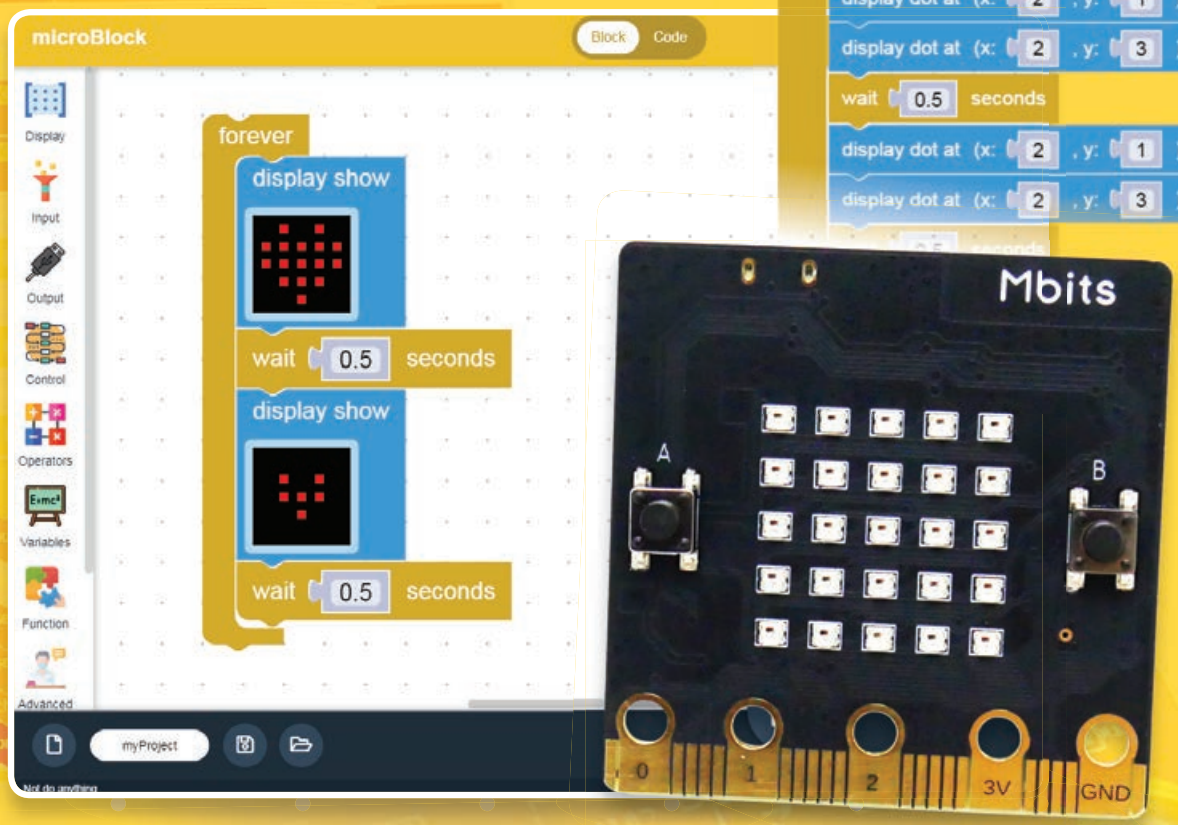
แรกเริ่มเขียนโค้ดกับ

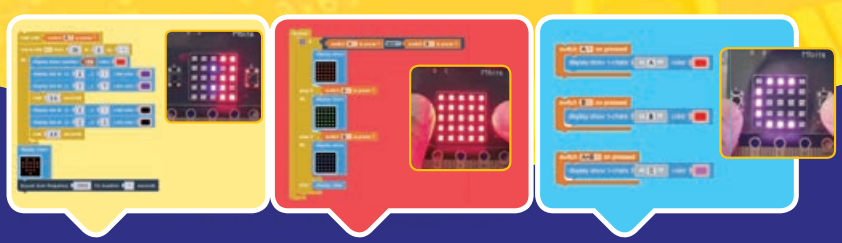
Mbits

บอร์ดไมโครคอนโทรลเลอร์เพื่อการเรียนรู้



พร้อมตัวอย่าง
โปรแกรมกว่า

20 แบบ



กฤษดา ใจเย็น
ธนโชติ ศรีภักดี

inex
Innovative Experiment
www.inex.co.th

แรกเริ่มเขียนโค้ดกับ Mbits

บอร์ดไมโครคอนโทรลเลอร์เพื่อการเรียนรู้

กฤษดา ใจเย็น
ธนโชติ ศรีภักดี

สงวนลิขสิทธิ์ตาม พ.ร.บ. ลิขสิทธิ์ พ.ศ. 2537
ห้ามการลอกเลียนไม่ว่าส่วนหนึ่งส่วนใดของหนังสือเล่มนี้ นอกจากจะได้รับอนุญาต

ใครควรใช้หนังสือเล่มนี้

1. นักเรียน นิสิต นักศึกษา และบุคคลทั่วไปที่มีความสนใจในการเรียนรู้วิทยาการคำนวณในเชิงการใช้งานร่วมกับอุปกรณ์ระบบสมองกลฝังตัว รวมถึงสาระการเรียนรู้ STEM ศึกษาและการโค้ดดิ้ง
2. สถาบันการศึกษาหรือโรงเรียนที่เปิดการเรียนการสอนที่ครอบคลุมสาระการเรียนรู้ด้าน STEM ศึกษาและวิทยาการคำนวณ
3. วิทยาลัยและมหาวิทยาลัย ที่มีการเปิดการเรียนการสอนวิชาอิเล็กทรอนิกส์หรือภาควิชาวิศวกรรมอิเล็กทรอนิกส์และคอมพิวเตอร์
4. คณาจารย์ที่มีความต้องการศึกษา และเตรียมการเรียนการสอนวิชาไมโครคอนโทรลเลอร์ รวมถึงวิทยาศาสตร์ประยุกต์ที่ต้องการบูรณาการความรู้ทางอิเล็กทรอนิกส์-ไมโครคอนโทรลเลอร์-การเขียนโปรแกรมคอมพิวเตอร์-การทดลองทางวิทยาศาสตร์ STEM ศึกษาและวิทยาการคำนวณตั้งแต่ระดับประถมศึกษา

ดำเนินการจัดพิมพ์และจำหน่ายโดย

บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด
108 ซ.สุขุมวิท 101/2 ถ.สุขุมวิท แขวงบางนา เขตบางนา กรุงเทพฯ 10260
โทรศัพท์ 0-2747-7001-4
โทรสาร 0-2747-7005

รายละเอียดที่ปรากฏในหนังสือเล่มนี้ได้ผ่านการตรวจทานอย่างละเอียดและถ้วนถี่ เพื่อให้มีความสมบูรณ์และถูกต้องมากที่สุดภายใต้เงื่อนไขและเวลาที่พึงมีก่อนการจัดพิมพ์เผยแพร่ ความเสียหายอันอาจเกิดจากการนำข้อมูลในหนังสือเล่มนี้ไปใช้ ทางบริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด มิได้มีภาระในการรับผิดชอบแต่ประการใด ความผิดพลาดคลาดเคลื่อนที่อาจมีและได้รับการจัดพิมพ์เผยแพร่ออกไปนั้น ทางบริษัทฯ จะพยายามชี้แจงและแก้ไขในการจัดพิมพ์ครั้งต่อไป

สารบัญ

บทที่ 1 แนะนำ Mbits V2 บอร์ดไมโครคอนโทรลเลอร์ เพื่อการเรียนรู้วิทยาการคำนวณเชิงปฏิบัติการและงาน IoT **5**

1.1 บอร์ดนี้มีที่มา 5

1.2 คุณสมบัติทางเทคนิค 5

บทที่ 2 ติดตั้งและใช้งานmicroBLOCK IDE **11**

2.1 คุณสมบัติเด่นของ microBlock IDE 11

2.2 ใช้งาน microBlock IDE กับ Mbits 13

บทที่ 3 ใช้งานอุปกรณ์อินพุต/เอาต์พุต **19**

3.1 Mbits กับ สวิตช์ A และ สวิตช์ B 20

3.2 Mbits ขับเสียงออกลำโพง 24

บทที่ 4 ไมโครโฟนและตัวตรวจวัดความเร่ง **29**

4.1 ไมโครโฟน 29

4.2 ตัวตรวจวัดความเร่ง Accelerometer 31

บทที่ 5 การใช้งาน Dashboard ของ microBlock IDE **37**

5.1 รู้จักกับหน้า Dashboard 37

5.2 รู้จักกับวิดเจ็ตต่างๆ ของ Dashboard ในโปรแกรม microBlock IDE 38

5.3 การใช้งาน Dashboard เบื้องต้น 40

บทที่ 6 แนะนำ AX-Mbits บอร์ดเชื่อมต่อพอร์ตอินพุตเอาต์พุตกับอุปกรณ์ภายนอกสำหรับ Mbits V2 บอร์ดไมโครคอนโทรลเลอร์เพื่อการเรียนรู้วิทยาการคำนวณเชิงปฏิบัติการ **47**

คำชี้แจงจากคณะผู้เขียน / เรียบเรียง

การนำเสนอข้อมูลเกี่ยวกับข้อมูลทางเทคนิคและเทคโนโลยีในหนังสือเล่มนี้ เกิดจากความต้องการที่จะอธิบายกระบวนการและหลักการทำงาน ของอุปกรณ์ในภาพรวมด้วยถ้อยคำที่ง่ายเพื่อสร้างความเข้าใจแก่ผู้อ่าน ดังนั้นการแปลคำศัพท์ทางเทคนิคหลายๆ คำอาจไม่ตรงตามข้อบัญญัติของราชบัณฑิตยสถาน และมีหลายๆ คำที่ยังไม่มีการบัญญัติอย่างเป็นทางการ คณะผู้เขียนจึงขออนุญาตบัญญัติศัพท์ขึ้นมาใช้ในการอธิบาย โดยมีข้อจำกัดเพื่ออ้างอิงในหนังสือเล่มนี้เท่านั้น

ทั้งนี้สาเหตุหลักของข้อชี้แจงนี้มาจาก การรวบรวมข้อมูลของอุปกรณ์ในระบบสมองกลฝังตัวและเทคโนโลยีหุ่นยนต์สำหรับการศึกษาเพื่อนำมาเรียบเรียงเป็นภาษาไทยนั้นทำได้ไม่ง่ายนัก ทางคณะผู้เขียนต้องทำการรวบรวมและทดลองเพื่อให้แน่ใจว่า ความเข้าใจในกระบวนการทำงานต่างๆ นั้นมีความคลาดเคลื่อนน้อยที่สุด

เมื่อต้องทำการเรียบเรียงออกมาเป็นภาษาไทย ศัพท์ทางเทคนิคหลายคำมีความหมายที่ทับซ้อนกันมาก การบัญญัติศัพท์จึงเกิดจากการปฏิบัติจริงร่วมกับความหมายทางภาษาศาสตร์ ดังนั้นหากมีความคลาดเคลื่อนหรือผิดพลาดเกิดขึ้น ทางคณะผู้เขียนขอน้อมรับและหากได้รับคำอธิบายหรือชี้แนะจากท่านผู้รู้จะได้ทำการชี้แจงและปรับปรุงข้อผิดพลาดที่อาจมีเหล่านั้นโดยเร็วที่สุด

ทั้งนี้เพื่อให้การพัฒนาสื่อทางวิชาการ โดยเฉพาะอย่างยิ่งกับความรู้ของเทคโนโลยีสมัยใหม่สามารถดำเนินไปได้อย่างต่อเนื่อง ภายใต้การมีส่วนร่วมของผู้รู้ในทุกภาคส่วน

ในหนังสือเล่มนี้ได้ทำการอ้างอิงถึงซอฟต์แวร์และเว็บไซต์หลายแห่ง ทางผู้จัดทำไม่อาจรับประกันหรือรับรองการคงอยู่ของสินค้าและบริการใดๆ ที่นำเสนอหรืออ้างอิงในหนังสือเล่มนี้ อย่างไรก็ตาม ในขณะจัดทำหนังสือเล่มนี้ ทางผู้จัดทำได้ทำการทดลองและทดสอบภายใต้ กรอบเวลาและการปรับปรุงล่าสุดในเวลานั้นๆ หากมีการปรับปรุงอื่นใดที่ทำให้การนำเสนอคลาดเคลื่อนหรือใช้ประโยชน์ไม่ได้ ทางผู้จัดทำจะพยายามปรับปรุงเนื้อหาและเผยแพร่ผ่านทางเว็บไซต์และสื่อสาธารณะของบริษัท อินโนเวตีฟ เอ็กเพอรนิเมนต์ จำกัด ที่ *www.inex.co.th* และ *www.facebook.com/innovativeexperiment*

บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด

บทที่ 1 แนะนำ Mbits V2 บอร์ดไมโครคอนโทรลเลอร์เพื่อการเรียนรู้วิทยาการคำนวณเชิงปฏิบัติการและงาน IoT

1.1 บอร์ดนี้มีที่มา

Mbits V2 เป็นผลงานความร่วมมือของ 3 บริษัท 2 ประเทศ ประกอบด้วย Shenzhen Elecrow Limited สาธารณรัฐประชาชนจีน (www.elecrow.com), บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด หรือ INEX (www.inex.co.th) และ Artronshop (www.artronshop.co.th) จากประเทศไทย โดย Elecrow ร่วมกับ INEX ผลิตบอร์ด Mbits V2 ภายใต้ข้อกำหนดทางฮาร์ดแวร์ที่ทำให้ใช้งานไมโครคอนโทรลเลอร์หลักบนบอร์ดนั่นคือ ESP32 ให้มีเสถียรภาพและเกิดประโยชน์สูงสุดเพื่อสนับสนุนการเรียนรู้ด้านสะเต็มศึกษาและวิทยาการคำนวณสำหรับประเทศไทยเป็นสำคัญ

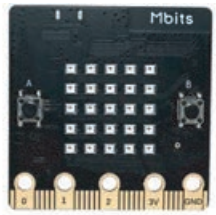
โดยเครื่องมือพัฒนาโปรแกรมสำหรับบอร์ด Mbits V2 นี้จะใช้โปรแกรมภาษาไพทอนแบบย่อส่วนหรือไมโครไพทอนเป็นหลัก ซึ่งสอดคล้องกับหลักสูตรวิทยาการคำนวณที่จัดทำโดยสถาบันส่งเสริมการสอนวิทยาศาสตร์และเทคโนโลยี หรือ สสวท. และเครื่องมือนี้คือ microBlock IDE ที่พัฒนาโดยคุณสนธยา นงนุช จาก Artronshop โดยรองรับการพัฒนาโค้ดในแบบบล็อก (Block code programming) สำหรับผู้เริ่มต้นในระดับประถมวัย และแบบโค้ดไมโครไพทอน (microPython text-based programming) ทำให้ครอบคลุมการเรียนรู้ในทุกระดับของเยาวชนและผู้เริ่มต้นในวงกว้าง

นอกจากนั้น ทาง Elecrow ก็ให้ความสนใจที่จะนำ microBlock IDE ไปใช้กับบอร์ดไมโครคอนโทรลเลอร์ ESP32 ตัวอื่นของบริษัท อันนำไปสู่โอกาสที่จะเผยแพร่ผลงานของโปรแกรมเมอร์คนไทยที่มีความสามารถทัดเทียมในระดับนานาชาติให้เป็นที่ยอมรับได้

1.2 คุณสมบัติทางเทคนิค

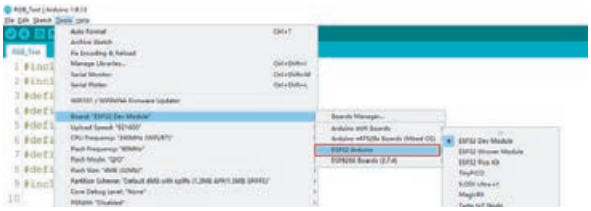
- ขนาดเล็ก 52 × 52 มม.
- ใช้ไมโครคอนโทรลเลอร์หลักเป็นโมดูล ESP32 Wrover-B มีหน่วยความจำแฟลชแบบ SPI 4MB และหน่วยความจำ PSRAM 8MB จึงรองรับการพัฒนาโปรแกรมด้วยภาษาไพทอนแบบย่อสวนได้อย่างมีประสิทธิภาพ
- เชื่อมต่อไร้สายความถี่ 2.4GHz ทั้ง WiFi และ Bluetooth แบบ BLE 4.2
- ส่วนแสดงผลเป็น LED 3 สี RGB แบบโปรแกรมได้ จำนวน 25 ดวง จัดเรียงแบบ 5x5 จุด
- มีลำโพงเปียโซสำหรับขับเสียงสัญญาณต่างๆ และเสียงดนตรี
- มีวงจรตรวจจับเสียงด้วยไมโครโฟนแบบ MEMS
- ติดตั้งตัวตรวจจับวัดความเร่ง 3 แกน ใช้เพื่อตรวจจับการเคลื่อนไหวและการเอียงได้
- มีสวิตช์กดติดปล่อยดับพร้อมใช้งาน 2 ตัว ระบุเป็น สวิตช์ A และ B
- ขั้วต่อพอร์ตอินพุตเอาต์พุตเป็นแบบ Edge Connector รูปแบบเดียวกับ micro:bit จึงนำไปใช้งานร่วมกับบอร์ดเสริมต่างๆ ของ micro:bit ได้ โดยต้องตรวจสอบข้อจำกัดทางเทคนิคร่วมด้วย เพราะความสามารถของขาพอร์ตอินพุตเอาต์พุตแต่ละขาของ ESP32 รองรับการทำงานได้หลากหลาย
- มีขาเชื่อมต่อระบบบัส I^2C เพื่อขยายการเชื่อมต่อกับอุปกรณ์ต่อพ่วงภายนอกได้
- มีสวิตช์ RESET และมี LED แสดงสถานะการทำงานของไมโครคอนโทรลเลอร์ เพื่อตรวจสอบการทำงานของบอร์ด
- เชื่อมต่อกับคอมพิวเตอร์ผ่านพอร์ต USB
- รองรับการพัฒนาโปรแกรมด้วย microBLOCK IDE , Let's Code และ Arduino IDE ซึ่งเลือกใช้ ESP-IDF เป็นคอมไพเลอร์หลัก

ร่วมมือออกแบบและผลิต
บอร์ดไมโครคอนโทรลเลอร์ Mbits V2



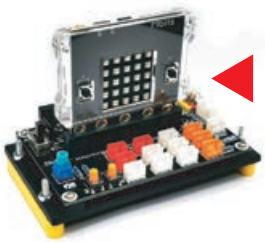
พัฒนาโปรแกรม microBlock IDE
https://microblock.app

Mbits V2 ยังรองรับการพัฒนาโปรแกรมด้วยภาษา C/C++ โดยใช้ Arduino IDE ร่วมกับชุดเครื่องมือทางซอฟต์แวร์ ESP-IDF โดย Espressif Systems



รองรับการพัฒนาโปรแกรมด้วยบล็อก

รองรับการพัฒนาโปรแกรมภาษาไพทอนแบบย่อส่วนหรือไมโครไพทอน (microPython)



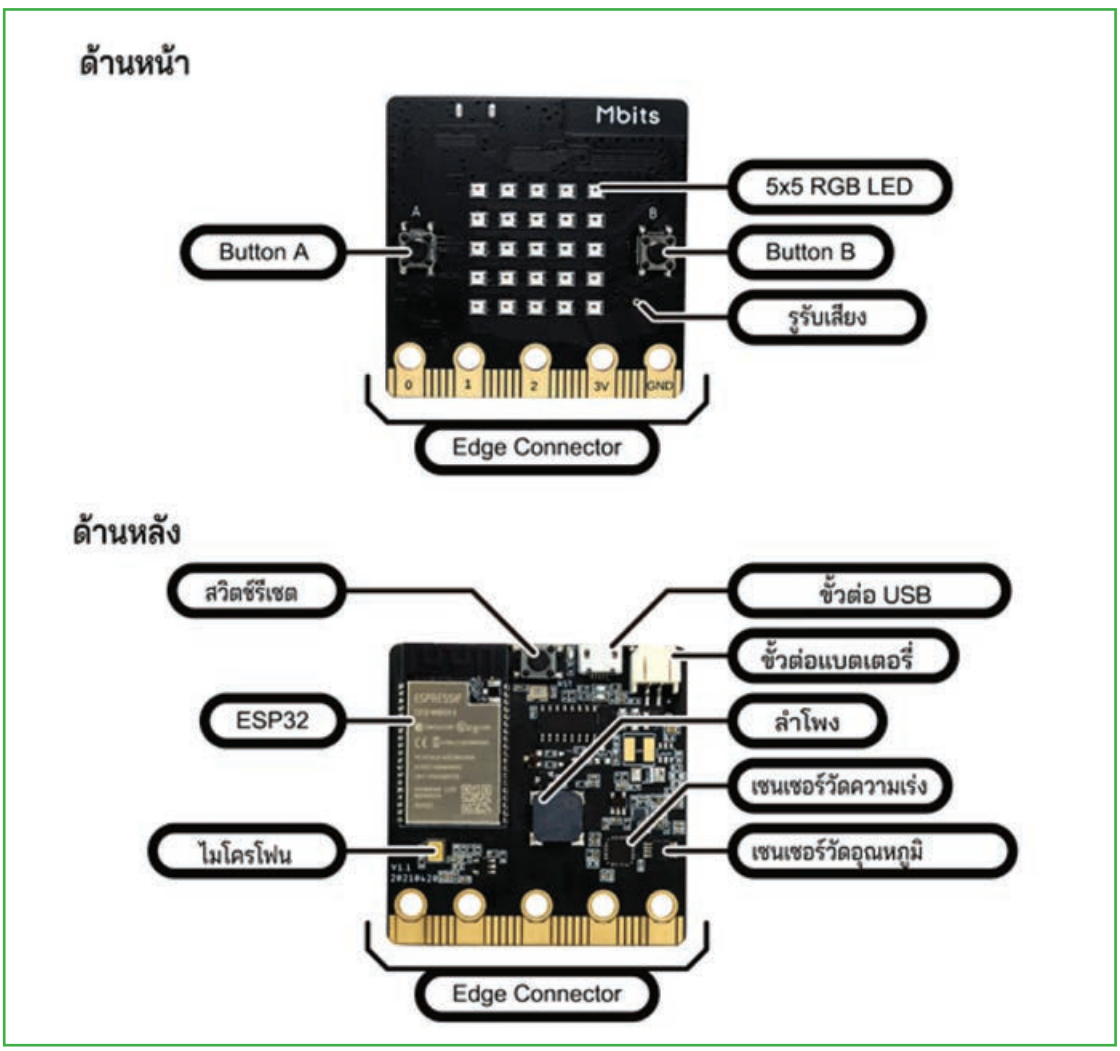
ติดตั้งกับบอร์ด AX-Mbits เพื่อใช้งานพอร์ตอินพุตเอาต์พุตของ Mbits V2 ได้สะดวก รองรับการเชื่อมต่อมินิบอร์ดตัวตรวจจับ อุปกรณ์อินพุตเอาต์พุต เซอร์โวมอเตอร์ (สูงสุด 4 ตัว)

รองรับการติดต่อกับคลาวด์เซิร์ฟเวอร์ NETPIE, Blynk, Senses IoT เพื่อพัฒนาเป็นอุปกรณ์ IoT ได้

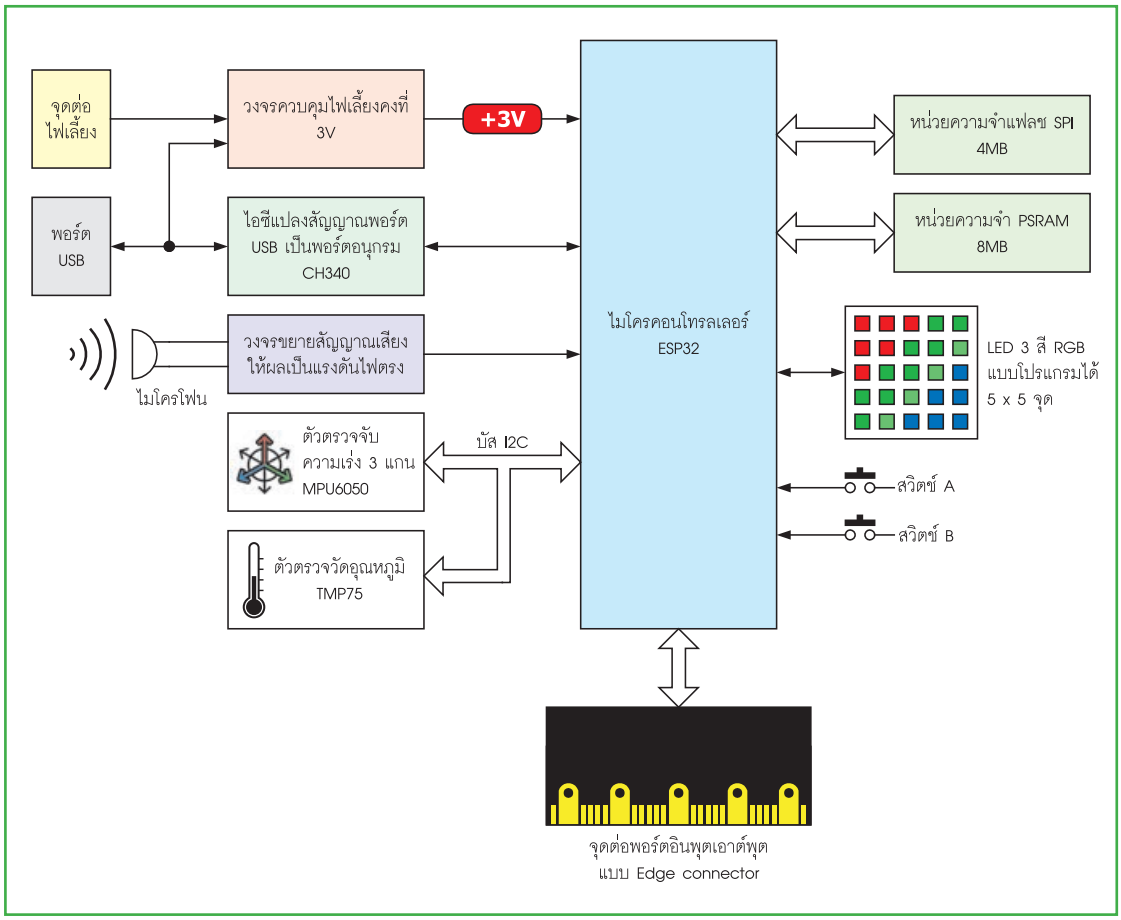
รูปที่ 1-1 ความร่วมมือในการพัฒนาฮาร์ดแวร์และซอฟต์แวร์สำหรับบอร์ด Mbits V2

ส่วนประกอบของ Mbits V2

รูปที่ 2 แสดงส่วนประกอบต่างๆ ของบอร์ด Mbits V2 มีไดอะแกรมการทำงานภายในแสดงในรูปที่ 3 และวงจรสมบูรณ์แสดงดังรูปที่ 4 อุปกรณ์หลักคือ โมดูลไมโครคอนโทรลเลอร์ ESP32 รุ่น Wrover-B มีหน่วยความจำแฟลช 4MB และหน่วยความจำ PSRAM เพิ่มเติมอีก 8MB ภายในบรรจุวงจรสื่อสารไร้สาย WiFi และบลูทูธ 4.2 รองรับการทำงานในแบบบลูทูธพลังงานต่ำหรือ BLE (Bluetooth Low Energy) มีวงจรเชื่อมต่ออุปกรณ์ภายนอกหลากหลายทั้งแบบบัส 2 สายหรือ I²C, บัส SPI, บัส UART และพอร์ตอินพุตเอาต์พุตทั้งแบบดิจิทัลและแอนะล็อก มีวงจรกำเนิดสัญญาณ PWM จึงทำให้นำบอร์ด Mbits ไปใช้ในการสร้างโครงงานได้อย่างหลากหลายตั้งแต่ระบบควบคุมอัตโนมัติพื้นฐาน ของเล่น เครื่องใช้แบบฉลาดที่ทำงานได้อย่างอัตโนมัติ โครงงานติดต่อกับตัวตรวจจับเพื่อสร้างระบบเก็บข้อมูล ไปจนถึงระบบควบคุมไร้สายผ่าน WiFi หรือ BLE หรือกระทั่งควบคุมผ่านเครือข่ายอินเทอร์เน็ต



รูปที่ 1-2 ส่วนประกอบของบอร์ด Mbits V2

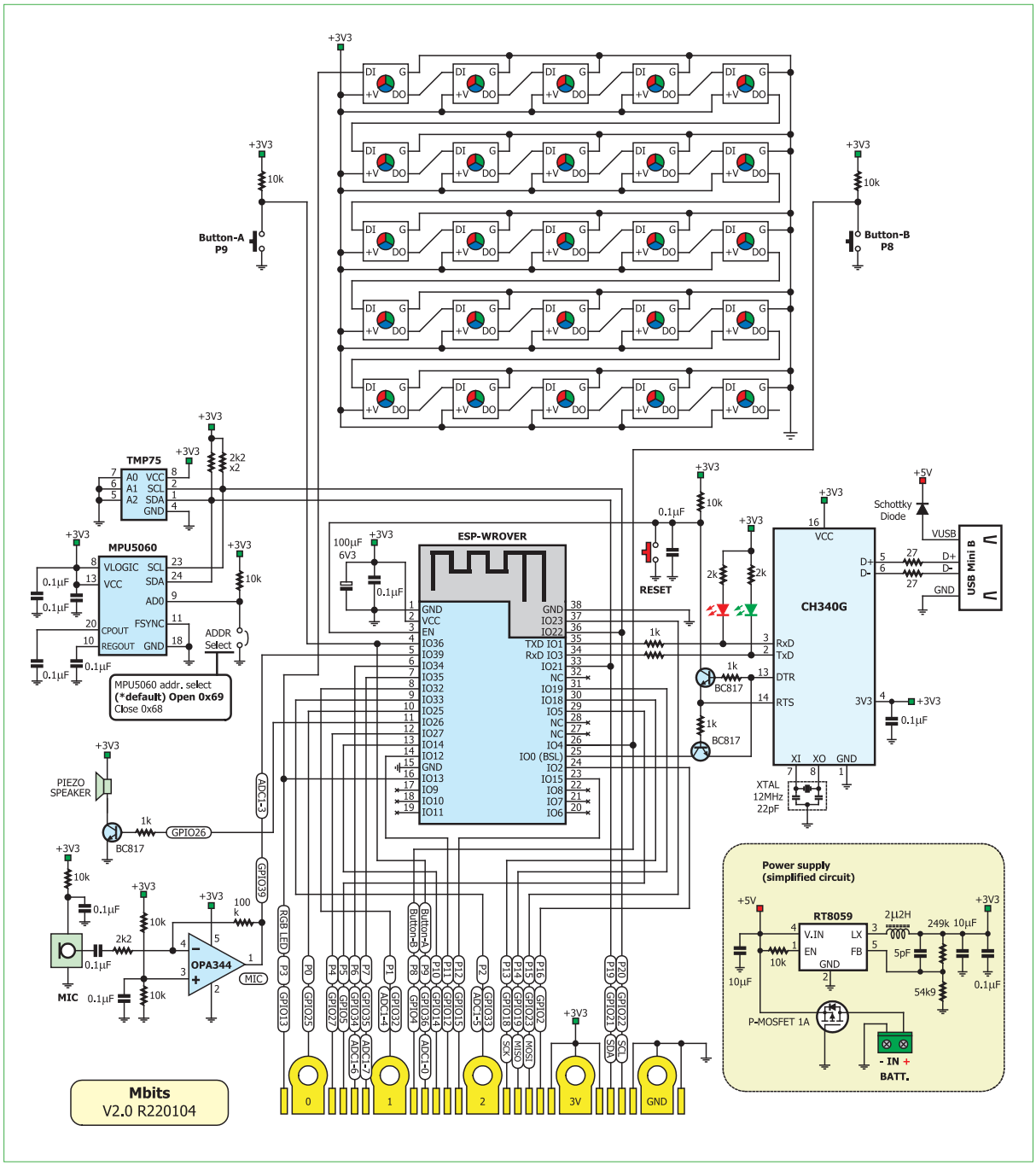


รูปที่ 1-3 ไดอะแกรมการทำงานของบอร์ด Mbits V2

การแสดงผลของ Mbits เป็น LED 3 สี RGB แบบโปรแกรมได้จำนวน 25 ดวง จัดเรียงในแบบ 5 x 5 จุด รองรับการแสดงผลข้อความ ตัวเลข สัญลักษณ์ เส้น จุด และกราฟิก 2 มิติที่เลือกสีในการแสดงผลได้ด้วยการเขียนโปรแกรม

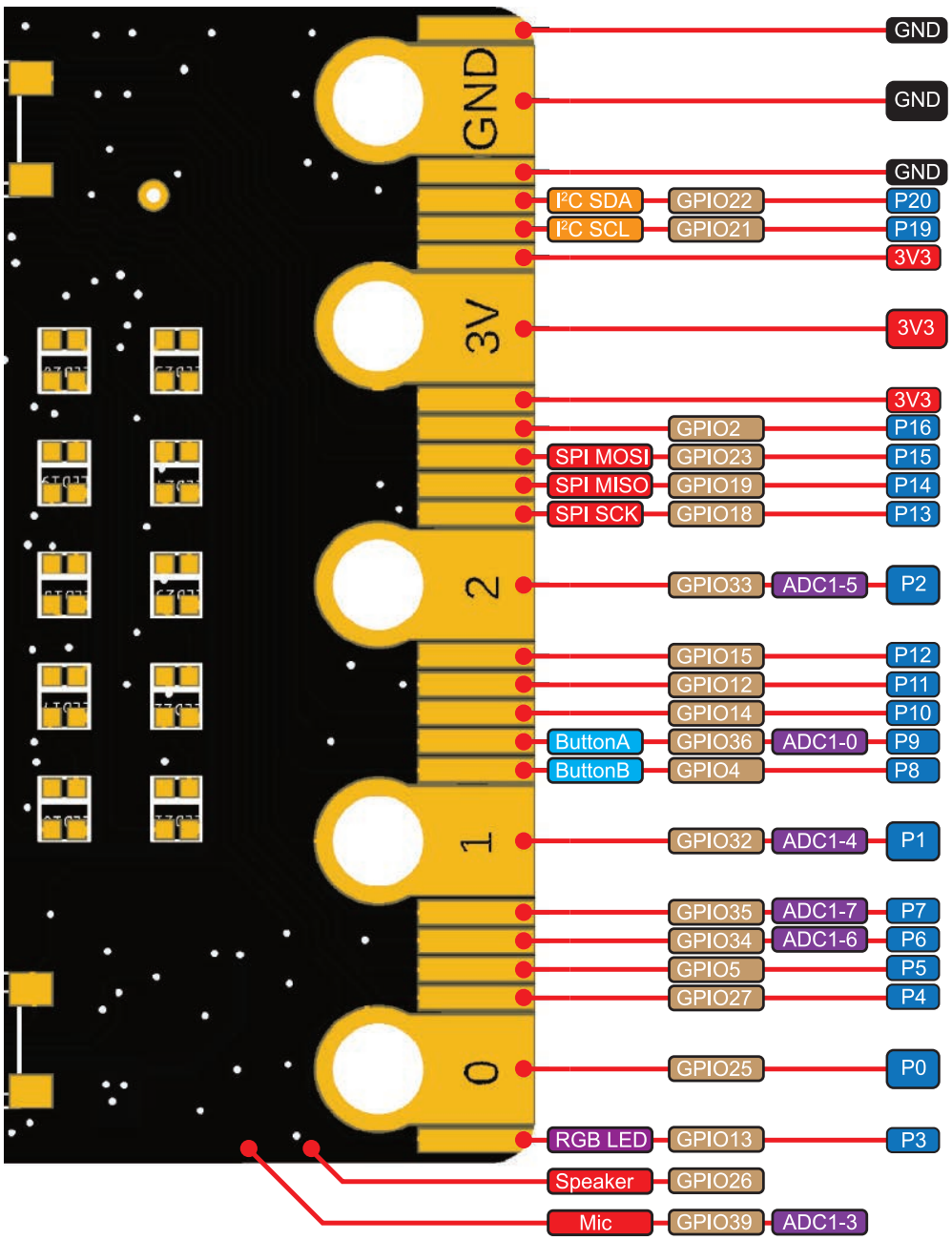
Mbits มีวงจรขับเสียงผ่านลำโพงเปียโซ ขับสัญญาณในย่านความถี่เสียงได้รวมถึงเสียงโน้ตดนตรี ในส่วนของตัวตรวจจับที่ติดตั้งมาพร้อมใช้งานประกอบด้วย

1. ตัวตรวจจับความเร่ง 3 แกนเบอร์ MPU6050 ใช้ตรวจจับความเร่ง ความเอียง การตกหรือเคลื่อนที่ของบอร์ด ติดต่อกับไมโครคอนโทรลเลอร์หลัก ESP32 ผ่านระบบบัส I^2C
2. ตัวตรวจวัดอุณหภูมิเบอร์ TMP75 ติดต่อกับไมโครคอนโทรลเลอร์หลัก ESP32 ผ่านระบบบัส I^2C
3. วงจรตรวจจับเสียงโดยใช้ไมโครโฟน ให้ผลการทำงานเป็นแรงดันไฟตรงที่แปรตามความดังของเสียง



รูปที่ 1-4 วงจรของบอร์ด Mbits V2

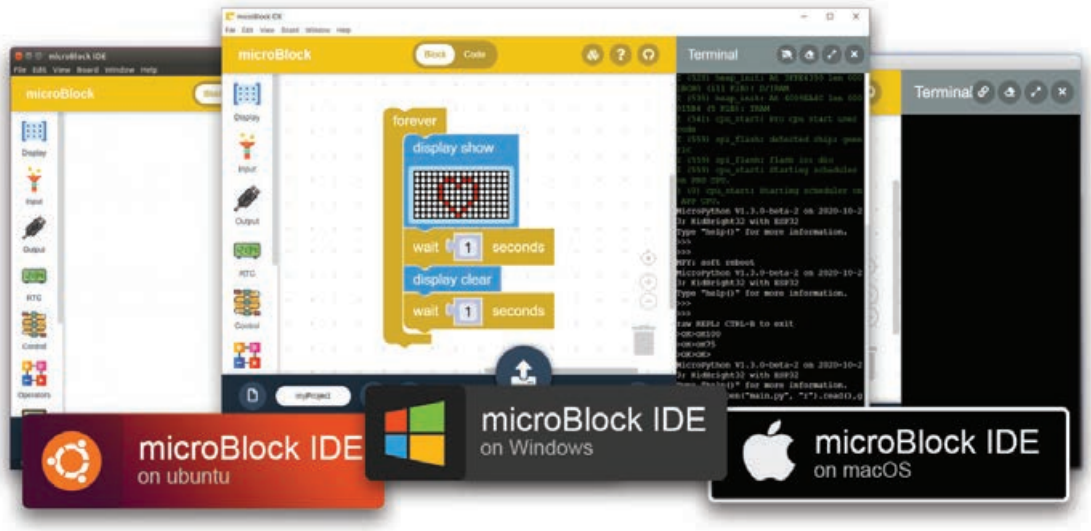
จุดต่ออุปกรณ์ภายนอกของ Mbits V2 เป็นจุดต่อแบบ Edge connector โดยแบ่งเป็นจุดต่อหลัก 5 จุด ที่เป็นรูขนาด 4 มม. และจุดต่อรองที่เป็นหน้าสัมผัสชุบทองรวม 20 จุด เป็นขาพอร์ตอินพุตเอาต์พุต P0 ถึง P19 ขาต่อไฟเลี้ยง 3V และกราวด์ ดังแสดงการกำหนดหน้าที่การทำงานของขาพอร์ตต่างๆ ในรูปที่ 5



รูปที่ 1-5 การจัดขาพอร์ตอินพุตเอาต์พุตของบอร์ด Mbits V2

บทที่ 2 ติดตั้งและใช้งาน microBLOCK IDE

microBlock IDE เป็นซอฟต์แวร์ที่ใช้พัฒนาโปรแกรมสำหรับ ESP32
ด้วยการลากบล็อกหรือเขียนโค้ดด้วยภาษา microPython

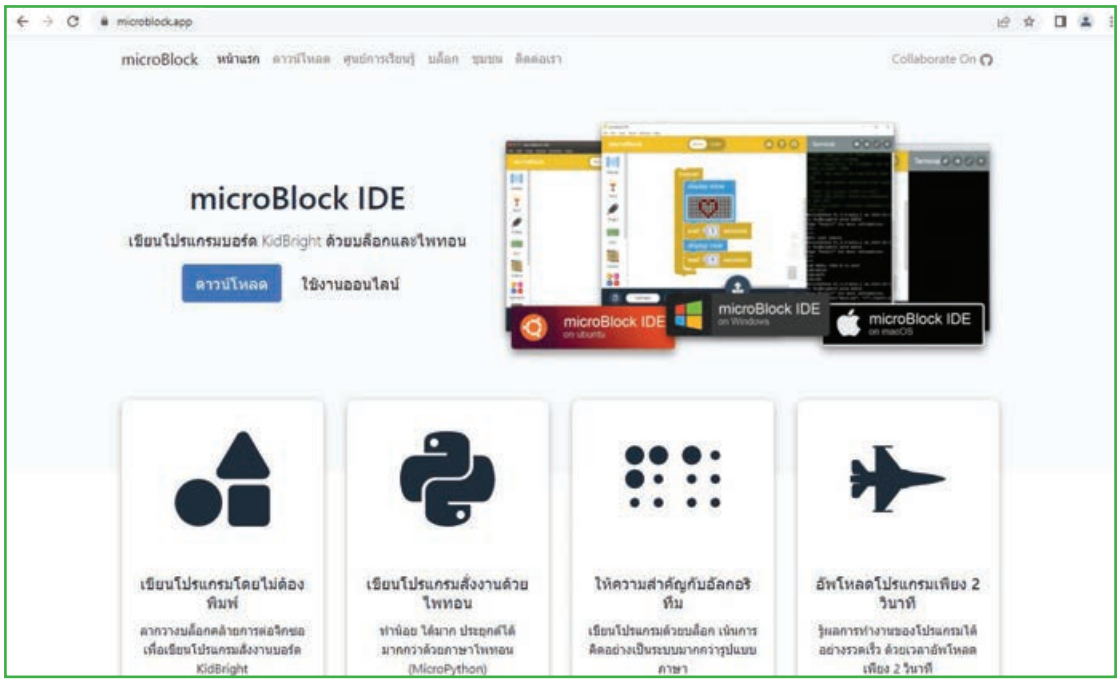


รูปที่ 2-1 ซอฟแวร์ microBlock IDE

2.1 คุณสมบัติเด่นของ microBlock IDE

- อัปโหลดโปรแกรมเร็วมากเพียง 2 วินาที
- สามารถค้นหาและเชื่อมต่อบอร์ด ESP32 (Mbits) โดยอัตโนมัติ
- มีหน้าต่าง Terminal ตรวจสอบผลการทำงานได้ง่าย
- มีระบบให้คำแนะนำการเขียนโค้ด (Auto Completion)
- ใช้งานบล็อกขับ LED และบล็อกสร้างเสียงเพลงได้ง่าย
- รองรับการเพิ่มบล็อกในโปรแกรมด้วย Extension
- มีระบบบันทึกไฟล์อัตโนมัติ หมดปัญหาไฟดับแล้วไฟล์งานหาย
- แปลงโค้ดจากบล็อกเป็น ไมโครไพทอน และสามารถแก้ไขโค้ดด้วยภาษาไมโครไพทอนได้เลย
- มีให้ใช้งานทั้งแบบ Offline ติดตั้งบนคอมพิวเตอร์และ Online ทำงานผ่านบราวเซอร์

สำรวจ https://microblock.app



รูปที่ 2-2

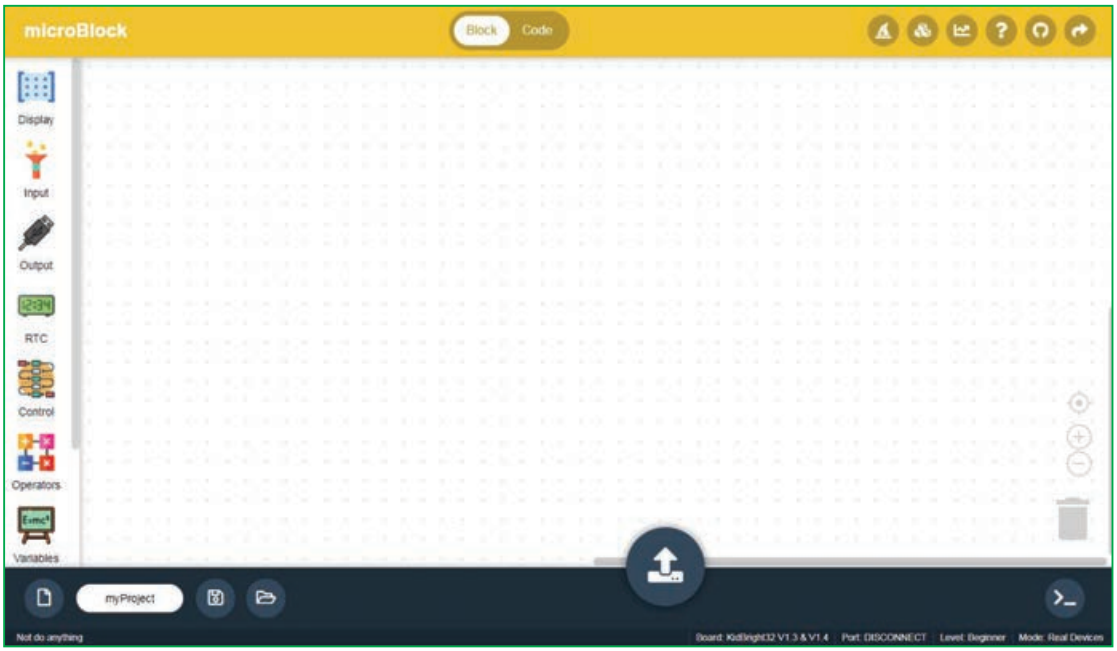
หน้าแรกของเว็ปไซต์ จะแนะนำคุณสมบัติต่างๆ ของ microBlock IDE มุ่งสู่การดาวน์โหลดโปรแกรมมาติดตั้ง กดที่ปุ่ม**ดาวน์โหลด**

รูปที่ 2-3

ดับเบิ้ลคลิกที่ไฟล์เพื่อติดตั้ง โปรแกรมจะใช้เวลาแตกไฟล์และติดตั้ง จากนั้นเปิดหน้าต่างของโปรแกรม microBlock IDE ขึ้นมา

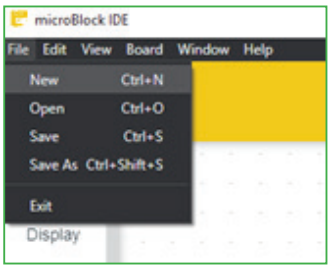
2.2 ใช้งาน microBlock IDE กับ Mbits

เมื่อเปิดโปรแกรมขึ้นมาครั้งแรก ตัวเลือกของบอร์ดที่ใช้อาจจะเป็นบอร์ดไมโครคอรโทรลเลอร์ตัวอื่นๆ การเลือกบอร์ดไมโครคอนโทรลเลอร์ ทำได้โดยการสร้างโปรเจ็กต์ใหม่

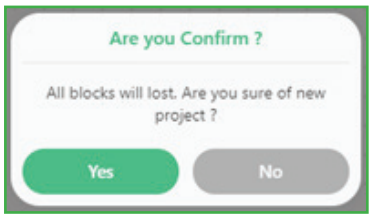


รูปที่ 2-4 หน้าตาโปรแกรม microBlock IDE

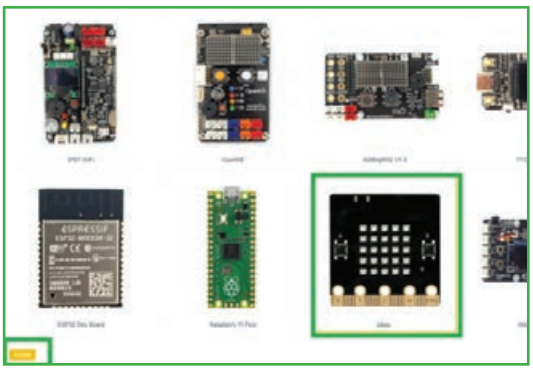
(1) ที่เมนู **File** เลือก **New** เพื่อสร้างโปรเจ็กต์ใหม่



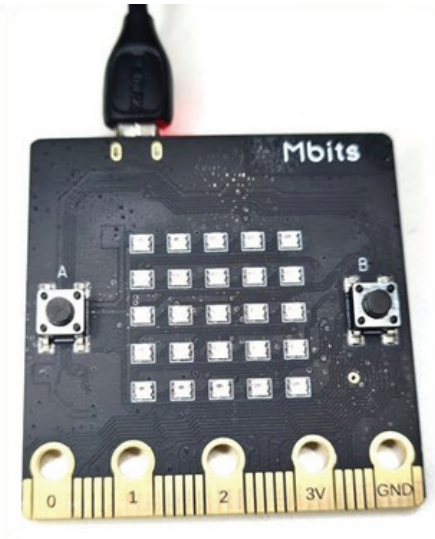
(2) โปรแกรมจะถามยืนยัน ว่าข้อมูลเดิมจะสูญหาย แน่ใจหรือไม่ให้ตอบ **YES**



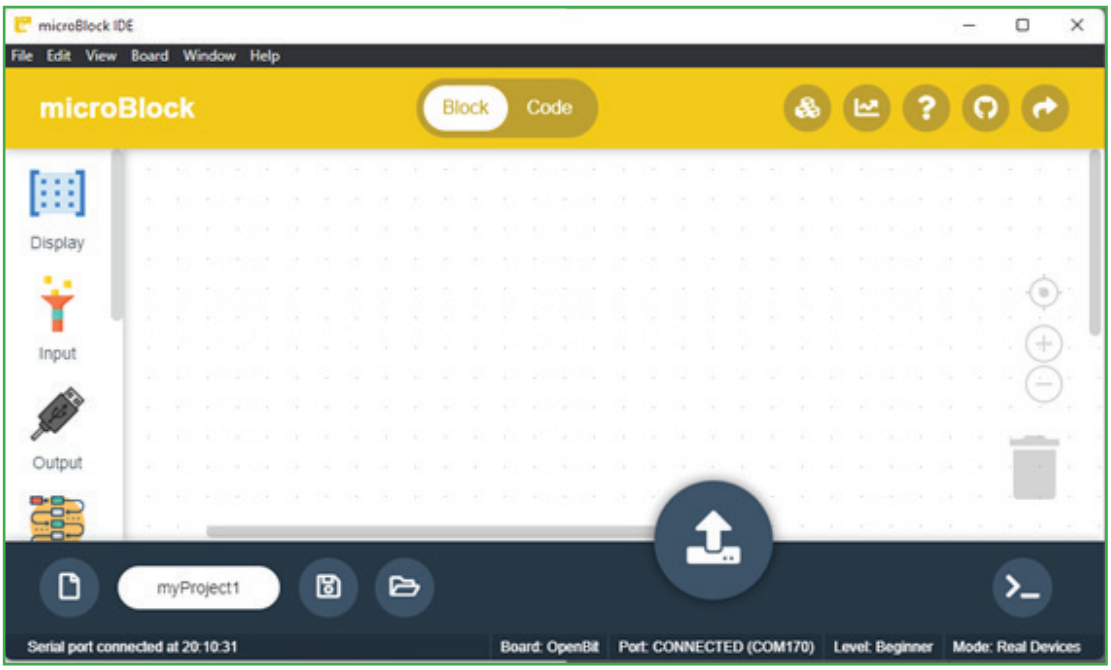
(3) จากบอร์ดต่างๆ ที่มี ให้เลือก **Mbits** พร้อมทั้งตั้งชื่อโปรเจ็กต์ จากนั้นเลื่อนลงด้านล่างกดปุ่ม **Create**



(4) เชื่อมบอร์ด Mbits เข้ากับคอมพิวเตอร์ผ่านสาย microUSB



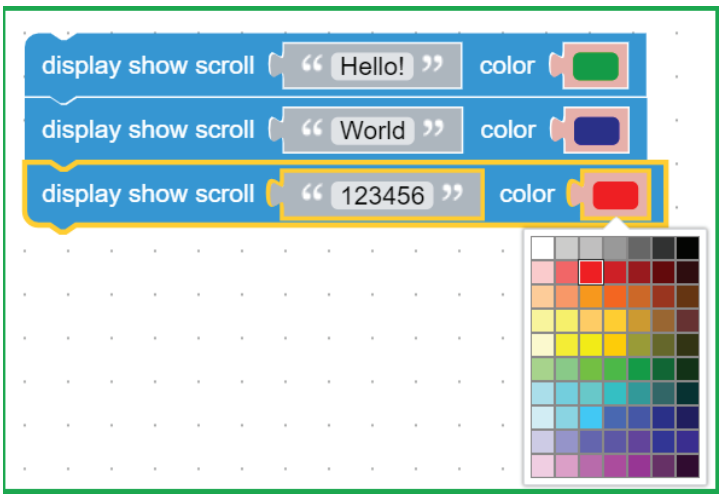
(5) แถบสถานะด้านล่างของโปรแกรมจะตรวจพบพอร์ตอนุกรมและเชื่อมต่อโดยอัตโนมัติ



ถึงตอนนี้ Mbits ก็พร้อมเขียนโค้ดด้วย microBlock IDE แล้ว

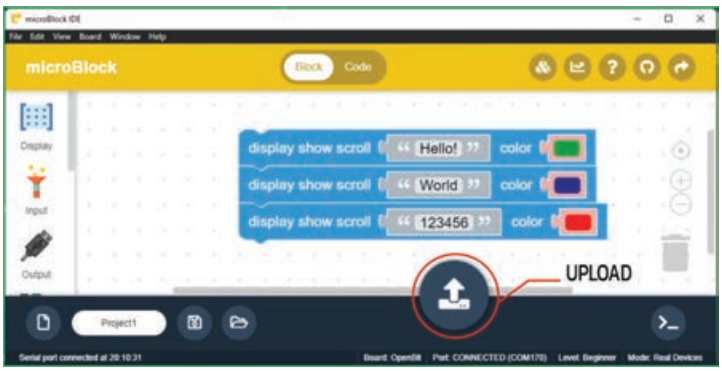
{ ตัวอย่างที่ 1 ข้อความเลื่อน }

เป็นการนำข้อความมาแสดงแบบเลื่อนที่ LED RGB 5×5 จุด โดยกำหนดสีของ LED ได้ตามต้องการ



โปรแกรมที่ 1-1

นำคำสั่ง display show scroll จากเมนู Display มาปรับเปลี่ยนข้อความและเลือกสี



รูปที่ 2-5

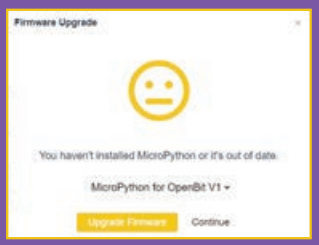
เมื่อวางบล็อกเป็นที่พอใจแล้วกดปุ่ม Upload เพื่อส่งโค้ดไปยังตัว Mbits

เมื่ออัปโหลดโค้ดเรียบร้อย บอร์ด Mbits จะเริ่มทำงานทันที

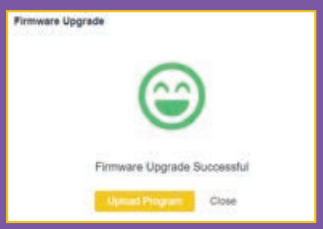
สแกน QR Code เพื่อดูผลลัพธ์

หมายเหตุ

ในการดาวน์โหลดโค้ดครั้งแรก โปรแกรมจะตรวจสอบเฟิร์มแวร์ของบอร์ด Mbits ก่อน ถ้าเป็นรุ่นเก่าจะมีหน้าต่างแจ้งให้ Upgrade Firmware กดปุ่ม Upgrade Firmware ได้เลย

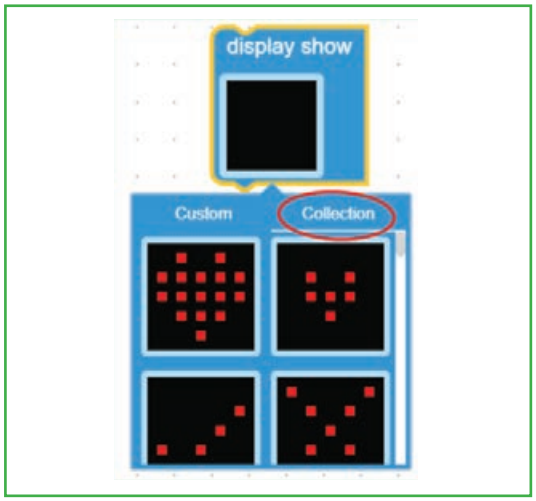


เมื่อเสร็จเรียบร้อยแล้ว จะมีหน้าต่างแสดงความพึงพอใจปรากฏขึ้นมา



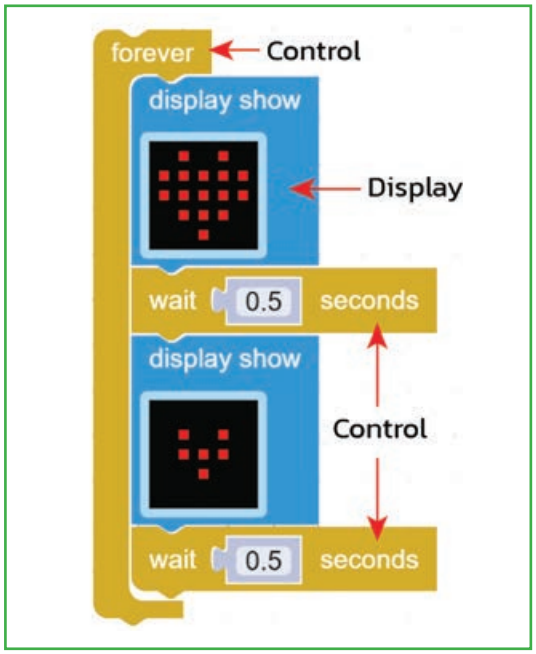
{ ตัวอย่างที่ 2 ไอคอนรูปหัวใจกะพริบ }

คำสั่ง display show ใช้กำหนดให้ LED RGB 5×5 ติดในตำแหน่งที่ต้องการเป็นสีใดๆ ก็ได้



รูปที่ 2-6

นอกจากนี้ใน Tab Collection ยังมีไอคอนต่างๆ ที่ทำไว้แล้วให้เลือกใช้งาน สามารถเลือกแล้วนำมาปรับแก้ได้ด้วย



โปรแกรมที่ 2-1

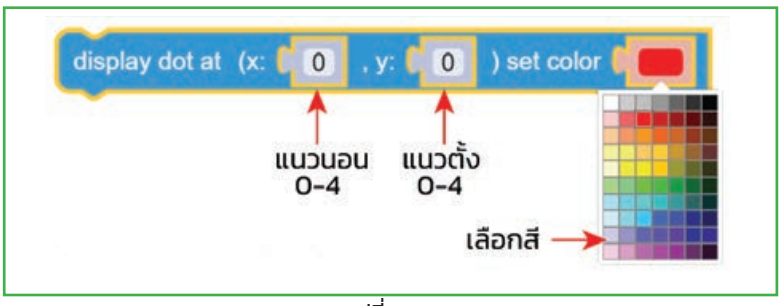
ตัวอย่างนี้จะนำไอคอนรูปหัวใจที่มีขนาดใหญ่และขนาดเล็ก มาแสดงสลับกัน หน่วงเวลาทุกๆ 0.5 วินาที

ผลลัพธ์ที่ได้จะเห็น LED RGB 5x5 จุดติดกะพริบ

สแกน QR Code เพื่อดูผลลัพธ์

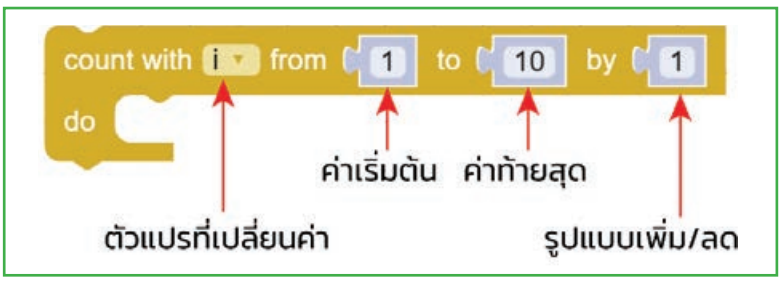
{ ตัวอย่างที่ 3 สั่ง LED ไล่ลำดับติดทีละดวง }

เพื่อให้ LED RGB 5×5 ติดในพิกัดที่กำหนด จะมีคำสั่ง display dot at ในหัวข้อ Display ระบุพิกัด x (0 ถึง 4) และ y (0 ถึง 4) ของ LED พร้อมทั้งระบุสีที่ต้องการได้ด้วย คำสั่งนี้ถ้าต้องการให้ LED RGB ดับต้องกำหนดเป็นสีดำได้



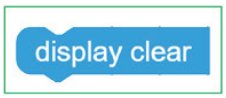
รูปที่ 2-7

เพื่อวนลูปทำซ้ำในช่วงที่กำหนด จะใช้คำสั่ง count with (เทียบเท่าคำสั่ง for) ในหัวข้อ Control เพื่อวนลูปและเปลี่ยนแปลงค่าตัวแปรไปจนถึงค่าที่กำหนด

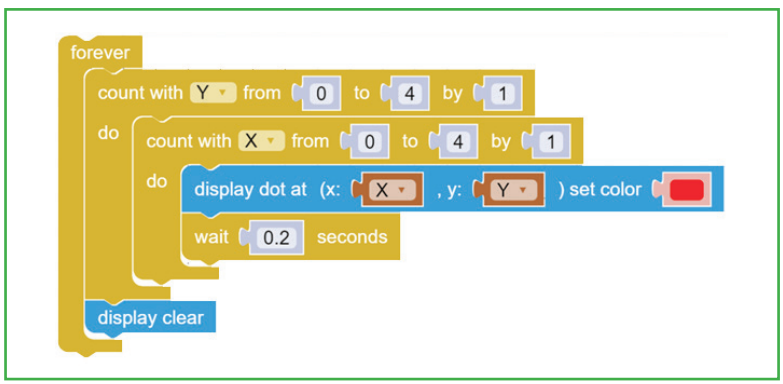


รูปที่ 2-8

เมื่อทำงานจนเสร็จครบ 1 รอบ จะเคลียร์ LED 5×5 จุด ด้วยคำสั่ง display clear



จากคำสั่งนำมาสร้างโค้ดได้ดังโปรแกรมที่ 3-1



โปรแกรมที่ 3-1

ผลลัพธ์ที่เกิดขึ้นจะเห็น LED RGB 5×5 จุด ของ Mbits ไล่ติดทีละดวงจนครบ

สแกน QR Code เพื่อดูผลลัพธ์

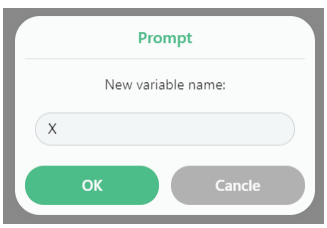
{ ตัวอย่างที่ 4 LED 5×5 สุ่มติด }

การสุ่มตัวเลข

ในการสุ่มตัวเลขใช้คำสั่ง random integer from ในหัวข้อ Operators จะคืนค่าตัวเลขสุ่มในช่วงที่กำหนดออกมาให้ ถ้าดัดแปลงโค้ดในตัวอย่างที่ 3 จะได้โปรแกรมที่กำหนดให้ LED ติดสุ่มได้

การกำหนดค่าตัวแปร

ในหัวข้อ Variable จะมีบล็อกคำสั่ง Create Variable เพื่อสร้างตัวแปรไว้จองพื้นที่หน่วยความจำเพื่อเก็บข้อมูล



เมื่อกดสร้าง จะมีหน้าต่าง ให้ระบุชื่อตัวแปร ในที่นี้ตั้งชื่อว่า X

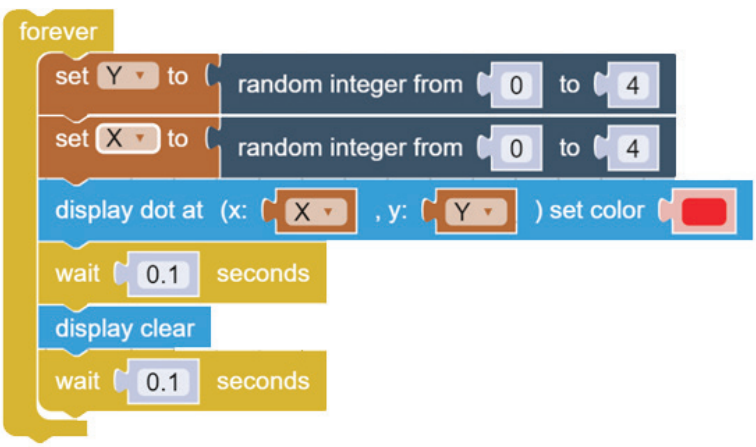
เมื่อกำหนดค่าตัวแปรเรียบร้อย จะมีบล็อกเพิ่ม 3 บล็อกคือ

สำหรับกำหนดค่าตัวแปรด้วยค่าที่ต้องการ

สำหรับเพิ่มหรือลดค่าตัวแปร

ผลลัพธ์ของค่าตัวแปร

เมื่อเข้าใจคำสั่งก็นำมาสร้างเป็นโค้ดดังโปรแกรมที่ 4-1

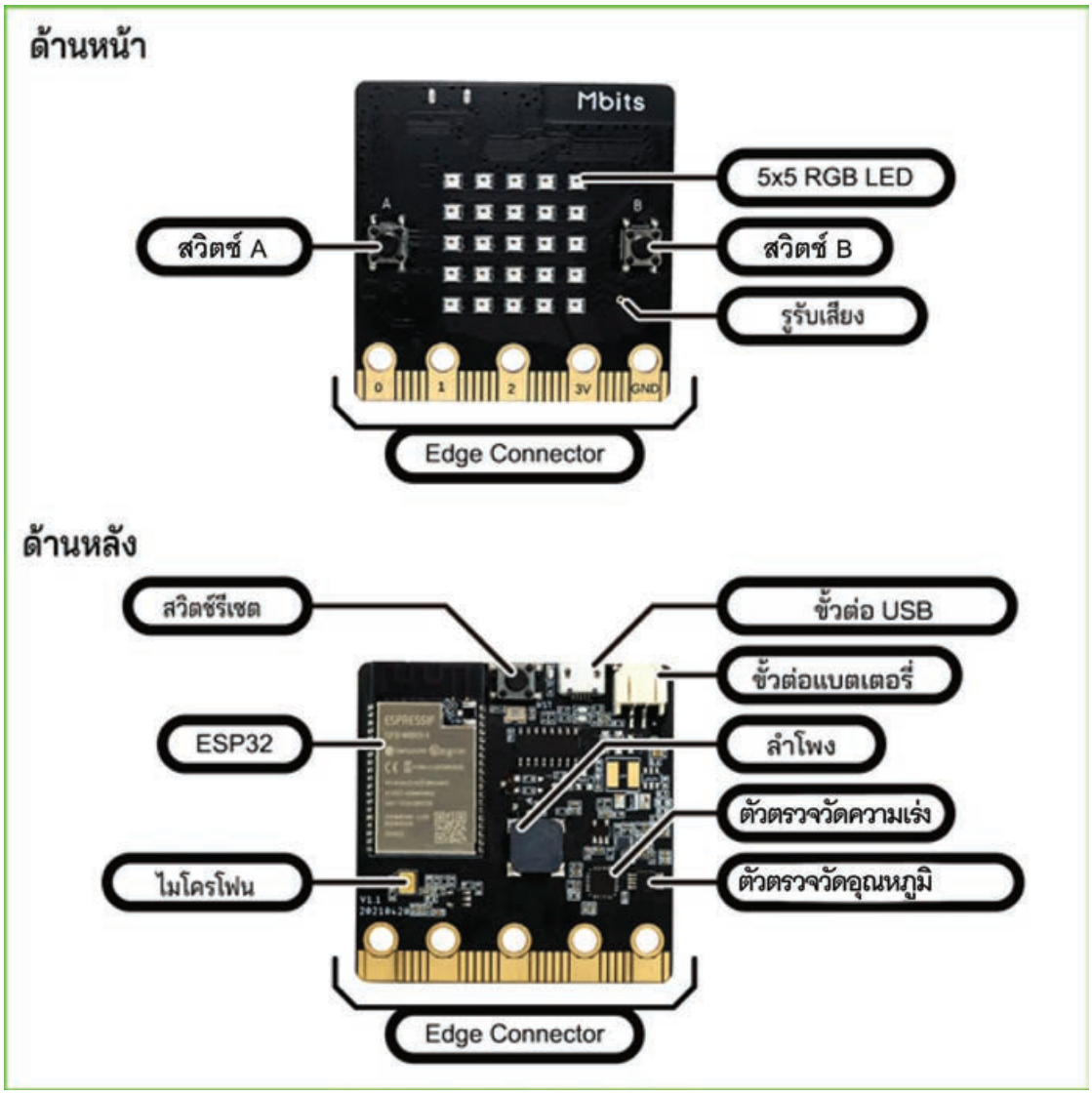


สแกน QR Code เพื่อดูผลลัพธ์

โปรแกรมที่ 4-1

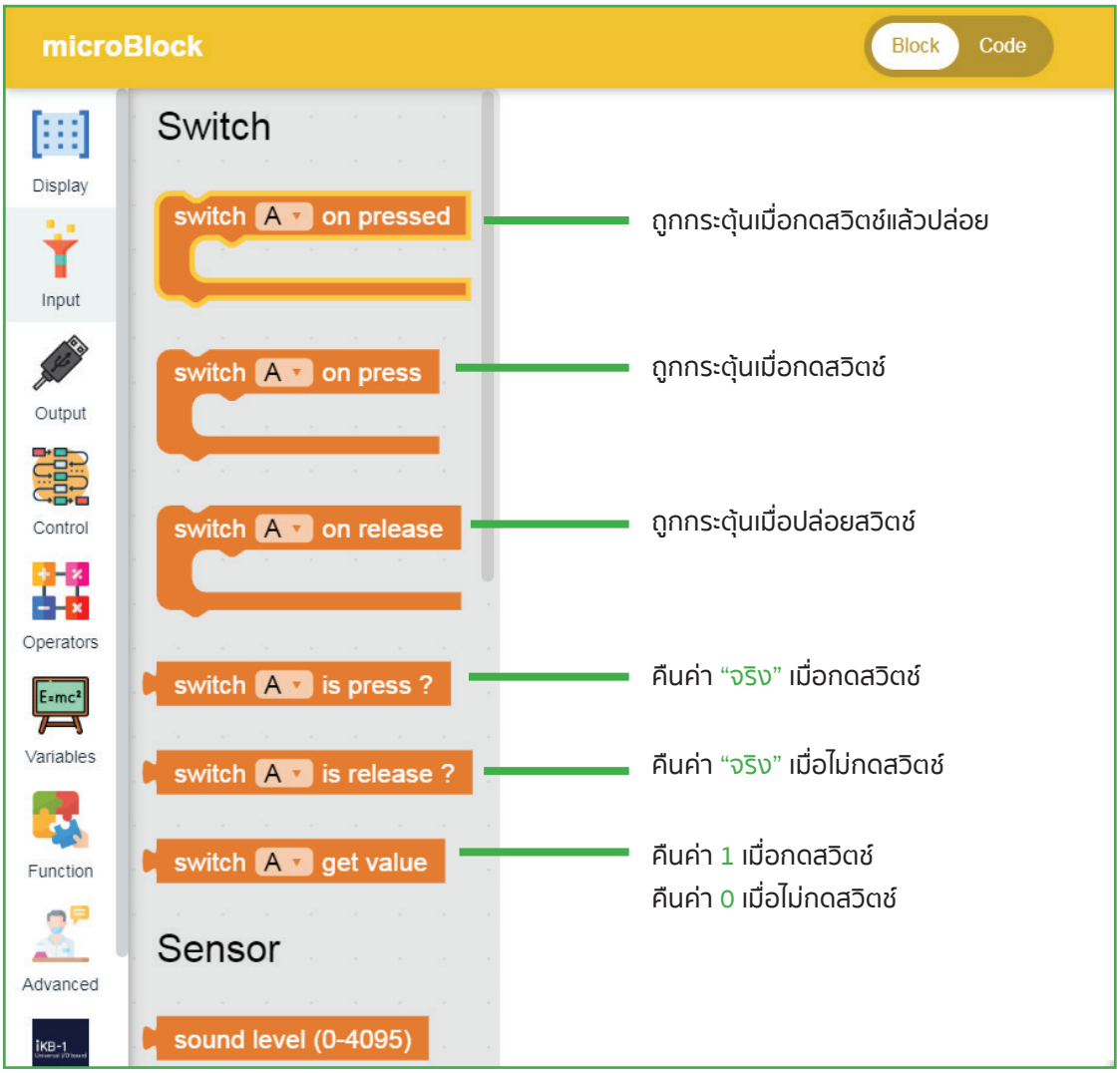
บทที่ 3 ใช้งานอุปกรณ์อินพุต/เอาต์พุต

บอร์ด Mbits ถือว่าเป็นบอร์ดที่สมบูรณ์พร้อมในเรื่องอุปกรณ์อินพุตเอาต์พุตที่ติดตั้งอยู่ในตัว ประกอบด้วย LED สวิตช์ ไมโครโฟน ลำโพง ตัวตรวจวัดอุณหภูมิและตัวตรวจวัดความเร่ง 3 แกน



รูปที่ 3-1 แสดงส่วนประกอบ Mbits V2

3.1 Mbits กับ สวิตช์ A และ สวิตช์ B



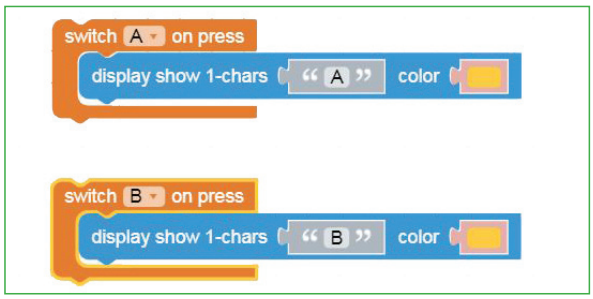
รูปที่ 3-2 แสดงบล็อกคำสั่งการใช้งานสวิตช์ A และ B ของ Mbits V2

{ ตัวอย่างที่ 5 อ่านค่าสวิตช์ A และ B }

เขียนโค้ดสั้นๆ ก่อนเพื่อให้เห็นความแตกต่างในการทำงานของบล็อกคำสั่ง on press กับ on pressed

รูปที่ 3-3

โปรแกรมที่ 5-1 รูปแบบ on press

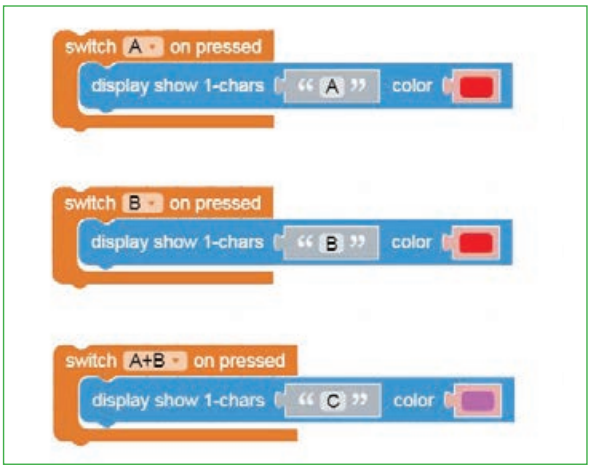


โปรแกรมที่ 5-1

สแกน QR Code เพื่อดูผลลัพธ์

ผลลัพธ์ของ on press จะเกิดขึ้นทันทีที่กดสวิตช์

โปรแกรมที่ 5-2 รูปแบบ on pressed

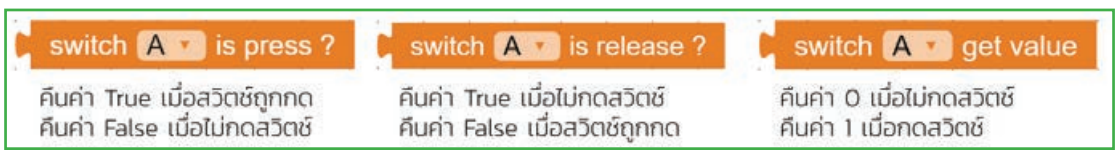


โปรแกรมที่ 5-2

สแกน QR Code เพื่อดูผลลัพธ์

ผลลัพธ์ของ on pressed จะเกิดขึ้นหลังจากที่ปล่อยสวิตช์แล้ว ดังนั้นถ้ากดสวิตช์ค้างอยู่จะยังไม่เห็นผลลัพธ์ปรากฏ

หมวดหมู่การคืนค่าจากสวิตช์



รูปที่ 3-4

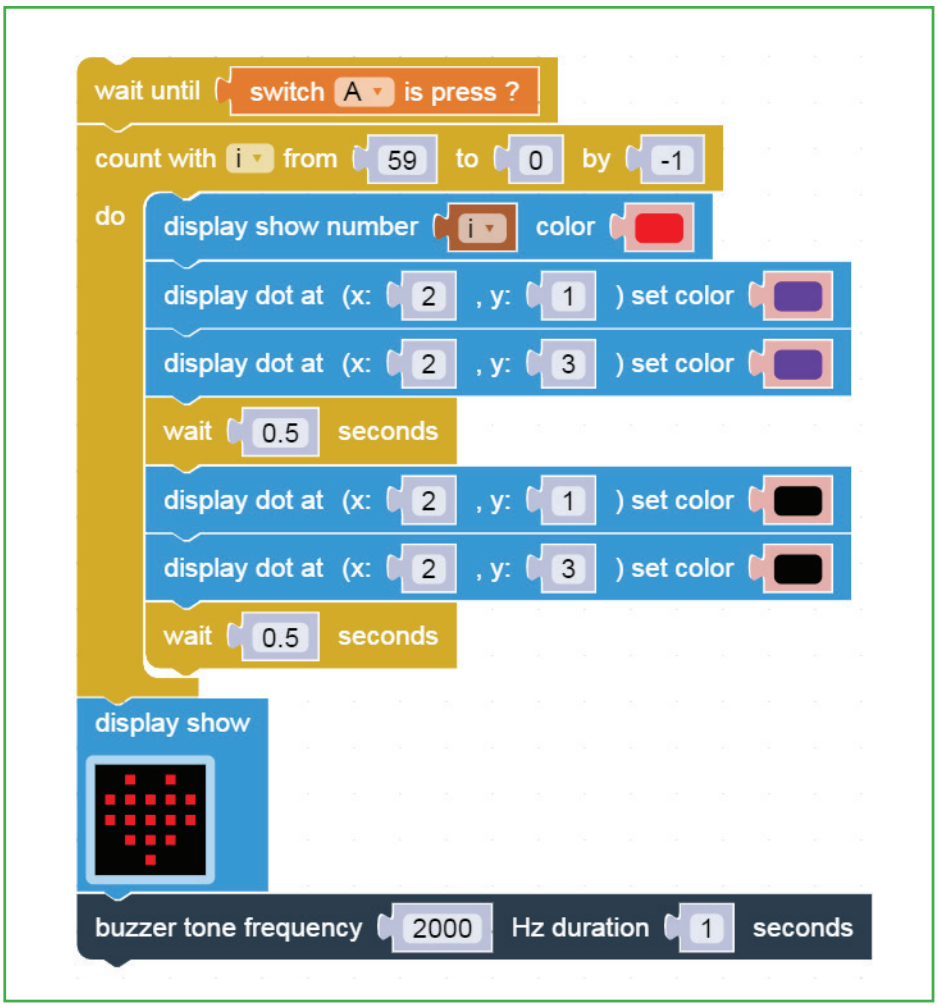
{ ตัวอย่างที่ 6 นับถอยหลัง 60 วินาที }

ตัวอย่างนี้จะใช้บล็อก wait until รอจนกระทั่งกดสวิตช์ A จึงไปทำคำสั่งถัดไป โดยจะนับค่าตัวเลขถอยหลังจาก 59 ไปจนถึง 0 ทุกๆ 1 วินาทีจะให้แสดง : ตรงกลางระหว่างตัวเลขด้วย

เมื่อนับครบให้แสดงรูปหัวใจพร้อมกับส่งเสียงออกลำโพง

คำสั่ง display show number จะแสดงตัวเลขแบบย่อ ทำให้การแสดงเลข 2 หลักทำได้บน LED 5×5 จุด

สแกน QR Code เพื่อดูผลลัพธ์



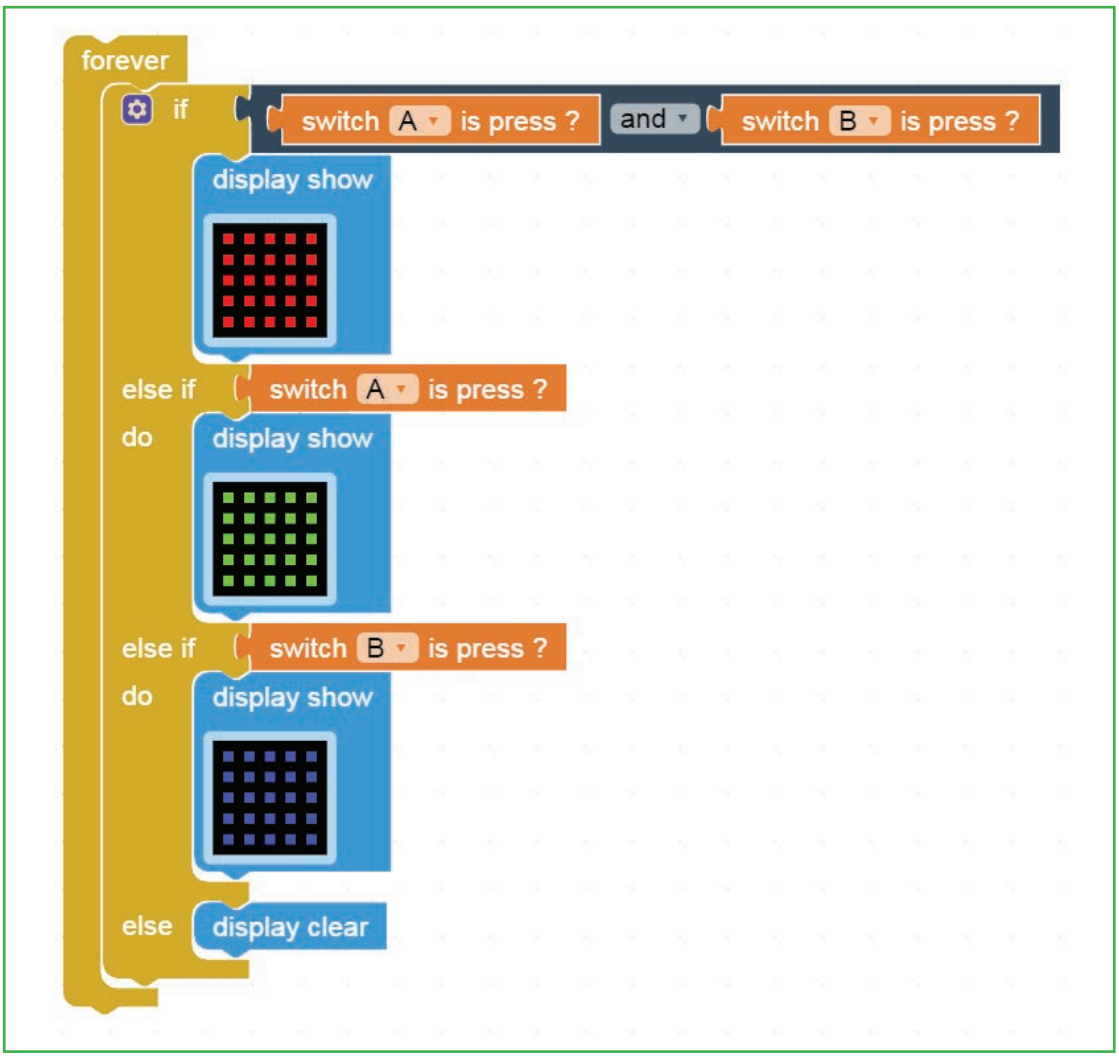
โปรแกรมที่ 6-1

{ ตัวอย่างที่ 7 แสดงสถานะการกดสวิตช์ที่ LED RGB 5×5 จุด}

วนลูปตรวจสอบเงื่อนไขการกดสวิตช์

เมื่อสวิตช์ A และ B ถูกกดทั้งคู่ ให้แสดง LED RGB 5×5 จุด เป็นสีแดง
กดสวิตช์ A ตัวเดียว ให้แสดง LED RGB 5×5 จุด เป็นสีเขียว
กดสวิตช์ B ตัวเดียว ให้แสดง LED RGB 5×5 จุด เป็นสีน้ำเงิน
ไม่มีการกดสวิตช์ ทำการเคลียร์หน้าจอ

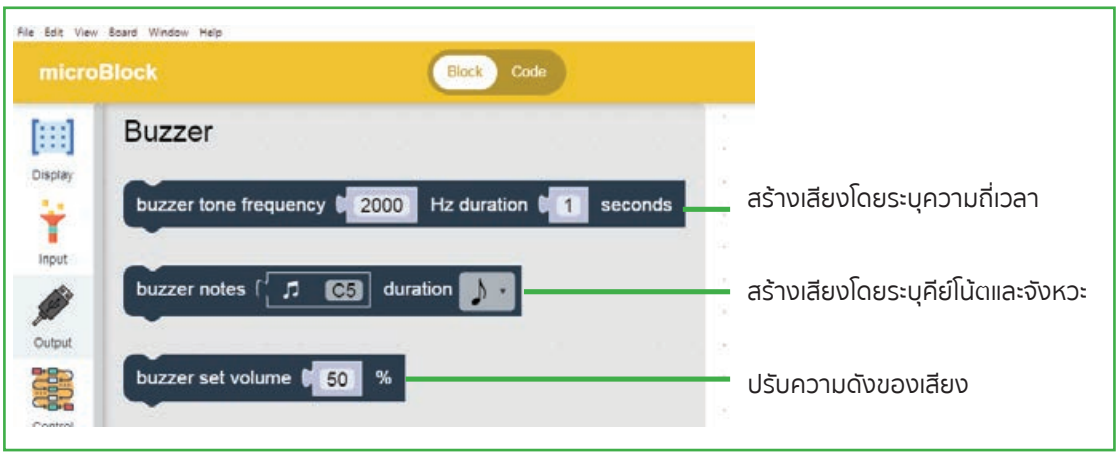
สแกน QR Code เพื่อดูผลลัพธ์



โปรแกรมที่ 7-1

3.2 Mbits ขับเสียงออกลำโพง

บนบอร์ด Mbits มีลำโพงขนาดเล็กติดตั้งอยู่ เพื่อขับเสียงออกมาจากการทำงานของคำสั่งในหัวข้อ **Output**



รูปที่ 3.5

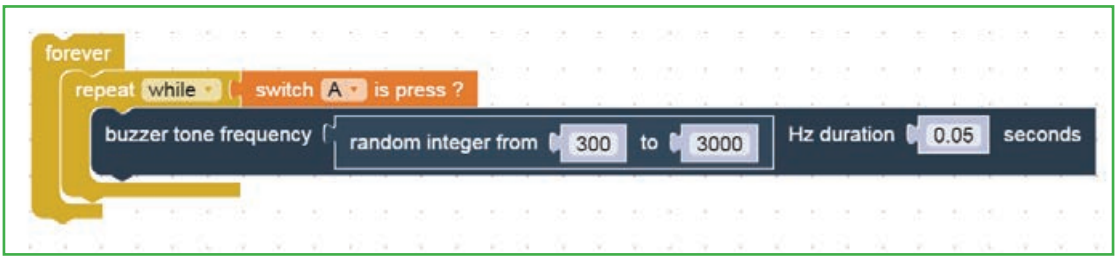
3.2.1 บล็อก buzzer tone frequency

สร้างเสียงโดยระบุความถี่ในหน่วยเป็นเฮิรตซ์ (Hz) และระยะเวลาให้เสียงดังในหน่วยเป็นวินาที สัญญาณเสียงที่ลำโพงตอบสนองได้ดีจะอยู่ในช่วง 300 ถึง 5000 Hz

{ ตัวอย่างที่ 8 สุ่มสร้างเสียงออกลำโพง }

ตัวอย่างนี้จะคอยตรวจสอบ **สวิตช์ A** ถ้ามีการกดอยู่จะให้สร้างเสียงต่อเนื่องไปเรื่อยๆ จนกว่าจะมีการปล่อยสวิตช์ โดยความถี่เสียงจะถูกสุ่มในช่วง 300 ถึง 3000 Hz

สแกน QR Code เพื่อดูผลลัพธ์

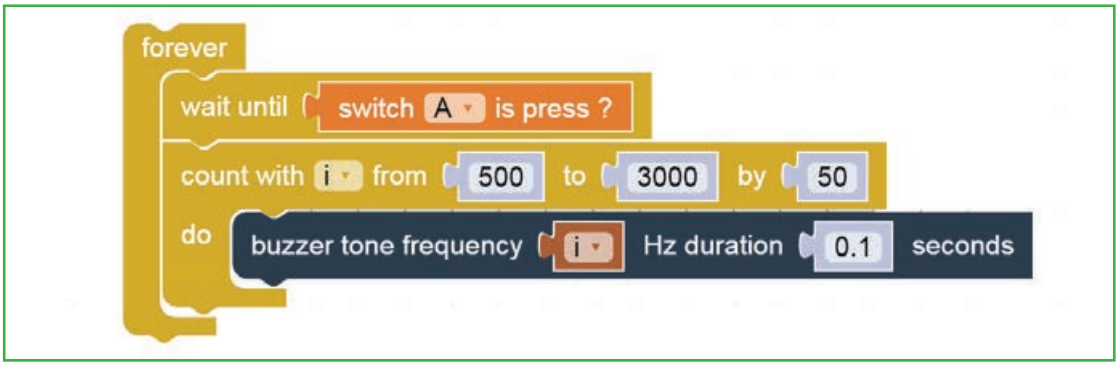


โปรแกรมที่ 8-1

{ ตัวอย่างที่ 9 ขับเสียงจากทุ้มไปแหลม }

เมื่อกด **สวิตช์ A** จะเป็นการสร้างเสียงไล่ลำดับจากเสียงทุ้มไปเสียงแหลม เมื่อถึงค่าความถี่สูงถึง 3000 แล้วก็จะกลับไปรอการกดสวิตช์รอบใหม่

สแกน QR Code เพื่อดูผลลัพธ์

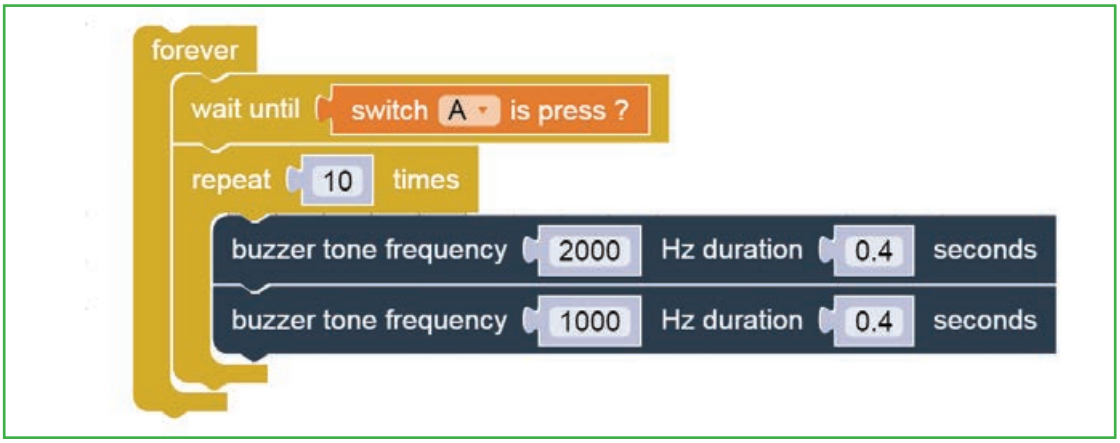


โปรแกรมที่ 9-1

{ ตัวอย่างที่ 10 สร้างเสียงไซเรน }

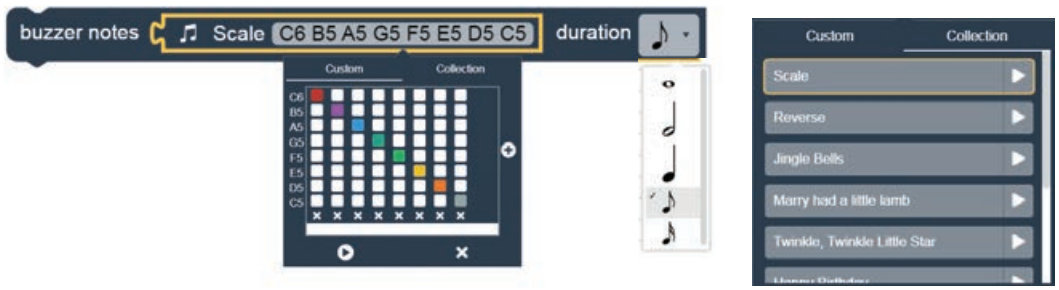
เมื่อกด **สวิตช์ A** จะวนลูป 10 ครั้งเพื่อสร้างเสียงออกลำโพงด้วยความถี่ 2 ความถี่สลับกัน ดังคล้ายกับเสียงไซเรน

สแกน QR Code เพื่อดูผลลัพธ์



โปรแกรมที่ 10-1

3.2.2 บล็อก buzzer note



รูปที่ 3-6

เป็นบล็อกคำสั่งที่ใช้สร้างเสียงแบบเดียวกับ **tone** แต่ระบุค่าเป็นคีย์โน้ตดนตรีและจังหวะของเสียง ต้องระบุค่า 2 ค่าคือ

- **Scale** คือชื่อตัวโน้ตดนตรีกำหนดได้หลายๆ ตัวโน้ตในคำสั่งเดียวและมี **Collection** เพื่อเรียกเพลงตัวอย่างมาเล่นและแก้ไขได้ พร้อมทั้งฟังเสียงโน้ตจากคอมพิวเตอร์ ก่อนการอัปโหลดลงไปยังบอร์ด mBits

- **duration** คือ ระยะเวลาที่ต้องการให้ขับเสียงดนตรีมีด้วยกัน 5 ค่า

โน้ตตัวกลมแทน 4 จังหวะ

โน้ตตัวขาวแทน 2 จังหวะ

โน้ตตัวดำแทน 1 จังหวะ

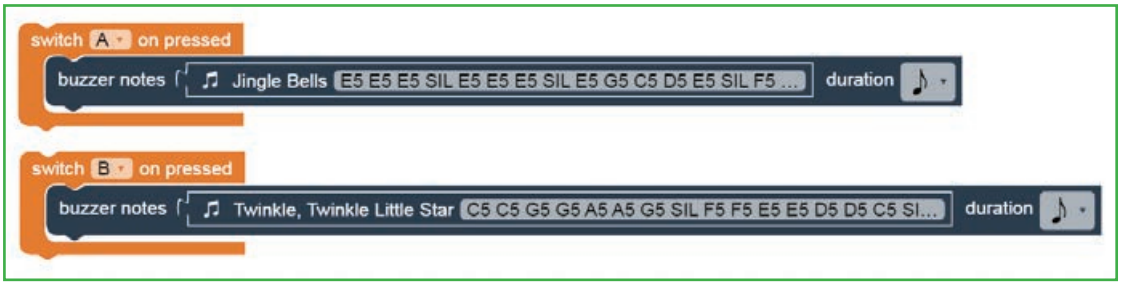
โน้ตเขบ็ต 1 ชั้นแทน 1/2 จังหวะ

โน้ตเขบ็ต 2 ชั้นแทน 1/4 จังหวะ

{ ตัวอย่างที่ 11 เสียงเพลง Twinkle,Twinkle Little Star และ Jingle Bells }

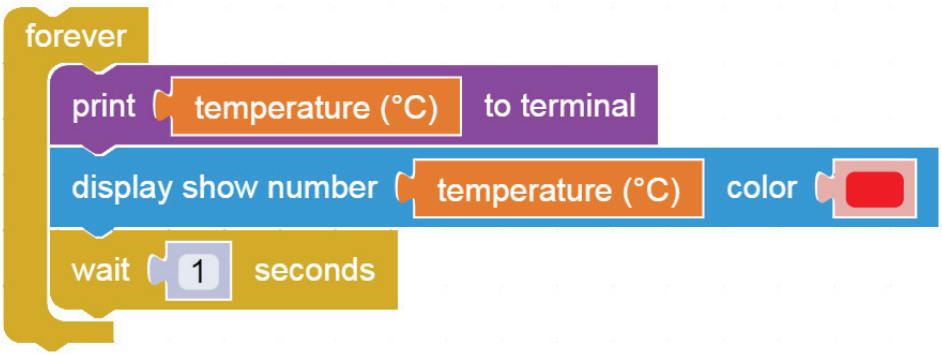
เป็นการเอาเมโลดี้ตัวอย่างจาก 9 ตัวอย่างมาลองเล่นเสียงจากการกด **สวิตช์** A และ B

สแกน QR Code เพื่อดูผลลัพธ์



โปรแกรมที่ 11-1

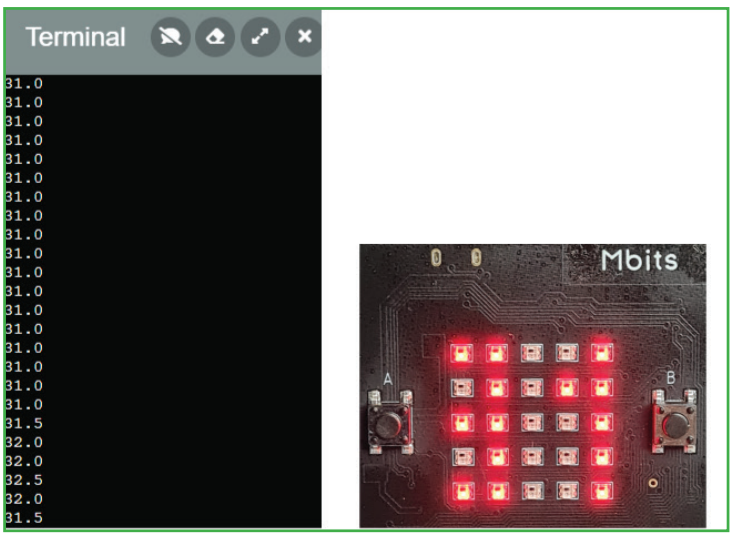
{ ตัวอย่างที่ 12 แสดงค่าอุณหภูมิที่ Terminals }



โปรแกรมที่ 12-1

ตัวอย่างนี้จะนำค่าของอุณหภูมิที่วัดได้มาแสดงที่ LED RGB 5×5 จุด และหน้าต่าง Terminal ทุกๆ 1 วินาที โดยค่าที่ Terminal จะแสดงในรูปเลขทศนิยมด้วย

สแกน QR Code เพื่อดูผลลัพธ์



รูปที่ 3-7 ผลลัพธ์แสดงที่หน้าต่าง Teminal และบนบอร์ด Mbits

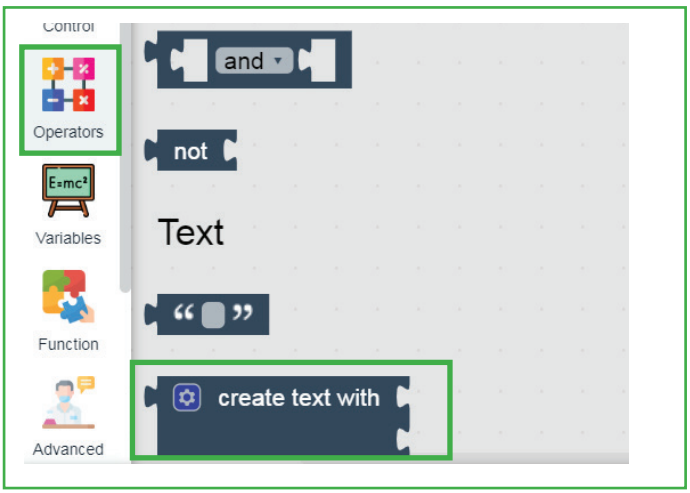
{ ตัวอย่างที่ 13 แสดงค่าอุณหภูมิพร้อมข้อความ (ภาษาไทย) }

ในหัวข้อ Operators จะมีชุดบล็อกคำสั่งเกี่ยวกับการจัดการข้อความชื่อ create text with ใช้เพื่อเชื่อมข้อความเข้าด้วยกัน

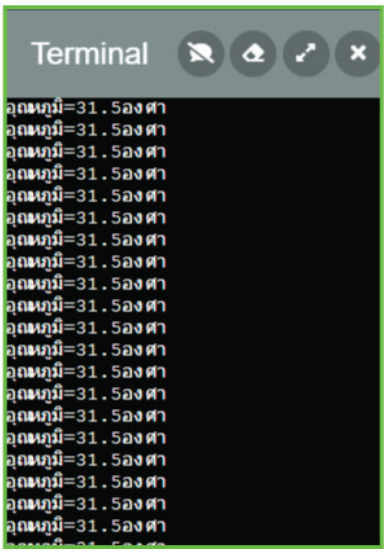
โดยตัวอย่างนี้จะเชื่อมข้อความ “อุณหภูมิ=” + temperature + “องศา” นำไปแสดงที่ Terminal

คำสั่ง create text with ค่าเริ่มต้นจะรวมข้อความได้ 2 ชุด (item) ถ้าต้องการเพิ่มต้องกดที่รูปเฟืองแล้วลาก item มาใส่เพิ่ม ในที่นี้ใช้ 3 ชุดข้อมูล

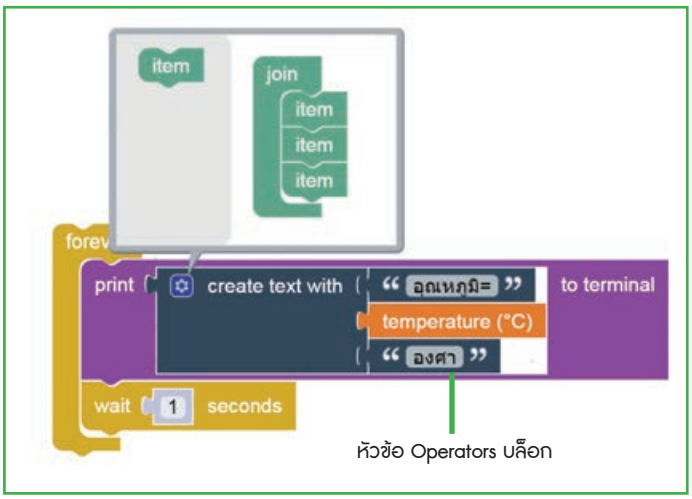
สแกน QR Code เพื่อดูผลลัพธ์



รูปที่ 3-8



รูปที่ 3-9 ผลลัพธ์แสดงที่หน้าต่าง Teminal

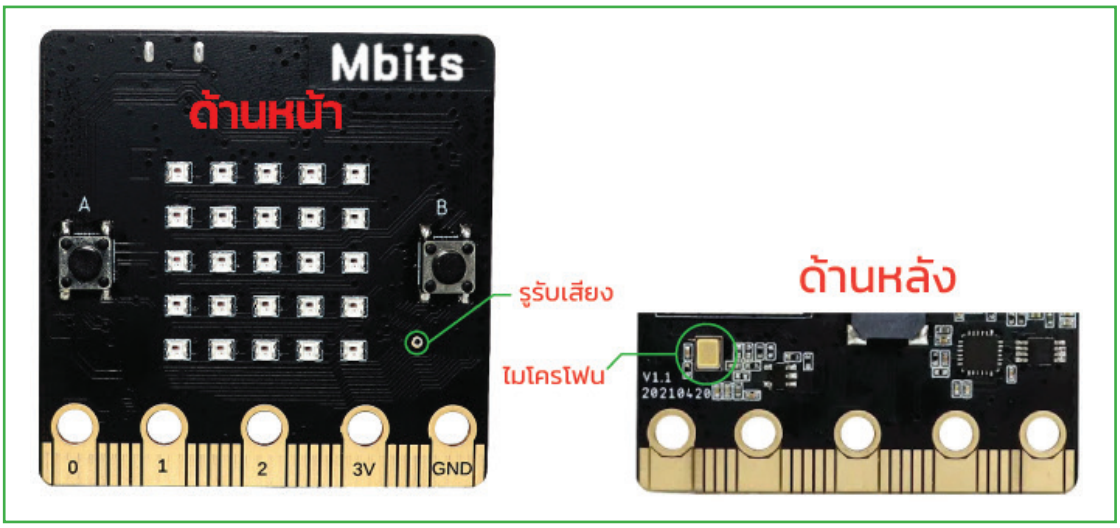


โปรแกรมที่ 13-1

บทที่ 4 ไมโครโฟนและตัวตรวจวัดความเร่ง

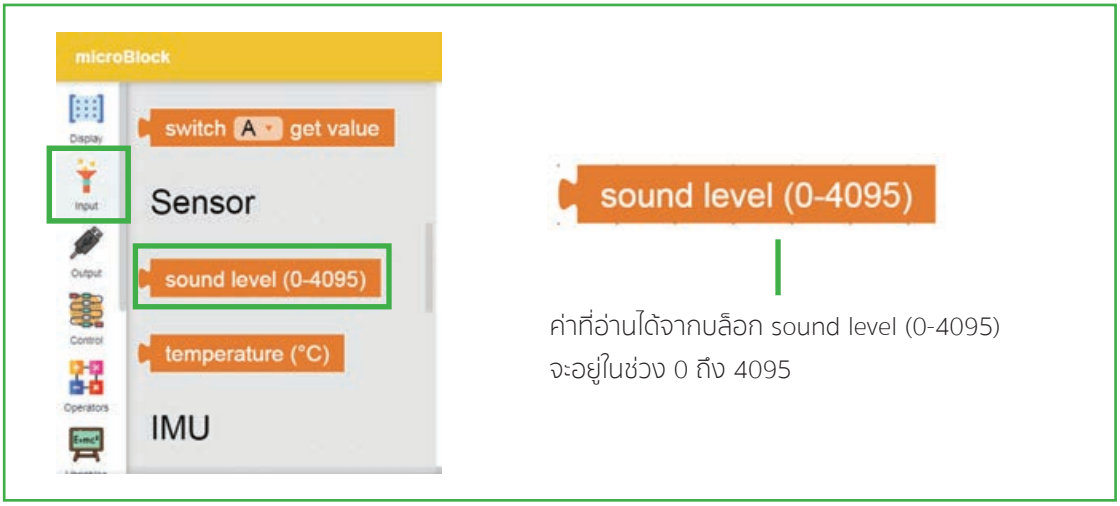
มาทำความรู้จักและใช้งานอุปกรณ์อินพุตสำคัญของบอร์ด Mbits
อีก 2 ตัวคือ ไมโครโฟนและตัวตรวจวัดความเร่ง

4.1 ไมโครโฟน



รูปที่ 4-1 ภาพแสดงตำแหน่งไมโครโฟนภาพด้านหน้าและด้านหลังของบอร์ด Mbits V2

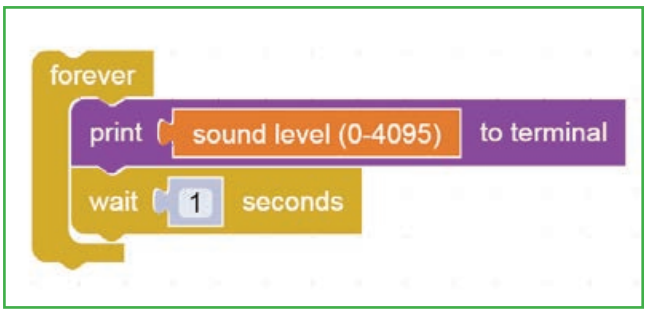
ไมโครโฟนทำหน้าที่เปลี่ยนสัญญาณเสียงให้เป็นสัญญาณไฟฟ้าแล้วป้อนเข้าไปยังวงจรแปลงสัญญาณแอนะล็อกเป็นดิจิทัลของ Mbits



รูปที่ 4-2 บล็อกคำสั่ง sound level (0-4095)

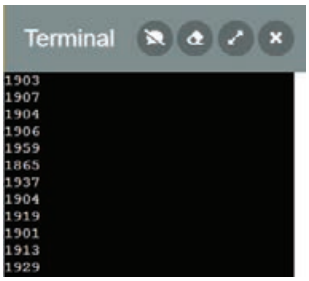
{ ตัวอย่างที่ 14 อ่านค่าจากไมโครโฟน แสดงผลที่หน้าต่าง Terminal }

ค่าเสียงในสภาวะปกติกับเมื่อได้รับการกระตุ้นจะมีค่าต่างกัน ทดสอบดูการเปลี่ยนแปลงของตัวเลขที่หน้าต่าง Terminal นำค่าเหล่านี้เป็นค่าสำหรับการเขียนโปรแกรมเปรียบเทียบในตัวอย่างต่อไป



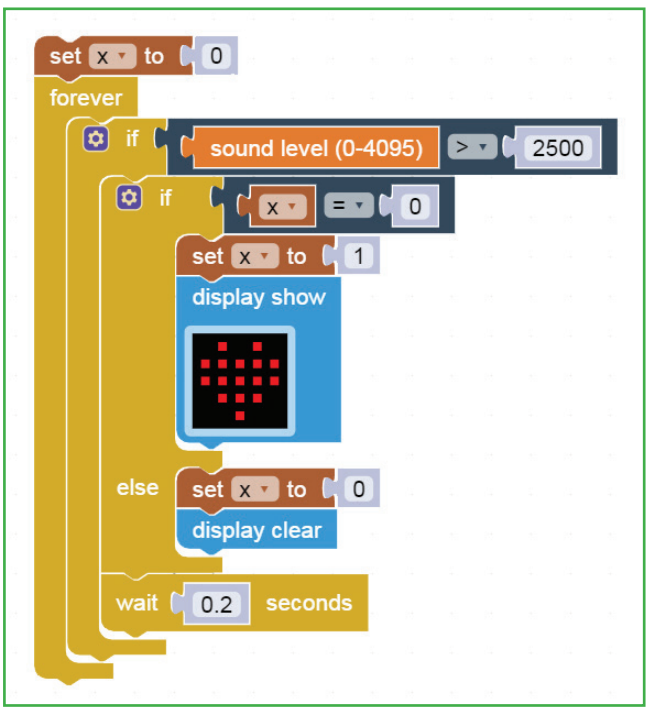
โปรแกรมที่ 14-1

สแกน QR Code เพื่อดูผลลัพธ์



รูปที่ 4-3 ผลลัพธ์แสดงที่หน้าต่าง Teminal

{ ตัวอย่างที่ 15 เปิด/ปิดไฟด้วยเสียง }



โปรแกรมที่ 15-1

สแกน QR Code เพื่อดูผลลัพธ์

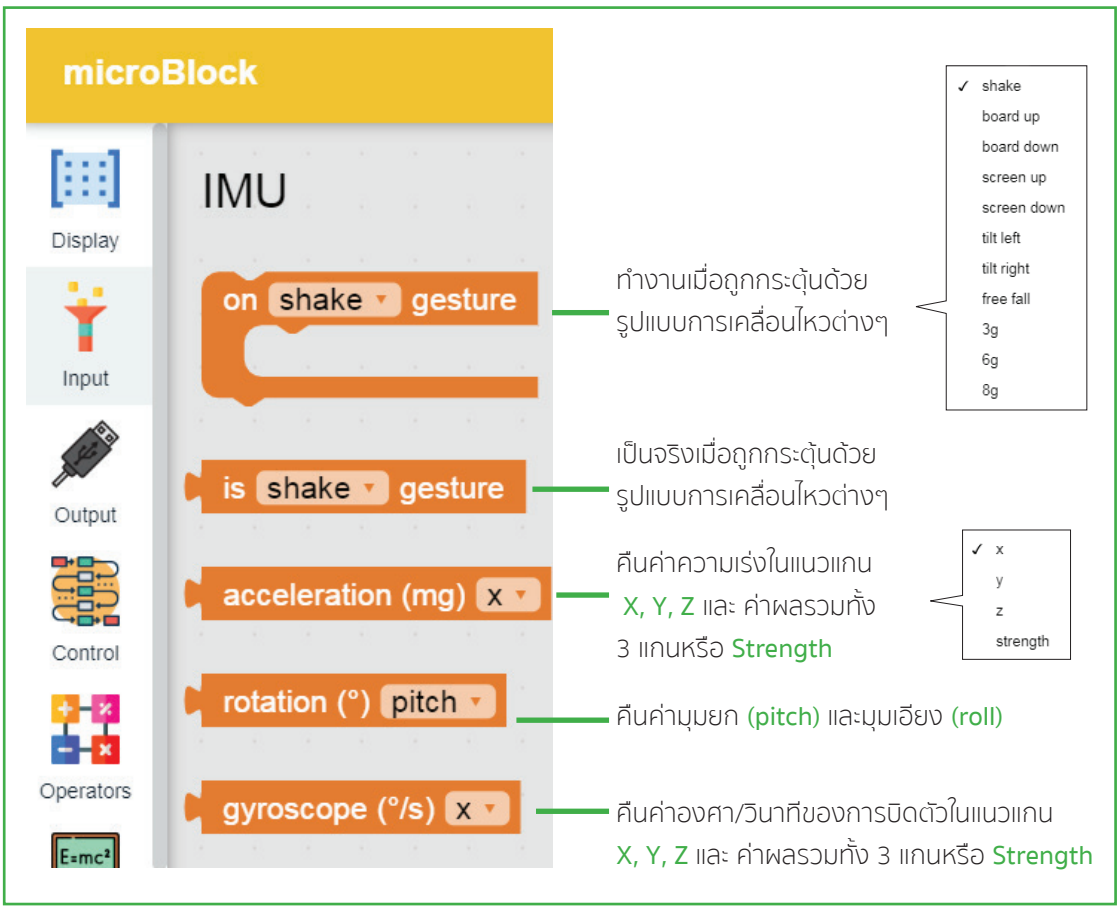
จากระดับเสียงที่อ่านค่าได้จากตัวอย่างก่อนหน้านี้พบว่า ค่าที่มากกว่าปกติคือค่าที่เกิน 2500 ขึ้นไป ดังนั้นจึงใช้ค่า 2500 เป็นจุดเปรียบเทียบ แต่ก่อนจะขับ LED RGB 5×5 จุดติดหรือดับจะต้องตรวจสอบก่อนว่า สถานะเดิมก่อนหน้านี้ (ตัวแปร X) มีค่าเป็นอย่างไร

ถ้ามีค่าเป็น 0 แสดงว่าเดิมดับอยู่ ก็สั่งให้ติด (เป็นรูปหัวใจ) และให้ค่าเปลี่ยนเป็น 1

ถ้าเดิมเป็น 1 แสดงว่าเดิมติดอยู่ก็สั่งให้ดับ และให้ค่า X เปลี่ยนเป็น 0 ท้ายสุดจะต้องหน่วงเวลาเล็กน้อยไม่ให้เกิดเหตุการณ์ซ้ำเร็วเกินไป

4.2 ตัวตรวจวัดความเร่ง Accelerometer

บนบอร์ด Mbits ติดตั้งตัวตรวจจับเบอร์ **MPU6050** เป็นตัวตรวจวัดความเร่ง 3 แกน นำค่าความเร่งมาประยุกต์ใช้งานได้หลากหลายตามชุดคำสั่งในหัวข้อ Input ดังนี้



รูปที่ 4-4 บล็อกคำสั่งหัวข้อ input>IMU

{ ตัวอย่างที่ 16 เขย่าสุ่มค่าตัวเลข 1 ถึง 9 }

เพื่อให้แสดงตัวเลขได้เต็มจอ จะใช้คำสั่ง display show 1-chars ทุกครั้งที่มีการเขย่า จะเป็นการกระตุ้นให้สุ่มตัวเลข แล้วนำมาแสดงที่ LED RGB 5×5 จุด

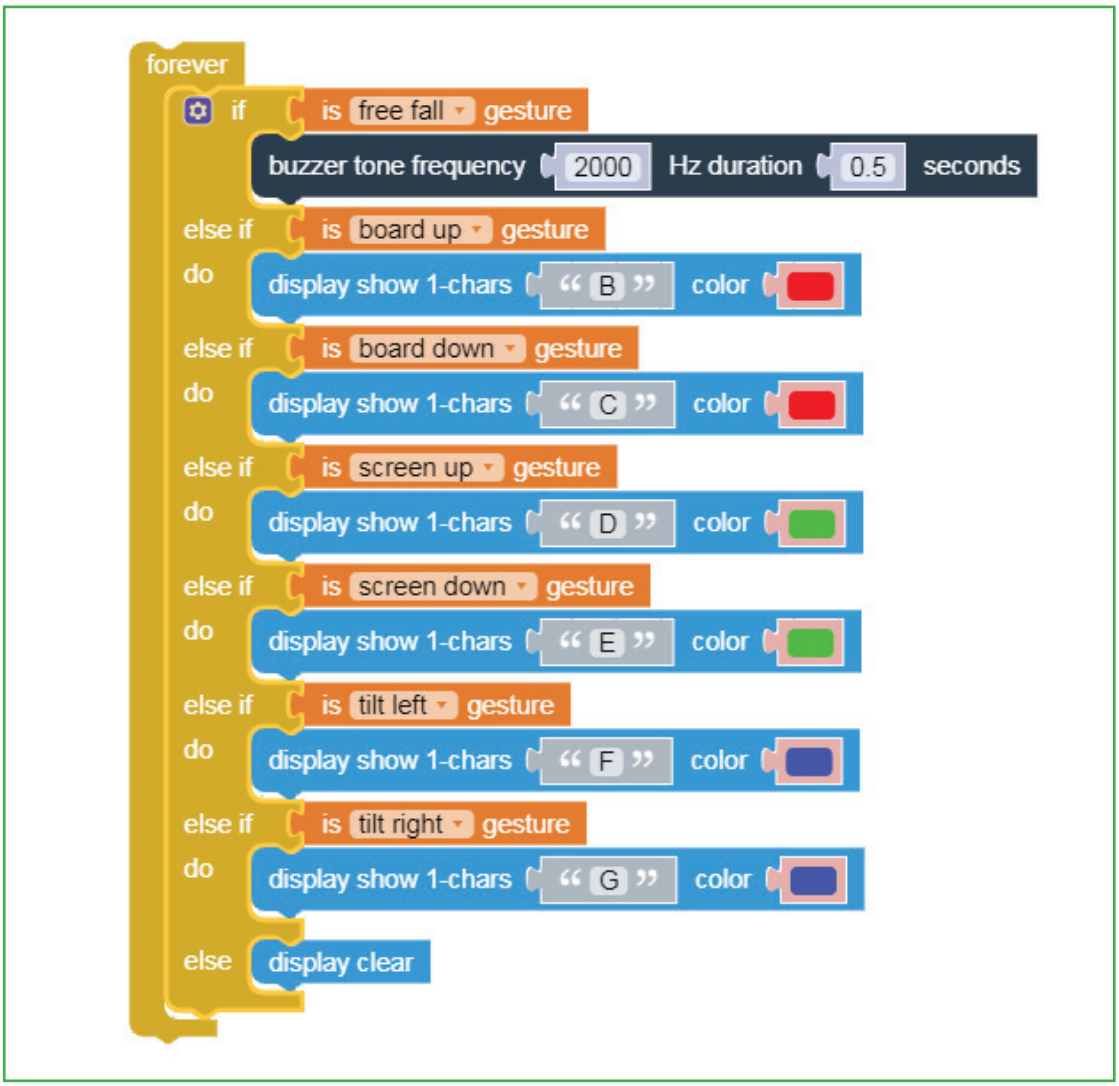
โปรแกรมที่ 16-1

สแกน QR Code เพื่อดูผลลัพธ์

{ ตัวอย่างที่ 17 กระตุ้นให้ทำงานด้วย Gesture ต่างๆ }

การกระตุ้นมีหลายรูปแบบตามรูป นำค่าเหล่านี้เขียนเป็นโค้ดเพื่อให้เห็นภาพ

รูปที่ 4-5



โปรแกรมที่ 17-1

การทำงาน

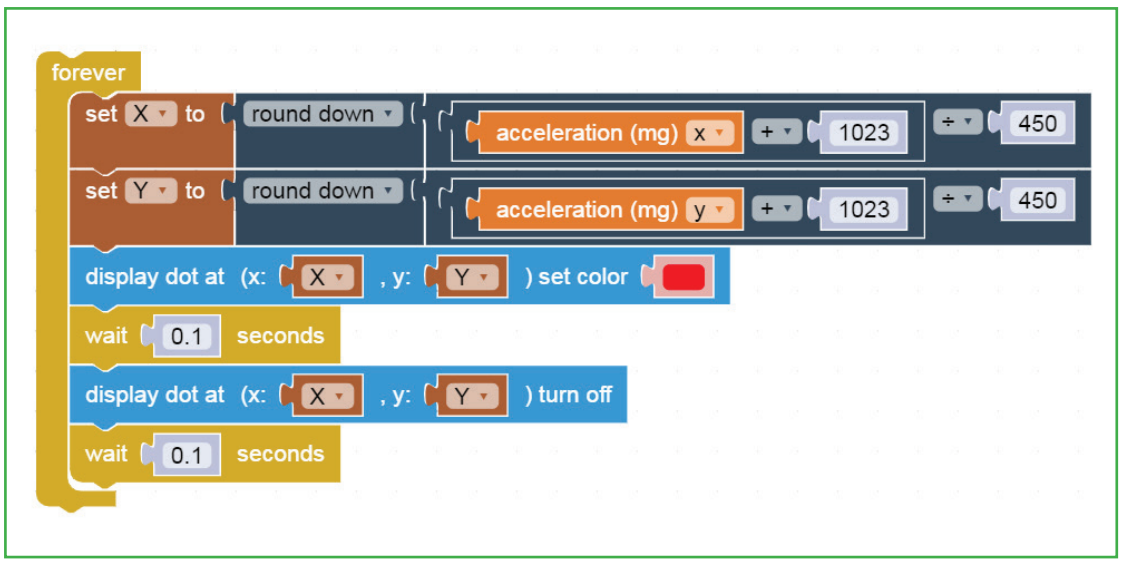
จากตัวอย่างจะตรวจสอบการหมุน การเอียงในทิศทางต่างๆ และแสดงผลออกมาเป็นตัวอักษร ยกเว้น **free fall** หรือการตกอิสระจะใช้เป็นการขับเสียงแทน เนื่องจากดูผลที่หน้าจอ LED RGB 5×5 จุด ได้ยาก

สแกน QR Code เพื่อดูผลลัพธ์

{ ตัวอย่างที่ 18 เอียงจุดไล่ติดกะพริบ }

ค่าที่อ่านได้จากบล็อก acceleration จะมีค่าอยู่ระหว่าง -1023 ถึง 1023 นำมาทำให้ค่าเป็นบวก โดยบวกกับ 1023 กลายเป็น 0 ถึง 2046 และแปลงไปใช้กับตำแหน่ง LED RGB 5×5 จุด ต้องแทนด้วยเลข 0 ถึง 4 จึงต้องหารด้วย 450 แต่ค่าการหารเป็นเลขทศนิยม ใช้คำสั่ง round down (ปัดเศษลง) ให้กลายเป็นเลขจำนวนเต็ม นำค่าไปพล็อตเป็นตำแหน่งของ LED RGB แกน X และ Y สุดท้ายกำหนดให้กะพริบด้วย จึงต้องสั่งให้ติดและดับที่ LED ในตำแหน่งเดิม

สแกน QR Code เพื่อดูผลลัพธ์



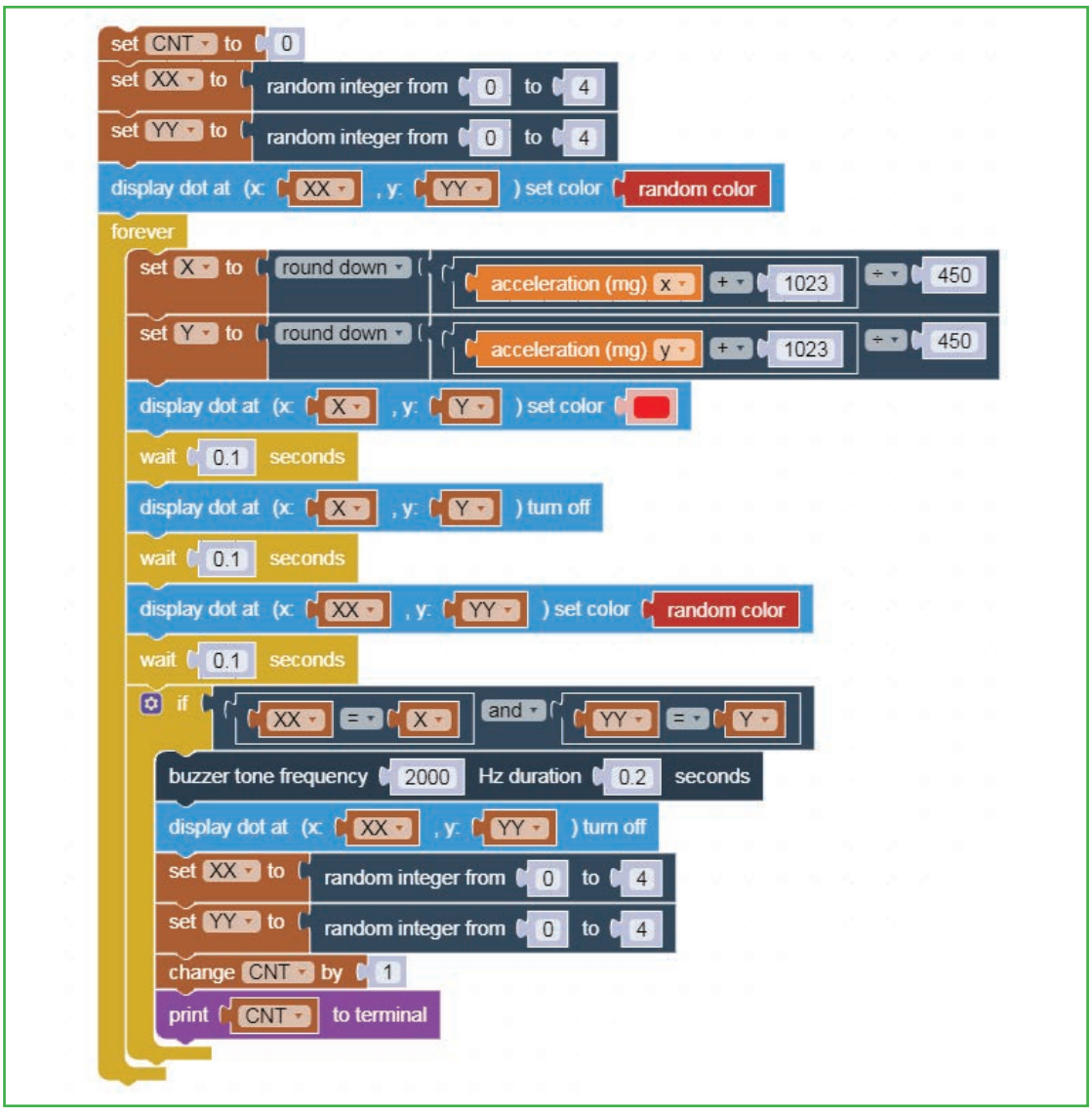
โปรแกรมที่ 18-1

{ ตัวอย่างที่ 19 เกมเอียงบอร์ดดับไฟ }

สแกน QR Code เพื่อดูผลลัพธ์

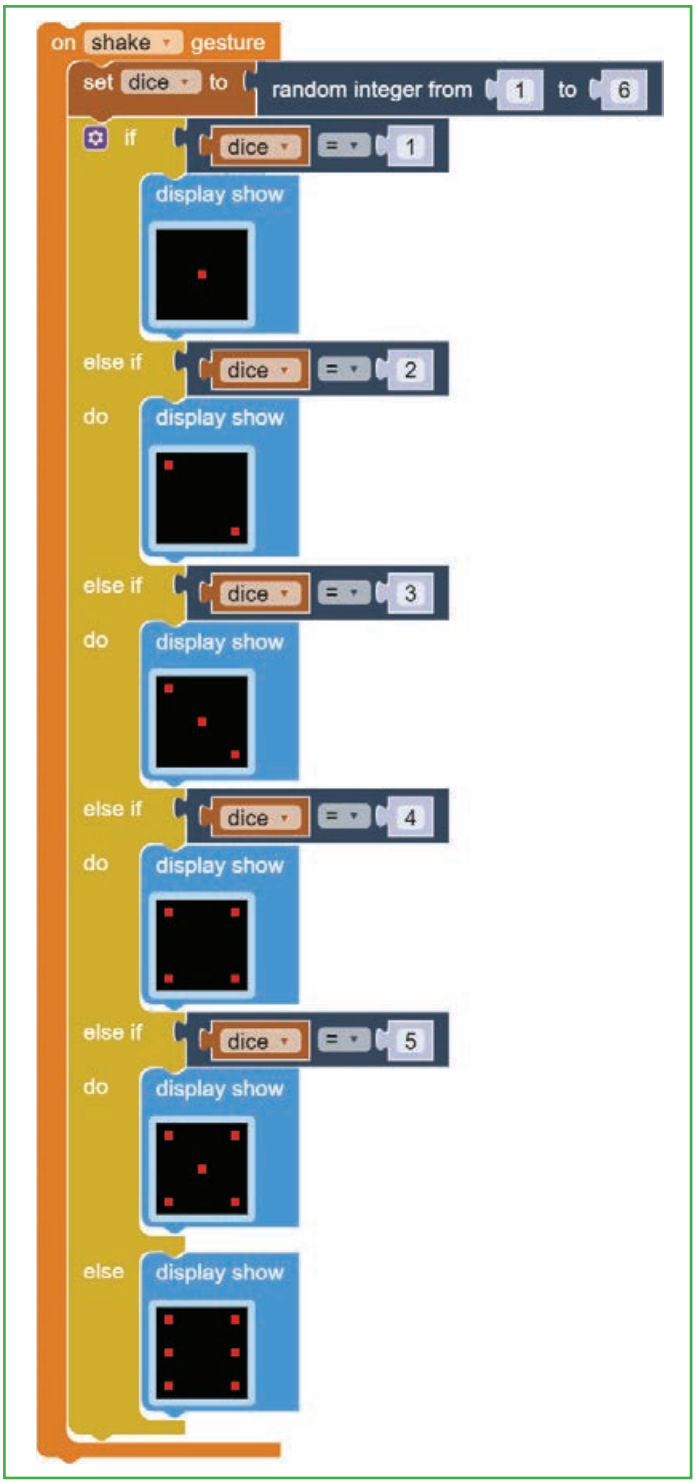
จากตัวอย่างก่อนหน้านี้ ดัดแปลงให้เป็นเกมง่ายๆ โดยเพิ่มตัวแปร XX และ YY สุ่มตำแหน่ง LED เป้าหมายติด เพื่อแยกแยะความต่าง LED เป้าหมายจะติดค้างและสุ่มสี ส่วน LED ค้นหาจะติดกะพริบสีแดง

ในการเลื่อนไปหาจะใช้การเอียงบอร์ด เมื่อตำแหน่งทั้งแกน X และ Y ตรงกับ LED เป้าหมาย ให้ขับเสียงออกลำโพงแล้วสุ่มตำแหน่งใหม่ เมื่อค้นหา LED เป้าหมายได้ 1 จุด จะเพิ่ม 1 คะแนน (ตัวแปร CNT) จากนั้นแสดงคะแนนที่หน้าต่าง Terminal



โปรแกรมที่ 19-1

{ ตัวอย่างที่ 20 เกมทอยลูกเต๋า }



โปรแกรมที่ 20-1

สแกน QR Code เพื่อดูผลลัพธ์

บอร์ด Mbits ทำการสุ่มตัวเลขจำนวนเต็มตั้งแต่ 1 ถึง 6 ด้วยคำสั่ง random integer from แล้วนำค่าของตัวเลขที่สุ่มได้ไปเก็บไว้ในตัวแปร dice

จากนั้น เมื่อโค้ดรันมาถึง if else if ก็จะตรวจสอบค่าตัวเลขในตัวแปร dice ว่าตรงกับตัวเลข 1,2,3,4 และ 5 หรือไม่ หากตรงกับตัวเลขใด LED ก็จะสว่างตามแบบที่กำหนดไว้

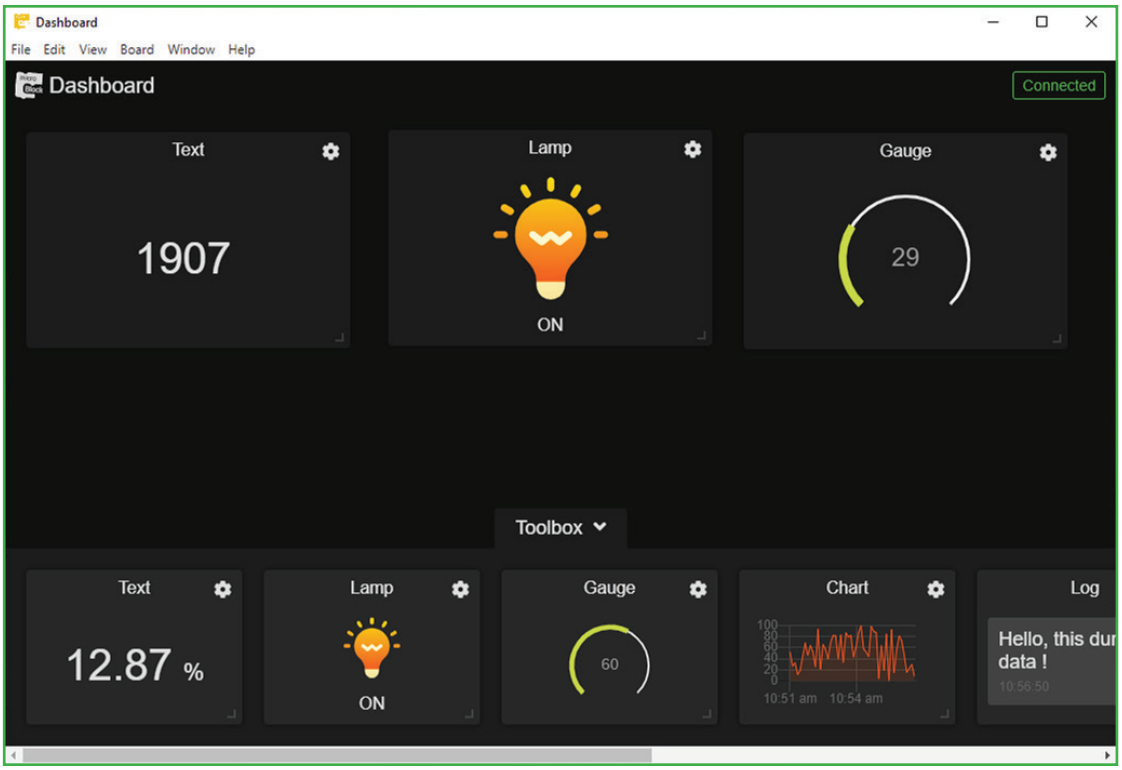
แต่หากไม่ตรงกับตัวเลขใดเลย แสดงว่าตัวเลขในตัวแปร dice มีค่าเท่ากับ 6 ก็จะทำงานตามโค้ดที่อยู่ในช่อง else แทน คือ แสดงลูกเต๋าหน้าเลข 6

บทที่ 5 การใช้งาน Dashboard ของ microBlock IDE

โปรแกรม microBlock IDE ตั้งแต่เวอร์ชัน 2.1.0 ขึ้นไป จะมีหน้าต่างแสดงผลการทำงานที่เรียกว่า Dashboard สำหรับแสดงข้อมูลแบบข้อความ ตัวเลข สถานะของ LED เกจหรือมาตรวัด กราฟ และปริมาณทางไฟฟ้าอื่นๆ ได้ สำหรับบอร์ด Mbits V2 ติดต่อกับส่วนแสดงผลหรือ Dashboard นี้ด้วยการสื่อสารข้อมูลผ่านอนุกรมผ่านทางพอร์ต USB อนุกรม โดยในระหว่างการทำงานร่วมกันจะต้องต่อสายเชื่อมต่อพอร์ต USB ของคอมพิวเตอร์กับบอร์ด Mbits V2 ตลอดเวลา

5.1 รู้จักกับหน้า Dashboard

Dashboard เป็นหน้าต่างสำหรับแสดงผลการทำงานในแบบกราฟิก ซึ่งเป็นที่นิยมมากในซอฟต์แวร์พัฒนาโปรแกรมสมัยใหม่ ซึ่งแตกต่างจากหน้าต่าง Terminal ที่แสดงข้อความหรือตัวเลขเท่านั้น ใน microBlock IDE ได้บรรจุความสามารถนี้เข้าไปในโปรแกรมด้วย ซึ่งจะแสดงผ่านเครื่องมือที่มีชื่อว่า วิดเจ็ต



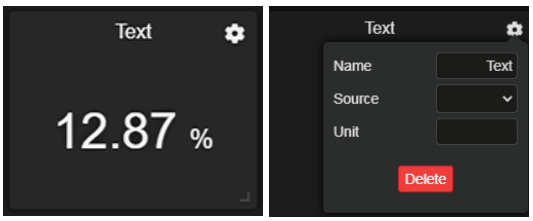
รูปที่ 5-1 แสดงตัวอย่างของ Dashboard ในโปรแกรม microBlock IDE

5.2 รู้จักกับวิดเจ็ตต่างๆ ของ Dashboard ในโปรแกรม microBlock IDE

ภายในแถบเครื่องมือหรือ Toolbox มีวิดเจ็ตสำหรับใช้งานบน Dashboard ที่แตกต่างกัน สรุปได้ดังนี้

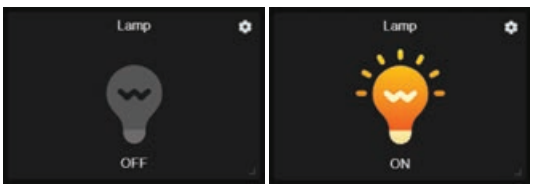
5.2.1 Text

เป็นวิดเจ็ตสำหรับแสดงข้อมูลแบบข้อความ ทั้งตัวอักษรและตัวเลข



Name : ชื่อของวิดเจ็ต
Source : ชื่อข้อมูล
Unit : หน่วยของค่าที่แสดง

5.2.2 Lamp



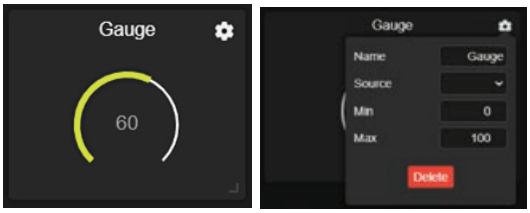
เป็นวิดเจ็ตใช้แสดงโคมไฟหรือหลอดไฟเหมาะกับการใช้งานข้อมูลที่เป็นดิจิทัล

- หากข้อมูลเป็น “0” หลอดไฟจะดับ
- หากข้อมูลเป็น “1” หลอดไฟจะติด

Name : ชื่อของวิดเจ็ต
Source : ชื่อข้อมูล

5.2.3 Gauge

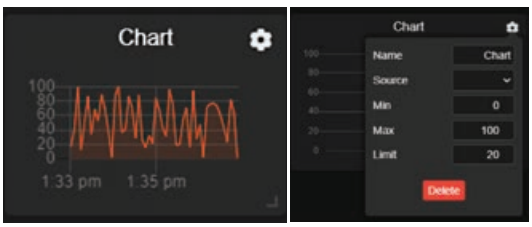
เป็นวิดเจ็ตใช้แสดงผลการทำงานในแบบมาตรวัดหรือเกจ



Name : ชื่อของวิดเจ็ต
Source : ชื่อข้อมูล
Min : ค่าต่ำสุดที่มาตรวัดหรือเกจแสดงได้
Max : ค่าสูงสุดที่มาตรวัดหรือเกจแสดงได้

5.2.4 Chart

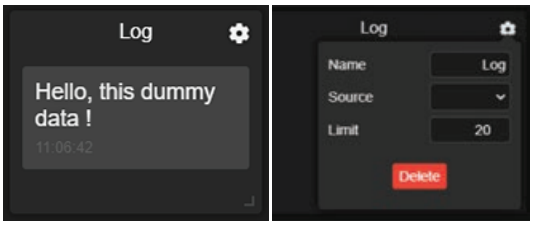
เป็นวิดเจ็ตใช้แสดงข้อมูลรูปกราฟที่บอกทั้งค่าของข้อมูลและเวลา



Name : ชื่อของวิดเจ็ต
Source : ชื่อข้อมูล
Min : ค่าต่ำสุดที่กราฟแสดงได้
Max : ค่าสูงสุดที่กราฟแสดงได้
Limit : จำนวนข้อมูลสูงสุดที่นำมาแสดงบนกราฟ

5.2.5 Log

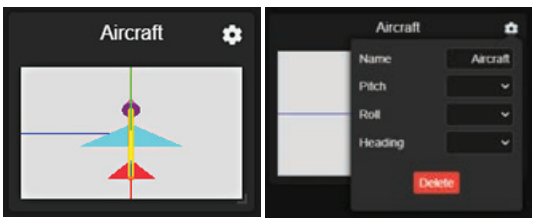
เป็นวิดเจ็ตแสดงชุดข้อมูลที่ประกอบไปด้วยข้อมูลและเวลา



Name : ชื่อของวิดเจ็ต
Source : ชื่อข้อมูล
Limit : จำนวนข้อมูลสูงสุดที่นำมาแสดง

5.2.6 Aircraft

เป็นวิดเจ็ตแสดงชุดข้อมูลในรูปแบบเครื่องบินจำลอง มีใน microBlock IDE V2.2.0 ขึ้นไป โดยปกติทำงานร่วมกับตัวตรวจวัดความเร่ง 3 แกน



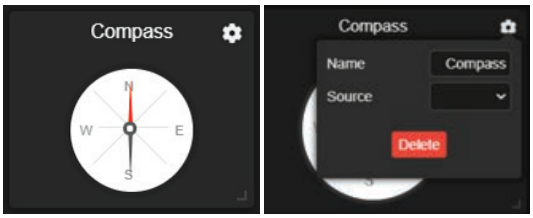
Name : ชื่อของวิดเจ็ต
Pitch : กำหนดค่าข้อมูลของมุม pitch
Roll : กำหนดค่าข้อมูลของมุม roll
Heading : กำหนดค่าข้อมูลของ heading

5.2.7 Game

เป็นมินิเกมส์ Pong ใช้ปุ่มลูกศรซ้ายและขวาบนคีย์บอร์ดในการควบคุม

5.2.8 Compass

เป็นวิดเจ็ตแสดงผลเข็มทิศ ปกติใช้งานร่วมกับโมดูลวัดสนามแม่เหล็กและเข็มทิสอิเล็กทรอนิกส์ (แนะนำเบอร์ LSM303)



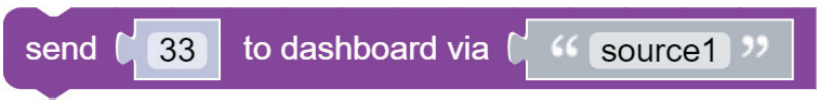
Name : ชื่อของวิดเจ็ต
Source : ชื่อข้อมูล

5.3 การใช้งาน Dashboard เบื้องต้น

การใช้งานหน้าต่าง Dashboard จะต้องสร้างโค้ดบนโปรแกรม microBlock IDE แล้วอัปโหลดไปยังบอร์ด Mbits V2 ก่อน เพื่อกำหนดชื่อและข้อมูลที่ต้องการ ส่งไปแสดงบนหน้า Dashboard ผ่านทางพอร์ต USB อนุกรม (USB serial port) โดยบล็อกคำสั่งที่ใช้ คือ บล็อกคำสั่ง send to dashboard

5.3.1 บล็อกคำสั่งสำหรับ Dashboard ของโปรแกรม microBlock IDE

บล็อกคำสั่งที่ใช้ในการส่งข้อมูลคือ send to dashboard อยู่ในแถบเครื่องมือ Advanced หัวข้อ Dashboard

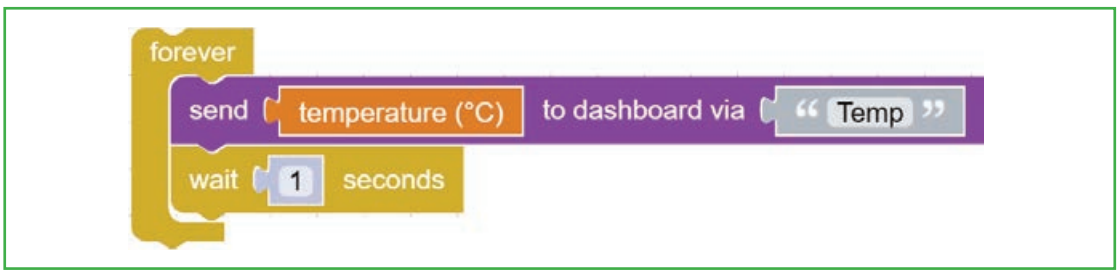


ใช้ส่งข้อมูลเพื่อนำไปแสดงผลบน Dashboard ประกอบด้วย

1. ข้อมูลที่ต้องการส่งเป็นตัวอักษรและตัวเลข โดยกำหนดในช่องที่อยู่ต่อจากคำสั่ง send
2. ชื่อข้อมูล เช่น ข้อมูลจากเซนเซอร์วัดอุณหภูมิ ให้ตั้งชื่อว่า “Temp” หรือ “Temperature_data” (ชื่อข้อมูลต้องไม่เว้นวรรคและไม่เป็นภาษาไทย) โดยกำหนดในช่อง “source1”

{ ตัวอย่างที่ 21 การแสดงค่าอุณหภูมิบนหน้า Dashboard }

หัวข้อนี้จะแสดงค่าข้อมูลที่อ่านได้จากตัวตรวจวัดอุณหภูมิบนบอร์ด Mbits ส่งไปแสดงผลเป็นตัวอักษรบนหน้าต่าง Dashboard มีขั้นตอนดังนี้

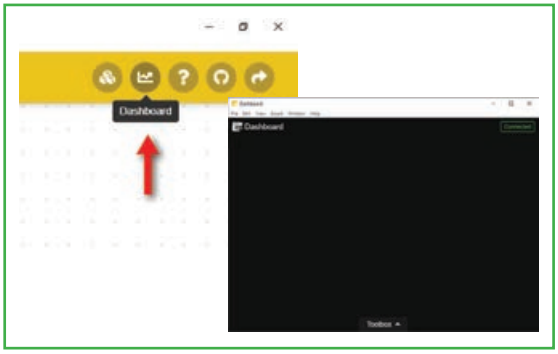


โปรแกรม 21-1

ทำการอัปโหลดโค้ดไปยังบอร์ด Mbits เท่านี้ก็เสร็จสิ้นสำหรับขั้นตอนสร้างโค้ดสำหรับบอร์ด Mbits

ต่อไปจะเป็นขั้นตอนการใช้งานและสร้างวิดเจ็ตบนหน้า Dashboard คลิกที่เมนู Dashboard ที่ด้านขวาบน เพื่อเปิดหน้าต่าง Dashboard

หน้าต่าง Dashboard ปรากฏขึ้นมาดังรูปที่ 5-2 ในการใช้งานครั้งแรกจะยังไม่มีวิดเจ็ต (widget) หรืออุปกรณ์ใดๆ หากเชื่อมต่อกับบอร์ด Mbits จะแสดงข้อความ Connected ที่มุมขวาบน (หากขึ้น Disconnect ให้ทำการเชื่อมต่อสายสัญญาณกับบอร์ด Mbits ก่อนใช้งานขั้นตอนถัดไป)



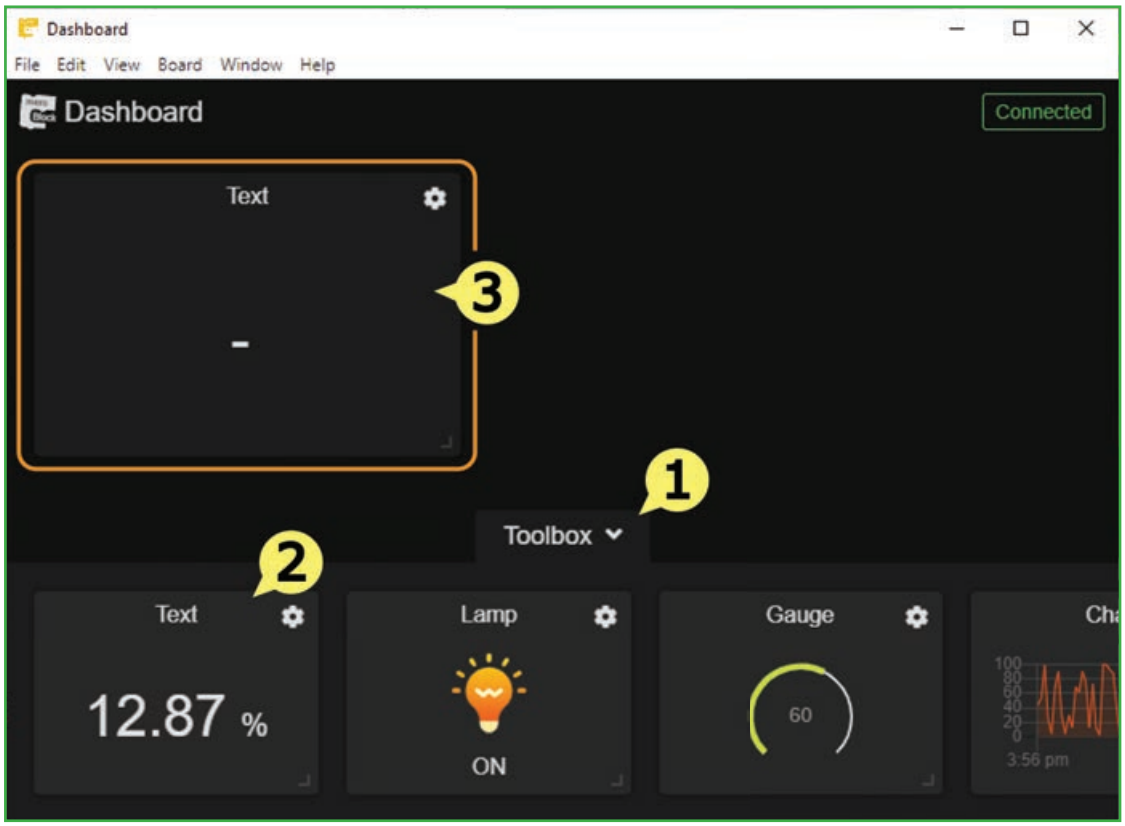
รูปที่ 5-2

5.3.2 เริ่มต้นเลือกและวางวิดเจ็ต

(1) คลิกที่ **Toolbox**

(2) คลิกเลือกวิดเจ็ตของเครื่องมือที่ต้องการ

(3) จากนั้นวิดเจ็ตจะปรากฏขึ้นมาบนพื้นที่แสดงผลในที่นี้เลือก **Text** ดังรูปที่ 5-3



รูปที่ 5-3 แสดงแนวทางการเลือกและวางวิดเจ็ตบน Dashboard ของ microBlock IDE

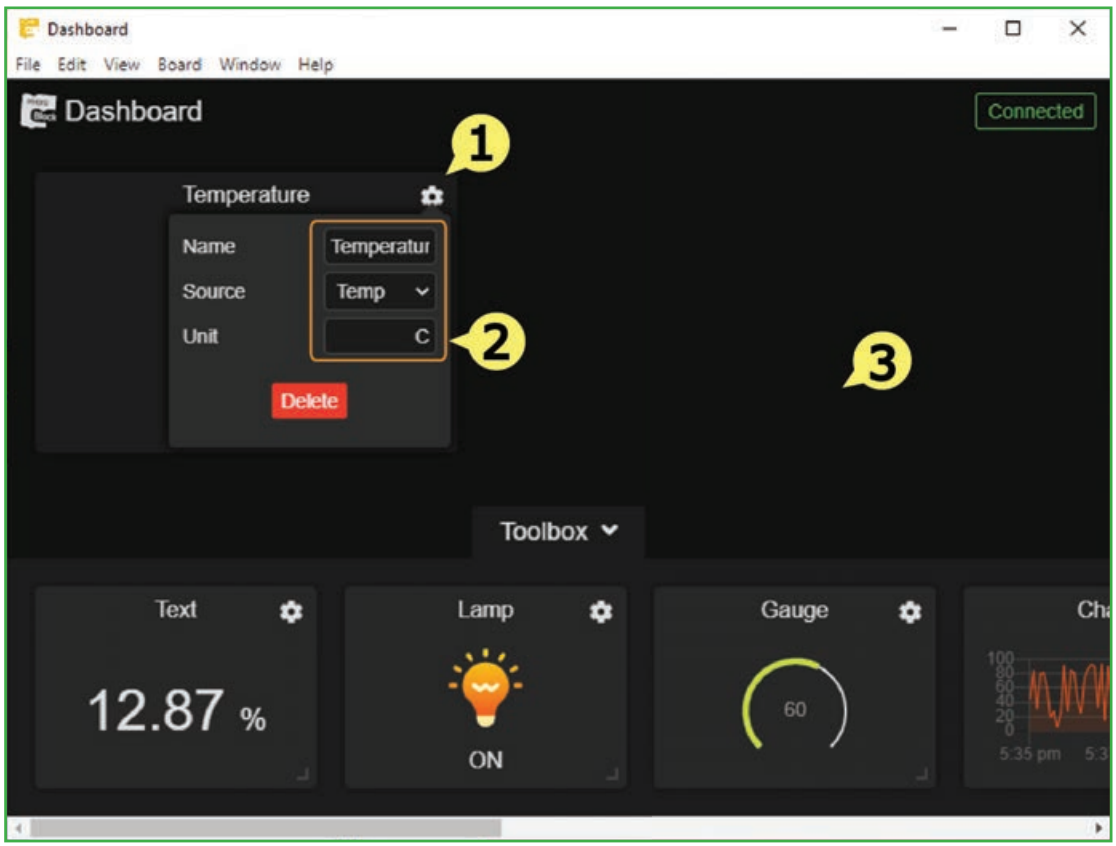
(4) การนำข้อมูลมาแสดง มีขั้นตอนดังรูปที่ 5-4

(4.1) คลิกที่ปุ่มรูปฟันเฟืองบริเวณด้านบนขวา

(4.2) หน้าต่างสำหรับตั้งค่าวิดเจ็ต **Text** ปรากฏขึ้นมา มีคุณสมบัติที่ต้องการตั้งค่าดังนี้

Name คือ ชื่อของวิดเจ็ต ผู้พัฒนาตั้งชื่ออะไรก็ได้ รองรับทั้งภาษาอังกฤษและภาษาไทย
Source คือ ชื่อข้อมูลที่ต้องการแสดงผลจากโปรแกรมที่เขียนไว้ก่อนหน้า
Unit คือ หน่วยของค่าที่แสดง เช่น องศาเซลเซียส (C)

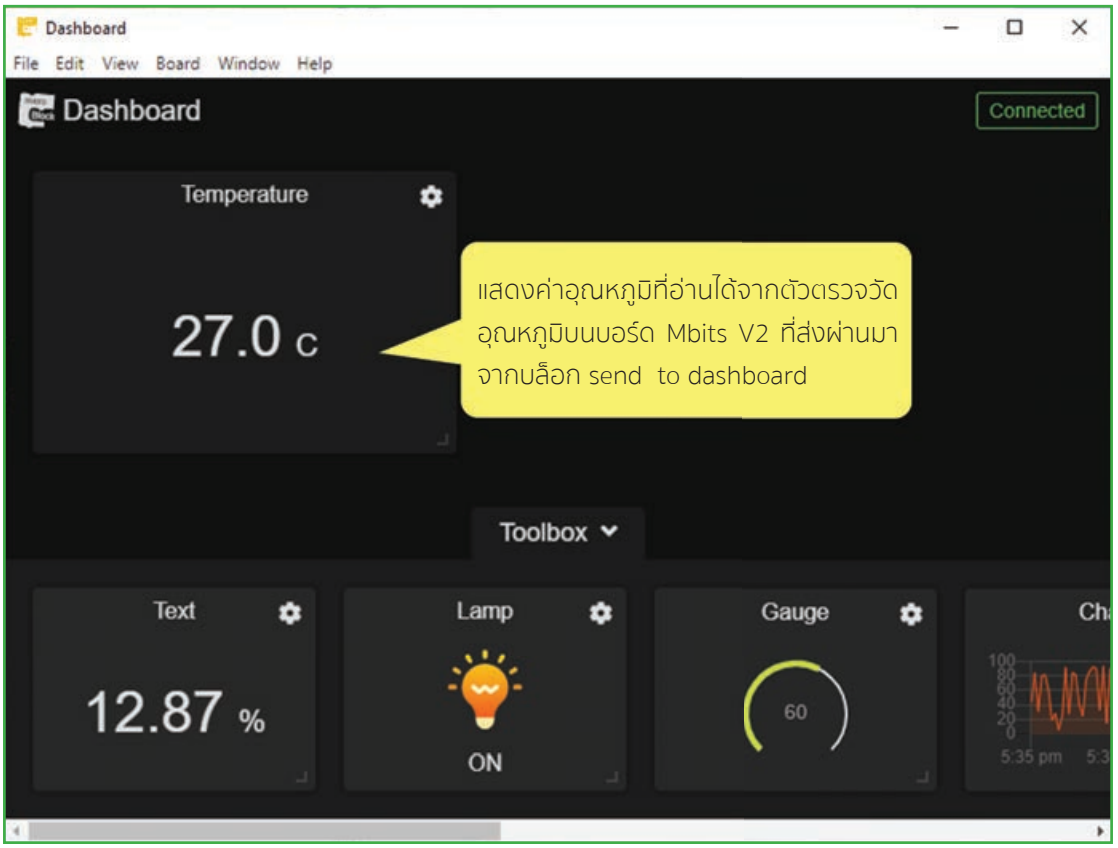
(4.3) คลิกไปที่พื้นที่ว่างของ Dashboard เพื่อปิดการตั้งค่าวิดเจ็ต



รูปที่ 5-4 แสดงขั้นตอนการนำข้อมูลมาแสดงบน Dashboard ของโปรแกรม microBlock IDE

ผลลัพธ์ที่ได้

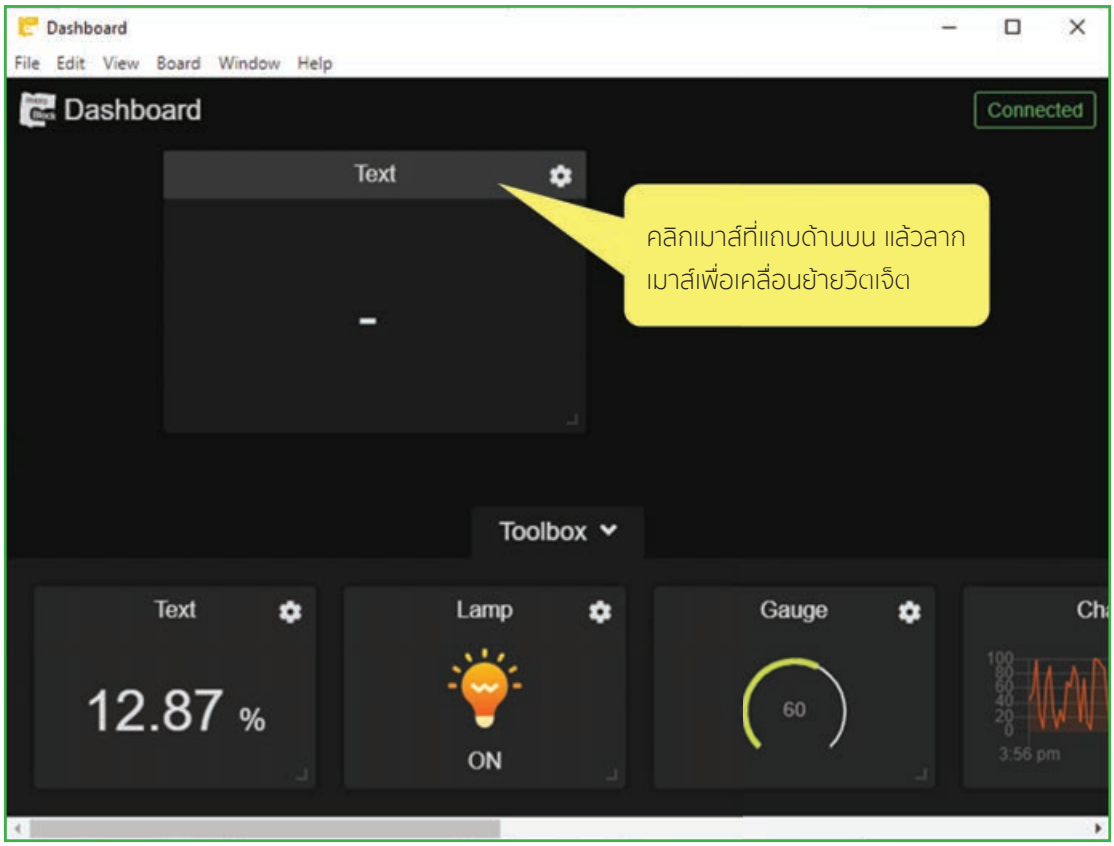
เมื่อทำการตั้งค่านำข้อมูลมาแสดงบนวิดเจ็ตสำเร็จ จะปรากฏค่าอุณหภูมิบนวิดเจ็ตที่ชื่อ Temperature โดยมีหน่วยเป็น C (องศาเซลเซียส)



รูปที่ 5-6 แสดง Dashboard สำหรับแสดงค่าอุณหภูมิที่อ่านได้จากบอร์ด Mbits V2

5.3.3 การย้ายตำแหน่งของวิดเจ็ต

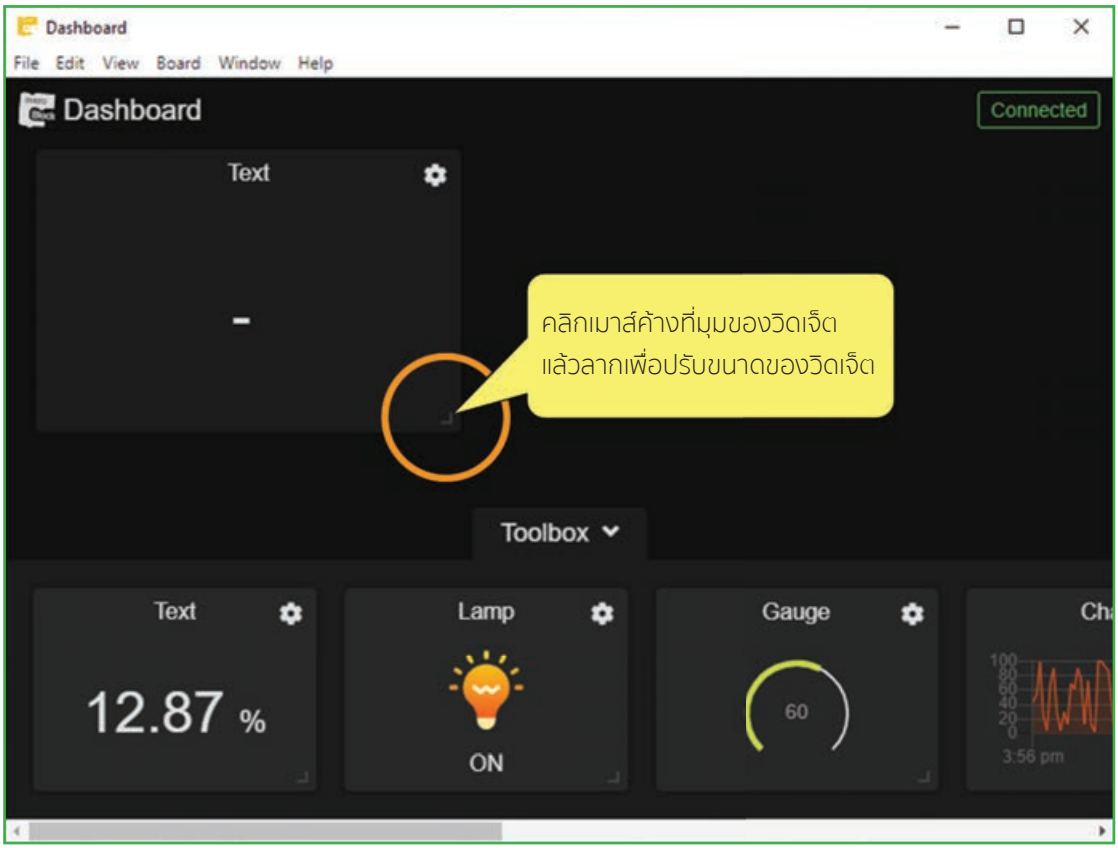
การย้ายตำแหน่งการวางของวิดเจ็ตหรืออุปกรณ์แสดงผลต่างๆ บน Dashboard ทำได้โดยการคลิกเมาส์ค้างที่แถบด้านบนของวิดเจ็ต จากนั้นทำการลากไปวางยังตำแหน่งที่ต้องการตามรูปที่ 5-7



รูปที่ 5-7 แสดงแนวทางการย้านตำแหน่งของวิดเจ็ตในหน้าต่าง Dashboard

5.3.4 การปรับขนาดของวิดเจ็ต

ผู้พัฒนาสามารถปรับขนาดของวิดเจ็ตบน Dashboard ได้ โดยคลิกเมาส์ค้างบริเวณมุมขวาล่างของวิดเจ็ต แล้วลากปรับขนาดตามความต้องการตามรูปที่ 5-8

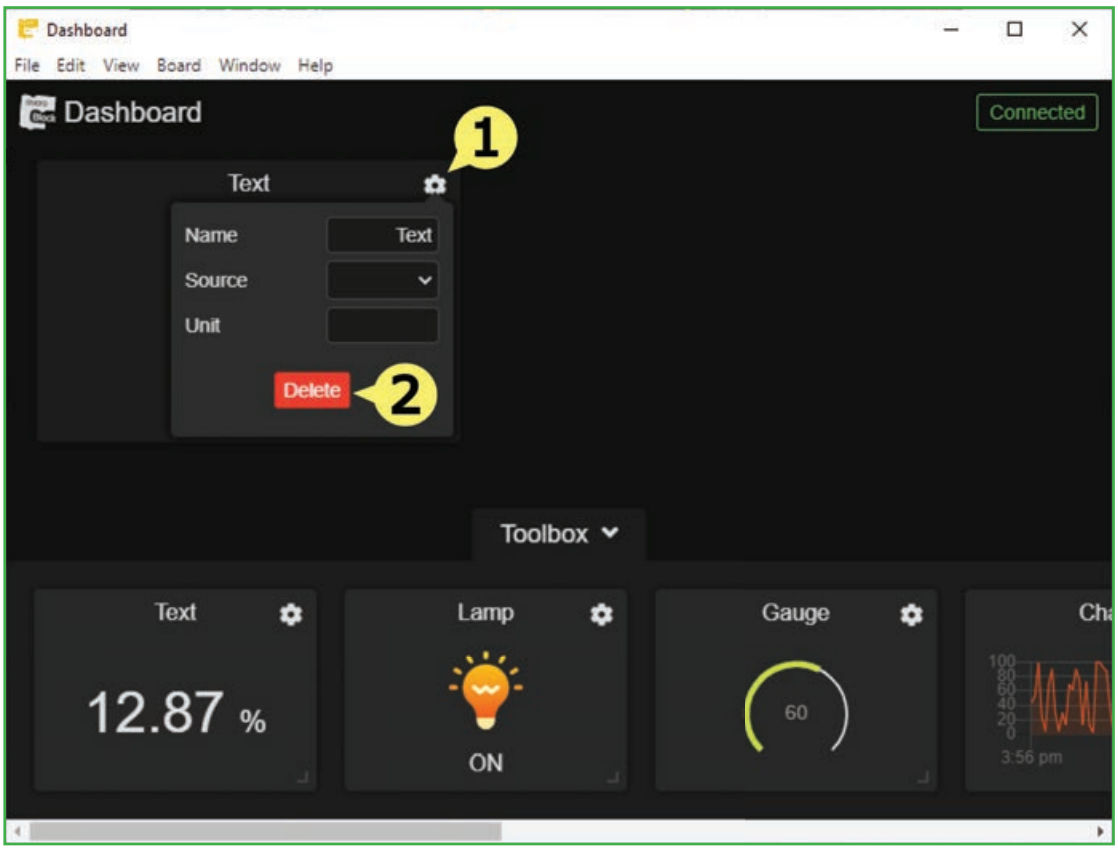


รูปที่ 5-8 แสดงแนวทางปรับขนาดวิดเจ็ตใน Dashboard ของโปรแกรม microBlock IDE

5.3.5 การลบวิตเจ็ต

แสดงขั้นตอนดังรูปที่ 5-9

(1) คลิกที่ปุ่มรูปฟันเฟืองบริเวณด้านบนขวาของวิดเจ็ตที่ต้องการลบ
(2) คลิกที่ปุ่ม **Delete** เพื่อลบ

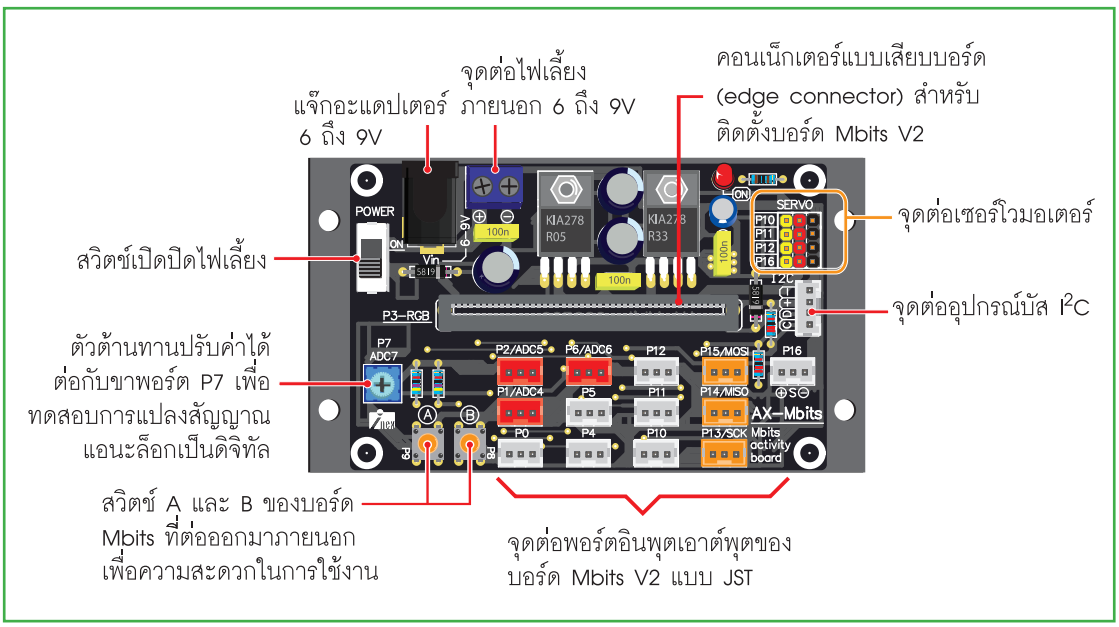


รูปที่ 5-9 แสดงการลบวิดเจ็ตในหน้าต่าง Dashboard ของโปรแกรม microBlock IDE

บทที่ 6

แนะนำ AX-Mbits บอร์ดเชื่อมต่อพอร์ตอินพุตเอาต์พุตกับอุปกรณ์ภายนอกสำหรับ Mbits V2 บอร์ดไมโครคอนโทรลเลอร์เพื่อการเรียนรู้วิทยาการคำนวณเชิงปฏิบัติการ

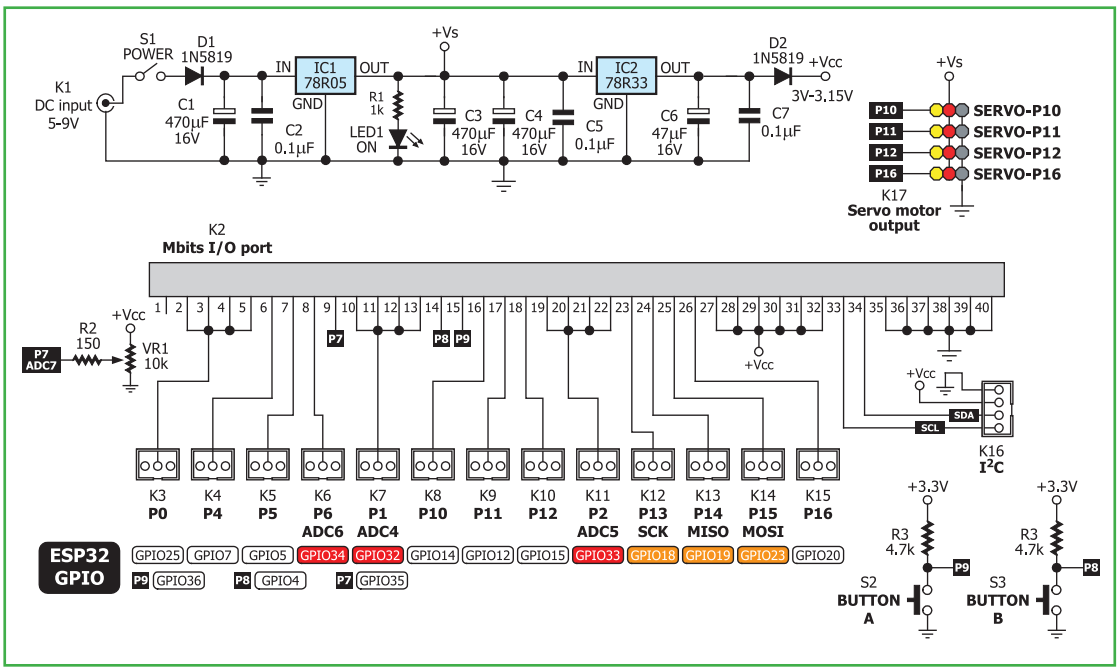
AX-Mbits เป็นหนึ่งในอุปกรณ์เสริมเพื่อสนับสนุนการใช้งานมินิบอร์ดไมโครคอนโทรลเลอร์เพื่อการเรียนรู้ Mbits V2 ที่พัฒนาและผลิตขึ้นในประเทศไทย โดยวิศวกรไทยจากบริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด (www.inex.co.th) โดยบอร์ดนี้ได้รับการออกแบบมาเพื่อช่วยอำนวยความสะดวกในการเรียนรู้และใช้งาน Mbits บอร์ดไมโครคอนโทรลเลอร์เพื่อการเรียนรู้วิทยาการคำนวณเชิงปฏิบัติการ โดยเฉพาะอย่างยิ่งกับการใช้งานพอร์ตอินพุตเอาต์พุตของบอร์ด Mbits



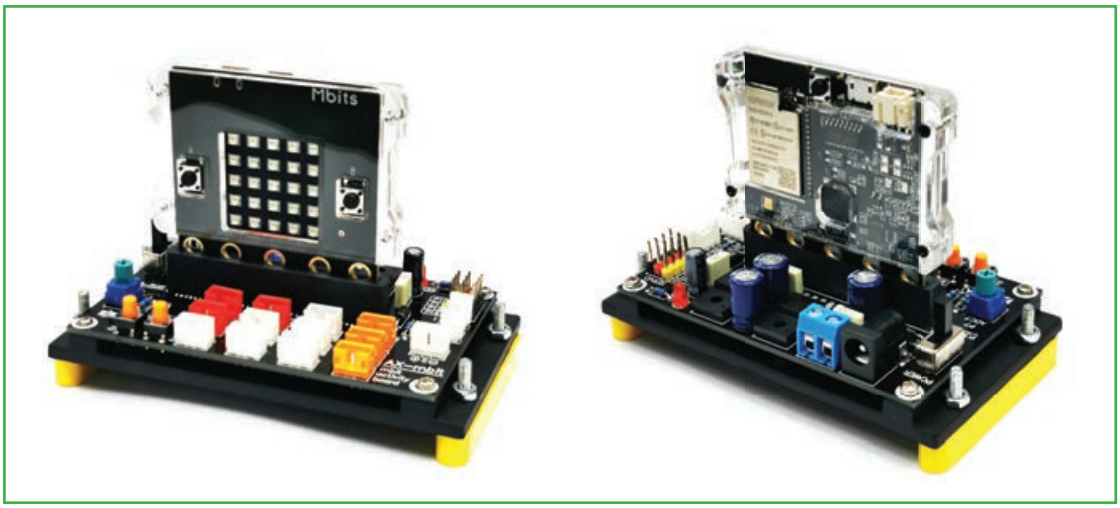
รูปที่ 6-1 แสดงส่วนประกอบของบอร์ด AX-Mbits

6.1 คุณสมบัติของ AX-Mbits

- มีคอนเน็กเตอร์แบบเสียบบอร์ดสำหรับติดตั้งบอร์ด Mbits
- มีจุดต่อไฟเลี้ยง 6 ถึง 9V เป็นแจ๊กอะแดปเตอร์และจุดต่อแบบขันสกรู พร้อมวงจรควบคุมไฟเลี้ยงคงที่ +5V และ +3.3V และวงจรป้องกันการต่อไฟกลับขั้ว
- มีจุดต่อพอร์ตสำคัญแบบ JST เพื่อต่อกับอุปกรณ์อินพุตเอาต์พุตของ INEX
- มีจุดต่ออุปกรณ์ระบบบัส I²C เพื่อต่อกับอุปกรณ์ภายนอกเพิ่มเติม
- มีจุดต่อเซอร์โวมอเตอร์ 4 ช่อง ใช้งานร่วมกับขาพอร์ต P10, P11, P12 และ P16 ของ Mbits V2 ใช้แรงดันจากไฟเลี้ยง +5V มาเลี้ยงเซอร์โวมอเตอร์ รองรับเซอร์โวมอเตอร์ขนาดเล็กที่ใช้ไฟเลี้ยง 4.8 ถึง 6V
- มีสวิตช์กดติดปล่อยดับ 2 ตัวต่อมาจากสวิตช์ A และ B ของ Mbits V2
- มีตัวต้านทานปรับค่าได้สำหรับทดลองอินพุตแอนะล็อก สำหรับปรับแรงดัน 0 ถึง +3V



รูปที่ 6-2 วงจรของบอร์ด AX-Mbits



รูปที่ 6-3 การต่อใช้งานของบอร์ด Mbits V2 กับบอร์ด AX-Mbits