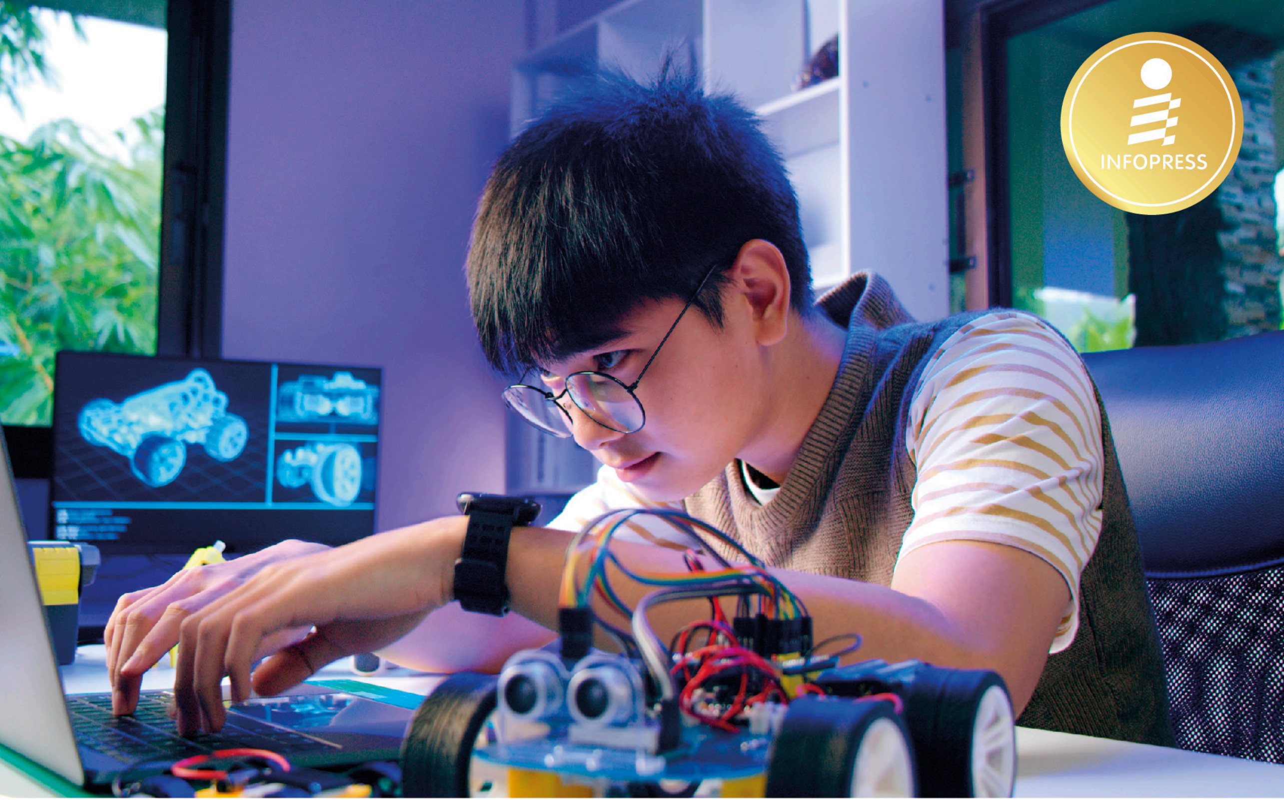


INFOPRESS

Practical Microcontroller Programming with ARDUINO

เรียนง่ายเป็นเร็ว

เปลี่ยน Arduino ให้เป็นเรื่องง่าย ให้ทุกคนเรียนได้อย่างรวดเร็ว

หนังสือเรียนไมโครคอนโทรลเลอร์ และการเขียนโปรแกรมควบคุมการทำงานด้วย Arduino เหมาะสำหรับผู้เริ่มต้นที่ไม่มีพื้นฐานมาก่อน

ผู้แต่ง ทศพล บ้านคลองสี่
อาจารย์มหาวิทยาลัยและวิทยากรพิเศษทั้งภาครัฐและเอกชน
เจ้าของเพจ/ยูทูบ : Aj. NesT The Series และ GlurGeek.com

Source Code
SERAZU.COM

IDC
PREMIER
มีเพียง “ความรู้” เท่านั้นที่มนุษย์ใช้พลิก “โลก”
และเปลี่ยนชีวิต เราจึงสร้างสรรค์ และส่งมอบ “ความรู้”
ในรูปแบบที่ดีกว่า เพื่อให้คนไทย “เรียนรู้” ได้ตลอดชีวิต
Only “Knowledge” can help human
change “The World” and “Their Lives”.
With this truth, it drives us to deliver
“Knowledge” for Thai being able to
“Learn” better everyday.
Think Beyond
INFOPRESS
Read comics
Little HEART

Practical Microcontroller & Programming with ARDUINO เรียนง่ายเป็นเร็ว

Writer	อาจารย์ทศพล บ้านคลองสี่
Editor	ภีรพล คชาเจริญ
Graphic Designer	ชวนันท์ รัตนะ
Page Layout	สุรัสวดี วงศ์จันทร์สุข
Proofreader	สุนทรี บรรลือศักดิ์
Publishing Coordinators	วรพล ณธิกุล, สุพัตรา อาจปรุ, ศรัณย์ คมขำ

โปรแกรม Arduino IDE เป็นเครื่องหมายการค้าของบริษัท Arduino LL, โปรแกรม Tinkercad เป็นเครื่องหมายการค้าของบริษัท Autodesk, Inc. และเครื่องหมายการค้าอื่นๆ ที่อ้างถึงเป็นของบริษัทนั้นๆ

สงวนลิขสิทธิ์ตามพระราชบัญญัติลิขสิทธิ์ พ.ศ. 2537 โดยบริษัท ไอดีซี พรีเมียร์ จำกัด ห้ามลอกเลียนไม่ว่าส่วนใดส่วนหนึ่งของหนังสือเล่มนี้ ไม่ว่าในรูปแบบใดๆ นอกจากจะได้รับอนุญาตเป็นลายลักษณ์อักษรจากผู้จัดพิมพ์เท่านั้น

บริษัท ไอดีซี พรีเมียร์ จำกัด จัดตั้งขึ้นเพื่อเผยแพร่ความรู้ที่มีคุณภาพสู่ผู้อ่านชาวไทย เรายินดีรับงานเขียนของนักวิชาการและนักเขียนทุกท่าน ท่านผู้สนใจกรุณาติดต่อผ่านทางอีเมลที่ infopress@idcpremier.com หรือทางโทรศัพท์หมายเลข 0-2962-1081 (อัตโนมัติ 10 คู่สาย) โทรสาร 0-2962-1084

สร้างสรรค์โดย

INFOPRESS

พิมพ์ครั้งที่ 1 มีนาคม 2565

ข้อมูลทางบรรณานุกรม

อาจารย์ทศพล บ้านคลองสี่

Practical Microcontroller & Programming with ARDUINO เรียนง่ายเป็นเร็ว

นนทบุรี : ไอดีซีฯ, 2565

334 หน้า

1. การเขียนโปรแกรมโดยใช้ภาษาโปรแกรมเฉพาะชนิด

I ชื่อเรื่อง

005.262

Barcode 885-916-100-945-0

ราคา 365 บาท

จัดพิมพ์และจัดจำหน่ายโดย

IDC PREMIER

บริษัท ไอดีซี พรีเมียร์ จำกัด

200 หมู่ 4 ชั้น 19 ห้อง 1901
อาคารจัสมินอินเตอร์เนชั่นแนลทาวเวอร์
ถ.แจ้งวัฒนะ อ.ปากเกร็ด จ.นนทบุรี 11120
โทรศัพท์ 0-2962-1081 (อัตโนมัติ 10 คู่สาย)
โทรสาร 0-2962-1084

สมาชิกสัมพันธ์

โทรศัพท์ 0-2962-1081-3 ต่อ 121
โทรสาร 0-2962-1084

ร้านค้าและตัวแทนจำหน่าย

โทรศัพท์ 0-2962-1081-3 ต่อ 112-114
โทรสาร 0-2962-1084

PREFACE

นับตั้งแต่ที่ Arduino เปิดตัวมา ก็ได้กลายมาเป็นหนึ่งในโอเพ่นซอร์สฮาร์ดแวร์ที่ประสบความสำเร็จอย่างมากในวงการการสร้างนวัตกรรมสิ่งประดิษฐ์ ผมได้ซื้อ Arduino ตัวแรกมาเมื่อหลายปีก่อน เพื่อนำมาใช้สอนให้กับนักศึกษาและผู้ที่สนใจด้วยความอยากรู้อยากเห็นว่า อุปกรณ์ตัวนี้มีความน่าสนใจอย่างไร ทำไมมีแต่คนพูดถึงกัน จึงได้เริ่มศึกษาค้นคว้าอย่างจริงจังจนได้มีโอกาสดูคลิปของ Massimo Banzi ผู้คิดค้น Arduino ที่ขึ้นพูดบนเวทีในงาน TED Talk ทำให้ผมรู้สึกประทับใจในเจตนารมณ์ของพวกเขาที่ว่า พวกเขากำลังพยายามสร้างไมโครคอนโทรลเลอร์ที่มีขนาดกะทัดรัด ราคาไม่แพง ใช้งานง่าย สามารถเชื่อมต่อกับอุปกรณ์อิเล็กทรอนิกส์พื้นฐาน และอุปกรณ์เซนเซอร์แบบต่างๆ ได้

โปรแกรมที่ใช้ในการเขียนโปรแกรมสำหรับสั่งงานบอร์ด Arduino ก็สามารถดาวน์โหลดได้ฟรี พร้อมทั้งรองรับการเขียนโปรแกรมภาษาซี ซึ่งเป็นภาษาพื้นฐานที่เหมาะกับผู้เริ่มต้นในการพัฒนาโปรแกรมทางด้านฮาร์ดแวร์ที่เรียนรู้ได้ง่าย เราสามารถหาซื้อบอร์ดนี้ได้ทั่วไป อีกทั้งยังมีแหล่งข้อมูลให้ศึกษา และสอนวิธีการใช้งานมากมายบนโลกออนไลน์ ทั้งจากเว็บ arduino.cc ที่เป็นของผู้พัฒนาเอง และแหล่งชุมชนต่างๆ ซึ่งเป็นสิ่งที่ดีมากสำหรับผู้เริ่มต้นที่อยากจะเรียนรู้การเขียนโปรแกรมร่วมกับฮาร์ดแวร์ เพื่อสร้างนวัตกรรมสิ่งประดิษฐ์ได้ด้วยตัวเอง

Arduino จึงเป็นบอร์ดที่จะช่วยสร้างแรงบันดาลใจใหม่ๆ ให้กับคนธรรมดาอย่างเราๆ ทั้งเด็กและผู้ใหญ่ ให้สามารถกล้าคิดค้น กล้าใส่ความคิดสร้างสรรค์ในการออกแบบ ส่งเสริมการพัฒนาทักษะการเขียนโปรแกรม เพื่อสร้างนวัตกรรมสิ่งประดิษฐ์ใหม่ๆ แบบง่ายๆ ได้ด้วยตัวเอง ตั้งแต่งานศิลปะเชิงโต้ตอบไปจนถึงการสร้างหุ่นยนต์ก็สามารถทำได้จริง

ดังนั้น หนังสือเล่มนี้ได้เรียบเรียงเนื้อหามาเพื่อสร้างแรงบันดาลใจ และประโยชน์ให้กับผู้เริ่มต้นพัฒนาโปรแกรมในการจะก้าวเข้าสู่นักประดิษฐ์นวัตกรรมรุ่นใหม่ โดยเน้นให้ผู้เรียนได้เข้าใจภาพรวมของ Arduino ตั้งแต่จุดเริ่มต้นไปจนถึงตัวอย่างการสร้างนวัตกรรมสิ่งประดิษฐ์ใหม่ๆ ที่น่าสนใจ รู้จักบอร์ดมาตรฐานของ Arduino และอุปกรณ์เสริมประเภทต่างๆ ตลอดจนการเลือกซื้อ Arduino ให้ตรงกับงานที่ต้องการประดิษฐ์ รู้จักการใช้งานซอฟต์แวร์เพื่อการเรียนรู้ Arduino แบบจำลองทั้งการออกแบบ การต่อวงจร และการเขียนโปรแกรมเพื่อจำลองการทำงานของ Arduino แบบออนไลน์ โดยไม่จำเป็นต้องใช้บอร์ดจริงเพื่อทดสอบแนวความคิด วิธีการ และผลการทดลองก่อนลงทุนซื้ออุปกรณ์จริง

ผู้เรียนจะได้เรียนรู้การใช้งานบอร์ด Arduino จริง ตั้งแต่เริ่มใช้งานครั้งแรกไปจนถึงการแสดงผลแบบต่างๆ และส่วนที่สำคัญที่สุด คือ การเขียนโปรแกรมควบคุมการทำงาน ซึ่งผมได้พยายามนำเอาประสบการณ์ที่มีทางด้านการสอนการเขียนโปรแกรม มาถ่ายทอดเรียบเรียงเป็นข้อความและรูปภาพ เพื่อให้เรียนได้ด้วยตัวเองแบบเห็นภาพที่เข้าใจง่ายที่สุด โดยเจาะลึกการเขียนโปรแกรม ตั้งแต่การรู้จักกลุ่มของคำสั่งต่างๆ ของ Arduino ที่มีให้ใช้งาน วิธีการเขียนโปรแกรม Arduino Sketch การใช้งานตัวแปร ชุดอักขระ ชนิดข้อมูลและตัวแปร ตัวดำเนินการ การใช้คำสั่งทำซ้ำ การใช้คำสั่งเงื่อนไข การใช้งานอาร์เรย์ และการสร้าง Functions ด้วยตัวเอง โดยถ่ายทอดออกมาเป็นบทเรียน Arduino Tutorials ภาคปฏิบัติ ที่สามารถเขียนโปรแกรมไปพร้อมกันแบบ Step-by-Step พร้อมด้วยรูปภาพ Flowchart เพื่อให้ผู้เรียนเห็นภาพการทำงานของโปรแกรมได้ด้วยตัวเอง

ผมหวังว่า หนังสือเล่มนี้จะเป็นจุดเริ่มต้นและแรงบันดาลใจที่ดีให้กับผู้เรียน ที่อยากจะเริ่มต้น และต้องการพัฒนาทักษะการเขียนโปรแกรมของตนเองให้ดียิ่งขึ้น และสามารถนำความรู้และทักษะที่ได้ไปต่อยอดในการสร้างโครงงานนวัตกรรมสิ่งประดิษฐ์ เพื่อส่งเสริมทางด้านการเรียน การประกอบธุรกิจ และการสร้างสรรค์สังคมให้ดียิ่งๆ ขึ้นได้

อาจารย์ทศพล บ้านคลองสี่
อาจารย์มหาวิทยาลัยและวิทยากรพิเศษทั้งภาครัฐและเอกชน
เจ้าของเพจ/ยูทูบ : Aj. NesT the Series และ GlurGeek.com

EDITOR'S NOTE

การเป็นบรรณาธิการหนังสือเล่มนี้ ทำให้ผมได้มีโอกาสรื้อฟื้นความรู้ในสาขาที่จบมา นั่นคือ สาขาอิเล็กทรอนิกส์ เอ็นจิเนียริ่ง พอได้อ่านต้นฉบับจบก็ยิ่งทำให้ผมสนุกเป็นอย่างยิ่ง เพราะได้เห็นว่าอุปสรรคต่างๆ ที่ผมเคยประสบในสมัยเรียนได้ถูกขจัดออกไปเสียจนหมดสิ้น จากศาสตร์ที่เคยเป็นเรื่องเฉพาะคนที่เรียนจบมาตรงสายกลับกลายเป็นเรื่องที่เปิดกว้างสำหรับทุกๆ คน มันเป็นเช่นนั้นได้อย่างไร?

มันคงจะเป็นไปไม่ได้ถ้าไม่มีฮีโร่เหล่านี้คือ นาย Massimo Banzi และเพื่อนๆ ผู้ออกแบบบอร์ดไมโครคอนโทรลเลอร์นามว่า “Arduino” ที่ตัดเอาความซับซ้อนออกไปเพื่อให้เป็นบอร์ดที่เรียนรู้ง่าย ขจัดสิ่งที่เป็นอุปสรรคในการพัฒนาออกไปเสียจนหมดสิ้น ด้วยการทำให้ฮาร์ดแวร์และซอฟต์แวร์เป็น Open Source และสร้าง Arduino Community เพื่อเป็นแหล่งสร้างแรงบันดาลใจ แบ่งปันไอเดีย ด้วยการสร้าง Arduino Project Hub ศูนย์รวมโปรเจ็กต์ที่ช่วยบ่มเพาะไอเดียชั้นดี สำหรับคนที่ยังคิดไม่ออกว่าจะประดิษฐ์อะไรดี และ Arduino Forum แหล่งแลกเปลี่ยนความรู้ระหว่างสมาชิก

ด้วยความที่เป็นกลุ่มนักออกแบบจากประเทศอิตาลี ที่ขึ้นชื่อในเรื่องของความคิดสร้างสรรค์ จึงมีแนวทางในการพัฒนา Arduino Products ที่แตกต่างจากยุคสมัยเดิม โดยได้พัฒนา Products ที่ครอบคลุมคนทุกระดับตลอดเส้นทางการศึกษา ทั้งในระดับนักเรียน นักศึกษา นักวิจัย เมกเกอร์ (Maker) ทั้งมือสมัครเล่นและมืออาชีพ มีตั้งแต่ชุดประกอบสำเร็จรูป (Kits) มีบอร์ดหลายรุ่นที่มีฟีเจอร์และฟอร์มแฟ็กเตอร์ขนาดต่างๆ เพื่อรองรับโครงงานที่มีสเกล และความซับซ้อนที่แตกต่างกัน และกลุ่มสินค้าขั้นสูง เช่น STEAM, Robotics & Drones, Internet Of Things (IoT), Industrial Automation เป็นต้น นอกจากนี้ยังมีวงจรขยายซัพพอร์ตมากมาย

หนังสือเล่มนี้เขียนโดย อาจารย์ทศพล บ้านคลองสี่ หรือ Aj. NesT the Series จะช่วยสร้างทักษะพื้นฐานทางอิเล็กทรอนิกส์ การต่อวงจร และการเขียนโปรแกรมควบคุมการทำงาน สามารถเริ่มต้นก้าวแรกได้อย่างรวดเร็วจากหนังสือ และ VDO Clip ใน YouTube Channel รวมถึงได้รับประสบการณ์จริงจากการทำ LAB ที่มีทั้งการต่อวงจรจริง และต่อวงจรจำลองผ่าน Simulation Software ผมเชื่อว่า ผู้เรียนจะได้รับประสบการณ์การเรียนรู้ที่ครบถ้วนบริบูรณ์ที่สุดจากหนังสือเล่มนี้

What is a Maker? : http://bitly.ws/oH6R

Makerspace : https://www.thekommon.co/makerspace-for-education/

ภีรพล คชาเจริญ

บรรณาธิการ

CONTENTS

Chapter 01
ภาพรวม Arduino สำหรับผู้เริ่มต้น

Arduino คืออะไร? 2
คลิปแนะนำ Arduino ใน 15 นาที 3
ทำไมบอร์ด Arduino จึงเป็นทางเลือกที่ดีที่สุด 4
Arduino บอร์ดเพื่อการศึกษา 6
Arduino จากก้าวเล็กๆ สู่ความยิ่งใหญ่ 10
รู้จัก Arduino ผ่านการบอกเล่าของ Massimo Banzi 11
คำบรรยายจากคลิป 11
ย้อนประวัติของ Arduino 23
วิวัฒนาการของบอร์ด Arduino 24
เริ่มต้นศึกษา Arduino อย่างไรดี 25
ศึกษา Arduino Projects ใน Arduino Project Hub 25
ศึกษา Arduino Projects ใน YouTube 26
ศึกษา Arduino Projects ใน HOWTOMECHATRONICS 27
องค์ประกอบการพัฒนา Arduino Projects 28
แหล่งข้อมูลศึกษาเพิ่มเติม (Arduino Tutorials) 30
บทสรุปท้ายบท 35

Chapter 02
รู้จักบอร์ดมาตรฐานของ Arduino และอุปกรณ์เสริม

โครงสร้างของ Microcontroller 38
เปรียบเทียบ Microprocessor & Microcontroller 39
ตระกูลของบอร์ดและชิป Microcontroller 40
รู้จักส่วนประกอบของบอร์ด Arduino 41
บอร์ด Arduino UNO 41
บอร์ด Arduino DUE 48
บอร์ด Arduino LEONARDO (with Headers) 52
บอร์ด Arduino MEGA 57

บอร์ด Arduino NANO 63
ข้อมูล Sensors/Modules, Motors และ Shields 67
ประเภทของ Sensors/Modules 67
ประเภทของ Arduino Motors และ Motor Driver Modules 71
ประเภทของ Arduino Shields 72
ตัวอย่างการใช้งานบอร์ด Arduino กับ Sensors 75
บทสรุปท้ายบท 79

Chapter 03 การเลือกซื้อ Arduino ให้เหมาะกับการใช้งาน

สินค้ากลุ่มไหนเหมาะกับใคร (Arduino Products) 82
ตารางเปรียบเทียบสเปคบอร์ดรุ่นต่างๆ 86
Arduino บอร์ดแท้บอร์ดเทียมดูกันอย่างไร 88
แหล่งเลือกซื้อบอร์ด Arduino และ Sensors 92
เลือกซื้อผ่านเว็บ arduino.cc 92
เลือกซื้อใน Lazada และ Shopee 93
เลือกซื้อผ่านร้านค้าออนไลน์ 94
บทสรุปท้ายบท 95

Chapter 04 ซอฟต์แวร์เพื่อการเรียนรู้ Arduino

Fritzing (Circuit Drawing) 98
จุดเด่นของโปรแกรม Fritzing 98
ดาวน์โหลดและติดตั้ง Fritzing 99
การเขียนวงจรและไดอะแกรมการเดินสายไฟด้วย Fritzing (Making Circuit & Wiring Diagrams) 101
การปรับขนาด-หมุน Breadboard 105
Zoom-in/Zoom-out 106
ทดลองเขียนวงจรกับโปรแกรม Fritzing ครั้งแรก 107
แหล่งศึกษาการใช้งานโปรแกรม Fritzing เพิ่มเติมทางออนไลน์ 115
Tinkercad (Arduino Simulator) 115
ส่วนประกอบของเว็บเพจ Learn Arduino 117
ทดลอง LED Light Up 118
ทดลองต่อวงจร Tone Melody 120
การแก้ไขส่วนประกอบ (Editing Components) 121
การเดินสาย (Wiring Components) 122
การเพิ่มชิ้นส่วนอุปกรณ์เพิ่มเติม (Adding Components) 124
บทสรุปท้ายบท 129

Chapter 05 เริ่มใช้งานบอร์ด Arduino ครั้งแรก

ทางเลือกในการเขียนโปรแกรม 132
Arduino Sketch 133
การเขียนโปรแกรมบน Arduino IDE ครั้งแรก 135
Step 1 ดาวน์โหลดและติดตั้ง Arduino IDE 135
Step 2 รู้จักส่วนประกอบของโปรแกรม Arduino IDE 136
Step 3 ตั้งค่าการเชื่อมต่อ Arduino IDE กับบอร์ด 141
Step 3.1 กำหนดรุ่นของบอร์ดที่ใช้ให้กับ IDE 141
Step 3.2 ตั้งค่า Port ที่ใช้ติดต่อกับบอร์ด 141
Step 3.3 ทดลองเขียนโปรแกรมบน Arduino IDE ครั้งแรก 143
การเขียนโปรแกรมบน Arduino Web Editor ครั้งแรก 144
Step 1 การใช้งาน Arduino Web Editor 144
Step 2 รู้จักส่วนประกอบของ Arduino Web Editor 146
Step 3 ตั้งค่าเชื่อมต่อ Arduino Web Editor กับบอร์ด 147

Step 3.1 กำหนดรุ่นของบอร์ดและการใช้งาน Port 147
Step 3.2 ทดลองเขียนโปรแกรมบน Arduino Web Editor ครั้งแรก 147
การศึกษา Arduino ในภาคปฏิบัติ 148
สาธิตตัวอย่างการเรียนผ่าน Arduino Examples 149
สาธิตตัวอย่างการเรียนรู้จาก Arduino Library 153
วิธีติดตั้ง Library ให้กับโปรแกรม Arduino 154
Serial Monitor 161
บทสรุปท้ายบท 166

Chapter 06
การเขียนโปรแกรมควบคุมการทำงาน

6.1 กลุ่มคำสั่งมีอะไรบ้าง (Arduino Programming) 168
6.1.1 โครงสร้างทางไวยากรณ์ (Structures) 168
6.1.2 ตัวแปรและค่าคงที่ (Values ; Variables, Constants) 169
6.1.3 ฟังก์ชัน (Functions) 170
6.2 วิธีเขียนโปรแกรมแบบ Arduino Sketch 172
6.2.1 โครงสร้างการเขียนโปรแกรม Arduino Sketch 172
6.2.2 การใส่คำอธิบายในโค้ด ; // และ /* ... */ 173
6.2.3 วงเล็บปีกกา { } และวงเล็บ () 174
6.2.4 เครื่องหมายเซมิโคลอน ; 174
Arduino Tutorial 1 ทดลองเขียนข้อความบนจอ Serial Monitor 174
Step 1 : อุปกรณ์ที่ใช้ในการทดลอง 175
Step 2 : คำสั่งที่ใช้ในการทดลอง 175
Step 3 : Flowchart Arduino Tutorial 1 176
Step 4 : Source Code Arduino_Tutorial_1.ino 176
Step 5 : วิธีการทดลอง Arduino Tutorial 1 177
Step 6 : ผลการทดลอง Arduino Tutorial 1 178
Step 7 : คำถามท้าย Arduino Tutorial 1 178
Arduino Tutorial 2 ทดลองเขียนโปรแกรมไฟกะพริบ 179
Step 1 : อุปกรณ์ที่ใช้ในการทดลอง 179
Step 2 : คำสั่งที่ใช้ในการทดลอง 179
Step 3 : Flowchart Arduino Tutorial 2 180
Step 4 : Source Code Arduino_Tutorial_2.ino 181
Step 5 : วิธีการทดลอง Arduino Tutorial 2 181
Step 6 : ผลการทดลองของ Arduino Tutorial 2 182
Step 7 : คำถามท้าย Arduino Tutorial 2 185
6.3 การใช้งานตัวแปร (Arduino Variables) 185
6.3.1 ตัวแปรภายนอกฟังก์ชันและตัวแปรภายในฟังก์ชัน 185
6.3.2 การกำหนดค่าเริ่มต้นให้กับตัวแปร 186
Arduino Tutorial 3 ทดลองการใช้งานตัวแปร Global Variable และ Local Variable 187
Step 1 : อุปกรณ์ที่ใช้ในการทดลอง 187
Step 2 : คำสั่งที่ใช้ในการทดลอง 188
Step 3 : Flowchart Arduino Tutorial 3 188
Step 4 : Source Code Arduino_Tutorial_3.ino 189
Step 5 : วิธีการทดลอง Arduino Tutorial 3 189
Step 6 : ผลการทดลองของ Arduino Tutorial 3 190
Step 7 : คำถามท้าย Arduino Tutorial 3 191
6.4 การตั้งชื่อตัวแปร (Naming the Variable) 191
Arduino Tutorial 4 ทดลองการตั้งชื่อตัวแปรที่ดี 192
Step 1 : อุปกรณ์ที่ใช้ในการทดลอง 192
Step 2 : คำสั่งที่ใช้ในการทดลอง 192
Step 3 : Flowchart Arduino Tutorial 4 193
Step 4 : Source Code Arduino_Tutorial_4.ino 194
Step 5 : วิธีการทดลอง Arduino Tutorial 4 195
Step 6 : ผลการทดลอง Arduino Tutorial 4 196
Step 7 : คำถามท้าย Arduino Tutorial 4 198

6.5 ชุดอักขระ (Arduino Strings)....................... 198
Arduino Tutorial 5 ทดลองการสร้างตัวแปรแบบ Strings....198
Step 1 : อุปกรณ์ที่ใช้ในการทดลอง....198
Step 2 : คำสั่งที่ใช้ในการทดลอง....198
Step 3 : Flowchart Arduino Tutorial 5....199
Step 4 : Source Code Arduino_Tutorial_5.ino....200
Step 5 : วิธีการทดลอง Arduino Tutorial 5....201
Step 6 : ผลการทดลอง Arduino Tutorial 5....202
Step 7 : คำถามท้าย Arduino Tutorial 5....204
6.6 ชนิดข้อมูลและตัวแปร (Data Types & Variables)....204
6.6.1 การกำหนดชนิดข้อมูล (Data Type)....204
Void Type (void)....205
Boolean Type (Bool)....205
Character Type (char)....205
Unsigned Character/Byte Type (unsigned char/byte)....208
Integer Type (int)....208
Unsigned Integer Type (unsigned int/word)....208
Long Type (long)....209
Unsigned Long Type (unsigned long)....209
Float และ Double Type (float/double)....209
String Type....209
6.6.2 อักขระหลีก (Escape Character)....210
Arduino Tutorial 6 ทดลองการใช้งาน Data Types และ Escape Characters....210
Step 1 : อุปกรณ์ที่ใช้ในการทดลอง....210
Step 2 : คำสั่งที่ใช้ในการทดลอง....211
Step 3 : Flowchart Arduino Tutorial 6....211
Step 4 : Source Code Arduino_Tutorial_6.ino....212
Step 5 : วิธีการทดลอง Arduino Tutorial 6....214
Step 6 : ผลการทดลอง Arduino Tutorial 6....215
Step 7 : คำถามท้าย Arduino Tutorial 6....215
6.7 โอเปอเรเตอร์ (Arduino Operators)....216
6.7.1 ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic Operators)....216
6.7.1.1 ลำดับความสำคัญของตัวดำเนินการ....217
Arduino Tutorial 7 ทดลองใช้งานตัวดำเนินการทางคณิตศาสตร์....217
Step 1 : อุปกรณ์ที่ใช้ในการทดลอง....217
Step 2 : คำสั่งที่ใช้ในการทดลอง....217
Step 3 : Flowchart Arduino Tutorial 7....218
Step 4 : Source Code Arduino_Tutorial_7.ino....219
Step 5 : วิธีการทดลอง Arduino Tutorial 7....220
Step 6 : ผลการทดลอง Arduino Tutorial 7....221
Step 7 : คำถามท้าย Arduino Tutorial 7....222
6.7.2 ตัวดำเนินการเปรียบเทียบ (Comparison Operators)....222
6.7.2.1 โอเปอเรเตอร์ที่ใช้ในการตัดสินใจ (Conditional or Ternary Operator)....223
Arduino Tutorial 8 ทดลองการใช้งานตัวดำเนินการเปรียบเทียบ....223
Step 1 : อุปกรณ์ที่ใช้ในการทดลอง....223
Step 2 : คำสั่งที่ใช้ในการทดลอง....223
Step 3 : Flowchart Arduino Tutorial 8....224
Step 4 : Source Code Arduino_Tutorial_8.ino....225
Step 5 : วิธีการทดลอง Arduino Tutorial 8....226
Step 6 : ผลการทดลอง Arduino Tutorial 8....227
Step 7 : คำถามท้าย Arduino Tutorial 8....227
6.7.3 ตัวดำเนินการบูลีน (Boolean Operators)....228
Arduino Tutorial 9 ทดลองการใช้งานตัวดำเนินการบูลีน....228
Step 1 : อุปกรณ์ที่ใช้ในการทดลอง....228
Step 2 : คำสั่งที่ใช้ในการทดลอง....228
Step 3 : Flowchart Arduino Tutorial 9....229
Step 4 : Source Code Arduino_Tutorial_9.ino....230
Step 5 : วิธีการทดลอง Arduino Tutorial 9....231

Step 6 : ผลการทดลอง Arduino Tutorial 9 231
Step 7 : คำถามท้าย Arduino Tutorial 9 232
6.7.4 ตัวดำเนินการแบบบิตไวส์ (Bitwise Operators) 232
6.7.4.1 พื้นฐานตารางค่าความจริง (Truth Table) 232
6.7.4.2 พื้นฐานระบบเลขฐานสอง (Binary Numbers) 233
Arduino Tutorial 10 ทดลองการใช้งานตัวดำเนินการแบบบิต 235
Step 1 : อุปกรณ์ที่ใช้ในการทดลอง 235
Step 2 : คำสั่งที่ใช้ในการทดลอง 235
Step 3 : Flowchart Arduino Tutorial 10 236
Step 4 : Source Code Arduino_Tutorial_10.ino 237
Step 5 : วิธีการทดลอง Arduino Tutorial 10 238
Step 6 : ผลการทดลอง Arduino Tutorial 10 239
Step 7 : คำถามท้าย Arduino Tutorial 10 239
6.7.5 ตัวดำเนินการผสม (Compound Operators) 240
Arduino Tutorial 11 ทดลองการใช้งานตัวดำเนินการผสม 241
Step 1 : อุปกรณ์ที่ใช้ในการทดลอง 241
Step 2 : คำสั่งที่ใช้ในการทดลอง 241
Step 3 : Flowchart Arduino Tutorial 11 242
Step 4 : Source Code Arduino_Tutorial_11.ino 243
Step 5 : วิธีการทดลอง Arduino Tutorial 11 244
Step 6 : ผลการทดลอง Arduino Tutorial 11 245
Step 7 : คำถามท้าย Arduino Tutorial 11 245
6.8 การใช้คำสั่งทำซ้ำ (for loop/while loop/do while) 246
Arduino Tutorial 12 ทดลองการใช้งานคำสั่งแบบทั่วไปในการควบคุมไฟกะพริบ 246
Step 1 : อุปกรณ์ที่ใช้ในการทดลอง 246
Step 2 : คำสั่งที่ใช้ในการทดลอง 246
Step 3 : Flowchart Arduino Tutorial 12 247
Step 4 : Source Code Arduino_Tutorial_12.ino 248
Step 5 : วิธีการทดลอง Arduino Tutorial 12 250
Step 6 : ผลการทดลอง Arduino Tutorial 12 251
Step 7 : คำถามท้าย Arduino Tutorial 12 252
การเขียนโปรแกรมโดยใช้คำสั่งลูป (Loop) 252
for loop 252
while loop 253
do while loop 257
Arduino Tutorial 13 ทดลองการใช้งานคำสั่งวนซ้ำแบบ for/while/do while 260
Step 1 : อุปกรณ์ที่ใช้ในการทดลอง 260
Step 2 : คำสั่งที่ใช้ในการทดลอง 260
Step 3 : Flowchart Arduino Tutorial 13 261
Step 4 : Source Code Arduino_Tutorial_13.ino 263
Step 5 : วิธีการทดลอง Arduino Tutorial 13 266
Step 6 : ผลการทดลอง Arduino Tutorial 13 268
Step 7 : คำถามท้าย Arduino Tutorial 13 269
Arduino Tutorial 14 การปรับแต่งโปรแกรมไฟกะพริบ 270
Step 1 : อุปกรณ์ที่ใช้ในการทดลอง 270
Step 2 : คำสั่งที่ใช้ในการทดลอง 270
Step 3 : Flowchart Arduino Tutorial 14 271
Step 4 : Source Code Arduino_Tutorial_14.ino 272
Step 5 : วิธีการทดลอง Arduino Tutorial 14 273
Step 6 : ผลการทดลอง Arduino Tutorial 14 274
Step 7 : คำถามท้าย Arduino Tutorial 14 275
6.9 การใช้คำสั่งเงื่อนไข if/else 275
Arduino Tutorial 15 ทดลองใช้คำสั่ง if/else เพื่อสั่งหลอดไฟ LED ที่ต้องการให้กะพริบ 277
Step 1 : อุปกรณ์ที่ใช้ในการทดลอง 278
Step 2 : คำสั่งที่ใช้ในการทดลอง 278
Step 3 : Flowchart Arduino Tutorial 15 แบบ if Statement 279
Step 4 : Source Code Arduino_Tutorial_15.ino 280

Step 5 : วิธีการทดลอง Arduino Tutorial 15
if Statement 282
Step 6 : Flowchart Arduino Tutorial 15
แบบ if/else Statement 283
Step 7 : Source Code Arduino Tutorial 15
แบบ if/else Statement 284
Step 8 : วิธีการทดลอง Arduino Tutorial 15
if/else Statement 285
Step 9 : ผลการทดลอง Arduino Tutorial 15 286
Step 10 : คำถามท้าย Arduino Tutorial 15 290

6.10 การใช้คำสั่งเลือกแบบ switch case 290

Arduino Tutorial 16 ทดลองใช้คำสั่ง switch เพื่อสั่ง หลอดไฟ LED ที่ต้องการให้กะพริบ 291

Step 1 : อุปกรณ์ที่ใช้ในการทดลอง 292
Step 2 : คำสั่งที่ใช้ในการทดลอง 292
Step 3 : Flowchart Arduino Tutorial 16 293
Step 4 : Source Code Arduino_Tutorial_16.ino 294
Step 5 : วิธีการทดลอง Arduino Tutorial 16 296
Step 6 : ผลการทดลอง Arduino Tutorial 16 297
Step 7: คำถามท้าย Arduino Tutorial 16 302

6.11 การใช้งานอาร์เรย์ (Arrays) 302

การเรียกใช้งานตัวแปร Array 303

Arduino Tutorial 17 ทดลองใช้งาน Array ควบคุมหลอดไฟ LED 305

Step 1 : อุปกรณ์ที่ใช้ในการทดลอง 305
Step 2 : คำสั่งที่ใช้ในการทดลอง 305
Step 3 : Flowchart Arduino Tutorial 17 306
Step 4 : Source Code Arduino_Tutorial_17.ino 307
Step 5 : วิธีการทดลอง Arduino Tutorial 17 308
Step 6 : ผลการทดลอง Arduino Tutorial 17 309
Step 7 : คำถามท้าย Arduino Tutorial 17 311

6.12 การสร้าง Functions ด้วยตัวเอง 312

Arduino Tutorial 18 ทดลองสร้างและเรียกใช้งาน ฟังก์ชัน เพื่อควบคุมหลอดไฟ LED 314

Step 1 : อุปกรณ์ที่ใช้ในการทดลอง 314
Step 2 : คำสั่งที่ใช้ในการทดลอง 314
Step 3 : Flowchart Arduino Tutorial 18 315
Step 4 : Source Code Arduino_Tutorial_18.ino 316
Step 5 : วิธีการทดลอง Arduino Tutorial 18 317
Step 6 : ผลการทดลอง Arduino Tutorial 18 318
Step 7 : คำถามท้าย Arduino Tutorial 18 321

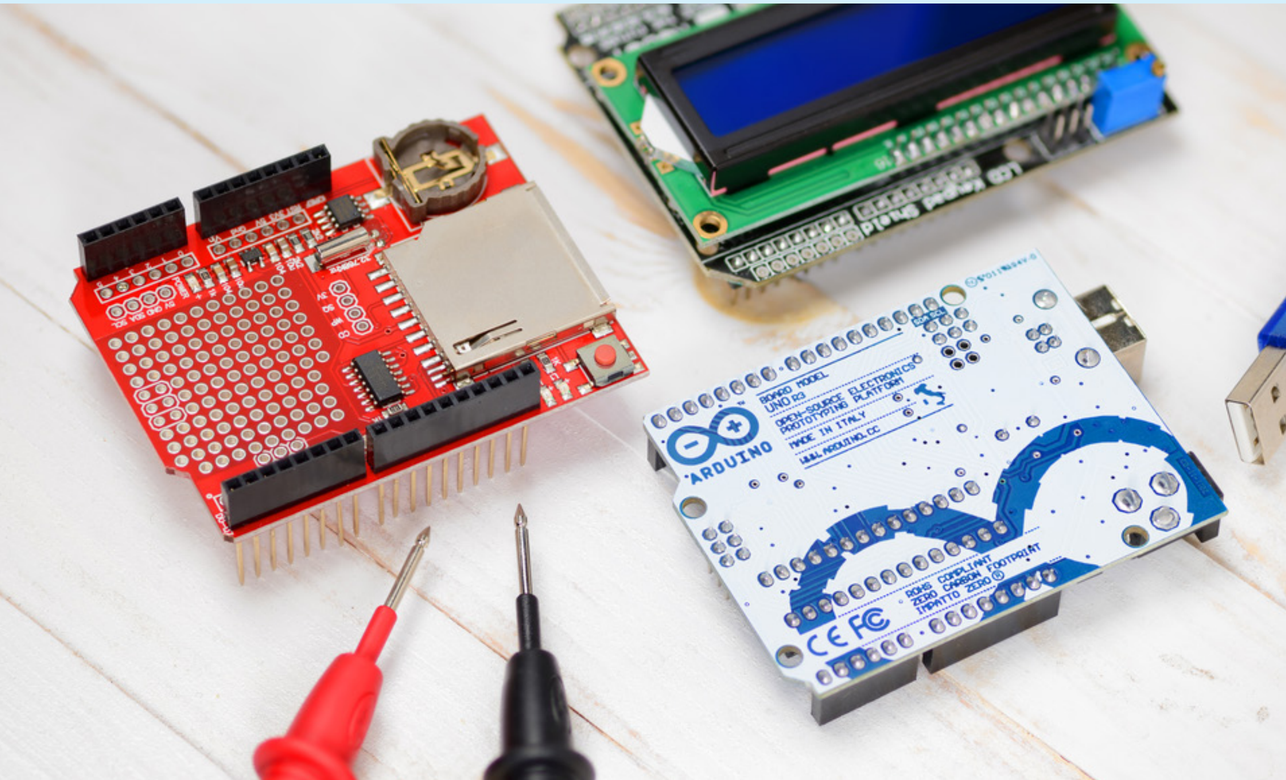
บทสรุปท้ายบท 322

CHAPTER

01

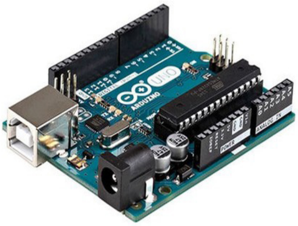
ภาพรวม Arduino สำหรับผู้เริ่มต้น

ในบทแรกนี้ ผมจะพาทุกท่านไปรู้จักกับโลกของ Arduino ในภาพกว้าง ให้รู้ว่ามันคืออะไร นำไปใช้ประโยชน์อย่างไร พาย้อนกลับไปหาต้นกำเนิดความเป็นมาของบอร์ด การเลือกใช้บอร์ด Arduino เพื่อการศึกษา การพัฒนาทักษะทางด้านอิเล็กทรอนิกส์ และการเขียนโปรแกรม องค์ประกอบของการพัฒนา Arduino Projects ตลอดจนแหล่งเรียนรู้เพิ่มเติมในรูปของคลิป VDO แสดงตัวอย่างโครงงานมากมายจากทั่วโลก



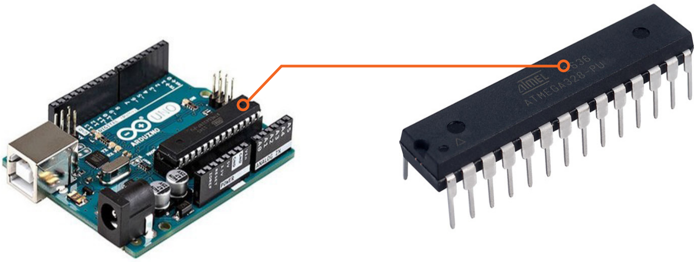
Arduino คืออะไร?

Arduino คือ บอร์ดไมโครคอนโทรลเลอร์ที่ใช้สร้างโปรเจ็กต์ที่เป็นสิ่งประดิษฐ์ที่เกี่ยวข้องในเชิงอิเล็กทรอนิกส์ (Electronics), วงจรไฟฟ้า (Circuit) และการเขียนโปรแกรม (Programming) โดยมีไมโครคอนโทรลเลอร์ในการควบคุมอุปกรณ์ทั้งหมด



▲ ภาพแสดง Arduino Logo และบอร์ดรุ่นมาตรฐาน Arduino UNO

ไมโครคอนโทรลเลอร์ (Microcontroller Unit : MCU) เป็นวงจรรวมขนาดเล็กที่ถูกออกแบบมาเพื่อควบคุมการทำงานเฉพาะในระบบฝังตัว หรือระบบสมองกลฝังตัว (Embedded System) โดยทั่วไปไมโครคอนโทรลเลอร์จะประกอบด้วยโพรเซสเซอร์ (Processor), หน่วยความจำ (Memory) และส่วนเชื่อมต่ออุปกรณ์ต่อพ่วง (I/O) ติดตั้งอยู่บนชิปตัวเดียวกัน (ชิปวงจรรวม หรือไอซี) จะเห็นว่าระบบไมโครคอนโทรลเลอร์นั้นคล้ายกับระบบคอมพิวเตอร์ ต่างกันเพียงเป็นระบบที่เล็กกว่า ขีดความสามารถน้อยกว่า เพราะมีหน้าที่ในการควบคุมการทำงานอยู่ในอุปกรณ์ หรือเครื่องใช้ไฟฟ้าที่มีการทำงานที่ซับซ้อนน้อยกว่านั่นเอง เราสามารถพบระบบฝังตัวได้ในอุปกรณ์ หรือเครื่องใช้ไฟฟ้าที่ต้องการระบบประมวลผลเพื่อควบคุมการทำงาน จึงอาจกล่าวได้ว่า **ไมโครคอนโทรลเลอร์ คือ ระบบคอนโทรลขนาดเล็กหรือระบบคอมพิวเตอร์ขนาดเล็ก**



Arduino Board　　　　Microcontroller

▲ รูปแสดงตำแหน่งของชิป Microcontroller บน Arduino Board

Arduino บอร์ดเพื่อการศึกษา

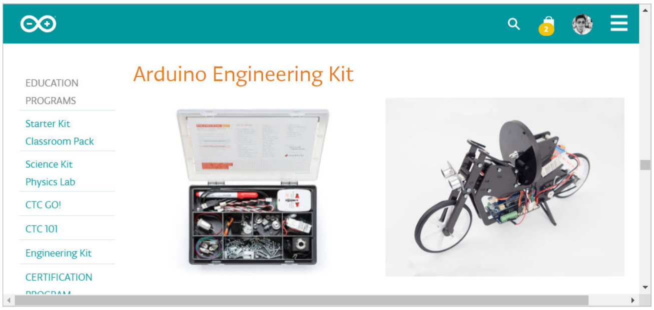
▲ Source : Intel IQ

เมื่อเข้าไปศึกษาที่เว็บไซต์หลักของ Arduino จะพบว่าที่นี่มีข้อมูลที่สนับสนุนในด้านการศึกษาเรียนรู้มากมาย อย่างเช่นหัวข้อ **"Teaching, Inspiring and Empowering!"** (https://www.arduino.cc/education) แสดงให้เห็นว่า ทาง Arduino ให้ความสำคัญกับการพัฒนาผลิตภัณฑ์เพื่อสนับสนุนภาคการศึกษาเป็นอย่างยิ่ง โดยมีการก่อตั้งหน่วยงานที่ชื่อ **"Arduino Education"** ขึ้นมา อันประกอบด้วยทีมงานที่จัดตั้งขึ้นโดยผู้เชี่ยวชาญทางด้านการศึกษา นักพัฒนาเนื้อหา วิศวกร และนักออกแบบการสื่อสารจากทั่วทุกมุมโลก โดยมีเป้าหมายมุ่งเน้นการพัฒนาระบบ STEAM รุ่นใหม่ๆ เพื่อตอบสนองความต้องการของนักเรียนและนักการศึกษา ตลอดหลักสูตรการศึกษาในระดับต่างๆ

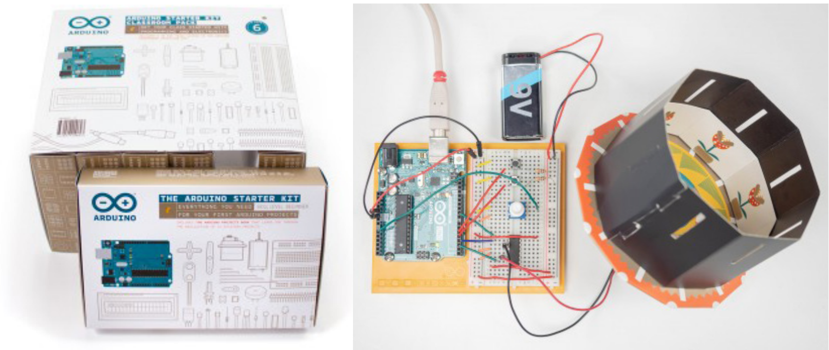
ภารกิจที่สำคัญ คือ การทำให้ทุกๆ คนสามารถเข้าถึงเทคโนโลยีได้ โดยพามันไปอยู่ในมือของนักเรียนและนักการศึกษาทุกๆ คน และเพื่อให้ภารกิจนี้เป็นจริง จึงได้ก่อตั้ง Arduino Education ขึ้นมาเพื่อปฏิบัติภารกิจนี้ให้บรรลุเป้าหมาย โดยเน้นการสร้างโปรแกรม STEAM ใหม่ๆ ที่รวมเอาวิทยาศาสตร์ เทคโนโลยี วิศวกรรม คณิตศาสตร์ และศิลปะ เพื่อรองรับความต้องการของผู้สอน และนักเรียนในภาคการศึกษาทุกระดับ

สถาบันการศึกษาหลายแห่งเลือกเข้าเว็บไซต์ Arduino.cc เพื่อเป็นแหล่งศึกษาข้อมูล ให้ความรู้แก่นักเรียนในหัวข้อต่างๆ ในหลากหลายวิธี

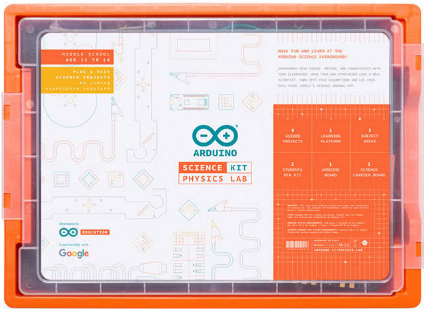
Develop	Arduino ได้รับการยอมรับอย่างกว้างขวางในแวดวงการศึกษาทั้งในระดับมัธยมศึกษาตอนปลาย ไปจนถึงมหาวิทยาลัย/วิทยาลัย/สถาบันวิจัย ในสาขา Engineer, Internet of Things, Robotics, Art และ Design
Teach and Learn	โรงเรียนมัธยมหลายแห่งบรรจุ Arduino ในหลักสูตรทางนวัตกรรมสำหรับการศึกษาข้ามหลักสูตร
Play	โรงเรียนประถมศึกษาใช้ของเล่นที่มีเทคโนโลยีของ Arduino ติดตั้งอยู่ภายในเพื่อแนะนำการเรียนรู้ทางกายภาพ ตรรกะ ทักษะการสร้าง และการแก้ปัญหา

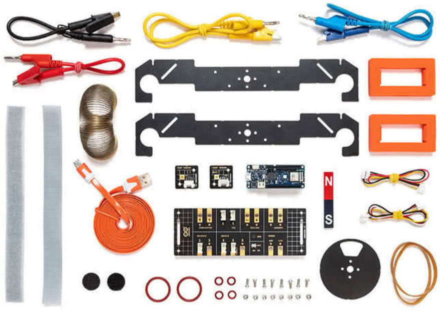


▲ รูปแสดงเมนู Arduino Kits สำหรับเลือกชุดอุปกรณ์ให้เหมาะกับความต้องการ



▲ รูปแสดง Starter Kit Classroom Pack แพ็คเกจที่เหมาะสำหรับชั้นเรียนที่สอนการเขียนโปรแกรมและอุปกรณ์อิเล็กทรอนิกส์





▲ ชุด Arduino Science Kit Physics Lab เหมาะสำหรับเด็กมัธยมปลาย (ช่วงอายุ 11-14)

ชมคลิปสาธิตโดยใช้คำค้นหาใน YouTube : "Arduino Science Kit Physics Lab"

www.youtube.com/watch?time_continue=33&v=4-U3JcdMoe0&feature=emb_title

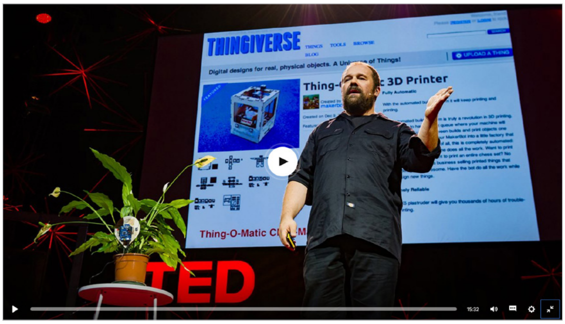
Arduino จากก้าวเล็กๆ สู่ความยิ่งใหญ่

เริ่มจากก้าวแรกที่ต้องการเพียงบอร์ดไมโครคอนโทรลเลอร์ ที่ทำให้เด็กได้นำไปใช้เรียนรู้งานทางด้านอิเล็กทรอนิกส์และการเขียนโปรแกรม ซึ่งก่อนที่จะมีบอร์ด Arduino เป็นเรื่องยากมากที่จะให้เด็กประถมมาเรียนเรื่องไมโครคอนโทรลเลอร์แล้วนำไปสร้างสิ่งประดิษฐ์ขึ้นมา เพราะบอร์ดไมโครคอนโทรลเลอร์ในยุคก่อนที่จะมี Arduino มีความซับซ้อน เรียนรู้ยาก ทีมผู้ก่อตั้ง คือ นาย Massimo Banzi และเพื่อนๆ จึงช่วยกันออกแบบบอร์ดไมโครคอนโทรลเลอร์ขนาดเล็กที่ชื่อว่า Arduino ที่ใช้งานง่าย เรียนรู้ได้อย่างรวดเร็วขึ้นมา และยังเผยแพร่ข้อมูลสู่สาธารณะอย่างหมดเปลือกแบบฟรีๆ

ลองจินตนาการกันต่อว่า ถ้าเราไม่มีบอร์ด Arduino เราก็ต้องไปซื้อชิป PIC หรือไม่ก็ต้องซื้อไมโครคอนโทรลเลอร์ตระกูลอื่นๆ มาใช้ ซึ่งมีราคาสูงและมีความยุ่งยากซับซ้อนในการเรียนรู้ ที่อาจจะเหมาะกับผู้ที่มีพื้นฐานทางอิเล็กทรอนิกส์ การต่อวงจร และการเขียนโค้ดเท่านั้น แต่ยากเกินไปสำหรับคนทั่วไป หรือแม้แต่เด็กชั้นประถมหรือมัธยมที่จะเข้าถึงได้ ซึ่ง Arduino ได้ทำลายข้อจำกัดเหล่านั้นลงได้ และพยายามจุดประกายให้กับทุกๆ คนได้ลุกขึ้นมาสร้างสิ่งประดิษฐ์ที่ยอดเยี่ยมเท่าที่พวกเขาจะจินตนาการได้ เกิดเป็นโปรเจ็กต์นับร้อยนับพัน ไม่ว่าจะเป็นของเด็กเล่นที่ไม่มีวางจำหน่ายที่ไหน เครื่องใช้ไฟฟ้าสำหรับสัตว์เลี้ยง สร้างเครื่องมือทางวิทยาศาสตร์ในราคาถูก อุปกรณ์เตือนภัยที่ใครๆ ก็ร่วมรายงานสถานการณ์ได้ ไปจนกระทั่งการสื่อสารกับดาวเทียมที่โคจรอยู่รอบโลกได้จริงๆ

> **คุณไม่จำเป็นต้องได้รับอนุญาต**
> **จากใครเพื่อสร้างสิ่งที่ยิ่งใหญ่**
>
> **คำพูดที่สร้างแรงบันดาลใจให้กับคนทั่วโลกของ Massimo Banzi**

รู้จัก Arduino ผ่านการบอกเล่าของ Massimo Banzi



▲ ชื่อคลิป “**How Arduino is open-sourcing imagination | Massimo Banzi**”

This talk was presented at an official TED conference https://www.ted.com

คำบรรยายจากคลิป

ไม่กี่อาทิตย์ที่ผ่านมา เพื่อนของผมให้รถของเล่นเป็นของขวัญแก่ลูกชายวัย 8 ปี แต่แทนที่จะไปซื้อจากห้างร้านทั่วๆ ไปอย่างที่ใครๆ ทำกัน เขาเลือกที่จะเข้าเว็บไซต์ THINGIVERSE แล้วดาวน์โหลดไฟล์มาไฟล์หนึ่งแล้วพรินต์ออกมาด้วย 3D Printer เครื่องนี้ (ตามรูป) ด้วยแนวคิดที่ว่า “คุณสามารถสร้างสิ่งต่างๆ ได้ด้วยเครื่องนี้ผ่านระบบดิจิทัล” ซึ่งเป็นแนวคิดที่นิตยสาร The Economist ยกย่องให้เป็นการปฏิวัติทางอุตสาหกรรมครั้งที่ 3 ผมต้องขอแย้งว่า จริงๆ แล้วครั้งที่ 4 กำลังเกิดขึ้นด้วยซ้ำ เกิดจากฮาร์ดแวร์โอเพ่นซอร์ส และการขับเคลื่อนของนักประดิษฐ์ เพราะพรินเตอร์ที่เพื่อนผมใช้ทำของเล่นนั้นก็มีที่มาจากโอเพ่นซอร์ส โดยคุณสามารถเข้าไปในเว็บไซต์ THINGIVERSE เพื่อดาวน์โหลดทุกสิ่งทุกอย่างที่ต้องใช้ ไม่ว่าจะเป็นไฟล์ วิธีการประกอบชิ้นส่วนอุปกรณ์ รวมถึงซอฟต์แวร์ วิธีการทั้งหมดอยู่ในนั้น นี่คือส่วนหนึ่งของชุมชนที่กว้างใหญ่ ที่มีคนนับพันจากทั่วโลกกำลังสร้างพรินเตอร์แบบนี้อยู่ และมีนวัตกรรมมากมายเกิดขึ้นอยู่เรื่อยๆ เพราะมันเป็นโอเพ่นซอร์ส คุณไม่จำเป็นต้องขออนุญาตใครเพื่อสร้างสิ่งที่ยิ่งใหญ่ มันเป็นเหตุการณ์คล้ายกับคอมพิวเตอร์พีซีที่เกิดขึ้นในปี ค.ศ. 1976 ที่บริษัท Apple และบริษัทอื่นๆ กำลังแข่งขันกันพัฒนา ซึ่งในอีกไม่กี่ปีนับจากนี้เราจะได้เห็นเหตุการณ์คล้ายๆ กัน

ศึกษา Arduino Projects ใน YouTube

เข้าไปที่เว็บไซต์ www.youtube.com แล้วพิมพ์คำว่า "**arduino project**" ในช่องค้นหา หรือใช้คีย์เวิร์ดเฉพาะโดยเพิ่มคำเพื่อค้นหาโปรเจ็กต์ตามความสนใจ เช่น arduino project robot, arduino spider robot, arduino robot arm, arduino stepper motor, arduino iot project เป็นต้น

Top 10 Arduino Projects 2020

คลิป **Top 10 Arduino Projects 2020** | Mind Blowing Arduino School Projects เมื่อเข้าไปที่คลิปหลักจะพบลิงค์เชื่อมไปยังโปรเจ็กต์ต่างๆ ที่ผู้อ่านจะพบรายละเอียดและข้อมูลทางด้านเทคนิคอย่างครบถ้วน เช่น Required Hardware/Parts Required, Schematic, Circuit File, Arduino Code

▲ คำค้นหาใน YouTube :

"Top 10 Arduino Projects 2020"

01 Bluetooth Nurf Turret
http://bitly.ws/or8P

02 Gesture Controlled Robot
http://bitly.ws/or8T

03 Buzz Wire with Score Counter
http://bitly.ws/or8V

04 Otto DIY Robot
http://bitly.ws/or94

05 Mind Controlled Drone
http://bitly.ws/or97

06 Sea Shells Light Music Box
http://bitly.ws/or9a

07 3D Printed Arduino Lawn Mower
http://bitly.ws/or9d

08 Web-Based Two-Player Game
http://bitly.ws/or9h

09 Control Robot Arm via Web
http://bitly.ws/or9o

10 Lightweight Arduino GSM Mobile Phone
http://bitly.ws/or9r

องค์ประกอบการพัฒนา Arduino Projects

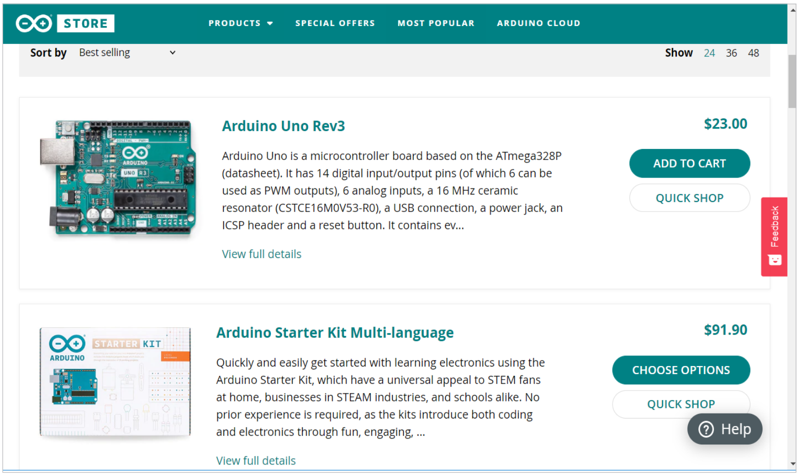
The word "Arduino" can mean 3 things

A Physical Piece of Hardware | A Programming Environment | A Community & Philosophy

▲ รูปแสดงองค์ประกอบการพัฒนา Arduino Projects

องค์ประกอบของการพัฒนา Arduino จะประกอบด้วย 3 ส่วนที่นำมาประยุกต์ใช้ทำงานร่วมกันได้แก่

1. **A Physical Piece of Hardware** คือ อุปกรณ์บอร์ด Arduino ที่ใช้ในการประมวลผลและควบคุมอุปกรณ์ Sensors ต่างๆ ด้วยไมโครคอนโทรลเลอร์ https://store.arduino.cc



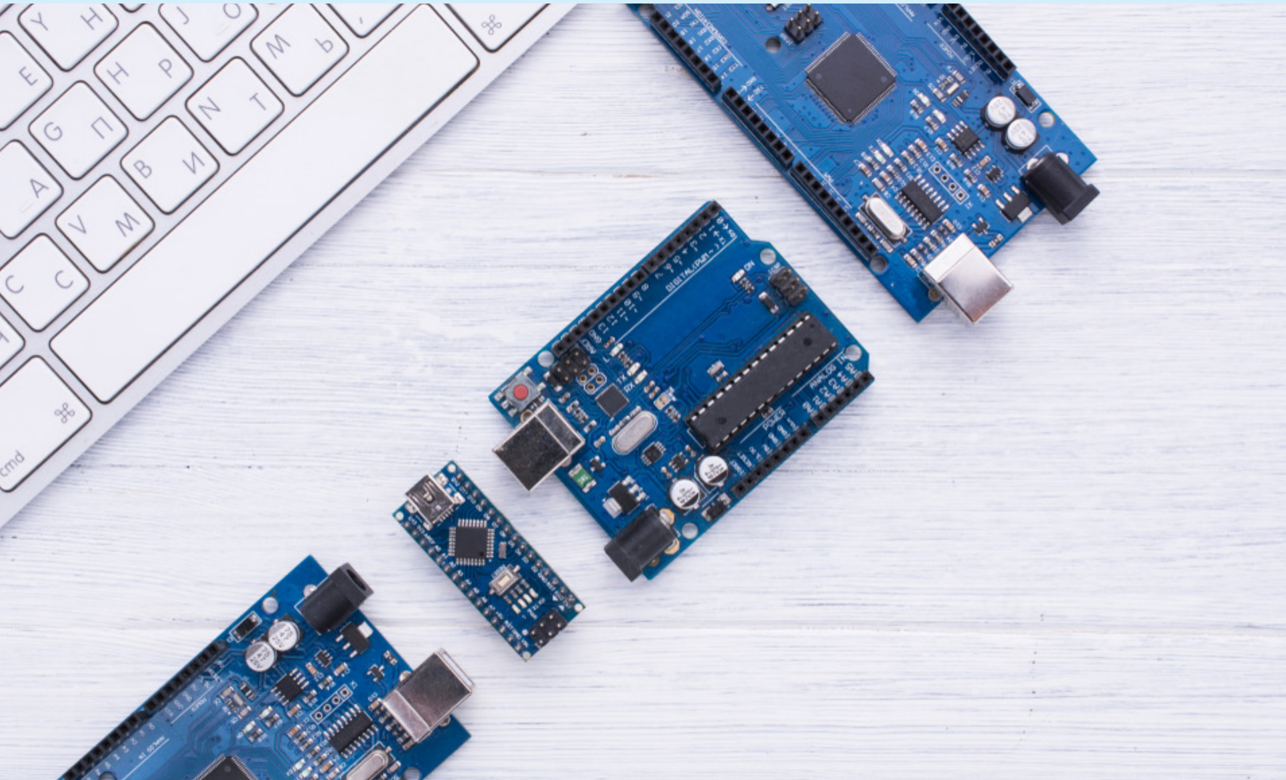
▲ ภาพแสดงตัวอย่างบางส่วนของ Arduino Hardware

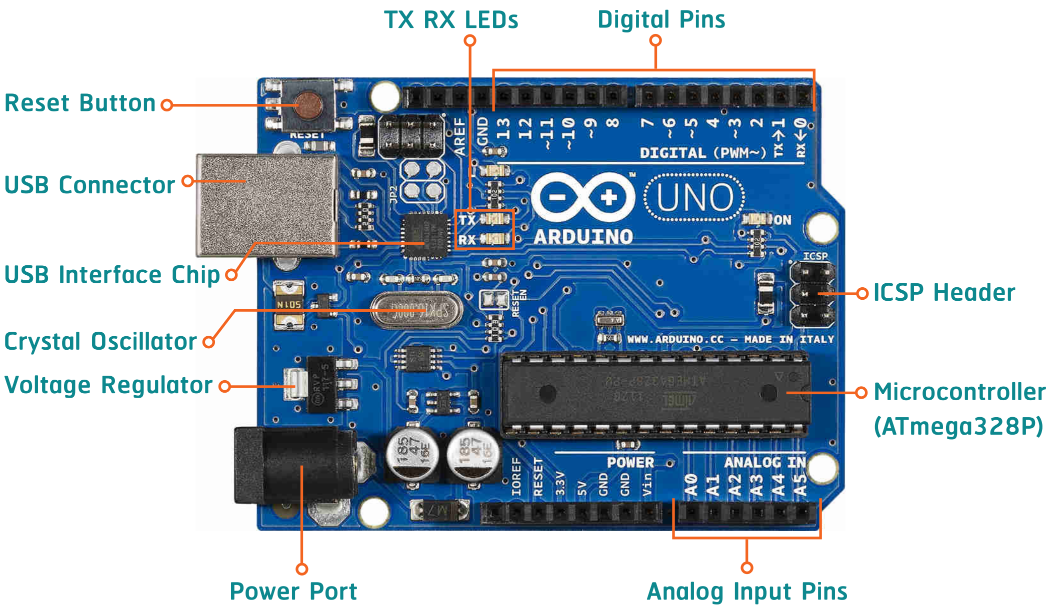
CHAPTER

02

รู้จักบอร์ดมาตรฐานของ Arduino และอุปกรณ์เสริม

เราได้รู้จักกับ Arduino ในภาพรวมกันไปแล้วในบทแรก สำหรับเนื้อหาในบทที่ 2 นี้ เราจะลงลึกในรายละเอียดของอุปกรณ์มากขึ้น เพื่อดูว่าบอร์ดไมโครคอนโทรลเลอร์มีส่วนประกอบอะไรบ้าง ส่วนไหนมีหน้าที่ทำอะไร โดยจะเริ่มที่ Arduino UNO ที่ถือเป็นบอร์ดรุ่นมาตรฐานที่เหมาะสำหรับผู้เริ่มต้นมากที่สุด หลังจากนั้นจะแนะนำบอร์ดตระกูลอื่นๆ ที่ได้รับความนิยม (Most Popular) พร้อมแนะนำอุปกรณ์เสริมอื่นๆ และตัวอย่างการประยุกต์ใช้งานบอร์ด Arduino กับ Sensors



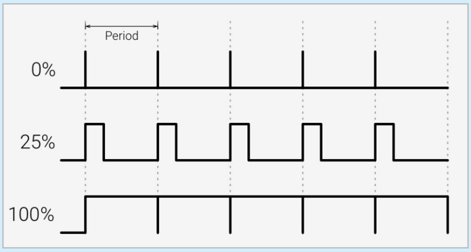


▲ รูปแสดงส่วนประกอบของบอร์ด Arduino UNO Rev 3

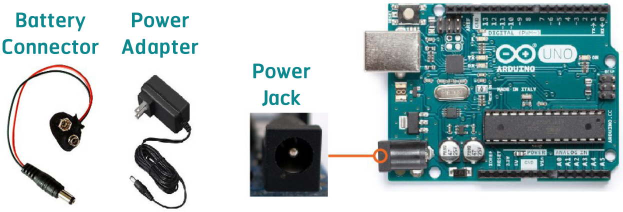
PWM คืออะไร?

PWM ย่อมาจากคำว่า **"Pulse Width Modulation"** คือ เทคนิคการเปลี่ยนช่วงความถี่ให้เหมาะสมกับการส่งสัญญาณ หรือที่เรียกว่า **"Modulation"** ถูกใช้เพื่อเข้ารหัสข้อความแปลงเป็นสัญญาณพัลส์ (Pulse) ซึ่งเป็นสัญญาณไฟฟ้าที่มีลักษณะเป็นคลื่นรูปสี่เหลี่ยม มีเพียงสถานะ High หรือ Low ต่อเนื่องซ้ำกันไปเรื่อยๆ ช่วงเวลาที่สัญญาณมีสถานะเป็น High ในหนึ่งรอบ (Period) นั้นจะถูกเรียกว่า "ความกว้างพัลส์" (Pulse Width) ซึ่งนิยมแสดงเป็นเปอร์เซ็นต์ ดังรูป

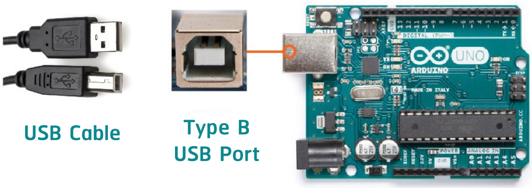
PWM เป็นเทคนิคที่ Arduino ใช้ในการควบคุมวงจร เช่นใช้ควบคุมความเร็วของมอเตอร์ไฟฟ้ากระแสตรง (DC Motor), ควบคุมการหรี่แสงของหลอดแอลอีดี (Dimming LED) เป็นต้น เมื่อขา (Pin) บนบอร์ดถูก Enable เพื่อเปิดใช้งาน PWM มันจะสร้างความถี่คงที่ ~ 500 Hz ในขณะที่รอบการทำงานจะเปลี่ยนไปตามพารามิเตอร์ที่ผู้ใช้ตั้งค่า



▲ รูปแสดงเปอร์เซ็นต์ของสัญญาณ Pulse



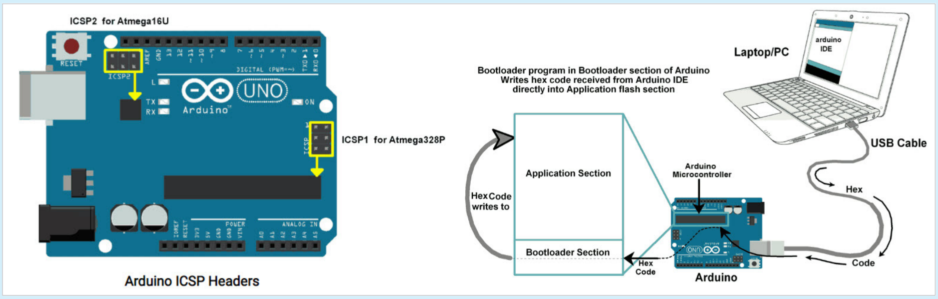
▲ รูปแสดง Power Jack ที่ใช้ต่อกับ Power Adapter หรือ Battery Connector



▲ รูปแสดงพอร์ต USB Type B สำหรับเชื่อมต่อกับคอมพิวเตอร์กับบอร์ด Arduino UNO

วิธีพัฒนาโค้ดเริ่มต้นการทำงานของบอร์ด (Flash Arduino Bootloader)

ในการปรับปรุงโค้ดเริ่มต้นการทำงานของบอร์ด (Flash Bootloader) ใหม่ เราต้องเขียนหรือติดตั้ง Bootloader ลงใน Bootloader Section ในพื้นที่หน่วยความจำ โดยปกติแล้วเราจะเขียนมันลงในชิป IC ก่อนจะบัดกรีลงบนแผ่น PCB ขณะที่ผู้ผลิตชิปไมโครคอนโทรลเลอร์บางราย เช่น Atmel หรือ Microchip ได้เตรียม ICSP Header (In-Circuit Serial Programming) ไว้ให้บนบอร์ดเพื่อใช้สำหรับแฟลชเมมโมรี สำหรับบอร์ด Arduino UNO จะมี ICSP Headers ไว้ให้ 2 หัว หัวหนึ่งสำหรับแฟลชชิป ATmega16U2 ที่ตำแหน่ง ICSP2 ส่วนอีกหัวสำหรับแฟลชชิป ATmega328 ที่ตำแหน่ง ICSP1 ดังรูป



▲ รูปแสดงตำแหน่ง ICSP Header บนบอร์ด Arduino UNO และวิธีแฟลช

www.electronicwings.com/arduino/basics-to-developing-bootloader-for-arduino

บอร์ด Arduino LEONARDO (with Headers)

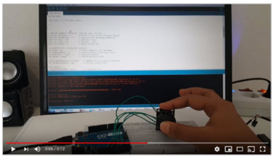
LEONARDO แตกต่างจากบอร์ดอื่นๆ คือ ใช้ชิปไมโครคอนโทรลเลอร์รุ่น ATmega32u4 ซึ่งมีการบรรจุ USB Communication เอาไว้พร้อมในตัว จึงไม่จำเป็นต้องใช้ตัวประมวลผลสำรองจากภายนอก ทำให้ LEONARDO มีความสามารถในการจำลองตัวเองเป็นเมาส์ คีย์บอร์ด จอยสติกได้ นอกจากนี้ยังมีพอร์ต Virtual (CDC) Serial Port/COM อีกด้วย

▲ **รูปแสดงบอร์ด Arduino LEONARDO with Headers ในมุมมองต่างๆ**

หากต้องการทำความเข้าใจเพิ่มเติม ให้เปิดเว็บไซต์ YouTube แล้วค้นหาคลิปดังต่อไปนี้

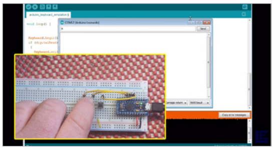
คลิป : "DIY Arduino Mouse Joystick"

https://www.youtube.com/watch?v=t8mE1ayw5bo



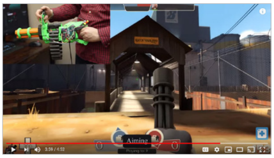
คลิป : "Joystick as mouse using Arduino Leonardo"

https://www.youtube.com/watch?v=u0DYwo009Yo



คลิป : "Arduino Keyboard Emulator"

https://www.youtube.com/watch?v=SHIcliL4O14&t=3s



คลิป : "Which Arduino to Buy for Making Video Game Controllers?"

https://www.youtube.com/watch?v=kqdTL_ICt8E

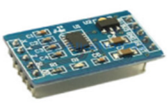
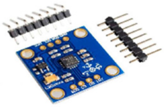
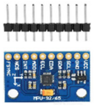
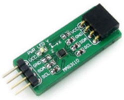
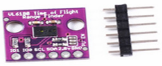
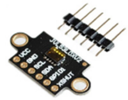
Arduino LEONARDO เป็นบอร์ดที่ใช้ชิปไอซีไมโครคอนโทรลเลอร์ (Microcontroller IC Chip) รุ่น ATMEGA32U4 (Datasheet), บอร์ดประกอบด้วยขารับและขาส่งสัญญาณดิจิทัล 20 ขา (7 ขาสำหรับเอาต์พุต PWM และขารับสัญญาณอะนาล็อก 12 ขา), คริสตัลผลิตสัญญาณ 16 MHz, หัวต่อแบบ USB Jack Type Micro-B, เพาเวอร์แจ็ค, ICSP Header และปุ่ม Reset เรียกว่ามีทุกสิ่งทุกอย่างที่จำเป็นเพื่อใช้ซัพพอร์ตการทำงานของไมโครคอนโทรลเลอร์ เมื่อต่อบอร์ดกับคอมพิวเตอร์ด้วยสาย USB ก็พร้อมทำงานได้ทันที (จะต่อกับอะแด็ปเตอร์หรือใช้แบตเตอรี่แทนก็ได้)

ตารางแสดง TECH SPECS	
Microcontroller (MCU)	ATmega32u4
Operating Voltage	5 V
Input Voltage (Recommended)	7-12 V
Input Voltage (Limits)	6-20 V
Digital I/O Pins	20
PWM Channels	7
Analog Input Channels	12
DC Current per I/O Pin	40 mA
DC Current for 3.3 V Pin	50 mA
Flash Memory	32 KB (ATmega32u4) of which 4 KB used by bootloader
SRAM	2.5 KB (ATmega32u4)
EEPROM	1 KB (ATmega32u4)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.3 mm
Weight	20 g

ข้อมูล Sensors/Modules, Motors และ Shields

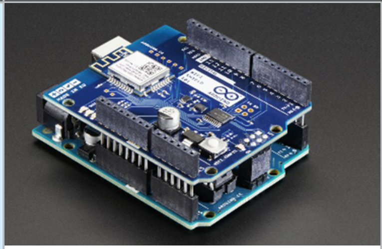
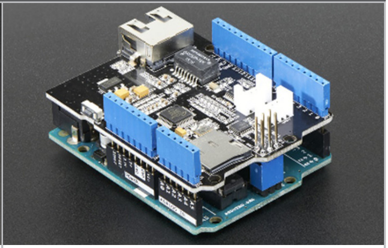
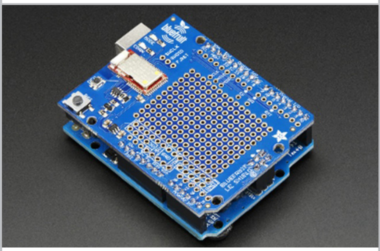
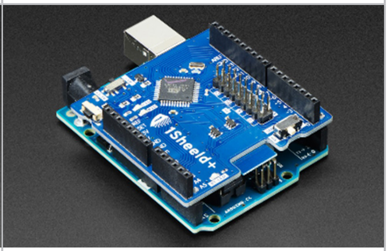
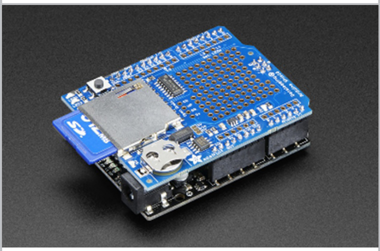
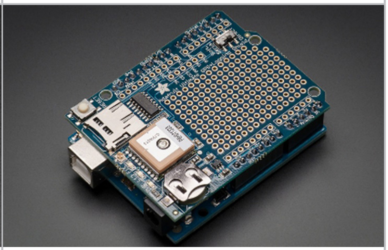
ประเภทของ Sensors/Modules

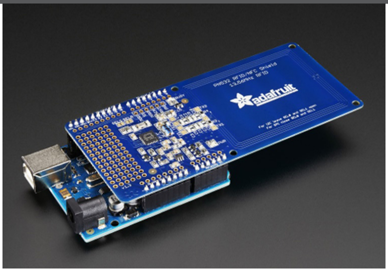
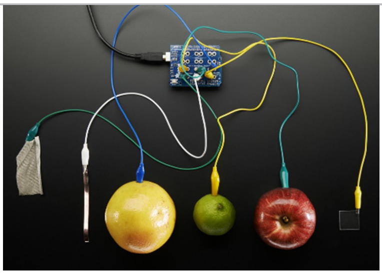
Arduino Sensors/Modules เป็นอุปกรณ์ต่อพ่วงเพื่อรับคำสั่งจากบอร์ด Arduino ให้ทำงานตามต้องการ โดย Sensors/Modules ที่ใช้ได้กับ Arduino แบ่งตามลักษณะการใช้งานดังนี้

Sensors/Modules	Examples			
เซนเซอร์ความเร่ง/ไจโร/IMU (Accelerometer, Gyroscope & IMU Sensors)	 Accelerometer (MMA7361)	 Gyroscope (GY-61 ADXL335)	 IMU (GY-9255 IMU 9DOF)	 Digital Compass Magnetometer (MAG3110)
เซนเซอร์วัดระยะทาง (Distance/Range Sensors)	Ultrasonic Sensor (HY-SRF05)	Infrared Distance Sensor (GP2Y0A710K0F)	 Ambient Light Sensor (VL6180)	 Laser Ranging Distance Sensor (VL53L0X V2)
เซนเซอร์แสงและการมองเห็น (Optical & Vision Sensors)	Camera Module (OV7670)	Infrared Barrier Module (OV7670)	LDR Photoresistor Sensor Module (OV7670)	Laser Module (OV7670)
เซนเซอร์ตรวจจับการเคลื่อนไหว (Motion Sensors)	Vibration Sensor (801S)	Microwave Motion Sensor (HB100)	PIR Motion Sensor (HC-SR501)	Knock Sensor (KY-031)

ประเภทของ Arduino Shields

Arduino Shields เป็นอุปกรณ์เสริมใช้สำหรับต่อพ่วงกับบอร์ด Arduino โดยการสวมเข้ากับ Digital Pin และ Analog Pin ซึ่งชิลด์เหล่านี้ส่วนใหญ่มาพร้อมกับ Library ที่ช่วยให้สามารถใช้งานชิลด์ได้ง่าย ไม่ต้องยุ่งยากกับการเขียนคำสั่งควบคุมเอง แบ่งตามลักษณะการใช้งานดังนี้

Arduino Shields	Examples	
WIFI/ETHERNET	 Arduino WiFi Shield 101	 Ethernet Shield for Arduino (W5500)
BLUETOOTH	 Adafruit Bluefruit LE Shield - Bluetooth LE for Arduino	 1Sheeld+ for iOS and Android
DATA LOGGING	 Adafruit Assembled Data Logging Shield	 Adafruit Ultimate GPS Logger Shield

Arduino Shields	Examples
NFC/RFID	 Adafruit PN532 NFC/RFID Controller Shield for Arduino + Extras
TOUCH	 Adafruit 12 x Capacitive Touch Shield for Arduino - MPR121

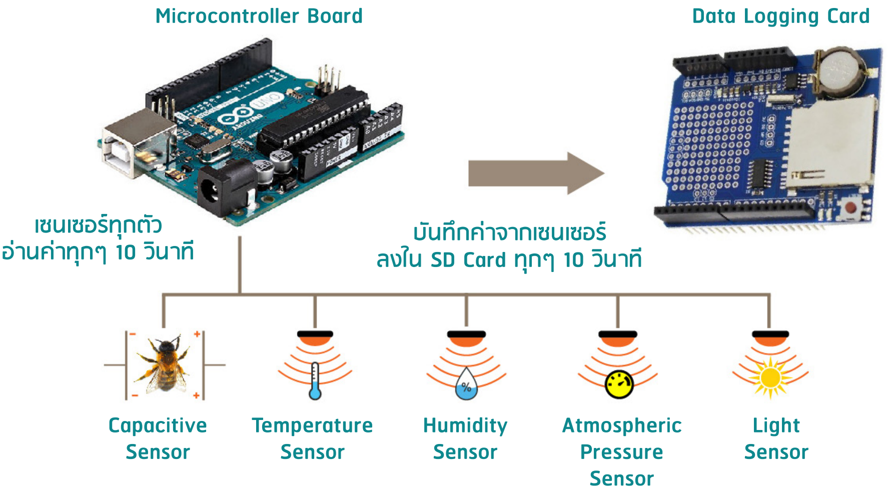
Credit : www.adafruit.com

ตัวอย่างการใช้งานบอร์ด Arduino กับ Sensors

ในการใช้งานบอร์ด Arduino โดยส่วนใหญ่จะใช้งานร่วมกับอุปกรณ์ Sensors แบบต่างๆ เพื่อควบคุมและสั่งการให้ทำงานตามที่ต้องการ ขออธิบายผ่านโปรเจ็กต์ที่ชื่อว่า “Monitoring Solitary Bees Using Open Technology” ซึ่งเป็นโครงการทางวิทยาศาสตร์ในการมอนิเตอร์พฤติกรรมการทำรังของผึ้งเมสันในสวนหลังบ้าน เป็นโปรเจ็กต์ของนาย Mike Teachman มือสมัครเล่นที่ชื่นชอบผึ้ง กับนาย Paul Perrault วิศวกรอาวุโสทางแอปพลิเคชัน

ผึ้งเมสัน (Mason Bee) จะมีพฤติกรรมที่แตกต่างจากผึ้งน้ำหวาน หรือฮันนี่บี (Honey Bee) โดยเมสันจะเป็นผึ้งที่ชอบอยู่ตามลำพัง มันไม่ใช่สัตว์สังคม ผึ้งตัวเมียจะมีความสมบูรณ์ ไม่มีผึ้งงาน พวกมันใช้ส่วนท้องขนเกสร พวกมันชอบทำรังในรูซึ่งมีส่วนสำคัญต่อการผสมเกสรของไม้ผลนานาชนิด

เบื้องหลังไอเดียของโปรเจ็กต์นี้คือ การเฝ้าติดตามดูพฤติกรรมของผึ้งเมสันที่แต่ละตัวจะอาศัยอยู่ตามลำพังแบบรังละตัว โดยใช้เซนเซอร์ติดไว้กับรังผึ้งทุกรัง ไม่ได้ติดเซนเซอร์เข้ากับตัวผึ้งแต่อย่างใด โดยเซนเซอร์ที่ใช้จะเป็นแบบ Capacitive Sensor ที่ใช้ตรวจจับความเคลื่อนไหวของผึ้งเมสันที่บินเข้าออกจากรัง และใช้บอร์ดไมโครคอนโทรลเลอร์ Arduino UNO เป็นระบบควบคุมการทำงาน ต่อมาได้มีการขยายขอบเขตการศึกษาเพิ่มเติม จากในช่วงแรกที่เราตรวจจับแค่การบินเข้าบินออกจากรังเท่านั้น แต่ได้เพิ่มการวัดกิจกรรมของผึ้งที่แสดงความรู้สึกออกมาได้อีกด้วย แต่จะต้องใช้เซนเซอร์ชนิดอื่นๆ เพิ่มเติม พร้อมกับการพัฒนาระบบการวัดที่จะใช้ในการรวบรวมข้อมูลจำนวนมาก



▲ รูปแสดงแผนภาพการทำงานของเครื่องมอนิเตอร์พฤติกรรมของผึ้งเมสัน

จากรูป เราสามารถแบ่งการทำงานออกเป็น 3 ส่วน ดังนี้

1. **ส่วน Input (Sensors)** ใช้สำหรับรับข้อมูลที่ตรวจวัดได้จากเซนเซอร์ ซึ่งมี 5 ชุดข้อมูล ได้แก่
   - **Capacitive Sensor** ใช้ตรวจจับวัตถุ ในที่นี้คือ ผึ้ง
   - **Temperature Sensor** ใช้ตรวจจับอุณหภูมิ
   - **Humidity Sensor** ใช้ตรวจจับความชื้น
   - **Atmospheric Pressure Sensor** ใช้วัดความดันอากาศ
   - **Light Sensor** ใช้วัดความเข้มแสง

CHAPTER

03

การเลือกซื้อ Arduino ให้เหมาะกับการใช้งาน

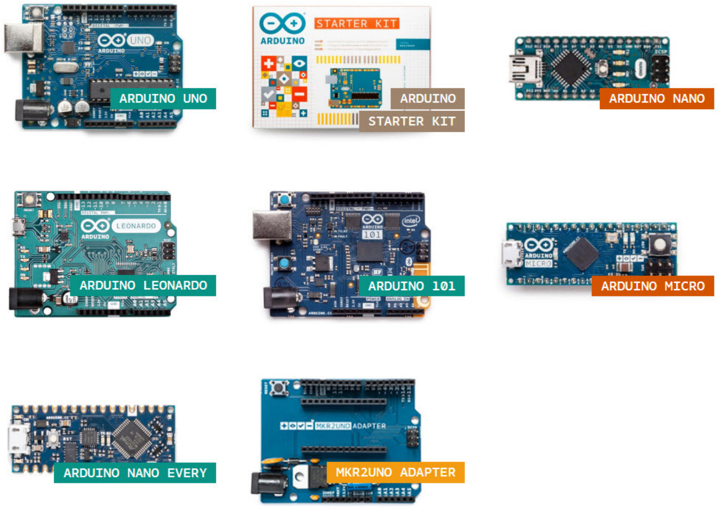
จากอดีตจนถึงปัจจุบัน ผลิตภัณฑ์ Arduino นั้นถูกผลิตออกมาเป็นจำนวนมาก ฉะนั้นในการเลือกซื้อ Arduino มาใช้งาน จำเป็นจะต้องศึกษา Arduino Products ชุดต่างๆ ให้ดีเสียก่อนว่า ผลิตภัณฑ์มีกี่กลุ่ม แต่ละกลุ่มเหมาะกับใคร เรียนรู้ว่ารุ่นไหนเหมาะจะใช้กับงานลักษณะไหน ไปดูตารางเปรียบเทียบสเปคเพื่อให้เห็นความแตกต่างชัดเจนมากขึ้น การตรวจสอบบอร์ดแท้ทำอย่างไร ทั้งหมดที่กล่าวมาก็เพื่อให้ผู้อ่านมีความรอบรู้อย่างเพียงพอ สามารถพิจารณาเลือกซื้อบอร์ดได้ด้วยตนเอง

สินค้ากลุ่มไหนเหมาะกับใคร (Arduino Products)

หากต้องการดูผลิตภัณฑ์ทั้งหมดของ Arduino ที่ผลิตจำหน่ายอย่างเป็นทางการ ได้แก่ Boards, Modules, Motors & Servos, Shields, Kits และ Accessories อื่นๆ สามารถเข้าไปสำรวจดูสินค้าต่างๆ ที่กล่าวมาได้ที่เว็บไซต์ www.arduino.cc เลือกเมนู **STORE** ซึ่งเป็นหน้าร้านแสดงรายการสินค้าทั้งหมด

หรือหากต้องการศึกษารายละเอียด เช่น การแบ่งหมวดหมู่สินค้า การเปรียบเทียบสเปคของสินค้าแต่ละรุ่น หรือหากมีข้อสงสัยว่า บอร์ด Arduino ของคุณเป็นของแท้หรือเทียม ก็สามารถเรียนรู้วิธีสังเกตบอร์ดเทียมได้ในบทนี้ ซึ่งผู้เขียนได้สรุปเนื้อหามาจากเว็บไซต์ www.arduino.cc

1. **Entry Level** เป็นกลุ่มสินค้าที่ออกแบบไว้สำหรับผู้เริ่มต้น เน้นใช้งานง่ายและเหมาะจะใช้เพื่อสร้างโปรเจ็กต์ชิ้นแรกสำหรับมือใหม่ที่กำลังก้าวสู่เมกเกอร์ (Maker) สินค้าในกลุ่มนี้ประกอบด้วยบอร์ดและโมดูลที่เหมาะในการเริ่มต้นเรียนรู้เกี่ยวกับบอร์ดอิเล็กทรอนิกส์ และการเขียนโค้ดมากที่สุด ในชุดการเรียนรู้ Arduino Starter Kit ก็จะแถมคู่มือที่ประกอบด้วย 15 บทเรียนที่จะสอนตั้งแต่เบสิกพื้นฐานไปจนถึงการทำโปรเจ็กต์ยากๆ กันเลยทีเดียว



▲ **กลุ่ม Entry Level บอร์ดมาตรฐานรุ่นต่างๆ เหมาะสำหรับผู้เริ่มต้น**

www.arduino.cc/en/Main/Products

CHAPTER

04

ซอฟต์แวร์ เพื่อการเรียนรู้ Arduino

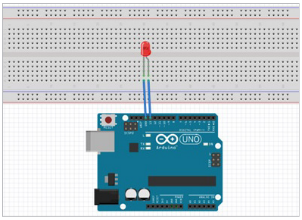
Circuit Drawing and Simulation

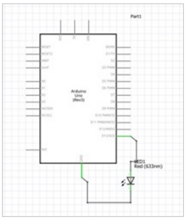
ในบทที่ 1 ถึง 3 เราได้ทำความรู้จักกับบอร์ด Arduino และอุปกรณ์เสริมต่างๆ ซึ่งเป็นเรื่องทางฮาร์ดแวร์ทั้งสิ้น สำหรับบทที่ 4 จะอธิบายในส่วนของซอฟต์แวร์และการเขียนโปรแกรม จากบทแรกเราได้เกริ่นนำซอฟต์แวร์สำหรับเขียนโค้ดกันไปบ้างแล้ว ซึ่งจะได้สรุปหลักการเขียนด้วยภาษา C กันในบทที่ 6 สำหรับบทที่ 4 จะแนะนำซอฟต์แวร์อีก 2 ส่วน ได้แก่ ซอฟต์แวร์สำหรับการเขียนวงจร (Circuit Drawing) และซอฟต์แวร์จำลองการทำงานแทนบอร์ดจริง (Arduino Simulator) โดยจะเน้นแสดงตัวอย่างเบื้องต้นแค่พอเข้าใจเท่านั้น เพื่อเป็นแนวทางให้ผู้อ่านนำไปศึกษาต่อด้วยตนเอง

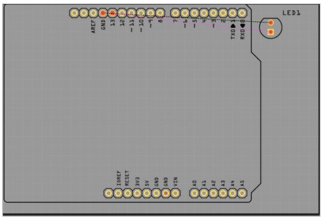
Fritzing (Circuit Drawing)

“เหตุผลที่แนะนำซอฟต์แวร์ตัวนี้เป็นตัวแรกก็เพราะว่า เว็บไซต์ Arduino แนะนำให้ใช้”

Fritzing คือ โปรแกรมที่ใช้จำลองการต่อบอร์ด Arduino เข้ากับอุปกรณ์อิเล็กทรอนิกส์โดยใช้ภาพกราฟิกแทนของจริง สามารถใช้เขียนแบบวงจรไฟฟ้า และออกแบบลายวงจรบนแผ่น PCB ได้อีกด้วย จึงสามารถแสดงวงจรในรูปแบบ Breadboard, Schematic และ PCB ได้ นอกจากนี้ยังสามารถใช้ออกแบบ แก้ไข และสร้างชิ้นส่วน (Parts) ของอุปกรณ์อิเล็กทรอนิกส์ขึ้นได้เองอีกด้วย


Breadboard


Schematic


PCB

▲ รูปแสดงวงจรในรูปแบบต่างๆ ที่สร้างจากโปรแกรม Fritzing

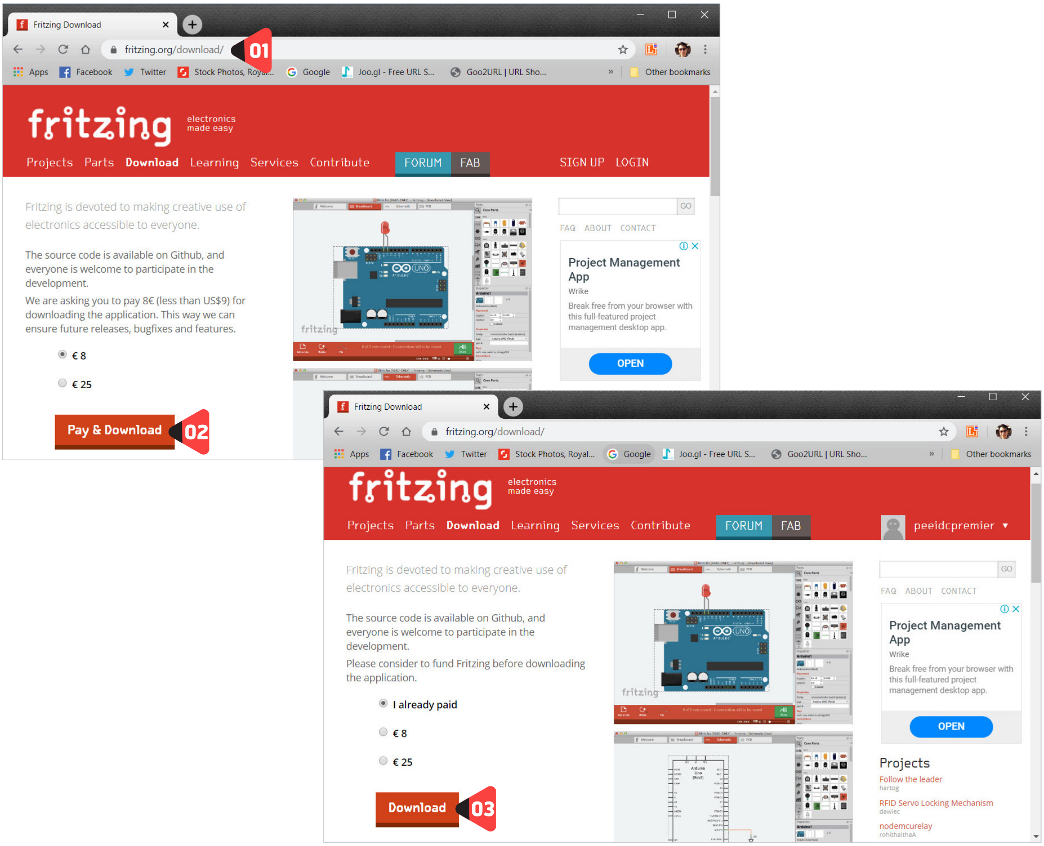
จุดเด่นของโปรแกรม Fritzing

1. มีบอร์ด Arduino และอุปกรณ์ที่เกี่ยวข้องมากมาย รองรับการเขียนวงจรหลากหลายรูปแบบ
2. สามารถใช้โปรแกรมนี้เพื่อเขียนโค้ดและอัพโหลดลงบอร์ด Arduino ได้เช่นเดียวกับ Arduino IDE เลยทีเดียว
3. ได้รับการพัฒนาเครื่องมือให้ยูสเซอร์ใช้ทำเอกสาร Arduino และเอกสารทางอิเล็กทรอนิกส์ทั้งหลาย
4. สามารถแชร์เอกสารกับคนอื่นๆ ได้ง่าย
5. รองรับการสอนวิชาอิเล็กทรอนิกส์ในห้องเรียน
6. ใช้สร้าง PCB Layout เพื่อใช้ในการผลิตด้วยเครื่องจักร
7. สนับสนุนฮาร์ด์แวร์โอเพ่นซอร์ส หมายถึง การเปิดเผยข้อมูลเบื้องหลังการสร้างและพัฒนา

8. โปรแกรมได้รับการออกแบบมาสำหรับคนในหลากหลายอาชีพ ไม่เว้นแม้กระทั่งมือสมัครเล่น
9. มีชุมชนออนไลน์เพื่อใช้เป็นแหล่งแลกเปลี่ยนความรู้และแชร์ประสบการณ์ (บอร์ดฟอรัม : https://forum.fritzing.org/)
10. มีบทเรียนสอนการใช้งานโปรแกรมบนเว็บไซต์ (https://fritzing.org/learning/)

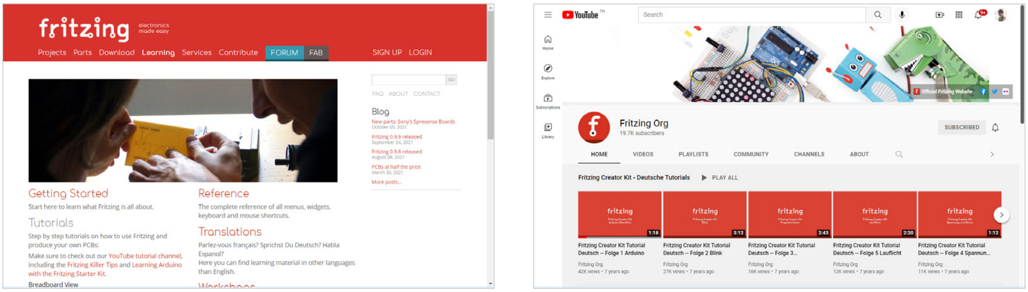
ดาวน์โหลดและติดตั้ง Fritzing

1. ให้เข้าไปที่เว็บไซต์ http://fritzing.org และคลิกเมนู **Download**
2. สมัครสมาชิก **Sign Up** และ **Login** เข้าใช้งาน
3. คลิกเลือกออปชัน **I already paid** และคลิกปุ่ม **Download**



แหล่งศึกษาการใช้งานโปรแกรม Fritzing เพิ่มเติมทางออนไลน์

เนื่องจากหนังสือเล่มนี้เน้นไปที่กระบวนการการเรียนรู้ และสร้างทักษะผ่านการทำ Lab และ Project ไม่ได้เน้นสอนพื้นฐานการใช้งานโปรแกรม หรือการเขียนโปรแกรม ผู้อ่านจึงจำเป็นต้องศึกษาด้วยตนเอง โดยสามารถเข้าไปเรียนรู้ได้ที่เว็บไซต์ของ Fritzing ที่เมนูหลัก Learning ในหัวข้อ Getting Started ซึ่งจะมีบทเรียนให้ศึกษาพื้นฐานครบทุกเรื่อง อีกทั้งยังมี YouTube Tutorial Channel ให้ดูในรูปแบบวิดีโออีกด้วย



▲ รูปแสดงหน้าเว็บเพจ Learning และ YouTube Channel ของเว็บไซต์ Fritzing สำหรับศึกษาการใช้งาน

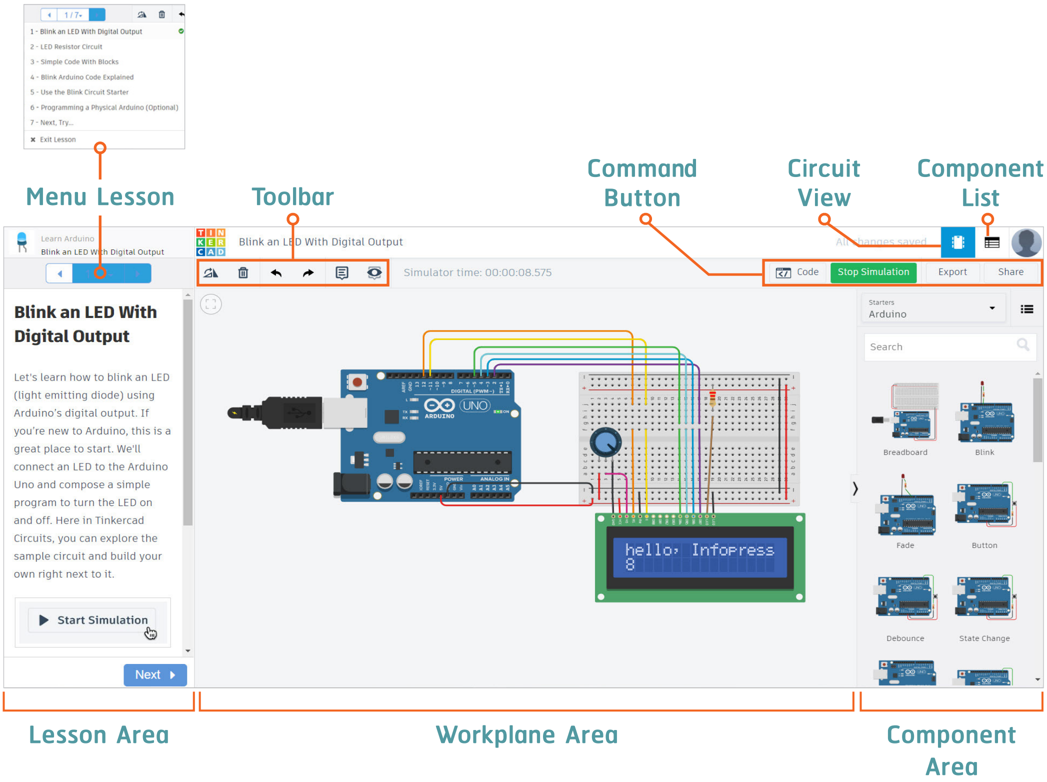
Tinkercad (Arduino Simulator)

Tinkercad คือ คอลเล็คชันของซอฟต์แวร์ในแบบฟรีออนไลน์ เป็นเครื่องมือที่จะช่วยในการฝึกคิดก่อนที่จะสร้างมันขึ้นมาจริงๆ แต่ที่เราจะเน้นกันในบทนี้คงมุ่งไปที่เครื่องมือที่ช่วยในการพัฒนา Program, Simulate, Assembly ในทาง Electronics ซึ่งจะช่วยในการจำลองการต่อวงจรเพื่อทดสอบก่อนต่อวงจรเพื่อใช้งานจริง ช่วยในการออกแบบและลดความเสียหาย หากวงจรที่เราออกแบบนั้นผิดพลาด นอกจากนี้ยังช่วยในการพัฒนาโปรแกรมทั้งในรูปของ Code (ภาษาโปรแกรมแบบข้อความ) หรือ CodeBlocks (ภาษาโปรแกรมแบบกราฟิกที่ใช้การคลิก-ลาก-วาง)

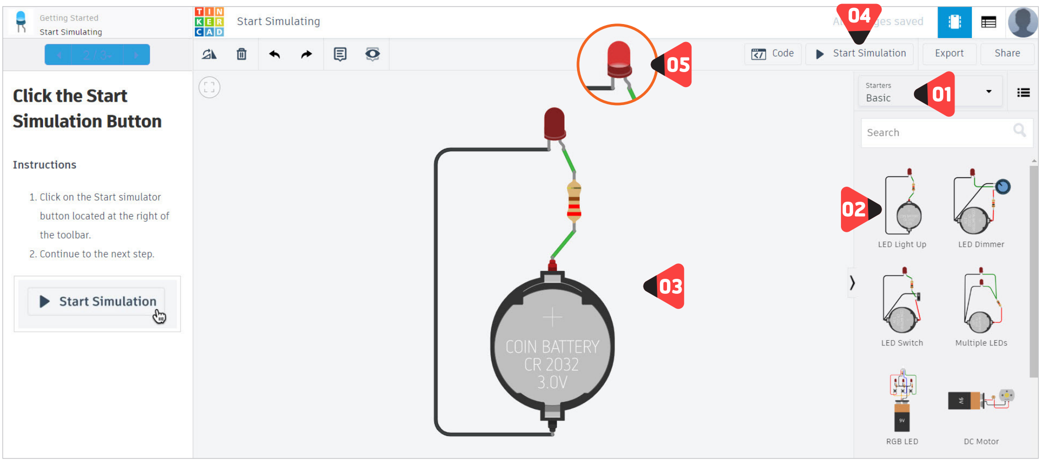
ส่วนประกอบของเว็บเพจ Learn Arduino

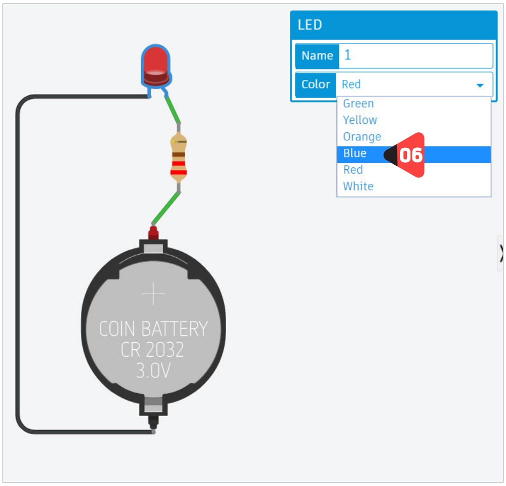
หน้าเว็บเพจ Learn Arduino ที่เราจะใช้ในการเรียนรู้ Arduino จะแบ่งออกเป็น 3 ส่วนหลักๆ ได้แก่ Lesson Area (ส่วนบทเรียน), Workplane Area (ส่วนพื้นที่สำหรับต่อวงจร/ประกอบชิ้นส่วน) และ Component Area (ส่วนแสดงรายการอุปกรณ์) ส่วนด้านบนของแต่ละพื้นที่จะเป็นแถบเครื่องมือ (Toolbar)

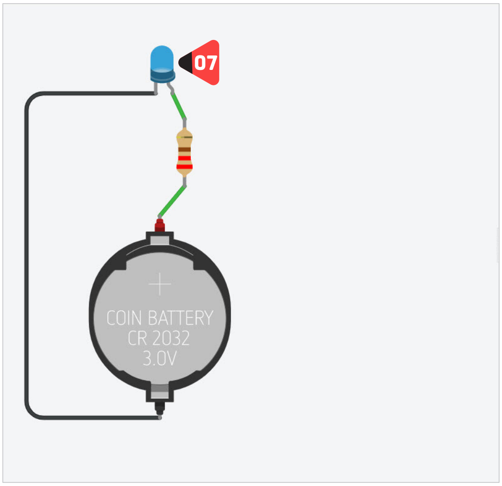
- **เครื่องมือของ Lesson Area** จะเป็น Menu Lesson มีปุ่มคลิกเดินหน้า-ถอยหลัง หรือ Move ไปยังหัวข้อที่ต้องการได้
- **เครื่องมือของ Workplane Area** จะมีไอคอนพื้นฐาน ได้แก่ Rotate, Trash, Undo, Redo, Annotation และ View/Hide
- **ปุ่มคำสั่งใช้งานทั่วไป** ได้แก่ Code, Start/Stop Simulation, Export และ Share (ปุ่มส่วนนี้ไม่ได้เกี่ยวข้องโดยตรงกับ Component Area แต่เป็นปุ่มใช้งานหลักทั่วไป)



หลอดสว่าง


Getting Started
Start Simulating
Start Simulating
Click the Start Simulation Button
Instructions
1. Click on the Start simulator button located at the right of the toolbar.
2. Continue to the next step.
Start Simulation
Code
Start Simulation
Export
Share
Starters
Basic
Search
LED Light Up
LED Dimmer
LED Switch
Multiple LEDs
RGB LED
DC Motor
COIN BATTERY
CR 2032
3.0V
01
02
03
04
05


LED
Name
1
Color
Red
Green
Yellow
Orange
Blue
Red
White
06
COIN BATTERY
CR 2032
3.0V


07
COIN BATTERY
CR 2032
3.0V

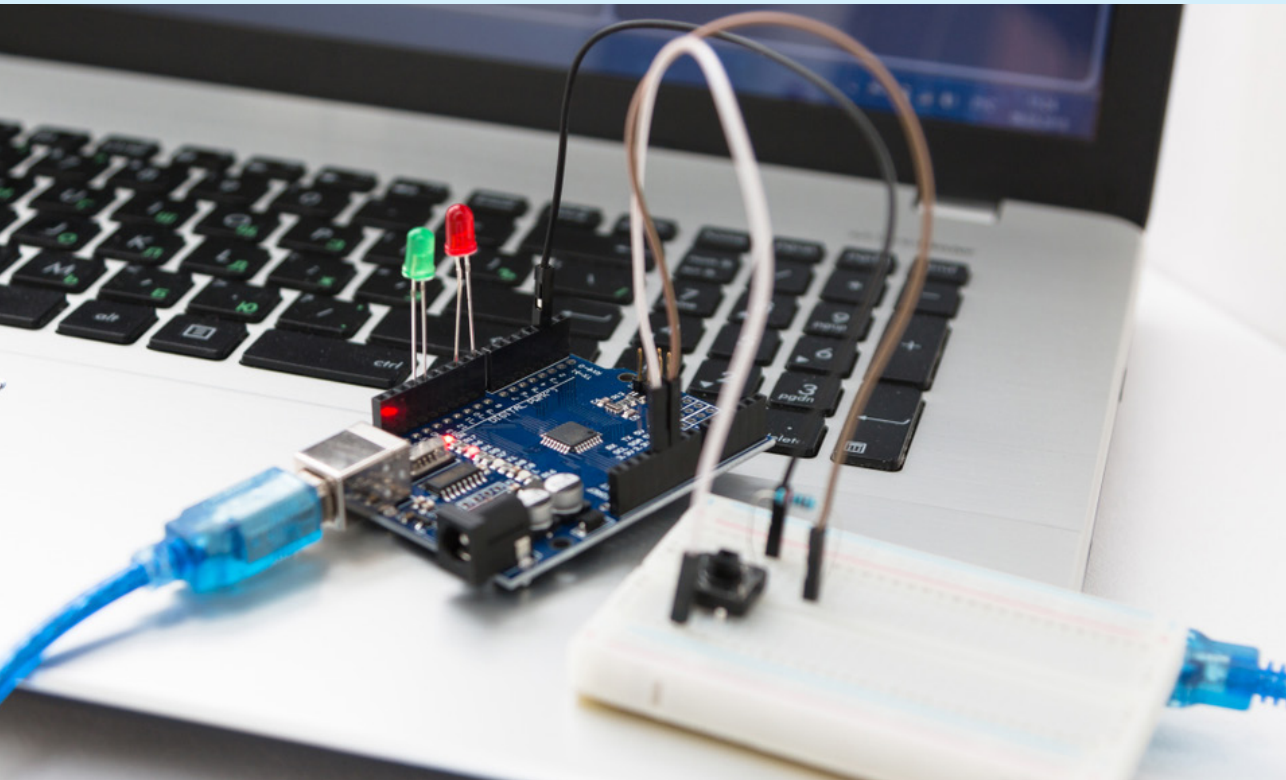
CHAPTER

05

เริ่มใช้งานบอร์ด Arduino ครั้งแรก

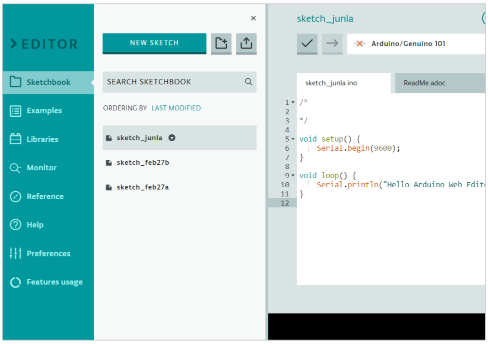
How to get started with Arduino

เมื่อเราตัดสินใจซื้อบอร์ด Arduino และได้มันมาไว้ในมือแล้ว แต่หลายคนคงติดปัญหาเดียวกัน คือ ไม่รู้จะเริ่มต้นอย่างไรกับบอร์ด? บทนี้คือวิธีการเริ่มต้นใช้งานบอร์ด Arduino ครอบคลุมการติดตั้งซอฟต์แวร์ การเชื่อมต่อ Arduino IDE กับ Arduino Board ต่อด้วยการเรียนรู้ก้าวแรกผ่าน Built-in Examples โดยในขั้นตอนนี้ผู้อ่านยังไม่จำเป็นต้องมีทักษะอะไรมากนัก แค่เปิดดูตัวอย่างและอ่านทำความเข้าใจ Document Project ต่อวงจรตามแบบ โหลดโค้ดลงบอร์ด สังเกตผลลัพธ์ และทำซ้ำกับตัวอย่างอื่นๆ



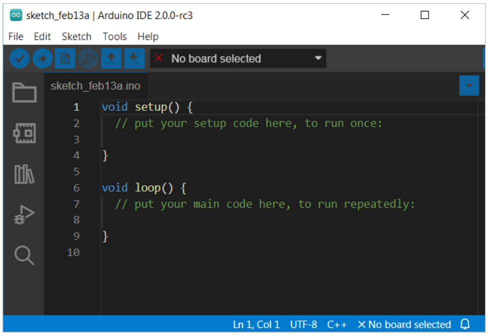
ทางเลือกในการเขียนโปรแกรม

บอร์ด Arduino นั้นจะทำอะไรไม่ได้เลยหากไม่มีชุดคำสั่งควบคุมการทำงาน โดยชุดคำสั่งนี้เราจะต้องเขียนขึ้นมาจากคอมพิวเตอร์แล้วโหลดเข้าไปที่บอร์ด เพื่อให้บอร์ดทำงานตามที่เราต้องการ ไม่ต่างจากคอมพิวเตอร์ซึ่งเป็นเพียงฮาร์ดแวร์ หากปราศจากซอฟต์แวร์แล้วมันก็หมดประโยชน์ ในการเขียนชุดคำสั่งหรือโค้ดดิง (Coding) เราเลือกได้ 2 ทาง ดังนี้



▲ รูปแสดงเว็บเพจ Arduino Web Editor

1. **Arduino Web Editor (Coding Online)** คือ การเขียนโปรแกรมผ่านทางออนไลน์ หรือ Arduino Cloud ในเว็บไซต์ arduino.cc กรณีนี้ต้องเชื่อมต่อกับอินเทอร์เน็ตในการเข้าใช้งานโปรแกรมแบบ Online IDE ไม่ต้องติดตั้งโปรแกรมลงเครื่อง เราสามารถเขียนโปรแกรมผ่านทาง Arduino Web Editor ได้โดยคลิกที่ Code Online แล้ว Login เพื่อเข้าใช้งาน จะปรากฏหน้าจอดังรูป



▲ รูปแสดงหน้าต่างโปรแกรม Arduino IDE

2. **Arduino IDE (Coding Offline)** เป็นการเขียนโปรแกรมบนคอมพิวเตอร์ของเราโดยไม่ต้องต่ออินเทอร์เน็ต กรณีนี้ต้องดาวน์โหลดและติดตั้งโปรแกรมลงในเครื่องคอมพิวเตอร์ก่อน โดยเลือกรุ่นของ Arduino IDE ตามระบบปฏิบัติการของเครื่องคอมพิวเตอร์ที่เราใช้ จากนั้นติดตั้งโปรแกรมตามขั้นตอน และเปิดโปรแกรมจะปรากฏหน้าจอดังรูป

การเขียนโปรแกรมบน Arduino IDE ครั้งแรก

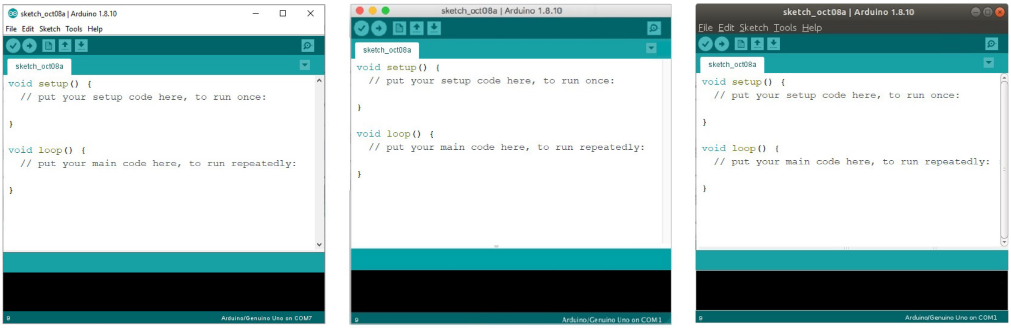
Step 1 ดาวน์โหลดและติดตั้ง Arduino IDE

Arduino IDE เป็นโปรแกรมใช้เขียนโปรแกรม หรือชุดคำสั่งที่สามารถแปลงภาษาที่มนุษย์เข้าใจ (Human-Readable) ซึ่งในที่นี้คือ ภาษา C/C++ เป็นภาษาที่คอมพิวเตอร์เข้าใจ (Machine-Readable) ผ่านตัวคอมไพเลอร์ (Compiler) แล้วอัพโหลดไฟล์โปรแกรมที่เขียนลงบนบอร์ด Arduino เพื่อให้มันทำงานตามคำสั่งที่เราได้เขียนโปรแกรมไว้

▲ เว็บเพจดาวน์โหลดโปรแกรม Arduino IDE

https://www.arduino.cc/en/software

เมื่อดาวน์โหลดและติดตั้งโปรแกรม Arduino IDE เสร็จแล้ว เมื่อเปิดใช้งานครั้งแรกจะปรากฏหน้าต่างโปรแกรมเหมือนกันแม้จะรันบนระบบปฏิบัติการที่แตกต่างกัน ดังรูป



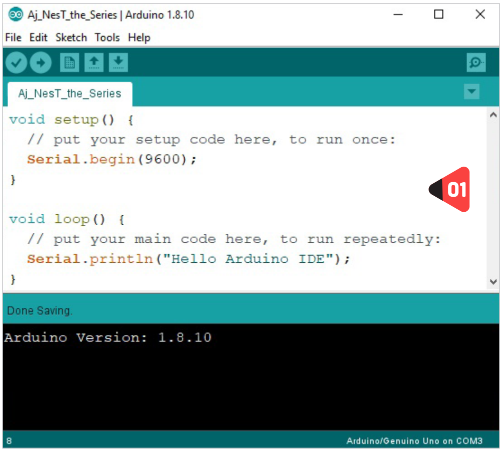
Windows macOS Linux

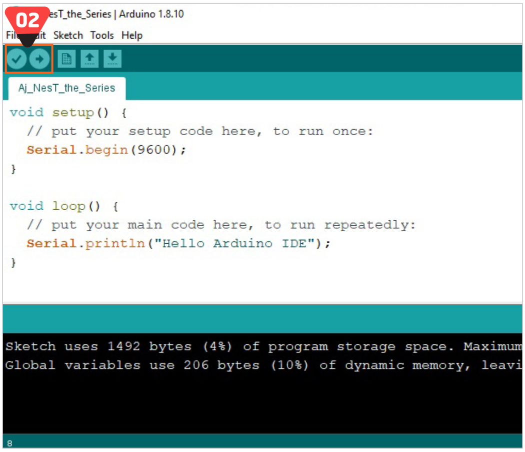
▲ หน้าตาโปรแกรม Arduino IDE

Step 3.3 ทดลองเขียนโปรแกรมบน Arduino IDE ครั้งแรก

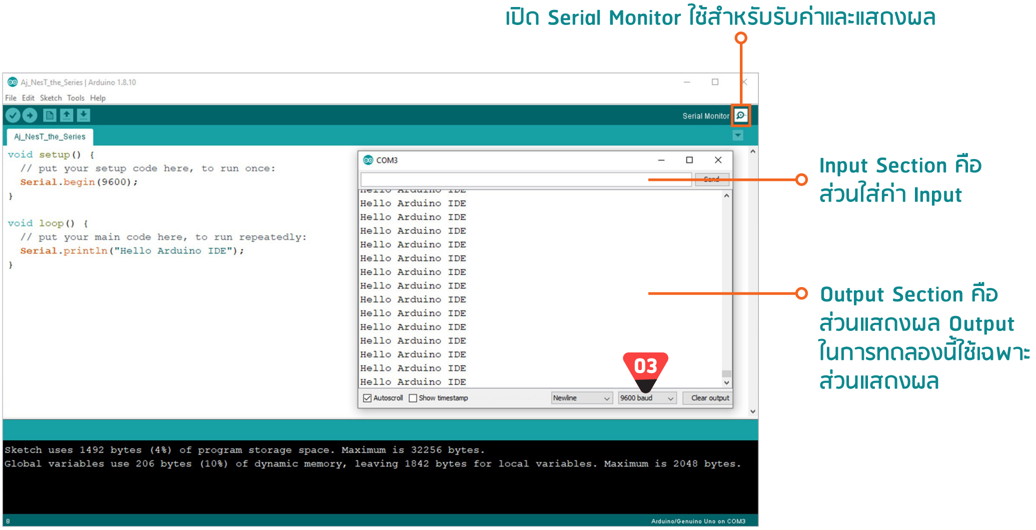
1. เขียนโปรแกรมแสดงผลออกทางหน้าจอ **Serial Monitor**

2. คลิกปุ่ม **Verify** และ **Upload**

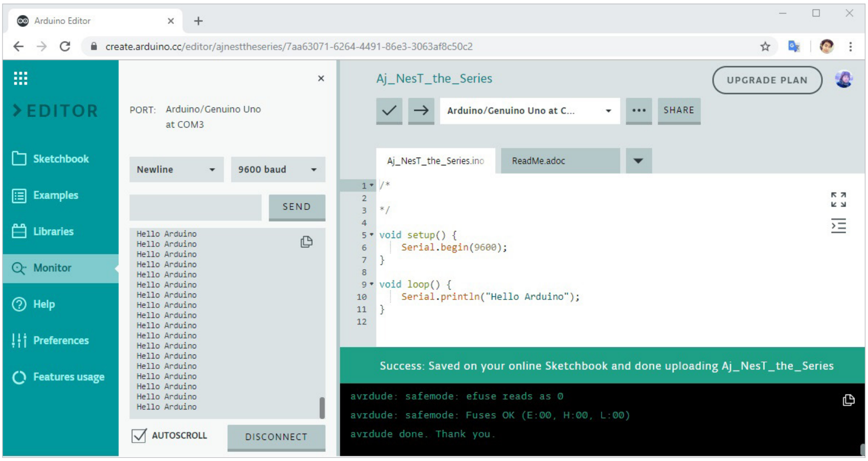




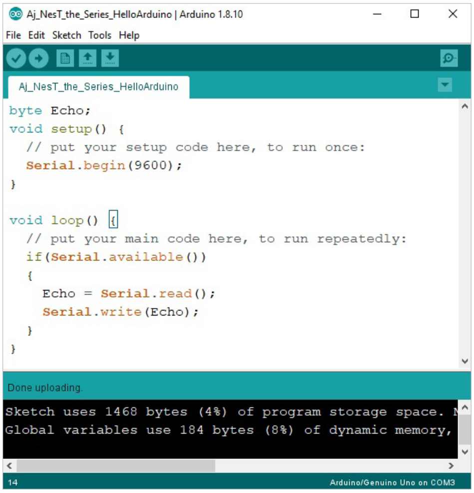
3. เปิด **Serial Monitor** กำหนดค่า **Baud Rate** เป็น **9600 baud** ผลลัพธ์หน้าจอจะแสดงคำว่า **Hello Arduino IDE** ขึ้นบรรทัดใหม่ และพิมพ์ซ้ำไปเรื่อยๆ

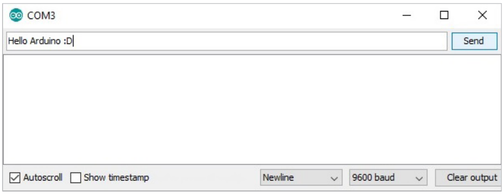


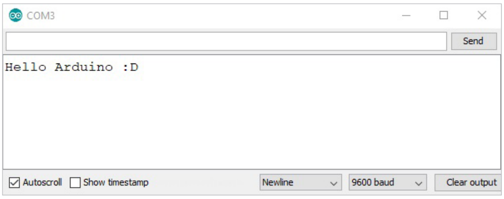
ทดสอบการแสดงผลข้อความ **“Hello Arduino”** บน Serial Monitor ในโปรแกรม Arduino Web Editor



ทดสอบการรับข้อความจากการพิมพ์ **“Hello Arduino :D”** และแสดงออกบน Serial Monitor ในโปรแกรม Arduino IDE

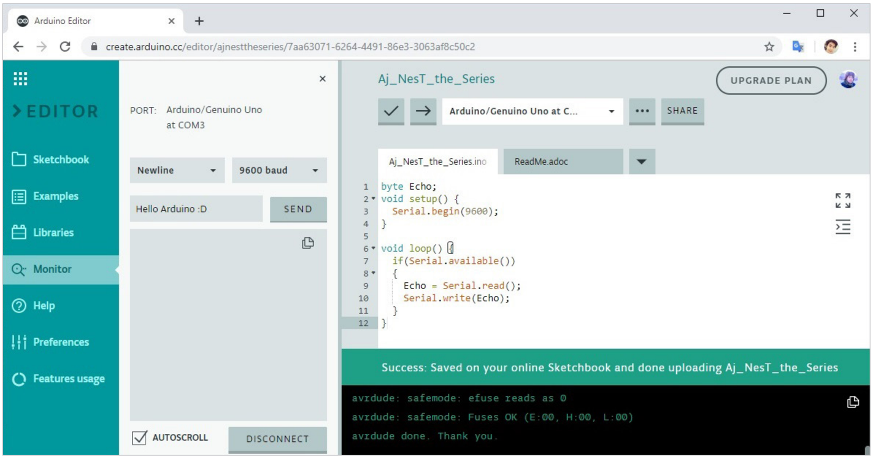




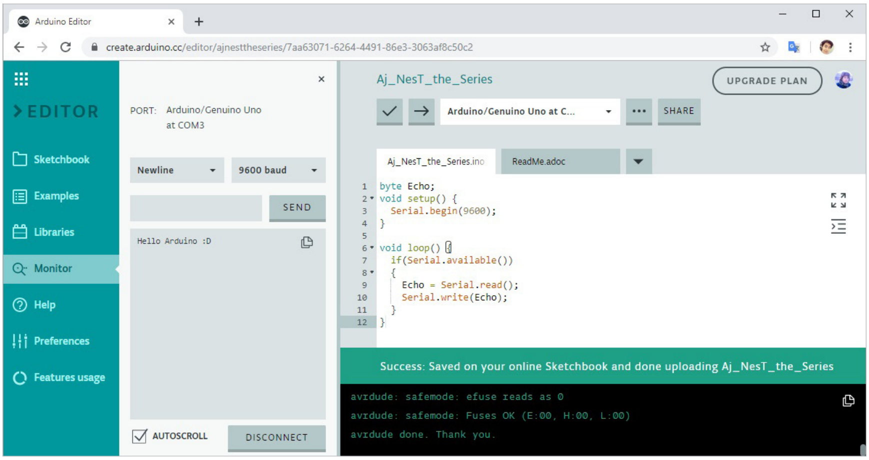


ทดสอบการรับข้อความจากการพิมพ์ **"Hello Arduino :D"** และแสดงออกบน Serial Monitor ในโปรแกรม Arduino Web Editor

รับข้อมูล Input ข้อความ "Hello Arduino :D"



แสดงข้อมูล Input ข้อความ "Hello Arduino :D" บนหน้าจอ Output ของ Serial Monitor



CHAPTER

06

การเขียนโปรแกรมควบคุมการทำงาน

Arduino Programming & Circuit Lab

ที่ผ่านมาเราได้เรียนรู้พื้นฐานการใช้งานซอฟต์แวร์ พร้อมทดลองเขียนโปรแกรมเบื้องต้น เพื่อเตรียมความพร้อมก่อนเริ่มเขียนโปรแกรมควบคุมการทำงานบอร์ด Arduino อย่างจริงจังกันในบทนี้ เริ่มตั้งแต่สรุปหลักการเขียนโปรแกรมด้วยภาษา C บนแพลตฟอร์ม Arduino พร้อมกันนี้ในหัวข้อ Arduino Tutorial จะเป็นการทำแล็บง่ายๆ ที่มีจุดประสงค์เพื่อให้ผู้เรียนเข้าถึงวิธีเขียนโปรแกรมในทางทฤษฎีและปฏิบัติไปพร้อมๆ กัน เพื่อสร้างความเข้าใจที่ลึกซึ้งมากยิ่งขึ้น

6.1 กลุ่มคำสั่งมีอะไรบ้าง (Arduino Programming)

เพื่อให้ทุกคนเข้าใจการเขียนโปรแกรม Arduino เราจะต้องรู้ก่อนว่ามีคำสั่งอะไรให้เราเล่นบ้าง หนังสือเล่มนี้ได้รวบรวมคำสั่งทั้งหมดโดยแบ่งออกเป็น 3 กลุ่ม ได้แก่ 1. Structures 2. Values และ 3. Functions

6.1.1 โครงสร้างทางไวยากรณ์ (Structures)

โครงสร้างทางไวยากรณ์ คือ กลุ่มคำสั่งประเภทโครงสร้างซึ่งในการเขียนโปรแกรมบน Arduino จะคล้ายกับการเขียนโปรแกรมโดยทั่วไป แต่ต่างกันตรงที่โครงสร้างการเขียนโปรแกรม Arduino จะมีฟังก์ชัน setup() และ loop() ในการทำงานเฉพาะของการเขียนโปรแกรมควบคุมบอร์ด Arduino

รายละเอียดคำสั่งประเภทโครงสร้าง ประกอบด้วย

1. **Sketch :** ตัวโปรแกรมหลัก
2. **Control Structure :** คำสั่งควบคุมการทำงาน
3. **Further Syntax :** ไวยากรณ์เสริม
4. **Operators :** ตัวดำเนินการต่างๆ

1

Sketch
loop()
setup()

2

Control Structure
break
continue
do...while
else
for
goto
if
return
switch...case
while

3

Further Syntax
#define (define)
#include (include)
/* */ (block comment)
// (single line comment)
; (semicolon)

4

Arithmetic Operators
% (remainder)
* (multiplication)
+ (addition)
- (subtraction)
/ (division)
= (assignment operator)

Comparison Operators
!= (not equal to)
< (less than)
<= (less than or equal to)
== (equal to)
> (greater than)
>= (greater than or equal to)

Boolean Operators
! (logical not)
&& (logical and)
|| (logical or)

Pointer Access Operators
& (reference operator)
* (dereference operator)

Bitwise Operators
& (bitwise and)
<< (bitshift left)
>> (bitshift right)
^ (bitwise xor)
| (bitwise or)
~ (bitwise not)

Compound Operators
%= (compound remainder)
&= (compound bitwise and)
*= (compound multiplication)
++ (increment)
+= (compound addition)
-- (decrement)
-= (compound subtraction)
/= (compound division)
^= (compound bitwise xor)
|= (compound bitwise or)

6.1.2 ตัวแปรและค่าคงที่ (Values ; Variables, Constants)

ตัวแปรและค่าคงที่ คือ กลุ่มคำสั่งประเภทกำหนดตัวแปร (Variables) และค่าคงที่ของตัวแปร (Constants) ของการเขียนโปรแกรม Arduino ในส่วนนี้จะคล้ายกับคำสั่งสำหรับการเขียนโปรแกรมโดยทั่วไป แต่ต่างกันตรงที่การกำหนดค่าคงที่ให้กับตัวแปรนั้น มีชื่อเฉพาะสำหรับเขียนโปรแกรมบน Arduino เช่น HIGH | LOW, INPUT | OUTPUT | INPUT_PULLUP, LED_BUILTIN เป็นต้น

รายละเอียดคำสั่งการจัดการค่าตัวแปรและค่าคงที่ ประกอบด้วย

1. **Constants** : การกำหนดค่าคงที่
2. **Conversion** : การแปลงประเภทข้อมูล
3. **Data Types** : ชนิดข้อมูล
4. **Variable Scope & Qualifiers** : การกำหนดขอบเขตและระบุชนิดข้อมูล
5. **Utilities** : คำสั่งช่วยอื่นๆ

Constants	1
HIGH | LOW	
INPUT | OUTPUT | INPUT_PULLUP	
LED_BUILTIN	
true | false	
Floating Point Constants	
Integer Constants	

Conversion	2
(unsigned int)	
(unsigned long)	
byte()	
char()	
float()	
int()	
long()	
word()	

Data Types	3
array	
bool	
boolean	
byte	
char	
double	
float	
int	
long	
short	
size_t	
string	
String()	
unsigned char	
unsigned int	
unsigned long	
void	
word	

Variable Scope & Qualifiers	4
const	
scope	
static	
volatile	

Utilities	5
PROGMEM	
sizeof()	

1 Digital I/O
- digitalRead()
- digitalWrite()
- pinMode()

2 Analog I/O
- analogRead()
- analogReference()
- analogWrite()

3 Zero, Due & MKR Family
- analogReadResolution()
- analogWriteResolution()

4 Advanced I/O
- noTone()
- pulseIn()
- pulseInLong()
- shiftIn()
- shiftOut()
- tone()

5 Time
- delay()
- delayMicroseconds()
- micros()
- millis()

6 Math
- abs()
- constrain()
- map()
- max()
- min()
- pow()
- sq()
- sqrt()

7 Trigonometry
- cos()
- sin()
- tan()

8 Characters
- isAlpha()
- isAlphaNumeric()
- isAscii()
- isControl()
- isDigit()
- isGraph()
- isHexadecimalDigit()
- isLowerCase()
- isPrintable()
- isPunct()
- isSpace()
- isUpperCase()
- isWhitespace()

9 Random Numbers
- random()
- randomSeed()

10 Bits and Bytes
- bit()
- bitClear()
- bitRead()
- bitSet()
- bitWrite()
- highByte()
- lowByte()

11 External Interrupts
- attachInterrupt()
- detachInterrupt()

12 Interrupts
- interrupts()
- noInterrupts()

13 Communication
- Serial
- Stream

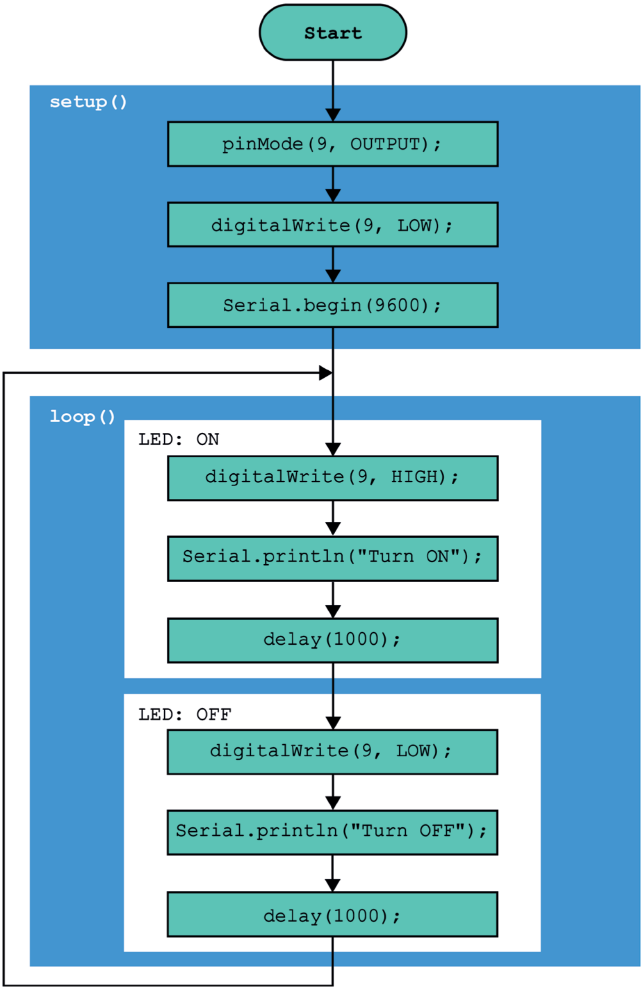
14 USB
- Keyboard
- Mouse

จากการรวบรวมคำสั่งต่างๆ ของการเขียนโปรแกรมบน Arduino เพื่อให้เกิดความเข้าใจ เราจะมาเริ่มทำความรู้จักกับคำสั่งต่างๆ ผ่านการเขียนโปรแกรมว่ามีการทำงานและเรียกใช้งานอย่างไร และผู้อ่านสามารถศึกษารายละเอียดอื่นๆ เพิ่มเติมได้ที่เว็บไซต์ของ Arduino ตามลิงค์ต่อไปนี้

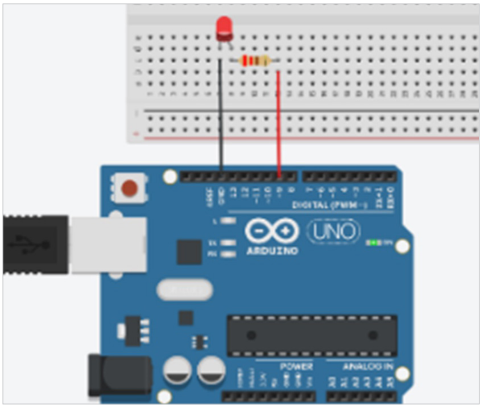
https://www.arduino.cc/reference/en/

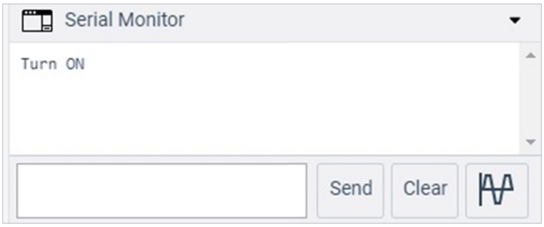
Parameter : Pin คือ หมายเลขขา Pin บนบอร์ด Arduino
Value : ค่า HIGH (เท่ากับ 1 V) หรือ LOW (เท่ากับ 0 V)

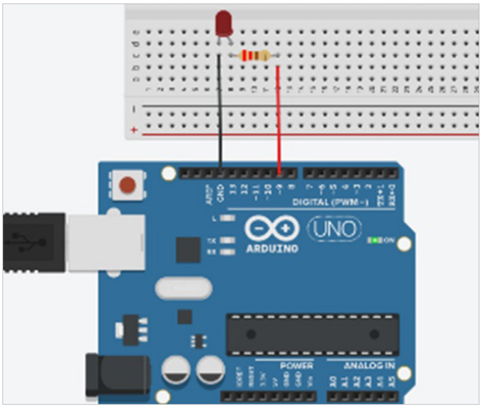
Step 3 : Flowchart Arduino Tutorial 2

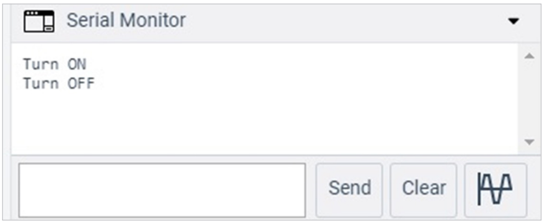


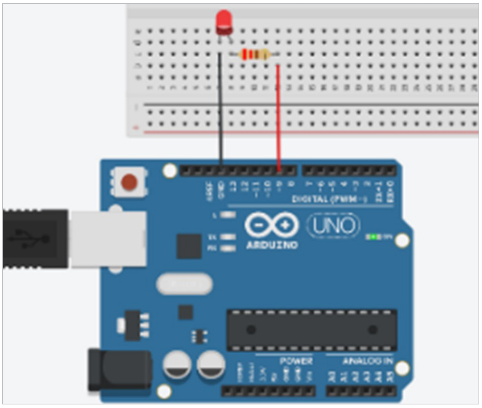
Tinkercad Output

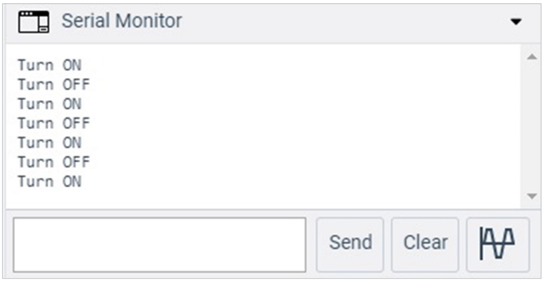




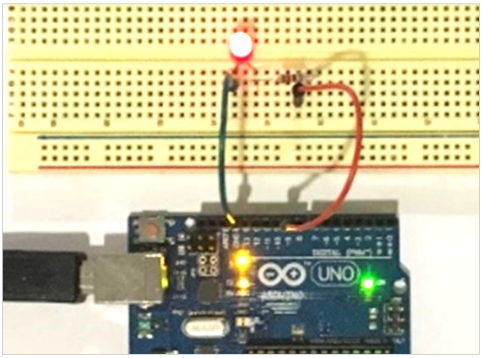


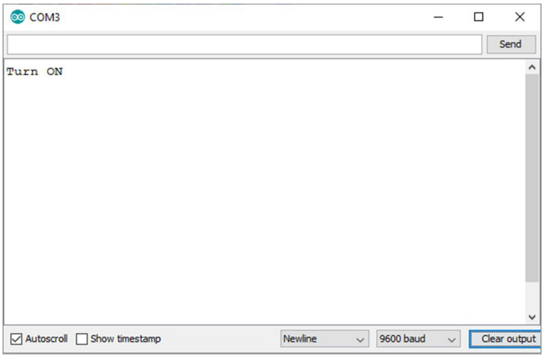


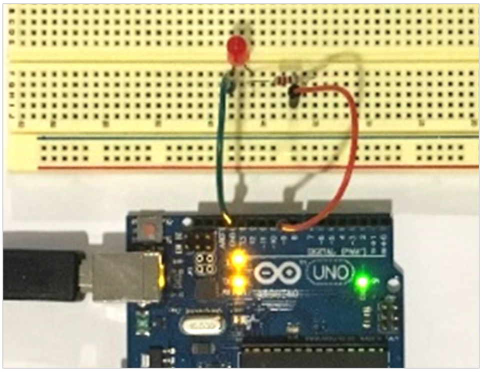


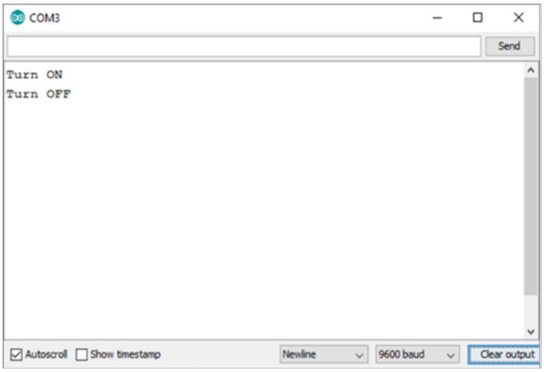


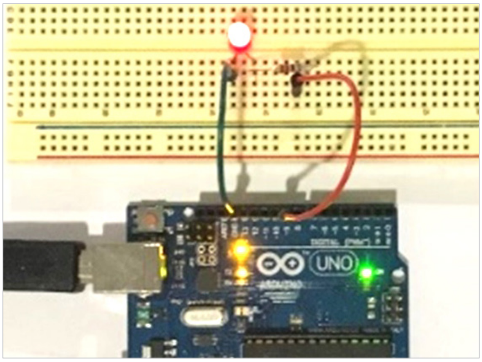
Arduino Board Output

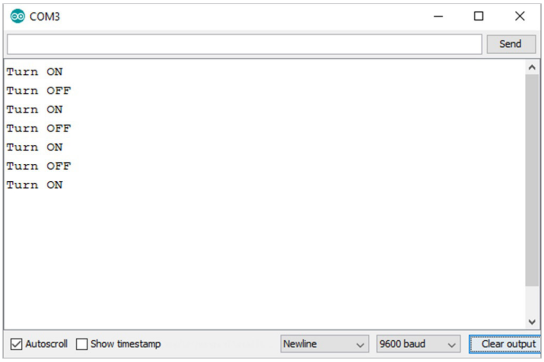






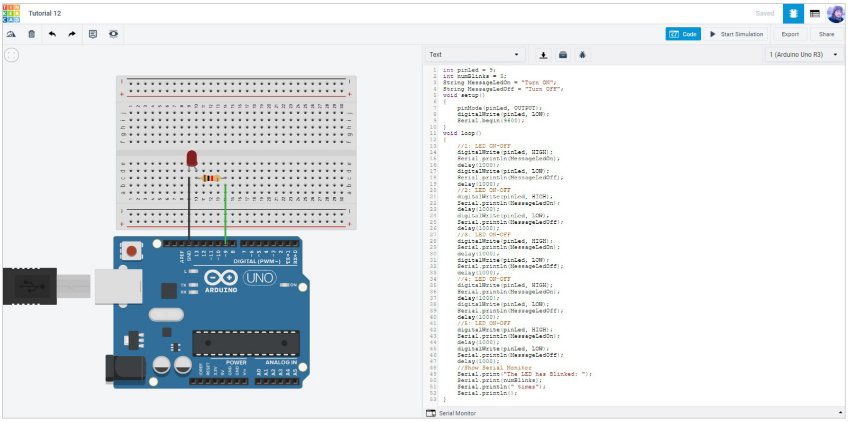




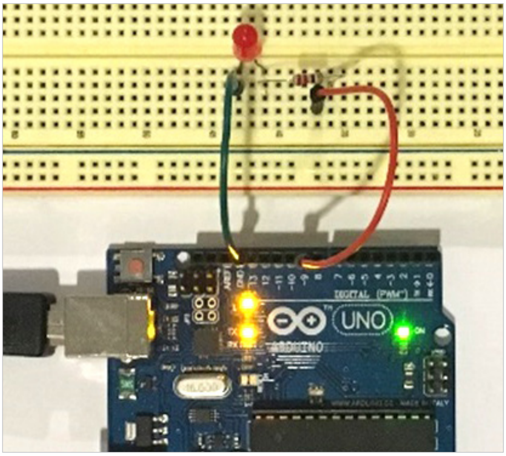


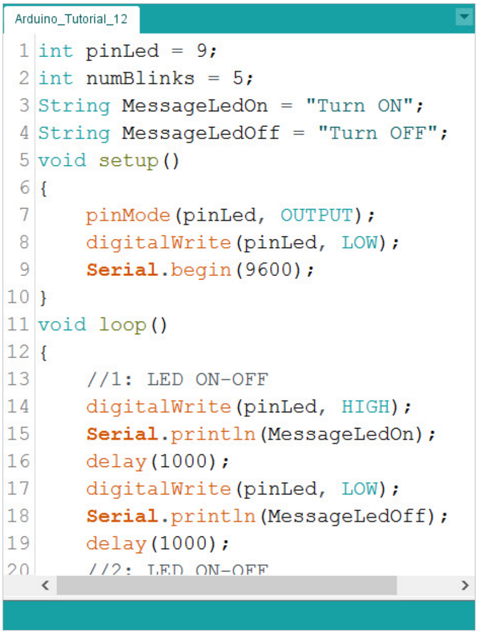
Step 5 : วิธีการทดลอง Arduino Tutorial 12

Tinkercad



Arduino Board






```
int pinLed = 9;
int numBlinks = 5;
String MessageLedOn = "Turn ON";
String MessageLedOff = "Turn OFF";
void setup()
{
    pinMode(pinLed, OUTPUT);
    digitalWrite(pinLed, LOW);
    Serial.begin(9600);
}
void loop()
{
    //1: LED ON-OFF
    digitalWrite(pinLed, HIGH);
    Serial.println(MessageLedOn);
    delay(1000);
    digitalWrite(pinLed, LOW);
    Serial.println(MessageLedOff);
    delay(1000);
    //2: LED ON-OFF
```

บทสรุปท้ายบท

การเรียนการเขียนโปรแกรมในบทนี้ เป็นการเรียนโครงสร้างไวยากรณ์ของภาษา C/C++ บนแพลตฟอร์ม Arduino ซึ่งมีลักษณะเฉพาะที่แตกต่างจากการเขียนโปรแกรมโดยปกติ ด้วยเหตุนี้จึงได้นำแล็บ Arduino Tutorial เข้ามาเป็นส่วนหนึ่งในการเรียน เพื่อจะได้เข้าใจ Syntax และ Parameter ของคำสั่ง และการนำไปใช้จริงในการเขียนโปรแกรมควบคุมการทำงาน

การสอนภาษา C อาจจะไม่ครบถ้วนครอบคลุมทุกคำสั่ง หรืออาจไม่ได้มีตัวอย่างการเขียนโปรแกรมที่หลากหลาย เหตุเพราะมีเนื้อหาอีกมากซึ่งต้องอาศัยเวลาอีกพอสมควรในการรวบรวมเรียบเรียงนั่นเอง เนื้อหาในบทนี้จึงถือเป็นจุดเริ่มต้นที่ไม่ยากเกินไปสำหรับมือใหม่หัดเขียนโปรแกรมควบคุมบอร์ดไมโครคอนโทรลเลอร์ ที่ต้องการก้าวแรกเพื่อสร้างกำลังใจ ก่อนจะมุ่งหน้าสู่การเรียนรู้ในระดับที่สูงขึ้นต่อไป