

# MIT App Inventor กับการพัฒนาเพื่อการประยุกต์ใช้จริง

## Real-world Applications Development with MIT App Inventor

ภูมินทร์ ดวงหาค้าง<sup>1</sup>

พัฒนันรี ศรีสมพันธ์<sup>1</sup>

โรเบิร์ต แบทชิงเจอร์<sup>2</sup>

<sup>1</sup>อาจารย์ประจำ คณะวิทยาศาสตร์ มหาวิทยาลัยพายัพ

<sup>2</sup>อาจารย์พิเศษประจำ ดร. คณะวิทยาศาสตร์ มหาวิทยาลัยพายัพ

E-mail: iampumin@gmail.com

### บทคัดย่อ

MIT App Inventor เป็นเครื่องมือที่ใช้สร้างแอปพลิเคชันที่ทำงานบนระบบปฏิบัติการแอนดรอยด์ (Android) โดย MIT App Inventor ใช้หลักการพัฒนาซอฟต์แวร์เชิงคอมโพเนนต์ (Component-based Software Development) ทำให้ผู้ใช้สามารถพัฒนาแอปพลิเคชันได้ง่าย โดยไม่ต้องเขียนรหัสคำสั่ง (Source code) ภาษาจาวา ทำให้บุคคลทั่วไปมองว่า MIT App Inventor ไม่สามารถพัฒนาแอปพลิเคชันเพื่อการประยุกต์ใช้จริงได้ (Real-world application) ดังนั้นในบทความนี้จึงต้องการนำเสนอการนำ MIT App Inventor ไปใช้ในการพัฒนาแอปพลิเคชันสำหรับใช้งานจริงในด้านต่างๆ รวมถึงปัจจัยต่างๆ ที่ทำให้ MIT App Inventor เป็นที่นิยมเพิ่มมากขึ้นในปัจจุบัน ซึ่งจะช่วยให้ผู้อ่านได้เห็นศักยภาพที่แท้จริงของ MIT App Inventor

**คำสำคัญ:** MIT App Inventor แอนดรอยด์ การประยุกต์ใช้จริง การพัฒนาซอฟต์แวร์เชิงคอมโพเนนต์

### ABSTRACT

MIT App Inventor is a tool used to create applications that run on the Android Operating System. MIT App Inventor uses component-based software development approach enabling users to develop applications so easily without coding in Java that many people have mentioned that MIT App Inventor cannot actually be used to develop real-world applications. This article introduces MIT App Inventor as a development environment for real-world application which are useful for various purposes and explains about various factors bring the MIT App Inventor is more and more popular nowadays. So readers can see the real potential of MIT App Inventor for creating practical applications.

**KEYWORDS:** MIT App Inventor, Android, Real-world applications, Component-based Software Development

### บทนำ

App Inventor ถูกสร้างขึ้นด้วยความร่วมมือของ บริษัท Google และ สถาบัน MIT (Massachusetts Institute of Technology) ซึ่งเป็นมหาวิทยาลัยชั้นนำทางด้านวิทยาศาสตร์และเทคโนโลยี โดยมีวัตถุประสงค์

เพื่อให้ผู้ใช้เข้าใจหลักการการพัฒนาแอปพลิเคชันสำหรับใช้งานบนอุปกรณ์ที่ใช้ระบบปฏิบัติการแอนดรอยด์ ด้วย App Inventor ผู้ใช้สามารถสร้างแอปพลิเคชันได้ง่ายโดยไม่ต้องเขียนรหัสคำสั่งภาษาจาวา แอปพลิเคชันที่พัฒนาโดยใช้ MIT App Inventor จัดเป็นเนทีฟแอปพลิเคชัน

(Native Application) คือ มีการจัดเตรียมภาษา และเครื่องมือสำหรับพัฒนาแอปพลิเคชันสำหรับระบบปฏิบัติการแอนดรอยด์เท่านั้น ซึ่งมีข้อดีคือ สามารถเข้าถึงฟังก์ชันการทำงานของระบบปฏิบัติการแอนดรอยด์ได้อย่างครบถ้วนและมีประสิทธิภาพ ทำให้แอปพลิเคชันมีประสิทธิภาพการทำงานสูงสุด มีความยืดหยุ่นสูง (Hybrid & native application development, ม.ป.ป.) ต่อมาบริษัท Google ได้ถอนตัวออกจากโครงการนี้ และส่งมอบให้ MIT เป็นผู้พัฒนา App Inventor ต่อ และได้เปลี่ยนชื่อเป็น MIT App Inventor โดยทางสถาบัน MIT ใช้เครื่องมือนี้เป็นพื้นฐานสำหรับการเรียนการสอนทางด้านการเขียนโปรแกรมเป็นหลัก ทำให้บุคคลทั่วไปมองว่า MIT App Inventor ใช้สำหรับการเรียนการสอนเท่านั้น ซึ่งจริงๆ แล้ว MIT App Inventor ได้จัดเตรียมคอมโพเนนต์ (Component) พื้นฐานที่เพียงพอสำหรับสร้างแอปพลิเคชันพื้นฐานที่สามารถทำงานได้ในระดับหนึ่ง อีกทั้งในปัจจุบันมี ชุมชน MIT App Inventor (MIT App Inventor Community) ซึ่งมีการพูดคุย แลกเปลี่ยนเทคนิคใหม่ๆ และวิธีการแก้ปัญหาเกี่ยวกับการพัฒนาแอปพลิเคชัน ทำให้การพัฒนาแอปพลิเคชัน โดยใช้ MIT App Inventor ก้าวหน้าไปอย่างมาก สิทธิในแอปพลิเคชันที่พัฒนาโดยใช้ MIT App Inventor เป็นสัญญาอนุญาตแบบเปิดกว้างโดยสงวนสิทธิเฉพาะแหล่งที่มา (Creative Commons Attribution 4.0 International License) คือ อนุญาตให้นำแอปพลิเคชันไปใช้ได้ ดัดแปลงแก้ไขแอปพลิเคชันได้ และนำแอปพลิเคชันไปใช้ในเชิงพาณิชย์ได้ แต่ต้องอ้างอิงถึงแหล่งที่มาหรือเจ้าของผลงาน ซึ่งเป็นประโยชน์ทั้งกับเจ้าของแอปพลิเคชัน ที่ยังคงมีสิทธิในแอปพลิเคชัน และยังสามารถเผยแพร่และแบ่งปันให้ผู้อื่นนำแอปพลิเคชันนั้นไปใช้งาน และพัฒนาต่อยอดได้ (Creative Commons Attribution 4.0 International, ม.ป.ป.)

ดังนั้นในบทความนี้จึงต้องการนำเสนอการนำ MIT App Inventor ไปใช้ในการพัฒนาแอปพลิเคชันสำหรับใช้งานจริงด้านต่างๆ รวมถึงปัจจัยต่างๆ ที่ทำให้ MIT App Inventor เป็นที่นิยมเพิ่มมากขึ้นในปัจจุบัน เพื่อให้ผู้อ่านได้เห็นศักยภาพที่แท้จริงของ MIT App Inventor

### การพัฒนาแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์

ขั้นตอนของการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์โดยทั่วไป (ขั้นตอนการ

พัฒนาแอปพลิเคชันแอนดรอยด์, ม.ป.ป.) จะมี 4 ขั้นตอน ดังนี้

1. สร้างโปรเจกต์ ในการพัฒนาแอปพลิเคชัน โดยจะต้องจัดเตรียมโพลเดอร์ เพื่อจัดเก็บรหัสคำสั่ง และไฟล์ต่างๆ ที่จำเป็นสำหรับการพัฒนาแอปพลิเคชัน
2. ออกแบบและพัฒนาหน้าจอส่วนติดต่อกับผู้ใช้ (User interface) โดยใช้ภาษาเอกซ์เอ็มแอล XML (Extensible Markup Language) ซึ่งจะได้ไฟล์ชื่อ main.xml สำหรับใช้เป็นตัวควบคุมเค้าโครง (Layout) ของส่วนประกอบต่างๆ เก็บไว้ในโพลเดอร์ res/layout
3. เขียนรหัสคำสั่งสำหรับการทำงานของแอปพลิเคชันตามที่ต้องการ โดยใช้รหัสคำสั่งภาษาจาวา (Java) ซึ่งรหัสคำสั่งจะเก็บไว้ในโพลเดอร์ src/package
4. ทดสอบแอปพลิเคชันที่พัฒนาในอุปกรณ์เสมือน AVD (Android Virtual Device) หรือ บนอุปกรณ์ที่ใช้ระบบปฏิบัติการแอนดรอยด์

โปรแกรมที่ใช้พัฒนาแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์ที่นิยมใช้ในปัจจุบัน มี 2 โปรแกรม คือ Eclipse ADT และ Android Studio ซึ่งแต่เดิมจะใช้โปรแกรม Eclipse ADT แต่ภายหลังทางทีมพัฒนา Eclipse ADT ได้พัฒนาโปรแกรม Android Studio ออกมาแทน ซึ่งทำงานได้ดีและมีประสิทธิภาพมากกว่าโปรแกรม Eclipse ADT (Guideline สำหรับการเริ่มต้นเขียนแอปพลิเคชันแอนดรอยด์, ม.ป.ป.)

จะเห็นได้ว่าการพัฒนาแอปพลิเคชันแอนดรอยด์ด้วยวิธีมาตรฐาน ผู้พัฒนาจะต้องมีความรู้และทักษะในการออกแบบหน้าจอโดยใช้ภาษาเอกซ์เอ็มแอล และการเขียนรหัสคำสั่งโดยใช้ภาษาจาวา

### การพัฒนาแอปพลิเคชันโดยใช้ MIT App Inventor

MIT App Inventor เป็นเครื่องมือสำหรับพัฒนาแอปพลิเคชันที่ทำงานบนระบบปฏิบัติการแอนดรอยด์ MIT App Inventor ในปัจจุบันเป็นรุ่น (Version) 2 ซึ่งหลังจากนี้ผู้เขียนจะใช้ชื่อ AI2 แทน MIT App Inventor 2 เพราะเป็นชื่อที่ใช้เรียกเครื่องมืออย่างแพร่หลายในปัจจุบัน โดย AI2 จะใช้หลักการพัฒนาซอฟต์แวร์เชิงคอมโพเนนต์ (Component-based Software Development) คือ อนุญาตให้ผู้ใช้ นำคอมโพเนนต์หรือชิ้นส่วนของซอฟต์แวร์มาประกอบกันเป็นระบบหรือแอปพลิเคชัน โดย AI2 ได้

จัดเตรียมคอมโพเนนต์พื้นฐาน เพื่อให้ผู้ใช้นำมาประกอบเป็นแอปพลิเคชันตามที่ต้องการได้ ดังแสดงในรูปที่ 1

ใน AI2 มีการแบ่งคอมโพเนนต์พื้นฐานเป็นกลุ่มดังนี้

User Interface components เป็นกลุ่มของคอมโพเนนต์ที่ใช้ติดต่อกับผู้ใช้ เช่น ส่วนรับข้อความหรือ ปุ่มคำสั่ง เป็นต้น

Layout components เป็นกลุ่มของคอมโพเนนต์ที่ควบคุมการวางตำแหน่งของคอมโพเนนต์ที่ใช้ติดต่อกับผู้ใช้บนหน้าจอ

Media components เป็นกลุ่มของคอมโพเนนต์ที่ใช้แสดงสื่อในรูปแบบต่างๆ เช่น รูปภาพ วิดีโอ เสียง เป็นต้น

Drawing and Animation components เป็นกลุ่มของคอมโพเนนต์ที่ใช้วาดรูป และแสดงภาพเคลื่อนไหว

Sensor components เป็นกลุ่มของคอมโพเนนต์ที่ใช้ทำงานร่วมกับเซ็นเซอร์ของอุปกรณ์ต่างๆ เช่น จีพีเอส (GPS) และ เข็มทิศ

Social components เป็นกลุ่มของคอมโพเนนต์ที่ใช้สื่อสารกับสังคมออนไลน์ เช่นการส่งเอสเอ็มเอส (SMS) การเรียกดูสมุดโทรศัพท์ หรือ การแชร์กับสังคมออนไลน์ เป็นต้น

Storage components เป็นกลุ่มของคอมโพเนนต์ที่ใช้จัดการกับข้อมูลในฐานข้อมูลขนาดเล็ก หรือข้อมูลในไฟล์

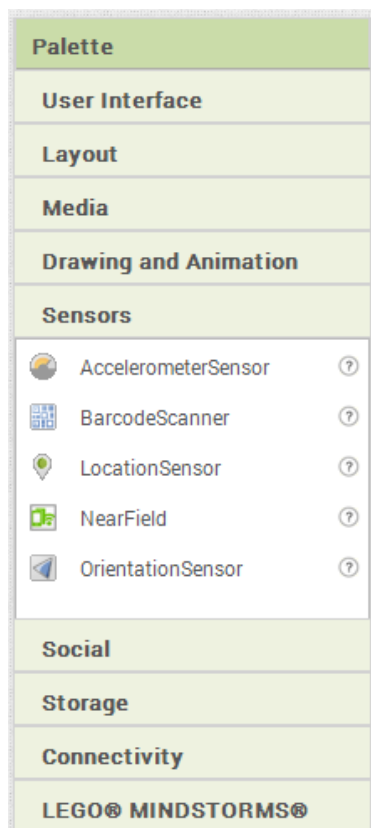
Connectivity components เป็นกลุ่มของคอมโพเนนต์ที่ใช้สร้างการเชื่อมต่อกับอุปกรณ์อื่นผ่านบลูทูธ และเชื่อมต่อกับแอปพลิเคชันที่พัฒนาโดยใช้ภาษาจาวา (Java Application)

LEGO® MINDSTORMS® components เป็นกลุ่มของคอมโพเนนต์ที่ใช้ควบคุมหุ่นยนต์เลโก (Lego Mindstorms NXT Robot)

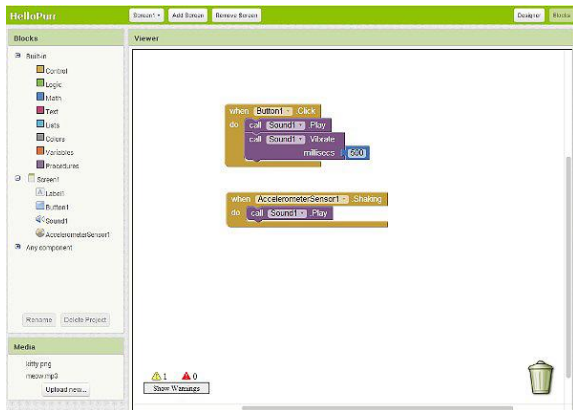
ประโยชน์ของคอมโพเนนต์เหล่านี้ คือ ทำให้การพัฒนาแอปพลิเคชันเร็วขึ้น เพราะแต่ละคอมโพเนนต์มีหน้าที่เฉพาะ มีการจัดการกับข้อมูลของตนเอง ทำให้นักพัฒนาไม่ต้องเสียเวลาในการทำงานพื้นฐานเหล่านั้น แต่จะใช้เวลาในการเลือกคอมโพเนนต์ที่เหมาะสมเพื่อนำมาประกอบกันเป็นแอปพลิเคชันแทน ซึ่งการพัฒนาแอปพลิเคชันในลักษณะนี้ จะเป็นการใช้ประโยชน์จากแนวคิดของการนำกลับมาใช้ใหม่ (Reuse) โดยการนำ

คอมโพเนนต์มาตรฐานใน AI2 หรือ คอมโพเนนต์ที่มีการแบ่งปันกัน(Share) ในชุมชนนักพัฒนามาใช้

การใช้งาน AI2 เป็นแบบมองเห็นได้ในขณะออกแบบ (Visually design) คือ เป็นลักษณะของภาษาภาพที่มีส่วนติดต่อกับผู้ใช้แบบลากแล้ววาง ทำให้ผู้ใช้สามารถพัฒนาแอปพลิเคชันได้โดยไม่ต้องเขียนรหัสคำสั่ง ผู้ใช้สามารถมองเห็นการออกแบบแอปพลิเคชันบนหน้าจอได้ทันที ผู้ใช้สามารถกำหนดองค์ประกอบหรือชิ้นส่วนของแอปพลิเคชัน แล้วระบุลักษณะหรือเหตุการณ์การทำงานของแอปพลิเคชัน โดยการกำหนดค่าที่ต้องการลงในบล็อก (Block) เหมือนกับการเขียนโปรแกรมด้วยภาษาใดภาษาหนึ่ง เช่น การกำหนดตัวแปร การสร้างเงื่อนไข การทำซ้ำ หรือแม้แต่การคำนวณ การเก็บข้อมูลในฐานข้อมูล และสร้างการพูดคุยกับบริการเว็บ เช่น ทวิตเตอร์ (Twitter) หรือ เฟซบุ๊ก (Facebook) (Guideline สำหรับการเริ่มต้นเขียนแอปพลิเคชันแอนดรอยด์, ม.ป.ป.) เป็นต้น ดังตัวอย่างในรูปที่ 2



รูปที่ 1 กลุ่มของคอมโพเนนต์ใน MIT App Inventor (Component types, n.d.)

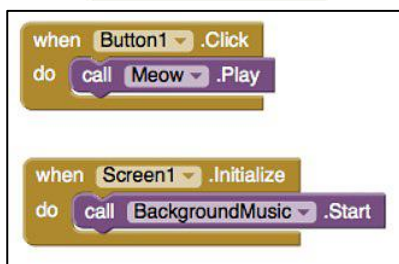


รูปที่ 2 การกำหนดรายละเอียดในบล็อก (The App Inventor environment, n.d.)

โครงสร้างบล็อกที่ใช้ใน AI2 Inventor จะอยู่ในรูปแบบของรูปทรงที่มีสี โดยแต่ละสีก็จะเป็นส่วนของข้อมูลเงื่อนไข โครงสร้างการควบคุม และการทำงาน ที่แบ่งแยกจากกันอย่างชัดเจน ซึ่งจะช่วยให้การพัฒนาแอปพลิเคชันง่ายต่อความเข้าใจ และมีความถูกต้องมากกว่าการใช้ผังงาน (Flowcharts) ในการพัฒนาแอปพลิเคชันแบบดั้งเดิม

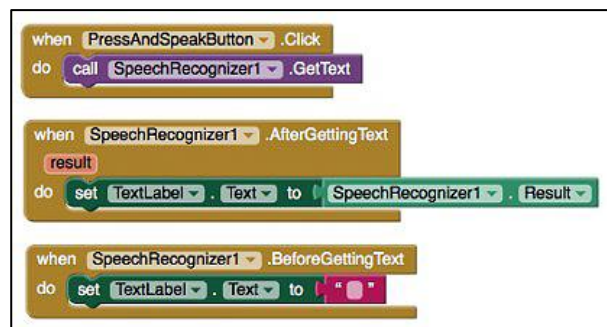
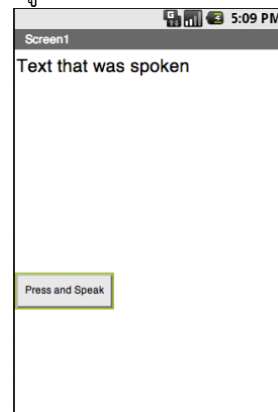
ถึงแม้วิธีการพัฒนาแอปพลิเคชันใน AI2 จะทำด้วยวิธีการง่ายๆ แต่ก็สามารถทำงานได้อย่างหลากหลาย เช่น

- 1) การเพิ่มสื่อมัลติมีเดีย ไม่ว่าจะเป็นเสียงรูปภาพ หรือ ภาพเคลื่อนไหว ดังรูปที่ 3



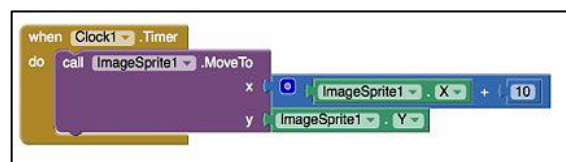
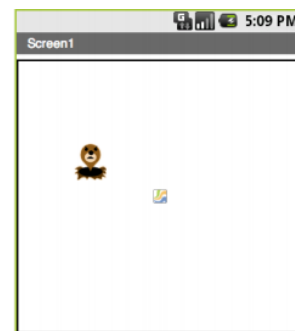
รูปที่ 3 การกำหนดรูปภาพ และเสียง (MIT App Inventor, n.d.)

- 2) การแสดงเสียงอ่านข้อความ (Speech recognition) ดังรูปที่ 4



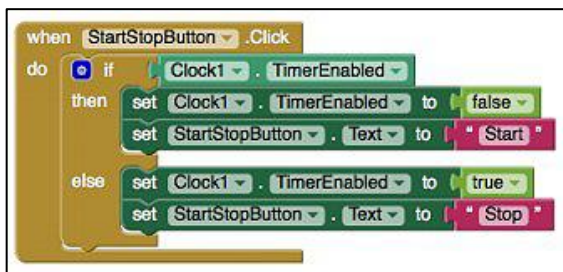
รูปที่ 4 การแสดงเสียงอ่านข้อความ (MIT App Inventor, n.d.)

- 3) การกำหนดการเคลื่อนไหวของวัตถุจากจุดหนึ่งไปยังอีกจุดหนึ่ง โดยการระบุเวลาที่ต้องการ ดังรูปที่ 5



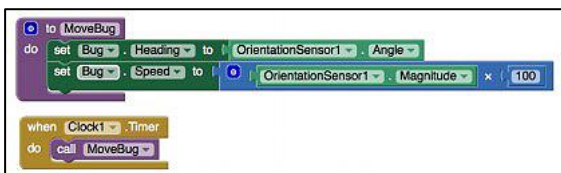
รูปที่ 5 การกำหนดการเคลื่อนไหวโดยระบุเวลา (MIT App Inventor, n.d.)

4) การกำหนดการเคลื่อนไหวนៃของวัตถุ โดยการคลิกปุ่ม เพื่อเริ่มหรือหยุดการเคลื่อนไหวนៃ ดังรูปที่ 6



รูปที่ 6 การกำหนดการเคลื่อนไหวนៃ โดยการกดปุ่ม Start หรือ Stop (MIT App Inventor, n.d.)

5) การกำหนดการเคลื่อนไหวนៃ โดยใช้วิธีการเอียงมือถือ ดังรูปที่ 7



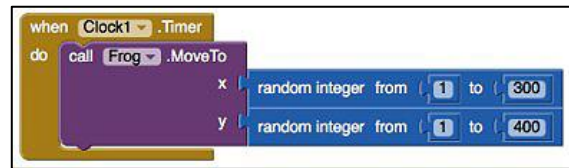
รูปที่ 7 การกำหนดการเคลื่อนไหวนៃ โดยการเอียงโทรศัพท์มือถือ (MIT App Inventor, n.d.)

6) การกำหนดการเคลื่อนไหวนៃของวัตถุ โดยการใช้นิ้วลากไปบนหน้าจอโทรศัพท์มือถือ ดังรูปที่ 8



รูปที่ 8 การกำหนดการเคลื่อนไหวนៃ โดยการใช้นิ้วบังคับบนหน้าจอโทรศัพท์มือถือ (MIT App Inventor, n.d.)

7) การสุ่มตัวเลข เพื่อใช้ในการกำหนดตำแหน่งของการแสดงวัตถุ (x,y) ดังรูปที่ 9



รูปที่ 9 การสุ่มตัวเลขเพื่อกำหนดตำแหน่งการแสดงวัตถุ (MIT App Inventor, n.d.)

8) การวาดเส้นหรือวัตถุโดยการลาก (drag) นิ้วแตะบนหน้าจอมือถือดังรูปที่ 10



รูปที่ 10 การวาดวัตถุโดยการลากนิ้ว (MIT App Inventor, n.d.)

9) การกำหนดการทำงานโดยผ่านการเขย่าโทรศัพท์มือถือ ดังรูปที่ 11



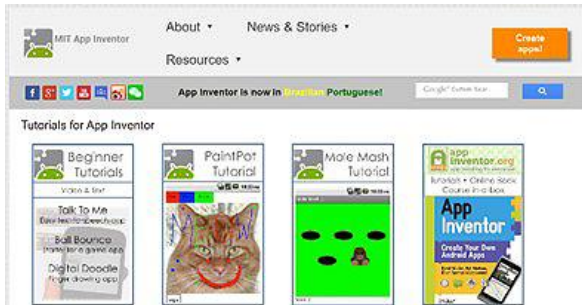
รูปที่ 11 การกำหนดการทำงานโดยการเขย่าโทรศัพท์มือถือ (MIT App Inventor, n.d.)

10) การกำหนดการทำงานหรือการเคลื่อนไหวนៃเมื่อวัตถุ 2 วัตถุชนกัน ดังรูปที่ 12



รูปที่ 12 การกำหนดการทำงานเมื่อวัตถุมาชนกัน (MIT App Inventor, n.d.)

ซึ่งการสร้างแอปพลิเคชันเพื่อทำงานพื้นฐานเหล่านี้ผู้ใช้สามารถเรียนรู้ด้วยตนเองผ่านเว็บไซต์ของ AI2 ดังแสดงในรูปที่ 13



รูปที่ 13 เว็บไซต์สอนการพัฒนาแอปพลิเคชันพื้นฐานโดยใช้ MIT App Inventor (The App Inventor environment, n.d.)

### การใช้ MIT App Inventor เพื่อพัฒนาแอปพลิเคชันใช้งานจริงด้านต่างๆ

ในส่วนนี้ผู้เขียนจะกล่าวถึงการใช้ AI2 พัฒนาแอปพลิเคชันใช้งานจริงด้านต่างๆ นอกเหนือจากการพัฒนาแอปพลิเคชันพื้นฐาน ที่มีการกล่าวถึงในคู่มือหรือเว็บไซต์ของ AI2

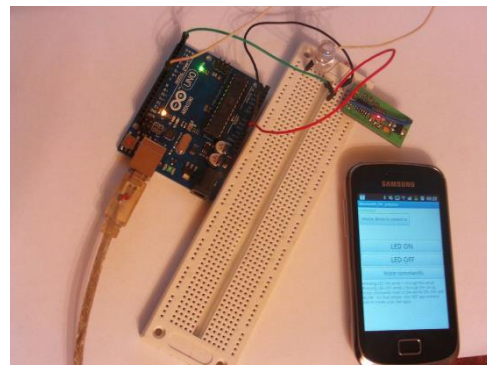
#### 1) การติดต่อกับฐานข้อมูล

ฐานข้อมูลเป็นสิ่งที่สำคัญที่สุดของแอปพลิเคชันทางธุรกิจ (Business application) เพราะต้องใช้เก็บข้อมูลต่างๆ เพื่อการดำเนินการทางธุรกิจ ซึ่ง AI2 สามารถสร้างแอปพลิเคชันสำหรับติดต่อกับฐานข้อมูลชนิดต่างๆ ได้อย่างหลากหลาย AI2 ได้จัดเตรียมคอมโพเนนต์พื้นฐานสำหรับติดต่อกับฐานข้อมูล ได้แก่ TinyDB และ TinyWebDB โดย TinyDB เป็นคอมโพเนนต์ที่ใช้ติดต่อกับฐานข้อมูลขนาดเล็กที่อยู่ในโทรศัพท์มือถือ ส่วน TinyWebDB เป็นคอมโพเนนต์ที่ใช้ติดต่อกับฐานข้อมูลที่อยู่บนเว็บไซต์ และนอกจากนั้นเรายังสามารถใช้ AI2 สร้างแอปพลิเคชันเพื่อติดต่อกับฐานข้อมูลมายเอสคิวแอล (MySQL) ซึ่งเป็นฐานข้อมูลที่นิยมใช้ในการเก็บข้อมูลสำหรับธุรกิจต่างๆ ในปัจจุบันเนื่องจากเป็นโปรแกรมที่มีขนาดเล็ก มีความเร็วสูง และไม่เสียค่าใช้จ่าย โดยการติดต่อกับฐานข้อมูลมายเอสคิวแอลจะติดต่อผ่าน Web API (Application Programming Interface) โดยใช้ภาษาพีเอชพี (PHP) คือ จะต้องทำการเขียนรหัสคำสั่งภาษาพีเอชพี เพื่อสร้างช่องทางการ

เชื่อมต่อกับฐานข้อมูลมายเอสคิวแอลเตรียมไว้ ดังบทความเรื่อง App Inventor-MySQL interface (MIT App Inventor, n.d.) และบทความ Connect App Inventor to MySQL Database (Connect App Inventor to MySQL Database, n.d.)

#### 2) การติดต่อกับอุปกรณ์ต่างๆ

AI2 ได้จัดเตรียมคอมโพเนนต์สำหรับการเชื่อมต่อกับอุปกรณ์ต่างๆ ได้แก่ Bluetooth Client และ Bluetooth Server สำหรับใช้สร้างการเชื่อมต่อกับอุปกรณ์ต่างๆ ผ่านทางบลูทูธ ทำให้แอปพลิเคชันที่พัฒนาโดยใช้ AI2 สามารถเชื่อมต่อกับอุปกรณ์ต่างๆ ได้อย่างหลากหลาย โดยเฉพาะการเชื่อมต่อกับบอร์ดไมโครคอนโทรลเลอร์ที่มีการพัฒนาแบบเปิดเผยรหัส (Open source) ที่มีผู้นิยมใช้เป็นอย่างมาก เพราะสามารถนำไปใช้ได้ฟรีและสามารถนำไปพัฒนาต่อยอดได้ ในปัจจุบันมีผู้ใช้บอร์ดไมโครคอนโทรลเลอร์ไปควบคุมอุปกรณ์ชนิดต่างๆ อย่างหลากหลาย และการที่ AI2 สามารถติดต่อกับบอร์ดไมโครคอนโทรลเลอร์ผ่านทางบลูทูธ ทำให้นักพัฒนาสามารถพัฒนาแอปพลิเคชันเพื่อควบคุมอุปกรณ์ต่างๆ เหล่านั้นได้เช่นกัน ดังตัวอย่างในรูปที่ 14



รูปที่ 14 การพัฒนาแอปพลิเคชันเพื่อเชื่อมต่อกับบอร์ดไมโครคอนโทรลเลอร์ (How to control arduino board using an android phone, 2013)

#### 3) แอปพลิเคชันใช้งานจริงอื่นๆ

บนเว็บไซต์ของ AI2 ได้มีการนำเสนอแอปพลิเคชันใช้งานจริง (MIT App Inventor stories, n.d.) ที่พัฒนาโดยใช้ AI2 ตัวอย่างเช่น การพัฒนาแอปพลิเคชันสำหรับคำนวณน้ำหนักสุทธิของระเบิดโดยใช้โทรศัพท์มือถือ ซึ่งพัฒนาโดยทหารนาวิกโยธินสหรัฐอเมริกา เพื่อที่จะนำระเบิดที่คำนวณน้ำหนักแล้วไปทำลายด้วยวิธีการที่เหมาะสม

อย่างปลอดภัย ดังรูปที่ 15 หรือการพัฒนาแอปพลิเคชัน สำหรับเก็บรวบรวมและบันทึกข้อมูลปริมาณน้ำฝนจาก เครื่องวัดปริมาณน้ำฝนที่กระจายอยู่ทั่วประเทศเฮติ เพื่อ งานช่วยเหลือด้านมนุษยธรรมในประเทศเฮติ ดังรูปที่ 16



รูปที่ 15 แอปพลิเคชันสำหรับคำนวณน้ำหนักสุทธิของ ระเบิด (MIT App Inventor stories, n.d.)



รูปที่ 16 แอปพลิเคชันสำหรับเก็บรวบรวมและบันทึก ข้อมูลปริมาณน้ำฝนในประเทศเฮติ (MIT App Inventor stories, n.d.)

เนื่องจากทางเว็บไซต์เปิดโอกาสให้นักพัฒนา สามารถเผยแพร่เรื่องราวของแอปพลิเคชันที่พัฒนาแล้ว นำไปใช้งานจริงได้ ทำให้ตัวอย่างของแอปพลิเคชันบน เว็บนี้ มีจำนวนเพิ่มขึ้นอย่างต่อเนื่อง ทำให้เห็นได้ว่า AI2 สามารถพัฒนาแอปพลิเคชันเพื่อนำไปใช้งานจริงได้อย่าง หลากหลายและมีประสิทธิภาพ

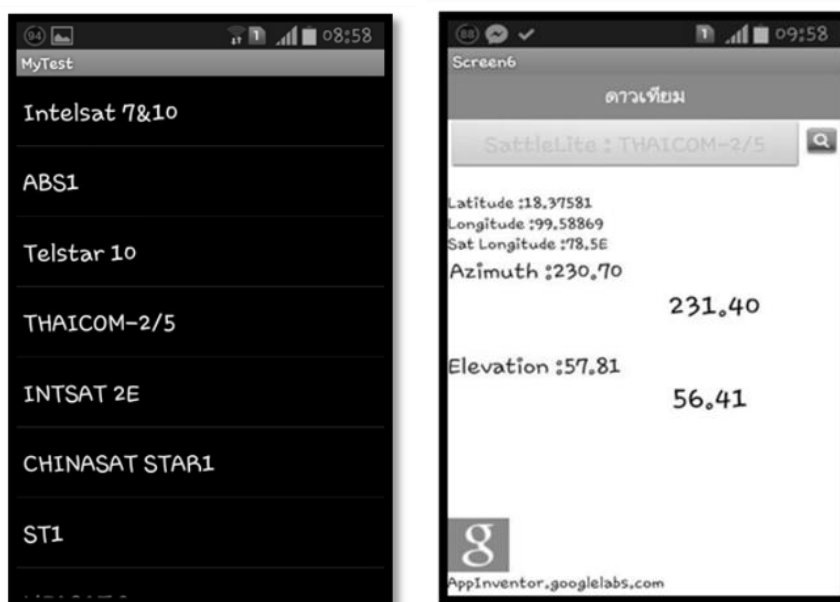
## แอปพลิเคชันพัฒนาโดยคนไทย

มงคล และนวพร (2558) ได้พัฒนาระบบตรวจจับ กลิ่น และแอปพลิเคชันติดตามกลิ่นบนโทรศัพท์มือถือผ่าน แทนการใช้งานสุนัขดมกลิ่นในส่วนของแอปพลิเคชันติดตาม กลิ่นบนโทรศัพท์มือถือ คุณค่าผู้วิจัยได้เลือกใช้โปรแกรม AI2 เป็นเครื่องมือในการพัฒนา ซึ่งช่วยให้คณะผู้วิจัย สามารถพัฒนาแอปพลิเคชันบนโทรศัพท์มือถือได้ โดยง่าย แอปพลิเคชันที่พัฒนาขึ้นสามารถติดต่อและ ควบคุมระบบตรวจจับกลิ่นที่พัฒนาบนอาอคยโนได้ อีกทั้ง แอปพลิเคชัน สามารถแสดงผลพื้ในรูปแบบของกราฟได้ อย่างสวยงาม นอกจากนี้แอปพลิเคชันสามารถอ่านออกเสียง เอร์เซ็นต์ของการเปรียบเทียบกลิ่นได้ (รูปที่ 17)

ประสงค์ (2558) ได้พัฒนาโปรแกรมช่วยติดตั้ง งานดาวเทียมบนระบบปฏิบัติการแอนดรอยด์ เพื่อช่วยลด ขั้นตอน อุปกรณ์ และเครื่องมือในการติดตั้งงานดาวเทียม โดยใช้คุณสมบัติของเซนเซอร์ต่างๆ ที่มีอยู่ในอุปกรณ์ สมาร์ทโฟนช่วยค้นหาตำแหน่งทิศทางของดาวเทียม ซึ่ง โปรแกรมหรือแอปพลิเคชันนี้พัฒนาขึ้นโดยใช้ AI2 และได้มี การทดสอบการทำงานของแอปพลิเคชันโดยเปรียบเทียบ กับการติดตั้งด้วยเครื่องมือแบบเดิม ปรากฏว่าแอปพลิเคชัน ที่พัฒนาขึ้น สามารถบอกข้อมูล ระบุพิกัดและใช้งานเป็น เครื่องมือช่วยในการติดตั้งงานดาวเทียมได้อย่างมี ประสิทธิภาพ (รูปที่ 18)



รูปที่ 17 แอปพลิเคชันบันทึกกลิ่นและติดตามกลิ่นเป้าหมาย (มงคล และ นวพร, 2558)



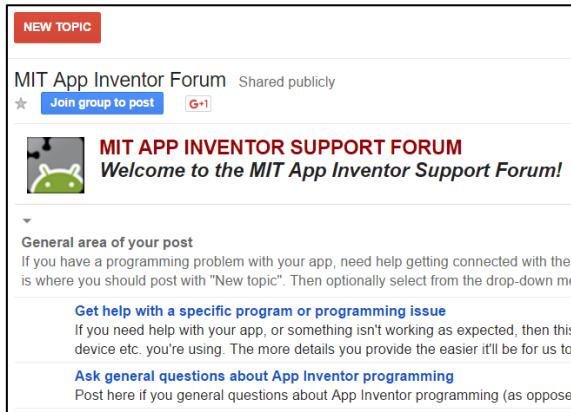
รูปที่ 18 การใช้งานโปรแกรมค้นหาหมุม ตำแหน่ง และดาวเทียม (ประสงค์, 2558)

### ชุมชน MIT App Inventor (MIT App Inventor Community)

ปัจจัยที่สำคัญที่ทำให้การพัฒนาแอปพลิเคชันโดยใช้ AI2 เป็นที่รู้จักแพร่หลายมากขึ้นในปัจจุบันคือชุมชน MIT App Inventor ซึ่งมีการสร้างกลุ่มอภิปรายแบบฟอรัม (Forum) หรือกลุ่มย่อย เพื่อพูดคุย แลกเปลี่ยนความรู้ความคิดเห็น และเทคนิคการพัฒนาแอปพลิเคชันใหม่ๆ ผ่านบริการกูเกิลกรุ๊ปส์ (Google Groups) ดังแสดง

ในรูปที่ 19 ตัวอย่างของกลุ่มอภิปราย เช่น กลุ่มที่พูดคุยปัญหาเกี่ยวกับการเขียนโปรแกรม หรือ กลุ่มที่พูดคุยเกี่ยวกับการนำ AI2 ไปใช้ในการเรียนการสอน หรือ กลุ่มที่พูดคุยเกี่ยวกับเทคนิคการพัฒนาแอปพลิเคชันโดยใช้ AI2 (MIT App Inventor Support Forum, n.d.) เป็นต้น นอกจากกลุ่มอภิปรายผ่านบริการกูเกิลกรุ๊ปส์แล้ว ยังมีการสร้างกลุ่มพูดคุยบนเฟซบุ๊ก (Facebook) (MIT App Inventor Facebook, n.d.)





รูปที่ 19 กลุ่มอภิปรายแบบฟอรัม

ชุมชน MIT App Inventor ตั้งเดิมาก่อนตั้งมานานกว่า 3 ปี (MIT App Inventor Classic RIP, 2015) ถึงแม้จุดประสงค์ของการก่อตั้งครั้งแรกเป็นเพียงโครงการนำร่องเพื่อให้ MIT App Inventor เป็นที่รู้จัก แต่ก็ถือว่าประสบความสำเร็จเป็นอย่างมาก มีการเผยแพร่แอปพลิเคชันมากกว่า 3.97 ล้านแอปพลิเคชัน โดยนักพัฒนา 1.7 ล้านคน และเมื่อเกิด AI2 ก็เกิดชุมชนใหม่ที่มีอัตราการเติบโตอย่างต่อเนื่องและรวดเร็ว ชุมชนจะทำให้เกิดปัญญาสะสม (Collective Intelligence) (วิวัฒน์, 2555) ซึ่งจะช่วยส่งเสริมและกระตุ้นให้เกิดแนวคิดในการพัฒนาแอปพลิเคชันที่สร้างสรรค์ และซับซ้อนกว่าการพัฒนาแอปพลิเคชันโดยลำพัง

นอกเหนือจากกลุ่มอภิปราย หรือกลุ่มพูดคุย แลกเปลี่ยนที่กล่าวมา ยังมีการจัดประกวดแอปพลิเคชันดีเด่นที่พัฒนาด้วย AI2 โดยที่นักพัฒนาสามารถส่ง แอปพลิเคชัน ไปยังเว็บไซต์ของ AI2 เพื่อเข้าร่วมประกวด จะมีการตัดสินและประกาศผลแอปพลิเคชันที่ชนะเลิศ ประจำเดือนประเภทต่างๆ ได้แก่ ประเภทการออกแบบ ประเภทสร้างสรรค์ ประเภทนวัตกรรม และประเภทสิ่งประดิษฐ์ ดังรูปที่ 20 ซึ่งการประกวดนี้เป็นส่วนสำคัญที่ช่วยส่งเสริมให้นักพัฒนาให้นำ AI2 ไปพัฒนาแอปพลิเคชันด้านต่างๆ อย่างหลากหลาย

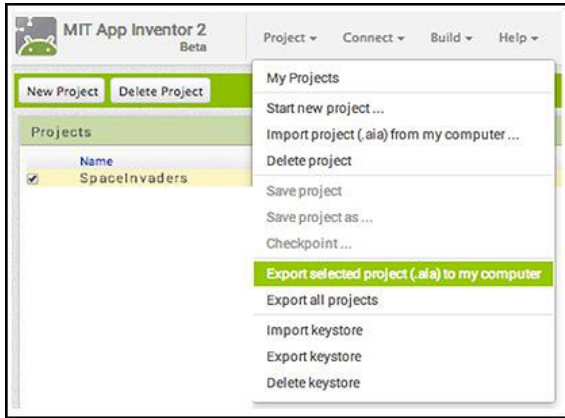


รูปที่ 20 การจัดประกวดแอปพลิเคชันดีเด่นประจำเดือน (App of the Month winners, n.d.)

### การแจกจ่ายแอปพลิเคชัน (Application Distribution) ที่พัฒนาโดย MIT App Inventor

เมื่อพัฒนาแอปพลิเคชันโดยใช้ AI2 เสร็จสมบูรณ์ นักพัฒนาสามารถแจกจ่ายแอปพลิเคชันได้ใน 2 รูปแบบ คือ รูปแบบของรหัสคำสั่ง (Source code form) ที่มีนามสกุลเป็น .aia เพื่อให้ให้นักพัฒนาคนอื่นสามารถเปิดแอปพลิเคชันนี้เพื่อศึกษาหรือแก้ไขได้ และรูปแบบที่รันได้ (Executable form) ซึ่งมีนามสกุลไฟล์ .apk เหมือนกับแอปพลิเคชันที่พัฒนาโดยใช้เครื่องมือมาตรฐานอื่น เช่น Android Studio โดยการแจกจ่ายสามารถใช้คำสั่งใน AI2 ได้โดยตรง ดังรูปที่ 21 และ 22 ตามลำดับ

แอปพลิเคชันที่มีนามสกุลไฟล์ .apk สามารถทำงานได้บนอุปกรณ์ต่างๆ ที่ใช้ระบบปฏิบัติการแอนดรอยด์ นักพัฒนาสามารถแจกจ่ายแอปพลิเคชันที่มีนามสกุลไฟล์ .apk ที่พัฒนาโดยใช้ AI2 ออกสู่สาธารณะ โดยโอนย้าย (Publish) แอปพลิเคชันนั้นไปไว้บน Google Play Store เพื่อให้ผู้ที่สนใจในแอปพลิเคชันนั้นสามารถดาวน์โหลดแอปพลิเคชันไปใช้งานได้



รูปที่ 21 การแจกจ่ายแอปพลิเคชันในรูปแบบรหัสคำสั่ง (.aia) (Sharing and packaging apps, n.d.)



รูปที่ 22 การแจกจ่ายแอปพลิเคชันในรูปแบบที่รันได้ (.apk) (Sharing and packaging apps, n.d.)

### ข้อจำกัดของ MIT App Inventor

แอปพลิเคชันที่พัฒนาโดยใช้ AI2 ในปัจจุบันยังมีข้อจำกัดบางประการ (App Inventor capabilities and limitations, n.d.) เช่น

คอมพิวเตอร์ในกลุ่มที่ใช้ติดต่อกับผู้ใช้บางตัว ยังมีข้อจำกัดบางประการ

ยังมีข้อจำกัดในเรื่องการเข้าถึงอุปกรณ์โทรศัพท์มือถือ เนื่องจากยังขาดคอมพิวเตอร์ที่ทำหน้าที่บางอย่าง เช่น การขาดคอมพิวเตอร์ที่ทำหน้าที่ในการบันทึกและอ่านแฟ้ม (File) ที่อยู่ในระบบจัดเก็บแฟ้มของโทรศัพท์มือถือ และการขาดคอมพิวเตอร์ที่ทำหน้าที่จัดการกับรายชื่อผู้ติดต่อ (Contact list)

ยังขาดคอมพิวเตอร์สำหรับเข้าถึงเว็บโดยตรง ต้องติดต่อผ่าน API คือ ต้องทำการเขียนรหัสคำสั่งเพื่อสร้างช่องทางการเชื่อมต่อเตรียมไว้บนเว็บไซต์

จากตัวอย่างของข้อจำกัดที่ได้กล่าวมาจะเห็นว่า เป็นข้อจำกัดที่ไม่ได้เกิดจากการทำงานของตัว AI2 แต่เป็นเพียงการขาดคอมพิวเตอร์ หรือส่วนเสริมที่ช่วยทำให้การพัฒนาแอปพลิเคชันง่ายและสะดวกขึ้น และด้วย

การที่ AI2 เป็นที่รู้จักอย่างแพร่หลายมากขึ้นในปัจจุบัน (Xie et al., 2015) มีการแลกเปลี่ยนความคิดเห็น และข้อเสนอแนะของนักพัฒนาในชุมชน AI2 อย่างกว้างขวาง ผู้เขียนคิดว่าทีมพัฒนา AI2 คงจะรับทราบถึงข้อจำกัดเหล่านั้น และพัฒนาปรับปรุง AI2 เพื่อตอบสนองต่อความต้องการของนักพัฒนาในอนาคตอันใกล้

### บทสรุป

บทความนี้นำเสนอการนำ MIT App Inventor ไปใช้ในการพัฒนาแอปพลิเคชันสำหรับใช้งานจริง ซึ่ง MIT App Inventor หรือ AI2 ใช้หลักการพัฒนาซอฟต์แวร์เชิงคอมโพเนนต์ และมีการใช้งานแบบมองเห็นได้ในขณะออกแบบ ทำให้ง่ายในการพัฒนาแอปพลิเคชันโดยที่ผู้ใช้ไม่จำเป็นต้องออกแบบหน้าจอโดยใช้ภาษาเอกซ์เอ็มแอล และเขียนรหัสคำสั่งภาษาจาวาเพื่อควบคุมการทำงาน ทำให้ผู้ที่ไม่มีความรู้ในการเขียนโปรแกรมสามารถพัฒนาแอปพลิเคชันสำหรับทำงานบนระบบปฏิบัติการแอนดรอยด์ได้ ถึงแม้การพัฒนาแอปพลิเคชันใน AI2 ทำได้โดยง่าย (Pokress and Veiga, 2013) แต่นักพัฒนา ก็สามารถสร้างแอปพลิเคชันที่มีประสิทธิภาพ และทำงานได้อย่างหลากหลาย ดังตัวอย่างที่ได้กล่าวมาในบทความ อีกทั้งยังมีชุมชน MIT App Inventor ที่คอยให้ความช่วยเหลือ และแลกเปลี่ยนเทคนิคใหม่ๆ ในการพัฒนาแอปพลิเคชัน ทำให้ MIT App Inventor เป็นที่นิยมเพิ่มขึ้นเป็นอย่างมากในปัจจุบัน ซึ่งผู้เขียนคิดว่าในอนาคต MIT App Inventor จะถูกพัฒนาและปรับปรุงให้มีความสามารถในการพัฒนาแอปพลิเคชันมากกว่าปัจจุบัน ซึ่งจะเป็นการปฏิวัติรูปแบบการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์ให้มีความง่ายและมีประสิทธิภาพมากยิ่งขึ้น นอกจากการใช้ AI2 ในการพัฒนาแอปพลิเคชันใช้งานจริง การนำ AI2 มาใช้ในการเรียนการสอนจะทำให้นักเรียนนักศึกษา มีโอกาสใหม่ๆ ในการพัฒนาทักษะที่จำเป็นในศตวรรษที่ 21 โดยเฉพาะความร่วมมือและการสื่อสารกับชุมชนนักพัฒนาทั้งภายในประเทศและต่างประเทศ เพื่อมีส่วนร่วมในการพัฒนาแอปพลิเคชันเพื่อแก้ปัญหาในโลกแห่งความเป็นจริง

## เอกสารอ้างอิง

ขั้นตอนการพัฒนาแอปพลิเคชันแอนดรอยด์. ม.ป.ป. สืบค้นเมื่อวันที่ 24 พฤศจิกายน 2559 จาก : <http://androidthai.in.th/conternt-android/108-step-developer-application-android.html>

ประสงค์ วงศ์ชัยบุตร. 2558. การพัฒนาโปรแกรมช่วยติดตั้งจานดาวเทียมบนระบบปฏิบัติการแอนดรอยด์. 2558. การประชุมวิชาการงานวิจัยและพัฒนาเชิงประยุกต์ ครั้งที่ 7 (ECTI-CARD 2015) เรื่อง **รู้ค่าพลังงานและสิ่งแวดล้อมเพื่อก้าวสู่ศตวรรษที่ 21**. วันที่ 8 กรกฎาคม 2558 จัดโดยมหาวิทยาลัยเทคโนโลยีราชมงคลศรีวิชัย วิทยาเขตตรัง และสมาคมวิชาการไฟฟ้า อิเล็กทรอนิกส์ โทรคมนาคม และสารสนเทศ

มงคล สีมาวงษ์ และนภาพร วิสิฐพงษ์พันธ์. 2558. ระบบตรวจจับกลิ่นและแอปพลิเคชันติดตามกลิ่นบนโทรศัพท์สมาร์ทโฟน. **การประชุมวิชาการระดับชาติด้านคอมพิวเตอร์และเทคโนโลยีสารสนเทศ ครั้งที่ 11 (NCCIT2015)**. วันที่ 2 – 3 กรกฎาคม 2558 ณ โรงแรมอโนมา กรุงเทพฯ จัดโดยคณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

วิวัฒน์ มีสุวรรณ. 2555. “ปัญญาสะสม” บนสังคมออนไลน์. **วารสารศึกษาศาสตร์ มหาวิทยาลัยนเรศวร** 14(1): 1-10.

App Inventor capabilities and limitations. n.d. Retrieved November 24, 2016, from <https://sites.google.com/site/appinventor/capabilities-limitations>

App of the Month winners. n.d. Retrieved November 24, 2016, from <http://appinventor.mit.edu/explore/app-month-gallery.html>

Connect App Inventor to MySQL Database. n.d. Retrieved November 24, 2016, from <http://www.newthinktank.com/2014/05/connect-app-inventor-mysql-database>

Component types. n.d. Retrieved November 24, 2016, from <http://ai2.appinventor.mit.edu/reference/components/>

Creative Commons Attribution 4.0 International. ม.ป.ป. สืบค้นเมื่อวันที่ 17 มีนาคม 2560 จาก : <https://creativecommons.org/licenses/by/4.0/>

Guideline สำหรับการเริ่มต้นเขียนแอปพลิเคชันแอนดรอยด์. ม.ป.ป. สืบค้นเมื่อวันที่ 24 พฤศจิกายน 2559 จาก : <http://www.akexorcist.com/2014/05/android-dev-tips-guideline.html>

How to control arduino board using an android phone. 2013. Retrieved November 24, 2016, from <https://blog.arduino.cc/2013/07/18/how-to-control-arduino-board-using-an-android-phone/>

Hybrid & native application development. ม.ป.ป. สืบค้นเมื่อวันที่ 17 มีนาคม 2560 จาก : <https://www.codebee.co.th/อะไรคือ-hybrid-vs-native-application.html>

MIT App Inventor. n.d. Retrieved November 28, 2016, from <http://appinventor.mit.edu/explore/>

MIT App Inventor Classic RIP. 2015. Retrieved November 24, 2016, from <http://appinventor.mit.edu/explore/blogs/hal/2015/07/mit-app.html>

MIT App Inventor Facebook. n.d. Retrieved November 24, 2016, from <https://www.facebook.com/MITAppInventor/>

MIT App Inventor stories. n.d. Retrieved November 24, 2016, from <http://appinventor.mit.edu/explore/stories.html>

MIT App Inventor Support Forum. n.d. Retrieved November 24, 2016, from <https://groups.google.com/forum/##forum/mitappinventortest>

Pokress, S.C. and Veiga, J.J.D. 2013. **MIT App Inventor enabling personal mobile computing**. arXiv preprint arXiv: 1310.2830v2. 24 Oct 2013.

Sharing and packaging apps. n.d. Retrieved November 24, 2016, from <http://appinventor.mit.edu/explore/ai2/share.html>

The App Inventor environment. n.d. Retrieved November 24, 2016, from [http://www.appinventor.org/HelloPurr2](http://www.appinventor.org>HelloPurr2)

Xie, B., Shabir, I., and Abelson, H. 2015. Measuring the usability and capability of app inventor to create mobile applications. **Proceedings of the ACM SIGPLAN Conference on Systems, Programming, Languages and Applications: Software for Humanity (SPLASH)**. October 25-30, 2015 in Pittsburgh, Pennsylvania, United States.