เทคโนโลยี

(วิทยาการคำนวณ)

ชั้นมัธยมศึกษาปีที่ 3

หน่วยการเรียนรู้ที่ 1

การพัฒนาแอปพลิเคชัน

แผนผังหัวข้อหน่วยการเรียนรู้

การพัฒนาแอปพลิเคชัน

- ความหมายและลักษณะสำคัญของแอปพลิเคชัน
- ขั้นตอนการพัฒนาโปรแกรม
- เครื่องมือที่ใช้ในการพัฒนาแอปพลิเคชัน
- การพัฒนาแอปพลิเคชันบนสมาร์ตโฟนด้วย MIT App Inventor
- อินเทอร์เน็ตของสรรพสิ่ง (Internet of Things)
- แอปพลิเคชันกับ Internet of Things
- การพัฒนาระบบ IoT เบื้องต้น

ตัวชี้วัด

พัฒนาแอปพลิเคชันที่มีการบูรณาการกับวิชาอื่นอย่างสร้างสรรค์ (ว 4.2 ม.3/1)

ความหมายและลักษณะสำคัญของแอปพลิเคชัน

แอปพลิเคชัน — ชุดคำสั่งหรือโปรแกรมที่ถูกพัฒนาขึ้น

ใช้ในการควบคุมการของทำงานของระบบคอมพิวเตอร์

- ระบบคอมพิวเตอร์ในอุตสาหกรรมต่าง ๆ

- คอมพิวเตอร์ส่วนบุคคล

- สมาร์ตโฟน

- สมาร์ตวอตช์

เพื่อให้ระบบคอมพิวเตอร์ทำงานตามคำสั่งตามที่ผู้ใช้ต้องการ

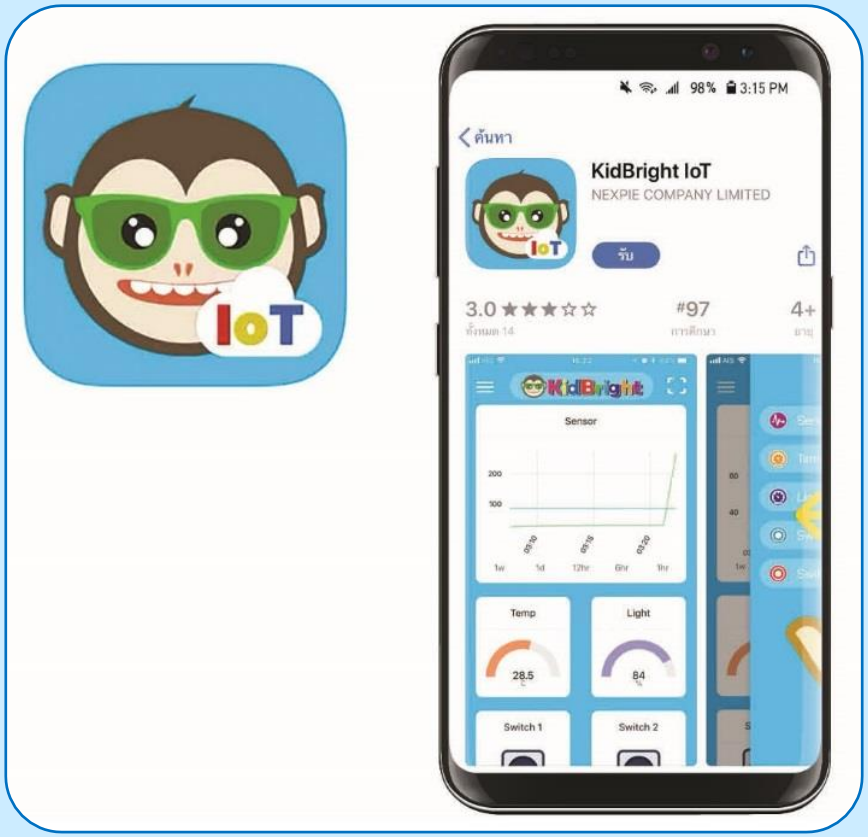
การจำแนกแอปพลิเคชันตามลักษณะการใช้งาน

- แอปพลิเคชันบนระบบคอมพิวเตอร์ส่วนบุคคล (Desktop Application)

  เป็นแอปพลิเคชันที่ทำงานบนระบบปฏิบัติการคอมพิวเตอร์ส่วนบุคคล หรืออาจจะเรียกว่า โปรแกรมคอมพิวเตอร์ก็ได้ เช่น โปรแกรมแปลงสกุลเงิน โปรแกรมผันเสียงวรรณยุกต์ โปรแกรมจัดการเอกสาร โปรแกรมนำเสนอ

- แอปพลิเคชันสำหรับระบบสมองกลฝังตัว (Microcontroller Application)

  เป็นแอปพลิเคชันที่ถูกพัฒนาเพื่อทำงานเฉพาะอย่างใดอย่างหนึ่งบนอุปกรณ์สมองกลฝังตัว เช่น ระบบรดน้ำต้นไม้อัตโนมัติ ระบบตรวจจับแก๊สรั่วภายในบ้าน ระบบเตือนภัยต่าง ๆ



- แอปพลิเคชันบนสมาร์ตโฟนหรือแท็บเล็ต (Mobile Application)

  เป็นแอปพลิเคชันที่ทำงานบนสมาร์ตโฟนหรือแท็บเล็ตเพื่ออำนวยความสะดวกหรือใช้งานในด้านต่าง ๆ ตามที่ต้องการ

- เว็บแอปพลิเคชัน (Web Application)

  เป็นแอปพลิเคชันที่ทำงานบนเว็บเบราว์เซอร์ สามารถเข้าถึงได้ง่าย ไม่ต้องแจกจ่ายหรือติดตั้งซอฟต์แวร์

ใช้งานในการเรียนการสอน

ใช้เก็บเอกสารต่าง ๆ

- แอปพลิเคชันบนอุปกรณ์พกพาอื่น ๆ

  เช่น สมาร์ตวอต์ช เป็นแอปพลิเคชันที่มีรูปแบบการทำงานเฉพาะบนอุปกรณ์ขนาดเล็กหรืออุปกรณ์พกพาอื่น ๆ เพื่ออำนวยความสะดวกอย่างง่าย

ขั้นตอนการพัฒนาโปรแกรม

ขั้นตอนที่ 1 การวิเคราะห์ปัญหา (Problem Analysis)

เป็นการวิเคราะห์ว่าปัญหาเป็นอย่างไร และหาวิธีการหรือระบบที่จะนำมาช่วยในการพัฒนาหรือแก้ไขปัญหาที่เกิดขึ้น

ขั้นตอนที่ 2 การออกแบบโปรแกรม (Program Design)

พูดคุยหาข้อสรุปเกี่ยวกับความต้องการของเจ้าของโครงการ เพื่อนำข้อมูลที่ได้มาออกแบบโปรแกรมหรือแอปพลิเคชันให้มีความสามารถตามที่เจ้าของโครงการต้องการ

ขั้นตอนที่ 3 การเขียนโปรแกรม (Program Coding)

นำโปรแกรมที่ได้ออกแบบไว้แล้วมาเขียนเป็นรหัสต้นฉบับ พัฒนาโปรแกรมให้มีคุณสมบัติตามที่ได้กำหนดไว้ สามารถทำงานได้ตามรูปแบบและบริบทที่ผู้ใช้ต้องการ

ขั้นตอนที่ 4 การทดสอบและแก้ไขโปรแกรม (Program Testing & Verification)

เมื่อพัฒนาแอปพลิเคชันเสร็จแล้วควรนำไปทดสอบการทำงานของระบบ เพื่อหาข้อผิดพลาดหรือการทำงานที่อาจจะไม่ถูกต้อง และปรับปรุงแก้ไขข้อบกพร่องนั้นให้สมบูรณ์

ขั้นตอนที่ 5 **การจัดทำเอกสารและคู่มือการใช้งาน (Program Documentation)**

จัดทำเอกสารประกอบการพัฒนาแอปพลิเคชันเพื่อบอกคุณสมบัติ องค์ประกอบ และข้อจำกัดต่าง ๆ ในการใช้งานโปรแกรม พร้อมจัดทำคู่มือการใช้งานเพื่อให้ผู้ใช้งานใหม่ทำความเข้าใจในการใช้งานโปรแกรมได้ง่ายและรวดเร็วที่สุด

ขั้นตอนที่ 6 **การใช้งานจริง (Program Implement)**

การนำระบบไปใช้งานจริง พร้อมทั้งติดตามตรวจสอบผลการใช้งานและข้อบกพร่องที่อาจจะเกิดขึ้น

ขั้นตอนที่ 7 **การปรับปรุงและพัฒนาโปรแกรม (Program Maintenance)**

ปรับปรุงและพัฒนาโปรแกรมให้ทันสมัยอยู่เสมอ

ตัวอย่าง ขั้นตอนการพัฒนาแอปพลิเคชันเกมบนสมาร์ตโฟน

"นักเรียนพบว่าในการเรียนภาษาอังกฤษนั้น มีคำศัพท์หลายคำที่ท่องจำได้ยาก แต่นักเรียนกลับสังเกตได้ว่าเพื่อนที่เล่นเกมโดยเกมนั้นมีการใช้ภาษาอังกฤษ ทำให้เพื่อนที่เล่นเกมหลายคนจำคำศัพท์เหล่านั้นได้ดี นักเรียนจึงมีแนวคิดที่จะสร้างเกมทายคำศัพท์ โดยต้องการพัฒนาให้เป็นเกมที่สามารถเล่นได้บนสมาร์ตโฟน"

จากสถานการณ์ข้างต้น นักเรียนสามารถใช้วัฏจักรการพัฒนาระบบงาน (System Development Life Cycle: SDLC) ในการพัฒนาแอปพลิเคชันที่ต้องการได้โดยมีกระบวนการ ดังนี้

ขั้นตอนที่ 1 **การวิเคราะห์ปัญหา (Problem Analysis)**

นักเรียนพบปัญหาในการเรียนภาษาอังกฤษที่ต้องท่องจำคำศัพท์ต่าง ๆ มากมาย และต้องการแก้ปัญหา โดยการสร้างเป็นแอปพลิเคชันช่วยจำและเรียนรู้เกี่ยวกับภาษาอังกฤษผ่านการเล่นเกม

ขั้นตอนที่ 2 การออกแบบโปรแกรม (Program Design)

ออกแบบโปรแกรมเป็นภาพวาดแสดงโครงสร้างต่าง ๆ ของแอปพลิเคชันอย่างละเอียด

ขั้นตอนที่ 3 การเขียนโปรแกรม (Program Coding)

ในการพัฒนาแอปพลิเคชันลักษณะนี้ นักเรียนสามารถใช้เครื่องมือในการพัฒนาแอปพลิเคชันในรูปแบบ Drag and Drop และ Block Programming ได้ เช่น App Inventor หรือ Thunkable

ขั้นตอนที่ 4 การทดสอบและแก้ไขโปรแกรม (Program Testing & Verification)

เมื่อพัฒนาแอปพลิเคชันนี้เรียบร้อยแล้ว ควรนำไปทดสอบการทำงานเพื่อสอบถามความพึงพอใจของผู้ใช้งานและตรวจหาข้อบกพร่อง สำหรับแอปพลิเคชันนี้สามารถนำไปทดลองกับนักเรียนกลุ่มที่ท่องจำคำศัพท์ได้ยาก และศึกษาความพึงพอใจของผู้ใช้งาน

ขั้นตอนที่ 5 การจัดทำเอกสารและคู่มือการใช้งาน (Program Documentation)

แอปพลิเคชันเรียนรู้เกี่ยวกับคำศัพท์ภาษาอังกฤษนี้ ควรมีเอกสารและคู่มือการใช้งาน อาจจะอยู่ในรูปของ Online Document เนื่องจากเป็นที่นิยมและใช้งานได้ทั่วไป

ขั้นตอนที่ 6 การใช้งานจริง (Program Implement)

การนำแอปพลิเคชันไปเผยแพร่และใช้งานจริง โดยการประชาสัมพันธ์ผ่านช่องทางต่าง ๆ เช่น การให้เพื่อนในห้องเรียนได้ใช้งาน การเผยแพร่ผ่านสื่อออนไลน์

ขั้นตอนที่ 7 การปรับปรุงและพัฒนาโปรแกรม (Program Maintenance)

หลังจากที่เผยแพร่แอปพลิเคชันนี้ไปแล้ว ควรติดตามและเก็บรวบรวมข้อมูลเกี่ยวกับผู้ใช้งานในด้านความพึงพอใจ และความต้องการอื่น ๆ เพื่อนำมาปรับปรุงและพัฒนาให้ดีขึ้นต่อไป

เครื่องมือที่ใช้ในการพัฒนาแอปพลิเคชัน

1. MIT App Inventor

เป็นเครื่องมือที่ใช้ในการสร้างแอปพลิเคชันที่ทำงานบนระบบปฏิบัติการแอนดรอยด์ (Android)

มีองค์ประกอบที่สำคัญ 2 ส่วน

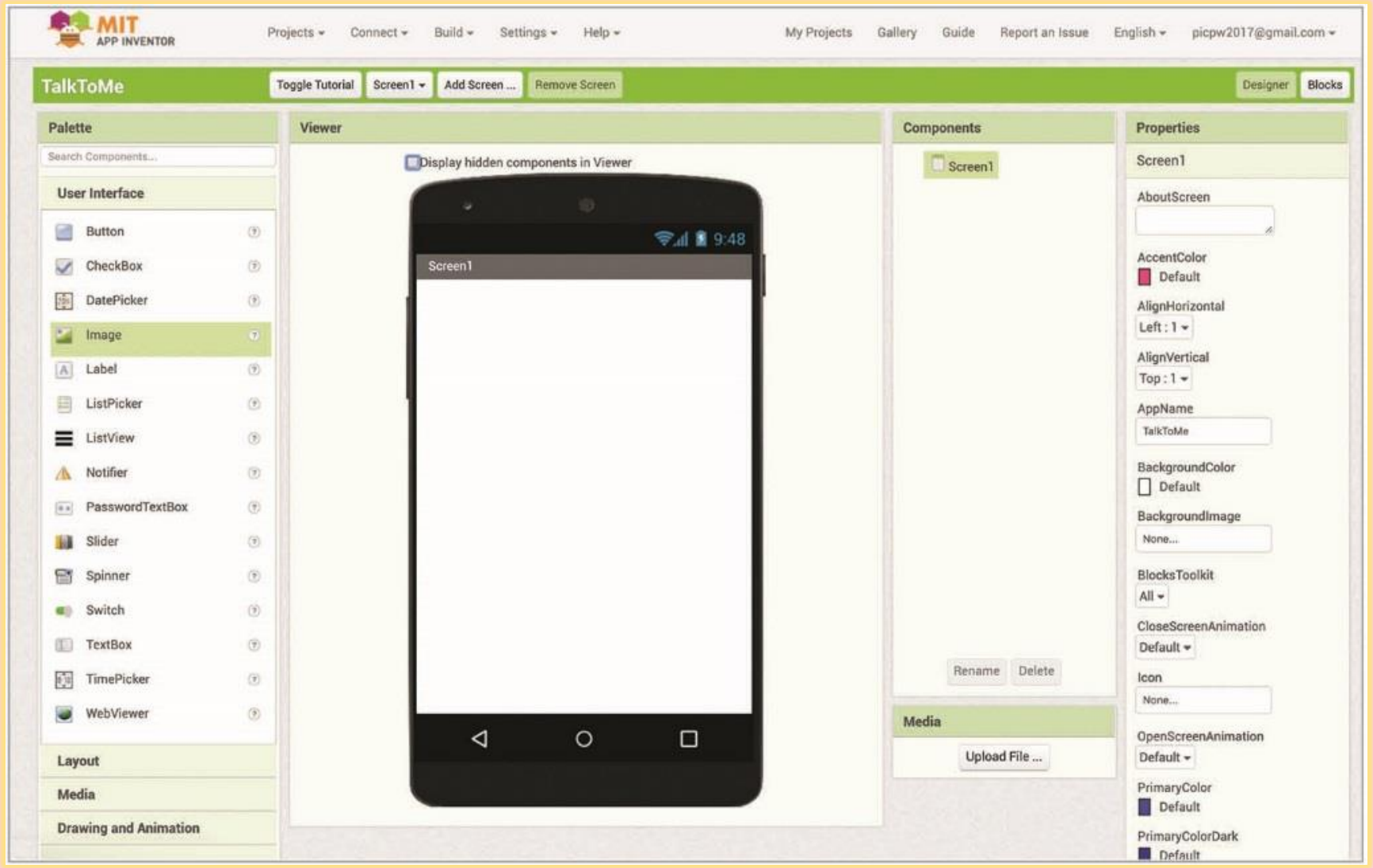
ส่วนของการออกแบบส่วนติดต่อผู้ใช้งาน (UI)

สามารถลากสิ่งที่ต้องการแสดงผลมาไว้บนหน้าจอแอปพลิเคชันและตกแต่งได้ตามต้องการ

ส่วนของการเขียนชุดคำสั่ง (Blocks)

คำสั่งที่ผู้พัฒนาสามารถเขียนคำสั่งให้แอปพลิเคชันทำงาน โดยคำสั่งที่ใช้อยู่ในรูปของบล็อกคำสั่งที่เชื่อมต่อกันไว้อย่างเป็นลำดับและมีระบบ

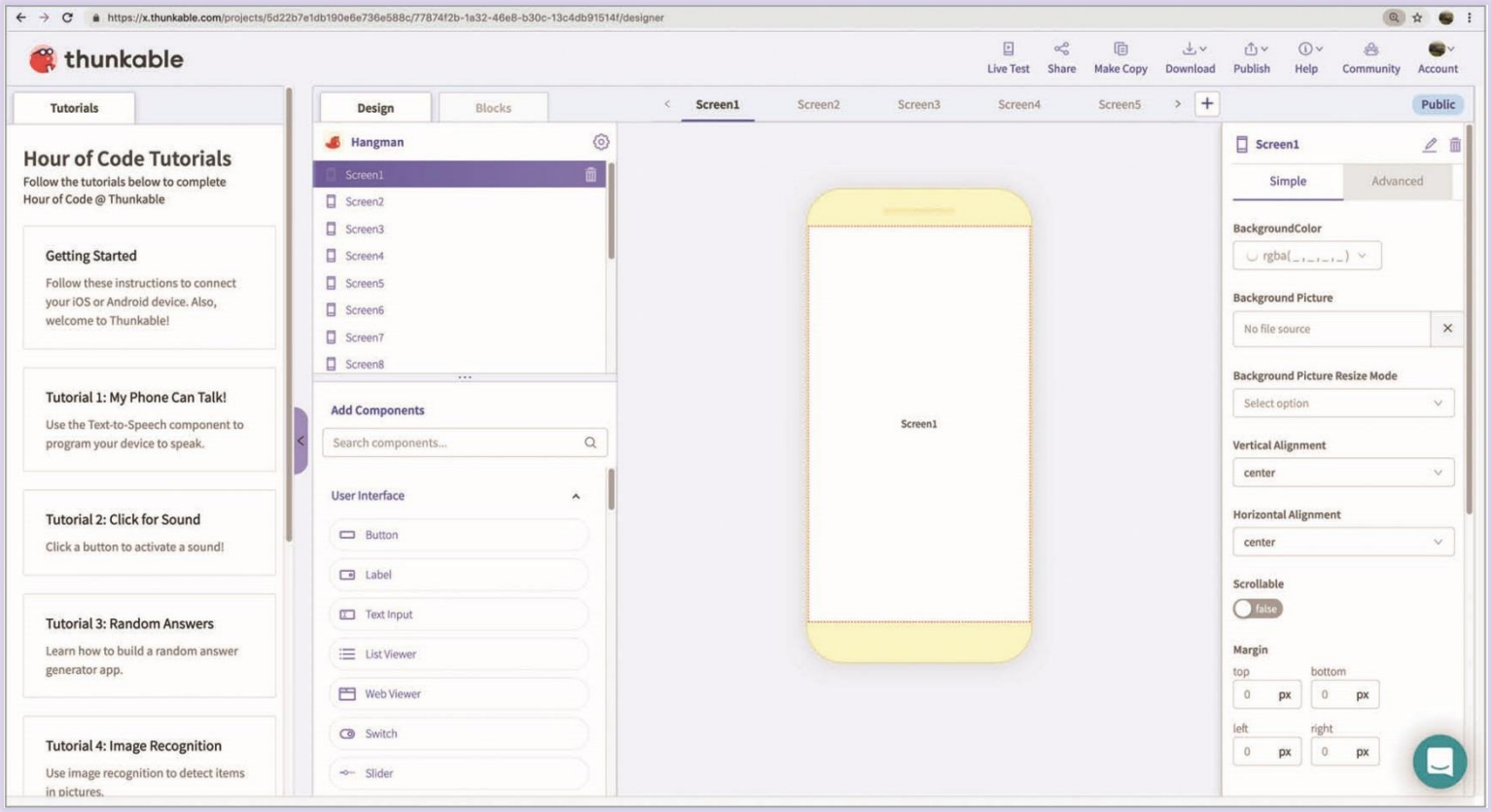
ตัวอย่างหน้าจอ MIT App Inventor



2. Thunkable

เป็นเครื่องมือในการพัฒนาแอปพลิเคชันที่มีรูปแบบเดียวกันกับ MIT App Inventor มีความแตกต่างกันไม่มาก สามารถพัฒนาแอปพลิเคชันพื้นฐานที่ไม่มีความซับซ้อนได้

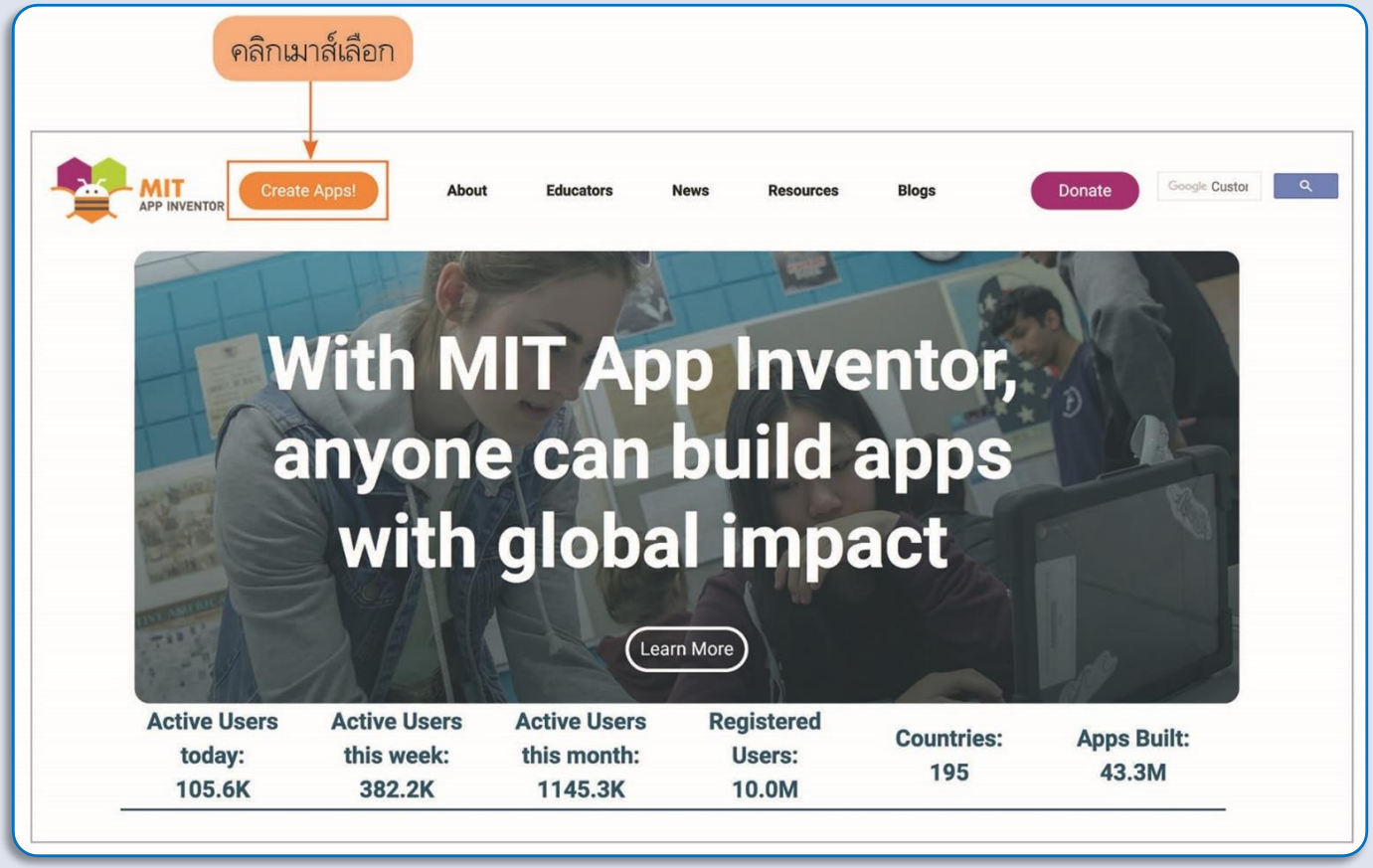
ภาพตัวอย่างหน้าจอ Thunkable



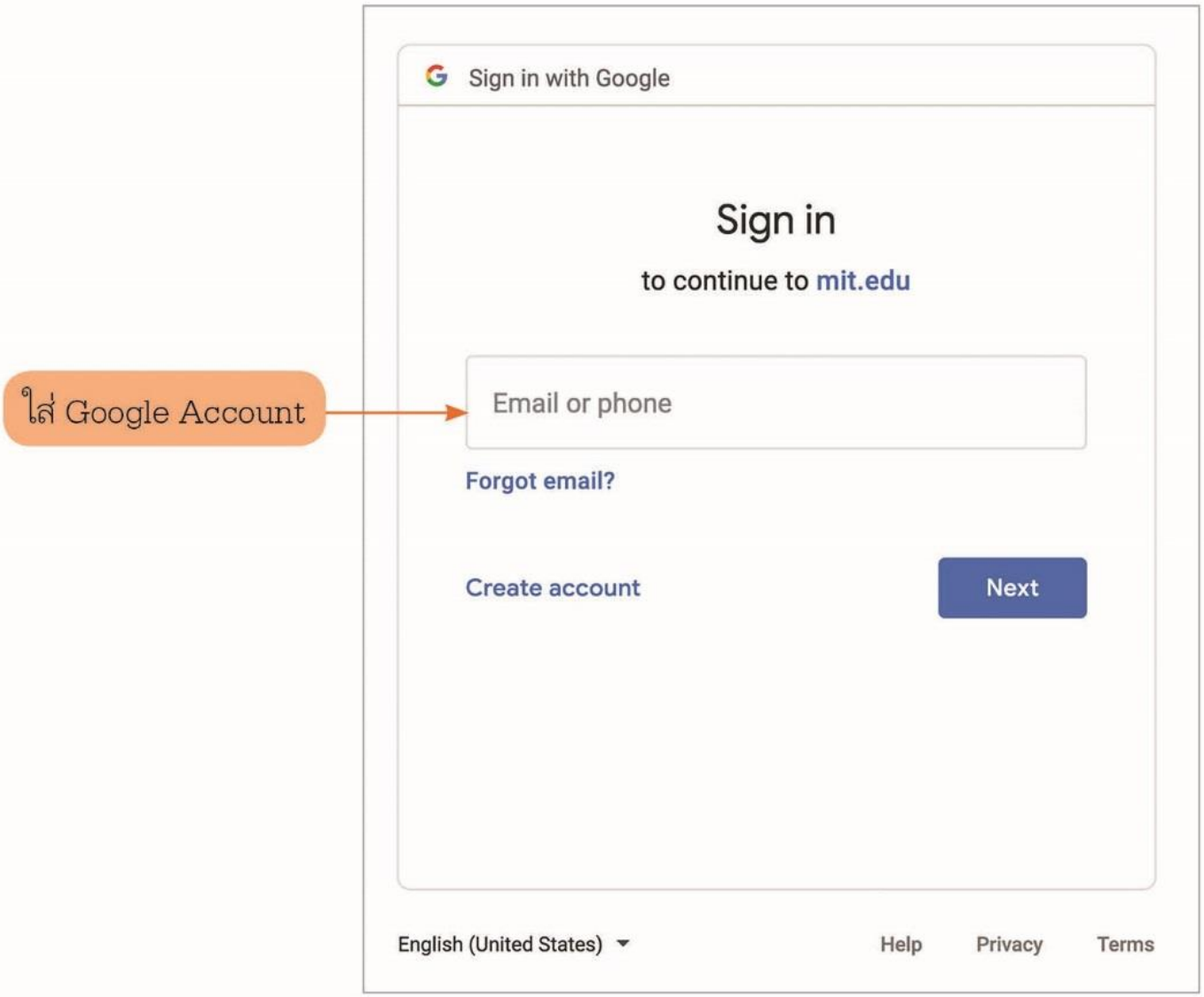
การพัฒนาแอปพลิเคชันบนสมาร์ตโฟนด้วย MIT App Inventor

ตัวอย่างที่ 1 การพัฒนาแอปพลิเคชันเครื่องคิดเลข

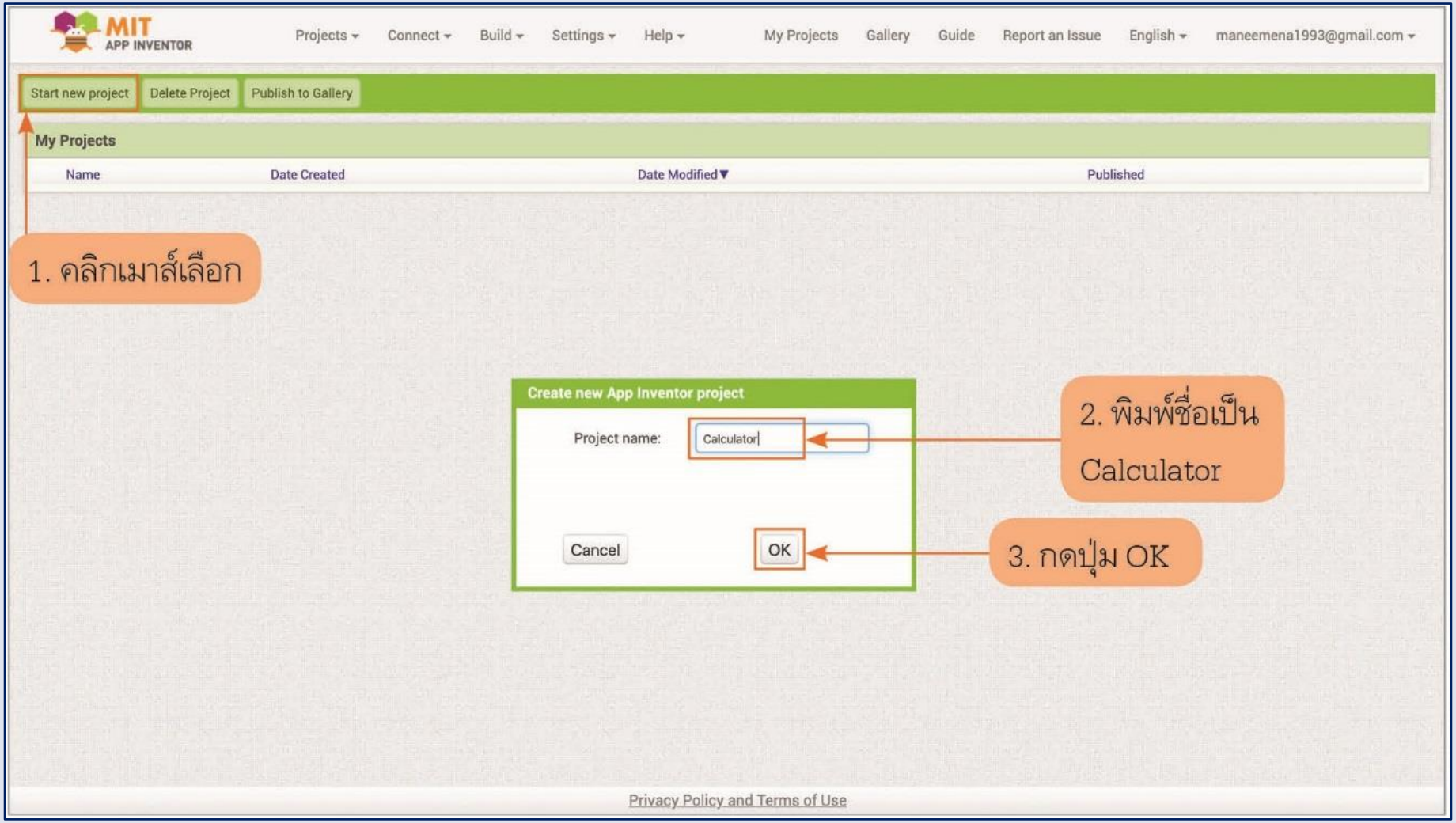
1. เข้าเว็บไซต์ https://appinventor.mit.edu/ และเข้าสู่ระบบด้วย Google Account



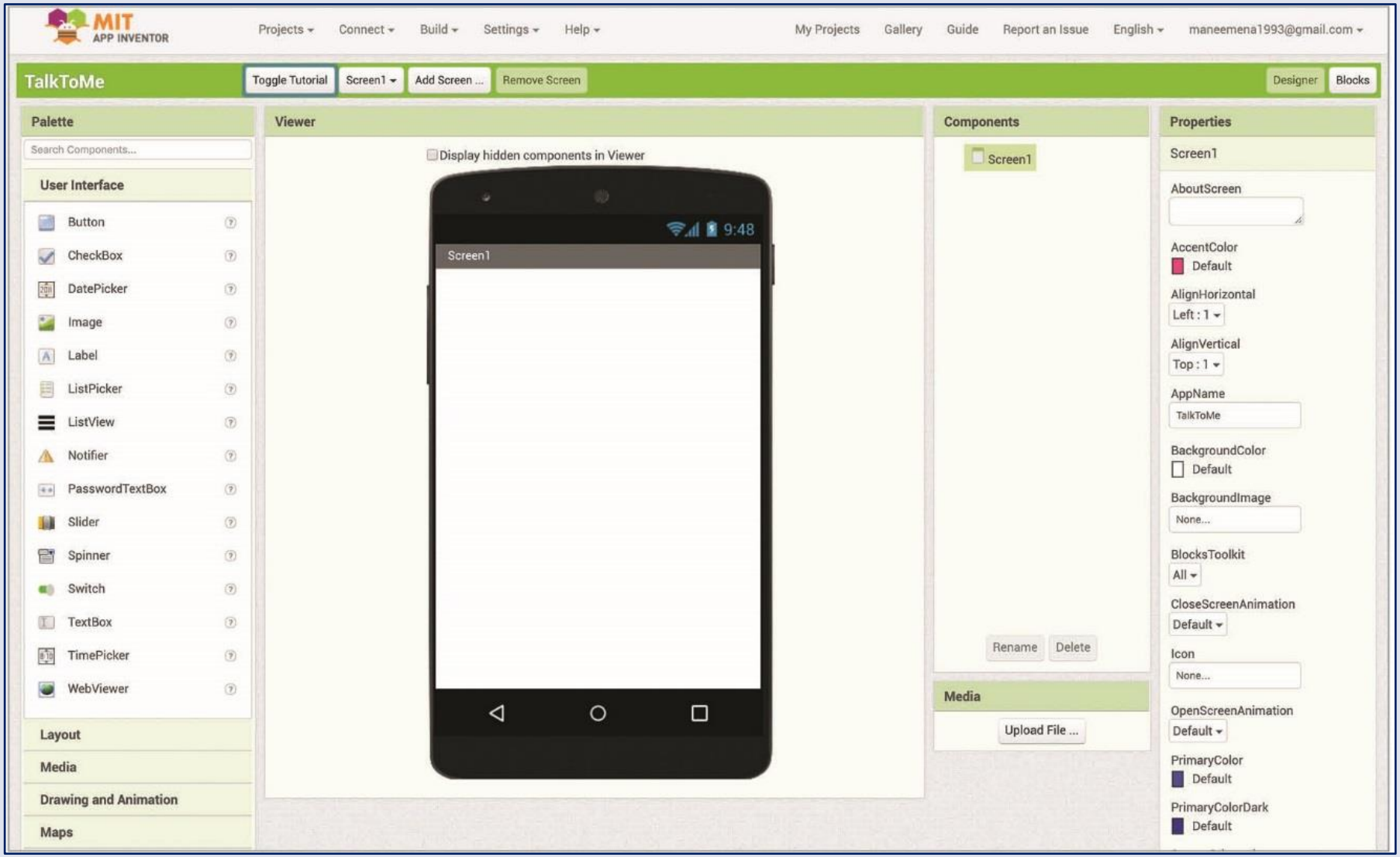
เทคโนโลยี (วิทยาการคำนวณ) ชั้นมัธยมศึกษาปีที่ 3



2. คลิกเมาส์เลือก Start new project และตั้งชื่อที่ต้องการ เช่น Calculator



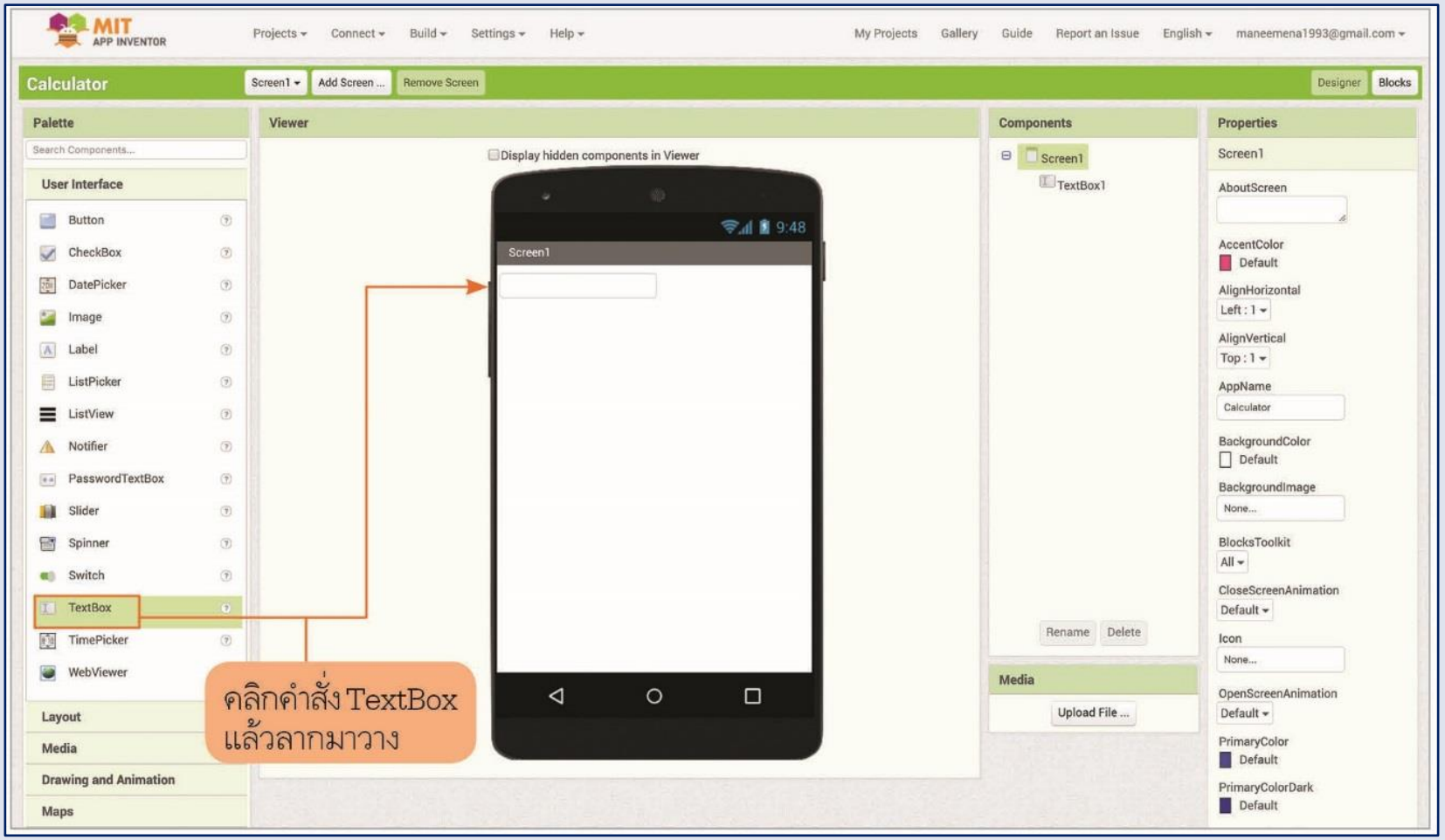
3. จะปรากฏหน้าจอโปรแกรมสำหรับสร้างแอปพลิเคชันขึ้นมา



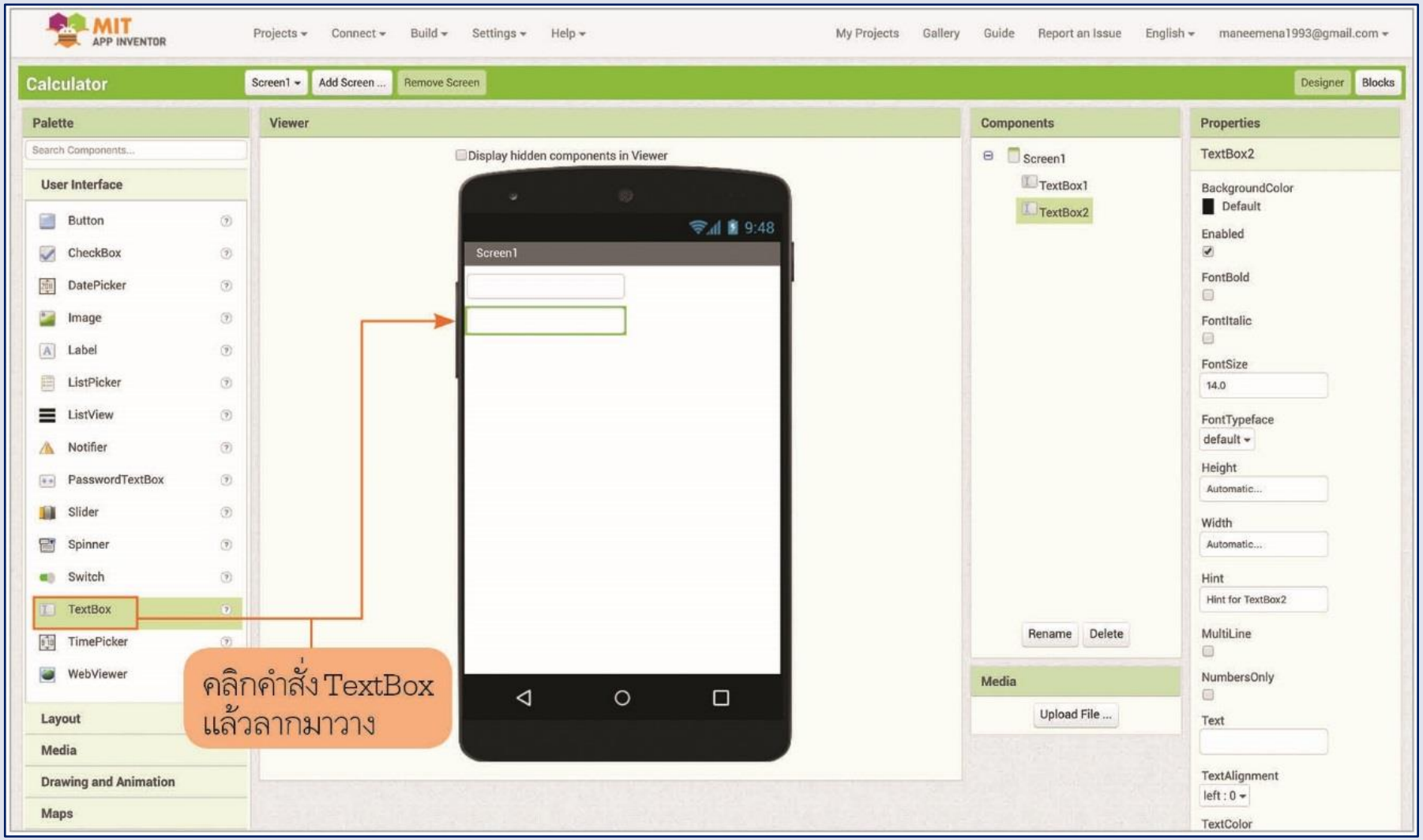
4. ออกแบบหน้าจอเครื่องคิดเลข

4.1 คลิกเมาส์คำสั่ง

แล้วลากมาวางบนหน้าจอสมาร์ตโฟน สำหรับรับค่าตัวเลขจำนวนที่ 1

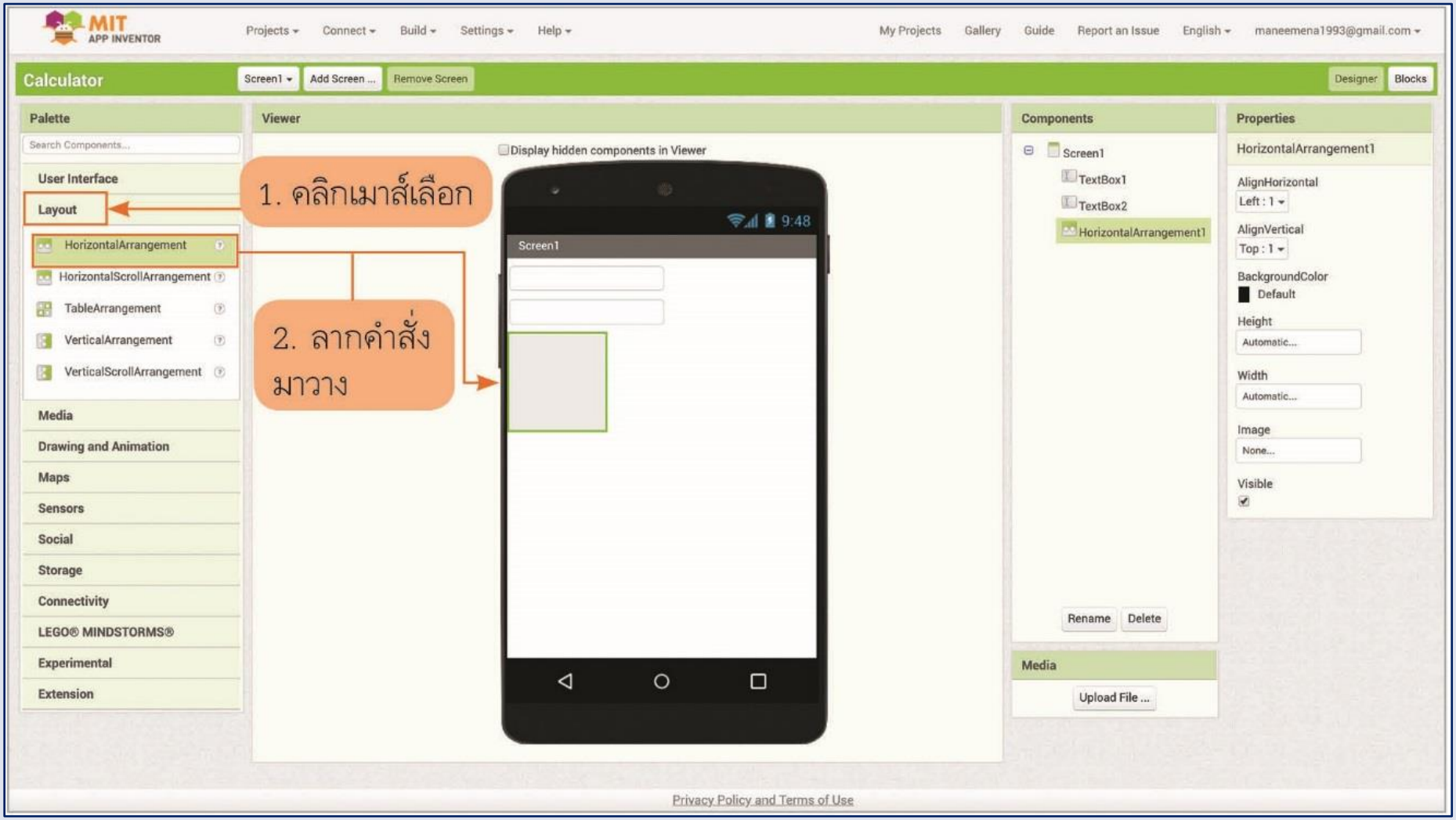


4.2 คลิกเมาส์คำสั่ง TextBox อีกครั้งแล้วลากมาวางบนหน้าจอสมาร์ตโฟน สำหรับรับค่าตัวเลขจำนวนที่ 2



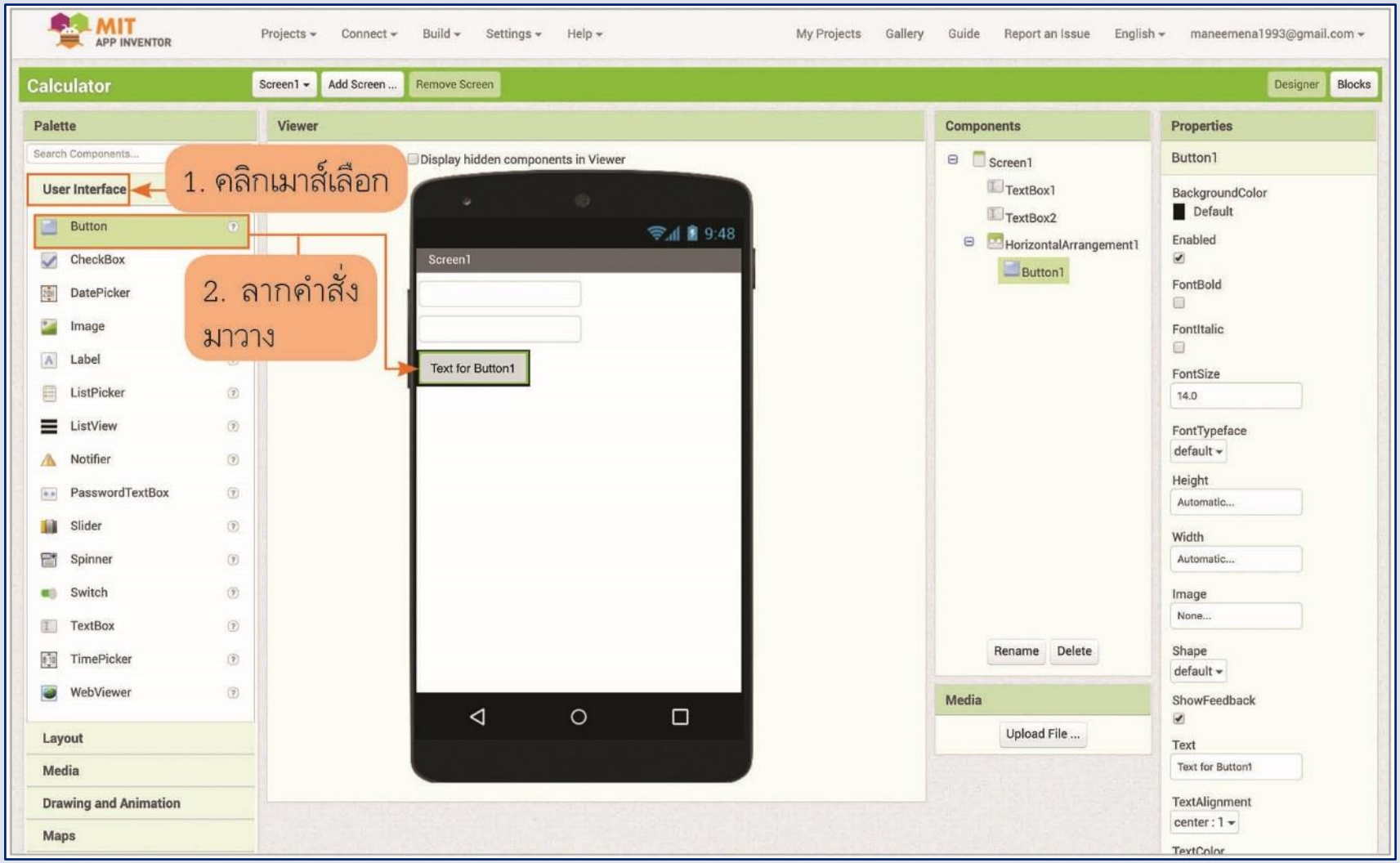
4.3 คลิกเมาส์คำสั่ง Layout แล้วลากคำสั่ง

มาวางบนหน้าจอสมาร์ตโฟน เพื่อจัดวางตัวดำเนินการ (บวก ลบ คูณ หาร) ให้อยู่ในแนวนอน

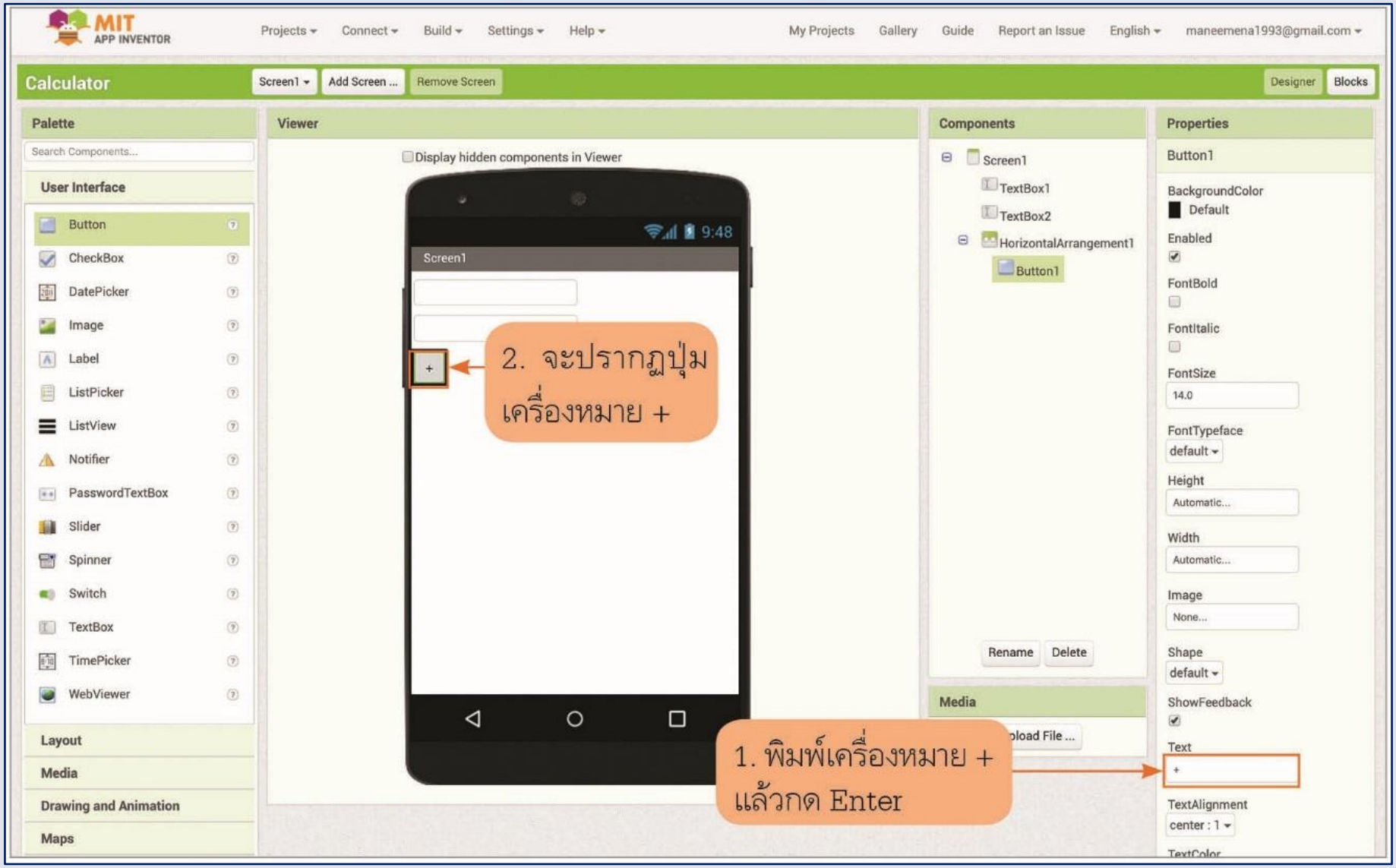


4.4 ใส่ปุ่มกดเครื่องหมายบวก (+) โดยคลิกเมาส์ที่คำสั่ง User Interface เลือก

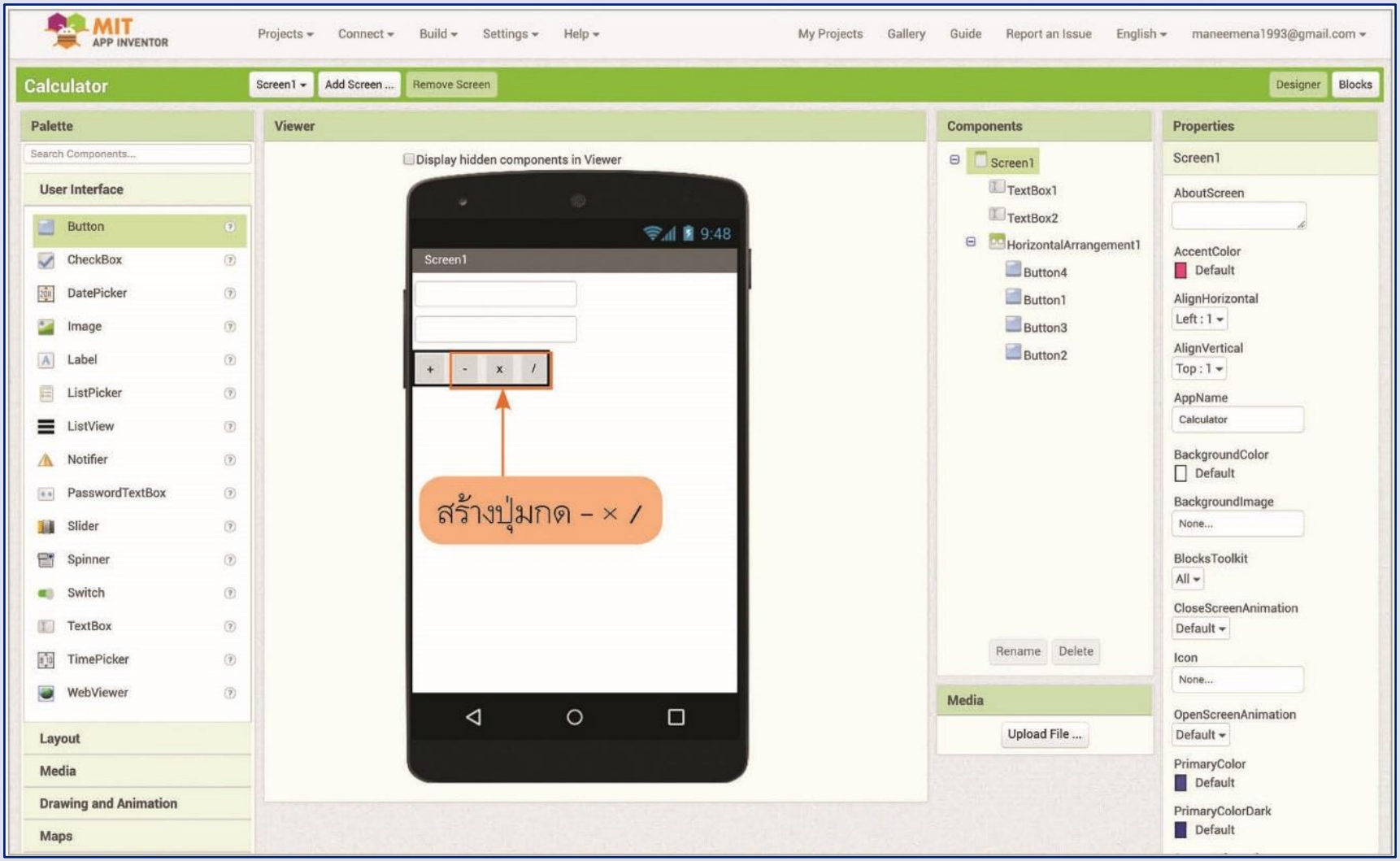
แล้วลากมาวาง



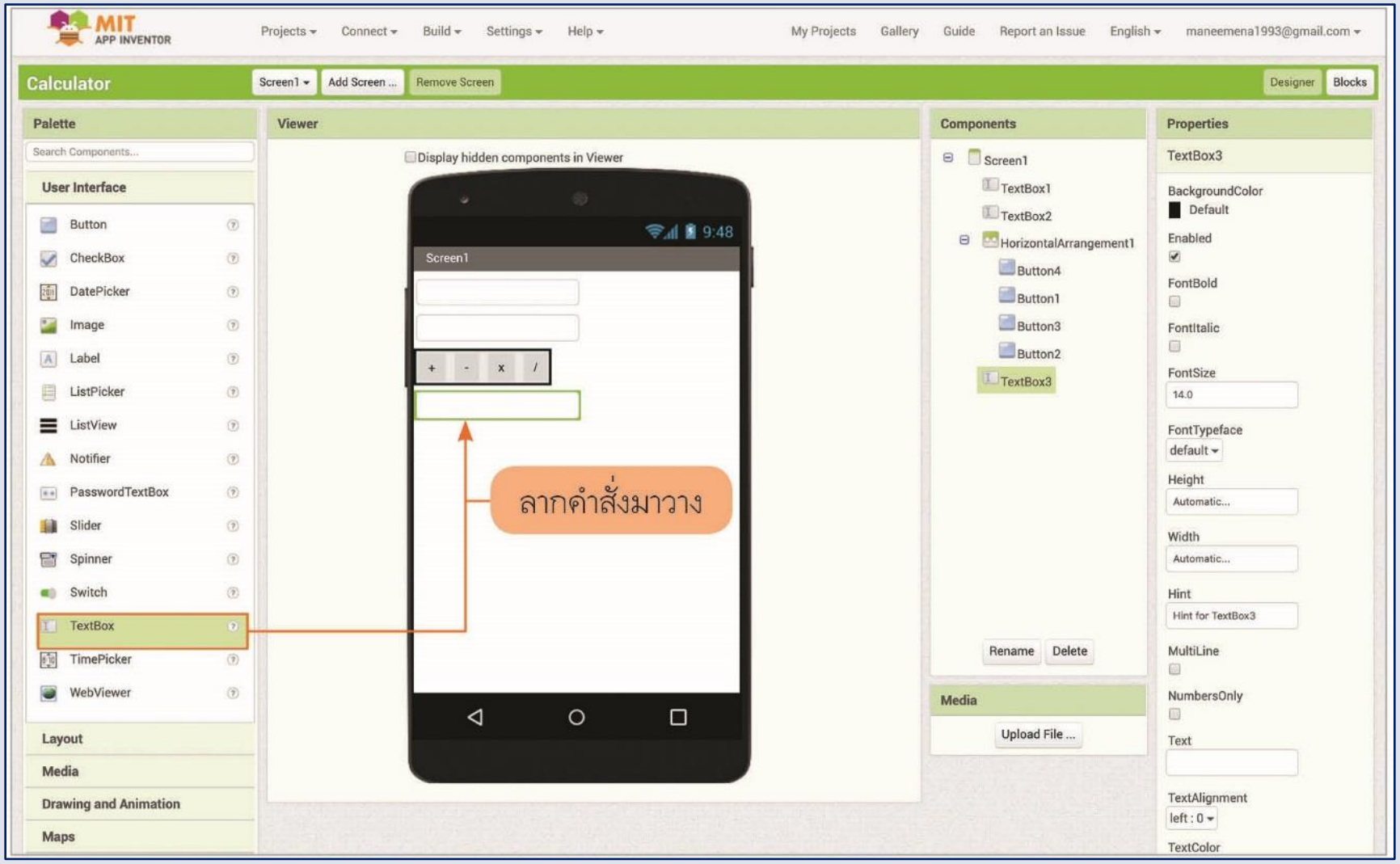
4.5 เปลี่ยนชื่อปุ่มกดเป็นเครื่องหมาย +



4.6 สร้างปุ่มกดเครื่องหมายลบ (–) คูณ (×) หาร (/) ให้ครบ
โดยใช้วิธีเดียวกันกับการสร้างปุ่มเครื่องหมายบวก (+)

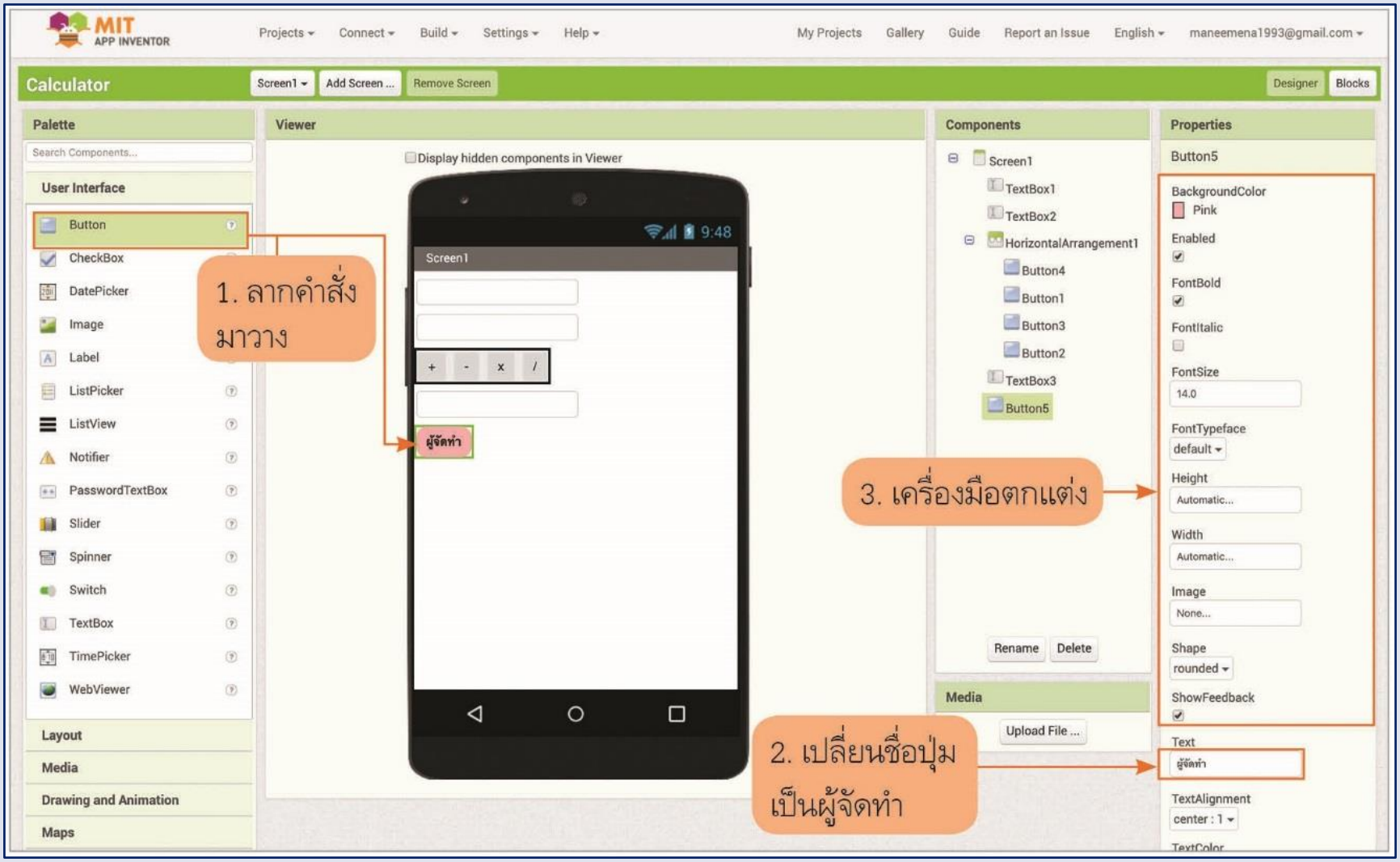


4.7 คลิกเมาส์คำสั่ง TextBox แล้วลากมาวางบนหน้าจอสมาร์ตโฟน สำหรับแสดงผลลัพธ์การคำนวณ



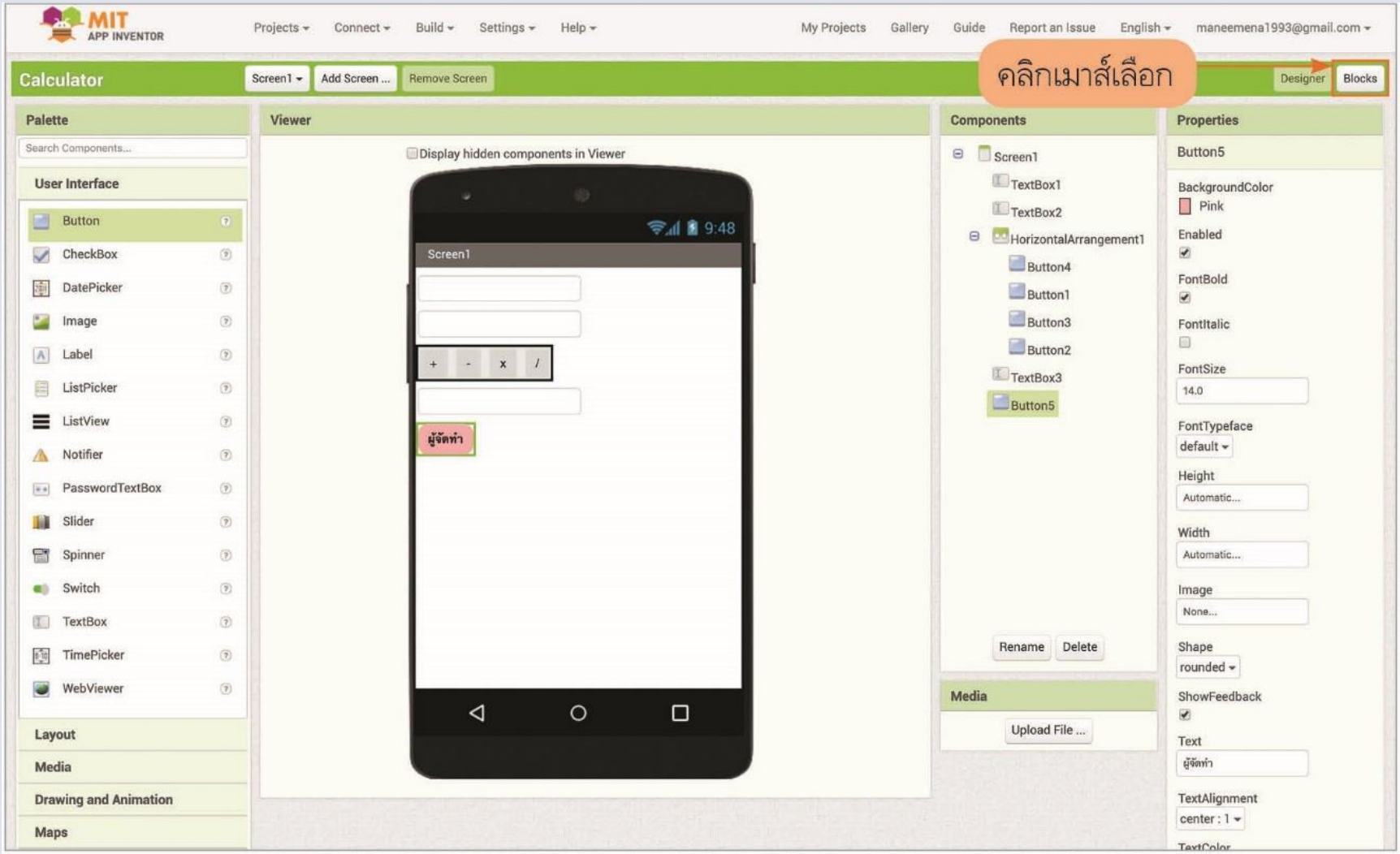
4.8 สร้างปุ่มผู้จัดทำโดยคลิกเมาส์ที่คำสั่ง

แล้วลากมาวางบนหน้าจอสมาร์ตโฟน จากนั้นเปลี่ยนชื่อปุ่มเป็น “ผู้จัดทำ” พร้อมตกแต่งให้สวยงาม

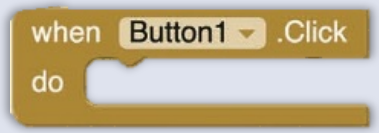


5. เขียนคำสั่งควบคุม

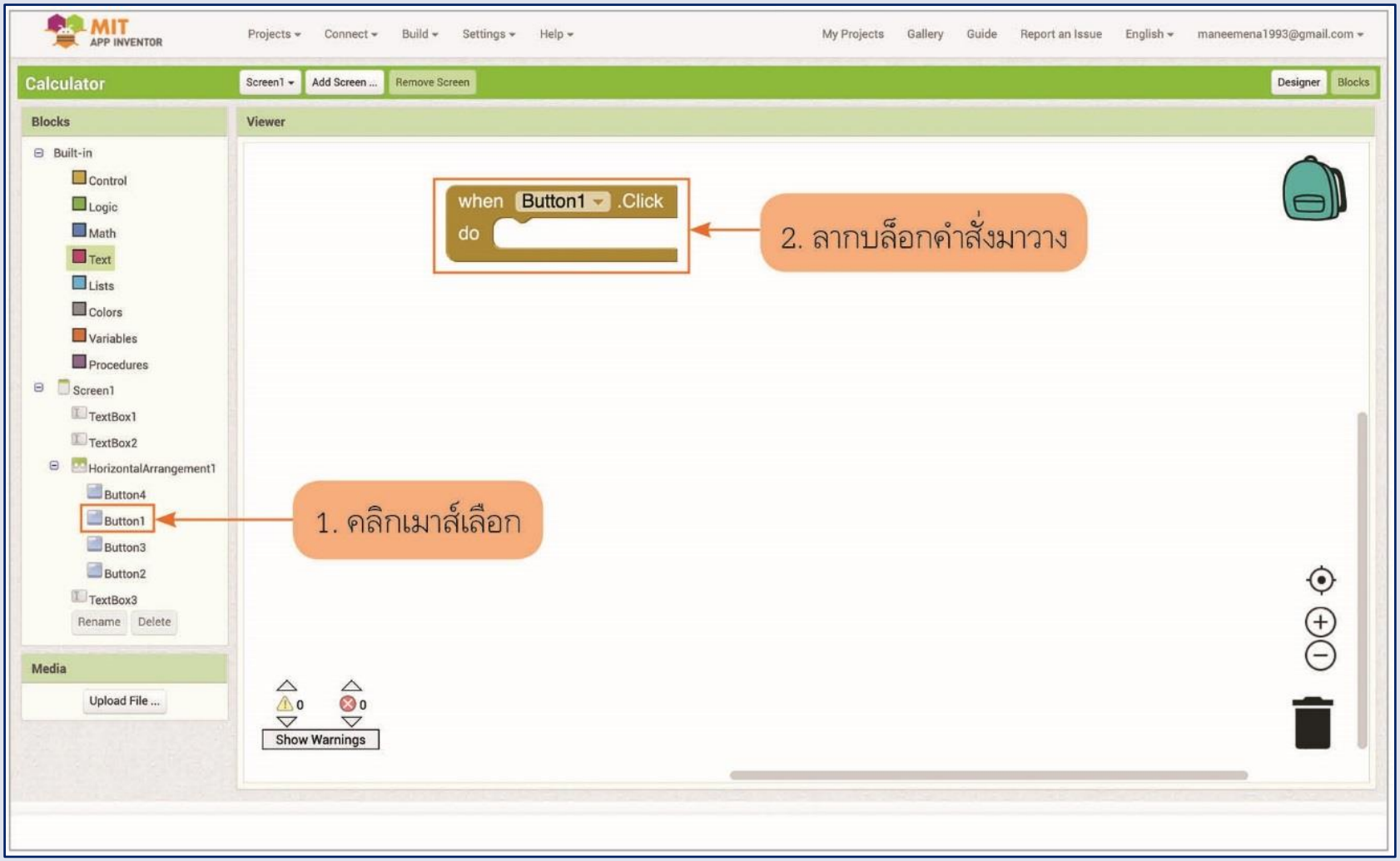
5.1 คลิกเมาส์เลือก Blocks เพื่อเขียนคำสั่งควบคุม



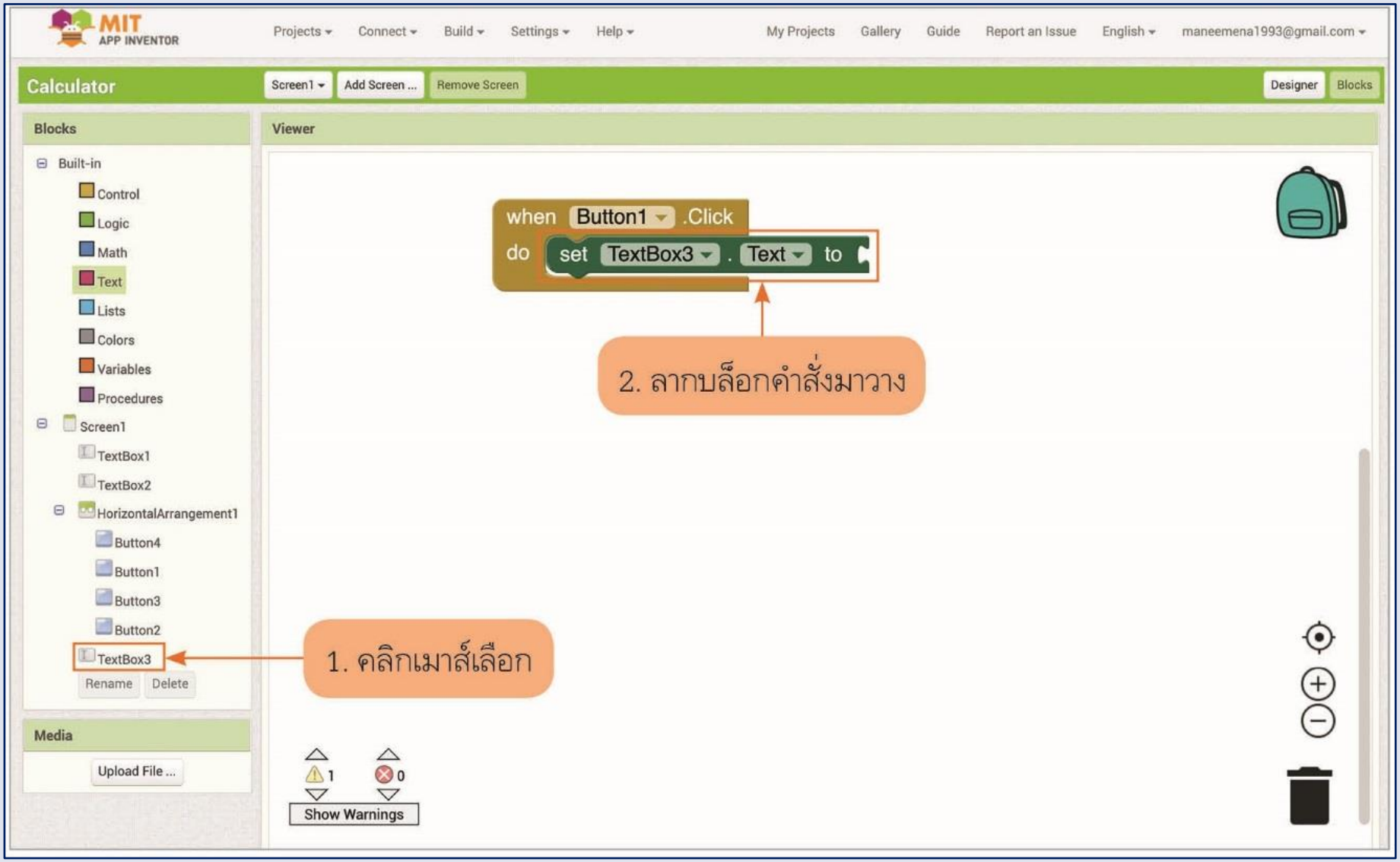
5.2 จะปรากฏหน้าจอสำหรับเขียนโปรแกรม จากนั้นคลิกเมาส์

แล้วลากบล็อกคำสั่ง 

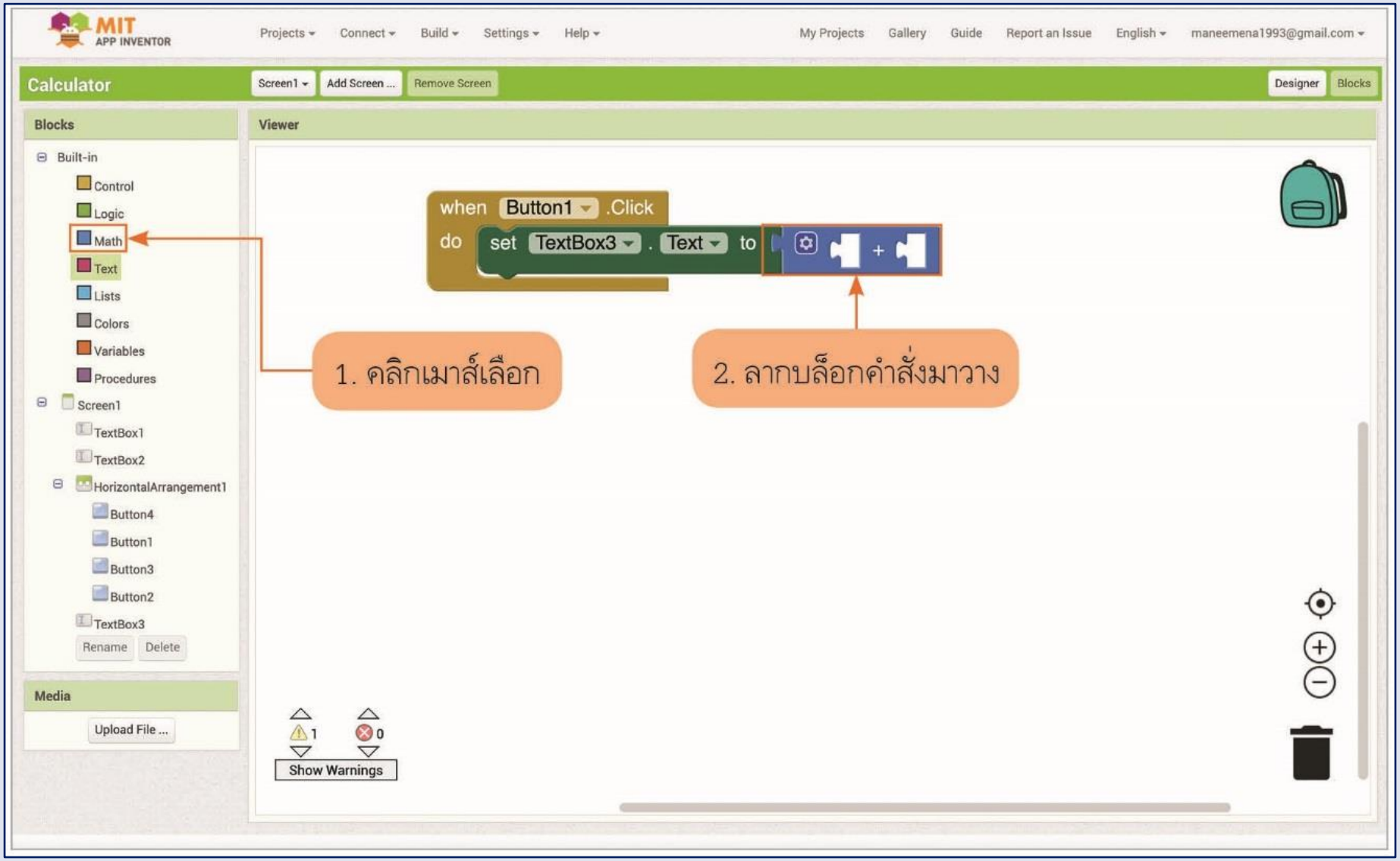
มาวาง



5.3 คลิกเมาส์เลือก TextBox3 แล้วลากบล็อกคำสั่ง set TextBox3 . Text to มาวาง



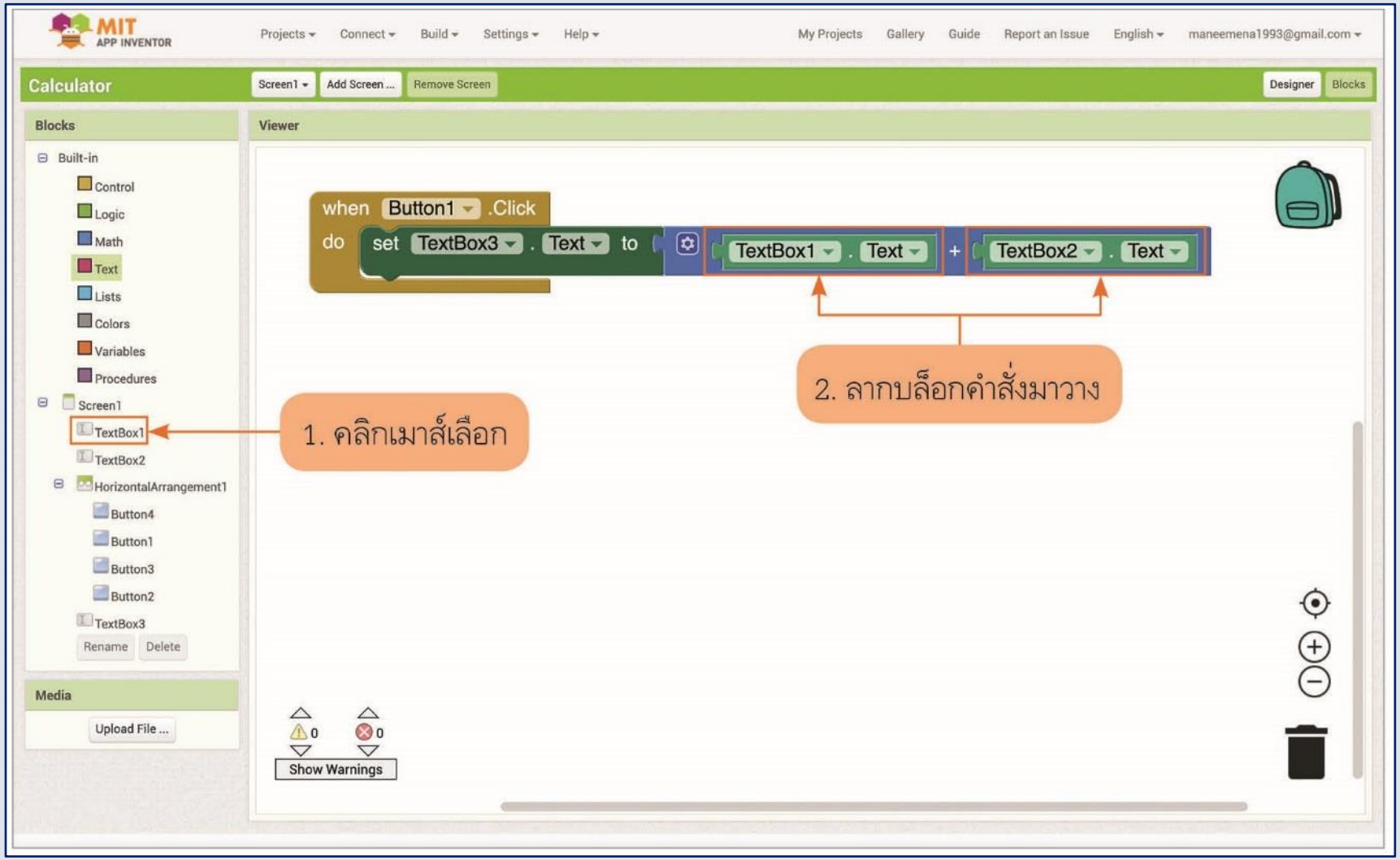
5.4 คลิกเมาส์เลือก แล้วลากบล็อกคำสั่ง มาวาง



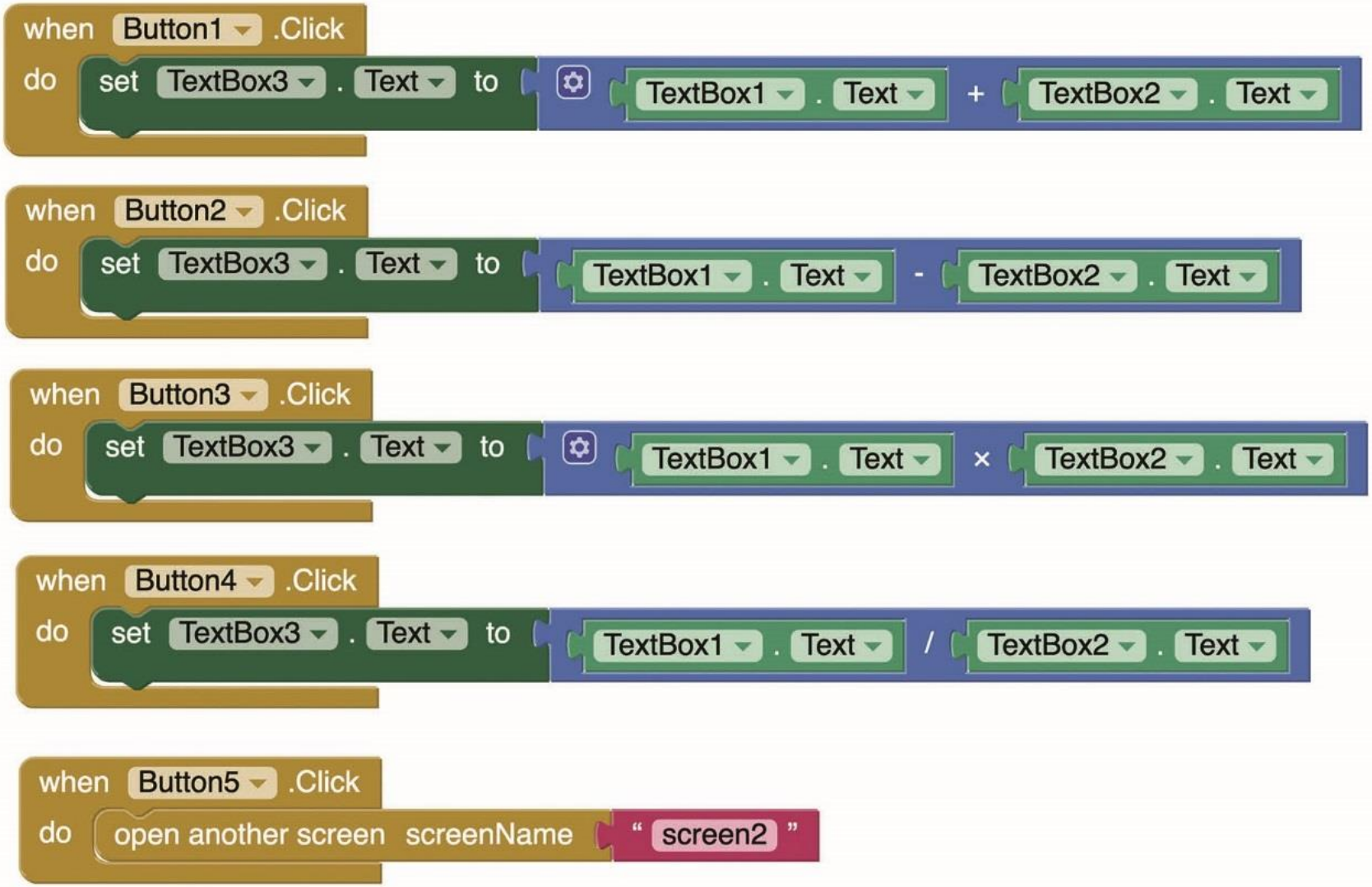
5.5 คลิกเมาส์เลือก

แล้วลากบล็อกคำสั่ง

มาวาง



5.6 เขียนคำสั่งควบคุมให้ครบ



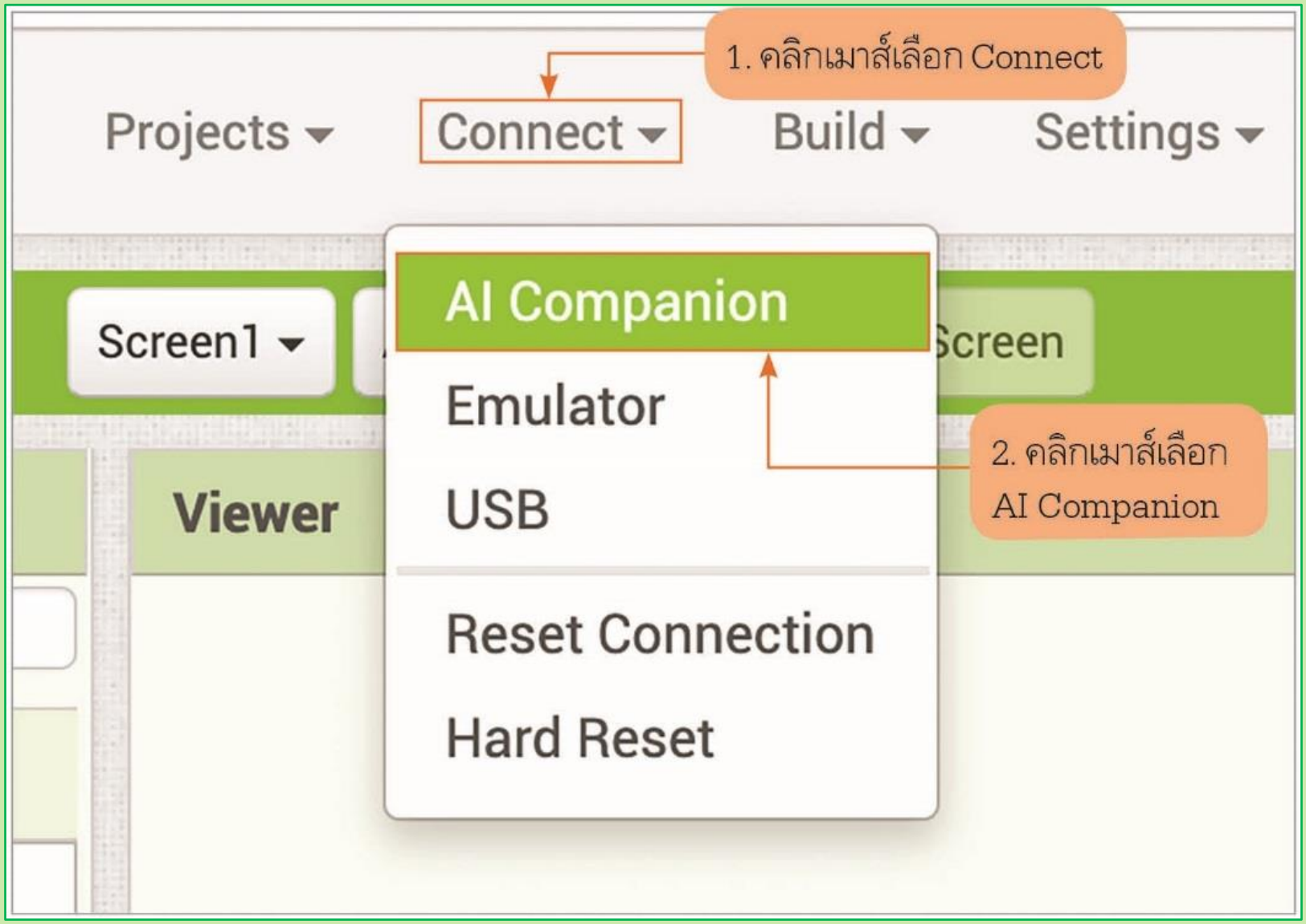
การทดสอบผลการทำงาน สามารถทำได้ 3 วิธี

วิธีที่ 1

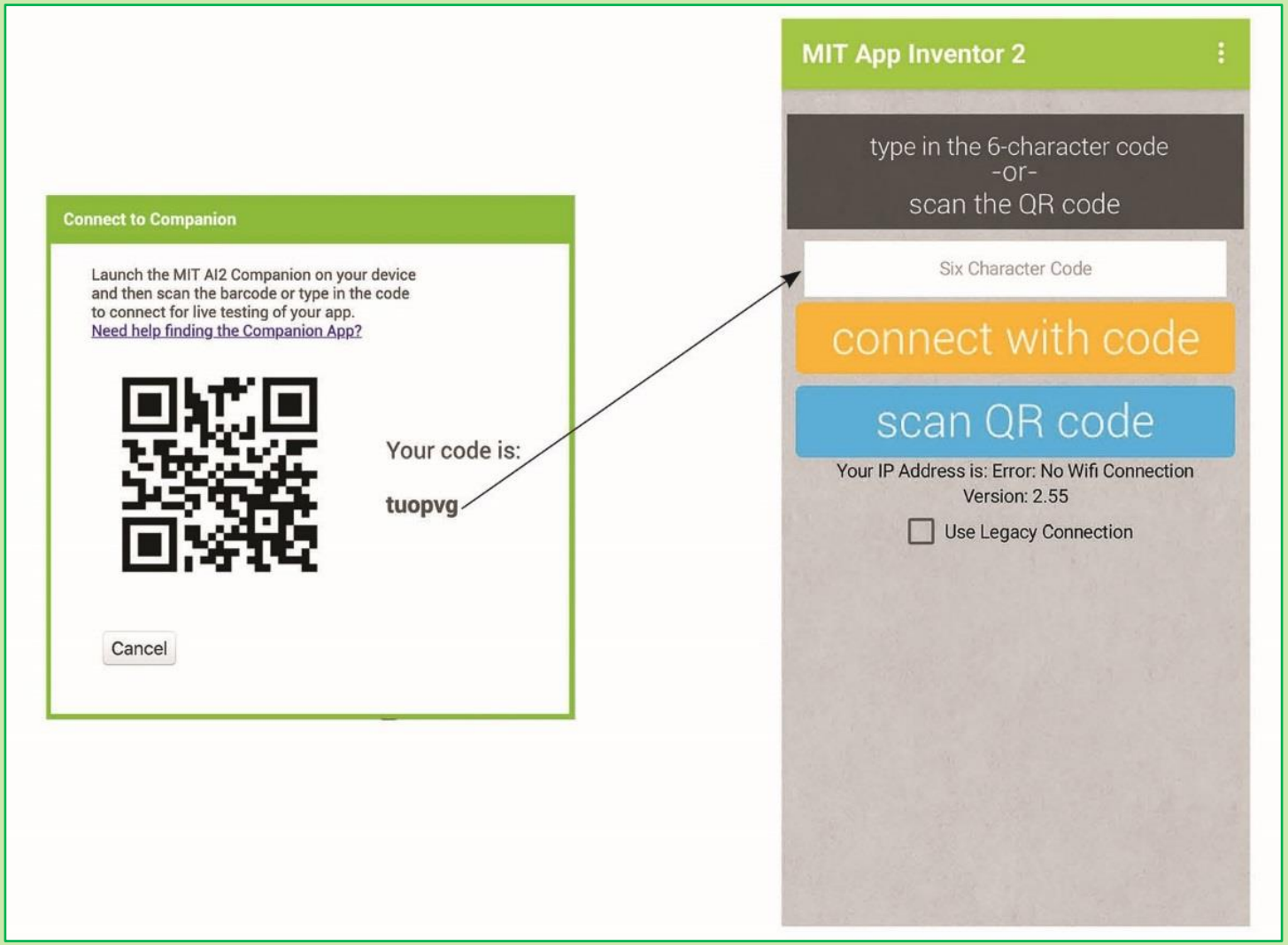
การเชื่อมต่อผ่านระบบเครือข่ายไร้สาย (WiFi) ด้วยแอปพลิเคชัน MIT AI2 Companion โดยปฏิบัติตามขั้นตอน ดังนี้

1. ดาวน์โหลดและติดตั้งแอปพลิเคชัน MIT AI2 Companion ในสมาร์ตโฟน (เฉพาะระบบปฏิบัติการ Android เท่านั้น)

2. คลิกเมาส์ที่เมนู Connect แล้วเลือก AI Companion ที่หน้าเว็บไซต์



3. เปิดแอปพลิเคชัน MIT AI2 Companion บนสมาร์ตโฟน แล้วใส่ Code หรือสแกน QR Code ที่ปรากฏบนหน้าเว็บไซต์



วิธีที่ 2

การใช้ Emulator เพื่อแสดงสมาร์ตโฟนจำลองบนเครื่องคอมพิวเตอร์ วิธีการนี้เหมาะสำหรับผู้ที่ไม่มีสมาร์ตโฟนที่เป็นระบบปฏิบัติการ Android โดยปฏิบัติตามขั้นตอน ดังนี้

ดาวน์โหลด/ติดตั้ง โปรแกรม aiStarter

เปิดโปรแกรม aiStarter

เลือกเมนู Connect

เลือก Emulator

ระบบจะประมวลผลและแสดงหน้าสมาร์ตโฟนจำลอง

วิธีที่ 3 การเชื่อมต่อผ่านสาย USB

ดาวน์โหลด/ติดตั้งโปรแกรม aiStarter

ดาวน์โหลด/ติดตั้งแอปพลิเคชัน MIT AI2 Companion ในสมาร์ตโฟน

เชื่อมต่อสมาร์ตโฟนกับคอมพิวเตอร์ด้วยสาย USB

หน้าเว็บไซต์เลือกเมนู Connect แล้วเลือก USB

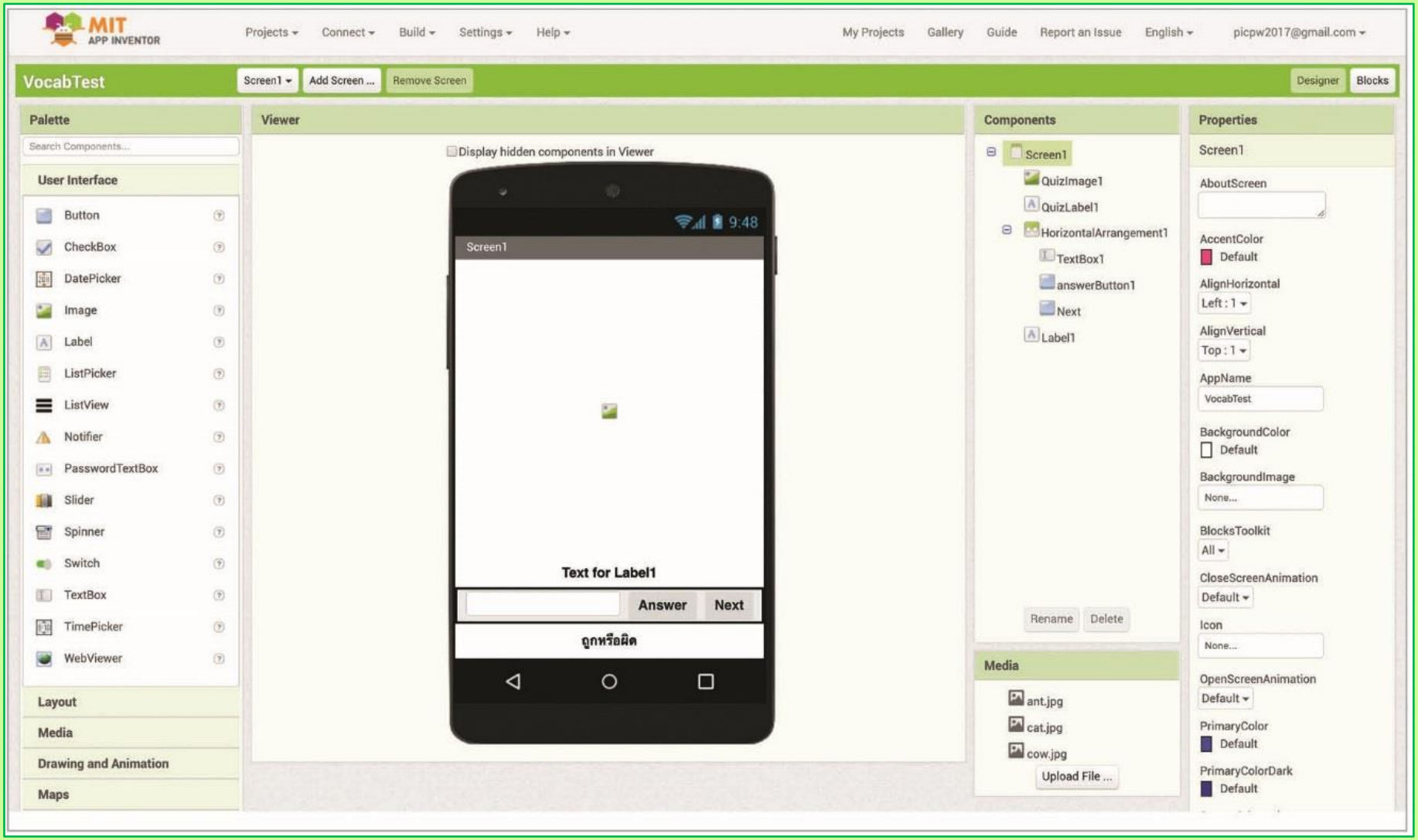
ตัวอย่างที่ 2 การพัฒนาแอปพลิเคชันตอบคำถาม

สร้างแอปพลิเคชันโดยใช้ MIT App Inventor

1. สร้างโพรเจกต์ใหม่และตั้งชื่อ “VocabTest”

2. ออกแบบหน้าจอ
   - Image กำหนดความกว้างและความสูง
   - Label1 สำหรับแสดงคำถาม
   - สร้าง Layout แบบแนวนอน (Horizontal)
     - TextBox1 สำหรับเป็นช่องให้ผู้ใช้งานพิมพ์คำตอบ
     - Button1 สำหรับให้ผู้ใช้งานกดส่งคำตอบ
     - Button2 สำหรับให้ผู้ใช้งานเปลี่ยนคำถามไปเป็นข้อถัดไป
     - Label1 สำหรับแสดงข้อความ “ถูก” หรือ “ผิด”

ตัวอย่างการออกแบบหน้าจอ



3. หาภาพตัวอย่างจำนวน 3 ภาพ และอัปโหลดเข้าสู่คลังสื่อ (Media)
เช่น ภาพมด ภาพแมว และภาพวัว โดยควรตั้งชื่อให้สื่อความหมาย

4. ชุดคำสั่งที่ใช้ในการควบคุมการทำงานของแอปพลิเคชัน

- กำหนดค่าเริ่มต้น โดยการสร้างตัวแปรจำนวน 4 ตัว ได้แก่

index ทำหน้าที่เก็บตำแหน่งของลำดับชุดข้อมูลที่จะแสดงผล

questionList ทำหน้าที่เก็บชุดคำถาม

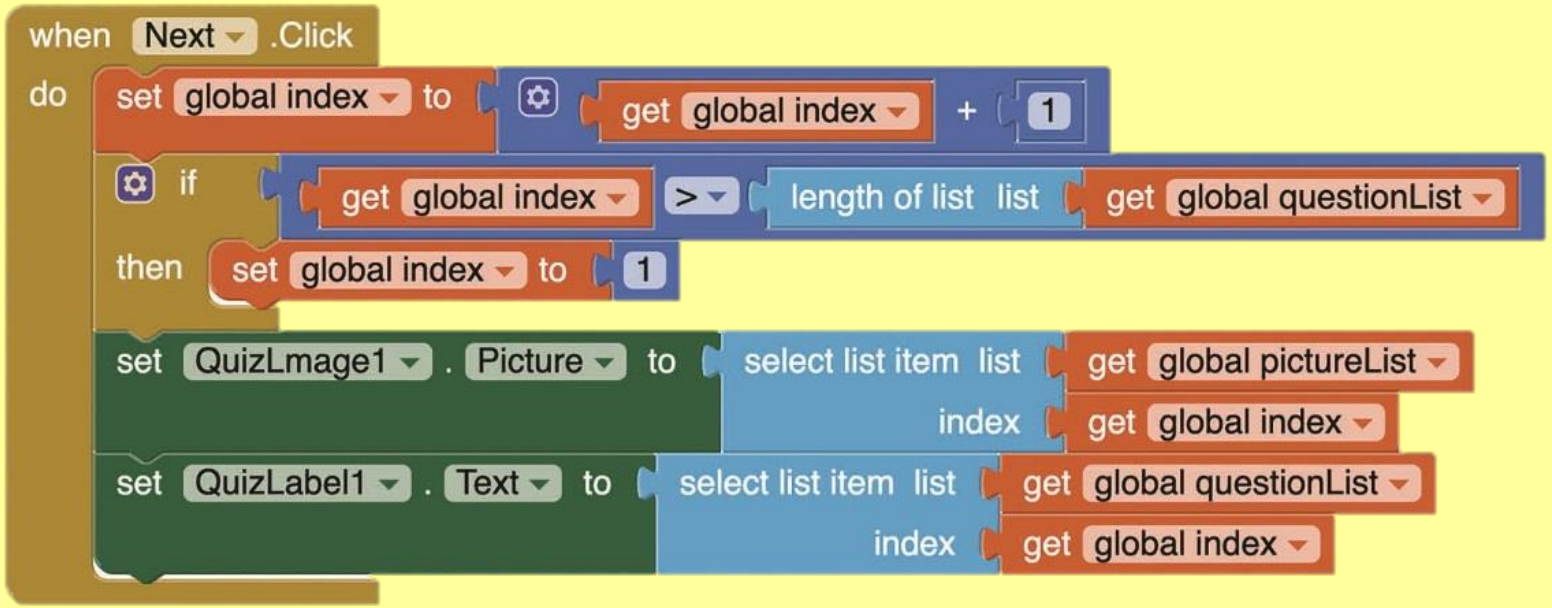
pictureList ทำหน้าที่เก็บชื่อของภาพ

AnswerList ทำหน้าที่เก็บคำตอบที่ถูกต้อง

- กำหนดค่าเริ่มต้นเมื่อแอปพลิเคชันเริ่มทำงาน โดยเมื่อแอปพลิเคชันทำงาน ระบบจะทำการกำหนดภาพและคำถามที่แสดงผลเป็นภาพและคำถามลำดับที่ 1

- คำสั่งควบคุมสำหรับปุ่มส่งคำตอบ มีหน้าที่ตรวจสอบคำตอบที่ผู้ใช้งานพิมพ์เข้ามาในช่อง TextBox1 โดยแปลงคำตอบให้อยู่ในรูปของตัวพิมพ์ใหญ่ แล้วตรวจสอบคำตอบที่กำหนดไว้ในลิสต์ของคำตอบว่าตรงกันหรือไม่ ถ้าตรงกันให้แสดงผลคำว่า “ถูกต้องครับ” ถ้าไม่ตรงกันให้แสดงผลคำว่า “ผิดครับ...พยายามอีกนิดนะ”

- คำสั่งควบคุมสำหรับปุ่มเลื่อนไปข้อถัดไปทำหน้าที่เลื่อนข้อคำถามไปยังข้อถัดไป โดยการเปลี่ยนลำดับการแสดงผลของภาพและข้อคำถาม ซึ่งมีการตรวจสอบลำดับการกดปุ่มถัดไปว่าเกินกว่าลำดับที่มีอยู่หรือไม่ (ในที่นี้มีอยู่ 3 ลำดับ) หากเกินให้กลับไปแสดงผลลำดับที่ 1 อีกครั้ง



5. ทดสอบผลการทำงานของแอปพลิเคชัน โดยทดสอบกับสมาร์ตโฟนหรือ ผ่าน Emulator

อินเทอร์เน็ตของสรรพสิ่ง (Internet of Things)

อินเทอร์เน็ตของสรรพสิ่ง
(Internet of Things) หรือ IoT

↓

อุปกรณ์หรือสิ่งของ
ที่สามารถเชื่อมต่อกับเครือข่าย
คอมพิวเตอร์หรืออินเทอร์เน็ตได้

↓

ทำให้มนุษย์สามารถสั่งการ
ควบคุมการใช้งานของอุปกรณ์นั้น ๆ
ผ่านทางเครือข่ายอินเตอร์เน็ตได้

→

เช่น การเปิด–ปิดอุปกรณ์ไฟฟ้า
ภายในบ้านผ่านทางสมาร์ตโฟน

ตัวอย่าง : อุปกรณ์ IoT ในชีวิตประจำวัน

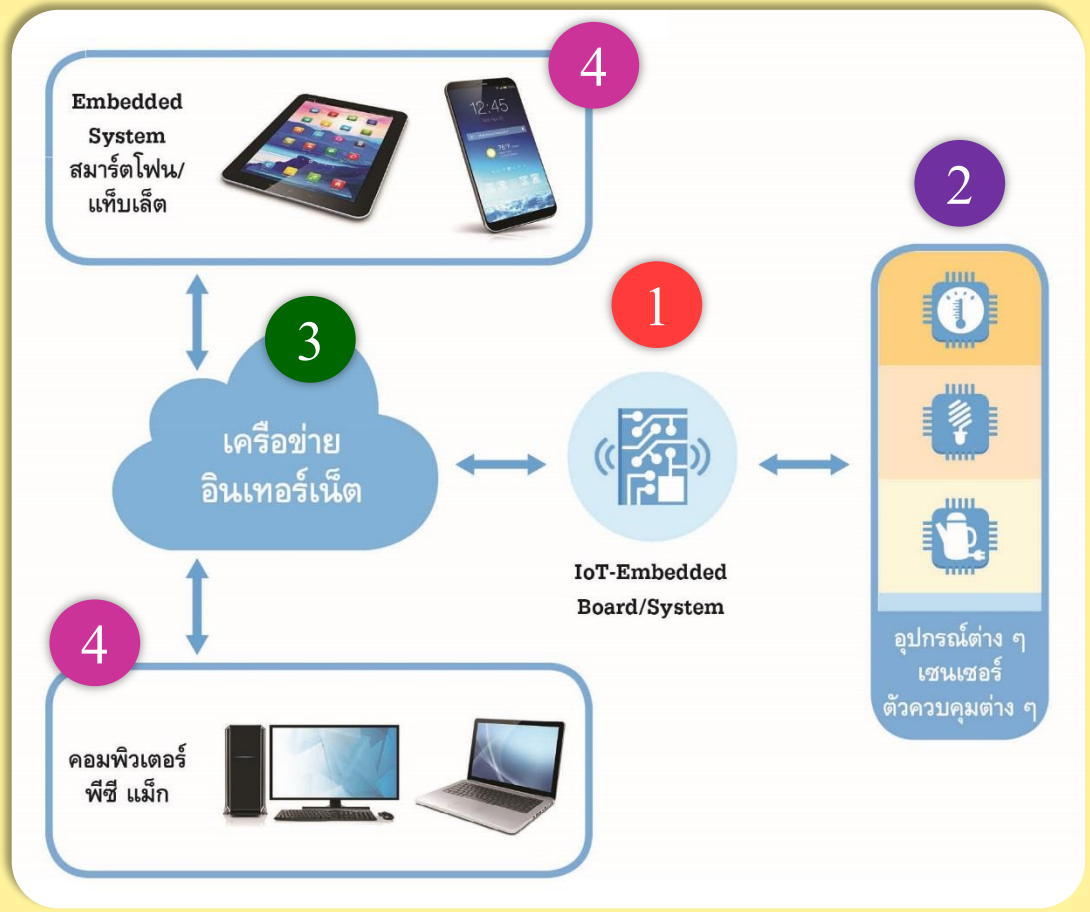
นาฬิกาอัจฉริยะ (Smart Watch)

หลอดไฟฟ้าที่สามารถควบคุมด้วยสมาร์ตโฟน

ตู้เย็นอัจฉริยะ

1. องค์ประกอบของเทคโนโลยี IoT

องค์ประกอบหลักของเทคโนโลยี IoT นั้นมีด้วยกัน 4 องค์ประกอบ ได้แก่



1. IoT บอร์ด
   หรือ Embedded Board/System
   คือ คอมพิวเตอร์ขนาดเล็ก
2. อุปกรณ์เชื่อมต่อ เช่น เซนเซอร์
   อุปกรณ์ Input และ Output
   เช่น สวิตช์ อุปกรณ์สะพานไฟ (Relay)
3. เครือข่ายอินเทอร์เน็ต
4. อุปกรณ์สื่อสาร หรือคอมพิวเตอร์
   หรือ Embedded System อื่น ๆ
   เช่น สมาร์ตโฟน โน้ตบุ๊ก แท็บเล็ต

2. อุปกรณ์สำหรับสร้างเทคโนโลยี IoT

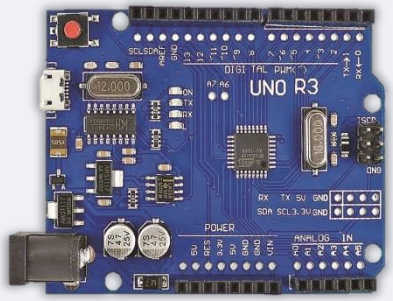
2.1 วงจรสมองกลฝังตัว (Embedded System) → มีลักษณะคล้ายกับคอมพิวเตอร์ คือ มีหน่วยรับข้อมูล หน่วยประมวลผล หน่วยความจำ และหน่วยแสดงผล

หน่วยประมวลผลหรือซีพียู

- ไมโครโพรเซสเซอร์ (Microprocessor: MP)
  - ต้องเชื่อมต่อกับหน่วยความจำภายนอก (Random Access Memory: RAM) เพื่ออ่านโปรแกรมสั่งงานและการเก็บข้อมูลก่อนหรือหลังประมวลผล
- ไมโครคอนโทรลเลอร์ (Microcontroller: MCU)
  - มีหน่วยความจำภายในตัว สามารถเขียนโปรแกรมลงบนตัวไมโครคอนโทรลเลอร์และประมวลผลในตัวของมันเองได้ มีพอร์ตหรือช่องสัญญาณเชื่อมต่อกับอุปกรณ์หรือวงจรภายนอก

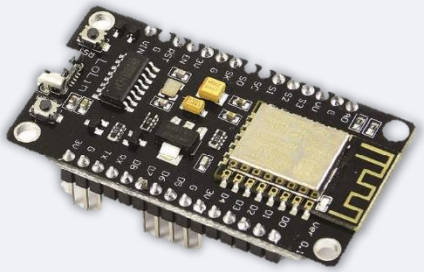
ตัวอย่าง : วงจรสมองกลฝังตัวที่สามารถนำมาใช้พัฒนาระบบ IoT

Arduino



Arduino อ่านว่า “อาดุยโน” เป็นวงจรที่ได้รับความนิยม เพราะเป็นระบบแบบ Open Source เหมาะสำหรับผู้เริ่มต้นศึกษา

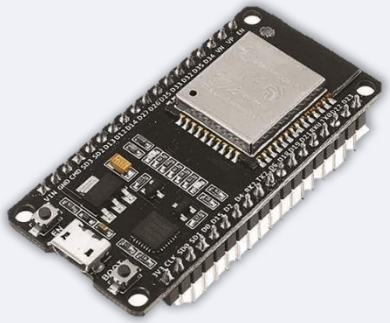
NodeMCU



สามารถใช้โปรแกรม Arduino IDE ทำงานร่วมกับ NodeMCU ได้ จึงทำให้ใช้ภาษา C/C++ ในการเขียนโปรแกรมได้ ทำให้สามารถใช้งานได้หลากหลายมากยิ่งขึ้น

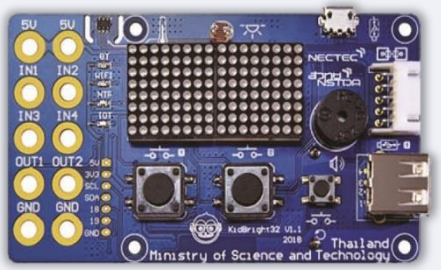
เทคโนโลยี (วิทยาการคำนวณ) ชั้นมัธยมศึกษาปีที่ 3

ESP32



สามารถในการเชื่อมต่อระบบเครือข่ายแบบไร้สายและบลูทูท รองรับการเขียนโปรแกรมแบบ Arduino และใช้ภาษา C ในการเขียนคำสั่งควบคุมได้

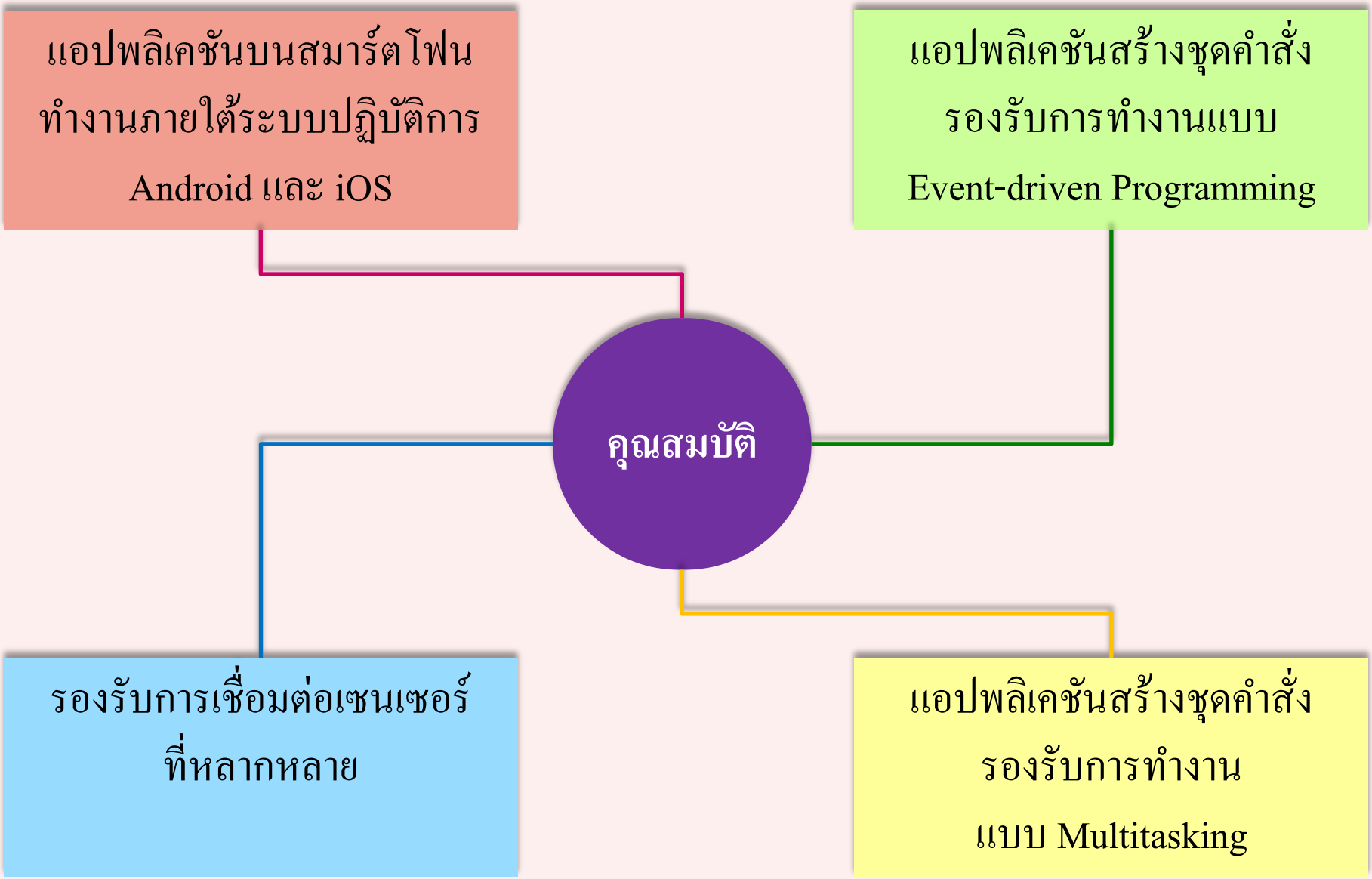
KidBright



มีการแสดงผลและเซนเซอร์แบบง่าย สามารถออกแบบและสร้างชุดคำสั่งแบบบล็อก (Block - structured Programming) ผ่านแอปพลิเคชันบนสมาร์ตโฟนได้

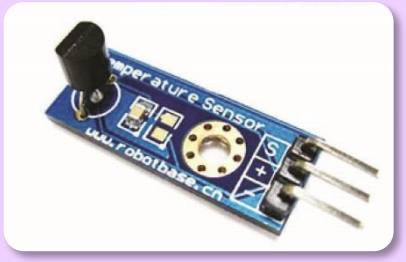
จุดเด่นของวงจรสมองกลฝังตัว

- บอร์ดวงจรประกอบด้วย
  เซนเซอร์พื้นฐาน → จอแสดงผล → Real-time Clock → ลำโพง
  สามารถนำไปประยุกต์ใช้งานได้หลากหลาย
- สร้างชุดคำสั่งแบบ Block-structured Programming
  ผ่านแอปพลิเคชันบนสมาร์ตโฟน
- ชุดคำสั่งถูกส่งไปยังบอร์ดวงจรสมองกลฝังตัว ผ่านเครือข่ายไร้สาย
  ทำให้ใช้งานได้ง่ายไม่จำเป็นต้องเชื่อมต่อสาย

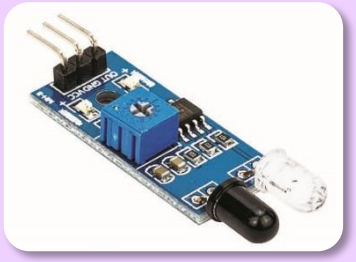

แอปพลิเคชันบนสมาร์ตโฟน
ทำงานภายใต้ระบบปฏิบัติการ
Android และ iOS
แอปพลิเคชันสร้างชุดคำสั่ง
รองรับการทำงานแบบ
Event-driven Programming
คุณสมบัติ
รองรับการเชื่อมต่อเซนเซอร์
ที่หลากหลาย
แอปพลิเคชันสร้างชุดคำสั่ง
รองรับการทำงาน
แบบ Multitasking

2.2 โมดูล หรือวงจรอิเล็กทรอนิกส์ → ทำหน้าที่อ่านค่าจากสภาพแวดล้อมภายนอก หรือแสดงผลลัพธ์ต่าง ๆ ตามที่ต้องการ

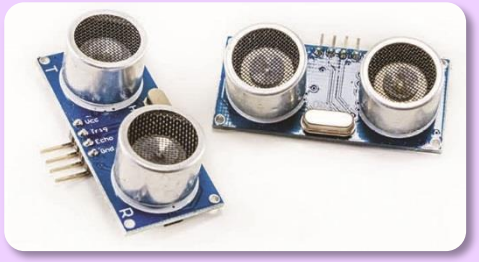
1) เซนเซอร์ มีหลายประเภท เช่น



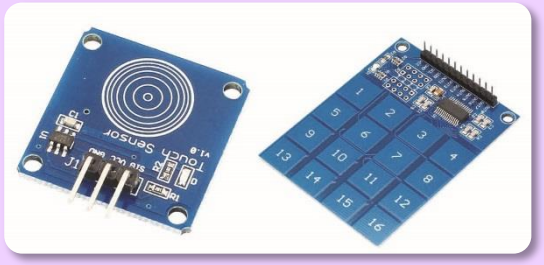
อุณหภูมิ ความร้อน



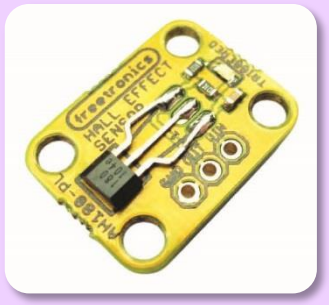
คลื่นอินฟราเรด



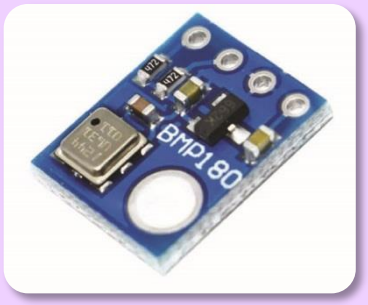
คลื่นอัลตราโซนิก



การสัมผัส

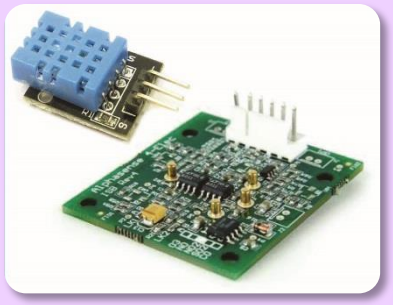


ตรวจจับวัตถุ (โลหะ)

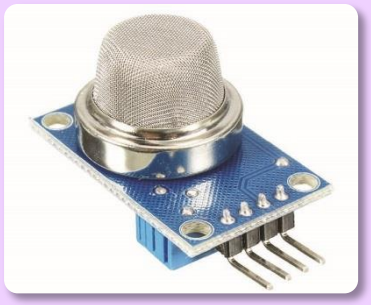


ความดัน

ระดับน้ำ



ความชื้น และสภาพแวดล้อม



ควันและแก๊ส

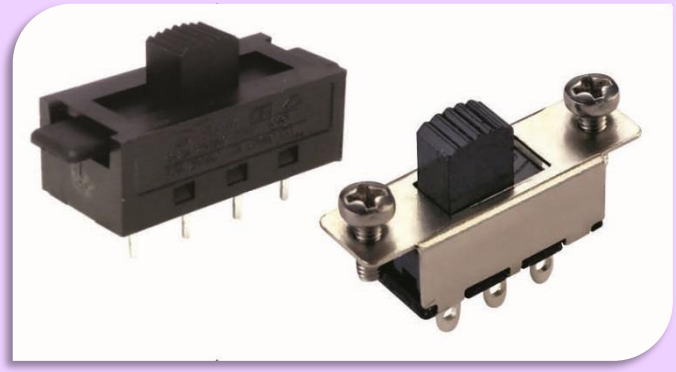
2) ปุ่มหรือสวิตช์แบบกลไก

มีหลายประเภท เช่น

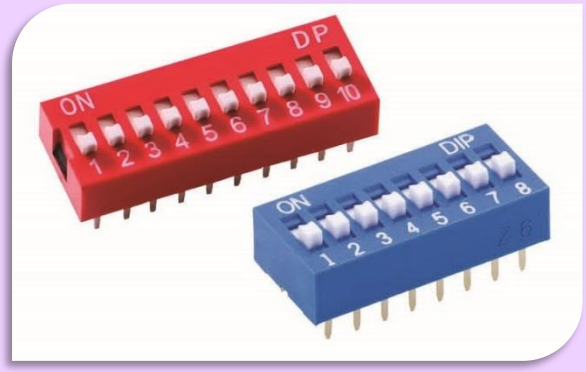
สวิตช์กดติดปล่อยดับ

สวิตช์เปิด-ปิด

สวิตช์แบบหมุน



สวิตช์เลื่อน

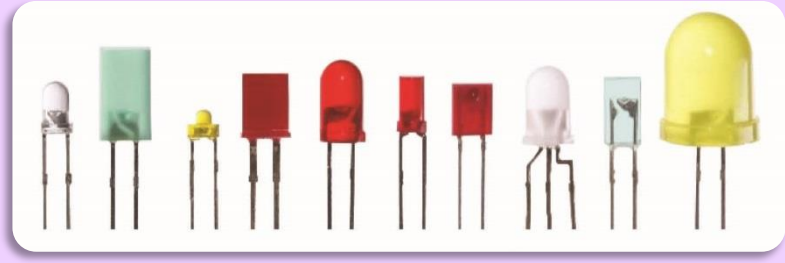


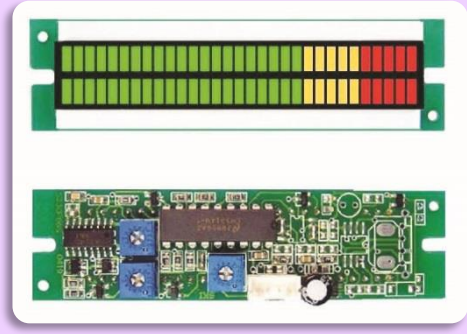
ดิปสวิตช์

สวิตช์จุดชนวน

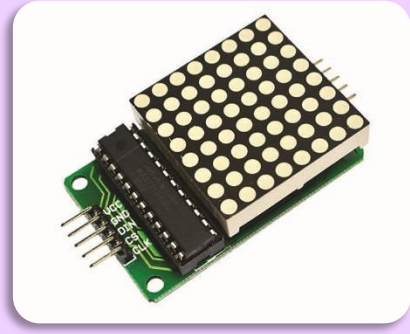
3) หลอดไฟฟ้าสำหรับงานอิเล็กทรอนิกส์และดิจิทัล

มีหลายประเภท เช่น

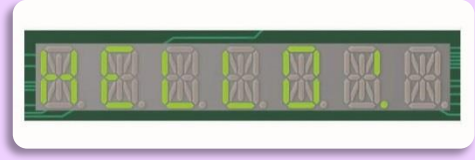




LED แบบบาร์ หรือแบบ VU meter ใช้แสดงค่าในรูปแบบของระดับต่าง ๆ จากน้อยไปมาก



LED แบบเมทริกซ์ ใช้แสดงผลเป็นตัวอักษรหรือตัวเลข โดยกำหนดการติดของดวงไฟได้อิสระ



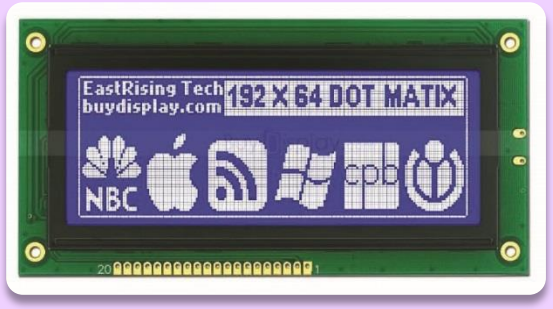
LED แบบ 7 ส่วน (7 segment) ใช้ในการแสดงค่าตัวเลข 0-9 ในแต่ละหลักและจุดทศนิยมได้

LED แบบ 14 ส่วน (14 segment) ใช้สำหรับการแสดงตัวอักษรและตัวเลข

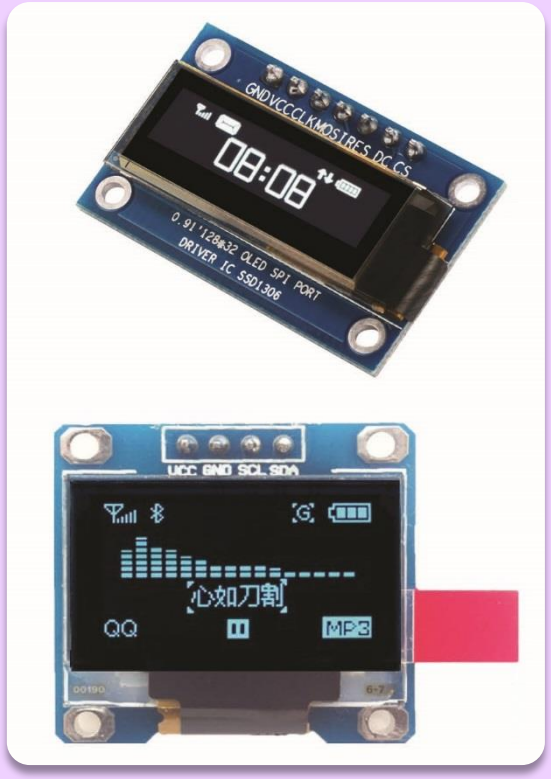
4) หน้าจอแสดงผลขนาดเล็ก

มีหลายประเภท เช่น

แอลซีดีดิสเพลย์โมดูล โดยส่วนใหญ่จะไม่สามารถแสดงตัวอักษรภาษาไทยได้



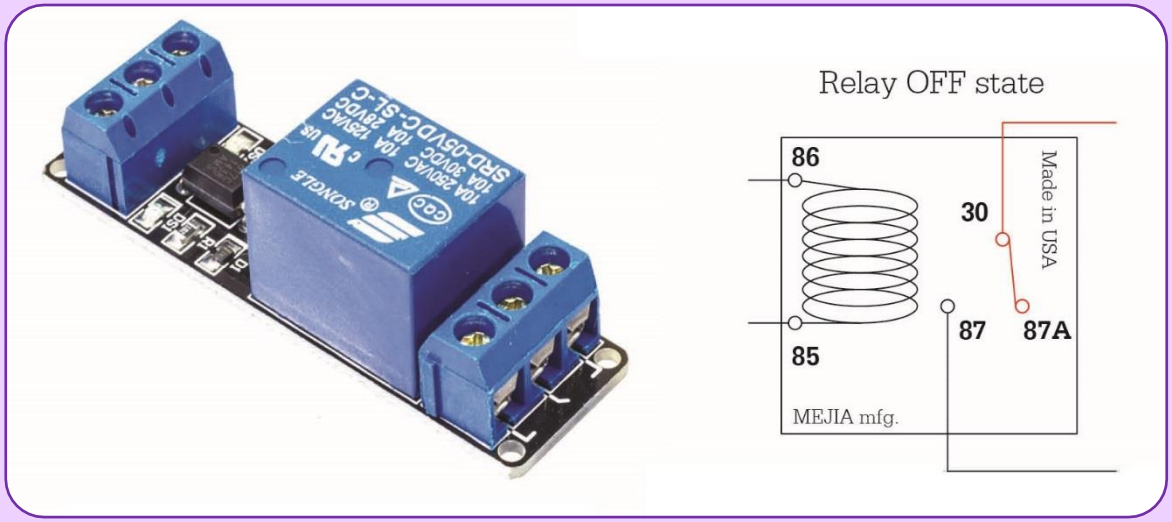
กราฟิกแอลซีดีดิสเพลย์โมดูล แสดงผลเป็นกราฟิกได้ แสดงเป็นตัวอักษรภาษาไทยได้



โอ-แอลอีดีดิสเพลย์โมดูล เป็นการพัฒนามาจาก LED ซึ่งจะให้แสงสว่างด้วยตัวเองโดยไม่ต้องใช้แสงพื้นหลัง ทำให้ได้ความชัดเจนและความสว่างที่คมชัดกว่า

5) วงจรควบคุมเปิด-ปิด

หากต้องการให้บอร์ด IoT นั้นสั่งการให้เปิดหรือปิดไฟส่องสว่างในบ้าน แต่เนื่องจากบอร์ด IoT นั้นใช้ไฟแรงดันต่ำ กระแสตรง ส่วนไฟฟ้าตามบ้านพักอาศัยใช้แรงดัน 220 โวลต์ กระแสสลับ เราจำเป็นต้องใช้วงจรควบคุมที่มีการแยกภาคสวิตช์หรือสะพานไฟที่แยกกันโดยการใช้อุปกรณ์ที่เรียกว่า รีเลย์ (Relay) มีหลายประเภท เช่น



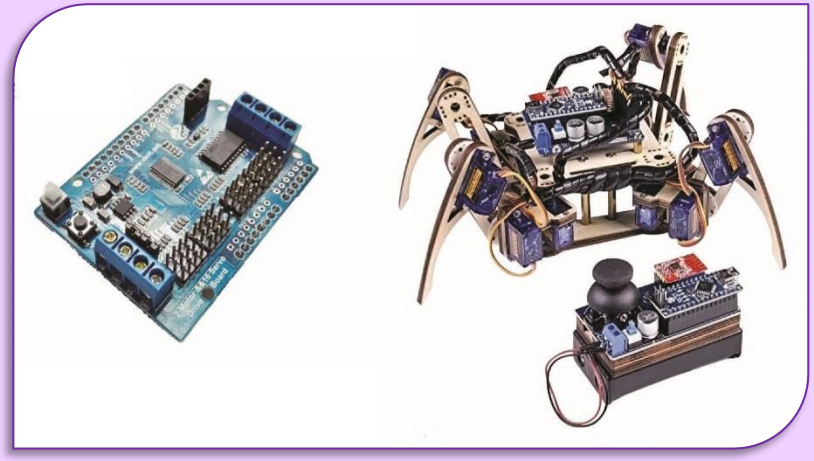
- รีเลย์ชนิดแม่เหล็ก หลักการ คือ การสั่งงานให้บอร์ด IoT จ่ายไฟกับวงจรรีเลย์ ตัวรีเลย์แบบแม่เหล็กนั้นเมื่อมีกระแสป้อนเข้าสู่ตัวรีเลย์ สนามแม่เหล็กจะกำเนิดขึ้นและมีการดูดสะพานไฟมาเชื่อมกัน ดังนั้น เราก็สามารถนำแหล่งจ่ายไฟอีกชุดมาเชื่อมต่อที่สะพานไฟนั้น

- รีเลย์ชนิดโซลิดสเตต (Solid State Relay) หลักการคล้ายกับรีเลย์แบบแม่เหล็ก แต่จะไม่มีกลไกใด ๆ การเปิดสะพานไฟนั้นจะใช้คุณสมบัติของปฏิกิริยาสารกึ่งตัวนำ โดยใช้ไฟแรงดันต่ำป้อนเพื่อให้สะพานไฟภายในทำงาน

6) โมดูลควบคุมเซอร์โวมอเตอร์

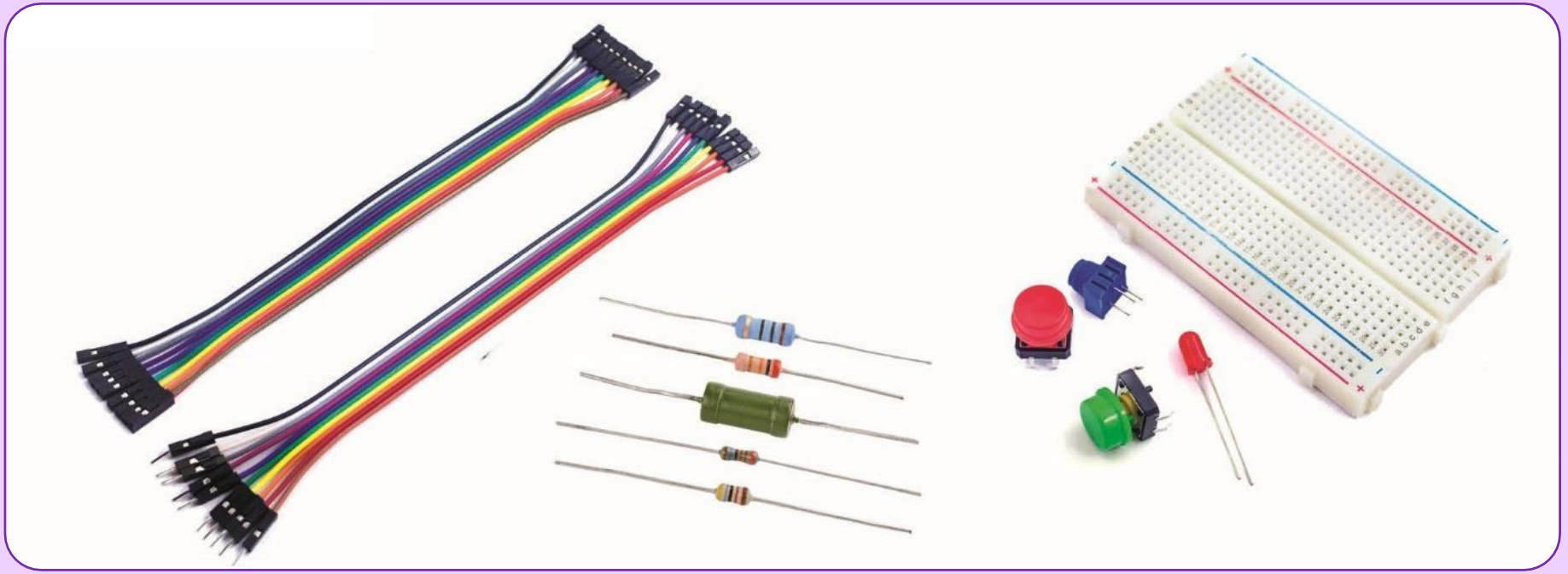
ใช้สำหรับควบคุมการหมุนของมอเตอร์แบบกำหนดองศาการหมุนได้ ซึ่งจะเหมาะสมกับการบังคับการเลี้ยว บังคับหางเสือเรือบังคับ หรือปีกและแพนหาง เครื่องบินเล็ก และบังคับการเคลื่อนไหวของแขนหุ่นยนต์หรือตัวหุ่นยนต์ ซึ่งวงจรนี้จะมีหลายช่องควบคุมให้ใช้หลายมอเตอร์

โมดูลควบคุมเซอร์โวมอเตอร์ที่ใช้ควบคุมการหมุนของหุ่นยนต์



7) โพรโทบอร์ด ตัวต้านทานขนาดต่าง ๆ และสายไฟฟ้าพร้อมขั้วแบบตัวผู้-ตัวเมีย ตัวผู้-ตัวผู้ ตัวเมีย-ตัวเมีย

โพรโทบอร์ด และตัวต้านทานขนาดต่าง ๆ



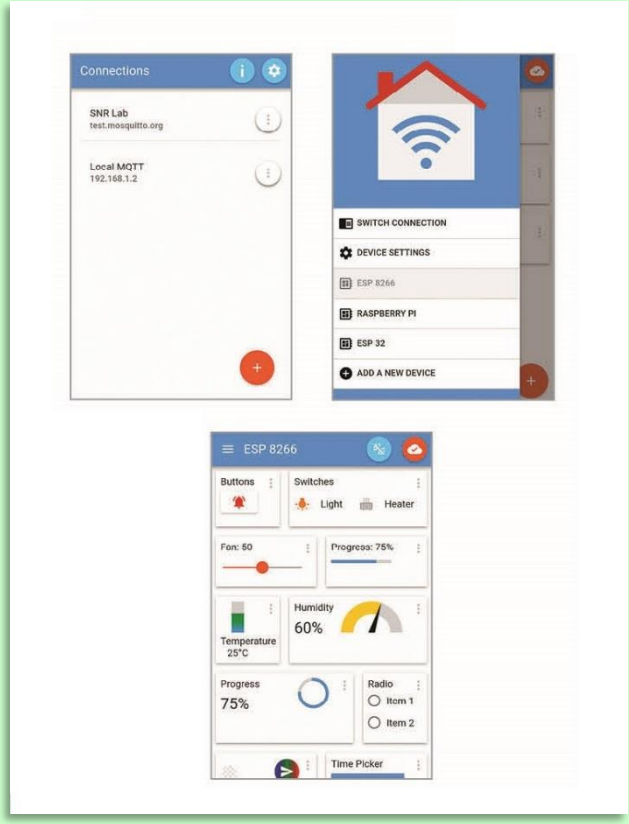
8) เครื่องมือวัดค่าทางไฟฟ้า

เช่น มัลติมิเตอร์ หรือออสซิลโลสโคปเพื่อวัดรูปแบบสัญญาณไฟฟ้า

แอปพลิเคชันกับ Internet of Things

แอปพลิเคชันที่ให้บริการด้าน IoT ในปัจจุบันมีอยู่ด้วยกันหลายแอปพลิเคชัน แต่แอปพลิเคชันที่ได้รับความนิยมและถูกนำมาใช้ในด้านการศึกษาและพัฒนาระบบ IoT มีดังนี้

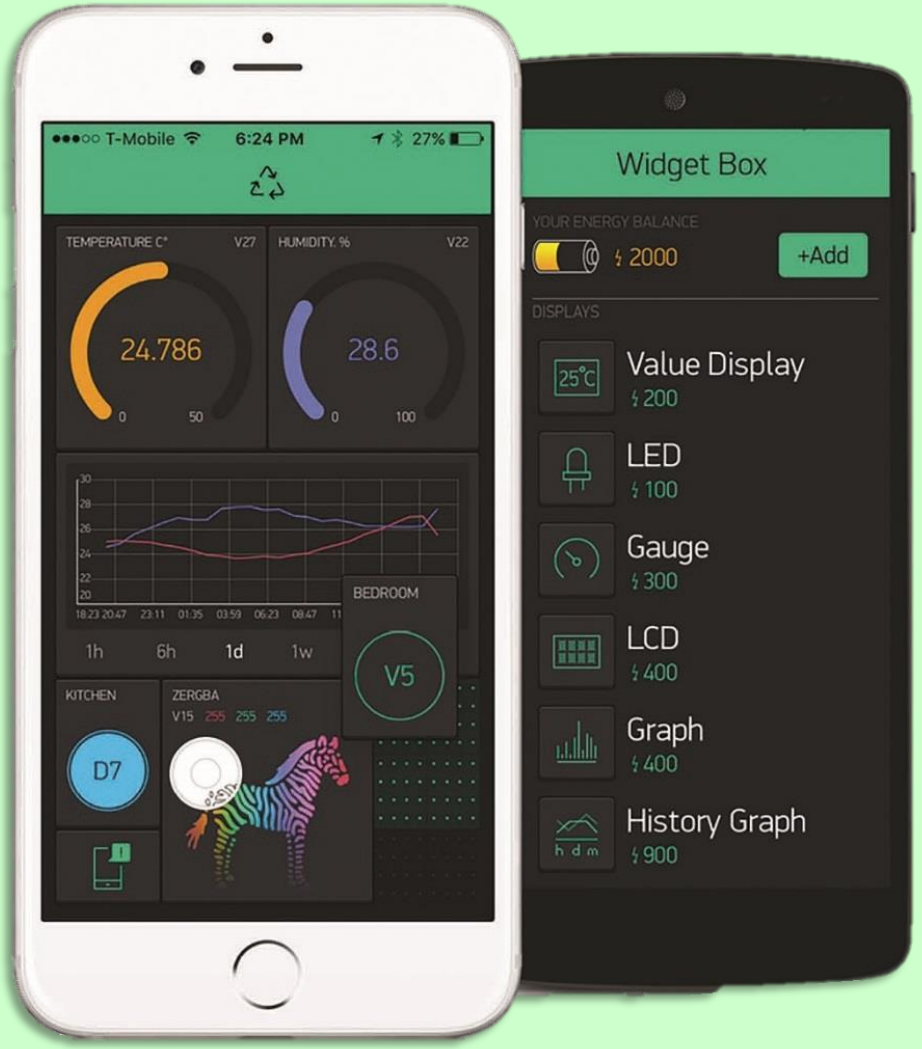
1. MQTT



MQTT จะมีหลักการในลักษณะที่มีตัวกลางในการแลกเปลี่ยนข้อมูล รับและส่งข้อมูลจากอุปกรณ์ต่าง ๆ ตัวกลางจะทำหน้าที่จัดการให้อัตโนมัติ ตัวกลางที่ว่านั้นจะเรียกว่า MQTT Broker ซึ่งหลักการ MQTT จะเป็นลักษณะที่เป็นการสมัคร รับหัวข้อข่าวและการตีพิมพ์ข่าวสารในหัวข้อนั้น ๆ ซึ่งกระทำได้โดยไม่มีข้อจำกัด

2. Blynk

Blynk คือ แอปพลิเคชันที่ถูกพัฒนาขึ้นมาเพื่ออำนวยความสะดวกในด้านการพัฒนางาน IoT ลักษณะสำคัญ คือ สามารถเขียนโปรแกรมได้ง่าย สามารถนำมาใช้งานเพื่อแสดงผลข้อมูลได้แบบทันทีทันใด (Realtime) อีกทั้งยังสามารถเชื่อมต่อหรือใช้งานกับอุปกรณ์สมองกลฝังตัวได้หลากหลายประเภท ทั้ง Arduino, ESP8266, ESP32 หรือ Raspberry Pi



3. NETPIE

NETPIE คือ แอปพลิเคชันที่ถูกออกแบบมาเพื่อทำงานด้าน IoT เช่นเดียวกันกับ MQTT และ Blynk พัฒนาโดยศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (NECTEC) จึงถือได้ว่าเป็นระบบของคนไทยที่ให้บริการในด้านงาน IoT ระบบแรก ซึ่งสามารถทำงานร่วมกับวงจรสมองกลฝังตัวได้หลายชนิด ทั้ง Arduino, ESP8266, ESP32 หรือ Respberry Pi

4. kidBright IoT

KidBright IoT คือ แอปพลิเคชันที่ใช้ระบบภายในเชื่อมโยงกับ NETPIE แต่ KidBright IoT สามารถทำงานร่วมกับวงจรสมองกลฝังตัว KidBright เท่านั้น มีรูปแบบการใช้งานที่ง่าย เพราะสามารถเขียนคำสั่งได้จาก KidBright IDE ในรูปแบบของบล็อกคำสั่ง แต่ก็มีข้อเสียที่สำคัญ คือ ผู้ใช้ไม่สามารถปรับแต่งหน้าจอแสดงผลของแอปพลิเคชันนี้ได้ หรือไม่สามารถเพิ่มฟังก์ชันการทำงานอื่น ๆ ได้

การพัฒนาระบบ IoT เบื้องต้น

ตัวอย่างที่ 1 การพัฒนาระบบเปิด-ปิดไฟด้วยสมาร์ตโฟน

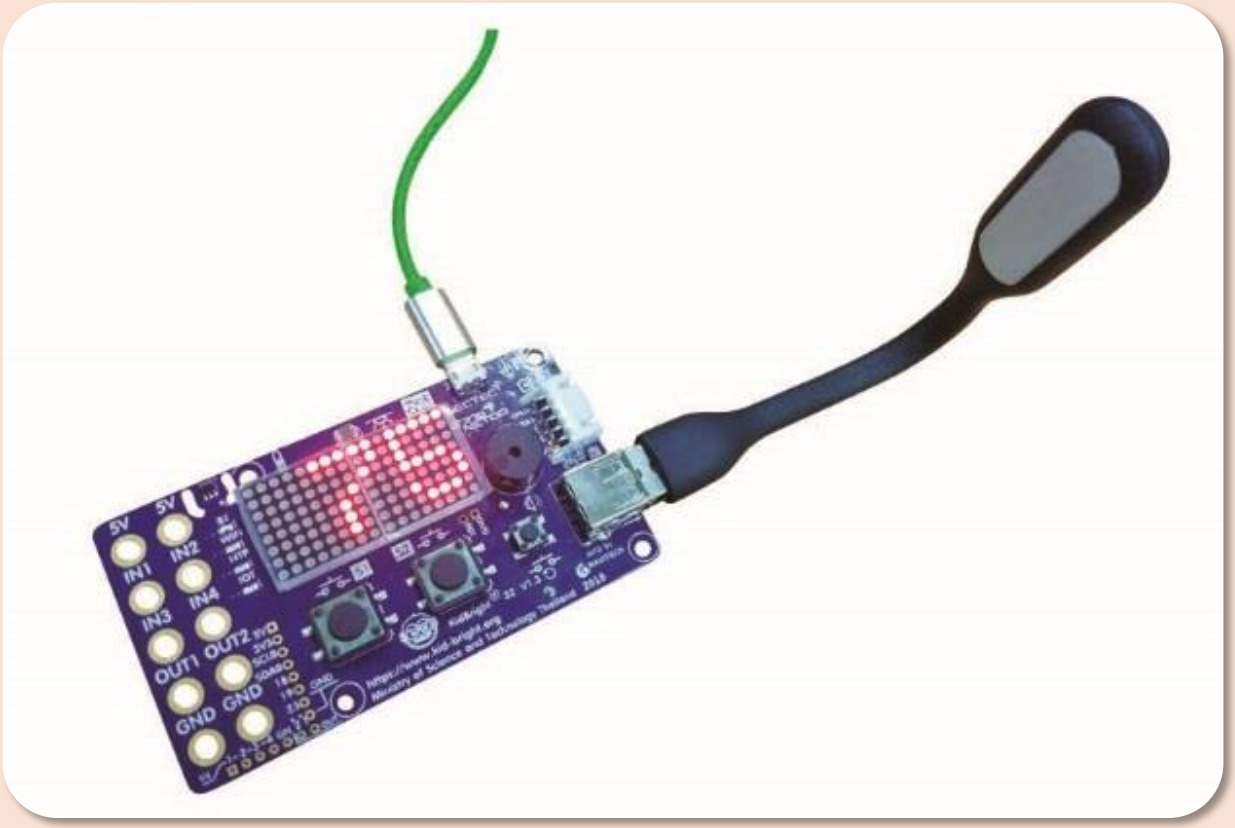
ระบบเปิด-ปิดไฟด้วยสมาร์ตโฟน คือ ระบบที่สามารถควบคุมวงจรไฟฟ้าผ่านสมาร์ตโฟนได้โดยที่ผู้ใช้งานจะสั่งงานผ่านสมาร์ตโฟน ซึ่งเชื่อมต่อกับระบบเครือข่ายและส่งสัญญาณควบคุมไปยังวงจรสมองกลฝังตัว ซึ่งในวงจรสมองกลฝังตัวจะมีคำสั่งที่ทำหน้าที่ตรวจสอบและรับข้อมูลที่ส่งมาจากสมาร์ตโฟน และสั่งงานให้ระบบต่าง ๆ ทำงานตามที่ผู้ใช้สั่งงาน

เลือกใช้

- บอร์ดวงจรสมองกลฝังตัว KidBright
- แอปพลิเคชัน KidBright IoT

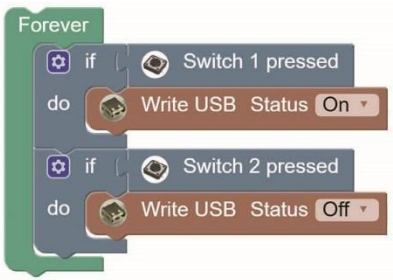
ขั้นตอนการพัฒนาระบบเปิด-ปิดไฟด้วยสมาร์ตโฟน

1. เชื่อมต่อบอร์ด KidBright เข้ากับหลอดไฟฟ้า USB

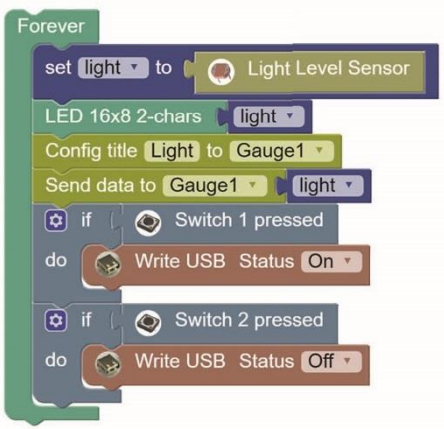


2. เขียนคำสั่งควบคุมด้วยโปรแกรม KidBright IDE บนคอมพิวเตอร์ สามารถเขียนได้ 3 รูปแบบ

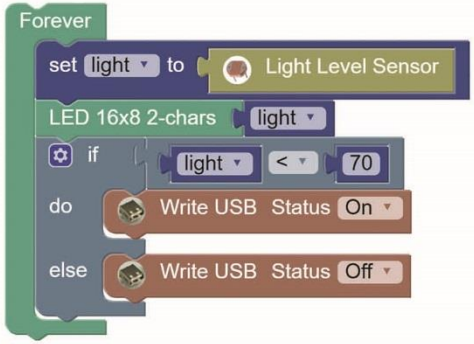
รูปแบบที่ 1



รูปแบบที่ 2



รูปแบบที่ 3



3. ตั้งค่าการเชื่อมต่อระบบเครือข่าย โดยคลิกที่ แล้วตั้งค่าการเชื่อมต่อโดย

SSID คือ ชื่อสัญญาณเครือข่ายไร้สาย

Password คือ รหัสผ่านของสัญญาณเครือข่ายไร้สาย

จากนั้นกดปุ่ม OK

4. เชื่อมต่อ KidBright เข้ากับคอมพิวเตอร์ โดยใช้สาย USB และทำการดาวน์โหลดโปรแกรมเข้าสู่บอร์ด KidBright ด้วยการกดปุ่ม

5. กดปุ่ม เพื่อแสดง QR Code สำหรับเชื่อมต่อกับสมาร์ตโฟน

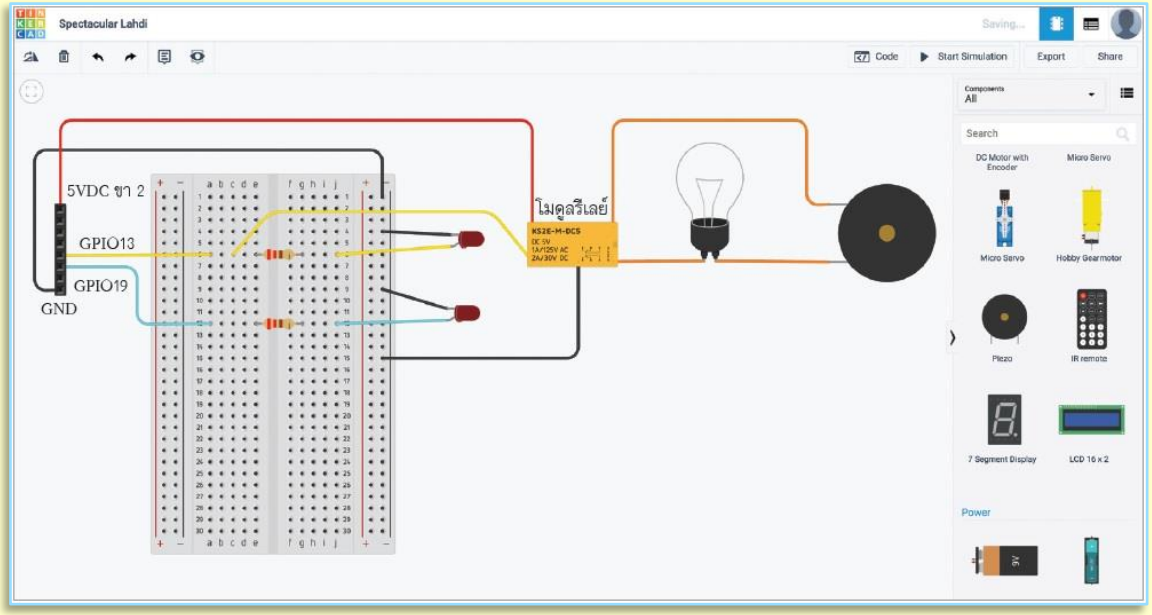
6. ในส่วนของสมาร์ตโฟนให้ทำการดาวน์โหลดแอปพลิเคชัน KidBright IoT แล้วทำการเปิดแอปพลิเคชัน กดปุ่มสแกน QR Code จากนั้นสแกน QR Code ที่เปิดไว้ในข้อ 5

7. ทดลองกดปุ่มสวิตช์ 1 และ 2 แล้วสังเกตการแสดงผลของหลอดไฟฟ้า USB

ตัวอย่างที่ 2 การใช้ MQTT และ Raspberry Pi 3 ในการพัฒนาระบบ IoT

1. การเชื่อมต่อวงจร

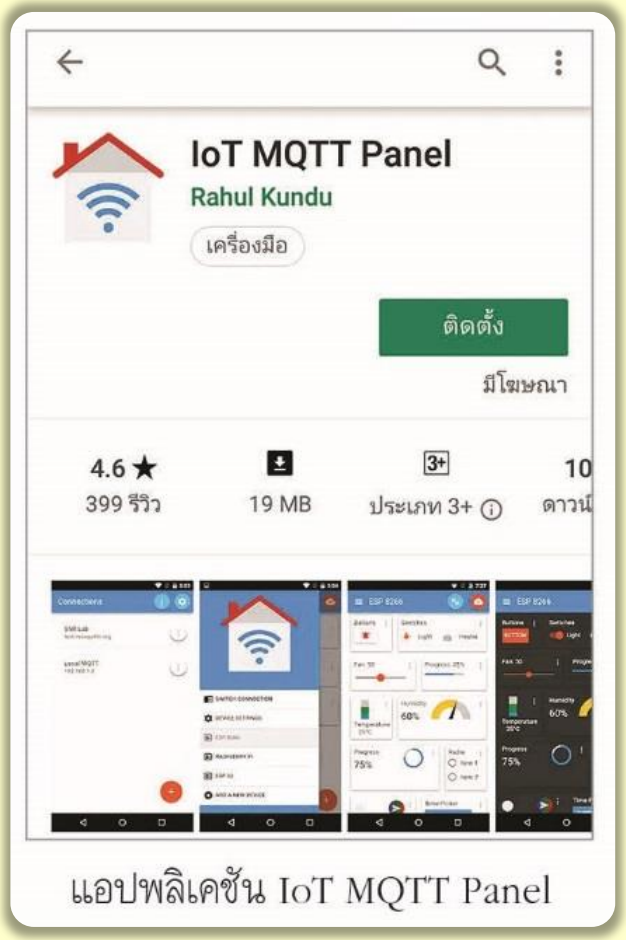
โปรแกรมต่อไปนี้จะสั่งงานให้รีเลย์ทำงาน (เปิด) ในช่วงเวลาหลัง 20.00 น. และหยุดทำงาน (ปิด) ในช่วงเวลาหลัง 22.00 น. และมีไฟกะพริบ (ดวงที่ 2) ทุก ๆ วินาที เพื่อแสดงว่าในขณะนั้นโปรแกรมกำลังทำงานอยู่



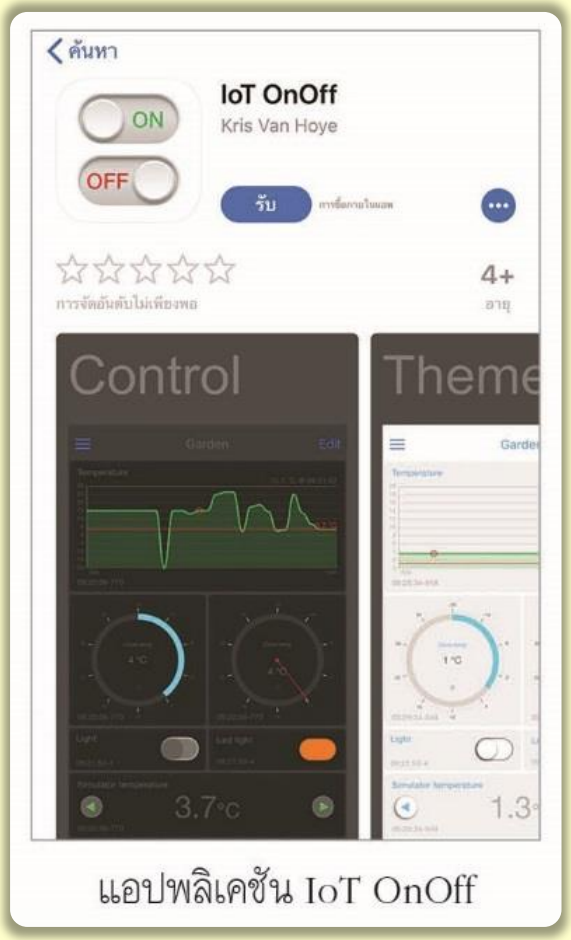
การเชื่อมต่อวงจรจะเชื่อมต่อสายไฟฟ้ากับโมดูลรีเลย์จากวงจร LED ดวงที่ 1 เพื่อมาใช้สั่งงานรีเลย์ และเพื่อให้ทราบว่ารีเลย์นั้นทำงานหรือไม่ หากทำงาน LED ดวงที่ 1 จะติดค้างไว้

2. การออกแบบหน้าจอด้วย MQTT

ออกแบบหน้าจอบนสมาร์ตโฟนโดยผ่านแอปพลิเคชันที่รองรับ MQTT ได้แก่ แอปพลิเคชัน IoT MQTT Panel (Android) หรือแอปพลิเคชัน IoT OnOff (iOS)



แอปพลิเคชัน IoT MQTT Panel



แอปพลิเคชัน IoT OnOff

3. การเขียนโปรแกรมควบคุมโดยใช้ Raspberry Pi 3

การเขียนโปรแกรมต่อไปนี้จะต้องให้บอร์ดเชื่อมกับเครือข่ายอินเทอร์เน็ต และสมาร์ตโฟนต้องผ่านเครือข่ายด้วยจึงจะทำงานได้อย่างสมบูรณ์

การเขียนโปรแกรมไพธอนกับบอร์ด Raspberry Pi 3 ให้ไฟติด-ดับ

ก่อนอื่นนักเรียนจะต้องลงโมดูลชุดคำสั่ง MQTT บน Raspberry Pi 3 ด้วยการเข้าที่ Terminal แล้วป้อนคำสั่ง sudo pip install paho-mqtt ระบบจะติดตั้งโมดูลชุดคำสั่ง MQTT ของ Paho ให้เรียกใช้ในการเขียนโปรแกรมได้

เข้าสู่โปรแกรม Geany และดูคำอธิบายเพื่อทำความเข้าใจ โดยการทำงานของโปรแกรมนี้จะสั่งงานให้รีเลย์และไฟดวงที่ 2 ติด 5 วินาที และดับ 5 วินาทีตามเวลาจริงของระบบ โดยใช้ชุดคำสั่ง datetime

```
1  #!/usr/local/bin/python
2  # -*- coding:utf-8 -*-
3  import paho.mqtt.client as mqtt
4  import RPi.GPIO as GPIO
5  from datetime import datetime
6
7  LED1status = "Off"
8  LED2status = "Off"
9  prefix = "/Thailand/M3"
10
11 def topic(sensor):
12     return (prefix+sensor)
13
```

เป็นการนำเข้าชุดคำสั่ง paho.mqtt.client ภายใต้นามของตัวแปร mqtt

เป็นการนำเข้าชุดคำสั่งที่เกี่ยวกับ GPIO และเวลาจริงของบอร์ด

เป็นการตั้งค่าตัวแปร สถานะของ LED ทั้ง 2 ตัว ให้อยู่ในสถานะปิด (Off) ตัวแปร prefix นั้นคือ คำนำหน้าหัวข้อข่าว เพื่อไปเชื่อมต่อกับชื่อของหัวข้อท้าย

def topic (sensor) เป็นการเขียนฟังก์ชันเพื่อเรียกใช้ในโปรแกรม

การเขียนตัวอักษร

def topic (sensor) เป็นการเขียนฟังก์ชันเพื่อเรียกใช้ในโปรแกรม เนื่องจากเวลา Publish จะต้องนำค่าหัวข้อย่อยของข่าว เช่น LED1, Relaystatus มาเชื่อมกันกับ prefix ทุกครั้งก่อน Publish หรือ Subscribe ซึ่งสามารถสังเกตได้ในโปรแกรมช่วงถัดไป ดังภาพ

```
14  ########################################
15  def on_message(client, userdata, message):
16      global LED1status
17      AppPayload = str(message.payload.decode("utf-
18      AppTopic = message.topic
19                                          1)
20
21                                  Payload == "pushed":
22
23              GPIO.output(LED1,1)
24              print "LED1 On"
25              LED1status = "On"
26          else:
27              GPIO.output(LED1,0)
28              print "LED1 Off"
29              LED1status = "Off"
30  ########################################
```

เป็นเทคนิคการสร้างฟังก์ชันภายในโปรแกรม

ประกาศตัวแปรภายนอก คือ LED1status เพราะจะมีการเรียกใช้ในบรรทัดที่ 22 25 และ 29

เป็นการแปลงข้อความที่ได้จากการ Subscribe ไว้
เพื่อดึงและแปลงเป็น Payload เก็บไว้

```
14  ##################################
15  def on_message(client, userdata, message):
16      global LED1status
17      AppPayload = str(message.payload.decode("utf-8"))
18      AppTopic = message.topic
19      print ("message received:",AppPayload)
20      print ("message topic=",AppTopic)
21      if AppTopic == topic("LED1") and AppPayload == "pushed":
22          if LED1status == "Off":
23              GPIO.output(LED1,1)
24
25
26
27
28
29              LED1status = "Off"
30  ##################################
```

เป็นการพิมพ์ข้อมูลภาระข่าวที่ได้และหัวข้อข่าวที่รับมา
ซึ่งจะพิมพ์ผลแสดงออกมาที่จอ Terminal ของบอร์ด
Raspberry Pi 3

เป็นการตรวจสอบตัวแปร AppTopic ส่วนบรรทัดที่ 22-29 นั้นจะเป็นเทคนิคการทำให้เมื่อมีการกดปุ่มโดยส่ง pushed มาจะทำให้ไฟดวงที่ 1 ติด โดยการใช้ตัวแปร

```
14  ######
15  def on
16      global LED1status
17      AppPayload = str(message.payload.decode("utf-8"))
18      AppTopic = message.topic
19      print ("message received:",AppPayload)
20      print ("message topic=",AppTopic)
21      if AppTopic == topic("LED1") and AppPayload == "pushed":
22          if LED1status == "Off":
23              GPIO.output(LED1,1)
24              print "LED1 On"
25              LED1status = "On"
26          else:
27              GPIO.output(LED1,0)
28              print "LED1 Off"
29              LED1status = "Off"
30  ##########################################
```

การนำ LED1status มาช่วยในการจดจำว่าสถานะติดหรือดับอยู่ โดยเมื่อสั่งให้ LED1 ติด ก็เก็บค่าตัวแปร LED1status เป็น On และหากมีการ Publish LED1 มาจากสมาร์ตโฟนเป็น pushed มาอีกครั้ง ก็จะเข้าสู่ else เพราะสถานะ LED1status เป็น On ก็จะถูกสั่งการให้ LED1 ดับ และเก็บค่าสถานะใหม่ LED1status เป็น Off ความหมายคือ กดปุ่มครั้งหนึ่งจะปิด กดปุ่มอีกครั้งจะเปิด เช่นนี้ เราเรียกลักษณะเทคนิคนี้ว่า Toggle (ท็อกเกิล)

```
31  LED1 = 13 #เป็นทั้งไฟ LED และรีเลย์
32  LED2 = 19 #BOARD 35
33  # GPIO.setwarnings(False)
34  GPIO.setmode(GPIO.BCM)   # GPIO.BOARD for Real Pin mode
35  GPIO.setup(LED1, GPIO.OUT)
36  GPIO.setup(LED2, GPIO.OUT)
37
38  broker_address = "test.mosquitto.org"
39  # Unique ID
40  client = mqtt.Client("WorawitIsr",transport="websockets")
41
42  #เวลาใช้งานคำสั่ง Subscribe จะเกี่ยวข้องกับการใช้เทคนิค callback
43  client.on_message=on_message      #def on_message จะเป็นโปรแกรม
44  #เชื่อมต่อกับ Broker ด้วย address ที่กำหนดไว้บรรทัดบน
45  client.connect(broker_address,8080)   #Connect MQTT Broker
46
47  #เมื่อใช้งาน Subscribe จะต้องมีส่วนนี้คือ loop_start และ loop_stop ท้ายโปรแกรม
48  client.loop_start() #Start the loop
49  client.subscribe(topic("LED1"))  #หัวข้อที่ Subscribe ใส่หลัง loop_start
50
```

เป็นการประกาศค่าขา GPIO และโหมดการทำงานของ GPIO

```
31  LED1 = 13 #เป็นทั้งไฟ LED และรีเลย์
32  LED2 = 19 #BOARD 35
33  # GPIO.setwarnings(False)
34  GPIO.setmode(GPIO.BCM)  # GPIO.BOARD for Real Pin mode
35  GPIO.setup(LED1, GPIO.OUT)
36  GPIO.setup(LED2, GPIO.OUT)
37
38  broker_address = "test.mosquitto.org"
39  # Unique ID
40  client = mqtt.Client("WorawitIsr",transport="websockets")
41
42  #เวลาใช้งานคำสั่ง Subscribe จะเกี่ยวข้องกับการใช้เทคนิค callback
43  client.on_message=on_message      # on_message จะเป็นโปรแกรม
44  #เชื่อมต่อกับ Broker ด้วย
45  client.connect MQTT Broker
46
47  #เมื่อใช้งาน Subscribe จะต้องมีคำสั่ง loop_start และ loop_stop ท้ายโปรแกรม
48  client.loop_start() #Start the loop
49  client.subscribe(topic("LED1")) #หัวข้อที่ Subscribe ใส่หลัง loop_start
50
```

เป็นการตั้งค่า Broker URL ซึ่งจะใช้ในบรรทัดที่ 45

จำเป็นต้องเปลี่ยนชื่อ โดยชื่อที่ใช้จะต้องไม่ซ้ำกันกับผู้อื่น

```
31  LED1 = 13 #เป็นทั้งไฟ LED และรีเลย์
32  LED2 = 19 #BOARD 35
33
34
35
36
37
38  broker_address = "test.mosquitto.org"
39  # Unique ID
40  client = mqtt.Client("WorawitIsr",transport="websockets")
41
42  #เวลาใช้งานคำสั่ง Subscribe จะเกี่ยวข้องกับการใช้เทคนิค callback
43  client.on_message=on_message      #def on_message จะเป็นโปรแกรม
44  #เชื่อมต่อกับ Broker ด้วย address ที่กำหนดไว้บรรทัดบน
45  client.connect(broker_address,8080)   #Connect MQTT Broker
46
47  #เมื่อใช้งาน Subscribe จะต้องมีส่วนนี้คือ loop_start และ loop_stop ท้ายโปรแกรม
48  client.loop_start() #Start the loop
49  client.subscribe(topic("LED1")) #หัวข้อที่ Subscribe ใส่หลัง loop_start
50
```

เป็นการตั้งค่าเมื่อมีการ Subscribe และมีข้อความส่งมาด้วย Publish จากผู้ส่งข่าว client.on_message จะกำหนดให้ไปทำงานที่ฟังก์ชัน on_message ที่เขียนไว้แล้วในบรรทัดที่ 15-29

เป็นการเชื่อมต่อกับ Broker ที่ระบุ URL ไว้ในบรรทัดที่ 38 และด้วยพอร์ตที่ 8080 (websockets)

```
31  LED1 = 13 #เป็นทั้งไฟ LED และรีเลย์
32  LED2 = 19 #BOARD 35
33  # GPIO.setwarnings(False)
34  GPIO.setmode(GPIO.BCM)   # GPIO.BOARD for Real Pin mode
35  GPIO.setup(LED1, GPIO.OUT)
36  GPIO.setup(LED2, GPIO.OUT)
37
38  broker_address = "test.mosquitto.org"
39
40
41
42
43
44
45  client.connect(broker_address,8080)   #Connect MQTT Broker
46
47  #เมื่อใช้งาน Subscribe จะต้องมีส่วนนี้คือ loop_start และ loop_stop ท้ายโปรแกรม
48  client.loop_start() #Start the loop
49  client.subscribe(topic("LED1")) #หัวข้อที่ Subscribe ใส่หลัง loop_start
50
```

เป็นบรรทัดที่จำเป็นต้องมีเมื่อมีการ Subscribe เพราะเป็นการสั่งให้มีการตรวจสอบการส่งข้อมูลตลอดเวลาด้วย client.loop_start( ) และหัวข้อข่าวที่สมัครไว้ คือ client.subscribe (topic(“LED1”) ซึ่งจะเป็น “/Thailand/M3/LED1”

การวนลูป

```
51  while True:
52      now = datetime.now()
53      hour = now.strftime("%H")
54      hournumber = int(hour)   #แปลงเป็นตัวแปรที่เป็นตัวเลข
55      sec = now.strftime("%S")
56      secnumber = int(sec)
57      #ไฟจะติดและดับทุก ๆ 5 วินาที โดยการตัดหลักสิบทิ้ง เอาหลักหน่วยมาเปรียบเทียบ
58      #นำ secnumber หาร 10 ปัดเศษทิ้งด้วย int และคูณ 10 กลับไปจะได้หลักสิบที่ไม่มีเศษ
59      #แล้วไปหักลบกับ secnumber เดิม จะได้หลักหน่วยมาใช้
```

เป็นการวนลูปกระทำการแสดงผลไฟติด-ดับของรีเลย์และ LED2

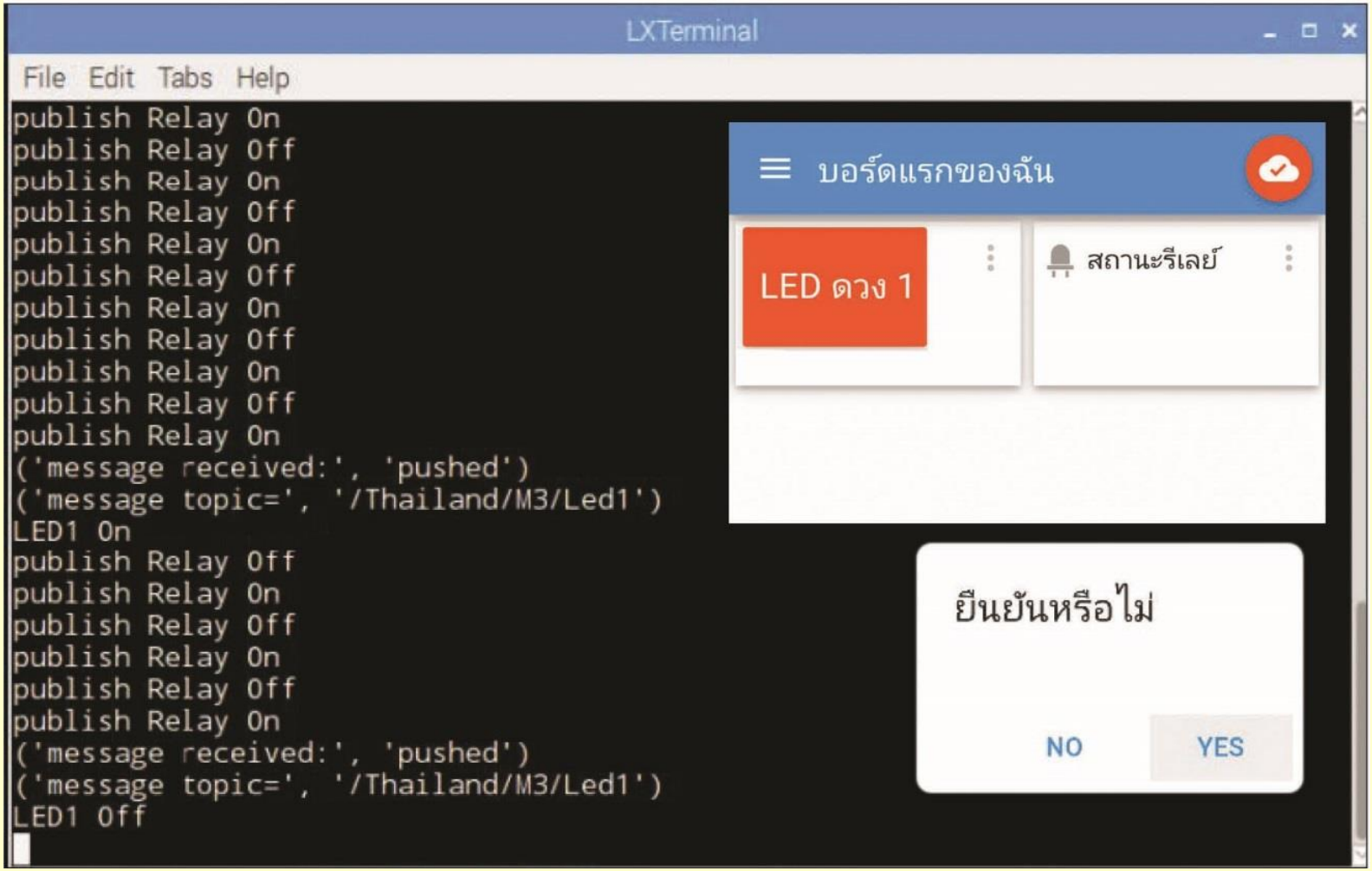
```
60      sec1digit = secnumber - int(secnumber/10)*10
61      if sec1digit <=5:
62          if LED2status == "Off":
63              GPIO.output(LED2,1)
64              client.publish(topic("Relaystatus"),"On")
65              print "publish Relay On"
66              LED2status = "On"
67      else:
68          if LED2status == "On":
69              GPIO.output(LED2,0)
70              client.publish(topic("Relaystatus"),"Off")
71              print "publish Relay Off"
72              LED2status = "Off"
```

จะได้ค่าของหลักหน่วยของวินาที ซึ่งเราจะนำมาใช้ในบล็อกของ if หาก 0-5 ก็จะสั่งการให้ LED2 ติด (ก็คือรีเลย์ด้วย) และตีพิมพ์ Publish หัวข้อข่าวว่า รีเลย์ติด และเก็บค่าสถานะ LED2status (ก็คือสถานะรีเลย์ด้วย) ไว้ว่าเป็น On และหากไม่ใช่เวลา 0-5 ก็สั่งการให้รีเลย์หรือ GPIO LED2 นั้นดับลง และตีพิมพ์หัวข้อข่าวว่า รีเลย์ปิดแล้ว และเก็บค่าสถานะของ LED2status เป็น Off

```
73
74  GPIO.cleanup()
75  client.loop_stop()
76  client.disconnect()
77
```

ภาพ การใช้ชุดคำสั่งเคลียร์พอร์ต GPIO และการปิดลูป

หลังจากบล็อกของ while ในช่วงบรรทัดที่ 51-72 แล้วก็ให้ใส่ชุดคำสั่งเคลียร์พอร์ต GPIO และการปิดลูป client.loop_stop( ) และการสิ้นสุดการเชื่อมต่อกับ Broker ด้วย client.disconnect( ) เป็นการจบโปรแกรมโดยสมบูรณ์



แบบทดสอบ

หน่วยการเรียนรู้ที่ 1

นักเรียนใช้ดินสอระบายลงใน ○ หน้าคำตอบที่ถูกต้องให้เต็มวง

1. ข้อใดเรียงขั้นตอนของวัฏจักรการพัฒนาระบบงาน (SDLC) ได้ถูกต้อง

ก. การใช้งานจริง
ข. การเขียนโปรแกรม
ค. การปรับปรุงและพัฒนาโปรแกรม
ง. การทดสอบและแก้ไขโปรแกรม
จ. วิเคราะห์ปัญหา
ฉ. การจัดทำเอกสารและคู่มือ
ช. การออกแบบโปรแกรม

① จ ช ข ก ค ง ฉ
② จ ช ข ง ฉ ก ค
③ ฉ จ ข ช ค ก ง
④ จ ช ข ง ฉ ค ก

เฉลย ② เพราะขั้นตอนของวัฏจักรการพัฒนาระบบงาน (SDLC) เริ่มจากการวิเคราะห์ปัญหาการออกแบบโปรแกรม การเขียนโปรแกรม การทดสอบและแก้ไขโปรแกรม การจัดทำเอกสารคู่มือ การใช้งานจริง การปรับปรุงและพัฒนาโปรแกรม

2. การเตรียมความพร้อมในการเขียนโปรแกรมมีความสำคัญอย่างไร

① เพื่อไม่ให้โปรแกรมทำงานผิดพลาดหรือเกิดปัญหาภายหลังจากนำไปใช้งานจริง

② เพื่อให้มีทักษะในการเขียนโปรแกรมเพื่อแก้ปัญหาตามที่ได้ออกแบบไว้

③ เพื่อให้ได้โปรแกรมที่ทำงานได้สมบูรณ์ทุกฟังก์ชันและมีประสิทธิภาพดีที่สุด

④ เพื่อให้รู้ว่าการเขียนโปรแกรมภาษาใดเหมาะกับการแก้ปัญหานั้น ๆ

เฉลย ② เพราะการเตรียมความพร้อมในการเขียนโปรแกรมนั้นก็เพื่อจะฝึกฝนให้มีทักษะในการเขียนโปรแกรมเพื่อแก้ปัญหา เข้าใจการทำงาน รูปแบบของภาษา และสามารถเขียนโปรแกรมแก้ปัญหาตามที่ได้ออกแบบไว้

3. ก่อนที่จะเขียนโปรแกรมแก้ปัญหา ผู้เขียนโปรแกรมควรทำสิ่งใดก่อน เพื่อให้การแก้ปัญหามีประสิทธิภาพมากที่สุด

① จัดหาอุปกรณ์และเครื่องมือที่จำเป็น

② ออกแบบผังงานการทำงานของโปรแกรม

③ ค้นหาการแก้ปัญหานั้น ๆ จากอินเทอร์เน็ต

④ ศึกษาการเขียนโปรแกรมจากสื่อหรือแหล่งเรียนรู้ให้เข้าใจ

เฉลย ② เพราะก่อนการเขียนโปรแกรมเพื่อแก้ปัญหาใด ๆ ควรทำการออกแบบผังงานการทำงานของโปรแกรมก่อน เพื่อให้การเขียนโปรแกรมเพื่อแก้ปัญหามีลำดับและกระบวนการที่ถูกต้องและสามารถตรวจสอบแก้ไขได้ง่าย

4. ข้อใดไม่ใช่ระบบประเภท IoT

① ระบบแสดงสถานะหลอดไฟฟ้าภายในบ้านมายังแอปพลิเคชัน

② ระบบควบคุมการเปิด-ปิดวาล์วน้ำด้วยสมาร์ตโฟน

③ ระบบรายงานสภาพอากาศทั่วประเทศไทย

④ ระบบเปิด-ปิดไฟฟ้าอัตโนมัติตามระดับแสง

เฉลย ④ เพราะเนื่องจากระบบเปิด-ปิดไฟฟ้าอัตโนมัติตามระดับแสงเป็นเพียงระบบอัตโนมัติ สามารถทำงานได้ด้วยตนเองและไม่ต้องเชื่อมต่อกับอินเทอร์เน็ต

5. ข้อใดกล่าวถึง MIT App Inventor ได้ถูกต้อง

① เป็นเครื่องมือที่ใช้สร้างแอปพลิเคชันในรูปของการเขียนโปรแกรมภาษาระดับสูง

② เป็นเครื่องมือที่ใช้สร้างแอปพลิเคชันในรูปของบล็อกคำสั่ง

③ เป็นเครื่องมือที่ใช้ทดสอบแอปพลิเคชันที่สร้างขึ้น

④ เป็นเครื่องมือที่ใช้ออกแบบผังงาน

เฉลย ② เพราะ MIT App Inventor เป็นเครื่องมือที่ใช้สร้างแอปพลิเคชันในรูปแบบของบล็อกคำสั่ง สามารถสร้างแอปพลิเคชันได้ง่าย

6. ข้อใดกล่าวถึง Embedded System ไม่ถูกต้อง

① เปรียบเสมือนกับคอมพิวเตอร์ขนาดเล็ก

② Embedded System เช่น Arduino UNO R3

③ สามารถเขียนคำสั่งควบคุมได้ด้วยภาษาไพธอนเท่านั้น

④ มีความสามารถในการประมวลผลและเชื่อมต่อกับอุปกรณ์ภายนอก

เฉลย ③ เพราะ Embedded System ไม่จำเป็นจะต้องเขียนคำสั่งควบคุมด้วยภาษาไพธอน เสมอไป บางอุปกรณ์ที่ใช้ภาษาอื่น ๆ เช่น Arduino ใช้ภาษาซีในการเขียนคำสั่งควบคุม

7. หากนักเรียนต้องการพัฒนาระบบควบคุมอุปกรณ์ไฟฟ้าภายในบ้าน นักเรียนจะต้องใช้อุปกรณ์อะไร เพื่อทำหน้าที่เชื่อมต่อสัญญาณไฟฟ้าระหว่างบอร์ดสมองกลฝังตัวกับอุปกรณ์ไฟฟ้า

① Relay

② MQTT

③ Protoboard

④ Raspberry Pi 3

เฉลย ① เพราะรีเลย์ (Relay) คือ สวิตช์ตัดต่อวงจรแบบอิเล็กทรอนิกส์ โดยใช้หลักการของแม่เหล็กไฟฟ้า จะทำงานเมื่อมีการจ่ายไฟไปตามกำหนด ทำให้เกิดวงจรเปิด เมื่อไม่มีการจ่ายไฟ จะทำให้เกิดวงจรปิด ทำให้อุปกรณ์ไฟฟ้าที่ใช้รีเลย์เป็นสวิตช์นั้นไม่ทำงาน

8. ข้อใดกล่าวถึงวงจรสมองกลฝังตัว KidBright ได้ถูกต้อง

① เขียนโปรแกรมคำสั่งด้วยภาษาไพธอน

② เขียนโปรแกรมคำสั่งด้วยภาษาจาวา

③ เขียนโปรแกรมคำสั่งด้วยภาษาซี

④ เขียนโปรแกรมคำสั่งแบบบล็อก

เฉลย ④ เพราะวงจรสมองกลฝังตัว KidBright สามารถเขียนโปรแกรมควบคุมด้วยบล็อกคำสั่ง

9. องค์ประกอบหลักของเทคโนโลยี IoT มีอะไรบ้าง

① Embedded Board/System, เครือข่ายอินเทอร์เน็ต

② อุปกรณ์เชื่อมต่อ, เครือข่ายอินเทอร์เน็ต, อุปกรณ์สื่อสาร

③ Embedded Board/System, เครือข่ายอินเทอร์เน็ต, อุปกรณ์สื่อสาร

④ Embedded Board/System, อุปกรณ์เชื่อมต่อ,
เครือข่ายอินเทอร์เน็ต, อุปกรณ์สื่อสาร

เฉลย ④ เพราะเทคโนโลยี IoT ประกอบไปด้วย
Embedded Board/System อุปกรณ์เชื่อมต่อ
เช่น เซนเซอร์, เครือข่ายอินเทอร์เน็ต, อุปกรณ์สื่อสาร

10. ถ้าต้องการสร้างเทคโนโลยี IoT ที่สามารถตรวจจับการเคลื่อนไหวได้ นักเรียนจะใช้เซนเซอร์ใดในการรับข้อมูล

① IR Sensor

② Level Sensor

③ Ultrasonic Sensor

④ Temperature Sensor

เฉลย ① เพราะ IR Sensor เป็นเซนเซอร์ที่ส่งคลื่นอินฟราเรด ใช้สำหรับตรวจจับสิ่งกีดขวาง หรือตรวจจับการเคลื่อนไหวได้