบของ JSON บน NETPIE
4. ภายในบล็อก On shadow updated จะคอยตรวจสอบค่าบนฐานข้อมูลว่ามีการ อัปเดตหรือไม่ หากมีการอัปเดตค่าเกิดขึ้นจึงจะทำงานตามคำสั่งที่อยู่ภายใน ซึ่งในที่นี้เราให้ทำการอ่านค่าอุณหภูมิล่าสุดจากฐานข้อมูลและนำมาแสดงบนจอ Dot Matrix LED ผ่าน Field ... as ... ในรูปแบบของตัวแปร
5. ภายในบล็อกคำสั่ง On connected จะทำการเชื่อมต่อกับ NETPIE และทำการตรวจสอบสถานะการเชื่อมต่อของอุปกรณ์หากเชื่อมต่อสำเร็จ จึงจะทำงานตามคำสั่งที่อยู่ภายในบล็อก ในที่นี้เป็นชุดคำสั่งให้แสดงข้อความคำว่า Connected ออกทาง Dot Matrix LED
6. ภายในบล็อกคำสั่ง On disconnected จะเป็นการเช็คการสูญเสียการเชื่อมต่อหากบอร์ดเสียการเชื่อมต่อกับ NETPIE แล้วจึงทำตามคำสั่งที่อยู่ภายใน ในที่นี้เป็นการแสดงข้อความคำว่า Lost Connection ออกทาง Dot Matrix LED และรอ 5 วินาทีเพื่อทำการ connect กับ NETPIE ใหม่อีกครั้ง



ส่วนที่ 2 การตั้งค่า Schema บนหน้าเว็บ NETPIE

ในการตั้งค่าบนหน้าเว็บของ NETPIE นั้นเราต้องทำการเขียน JSON ให้กับ Schema ซึ่งกำหนดชื่อตัวแปร การแปลงค่า ชนิดตัวแปร และระยะเวลาที่ต้องการเก็บลงฐานข้อมูล ซึ่งสามารถกำหนดค่าได้ตามตัวอย่างดังรูป

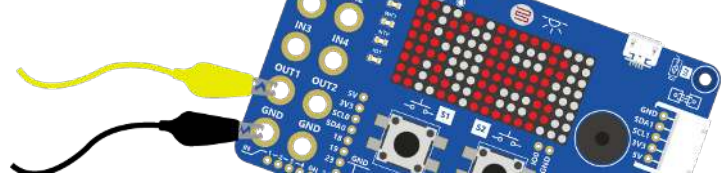
```
1- {  
2-   "additionalProperties": false,  
3-   "properties": {  
4-     "temp": {  
5-       "operation": {  
6-         "store": {  
7-           "ttl": "30d"  
8-         },  
9-       },  
10-      "type": "number"  
11-    }  
12-  }  
13- }
```

รูปภาพ 8-19 รูปภาพการตั้งค่า schema โดย JSON

เมื่อทำการเชื่อมต่อสำเร็จค่าอุณหภูมิที่เราส่งไปนั้นจะถูกเก็บไว้ใน shadow ซึ่งเราสามารถเรียกดูค่าล่าสุดที่เกิดการอัปเดตได้ดังรูป

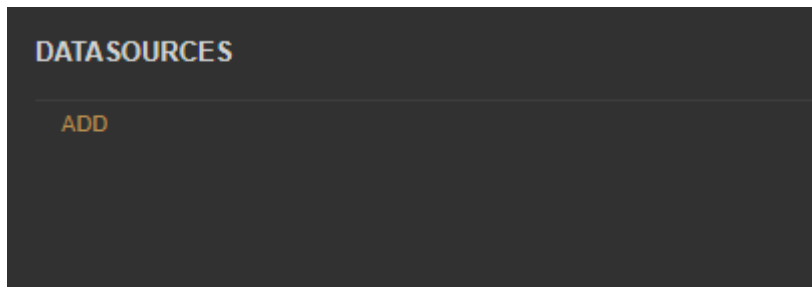
```
Tree  
Select a node...  
object {1}  
temp : 29.75
```

รูปภาพ 8-20 ค่าอุณหภูมิล่าสุดในฐานข้อมูล shadow



ส่วนที่ 3 การตั้งค่า NETPIE Free Board

1. ทำการเพิ่มอุปกรณ์ที่ต้องการให้ Freeboard ทำการสื่อสารด้วยโดยทำการกดที่ปุ่ม ADD และทำการใส่ Client ID และ Token ของ Device ของอุปกรณ์ที่ต้องการสื่อสารด้วย และทำการเปิดระบบ FEED ขึ้นเพื่อเรียกใช้ฐานข้อมูลของอุปกรณ์



รูปภาพ 8-21 ภาพแสดงปุ่ม ADD เพิ่มอุปกรณ์ในหน้า Freeboard

DATASOURCE

NAME:

DEVICE ID:
Client ID ของ Device ที่ต้องการหาข้อมูล

DEVICE TOKEN:
Token ของ Device ที่ต้องการหาข้อมูล

SUBSCRIBED TOPICS:
Topic ที่ต้องการ Subscribe

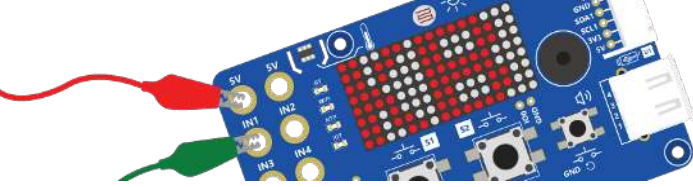
FEED: ☒ YES

SINCE:
Display data points since ... ago.
Hour

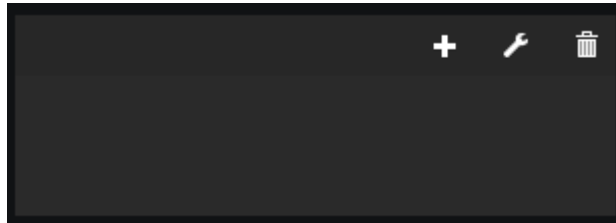
DOWN SAMPLING:
Resolution of the data points.
Minute

SAVE CANCEL

รูปภาพ 8-22 ภาพแสดงการตั้งค่าเพื่อให้ Freeboard สื่อสารกับอุปกรณ์ที่เราต้องการ

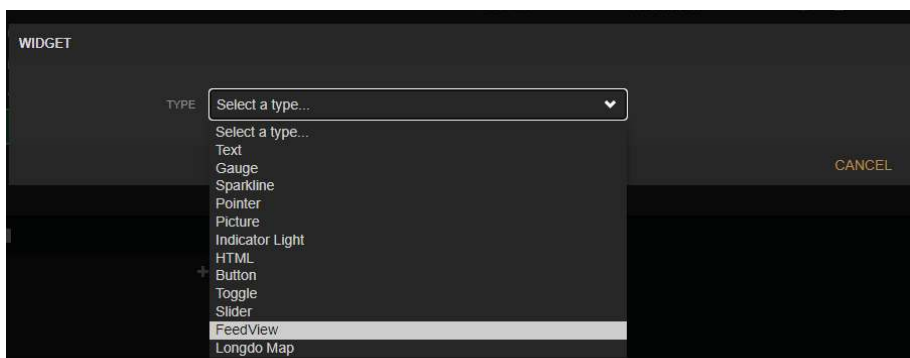


2. จากนั้นให้ทำการ ADD PANE เพื่อสร้างพื้นที่สำหรับการเพิ่มอุปกรณ์แสดงผล และควบคุมต่าง ๆ ของ NETPIE Freeboard

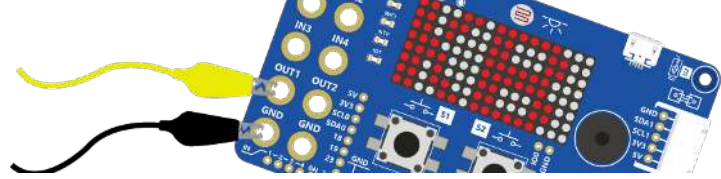


รูปภาพ 8-23 รูปแสดง PANE พื้นที่สำหรับเพิ่มอุปกรณ์แสดงผล และควบคุมต่าง ๆ

3. ทำการเลือกอุปกรณ์ NETPIE FeedView เพื่อนำข้อมูลจากฐานข้อมูลมาแสดงในรูปแบบกราฟโดยให้ทำการตั้งค่าต่าง ๆ ดังรูป โดยมี FILTER เป็นชื่อตัวแปรที่ต้องการนำค่ามาแสดง (ชื่อตัวแปรที่ทำการตั้งชื่อไว้บน JSON)



รูปภาพ 8-24 รูปแสดงการเลือก FeedView



WIDGET

TYPE: **FeedView**

TITLE: **Temperature in the room**

DATA SOURCE: **datasources["TEST111"]["feed"]** + DATASOURCE X JS EDITOR

FILTER: **temp**
Data fields separated with commas e.g. temp,humidity,light. Blank means display all fields.

TYPE OF CHART: **Line**

X AXIS TITLE:

Y AXIS TITLE:

BEGIN AT 0: ☐ NO

LINE COLORS:
enter the color set separated by commas e.g. #FF0000,0000FF,0000FF or leave blank for the default color set

MAKER: **YES** ☒

MULTIPLE AXIS: **YES** ☒

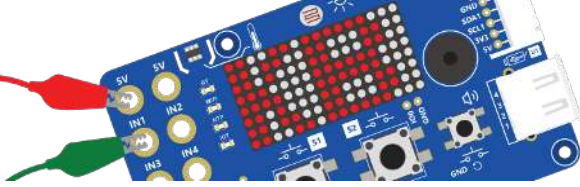
HEIGHT BLOCKS: **4**

SAVE CANCEL

รูปภาพ 8-25 รูปแสดงการตั้งค่า FeedView

4. เมื่อทำการตั้งค่าเสร็จเป็นที่เรียบร้อยแล้วให้ทำการ SAVE จะมีกราฟแสดงค่าอุณหภูมิเทียบกับเวลาแสดงขึ้นมาในหน้า Freeboard

นอกจากนี้แล้ว NETPIE สามารถเชื่อมต่อกับ server ภายนอกหรือที่เรียกว่า Web Hooks ได้อีกด้วยซึ่งใช้ในการแจ้งเตือน หรือส่งข้อความกับ server หรือ API ต่าง ๆ ภายนอกได้ ซึ่งผู้ใช้สามารถศึกษาข้อมูลเพิ่มเติมเกี่ยวกับการใช้งาน NETPIE Shadow และ Event Hooks เพิ่มเติมได้จาก <https://netpie.io/tutorials/NodeMCU>



บทที่ 9

การใช้งาน Line Notify ร่วมกับบอร์ด KidBright

9.1 รู้จักกับ Line Notify

แอปพลิเคชัน Line ถือเป็นแอปพลิเคชันยอดนิยมสำหรับชาวไทยที่ถูกนำมาใช้เพื่อติดต่อสื่อสารในชีวิตประจำวัน ในขณะนี้แอปพลิเคชัน Line ได้มีบริการที่ชื่อ Line Notify ซึ่งเป็นบริการที่อนุญาตให้ระบบอัตโนมัติต่าง ๆ ตัวอย่างเช่น แอปพลิเคชันในแพลตฟอร์มจำพวกเว็บ มีมือถือ หรือบอร์ดไมโครคอนโทรลเลอร์ที่มีการเชื่อมต่ออินเทอร์เน็ต สามารถส่งข้อความเข้าสู่ห้องแชทของแอปพลิเคชัน Line ได้

Line Notify จะทำตัวเสมือนกับเป็นเพื่อนของเราในแอปพลิเคชัน Line และ ข้อความที่ส่งมาหาเราจะอยู่ในรูปแบบ [ชื่อ]: [ข้อความ] ซึ่งนักพัฒนาจะเป็นผู้กำหนดชื่อและข้อความได้เอง เช่น หากกำหนดชื่อเป็น “KidBright” และกำหนดข้อความเป็น “สวัสดี” ก็จะได้รับข้อความเข้าห้องแชทเป็น “KidBright: สวัสดี” เป็นต้น สำหรับการตั้งค่าการใช้งานจะอยู่ในส่วนต่อไป

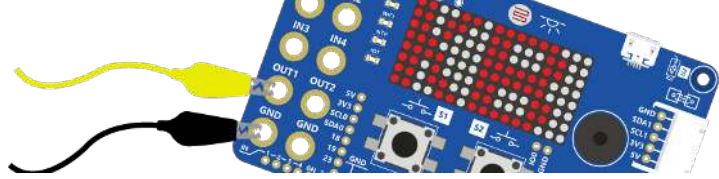
สามารถดูข้อมูลเพิ่มเติมที่เกี่ยวข้องกับ Line Notify ได้จากเว็บไซต์ของ Line Notify ที่ <https://notify-bot.line.me/en/>

9.2 การเพิ่ม Line Notify เป็นเพื่อน

สามารถทำได้ง่าย ๆ โดยการสแกน QR Code นี้ผ่านแอปพลิเคชัน Line



รูปภาพ 9-1 ภาพแสดง QR Code สำหรับเพิ่ม Line Notify เป็นเพื่อน

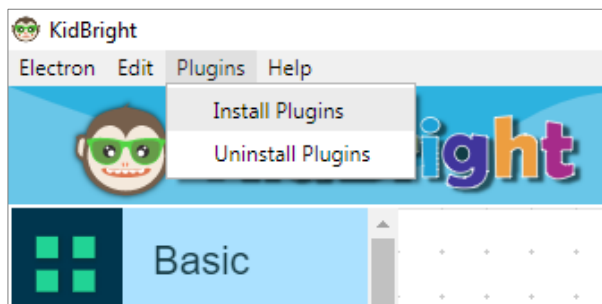


9.3 การติดตั้ง Plugin ใน KidBright IDE เพื่อใช้งาน Line Notify

มีนักพัฒนาชาวไทยหลายคนที่ร่วมด้วยช่วยกันพัฒนา Plugin สำหรับบอร์ด KidBright หนึ่งในนั้นคือนักพัฒนาที่ใช้ชื่อว่า maxpomer ที่เป็นผู้พัฒนา Plugin ชื่อ linenotify ซึ่งถูกนำมาใช้สำหรับส่งข้อความ รูปภาพ และสติ๊กเกอร์ผ่านบริการ Line Notify ในการนี้เองนักพัฒนาคนอื่น ๆ จึงสามารถใช้ block เพื่อเชื่อมต่อกับบริการ Line Notify ได้ในโปรแกรม KidBright IDE

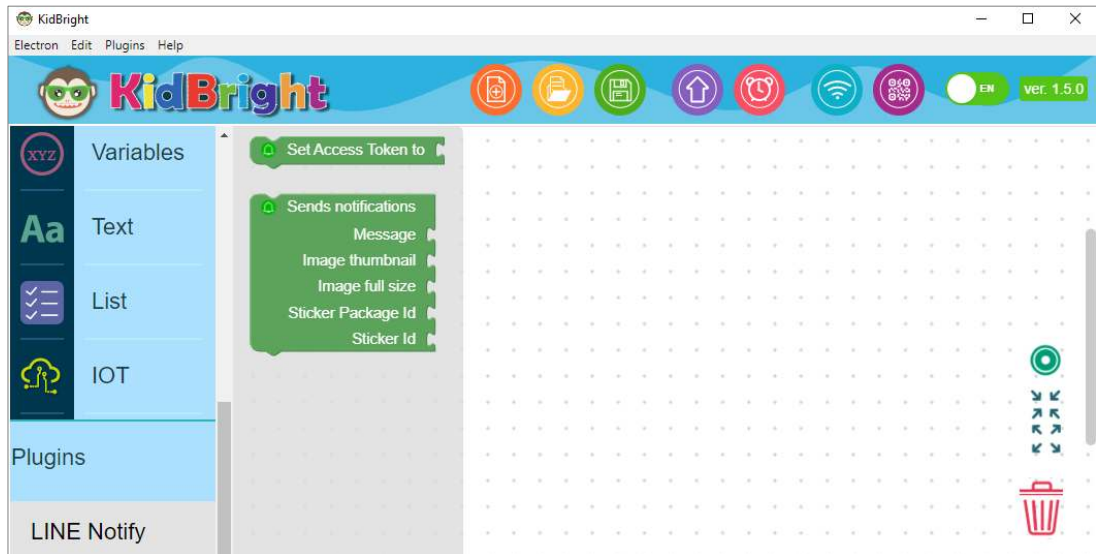
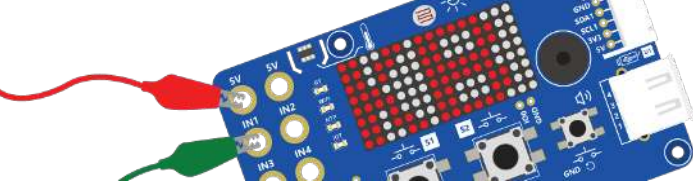
ขั้นตอนการติดตั้ง Plugin linenotify ใน KidBright IDE สามารถทำได้ ดังนี้

1. ดาวน์โหลด Plugin ได้ที่เว็บไซต์ Github ของ kidstore ซึ่งอยู่ที่ <https://github.com/kidbright-plugins-by-kidstore/linenotify/releases/> ให้ทำการดาวน์โหลดไฟล์ zip
2. กดที่เมนู Plugins แล้วจึงกด Install Plugins



รูปภาพ 9-2 ภาพแสดงเมนู Install Plugins

3. เลือกไฟล์ .zip ที่ดาวน์โหลดมา แล้วกดปุ่ม Open
4. โปรแกรม KidBright IDE จะทำการติดตั้ง Plugin แล้วจะทำการรีสตาร์ทโปรแกรมให้โดยอัตโนมัติ
5. สามารถตรวจสอบว่าการติดตั้งสำเร็จ ได้โดยดูว่ามี Block สำหรับ Line Notify ในหมวด Plugins

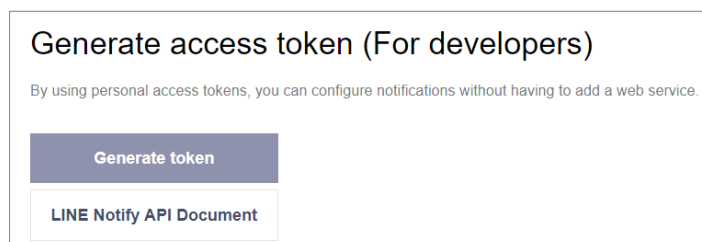


รูปภาพ 9-3 ภาพแสดง Block คำสั่งสำหรับ Line Notify

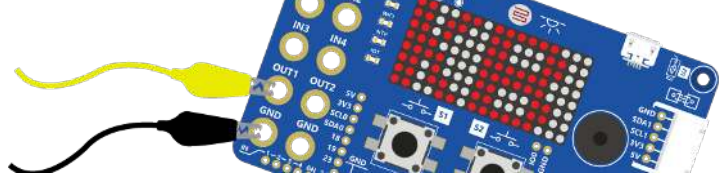
9.4 วิธีการใช้งาน Line Notify ร่วมกับบอร์ด KidBright

เนื่องจากการใช้งาน Line Notify สามารถเข้าใจได้ง่าย จึงจะนำเสนอการใช้งานโดยการแสดงตัวอย่างให้เห็น โดยตัวอย่างที่นำมาแสดงคือการส่งค่าอุณหภูมิเข้าสู่ห้องแชท ซึ่งมีวิธีการทำดังนี้

1. บัญชี Line จะใช้ได้เฉพาะบัญชีที่ลงทะเบียน email แล้วเท่านั้น หากยังไม่เคยลงทะเบียนมาก่อน ให้ลงทะเบียนในแอปพลิเคชัน Line ก่อน โดยกดไปที่เมนู Setting > Account แล้วทำการตั้งค่า Email และ Password พร้อมทั้งตั้งค่า Allow Login เพื่อให้สามารถใช้งานผ่านเว็บได้
2. เข้าไปที่หน้าเว็บ <https://notify-bot.line.me/my/> แล้ว login ด้วยบัญชี Line โดยกรอก Email และ Password
3. เมื่อผ่านหน้า Login แล้วในหน้าเว็บจะเป็นดังรูปภาพ 9-4 ให้กดที่ปุ่ม Generate token



รูปภาพ 9-4 ภาพแสดงหน้าเว็บเพื่อสร้าง Token แทนตัวตนที่ใช้เชื่อมต่อกับบริการ Line Notify



4. จะมีหน้าต่าง Pop-Up ขึ้นมาดังรูปภาพ 9-5 เป็นหน้าต่างสำหรับกรอกข้อมูลที่จำเป็นสำหรับการสร้าง token

Generate token

Please enter a token name to be displayed before each notification.

KidBright

Select a chat to send notifications to.

Search by group name

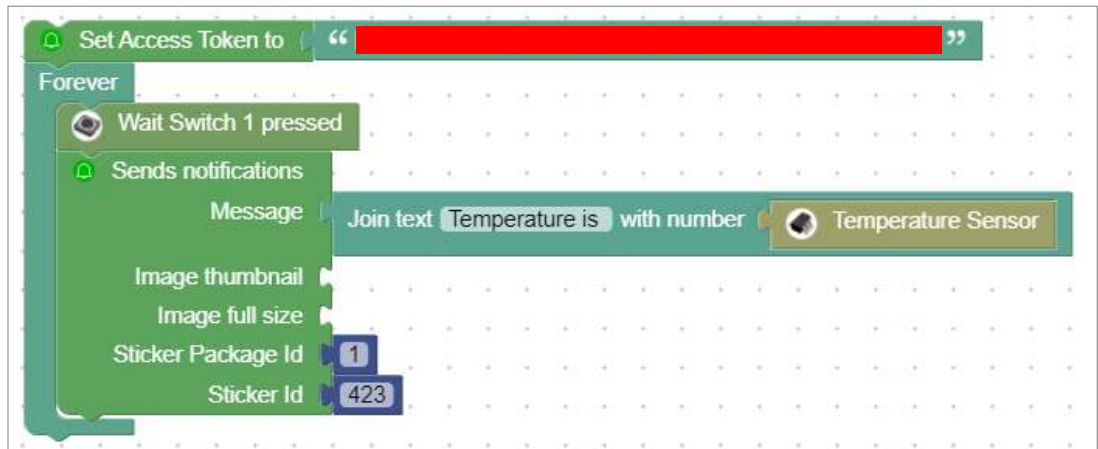
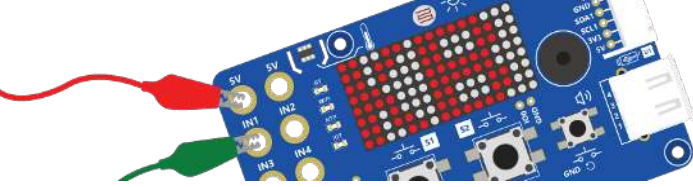
1-on-1 chat with LINE Notify

Note: Revealing your personal access token can allow a third party to obtain the names of your connected chats as well as your profile name.

Generate token

รูปภาพ 9-5 ภาพแสดงหน้าต่าง Pop-Up เพื่อการสร้าง Token

5. ในช่องที่ 1 ให้กรอกชื่อที่จะใช้แสดง เช่น KidBright เป็นต้น
6. ในช่องที่ 2 จะให้เลือกห้องแชทที่ข้อความจะถูกแสดง ซึ่งสามารถเลือกได้ว่าจะเป็นห้องแชทสำหรับตัวเองเท่านั้น (1-on-1 chat with Line Notify) หรือห้องแชทสำหรับกลุ่ม (เลือกได้จากชื่อกลุ่ม)
7. กดปุ่ม Generate token
8. ระบบจะสร้าง Token ใหม่ และมีหน้าต่างแสดง Token นั้น ทั้งนี้ Token จะเป็นอักขระที่ประกอบไปด้วยตัวเลขหรือตัวอักษร ให้กดที่ปุ่ม Copy และจดบันทึก Token นี้เก็บไว้ใช้ในภายหลัง
9. ต่อ Block คำสั่ง ดังรูปภาพ 9-6 โดยการ Paste Token ใส่ในช่อง “Set Access Token to”



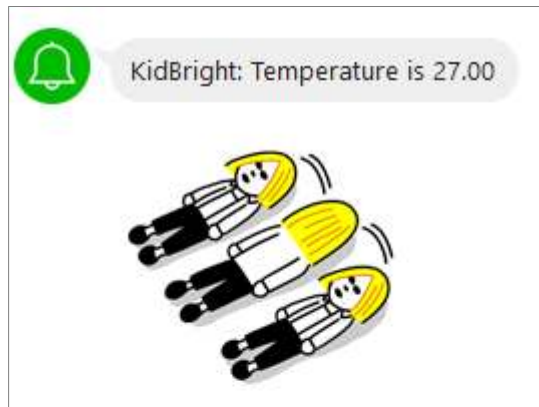
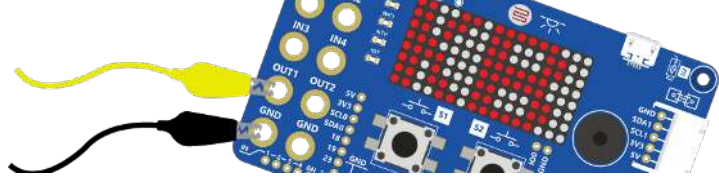
รูปภาพ 9-6 ภาพการต่อ Block คำสั่ง เพื่อใช้งาน Line Notify

10. ตั้งค่าการเชื่อมต่ออินเทอร์เน็ต โดยกดที่สัญลักษณ์ Wi-Fi และตั้งค่าการเชื่อมต่อได้แก่ SSID และ Password ดังรูปภาพ 9-7 โดย SSID คือชื่อของสัญญาณไวไฟ และ Password คือรหัสผ่าน แล้ว OK



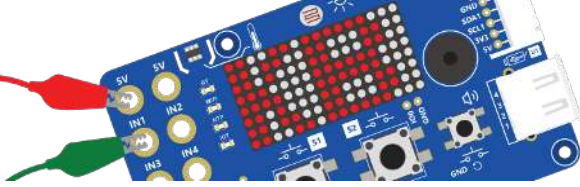
รูปภาพ 9-7 ภาพแสดงการตั้งค่า Wi-Fi

11. ทำการ Upload โดยกดที่ปุ่ม Upload
12. เมื่อการ Upload เสร็จสิ้น ให้ทดลองกดที่ปุ่ม SW1 บนบอร์ด KidBright
13. ตรวจสอบว่ามีข้อความเข้าที่ห้องแชทที่ตั้งค่าเอาไว้ ซึ่งข้อความจะอยู่ในรูปแบบที่กำหนด โดยมีชื่อคือ KidBright และข้อความแสดงอุณหภูมิ พร้อมทั้งมี Sticker ตามที่กำหนดอีกด้วย ดังรูปภาพ 9-8



รูปภาพ 9-8 ภาพแสดงตัวอย่างข้อความที่จะได้รับจาก Line Notify

14. ผู้ใช้งานสามารถเปลี่ยนเลข Sticker Package ID และ Sticker ID ของ Sticker ได้โดยดูจาก link นี้ https://devdocs.line.me/files/sticker_list.pdf



บทที่ 10

เปรียบเทียบการเขียนโปรแกรมด้วย KidBright IDE และ Arduino

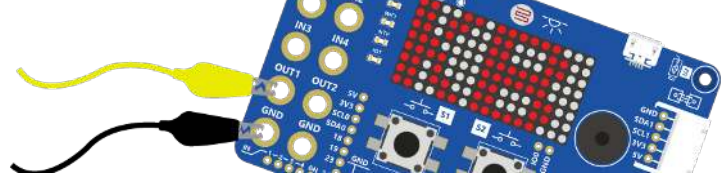
10.1 การเขียนโปรแกรมไมโครคอนโทรลเลอร์ด้วย Arduino

Arduino คือ ชื่อของโครงการโอเพนซอร์สอิเล็กทรอนิกส์ทั้งเชิงฮาร์ดแวร์และซอฟต์แวร์ ซึ่งโครงการ Arduino มีการผลิตบอร์ดไมโครคอนโทรลเลอร์หลากหลายรุ่นและมี IDE สำหรับเขียนโปรแกรมเพื่อควบคุมบอร์ดไมโครคอนโทรลเลอร์เหล่านั้น โดยในเริ่มแรก Arduino มักถูกออกแบบโดยใช้ชิปตระกูล AVR แต่ในเวลาถัดมา มีนักพัฒนาที่ทำฟังก์ชันเสริมให้โปรแกรม Arduino เพื่อควบคุมบอร์ดไมโครคอนโทรลเลอร์จากผู้ผลิตรายอื่น ๆ ได้ เช่น ESP8266 และ ESP32 จากบริษัท Espressif เป็นต้น

ทั้งนี้ บอร์ด KidBright ซึ่งเป็นบอร์ดที่ใช้ไมโครคอนโทรลเลอร์ ESP32 ทำให้เราสามารถเขียนโปรแกรม Arduino ในการเขียนโปรแกรมบอร์ด KidBright ได้



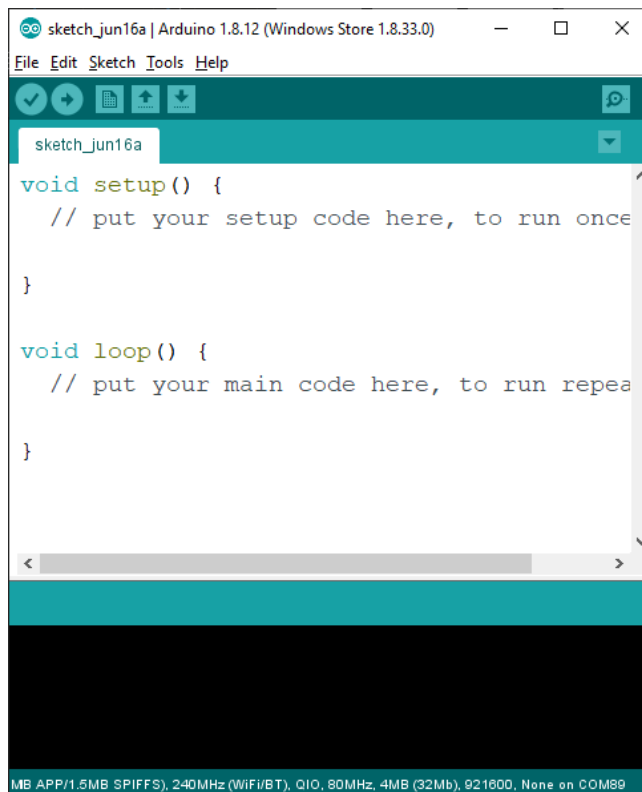
รูปภาพ 10-1 ภาพโลโก้ของ Arduino Project



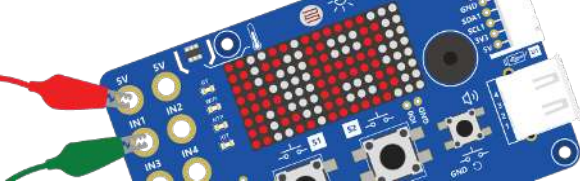
10.2 การติดตั้งโปรแกรม Arduino IDE

โปรแกรม Arduino ถูกพัฒนาโดยภาษา Java ทำให้สามารถทำงานได้บนคอมพิวเตอร์เกือบทุกเครื่องไม่ว่าจะเป็นระบบปฏิบัติการ Windows, macOS หรือ Linux โดยสามารถดาวน์โหลดโปรแกรมได้จากเว็บไซต์ <https://www.arduino.cc/en/main/software> และการติดตั้งโปรแกรมสามารถทำได้ตามปรกติตามที่ระบบปฏิบัติการนั้น ๆ ได้แก่ Windows Installer, Microsoft Store App, Mac OS X Application, Ubuntu Software Center, Snap Store, AUR, Shell Script ฯลฯ

เมื่อติดตั้งสำเร็จให้ทดลองเปิดโปรแกรม Arduino จะได้โปรแกรมที่มีหน้าต่างดังรูปภาพ 10-2



รูปภาพ 10-2 ภาพแสดงหน้าจอ User Interface ของโปรแกรม Arduino IDE



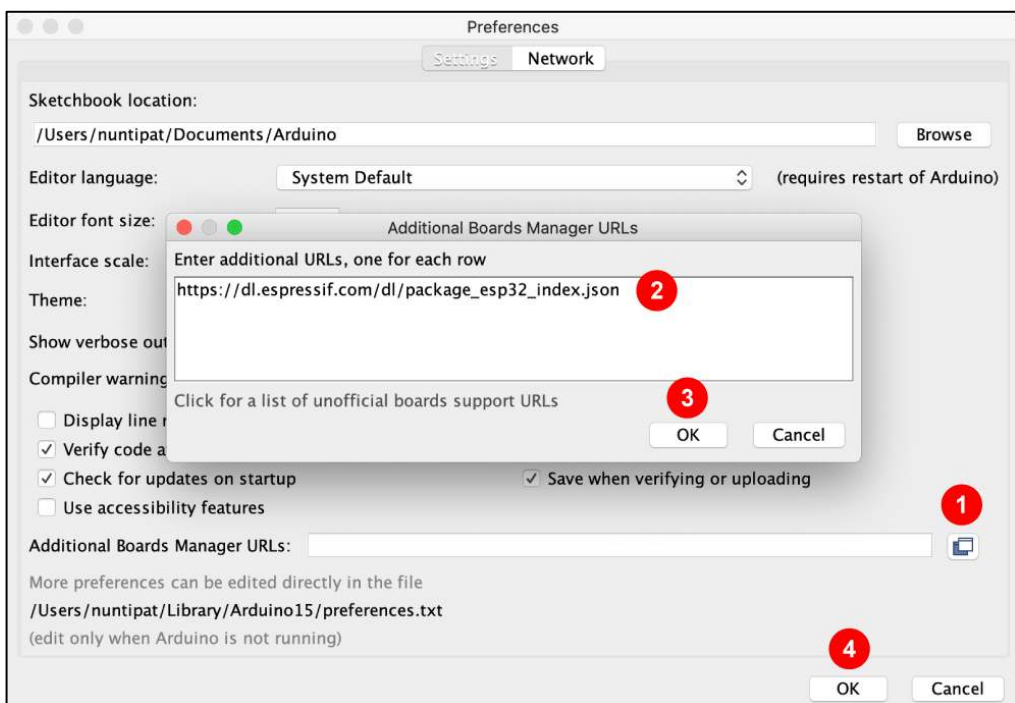
10.3 การตั้งค่าในโปรแกรม Arduino IDE เพื่อเขียนโปรแกรมบอร์ด KidBright

ขั้นตอนการตั้งค่าในโปรแกรม Arduino เพื่อใช้สำหรับการเขียนโปรแกรมควบคุมบอร์ด KidBright มีดังนี้

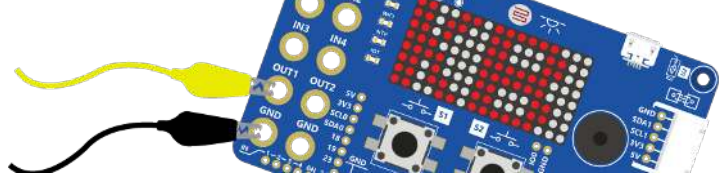
10.3.1 ติดตั้งบอร์ด ESP32 ใน Arduino IDE

โปรแกรม Arduino IDE มีคำอธิบายบอร์ดและโปรแกรมเมอร์เริ่มต้นเพื่อใช้งานกับไมโครคอนโทรลเลอร์ตระกูล AVR หากเราต้องการใช้ Arduino IDE เพื่อเขียนโปรแกรมบอร์ด KidBright ซึ่งเป็นบอร์ดที่ใช้ไมโครคอนโทรลเลอร์ตระกูล ESP32 เราจะต้องเพิ่มคำอธิบายบอร์ดและโปรแกรมเมอร์ในการตั้งค่าครั้งแรก ทั้งนี้ การเพิ่มคำอธิบายบอร์ดและโปรแกรมเมอร์สำหรับบอร์ด ESP32 เข้าสู่โปรแกรม Arduino สามารถทำได้ดังนี้

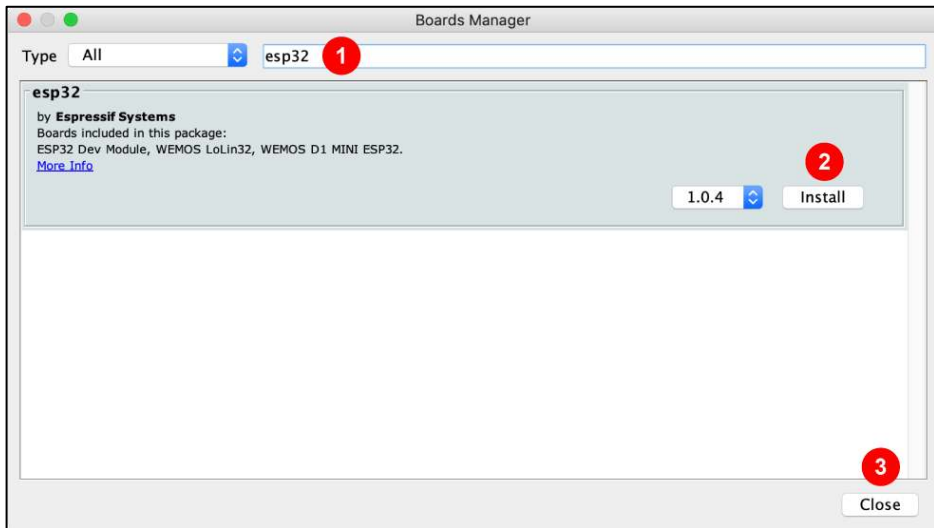
1. ไปที่เมนู File > Preferences ในแถบ Additional Boards Manager URLs ให้ทำการเพิ่ม URL https://dl.espressif.com/dl/package_esp32_index.json ดังรูปภาพ 10-3 ทั้งนี้หากมีข้อมูลอยู่หลาย URLs ให้ใส่แต่ละ URL แยกบรรทัดกัน โดยมีเพียงบรรทัดละ 1 URL กดปุ่ม Ok เพื่อเปิดหน้าต่าง Additional Board Manager URLs และหน้าต่าง Preferences



รูปภาพ 10-3 ภาพแสดงวิธีการเพิ่ม Board Manager URL ใน Arduino

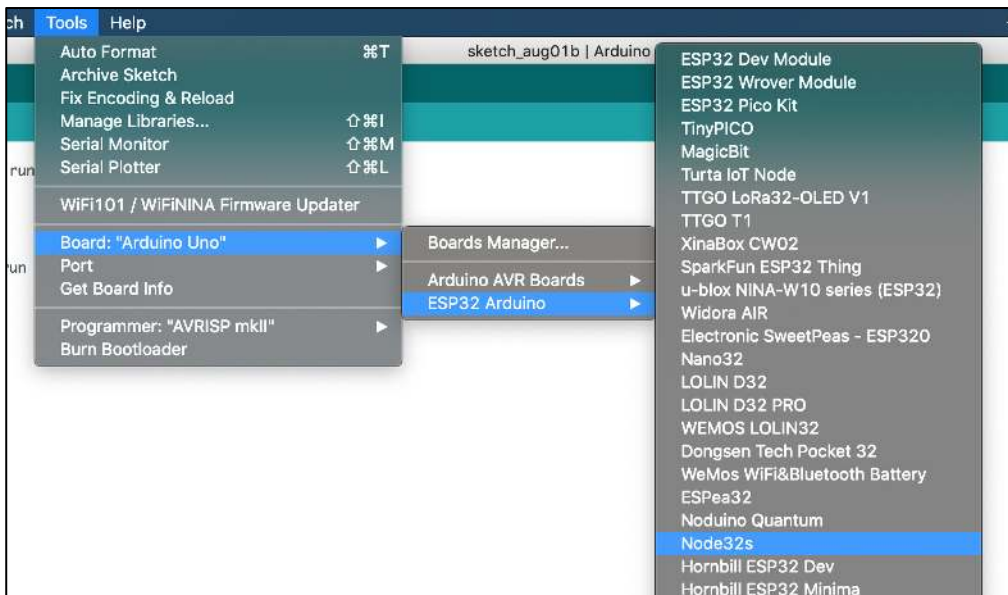


2. ไปที่ Tools > Board > Boards Manager แล้วทำการค้นหาคำว่า ESP32 และกดที่ปุ่ม Install ดังรูปภาพ 10-4 จากนั้นกด Close เพื่อปิดหน้าต่าง Boards Manager

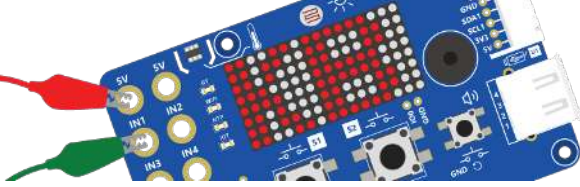


รูปภาพ 10-4 ภาพแสดงการติดตั้งบอร์ด ESP32 โดยใช้ Boards Manager

3. เลือกบอร์ดโดยไปที่เมนู Tools > Board: “xxx” แล้วเลือก Node32s ดังรูปภาพ 10-5



รูปภาพ 10-5 ภาพแสดงวิธีการตั้งค่าบอร์ดในโปรแกรม Arduino IDE



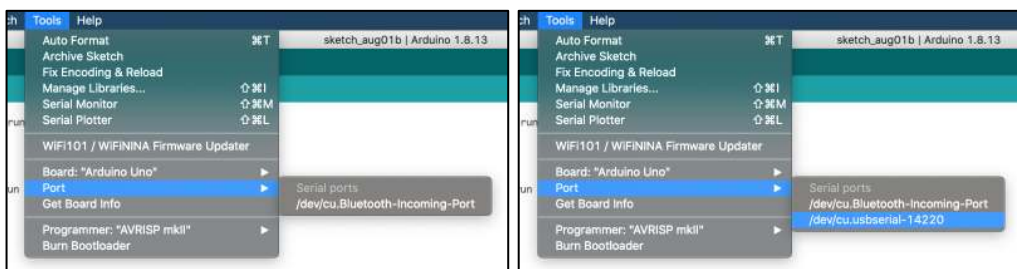
10.3.2 ตรวจสอบการเชื่อมต่อกับบอร์ด KidBright

เฉกเช่นเดียวกับ KidBright IDE การใช้งาน Arduino IDE เพื่อเขียนโปรแกรมควบคุมบอร์ด KidBright มีความจำเป็นต้องทำให้โปรแกรม Arduino IDE มองเห็นบอร์ด ทั้งนี้ เราเชื่อมต่อคอมพิวเตอร์ของเราเข้ากับบอร์ด KidBright ผ่านสาย USB ซึ่งเป็นวิธีการส่งข้อมูลแบบ Serial ทั้งนี้ ในแต่ละระบบปฏิบัติการคอมพิวเตอร์จะตั้งชื่อการเชื่อมต่อของบอร์ด KidBright กับคอมพิวเตอร์นี้ว่า Serial Port หรือเรียกสั้น ๆ ว่า Port

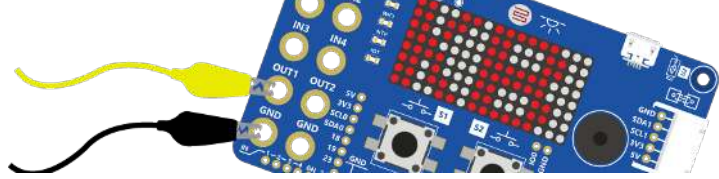
หากคอมพิวเตอร์ที่ใช้ไม่เคยได้รับการติดตั้ง Driver สำหรับบอร์ด KidBright มาก่อน คอมพิวเตอร์จะไม่สามารถตั้งชื่อให้กับการเชื่อมต่อได้ จึงจำเป็นต้องติดตั้ง Driver สำหรับบอร์ด KidBright ในคอมพิวเตอร์ก่อนเสมอ ยกเว้นกรณีที่เครื่องคอมพิวเตอร์บางเครื่องจะมี Driver ติดตั้งมาให้แล้วหรือติดตั้งโดยอัตโนมัติเมื่อเสียบอุปกรณ์ก็จะไม่จำเป็นต้องติดตั้ง Driver ใหม่ให้ซ้ำซ้อนอีก นอกจากนี้ โปรแกรม Arduino IDE ยังสามารถทำงานได้กับบอร์ดจากหลากหลายผู้ผลิต ทำให้โปรแกรม Arduino IDE มีทางเลือกให้นักพัฒนาสามารถเลือกได้ว่าจะใช้ Port ใดในการอัปโหลดโปรแกรมเข้าสู่บอร์ดอีกด้วย

ขั้นตอนต่อไปนี้จะเป็นการติดตั้ง Driver และวิธีการเลือกการเชื่อมต่อที่ถูกต้องในโปรแกรม Arduino IDE สำหรับใช้กับบอร์ด KidBright

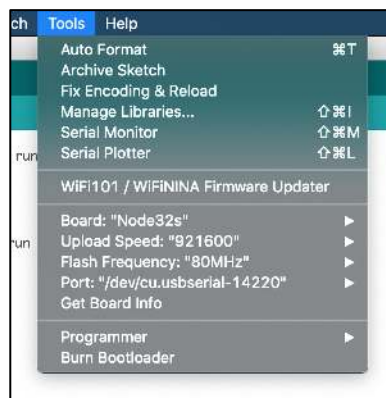
1. หากคอมพิวเตอร์และบอร์ด KidBright กำลังเสียบสาย USB อยู่ ให้ถอดสาย USB ก่อน
2. ที่เมนู Tools > Port จะเป็นเมนูที่ใช้สำหรับเลือกว่า จะใช้การเชื่อมต่อ Serial Port ใด ซึ่งเป็นไปได้หลายกรณี เช่น กรณีที่คอมพิวเตอร์ไม่มีการเชื่อมต่อกับ Serial Port ใดเลยจะได้ว่าเมนู Port จะเป็นสีเทา ดังรูปภาพ 10-6 (ซ้าย) และกรณีที่คอมพิวเตอร์มีการเชื่อมต่อ Serial Port อยู่แล้วจะได้ว่าเมนู Port จะสามารถกดเพื่อดูได้ว่ามี Port ใดอยู่บ้าง ดังรูปภาพ 10-6 (ขวา)



รูปภาพ 10-6 ภาพแสดงเมนู Port เชื่อมต่อ ในโปรแกรม Arduino IDE

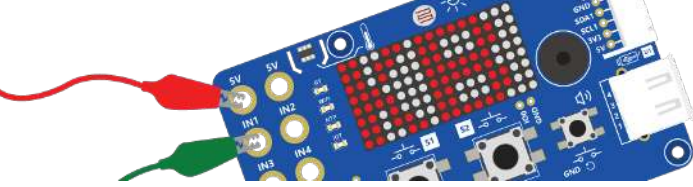


3. ให้จำว่าในขณะที่เรายังไม่เสียบสายนี้มี Port อะไรอยู่บ้าง
4. เสียบสาย USB เพื่อเชื่อมต่อคอมพิวเตอร์และบอร์ด KidBright
5. ตรวจสอบที่เมนู Port อีกครั้ง ว่าโปรแกรม Arduino พบบอร์ด KidBright หรือไม่โดยดูว่ามี Port เพิ่มขึ้นมากกว่าตอนที่ไม่มีเสียบสายหรือไม่ หากมีเพิ่มขึ้นให้เลือกการเชื่อมต่อนั้น
6. เมื่อเลือกบอร์ดและการเชื่อมต่อเรียบร้อยแล้ว จะเห็นการตั้งค่าเมนูเป็นดังรูปภาพ 10-7 ซึ่งในช่อง Port อาจจะแตกต่างกันได้ ตามการเชื่อมต่อที่ระบบปฏิบัติการตั้งชื่อให้ หากใช้ระบบปฏิบัติการ Windows จะได้ชื่อนำหน้าว่า COM ตามด้วยตัวเลข หรือหากใช้ระบบปฏิบัติการ macOS หรือ Linux จะได้ชื่อขึ้นต้นว่า /dev/tty หรือ /dev/cu ทั้งนี้ ข้อสำคัญคือให้เลือกการเชื่อมต่อที่เพิ่มขึ้นหลังจากที่เราเสียบสาย USB เพื่อเป็นการยืนยันว่าเราได้เลือกใช้การเชื่อมต่อกับบอร์ด KidBright อยู่

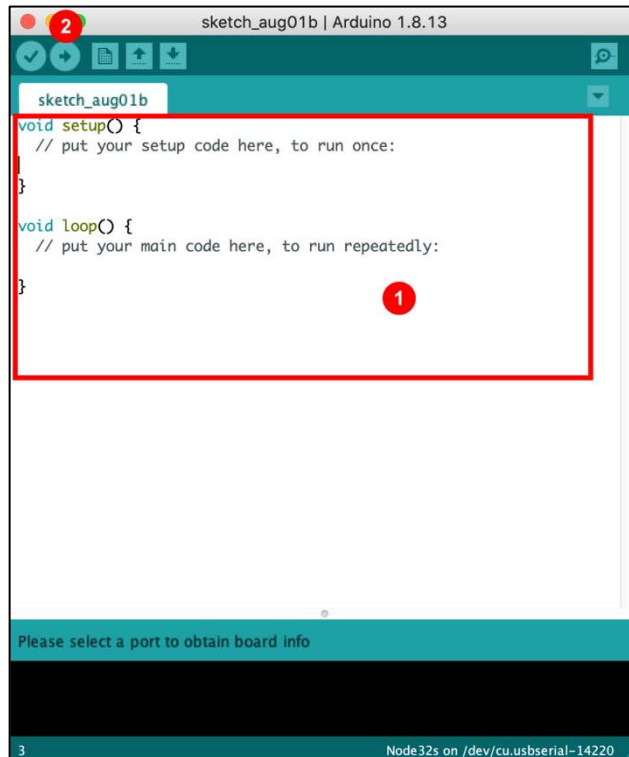


รูปภาพ 10-7 ภาพแสดงบอร์ดและการเชื่อมต่อที่ถูกเลือกแล้ว

7. หากไม่พบการเชื่อมต่อเพิ่มขึ้น เป็นไปได้ว่าระบบปฏิบัติการอาจจะยังไม่มี Driver ของบอร์ด KidBright ให้ทำการติดตั้ง Driver ของบอร์ด KidBright ก่อน ทั้งนี้ ต้องเลือก Driver ให้ตรงกับรุ่นของบอร์ด KidBright ด้วย ซึ่งสามารถดาวน์โหลดได้จาก <https://www.kid-bright.org/kidbright/downloads/> เว็บไซต์นี้จะมี Driver แยกตามรุ่นของบอร์ดอยู่แล้ว ให้ดาวน์โหลดและติดตั้ง Driver ให้เสร็จเรียบร้อย แล้วจึงตรวจสอบที่เมนู Port ใหม่อีกครั้ง
8. หากติดตั้ง Driver แล้วยังคงไม่พบการเชื่อมต่ออีก และสามารถทดลองกับคอมพิวเตอร์เครื่องอื่นได้ หากยังคงไม่พบการเชื่อมต่อในคอมพิวเตอร์เครื่องอื่นจะเป็นไปได้ว่าบอร์ด KidBright มีปัญหา

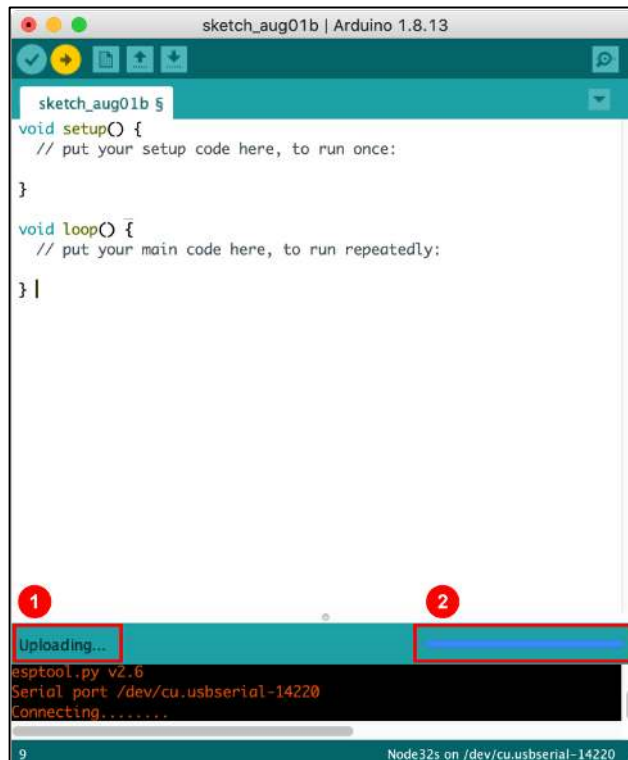
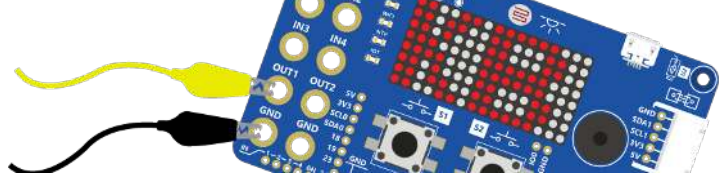


9. เมื่อเลือกการเชื่อมต่อเสร็จแล้ว ให้ใช้โค้ดที่ได้รับเมื่อเปิดโปรแกรมครั้งแรกซึ่งยังไม่ถูกแก้ไข หรือโค้ดจากเมนู File > New เพื่อทดลองอัปโหลดโปรแกรมเข้าสู่บอร์ด KidBright โดยกดที่เมนูอัปโหลดดังรูปภาพ 10-8



รูปภาพ 10-8 ภาพแสดงการอัปโหลดโปรแกรม

10. สามารถดูสถานะอัปโหลดและแถบความก้าวหน้า (Progress Bar) ได้จากแถบที่บริเวณส่วนล่างของโปรแกรกดังรูปภาพ 10-9 โดยสถานะจะมีด้วยกันหลากหลายแบบ สถานะทั่วไปที่แสดงว่าการทำงานยังคงถูกต้อง ได้แก่ Compiling Sketch, Uploading และ Done Uploading ส่วนสถานะที่แสดงว่าการทำงานไม่ถูกต้อง จะแสดงออกเป็น Error message และแถบสีฟ้าจะเปลี่ยนเป็นแถบสีแดง ซึ่งอาจจะเกิดได้จาก Code ไม่ถูกต้อง หรืออาจจะอัปโหลดไม่สำเร็จ ซึ่งจะต้องแก้ไขเป็นกรณีไป เมื่อแก้ไขแล้วจึงลองอัปโหลดใหม่อีกครั้ง
11. หากสถานะเป็น Done Upload แสดงว่าการตั้งค่าบอร์ดเสร็จสิ้นแล้ว และสามารถอัปโหลดโปรแกรมเข้าสู่บอร์ด KidBright ได้สำเร็จ

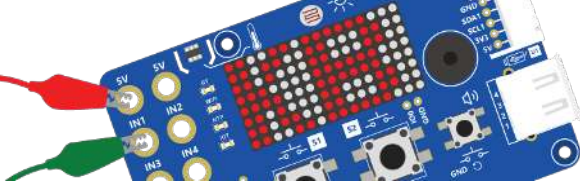


รูปภาพ 10-9 ภาพแสดงสถานะอัปโหลดและแถบความก้าวหน้า

ทั้งนี้สำหรับระบบปฏิบัติการ Linux อาจจำเป็นต้องมีการตั้งค่าเพิ่มเติมเพื่อให้โปรแกรม Arduino สามารถเชื่อมต่อกับบอร์ดผ่าน Serial Port ได้ โดยผู้พัฒนาสามารถศึกษาขั้นตอนเพิ่มเติมได้จากเว็บไซต์ <https://docs.platformio.org/en/latest/faq.html#platformio-udev-rules>

10.4 การติดตั้ง Library ใน Arduino IDE สำหรับใช้งานบอร์ด KidBright

เนื่องจากโปรแกรม Arduino IDE สามารถใช้เพื่อพัฒนาโปรแกรมสำหรับบอร์ดที่หลากหลายและถูกพัฒนาโดยนักพัฒนาจากทั่วโลก ทำให้ทีมพัฒนาโปรแกรม Arduino มีการกำหนดรูปแบบของ Library หรือชุดโค้ดสำเร็จ และอนุญาตให้นักพัฒนาจากทั่วโลกสามารถสร้างและใช้งาน Library เหล่านี้ได้ โดยมาก library เหล่านี้มักจะเป็นชุดคำสั่งสำหรับการใช้งาน Sensor หรือ Actuator หรือ การเชื่อมต่อกับ cloud platform ผ่านโปรโตคอลต่าง ๆ ซึ่งจะช่วยให้ นักพัฒนาทั่วไปสามารถเขียนโค้ดได้สั้นลง และอาจไม่จำเป็นต้องคำนึงถึงรายละเอียดเชิงลึกของอุปกรณ์หรือโปรโตคอล

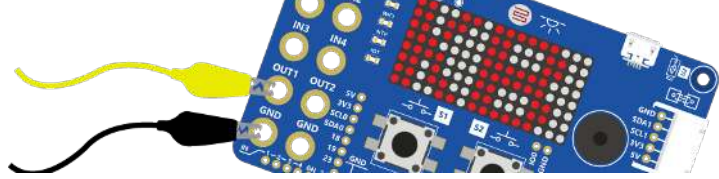


เนื่องจากทีมพัฒนา Arduino IDE อนุญาตให้นักพัฒนาทุกคนสามารถสร้าง Library ของตนเองได้ เราจึงอาจจะเจอ Library ที่ไม่มีคุณภาพหรือใช้การไม่ได้หรืออาจใช้การได้เฉพาะในบอร์ดอื่น การเลือก Library ที่มีคุณภาพจึงเป็นสิ่งจำเป็น โดย Library ที่มีคุณภาพมักจะมีการทดสอบกับอุปกรณ์ที่แตกต่างหลากหลาย มีข้อจำกัดการใช้งานน้อย และถ้ามีข้อจำกัดก็มักจะมีการระบุข้อจำกัดการใช้งานไว้ในคำอธิบายหรือเอกสารประกอบการใช้งานเสมอ โดยส่วนมาก Library ที่มีคุณภาพมักจะถูกสร้างจากบริษัทที่ผลิตอุปกรณ์เหล่านั้นเอง หรือเป็นนักพัฒนาที่ทำงานกับอุปกรณ์เหล่านี้เป็นประจำ

ในการติดตั้ง Library สำหรับใช้งาน เราสามารถเลือกติดตั้ง Library บางตัวได้ตามการใช้งานตามอุปกรณ์จริง ซึ่งไม่จำเป็นต้องติดตั้งทุกตัว สามารถดูอุปกรณ์ รวมทั้งชื่อของ Library และวิธีการติดตั้งสำหรับบอร์ด KidBright ได้จากตาราง 10-1

อุปกรณ์	ชื่อ Library	รูปแบบการติดตั้ง Library
Dot Matrix LED	Adafruit LED Backpack Library https://github.com/adafruit/Adafruit_LED_Backpack	ผ่าน Library Manager
Real Time Clock	MCP7941X https://github.com/ichilton/mcp7941x_arduino/archive/master.zip	โดย Zip File
Temperature Sensor	Wire	มีอยู่แล้ว ไม่ต้องติดตั้งเพิ่มเติม
Light Sensor	Arduino	มีอยู่แล้ว ไม่ต้องติดตั้งเพิ่มเติม
Buzzer	Arduino	มีอยู่แล้ว ไม่ต้องติดตั้งเพิ่มเติม
Button	Arduino	มีอยู่แล้ว ไม่ต้องติดตั้งเพิ่มเติม

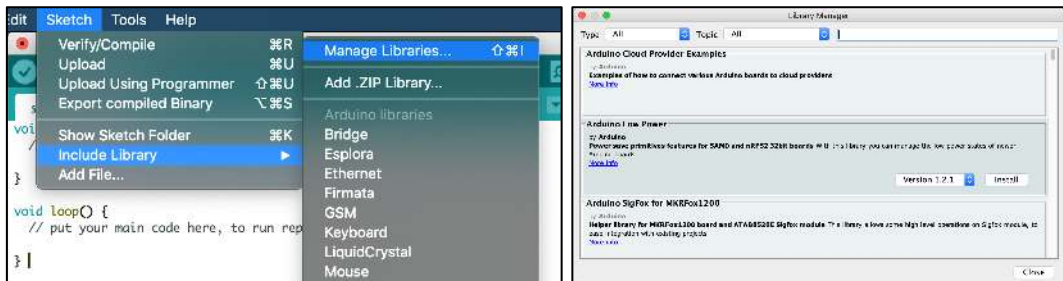
ตาราง 10-1 ตารางแสดงอุปกรณ์ ชื่อ Library และรูปแบบการติดตั้ง Library



10.4.1 การติดตั้ง Library ด้วย Library Manager ในโปรแกรม Arduino IDE

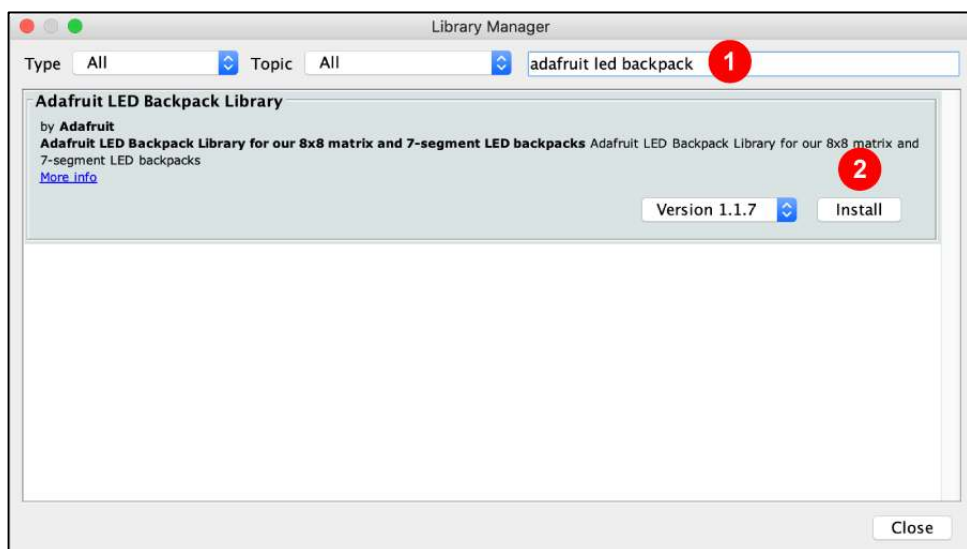
การติดตั้งสามารถทำได้โดยทำตามขั้นตอนต่อไปนี้

1. เลือกเมนู Tools > Manage Libraries จะมีหน้าต่าง Library Manager ขึ้นมาดังรูปภาพ 10-10



รูปภาพ 10-10 ภาพแสดงเมนู Manage Libraries และหน้าต่าง Library Manager

2. พิมพ์ชื่อ Library ที่ต้องการติดตั้ง และกด Install

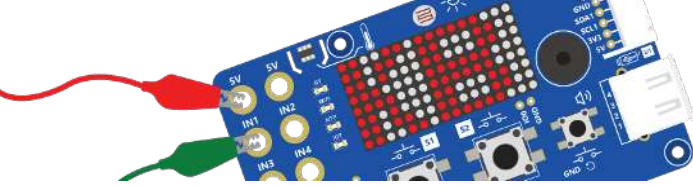


รูปภาพ 10-11 ภาพแสดงการค้นหา และปุ่ม Install เพื่อติดตั้ง library

10.4.2 การติดตั้ง Library โดย Zip File ในโปรแกรม Arduino IDE

การติดตั้งสามารถทำได้โดยทำตามขั้นตอนต่อไปนี้

1. ดาวน์โหลดไฟล์ zip ของ library
2. เลือกเมนู Sketch > Include Library > Add .zip library
3. เลือกไฟล์ zip ที่ดาวน์โหลดไว้ และกด Open



10.5 เปรียบเทียบการใช้งาน KidBright IDE และ Arduino

ในขั้นตอนการเขียนโปรแกรมด้วย KidBright IDE และ Arduino จะมีข้อแตกต่างกันอยู่มาก โดยข้อแตกต่างหลัก ได้แก่

1. ในโปรแกรม KidBright IDE จะให้ผู้ใช้เขียนโปรแกรมในแบบ Block-structured programming แต่โปรแกรม Arduino IDE จะใช้การเขียนโปรแกรมแบบ Coding ด้วยภาษา C/C++
2. โปรแกรม Arduino จะมีการตั้งค่าบอร์ด และ Connection เพื่อเลือกว่าขณะนี้กำลังพัฒนาโปรแกรมอยู่ที่บอร์ดใด ในขณะที่ KidBright IDE ไม่จำเป็นต้องตั้งค่าเพราะว่าโปรแกรมจะถูกใช้กับบอร์ด KidBright อยู่แล้ว
3. ในโปรแกรม KidBright IDE จะมี Block Library ที่จำเป็นสำหรับบอร์ด KidBright อยู่แล้ว เนื่องจาก KidBright IDE ถูกสร้างขึ้นเพื่อให้ใช้งานกับบอร์ด KidBright โดยเฉพาะจึงมี Library ที่จำเป็นถูกติดตั้งมาอยู่แล้วตั้งแต่เริ่มต้น ในขณะที่การใช้งาน Arduino IDE ผู้ใช้จะต้องมีการเพิ่ม Library เข้าไปในขั้นตอนการเขียนโปรแกรมด้วย เนื่องจากโปรแกรม Arduino สามารถใช้งานได้กับบอร์ดหลากหลายจึงไม่ได้ติดตั้ง Library สำหรับเซนเซอร์ต่าง ๆ ของบอร์ด KidBright มาตั้งแต่เริ่มต้น ทั้งนี้ แนวคิดการเพิ่ม Library จะคล้ายกับการเพิ่ม Plugin ในโปรแกรม KidBright IDE

10.6 เปรียบเทียบตัวอย่างโปรแกรมที่สร้างด้วย KidBright IDE และ Arduino IDE

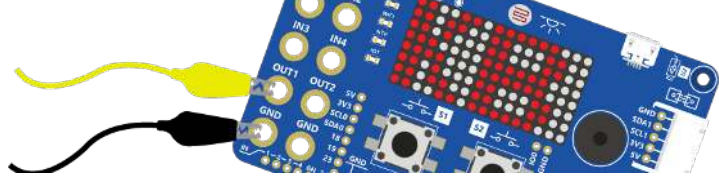
ในเนื้อหาส่วนนี้ จะเป็นการยกตัวอย่างเปรียบเทียบการเขียนโปรแกรมด้วย KidBright IDE และ Arduino IDE สำหรับการใช้งานหลากหลายแบบ

10.6.1 การแสดงข้อความผ่าน Dot Matrix LED

ใน KidBright IDE สามารถทำได้โดยต่อบล็อกดังรูปภาพ 10-12



รูปภาพ 10-12 ภาพโปรแกรมแสดงข้อความบน Dot Matrix LED ด้วย KidBright IDE



อธิบายการทำงานของบล็อกได้ดังนี้

1. Forever เป็นการบอกว่าจะทำตลอดไป
2. LED 16x8 Scroll เป็นการเลื่อนข้อความบน Dot Matrix LED
3. Delay เป็นการเว้นการทำงาน เพื่รอให้การเลื่อนข้อความเสร็จสิ้น

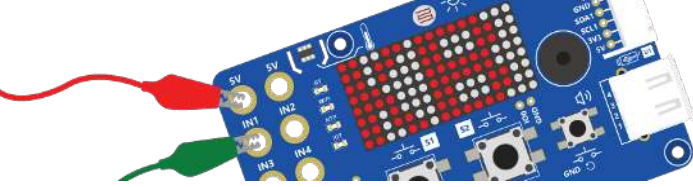
ต่อไปจะเป็นตัวอย่างการสั่งการ Dot Matrix LED ในโปรแกรม Arduino IDE

```
1 #include <Wire.h>
2 #include <Adafruit_GFX.h>
3 #include "Adafruit_LEDBackpack.h"
4
5 Adafruit_8x16minimatrix matrix;
6
7 void setup() {
8   matrix.begin(0x70);
9   matrix.setRotation(1);
10  matrix.setTextSize(1);
11  matrix.setTextColor(LED_ON);
12  matrix.setTextWrap(false);
13 }
14
15 void loop() {
16   for (int8_t x = 16; x >= -75; x--) {
17     matrix.clear();
18     matrix.setCursor(x, 0);
19     matrix.print("Hello World!");
20     matrix.writeDisplay();
21     delay(100);
22   }
23 }
```

รูปภาพ 10-13 ภาพโค้ดแสดงผลข้อความบน Dot Matrix LED ในโปรแกรม Arduino IDE

อธิบายการทำงานของโค้ดภาษา C/C++ ใน Arduino IDE ได้ดังนี้

1. บรรทัดที่ 1 – 3 เป็นการนำเข้า library ที่จำเป็น
2. บรรทัดที่ 5 เป็นการสร้าง Object ชื่อ matrix เพื่อควบคุม Dot Matrix LED
3. บรรทัดที่ 7 – 13 เป็นการตั้งค่าสำหรับการควบคุม matrix เช่น address ของ DOT Matrix LED การหมุนอักษร ขนาดอักษร และตั้งค่าการตัดข้อความ
4. บรรทัดที่ 15 คำสั่ง loop จะดำเนินการสิ่งที่อยู่ในปีกกาตลอดไป เปรียบเสมือนบล็อก Forever ใน KidBright IDE



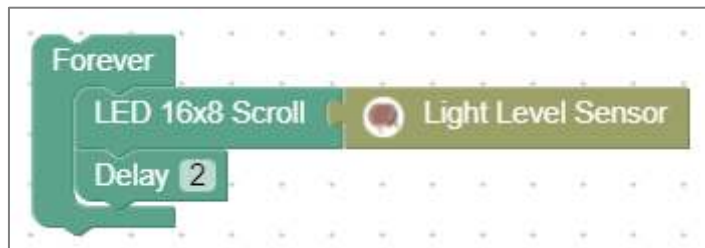
5. บรรทัดที่ 16 – 22 เป็นการสั่งการ matrix แสดงข้อความ ณ ตำแหน่งต่าง ๆ ในรูปแบบแอนิเมชัน ซึ่งนักพัฒนาสามารถปรับเปลี่ยนพารามิเตอร์ในการเคลื่อนไหวได้ ได้แก่ ความเร็วในการเคลื่อนไหว ตำแหน่งเริ่มต้นและสิ้นสุดของการ scroll

ดังที่ได้กล่าวไปในข้างต้น KidBright IDE มีการติดตั้ง Library ที่จำเป็นสำหรับการใช้งานอุปกรณ์มาอยู่แล้ว ทำให้ในขณะที่เขียนโปรแกรมผู้ใช้เพียงต่อบล็อกไม่กี่บล็อก ก็สามารถทำการควบคุม Dot Matrix LED ได้ ในขณะที่โปรแกรม Arduino IDE ผู้ใช้จะต้องทำการเขียนโปรแกรมเพื่อเลื่อนข้อความด้วยตนเองทุกขั้นตอน

การหมุนข้อความ ขนาดข้อความ การตัดคำ การปรับความเร็ว และตำแหน่งเริ่มต้น – สิ้นสุดของการเลื่อนข้อความ ไม่สามารถทำได้โดยง่ายใน KidBright IDE เพราะบล็อกคำสั่งไม่ได้อนุญาตให้ทำได้โดยตรง ในขณะที่โปรแกรม Arduino IDE อนุญาตให้นักพัฒนาสามารถควบคุมได้ทั้งหมด ความสามารถของ IDE จึงเป็นข้อคำนึงเสมอว่าจะเลือกใช้ IDE ตัวใด

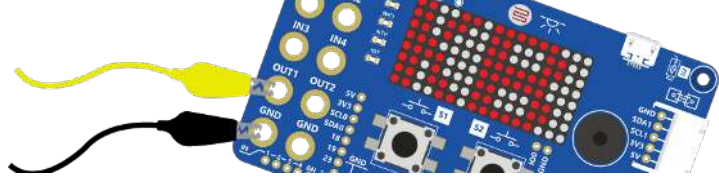
10.6.2 ทราอ่านค่า Light Sensor เพื่อนำมาแสดงบน Dot Matrix LED

ใน KidBright IDE สามารถทำได้โดยต่อบล็อกดังรูปภาพ 10-14



รูปภาพ 10-14 ภาพโปรแกรมแสดงผล Light Level บน Dot Matrix LED ในโปรแกรม KidBright IDE

หลักการของการแสดงผลจะคล้ายกับตัวอย่าง 10.6.1 เพียงแต่เปลี่ยนจากข้อความเป็นค่าของระดับแสง ในตัวอย่างต่อไปนี้จะเป็นการแสดง Light Level บน Dot Matrix LED ในโปรแกรม Arduino IDE

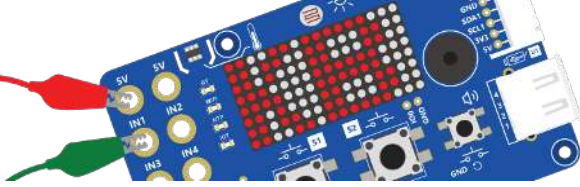


```
1 #include <Wire.h>
2 #include <Adafruit_GFX.h>
3 #include "Adafruit_LEDBackpack.h"
4
5 Adafruit_8x16minimatrix matrix;
6
7 int getLightLevel() {
8     int value = min((int) analogRead(36), 3400);
9     int light = ((3400 - value) * 100) / 3400;
10    return light;
11 }
12
13 void setup() {
14     /* setup analog range for LDR Light Sensor*/
15     analogSetWidth(12);
16     analogSetAttenuation(ADC_0db);
17
18     /* setup Dot Matrix LED */
19     matrix.begin(0x70);
20     matrix.setRotation(1);
21     matrix.setTextSize(1);
22     matrix.setTextColor(LED_ON);
23     matrix.setTextWrap(false);
24 }
25
26 void loop() {
27     int lightLevel = getLightLevel();
28     for (int8_t x = 16; x >= -16; x--) {
29         matrix.clear();
30         matrix.setCursor(x, 0);
31         matrix.print(lightLevel);
32         matrix.writeDisplay();
33         delay(100);
34     }
35 }
```

รูปภาพ 10-15 ภาพโค้ดแสดงผล Light Level บน Dot Matrix LED ในโปรแกรม Arduino IDE

อธิบายการทำงานของโค้ดภาษา C/C++ ใน Arduino IDE ได้ดังนี้

1. บรรทัดที่ 1 – 3 เป็นการนำเข้า library ที่จำเป็น
2. บรรทัดที่ 5 เป็นการสร้าง Object ชื่อ matrix เพื่อควบคุม Dot Matrix LED

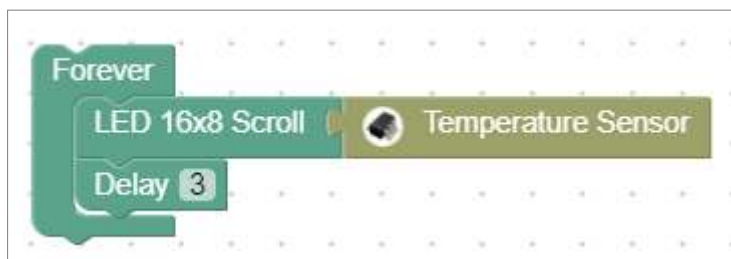


3. บรรทัดที่ 7 – 11 เป็นการสร้าง Function เพื่ออ่านค่า Analog Pin หมายเลข 36 ของไมโครคอนโทรลเลอร์ ESP32 และคำนวณเป็นระดับความเข้มแสง
4. บรรทัดที่ 13 เป็นการสร้าง Function setup เพื่อตั้งค่าอุปกรณ์ ก่อนการใช้งาน
5. บรรทัดที่ 14 – 16 เป็นการตั้งค่าโหมดของการอ่านค่า Analog Sensor ซึ่งในที่นี้คือการตั้งค่าสำหรับอ่านค่า LDR Light Sensor
6. บรรทัดที่ 18 – 23 เป็นการตั้งค่าสำหรับการควบคุม matrix
7. บรรทัดที่ 26 คำสั่ง loop จะดำเนินการสิ่งที่อยู่ในปีกกาตลอดไป
8. บรรทัดที่ 27 เป็นการเรียกใช้ Function getLightLevel เพื่อคำนวณระดับความเข้มแสง และเก็บผลลัพธ์การคำนวณไว้ในตัวแปรชื่อ lightLevel
9. บรรทัดที่ 28 – 33 เป็นการสั่งการ matrix ให้แสดงค่าระดับความเข้มแสงในรูปแบบแอนิเมชัน

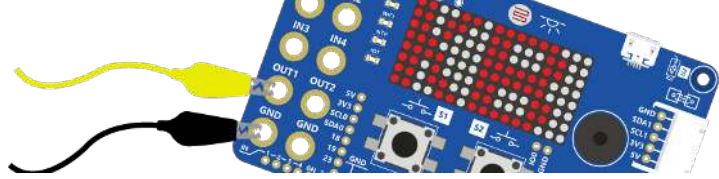
สังเกตว่าการอ่านค่า LDR Light Sensor สำหรับบอร์ด KidBright จะมีรูปแบบเฉพาะตัว ซึ่งอาจจะแตกต่างกับการอ่านค่าแสงจาก LDR Sensor ในบอร์ดอื่น ๆ ทั้งนี้เป็นเพราะเรากำหนดโหมดของ Analog Sensor ตามโหมดที่ KidBright IDE ใช้เป็นค่าเริ่มต้น และการคำนวณค่า Light Level ก็มีการใช้ค่าคงที่ 3400 ซึ่งก็เป็น calibration factor ที่ใช้ใน KidBright IDE อีกด้วย

10.6.3 การอ่านค่า Temperature Sensor เพื่อนำมาแสดงบน Dot Matrix LED

คำสั่งที่ใช้แสดงผลบน Dot Matrix LED จะมีลักษณะเหมือนกับตัวอย่างก่อนหน้านี้ทั้งใน 10.6.1 และ 10.6.2 เพียงแต่เปลี่ยนแปลงตัวแปรที่จะใช้แสดงผลว่าจะเป็นข้อความ หรือ ค่าระดับแสง หรือค่าอุณหภูมิ หากเราต้องการแสดงผลค่าอุณหภูมิให้เราเลือกใช้บล็อก Temperature Sensor ใน KidBright IDE สามารถทำได้โดยต่อบล็อกดังรูปภาพ 10-16



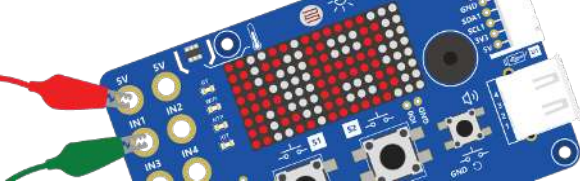
รูปภาพ 10-16 ภาพโปรแกรมแสดงผลอุณหภูมิบน Dot Matrix LED ในโปรแกรม KidBright IDE



ใน Arduino IDE สามารถทำได้โดยโค้ดต่อไปนี้

```
1 #include <Wire.h>
2 #include <Adafruit_GFX.h>
3 #include "Adafruit_LEDBackpack.h"
4
5 Adafruit_8x16minimatrix matrix;
6
7 double getTemperature() {
8   Wire1.beginTransmission(0x4D);
9   Wire1.write(0x00); // Temperature Data Register
10  Wire1.endTransmission();
11
12  uint8_t count = Wire1.requestFrom(0x4D, 2);
13  double temp = 0.0;
14  if (count == 2) {
15    byte data[2];
16    data[0] = Wire1.read();
17    data[1] = Wire1.read();
18    temp = 0.25 * (data[0] << 3 | (data[1] >> 5 & 0x07));
19  }
20  return temp;
21 }
22
23 void setup() {
24   /* setup I2C for Temperature Sensor*/
25   Wire1.begin(4, 5);
26
27   /* setup Dot Matrix LED */
28   matrix.begin(0x70);
29   matrix.setRotation(1);
30   matrix.setTextSize(1);
31   matrix.setTextColor(LED_ON);
32   matrix.setTextWrap(false);
33 }
34
35 void loop() {
36   double celcius = getTemperature();
37   for (int8_t x = 16; x >= -16; x--) {
38     matrix.clear();
39     matrix.setCursor(x, 0);
40     matrix.print(celcius);
41     matrix.writeDisplay();
42     delay(100);
43   }
44 }
```

รูปภาพ 10-17 ภาพโค้ดแสดงผลอุณหภูมิบน Dot Matrix LED ในโปรแกรม Arduino IDE

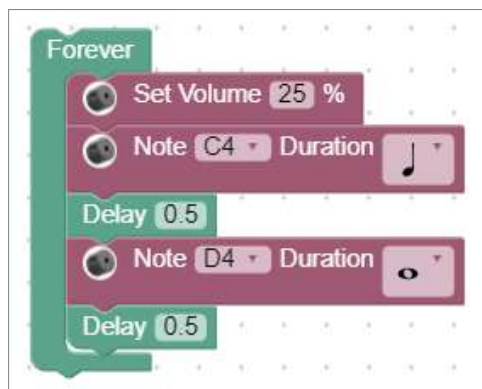


อธิบายการทำงานของโค้ดภาษา C/C++ ใน Arduino IDE ได้ดังนี้

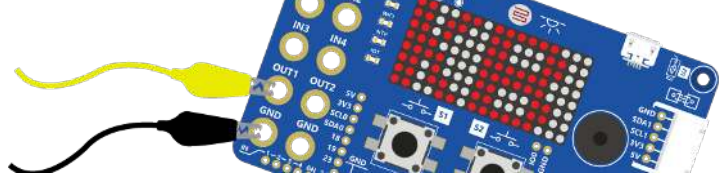
1. บรรทัดที่ 1 – 3 เป็นการนำเข้า library ที่จำเป็น
2. บรรทัดที่ 5 เป็นการสร้าง Object ชื่อ matrix เพื่อควบคุม Dot Matrix LED
3. บรรทัดที่ 7 – 21 เป็นการสร้าง Function เพื่ออ่านข้อมูลผ่านโปรโตคอล I2C ที่ Address 0x4D แล้วทำการคำนวณเป็นอุณหภูมิในหน่วย celcius
4. บรรทัดที่ 23 เป็นการสร้าง Function setup เพื่อตั้งค่าอุปกรณ์ ก่อนการใช้งาน
5. บรรทัดที่ 24 – 25 เป็นการตั้งค่าว่าจะใช้ I2C บน Pin ที่ 4 และ 5 ซึ่งมีการเชื่อมต่อกับอุปกรณ์ Temperature Sensor LM73
6. บรรทัดที่ 27 – 32 เป็นการตั้งค่าสำหรับการควบคุม matrix
7. บรรทัดที่ 35 คำสั่ง loop จะดำเนินการสิ่งที่อยู่ในปีกกาตลอดไป
8. บรรทัดที่ 36 เป็นการเรียกใช้ Function getTemperature เพื่อคำนวณอุณหภูมิและเก็บผลลัพธ์การคำนวณใส่ในตัวแปรชื่อ celcius
9. บรรทัดที่ 37 – 43 เป็นการสั่งการ matrix ให้แสดงค่าอุณหภูมิในรูปแบบแอนิเมชัน
สังเกตว่า Wire1 เป็น Object ที่ไม่ถูกสร้างจากโค้ดข้างต้น เหตุผลเพราะ Wire1 นั้นถูกสร้างอยู่ใน Wire.h อยู่แล้ว จึงสามารถนำมาใช้ได้เลยโดยไม่ต้องสร้างใหม่

10.6.4 การควบคุม Buzzer เพื่อเล่นเสียงโน้ตดนตรี

การเขียนโปรแกรมควบคุมการเล่นเสียงดนตรีในโปรแกรม KidBright IDE สามารถทำได้โดยต่อบล็อกต่าง ๆ เข้าด้วยกันดังรูปภาพ 10-18



รูปภาพ 10-18 ภาพโปรแกรมสำหรับควบคุมการเล่นเสียงดนตรีใน KidBright IDE



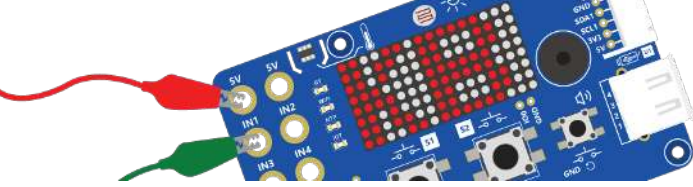
อธิบายการทำงานของบล็อกต่าง ๆ ได้ดังนี้

1. Set Volume เป็นการตั้งค่าความดังของเสียงเป็น 25%
2. Note / Duration เป็นการเปิดเล่นเสียงตัวโน้ต ซึ่งจะเปิดเสียงไว้นานหรือเร็วก็ตามแต่ชนิดของตัวโน้ตที่กำหนด และเมื่อครบกำหนดเวลาแล้วเสียงจะดับ
3. Delay เป็นการเว้นเสียงระหว่างตัวโน้ต เป็นเวลา 0.5 วินาที โดยในขณะเวลาที่เว้นจะไม่มีเสียง buzzer ดังขึ้น

ส่วนการเขียนโปรแกรมควบคุมการเล่นเสียงดนตรีในโปรแกรม Arduino IDE สามารถทำได้ดังโค้ดตัวอย่างต่อไปนี้

```
1 const int ch = 0;
2 const int bpm = 120;
3 const int quarter_note_delay = (60 * 1000) / bpm;
4
5 void setup() {
6   ledcSetup(ch, 0, 8);
7   ledcAttachPin(13, ch);
8 }
9
10 void loop() {
11   ledcWrite(ch, 25);
12
13   ledcWriteNote(ch, NOTE_C, 4);
14   delay(quarter_note_delay);
15   ledcWriteTone(ch, 0);
16
17   delay(500);
18
19   ledcWriteNote(ch, NOTE_D, 4);
20   delay(4 * quarter_note_delay);
21   ledcWriteTone(ch, 0);
22
23   delay(500);
24 }
```

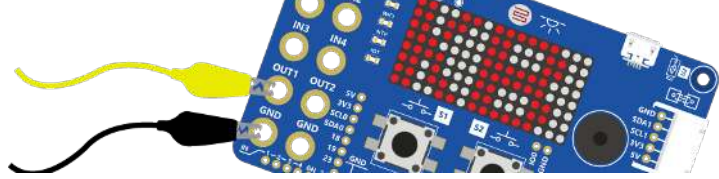
รูปภาพ 10-19 ภาพโค้ดสำหรับควบคุมการเล่นเสียงดนตรีใน Arduino IDE



อธิบายการทำงานของโค้ดภาษา C/C++ ใน Arduino IDE ได้ดังนี้

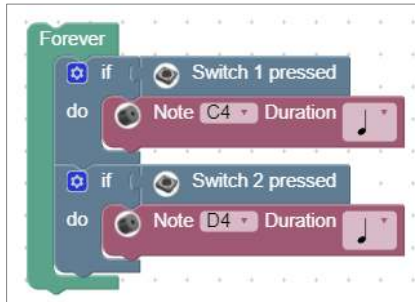
1. บรรทัดที่ 1 – 3 เป็นการประกาศตัวแปรต่าง ๆ ที่ใช้ในโปรแกรม ได้แก่ ค่า ch ซึ่งเป็นตัวเลขแทน channel ซึ่งจะใช้ในการระบุ Group และ Timer ว่าจะใช้ตัวใด ค่า bpm หรือ beat per minute ในทางดนตรี และค่า quarter_note_delay ซึ่งแทนความนานของโน้ตตัวดำในหน่วยมิลลิวินาที
2. บรรทัดที่ 6 – 7 เป็นการตั้งค่า Timer และกำหนดการใช้งานเข้ากับ Pin โดยคำสั่ง ledcSetup เป็นการกำหนดให้ Timer ที่แทนด้วยเลข ch มีความถี่เป็น 0 และใช้ resolution เป็น 8 ส่วนคำสั่ง ledcAttachPin เป็นคำสั่งที่จะแนบ Timer เข้ากับ Pin หมายเลข 13 ที่ต่ออยู่กับ Buzzer ของบอร์ด KidBright
3. บรรทัดที่ 10 คำสั่ง loop จะดำเนินการสิ่งที่อยู่ในปีกกาตลอดไป
4. บรรทัดที่ 11 เป็นการกำหนด duty cycle ของ Timer หรือก็คือการตั้งค่าความดังของ Buzzer นั้นเอง ซึ่งในตัวอย่างตั้งไว้เป็น 25%
5. บรรทัดที่ 13 และ 19 เป็นการตั้งค่า frequency ให้กับ Timer เป็นความถี่ของตัวโน้ตดนตรีในอีกท่วงต่าง ๆ เพื่อให้ buzzer เล่นเสียง ทั้งนี้ ตัวแทนตัวโน้ตที่สามารถถูกใช้ในโค้ดได้ มีดังนี้ NOTE_C, NOTE-Cs, NOTE_D, NOTE_Eb, NOTE_E, NOTE_F, NOTE_Fs, NOTE_G, NOTE_Gs, NOTE_A, NOTE_Bb, NOTE_B และ NOTE_MAX
6. บรรทัดที่ 14 และ 20 เป็นคำสั่งให้บอร์ด KidBright ไม่ทำอะไรเพิ่มเติมเป็นเวลานานเท่ากับชนิดของตัวโน้ตนั้น ๆ
7. บรรทัดที่ 15 และ 21 เป็นคำสั่งให้ดับเสียงของ Buzzer โดยการตั้งค่าความถี่ให้เป็น 0 Hertz
8. บรรทัดที่ 17 และ 23 เป็นคำสั่งให้บอร์ด KidBright ไม่ทำอะไรเพิ่มเติม เป็นเวลา 500 มิลลิวินาที หรือ 0.5 วินาที

สังเกตว่าการเล่นเสียงดนตรีทั้งในโปรแกรม KidBright IDE และ Arduino IDE มีชุดคำสั่งสำเร็จรูปที่ทำให้ผู้ใช้โปรแกรมสามารถเลือกเล่นตัวโน้ต เลือกอ็อกเทฟ เลือกความนานของตัวโน้ต และระดับความดังของเสียงตามความต้องการได้ทั้งคู่



10.6.5 การใช้งานปุ่มกดบนบอร์ด KidBright เพื่อเล่นเสียงดนตรี

การใช้ KidBright IDE เพื่อตรวจจับการกดปุ่มบนบอร์ด KidBright สามารถทำได้โดยเติมเงื่อนไขเพื่อตรวจสอบสถานะปุ่มกด เช่น หากมีการกดอยู่ให้ทำการเล่นเสียง เช่นตัวอย่างนี้



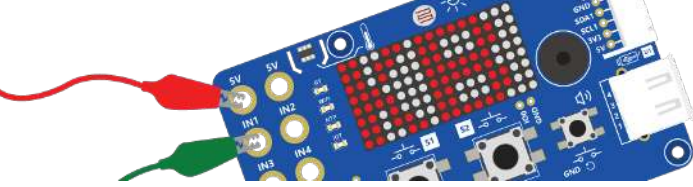
รูปภาพ 10-20 ภาพโปรแกรมสำหรับใช้งานปุ่มกดบนบอร์ด KidBright เพื่อเล่นเสียงดนตรี ใน KidBright IDE

คำอธิบายการทำงานของโปรแกรม มีดังนี้

1. if Switch is pressed เป็นการตรวจสอบเงื่อนไข ว่าปุ่มถูกกดอยู่จริงหรือไม่
 2. do จะยอมให้บล็อกภายในทำงานได้ หากปุ่มถูกกดอยู่จริง
 3. Note / Duration เป็นคำสั่งเล่นเสียงดนตรี เช่นเดียวกับในตัวอย่างก่อนหน้านี้
- ในการทำงานเดียวกัน เราสามารถใช้ Arduino IDE ทำได้โดยใช้โค้ดต่อไปนี้

```
1  const int ch = 0;
2  const int bpm = 120;
3  const int quarter_note_delay = (60 * 1000) / bpm;
4
5  const uint8_t SW1 = 16;
6  const uint8_t SW2 = 14;
7
8  void setup() {
9    ledcSetup(ch, 0, 8);
10   ledcAttachPin(13, ch);
11
12   pinMode(SW1, INPUT_PULLUP);
13   pinMode(SW2, INPUT_PULLUP);
14 }
15
16 void loop() {
17   if (digitalRead(SW1) == LOW) {
18     ledcWriteNote(ch, NOTE_C, 4);
19     delay(quarter_note_delay);
20     ledcWriteTone(ch, 0);
21   }
22   if (digitalRead(SW2) == LOW) {
23     ledcWriteNote(ch, NOTE_D, 4);
24     delay(quarter_note_delay);
25     ledcWriteTone(ch, 0);
26   }
27 }
```

รูปภาพ 10-21 ภาพโปรแกรมสำหรับใช้งานปุ่มกดบนบอร์ด KidBright เพื่อเล่นเสียงดนตรี ใน KidBright IDE



อธิบายการทำงานของโค้ดภาษา C/C++ ใน Arduino IDE ได้ดังนี้

1. บรรทัดที่ 1 – 3 เป็นการประกาศตัวแปรต่าง ๆ ที่ใช้ในโปรแกรม ซึ่งเกี่ยวข้องกับการควบคุมเสียง
2. บรรทัดที่ 5 – 6 เป็นการประกาศตัวแปรหมายเลข Pin ของปุ่มกด ได้แก่ ปุ่ม S1 เป็นเลข 16 และ ปุ่ม S2 เป็นเลข 14
3. บรรทัดที่ 9 – 10 เป็นการตั้งค่าที่เกี่ยวข้องกับการควบคุมเสียง
4. บรรทัดที่ 12 – 13 เป็นการตั้งค่าโหมดการทำงานของ Pin ซึ่งเกี่ยวข้องกับปุ่มกด
5. บรรทัดที่ 16 คำสั่ง loop จะดำเนินการสิ่งที่อยู่ในปีกกาตลอดไป
6. บรรทัดที่ 17 และ 22 เป็นตรวจสอบว่าปุ่มถูกกดอยู่หรือไม่
7. บรรทัดที่ 18 – 20 และ 23 – 25 เป็นการเล่นเสียงโน้ตดนตรี เช่นเดียวกับตัวอย่างก่อนหน้า

