

Workshop Block-Set

ที่มาของ Block-Set

เรียนรู้วิธีการสร้างระบบควบคุมด้วยไมโครคอนโทรลเลอร์ จะต้องใช้เวลาการเรียนรู้มาก (Learning Curve) เพราะมันประกอบไปด้วยศาสตร์หลายอย่าง มารวมๆกัน

1. อิเล็กทรอนิกส์เบื้องต้น ต่อรูจักอุปกรณ์ และ วิธีการนำไปใช้
2. การเขียนโปรแกรม ทั้ง syntax ทั้ง รูปแบบภาษา ที่นำไปใช้ ต้องคุ้นเคยและแก้ปัญหาได้
3. ไอเดียสร้างสรรค์ สิ่งนี้ ต้องรู้ ต้องเห็นให้มาก ถึงจะเข้าใจ และเกิดไอเดียนำไปประยุกต์ได้

ดังนั้นถ้าผู้เริ่มต้นจะเข้าใจ ทุกเรื่องได้ ต้องใช้เวลามาก มีหลายคนอาจจะยอมแพ้ไปก่อน หรือ มารีบหัดเรียนรู้ ตอนที่จะใช้คิดว่ามีตัวอย่างเยอะในเน็ต มันก็ไม่ทันแล้ว เนื่องด้วยเหตุผลต้องใช้เวลาการเรียนรู้มากดังกล่าว

จากประสบการณ์ที่ทางผมจัด workshop อบรม arduino เบื้องต้น มาหลายครั้ง ทำให้เข้าใจเวลาส่วนใหญ่ที่ใช้ไป ส่วนมากจะเสียเวลาไปกับ การตรวจสอบการต่อวงจรของผู้อบรม และ การเขียนโปรแกรม ผู้ศึกษาที่ยังจดจำ syntax ไม่ได้ ไม่เข้าใจเรื่องประกาศตัวแปร ทำให้ workshop อาจใช้เวลาหลายชั่วโมง ดังนั้นคนที่เพิ่งเข้ามาหัดเขียนโปรแกรมเลย จะทำให้เข้าใจส่วนนี้ได้เลยเป็นเรื่องยากมาก

อุปกรณ์ชุดนี้ ตั้งใจจะทำให้ผู้เริ่มต้น ได้ศึกษาการควบคุมด้วยไมโครคอนโทรลเลอร์ ได้อย่างรวดเร็ว โดยตัดส่วนของต่ออุปกรณ์ลงไป ใช้เป็น shield สำหรับชุดทดลอง Fun Basic I/O Shield บน Arduino และ ใช้ ArduBlock แทน การเขียน code แบบเดิม เพื่อลดเวลาการผิดพลาดในการ Debug โปรแกรมลง

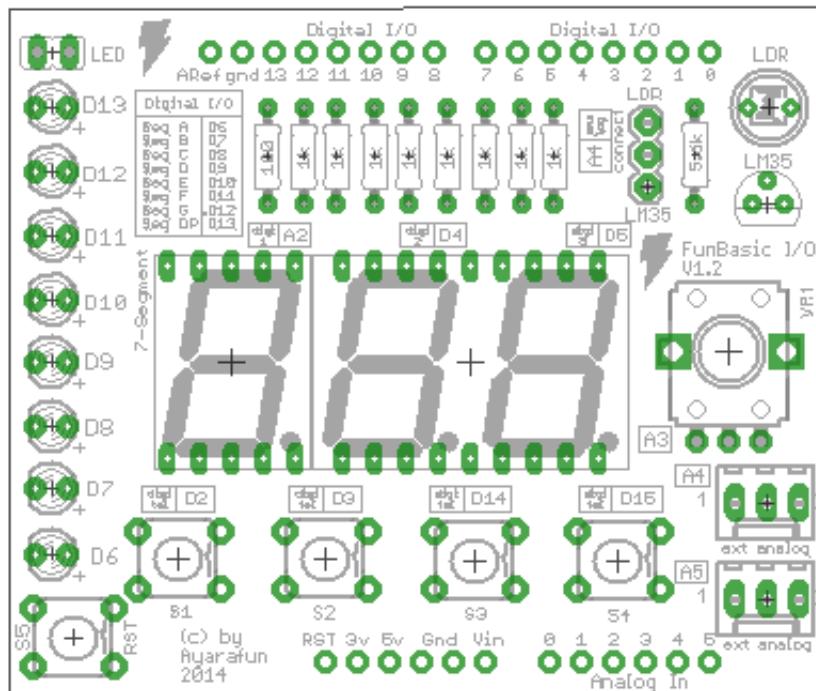
คำเตือน

Ardublock เป็นการเขียนโปรแกรม แบบ script เมื่อเข้าใจควบคุมไมโครคอนโทรลเลอร์ด้วย Workshop นี้แล้ว ผู้ศึกษาก็ต้องไปขยายผลต่อ ไปศึกษาวิธีการเขียนโปรแกรมที่ถูกต้อง ไม่ว่าจะ เป็น c/c++ มันจะ ยืดหยุ่นกว่า สามารถเขียนโปรแกรม ควบคุมอุปกรณ์ได้มากกว่า ชุด shield ที่ ออกแบบไว้

Fun Basic I/O

เป็นชุดทดลอง ที่รวมกับในรูปแบบ shield ของ arduino ในการฝึกหัดการเขียนโปรแกรมไมโครคอนโทรลเลอร์ ประกอบไปด้วย

- LED 8 ดวง
- 7-Segment 3 หลัก
- ปุ่มกด 4 ดวง
- Light Sensor 1 ชุด
- Temperature Sensor 1 ชุด
- Potentiometer 1 ชุด

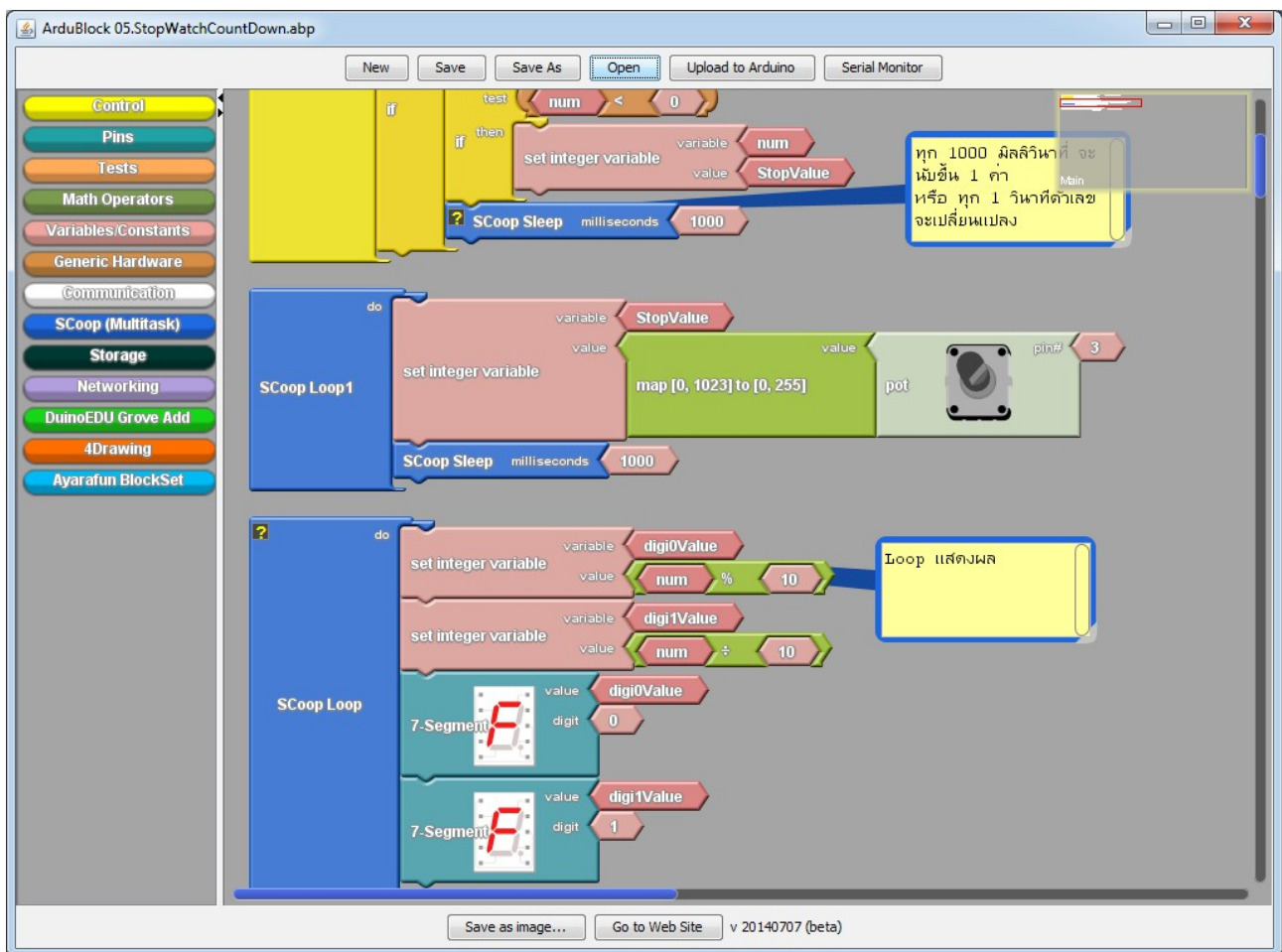


+

Ardublock

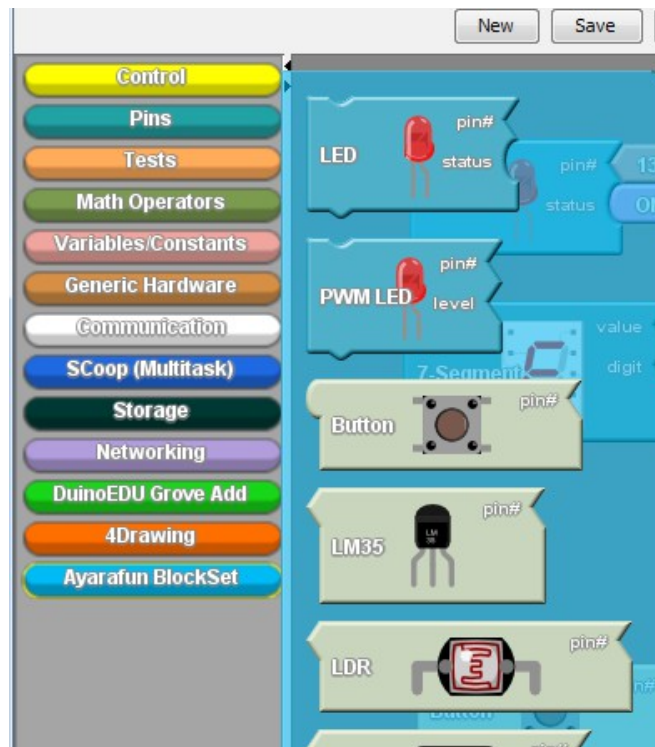
ArduBlock เป็นโปรแกรมภาษาคอมพิวเตอร์ ที่ใช้การเขียนโปรแกรมในรูปแบบกราฟิก (Graphical Programming Language) ลาก คลิก และวาง แทนการเขียนโปรแกรมด้วย code ที่ละบรรทัด เพื่อลดความซับซ้อนเกินความจำเป็นของผู้เริ่มศึกษา ไมโครคอนโทรลเลอร์

ArduBlock ถูกพัฒนาโดยนาย [David Li](#) โดยพัฒนาต่อมาจาก OpenBlocks ของ Media Lab MIT ซึ่งพัฒนาบน Java เช่นเดียวกัน จึงทำให้ OpenBlock กับ Arduino เชื่อมต่อกันได้ดี โดย OpenBlock ถูกนำไปใช้ใน App Inventor เช่นกัน จึงทำให้ หน้าตาเหมือนกัน โดยไม่ต้องสงสัย ซึ่งนอกจาก OpenBlock แล้ว ยังมี Scratch ที่มีหน้าตา หลักการเดียวกันเลย ซึ่งก็มาจาก เช่นเดียวกัน

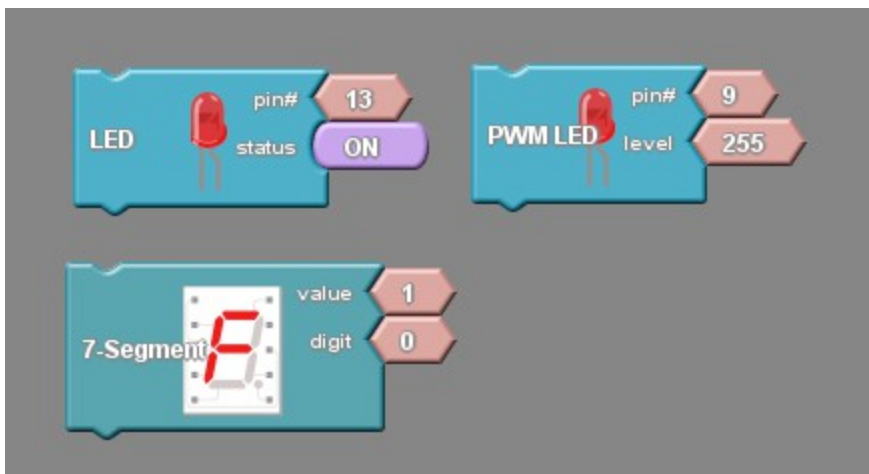


การโปรแกรมภาษา ด้วยภาพกราฟิก จะทำให้เราเข้าใจโปรแกรมเป็นโมดูลภาพ จะเข้าใจว่าแต่ละโมดูลควบคุมด้วยคอมพิวเตอร์ จะต้องประกอบไปด้วยอะไรบ้าง ซึ่งทำให้ผู้ที่ไม่มีทักษะทางคอมพิวเตอร์ หรือ อยู่ในสายงานเกี่ยวข้องกับคอมพิวเตอร์ ได้เริ่มต้นศึกษาเขียนโปรแกรมบน ไมโครคอนโทรลเลอร์ และ Arduino ได้รวดเร็วยิ่งขึ้น

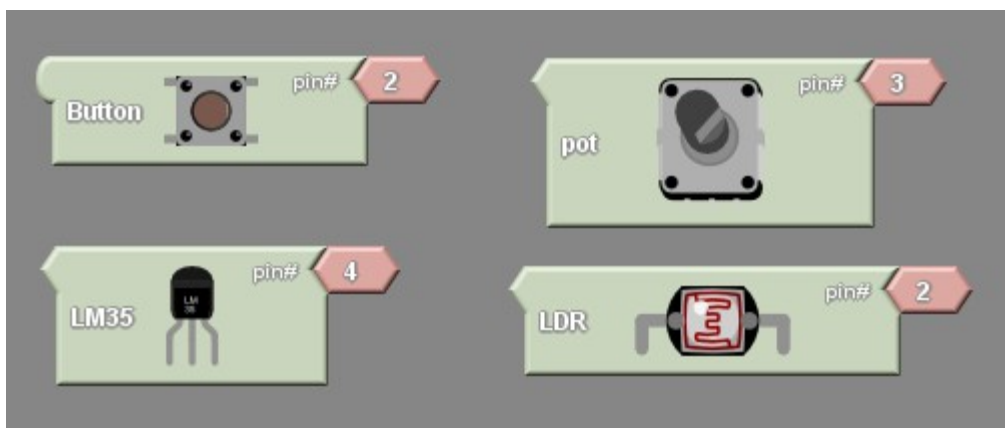
Bricks ในส่วนนี้ทางเราได้ทำ Block ขึ้นมาใหม่ ชื่อ Ayarafun BlockSet สำหรับคนที่ใช้ ArduBlock ซึ่งจะช่วยให้เราได้ทดลองอะไรได้อย่างรวดเร็วยิ่งขึ้น



บล็อกของ Hardware



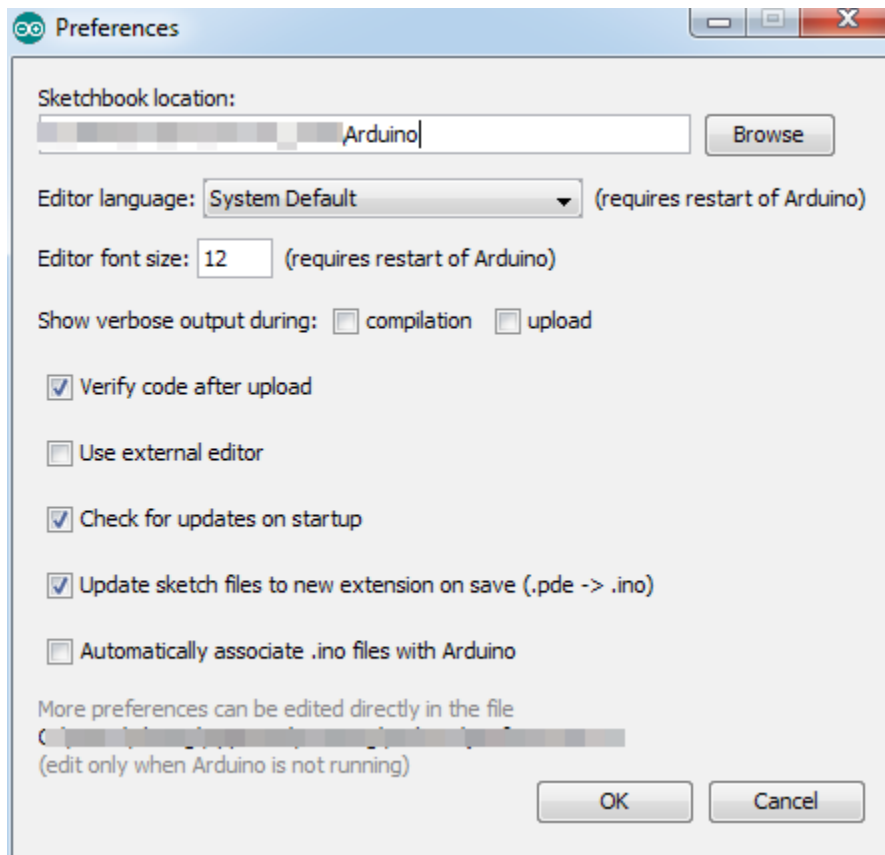
บล็อกอ่านค่าจาก Sensor



การติดตั้ง

Ardublock เป็นโปรแกรมประเภท Third-party จะทำงานร่วมกับ Arduino

1. ดาวน์โหลดไฟล์ ardublock-all.jar จาก ArduBlock หรือ ถ้าใช้ ArduBlock
2. คลิก เมนู Open / Preferences จากนั้นมองหา "Sketchbook location"



3. สร้าง Path `"/tools/ArduBlockTool/tool/"` อ้างอิงจากใน "Sketchbook location" แล้วนำไฟล์ `ardublock-all.jar` ใส่ในโฟลเดอร์ ดังกล่าว

ตัวอย่าง สมมติว่า ผู้ใช้ชื่อ "[user]"

- สำหรับ Mac, `/Users/[user]/Documents/Arduino/tools/ArduBlockTool/tool/ardublock-all.jar`
- สำหรับ Linux, `/home/[user]/sketchbook/tools/ArduBlockTool/tool/ardublock-all.jar`
- สำหรับ Windows, `C:\Users\[user]\Documents\Arduino`

*ตัวอักษรเล็กใหญ่มีผลนะครับ

Ex 1. Blinky LED (\example\01.Basics\01.Blink.abp)

Blinking LED คือ ไฟกระพริบ การหัดเขียนโปรแกรม เราจะเริ่มต้นทดสอบการเขียนโปรแกรมง่าย ให้ได้ Output ออกมาอย่างรวดเร็ว เหมือนกับ ในทางคอมพิวเตอร์เราจะเขียนโปรแกรม “Hello World” ให้คอมพิวเตอร์ส่งข้อความออกมาทักทายทางจอภาพให้เร็วที่สุด ก็เพื่อให้เราคุ่นเคยกับการเขียนโปรแกรมก่อน

Sketch with Block

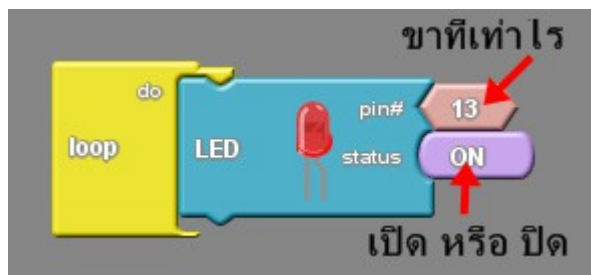
วิธีการใช้งาน ArduBlock คำสั่งในบล็อกจะเป็นรูปแบบภาพ ลากมาประกอบกัน โดยทุกโปรแกรม จะต้อง มี **Loop “do ...”** หรือ **Program “setup... loop...”** ดังภาพ



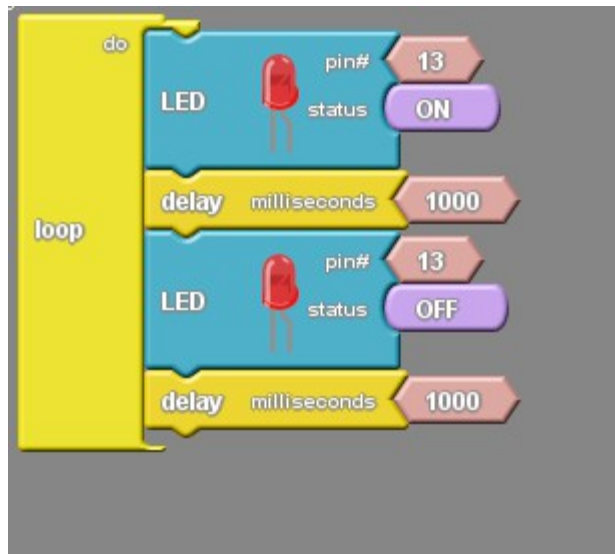
Block Program จะมีส่วน Setup เพิ่มขึ้นมา เพื่อให้สามารถ initial ค่าต่างๆ ซึ่งในโปรแกรมจะเข้าไปส่วนของ Setup แต่ครั้งแรกครั้งเดียว ส่วน loop จะเป็นส่วนที่บรรจุโปรแกรมการทำงานหลัก

Block Loop จะเป็น Block ที่ตัดส่วน setup เข้าไป จะเป็นส่วนที่บรรจุโปรแกรมการทำงานหลัก ในหนึ่งโปรแกรมจะมี Loop ได้เพียงชิ้นเดียว

จากนั้น ใน Ayarafun BlockSet เลือกบล็อก LED ใส่ในบล็อก Loop ดังรูป ให้เราตั้งค่า status ว่าจะให้ LED เปิดไฟ (ON) และ ปิดไฟ (OFF) ในหนึ่งบล็อกคือ หนึ่งคำสั่ง



สำหรับไฟกระพริบ คือการเปิดไฟ หน่วงเวลาให้ไฟติด และ ปิดไฟ หน่วงเวลาให้ไฟดับ ดังรูป จะทำงานจากบล็อกแรกก่อน และ ทำงานที่บล็อกถัดไป

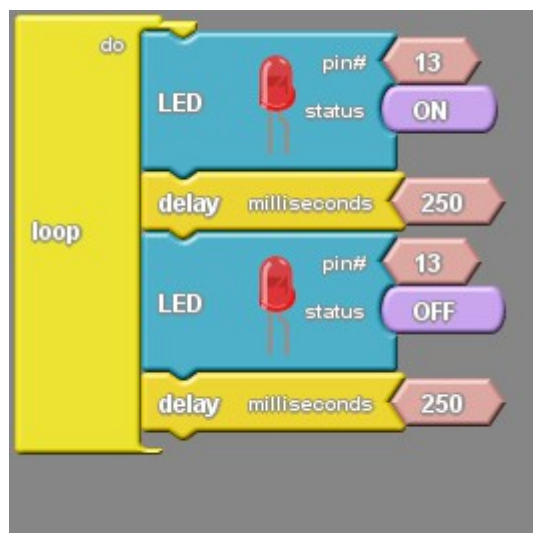


เอาล่ะครับ ตอนนี้เราได้ไฟกระพริบแล้ว

Ex 2. Blinky LED II – Too Fast, Too Slow

รอบนี้ เราตั้งใจง่ายๆ จะทำอย่างไรให้ ไฟมันกระพริบ มันกระพริบช้าหรือเร็วขึ้นได้ ไฟจะติดนาน หรือ ช้า เกิดจากจังหวะการหน่วงเวลา

ฉะนั้น ถ้าเราอยากเปลี่ยนให้มันกระพริบเร็วหรือช้า ให้เปลี่ยนค่า ที่ต่อท้าย Block Delay จาก 1000 เป็นค่า 250

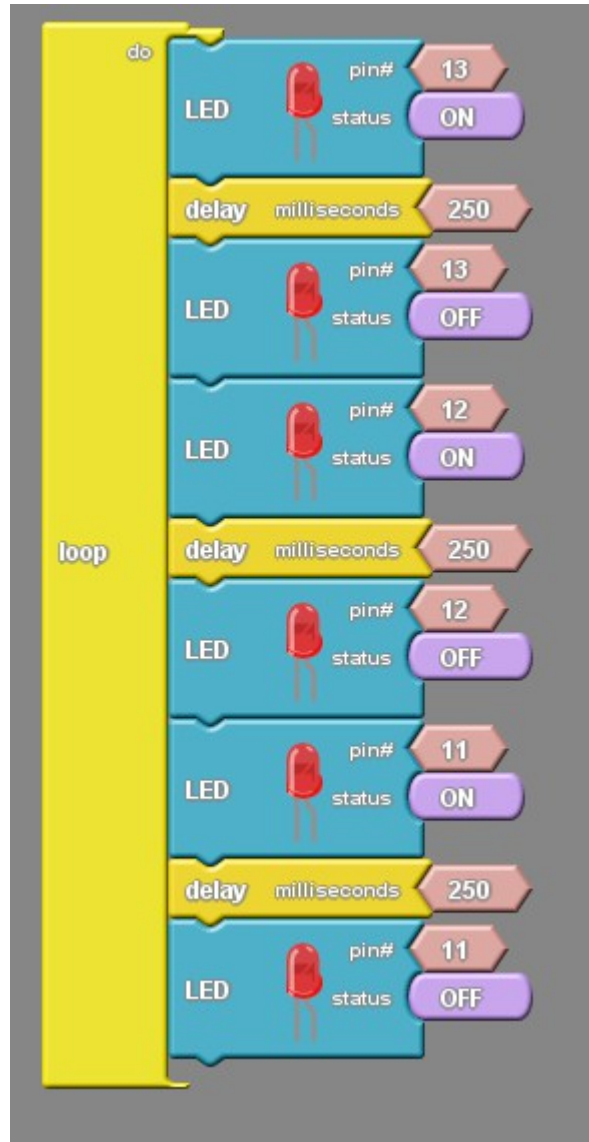


แล้ว Upload เข้าไปยังบอร์ด ไฟจะกระพริบถี่ขึ้นแล้วนะครับ

มาถึงตอนนี้ เรามาลองเพิ่มเติมอีกนิดหน่อยครับ ถ้าอยากจะให้กระพริบที่ตำแหน่งอื่น แต่เปลี่ยนจาก Pin# 13 ไปเป็น Pin อื่น 6-13

Ex. 3.1 LED Chasing (example\01.Basics\02.ledChasing.abp)

ไฟวิ่ง ไฟดาวตก จะเป็นไฟที่เราเห็นได้ทั่วไป ตามหลักการ มันการเปิดปิดไฟ แบบมีทิศทาง สมมติว่ามันวิ่งไปข้างหน้า ดวงที่อยู่ข้างหน้าถ้าดับไปจะสว่าง ดวงที่อยู่ตามหลังจะดับไป จึงทำให้ออกมาเป็น Sketch Block ดังรูป

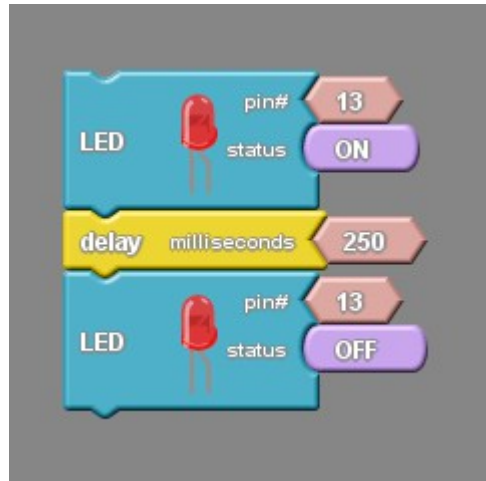


ลอง Upload ดู ตอนนี้ เราจะได้ ไฟวิ่งแบบ 3 ดวง ครับ ลองคิดว่า ถ้าจะทำไฟวิ่ง มากกว่า 3 ดวง block จะต่อยาวไปขนาดไหน

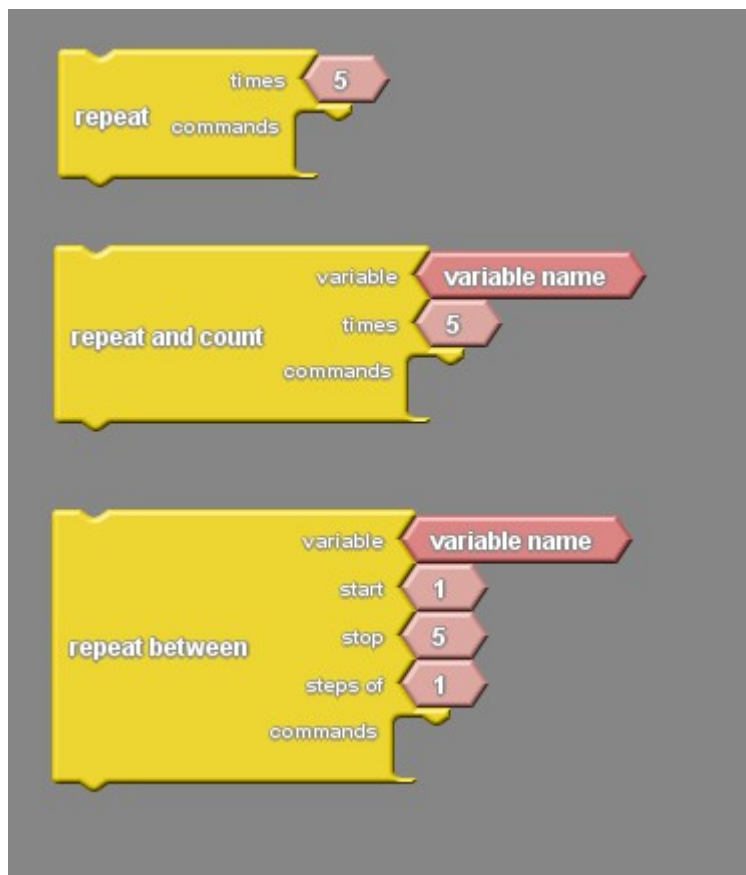
Ex 3.2 LED Chasing V2

จากตัวอย่างที่ผ่านมา จะเห็นว่า ถ้าจะทำไฟวิ่ง 8 ดวงบล็อกโปรแกรมเราจะยาวมาก มาถึงตัวอย่างนี้ เราจะบอกเทคนิคการเขียนโปรแกรมกันบ้าง

จากบล็อกดังกล่าว เราจะเห็นไอดีเดียวว่า ไฟวิ่ง จะเกิดจากการเปิดไฟดวงที่ 1 หน่วงเวลา และ ปิดไฟดวงที่ 1 ในขณะเดียวกัน ก็เปิดไฟ ดวงถัดไปด้วย

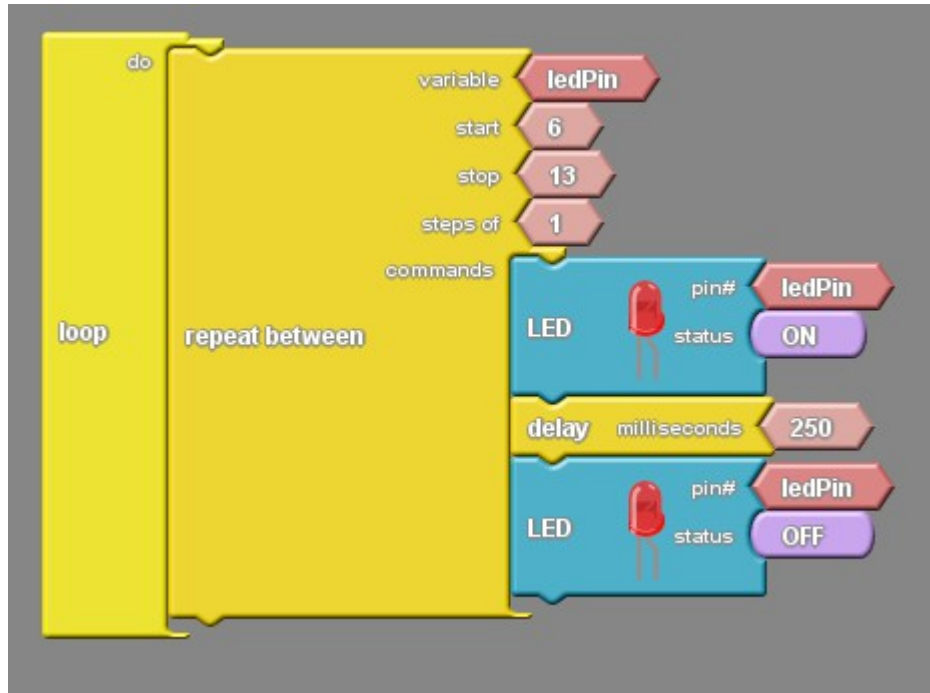


ซึ่งจะเห็นว่า มันเป็นกระบวนการซ้ำๆ ซึ่งใน ArduBlock มีฟังก์ชันสำหรับเรียกใช้งานซ้ำๆ ที่ขอบล็อกว่า Repeat โดย Block Repeat จะมีอยู่สามรูปแบบ



Block Repeat

1. Repeat แบบกำหนดจำนวนครั้ง
2. Repeat and count ตั้งค่าตัวแปร และ กำหนดจำนวนครั้งได้
3. Repeat between ตั้งค่าตัวแปร และ กำหนดจุดเริ่มต้น และ สิ้นสุดได้



ซึ่งที่เหมาะสมกับงานนี้ น่าจะ **repeat between** โดยเราตั้งชื่อ ตัวแปร (variable) ชื่อ ledPin แทนค่า 6 ถึง 13 และในส่วนของ ช่องคำสั่ง (command) เรานำ block ที่ควบคุมไฟ เปิด และ ปิด ใส่ลงไป ในตอนนีไฟจะวิ่งจาก ตำแหน่ง D6 ถึง D13 ได้แล้ว

ทิศทางไฟวิ่ง

