Raspberry Pi 3 บอร์ดดอมพิวเตอร์ขนาดเล็ก

รู้จักแล:การใช้งานเบื้องต้น



ผศ.โอภาส ศิริครรชิตถาวร

วิทยาลัยเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

นวพร เหล่าวัฒนธรรม กฤษดา ใจเย็น วรพจน์ กรแก้ววัฒนกุล ธีรวุธ จิตพรมมา



2 • Raspberry Pi 3 : รู้จักและการใช้งานเบื้องตัน

Raspberry Pi 3 บอร์ดดอมพิวเตอร์ 32 บิต : รู้จักและการใช้งานเบื้องต้น

ผศ.โอภาส ศิริครรชิตถาวร, สมเกียรติ กิจวงศ์วัฒนะ, นวพร เหล่าวัฒนธรรม, กฤษดา ใจเย็น, วรพจน์ กรแก้ววัฒนกุล, ธีรวุธ จิตพรมมา

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

ห้ามการลอกเลียนไม่ว่าส่วนหนึ่งส่วนใดของหนังสือเล่มนี้ นอกจากจะได้รับอนุญาต

ใครควรใช้หนังสือเล่มนี้

1. นักเรียน นิสิต นักศึกษา และบุคคลทั่วไปที่มีความสนใจใช้งานบอร์ดคอมพิวเตอร์ 32 บิตขนาดเล็ก สมัยใหม่ที่ติดตั้งระบบปฏิบัติการที่เหมาะสมสำหรับการใช้งานเป็น Embedded Computer หรือ Embedded PC ที่มีประสิทธิภาพสูง

2. สถาบันการศึกษา โรงเรียน วิทยาลัย มหาวิทยาลัย ที่มีการเปิดการเรียนการสอนวิชาอิเล็กทรอนิกส์ หรือภาควิชาวิศวกรรมอิเล็กทรอนิกส์และคอมพิวเตอร์

 คณาจารย์ที่มีความต้องการศึกษา และเตรียมการเรียนการสอนเกี่ยวกับระบบสมองกลฝังตัวทั้งใน ระดับเบื้องต้นจนถึงขั้นก้าวหน้าในระดับอาชีวศึกษา และปริญญาตรี

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

รายละเอียดที่ปรากฏในหนังสือเล่มนี้ได้ผ่านการตรวจทานอย่างละเอียดและถ้วนถี่ เพื่อให้มีความสมบูรณ์และ ถูกต้องมากที่สุดภายใต้เงื่อนไขและเวลาที่พึงมีก่อนการจัดพิมพ์เผยแพร่ ความเสียหายอันอาจเกิดจากการนำข้อมูล ในหนังสือเล่มนี้ไปใช้ ทางบริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด มิได้มีภาระในการรับผิดชอบแต่ประการใด ความผิดพลาดคลาดเคลื่อนที่อาจมีและได้รับการจัดพิมพ์เผยแพร่ออกไปนั้น ทางบริษัทฯ จะพยายามชี้แจงและแก้ไข ในการจัดพิมพ์ครั้งต่อไป

ด่าน่า

Raspberry Pi คือบอร์ดไมโครคอมพิวเตอร์แบบแแผ่นเดี่ยวที่โด่งดังที่สุดเท่าที่เคยมีมา มันคือ บอร์ดคอมพิวเตอร์ชื่อผลไม้ที่บรรจุความสามารถไว้อย่างมากมาย รองรับระบบปฏิบัติการ Linux สำหรับการพัฒนาไปสู่บอร์ด Embedded Linux ในราคาประหยัด แต่อุดมด้วยความสามารถขั้นเทพ มาพร้อมจุดเชื่อมต่ออุปกรณ์อินพุตเอาต์พุตทั้งผ่านพอร์ต USB, LAN, HDMI, ช่องสัญญาณภาพ และ GPIO สำหรับต่อกับวงจรหรืออุปกรณ์อิเล็กทรอนิกส์

Raspberry Pi เป็นบอร์ดคอมพิวเตอร์ 32 บิตที่ได้รับความนิยมสูงเป็นประวัติการณ์ของวงการ ระบบสมองกลฝังตัวโลก ด้วยยอดจำหน่ายมากกว่า 1 ล้านบอร์ดในเวลาไม่ถึงหนึ่งปีหลังจากที่ออก จำหน่ายครั้งแรกในวันที่ 29 กุมภาพันธ์ ค.ศ. 2012 และนับถึงวันที่จัดทำหนังสือเล่มนี้มียอดจำหน่าย ไปมากกว่า 10 ล้านชิ้น (นับถึงเดือนกันยายน ค.ศ. 2016) Raspberry Pi มีดีอย่างไร อะไรเป็นปัจจัย หลักที่ทำให้บอร์ดคอมพิวเตอร์ 32 บิตขนาดเล็กนี้โด่งดังไปทั่วโลก

ประเด็นหลักคือ *คุณสมบัติทางเทคนิคที่ยอดเยี่ยมที่มาพร้อมกับราคาที่ไม่แพง* ราคาของ Raspberry Pi ที่เปิดจำหน่ายในอังกฤษคือ US\$35 หรือ GBP25 (ราคาขายในไทยประมาณ 1,605 ถึง 2,140 บาท รวมภาษีและค่าส่งแล้ว) นั่นทำให้นักเล่นบอร์ดคอมพิวเตอร์แทบทุกรายให้ความสนใจในทันที **ด้านคุณสมบัติทางเทคนิค** ด้วยการใช้ชิปควบคุมหลักเป็นชิปของ Broadcom เบอร์ BCM2835 ใน Raspberry Pi รุ่นแรกที่รวมซีพียู, หน่วยประมวลกราฟิกหรือ GPU และหน่วยความจำ SDRAM ไว้ภาย ในตัวถังเดียวกัน โดย BCM2835 บรรจุซีพียู ARM11 ความเร็ว 700MHz, หน่วยประมวลกราฟิกหรือ GPU รุ่น Broadcom VideoCore IV ที่รองรับการแสดงผลผ่านจอภาพแบบ HDMI ด้วยความละเอียดในระดับ HD 1080p และมีหน่วยความจำ SDRAM มากถึง 512MB (เพิ่มเป็น 1GB ในรุ่น Raspberry Pi 2)

Raspberry Pi ยังมีจุดต่อ USB 2.0 จำนวน 2 พอร์ต (เพิ่มเป็น 4 พอร์ตในรุ่น Raspberry Pi B+, 2 และ 3) รองรับการต่อเมาส์ คีย์บอร์ด และอุปกรณ์ USB อื่นๆ, แจ๊ก AV และ HDMI เพื่อต่อกับโทรทัศน์ หรือจอแสดงผลที่มีจุดต่อแบบ RCA ตัวเมียหรือ HDMI, จุดต่ออีเธอร์เน็ตหรือ LAN, คอนเน็กเตอร์หรือ จุดต่อพอร์ตอินพุตเอาต์พุต (General Purpose Input/Output : GPIO) ที่มีขาต่อบัส SPI (Serial Peripheral Interface Bus), I²C, I²S, ขาสัญญาณรับส่งข้อมูลอนุกรมหรือ UART และซ็อกเก็ตของ SD การ์ดสำหรับเสียบ SD การ์ดที่ติดตั้งระบบปฏิบัติการไว้แล้ว

นั่นคือ Raspberry Pi สามารถใช้งานเป็นคอมพิวเตอร์ตัวน้อยเพื่อการเขียนโปรแกรมทั่วๆ ไป ก็ได้ หรือจะใช้เป็นบอร์ดควบคุมตัวเก่งในระบบสมองกลฝังตัวที่มีการเชื่อมต่อกับวงจรหรืออุปกรณ์ อิเล็กทรอนิกส์ก็ได้

4 • Raspberry Pi 3 : รู้จักแล:การใช้งานเบื้องตัน

ผู้ผลิต Raspberry Pi คือ **มูลนิธิ Raspberry Pi** (Raspberry Pi Foundation – www. raspberrypi.org) ยังพัฒนาบอร์ดอย่างต่อเนื่อง เป็นรุ่น Raspberry Pi 2 ที่เปลี่ยนซีพียูเป็นเบอร์ BCM2836 อันเป็นชิป ARM Cortex-A7 ที่มี 4 แกน ความเร็วในการประมวลผลสูงขึ้นถึง 900MHz และ เพิ่มหน่วยความจำแรมเป็น 1GB ทำให้ Raspberry Pi 2 ทำงานเร็วขึ้นถึง 6 เท่า ด้วยราคาจำหน่ายเดิม จนถึงต้นปี ค.ศ. 2016 ได้พัฒนาต่อเป็น Raspberry Pi 3 ที่ทำงานเร็วขึ้น 10 เท่าเมื่อเทียบกับรุ่นแรก

ใน Raspberry Pi 3 ได้เปลี่ยนซีพียูเป็น BCM2837 ซึ่งเป็น ARM Cortex A53 ขนาด 64 บิต แบบ 4 แกน ความเร็ว 1.2GHz และได้รวมวงจรสื่อสารไร้สายทั้ง WiFi และ BLE (Bluetooth Low Energy) หรือบลูทูธกำลังงานต่ำไว้ในตัว ทำให้ Raspberry Pi 3 ติดต่อกับเครือข่ายในแบบไร้สายได้ โดยไม่ต้องใช้ USB WiFi ดองเกิลอีกต่อไป

ระบบปฏิบัติการ Linux เป็นตัวช่วยสำคัญของ Raspberry Pi โดยติดตั้งลงใน SD การ์ดที่ทำ หน้าที่เสมือนเป็นฮาร์ดดิสก์ของคอมพิวเตอร์ ทำให้ Raspberry Pi กลายเป็นคอมพิวเตอร์สมบูรณ์แบบ ขนาดเล็กที่ใครๆ ก็ใช้งาน นับเป็นการปฏิวัติวงการ Embedded PC ครั้งใหญ่ที่สุดในประวัติศาสตร์ ด้วยการพัฒนาอย่างต่อเนื่องทำให้ปัจจุบันระบบปฏิบัติการของ Raspberry Pi มีความสวยงาม และ รวบรวมซอฟต์แวร์ที่สำคัญสำหรับการพัฒนาโปรแกรมไว้ให้พร้อมใช้งาน ทำให้เกิดความสะดวกในการ ใช้งานเพิ่มขึ้นเป็นอย่างมาก

เพราะราคาของ Raspberry Pi ถูกและมีขีดความสามารถสูง ทำให้มีโปรแกรมเมอร์มือสมัคร เล่นและชั้นเซียนเข้าถึงได้มาก ต่างก็ช่วยกันพัฒนาโปรแกรมประยุกต์ออกมาอย่างมากมาย รวมถึง ซอฟต์แวร์ที่ใช้ในการเรียนรู้การเขียนโปรแกรมอย่างง่ายอย่าง Scratch ส่งผลให้เยาวชนสามารถเข้า ถึงการเขียนโปรแกรมและใช้งานคอมพิวเตอร์ได้อย่างกว้างขวาง ไม่เพียงเท่านั้น Raspberry Pi ยัง เชื่อมต่อกับเครือข่ายอินเทอร์เน็ตได้อย่างง่ายดาย มีเว็บบราวเซอร์ให้เข้าถึงเว็บไซต์ต่างๆ ได้ ทั้งยัง ทำงานร่วมกันบนเครือข่ายได้สะดวก ทั้งหมดนี้เราพบเห็นได้ทั่วไปกับคอมพิวเตอร์สมัยใหม่แทบทุกตัว แต่ไม่ใช่ที่ราคาต่ำกว่า 2,500 บาท

ดังนั้น จึงเชื่อได้อย่างสนิทใจว่า Raspberry Pi คือหนึ่งในบอร์ดคอมพิวเตอร์ที่ผู้สนใจในระบบ สมองกลฝังตัวสมัยใหม่ส่วนใหญ่ต้องเรียนรู้และใช้งานให้เป็น หนังสือเล่มนี้จึงเกิดขึ้นเพื่อเป็นตัวช่วย สำหรับผู้สนใจเริ่มต้นใช้งานบอร์ดคอมพิวเตอร์ตัวเก่งและยอดนิยมตัวนี้

> **ชัยวัฒน์ ลิ้มพรจิตรวิไล** บรรณาธิการ

ด่าซี้แจงจากดณ:ผู้เขียน/เรียบเรียง

การนำเสนอข้อมูลเกี่ยวกับข้อมูลทางเทคนิคและเทคโนโลยีในหนังสือเล่มนี้ เกิดจากความต้อง การที่จะอธิบายกระบวนการและหลักการทำงาน ของอุปกรณ์ในภาพรวมด้วยถ้อยคำที่ง่ายเพื่อสร้าง ความเข้าใจแก่ผู้อ่าน ดังนั้นการแปลคำศัพท์ทางเทคนิคหลายๆ คำอาจไม่ตรงตามข้อบัญญัติของราช บัณฑิตยสถาน และมีหลายๆ คำที่ยังไม่มีการบัญญัติอย่างเป็นทางการ คณะผู้เขียนจึงขออนุญาต บัญญัติศัพท์ขึ้นมาใช้ในการอธิบาย โดยมีข้อจำกัดเพื่ออ้างอิงในหนังสือเล่มนี้เท่านั้น

ทั้งนี้สาเหตุหลักของข้อชี้แจงนี้มาจาก การรวบรวมข้อมูลของอุปกรณ์ในระบบสมองกลฝังตัว และเทคโนโลยีหุ่นยนต์สำหรับการศึกษาเพื่อนำมาเรียบเรียงเป็นภาษาไทยนั้นทำได้ไม่ง่ายนัก ทางคณะ ผู้เขียนต้องทำการรวบรวมและทดลองเพื่อให้แน่ใจว่า ความเข้าใจในกระบวนการทำงานต่างๆ นั้นมีความ คลาดเคลื่อนน้อยที่สุด

เมื่อต้องทำการเรียบเรียงออกมาเป็นภาษาไทย ศัพท์ทางเทคนิคหลายคำมีความหมายที่ทับซ้อน กันมาก การบัญญัติศัพท์จึงเกิดจากการปฏิบัติจริงร่วมกับความหมายทางภาษาศาสตร์ ดังนั้นหากมีความ คลาดเคลื่อนหรือผิดพลาดเกิดขึ้น ทางคณะผู้เขียนขอน้อมรับและหากได้รับคำอธิบายหรือชี้แนะจากท่าน ผู้รู้จะได้ทำการชี้แจงและปรับปรุงข้อผิดพลาดที่อาจมีเหล่านั้นโดยเร็วที่สุด

ทั้งนี้เพื่อให้การพัฒนาสื่อทางวิชาการ โดยเฉพาะอย่างยิ่งกับความรู้ของเทคโนโลยีสมัยใหม่ สามารถดำเนินไปได้อย่างต่อเนื่อง ภายใต้การมีส่วนร่วมของผู้รู้ในทุกภาคส่วน

บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด

6 • Raspberry Pi 3 : รู้จักและการใช้งานเบื้องตัน

สารบัญ

บทที่ 1	Raspberry Pi รายงานตัว7
บทที่ 2	เตรียมการใช้งาน Raspberry Pi 3 25
บทที่ 3	Raspberry Pi กับการเชื่อมต่อระบบเครือข่ายไร้สาย WiFi และการควบคุมผ่านคอมพิวเตอร์จากภายนอก47
บทที่ 4	คำสั่งพื้นฐานบนเทอร์มินอลของระบบปฏิบัติการ Linux 67
บทที่ 5	เริ่มต้นรู้จักกับภาษา Python 89
บทที่ 6	Raspberry Pi 3 กับการพัฒนาโปรแกรมภาษา Python ด้วยซอฟต์แวร์ Geany119
บทที่ 7	การติดต่อพอร์ตอินพุตเอาต์พุตอเนกประสงค์ หรือ GPIO ของ Raspberry Pi 3131
บทที่ 8	การตั้งค่าให้ชุดคำสั่งทำงานหลังจากบอร์ด Raspberry Pi บูตระบบเสร็จสิ้น163

บทที่ 1 Raspberrry Pi รายงานตัว



Raspberry Pi คือบอร์ดคอมพิวเตอร์ขนาดเล็กที่มีราคาถูก ทว่ามีความสามารถเทียบเท่ากับ คอมพิวเตอร์ขนาดย่อมๆ รองรับการใช้งานได้เหมือนคอมพิวเตอร์เครื่องหนึ่ง ต่อจอภาพและอุปกรณ์ USB เพื่อใช้งานได้ ไม่ว่าจะเป็นเมาส์ คีย์บอร์ด หรือ USB WiFi ทั้งยังมีจุดเด่นที่ต่างจากคอมพิวเตอร์ ทั่วไปคือ มีพอร์ตอินพุตเอาต์พุตหรือ GPIO ให้ใช้งาน จึงทำให้นำ Raspberry Pi นำไปต่อกับอุปกรณ์ ต่างๆ เพื่อประยุกต์การทำงานที่เกี่ยวกับอิเล็กทรอนิกส์ได้

จุดประสงค์ของบอร์ด Raspberry Pi คือ ใช้ในด้านการศึกษาและเรียนรู้สำหรับเยาวชน แต่ด้วย กวามสามารถ ขนาดและราคา ทำให้ Raspberry Pi กลายเป็นบอร์ดยอดนิยมสำหรับนักเล่น นักทดลอง ระบบสมองกลฝังตัวในวงกว้างเป็นอย่างมาก เพราะประยุกต์ใช้งานได้หลากหลาย นอกเหนือจากด้าน การศึกษา ไม่ว่าจะเป็นระบบอัตโนมัติอย่าง Home Automation หรือระบบเซิร์ฟเวอร์หรือหน่วยบริการ ข้อมูลขนาดย่อมก็ทำได้ ในปัจจุบันมีผู้ผลิตสินค้าหลายรายนำ Raspberry Pi ไปใช้งานจริงในเชิงพาณิชย์

1.1 จุดกำเนิดของ Rasapberry Pi

Raspberry Pi ถูกสร้างขึ้นภายใต้แนวคิดที่ไม่ซับซ้อนว่า "ต้องการให้เด็กๆ หัดเขียนโปรแกรมด้วย เครื่องมือที่ราคาถูก" แนวคิดนี้มาจากมูลนิธิ Raspberry Pi (www.raspberrypi.org) โดยในตอนแรกไม่ ได้คิดถึงการสร้างคอมพิวเตอร์ขนาดเล็กเลยด้วยซ้ำ ทางที่จะทำให้เด็กๆ สนใจและหัดเขียนโปรแกรมคือ หาเครื่องมือให้พวกเขาในแบบฟรีหรือถูกที่สุดเท่าที่จะเป็นไปได้ Raspberry Pi จึงถูกสร้างขึ้นโดย Eben Vpton และคณะทำงานในมูลนิธิ Raspberry Pi มีการแจกจ่ายแก่ผู้เรียนในสาขาวิทยาการคอมพิวเตอร์ ของมหาวิทยาลัยเคมบริดจ์ สหราชอาณาจักร เพื่อเรียนรู้และระดมสมอง โดยมีเป้าหมายไปสู่การพัฒนา สื่อเพื่อช่วยให้เด็กๆ สนใจในการเขียนโปรแกรม นั่นเป็นจุดเริ่มต้นที่ทำให้มีคนรู้จัก Raspberry Pi แต่ก็ ยังจำกัดอยู่ในแวดวงการศึกษาระดับมหาวิทยาลัยเท่านั้น

ย้อนหลังไปในปี ค.ศ. 2006 ต้นแบบแรกของ Raspberry Pi ใช้ไมโครคอนโทรลเลอร์ ATmega644 ของ Atmel โดยมีการเผยแพร่วงจรและแบบลายทองแดงของแผ่นวงจรพิมพ์สู่สาธารณะ ด้วยการคำเนิน การในลักษณะนี้ Eben Upton ผู้ให้กำเนิด Raspberry Pi สามารถรวบรวมกลุ่มของครู อาจารย์ นักวิชาการ และผู้เชี่ยวชาญด้านคอมพิวเตอร์ ทั้งด้านซอฟต์แวร์และฮาร์ดแวร์ เพื่อแลกเปลี่ยนข้อมูล ข้อคิดเห็น จนนำไปสู่การเลือกชิปประมวลผลตัวใหม่โดยได้รับแรงบันดาลใจจากเครื่องคอมพิวเตอร์ BBC Micro ของ Acom ซึ่งเป็นผู้ผลิตชิป ARM *เดิม Raspberry Pi ถูกตั้งชื่อว่า ABC computer* เนื่องจากต้องการสื่อ สารทางการตลาดว่า *นี่คือคอมพิวเตอร์เพื่อการเริ่มต้นเทียบได้กับหนังสือเรียนภาษาอังกฤษ ABC นั่นเอง*

8 • Raspberry Pi 3 : รัจักและการใช้งานเบื้องตัน

้จากนั้น Upton ได้ผลิตบอร์ดคอมพิวเตอร์ต้นแบบตัวใหม่ออกมาโดยใช้ ARM โปรเซสเซอร์ มี ้งนาดเท่ากับ USB แฟลช ใครฟ โคยมีจุดต่อพอร์ต USB 1 ช่อง และพอร์ตต่อจอภาพ HDMI อีก 1 ช่อง ้นั่นแสดงให้เห็นถึงวิสัยทัศน์ของ Upton ที่เชื่อว่า จอแสดงผลกวามละเอียคสูงที่ใช้ช่องทางการสื่อสารแบบ HDMI (High-Definition Multimedia Interface) จะเป็นคำตอบสำหรับการใช้งานคอมพิวเตอร์ในยุคใหม่ และ Raspberry Pi จะต้องมีความสามารถนี้เป็นหลัก

Upton ต้องรออีกถึง 5 ปี จนถึงเดือนสิงหาคม ปี ค.ศ. 2011 บอร์ดเวอร์ชัน Alpha จำนวน 50 ้ชุดได้ถูกผลิตออกมา เพื่อใช้ในการทดสอบและนำเสนอต่อผู้สนใจ จนถึงเดือนธันวาคมในปีเดียวกัน Raspberry Pi Model B ในรุ่น Beta จำนวน 25 บอร์คได้รับการทดสอบอีกครั้ง โดยคราวนี้รันด้วย ระบบปฏิบัติการ Linux เล่นไฟล์วิดีโอความละเอียดในระดับ HD ด้วยความละเอียด 1080p ทำเอาโลก ทั้งใบแทบจะหยุดหมุน สปอตไลต์ของวงการ Embedded System และคอมพิวเตอร์ขนาดเล็กหันไป ้ส่องและ โฟกัสไปที่ Raspberr Pi แทบจะในทันที และเฝ้ารอวันเปิดตัวอย่างเป็นทางการ

ในสัปดาห์แรกของปีค.ศ. 2012 บอร์ด Raspberry Pi จำนวน 10 บอร์ดถูกส่งไปประมูลในเว็บ ใซต์ E-Bay หนึ่งในนั้นได้รับการประมูลจากพิพิธภัณฑ์ The Centre for Computing History ใน Suffolk สหราชอาณาจักร บอร์ด Raspberry Pi ล็อตแรกทั้ง 10 บอร์ดสามารถสร้างมูลค่าจากการประมูลได้ มากกว่า 16,000 ปอนด์ โคยบอร์ดที่มีเลขรหัสประจำตัว 01 มีมูลค่าสูงถึง 3,500 ปอนด์



techcentral.co.za)

รูปที่ 1–1 Eben Upton ผู้ให้กำเนิดบอร์ด รูปที่ 1–2 บอร์ดต้นแบบรุ่นแรกของ Raspberry Pi ที่พัฒนาขึ้น Raspberry Pi (ภาพจาก http://www. ในปี ค.ศ. 2006 โดยใช้ ATmega644 ของ Atmel ทำงานด้วย สัญญาณนาฬิกาความถี่ 22.1MHz มีหน่วยความจำ SRAM 512 กิโลไบต์ สร้างสัญญาณภาพที่ความละเอียด 320x240 จุด (ภาพจาก http://www.raspberrypi.org)



รูปที่ 1–3 Eben Upton ในวัยหนุ่มกับ ต้นแบบของ Raspberry Pi ที่เริ่มต้นใช้ชิป ARM ขนาดเท่ากับ USB แฟลชไดรฟ มีจุดต่อพอร์ต USB และ HDMI อย่างละ 1 ช่อง (ภาพจาก http:/www.ixwebhost ing.mobi/2011/12/29/12379. html)

ต่อมาโครงการ Raspberry Pi นี้ได้รับการขยายผลโดย BBC (British Broadcasting Corporation - www.bbc.com) อันเป็นเครือข่ายวิทยุและ โทรทัศน์ที่ทรงอิทธิพลของสหราชอาณาจักรและของโลก ทันทีที่ทั่วโลกได้รับข่าวสารของ Raspberry Pi ยอดการสั่งซื้อก็หลั่งไหลมาอย่างมากมาย บริษัทที่ดูแล การผลิตและจำหน่ายทั้ง RS และ Element 14 ต้องวางแผนการผลิตใหม่ เพื่อรองรับกับกระแสความ นิยมในตัว Raspberry Pi

Raspberry Pi ขายได้ 10,000 ชุดภายในชั่วโมงแรกที่มีการเปิดขายอย่างเป็นทางการ จากนั้น สินค้าก็ขาดตลาด ต้องมีการสั่งจองล่วงหน้า บางรายต้องรอนานกว่า 12 สัปดาห์ เมื่อ Raspberry Pi กลับเข้ามามีจำหน่ายอีกครั้งด้วยจำนวนกว่า 100,000 บอร์ด ผู้เกี่ยวข้องต้องพบกับความประหลาด ใจอีกครั้งเมื่อมันถูกจำหน่ายหมดภายในวันแรกของการเปิดจำหน่าย จนถึงต้นปี ค.ศ. 2013 ยอด จำหน่ายของ Raspberry Pi ก็ผ่านหลัก 1 ล้านชิ้น นอกจากนั้น Google ยักษ์ใหญ่ของอุตสาหกรรมไอ ทีอีกรายก็ได้บริจาก Raspberry Pi กว่า 15,000 บอร์ดแก่โรงเรียนทั่วสหราชอาณาจักร เพื่อส่งเสริม ให้เด็กๆ ได้เรียนรู้การเขียนโปรแกรมคอมพิวเตอร์ในแบบง่ายๆ อันเป็นการสร้างฐานในการพัฒนา บุคลากรด้านไอทีสำหรับอนาคต

รูปที่ 1–4 Eben Upton ในฐานะผู้ก่อตั้ง มูลนิธิ Raspberry Pi มอบสิทธิ์ในการ ผลิตและจำหน่ายบอร์ด Raspberry Pi แก่ Element 14 และ RS Components (ภาพจาก http://www.cambridge– news.co.uk)



1.2 Raspberry Pi ทำอะไรได้บ้าง

ทุกอย่างที่คอมพิวเตอร์ใหญ่ๆ ทำได้ Raspberry Pi ก็ทำได้ ! แถมยังมีความยืดหยุ่นมากกว่า เมื่อนำไปใช้ในโครงงานระบบสมองกลฝังตัวเพราะมีขนาดเล็กกว่ามาก มีพอร์ต GPIO สำหรับเชื่อม ต่อกับอุปกรณ์อิเล็กทรอนิกส์ได้สะดวก ทำให้การส่งสัญญาณเพื่อควบคุมอุปกรณ์ภายนอกอื่นๆ ทำได้อย่างมีประสิทธิภาพและเป็นไปได้สำหรับคนในทุกระดับเพียงมีความรู้ด้านการเขียนโปรแกรม

มีโครงงานจำนวนมากในเครือข่ายอินเทอร์เน็ตให้ดูเป็นตัวอย่าง ที่โคคเค่นมากคือ การนำไป ใช้งานด้านความบันเทิง ทำเป็นเครื่องเล่นสื่อผสมหรือ Media player ราคาประหยัด โดยต่อ Raspberry Pi เข้ากับฮาร์คดิสก์ที่บรรจุไฟล์เพลงหรือภาพยนต์ แล้วทำการเรียกออกมาเพื่อเล่นกลับภายใต้คุณภาพ ในระดับ HD เมื่อใช้กับจอภาพแบบ HDMI หรือการนำไปสร้างเป็นสมาร์ตทีวีหรือ โทรทัศน์อัจฉริยะ ที่รับชมรายการผ่านเครือข่ายอินเทอร์เน็ต สามารถท่องเว็บไซต์ เล่นเกมออนไลน์ โดยทั้งหมดนี้ไม่ต้อง ใช้คอมพิวเตอร์ขนาดใหญ่อีกต่อไป เพียงมี Raspberry Pi, แหล่งจ่ายไฟที่มีเสถียรภาพสูง และจอ HDMI ร่วมกับเครือข่ายอินเทอร์เน็ตความเร็วสูง ทุกอย่างทั้งความบันเทิงและสื่อความรู้ก็จะอยู่ในกำมือของ ผู้ใช้งาน

นอกจากนั้นยังนำ Raspberry Pi ไปเชื่อมต่อกับตัวตรวจจับและระบบสื่อสารไร้สายเพื่อสร้าง ระบบคาต้าล็อกเกอร์หรือระบบมอนิเตอร์ที่ต้องการงานประมวลผลข้อมูลที่สูงทั้งจำนวนและความ เร็ว นำไปสร้างหุ่นยนต์ที่มีขีคความสามารถสูงทั้งเพื่อการแข่งขันและใช้ในอุตสาหกรรม ทั้งหมคนี้ ทำได้ไม่ยากและเป็นไปได้ด้วยบอร์ค Raspberry Pi

1.2 รุ่นของบอร์ด Raspberry Pi

มูลนิธิ Raspberry Pi ได้พัฒนาและผลิตบอร์ด Raspberry Pi (นับถึงกันยายน ค.ศ. 2016) ออก มาจำหน่าย รวมแล้ว 8 รุ่น (จำหน่ายในตลาดสากล) ตามลำดับเวลาดังนี้

- 1. Raspberry Pi Model B
- 2. Raspberry Pi Model A
- 3. Raspberry Pi Model B+
- 4. Raspberry Pi Model A+
- 5. Raspberry Pi Compute module
- 6. Raspberry Pi 2
- 7. Rasberry Pi Zero
- 8. Raspberry Pi 3



รูปที่ 1 – 5 ตัวอย่างการนำไป Raspberry Pi ไปใช้ในการสร้างสิ่งประดิษฐ์

- (1) คอมพิวเตอร์เด็กแนวในสไตล์ R-Pi
- (2) สร้างซูเปอร์คอมพิวเตอร์ โดยใช้ตัวต่อ Lego มาสร้างเป็นกล่องบรรจุ
- (3) นาฬิกาย้อนยุคจากหลอด Nixxie ควบคุมด้วย Raspberry
- (4) ตู้เกมแบบ Retro
- (5) Rasperry Pi media center ดูหนัง ฟังเพลง ท่องเน็ต โดยไม่ต้องใช้คอมพิวเตอร์พีซี

(6) การทดลองใช้งาน GPIO เพื่อเชื่อมต่อวงจรอิเล็กทรอนิกส์ อีกหนึ่งความสามารถที่หาได้ ไม่ง่ายจากบอร์ด Embedded Linux รุ่นอื่น ๆ แต่ Raspberry Pi จัดให้ !

12 • Raspberry Pi 3 : รัจักแล:การใช้งานเบื้องต้น

1.2.1 Raspberry Pi Model B

เป็นบอร์ด Raspberry Pi ที่ออกวางจำหน่ายในเชิงพาณิชย์รุ่นแรก โดยใช้ซีพียู BCM2835 ซึ่ง เป็นชิป ARM11 คอร์ ARM1176JZF-S ความเร็ว 700MHz ที่รวมหน่วยประมวลกราฟิกหรือ GPU ไว้ภายในตัวถังเดียวกัน มีหน่วยความจำแรม 256MB ต่อมาเพิ่มเป็น 512MB

้ด้านพอร์ตเชื่อมต่ออุปกรณ์ภายนอก ประกอบด้วย พอร์ต USB 2.0 (2 พอร์ต), แจ๊ก RCA และ HDMI เอาต์พุตสัญญาณวิดีโอสำหรับต่อกับโทรทัศน์หรือจอแสดงผลที่มีจุดต่อแบบ RCA ตัวเมียหรือ HDMI, จุดต่อเอาต์พุตเสียงเป็นแจ๊กหูฟัง 3.5 มม., จุดต่ออีเธอร์เน็ตหรือจุดต่อระบบ LAN, คอนเน็กเตอร์ หรืองุดต่อพอร์ตอินพุตเอาต์พุต (General Purpose Input/Output : GPIO) ที่มีขาต่อบัส SPI (Serial Peripheral Interface Bus), I?C, I?S, ขาสัญญาณรับส่งข้อมูลอนุกรมหรือ UART และซ็อกเก็ตของ SD การ์ดสำหรับ เสียบ SD การ์ดที่ติดตั้งระบบปฏิบัติการเรียบร้อยแล้ว

1.2.2 Raspberry Pi Model A

เป็นบอร์ด Raspberry Pi ที่ออกตามหลังมา โดยมีการตัดอุปกรณ์ที่ไม่ได้ใช้งานออกไป เพื่อ ทำให้ต้นทุนลดลง สำหรับนำไปผลิตเป็นสินค้าเพื่อให้แข่งขันได้ โดยยังคงใช้ซีพียู BCM2835 หน่วย ้ความจำแรม 512MB มีพอร์ต USB 2.0 เหลือ 1 ช่อง, แจ๊ก RCA และ HDMI, จุดต่อเอาต์พุตเสียงเป็น แจ๊กหูฟัง 3.5 มม., จุดต่อพอร์ตอินพุตเอาต์พุต GPIO และซ็อกเก็ตของ SD การ์ด โดยตัดอุปกรณ์ใน ้ส่วนของการเชื่อมต่อ LAN ออกไป



รูปที่ 1-6 บอร์ด Raspberry Pi Model B ที่ออกวาง รูปที่ 1-7 บอร์ด Raspberry Pi Model A ที่ใช้แผ่นวงจร จำหน่ายเป็นรุ่นแรก



พิมพ์แบบเดียวกับ Model B แต่ไม่ติดตั้งหรือ ตัดอุปกรณ์บางตัวออกไป





รูปที่ 1-8 บอร์ด Raspberry Pi Model B+ ที่มีการปรับอย่างมาก รูปที่ 1-9 บอร์ด Raspberry Pi Model A+ ด้านรูปลักษณ์และจุดต่ออุปกรณ์ภายนอก

. ที่ออกแบบแผ่นวงจรพิมพ์ใหม่ทั้งหมด เพื่อให้มีขนาดเล็กลง

1.2.3 Raspberry Pi Model B+

หลังจากได้รับคำแนะนำอย่างมากมาย ทางมูลนิธิ Raspberry Pi จึงตัดสินใจปรับรูปแบบของ บอร์ค Raspberry Pi ใหม่ โดยยังคงใช้ซีพียู BCM2835 หน่วยความจำแรม 512MB มีจุดต่ออีเธอร์เนีต และพอร์ต HDMI

จุดที่ปรับคือ เพิ่มพอร์ต USB 2.0 เป็น 4 พอร์ต แล้วรวมแจ๊ก RCA แจ๊กหูฟังเข้าไว้ด้วยกัน กลายเป็นแจ๊ก AV ด้านคอนเน็กเตอร์หรือจุดต่อพอร์ตอินพุตเอาต์พุต GPIO เพิ่มเป็น 40 งา และเปลี่ยน ซ็อกเก็ตของ SD การ์ดเป็นแบบ micro SD การ์ด เพื่อรองรับการ์ดที่มีความจุสูงได้

1.2.4 Raspberry Pi Model A+

รุ่น A+ ได้รับการออกแบบแผงวงจรใหม่ทั้งหมด โดยไม่ใช้วิธีเดียวกับรุ่น A ทำให้บอร์ดมี ้งนาดเล็กลง นำไปติดตั้งในผลิตภัณฑ์ได้ง่ายขึ้น และต้นทุนรวมลดลงอย่างมีนัยสำคัญ โดยส่วนที่ตัด ้ออกคือ จุดต่ออีเธอร์เนีต และลดจำนวนพอร์ต USB ลงเหลือ 1 พอร์ตเหมือนกับรุ่น A

1.2.5 Raspberry Pi compute module

สำหรับบอร์ค Raspberry Pi รุ่นนี้ได้รับการพัฒนาขึ้นเพื่อนำไปสร้างระบบคอมพิวเตอร์ในแบบ เฉพาะตัว (custom system) โดยตัดส่วนของซีพียูและอุปกรณ์สำคัญออกมาเป็น โมดูลกล้าย โมดูลหน่วย ความจำ ดังรูปที่ 1-10 (ซ้ายสุด) ในโมดูลหลักประกอบด้วย โปรเซสเซอร์ BCM2835, แรม 512MB และหน่วยความจำแฟลช 4GB บรรจุลงในแผ่นวงจรพิมพ์ที่มีขนาคเพียง 67.6 × 30 มม. ตามขนาค มาตรฐานของโมดูล DDR2 SODIMM

14 • Raspberry Pi 3 : รู้จักและการใช้งานเบื้องตัน



รูปที่ 1–10 หน้าตาของ Rasberry Pi Compute Module Development kit (ภาพจาก http://www.cnx– software.com)

เพื่ออำนวยความสะดวกในการใช้งานโมดูล Raspberry Pi แบบนี้ บอร์ด Compute Module IO จึงเกิดขึ้น (รูปที่ 1-10 ลำดับที่ 2 จากซ้าย) สำหรับนำโมดูล Raspberry Pi compute ไปติดตั้ง โดยบอร์ด IO นี้จะมีจุดต่อไฟเลี้ยง, จุดต่อพอร์ต USB ทั้งแบบ A และ micro USB สำหรับบูตระบบจากโฮสต์ที่ ตอ่เข้ามาร่วมใช้งานด้วย, จุดต่อพอร์ต GPIO และจุดต่อ HDMI ส่วนการติดต่อกับโมดูลกล้องและ จอแสดงผลผ่านพอร์ต CSI และ DSI ก็จะมีบอร์ดอะแดปเตอร์มาช่วยดังในรูปที่ 1-10 (ขวามือ)

1.2.6 Raspberry Pi 2

หลังจากบอร์ด Raspberry Pi ออกวางจำหน่าย ก็มีผู้ผลิตอื่นๆ หลายรายผลิตบอร์ดคอมพิวเตอร์ ที่ทำงานในลักษณะเดียวกันออกมาจำนวนมาก ทางมูลนิชิ Raspberry Pi จึงได้ผลิตบอร์ดรุ่นใหม่ออก มา โดยเปลี่ยนซีพียูเป็น BCM2836 ซึ่งเป็นชิป ARM Cortex-A7 แบบ 4 แกน และเพิ่มหน่วยความจำ เป็น 1GB กำหนดชื่อเป็น **Raspberry Pi 2** มีความเร็วในการทำงานสูงขึ้นถึง 6 เท่า

การเปลี่ยนแปลงส่งผลต่อตลาคในทันที บอร์ค Raspberry Pi 2 ใค้รับการตอบรับคีมาก Microsoft ประกาศพัฒนาระบบปฏิบัติการวินโควส์ 10 ในรุ่นพิเศษ (Windows 10 IoT) ที่รองรับ Raspberry Pi 2 (เปิคให้คาวน์โหลคมาใช้งานได้ฟรีตั้งแต่ 29 กรกฎาคม พ.ศ. 2558) จึงทำให้บอร์ค Raspberry Pi 2 กลายเป็นบอร์คที่ผู้สนใจงานค้านนี้ต้องมี

รูปที่ 1–11 บอร์ด Raspberry Pi 2 ที่มีการปรับ หน้าตาเล็กน้อยเนื่องจากมีการเปลี่ยนซีพียู และเพิ่มความจุของหน่วยความจำเป็น 1GB





รูปที่ 1–12 บอร์ด Raspberry Pi Zero ซึ่งเป็นบอร์ดรุ่นที่เล็กสุด (ในขณะนี้) ของครอบครัว Raspberry Pi ในรูปขวามือเป็นการเปรียบเทียบกับขนาดของธนบัตร 5 เหรียญสหรัฐ

1.2.7 Raspberry Pi Zero

เป็นบอร์ดที่มีขนาดเล็กและราคาถูกที่สุดในอนุกรม Raspberry Pi มีราคาขายปลีก 5 เหรียญ สหรัฐ (ตามราคาในเว็บไซต์อย่างเป็นทางการ) Raspberry Pi Zero นำชิป BCM2835 แบบที่ใช้ใน Raspberry Pi A, B, A+ และ B+ มาใช้ รันด้วยความถิ่ของสัญญาณนาฬิกา ที่สูงขึ้นเป็น 1GHz มีหน่วย ความจำแรม 512MB และตัดลดชิปอื่นๆ ออกไปจากบอร์ด เหลือเพียงซีพียูและภาคจ่ายไฟ ด้านการ เชื่อมต่ออุปกรณ์อื่นๆ ก็ถูกตัดออกไปหมด เหลือเพียงพอร์ต USB 1 พอร์ตที่ลากออกมาจาก ซีพียู, พอร์ ตอินพุตเอาต์พุต GPIO 40 ขา, จุดต่อ RCA composite และจุดต่อ HDMI แบบมินิเท่านั้น ขนาดของ บอร์ดกือ ขนาด 65 x 30 x 5 มม.

เมื่อเทียบกับราคา 5 เหรียญสหรัฐของ Raspberry Pi Zero ถือว่า คุ้มราคามาก เพราะนี่คือ บอร์ ดคอมพิวเตอร์ที่มีซีพียู 1GHz แรม 512MB มีระบบปฏิบัติการเป็น Linux นอกจากนั้น ยังใช้งานกับ บอร์ด HAT (Hardware Attach on Top) ของ Raspberry Pi ได้ ในการทดสอบ

ด้วยขนาดที่เล็ก ทำให้นำไปประยุกต์ใช้ในโครงงานต่างๆ ได้ง่ายขึ้น เช่น สร้างโครน หุ่นยนต์ หรืออุปกรณ์สวมใส่อิเล็กทรอนิกส์แบบต่างๆ รวมถึงการนำไปให้เด็กๆ หัดใช้คอมพิวเตอร์เพื่อเล่นเกม Minecraft หรือให้หัดเขียนโปรแกรมด้วย Scratch อันเป็นโปรแกรมแบบลากแล้ววางเหมาะสำหรับ การสอนเด็กๆ ให้หันมาสนใจการเขียนโปรแกรม

1.2.8 Raspberry Pi 3

วันที่ 29 กุมภาพันธ์ พ.ศ. 2559 ที่ผ่านมาคือวันครบรอบปีที่ 4 ของ Raspberry Pi ทาง Raspberry Pi Foundation ได้นำ Raspberry Pi 3 ออกสู่ตลาดในวันครบรอบวันคล้ายวันเกิด Raspberry Pi 3 มาพร้อม กับซีพียูตัวใหม่ 64 บิต และติดตั้งชิปวงจรสื่อสารไร้สายสมัยใหม่ที่รวม WiFi และ BLE (Bluetooth Low Energy) ไว้ในตัวถังเดียวกัน เพื่อทำให้ Raspberry Pi 3 มีขีดความสามารถสูงขึ้น ใช้งานพอร์ต USB ใน การต่อกับอุปกรณ์ได้มากขึ้น เพราะไม่ต้องสำรองไว้สำหรับ USB WiFi และบลูทูธดองเกิลดังที่ผ่านมา

Raspberry Pi 3 ใช้ชิปประมวลผลหลักเป็น Broadcom BCM2837 ARM Cortex-A53 แบบ 4 แกน ความเร็ว 1.2GHz จึงเร็วกว่า Raspberry Pi รุ่นแรก 10 เท่า มีหน่วยความจำแรม 1GB และพอร์ต เชื่อมต่อต่างๆ อยู่ครบเหมือนรุ่น Raspberry Pi 2 ความ โคคเค่นแตกต่างจึงไปอยู่ที่การบรรจุวงจรWiFi และ Bluetooth 4 (Bluetooth Low Energy - BLE) ไว้บนบอร์คเลย ทำให้การเชื่อมต่อเครือข่ายแบบไร้ สายง่ายและสะควกขึ้นมาก แจ๊ก AV รวมเอาต์พุตเสียงและสัญญาณภาพหรือวิดีโอไว้ด้วยกันเป็น แจ๊กหูฟัง 3.5 มม.

ด้วยการปรับปรุงอย่างเข้าใจในการใช้งาน ทำให้ Raspberry Pi 3 กลายเป็นบอร์คหลักใหม่ของ ผู้ใช้งาน แม้ว่าจะมีราคาแพงขึ้นอีกเล็กน้อยก็ตาม



รูปที่ 1–13 หน้าตาของบอร์ด Raspberry Pi 3 จะเห็นว่า ในภาพรวมมีความคล้ายคลึงกับบอร์ด Raspberry Pi 2 ค่อนข้างมาก (ภาพจาก https://cdn.infoq.com)

1.3 คุณสมบัติทางเทคนิคที่สำคัญของ Raspberry Pi 3

● ชิปประมวลผลหลัก: Broadcom BCM2837 ARM Cortex-A53 แบบ 4 แกน ความเร็ว 1.2GHz

 หน่วยประมวลกราฟิกหรือ GPU : Broadcom VideoCore IV dual-core GPU หรือเทียบเท่า รองรับการแสดงผลผ่านจอ HDMI

• หน่วยความจำ SDRAM : 1GB LPDDR2

• ส่วนสื่อสารข้อมูล : พอร์ตอีเธอร์เน็ต 10/100, WiFi 2.4GHz 802.11n และ Bluetooth 4 (Bluetooth Low Energy - BLE)

• จุดต่อ :

USB 2.0 จำนวน 4 พอร์ต รองรับการขยายผ่าน USB ฮับ ใช้เชื่อมต่อกับอุปกรณ์ USB ได้ หลายแบบ อาทิ เมาส์, คีย์บอร์ค, WiFi คองเกิล, บลูทูธคองเกิล, ฮาร์คคิสก์ภายนอก, USB แฟลชไครฟ เป็นต้น

แจ๊ก AV ที่รวมเอาต์พุตเสียงและสัญญาณภาพหรือวิดีโอไว้ด้วยกันเป็นแจ๊กหูฟัง 3.5 มม.

คอนเน็กเตอร์หรือจุดต่อพอร์ตอินพุตเอาต์พุต 40 บา (General Purpose Input/Output : GPIO) ที่มีขาต่อบัส SPI (Serial Peripheral Interface Bus), I²C, I²S, บาสัญญาณรับส่งข้อมูลอนุกรมหรือ UART

ซ็อกเก็ตของ micro SD การ์คสำหรับเสียบ micro SD การ์คที่ติคตั้งระบบปฏิบัติการแล้ว

จุดต่อ CSI สำหรับต่อโมคูลกล้อง Raspberry Pi

จุคต่อ DSI สำหรับจอแสคงผล LCD

จุดต่อไฟเลี้ยง +5V ผ่านกอนเน็กเตอร์ microUSB

 ความต้องการไฟเลี้ยง : +5V 2.5A เป็นอย่างน้อย แนะนำใช้งานกับอะแคปเตอร์ไฟตรง +5V
 2.5A ขึ้นไป หรือเครื่องจ่ายไฟสำรอง Power Bank +5V ที่มีความสามารถในการจ่ายกระแสไฟฟ้า มากกว่า 2A ขึ้นไป

● ขนาด : 85 x 56 มม.



1.4 ส่วนประกอบที่สำคัญของ Raspberry Pi 3

ในรูปที่ 1-14 แสดงส่วนประกอบที่สำคัญทั้งหมดของบอร์ด Raspberry Pi 3 ข้อมูลเบื้องต้นที่ ควรทราบเกี่ยวกับบอร์ด Raspberry Pi 3 สรุปได้ดังนี้

1.4.1 สมองกลใหม่ 64 บิต

ซีพียูใหม่ของ Raspberry Pi 3 คือ Broadcom BCM2837 เป็นซิปที่ใช้ ARM Cortex-A53 แบบ 64 บิต ทำงานที่ความเร็ว 1.2GHz และมี Cache 32kB ที่ Level 1 เพื่อช่วยให้ซีพียูเข้าถึงคำสั่งที่ใช้ บ่อยที่สุด ได้เร็วขึ้น และ Cache 512kB ที่ Level 2 ช่วยให้ซีพียูทำงานได้เร็วขึ้นกว่าเดิม ต่างจาก Raspberry Pi 2 ที่ไม่มีหน่วยความจำแคช (Cache memory) เชื่อมต่อกับหน่วยความจำแรมที่มีขนาด 1GB เป็นแบบ LPDDR2 (Low Power DDR2) ที่กินพลังงานไฟฟ้าต่ำ และมีชิปประมวลผลกราฟิก เป็น VideoCore IV แบบ 2 แกนที่รองรับ OpenGL ES 2.0, OpenVG และรองรับ Decode H.264 ระดับ 1080p30



รูปที่ 1–15 ชิป Broadcom BCM2837 สมองกลตัวใหม่ของ Raspberry Pi 3

20 • Raspberry Pi 3 : รู้จักแล:การใช้งานเบื้องตัน



รูปที่ 1–16 แสดงส่วนเชื่อมต่ออุปกรณ์ภายนอกของบอร์ด Raspberry Pi 3

1.4.2 จุดต่ออุปกรณ์ภายนอก

มีจุดต่อ HDMI สำหรับต่อกับจอแสดงผลความละเอียดสูง รองรับ HDMI 1.3 และ 1.4, มีแจ๊ก AV 3.5 มม. สำหรับเอาต์พุตเสียงสเตอริโอและสัญญาณภาพแบบ Component Video, มีจุดต่อแหล่งจ่าย ไฟ +5V ผ่านทางคอนเน็กเตอร์ microUSB, รองรับการเชื่อมต่อ โมดูลกล้องผ่านคอนเน็กเตอร์ CSI (Camera Serial Interface), เชื่อมต่อกับจอแสดงผลผ่านคอนเน็กเตอร์ DSI (Display Serial Interface), รองรับการ เชื่อมต่อเครือข่าย LAN ผ่านจุดต่ออีเธอร์เน็ต และจุดต่อพอร์ต USB 4 ช่อง เพื่อต่อกับอุปกรณ์ภายนอก

1.4.3 ส่วนติดต่อการ์ดหน่วยความจำ

หน่วยความจำข้อมูลและระบบปฏิบัติการของบอร์ค Raspberry Pi 3 จะถูกเก็บไว้อยู่ในแผ่น หน่วยความจำmicro SD การ์ค โคยมีช่องเสียบ micro SD การ์คอยู่ที่ใต้บอร์ค



รูปที่ 1–17 แสดงซ็อกเก็ตของ micro SD การ์ดของบอร์ด Raspberry Pi 3 ที่ติดตั้งอยู่ใต้บอร์ด



รูปที่ 1–18 ชิป **BCM**43438 ที่รวม WiFi และบลูทูธ 4.1 ไว้ในตัวเดียวติดตั้งอยู่ด้านล่างของบอร์ด ส่วน สายอากาศจะอยู่ด้านบน

1.4.4 จุดต่อพอร์ต GPIO

ที่ขาดไม่ได้เลยสำหรับบอร์ด Raspberry Pi 2 นั่นคือ พอร์ตอินพุตเอาต์พุตดิจิตอลอเนกประสงค์ หรือ GPIO 40 ขาเพื่อนำไปต่อเข้ากับวงจรอิเล็กทรอนิกส์ต่างๆ มีขาพอร์ตที่รองรับระดับสัญญาณลอจิก 3.3V ให้ใช้งานรวม 26 ขา รองรับ UART, I²C และ SPI แต่ไม่รองรับอินพุตเอาต์พุตอะนาลอกโดยตรง

1.4.5 คู่หูไร้สาย WiFi และ BLE

ความสามารถใหม่ที่สร้างความพอใจให้กับผู้ใช้งานอย่างมากคือ การผนวกชิป BCM43438 ซึ่ง ภายในมีวงจรสื่อสารข้อมูลไร้สายสมัยใหม่ทั้ง WiFi หรือ Wireless LAN และบลูทูธ 4.1 ซึ่งรองรับทั้ง บลูทูธมาตรฐานและแบบพลังงานต่ำ (BLE) อยู่ในตัวเดียวกัน ทำให้บอร์ด RaspberryPi 3 เชื่อมต่อ เครือข่ายและอินเทอร์เน็ตในแบบไร้สายด้วย WiFi ได้เลย โดยไม่ต้องใช้ USB WiFi ดองเกิลอีกต่อไป



รูปที่ 1–19 อะแดปเตอร์ +5V 2.5A ที่ทางผู้ผลิตคือ Raspberry Pi Foundation แนะนำให้ใช้เป็นแหล่งจ่าย ไฟเลี้ยงสำหรับบอร์ด Raspberry Pi 3

1.4.6 ไฟเลี้ยงที่คู่ควร

เนื่องจาก Raspberry Pi 3 ใช้ซีพียูที่มีความ เร็วมากขึ้น และมีการเพิ่ม WiFi กับบลูทูธ ส่งผล ให้มีความต้องการ ใช้พลังงานมากขึ้น โดย Raspberry Pi 3 จะใช้กระแส ไฟฟ้าเพิ่มเป็น 2.5A ที่ไฟเลี้ยง +5V จึงจะทำงานได้เต็มประสิทธิภาพ แต่ถ้าหากงานที่ใช้ไม่ได้เน้นเรื่องการสื่อสาร ข้อมูลไร้สายอาจเลือกใช้อะแดปเตอร์ +5V 2A ได้

1.4.7 ระบบปฏิบัติการใหม่ Raspbian Jessie with Pixel

ในวันที่ Raspberry Pi 3 ออกวางตลาด ทาง RaspberryPi Foundation ได้ทำการแนะนำระบบ ปฏิบัติการเวอร์ชั่นใหม่ให้ได้ใช้งานไปพร้อมกัน นั่นคือ **Raspbian Jessie** โดยดาวน์โหลดได้ที่ *https:/* /www.raspberrypi.org/downloads/raspbian/ โดยมีการปรับปรุงการเข้าถึงหน้าเดสก์ท็อปให้เร็วขึ้น มี หน้าต่างสำหรับตั้งค่าการทำงานต่างๆ ในแบบกราฟิกที่เป็นไดอะล็อกบ็อก ทำให้การตั้งค่าทำได้ง่าย สะดวก และดูร่วมสมัยมากขึ้น

	-				
RASPBIAN					
Raspbian is the Fou it with <u>NOOBS</u> or do	ndation's official suppo wnload the image belo	orted operating system. Y w and follow our <u>installa</u> :	'ou can install <mark>tion guide</mark> .		
Raspbian comes pre and general use. It h	e-installed with plenty o nas Python, Scratch, Sor	of software for education, nic Pi, Java, Mathematica	programming a and more.		
SHA-1: e0eeb96e Note: Raspbian and NOOB	RASPBIAN JESS Image with PIXEL deskt Version: S Release date 21 Kernel version: 4 Release notes: L: Download Torrent 225a10b3bd4b57454317b065 S contain Java SE Platform Pro	SIE WITH PIXEL op based on Debian Jessie eptember 2016 016-09-23 .4 ink Download ZIP 53d3d09d46 soducts, licensed to you under the	SHA-1: 3a34e7b0	RASPBIAN JES Minimal image based Version: Release date: Kernel version: Release notes: Download Torrent Sele6e9042294b2906514	SSIE LITE on Debian Jessie September 2016 2016-09-23 4.4 Link Download ZIP 4748625bea8
Licence Agreement availab Mathematica and the Wolf	ole <u>here</u> . ram Language are included in 1	this release under license and wit	th permission of		
Wolfram Research, Inc. and agree to be bound by the \	d may be used for non-comme Wolfram Raspberry Pi Bundle Li	rcial purposes only. By using this icensee Agreement available <u>her</u>	s software you E		

รูปที่ 1–20 เว็บเพจสำหรับดาวน์โหลดระบบปฏิบัติการ Rasbian Jessie เวอร์ชันล่าสุดสำหรับ Raspberry Pi 3



รูปที่ 1–21 หน้าต่างเดสก์ท็อปของระบบปฏิบัติการ Rasbian Jessie ที่จะพบเมื่อเริ่มใช้งานบอร์ด Raspberry Pi 3

ทั้งหมดนี้คือ ปฐมบทของการแนะนำให้รู้จักกับบอร์ดไมโครคอมพิวเตอร์แผ่นเดี่ยวที่โด่งดัง ที่สุด พัฒนาโปรแกรมได้หลากหลายบนระบบปฏิบัติการ Linux บอร์ดไมโครฯ ที่ใช้ชื่อผลไม้ที่อุดม ไปด้วยสารอาหารที่เป็นประโยชน์ต่อร่างกายและนี่คือบอร์ดสำหรับนักพัฒนา Embedded Computer ยุคใหม่ที่มาพร้อมกับความโดดเด่นทั้งคุณสมบัติและราคาที่ใครๆ ก็เข้าถึงได้



บทที่ 2 <mark>เตรียมการใช้งาน Raspberry Pi 3</mark>



เนื่องจากบอร์ด Raspberry Pi 3 เป็นบอร์ดคอมพิวเตอร์ขนาดเล็กแบบหนึ่ง จึงต้องมีการติดตั้ง หน่วยความจำหลักของระบบ ติดตั้งระบบปฏิบัติการ และเชื่อมต่ออุปกรณ์พื้นฐานเพื่อตั้งค่าการทำงาน

2.1 เตรียมอุปกรณ์และเครื่องมือ

เตรียมอุปกรณ์ดังนี้

1. บอร์ด Raspberry Pi 3

2. แผ่นหน่วยความจำmicroSD การ์ค สำหรับเก็บระบบปฏิบัติการและข้อมูล ควรมีความจุตั้งแต่ 8GB คลาส 4 ขึ้นไป ซึ่งก็คือ SDHC การ์ค แนะนำ Class 10 จะเป็นแบบ FAT32 หรือ NTFS ก็ได้

 สาย microUSB ใช้สำหรับต่อกับแหล่งจ่ายไฟ +5V (แนะนำ 2.5A ขึ้นไป) ที่มีจุดต่อแบบ USB เช่นอะแดปเตอร์ของแท็บเล็ตแอนดรอยด์หรือวินโดวส์, เครื่องจ่ายไฟสำรองหรือ เพาเวอร์แบงก์ (Power Bank) ขนาด 3,000mA ขึ้นไป ไม่ควรใช้แหล่งจ่ายไฟจากพอร์ต USB ของคอมพิวเตอร์ เนื่อง จากมีความสามารถในการจ่ายกระแสไฟฟ้าไม่เพียงพอ

4. อะแคปเตอร์ไฟตรง +5V 2.5A ขึ้นไป ที่มีจุดต่อแบบ USB หรือเครื่องจ่ายไฟสำรองหรือ เพาเวอร์แบงก์ (Power Bank) ขนาด 3000mA ขึ้นไป หรืออะแคปเตอร์ +5V 2.5A ขึ้นไปที่มีหัวต่อ แบบ microUSB หากใช้แบบนี้ สาย microUSB ในข้อ 3. ไม่ต้องใช้

5. สาย HDMI สำหรับต่อกับจอแสดงผล สำหรับผู้ใช้งานที่มีจอแสดงผลแบบ DVi และ VGA จะต้องใช้ตัวแปลงหรือสายแปลงสัญญาณ HDMI เป็น DVi หรือVGA มาต่อแทน

6. คีย์บอร์ดและเมาส์แบบ USB สำหรับควบคุมการทำงานของบอร์ด Raspberry Pi 3

7. สาย LAN (มีหรือไม่มีก็ได้) สำหรับเชื่อมต่อกับระบบเครือข่ายและอินเทอร์เนีต

8. สาย AV ที่มีหัวเสียบค้านหนึ่งเป็นปลั๊กหูฟัง 4 ขั้ว และปลายอีกค้านหนึ่งเป็นปลั๊ก RCA 3 หัว สำหรับสัญญาณเสียงช่องซ้าย, ขวา และสัญญาณวิคีโอ หรือสายหูฟังสเตอริโอ 3.5 มม. ในกรณีที่ไม่ต้อง การใช้สัญญาณวิคีโอที่จุคนี้ (มีหรือไม่มีก็ได้)

9. ลำโพงขยายเสียงที่มีจุดต่อสัญญาณอินพุตเป็นแจ๊ก RCA หรือแจ๊กหูฟังสเตอริโอ 3.5 มม. เพื่อต่อกับแจ๊ก AV ของบอร์ด Raspberry Pi 3 (มีหรือไม่มีก็ได้)

รูปที่ 2-1 แสดงอุปกรณ์ทั้งหมดที่ควรมีสำหรับใช้งานบอร์ด Raspberry Pi 3 และ Model B ทุกรุ่น



รูปที่ 2-1 อุปกรณ์ที่ต้องเตรียมสำหรับการเริ่มต้นใช้งานบอร์ดคอมพิวเตอร์ Raspberry Pi ทุกรุ่น (ในที่นี้เลือกใช้กับ Raspberry Pi 3)

2.2 ติดตั้งระบบปฏิบัติการลงใน microSD การ์ด

(ในกรณีจัคซื้อ Raspberry Pi 3 Starter kit จาก inex ให้ข้ามหัวข้อนี้เพราะ ได้จัดการให้พร้อม ใช้งานแล้ว)

สำหรับ Raspberry Pi 3 ใช้หน่วยความจำ microSD การ์คในการทำงานและติดตั้งระบบปฏิบัติการ ผู้ใช้งานสามารถสลับเปลี่ยน microSD การ์คได้ตามต้องการ โดยติดตั้งระบบปฏิบัติการที่ต่างกันในแต่ละแผ่น

สำหรับไฟล์ติดตั้งของระบบปฏิบัติการจะเรียกกันว่า ไฟล์อิมเมจ (image file) สำหรับบอร์ด Raspberry Pi 3 มีหลายแบบให้เลือกใช้งาน ไม่ว่าจะเป็น Raspbian, Pidora หรือ Raspbmc เป็นต้น สำหรับในหนังสือเล่มนี้เลือกใช้ Raspbian ดาวน์โหลดได้ฟรีจาก*http://www.raspberrypi.org/* downloads เลือกหัวข้อ RASPBIAN จากนั้นทำการแตกไฟล์ .zip ออกมาจะได้เป็นไฟล์อิมเมจ .img

ในการติดตั้งไฟล์อิมเมจลงใน microSD การ์ดจะต้องทำผ่านคอมพิวเตอร์ด้วยการใช้ตัวอ่าน เขียนการ์ดหน่วยความจำอาจเป็นแบบติดตั้งมาตัวคอมพิวเตอร์ หรือแบบต่อภายนอกผ่านพอร์ต USB ก็ได้ หากตัวอ่านเขียนการ์ดหน่วยความจำไม่สามารถใช้งานกับ microSD การ์ด จะต้องใช้ตัวแปลง ขนาดเป็น SD การ์ด ซึ่งปกติจะมาพร้อมกับตัว micro SD การ์ด

สำหรับระบบปฏิบัติการวินโดวส์แนะนำให้ใช้ซอฟต์แวร์ที่ชื่อว่า Win32 Disk Imager โดย ดาวน์โหลดได้จาก*http://www.softpedia.com/get/CD-DVD-Tools/Data-CD-DVD-Burning/Win32-Disk-Imager.shtml* จากนั้นทำการติดตั้งให้เรียบร้อย จะเข้าสู่ขั้นตอนการติดตั้งไฟล์อิมเมจดังนี้

(2.2.1)เสียบ microSD การ์คเข้าที่ตัวอ่านเขียนการ์คหน่วยความจำ ที่คอมพิวเตอร์ให้รันซอฟต์แวร์ Win32 Disk Imager ขึ้นมา สังเกตที่ช่อง Device ว่า เป็นไครฟของ SD การ์คหรือไม่ จากตัวอย่างจะเป็น ไครฟ I ต้องตรวจสอบให้แน่ใจว่าไม่ผิด เพราะถ้าเลือกผิดอาจทำให้ข้อมูลในไครฟนั้นๆ หายไปได้ จากนั้นคลิกที่ปุ่มรูปแฟ้มที่อยู่ในช่อง Image File เพื่อเลือกไฟล์อิมเมจที่ต้องการติดตั้ง

♥♥Win32 Disk Ima - Image File	iger			
Copy MD5 Has	h:			_
Version: 0.9.5	Cancel	Read	Write	Exit

(2.2.2) เลือกไฟล์อิมเมจของระบบปฏิบัติการ Raspbian ทำการตรวจสอบไครฟที่จะติดตั้งไฟล์ อิมเมจอีกครั้ง เพื่อป้องกันความผิดพลาด เมื่อตรวจสอบจนแน่ใจแล้ว ให้กลิกปุ่ม Write เพื่อเริ่มเขียน ข้อมูลของไฟล์อิมเมจลงใน SD การ์ด ในขั้นตอนการเขียนข้อมูลจะใช้เวลาก่อนข้างนาน เพราะข้อมูล มีขนาดใหญ่ ทั้งนี้ยังขึ้นอยู่กับความเร็วในการเขียนข้อมูลของ SD การ์ดด้วย

🍓 Win32 Disk Ima – Imane Eile	ager			
/2015-02-16-raspbia	n-wheezy/2015-	02-16-raspbian	-wheezy.img	
Copy MD5 Has Progress	sh:			
Version: 0.9.5	Cancel	Read	Write	Exit

(2.2.3) เมื่อเสร็จแล้ว ถอดแผ่น microSD การ์ดออกจากตัวอ่านเขียนการ์ดหน่วยความจำ นำไปติดตั้งใช้งานกับบอร์ด Raspberry Pi 3 ต่อไป



2.3 ทดสอบการทำงานเบื้องต้น

มีขั้นตอนโดยสรุปดังนี้

(2.3.1) ต่อสายคีย์บอร์ค, เมาส์, จอภาพ และแหล่งจ่ายไฟให้เรียบร้อย จากนั้นจ่ายไฟให้กับ บอร์ค Raspberry Pi 3 คังรูปที่ 2-2

(2.32) จากนั้นบอร์ค Raspberry Pi 3 จะเริ่มทำงาน แสคงหน้าต่างต้อนรับเข้าสู่ระบบปฏิบัติ การในรูปแบบใหม่ คังรูปที่ 2-3 รอครู่หนึ่งเมื่อบูตเรียบร้อยจะปรากฏหน้าต่างเคสก์ท็อปคังรูปที่ 2-4



รูปที่ 2-2 ต่ออุปกรณ์ต่าง ๆ ที่จำเป็นสำหรับเริ่มต้นใช้งานบอร์ด Raspberry Pi 3

30 • Raspberry Pi 3 : รู้จักและการใช้งานเบื้องตัน



รูปที่ 2-3 หน้าต่างแสดงการบูตระบบของ Raspberry Pi 3



รูปที่ 2-4 หน้าต่าง Desktop ของระบบปฏิบัติการ Raspbian Jessie with Pixel ในโหมดกราฟิก

2.4 การใช้งานกับระบบเครือข่ายและเชื่อมต่ออินเทอร์เน็ต

มีขั้นตอนดังต่อไปนี้

(2.4.1) เมื่อต้องการเชื่อมต่ออินเทอร์เน็ตให้ต่อสาย LAN เข้าที่จุดต่อ LAN ซึ่งเป็นคอนเน็กเตอร์ RJ-45 จะใช้งานได้ทันที โดยไม่ต้องทำการตั้งค่าใดๆ (ทั้งนี้ ผู้ใช้งานต้องมีการเปิดใช้บริการอินเทอร์ เน็ตจากผู้ให้บริการด้วย) ดังรูปที่ 2-5

(2.4.1.2) ที่หน้าต่างเคสก์ท็อปบริเวณกรอบด้านบนขวา จะมีสัญลักษณ์แสดงสถานะการเชื่อม ต่อดังรูปที่ 2-6



รูปที่ 2-5 ต่อสาย LAN เข้ากับบอร์ด Raspberry Pi 3



รูปที่ 2–6 การแสดงสัญลักษณ์เพื่อแจ้งสถานะว่า Raspberry Pi 3 เชื่อมต่อกับเครือข่ายอินเทอร์เน็ตผ่านทาง พอร์ตอีเธอร์เน็ตแล้ว

(รูปซ้าย) แสดงสถานะยังไม่เชื่อมต่อ

(รูปขวา) แสดงสถานะเชื่อมต่อแล้ว

32 • Raspberry Pi 3 : รู้จักแล:การใช้งานเบื้องต้น

2.4.2 การเชื่อมต่ออินเทอร์เน็ตผ่าน Wi-Fi

บนบอร์ค Raspberry Pi 3 ได้ติดตั้งวงจรเชื่อมต่อเครือข่ายไร้สาย Wi-Fi ไว้แล้ว จึงพร้อมใช้งาน มีขั้นตอนการทคสอบคังนี้

(2.4.2.1) คลิกที่สัญลักษณ์แสดงการเชื่อมต่อเครือข่ายดังรูป จากนั้นเลือกชื่อเครือข่าย WiFi ที่ต้องการติดต่อตามต้องการ



(2.4.2.2) ป้อนรหัสผ่านของเครือข่าย WiFi หากมีความต้องการทราบรหัสผ่าน เมื่อป้อนรหัส ผ่านแล้ว คลิกปุ่ม OK



(2.4.2.3) ถ้าหากเชื่อมต่อได้สำเร็จ สัญลักษณ์แสดงสถานะจะเปลี่ยนเป็นจุดกระจายคลื่นดังรูป



(2.4.2.4) เมื่อคลิกที่สัญลักษณ์จุดกระจายคลื่นอีกครั้ง จะเห็นว่า มีเครื่องหมายถูกที่หน้าชื่อของ เครือข่ายหรืออุปกรณ์ WiFi ที่บอร์ด Raspberry Pi 3 กำลังเชื่อมต่อ



(2.4.2.5) เมื่อเชื่อมต่ออินเทอร์เน็ตได้แล้ว ทำการทคสอบด้วยการเปิดเว็บบราวเซอร์ แล้วเข้า เว็บไซต์ตามต้องการ เช่น www.google.com เพื่อตรวจสอบว่า ใช้งานได้จริง



34 • Raspberry Pi 3 : รู้จักและการใช้งานเบื้องต้น

2.5 การตั้งค่า Raspberry Pi เบื้องต้น

การตั้งค่าการทำงานของระบบปฏิบัติการ Rasbian Jessie with Pixel สำหรับ Raspberry Pi 3 สามารถตั้งค่าในโหมดกราฟิกได้ ซึ่งแตกต่างจากในระบบปฏิบัติการรุ่นเดิมที่ต้องกระทำผ่านทาง หน้าต่างบรรทัดคำสั่งหรือคอมมานด์ไลน์เท่านั้น โดยเลือกไปที่ Preferences > Raspberry Pi Configuration



เมื่อคลิกเลือกจะปรากฏหน้าต่างตั้งค่าดังนี้

System	Interfaces	Performance	Localisation	
Filesystem:			Expand Filesy	stem
Password:			Change Passv	vord.
Hostname: raspberrypi				
Boot:		⊙ To Desktop ○ To CLI		
Auto Login:		☑ Login as user 'pi'		
Network at Boo	ot:		Wait for net	work
Underscan:		● Enabled		
Rastrack:			Add to Rastra	ack
		C	ancel	Ж

2.5.1 การเพิ่มหน่วยความจำให้มากที่สุด

(2.5.1.1) เลือกที่แท็บ System ดูหัวข้อ Filesystem คลิกปุ่ม Expand Filesystem

Raspberry	y Pi Configuration 🛛 🗕 🗖 🕨		
System Interfaces	Performance Localisation		
Filesystem:	Expand Filesystem		
Password:	Change Password		
Hostname:	raspberrypi		
Boot:	⊙ To Desktop ○ To CLI		
Auto Login:	☑ Login as user 'pi'		
Network at Boot:	Wait for network		
Underscan:	⊙ Enabled		
Rastrack:	Add to Rastrack		
	Cancel OK		

(2.5.1.2) จากนั้นคลิกปุ่ม OK เพื่อยืนยันการตั้งค่า อย่างไรก็ตาม การตั้งค่าพื้นที่หน่วยความ จำจะไม่เกิดผลขึ้นในทันที จนกว่าจะทำการรีสตาร์ตระบบใหม่



2.5.2 การเปิดใช้งานส่วนเชื่อมต่ออุปกรณ์ภายนอก หรือ Interfaces

Raspberry Pi Configuration 🛛 🗕 🗖				
System	Interfaces	Performance	Localisation	
Camera:		⊖ Enabled	 Disabled 	
SSH:		Enabled	O Disabled	
VNC:		Enabled	○ Disabled	
SPI:		Enabled	 Disabled 	
12C:		Enabled	• Disabled	
Serial:		Enabled	 Disabled 	
1-Wire:		○ Enabled	 Disabled 	
Remote GPIO:		Enabled	 Disabled 	
		Са	Incel OK	

(2.5.2.1) .เลือกที่แท็บ Interfaces

(2.5.2.2) ต้องการเปิดใช้งานรายการใด ทำการคลิกเลือก Enabled หากต้องการปิดให้คลิกเลือก Disabled โดยมีรายละเอียดดังนี้

- 1. Camera : เปิดใช้กล้องที่ต่อผ่านพอร์ต CSI
- 2. SSH : เปิดใช้ความสามารถการรีโมตผ่านช่องทาง SSH
- 3. VNC : เปิดใช้ความสามารถการรีโมตผ่านช่องทาง VNC
- 4. SPI : เปิดใช้ความสามารถการเชื่อมต่อกับอุปกรณ์บัส SPI ของพอร์ต GPIO
- 5. I2C : เปิดใช้ความสามารถการเชื่อมต่อกับอุปกรณ์บัส I²C ของพอร์ต GPIO
- 6. Serial : เปิดใช้ความสามารถการเชื่อมต่ออุปกรณ์อนุกรมด้วย UART ของพอร์ต GPIO
- 7. **1-Wire** : เปิดใช้ความสามารถการเชื่อมต่ออุปกรณ์ระบบบัส 1 สายของพอร์ต GPIO
- 8. Remote GPIO : เปิดใช้ความสามารถในการควบคุมขาพอร์ต GPIO ผ่านการ Remote
| | Raspberry | Pi Configuratio | n – 🗆 🗙 |
|--------------|------------|-----------------|------------------------------|
| System | Interfaces | Performance | Localisation |
| Camera: | | ○ Enabled | Disabled |
| SSH: | | Enabled | O Disabled |
| VNC: | | Enabled | O Disabled |
| SPI: | | • Enabled | O Disabled |
| I2C: | | • Enabled | O Disabled |
| Serial: | | O Enabled | Disabled |
| 1-Wire: | | • Enabled | O Disabled |
| Remote GPIO: | | Enabled | Disabled |
| | | Са | ncel OK |

ในที่นี้นี้จะเปิดใช้งานเท่าที่จำเป็น มีดังนี้ SSH, VNC, SPI, I2C และ 1-Wire

2.5.3 การตั้งค่าเวลา

(2.5.3.1) เลือกที่เมนู Localisation

	Raspberry	Pi Configuratio	on – 🗆 🗙
System	Interfaces	Performance	Localisation
Locale:			Set Locale
Timezone:		a so	Set Timezone
Keyboard:			Set Keyboard
WiFi Country:		(Set WiFi Country
		Ca	ancel OK

(2.5.3.2) คลิกที่ปุ่ม Set Timezone จากนั้นให้ตั้งค่าตามนี้

- 1. Area : Asia
- 2. Location : Bangkok

เวลาจะถูกตั้งให้ตรงก็ต่อเมื่อเชื่อมต่อเครือข่ายอินเทอร์เน็ตได้

	Raspberry	Pi Configuratio	'n	×
System	Interfaces	Performance	Localisa	tion
Locale:			Set Lo	cale
Timezone:			Set Tim	ezone
Keyboard:		Timezone –	□ × ^e)	board
WiFi Country:	Area:	Asia	- 1	Country
	Location:	Bangkok	•	
	C	ancel OI	к	
		Ca	ancel	OK

2.6 การเลือกเปิดหน้าต่าง Terminal

สิ่งหนึ่งที่ควรทราบในการตั้งค่าการทำงานรวมถึงการใช้งาน Raspberry Pi คือ การใช้งานหน้า ต่างเทอร์มินอล (Terminal) การเปิดหน้าต่าง Terminal เพื่อพิมพ์กำสั่งต่างๆ ของระบบปฏิบัติการทำ ได้ 2 ทางกือ

1. คลิกเปิดที่ไอคอน Terminal ที่อยู่ในแถบเมนูด้านบนของจอภาพดังรูป





2. เลือกไปที่ Accessories > Terminal ดังรูป

เมื่อเลือกเปิดหน้าต่าง Terminal แล้ว รอสักครู่ หน้าต่าง Terminal จะปรากฏขึ้นพร้อมกับแสดง คอมมานด์พร้อมพ์ pi@raspberrypi:~\$ จากนั้นผู้ใช้งานสามารถพิมพ์คำสั่งต่างๆ ลงได้ทันที



2.7 การตั้งค่าคีย์บอร์ดและเลือกภาษาผ่านทางหน้าต่าง Configuration ในโหมดกราฟิก

หากผู้ใช้งาน Raspberry Pi เมื่อได้ติดตั้งระบบปฏิบัติการตัวใหม่ Raspbian Jessie with Pixel จากทาง Raspberrypi.org การใช้งานคีย์บอร์ดยังไม่สมบรูณ์ดี เนื่องจากคีย์บอร์ดที่ต่อใช้งานจริงอาจ มีการวางตำแหน่งคีย์หรือเลย์เอาต์ (layout) ไม่ตรงกันกับที่กำหนดในระบบปฏิบัติการ มีขั้นตอนการ ทดสอบและตั้งก่าดังนี้

(2.7.1) ทดสอบคีย์บอร์ดที่ต่อใช้งาน โดยเข้าไปที่ Preferences >Mouse and Keyboard Settings



(2.7.2) เลือกแท็ป Keyboard จากนั้นกคคีย์ # ถ้าขึ้นอักขระ £ แสดงว่ายังตั้งค่า Keyboard Layout ไม่ถูกต้อง



(2.7.3) คลิกที่ปุ่ม Keyboard Layout หน้าต่าง Keyboard Layout จะปรากฏขึ้นมา ในกรอบ Country ให้เลือก United States แล้วทคสอบกคคีย์ # ในช่อง Test here to test your keyboard ตรวจสอบว่า แสดงผลตรงกันหรือไม่

Spanish (Latin America Cherokee	Country Taiwan, Province of Chi	Variant
Spanish (Latin America Cherokee	Taiwan, Province of Chi	Spanish (Latin Amorica
Cherokee		opanion (Latin America
	Tajikistan	Cherokee
English (US, with euro c	Tanzania, United Repul	English (US, with euro
English (US, internation	Thailand	English (US, internation
English (US, alternative	Turkey	English (US, alternative
English (Colemak)	Turkmenistan	English (Colemak)
English (Dvorak)	Ukraine	English (Dvorak)
English (Dvorak, interna	United Kingdom	English (Dvorak, intern
English (Dvorak alterna	United States	English (Dvorak alterna
English (left handed Dv	Uzbekistan	English (left handed Dy
English (right handed D	Viet Nam	English (right handed [
d	Type here to test your keyb	oard #
	English (US, internation English (US, alternative English (Colemak) English (Dvorak) English (Dvorak, interna English (Dvorak alterna English (left handed Dv English (right handed D	English (US, internation Thailand English (US, alternative Turkey English (Colemak) Turkmenistan English (Dvorak) Ukraine English (Dvorak, interna United Kingdom English (Dvorak alterna United States English (left handed Dv Uzbekistan English (right handed D Viet Nam Marcel OK

(2.7.4) หากต้องการเปลี่ยนเป็นภาษาไทยให้เลือก Thailand จากนั้นทดสอบด้วยการพิมพ์ข้อ ความลงในช่องทดสอบ จากนั้นคลิกปุ่ม OK เพื่อบันทึก



รูปที่ 2-7 แสดงตำแหน่งของหน้าสัมผัสของปลั๊ก AV ที่ใช้งานกับแจ๊ก AV ของ Raspberry Pi

2.8 การตั้งค่าเพื่อเลือกการแสดงผลผ่าน HDMI หรือแจ๊ก AV

Raspberry Pi 3 รองรับการต่อจอแสดงผลหลัก 2 แบบคือ ต่อกับจอแสดงผลแบบ HDMI และ ต่อกับแจ๊ก AV ในการใช้งานผ่านพอร์ต HDMI ผู้ใช้งานสามารถใช้งานได้โดยไม่ต้องตั้งค่าใดๆ แต่ ถ้าหากต้องการให้ Raspberry Pi แสดงภาพผ่านแจ๊ก AV มีขั้นตอนที่ต้องดำเนินการดังนี้

(2.8.1) ตรวจสอบปลั๊ก AV ที่นำมาต่อใช้งาน จะต้องมีการจัดขาของหน้าสัมผัสดังรูปที่ 2-7

(2.8.2) เปิดหน้าต่าง Terminal พิมพ์คำสั่ง

sudo nano /boot/config.txt จากนั้นให้แก้ไขไฟล์ โดยพิมพ์เครื่องหมาย # ไว้ข้างหน้าข้อความ ดังรูป

uncomment if hdmi display is not detected and composite is being output #hdmi_force_hotplug=1

สั่งเปิดใช้ช่องสัญญาณภาพออกทางแจ๊ก AV ด้วยการเอาเครื่องหมาย **#** หน้าบรรทัด sdtv_mode=2 ออก

```
# uncomment for composite PAL
sdtv_mode=2
```

(2.8.3) บันทึกการแก้ไขไฟล์ *โดยกดคีย์ Ctrl กับ X และ Y แล้วตามด้วย Enter* จากนั้นทำการ รีบูตระบบใหม่อีกครั้งด้วยคำสั่ง sudo reboot

2.9 การตั้งค่าเลือกการขับเสียงผ่านแจ๊ก AV

Raspberry Pi 3 มีช่องทางออกของสัญญาณเสียง 2 ช่อง คือ พอร์ต HDMI และแจ๊ก AV 3.5 มม. การตั้งค่าเพื่อเลือกช่องทางออกของสัญญาณเสียงมีขั้นตอนดังนี้

คลิกเมาส์ปุ่มขวาที่ไอคอนรูปลำโพงของแถบเมนูที่ด้านบนของหน้าต่างเดสก์ท็อป

- ✓ 2 * 1
 % 13:03 ▲
 ✓ Analog
 HDMI
- เลือกหัวข้อ Analog หากต้องการให้สัญญาณเสียงถูกขับออกแจ๊ก AV

● เลือกหัวข้อ HDMI หากต้องการให้สัญญาณเสียงถูกขับออกทางพอร์ต HDMI



2.10 ข้อควรทราบเกี่ยวกับความจุของ microSD การ์ด หลังจากติดตั้งระบบ ปฏิบัติการ

หลังจากติดตั้งระบบปฏิบัติกาารบนบอร์ด Raspberry Pi 3 จะมีการแบ่งพื้นที่ของ microSD การ์ด ออกเป็นหลายส่วน หากนำ microSD การ์ด ไปใช้งานปกติเหมือนเดิมหรือนำ ไปตรวจสอบพื้นที่ว่างด้วย Windows Explorer บนระบบปฏิบัติการวินโดวส์ จะพบว่า ความจุของ microSD การ์ดลดลงมาก ทั้งนี้ เนื่องจากระบบปฏิบัติการณ์วินโดวส์ไม่สามารถมองเห็นพื้นที่ดังกล่าวได้

สิ่งที่เกิดขึ้นจึงไม่ใช่เรื่องผิดปกติอย่างใด เมื่อนำกลับมาทำงานบนระบบปฏิบัติการ Raspbian บนตัวบอร์ด Raspberry Pi 3 จะเห็นพื้นที่ของ microSD การ์ดทั้งหมดเองตามปกติ 44 • Raspberry Pi 3 : รู้จักและการใช้งานเบื้องต้น

2.11 การเปลี่ยนรหัสผ่านผู้ใช้งาน

เป็นที่ทราบกันอยู่แล้วว่า เมื่อเริ่มต้นใช้งานบอร์ด Raspberry Pi นั้น จะมีการกำหนดค่าเริ่ม ต้นของชื่อผู้ใช้คือ pi และรหัสผ่านคือ raspberry

หากมีการนำ Raspberry Pi ไปใช้ในการผลิตเป็นสินค้า ถ้าไม่เปลี่ยนรหัสผ่าน น่าจะไม่ปลอดภัย เพราะอาจมีผู้ไม่ประสงค์ดีแอบเข้ามาทำอะไรกับ Raspberry Pi ได้ ดังนั้นทางที่ดีจึงควรเปลี่ยนรหัส ผ่านเพื่อความปลอดภัย

การเปลี่ยนรหัสผ่านทำได้ดังนี้

(2.11.1)เปิดหน้าต่าง Raspberry Pi Configuration โดยเลือกไปที่ Preferences > Raspberry Pi Configuration

(2.11.2) เลือกที่แท็ป System คลิกที่ปุ่ม Change Password

Raspberr	y Pi Configuratio	on _ ≡ ×	
System Interfaces	Performance	Localisation	
Filesystem:		Expand Filesystem	
Password:	-	Change Password	
Hostname:	raspberrypi		
Boot:	⊙ To Desktop ○ To CLI		
Auto Login:		🖌 Login as user 'pi'	
Network at Boot:		Wait for network	
Splash Screen:	 Enabled 	O Disabled	
Underscan:	O Enabled	• Disabled	
	Ci	ancel OK	

(2.11.3) ใส่รหัสผ่านเดิม แล้วตามด้วยรหัสผ่านใหม่ที่ต้องการ 2 ครั้ง จากนั้นคลิกปุ่ม **OK**

Change F	Password 🗕 🗖 🗙
Current password:	
Enter new password:	
Confirm new password:	
	Cancel OK

(2.11.4) จะปรากฏกล่องข้อความแจ้งว่า **รหัสผ่านถูกเปลี่ยนแปลงแล้ว** คลิกปุ่ม **OK** เพื่อตอบรับ

2.12 การใช้งานบลูทูธของ Raspberry Pi 3 เบื้องต้น

ตัวอย่างการใช้งานบลูทูธของ Raspberry Pi 3 เป็นการใช้งานกับเมาส์บลูทูธของ Apple มีขั้นตอน ดังนี้

(2.12.1) เปิดสวิตช์ที่ตัวเมาส์บลูทูธ จากนั้นไปที่ไอคอนบลูทูธ คลิกที่ไอคอน เลือกรายการ Add Device



(2.12.2) รอสักครู่ จะเห็นอุปกรณ์บลูทูธที่มองเห็นและติดต่อได้ ในที่นี้คือ เมาส์บลูทูธ



(2.12.2) คลิกที่รายการอุปกรณ์ที่ต้องการเชื่อมต่อ จากนั้นคลิกที่ปุ่ม **Pair**



(2.12.3) รอจนกว่า อุปกรณ์เชื่อมต่อกันได้ โดยแสดงข้อความว่า Connected successfully เมาส์ บลูทูฐพร้อมใช้งานทันที ทคลองใช้งานโดยเลื่อนเคอร์เซอร์ของเมาส์ มายังปุ่ม OK แล้วคลิกกค



2.13 การเลือกให้ Raspberry Pi ทำการบูตเพื่อเข้าโหมด Command Line

Raspberry Pi 3 สามารถกำหนดให้บูตระบบได้ทั้งในโหมดบรรทัดกำสั่งหรือ command line และโหมดกราฟิก ก่าตั้งต้นของระบบปฏบัติการที่ติดตั้งมาแต่แรกคือ โหมดกราฟิกที่ไม่ต้องใส่รหัสผ่าน หากต้องการให้บูตระบบเพื่อเข้าโหมดบรรทัดกำสั่งหรือ command line ดำเนินการตามขั้นตอนต่อไปนี้

(2.13.1) เข้าไปที่ Preferences > Raspberry Pi Configuration

(2.13.2) เลือกที่แท็บ System สังเกตที่หัวข้อ Boot ให้เลือก To CLI จะเป็นการบูตเข้าสู่โหมด บรรทัดคำสั่งหรือ command line

	Raspberry	Pi Configuratio	on 🗕 🗆 🗙
System	Interfaces	Performance	Localisation
Filesystem:			Expand Filesystem
Password:			Change Password
Hostname:		raspberrypi	
Boot:	-	O To Desk	top 💿 To CLI
Auto Login:		-	🗆 Login as user 'pi'
Network at Bo	ot:		Wait for network
Splash Screen	1	C Enabled	Disabled
Underscan:		© Enabled	 Disabled
		С	ancel OK

(2.13.3) ที่รายการ Auto Login เป็นการกำหนดให้เข้าสู่ระบบปฏิบัติการโดยไม่ต้องใส่ Username และ Password ในที่นี้จะเลือกเข้าโหมด command line และต้องใส่ Username และ Password ด้วย ดังนั้นจึงต้องตั้งก่าต่างๆ ตามที่แสดงในรูปด้านล่าง จากนั้นกลิกปุ่ม OK แล้วทำการรีบูต

เพียงเท่านี้ RaspberryPi 3 สามารถบูตเข้าโหมคบรรทัดคำสั่งหรือ command line ได้แล้ว เมื่อ Raspberry Pi 3 บูตเข้าสู่โหมด command line แล้ว ให้ป้อนรหัสต่างๆ ดังนี้

```
Username : pi
Password : raspberry
```

ถ้ามีการตั้งรหัสผ่านใหม่มาไว้ก่อนล่วงหน้าผู้ใช้งานต้องป้อนรหัสผ่านใหม่ จากนั้นหากต้อง การเข้า โหมดกราฟิก ให้พิมพ์คำสั่ง startx ที่หน้าต่าง Terminal ตัวบอร์ด Raspberry Pi 3 ก็จะเข้าสู่ โหมดกราฟิกเพื่อทำงานต่อไป



บทที่ 3 Raspberry Pi กับการเชื่อมต่อระบบเครือข่าย ไร้สาย WiFi และการควบคุมผ่านคอมพิวเตอร์ จากภายนอก

บอร์ด Raspberry Pi 3 มีจุดต่อระบบเครือข่ายหรือ LAN ผ่านจุดต่ออีเธอร์เน็ตที่เป็นคอนเน็ก เตอร์แบบ RJ-45 และด้วยการใช้ระบบปฏิบัติการ Raspbian ในเวอร์ชันใหม่ๆ ได้บรรจุความสามารถ นี้ไว้เรียบร้อย ทันทีที่บูตระบบเสร็จสิ้น บอร์ด Raspberry Pi 3 จะสามารถติดต่อกับระบบเครือข่าย แบบใช้สายได้ทันที รวมถึงการเชื่อมต่อกับอินเทอร์เน็ตด้วย หากเครือข่ายที่ทำการเชื่อมต่อนั้นเปิดการ ติดต่อเครือข่ายอินเทอร์เน็ตไว้แล้ว

ในการใช้งานสมัยใหม่ การเชื่อมต่อเครือข่ายมุ่งสู่การเชื่อมต่อแบบไร้สายผ่าน WiFi (ถ้าเรียก ให้ถูกต้องคือ WLAN หรือ Wireless Local Area Network ตามมาตรฐาน IEEE802.11– คำว่า WiFi เป็นชื่อเครื่องหมายการค้าของกลุ่ม Wi-Fi Alliance อันเป็นการรวมตัวกันของบริษัทผู้ผลิตสินค้าและ บริการที่เกี่ยวข้องกับอุปกรณ์ระบบเครือข่ายไร้สายหรือ WLAN อาทิ Nokia, Intersil, 3COM, Motorola ภายใต้การสนับสนุนจาก Apple, Samsung, LG, Sony, Microsoft, Texas Instrument และ T-mobile) บอร์ด Raspberry Pi 3 มีวงจรหรือโมดูล WiFi (WLAN) ติดตั้งอยู่ในตัว จึงไม่ต้องใช้อุปกรณ์ที่เรียก ว่า **USB WiFi ดองเกิล** ต่อเข้าทางพอร์ต USB เพียงทำการตั้งค่าเล็กน้อย ก็จะทำให้บอร์ค Raspberyy Pi 3 เชื่อมต่อระบบเครือข่ายในแบบไร้สายได้ หรือใช้งานอินเทอร์เน็ตผ่าน WiFi ได้ด้วยเช่นกัน

3.1 การตั้งค่าการทำงานของ Raspberry Pi 3 เพื่อเชื่อมต่อ WiFi ผ่าน Terminal

ในระบบปฏิบัติการ Raspbian เวอร์ชันล่าสุดมีการตั้งค่าเพื่อให้วงจร WiFi บนบอร์ด Raspberry Pi 3 ติดต่อกับเครือข่าย WiFi ได้ค่อนข้างง่าย *สิ่งที่จะต้องตั้งค่าจึงมีเพียงการกำหนดเครือข่ายที่ต้อง* การเชื่อมต่อเท่านั้น

สำหรับการใช้งาน Raspbian ในโหมดกราฟิกหรือ GUI มีโปรแกรมสำหรับเชื่อมต่อ WiFi ไว้ อยู่แล้ว แต่การใช้งานผ่านเทอร์มินอล (Terminal) หรือสั่งงานด้วยกอมมานด์ไลน์ (Command Line) มีวิธีการเชื่อมต่อที่ต่างกันไปดังนี้

(3.1.1) ตรวจสอบการเชื่อมต่อบอร์ค Raspberry Pi 3 กับอุปกรณ์ต่างๆ ทั้งสายไฟเลี้ยง, สาย ต่อจอภาพ HDMI, คีย์บอร์ค, เมาส์ และ SD การ์คที่ติคตั้งระบบปฏิบัติการ Raspbian เวอร์ชันล่าสุค คังรูปที่ 3-1



รูปที่3-1 การเชื่อมต่ออุปกรณ์ต่างเข้ากับบอร์ด Raspberry Pi 3 เพื่อเชื่อมต่อเครือข่ายแบบไร้สายผ่าน WiFi

(3.1.2) จ่ายไฟแล้วเข้าสู่หน้าต่างเทอร์มินอล แล้วพิมพ์คำสั่ง sudo nano /etc/network/ interfaces เพื่อตรวจสอบการกำหนดค่า Network Connection ของระบบ

```
pi@raspberrypi ~ $ sudo nano /etc/network/interfaces
```

จากกำสั่งข้างต้น **nano** ในที่นี้ก็คือโปรแกรมเท็กซ์เอคิเตอร์ตัวหนึ่ง ไฟล์ที่เปิดคือ ไฟล์ ระบบสำหรับตั้งก่าเกรือข่าย เมื่อรันกำสั่ง โปรแกรมจะแสดงข้อมูลดังรูป



wlan0 คือ Wireless LAN หรือ WiFi ที่บอร์ด Raspberry Pi 3 ใช้เชื่อมต่อกับ เครือข่ายและอินเทอร์เน็ต (3.1.3) ถ้าต้องการปิดไฟล์ดังกล่าว ให้กด Ctrl และ X ในกรณีที่มีการแก้ไขไฟล์ข้อมูล โปรแกรม จะสอบถามว่า ต้องการให้บันทึกด้วยหรือไม่ ให้กดปุ่ม Y เพื่อทำการบันทึกไฟล์



(3.1.4) จากนั้นโปรแกรมจะถามชื่อไฟล์ที่ต้องการบันทึก ให้กคปุ่ม Enter เพื่อบันทึกทับไฟล์เคิม

File Name to Write: /etc/network/interfaces ^G Get Help M-D DOS Format M-A Append M-B Backup File ^C Cancel M-M Mac Format M-P Prepend

(3.1.5) ถำดับต่อไปเป็นการตั้งค่า WiFi ที่ต้องการเชื่อมต่อ ให้พิมพ์กำสั่ง sudo nano /etc/ wpa_supplicant/wpa_supplicant.conf

```
pi@raspberrypi ~ $ sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

(3.1.6) ทำการตั้งก่า WiFi ที่จะเชื่อมต่อ โดยกำหนดชื่อ WiFi ที่ต้องการเชื่อมต่อใน ssid และ กำหนดรหัสผ่านที่บรรทัด psk ส่วนพารามิเตอร์อื่นๆ ให้กำหนดตามรูปต่อไปนี้



(3.1.7) เมื่อกำหนดเสร็จแล้ว บันทึกแล้วปิดไฟล์ จากนั้นรีบูตระบบด้วยคำสั่ง sudo reboot



(3.1.8) เมื่อรีบูตระบบเสร็จสิ้น พิมพ์คำสั่ง ifconfig เพื่อตรวจสอบสถานะของการเชื่อมต่อ เครือข่ายหรือ Network Connection

pi@raspbe	rrypi ~ \$ ifconfig
eth0	Link encap:Ethernet HWaddr b8:27:eb:ee:d2:29 UP BROADCAST MULTICAST MTU:1500 Metric:1 RX packets:0 errors:0 dropped:0 overruns:0 frame:0 TX packets:0 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000 RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
10	Link encap:Local Loopback inet addr:127.0.0.1 Mask:255.0.0.0 UP LOOPBACK RUNNING MTU:65536 Metric:1 RX packets:8 errors:0 dropped:0 overruns:0 frame:0 TX packets:8 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:0 RX bytes:1104 (1.0 KiB) TX bytes:1104 (1.0 KiB)
wlan0	Link encap:Ethernet HWaddr 48:02:2a:cd:49:e1 inet addr:192.168.1.37 Bcast:192.168.1.255 Mask:255.255.255.0 UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1 RX packets:8577 errors:0 dropped:91 overruns:0 frame:0 TX packets:852 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000 RX bytes:865284 (845.0 KiB) TX bytes:148687 (145.2 KiB)

ให้สังเกตที่ **wlan0** จะเห็นว่า บรรทัดที่ 2 ตรงคำว่า **inet addr** มีเลข IP แอดเดรสอยู่ นั่นหมายความว่า เชื่อมต่อ WiFi ได้แล้ว และ IP แอดเดรสนั้นก็คือ IP แอดเดรสของบอร์ด Raspberry Pi

(3.1.9) ในกรณีที่เชื่อมต่อไม่ได้ จะไม่มี inet addr แสดงให้เห็น ทำการตรวจสอบการตั้ง ก่า ssid และรหัสผ่านจากขั้นตอน (3.1.5) ใหม่อีกครั้ง รวมถึงตรวจสอบด้วยว่า ตัวปล่อยสัญญาณ WiFi หรือเราเตอร์อยู่ในระยะที่เชื่อมต่อกันได้หรือไม่

(3.1.10) การตรวจสอบการเชื่อมต่ออินเทอร์เน็ตทำได้ง่ายๆ ด้วยการ Ping สัญญาณไปที่ Google โดยพิมพ์กำสั่ง ping www.google.com

```
pi@raspberrypi ~ $ ping www.google.com
PING www.google.com (173.194.120.147) 56(84) bytes of data.
64 bytes from kul06s08-in-f19.1e100.net (173.194.120.147): icmp_req=1 ttl=52 time=34.9 ms
64 bytes from kul06s08-in-f19.1e100.net (173.194.120.147): icmp_req=2 ttl=52 time=32.4 ms
```

จะเห็นข้อความที่ส่งกลับมารวมไปถึงระยะเวลาของการ Ping สัญญาณด้วย นั่นหมายความว่า บอร์ด Raspberry Pi 3 เชื่อมต่อกับเครือข่ายอินเทอร์เน็ตได้แล้ว

3.2 การควบคุม Raspberry Pi ผ่าน Secure Shell (SSH)

ถึงแม้ว่าบอร์คคอมพิวเตอร์ Raspberry Pi ทุกรุ่นจะรองรับการเชื่อมต่อกับอุปกรณ์ USB อย่าง คีย์บอร์ค เมาส์ และจอแสดงผลได้อยู่แล้ว แต่ในการใช้งานจริงก็ไม่จำเป็นต้องต่อกับอุปกรณ์เหล่านี้ เสมอไป เพื่อลดขนาดของของระบบลง แต่ยังคงทำงานหรือใช้งานได้

โดยปกติผู้ใช้งานบอร์ด Raspberry Pi มักใช้งานในแบบเดียวกับใช้คอมพิวเตอร์ นั่นคือ ต่อเมาส์ ก็ย์บอร์ด และจอภาพเข้ากับบอร์ด Raspberry Pi จากนั้นทำการเขียนโปรแกรมหรือทำงานตามที่ต้องการ สำหรับในบทความนี้ขอนำเสนอการใช้งาน Raspberry Pi ในอีกลักษณะหนึ่ง โดยใช้คอมพิวเตอร์ทำการ รีโมตหรือเชื่อมต่อกับบอร์ด Raspberry Pi จากภายนอกผ่านระบบเครือข่ายแล้วทำการรันโปรแกรมหรือ ใช้งานบอร์ด Raspberry Pi ผ่านการควบคุมหรือกำหนดการใช้งานจากคอมพิวเตอร์อีกตัวหนึ่ง ทำให้ไม่ ต้องใช้จอภาพ, เมาส์ และคีย์บอร์ดเพื่อต่อใช้งาน บอร์ด Raspberry Pi จึงทำงานได้อย่างกล่องตัว ไม่ต้อง มีอุปกรณ์ต่อพ่วงมากมาย การติดต่อในลักษณะดังกล่าวนี้ใช้กระบวนการที่ชื่อว่า SSH

3.2.1 SSH คืออะไร

เดิมทีการควบคุมคอมพิวเตอร์ระยะ ใกลผ่านทางระบบเครือข่ายหรือเน็ตเวิร์ก มักใช้โปรแกรม telnet, rlogin หรือ rshแต่มีปัญหาจากแฮกเกอร์ที่คอยดักจับข้อมูลในเน็ตเวิร์กระหว่างคอมพิวเตอร์ทั้งสอง เครื่อง ทั้งนี้เป็นเพราะข้อมูลจากโปรแกรมที่กล่าวมาข้างต้นทั้งหมดไม่มีการเข้ารหัส จนกระทั่งในปี ค.ศ. 1955 Tatu Ylonen ได้พัฒนาโปร โตคอลชื่อ SSH ย่อมาจาก Secure Shell เป็นเน็ตเวิร์กโปร โตคอลใช้ ควบคุมคอมพิวเตอร์ระยะ ใกลที่มีการเข้ารหัสด้วย public key กล่าวคือ SSH จะเข้ารหัสข้อมูล ส่งไป ทางเน็ตเวิร์ก เมื่อถึงปลายทางก็จะถอดรหัสข้อมูลให้เอง ทำให้การดักจับข้อมูลทำได้ยากขึ้น

จึงอาจกล่าวได้ง่ายๆ ว่า SSH เป็นการใช้คอมพิวเตอร์สักเครื่องที่อยู่ในวงเครือข่ายหรือวง LAN เดียวกับบอร์ด Raspberry Pi ต่อเข้าไปควบคุมสั่งงานนั่นเอง ด้วยวิธีนี้ไม่จำเป็นต้องต่อบอร์ด Raspberry Pi กับจอแสดงผล, คีย์บอร์ด และเมาส์แต่อย่างใด เพราะสั่งงานจากคอมพิวเตอร์ที่เชื่อม ต่อได้ทันที โดยมีเงื่อนไขว่า ทั้งคอมพิวเตอร์และบอร์ค Raspberry Pi จะต้องอยู่ในเครือข่ายเดียวกัน หรือในวง LAN เดียวกัน



รูปที่ 3-2 การเชื่อมต่อบอร์ด Raspberry Pi เพื่อใช้งานด้วยวิธีการ SSH ซึ่งทำได้ทั้งแบบต่อผ่านเราเตอร์ (รูป A) หรือต่อตรงในแบบ peer-to-peer (รูป B)

3.2.2 การเชื่อมต่อทางฮาร์ดแวร์

ในการใช้งานบอร์ค Raspberry Pi ด้วยวิธีการ SSH หรือรี โมตเข้าไปทำงานนั้น ให้ต่อคอมพิวเตอร์ ที่ใช้งานและบอร์ค Raspberry Pi เข้ากับเราเตอร์ผ่านทางจุคต่อ LAN คังรูปที่ 3-2

3.3.3 ติดต่อแบบ SSH ด้วยโปรแกรม Tera Term

สำหรับโปรแกรมบนคอมพิวเตอร์ที่ใช้ควบคุมหรือติดต่อกับบอร์ด Raspberry Pi ผ่าน SSH นั้น หลายตัว ในที่นี้ใช้โปรแกรม **Tera Term** ดาวน์โหลดได้ฟรีจาก *http://en.sourceforge.jp/projects/ttssh2/ releases/*

ที่บอร์ค Raspberry Pi นั้น ระบบปฏิบัติการ Raspbian รุ่นใหม่ๆ ได้กำหนดค่าและเปิดใช้งาน SSH ไว้แล้ว จึงเชื่อมต่อผ่าน SSH ได้ทันที

เมื่อเปิดโปรแกรม Tera Term ขึ้นมาให้ใส่ IP แอดเดรสของ Raspberry Pi (ตรวจสอบได้ด้วย คำสั่ง ifconfig) และเลือก Service เป็น SSH ดังรูปที่ 3-3 คลิกปุ่ม OK

เมื่อเชื่อมต่อกับ Raspberry Pi ได้แล้ว จะต้องทำการล็อกอินก่อน โดยใช้ Username และ Password เดียวกันกับที่ใช้ในตอนบูตระบบ จากนั้นคลิกปุ่ม OK เพื่อทำการเชื่อมต่อ ดังรูปที่ 3-4

หลังจากนั้น หน้าต่าง VT มอนิเตอร์จะปรากฏขึ้นมา ดังรูปที่ 3-5 อันเป็นหน้าจอการทำงานใน ปัจจุบันของบอร์ค Raspberr Pi 2 นับจากนี้ผู้ใช้งานสามารถควบคุมหรือใช้งานบอร์ค Raspberry Pi ผ่าน ทางหน้าต่างนี้ได้แล้ว

-	Tera Term: New o	connection	×
● TCP/IP	Host: 192.168. ✓ History Service: ○ Telnet ● SSH ○ Other	TCP port#: 22 SSH version: SSH2 Protocol: UNSPEC	× ×
Serial	Port:		~
	OK Cance	I Help	

รูปที่ 3–3 หน้าต่างของโปรแกรม Tera Term ที่ใช้ในการเชื่อมต่อเพื่อควบคุมการทำงานของบอร์ด Raspberry Pi ด้วยวิธีการ SSH

54 • Raspberry Pi 3 : รู้จักและการใช้งานเบื้องตัน

	SSH Authentication 🛛 🗕 🗖 🗙			
Logging in to 192.168	3.1.37			
Authentication require	ed.			
User name:	pi			
Passphrase:	•••••			
	Remember password in memory			
	Forward agent			
Use plain passw	ord to log in			
Use RSA/DSA/E	CDSA/ED25519 key to log in Private key file:			
Use rhosts to lo	g in (SSH1) Local user name:			
	Host private key file:			
O Use challenge/r	esponse to log in(keyboard-interactive)			
O Use Pageant to log in				
	OK Disconnect			

รูปที่ 3-4 ล็อกอินเข้าสู่ระบบ



รูปที่ 3–5 หน้าต่าง VT มอนิเตอร์ของโปรแกรม Tera Term ที่ใช้เป็นจอแสดงผลของบอร์ด Raspberry Pi หลังจากเชื่อมต่อกับคอมพิวเตอร์เพื่อแลกเปลี่ยนข้อมูลด้วยวิธีการ SSH

3.4 การควบคุมบอร์ด Raspberry Pi 3 จากระยะไกลด้วยคอมพิวเตอร์

ปกติแล้วการ ใช้งานบอร์ค Raspberry Pi จะเน้นไปที่การ ใช้งานแบบ โคยลำพังเหมือนกับ คอมพิวเตอร์เครื่องหนึ่งที่ต่อคีย์บอร์ค,เมาส์, จอภาพ และแหล่งจ่ายไฟ หากต้องการเชื่อมต่อกับเครือข่าย อินเทอร์เน็ต ก็ต่อสาย LAN เข้าที่จุคต่อพอร์ตอีเธอร์เน็ตหรือใช้ WiFi ที่มาพร้อมกับ Raspberry Pi 3 ก็ได้

ยังมีการใช้งานบอร์ด Raspberry Pi 3 ในอีกรูปแบบหนึ่ง โดยไม่ต้องต่อเมาส์, คีย์บอร์ด และ จอภาพ เพียงต่อกับคอมพิวเตอร์อีกตัวหนึ่งด้วยสาย LAN ผ่านจุดต่อพอร์ตอีเธอร์เน็ต เรียกวิธีการใน ถักษณะนี้ว่า การเชื่อมต่อระยะไกล หรือ Remote Desktop Connection (RDC) หรือเรียกสั้นๆ ว่า การรึ โมต (remote) ด้วยวิธีการนี้จะทำให้ผู้ใช้งานสามารถสั่งงานหรือใช้งานบอร์ด Raspberry Pi ผ่านทาง หน้าจอของคอมพิวเตอร์ ได้ จึงไม่ต้องต่อเมาส์, คีย์บอร์ด และจอภาพให้กับบอร์ด Raspberry Pi แต่ ประการใด เพราะจะใช้อุปกรณ์เหล่านั้นจากคอมพิวเตอร์ที่ทำการเชื่อมต่อ จึงทำให้ผู้ใช้งานสามารถ สร้างระบบควบคุมที่ติดต่อหรือตรวจสอบการทำงานกับคอมพิวเตอร์ที่ใช้ควบคุมในระยะไกลได้

ขั้นตอนในการรี โมตบอร์ค Raspberry Pi ด้วยคอมพิวเตอร์อีกเครื่องหนึ่งทำได้ไม่ยาก โดยใช้ โปรแกรมที่ชื่อว่า **xrdp (x-windows remote desktop protocol)**

3.4.1 ติดตั้งโปรแกรม xrdp

มีขั้นตอนโดยสรุปดังนี้

(1) เชื่อมต่อเมาส์, คีย์บอร์ด และจอภาพให้กับ Raspberry Pi 3

(2) เชื่อมต่อกับบอร์ค Rapberry Pi 2 กับเครือข่ายอินเทอร์เนีต

(3)เปิดหน้าต่างเทอร์มินอลในโหมดคอมมานด์ไลน์หรือเข้าสู่โหมดกราฟิกด้วยการรันคำสั่ง startx ก็ได้ รอจนกระทั่งระบบพร้อมทำงาน แล้วเปิดหน้าต่าง LX Terminal

(4) พิมพ์กำสั่งต่อไปนี้เพื่อทำการดาวน์โหลดและติดตั้งโปรแกรม

sudo apt-get install xrdp

3.4.2 ติดตั้งและใช้งานโปรแกรม Advanced IP Scanner

เนื่องจากการเชื่อมต่อแบบ RDC ต้องใช้ช่องทางผ่านพอร์ตอีเธอร์เน็ต สิ่งหนึ่งที่ต้องทราบคือ *หมายเลข IP แอดเดรสของบอร์ด Raspberry Pi 3 ที่นำมาเชื่อมต่อ* ซึ่งหมายเลข IP แอคเครสนี้ คอมพิวเตอร์ที่เชื่อมต่อด้วยจะเป็นตัวกำหนดให้ การตรวจสอบหมายเลข IP แอดเดรส จะต้องใช้เครื่อง มือช่วย ในที่นี้ขอแนะนำโปรแกรม Advanced IP Scanner

56 • Raspberry Pi 3 : รู้จักและการใช้งานเบื้องตัน

Advanced IP Scanner เป็นซอฟต์แวร์ที่ช่วยให้ผู้ใช้ก้นหาและตรวจสอบหมายเลข IP แอดเดรส ของอุปกรณ์ที่เชื่อมต่ออยู่บนเครือข่ายเดียวกัน โดยซอฟท์แวร์นี้รองรับการติดต่อกับ โปร โตคอลแบบต่างๆ มากมาย อาทิ HTTP, HTTPS, FTP รวมถึงการก้นหาเครือข่ายเพื่อรับทราบข้อมูลเพิ่มเติมเกี่ยวกับอุปกรณ์ ที่เชื่อมต่อกัน ได้แก่ ชื่อของกอมพิวเตอร์หรืออุปกรณ์ และ MAC แอดเดรส ทำการ ดาวน์โหลด โปรแกรม มาใช้งานได้โดยไม่มีก่าใช้จ่ายจาก *http://www.advanced-ip-scanner.com* เมื่อดาวน์โหลดมาแล้ว ติดตั้ง ให้เรียบร้อย ซึ่งมีขั้นตอนเหมือนกับการติดตั้งโปรแกรมประยุกต์อื่นๆ บนระบบปฏิบัติการวินโดวส์

(1) เชื่อมต่อและค้นหาขอบเขตของเลข IP แอดเดรส

(1.1) เริ่มต้นการค้นหา IP แอคเครสของบอร์ค Raspberry Pi 3 ด้วยการต่อสาย LAN ระหว่างพอร์ตอีเธอร์เน็ตของ Raspberry Pi 3 กับคอมพิวเตอร์

<u>2</u>		Advanced IP	Scanner		×
ไฟล์ ปฏิบัติกา	ร การตั้งค่า ดู ช่วยเหลือ				
▶ สแกน					
192.168.137.1	- 192.168.137.254		<i>ตัวอย่าง</i> : 1	192. 168. 0. 1 - 192. 168. 0. 100, 192. 168. 0. 20	o 🗸
รายการผลลัพธ์	รายการโปรด				
สถานะ	ชื่อ	IP	ผู้ผลิต	MAC แอ็ดเดรส	
0 มีชีวิต, 0 ตาย, 0	ที่ไม่รู้จัก				1.1

(1.2) เปิดโปรแกรม Advanced IP Scanner ขึ้นมา จะมีรายละเอียดดังรูป

(1.3) การตรวจหาเลข IP แอคเครสนั้น จะต้องทราบขอบเขตของจำนวน IP แอคเครส ก่อน ดังนั้นอุปกรณ์ที่ต้องการทราบเลข IP แอคเครสจึงต้องอยู่ในวง LAN เดียวกัน จึงจะตรวจหาหรือ สแกนหาเลข IP แอคเครสได้ เปิด Control Panel แล้วไปที่ Network and Internet >> Network and Sharing Center จากนั้นคลิกที่ Ethernet ดังรูป

¥	Control Panel\Network and Inter	net\Network and Sharing Center -	□ ×
🔄 🍥 🔻 🛉 😟 ד Control Pa	nel → Network and Internet → Network an	d Sharing Center v 🖒 Search Control Panel	م
Control Panel Home View your basic network information and set up connections			
Change adapter settings	View your active networks		
Change advanced sharing settings	Unidentified network Public network	Access type: No Internet access Connections: 🔋 Ethernet	
	Change your networking settings		
	Set up a broadband, dial-up, o	twork r VPN connection; or set up a router or access point.	
	Troubleshoot problems Diagnose and repair network p	roblems, or get troubleshooting information.	
See also		· · · · · · · · · · · · · · · · · · ·	
HomeGroup			
Internet Options			
Windows Firewall			

(1.4) หน้าต่าง Ethernet Status ที่แสดงสถานะของ Ethernet จะปรากฏขึ้น คลิกที่ Details...

9	Ethernet	Status	×
General			
Connection			-
IPv4 Connectivity		No Internet access	
IPv6 Connectivity	:	No network access	
Media State:		Enabled	
Duration:		01:08:51	
Speed:		100.0 Mbps	
Details			
Activity	Sent —	Received	-
Bytes:	25,878	171,336	
Properties	😗 Disable	Diagnose	
		Close	

(1.5) เมื่อคลิกแล้ว หน้าต่าง Network Connection Details ปรากฏขึ้นมา แสดงราย ละเอียดของการเชื่อมต่อดังรูป

Property	Value
Connection-specific DN	
Description	Broadcom NetXtreme Gigabit Ethernet
Physical Address	10-DD-B1-D8-FC-63
DHCP Enabled	No
IPv4 Address	192.168.137.1
IPv4 Subnet Mask	255.255.255.0
Pv4 Default Gateway	
Pv4 DNS Server	
IPv4 WINS Server	
NetBIOS over Topip En	Yes
Link-local IPv6 Address IPv6 Default Gateway	fe80::4957:9218:bb35:58d5%12
IPv6 DNS Servers	fec0:0:0:ffff::1%1
	fec0:0:0.ffff::2%1
	fec0:0:0ffff::3%1

ข้อมูลที่สำคัญสำหรับการหาขอบเขต IP แอคเครสคือ IPv4 Address และ IPv4

Subnet Mask โดยที่

IPv4 Address คือ หมายเลข IP แอคเครสของคอมพิวเตอร์ที่ทำหน่าที่ง่ายหรือ กำหนคเลข IP แอคเครส ให้แก่อุปกรณ์ที่ต่ออยู่ในวง LAN เคียวกัน

IPv4 Subnet Mask คือ จำนวนหมายเลข IP แอคเครสของอุปกรณ์ที่เป็นไป ไค้ภายในวง LAN เคียวกัน

(2) การคำนวนหาขอบเขตของเลข IP แอดเดรส

(2.1) <u>การคำนวนหาหมายเลขเริ่มต้นของ IP แอดเดรสหรือ Network IP</u>

เลข IP แอคเครส มีข้อมูล 4 ชุค ชุคละ 8 บิต และจะกั่นด้วย . (จุค) หากมีการกระทำ ทางด้านลอจิกจะต้องกระทำให้ตรงชุดด้วย จากขั้นตอนที่ (1.5) ในหัวข้อ 3.3.2 ได้หมายเลขของ IPv4 Address = 192.168.137.1 เมื่อแปลงเป็นเลขฐานสอง จะได้ 11000000.10101000.10001001.00000001

ส่วน IPv4 Subnet Mask = 255.255.0 เมื่อทำการแปลงเป็นเลขฐานสอง จะ ได้เป็น 11111111111111111111111100000000 นำค่าของ IPv4 Address มาแอนค์กับ IPv4 Subnet Mask จะ ได้ผลลัพธ์เป็น

11000000.10101000.10001001.00000000

แปลงเป็นเลขฐานสิบจะได้

192.168.137.0

จึงใค้ค่าเริ่มต้นของ IP แอคเครสเท่ากับ 192.168.137.0

(2.2) <u>การคำนวนหาหมายเลขสิ้นสุดของ IP แอดเดรสหรือ Broadcast IP</u>

วิธีที่ง่ายที่สุดคือ เมื่อทราบแล้วว่า ในแต่ละชุดข้อมูล มี 8 บิต มีค่า 0 ถึง 255 ให้พิจารณา จาก Subnet Mask ก่อนจะเห็น ได้ว่าชุดที่ 4 ของ Subnet Mask เป็นเลข 0 นอกนั้นมีค่าเป็นหมายเลข 255 ทั้ง 3 ชุด ดังนั้นจำนวนของหมายเลข Subnet Mask ที่จะทำให้จากเดิมคือ 255.255.255.0 ไปถึง 255.255.255.255 มีหมายเลขที่เป็นไปได้ 256 จำนวนหรือ 0 ถึง 255 ส่งผลให้หมายเลข IP แอดเดรส ที่อยู่ในวง LAN นี้เริ่มจาก Network IP คือ **192.168.137.0** ไปถึง**192.168.137.255** ก็จะ ได้ทั้งหมด 256 หมายเลข ดังนั้นขอบเขตหมายเลข IP แอดเดรสคือ 192.168.137.0 ถึง 192.168.137.255

(3) เริ่มต้นการตรวจหา IP แอดเดรสของอุปกรณ์

(3.1)เมื่อได้ขอบเขตที่ต้องการหาหมายเลข IP แอดเดรสแล้ว ให้นำหมายเลขนั้นใส่ใน ช่องหมายเลขแอดเดรส ดังรูป จากนั้นคลิกที่ปุ่มสแกน เพื่อทำการตรวจหา

đ			Advanced	I IP Scanner		- 🗆 ×	
ไฟล์ ปฏิบัติการ	ร การตั้งค่า ดู ช่ว	วยเหลือ					
▶ สแกน							
192.168.137.1	- 192.168.137.254				ด้วอย่าง: 192.168.0.1-192.168.0.100,	192.168.0.200	~
รายการผลลัพธ์	รายการโปรด						
สถานะ	ชื่อ		IP	ผุ้ผลิต			
<						>	
0 มีชีวิต, 0 ตาย, 0	ที่ไม่รู้จัก						

(3.2) เมื่อสแกนหรือตรวจหาเสร็จแล้ว โปรแกรมจะแสดงหมายเลข IP แอดเครสของ อุปกรณ์ทุกตัวที่ต่ออยู่ในวง LAN เดียวกันดังรูป

2	Ad	vanced IP Scar	ner		- 5	×
ไฟล์ ปฏิบัติกา	ร การตั้งค่า ดู ช่วยเหลือ					
▶ สแกน						
192.168.137.1	- 192.168.137.254		ตัวอย่าง: 192.168.	0. 1- 192. 168.0. 100,	192, 16	8. <i>0.200</i> 🗸
รายการผลลัพธ์	รายการโปรด					
สถานะ	ชื่อ	IP	ผู้ผลิต	MAC แอ็ดเดรส		
	INEX2	192.168.137.1	Apple	10:DD:B1:D8:		
⊳ 📮	raspberrypi.mshome.net	192.168.137.85	Raspberry Pi Foundation	B8:27:EB:8A:		
2 มีชีวิต, 0 ตาย, 2	52 ที่ไม่รู้จัก					

ด้วยขั้นตอนที่นำเสนอ ผู้ใช้งานจะทราบว่า หมายเลข IP แอดเดรสของ Raspberry Pi 3 เป็น หมายเลขอะไร ซึ่งอาจจะดูจากชื่อหรือผู้ผลิตก็ได้ โดยนำหมายเลข IP แอดเดรสไปทำการรีโมตหรือ เรียกอีกอย่างคือการควบคุมคอมพิวเตอร์ระยะไกลได้

3.4.3 ทดสอบใช้งาน

หลังจากติดตั้งโปรแกรม xrdp และ Advanced IP Scanner แล้ว ให้ทำการตรวจสอบหาเลข IP แอดเดรส ของบอร์ด Raspberry Pi 3 เมื่อได้เลขมาแล้ว จะเข้าสู่การทดสอบใช้งาน

- (1) ปลดเมาส์, คีย์บอร์ด และจอภาพออกจากบอร์ด Raspberry Pi
- (2) ต่อสาย LAN ระหว่างพอร์ตอีเธอร์เน็ตของ Raspberry Pi กับคอมพิวเตอร์
- (3) เปิดโปรแกรม Remote Desktop Connection ดังรูป

5	Remote Desktop Connection 🗧 🗆 🗙
-	Remote Desktop Connection
Computer: User name: You will be a	192.168.1.36 ✓ None specified sked for credentials when you connect.
Show C	Dptions Connect Help

(4) ใส่ IP แอคเครสของบอร์ค Raspberry Pi 3 (ได้มาจากการตรวจหาด้วยโปรแกรม Advanced IP Scanner) ที่ช่อง Computer: แล้วคลิกปุ่ม Connect เพื่อทำการเชื่อมต่อ

เมื่อเชื่อมได้แล้วจะปรากฏหน้าต่าง Login to xrdp ดังรูป

Lo	ogin to ×rdp	1	
	-375	Module	sesman-Xvnc
		username	pi
		password	******
	SAL A		
	KKC ~ ~		
L		ОК	Cancel Help
		1	

(5) ใส่ username เป็น pi และ password เป็น raspberry (เป็น username และ password มาตรฐาน - ให้ป้อนรหัสผ่านและชื่อผู้ใช้งานหากมีการกำหนดให้แตกต่างออกไป) แล้วคลิกปุ่ม OK
 (6) จะปรากฏหน้าเดสก์ท็อปของ Raspbian Jessie with Pixel ขึ้นมา



ด้วยขั้นตอนทั้งหมดจะช่วยให้ใช้งานบอร์ด Raspberry Pi ผ่านกอมพิวเตอร์อีกเกรื่องหนึ่งที่ ต่อห่างออกไปได้

3.5 การควบคุมบอร์ด Raspberry Pi 3 ด้วยคอมพิวเตอร์โดยใช้ VNC เมื่อทำงานในโหมดกราฟิก

นอกจากการใช้วิธีการ SSH และ xrdp ในการเข้าถึงเพื่อแลกเปลี่ยนข้อมูลหรือควบคุมการทำ งานของบอร์ด Raspberry Pi จากคอมพิวเตอร์อีกเครื่องหนึ่ง ยังมีอีกหนึ่งวิธีที่ได้รับความนิยม นั่นคือ การใช้โปรแกรม VNC

VNC ย่อมาจาก Virtual Network Computing เป็นโปรแกรมประเภทรีโมตคอนโทรลเพื่อ ติดต่อจากคอมพิวเตอร์เครื่องหนึ่งไปยังคอมพิวเตอร์อีกเครื่องหนึ่ง หรือเรียกอีกอย่างคือ ระหว่างเครื่อง ใคลเอ็นต์ (client) กับเซิร์ฟเวอร์ (server) โดย VNC เป็นระบบที่ใช้ทรัพยากรอย่างเช่น หน้าจอ คอมพิวเตอร์, เมาส์ และ Keyboard สำหรับทำการติดต่อหรือรีโมตไปยังคอมพิวเตอร์เครื่องอื่นๆ ทั้ง ยังทำงานร่วมกันได้หลายเครื่องพร้อมๆ กันด้วย

ประโยชน์ของ VNC โดยสรุปมีดังนี้

- File-Transfer : รองรับการรับและส่งไฟล์ต่างๆ
- Text Chat : รองรับการสนทนาผ่านเครือข่าย
- Auto Reconnect : มีความสามารถในการเชื่อมต่อใหม่โดยอัตโนมัติ

ในระบบปฏิบัติการ Raspbian Jessie with Pixel ตัวใหม่ล่าสุด (Release : 2016-09-23) ได้ทำการ ติดตั้ง VNC Server ไว้แล้ว โดยที่ผู้ใช้งานไม่ต้องทำการติดตั้งเข้าไปใหม่เพียงเปิดใช้การทำงานของ VNC เท่านั้น ก็จะใช้งาน VNC ได้ทันที

3.5.1 การตั้งค่าของอุปกรณ์

การตั้งค่าในส่วน Raspberry Pi (กำหนดเป็นเซิร์ฟเวอร์)

ทำการเปิดใช้งาน VNC โดยเข้าไปที่ Preferences > Raspberry Pi Configuration จากนั้น เลือกที่แท็บ Interfaces คลิกเลือกเปิดการใช้งาน VNC ดังรูป จากนั้นคลิกปุ่ม OK



การตั้งค่าในส่วนคอมพิวเตอร์ (กำหนดเป็นไคลเอ็นต์)

ที่คอมพิวเตอร์จะต้องได้รับการติดตั้งโปรแกรม VNC Viewer ไว้ก่อน คาวน์โหลดได้จาก https://www.realvnc.com/download/vnc/ จากนั้นทำการติดตั้งให้เรียบร้อย

3.5.2 การใช้งานโปรแกรม VNC Viewer

สิ่งที่ต้องทราบก่อนที่จะใช้โปรแกรมคือ หมายเลข IP แอดเดรสของ Raspberry Pi ที่ได้จาก การค้นหา โดยต้องเชื่อมต่อบอร์ด Raspberry Pi 3 เข้ากับคอมพิวเตอร์ผ่านทางพอร์ตอีเธอร์เน็ตหรือ ใช้ WiFi โดยต้องต่อให้อยู่ในวงของเครือข่ายเดียวกัน จากนั้นทำการค้นหาหมายเลข IP แอดเดรส

V2 VNC Viewer File View Help	- 0	×
Enter a VNC Server address or search	Le Sign in	-
There are no connections in your address bo	k at present.	
Sign in to your RealVNC account to automatically disc	ver team computers.	
Alternatively, enter the VNC Server IP address or hostname in th	Search bar to connect direct.	

จากนั้นนำหมายเลข IP แอคเครสไปใส่ในช่อง Enter a VNC Server address or search ดังรูป แล้วกคลีย์ Enter



ใส่ Username : pi และ Password : raspberry หากมีการตั้งรหัสผ่านใหม่ ให้ใช่รหัสผ่านนั้น จากนั้นคลิกปุ่ม OK

VNC Server:	192.168.1.2::5900
Username:	pi
Password:	•••••
Remembe	er password
Catchnbrase	: Sweet Agatha crystal. Premium press Abraham.
cattenprirase	3 3 1

จะปรากฏหน้าต่างเดสก์ท็อปของ Raspberry Pi 3 ที่ใช้ระบบปฏิบัติการ Raspbian Jessie with Pixel นับจากนี้ผู้ใช้งานจะสามารถติดต่อและใช้งานบอร์ด Raspberry Pi ผ่านทางคอมพิวเตอร์อีกตัว หนึ่งได้ โดยไม่ต้องต่อจอภาพ, เมาส์ และกีย์บอร์ดเพิ่มเติม





^{บทที่ 4} ดำสั่งพื้นฐานบนเทอร์มินอลของ ระบบปฏิบัติการ Linvx



ระบบปฏิบัติการหลักที่ติดตั้งลงในบอร์ดกอมพิวเตอร์ Raspberry Pi ในทุกรุ่นกือ Raspbian ซึ่งเป็น ระบบปฏิบัติการ Linux แบบหนึ่งที่พัฒนาขึ้นมาเพื่อรองรับบอร์ดกอมพิวเตอร์ขนาดเล็กอย่าง Raspberry Pi โดยเฉพาะ **ในการใช้งานระบบปฏิบัติการ Linux มี 2 โหมด**กือ *โหมดคอมมานด์ไลน์* (กล้ายกับระบบปฏิบัติ การ DOS) และ *โหมดกราฟิก* (กล้ายระบบปฏิบัติการวิน โดวส์หรือ MAC OS) โดยส่วนใหญ่จะนิยมทำงาน กันใน โหมดกอมมานด์ไลน์เนื่องจากเร็วและ ไม่ใช้ทรัพยากรมาก อย่างไรก็ตาม การทำงานใน โหมดกราฟิก แท้ที่จริงก็มีกำสั่งและการประมวลผลแบบคอมมานด์ไลน์จำนวนมากอยู่เบื้องหลัง ดังนั้นการทำความรู้จัก กำสั่งพื้นฐานบนเทอร์มินอลของระบบปฏิบัติการ Linux จึงเป็นเรื่องที่จำเป็น



รูปที่ 4–1 หน้าจอเมื่อบอร์ด Raspberry Pi 3 ทำการบูต, เตรียมความพร้อมและตรวจสอบทรัพยากรทั้งหมด

4.1 เทอร์มินอลคืออะไร

โดยปกติเมื่อบอร์ด Raspberry Pi เริ่มทำงาน มันจะทำการเตรียมความพร้อมและตรวจสอบ ทรัพยากรที่เกี่ยวข้องทั้งหมด ซึ่งหากมีการต่อจอแสดงผลไว้ จะเห็นกระบวนการต่างๆ แสดงออกมา เรียงกันเป็นบรรทัด ดังรูปที่ 4-1 ซึ่งขั้นตอนนี้มีในคอมพิวเตอร์ทุกระบบปฏิบัติการ เพียงแต่ว่า ผู้ผลิต แต่ละรายทำการซ่อนหน้าจอในส่วนนี้ ที่ผู้ใช้งานเห็นในคอมพิวเตอร์ส่วนใหญ่อาจเป็นจอว่างๆ หรือ เป็นหน้าจอแสดงโลโก้ของระบบปฏิบัติการนั้นๆ จนกระทั่งการเตรียมความพร้อมเสร็จสิ้นก็จะเข้าสู่ หน้าจอของการเริ่มต้นการทำงาน

สำหรับระบบปฏิบัติการ Raspbian รุ่นก่อนหน้า Raspbian Jessie with Pixel หากบอร์ด Raspberry Pi ไม่ได้ถูกตั้งให้บูตเข้าสู่โหมดกราฟิก โดยอัต โนมัติ เมื่อการบูตเสร็จสิ้น มันจะแสดงหน้าจอใน โหมดคอม มานด์ ไลน์และเชลพร้อมพ์ pi@raspberrypi \$ เพื่อรอรับการป้อนคำสั่ง ดังรูปที่ 4-2 หน้าจอใน ลักษณะนี้เรียกว่า เทอร์มินอล (Terminal) สำหรับผู้ที่เคยใช้งาน DOS จะกุ้นเคย แต่สำหรับผู้ใช้งานที่เคย ใช้งานเฉพาะวินโดวส์หรือ MAC OS อาจไม่รู้จัก

เมื่อมาถึงตรงนี้ บอร์ค Raspberry Pi พร้อมทำงาน เมื่อมีการป้อนกำสั่งซึ่งจะ ได้นำเสนอข้อมูล โดยสรุปหัวข้อต่อไป

```
raspberrypi login: pi
Password:
Last login: Mon Feb 23 19:37:59 GMT-7 2015 from 192.168.1.50 on pts/2
Linux raspberrypi 3.18.7-v7+ #755 SMP PREEMPT Thu Feb 12 17:20:48 GMT 2015 armv71
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
pi@raspberrypi * $
```

รูปที่ 4–2 เมื่อบูตระบบเสร็จสิ้น และป้อนรหัสผ่านเพื่อเข้าสู่การทำงาน หน้าจอของการทำงานจะเข้าสู่โหมด คอมมานด์ไลน์ พร้อมรับคำสั่งเพื่อทำงานต่อไป

4.2 โครงสร้างใดเร็กตอรี่หลักของระบบปฏิบัติการ Linux

เพื่อให้การเรียนรู้เกี่ยวกับระบบปฏิบัติการ Raspbian ที่ติดตั้งลงบนบอร์ด Raspberry Pi เป็น ไปอย่างมีประสิทธิภาพ ในหัวข้อต่อจากนี้จะได้อธิบายถึงภาพรวมของโครงสร้างการจัดการไฟล์และ ไดเร็กตอรี่ของระบบปฏิบัติการ Linux เพื่อให้เกิดความเข้าใจที่ตรงกัน

เหตุผลที่ต้องทำความเข้าใจในเรื่องนี้เนื่องจากระบบปฏิบัติการ Raspbian เป็นระบบปฏิบัติการ ที่ได้รับการสืบทอดมาจาก Linux โดยมีการปรับปรุงแก้ไขไฟล์ระบบบางส่วนเพื่อให้ใช้ทรัพยากรลดลง ทำให้ติดตั้งลงบนบอร์ด Raspberry Pi 3 ได้ ดังนั้นการทำความเข้าใจในโครงสร้างไดเร็กตอรี่ของระบบ ปฏิบัติการ Linux จึงเป็นการทำความเข้าใจโครงสร้างไดเร็กตอรี่ของระบบปฏิบัติการ Raspbian ไปด้วย

โครงสร้างใดเร็กตอรี่หลักของระบบปฏิบัติการ Linux แสดงด้วยแผนภาพในรูปที่4-3 ผู้เรียนรู้และ พัฒนาโปรแกรมบน Raspberry Pi จะได้ทราบถึงตำแหน่ง, เส้นทาง (พาธ) ของไดเร็กตอรี่และไฟล์ เพื่อ ให้เข้าถึงและเรียกใช้ง่ายขึ้น เนื่องจากโดยส่วนใหญ่การพัฒนาโปรแกรมบน Raspberry Pi 3 มักกระทำ ในโหมดคอมมานด์ไลน์ผ่านหน้าต่างเทอร์มินอล หากไม่ทราบถึงโครงสร้างไดเร็กตอรี่ อาจทำให้ต้องใช้ เวลาในการค้นหาไดเร็กตอรี่ที่จัดเก็บไฟล์ หรืออาจหาไม่พบได้ แผนภาพนี้จะช่วยลดปัญหาดังกล่าวลง

system root /	
	เก็บคำสั่งระบบหลัก
	เกบสวนบูตระบบ องโกรณ์ต่าง ๆ
	- เก็บค่าติดตั้งอื่นๆ
home	ไดเร็กตอรี่ส่วนผู้ใช้งาน
lib	ไลบารีสำหรับไฟล์ระบบเรียกใช้งาน
lost+found	เก็บขึ้นส่วนที่ได้จากการตรวจกู้ไฟล์ด้วย fsck
- media	ตำแหน่ง mount ของสื่อเก็บข้อมูลแบบถอดได้ เช่น ซีดีรอม, USB ดิสก์, การ์ดรีดเดอร์
mnt	ตำแหน่ง mount ของระบบระหว่างทำงาน
- opt	เก็บส่วน Add-on ของโปรแกรมอื่นๆ
proc	ส่วนเก็บข้อมูลระหว่างการทำงาน รวมถึงสถานะของระบบ
root	ไดเร็กตอรี่ของ root
sbin	เก็บคำสั่งทำงานสำหรับผู้ดูแลระบบ
sys	ไฟล์ระบบที่ติดต่อ Kernel
tmp	พื้นที่ส่วนกลางสำหรับเก็บข้อมูลระหว่างทำงาน (temporary)
usr	พื้นที่ไฟล์งานระบบที่ผู้ใช้ทั่วไปเข้าถึงได้
L var	พื้นที่เก็บข้อมูล ตัวแปร ไฟล์ส่วนกลางอื่นๆ ที่ใช้งานกันในระบบ
	รวมถึงบริการไฟล์สำหรับเครือข่ายด้วย

รูปที่ 4-3 โครงสร้างไดเร็กตอรี่หลักของระบบปฏิบัติการ Linux

4.3 การใช้งานคำสั่งบนเทอร์มินอลของระบบปฏิบัติการ Rasbian

เมื่อบูตระบบจนเสร็จเรียบร้อย หน้าจอแรกสุดบนเทอร์มินอลคือ รอล็อกอินด้วยชื่อผู้ใช้ pi รหัสผ่านเป็น raspberry (จะไม่ปรากฏให้เห็นขณะพิมพ์ป้อนรหัส) ขั้นตอนนี้อาจข้ามได้ หากมีการตั้ง ให้ทำการล็อกอินแบบอัตโนมัติ

login: pi Password: <mark>raspberry</mark> Last login: Thu Aug 27 14:38:30 ICT 2015 from service.mshome.net on pts/0 Linux raspberrypi 3.18.11-v7+ #781 SMP PREEMPT Tue Apr 21 18:07:59 BST 2015 armv71
The programs included with the Debian GNU/Linux system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law. pi@raspberrypi \$

เมื่อเข้าสู่เชลพร้อมพ์สำหรับป้อนกำสั่ง จะแสดงคังนี้

pi@raspberrypi 🍈 💲 🔄

มีความหมายว่า ชื่อผู้ใช้ pi ทำงานบนโฮสต์เนม (hostname) ชื่อ raspberry

~ คือ home directory หรือไดเร็กตอรี่ตั้งต้นของผู้ใช้ ซึ่งอยู่ที่ /home/pi

ทดลองเปลี่ยนไปที่ใดเร็กตอรี่ root / โดยการใช้คำสั่ง cd /

pi@raspberrypi ~ \$ cd / pi@raspberrypi / \$ _

ถ้าต้องการดูว่ามีอะไรอยู่ในไดเร็กตอรี่ที่อยู่ ณ ขณะนี้บ้าง ให้พิมพ์กำสั่ง 1s เพื่อแสดงรายการ ไฟล์แบบสรุป แต่ถ้าต้องการดูรายละเอียครวมถึงรายการที่ซ่อนอยู่ ใช้กำสั่ง 1s -a1 จะแสดงผล ต่างกัน ดังรูปที่ 4-4

รายละเอียดที่แสดงนั้นประกอบด้วยข้อมูลหลายส่วน ดังแสดงรายละเอียดโดยสรุปไว้ในรูปที่ 4-5 นี้

pi@raspber bin boot	rypi .bak	/\$] etc	ls lil	b	ſ	nedi	ia o	pt	r	oot	sbin	srv	tmp	var
boot dev		home	e los	st+four	nd r	nnt	р	roc	: r	un	selinu×	(sys	s usr	
pi@raspber	rvpi	/ \$]	ls -ai	1										
total 100														
drwxr-xr-x	23	root	root	4096	Aug	24	09:5	2.						
drwxr-xr-x	23	root	root	4096	Aug	24	09:5	2.						
drwxr-xr-x	2	root	root	4096	Aug	13	11:4	8 b	oin					
drwxr-xr-x	6	root	root	16384	Jan	1	197	0 b	boot					
drwxr-xr-x	3	root	root	4096	Apr	23	15:2	6 b	boot	.bak				
drwxr-xr-x	12	root	root	3220	Aug	26	16:2	3 <mark>c</mark>	lev					
drwxr-xr-x	113	root	root	4096	Aug	27	15:5	1 e	etc					
drwxr-xr-x	Э	root	root	4096	Feb	15	201	5 h	nome					
drwxr-xr-x	13	root	root	4096	Apr	23	21:5	1 1	lib					
drwx	2	root	root	16384	Feb	15	201	51	lost	+foui	nd			
drwxr-xr-x	2	root	root	4096	Aug	19	15:3	<u>6</u> m	nedi	a				
drwxr-xr-x	8	root	root	4096	Aug	19	15:3	<u>6</u>	int					
drwxr-xr-x	6	root	root	4096	Feb	16	201	5 o	pt					
dr-xr-xr-x	100	root	root	0	Jan	1	197	0 p	proc					
drwx	. 4	root	root	4096	Aug	19	14:4	Į r	oot					
drwxr-xr-x	14	root	root	560	Aug	26	16:2	4 r	un					
drwxr-xr-x	2	root	root	4096	Aug	13	11:5	3 S	sbiņ					
drwxr-xr-x	2	root	root	4096	Jun	20	201	2 5	seli	nux				
drwxr-xr-x	3	root	root	4096	Aug	19	14:0	<u>9</u> s	srv					
dr-xr-xr-x	11	root	root	V.	And	21	14:2	/ 5	sys					
drwxrwxrwx	4	root	root	4096	Aug	27	15:3	9	tmp					
drwxr-xr-x	11	root	root	4096	Mar	24	22:3	3 נ	isr					
drwxrwxrwx	12	root	root	4096	Hug	19	14:4	ί υ	/an					

รูปที่ 4-4 หน้าต่างเทอร์มินอลแสดงการใช้คำสั่ง 1s และ 1s -a1 ในการแสดงรายการ ของไฟล์ในไดเร็กตอรี่



รูปที่ 4–5 คำอธิบายของข้อมูลภายในไดเร็กตอรี่เมื่อเรียกให้แสดงด้วยคำสั่ง 1s -a1 ของระบบปฏิบัติการ Raspbian ที่ใช้พื้นฐานสำคัญมาจากระบบปฏิบัติการ Linux

72 • Raspberry Pi 3 : รู้จักและการใช้งานเบื้องตัน

4.4 สิ่งที่ควรทราบเบื้องต้นในการใช้งานบนระบบปฏิบัติการ Linux

การใช้งานคำสั่งบนระบบปฏิบัติการ Linux แตกต่างจากระบบปฏิบัติการ DOS (หรือ Command Prompt บนระบบปฏิบัติการวิน โดวส์) ดังนี้

 ชื่อคำสั่ง ชื่อ ใดเร็กตอรี่ ชื่อ ใฟล์ และพารามิเตอร์ต่างๆ ทั้งหมดในระบบปฏิบัติการ Linux อักษร ตัวพิมพ์เล็กและตัวพิมพ์ใหญ่ มีความหมายต่างกัน ทดแทนกันไม่ได้ จำเป็นต้องเรียกให้ถูกต้องเสมอ

ระบบปฏิบัติการ Linux ใช้ช่องว่าง (space seperation) ในการแยกคำสั่งกับพารามิเตอร์ เช่น
 เดียวกับระบบปฏิบัติการ UNIX ดั้งเดิม จะพิมพ์ย่อติดๆ กันแบบระบบปฏิบัติการ DOS เช่น cd..
 ไม่ได้ จะต้องพิมพ์เป็น cd ..

- ตัวกั่นใคเร็กตอรี่ใช้ / (ในขณะที่ระบบปฏิบัติการ DOS ใช้ \)
- หมายถึงตำแหน่ง home directory หรือไดเร็กตอรี่ตั้งต้นของผู้ใช้
- . หมายถึงตำแหน่ง ใดเร็กตอรี่ปัจจุบัน (เหมือนระบบปฏิบัติการ DOS)

แหมายถึงตำแหน่ง ไคเร็กตอรี่ที่อยู่ถำดับบนขึ้น ไปหนึ่งอันดับของปัจจุบัน (เหมือนระบบ ปฏิบัติการ DOS) เช่น ถ้าปัจจุบันอยู่ที่ตำแหน่ง /var/www ถ้าพิมพ์กำสั่ง cd .. จะเป็นการกลับมา ยัง /var

 โดยปกติไฟล์สั่งงานระบบเกือบทุกตัว จะมีคู่มือการใช้คำสั่งหรือ manual มาด้วย หากต้อง การทราบวิธีการใช้พารามิเตอร์ของคำสั่งอย่างละเอียด ให้พิมพ์ man นำหน้าคำสั่ง เพื่อเปิดวิธีใช้งาน เช่น man 1s หน้าจอจะแสดงรายละเอียดของคำสั่ง 1s พร้อมพารามิเตอร์ที่ใช้งานได้ขึ้นมา

 โดยปกติระบบปฏิบัติการ Raspbian จะล็อกอินด้วย pi ซึ่งถือสิทธิ์ผู้ใช้ปกติ หากต้องการ ทำงานในสิทธิ์ของ super user หรือ root เพื่อเข้าถึงหรือแก้ไขระบบ จะใช้การสั่งงานผ่านโหมดคำสั่ง sudo (super user do) เช่น

sudo nano /boot/config.txt

เรียกตัวแก้ไขไฟล์ตั้งค่าในระบบ ซึ่งต้องใช้สิทธิ์ของ super user ในการแก้ไข

 ระหว่างทำงานคำสั่งบนเทอร์มินอล หากต้องการจบการทำงาน โดยปกติจะใช้การกดแป้น Ctrl และ C ซึ่งจะส่งสัญญาณ SIGTERM (signal terminate) ให้ระบบเพื่อหยุดทำงาน แต่ถ้าโปรแกรม ที่ทำงานอยู่ขณะนั้นเป็นกระบวนการหรือ process ที่ทำงานในระดับที่สูงกว่าหรือไม่ตอบสนอง สัญญาณนี้ จำเป็นต้องทำการเปิดใช้เทอร์มินอลอื่น แล้วใช้คำสั่ง ps เพื่อดู process ID ที่ทำงานก่อน หน้า แล้วใช้กำสั่ง kill -9 ตามด้วยเลข process ID ที่ได้จากกำสั่ง ps เพื่อบังกับให้หยุด
ระหว่างทำงานกำสั่งบนเทอร์มินอล หากกดแป้น Ctrl และ Z แล้วออกมาที่หน้าเชลพร้อมพ์ เป็นเพียงการหยุดกระบวนการแล้วสลับมากลับมาที่เชลพร้อมพ์ชั่วคราวเท่านั้น แต่ตัว process ยังคง อยู่ และพร้อมจะกลับไปทำงานต่อเมื่อพิมพ์กำสั่ง fg

ห้ามถอด microSD การ์ดออกจากบอร์ดอย่างเด็ดขาดในขณะทำงาน

 ก่อนปิดเครื่องหรือตัดไฟเลี้ยงออกจากระบบ ควรทำการหยุดระบบให้เรียบร้อยก่อน โดย ใช้กำสั่ง sudo halt หรือ sudo poweroff หรือ sudo shutdown now รอจนระบบหยุด ทำงานโดยสมบูรณ์ก่อน จึงตัดไฟเลี้ยง

4.5 ตัวประมวลไปป์ไลน์ (Pipelines) และ ตัวจัดการเปลี่ยนทิศทาง I/O

ในระบบปฏิบัติการ Linux มีตัวควบคุมการทำงานระหว่างกำสั่งในคราวเดียว อันได้แก่ .

> ใช้เปลี่ยนทิศทางของเอาต์พุต I/O ไปยังไฟล์ I/O ที่ระบุได้ เช่น การเขียนหน้าจอลงไฟล์

ls -al > output.txt

>> ใช้สำหรับเปลี่ยนทิศทางของเอาต์พุต I/O ไปยังไฟล์ I/O เช่นเดียวกับ > แต่เป็นการเพิ่ม ท้ายไฟล์

> echo "Hello" > test.txt echo "Raspberry" >> test.txt ถ้าเปิดไฟล์ test.txt ดู จะเห็นข้อความ

Hello

```
Raspberry
```

< ใช้ในกรณีรับอินพุตจากไฟล์ I/O มักใช้ในการเปิดไฟล์ในโปรแกรม

I เป็นไปป์ใช้สำหรับการประมวลผลโดยส่งผ่านเอาต์พุตมาผ่านกำสั่งอีกทอดหนึ่ง เช่น

ls -al | more

กรณีไฟล์ที่แสดงมีจำนวนมากเกินกว่าจะแสดงได้ในหน้าจอเดียว คำสั่ง **more** จะเป็น ตัวจำกัดให้เห็นเพียงหน้าจอเดียวก่อน แล้วรอผู้ใช้กดแป้นคีย์บอร์ดเพื่อดำเนินการแสดงหน้าจอถัดไป

tee ใช้ในกรณีที่เป็นไปปกลาง เพื่อพักข้อมูลก่อน แล้วส่งผ่านไปยังไปป์หลังอีกทอดหนึ่ง เช่น

ps ax | tee list_process.txt | more

อันดับแรกจะเก็บผลลัพธ์ที่ได้จากคำสั่ง **ps** มาไว้ในไฟล์ list_process.txt ก่อน แล้ว ส่งผ่านไปแสดงบนหน้าจอผ่านทางคำสั่ง **more** ซึ่งถ้า process มีมากกว่าการแสดงในหน้าจอเดียว คำสั่ง tee จะจัดการแสดงหน้าจอในหน้าต่อไปหลังจากมีการกดคีย์บอร์ด

4.6 คำสั่งที่ควรทราบของระบบปฏิบัติการ Linux ที่ใช้กับ Raspberry Pi 3

4.6.1 **ls** (List Directory Contents)

คำสั่งแสดงรายชื่อไฟล์ในไดเร็กตอรี่นั้นๆ ใช้ร่วมกับอักงระช่วยเหลือ (wildcard) เช่น ? และ * ได้ รองรับพารามิเตอร์ต่อท้ายได้หลายแบบ (รายละเอียดให้ดูจาก man ls) ที่เรียกใช้งานบ่อยได้แก่

-a แสดงชื่อไฟล์ที่ซ่อนไว้ให้เห็น (มี . นำหน้าชื่อ)

pi@raspberrypi ~ \$ ls -a config .gstreame dbus .gvfs Adafruit_Python_DHT Desktop .icons .arduino .dmrc .idlerc .bash_history Downloads .java .bash_logout .fceux .lazarus .bashrc .fontconfig .local .cache .gconf logs	r-0.10 .Mathematica .thumbnails .pki .vnc .profile wiringPi .pulse .WolframEngine .pulse-cookie .Xauthority python_games .xsession-errors .sketchbook .xsession-errors.old .sonic-pi
--	---

-1 แสดงชื่อไฟล์แบบแจกแจงรายละเอียด

pi@raspberryp	i Ű	\$	ls -l				
total 28							
drwxr-xr-x 10	pi	pi	4096	Aug	14	11:33	Adafruit_Python_DHT
drwxr-xr-x 7	pi	pi	4096	Aug	19	14:44	Desktop
drwx 2	pi	pi	4096	Apr	23	16:50	Downloads
drwxr-xr-x 2	pi	pi	4096	Aug	19	13:26	logs
drwxrwxr-x 2	\mathbf{pi}	pi	4096	Jan	27	2015	python_games
drwxr-xr-x 2	pi	pi	4096	Mar	24	22:42	sketchbook
drwxr-xr-x 9	pi	pi	4096	Apr	23	16:28	wiringPi

- al แสดงชื่อไฟล์แบบแจกแจงรายละเอียดและแสดงชื่อที่ซ่อนด้วย

pi@raspberryp	i ~ \$	ls –al					
total 324							
drwxr-xr-x 30	pi	pi	4096	Aug	27	13:48	
drwxr-xr-x 3	root	root	4096	Feb	15	2015	
drwxr-xr-x 10	pi	pi	4096	Aug	14	11:33	Adafruit_Python_DHT
drwxr-xr-x 2	pi	pi	4096	Mar	24	22:42	.arduino
-rw 1	. pi	pi	1598	Aug	26	16:23	.bash_history
-rw-rr 1	. pi	pi	220	Feb	15	2015	.bash_logout
-rw-rr 1	pi	pi	3243	Feb	15	2015	.bashrc
drwxr-xr-x 12	pi	pi	4096	Apr	23	22:17	. cache
drwxr-xr-x 21	pi	pi	4096	Aug	13	16:05	.config
drwx 3	pi	pi	4096	Mar	24	21:53	dbus
drwxr-xr-x	pi bi	pi	4096	Hug	19	14:44	Desktop
-rw-rr l	pi	pi	, 35	Hug	Τä	18:22	,dmrç
drwx 2	pi	pi	4096	Hpr	23	16:50	Downloads
drwx b	pi	pi	4096	Hpr	23	21:20	.tceux
drwxr-xr-x 2	pi	pi	4096	Hug	19	13:25	.tontcontig
drwx	pi	pi	4096	Hpr	23	22:02	.gcont
drwxr-xr-x 2	рi	pi	4096	Hpr	23	22:01	.gstreamer-0.10
drwx 2	pi	pi	4096	Hug	<u>24</u>	03:08	.gvts
drwx	pi	pi	4096	Mar	24	22:26	.1cons
drwxr-xr-x 2	pi	pi	4096	Hpr	23	12:22	.idlerc
drwxr-xr-x	pi	p1	4096	Mar	24	22:42	. java
drwxr-xr-x c	p1	pi	4090	Hpr	23	20:37	. lazarus
arwx a	pi	pi	4070	Mar	24	21:00	,10Ca1
drwxr-xr-x Z	рі	pi	4096	Hug	17	10.00	
arwxr-xr-x 5	p1	pi	4096	Hpr	23	18:28	.Mathematica
arwx a	p1	p1	4096	нpr	ZJ	19:98	.pK1

4.6.2 cd (Change Directory)

คำสั่งเปิดเข้าสู่ไดเร็กตอรี่ที่ระบุ ใช้การอ้างอิงแบบสัมบูรณ์หรือ absolute ระบุตำแหน่งโดยตรง หรือแบบสัมพัทธ์หรือ relative โดยใช้ . หรือ .. ร่วมได้

```
<u>ตัวอย่างที่ 4-1</u>
<u>คำสั่งเปิดเข้าสู่ไดเร็กตอรี่ที่ระบุ ใช้การอ้างอิงแบบสัมบูรณ์หรือ absolute</u>
cd /
cd /home/pi
```

cd ~ (~ แทนตำแหน่ง home ของผู้ใช้ ซึ่งมีความหมายเดียวกับ /home/pi)

cd ~/Desktop

<u>ตัวอย่างที่ 4-2</u>

```
<u>คำสั่งเปิดเข้าสู่ไดเร็กตอรี่ที่ระบุ ใช้การอ้างอิงแบบสัมพัทธ์หรือ relative</u>
```

ถ้าเดิมอยู่ที่ ~/Desktop ต้องการไปที่ ~/Download ใช้ .. ขึ้นไป 1 อันดับ แล้วเข้าสู่ไดเร็กตอรี่ Downloads

> pi@raspberrypi ~/Desktop \$ cd /var/www pi@raspberrypi /var/www \$

ถ้าเดิมอยู่ที่ /usr/local/bin ต้องการไปที่ไดเร็กตอรี่ /var/www ใช้ .. ต่อกัน 3 ครั้งแล้วต่อด้วย var/www ก็ได้

> pi@raspberrypi ~/Desktop \$ cd ../Downloads pi@raspberrypi ~/Downloads \$ _

แต่ถ้าหากผู้ใช้ไม่มี permission หรือไม่ได้รับสิทธิ์ในการเข้าถึงไดเร็กตอรี่ จะเข้าสู่ไดเร็กตอรี่ ที่เรียกไม่ได้ 76 • Raspberry Pi 3 : รู้จักและการใช้งานเบื้องตัน

4.6.3 mkdir (Make Directory)

เป็นคำสั่งสร้างใดเร็กตอรี่ใหม่ตามที่ระบุ ใช้การอ้างอิงแบบสัมบูรณ์ ระบุตำแหน่งโดยตรง หรือแบบสัมพัทธ์กี่ได้

<u>ตัวอย่างที่ 4-3 สร้าง Examples บน Deskto</u>p

```
pi@raspberrypi ~/Desktop $ mkdir Examples
pi@raspberrypi ~/Desktop $ ls -1
total 16
drwxr-xr-x 2 pi pi 4096 Aug 30 00:40 Examples
```

ใช้คำสั่ง **1s** เพื่อเข้าดูรายละเอียด จะเห็นว่า pi เป็น owner และ pi มี permission หรือมี สิทธ์เข้าถึงจึงเห็นรายละเอียดของไฟล์ Examples ได้

4.6.4 sudo (Super User Do)

ใช้นำหน้าคำสั่งใดๆ ที่ต้องการใช้สิทธิ์ (Granted) ของ super user หรือ root ซึ่งเข้าถึง ปรับ แต่งแก้ไขส่วนประกอบของระบบได้ทุกอย่าง และอยู่เหนือ permission หรือสิทธิ์ในการเข้าถึงไฟล์ และไคเร็กตอรี่ทั้งปวง

<u>ตัวอย่างที่ 4-4</u>

โดยปกติสำหรับไฟล์ตั้งค่าระบบ ผู้ใช้ปกติจะแก้ไขและบันทึกไม่ได้ เมื่อใช้คำสั่ง sudo นำหน้า nano ก็จะเข้าถึงไฟล์นั้นได้

pi@raspberrypi ~ \$ sudo nano /boot/config.txt

4.6.5 rmdir (Remove Directory)

เป็นคำสั่งลบไดเร็กทอรี่ตามที่ระบุ ก่อนใช้กำสั่งนี้ จะต้องไม่มีไฟล์เหลือในไดเร็กตอรี่ด้วย จึงจะลบออกได้ แต่หากต้องการลบโดยไม่สนใจไฟล์ด้านใน ให้ใช้กำสั่ง m –Rf แทน

ร<u>ูปแบบ</u> rmdir ชื่อไดเร็กตอรี่ที่ต้องการลบ <u>ตัวอย่างที่ 4-5 ลบ Examples บน Deskto</u>p pi@raspberrypi ~/Desktop \$ rmdir Examples

4.6.6 rm (Remove)

เป็นคำสั่งลบไฟล์ที่รองรับทั้งการระบุไฟล์โดยตรง และใช้ร่วมกับอักขระช่วยเหลือทั้ง ? และ * ได้ รวมถึงรองรับพารามิเตอร์ต่อท้ายได้หลายแบบ (รายละเอียดให้ดูจาก man rm)

pi@raspberrypi ~/Desktop	\$ ls -al
	a aa aa ia
drwxr-xr-x 2 pi pi 4096	Hug 30 00:48 .
drwxr-xr-x 25 pi pi 4096	Aug 30 17:52
-rw-rr 1 pi pi 5316	Aug 29 10:21 geany.desktop
-rw-rr 1 pi pi 16	Aug 29 09:24 test.py
pi@raspberrypi ~/Desktop	\$ rm test.py
pi@raspberrypi ~/Desktop	\$ ls -al
total 16	
drwxr-xr-x 2 pi pi 4096	Aug 30 17:55 .
drwxr-xr-x 25 pi pi 4096	Aug 30 17:52
-rw-rr 1 pi pi 5316	Aug 29 10:21 geany.desktop
pi@raspberrypi ~/Desktop	\$

แต่ถ้าไม่มีสิทธิ์หรือ permission ในการแก้ไข ก็ต้องใช้ sudo นำหน้าไฟล์เพื่อลบด้วยสิทธิ์ของ super user อีกกรณีที่ใช้บ่อยคือ พารามิเตอร์ -Rf เพื่อลบไดเร็กตอรี่พร้อมไฟล์ด้านในทั้งหมด

<u>ตัวอย่างที่ 4-6</u>

pi@raspberrypi ~/Exa pi@raspberrypi ~ \$ r pi@raspberrypi ~ \$ l	mples \$ cd m -Rf Examp s	 les	
Adafruit_Python_DHT	Downloads	Pictures	simpletest.py
Desktop	indiecity	Public	Templates
Documents	Music	python_games	Videos

แสดงการใช้คำสั่ง **rm** ที่ต่อท้ายด้วยพารามิเตอร์ **-**R**f** สำหรับลบไดเร็กตอรี่และไฟล์ภายในทั้งหมด

4.6.7 cp (Copy)

เป็นคำสั่งคัดลอกไฟล์ที่ระบุไปยังที่อยู่ปลายทาง ใช้ร่วมกับอักงระช่วยเหลือทั้ง ? และ * ได้ นอกจากนั้นยังรองรับการใส่พารามิเตอร์ต่อท้ายได้หลายแบบ (รายละเอียคให้ดูจาก man cp)

pi@raspberrypi ~/Desktop \$ ls geany.desktop test.py pi@raspberrypi ~/Desktop \$ cp test.py/Examples
<u>ตัวอย่างที่ 4-7</u>
pi@raspberrypi ~/Adafruit_Python_DHT/examples \$ ls AdafruitDHT.py google_spreadsheet.py simpletest.py pi@raspberrypi ~/Adafruit_Python_DHT/examples \$ cp *.py ~/Examples pi@raspberrypi ~/Adafruit_Python_DHT/examples \$ ls ~/Examples AdafruitDHT.py google_spreadsheet.py simpletest.py

ใช้อักขระช่วยเหลือ * เพื่อคัดลอกไฟล์ชื่อใดก็ได้ที่มี .py ต่อท้าย

4.6.8 mv (Move)

เป็นกำสั่งย้ายไฟล์ที่ระบุไปยังปลายทาง คล้ายกับกำสั่ง cp แต่จะลบไฟล์ต้นทางด้วย

<u>รูปแบบ</u>

1. ในกรณีที่ต้องการย้ายไฟล์ไปยังไดเร็กตอรี่ปลายทางเท่านั้น

mv ไฟล์ที่ต้องการย้าย ไดเร็กตอรี่ปลายทาง

2. ในกรณีที่ต้องการย้ายไฟล์ไปยังไดเร็กตอรี่ปลายทาง และเปลี่ยนชื่อไฟล์

mv ไฟล์ที่ต้องการย้าย ไดเร็กตอรี่ปลายทาง/กำหนดชื่อไฟล์ปลายทางที่เกิด ขึ้นหลังการย้ายไฟล์

<u>ตัวอย่างที่ 4-8</u>

```
pi@raspberrypi ~/Examples $ ls
AdafruitDHT.py google_spreadsheet.py simpletest.py test.py
pi@raspberrypi ~/Examples $ mv simpletest.py ..
pi@raspberrypi ~/Examples $ ls
AdafruitDHT.py google_spreadsheet.py test.py
pi@raspberrypi ~/Examples $ ls ../*.py
../simpletest.py
```

4.6.9 cat (Concatenation)

เป็นกำสั่งแสดงข้อมูลที่อยู่ในไฟล์นั้นๆ ในรูปของแฟ้มข้อความ หรือ Plain Text

<u>รูปแบบ</u>

cat filenames

โดยที่

filenames - ไฟล์ที่ต้องการให้แสดงข้อมูล

<u>ตัวอย่างที่ 4-9</u>



เป็นการกำหนดให้แสดงข้อมูลของไฟล์ sample.py ในรูปของข้อความ เมื่อป้อนคำสั่งแล้วกด Enter บอร์ด Raspberry Pi แสดงข้อมูลของไฟล์ในบรรทัดต่อๆ มา จนครบ แล้วกลับมาที่เซลพร้อมพ์ นอกจากนั้น คำสั่งนี้ยังนำข้อมูลที่ได้ไปรวมกับอีกไฟล์ได้ โดยข้อมูลจะอยู่ต่อท้ายจากของเดิม ที่มีอยู่ในไฟล์นั้นๆ และถ้าไฟล์ปลายทางยังไม่มี ก็จะสร้างไฟล์ขึ้นมาให้ทันที ดังตัวอย่างที่ 4-11

<u>ตัวอย่างที่ 4-10</u>

```
pi@raspberrypi ~/Desktop/rpi $ ls -1
total 8
-rwxrw-r-- 1 pi users 105 Apr 14 00:29 sample.py
-rw-r--r-- 1 pi pi 50 Apr 14 00:38 test.py
pi@raspberrypi ~/Desktop/rpi $ cat test py
#Hello Raspberry Pi
#This is test file for python
pi@raspberrypi ~/Desktop/rpi $ cat sample.py >> test.py
pi@raspberrypi ~/Desktop/rpi $ cat test.py
#Hello Raspberry Pi
#This is test file for python
import RPi.GPIO
import timer
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
print "Hello Raspberry Pi"
pi@raspberrypi ~/Desktop/rpi $
```

ในตัวอย่างนี้เป็นการนำข้อมูลของไฟล์ sample.py ไปต่อท้ายข้อมูลในไฟล์ test.py โดยในตอน เริ่มต้นใช้คำสั่ง **cat test.py** เพื่อแสดงให้เห็นข้อมูลในไฟล์ test.py ก่อน ซึ่งมี 2 บรรทัดคือ

```
#Hello Raspberry Pi
#This is test file for python
```

จากนั้นรันคำสั่ง **cat sample.py >> test.py** เพื่อนำข้อมูลจากไฟล์ sample.py ซึ่งมีรายละเอียดแสดงในตัวอย่างที่ 4-9 ไปต่อท้ายข้อมูลในไฟล์ test.py

ถัดจากนั้น รันคำสั่ง **cat test.py** อีกครั้ง เพื่อแสดงข้อมูลในไฟล์ test.py หลังจากที่มีการนำ ข้อมูลจากไฟล์ sample.py มาต่อท้าย จะเห็นข้อมูลเพิ่มขึ้นอีก 5 บรรทัดดังนี้

```
#Hello Raspberry Pi
#This is test file for python
import Rpi.GPIO
import timer
GPIO.setwaring(False)
GPIO.setmode(GPIO.BCM)
print "Hello Raspberry Pi"
แล้วกลับมาที่เซลพร้อมพ์
```

4.6.10 echo

เป็นคำสั่งส่งผ่านข้อความที่พิมพ์ไปยังเอาต์พุต ถ้าไม่ระบุปลายทางจะเป็นการแสดงออกมาที่ เทอร์มินอล

ถ้าระบุเอาต์พุตด้วย > และต่อท้ายด้วยชื่อ ไฟล์ จะเป็นการเขียนข้อความลง ไฟล์นั้น ถ้ามี ไฟล์
 เดิมจะถูกลบและเขียนข้อความใหม่ทับ



ถ้าต้องการเขียนเพิ่มต่อท้ายไฟล์ จะใช้พารามิเตอร์ >> แทน



4.6.11 pwd (Print Working Directory)

เป็นกำสั่งแสดงตำแหน่งของใดเร็กตอรี่ปัจจุบัน

pi@raspberrypi ~/Desktop \$ pwd /home/pi/Desktop

4.6.12 hostname

เป็นกำสั่งแสคงชื่อเครื่องปัจจุบันที่ทำงาน

pi@raspberrypi ~/Desktop \$ hostname raspberrypi

4.6.13 df (Disk Free)

เป็นกำสั่งสำหรับดูการใช้พื้นที่ว่างสำหรับเก็บข้อมูลใน microSD การ์ด

pi@raspberrypi	~/Desktop	\$ df			
Filesystem	1K-blocks	Used	Available	Use%	Mounted on
/dev/root	7319248	3978640	2992976	58%	/
devtmpfs	372316	0	372316	0%	/dev
tmpfs	75328	236	75092	1%	/run
tmpfs	5120	0	5120	0%	/run/lock

4.6.14 date

เป็นกำสั่งแสดงหรือตั้งก่าวันเวลาของเกรื่อง

- ถ้าพิมพ์ date จะเป็นการแสดงเวลาของระบบออกมา
- ถ้าต้องการตั้งก่าวันเวลา จะต้องใช้พารามิเตอร์-s ตามด้วยข้อความวันเวลา ดังตัวอย่าง และต้องใช้สิทธิ์ super user ด้วย

<u>ตัวอย่างที่ 4-11</u>

pi@raspberrypi ~/Desktop \$ date Sun Aug 30 18:09:38 ICT 2015 pi@raspberrypi ~/Desktop \$ date -s "Sun Aug 30 18:08:38 ICT 2015" date: cannot set date: Operation not permitted Sun Aug 30 18:08:38 ICT 2015 pi@raspberrypi ~/Desktop \$ sudo date -s "Sun Aug 30 18:08:38 ICT 2015" Sun Aug 30 18:08:38 ICT 2015

4.6.15 chmod (Change file Mode bit)

เป็นคำสั่งเปลี่ยนโหมดไฟล์และไดเร็กตอรี่ ใช้ในการแก้ไขสิทธิ์ในการเข้าถึงไฟล์หรือ permission การแก้ไข permission ด้วยคำสั่ง chmod นิยมทำสองวิธีหลักๆ คือ

1. อ้างอิงแบบ u g o a (user, group, other, all permission) +- (เปิด,ปิด)

r w x (read,write,execute)

ls -l output.txt
19 Aug 30 20:51 output.txt
chmod u+rwx output.txt
ls -l output.txt
19 Aug 30 20:51 output.txt
chmod go+rx output.txt
ls -l output.txt
19 Aug 30 20:51 output.txt

 2. อีกวิธีหนึ่งที่ยืดหยุ่นกว่าและนิยมใช้มากกว่าคือ ระบุ permission rwx 3 บิต 3 ชุดเรียงกัน เขียนในรูปแบบเลขฐานแปด เช่น rwx r-x r-x เรียกเป็น 755

ls -l output.txt
19 Aug 30 20:51 output.txt
chmod 755 output.txt
ls -l output.txt
19 Aug 30 20:51 output.txt

แต่ถ้าไม่มีสิทธิ์ในการแก้ไข ก็ต้องใช้คำสั่ง **sudo** นำหน้าไฟล์เพื่อตั้งค่าด้วยสิทธิ์ของ super user

82 • Raspberry Pi 3 : รู้จักแล:การใช้งานเบื้องตัน

4.6.16 chgrp (Change Group ownership)

เป็นกำสั่งเปลี่ยนกลุ่มไฟล์และ ใคเร็กตอรี่

ร<u>ูปแบบคำสั่ง</u> chgrp ชื่อกลุ่ม ชื่อไฟล์หรือไดเร็กตอรี่ที่ต้องการเปลี่ยน

> pi@raspberrypi ~ \$ ls -l output.txt -rwxr-xr-x 1 pi pi 19 Aug 30 20:51 output.txt pi@raspberrypi ~ \$ sudo chgrp root output.txt pi@raspberrypi ~ \$ ls -l output.txt -rwxr-xr-x 1 pi root 19 Aug 30 20:51 output.txt

4.6.17 chown (Change file Owner and Group)

เป็นกำสั่งเปลี่ยนเจ้าของไฟล์และ ใคเร็กตอรี่

<u>รูปแบบคำสั่ง</u>

chown ชื่อผู้ใช้ ชื่อไฟล์หรือไดเร็กตอรี่ที่ต้องการเปลี่ยน

pi@raspberrypi ~ \$ ls -l output.txt -rwxr-xr-x 1 pi pi 19 Aug 30 20:51 output.txt pi@raspberrypi ~ \$ sudo chown root output.txt pi@raspberrypi ~ \$ ls -l output.txt -rwxr-xr-x 1 root pi 19 Aug 30 20:51 output.txt

หากต้องการเปลี่ยนทั้ง owner และ group ในคราวเดียวกันจะใช้ owner:group ดังนี้

pi@raspberrypi 🏾 \$	ls -l output.txt
-rwxr-xr-x 1 pi pi	19 Aug 30 20:51 output.txt
pi@raspberrypi 🍧 💲	sudo chown root:root output.txt
pi@raspberrypi 🌷 💲	ls -l output.txt
-rwxr-xr-x 1 root r	oot 19 Aug 30 20:51 output.txt

4.6.18 grep (Print line matching a pattern)

เป็นคำสั่งสำหรับตรวจและแสดงข้อความที่ค้นหา ปกติจะใช้ร่วมกับคำสั่งที่ส่งข้อความผ่าน ไปป์ | มาอีกทอดหนึ่ง (| คืออักขระที่เป็นคีย์บนของคีย์ \ ไม่ใช่ตัวอักษรแอลตัวพิมพ์เล็ก)

<u>ตัวอย่างที่ 4-12</u>

ในตัวอย่างนี้ต้องการค้นหาข้อความ output ผลการค้นหาคือ พบไฟล์ output.txt

4.6.19 ps (Process report snapshot)

เป็นกำสั่งดูกระบวนการหรือ process ของกำสั่งต่างๆ ที่ทำงานในระบบ ณ เวลานั้น

ปกติมักจะต่อท้ายด้วยพารามิเตอร์ **ax** เพื่อดูกำสั่งที่ทำงานอยู่เบื้องหลังด้วย และเพื่อความ สะดวกในการไล่ดู จะใช้ร่วมกับ | **grep** เพื่อแยกเอาเฉพาะ process ที่สนใจมาแสดง

<u>ตัวอย่างที่ 4-13</u>



ในตัวอย่างนี้ ต้องการดู process ของ python3 ที่ทำงานอยู่

84 • Raspberry Pi 3 : รู้จักและการใช้งานเบื้องตัน

4.6.20 kill (kill process)

เป็นคำสั่งยกเลิกการทำงานของ process ID ที่ระบุ

ถ้าหากเป็น process ที่สามารถหยุคได้ด้วยสัญญาณ SIGTERM ปกติ เพียงเรียกคำสั่ง

sudo kill ตามด้วยเลข process ID

กระบวนการหรือ process ที่ระบุนั้นจะหยุดทำงานทันที

แต่ถ้าเป็น process ที่ไม่รับสัญญาณ SIGTERM จะเพิ่มพารามิเตอร์ **- 9** นำหน้าเลข process ID เพื่อบังคับให้หยุด ดังรูป

pi@raspberrypi	ິ \$	ps ax grep python3
2814 pts/1	S+	0:00 sudo <mark>python3</mark> test.py
2815 pts/1	S+	0:00 python3 test.py
2837 pts/0	S+	0:00 grepcolor=auto python3
pi@raspberrypi	<u> </u>	sudo kill -9 2815
p1@raspberryp1	<u>چ</u>	ps ax grep python3
2842 pts/0	К+	0:00 grepcolor=auto python3

4.6.21 passwd (change password)

เป็นคำสั่งเปลี่ยนรหัสผ่านของผู้ใช้ปัจจุบัน เมื่อรันคำสั่งจะถามรหัสผ่านเดิมก่อน จากนั้นจึง พิมพ์รหัสผ่านใหม่สองครั้งเพื่อยืนยันการแก้ไข

pi@raspberrypi 🌷 💲 passwd
Changing password for pi.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully

4.6.22 clear

เป็นกำสั่งถ้างข้อความบนหน้าจอเทอร์มินอล

<u>ตัวอย่างที่ 4-14</u>

หน้าจอเทอร์มินอลก่อนรันคำสั่ง clear

drwxr-xr-x	2 pi	pi	4096	Aug 29	10:12	Templa	tes	
drwxr-xr-x	3 pi	pi	4096	Aug 29	05:26	. theme	S	
drwx	4 pi	pi	4096	Aug 29	09:22	. thumb	nails	
drwxr-xr-x	2 pi	pi	4096	Aug 29	10:12	Videos		
drwx	2 pi	pi	4096	Aug 30	00:00	.vnc		
-rw	1 pi	pi	57	Aug 30	21:25	.Xauth	ority	
-rw	1 pi	pi	46742	Aug 30	00:00	.xsess	ion-errors	
pi@raspberr	vpi 🌷	\$ ls						
Adafruit_Py	thon_D	HT Doci	ments in	diecity	outpu	ut.txt	Public	Templates
Desktop		Dowr	nloads Mu	sic	Pictu	unes	python_games	Videos
pi@raspberrypi ~ \$ clear_								

หน้าจอเทอร์มินอลหลังรันคำสั่ง clear

pi@raspberrypi 🌷 💲		

4.6.23 ifconfig (Network Interface Config)

เป็นคำสั่งตรวจสอบการต่อเชื่อมเครือข่าย

<u>ตัวอย่างที่ 4-15</u>

pi@raspber	rypi 👕 💲 ifconfig
eth0	Link encap:Ethernet HWaddr b8:27:eb:07:b4:e0
	inet addr:192.168.1.44 Bcast:192.168.1.255 Mask:255.255.255.0
	UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
	RX packets:25456 errors:0 dropped:3 overruns:0 frame:0
	TX packets:3749 errors:0 dropped:0 overruns:0 carrier:0
	collisions:0 txqueuelen:1000
	RX bytes:1291062 (1.2 MiB) TX bytes:477509 (466.3 KiB)
lo	Link encap:Local Loopback
	inet addr:127.0.0.1 Mask:255.0.0.0
	UP LOOPBACK RUNNING MTU:65536 Metric:1
	RX packets:0 errors:0 dropped:0 overruns:0 frame:0
	TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
	collisions: A txqueuelen: A
	BX butes:0 (0,0,B) TX butes:0 (0,0,B)

จากตัวอย่างแสดงให้เห็นว่า บอร์ด Raspberry Pi เชื่อมต่อกับเครือข่ายผ่านพอร์ตอีเธอร์เน็ต

(LAN) ทางช่องทาง eth0

4.6.24 ping

เป็นกำสั่งตรวจสอบการตอบสนองของ IP หรือ hostname ที่ระบุ

ถ้าไม่ใช้พารามิเตอร์ – c ตามด้วยจำนวนครั้งที่ต้องการ ping โปรแกรมจะทำการ ping อย่าง ต่อเนื่อง หากต้องการจบการทำงาน ให้กดกีย์ Ctrl และ C

pi@raspberryp:	~ \$ ping 192.168.1.1
PING 192.168.1	1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from	192.168.1.1: icmp_req=1 ttl=64 time=0.866 ms
64 bytes from	192.168.1.1: icmp_req=2 ttl=64 time=0.663 ms
64 bytes from	192.168.1.1: icmp_req=3 ttl=64 time=0.668 ms
64 bytes from	192.168.1.1: icmp_req=4 ttl=64 time=0.639 ms
64 bytes from	192.168.1.1: icmp_req=5 ttl=64 time=0.624 ms
^Z	
[3]+ Stopped	ping 192.168.1.1

4.6.25 nano

เป็นคำสั่งเรียกโปรแกรมเท็กซ์เอดิเตอร์ที่ชื่อว่า **nano** ขึ้นมาทำงาน โดย nano เป็นโปรแกรม อำนวยความสะดวกสำหรับแก้ไขไฟล์ข้อความ

ร<u>ูปแบบคำสั่ง</u> nano ไฟล์ที่ต้องการแก้ไข เมื่อรัน จะเห็นหน้าโปรแกรมดังรูป



4.6.26 apt-get (Advance Package Tools : get)

เป็นกำสั่งจัดการแพ็กเกจภายในเครื่อง ช่วยในก้นหา ติดตั้ง และอัปเดตแพ็กเกจให้สะดวกขึ้น กวรใช้ควบกู่กับกำสั่ง sudo

<u>รูปแบบคำสั่ง</u>

ที่ใช้งานบ่อยคือ

sudo apt-get install ชื่อแพ็กเก็จที่ต้องการติดตั้งเพิ่ม sudo apt-get remove ชื่อแพ็กเก็จที่ต้องการถอนออกจากระบบ

หากต้องการตรวจสอบการปรับเวอร์ชั่นแพ็กเกจโปรแกรมต่างๆที่มีอยู่ทั้งหมด จะใช้

sudo apt-get update

หลังจากตรวจสอบแล้ว ถ้าต้องการปรับเวอร์ชั่นใหม่ จะใช้คำสั่งนี้ตามหลัง

sudo apt-get upgrade

4.2.27 wget

เป็นกำสั่งเรียกใช้งานตัวช่วยเหลือในการคาวน์โหลคไฟล์จากอินเทอร์เนีต

<u>รูปแบบ</u>

wget URL

โดยที่

URL - ตำแหน่งเว็บไซต์ของไฟล์ที่ต้องการดาวน์โหลด

<u>ตัวอย่างที่ 4-16</u>



จากตัวอย่างเป็นการสั่งให้บอร์ด Raspberry Pi ดาวน์โหลดไฟล์ sample_music.mp3 จากเว็บไซต์ sleepingforless.com โดยมีการแสดงข้อมูลของไฟล์ที่ทำการดาวน์โหลดด้วย เมื่อการ ดาวน์โหลดเสร็จสิ้นจะกลับมายังเชลพร้อมพ์ 88 • Raspberry Pi 3 : รู้จักแล:การใช้งานเบื้องตัน

4.6.28 reboot

เป็นคำสั่งรีบูตระบบใหม่

<u>รูปแบบคำสั่ง</u> sudo reboot

4.6.29 poweroff หรือ halt หรือ shutdown

เป็นคำสั่งปิดระบบ

<u>รูปแบบคำสั่ง</u>

sudo poweroff

หรือ

sudo halt

หรือ

sudo shutdown เวลา โดยที่ค่าเวลามีหน่วยเป็นวินาที

<u>ตัวอย่างที่ 4-17</u> sudo shutdown 60 เป็นการสั่งปิดการระบภายใน 60 วินาที

<u>ตัวอย่างที่ 4-18</u> sudo poweroff เป็นคำสั่งปิดการทำงานของระบบทันที

sudo halt เป็นคำสั่งปิดการทำงานของระบบทันที

sudo shutdown now เป็นคำสั่งปิดการทำงานของระบบทันที

ข้อมูลคำสั่งของระบบปฏิบัติการ Linux ยังมีอีกมากมาย ทั้งหมดที่นำเสนอในที่นี้เป็นเพียงส่วน หนึ่งที่มีการ ใช้งานบ่อยๆ กับบอร์ดคอมพิวเตอร์ Raspberry Pi เท่านั้น



^{บทที่ 5} เริ่มต้นรู้จักกับภาษา Python



ในการพัฒนาโปรแกรมสำหรับ Raspberry Pi 3 ทำได้หลากหลายภาษา ทั้ง C, BASIC, JAVA, Pascal, Assembly หรือ Python ซึ่งเหมือนกับคอมพิวเตอร์ที่ใช้งานทั่วไป เพราะ Raspberry Pi 3 ก็คือ บอร์คคอมพิวเตอร์ที่ติดตั้งระบบปฏิบัติการ Raspbian อันเป็นหนึ่งในระบบปฏิบัติการ Linux ที่มีการ ปรับให้ทำงานบนบอร์ค Raspberry Pi ได้อย่างมีประสิทธิภาพ

ภาษาที่นิยมใช้ในการพัฒนาโปรแกรมบน Raspberry Pi คือ Python และมีตัวอย่างโปรแกรม ให้ดาวน์โหลดมาเรียนรู้และใช้งานมากมายในเครือข่ายอินเทอร์เน็ต ในบทนี้นำเสนอเนื้อหาอันเป็น การแนะนำเกี่ยวกับ Python เบื้องต้น

5.1 Python คืออะไร?

Python เป็นภาษาโปรแกรมระคับสูงที่ถือกำเนิดขึ้นเมื่อปี ค.ศ. 1990 โดย Guido van Rossum ซึ่งเป็นโปรแกรมเมอร์ชาวดัตช์หรือเนเธอร์แลนด์ *โดยชื่อ Python มาจากภาพยนต์ซีรีส์ของสถานี โทรทัศน์ BBC ที่ชื่อว่า Monty Python's Flying Circus* ในปัจจุบันนี้ภาษา Python อยู่ภายใต้การดูแล ของมูลนิธิซอฟต์แวร์ Python



รูปที่ 5-2 Guido van Rossum บิดา แห่ง Python







รูปที่ 5–3 แสดงสัดส่วนของจำนวนการ ใช้งานและพัฒนาโปรแกรมด้วยภาษา คอมพิวเตอร์ จะเห็นว่า ภาษา Python ได้รับความนิยมสูงสุด Python เป็นอีกหนึ่งภาษาโปรแกรมคอมพิวเตอร์ที่ได้รับความนิยมสูงมากในปัจจุบัน เนื่องจาก ตัวภาษาทำความเข้าใจได้ไม่ยาก มี Syntax ที่ไม่ยุ่งยากเมื่อเปรียบเทียบกับภาษาอื่น จึงทำให้เรียนรู้ได้ ง่าย และมีการทำงานแบบอินเตอร์พรีตเตอร์ (Interpreter) หรือแบบแปลภาษา

5.2 เริ่มใช้งาน Python บน Raspberry Pi

ในการใช้งาน Python บน Raspberry Pi ทำได้ 2 แบบคือ

 เชลสคริปต์ (Shell Script) โดยเขียนชุดคำสั่งไว้ในไฟล์ใดๆ แล้วสั่งให้ Python ทำการเอ็กซิคิวต์ ไฟล์นั้นๆ

2. อินเตอร์แอกตีฟเชล (Interactive Shell) ที่ให้ผู้พัฒนาโปรแกรมพิมพ์กำสั่งลงในเทอร์มินอลได้ ทันที แล้ว Python จะทำงานทันทีในทุกๆ บรรทัดที่พิมพ์ เหมาะแก่การฝึกเขียนโปรแกรม Python ได้ เป็นอย่างดี

5.2.1 การใช้งานเชลสคริปต์ของ Python บน Raspberry Pi

หลักการของเชลสคริปต์คือ สร้างไฟล์ที่มีชุดคำสั่งที่พิมพ์เตรียมไว้แล้ว แล้วสั่งให้ชุดคำสั่งที่ อยู่ในไฟล์นั้นทำงาน

```
แนวทางการใช้งานเชลสคริปต์มีดังนี้
```

(1) สร้างไฟล์ขึ้นมาหนึ่งไฟล์ โดยใช้ Nano ในการสร้างไฟล์ แล้วใส่คำสั่งไว้ข้างใน โดยพิมพ์ คำสั่งว่า sudo nano mycode.py



(2) ที่ข้าง Nano Text Editor ให้เพิ่มคำสั่ง print ("Python is very cool").



(3) ทำการปิดโปรแกรม Nano ลง พร้อมกับบันทึกไฟล์ดังกล่าวไว้ด้วย (กด Ctrl และ X ตามด้วย Y แล้วกด Enter) (4) วิธีการสั่งให้ไฟล์ดังกล่าวทำงาน จะใช้คำสั่งว่า python แล้วตามด้วยชื่อไฟล์นั้นๆ ดังนั้น คำสั่งจึงเป็น python mycode.py

pi@raspberrypi ~ \$ python3 mycode.py

(5) เมื่อกด Enter จะเห็นข้อความ Python is very cool แสดงขึ้นมา ซึ่งเกิดจากกำสั่ง ที่สร้างไว้ในไฟล์ mycode.py นั่นเอง

```
pi@raspberrypi ~ $ python3 mycode.py
Python is very cool
pi@raspberrypi ~ $
```

5.2.2 การใช้งานอินเตอร์แอกตีฟเชลของ Python บน Raspberry Pi 3

(1) เข้าสู่หน้าต่างเทอร์มินอล แล้วรันคำสั่ง **python** เพื่อเรียกอินเตอร์แอกตีฟเชลขึ้นมา

pi@raspberrypi ~ \$ python3

(2) เมื่อกด Enter อินเตอร์แอกตีฟเชลของ Python เริ่มทำงาน และหยุดที่บรรทัด >>> ซึ่งบรรทัด ดังกล่าวนี้กือ ตำแหน่งสำหรับพิมพ์กำสั่งของโปรแกรม

```
pi@raspberrypi ~ $ python3
Python 3.2.3 (default, Mar 1 2013, 11:53:50)
[GCC 4.6.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

(3) ให้พิมพ์กำสั่ง print ("Python is very cool")

```
pi@raspberrypi ~ $ python3
Python 3.2.3 (default, Mar 1 2013, 11:53:50)
[GCC 4.6.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> print ("Python is very cool")
```

(4) เมื่อกด Enter จะเห็นว่า ที่บรรทัดถัดไปมีการแสดงข้อความว่า Python is very cool

```
pi@raspberrypi ~ $ python3
Python 3.2.3 (default, Mar 1 2013, 11:53:50)
[GCC 4.6.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> print ("Python is very cool")
Python is very cool
>>>
```

ข้อความที่แสดงนั้นเกิดมาจากการใช้คำสั่ง print หากต้องการให้แสดงข้อความใด ข้อความ นั้นๆ จะถูกแสดงในบรรทัดถัดไป นี่คือรูปแบบการทำงานของอินเตอร์แอกตีฟเชลนั่นเอง เมื่อทำงาน เสร็จแล้วก็จะแสดงสัญลักษณ์ >>> เพื่อให้พิมพ์คำสั่งต่อไปได้ทันที 92 • Raspberry Pi 3 : รู้จักและการใช้งานเบื้องตัน

5.3 คำสั่งprint

เป็นกำสั่งใช้แสดงข้อมูลหรือข้อกวามใดๆ บนหน้าจอแสดงผล นับเป็นหนึ่งในกำสั่งพื้นฐาน ที่สำคัญของ Python

<u>รูปแบบ</u> print (ข้อมูลหรือข้อความ) <u>ตัวอย่างที่ 5-1</u> >>> print ("Innovative Experiment") แลที่ได้ · Innovative Experiment >>> print ('Innovative Experiment') ผลที่ได้ : Innovative Experiment >>> print ("Hi! I'm R-Pi lover") ผลที่ได้ : Hi! I'm R-Pi lover >>> print ('Hi! I\'m R-Pi lover') ผลที่ได้: Hi! I'm R-Pi lover >>> print ("""This is ... new line ... command""") ผลที่ได้ : This is new line command >>> print ("This is\nnew line\ncommand") ผลที่ได้ : This is new line command

5.4 เครื่องหมายพิเศษบางส่วนใน Python

<u>รูปแบบ</u>	<u>ผลลัพธ์</u>
- \'	N
$\backslash ''$	w
١١	λ
\a	มีเสียงดังหนึ่งครั้ง
\f	ขึ้นแถวใหม่
	>>> print ("AA\fAA") AA AA
\n	ขึ้นบรรทัดใหม่และชิดซ้ายมือ
	>>> print ("AA\nAA") AA AA
\t	แท็บหนึ่งครั้ง

🥐 python Note

สำหรับการพัฒนาแอปพลิเคชั่นให้แก่บอร์ด Raspberry Pi ด้วยโปรแกรมภาษา Python เป็น สิ่งที่ได้รับความนิยมสูงมาก มีโค้ดตัวอย่างจำนวนไม่น้อยในเว็บไซต์พัฒนาขึ้นด้วย Python2 ในขณะที่ปัจจุบัน (ในขณะที่ทำหนังสือเล่มนี้) นักพัฒนาโปรแกรมส่วนใหญ่เริ่มปรับมาใช้ Python3 แล้ว โดย Python3 ได้รับการพัฒนาขึ้นมาใหม่และไม่คอมแพตติเบิลหรือไม่เข้ากัน กับPython2 ที่มีอยู่เดิม ดังนั้น หากนำโค้ดที่เขียนขึ้นจาก Python2 มาใช้งานกับ Python3 จะต้องตรวจสอบความเข้ากันคำสั่งเสียก่อน



94 • Raspberry Pi 3 : รู้จักแล:การใช้งานเบื้องตัน

5.5 ตัวแปร (Variables)

ตัวแปรหรือ Variable เป็นชื่อเรียกที่เก็บข้อมูลเพื่อใช้ในโปรแกรม ผู้พัฒนาสามารถตั้งชื่อตัว แปรได้ โดยมีเงื่อนไขดังนี้

- ใช้ตัวอักษร a ถึง z และ A ถึง Z
- ใช้เครื่องหมาย _ (Underscore) ได้เท่านั้น ห้ามใช้เครื่องหมายพิเศษอื่นๆ
- ใช้ตัวเลข 0 ถึง 9
- ตัวอักษรตัวแรกของชื่อตัวแปร ห้ามใช้ตัวเลข
- ห้ามตั้งชื่อตรงกับคำสงวน

5.5.1 คำสงวน

ประกอบด้วย false, none, true, and, as, assert, break, class, continue, def, del, elif, else, except, finally, for, from, global, if, not, or, pass, raise, return, try, while, with และ yield

ทคสอบได้ด้วยการพิมพ์กำสั่ง

```
>>> import keyword
>>> print (keyword.kwlist)
ผลที่ได้
```

```
['False', 'None', 'True', 'and', 'as', 'assert', 'break',
'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally',
'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda',
'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while',
'with', 'yield']
```

5.5.2 ตัวอย่างการสร้างตัวแปร

<u>ตัวอย่างที่ 5-2</u>

>>> age = 25 >>> print (age) ผลที่ได้: **25**

จากตัวอย่างเป็นการสร้างตัวแปรที่มีชื่อว่า age และเก็บข้อมูลไว้เป็นเลข 25 ดังนั้นเมื่อสั่ง ให้แสดงข้อมูลที่อยู่ในตัวแปร age ก็จะแสดงค่า 25 ออกมา

<u>ตัวอย่างที่ 5-3</u>

```
>>> message = "Hi everyone."
>>> print (message + "I'm python developer.")
ผลที่ได้: Hi everyone. I'm python developer.
```

<u>ตัวอย่างที่ 5-4</u>

```
>>> message1 = "Hi everyone."
>>> message2 = "I'm Python developer."
>>> message3 = "Nice to meet you."
>>> print (message1 + message2 + message3)
µanlo::
```

Hi everyone.I'm Python developer.Nice to meet you. จะเห็นว่า ด้วยการใช้คำสั่งแบบนี้ เมื่อแสดงผลจะ<u>ไม่มีเว้นวรรค</u>ระหว่างข้อความ

<u>ตัวอย่างที่ 5-5</u>

```
>>> message1 = "Hi everyone."
>>> message2 = "I'm Python developer."
>>> message3 = "Nice to meet you."
>>> print (message1, message2, message3)
ผลที่ได้:
```

```
Hi everyone. I'm Python developer. Nice to meet you.
จะเห็นว่า ด้วยการใช้คำสั่งแบบนี้ เมื่อแสดงผลจะ<u>เว้นวรรค</u>ระหว่างข้อความ
```

<u>ตัวอย่างที่ 5-6</u>

```
>>> width = 10.56
>>> height = 3.38
>>> area = width * height
>>> print (area)
ผลที่ได้ · 35 6928
```

เป็นการคูณค่าระหว่างตัวแปร width และ height แล้วเก็บไว้ในตัวแปร area ถ้าสังเกต นี่คือ การเขียนคำสั่งเพื่อคำนวณหาพื้นที่

5.5.3 ประเภทของตัวแปร (Primitive Data Types)

ตัวแปรในภาษา Python มี 4 ประเภทคือ

1. float (Floating-point number) : เลขทศนิยม มีค่า

2. int (integer) : เลขจำนวนเต็ม มีค่า -2³¹ ถึง 2³¹-1

3. long (Long integer) : เลขจำนวนเต็มขนาดใหญ่ (เก็บข้อมูลตัวเลขได้มากกว่า int) มีค่าไม่จำกัด

4. st (Character string) : ข้อความตัวอักษร

5.5.4 วิธีตรวจสอบประเภทของตัวแปรใดๆ

เนื่องจากภาษา Python ไม่จำเป็นต้องประกาศประเภทของตัวแปร แต่จะถูกกำหนดจากข้อมูล (ประเภทตัวแปรสามารถเปลี่ยนแปลงได้ตลอดเวลา) ดังนั้นหากต้องการทราบว่า ตัวแปรนั้นๆ เป็น ประเภทใด จะต้องใช้กำสั่ง **type** เพื่อตรวจสอบ

<u>ตัวอย่างที่ 5-7</u>

```
>>> money = 2500
>>> type (money)
ผลที่ได้:
```

<type 'int'>

แสดงประเภทหรือชนิดของตัวแปร money เป็นเลขจำนวนเต็ม

<u>ตัวอย่างที่ 5-8</u>

```
>>> height = 179.3
>>> type (height)
ผลที่ได้:
```

<type 'float'>

แสดงประเภทหรือชนิดของตัวแปร height เป็นเลขทศนิยม

<u>ตัวอย่างที่ 5-9</u>

```
>>> milli_sec = 31020000000
>>> type (milli_sec)
ผลที่ได้:
```

<type 'long'>

แสดงประเภทหรือชนิดของตัวแปร milli_sec เป็นเลขจำนวนเต็มขนาดใหญ่

<u>ตัวอย่างที่ 5-10</u>

```
>>> name = "INEX"
>>> type (name)
ผลที่ได้:
```

<type 'str'>

แสดงประเภทหรือชนิดของตัวแปร name เป็นข้อความ

ในกรณีที่ต้องการตรวจสอบว่า ตัวแปรใดๆ เป็นประเภทเดียวกับที่ต้องการตรวจสอบหรือไม่ จะต้องใช้กำสั่ง type ในลักษณะตามตัวอย่างต่อไปนี้

<u>ตัวอย่างที่ 5-11</u>

```
>>> numberOfDay = 301
>>> type (numberOfDay) is int
ผลที่ได้: True
```

เป็นการตรวจสอบว่า ค่าในตัวแปร numberOfDay เป็นตัวแปรแบบเลขจำนวนเต็ม (int) หรือไม่ จากการกำหนดค่าตัวแปร พบว่า ตัวแปร numberOfDay มีการกำหนดค่าเป็น 301 ดังนั้น จึงเป็น ตัวแปรชนิดเลขจำนวนเต็ม ผลลัพธ์ที่ได้จึงเป็นจริงหรือ True

<u>ตัวอย่างที่ 5-12</u>

```
>>> numberOfDay = 301
>>> type (numberOfDay) is str
ผลที่ได้: False
```

จากตัวอย่างนี้เป็นการตรวจสอบว่า ค่าในตัวแปร numberOfDay เป็นตัวแปรแบบข้อความ (st) หรือไม่ จากการกำหนดค่าตัวแปร numberOfDay พบว่า ตัวแปรนี้เป็นแบบเลขจำนวนเต็ม ดังนั้น ผลลัพธ์ที่ได้จึงเป็นเท็จหรือ False 98 • Raspberry Pi 3 : รู้จักและการใช้งานเบื้องตัน

5.6 การรับค่าจากคีย์บอร์ด (Keyboard Input)

นอกเหนือจากการกำหนดข้อมูลให้กับตัวแปรโดยตรงแล้ว ใน Python ยังกำหนดชนิดข้อมูล ให้กับข้อมูลที่รับมาจากกีย์บอร์ดได้ด้วย โดยใช้กำสั่ง input เก็บข้อมูลเป็น int, long หรือ float เท่านั้น

<u>ตัวอย่างที่ 5-13</u>

```
>>> age = input ("How old are you? : ")
How old are you : 31
>>> print (age)
31
>>> type (age)
<type 'int'>
```

เป็นการใช้คำสั่ง input ในการรับค่าตัวเลข แล้วนำไปเก็บไว้ในตัวแปร age แล้วทำการพิมพ์ ค่าที่ได้ออกมาและชนิดของตัวแปร

<u>ตัวอย่างที่ 5-14</u>

```
>>> weight = input ("Weight : ")
weight : 75.3
>>> print (weight)
75.3
>>> type (weight)
<type 'float'>
```

<u>ตัวอย่างที่ 5-15</u>

```
>>> height = input ("Height : ")
Height = 10
>>> width = input ("Widht : ")
Widht = 20
>>> print ("Area =", width * height)
```

ในกรณีที่ต้องการรับข้อมูลจากคีย์บอร์คเป็นแบบ str จะใช้คำสั่ง raw_input

<u>ตัวอย่างที่ 5-16</u>

```
>>> name = raw_input ("What's your name? : ")
What's your name? : Python
>>> print (name)
Python
>>> type (name)
<type="str">
```

จากตัวอย่างเป็นการรับข้อความเพื่อเก็บไว้ในตัวแปร name จากนั้นทำการพิมพ์แสดงข้อมูล ของตัวแปร name และชนิดของตัวแปร นั่นคือ str โดยมีข้อความ Python เก็บไว้ในตัวแปร name

5.7 เครื่องหมายคำนวณทางคณิตศาสตร์พื้นฐาน (Math Operators)

ประกอบด้วย

+	Addition	บวก
-	Subtraction	ลบ
*	Multiplication	คูณ
/	Division	หาร (ส่วน)
%	Modulus	หารเอาเศษ
//	Floor division	หาร ปัดเศษ
**	Exponentiation	ยกกำลัง

<u>ตัวอย่างที่ 5-17</u>

x += 4	มีค่าเท่ากับ	x = x + 4
x *= 3	มีค่าเท่ากับ	x = x * 3
x /= 6	มีค่าเท่ากับ	x = x / 6
x **= 2	มีค่าเท่ากับ	x = x ** 2
x %= 5	มีค่าเท่ากับ	x = x % 5

5.8 List IIa= Tuple

5.8.1 List

List เป็นตัวแปรที่อยู่ในรูปของอะเรย์ มีหน้าที่เก็บข้อมูลจำนวนหลายๆ ตัว

5.8.1.1 การสร้างตัวแปรแบบ List

การสร้างตัวแปรแบบ List ทำได้ง่ายๆ ดังตัวอย่าง

```
<u>ตัวอย่างที่ 5-18</u>

>>> my_list = [15, 12, 17, 13]

>>> print (my_list)
ผลลัพธ์ที่ได้ : [15, 12, 17, 13]

จากตัวอย่างนี้ ตัวแปร my_list มีสมาชิก 4 ตัวคือ 15, 12, 17 และ 13

ตัวอย่างที่ 5-19
```

```
>>> my_list = [15, 12, 17, 13]
>>> type (my_list)
```

ผลลัพธ์ที่ได้ : **<type 'list'>** จากตัวอย่างนี้เป็นการตรวจสอบชนิดของตัวแปร my_list ซึ่งได้ผลเป็นตัวแปรแบบ List

5.8.1.2 วิธีการเรียกใช้งานข้อมูลที่อยู่ใน List

ทำได้ด้วยการระบุตำแหน่งของสมาชิกในตัวแปรแบบ list โดยตำแหน่งของตัวแปรอาจเรียกว่า เลขดัชนี โดยมีก่าตั้งแต่ 0 เริ่มจากสมาชิกที่อยู่ซ้ายสุด

<u>ตัวอย่างที่ 5-20</u>

```
>>> data = [30, 15, 20, 60, 5, 100]
>>> print (data[3])
```

ผลลัพธ์ที่ได้ : **60** เป็นการสั่งให้แสดงค่าของสมาชิกลำดับ 3 ในตัวแปร _{data} ซึ่งเป็นตัวแปร แบบ list ซึ่งก็คือค่า 60 โดยลำดับ 0 คือ 30, ลำดับ 1 คือ 15, ลำดับ 2 คือ 20, ลำดับ 4 คือ 5 และลำดับ 5 สุดท้ายมีค่า 100

<u>ตัวอย่างที่ 5-21</u>

```
>>> names = ["Ivy", "Dan", "Thomas"]
>>> person = names[1]
>>> print (person)
```

ผลลัพธ์ที่ได้ : **Dan** จากตัวอย่างนี้เป็นการสั่งให้แสดงชื่อของสมาชิกลำดับ 1 ในตัวแปร _{names} ซึ่งเป็นตัวแปรแบบ str ซึ่งก็คือ ข้อความ Dan แล้วนำเก็บไว้ในตัวแปร _{person} จากนั้นแสดงค่าของ ตัวแปร _{person} ด้วยคำสั่ง _{print} **ข้อควรระวัง** - กรณีที่ระบุตำแหน่งของข้อมูลเกิน จะทำให้โปรแกรมทำงานผิดพลาดได้ ดังตัวอย่างต่อไปนี้

<u>ตัวอย่างที่ 5-22</u>

```
>>> numbers = [10, 20, 30]
>>> print (number[5])
ผลลัพธ์ที่ได้
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
Index Error: list index out of range
```

จากตัวอย่างแสดงให้เห็นการทำงานที่ผิดพลาด เนื่องจากกำหนดขอบเขตของตำแหน่งหรือเลข ดัชนีของตัวแปรเกินขอบเขตที่กำหนดไว้

5.8.1.3 การเรียกใช้งานข้อมูลหลายตัวพร้อมกัน

ผู้พัฒนาโปรแกรมสามารถเรียกใช้งานข้อมูลของตัวแปร List ได้หลายตัวพร้อมกัน มีรูปแบบเป็น

ชื่อตัวแปร[x : y]

โดยที่

X คือลำดับของข้อมูลตัวแรกสุดที่ต้องการเรียกใช้งาน

Y คือลำดับของข้อมูลที่ต้องการสิ้นสุด

<u>ตัวอย่างที่ 5-23</u>

```
>>> letters = ["Ant", "Bird", "Cat", "Dog", "Egg", "Fish"]
>>> new_letters = letters[1 : 5]
>>> print (new_letters]
ผลลัพธ์ที่ได้ : ['Bird', 'Cat', 'Dog', 'Egg']
```

จากตัวอย่างเป็นการเรียกข้อมูลในลำดับ 1 ไปจนถึงข้อมูลลำดับ 4 เนื่องจากกำหนดลำดับ ที่สิ้นสุดไว้ที่ลำดับ 5 ดังนั้นจากคำสั่งในตัวอย่างนี้จึงเรียกใช้งานข้อมูลลำดับ 1 ถึง 4 ของตัวแปร new_letters ในกรณีที่ระบุลำดับของข้อมูลที่ต้องการสิ้นสุดเกินจำนวนที่มีในตัวแปรชนิด List ตัวนั้นๆ โปรแกรมยังคงทำงานได้ปกติ แต่ข้อมูลที่ได้ก็จะเป็นข้อมูลเท่าที่มี เพราะไม่สามารถเรียกข้อมูลใน ลำดับที่เกินออกมาใช้งานได้

<u>ตัวอย่างที่ 5-24</u>

```
>>> letters = ["Ant", "Bird", "Cat", "Dog", "Egg", "Fish"]
>>> new_letters = letters[2 : 10]
>>> print (new_letters]
ผลลัพธ์ที่ได้ :
```

['Cat', 'Dog', 'Egg', 'Fish']

จากตัวอย่างนี้ ตัวแปร _{new_letters} เป็นตัวแปรชนิด List ที่มีสมาชิก 6 ตัว แต่มีการเรียก ข้อมูลของสมาชิกมาใช้งานตั้งแต่ลำดับ 2 ไปถึง 9 (เนื่องจากกำหนดขอบเขตสิ้นสุดไว้ที่ลำดับ 10) ซึ่งเกินจากจำนวนของสมาชิกที่มีอยู่ ดังนั้นในการแสดงผลจึงแสดงข้อมูลของสมาชิกเท่าที่มี

5.8.1.4 การแก้ไขข้อมูลที่อยู่ใน List

ด้วยความยึดหยุ่นของภาษา Python ทำให้การแก้ไขข้อมูลของสมาชิกในตัวแปรชนิด List ทำได้ง่ายมากเพียงระบุลำดับของสมาชิกแล้วกำหนดค่าหรือข้อมูลที่ต้องการแก้ไขข้อมูลของสมาชิก ในตัวแปรก็จะเปลี่ยนแปลงได้ทันที

<u>ตัวอย่างที่ 5-25</u>

```
>>> days = ["Monday", "Tuesday", "Saturday"]
>>> days[1] = "Friday"
>>> print (days)
ผลลัพธ์ที่ได้ :
```

['Monday', 'Friday', 'Saturday']

จากตัวอย่าง ต้องการแก้ไขข้อมูลในลำดับ 1 จากเดิมคือ Tuesday เปลี่ยนเป็น Friday ทำได้ ง่ายตามที่แสดงในบรรทัด >>> days[1] = "Friday" เลข 1 คือ ลำดับ 1 ของตัวแปร days ซึ่งก็คือ Tuesday เมื่อรันคำสั่ง days[1] = "Friday" แล้ว ค่าข้อมูลในลำดับ 1 จะเปลี่ยนเป็น Friday ตามที่แสดงด้วยคำสั่ง print

5.8.1.5 การรองรับข้อมูลต่างประเภทของตัวแปร List

ข้อมูลที่เก็บไว้ในตัวแปรชนิด List ไม่จำเป็นต้องเป็นประเภทเดียวกันก็ได้ เพราะ List นั้นรอง รับข้อมูลต่างประเภทได้ ดังตัวอย่างต่อไปนี้

<u>ตัวอย่างที่ 5-26</u>

```
>>> data = [162, 91.4, "Wording"]
>>> print (data)
ผลลัพธ์ที่ได้ :
```

```
[162, 91.4, 'Wording']
```

จากตัวอย่างนี้ ตัวแปร data ประกอบด้วยข้อมูลที่เป็นเลขจำนวนเต็ม (int) คือ 162, เลข ทศนิยม (float) คือ 91.4 และตัวอักษร (str) คือ Wording

หากต้องการตรวจสอบประเภทของข้อมูลแต่ละตัวในตัวแปรขนิด List ก็ทำได้ด้วยกำสั่ง type ดังตัวอย่างต่อไปนี้

<u>ตัวอย่างที่ 5-27</u>

```
>>> data = [162, 91.4, "Wording"]
>>> type (data[0])
ผลลัพธ์ที่ได้ :
```

<type 'int'>

จากตัวอย่างนี้ต้องการตรวจสอบประเภทของข้อมูลในลำดับ 0 ของตัวแปร data นั่นคือ 162 เป็นข้อมูลประเภท int

<u>ตัวอย่างที่ 5-28</u>

```
>>> data = [162, 91.4, "Wording"]
>>> type (data[1])
ผลลัพธ์ที่ได้ :
```

<type 'float'>

จากตัวอย่างนี้ต้องการตรวจสอบประเภทของข้อมูลในลำดับ 1 ของตัวแปร data นั่นคือ 91.4 เป็นข้อมูลประเภท float

<u>ตัวอย่างที่ 5-29</u>

```
>>> data = [162, 91.4, "Wording"]
>>> type (data[2])
ผลลัพธ์ที่ได้ :
```

<type `str>

จากตัวอย่างนี้เป็นตรวจสอบประเภทของข้อมูลในลำดับ 2 ของตัวแปร data นั่นคือ Wording พบว่าเป็นข้อมูลประเภทตัวอักษรหรือ str

5.8.1.6 ชุดข้อมูลอะเรย์หลายมิติ

ตัวแปรชนิด List รองรับการกำหนดค่าในแบบชุดข้อมูลอะเรย์หลายมิติได้

<u>ตัวอย่างที่ 5-30</u>

```
>>> matrix = [[1, 2, 3], [35, 50, 71], [5, 25, 125]]
>>> print (matrix[0][2])
```

```
ผลลัพธ์ที่ได้ : 3 ในตัวอย่างนี้กำหนดให้ตัวแปร <sub>matrix</sub> เป็นตัวแปรอะเรย์ 3x3 และเลือก
แสดงค่าของข้อมูลในตำแหน่ง [0,2] ซึ่งก็คือ เลข 3
```

5.8.2 Tuple

Tuple เป็นตัวแปรที่อยู่ในรูปของอะเรย์ที่มีหน้าที่เก็บข้อมูลจำนวนหลายๆ ตัว มีคุณสมบัติ กล้ายกับ List แต่ไม่สามารถแก้ไขข้อมูลที่อยู่ใน Tuple ได้

5.8.2.1 การสร้างตัวแปรแบบ Tuple

การสร้างตัวแปรแบบ Tuple เหมือนกับตัวแปรชนิด List เพียงเปลี่ยนสัญลักษณ์ขอบเขตของ ข้อมูลจาก [] เป็น () ดังตัวอย่าง

<u>ตัวอย่างที่ 5-31</u>

```
>>> my_tuple = (2, 4, 5, 10)
>>> print (my_tuple)
ผลลัพธ์ที่ได้ :
```

```
(2, 4, 5, 10)
```

ในตัวอย่างนี้ตัวแปร _{my_tuple} มีสมาชิก 4 ตัวคือ 2, 4, 5 และ 10

<u>ตัวอย่างที่ 5-32</u>

```
>>> my_tuple = (2, 4, 5, 10)
>>> type (my_tuple)
ผลลัพธ์ที่ได้ :
```

<type 'tuple'>

ในตัวอย่างนี้เป็นการตรวจสอบชนิดของข้อมูลในตัวแปร my_tuple

5.8.2.2 การแก้ไขข้อมูลที่อยู่ใน Tuple

การแก้ไขข้อมูลของสมาชิกในตัวแปรชนิด Tuple ทำไม่ได้ เว้นแต่ว่าจะทำการกำหนดค่าใหม่ ทั้งหมด

<u>ตัวอย่างที่ 5-33 (กำหนดค่าใหม่ทั้งหมด)</u>

```
>>> tuples = 2, 4, 6, 8, 10, 12
>>> tuples = 1, 3, 5
>>> print (tuples)
ผลลัพธ์ที่ได้ :
```

(1, 3, 5)

ในตัวอย่างนี้ ช่วงแรกทำการกำหนดค่าของตัวแปร tuples ไว้อย่างหนึ่ง เนื่องจากเข้าไปแก้ไข ค่าในตัวแปรชนิด Tuple ไม่ได้ จึงต้องใช้การกำหนดค่าใหม่ดังคำสั่งบรรทัด >>> tuples = 1, 3, 5

<u>ตัวอย่างที่ 5-34 (แก้ไขข้อมูลใน Tuple ไม่ได้)</u>

```
>>> tuples = 2, 4, 6, 8, 10
>>> tuples[1] = 5
ພຄລັพธ์ที่ได้ :
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
NameError: name `5' is not defined
```

ในตัวอย่างนี้ ต้องการเปลี่ยนค่าของข้อมูลลำดับ 1 ของตัวแปร tuples จะพบว่า **ทำไม่ได้** จึงเกิดการแจ้งความผิดพลาดขึ้นมาให้ทราบ

5.9 การทำงานแบบมีเงื่อนใข (Condition Operation)

ในการเขียนโปรแกรมทุกภาษาจะต้องมีการตัดสินใจเลือกกระทำคำสั่งตามเงื่อนไขที่กำหนด เพื่อให้เกิดการทำงานของโปรแกรมที่มีความซับซ้อนสูงมากขึ้น ในภาษา Python มีการทำงานแบบ มีเงื่อนไขอยู่ 3 แบบหลักๆ คือ if, for และ while

สิ่งหนึ่งที่ต้องทราบเป็นความรู้พื้นฐานของการทำงานแบบมีเงื่อนไขคือ เครื่องหมายที่ใช้ใน การเปรียบเทียบค่า ประกอบด้วย

- == เท่ากับ
- **! =** ไม่เท่ากับ
- <> ไม่เท่ากับ
- > มากกว่า
- < น้อยกว่า
- >= มากกว่าหรือเท่ากับ
- <= น้อยกว่าหรือเท่ากับ

5.9.1 การทำงานแบบมีเงื่อนไขด้วยกลุ่มคำสั่ง ± £

5.9.1.1 if

เป็นคำสั่งสำหรับกำหนดให้ตัดสินใจเลือกทำงาน เมื่อเงื่อนไขตรงตามที่กำหนดไว้

ร<u>ูปแบบ</u> if (เงื่อนไข): คำสั่ง 1 คำสั่ง 2

ในภาษา Python การเขียนบล็อกของคำสั่งเพื่อควบคุมการทำงานตามเงื่อนไข จะไม่ใช้ เครื่อง หมาย {} (ปีกกาเปิดปิด) แต่จะกำหนดช่วงด้วยการย่อหน้าโดยกดปุ่ม Tab แทน

```
<u>ตัวอย่างที่ 5-35</u>
x = 30
if (x > 6):
print ("Value is more than 6")
ผลลัพธ์ที่ได้ : Value is more than 6
```

<u>ตัวอย่างที่ 5-36</u>

```
x = 67
if (x > 10):
    print ("More than 10")
if (x > 40):
    print ("More than 40")
if (x > 70):
    print ("More than 70")
if (x > 100):
    print ("More than 100")
ผลลัพธ์ที่ได้ :
    More than 10
    More than 40
```

ในตัวอย่างนี้เป็นการตรวจสอบว่า ค่าของตัวแปร x อยู่ในช่วงใด และพิมพ์แสดงข้อความของ ช่วงนั้นๆ ออกมา จากตัวอย่างตัวแปร x มีค่า 67 ดังนั้นเงื่อนไขที่ตรงมี 2 เงื่อนไขคือ

```
if (x > 10):
print ("More than 10")
และ
if (x > 40):
print ("More than 40")
จึงแสดงข้อความ More than 10 และ More than 40
```

จะเห็นว่า เมื่อเงื่อนไขที่อยู่ใน if นั้นถูกต้อง กำสั่งที่อยู่ใน if จะทำงานโดยทันที และในกำสั่ง if แต่ละตัวรองรับกำสั่งได้มากกว่าหนึ่งกำสั่ง ยกตัวอย่าง

```
<u>ตัวอย่างที่ 5-37</u>
```

```
y = 5
if (y > 2):
print ("Value is more than 2")
y = y + 10
print ("Result=", y)
ผลลัพธ์ที่ได้ :
Value is more than 2
Result = 15
```

ในตัวอย่างนี้ แสดงให้เห็นถึงการกระทำคำสั่งภายใต้ if หากค่าของ _Y มากกว่า 2 จะแสดง ข้อความ Value is more than 2 และทำการคำนวณต่อ นั่นคือ _Y = _Y + 10 จากนั้นจึงพิมพ์ ค่าของตัวแปรแสดงออกมา

<u>ตัวอย่างที่ 5-38</u>

```
y = 5
if (y < 2):
    print ("Value is less than 2")
    y = y + 10
print ("Result=", y)
ผลลัพธ์ที่ได้: Result = 5</pre>
```

ในตัวอย่างนี้ แสดงให้เห็นถึงการทำงานเมื่อเงื่อนไขของคำสั่ง if ไม่เป็นจริง ก็จะไม่มีการทำ งานในบล็อกของคำสั่ง if โปรแกรมจะกระโดดข้ามมายังคำสั่ง print ("Result=", y) เพื่อ แสดงผลค่าของ y ทันทั จึงได้ผลลัพธ์เป็น Result = 5

5.9.1.2 else

else เป็นกำสั่งที่ทำงานแบบมีเงื่อนไขนอกเหนือจากที่กำหนดไว้ใน if จะต้องใช้กู่กับ if เสมอ และไม่สามารถใช้งานซ้ำในเงื่อนไขชุดนั้นๆ ได้

ร<u>ูปแบบ</u> if (เงื่อนไข): คำสั่ง 1 else: คำสั่ง 2

จะเห็นว่า else ไม่จำเป็นต้องกำหนดเงื่อนไขใดๆ เพราะ else จะทำงานก็ต่อเมื่อมีการกำหนด เงื่อนไขอื่นนอกเหนือไปจากที่กำหนดไว้ในกำสั่ง if

```
<u>ตัวอย่างที่ 5-39</u>
```

```
x = 15
if (x < 5):
    print ("Less than 5")
else:
    print ("Equal or more than 5")
Waamon file : Equal or more than 5</pre>
```

ในตัวอย่างนี้ กำหนดเงื่อนไขแรกที่คำสั่ง if ว่า ถ้า x น้อยกว่า 5 ให้พิมพ์ข้อความ Less than 5 จากนั้นจะมีการทดสอบเงื่อนไขเพิ่มด้วยคำสั่ง else ว่า ถ้า x ไม่น้อยกว่า 5 ให้พิมพ์ข้อความ Equal or more than 5
else จะใช้ต่อท้ายคำสั่ง if ได้เพียง 1 ครั้งเท่านั้น แต่ใช้งานร่วมกับคำสั่ง if หลายๆ ตัวได้

<u>ตัวอย่างที่ 5-40</u>

```
x = 0
if (x > 1):
    print ("More than 1")
else:
    print ("Equal or less than 1")
if (x < 4):
    print ("Less than 4")
else:
    print ("Equal or more than 4")
ผลลัพธ์ที่ได้
    Equal or less than 1</pre>
```

Less than 4

ในตัวอย่างนี้ มีการใช้คำสั่ if รวม 2 ช่วง

ช่วงแรกตรวจสอบว่า x มากกว่า 1 หรือไม่ และอีกช่วงคือ x น้อยกว่า 4 หรือไม่

ในช่วงแรก จะตรวจสอบว่า x มากกว่า 1 หรือไม่ ถ้าใช่ จะพิมพ์ข้อความ More than 1 ถ้าไม่ใช่ คำสั่ง else จะเข้ามารับช่วงการทำงานต่อ เพื่อกำหนดให้แสดงข้อความ Equal or less than 1 เช่นเดียวกับในช่วงที่ 2 ที่ตรวจสอบว่า x น้อยกว่า 4 หรือไม่ ถ้าใช่ จะแสดงข้อความ Less than 4 หากไม่ใช่ คำสั่ง else จะกำหนดให้ทำงานต่อเพื่อแสดงข้อความ Equal or more than 4

จะเห็นว่า คำสั่ง else จะใช้งานต่อจากคำสั่ง if เสมอ และถ้ามีคำสั่ง if กี่ชุด คำสั่ง else ก็จะนำมาใช้งานได้ตามจำนวนของคำสั่ง if ที่ปรากฏในโปรแกรม

<u>ตัวอย่างที่ 5-41</u>

SyntaxError: invalid syntax

จากตัวอย่างนี้ จะเห็นว่าไม่อาจใช้คำสั่ง else ได้มากกว่า 1 ครั้งต่อการใช้คำสั่ง if หนึ่งชุด

5.9.1.3 else-if

เนื่องจาก if เป็นคำสั่งใช้ตรวจสอบเงื่อนไขตามที่กำหนดไว้ เมื่อมีการใช้งาน if จำนวนมาก อาจทำให้เกิดการทำงานภายในคำสั่ง if แต่ละตัวซ้ำซ้อนกันได้ เช่น

```
x = 2

if (x > 5):

print ("Result >", 5)

if (x > 4):

print ("Result >", 4)

if (x > 3):

print ("Result >", 3)

if (x > 2):

print ("Result >", 2)

if (x > 1):

print ("Result >", 1)

ພລລັพธ์ที่ໄດ້

Result > 5

Result > 4
```

```
Result > 3
```

จะเห็นว่ามี 3 เงื่อนไขที่ถูกต้อง จึงทำให้คำสั่งที่อยู่ในทั้ง 3 เงื่อนไขนั้นทำงาน

ทำให้กำสั่ง else-if (เวลาใช้งานจริงจะเหลือเพียง elif) ถูกนำมาใช้งาน เพื่อให้โปรแกรม ของการตรวจสอบเงื่อนไขในกลุ่มนั้นๆ เลือกทำงาน "เงื่อนไขใคเงื่อนไขหนึ่ง" เท่านั้น

```
กรณีใช้งานร่วมกับคำสั่ง else
<u>รูปแบบ</u>
if (เงื่อนไข 1):
                                       if (เงื่อนไข 1):
   ดำสั่ง 1
                                              ดำสั่ง 1
elif (เงื่อนไข 2):
                                       elif (เงื่อนไข 2):
   คำสั่ง 2
                                              คำสั่ง 2
elif (เงื่อนไข 3):
                                       elif (เงื่อนไข 3):
   คำสั่ง 3
                                              ดำสั่ง 3
                                       else
                                              คำสั่งนอกเหนือจากทุกเงื่อนไข
```

การใช้กำสั่ง else-if เหมือนกับ else คือ ต้องใช้งานร่วมกับif เท่านั้น แต่ else-if เรียกใช้ได้หลายตัวภายใต้กำสั่ง if ชุดเดียวกัน

<u>ตัวอย่างที่ 5-42</u>

```
x = 2
if (x > 5):
    print ("Result >", 5)
elif (x > 4):
    print ("Result >", 4)
elif (x > 3):
    print ("Result >", 3)
elif (x > 2):
    print ("Result >", 2)
elif (x > 1):
    print ("Result >", 1)
ผลลัพธ์ที่ได้ :
```

Result > 1

การทำงานของตัวอย่างนี้คือ โปรแกรมจะตรวจสอบว่า x มากกว่า 5 หรือไม่ ถ้าใช่ ก็ทำการแสดง ข้อความ Result > 5 แล้วออกจากบล็อกคำสั่งของ if แต่ถ้าไม่ใช่จะตรวจสอบต่อว่า มากกว่า 4, มากกว่า 3, มากกว่า 2 และมากกว่า 1 ด้วยคำสั่ง elif จากตัวอย่างค่าของ x คือ 2 จึงตรงกับเงื่อนไขสุดท้ายที่ ว่า elif (x > 1) จึงเข้าไปทำคำสั่งในบล็อก นั่นคือ print ("Result >", 1) เป็นการแสดง ข้อความ Result > 1

จากโปรแกรมตัวอย่างที่ 5-42 เมื่อเงื่อนไขใดก็ตามเกิดเป็นจริงและมีการกระทำคำสั่งภายใต้ เงื่อนไขนั้นๆ ไปแล้ว โปรแกรมจะออกจากการตรวจสอบด้วยคำสั่ง ± £ ในชุดนั้นๆ ทันที แม้ว่า จะมี เงื่อนไขอื่นๆ ที่ถูกต้องรวมอยู่ก็ตาม

5.9.2 การทำงานวนรอบแบบมีเงื่อนไขด้วยคำสั่ง for

for เป็นการวนการทำงานวนลูปแบบมีเงื่อนไข โดยจะวนทำงานของคำสั่งชุดเดิมๆ ไปเรื่อยๆ แต่จะมีจำนวนครั้งในการวนที่เจาะจง จึงเหมาะสำหรับการกำหนดรอบของการวนทำงานที่ทราบ จำนวนครั้งที่แน่นอน

<u>รูปแบบ</u> for ตัวแปร in ชุดข้อมูล: คำสั่ง 1 คำสั่ง 2 <u>ตัวอย่างที่ 5-43</u> for number in [1, 3, 5, 7, 9, 11]: print (number) ผลลัพธ์ที่ได้ : 1 3 5 7 9

จากตัวอย่างนี้ จะเห็นว่า คำสั่ง for จะวนทำงาน 6 ครั้ง ตามจำนวนข้อมูลที่มีในตัวแปร number ที่เป็นตัวแปรแบบ List นี่คืออีกหนึ่งข้อดีของ Python ที่สามารถนำตัวแปร List หรือ Tuple มาใช้ในคำสั่ง for ได้

ถ้าต้องการวนทำงานโดยมีค่าเป็นชุดตัวเลขที่เรียงลำดับอย่างแน่นอน ผู้พัฒนาโปรแกรม สามารถใช้คำสั่ง _{range} เข้ามาช่วยได้

<u>ตัวอย่างที่ 5-44</u>

```
for x in range (4):
print (x)
ผลลัพธ์ที่ได้
0
1
2
3
```

เลข 4 ที่กำหนดลงไปใน _{range} คือจำนวนครั้ง โดยค่าตัวเลขที่ได้จะเริ่มตั้งแต่ 0 และนับไป จนครบ 4 ครั้ง จึงเป็น 0, 1, 2 และ 3 ในกรณีที่ต้องการกำหนดเลขเริ่มต้นด้วย ก็ทำการกำหนดค่าในกำสั่ง range ได้เลย

<u>ตัวอย่างที่ 5-45</u>

```
for x in range (5, 10):

print (x)

ມຸດຄັ້າທີ່ໄດ້ :

5

6

7

8

9
```

เลข 5 ที่กำหนดลงไปในคำสั่ง range คือ เลขเริ่มต้น และเลข 10 คือ ขอบเขตสิ้นสุด

การกำหนดจำนวนการวนทำงานของคำสั่ง for โดยใช้คำสั่ง range เข้ามาช่วย ยังมีอีกหนึ่ง ความสามารถที่น่าสนใจคือ <mark>กำหนดค่าด้วยผลต่างของเลขตัวถัดไป</mark> ดังตัวอย่างที่ 5-46

<u>ตัวอย่างที่ 5-46</u>

```
for x in range (2, 20, 3):
print (x)
ผลลัพธ์ที่ได้ :
2
5
8
11
14
17
```

จากตัวอย่างนี้ที่คำสั่ง range เป็นการกำหนดให้เริ่มต้นที่เลข 2 และสิ้นสุดที่ 20 โดยเพิ่มค่า ขึ้นครั้งละ เมื่อโปรแกรมนี้ทำงานจริง จะสังเกตเห็นว่า คำสั่ง _{for} จะทำงานจนถึงค่าที่เป็น 17 ไม่ใช่ 20 เนื่องจากเลข 20 เป็นขอบเขต จึงไม่อาจนำมาประมวลผลได้

5.9.3 การทำงานวนรอบแบบมีเงื่อนไขด้วยคำสั่ง while

while เป็นการวนการทำงานหรือวนลูปแบบมีเงื่อนไขว่า จะวนทำงานกำสั่งชุดเดิมไปเรื่อยๆ จนกว่าเงื่อนไขที่กำหนดไว้จะไม่เป็นจริง โดยจะทำการตรวจสอบเงื่อนไขก่อน แล้วจึงทำงานตามชุด กำสั่ง

```
<u>รูปแบบ</u>
while (เงื่อนไข):
   ดำสั่ง 1
   คำสั่ง 2
   คำสั่ง 3
<u>ตัวอย่างที่ 5-47</u>
count = 0
while (count < 20):
   count += 3
   print ("Count =", count)
ผลลัพก์ที่ได้
   Count = 3
   Count = 6
   Count = 9
   Count = 12
   Count = 15
   Count = 18
   Count = 21
```

จากตัวอย่างนี้ ตัวแปร count เพิ่มค่าขึ้นครั้งละ 3 เมื่อมีค่าน้อยกว่า 20 เมื่อวนทำงานไป เรื่อยๆ จนถึงเมื่อตัวแปร count มีค่าเป็น 18 ซึ่งยังน้อยกว่า 20 ทำให้ต้องเพิ่มค่าอีก 3 กลายเป็น 21 และทำให้ไม่ตรงกับเงื่อนไข จึงทำให้ชุดคำสั่งที่อยู่ในลูป while หยุดทำงานทันที ในกรณีที่ต้องการออกจากการทำงานในลูป while กลางคัน ให้ใช้คำสั่ง break

<u>ตัวอย่างที่ 5-48</u>

```
import random
count = 0
while (1):
    count += 1
    x = random.randint (1, 10)
    print ("Random", x)
    if (x == 10):
        break;
        print ("Random count", count)
ผลลัพธ์ที่ได้
    Random 6
    Random 6
    Random 4
    Random 2
    Random 10
    Random Count 4
```

จากตัวอย่างเป็นการสุ่มตัวเลขระหว่าง 1 ถึง 10 ขึ้นมา แล้วแสดงผลออกทางหน้าจอ ใน ระหว่างการสุ่มแต่ละครั้งจะมีการนับด้วยว่า สุ่มไปกี่ครั้ง โดยเก็บไว้ในตัวแปร _{count} เมื่อสุ่มได้เลข 10 ก็จะทำการออกจากลูป _{while} ในทันที แล้วแสดงจำนวนครั้งที่ทำการสุ่ม จากโปรแกรมตัวอย่าง จำนวนครั้งในการสุ่งคือ 4 จากข้อความแสดงผลตัวสุดท้าย Random Count 4 116 • Raspberry Pi 3 : รู้จักและการใช้งานเบื้องตัน

5.10 ฟังก์ชั่นของ Python

ฟังก์ชั่นถือเป็นส่วนสำคัญของภาษาคอมพิวเตอร์ทุกภาษาที่ขาดไปไม่ได้ เพราะเป็นเสมือนชุด กำสั่งที่เตรียมพร้อมสำหรับใช้งาน เหมาะสำหรับการเรียกชุดกำสั่งเดิมๆ ซ้ำ ช่วยลดความซับซ้อนของ กำสั่งในโปรแกรม อีกทั้งยังช่วยให้กำสั่งในโปรแกรมดูเข้าใจได้ง่ายขึ้น

รูปแบบฟังก์ชั่นในโปรแกรมภาษา Python เป็นดังนี้

```
def ชื่อฟังก์ชั่น (พารามิเตอร์):
```

คำสั่ง ตัวอย่างที่ 5-49

```
def print_ok ():
print ("OK")
print_ok()
print_ok()
print_ok()
ผลลัพธ์ที่ได้ :
OK
OK
```

การสร้างฟังก์ชั่นในภาษา Python มีเงื่อนไขหลักคือ จะต้องสร้างการเรียกใช้งานฟังก์ชั่นนั้นๆ ด้วย เพราะว่า ภาษา Python ทำงานแบบอินเตอร์พรีตเตอร์ จึงไม่อาจประกาศไว้หลังจากคำสั่งเรียก ใช้งานได้

นอกจากนั้น ฟังก์ชั่นในภาษา Python ยังรองรับการกำหนดพารามิเตอร์เพื่อกำหนดข้อมูลตัว แปรที่นำมาใช้งานภายในโปรแกรมได้ด้วย

<u>ตัวอย่างที่ 5-50</u>

```
def get_triangle_area (width, height):
area = width * height / 2
print ("Area =", area)
get_triangle_area (100, 30)
ผลลัพธ์ที่ได้ : Area = 1500
```

ในตัวอย่างนี้เป็นการสร้างฟังก์ชั่นหาพื้นที่สามเหลี่ยม ชื่อฟังก์ชั่นคือ get_triangle_area มีพารามิเตอร์ 2 ตัวคือ width ความกว้างของฐาน และ height คือ ความสูงของรูปสามเหลี่ยม ในกรณีที่ต้องการส่งก่าผลลัพธ์ที่ได้จากฟังก์ชั่น เรียกใช้งานกำสั่ง Return ได้เหมือนภาษาอื่นๆ

<u>ตัวอย่างที่ 5-51</u>

```
def get_triangle_area (width, height):
    area = width * height / 2
    return (area)
result = get_triangle_area (50, 12)
print ("Result Area =", result)
ພຸລລັพก์ที่ได้ : Result Area = 300
```

ในตัวอย่างนี้เป็นการสร้างฟังก์ชั่นหาพื้นที่สามเหลี่ยม ชื่อฟังก์ชั่นคือ get_triangle_area โดยต่อยอดมาจากตัวอย่างที่ 5-50 นั่นคือ เพิ่มคำสั่ง return (area) เข้ามา เพื่อทำให้ฟังก์ชั่น สามารถส่งค่ากลับไปได้ จากนั้นจึงนำค่าผลลัพธ์ของฟังก์ชั่นไปเก็บไว้ในตัวแปร แล้วทำการแสดงผล ด้วยคำสั่ง print ต่อไป

นอกจากนี้ภาษา Python ยังมีความพิเศษตรงที่ กำหนดค่าเริ่มต้นให้กับพารามิเตอร์ในฟังก์ชั่นได้ ด้วย จึงทำให้ผู้พัฒนาโปรแกรมเรียกใช้งานฟังก์ชั่นนั้นๆ ได้โดยกำหนดค่าพารามิเตอร์เหล่านั้นหรือไม่ก็ได้

<u>ตัวอย่างที่ 5-52</u>

```
2nd Result = 50
```

```
3rd Result = 180
```

ในตัวอย่างนี้ กำหนดให้ค่าเริ่มต้นของพารามิเตอร์คือ width = 30 และ height =5 เมื่อ เรียกใช้ฟังก์ชั่น get_square_area () ในแบบไม่กำหนดพารามิเตอร์ มันจะดึงค่า width = 30 และ height =5 มาใช้ในการคำนวณ ผลลัพธ์ที่ได้จึงเป็น 150 ตามคำตอบแรกที่เก็บไว้ในตัวแปร Result1 ส่วนใน result2 กำหนดพารามิเตอร์เพียงตัวเดียว ฟังก์ชั่นจะมองว่า กำหนดให้พารามิเตอร์ ตัวแรก นั่นคือ width ดังนั้นในการทำงาน จะดึงค่า height =5 มาคำนวณ จึงได้คำตอบเป็น 50 (มาจาก 10 x 5)

สุดท้ายใน result3 มีการกำหนดพารามิเตอร์มาครบทั้งสองตัว จึงนำค่ามาคูณกันได้ทันที ได้ผลลัพธ์เป็น 180 (มาจาก 12 x 15)

จะเห็นว่า ถ้ามีการกำหนดค่าเริ่มต้นให้กับพารามิเตอร์แล้ว เวลาเรียกใช้งานฟังก์ชั่นจะสามารถ เลือกได้ว่าจะกำหนดค่าให้กับพารามิเตอร์นั้นๆ ใหม่หรือไม่ ถ้าไม่ได้กำหนด โปรแกรมจะก็เรียกค่าเริ่ม ต้นที่กำหนดไว้ในฟังก์ชั่นมาใช้งานแทน

ข้อควรระวัง – ถ้าไม่ได้มีการกำหนดค่าเริ่มต้นให้กับฟังก์ชั่น เวลาเรียกใช้งานฟังก์ชั่นนั้นๆ จะต้อง กำหนดค่าพารามิเตอร์ด้วยทุกครั้ง ไม่เช่นกันโปรแกรมจะเกิดข้อผิดพลาดขึ้น ดังตัวอย่างที่ 5-53 ต่อไปนี้

<u>ตัวอย่างที่ 5-53</u>

```
def get_square_area(width, height=5):
    return width * height
result = get_square_area()
print ("Result =", result)
ผลลัพธ์ที่ได้ :
```

```
pi@raspberrypi ~ $ sudo python simple_fn.py
Traceback (most recent call last):
   File "simple_fn.py", line 4, in <module>
      result = get_square_area()
TypeError: get_square_area() takes at least 1 argument (0 given)
```

ทั้งหมดที่นำเสนอในบทนี้เป็นเพียงข้อมูลและตัวอย่างเบื้องต้นของการเขียนโปรแกรมด้วย ภาษา Python ผู้ที่สนใจศึกษาเพิ่มเติมขอแนะนำให้จัดหาหนังสือการเขียนโปรแกรมภาษา Python มาอ่านและศึกษาเพิ่มเติม โดยหนังสือที่ขอแนะนำคือ หนังสือรายวิชาเพิ่มเติม เทคโนโลยีสารสนเทศ และการสื่อสาร ภาษาไพธอน จัดพิมพ์โดย สถาบันส่งเสริมการสอนวิทยาศาสตร์และเทคโนโลยี (สสวท.)



^{บทที่ 6} Raspberry Pi 3 กับการพัฒนาโปรแกรม ภาษา Python ด้วยซอฟต์แวร์ Geany



ในบทที่ 5 นำเสนอเกี่ยวกับที่มาและพื้นฐานของภาษา Python รวมถึงแนวทางในการพัฒนา โปรแกรม โดยทั้งหมดในบทที่ 5 เป็นการพัฒนาโปรแกรมบนเท็กซ์เอดิเตอร์ Nano ที่มากับระบบ ปฏิบัติการ Raspbian ซึ่งมีลักษณะเป็นหน้าต่างแบบ DOS สำหรับผู้ใช้งานและนักพัฒนาที่ต้องการ เครื่องมือพัฒนาโปรแกรมภาษา Python ในแบบกราฟิกหรือแบบที่เรียกว่า IDE (Integrated Development Environment) ที่รวบรวมกำสั่งและเครื่องมือต่างๆ ไว้ในหน้าต่างเดียว ในที่นี้ขอแนะ นำซอฟต์แวร์ที่ชื่อ Geany

6.1 รู้จักกับ Geany

Geany เป็นซอฟต์แวร์สำหรับเขียนข้อความหรือเท็กซ์เอคิเตอร์ (text editor) ที่บรรจุเครื่องมือ ต่างๆ ไว้พร้อมใช้งานภายใต้หน้าต่างเดียวกัน แต่มีขนาคของซอฟต์แวร์ที่ไม่ใหญ่ เพื่อให้การประมวล ผลทำได้อย่างรวคเร็ว ทั้งยังรองรับคอมไพเลอร์หรือตัวแปลภาษาได้อย่างกว้างขวาง

เว็บไซต์อย่างเป็นทางการของ Geany คือ *http://www.geany.org* ผู้สนใจใช้งานคาวน์โหลด มาใช้ได้โดยไม่มีค่าใช้จ่าย จึงมีนักพัฒนาที่นำ Geany มาติดตั้งคอมไพเลอร์ของภาษา Python สำหรับ ใช้กับบอร์ด Raspberry Pi เพื่อเขียนโปรแกรมควบคุมการทำงานด้วยภาษา Python ทั้งการประมวล ผลทางซอฟต์แวร์และการติดต่อสั่งงานทางฮาร์ดแวร์ผ่านพอร์ตอินพุตเอาต์พุตอเนกประสงค์หรือ GPIO (General Purpose Input Output) ของบอร์ด Raspberry Pi

สำหรับท่านที่จัดซื้อชุดเรียนรู้ Raspberry Pi 3 จาก INEX ในแผ่น microSD การ์ดได้ทำการ ติดตั้ง Geany ไว้พร้อมใช้งานบนระบบปฏิบัติการ Raspbian แล้ว

หากระบบปฏิบัติการ Raspbian ที่มีอยู่ไม่มีซอฟต์แวร์ Geany ติดตั้งไว้ ผู้ที่ต้องการใช้งาน สามารถทำการติดตั้งได้เอง โดย

(1) เชื่อมต่อบอร์ค Raspberry Pi 3 เข้ากับเครือข่ายอินเทอร์เน็ต

(2) หลังจากบูตระบบเสร็จสิ้น ให้เข้าสู่หน้าต่างเทอร์มินอล นั่นคือ LX Terminal

(3) ติดตั้งโปรแกรมด้วยการพิมพ์คำสั่ง

pi@raspberrypi ~/code \$ sudo apt-get update pi@raspberrypi ~/code \$ sudo apt-get install geany

(4) จากนั้นรอจนกระทั่งการติดตั้งโปรแกรมเสร็จสิ้น

6.2 ส่วนประกอบของ Geany

เมื่อติดตั้งโปรแกรมเสร็จแล้ว ทำการเรียกใช้งานระบบปฏิบัติการในแบบกราฟิก โดยการกลับ ออกไปยังเชลพร้อมพ์หลักแล้วพิมพ์คำสั่ง startx เพื่อเลือกให้ทำงานบนระบบปฏิบัติการ Raspbian แบบกราฟิก จากนั้นค้นหาไอคอนของ Geany แล้วทำการดับเบิลคลิกเพื่อรันให้ทำงาน



หรือเลือกจากเมนู Programming > Geany ดังรูป

🗼 Accessories 🔹 🕨]	
🔕 Education 🔹 🕨 🕨		
🚅 Graphics 🔹 🕨 🕨		
🕥 Internet 🔹 🕨 🕨		
💾 Other 🔹 🕨 🕨		
🛒 Programming 💦 🕨 🕨	🍖 Geany	
👆 Sound & Video 🔹 🕨	🥐 IDLE	
🏐 System Tools 🔹 🕨 🕨	🥐 IDLE 3	
Preferences	🥸 Scratch	
	🕬 Sonic Pi	
Kun	🥘 Squeak	
🖸 Logout		
🔨 🗉 🔍 🗖 🗖	💻 pi@raspberry	/pi:

เมื่อเปิคโปรแกรมขึ้นมา หน้าต่างหลักของโปรแกรมเป็นดังรูปที่ 6-1

🤕 untitled - Geany 📮 🗖 💌
Eile Edit Search View Document Project Build Tools Help
Symbols Documents untitled X
3 2 =
Status 09.53:02: Imis is deally 1.22. 09.53:02: New file "untitled" opened.
Messages
Scribble 4
Terminal
This is Geany 1.22
มีส่วนประกอบดังนี้
 แถบครื่องมือ ส่วนที่ใช้เขียนคำสั่ง
3 ส่วนแสดงผลที่อยู่ของไฟล์เมื่อเลือก Documents และแสดงส่วนประกอบต่างๆ ของโปรแกรม ที่เขียนเมื่อเลือก Symbols
4 ส่วนแสดงผลลัพธ์ต่างๆ เช่น ผลลัพธ์ของการคอมไพล์โปรแกรม

รูปที่ 6–1 ส่วนประกอบของหน้าต่างหลักของ Geany ซอฟต์แวร์เท็กซ์เอดิเตอร์สำหรับพัฒนาโปรแกรมให้แก่ Raspberry Pi

6.2.1 แถบเครื่องมือ



มีส่วนประกอบดังนี้

 สร้างไฟล์ใหม่โดยไม่ระบุนามสกุล ถ้าต้องการสร้างไฟล์ใหม่ที่ระบุบนามสกุลด้วย ให้เลือก ที่รูปลูกศรชี้ลงจะปรากฏเมนูให้เลือกดังรูป

@		untitled - Geany
<u>Ele E</u> dit <u>S</u> earch <u>V</u> iew	Document Project	<u>B</u> uild <u>T</u> ools <u>H</u> elp
💊 <u>N</u> ew	Ctrl+N	
📓 New (with <u>T</u> emplate)	>	file.html
Open Open Selected File Recent Files	Ctrl+0 Shift+Ctrl+0 >	file.php file.rb file.tex
Save As	Ctrl+S	main.c main.cxx
Save All	shift+ctrl+s	main.java
🔊 <u>Re</u> load As	Ctri+R	main.py program.pas
Properties		

(2) เปิดไฟล์ ถ้าต้องการเปิดไฟล์ที่เคยเปิดมาแล้ว ให้เลือกที่รูปลูกศรชี้ลง จะปรากฏเมนูขึ้น มาให้เลือก

- (3) บันทึกไฟล์ที่เลือกไว้
- (4) บันทึกไฟล์ทั้งหมดที่เปิด
- (5) เรียกคืนรูปแบบเดิมทั้งหมด จะใช้ในกรณีที่ยังไม่ได้บันทึกไฟล์
- 6 ปิดไฟล์ที่เลือกไว้
- 7 คอมไพล์ไฟล์สคริปต์
- 8 รันไฟล์สคริปต์
- (9) ค้นหาคำที่อยู่ในไฟล์
- 10 เลือกหมายเลขบรรทัดที่ต้องการดูคำสั่ง
- 11 ออกจากโปรแกรม

6.2.2 การเลือกรูปแบบตัวอักษร

ผู้พัฒนาโปรแกรมสามารถเลือกรูปแบบหรือฟอนต์ (font) ของตัวอักษรที่แสดงผลบนพื้นที่ เขียนคำสั่ง โดยเลือกไปที่เมนู View > Change Font จากนั้นจะปรากฏหน้าต่างสำหรับตั้งค่า ดังรูป

1	g) Cł	noose	font	[- • ×
	<u>F</u> amily:		<u>S</u> tyle:		Si <u>z</u> e:
	FreeMono	^	Regular	-	14
	FreeSans		Italic		
	FreeSerif		Bold		12
	Monospace		Bold Italic	•	13 =
	Nimbus Mono L	Ξ			14
	Nimbus Roman No9 L				15
	Nimbus Sans L	~			16
	Preview:				
	abcdefghijk ABCDEFG	GHIJ	к		
		<u> </u>	Cancel 🖌 Apply	4	<u>о</u> к

เมื่อตั้งค่าแล้ว คลิกปุ่ม OK เพื่อยืนยัน

6.2.3 ตั้งค่าการคอมไพล์และรันโปรแกรมหรือเอ็กซิคิวต์

เนื่องจากภาษา Python ที่ใช้กับบอร์ด Raspberry Pi 3 ในขณะที่จัดทำหนังสือเล่มนี้มี 2 เวอร์ชัน หลักคือ Python2 และ Python3 ผู้พัฒนาโปรแกรมอาจมีโค้ดของภาษา Python ทั้ง 2 เวอร์ชัน หากทำ การคอมไพล์ (compile) และเอ็กซิคิวต์ (execute) หรือรันโปรแกรมไม่ตรงเวอร์ชัน อาจทำให้ไฟล์ สคริปต์ทำงานไม่ได้ หรือคอมไพล์ไม่ผ่าน จึงขอแนะนำการเพิ่มเมนู เพื่อให้ผู้ใช้งาน Geany ได้มีทาง เลือกในการคอมไพล์และเอ็กซิคิวต์หรือรันโปรแกรมดังนี้

(1) ไปที่ Build จะปรากฏเมนูให้เลือกดังรูป

∲ ≩ <u>C</u> ompile	F8
<u>M</u> ake	Shift+F9
Make Custom <u>T</u> arget	Shift+Ctrl+F9
Make <u>O</u> bject	Shift+F8
<u>N</u> ext Error	
Previous Error	
୍ଞଞ୍ଚି <u>E</u> xecute	F5
🕮 <u>S</u> et Build Commands	

จะเห็นว่ามีคำสั่ง Complie และ Execute ให้เลือก โดยทั้งสองคำสั่งใช้สำหรับ Python3

(2) จากนั้นให้คลิกที่ Set Build Commands จะปรากฏตารางขึ้นมา ให้ทำการเพิ่ม Label และ Command ดังรูป แล้วคลิก OK

9		Set Build Commands		_ 🗆 🗙
#	Label	Command	Working directory	Reset
Py	thon commands			
1.	<u>C</u> ompile	python3 -m py_compile "%f		
2.	<u>C</u> ompile Python2	python -m py_compile "%f"		
3.				
Er	ror regular expression:			
Ind	lependent commands			
1.	<u>M</u> ake	make		
2.	Make Custom <u>T</u> arget	make		
3.	Make <u>O</u> bject	make %e.o		
4.				
Er	ror regular expression:			
No	te: Item 2 opens a dialog	and appends the response to t	the command.	
EX	ecute commands			
1.	<u>E</u> xecute	sudo python3 "%f"		
2.	Execute Python2	sudo python "%f"		
%d	, %e, %f, %p are substitut	ed in command and directory fi	elds, see manual for details	
			<u>Cancel</u>	🤩 <u>о</u> к

(3) จากนั้นไปที่ Build จะเห็นว่า มีเมนูเพิ่มขึ้นมาจากเดิมดังรูป

∲ ≩ <u>C</u> ompile	F8
ទ <u>C</u> ompile Python2	F9
<u>M</u> ake	Shift+F9
Make Custom <u>T</u> arget	Shift+Ctrl+F9
Make <u>O</u> bject	Shift+F8
☆ <u>N</u> ext Error <u>P</u> revious Error	
	F5
🕮 Set Build Commands	

@	LAB01_00SWLED.py - /home/pi/Desktop/PythonDemo - Geany	_ = ×
<u>F</u> ile <u>E</u> dit <u>S</u> earch <u>V</u> iev	w <u>D</u> ocument <u>P</u> roject <u>B</u> uild <u>T</u> ools <u>H</u> elp	
P • 🖬 • 🛔] 💭 📐 😪 😒 🕸 🍫 🛸 🔚 🖂 🖉 🖉	🥖 🗞 💽
Symbols Documents	LAB01_00SWLED.py 🗙	
✓ Variables	<pre>1 #!/usr/bin/env python 2 # -*- coding: utf-8 -*- 3 4 import RPi.GPIO as GPIO 5 import time 6 GPIO.setwarnings(False) 7 GPIO.setwode(GPIO.BCM) 8 GPIO.setup(18,GPIO.OUT) 9 GPIO.setup(17,GPIO.IN) 10 #while (1): 11 \$ state=GPIO.input(17) 12 GPIO.output(18,state) 13 print 'SW='+ str(state) 14 15 16 17</pre>	=
Status 16:27:34: File // 16:27:50: File //	/home/pi/Desktop/PythonDemo/LAB01_00SWLED.py saved. /home/pi/Desktop/PythonDemo/LAB01_00SWLED.py saved.	
Compiler 17:27:03: Did n	not send document /home/pi/Desktop/PythonDemo/LAB01_00SWLED.py to the printing subsystem.	
Messages 17:27:33: File u	untitled closed.	
Scribble 17:39:34: New f 17:39:36: File u	file "untitled" opened. untitled closed.	Ξ
Terminal 17:39:55: New f	file "untitled" opened.	
17:39:56: File u	untitled closed.	v
line: 11 / 17 col: 0 sel: 0	0 INS TAB mode: Win (CRLF) encoding: UTF-8 filetype: Python scope: unknown	

รูปที่ 6-2 แสดงตัวอย่างหน้าต่างของ Geany เมื่อมีการเปิดไฟล์ของโปรแกรมภาษา Python ขึ้นมาใช้งาน

นับจากนี้ผู้พัฒนาจะสามารถเลือกคอมไพล์และเอ็กซิคิวต์หรือรันโปรแกรมภาษา Python ได้ทั้ง Python 2 และ Python 3 ผ่านทางเมนู Build ได้ตามต้องการ

อนึ่ง โปรแกรมตัวอย่างของภาษา Python ที่พัฒนาขึ้นเพื่อควบคุมการทำงานของ Raspberry Pi 3 ในหนังสือเล่มนี้นับจากบทนี้จะใช้ Geany เป็นเท็กซ์เอคิเตอร์หลักและใช้ Python3 ในการพัฒนา โปรแกรม

6.3 ตัวอย่างการทดสอบโปรแกรมภาษา Python ด้วย Geany

ในการเขียนโปรแกรมภาษา Python สำหรับ Raspberry Pi 3 ในที่นี้ ขอแนะนำให้ใช้ Python 3 เป็นหลัก สำหรับการพัฒนาโปรแกรมบน Geany เบื้องค้น มีขั้นตอนดังนี้

(1) สร้างไฟล์สคริปต์ใหม่ โดยไปที่ File > New(With_Template) แล้วคลิกที่ main.py ดังรูป

@	LED_Bli		_ • ×
<u>Eile E</u> dit <u>S</u> earch <u>V</u> iew <u>D</u> ocument	<u>P</u> roject <u>B</u> uild <u>T</u> ools	Help	
New 0	Ctrl+N	🚱 🌭 🗸 🎭 i 🌆 i 🚺 📝 🔿 i 📝 🖇	
P New (with Template)	>		
🖻 Open 🛛	Ctrl+O		
Open Selected File Shift+0	Ctrl+O file.rb	PI0	\sim
Recent <u>F</u> iles	> file.tex	.se)	
A Save	main.c	(M)	
Save As	main.cxx	JT)	
La Save All Shift+(Ctrl+S		
Beload 0	main.java	·)	
Reload As	main.py		=
Descrition	main.vala)	
lal Properties	program.pas		
Page Set <u>u</u> p			
Print (Ctrl+P		
X <u>C</u> lose C	trl+W		
X Close Other Documents			
X Close All Shift+C	lul+w		V
🕞 Quit 🛛	Ctrl+Q		
Status 17:36:21: This is Geany 1.22.			
Compiler 17:36:22: File /home/pi/Desktop/Ll 17:36:44: File /home/pi/Desktop/Ll	ED_Dlinking.py opened(1). ED_Blinking.py saved.		
17:38:41: New file "untitled py" op	sened		
Interstages 17:39:02: File untitled.py closed.	- 1		
Scribble 17:39:04: New Tie untitled open 17:39:06: File untitled closed.	ea.		
Terminal			
	1		
line: 11 / 13 col: 21 sel: 0 INS TAB	mode: Unix (LF) end	coding: UTF-8 Tiletype: Python scope: unknown	



(2) จะปรากฏไฟล์สคริปต์ที่มีชื่อว่า untitled.py คังรูป[20]

(3) พิมพ์คำสั่งต่อไปนี้

```
print ('Python',3)
print ('Hello','World!')
print ("Hello","World!")
```

(4) บันทึกไฟล์สคริปต์ โดยไปที่ File > Save เลือกที่อยู่ของ File แล้วตั้งชื่อว่า HelloPython3.py ในที่นี้จะเก็บไฟล์ไว้ที่ Desktop แล้วคลิก Save

٢			Save File			_ 0
Name:	HelloPyth	on2.py				
Save in <u>f</u> older:	< 🕅 pi	Desktop				Create Folde
<u>P</u> laces		Name		~	Size	Modified
Q Search		mes				04/23/2015
🕙 Recently Use	d	🛅pycache				15:04
🔝 pi		🛅 PythonDemo				13:06
🐻 Desktop		🛅 sound				04/23/2015
🗐 File System		🛅 video				04/23/2015
	:	📄 geany.desktop			5.2 KB	07/07/2012
+ -		LED_Blinking.py			223 bytes	17:36
Open file in a	new tab			R <u>e</u> name	Cancel	Save

(5) ทดสอบคอมไพล์และเอ็กซิคิวต์ คำสั่ง print ที่พิมพ์ลงในไฟล์สคริปต์เป็นคำสั่งแสดงข้อ ความออกทางหน้าจอภาพ รูปแบบคำสั่ง print ของ Python3 และ 2 แตกต่างกัน ในที่นี้จะใช้ Python3 เป็นหลัก ดังนั้นหากผู้พัฒนาโปรแกรมมีตัวอย่างโปรแกรมเดิมของ Python2 เมื่อนำมาคอมไพล์ หรือเอ็กซิคิวต์ด้วย Python3 จะต้องทำการแก้ไชคำสั่งให้ถูกต้องก่อน โดยศึกษาได้จาก help ของ Python3

(5.1) เริ่มต้นด้วยการคอมไพล์

(5.1.1) ไปที่ **Build** จะปรากฎเมนูดังรูป



(5.1.2) คลิกที่ **Compile** เพื่อใช้ Python3 คอมไพเลอร์ในการคอมไพล์ จากนั้นผลลัพธ์ ของการคอมไพล์จะปรากฏที่กรอบด้านล่างของหน้าต่างโปรแกรม โดยแสดงข้อความว่า **Compila** tion finished successfully แสดงว่ารูปแบบคำสั่งถูกต้องพร้อมที่จะทำการเอ็กซิคิวต์หรือรันต่อไป

(5.2) การเอ็กซิคิวต์หรือสั่งให้รันโปรแกรม

(5.2.1) ไปที่ Build ทำการเลือกเอ็กซิคิวต์ด้วย Python3 โดยคลิกเลือกที่ Execute ดังรูป







นี่คือตัวอย่างการใช้งาน Geany ในการพัฒนาโปรแกรมภาษา Python ให้แก่บอร์ค Raspberry Pi โดยตัว Geany เองรองรับและทำงานร่วมกับคอม ไพเลอร์ได้หลายตัว ทั้ง Python2, Python3 สำหรับ ภาษา Python หากเป็นภาษา C คอม ไพเลอร์ที่นิยมสำหรับผู้เริ่มต้นคือ WiringPi

สำหรับการเรียนรู้เพื่อเขียนโปรแกรมควบคุมการทำงานของบอร์ค Raspberry Pi 3 ในหนังสือเล่มนี้เลือกใช้ภาษา Python และเป็นรุ่น Python3 โคยได้ทำการผนวก คอมไพเลอร์ไว้ใน Geany ตั้งแต่ขั้นตอนการสร้างอิมเมจไฟล์สำหรับระบบปฏิบัติการ Raspbian ที่ทาง INEX จัดทำขึ้น ทั้งนี้เพื่ออำนวยความสะควกในการเรียนรู้และพัฒนา โปรแกรม



บทที่ 7 การติดต่อพอร์ตอินพุตเอาต์พุตอเนกปร:สงค์ หรือ GPIO ของ Raspberry Pi 3



สิ่งที่ทำให้บอร์ด Raspberry Pi เป็นที่นิยมนั้นไม่ได้มีเพียงเรื่องราคาและขนาด แต่บอร์ด Raspberry Pi เป็นบอร์ดคอมพิวเตอร์ที่มีขาพอร์ตอินพุตเอาต์พุตอเนกประสงค์หรือ GPIO (General Purpose Input Output) สำหรับนำไปเชื่อมต่อกับวงจรอิเล็กทรอนิกส์ได้ จึงทำให้นำบอร์ด Raspberry Pi ไปสร้างสรรค์ โครงงานและสิ่งประดิษฐ์ได้อย่างมากมาย และนั่นคือจุดขายที่สำคัญที่นำโปรแกรมเมอร์มาพบกับนัก ทดลองวงจรอิเล็กทรอนิกส์ ส่งผลให้เกิดกลุ่มของผู้ใช้งานที่กว้างขึ้นอย่างมากมายมหาศาล

7.1 การจัดขา GPIO

บอร์ด Raspberry Pi 3 มีกอนเน็กเตอร์สำหรับขาพอร์ต GPIO 40 ขา แบ่งเป็น 2 แถว แถวละ 20 ขา โดยแต่ละขาจะมีหน้าที่ต่างกันดังแสดงในรูปที่ 72 จะเห็นได้ว่า มีขาพอร์ตอินพุตเอาต์พุตสำหรับใช้ งานรวม 26 ขา มีระดับสัญญาณลอจิกเป็น 0 และ +3.3V จึงทำให้ Raspberry Pi 3 สามารถเชื่อมต่อ อุปกรณ์อิเล็กทรอนิกส์ได้หลากหลาย ตั้งแต่ LED, สวิตช์ ไปจนถึงไอซีหน้าที่พิเศษหรือตัวตรวจจับ ต่างๆ ที่ใช้การเชื่อมต่อผ่านบัส UART, I²C และ SPI



รูปที่ 7-1 แสดงตำแหน่งของคอนเน็กเตอร์ GPIO 40 ขาบนบอร์ด Raspberry Pi 3



รูปที่ 7-2 การจัดขาคอนเน็กเตอร์ GPIO ของบอร์ด Raspberry Pi 3

้งาพอร์ตอินพุตเอาต์พุตของ Raspberry Pi 3 มีทั้งสิ้น 26 งา มีการแบ่งปันหน้าที่การทำงานดังนี้

1. เป็นขาพอร์ตอินพุตเอาต์พุตดิจิตอล 26 ขา

2. ใช้งานเป็นขาเอาต์พุตสัญญาณ PWM ได้ 26 ขา

3. ใช้งานเป็นขาเชื่อมต่อบัส I²C 1 ช่อง (2 ขา : SDA และ SCL โดยแบ่งปันการทำงานกับขา พอร์ต GPIO2 และ 3)

4. ใช้งานเป็นขาเชื่อมต่อพอร์ตอนุกรมสำหรับสื่อสารข้อมูลอนุกรม UART 1 ช่อง (2 ขา : TxD และ RxD โดยแบ่งปันการทำงานกับขาพอร์ต GPIO14 และ 15)

5. ขาเชื่อมต่อบัส SPI 1 ช่อง (3 ขาหลัก : MOSI, MISO, SCK โดยแบ่งปันการทำงานกับขา พอร์ต GPIO10, 9 และ 11 ตามลำคับ พร้อมกับ 2 ขาเสริม : CE0 และ CE1 โดยแบ่งปันการทำงานกับ ขาพอร์ต GPIO8 และ 7 ตามลำคับ)

นอกจากนี้ยังมีขาพอร์ตสำหรับเชื่อมต่อหน่วยความจำอีอีพรอมอนุกรมภายนอก (Serial EEPROM) นั่นคือ ขา **ID_SCL** (ขาสัญญาณนาฬิกา) และ **ID_SDA** (ขาข้อมูล) เพื่อใช้กำหนดชื่อของฮาร์ดแวร์ที่นำมา ต่อกับพอร์ต GPIO ของ Raspberry Pi 3 ภายใต้มาตรฐาน **HAT** (Hardware Attached on Top) ขาพอร์ต GPIO ทั้งหมครับและขับสัญญาณได้สูงสุด +3.3V ไม่สามารถรับแรงคัน +5V ได้เมื่อ ทำงานเป็นพอร์ตอินพุต ด้านความสามารถในการรับและง่ายกระแสไฟฟ้าของพอร์ต GPIO แต่ละขาไม่ เกิน 50mA แต่รวมทุกขาพอร์ตจะง่ายกระแสไฟฟ้าได้ไม่เกิน 300mA เมื่อทำงานเป็นพอร์ตเอาต์พุต

ดังนั้นในการใช้งานขาพอร์ต GPIO ของ Raspberry Pi จึงต้องระมัดระวังอย่าป้อนแรงดัน +5V เข้าที่ขาพอร์ตโดยตรงอย่างเด็ดขาด เพราะจะทำให้วงจรของขาพอร์ต GPIO เสียหายได้ในทันที แม้ว่าที่จุดต่อหรือคอนเน็กเตอร์ GPIO ของ Raspberry Pi จะมีจุดต่อ +5V ไว้ให้ใช้งานก็ตาม ทั้งนี้ การมีจุดต่อแรงดัน +5V (ซึ่งมาจากพอร์ต microUSB ของบอร์ด Raspberry Pi) ก็เพื่อประโยชน์ใน การเป็นไฟเลี้ยงให้อุปกรณ์ต่อพ่วงที่นำมาต่อกับบอร์ด Raspberry Pi ซึ่งต้องการไฟเลี้ยง +5V จะได้ ไม่ต้องจัดหาแหล่งจ่ายไฟเลี้ยงเพิ่มเติม ทำให้การใช้งานพอร์ต GPIO ของ Raspberry Pi กับอุปกรณ์ ต่อพ่วงภายนอกทำได้สะดวกขึ้นอย่างมาก

7.2 Rpi-I/O40 บอร์ดอินพุตเอาต์พุตสำหรับ Raspberry Pi

ในการใช้งานเพื่อเชื่อมต่อขาพอร์ต GPIO ของบอร์ค Raspberry Pi เพื่อความปลอคภัยและสะควก ในการต่อวงจร จึงควรต่อสายออกมา แล้วทำการต่อวงจรทคลองบนเบรคบอร์คหรือแผงต่อวงจร อุปกรณ์ที่แนะนำในบทนี้คือ **บอร์ด Rpi-I/O40** ที่มีพื้นที่สำหรับการสร้างวงจรหรือ Proto-Area ขนาคใหญ่ ที่เทียบเท่ากับเบรคอร์คขนาค 400 จุดต่อ รวมถึงมีการจัคสรรขาพอร์ตสำหรับการเชื่อมต่อเพื่อทคลองหรือ เรียนรู้ได้สะควกขึ้น รวมทั้งยังมีจุดต่อเซอร์โวมอเตอร์และจุดต่อไฟเลี้ยงสำหรับเซอร์โวมอเตอร์ไว้ให้ด้วย เพื่อช่วยเสริมในการนำแผงวงจรสำหรับทคลองตัวนี้ไปใช้ให้เกิดประโยชน์ที่น่าทึ่งต่อไป

ในรูปที่ 7-3 แสดงหน้าตาของบอร์ด Rpi-I/O40 พร้อมทั้งสายแพที่ใช้ในการเชื่อมต่อกับบอร์ด Raspberry Pi และตัวอย่างการต่อใช้งานจริง



รูปที่ 7-3 บอร์ด Rpi-I/O40 พร้อมสายแพที่ใช้ในการเชื่อมต่อกับบอร์ด Raspberry Pi 3



รูปที่ 7-4 วงจรของ Rpi-I/O40 บอร์ดทดลองและพัฒนาโครงงานด้วย Raspberry Pi 3

7.2.1 วงจรของบอร์ด RPi-I/O40

รูปที่ 7-4 แสดงวงจรของบอร์ด RPi-I/O40 สัญญาณจากกอนเน็กเตอร์ GPIO ของบอร์ด Raspberry Pi 3 จะถูกต่อเข้ามาที่กอนเน็กเตอร์ K1 จากนั้นจะถูกเชื่อมต่อมายังกอนเน็กเตอร์ K2 ถึง K7 โดยขาพอร์ตอินพุตเอาต์พุตหลักๆ จะถูกต่อมายังกอนเน็กเตอร์ K3 และ K6 โดย K3 เป็นกอน เน็กเตอร์ IDC ตัวเมีย 27 ขา แถวคู่ (รวม 54 ขา) และ K6 เป็นกอนเน็กเตอร์ IDC ตัวผู้ 27 ขา แถวเดี่ยว ส่วน K2 และ K5 เป็นจุดต่อไฟเลี้ยง +3.3V, +5V และกราวด์ โดย K2 เป็นกอนเน็กเตอร์ IDC ตัวผู้ 27 ขา แถวเดี่ยว 4 ขา แถวคู่ (รวม 8 ขา) และ K5 เป็นกอนเน็กเตอร์ IDC ตัวผู้ 4 ขา แถวเดี่ยว สำหรับ K4 และ K7 เป็นจุดต่อไอซีหน่วยกวามจำอีอีพรอมอนุกรม ซึ่งมีด้วยกัน 4 ขากือ +3.3V, กราวด์, ID_SCL และ ID_SDA โดย K4 เป็นกอนเน็กเตอร์ IDC ตัวเมีย 4 ขา แถวคู่ (รวม 8 ขา) และ K7 เป็นกอนเน็กเตอร์ IDC ตัวผู้ 4 ขา แถวเดี่ยว



รูปที่ 7–5 แสดงการใช้งานพื้นที่สร้างวงจรของบอร์ด **Rpi-I/O40** (ซ้าย) ปล่อยว่างไว้เพื่อรอรับการบัดกรีอุปกรณ์เพื่อสร้างวงจร (ขวา) ติดตั้งเบรดบอร์ดเพื่อทดลองวงจร

นอกจากนั้นบนบอร์ด RPi-I/O40 ยังเตรียมวงจรเชื่อมต่อเซอร์ โวมอเตอร์ไว้เพิ่มเติมอีก 2 ช่อง โดยไฟเลี้ยงเซอร์ โวมอเตอร์ถูกต่อเข้าที่จุดต่อ K8 มีตัวเก็บประจุ C3 ค่า 470µF 10V เพื่อช่วยในการ รักษาระดับแรงดันไฟเลี้ยงเซอร์ โวมอเตอร์ให้มีความแน่นอน ในขณะที่เซอร์ โวมอเตอร์ทำงาน ส่วน ขาพอร์ตสำหรับขับเซอร์ โวมอเตอร์จะถูกต่อมายังคอนเน็กเตอร์ K9 และ K10 โดยแต่ละตัวจะ ประกอบด้วยคอนเน็กเตอร์ IDC 3 ขาตัวผู้และตัวเมียอย่างละ 1 ตัว หากต้องการใช้ขาพอร์ตของ Raspberry Pi 3 เพื่อขับเซอร์ โวมอเตอร์ ให้ต่อขาพอร์ตมายังจุดต่อ K9 หรือ K10





7.2.2 การใช้งานบอร์ด Rpi-I/O40

บนบอร์ค Rpi-I/O40 พื้นที่ว่างสำหรับสร้างวงจรที่มีจุดบัคกรีและการเชื่อมต่อตรงกับแผงต่อ วงจรหรือเบรคบอร์ค จึงอาจเว้นไว้เพื่อรองรับการบัคกรีสร้างวงจร หรือนำเบรคบอร์คขนาค 400 จุดมาติดตั้งก็ได้ ดังรูปที่ 7-5

ในการใช้งานต้องนำสายแพ 40 เส้นมาเชื่อมต่อระหว่างบอร์ค RPi-I/O40 กับ Raspberry Pi 3 โดยสายแพที่ใช้จะมีหน้าตาและตำแหน่งการยึดคอนเน็กเตอร์เข้ากับสายแพคังรูปที่ 7-6



- มีสวิตช์กดติดปล่อยดับพร้อมใช้งาน ต่อกับขา GPIO13 ทำงานแบบลอจิกต่ำ
- มีสวิตช์แบบจอยสติ๊ก 4 ทิศทางและปุ่มกดกึ่งกลาง ต่อกับขา GPIO6, 17, 18, 22 และ 23
- มีจอแสดงผล OLED 0.96 นิ้ว ความละเอียด 128x64 จุด ใช้การเชื่อมต่อแบบบัส I2C
- มี LED 3 สี่ RGB แบบอนุกรมและโปรแกรมได้เบอร์ WS2812B ต่อกับขาพอร์ต GPIO12
- ขนาด 65 x 56 มม. โดยออกแบบรูปร่างของแผ่นวงจรพิมพ์ให้มีขนาด ระยะห่างของรูยึด ช่องว่างต่างๆ เป็น ไปตามข้อกำหนดของการออกแบบบอร์ด HAT ของ Raspberry Pi Foundation

รูปที่ 7-7 ส่วนประกอบของ LeafHAT บอร์ดเชื่อมต่ออุปกรณ์อินพุตเอาต์พุตสำหรับพอร์ต GPIO ของ Raspberry Pi

7.3 LeafHAT บอร์ดเชื่อมต่ออุปกรณ์อินพุตเอาต์พุตสำหรับพอร์ต GPIO ของ Raspberry Pi

เพื่อให้การทดลองเชื่อมต่ออุปกรณ์ภายนอกกับพอร์ต GPIO ของ Raspberry Pi ง่ายสำหรับนัก ทดลองมือใหม่ และสะดวกสำหรับนักทดลองที่มีประสบการณ์มาแล้ว จึงขอแนะนำบอร์คเชื่อมต่อ อุปกรณ์อินพุตเอาต์พุตที่ชื่อ **LeafHAT** มาใช้เป็นหลักในทุกตัวอย่างการทดลองของหนังสือเล่มนี้ โดย LeafHAT ได้รับการออกแบบภายใต้ข้อกำหนดของการพัฒนาบอร์ด HAT (Hardware Attach on Top) ของ Raspberry Pi Foundation สำหรับติดตั้งเข้ากับพอร์ต GPIO 40 ขาของบอร์ด Raspberry Pi

ในรูปที่ 7-7 แสดงส่วนประกอบและคุณสมบัติทางเทคนิคที่สำคัญของบอร์ด LeafHAT ส่วน วงจรสมบูรณ์แสดงในรูปที่ 7-8



รูปที่ 7-8 วงจรสมบูรณ์ของบอร์ด LeafHAT



รูปที่ 7-9 ตัวอย่างการติดตั้งใช้งานร่วมกับบอร์ด Raspberry Pi 3, แผงวงจรอุปกรณ์อินพุตเอาต์พุต ในแบบต่าง ๆ

การใช้งานร่วมกับ Raspberry Pi 3 ทำได้ง่ายมาก เพียงเสียบบอร์ด LeafHAT เข้ากับคอนเน็ก เตอร์ GPIO 40 ขาของบอร์ด Raspberry Pi 3 ตามทิศทางให้ถูกต้อง ดังรูปที่ 7-9 บอร์ด LeafHAT ก็พร้อม ใช้งานกับ Raspberry Pi 3 แล้ว

บอร์ด LeafHAT ยังใช้งานได้กับบอร์ด Raspberry Pi ทุกรุ่นที่มีคอนเน็กเตอร์ GPIO 40 ขา โดยไม่ต้องคัดแปลงแต่อย่างใด

7.4 การควบคุม GPIO บนบอร์ด Raspberry Pi 3 ด้วยโปรแกรมภาษา Python เบื้องต้น

สำหรับการเขียนโปรแกรมควบคุม GPIO บนบอร์ค Raspberry Pi 3 ทำได้หลายวิธี ไม่ว่าจะ เป็นการใช้โปรแกรมภาษา Python, C/C++, Java หรือแม้แต่ Javascript ก็ตาม

วิธีที่แนะนำในที่นี้ เป็นการสั่งงานผ่านไฟล์สคริปต์ที่สร้างขึ้นด้วยภาษา Python โดยมีไลบรารีที่ ชื่อว่า **RPi.GPIO** เป็นหัวใจหลัก ซึ่งในระบบปฏิบัติการ Raspbian เวอร์ชันใหม่ๆ จะติดตั้งไลบรารี RPi.GPIO มาให้แล้ว จึงไม่จำเป็นต้องติดตั้งเพิ่ม

แต่ถ้ายังไม่ได้ติดตั้ง ให้เข้าสู่หน้าต่างเทอร์มินอล แล้วพิมพ์กำสั่งที่เชลพร้อมพ์ ดังนี้

sudo apt-get install python3-rpi.gpio

7.4.1 การกำหนดขา GPIO

บอร์ค Raspberry Pi 3 จะมีการเรียกใช้งานขา GPIO ในภาษา Python อยู่ 2 แบบคือ อิงตามเลขขา บนบอร์ค (Board) และอิงตามขาของตัวชิป Broadcom (BCM) การเลือกรูปแบบการจัดขาใช้คำสั่ง

> GPIO.setmode(GPIO.BOARD) GPIO.setmode(GPIO.BCM)

ตัวอย่างในหนังสือเล่มนี้ใช้การอ้างอิงขาพอร์ตตามขาของตัวชิป Broadcom (GPIO.BCM)

7.4.2 ติดต่อ GPIO ของบอร์ด Raspberry Pi 3 เพื่อสั่งงานขาพอร์ตเอาต์พุตเบื้องต้น ด้วยภาษา Python

ตัวอย่างแรกของการใช้งานขา GPIO ของบอร์ด Raspberry Pi เบื้องต้น เป็นการใช้งานขา GPIO5 เป็นขาพอร์ตเอาต์พุตเพื่อขับ LED ให้ติดดับสลับกัน หรือทำงานเป็นไฟกะพริบ 1 ดวง

(1) ต่อวงจรเพื่อเชื่อมต่อบอร์ค Raspberry Pi 3 กับ LED ดังวงจรในรูปที่ 7-10 หรือ 7-11 ใน กรณีใช้บอร์ค LeafHAT และ ZX-LED ในการทคลอง

(2) เปิดซอฟต์แวร์ Geany เพื่อสร้างไฟล์สคริปต์ โดยเขียนโปรแกรมที่ 7-1

(3) บันทึกไฟล์สคริปต์ โดยตั้งชื่อว่า LED_Blinking.py

(4) เอ็กซิคิวต์ด้วย Python3 โดยไปที่เมนู Bulid > Execute

หน้าต่าง LX Terminal จะปรากฏขึ้นมาเพื่อแสดงสถานะลอจิกของขาพอร์ต และ LED ที่ต่อกับขาพอร์ต GPIO5 จะติดดับด้วยอัตรา 1 วินาที ถ้าต้องการหยุดการทำงานให้กด Ctrl และ C (กดคีย์ Ctrl ค้างไว้ แล้วกดคีย์ C ตาม)



รูปที่ 7–10 วงจรทดลองใช้งานขาพอร์ต GPIO ของบอร์ด Raspberry Pi 3 อย่างง่ายด้วยการสร้างวงจร ไฟกะพริบ



รูปที่ 7-11 วงจรทดลองใช้งานขาพอร์ต GPIO ของบอร์ด Raspberry Pi 3 อย่างง่ายด้วยการสร้างวงจร ไฟกะพริบ โดยใช้บอร์ด LeafHAT

```
import RPi.GPIO as GPIO
import time
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(5,GPIO.OUT)
while True:
    GPIO.output(5,0)
    print('GPIO5=OFF')
    time.sleep(1)
    GPIO.output(5,1)
    print('GPIO5=ON')
    time.sleep(1)
```

โปรแกรมที่ 7–1 โค้ดภาษา Python สำหรับใช้งานขาพอร์ต GPI05 ของ Rapberry Pi 2 ขับ LED (ใช้ Python3 คอมไพเลอร์)

7.5 ใช้งานพอร์ต GPIO ของ Raspberry Pi 3 เป็นพอร์ตอินพุตเอาต์พุตดิจิตอล

พอร์ตอินพุตดิจิตอลและเอาต์พุตดิจิตอลเป็นคุณสมบัติพื้นฐานของ GPIO ของ Raspberry Pi 3 เป็นการสั่งให้สถานะของขา GPIO เป็นลอจิกสูง (High) หรือลอจิกต่ำ (Low) ในการใช้งานพอร์ตอินพุต เอาต์พุตดิจิตอลทุกครั้งจะต้องกำหนดให้ขา GPIO นั้นๆ ทราบด้วยว่า ทำงานแบบใด เพราะ GPIO ของ Raspberry Pi 3 ไม่สามารถทำงานเป็นพอร์ตอินพุตและเอาต์พุตดิจิตอลได้พร้อมกัน

7.5.1 การกำหนดทิศทางของขาพอร์ต

ตัวอย่างการกำหนดการทำงานให้กับขา GPIO

GPIO.setup(5, GPIO.OUT) กำหนดให้ขา GPIO5 เป็นพอร์ตเอาต์พุตดิจิตอล GPIO.setup(6, GPIO.IN) กำหนดให้ขา GPIO6 เป็นพอร์ตอินพุตดิจิตอล

สำหรับขาพอร์ตอินพุตดิจิตอลยังกำหนดได้ด้วยว่า จะให้มีการต่อพูลอัป (pull up -กำหนดให้ เป็นลอจิกสูง) หรือพูลดาวน์ (pull down - กำหนดให้เป็นลอจิกต่ำ) ให้กับขาพอร์ตในขณะยังไม่มีการ ป้อนสัญญาณเข้ามา โดยกำหนดได้ดังนี้

> <u>สำหรับพูลอัป:</u> GPIO.setup(5, GPIO.IN, pull_up_down=GPIO.PUD_UP) <u>สำหรับพูลดาวน์</u>: GPIO.setup(5, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

7.5.2 พอร์ตเอาต์พุตดิจิตอล

เมื่อต้องการให้ขา GPIO ขับสัญญาณตามที่กำหนด จะต้องกำหนดให้ทำงานเป็นพอร์ตเอาต์พุต ดิจิตอล สัญญาณที่ขับได้ มี 2 สถานะคือ HIGH หรือลอจิกสูง หรือ "1" กับ LOW หรือลอจิกต่ำ หรือ "0"

```
ร<u>ูปแบบคำสั่ง</u>

GPIO.output(ขาพอร์ต, สถานะ)

<u>ตัวอย่างที่ 7-1</u>

GPIO.output(19, GPIO.HIGH)

กำหนดให้ขา GPIO19 ขับสัญญาณลอจิกสูง

GPIO.output(2, GPIO.LOW)

กำหนดให้ขา GPIO2 ขับสัญญาณลอจิกต่ำ
```

7.5.2.1 ทดลองสร้างไฟวิ่ง LED 3 ดวงอย่างง่าย

(1) ต่อวงจรตามรูปที่ 7-12

(2) เปิดซอฟต์แวร์ Geany แล้วสร้างไฟล์สคริปต์ จากนั้นพิมพ์คำสั่งตามโปรแกรมที่ 7-2

(3) ทำการบันทึกไฟล์ LED3Scrolling.py

(4) เอ็กซิคิวต์ด้วย Python3 โดยไปที่เมนู Bulid > Execute หรือรันด้วยคำสั่ง sudo python3 LED6Scrolling.py

เมื่อ โปรแกรมเริ่มทำงาน LED ทั้ง 3 ควงจะติดคับ ในลักษณะ ไฟวิ่ง ถ้าต้องการ ให้หยุด ทำงาน กด Ctrl และ C (กดคีย์ Ctrl ค้าง ไว้ แล้วกดคีย์ C ตาม)



รูปที่ 7-12 วงจรทดลองใช้งานขาพอร์ต GPIO ของบอร์ด Raspberry Pi 3 สร้างวงจรไฟวิ่ง 3 ดวง โดย ใช้บอร์ด LeafHAT และ ZX-LED

```
import RPi.GPIO as GPIO
import time
GPIO.setwarnings(False)
GPIO.setmode (GPIO.BCM)
pin1 = 24
pin2 = 25
pin3 = 26
GPIO.setup(pin1, GPIO.OUT)
GPIO.setup(pin2, GPIO.OUT)
GPIO.setup(pin3, GPIO.OUT)
GPIO.output(pin1, False)
GPIO.output(pin2, False)
GPIO.output(pin3, False)
while(1):
         GPIO.output(pin1, GPIO.HIGH)
         time.sleep(0.3)
         GPIO.output(pin1, GPIO.LOW)
         GPIO.output(pin2, GPIO.HIGH)
         time.sleep(0.3)
         GPIO.output(pin2, GPIO.LOW)
         GPIO.output(pin3, GPIO.HIGH)
         time.sleep(0.3)
         GPIO.output(pin3, GPIO.LOW)
```

```
โปรแกรมที่ 7-2 โค้ดภาษา Python สำหรับใช้งานขาพอร์ต GPIO ของบอร์ด Raspberry Pi 3 เป็นขาพอร์ต
เอาต์พุตดิจิตอลเพื่อขับ LED 3 ดวง
```
7.5.2.2 สร้างใฟวิ่ง LED 3 ดวงแบบกระชับ

(1) ใช้วงจรในรูปที่ 7-12 ในการทคลอง

(2) เปิดซอฟต์แวร์ Geany แล้วสร้างไฟล์สคริปต์ จากนั้นพิมพ์คำสั่งตามโปรแกรมที่ 7-3

(3) ทำการบันทึกไฟล์ชื่อ LED3Scrolling-Short.py

(4) เอ็กซิคิวต์ด้วย Python3 โดยไปที่เมนู Bulid > Execute หรือรันผ่านเทอร์มินอลด้วยคำสั่ง sudo python3 LED3Scrolling-Short.py

เมื่อโปรแกรมเริ่มทำงาน LED ทั้ง 3 ควงจะติดดับในลักษณะไฟวิ่งเหมือนกับ โปรแกรม LED3Scrolling แต่ถ้าพิจารณาจากโค้ดแล้ว จะพบว่า โปรแกรม LED3Scrolling-Short จะสั้น และกระชับกว่า

```
import RPi.GPIO as GPIO
import time
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
pin_set = [24, 25, 26]
for pin in pin_set:
   GPIO.setup(pin, GPIO.OUT)
   GPIO.output(pin, False)
while(1):
   for pin in pin_set:
     GPIO.output(pin, GPIO.HIGH)
     time.sleep(0.3)
     GPIO.output(pin, GPIO.LOW)
```

โปรแกรมที่ 7–3 โค้ดภาษา Python สำหรับใช้งานขาพอร์ต GPIO ของบอร์ด Raspberry PI 3 เป็นขาพอร์ต เอาต์พุตดิจิตอลเพื่อขับ LED 3 ดวง ที่ปรับปรุงจากโปรแกรมที่ 7–2 ให้มีความกระชับมากขึ้น

7.5.3 พอร์ตอินพุตดิจิตอล

เมื่อต้องการสั่งงานให้ขา GPIO รับสัญญาณคิจิตอลจากภายนอก ต้องกำหนดให้ GPIO ทำงานเป็นพอร์ตออินพุตสัญญาณที่รับได้มี 2 สถานะคือ HIGH หรือลอจิกสูง หรือ "1" กับ LOW หรือ ลอจิกต่ำ หรือ "0" โดยพอร์ตอินพุตดิจิตอลมีวิธีรับสัญญาณได้ 2 แบบใหญ่ๆ คือ แบบซิงโครนัส (synchronous) และอะซิงโครนัส (asynchronous)

7.5.3.1 พอร์ตอินพุตดิจิตอลแบบซิงโครนัส

การใช้งานพอร์ตอินพุตดิจิตอลแบบซิงโครนัสมี 2 วิธีคือ

วิธีที่ 1 เป็นการอ่านค่าสถานะ ณ เวลานั้นของขา GPIO เมื่ออ่านค่าได้แล้ว จึงกระทำ คำสั่งต่อไป

วิธีที่ 2 เป็นการหยุครอจนกว่าสัญญาณจะตรงกับที่กำหนคไว้ จึงกระทำคำสั่งต่อไป

(ก) ใช้งานพอร์ตอินพุตดิจิตอลแบบซิงโครนัสด้วยวิธีที่ 1

ในวิธีนี้จะเป็นการอ่านค่าสถานะในปัจจุบันของขา GPIO เมื่ออ่านค่าได้แล้ว จึงกระทำคำสั่ง ต่อไป ต้องใช้คำสั่งต่อไปนี้ในการกำหนดลักษณะการทำงาน

state = GPIO.input(ขาพอร์ต)

```
import RPi.GPIO as GPIO
import time
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
input_pin = 27
GPIO.setup(input_pin, GPIO.IN)
while(1):
   state = GPIO.input(input_pin)
   print (state)
   time.sleep(0.5)
```

โปรแกรมที่ 7-4 โค้ดภาษา Python ไฟล์ SynchronousInput1.py สำหรับอ่านค่าสถานะของขาพอร์ต 27 ของบอร์ด Raspberry Pi 3 เมื่อทำงานเป็นขาพอร์ตอินพุต





<u>ตัวอย่างที่ 7-2</u>

- (1) ต่อวงจรตามรูปที่ 7-13
- (2) เข้าสู่โปรแกรม Geany แล้วเขียนโปรแกรมที่ 7-4
- (3) บันทึกไฟล์ SynchronousInput1.py ทำการคอมไพล์และเอ็กซิคิวต์ เพื่อทดสอบการทำงาน

โปรแกรมก็จะคอยอ่านค่าสถานะของขา GPIO27 ว่า มีสถานะเป็นอย่างไร ถ้ามีสถานะ เป็น High จะแสดงค่าออกมาเป็นเลข 1 และถ้าเป็น Low ก็จะแสดงเป็น 0 โดยจะอ่านค่าทุกๆ 0.5 วินาที แล้วแสดงบนหน้าต่างเทอร์มินอล ทดลองง่ายๆ ด้วยการกดและปล่อยสวิตช์ เมื่อขาพอร์ตต่อ ไฟเลี้ยง +3.3V จะอ่านค่าสถานะเป็น "1" หากต่อลงกราวด์จะแสดงสถานะ "0"

(4) การแสดงผลลัพธ์เป็นเลข 0 และ 1 อาจจะดูเข้าใจยาก ดังนั้นจึงควรปรับเปลี่ยนการแสดง ค่าให้เหมาะสมมากขึ้นด้วยการแสดงเป็นข้อความบอกว่า ปุ่มถูกกดหรือไม่ถูกกด

- (5) เขียนโปรแกรมทดสอบเพิ่มเติมดังโปรแกรมที่ 7-5
- (6) บันทึกไฟล์ชื่อ SynchronousInput2.py ทำการคอมไพล์และเอ็กซิคิวต์ เพื่อทดสอบการทำงาน
- (7) ทดลองกดสวิตช์แล้วปล่อยสลับกัน สังเกตผลการทำงานที่หน้าต่างเทอร์มินอล

เมื่อปุ่มหรือสวิตช์ไม่ถูกกดจะแสดงข้อความว่า Button OK และเมื่อปุ่มหรือสวิตช์ถูก กดจะแสดงข้อความว่า Button Pressed ที่จอแสดงผล ดังรูปที่ 7-14

148 • Raspberry Pi 3 : รู้จักแล:การใช้งานเบื้องตัน

			LXTerr	ninal 📃 🗖 💌]
<u>F</u> ile	<u>E</u> dit	<u>T</u> abs	<u>H</u> elp		
0					-
1					l
1					l
0					l
1					l
0					l
1					l
0					l
0					l
T					l
0					l
0					
1					
1					
0					1
1					
					4

รูปที่ 7–14 ผลการทำงานของโปรแกรมที่ 7–4 เป็นการแสดงสถานะลอจิกของขา 27 ของพอร์ต GPIO ของบอร์ด Raspberry Pi 3 ที่ใช้งานเป็นขาพอร์ตอินพุต

```
import RPi.GPIO as GPIO
import time
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
input_pin = 27
GPIO.setup(input_pin, GPIO.IN)
while(1):
   state = GPIO.input(input_pin)
   if(state == GPIO.HIGH):
      print ("Button Pressed")
   elif(state == GPIO.LOW):
      print ("Button OK")
   time.sleep(0.5)
```

โปรแกรมที่ 7–5 โค้ดภาษา Python ไฟล์ SynchronousInput2.py สำหรับใช้งานพอร์ต GPIO ของบอร์ด Raspberry Pi 3 เป็นอินพุตทำงานแบบซิงโครนัสในลักษณะที่เมื่ออ่านค่าได้แล้ว ให้กระทำคำสั่งต่อไป

-			LXTerminal	. • ×
<u>F</u> ile <u>E</u> dit	<u>T</u> abs	<u>H</u> elp		
Button	0K			^
Button	Press	sed		
Button	Press	sed		
Button	0K			
Button	Press	sed		
Button	Press	sed		
Button	Press	sed		
Button	Press	sed		
Button	0K			
Button	Press	sed		
Button	0K			
Button	0K			
Button	Press	sed		
Button	Press	sed		
Button	Press	sed		
Button	Press	sed		≡
Button	Press	sed		\sim

รูปที่ 7–15 ผลการทำงานของโปรแกรมที่ 7–5 เป็นการแสดงสถานะลอจิกของขา GPIO27 ของพอร์ต GPIO ของบอร์ด Raspberry Pi 3 ที่ใช้งานเป็นขาพอร์ตอินพุตเมื่อมีการกดและปล่อยสวิตช์

```
(ข) ใช้งานพอร์ตอินพุตดิจิตอลแบบซิงโครนัสด้วยวิธีที่ 2
```

เป็นการรอจนกว่าค่าสัญญาณจะตรงกับที่กำหนดไว้ จึงกระทำคำสั่งต่อไป ให้ใช้คำสั่งต่อไปนี้ state = GPIO.wait_for_edge (ขาพอร์ต, ขอบขาของสัญญาณที่ต้องการอ่านค่า) เช่น

<u>ตัวอย่างที่ 7-3</u>

(1) ใช้วงจรจากรูปที่ 7-13 ในการทดลอง เข้าสู่โปรแกรม Geany แล้วเขียนโปรแกรมที่ 7-6
 (2) บันทึกไฟล์ SynchronousInputEdge.py ทำการคอมไพล์และเอ็กซิคิวต์ โดยไปที่เมนู Bulid
 > Execute หรือรันผ่านเทอร์มินอลด้วยคำสั่ง sudo python3 SynchronousInputEdge.py
 โปรแกรมนี้เป็นการรอสัญญาณขอบขาขึ้นที่ขา GPIO27 โดยจะหยุดรอที่คำสั่ง
 GPIO.wait_for_edge เมื่อกดสวิตช์ สถานะที่ขาพอร์ตจะเป็น "1" หรือเกิดสัญญาณขอบขาขึ้น ทำให้
 เงื่อนไขเป็นจริง แสดงสถานะออกทางจอแสดงผล แล้วรอ 0.2 วินาทีก่อนวนกลับไปอ่านค่าใหม่อีกครั้ง

```
import RPi.GPIO as GPIO
import time
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
input_pin = 27
GPIO.setup(input_pin, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
while(1):
    GPIO.wait_for_edge(input_pin,GPIO.RISING)
    print ("Button Pressed")
    time.sleep(0.2)
```

<u>คำอธิบายโปรแกรมเพิ่มเติม</u>

ที่บรรทัด GPIO.setup (input_pin, GPIO.IN, pull_up_down=GPIO.PUD_DOWN) เป็นการกำหนดให้ขาพอร์ตที่เลือก ในที่นี้คือขา GPIO27 จากคำสั่ง input_pin = 27 เป็นอินพุต และมีการกำหนด สถานะลอจิกตั้งต้นของขาพอร์ตเป็นแบบพูลดาวน์หรือเป็น "O"

คำสั่งตรวจสอบสถานะของขาพอร์ตคือ GPIO.wait_for_edge(input_pin, GPIO.RISING) หากพบสัญญาณขอบขาขึ้น (Rising edge) ก็จะทำงานในคำสั่งถัดไป ซึ่งก็คือ คำสั่ง print ("Button Pressed") เพื่อแสดงสถานะของการกดสวิตซ์ไปยังจอแสดงผล

โปรแกรมที่ 7–6 โค้ดภาษา Python ไฟล์ SynchronousInputEdge.py สำหรับใช้งาน GPIO ของ Raspberry Pi 3 เป็น อินพุตทำงานแบบซิงโครนัสในลักษณะรอจนกว่าสถานะของขาพอร์ตตรงกับที่กำหนดไว้ จึงกระทำคำสั่งต่อไป

7.5.3.2 พอร์ตอินพุตดิจิตอลแบบอะซิงโครนัส

การใช้งานพอร์ตอินพุตดิจิตอลแบบนี้เป็นการใช้อีเวนต์ (Event) หรือเหตุการณ์เข้ามาช่วย โดยไม่จำเป็นต้องคอยเขียนกำสั่งวนตรวจสอบก่าตลอดเวลา เพียงกำหนดอีเวนต์หรือเหตุการณ์ที่จะ เกิดขึ้น และเมื่อเกิดอีเวนต์ตามที่กำหนดไว้ โปรแกรมจะทำงานตามชุดกำสั่งในส่วนนั้นให้ทันที

<u>ตัวอย่างที่ 7-4</u>

(1) ยังคงใช้วงจรในรูปที่ 7-13 ในการทดลอง เข้าสู่โปรแกรม Geany แล้วเขียนโปรแกรมที่ 7-7

(2) บันทึกไฟล์ AsynchronousInput.py ทำการคอมไพล์และเอ็กซิคิวต์ เพื่อทดสอบการทำงาน

ผลการทำงานจะเหมือนกับโปรแกรมที่ 7-6 หากแต่ลักษณะของโปรแกรมแตกต่างกัน โดยในโปรแกรมที่ 7-7 นี้กำหนดให้ขาพอร์ตอินพุตทำงานในแบบอะซิงโครนัส

```
import RPi.GPIO as GPIO
import time
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
input_pin = 27
def button_pressed(channel):
    print ("Button Pressed")
GPIO.setup(input_pin, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.add_event_detect(input_pin, GPIO.RISING, callback=button_pressed,
bouncetime=300)
while(1):
    pass
```

<u>คำอธิบายโปรแกรมเพิ่มเติม</u>

ในโปรแกรมนี้สร้างฟังก์ชั่นที่ชื่อว่า button_pressed เตรียมไว้ จากนั้นกำหนดขา GPIO27 ให้เป็น พอร์ตอินพุตดิจิตอล และกำหนดให้มีการต่อพูลดาวน์ไว้ จากนั้นกำหนดเงื่อนไขให้กับขา GPIO27 ว่า ถ้ามีสัญญาณ ที่เป็นขอบขาขึ้น (Rising) ก็จะเรียกให้ฟังก์ชั่น button_pressed ทำงาน และกำหนดระยะเวลาสำหรับดีเบาซ์ (debounce) หรือการแก้สัญญาณรบกวนจากการกดสวิตช์ไว้ที่ **300** มิลลิวินาที

เมื่อกดสวิตช์ จะทำให้มีสัญญาณขอบขาขึ้นที่ขา GPIO27 ฟังก์ชั่น button_pressed ถูกเรียกให้ทำงาน จะแสดงข้อความบนหน้าจอว่า สวิตช์ที่ต่อกับขา GPIO27 ถูกกด ให้สังเกตที่ตัวแปร channel ที่อยู่ในฟังก์ชั่น button_pressed ตัวแปรนี้ก็คือ ขา GPIO ตัวที่ใช้งานนั่นเอง ในกรณีนี้คือขา GPIO27

โปรแกรมที่ 7–7 โค้ดภาษา Python ไฟล์ AsynchronousInput.py สำหรับใช้งาน GPIO ของบอร์ด Raspberry Pi 3 เป็นอินพุตทำงานแบบอะซิงโครนัส เมื่อมีเหตุการณ์หรืออีเวนต์ (Event) ตามที่กำหนดไว้เกิดขึ้น โปรแกรมจะทำงานตามชุดคำสั่งที่กำหนดทันที

7.5.4 ตัวอย่างการใช้งานพอร์ตอินพุตเอาต์พุต

7.5.4.1 สวิตช์ควบคุม LED

ในตัวอย่างนี้เป็นการกำหนดให้ Raspberry Pi 3 อ่านค่าจากขา GPIO27 ที่ต่อกับสวิตช์เพื่อนำ ค่ามาขับ LED ที่ต่อกับขา GPIO5

(1) ต่อวงจรตามรูปที่ 7-16

(2) สร้างไฟล์สคริปต์ด้วย Geanny หรือ nano เท็กซ์เอดิเอตร์

(3) พิมพ์คำสั่งตามโปรแกรมที่ 7-8

(4) ทำการบันทึกไฟล์ชื่อ SwitchLED.py

(5) สั่งให้ไฟล์สกริปต์ทำงาน โดยไปที่เมนู Bulid > Execute หรือรันไฟล์สกริปต์ด้วย Terminal โดยใช้กำสั่ง sudo python3 SwitchLED.py

(6) สังเกตการทำงานของ LED และหน้าจอแสดงผลเมื่อไม่ได้กดสวิตช์ และหลังจากกดสวิตช์

เมื่อยัง ไม่ได้กดสวิตช์ ที่ขา GPIO27 จะมีค่าเป็น 1 และส่งผลให้ ขา GPIO5 ที่กำหนด ไว้ เป็นเอาต์พุต มีสะถานะเป็น 1 ทำให้ LED ที่ต่อกับขา GPIO5 สว่าง และเมื่อกดสวิตช์ สถานะที่ขา GPIO27 จะมีค่าเป็น 0 ส่งผลให้ LED ที่ต่อกับขา GPIO5 ดับ และที่จอแสดงผลจะแสดงข้อความดังนี้

sw=1 หมายความว่าไม่ได้กดสวิตช์

sw=0 หมายความว่า กคสวิตช์

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setup(5,GPIO.OUT)
GPIO.setup(27,GPIO.IN)
while (1):
   state=GPIO.input(27)
   GPIO.output(5,state)
   print("SW=",state)
```

โปรแกรมที่ 7–8 โค้ดภาษา Python ไฟล์ SwitchLED.py สำหรับใช้งาน GPIO ของบอร์ด Raspberry Pi 3 ทั้งอินพุตและเอาต์พุตให้ทำงานร่วมกัน โดยอ่านค่าสถานะของสวิตช์จากพอร์ตอินพุตมาควบคุม การทำงานของ LED ที่ต่อกับขาพอร์ตเอาต์พุต



รูปที่7–16 วงจรทดสอบการทำงานของพอร์ต GPIO เบื้องต้นในลักษณะทำงานร่วมกันทั้งอินพุตและเอาต์พุต



รูปที่ 7–17 วงจรสวิตข์ควบคุมไฟวิ่ง 3 ดวง เป็นวงจรทดสอบการทำงานของพอร์ต GPIO เบื้องต้นในลักษณะ ทำงานร่วมกันทั้งอินพุตและเอาต์พุต

7.5.4.2 สวิตช์ควบคุมไฟวิ่ง LED

ในตัวอย่างนี้เป็นการกำหนดให้ Raspberry Pi 3 อ่านก่าจากขา 27 ที่ต่อกับสวิตช์เพื่อนำสถานะ มาควบคุมการทำงานของไฟวิ่ง LED ที่ต่อกับขา 24, 25 และ 26

(1) ต่อวงจรตามรูปที่ 7-17 จากนั้นสร้างไฟล์สคริปต์ด้วย Geanny หรือ Nano เท็กซ์เอดิเตอร์

(2) พิมพ์คำสั่งตามโปรแกรมที่ 7-9

```
import RPi.GPIO as GPIO
import time
GPIO.setmode (GPIO.BCM)
my pin=[24,25,26]
count pin=0;
count state=1;
input_pin=27
GPIO.setup(input pin, GPIO.IN, pull up down=GPIO.PUD DOWN)
for pin in my pin:
  GPIO.setup(pin,GPIO.OUT)
 count pin+=1
while (\overline{1}):
  GPIO.wait_for_edge(input_pin, GPIO.RISING)
  count state+=1
  state=count state%2
  print("State=", state)
  if state==0:
    for i in range(count pin):
       GPIO.output(my_pin[i],1)
       time.sleep(0.5)
       GPIO.output(my pin[i],0)
  elif state==1:
    for i in range(count pin):
       i=count pin-1-i
       GPIO.output(my_pin[i],1)
       time.sleep(0.5)
       GPIO.output(my pin[i],0)
```

<u>คำอธิบายโปรแกรมเพิ่มเติม</u>

ในโปรแกรมนี้ ใช้สัญญาณขอบขาขึ้นในขณะที่ปล่อยสวิตช์เป็นตัวกำหนดการทำงานของ LED ด้วยคำสั่ง GPIO.wait_for_edge(input_pin, GPIO.RISING) ซึ่งจะรอจนกว่าจะมีสัญญาณขอบขาขึ้นปรากฏ ขึ้นมา จากนั้นเพิ่ม count_state 1 ค่า แล้วหารด้วย 2 โดยใช้คำสั่ง mod(%) เพื่อหาเศษ ผลลัพท์ที่ได้คือ 0 และ 1

นั่นหมายความว่า ถ้า count_state เป็นจำนวนคู่ ผลลัพธ์จะเป็น "O" และถ้า count_state เป็น จำนวนคี่ ผลลัพธ์จะเป็น "1" จากนั้นนำผลลัพธ์มาเป็นเงื่อนไขในการตรวจสอบต่อไป

โปรแกรมที่ 7–9 โค้ดภาษา Python ไฟล์ SwitchLED6Running.py สำหรับใช้งานพอร์ต GPIO ของบอร์ด Raspberry Pi 3 ทั้งอินพุตและเอาต์พุตให้ทำงานร่วมกัน โดยอ่านค่าสถานะของสวิตซ์จากพอร์ตอินพุต มาควบคุมการทำงานของไฟวิ่ง LED 3 ดวงที่ต่อกับขาพอร์ตเอาต์พุต 3 ขา (3) ทำการบันทึกไฟล์ชื่อ SwitchLED3Running.py

(4) สั่งให้ไฟล์สคริปต์ทำงาน โดยไปที่เมนู Bulid > Execute ในหน้าต่างของ Geany หรือรัน ไฟล์สกริปต์ด้วยเทอร์มินอล โดยใช้กำสั่ง sudo python3 SwitchLED3Running.py

(5) สังเกตการทำงานของ LED และจอแสดงผลเมื่อไม่ได้กดสวิตช์ และหลังจากกดสวิตช์ค้างไว้

เมื่อกดสวิตซ์ วงจรไฟวิ่ง LED ยังไม่ทำงาน ทันทีที่ปล่อยสวิตซ์ วงจรไฟวิ่ง LED จะเริ่มทำงาน โดย LED ที่ต่อกับขา GPIO24 จะติดเป็นดวงแรกนาน 0.5 วินาทีแล้วดับลง LED ที่ต่อ กับขา GPIO25 จะติดขึ้นมาแทน ตามด้วย LED ที่ต่อกับขา GPIO 26 แล้วหยุดทำงาน เพื่อรอให้กด และปล่อยสวิตซ์ที่ต่อกับขา GPIO27 อีกครั้ง

7.6 การสร้างสัญญาณ PWM สำหรับ GPIO ของ Raspberry Pi 3

สัญญาณ PWM (Pulse-width Modulation) หรือสัญญาณมอดูเลชั่นทางความกว้างของ พัลส์เป็นสัญญาณดิจิตอลอีกแบบหนึ่งที่เป็นที่นิยมใช้ในการควบคุมความเร็วมอเตอร์ และควบคุม ความสว่างของ LED โดยใช้ช่วงเวลาที่สัญญาณ PWM เป็นลอจิกสูงในการควบคุม โดยแรงดันที่ใช้ ในการควบคุมความเร็วของมอเตอร์หรือความสว่างของ LED จะขึ้นกับความกว้างของสัญญาณพัลส์ บวกของสัญญาณ PWM ดังนั้นหากพัลส์บวกมีความกว้างมาก แรงดันไฟตรงที่ได้จะสูง

7.6.1 แนะนำสัญญาณ PWM

ในรูปที่ 7-18 แสดงตัวอย่างสัญญาณ PWM ซึ่งมีลักษณะเป็นสัญญาณพัลส์ที่มีความกว้างค่า หนึ่งที่สามารถกำหนดได้ โดยมีค่าของคาบเวลาที่คงที่ เนื่องจากสัญญาณ PWM มีทั้งช่วงเวลาที่เป็น ลอจิกสูงและต่ำสลับกัน มากน้อยขึ้นกับการกำหนดความกว้างของพัลส์ ซึ่งเรียกสัดส่วนของสัญญาณ พัลส์บวก (ช่วงที่เป็นลอจิกสูง) กับคาบเวลาทั้งหมดของสัญญาณพัลส์ว่า ดิวตี้ไซเกิล (duty cycle) โดย คิดค่าดิวตี้ไซเกิลเป็นเปอร์เซ็นต์ของค่าความกว้างพัลส์ ทั้งหมด



รูปที่ 7-18 ตัวอย่างของสัญญาณ PWM

156 • Raspberry Pi 3 : รู้จักและการใช้งานเบื้องตัน



รูปที่ 7–19 แสดงตัวอย่างสัญญาณ PWM ที่มีค่าดิวตี้ไซเกิลต่าง ๆ กัน

เมื่อสัญญาณมีลักษณะเป็นพัลส์ดังรูปที่ 7-18 ก็จะทำให้แรงดันไฟตรงของสัญญาณ PWM นี้มี ถูกเฉลี่ยตามอัตราส่วนระหว่างความกว้างพัลส์กับคาบเวลาของสัญญาณหรือดิวตี้ไซเกิล หากค่าของ ดิวตี้ไซเกิลมาก แรงดันไฟตรงเฉลี่ยของสัญญาณ PWM จะสูง ในทางตรงข้าม หากค่าดิวตี้ไซเกิลต่ำ แรงดันเฉลี่ยก็จะต่ำ

ในรูปที่ 7-19 แสดงให้เห็นถึงสัญญาณ PWM ที่มีค่าดิวตี้ไซเกิลต่างๆ กัน จะเห็นว่า คาบเวลา ของสัญญาณจะคงที่ แต่ความกว้างของสัญญาณพัลส์บวกแตกต่างกัน ตัวอย่างจากรูปที่ 7-18 พัลส์ ลูกซ้ายสุดมีค่าดิวตี้ไซเกิล 90% หมายถึง ความกว้างของพัลส์ช่วงบวกมีความกว้างเป็น 90% ของความกว้างทั้งหมด ดังนั้นแรงดันเฉลี่ยที่ได้เท่ากับ (90×3.3)/100=2.97V (ที่ไฟเลี้ยงวงจร +3.3V) และถ้าหากสัญญาณ PWM มีดิวตี้ไซเกิล 50% แรงดันเฉลี่ยจะเท่ากับ 1.65V ลดลงเหลือ 0.99V ที่ดิวตี้ ไซเกิล 30% และ 0.33V ที่ดิวตี้ไซเกิล 10%

ผู้พัฒนาสามารถปรับค่าดิวตี้ไซเกิล, คาบเวลา และความถี่ของสัญญาณ PWM ได้ตามจุด ประสงค์ในการใช้งาน โดยมีการคำนวณเบื้องต้นดังนี้

<u>ตัวอย่างที่ 7-5</u>

สัญญาณ PWM มีคาบเวลา 20 มิลลิวินาที มีความกว้างพัลส์ 5 มิลลิวินาที จะมีดิวตี้ไซเกิล เท่ากับ 5 / 2 x 100% = 25%

<u>ตัวอย่างที่ 7-6</u>

สัญญาณ PWM ความถี่ 4kHz และมีดิวตี้ไซเกิล 18% คำนวณหาคาบเวลาและความกว้าง ของพัลส์ได้ดังนี้

คาบเวลา = 1/4000Hz = 0.25 มิลลิวินาที่หรือ 250 ไมโครวินาที

ความกว้างพัลส์ = 250 x 18% / 100% = 45 ไมโครวินาที

7.6.2 Raspberry Pi 3 กับการสร้างสัญญาณ PWM

ในการสร้างสัญญาณ PWM บน Raspberry Pi 3 นั้น ทำได้ง่ายๆ ด้วยคำสั่ง PWM ที่มีอยู่แล้ว ใน RPi.GPIO สิ่งที่ต้องทำคือ กำหนดขาที่ต้องการขับสัญญาณ PWM, ความถี่ และค่าดิวตี้ไซเกิล

ในซีพียู Broadcom BCM2837 ของ Raspberry Pi 3 มีฮาร์ดแวร์สร้างสัญญาณ PWM แล้ว โดย กำหนดให้ขา GPIO12 กับ 18 เป็นขาเอาต์พุต PWM0 และขา GPIO13 เป็นเอาต์พุต PWM1 หากมี การเลือกใช้งาน แต่ในไลบรารี RPi.GPIO ไม่ได้เรียกใช้ โดยเลือกให้กำเนิดสัญญาณ PWM ด้วยกระบวน การทางซอฟต์แวร์แทน หรือเรียกว่า **PWM ทางซอฟต์แวร์ (Software PWM)** ซึ่งมีข้อดีคือ ทำให้ขาพอร์ต GPIO ทุกขาสร้างสัญญาณ PWM ได้

การกำหนดให้ขาพอร์ต GPIO ของ Raspberry Pi 3 เป็นขาเอาต์พุต PWM จะต้องเรียกใช้โมดูล RPi.GPIO และกำหนดให้ขาพอร์ตที่ต้องการใช้งานเป็นเอาต์พุตก่อน ด้วยกำสั่ง

GPIO.setup (pin, GPIO.OUT) ส่วนคำสั่งสำหรับการสร้างสัญญาณ PWM ประกอบด้วย

7.6.2.1 คำสั่งเลือกขาพอร์ตและความถื่

<u>รูปแบบคำสั่ง</u>

p = GPIO.PWM(pin,frequency)

<u>พารามิเตอร์</u>

p คือ ตัวแปรใดๆ pin คือ ขาพอร์ต GPIO frequency คือ ความถี่ของสัญญาณ PWM ในหน่วย Hz

<u>ตัวอย่างที่ 7-7</u>

p = GPIO.PWM(25,50) # ใช้ขาพอร์ต GPIO25 ขับสัญญาณ PWM ด้วยความถี่ 50Hz

7.6.2.2 คำสั่งเริ่มต้นการสร้างสัญญาณ PWM ด้วยการกำหนดค่าดิวตี้ไซเกิล

<u>รูปแบบคำสั่ง</u>

p.start(dutycycle)

<u>พารามิเตอร์</u>

p คือ ตัวแปรใดๆ dutycycle คือ ค่าดิวตี้ไซเกิล มีค่า 0.0 ถึง 100.0

<u>ตัวอย่างที่ 7-8</u>

p.start = (0)

ไม่มีสัญญาณออกจากขาพอร์ต เนื่องจากค่าดิวตี้ไซเกิล = 0%

p.start = (50)

กำเนิดสัญญาณ PWM ที่มีค่าดิวตี้ไซเกิล 50%

7.6.2.3 คำสั่งเปลี่ยนค่าความถื่

<u>รูปแบบคำสั่ง</u>

p.ChangeFrequrency(frequency)

<u>พารามิเตอร์</u>

p คือ ตัวแปรใดๆ

frequency คือ ความถี่ของสัญญาณ PWM ในหน่วย Hz

<u>ตัวอย่างที่ 7-9</u>

p.ChangeFrequency(100) # เปลี่ยนความถี่ของสัญญาณ PWM เป็น 100Hz

7.6.2.4 คำสั่งเปลี่ยนค่าดิวตี้ไซเกิล

<u>รูปแบบคำสั่ง</u>

p.ChangeDutyCycle(dutycycle)

<u>พารามิเตอร์</u>

p คือ ตัวแปรใดๆ dutycycle คือ ค่าดิวตี้ไซเกิล มีค่า 0.0 ถึง 100.0

<u>ตัวอย่างที่ 7-10</u>

p.ChangeDutyCycle(90) # เปลี่ยนค่าดิวตี้ไซเกิลเป็น 90%

7.6.2.5 คำสั่งหยุดการสร้างสัญญาณ PWM

<u>รูปแบบคำสั่ง</u> p.stop() <u>พารามิเตอร์</u> p คือ ตัวแปรใดๆ

7.6.3 ตัวอย่างการสร้างสัญญาณ PWM ของ Raspberry Pi 3

7.6.3.1 ขับ LED กะพริบด้วยสัญญาณ PWM

(1) ต่อวงจรตามรูปที่ 7-20 โดยต่อแผงวงจร ZX-LED เข้าที่ขา GPIO25 ผ่านบอร์ด LeafHAT

- (2) สร้างไฟล์สคริปต์ด้วย Geanny หรือ nano เท็กซ์เอดิเตอร์ พิมพ์กำสั่งตามโปรแกรมที่ 7-10
- (3) บันทึกไฟล์ชื่อ PWMblink.py แล้วทำการคอมไพล์และเอ็กซิคิวต์เพื่อทคสอบการทำงาน

โปรแกรมนี้จะสร้างสัญญาณ PWM ความถี่ 1Hz มีค่าดิวตี้ไซเกิล 50% ส่งออกไปยังขา GPIO25 โดยสัญญาณ PWM ที่ขับออกมานี้จะทำให้ LED ที่ต่อกับขา GPIO25 กะพริบทุกๆ 0.5 วินาที

```
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setup(25,GPIO.OUT)
blink = GPIO.PWM(25, 1)
print ('Start Blinking')
blink.start(50)
while True:
   pass
```

Generate PWM at GPIO25 with 1Hz frequency # Start PWM by 50% duty cycle





รูปที่ 7-20 วงจรทดลองสร้างสัญญาณ PWM ของบอร์ด Raspberry Pi 3 ผ่านทางขา GPIO25

7.6.3.2 ปรับความสว่างของ LED ด้วยสัญญาณ PWM

(1) ยังคงใช้วงจรตามรูปที่ 7-20 ในการทคลอง

- (2) เข้าสู่โปรแกรม Geany หรือ nano เท็กซ์เอดิเตอร์ แล้วเขียนโปรแกรมที่ 7-11
- (3) บันทึกไฟล์ชื่อ PWMLED.py แล้วทำการคอมไพล์และเอ็กซิคิวต์เพื่อทดสอบการทำงาน

โปรแกรมนี้จะส่งสัญญาณ PWM ออกไปทางขา GPIO25 เพื่อขับให้ LED สว่างเพิ่ม ขึ้นเรื่อยๆ จนสว่างที่สุดที่ค่าดิวตี้ไซเกิล100% และเมื่อสว่างเต็มที่แล้ว สัญญาณ PWM จะลดค่าดิวตี้ ไซเกิลลง ทำให้ LED ค่อยหรี่ๆ ลงจนดับ แล้ววนทำงานในลักษณะนี้ตลอดเวลา หากต้องการให้หยุด ทำงานให้กด Ctrl และ C (กดคีย์ Ctrl ค้างไว้ แล้วกดคีย์ C ตาม)

```
import RPi.GPIO as GPIO
import time
GPIO.setwarnings(False)
GPIO.setmode (GPIO.BCM)
GPIO.setup(25, GPIO.OUT)
blink = GPIO.PWM(25, 1000)
multiply = 1
duty = 0
while True:
  blink.start(duty)
  time.sleep(0.01)
  if duty >= 100:
    multiply = -1
  if duty <= 0:
    multiply = 1
  duty += multiply
  print (duty)
```

โปรแกรมที่ 7-11 โค้ดภาษา Python ไฟล์ PWMLED.py สำหรับใช้งาน GPIO ของบอร์ด Raspberry Pi 3 เพื่อสร้างสัญญาณ PWM สำหรับขับและควบคุมความสว่างของ LED

7.6.3.3 สร้างเสียงเตือนด้วย PWM

- (1) ต่อวงจรตามรูปที่ 7-21
- (2) เข้าสู่โปรแกรม Geany หรือ nano เท็กซ์เอคิเตอร์ แล้วเขียนโปรแกรมที่ 7-12
- (3) บันทึกไฟล์ชื่อ PWMpiezo.py



รูปที่ 7-21 วงจรทดลองสร้างสัญญาณ PWM จากพอร์ต GPIO ของ Raspberry Pi 3 เพื่อขับออกลำโพงเปียโซ

```
import RPi.GPIO as GPIO
import time
from datetime import datetime
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(24,GPIO.OUT)
blink = GPIO.PWM(24,500)
blink.start(0)
while True:
    da=datetime.now()
    microsec=da.microsecond
    if microsec > 700000:
        blink.ChangeDutyCycle(50)
    else:
        blink.ChangeDutyCycle(0)
```

โปรแกรมที่ 7–12 โค้ดภาษา Python ไฟล์ PWMpiezo.py สำหรับใช้งานพอร์ต GPIO ของบอร์ด Raspberry Pi 3 เพื่อสร้างสัญญาณ PWM สำหรับขับเสียงออกทางลำโพงเปียโซ

(4) สั่งให้ไฟล์สคริปต์ทำงาน โดยไปที่เมนู Bulid > Execute หรือรันไฟล์สคริปต์ด้วยการเปิด หน้าต่าง Terminal โดยใช้กำสั่ง sudo python3 PWMpiezo.py

จะ ได้ยินเสียงดังที่ลำโพงเปียโซซึ่งต่อกับขา GPIO24 ทุกๆ 0.7 วินาที และดังอยู่นาน 0.3 วินาที โดยสัญญาณ PWM ที่สร้างขึ้นนี้มีความถี่ 500Hz ส่วนระยะเวลาของการกำเนิดเสียง ใช้ พารามิเตอร์ microsecond ที่มีค่า 0 ถึง 999,999 ในการกำหนดเวลาของการกำหนดเสียง

โดยค่าเวลานี้มาจากคำสั่ง da=datetime.now() เมื่อค่าเวลาของพารามิเตอร์ microsecond น้อยกว่า 700,000 จะทำให้คำสั่ง blink.ChangeDutyCycle(0) ใด้ค่าของดิวตี้ ไซเกิลเท่ากับ 0 จึงไม่มีการขับสัญญาณเสียงเกิดขึ้น เมื่อ ค่าเวลาของพารามิเตอร์microsecond มากกว่า 700,000 คำสั่ง blink.ChangeDutyCycle(50) ทำงาน เปลี่ยนค่าของดิวตี้ไซเกิลเท่ากับ 50% จึงเกิด การขับสัญญาณเสียงขึ้น และมีเสียงออกลำโพงเปียโซอย่างต่อเนื่องจนกระทั่งค่าเวลาของพารามิเตอร์ microsecond กลับเป็น 0 ดังนั้นจึงมีเสียงสัญญาณดังขึ้นนานประมาณ 300,000 ไมโครวินาทีหรือ 0.3 วินาที และเงียบ 0.7 วินาที



^{บทที่ 7} การตั้งด่าให้ชุดดำสั่งทำงานหลังจาก บอร์ด Raspberry Pi บูตร:บบเสร็จสิ้น



ในการใช้งานตามปกติของบอร์ด Raspberry Pi จะมีขั้นตอนคือ จ่ายไฟ รอการบูตระบบปฏิบัติ การ จากนั้นจะเข้าสู่ขั้นตอนการ log in ผู้ใช้งานต้องป้อนชื่อและรหัสผ่านหลัก หากต้องการรันไฟล์ใด ก็เรียกให้ทำงานด้วยกำสั่ง sudo ในกรณีที่ทำงานในโหมดกอมมานไลน์หรือบรรทัดกำสั่ง หรือรันกำสั่ง startx เพื่อเข้าสู่โหมดกราฟิก (มีหน้าตากล้ายๆ วินโดวส์) แล้วก่อยไปเลือกรันโปรแกรมตามต้องการ

อย่างไรก็ดี ด้วยความที่บอร์ด Raspberry Pi มีความเป็นอุปกรณ์ในแบบสมองกลฝังตัวสูง จึงมี ความต้องการให้ระบบเข้าสู่การรันโปรแกรมหรือคำสั่งที่ต้องการในแบบอัตโนมัติ โดยมีการ log in แบบอัตโนมัติและไม่ต้องมีการป้อนคำสั่งเพื่อรันโปรแกรม นั่นคือ ต้องการให้ทำงานเหมือนกับระบบ สมองกลฝังตัวทั่วๆ ไป ที่ทำงานตามโปรแกรมที่กำหนดได้ทันทีหลังจากจ่ายไฟเลี้ยงหรือรีเซตระบบ

เนื่องจากบอร์ด Raspberry Pi มีการทำงานในลักษณะที่มีระบบปฏิบัติการเป็นตัวกำหนดการ ทำงาน หากต้องการให้เกิดการรันคำสั่งหรือโปรแกรมใดๆ อย่างอัตโนมัติ จะต้องมีการกำหนดหรือ ตั้งค่าการทำงานไว้ล่วงหน้า จะเรียกการทำงานในลักษณะนี้ว่า ออโต้สคิปต์ (auto script)

8.1 หลักการ

การทำให้โปรแกรมที่ต้องการเริ่มทำงานหลังจากบอร์ด Raspberry Pi บูตระบบเสร็จสิ้น จะใช้ วิธีกำหนดสลิปต์หรือชุดคำสั่งที่ต้องการให้ทำงานไปเก็บไว้ในส่วนหนึ่งของการบูตระบบ เมื่อระบบ บูตหรือเริ่มต้นทำงาน ก็จะไปเรียกคำสั่งที่เตรียมไว้ขึ้นมาทำงานทันที โดยเป็นการทำงานด้านหลัง (background) โดยระบบปฏิบัติการยังกงรอรับชุดคำสั่งที่ต้องการทำงานอื่นอยู่ด้วย (foreground) ทั้ง นี้เพราะ Raspberry Pi 3 และระบบปฏิบัติการ Raspbian รองรับการทำงานแบบมัลติทาสกิ้ง (mutltasking) จึงทำงานกับกำสั่งหรือโปรแกรมได้ทั้งแบบด้านหลังและตรงหน้าได้ ผู้ใช้งานสามารถหยุด การทำงานของโปรแกรมหรือชุดกำสั่งที่เป็นออโต้สกริปต์นี้ได้ตลอดเวลา

การหยุดโปรแกรมที่ทำงานอยู่เบื้องหลังขอแนะนำ 2 แนวทางคือ

 หยุดด้วยกระบวนการทางฮาร์ดแวร์ เช่น การต่อสวิตช์ หรือต่ออุปกรณ์ตรวจจับไว้ เมื่อ สวิตช์ถูกกด หรือตัวตรวจจับทำงาน ก็จะมีกระบวนการสั่งให้โปรแกรมหยุดทำงานทันที

2. หยุดด้วยกระบวนการทางซอฟต์แวร์



รูปที่ 8–1 วงจรสำหรับการทดลองรันโปรแกรมอัตโนมัติหลังจากที่ Rasapberry Pi บูตระบบเสร็จ

8.2 การสร้างชุดคำสั่งให้รันแบบอัตโนมัติ

(8.2.1) เริ่มต้นด้วยการสร้างชุดคำสั่งที่ต้องการให้ทำงานแบบอัตโนมัติหลังจากบูตระบบปฏิบัติ การเสร็จสิ้นด้วย Geany หรือ nano เท็กซ์อดิเตอร์ก็ได้ โดยในที่นี้เลือกไฟล์สคริปต์ของโปรแกรมภาษา Python ชื่อ LED.py ดังแสดงในโปรแกรมที่ 8-1 มีการต่อวงจรแสดงในรูปที่ 8-1

(8.2.2) บันทึกไฟล์ชื่อ LED.py

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setup(5,GPIO.OUT)
while True :
    GPIO.output(5,True)
    time.sleep(0.3)
    GPIO.output(5,False)
    time.sleep(0.3)
```

โปรแกรมที่ 8–1 ไฟล์ LED.py โปรแกรมภาษา Python สำหรับควบคุมการทำงานของ LED โดยใช้พอร์ต GPIO ของ Raspberry Pi 3 และกำหนดให้โปรแกรมนี้ทำงานทันทีหลังจากการบูตระบบเสร็จสิ้น (8.2.3) แล้วเลือกเมนู **Build > Execute** บนโปรแกรม Geany หรือ เปิดหน้าต่าง Terminal แล้ว สั่งรันด้วยกำสั่ง

```
sudo python3 LED.py
```

้ ต้องจดจำตำแหน่งของใดเร็กตอรี่ที่เก็บโปรแกรม ซึ่งในที่นี้คือ /home/pi/Desktop/LED.py

(8.2.4) เปิด Terminal แล้วทำการแก้ไขไฟล์ rc.local โดยพิมพ์ชุดคำสั่ง

```
sudo nano /etc/rc.local
pi@raspberrypi ~ $ sudo nano /etc/rc.local
```

(8.2.5) จะพบชุคคำสั่งต่างๆ แล้วให้พิมพ์ชุคคำสั่ง

```
sudo python3 /home/pi/Desktop/LED.py &
```

ก่อนบรรทัด exit0 ดังรูป โดยเครื่องหมาย & ต่อท้ายคำสั่งเป็นการบอกว่าคำสั่งนี้ ให้ทำงานอยู่เบื้องหลัง)



(8.2.6) บันทึกการแก้ไขไฟล์โดยกดคีย์ Ctrl ตามด้วย X และ Y จากนั้นกดคีย์ Enter
 (8.2.7) เมื่อแก้ไขไฟล์เสร็จเรียบร้อยแล้ว ทำการรีบูตใหม่ด้วยคำสั่ง sudo reboot

pi@raspberrypi ~ \$ sudo reboot

(8.2.8) เมื่อบอร์ดเริ่มกลับมาทำงานใหม่ จะเห็นว่า LED กะพริบหลังจากการบูตระบบ และ ทำงานตลอดเวลาโดยที่ไม่ต้องพิมพ์คำสั่งให้ทำงาน 166 • Raspberry Pi 3 : รู้จักและการใช้งานเบื้องตัน

8.3 วิธีการหยุดการทำงานหรือ Kill process ที่ทำงานอยู่

มีขั้นตอน โดยสรุปดังนี้

(8.3.1) เชื่อมต่อคอมพิวเตอร์ภายนอกเข้ากับ Raspberry Pi เพื่อทำการรี โมต

(8.3.2) เปิด Terminal ทำการค้นหา Process ID ของโปรแกรมที่ทำงานอยู่ โดยใช้คำสั่ง

```
ps ax | grep {ชื่อไฟล์}
```



(8.3.3) จะเห็นได้ว่า ไฟล์ LED.py มีส่วนประกอบหรือ Process ที่ทำงานอยู่ 3 ตัว ซึ่งจะเห็น เลข Process ID อยู่ด้วย

pi@raspberrypi	~ \$ 1	os ax	grep LED.py
2207 ?	S	0:00	sudo python3 /home/pi/Desktop/LED.py
2216 ?	S	0:00	python3 /home/pi/Desktop/LED.py
2255 pts/0	S+	0:00	grepcolor=auto LED.py
pi@raspberrypi	~ \$		

(8.3.4) ใช้คำสั่ง **ห111** เพื่อหยุดการทำงาน โดยระบุ Process ID ที่ต้องการจะปิด ดังนี้

sudo kill 2207

(8.3.5) หลังจากชุดคำสั่งหยุดทำงานแล้ว ตรวจสอบ Process ที่ทำงานอีกครั้ง จะพบว่าไม่มี Process ของโปรแกรม LED.py หลงเหลืออยู่



การออกจากโปรแกรมหรือหยุดการทำงานของโปรแกรม ยังทำได้โดยใช้ฮาร์ดแวร์เพิ่มเติม อาทิ ใช้สวิตช์ หรือใช้กีย์บอร์ดพิมพ์กำสั่งใดๆ เพราะในทางปฏิบัติจริง ไฟถ์สคริปต์หรือโปรแกรมถูก กำหนดให้ทำงานอยู่เบื้องหลัง ระบบปฏิบัติการยังกงรอรับกำสั่งใหม่อยู่ตลอดเวลา ถึงแม้ว่า ยังกงมี การรันไฟถ์สกริปต์เพื่อทำงานหลักอยู่ก็ตาม

