

หลักสูตรการเขียนโปรแกรม

Coding Language



Raspberry Pi



บริษัท 168 เอ็ดดูเคชั่น จำกัด

เลขที่ 128 อาคารพญาไทพลาซ่า ชั้นที่ 11 ห้องเลขที่ 117

ถนนพญาไท แขวงทุ่งพญาไท เขตราชเทวี กทม. โทร. 02-129-3208



คำนำ

จากนโยบายการศึกษาของกระทรวงศึกษาธิการที่มุ่งเน้นให้นักเรียนมีการเรียนรู้ในภาษาที่ 3 หรือภาษาโค้ดดิ้ง (Coding) โดยเริ่มตั้งแต่ชั้นประถมศึกษาปีที่ 1 เป็นต้นไป เพื่อส่งเสริมให้เกิด I-Innovation หรือ นวัตกรรมที่จะสามารถนำไปใช้ได้จริงและเกิดประโยชน์แก่คนหมู่มาก ทำให้ประเทศไทยมีบุคลากรที่สร้างนวัตกรรมหน้าใหม่ที่สามารถผลิตนวัตกรรมที่จะเปลี่ยนโลกได้ในทุกสาขาอาชีพ ซึ่งการพัฒนาในรูปแบบต่างๆ เหล่านี้จะต้องเป็นบุคคลที่มีความคิดสร้างสรรค์ มีความคิดที่เป็นระบบและเข้าใจในภาษาดิจิทัล เพื่อจะสร้างผู้นำทางด้านเทคโนโลยีใหม่ๆ การเรียนรู้โค้ดดิ้ง (Coding) จึงจำเป็นและสำคัญมากต่อการศึกษาชาติและโรงเรียนต่างๆ จำเป็นต้องพัฒนาการเรียนการสอนโค้ดดิ้งในทุกระดับชั้น

หลักสูตรการเรียนการสอนสำหรับบอร์ด Raspberry Pi นี้จะเป็นการแนะนำเนื้อหาและโปรแกรมที่ใช้สำหรับบอร์ด Raspberry Pi รวมทั้งอุปกรณ์ที่ใช้ร่วมกับบอร์ดเพื่อให้แสดงผลในรูปแบบต่าง ๆ รวมทั้งโครงงานสำหรับการเรียนการสอนในตัวบอร์ด หลังจากที่ผู้เรียนได้เรียนรู้ตั้งแต่หน่วยการเรียนรู้ที่ 1 จนถึงหน่วยการเรียนรู้สุดท้ายแล้วผู้เรียนจะเข้าใจวิธีการใช้งานและสามารถนำไปประยุกต์เป็นโครงงานและการใช้งานจริงร่วมกับโปรแกรมอื่นที่รองรับได้เป็นอย่างดี

ซึ่งบอร์ดราสเบอร์รี่นี้จะเป็นอุปกรณ์ที่เรียกว่าเป็นคอมพิวเตอร์ขนาดเล็กที่สามารถเชื่อมต่อกับจอมอนิเตอร์ คีย์บอร์ด และเมาส์ได้ สามารถนำมาประยุกต์ใช้ในการทำโครงงานทางด้านอิเล็กทรอนิกส์ การเขียนโปรแกรม หรือเป็นเครื่องคอมพิวเตอร์ตั้งโต๊ะขนาดเล็ก ไม่ว่าจะเป็นการทำงาน Spreadsheet Word Processing ท่องอินเทอร์เน็ต ส่งอีเมล หรือเล่นเกมส์ อีกทั้งยังสามารถเล่นไฟล์วิดีโอความละเอียดสูง (High-Definition) ได้อีกด้วย บอร์ด Raspberry Pi รองรับระบบปฏิบัติการลินุกซ์ (Linux Operating System) ได้หลายระบบ เช่น Raspbian (Debian) Pidora (Fedora) และ Arch Linux เป็นต้น โดยติดตั้งบน SD Card บอร์ด Raspberry Pi นี้ถูกออกแบบมาให้มี CPU GPU และ RAM อยู่ภายในชิปเดียวกัน มีจุดเชื่อมต่อ GPIO ให้ผู้ใช้สามารถนำไปใช้ร่วมกับอุปกรณ์อิเล็กทรอนิกส์อื่นๆ ได้อีกด้วย

ซึ่งทางผู้จัดทำหวังเป็นอย่างยิ่งว่าสถานศึกษาที่ได้นำหลักสูตรการเรียนการสอน Coding เรื่องหลักสูตร Raspberry Pi นี้ไปเรียนรู้จะสามารถทำการเรียนการสอนหลักสูตร Coding ได้เป็นอย่างดีและสามารถประยุกต์ใช้รวมทั้งสร้างโครงงานได้เป็นอย่างดี

สารบัญ

เรื่อง	หน้า
ความรู้เบื้องต้นเกี่ยวกับบอร์ด Raspberry Pi และอุปกรณ์เชื่อมต่อ	1
การติดตั้งระบบปฏิบัติการ Raspbian และการตั้งค่าเบื้องต้น	19
แนะนำภาษา Python และ Scratch เบื้องต้น	53
การเขียนโปรแกรมเบื้องต้นเพื่อควบคุม Input – Output ของขา GPIO	89
การใช้งาน Sensor โมดูล และอุปกรณ์ที่เชื่อมต่อบอร์ด Raspberry Pi	128
พัฒนาระบบ Internet Of Things จากบอร์ด Raspberry Pi	250
สร้างหุ่นยนต์เคลื่อนที่ด้วยบอร์ด Raspberry Pi	320

คำอธิบายรายวิชา

วิชา Raspberry Pi กลุ่มสาระการเรียนรู้วิทยาศาสตร์

ระดับชั้นมัธยมศึกษา ภาคเรียนที่ ๑-๒ เวลา ๔๐ ชั่วโมง จำนวน ๑ หน่วยกิต

ศึกษาบอร์ด Raspberry Pi และอุปกรณ์เชื่อมต่อต่างๆ เช่น เซนเซอร์ โมดูลต่างๆ การติดตั้งระบบปฏิบัติการ Raspbian การตั้งค่าบอร์ดเบื้องต้น หลักการพัฒนาโปรแกรมด้วยภาษาไพทอนและ Scratch เบื้องต้น การต่อวงจรไฟฟ้าและการเขียนโปรแกรมควบคุมอุปกรณ์อินพุต-เอาต์พุตของขา GPIO การเขียนโปรแกรมควบคุมเซนเซอร์ โมดูล และอุปกรณ์อื่นๆ ของบอร์ด Raspberry Pi ประยุกต์ใช้เซนเซอร์และโมดูลต่างๆ มาพัฒนาต่อยอดสู่ระบบ Internet of Things (IOT) และการพัฒนาหุ่นยนต์ด้วยบอร์ด Raspberry Pi

โดยใช้กระบวนการฝึกปฏิบัติ กระบวนการคิดวิเคราะห์ นำทักษะการเขียนโปรแกรมภาษาไพทอน และ Scratch มาสร้างโครงงานด้วยบอร์ด Raspberry Pi ได้ด้วยตนเอง

รักการค้นคว้าและวางแผนการทำงานอย่างเป็นระบบ ละเอียดยรอบคอบ มีความคิดสร้างสรรค์ นำเทคโนโลยีมาใช้ในการชีวิตประจำวันอย่างมีจิตสำนึกและมีความรับผิดชอบ

ตัวชี้วัด

ว ๔.๑ ม.๑/๕ ม.๒/๕ ม.๓/๕ ม.๔/๕

ว ๔.๒ ม.๑/๒ ม.๒/๒ ม.๓/๑

รวม ๗ ตัวชี้วัด

ตัวชี้วัดและสาระการเรียนรู้

วิชา Raspberry Pi กลุ่มสาระการเรียนรู้วิทยาศาสตร์

ระดับชั้นมัธยมศึกษา ภาคเรียนที่ ๑-๒ เวลา ๔๐ ชั่วโมง จำนวน ๑ หน่วยกิต

สาระที่ ๔ เทคโนโลยี

มาตรฐาน ว ๔.๑ เข้าใจแนวคิดหลักของเทคโนโลยีเพื่อการดำรงชีวิตในสังคมที่มีการเปลี่ยนแปลงอย่างรวดเร็ว ใช้ความรู้และทักษะทางด้านวิทยาศาสตร์ คณิตศาสตร์ และศาสตร์อื่น ๆ เพื่อแก้ปัญหาหรือพัฒนางานอย่างมีความคิดสร้างสรรค์ด้วยกระบวนการออกแบบเชิงวิศวกรรม เลือกใช้เทคโนโลยีอย่างเหมาะสม โดยคำนึงถึงผลกระทบต่อชีวิต สังคม และสิ่งแวดล้อม

ลำดับ	ตัวชี้วัด	สาระการเรียนรู้แกนกลาง
ม.๑	๕. ใช้ความรู้และทักษะเกี่ยวกับวัสดุ อุปกรณ์ เครื่องมือ กลไก ไฟฟ้า หรือ อิเล็กทรอนิกส์ เพื่อแก้ปัญหาได้อย่างถูกต้องเหมาะสม และปลอดภัย	- การสร้างชิ้นงานอาจใช้ความรู้เรื่องกลไกไฟฟ้า อิเล็กทรอนิกส์ เช่น LED บัชเซอร์ มอเตอร์ วงจรไฟฟ้า - อุปกรณ์และเครื่องมือในการสร้างชิ้นงานหรือพัฒนาวิธีการมีหลายประเภท ต้องเลือกใช้ให้ถูกต้องเหมาะสม และปลอดภัย รวมทั้งรู้จักเก็บรักษา
ม.๒	๕. ใช้ความรู้และทักษะเกี่ยวกับวัสดุ อุปกรณ์ เครื่องมือ กลไก ไฟฟ้า หรือ อิเล็กทรอนิกส์ เพื่อแก้ปัญหาได้อย่างถูกต้องเหมาะสม และปลอดภัย	- การสร้างชิ้นงานอาจใช้ความรู้เรื่องกลไกไฟฟ้า อิเล็กทรอนิกส์ เช่น LED บัชเซอร์ มอเตอร์ วงจรไฟฟ้า - อุปกรณ์และเครื่องมือในการสร้างชิ้นงานหรือพัฒนาวิธีการมีหลายประเภท ต้องเลือกใช้ให้ถูกต้องเหมาะสม และปลอดภัย รวมทั้งรู้จักเก็บรักษา
ม.๓	๕. ใช้ความรู้และทักษะเกี่ยวกับวัสดุ อุปกรณ์ เครื่องมือ กลไก ไฟฟ้า หรือ อิเล็กทรอนิกส์ เพื่อแก้ปัญหาได้อย่างถูกต้องเหมาะสม และปลอดภัย	- การสร้างชิ้นงานอาจใช้ความรู้เรื่องกลไกไฟฟ้า อิเล็กทรอนิกส์ เช่น LED บัชเซอร์ มอเตอร์ วงจรไฟฟ้า - อุปกรณ์และเครื่องมือในการสร้างชิ้นงานหรือพัฒนาวิธีการมีหลายประเภท ต้องเลือกใช้ให้ถูกต้องเหมาะสม และปลอดภัย รวมทั้งรู้จักเก็บรักษา

ม.๔	๕. ใช้ความรู้และทักษะเกี่ยวกับวัสดุ อุปกรณ์ เครื่องมือ กลไก ไฟฟ้า หรือ อิเล็กทรอนิกส์ เพื่อแก้ปัญหาได้อย่างถูกต้องเหมาะสม และปลอดภัย	• การสร้างชิ้นงานอาจใช้ความรู้ เรื่องกลไก ไฟฟ้า อิเล็กทรอนิกส์ เช่น LED บัชเซอร์ มอเตอร์ วงจรไฟฟ้า • อุปกรณ์และเครื่องมือในการสร้างชิ้นงานหรือพัฒนาวิธีการมีหลายประเภท ต้องเลือกใช้ให้ถูกต้องเหมาะสม และปลอดภัย รวมทั้งรู้จักเก็บรักษา
-----	--	---

มาตรฐาน ว ๔.๒ เข้าใจและใช้แนวคิดเชิงคำนวณในการแก้ปัญหาที่พบในชีวิตจริงอย่างเป็นขั้นตอนและเป็นระบบ ใช้เทคโนโลยีสารสนเทศและการสื่อสารในการเรียนรู้ การทำงาน และการแก้ปัญหาได้อย่างมีประสิทธิภาพ รู้เท่าทัน และมีจริยธรรม

ลำดับ	ตัวชี้วัด	สาระการเรียนรู้แกนกลาง
ม.๑	๒. ออกแบบและเขียนโปรแกรมอย่างง่าย เพื่อแก้ปัญหาทางคณิตศาสตร์หรือวิทยาศาสตร์	• การออกแบบและเขียนโปรแกรมที่มีการใช้ตัวแปร เงื่อนไข วนซ้ำ • การออกแบบอัลกอริทึม เพื่อแก้ปัญหาทางคณิตศาสตร์ วิทยาศาสตร์อย่างง่าย อาจใช้แนวคิดเชิงนามธรรมในการออกแบบ เพื่อให้การแก้ปัญหามีประสิทธิภาพ • การแก้ปัญหอย่างเป็นขั้นตอนจะช่วยให้แก้ปัญหาได้อย่างมีประสิทธิภาพ • ซอฟต์แวร์ที่ใช้ในการเขียนโปรแกรม เช่น Scratch, python, java, c • ตัวอย่างโปรแกรม เช่น โปรแกรมสมการการเคลื่อนที่ โปรแกรมคำนวณหาพื้นที่ โปรแกรมคำนวณดัชนีมวลกาย
ม.๒	๒. ออกแบบและเขียนโปรแกรมที่ใช้ตรรกะ และฟังก์ชันในการแก้ปัญหา	• ตัวดำเนินการบูลีน • ฟังก์ชัน • การออกแบบและเขียนโปรแกรมที่มีการใช้ตรรกะ และฟังก์ชัน

		• การออกแบบอัลกอริทึม เพื่อแก้ปัญหาอาจใช้แนวคิดเชิงคำนวณในการออกแบบ เพื่อให้การแก้ปัญหามีประสิทธิภาพ • การแก้ปัญหอย่างเป็นขั้นตอนจะช่วยให้แก้ปัญหาได้อย่างมีประสิทธิภาพ • ซอฟต์แวร์ที่ใช้ในการเขียนโปรแกรม เช่น Scratch, python, java, c • ตัวอย่างโปรแกรม เช่น โปรแกรมตัดเกรดหาคำตอบทั้งหมดของสมการหลายตัวแปร
ม.๓	๑. พัฒนาแอปพลิเคชันที่มีการบูรณาการกับวิชาอื่นอย่างสร้างสรรค์	• ขั้นตอนการพัฒนาแอปพลิเคชัน • Internet of Things (IoT) • ซอฟต์แวร์ที่ใช้ในการพัฒนาแอปพลิเคชัน เช่น Scratch, python, java, c, AppInventor • ตัวอย่างแอปพลิเคชัน เช่น โปรแกรมแปลงสกุลเงิน โปรแกรมผันเสียงวรรณยุกต์ โปรแกรมจำลองการแบ่งเซลล์ ระบบรดน้ำอัตโนมัติ

โครงสร้างรายวิชา

วิชา Raspberry Pi กลุ่มสาระการเรียนรู้วิทยาศาสตร์

ระดับชั้นมัธยมศึกษา ภาคเรียนที่ ๑-๒ เวลา ๔๐ ชั่วโมง จำนวน ๑ หน่วยกิต

ลำดับ	ชื่อหน่วยการเรียนรู้	สาระสำคัญ (key Concept)	มาตรฐานการเรียนรู้/ ตัวชี้วัด	เวลา (ชม.)	น้ำหนัก คะแนน
๑	ความรู้เบื้องต้นเกี่ยวกับบอร์ด Raspberry Pi และอุปกรณ์เชื่อมต่อ	- ความรู้และส่วนประกอบของบอร์ด Raspberry Pi - อุปกรณ์ที่เชื่อมต่อกับตัวบอร์ด	ว ๔.๑ ม.๑/๕ ว ๔.๑ ม.๒/๕ ว ๔.๑ ม.๓/๕ ว ๔.๑ ม.๔/๕	๒	๕
๒	การติดตั้งระบบปฏิบัติการ Raspbian และการตั้งค่าเบื้องต้น	- ขั้นตอนการติดตั้งระบบปฏิบัติการ Raspbian - การตั้งค่าเบื้องต้น	ว ๔.๑ ม.๑/๕ ว ๔.๑ ม.๒/๕ ว ๔.๑ ม.๓/๕ ว ๔.๑ ม.๔/๕	๒	๕
๓	แนะนำภาษา Python และ Scratch เบื้องต้น	- ความรู้พื้นฐานภาษา Python เพื่อใช้เขียนคำสั่งกับบอร์ด Raspberry Pi - ความรู้พื้นฐานภาษา Scratch เพื่อใช้เขียนคำสั่งกับบอร์ด Raspberry Pi	ว ๔.๒ ม.๑/๒ ว ๔.๒ ม.๒/๒	๕	๑๐
๔	การเขียนโปรแกรมเบื้องต้นเพื่อควบคุม Input – Output ของขา GPIO	การเขียนโปรแกรมเพื่อควบคุมอุปกรณ์ INPUT/OUTPUT ที่เชื่อมต่อเข้ากับขา GPIO	ว ๔.๒ ม.๑/๒ ว ๔.๒ ม.๒/๒	๕	๑๐
๕	การใช้งาน Sensor โมดูล และอุปกรณ์ที่เชื่อมต่อบอร์ด Raspberry Pi	- การใช้งาน Sensor - การใช้งานโมดูล - การใช้งานอุปกรณ์อื่นๆ	ว ๔.๒ ม.๑/๒ ว ๔.๒ ม.๒/๒	๑๐	๑๐

๖	พัฒนาระบบ Internet Of Things จากบอร์ด Raspberry Pi	การเขียนโปรแกรมคำสั่งให้กับบอร์ด Raspberry Pi ในการพัฒนาระบบ IOT	ว ๔.๒ ม.๑/๒ ว ๔.๒ ม.๒/๒ ว ๔.๒ ม.๓/๑	๖	๑๐
๗	สร้างหุ่นยนต์เคลื่อนที่ด้วยบอร์ด Raspberry Pi	การเขียนโปรแกรมคำสั่งในการสร้างเงื่อนไขและควบคุมการเคลื่อนที่ของหุ่นยนต์	ว ๔.๒ ม.๑/๒ ว ๔.๒ ม.๒/๒ ว ๔.๒ ม.๓/๑	๖	๑๐
กิจกรรม				๑	๑๐
ข้อสอบกลางภาค				๑	๑๐
ข้อสอบปลายภาค				๒	๒๐
รวม				๔๐	๑๐๐

หน่วยการเรียนรู้ที่ ๑

ความรู้เบื้องต้นเกี่ยวกับบอร์ด Raspberry Pi และอุปกรณ์เชื่อมต่อ

กลุ่มสาระการเรียนรู้วิทยาศาสตร์

สาระที่ ๔ เทคโนโลยี

มาตรฐาน ว ๔.๑ เข้าใจแนวคิดหลักของเทคโนโลยีเพื่อการดำรงชีวิตในสังคมที่มีการเปลี่ยนแปลงอย่างรวดเร็ว ใช้ความรู้และทักษะทางด้านวิทยาศาสตร์ คณิตศาสตร์ และศาสตร์อื่น ๆ เพื่อแก้ปัญหาหรือพัฒนางานอย่างมีความคิดสร้างสรรค์ด้วยกระบวนการออกแบบเชิงวิศวกรรม เลือกใช้เทคโนโลยีอย่างเหมาะสมโดยคำนึงถึงผลกระทบต่อชีวิต สังคม และสิ่งแวดล้อม

ตัวชี้วัด

- | | |
|-------------|--|
| ว ๔.๑ ม.๑/๕ | ใช้ความรู้และทักษะเกี่ยวกับวัสดุ อุปกรณ์ เครื่องมือ กลไก ไฟฟ้า หรืออิเล็กทรอนิกส์ เพื่อแก้ปัญหาได้อย่างถูกต้อง เหมาะสมและปลอดภัย |
| ว ๔.๑ ม.๒/๕ | ใช้ความรู้และทักษะเกี่ยวกับวัสดุ อุปกรณ์ เครื่องมือ กลไก ไฟฟ้า หรืออิเล็กทรอนิกส์ เพื่อแก้ปัญหาได้อย่างถูกต้อง เหมาะสมและปลอดภัย |
| ว ๔.๑ ม.๓/๕ | ใช้ความรู้และทักษะเกี่ยวกับวัสดุ อุปกรณ์ เครื่องมือ กลไก ไฟฟ้า หรืออิเล็กทรอนิกส์ เพื่อแก้ปัญหาได้อย่างถูกต้อง เหมาะสมและปลอดภัย |
| ว ๔.๑ ม.๔/๕ | ใช้ความรู้และทักษะเกี่ยวกับวัสดุ อุปกรณ์ เครื่องมือ กลไก ไฟฟ้า หรืออิเล็กทรอนิกส์ เพื่อแก้ปัญหาได้อย่างถูกต้อง เหมาะสมและปลอดภัย |

สาระสำคัญ

๑. ความรู้และส่วนประกอบของบอร์ด Raspberry Pi
๒. อุปกรณ์ที่เชื่อมต่อกับตัวบอร์ด

สาระการเรียนรู้

ความรู้

- ความเป็นมา ลักษณะ และส่วนประกอบของบอร์ด Raspberry Pi
- อุปกรณ์ต่างๆ ที่เชื่อมต่อเพื่อใช้งานร่วมกับบอร์ด Raspberry pi

ทักษะ/กระบวนการ

- รับรู้ข้อมูลและศึกษาด้วยวิธีการต่างๆ โดยใช้สื่อประกอบเพื่อกระตุ้นการเรียนรู้

- ตอบคำถาม แสดงความคิดเห็นกับสิ่งที่กำหนดเน้นการใช้เหตุผลด้วยหลักการกฎเกณฑ์ อ้างหลักฐานข้อมูลประกอบให้น่าเชื่อถือ
- สังเคราะห์ความรู้ด้วยตนเอง โดยอาจจัดทำในรูปของรายงาน

คุณลักษณะที่พึงประสงค์

- รักษาติ ศาสน์ กษัตริย์
- ซื่อสัตย์
- มีวินัย
- ใฝ่เรียนรู้
- อยู่อย่างพอเพียง
- มุ่งมั่นในการทำงาน
- รักความเป็นไทย
- มีจิตสาธารณะ

หน่วยการเรียนรู้ที่ 1

ความรู้เบื้องต้นเกี่ยวกับบอร์ด Raspberry Pi และอุปกรณ์เชื่อมต่อ

ความเป็นมาของ บอร์ด Raspberry Pi

Raspberry Pi (ออกเสียงว่า ราส-เบอร์-รี่-พาย) เป็นเครื่องคอมพิวเตอร์ขนาดเล็ก ที่มีขนาดเพียงเท่ากับบัตรเครดิต ที่สำคัญคือ Raspberry Pi มีราคาที่ถูกมาก เมื่อเทียบกับคอมพิวเตอร์เดสก์ท็อปปกติ ทำงานได้เหมือนเครื่องคอมพิวเตอร์ทุกอย่าง สามารถต่อ Raspberry Pi เข้ากับจอคอมพิวเตอร์หรือจอทีวีที่รองรับ HDMI หรือถ้าไม่มีพอร์ต HDMI ก็ไม่ต้องก็สามารถต่อผ่านสายสัญญาณวีดีโอปกติ (เส้นสีเหลือง) ได้เช่นกัน แต่ความละเอียดอาจจะต่ำกว่า

นอกจากต่อจอแสดงผลแล้ว ก็ต้องต่ออุปกรณ์รับข้อมูล Raspberry Pi นี้รองรับเมาส์และคีย์บอร์ดผ่าน USB port ปกติ เพราะฉะนั้นสามารถนำเมาส์และคีย์บอร์ดที่มีอยู่แล้วมาต่อได้เลย ระบบจ่ายไฟของ Raspberry Pi เพียงเสียบสาย Mini USB ที่เราใช้ชาร์จมือถือและอุปกรณ์อื่นๆ เข้ากับคอมพิวเตอร์ หรือเข้ากับหัวชาร์จไฟมือถือก็ได้เช่นกัน

(Raspberry Pi) เกิดขึ้นในปี 2549 ที่มหาวิทยาลัยเคมบริดจ์ ประเทศอังกฤษ โดยผู้สร้างทั้งสี่คนคือ อีเบน อัฟตัน, ร็อบ มุลลินส์, แจ็ก แลง และ อลัน มายครอฟท์ มีจุดมุ่งหมายที่จะให้ Raspberry Pi เป็นคอมพิวเตอร์ราคาย่อมเยาที่ใครๆ ก็สามารถหามาครอบครองได้ และสามารถศึกษาการทำงานของคอมพิวเตอร์พร้อมทั้งเขียนโปรแกรมง่ายๆ ได้ทันที การที่ Raspberry Pi เป็นบอร์ดวงจรรวมที่เปลือยเปล่า ทำให้เด็ก ๆ ได้เห็นชิ้นส่วนทั้งหมดที่เป็นส่วนประกอบของคอมพิวเตอร์ได้อย่างชัดเจน ซึ่งจะทำให้เข้าใจการทำงานของคอมพิวเตอร์ในปัจจุบันที่มาในกล่องสวยงามได้มากขึ้น

Raspberry Pi ทำอะไรได้บ้าง

Raspberry Pi กับคอมพิวเตอร์ เช่น

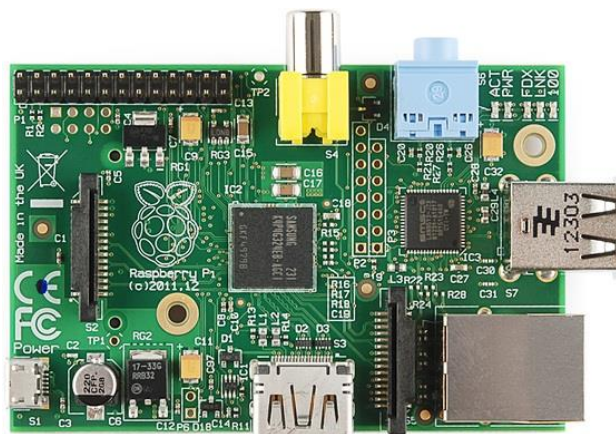
- ใช้เป็นคอมพิวเตอร์ส่วนบุคคล เพื่อดูหนัง ฟังเพลง เล่นอินเทอร์เน็ต
- ใช้เป็นเว็บเซิร์ฟเวอร์ สำหรับเปิดเว็บไซต์ขนาดเล็ก
- ใช้ทำระบบตรวจจับใบหน้า

ส่วนนอกเหนือจากที่คอมพิวเตอร์ทั่วๆ ไปทำได้ก็คงเป็นเรื่องของการสื่อสาร และควบคุมอุปกรณ์ไฟฟ้า อิเล็กทรอนิกส์ต่างๆ เช่น

- ระบบเปิด/ปิดหลอดไฟ หรือเครื่องใช้ไฟฟ้าแบบอัตโนมัติ

- ใช้ควบคุมการทำงานของหุ่นยนต์ หรือเครื่องจักร
- ใช้ทำประตูล็อคไฟฟ้า ล็อค/ปลดล็อคด้วยรีโมท หรือสั่งงานด้วยเสียง
- ใช้ทำสถานีวัดอุณหภูมิ และความชื้นในอากาศ
- ใช้ทำระบบกล้องวงจรปิด

รู้จักกับบอร์ด Raspberry Pi



Raspberry Pi คืออะไร

บอร์ดคอมพิวเตอร์ขนาดเล็กที่สามารถเชื่อมต่อกับจอมอนิเตอร์ คีย์บอร์ด และเมาส์ได้ สามารถนำมาประยุกต์ใช้ในการทำโครงงานทางด้านอิเล็กทรอนิกส์ การเขียนโปรแกรม หรือเป็นเครื่องคอมพิวเตอร์ตั้งโต๊ะขนาดเล็ก ไม่ว่าจะเป็นการทำงาน Spreadsheet Word Processing ท่องอินเทอร์เน็ต ส่งอีเมล หรือเล่นเกมส์ อีกทั้งยังสามารถเล่นไฟล์วีดีโอความละเอียดสูง (High-Definition) ได้อีกด้วย

บอร์ด Raspberry Pi รองรับระบบปฏิบัติการลินุกซ์ (Linux Operating System) ได้หลายระบบ เช่น Raspbian (Debian) Pidora (Fedora) และ Arch Linux เป็นต้น โดยติดตั้งบน SD Card บอร์ด Raspberry Pi นี้ถูกออกแบบมาให้มี CPU GPU และ RAM อยู่ภายในชิปเดียวกัน มีจุดเชื่อมต่อ GPIO ให้ผู้ใช้สามารถนำไปใช้ร่วมกับอุปกรณ์อิเล็กทรอนิกส์อื่นๆ ได้อีกด้วย

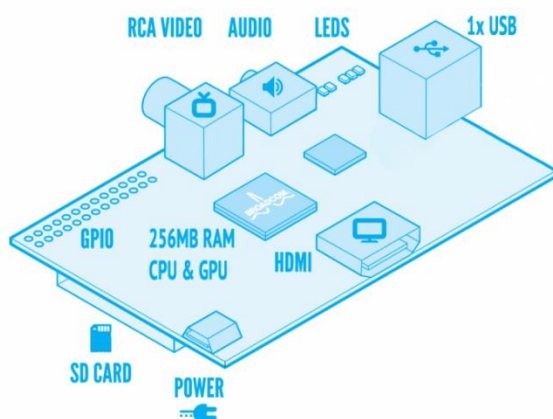
คุณสมบัติทางเทคนิคของบอร์ด

บอร์ด Raspberry Pi ปัจจุบันมีด้วยกัน 2 โมเดล คือ โมเดล A และ โมเดล B ซึ่งทั้ง 2 โมเดลมีคุณสมบัติทางเทคนิคที่ใกล้เคียงกัน แตกต่างกันเพียงบางส่วน รายละเอียดดังตาราง

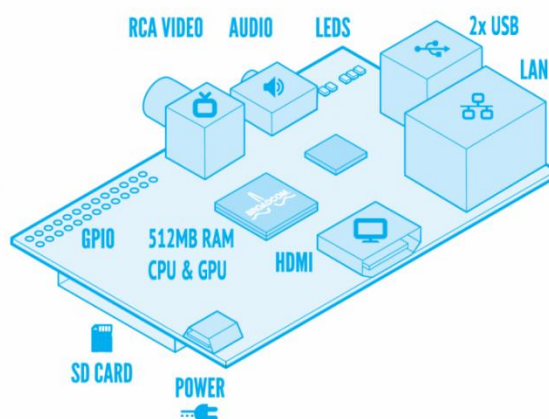
	โมเดล A	โมเดล B (Revision 2)
System on a chip (SoC)	Broadcom BCM2835 (CPU, GPU, DSP, SDRAM and Single USB Port)	
CPU	700MHz ARM1176JZF-S core (ARM11 family, ARMv6 instruction set)	
GPU	Broadcom VideoCore IV @ 250 MHz OpenGL ES 2.0 (24 GFLOPS) MPEG-2 and VC-1, 1080p 30 h.264/MPEG-4 AVC high-profile decoder and encoder	
Memory (SDRAM)	256 MB (Shared with GPU)	512 MB (Shared with GPU)
USB 2.0 Ports	1 (direct form BCM2835)	2 (via the built in integrated 3-port USB hub)
Video Input	A CSI input connector allows for the connection of RPF designed camera module (ออกแบบมาให้เชื่อมต่อกับ Raspberry Pi Camera Module โดยเฉพาะ)	
Video Outputs	Composite RCA (PAL and NTSC), HDMI (rev 1.3 & 1.4), raw LCD Panels via DSI 14 HDMI resolutions from 640x350 to 1920x1200 plus various PAL and NTSC standards. (มีทั้งสองแบบ คือ แบบ RCA และแบบ HDMI)	
Audio Outputs	3.5 mm jack, HDMI, and as of revision 2 boards, I2S audio (also potentially for audio input)	
Onboard storage	SD/ MMC/ SDIO card slot (3.3V card power support only)	
Onboard network	None	10/100 Ethernet (8P8C) USB adapter on the third port of the USB hub
Low-level peripherals Low-level peripherals	8 x GPIO, UART, I2C Bus, SPI Bus with two chip selects, I2S audio +3.3V, +5V, Ground	
Power ratings	300 mA (1.5 W)	700 mA (3.5 W)
Power source	5 Volt via Micro USB or GPIO header	
Size	85.60 mm x 53. Mm (3.370 inch x 2.125 inch)	
Weight	45 g. (1.6 oz.)	

ตัวอย่างโครงสร้างบอร์ด Raspberry Pi ทั้ง 2 โมเดล

RASPBERRY PI MODEL A



RASPBERRY PI MODEL B



ส่วนประกอบของบอร์ด Raspberry Pi (Model B)



1. คอนเน็กเตอร์สำหรับเชื่อมต่อ Input – Output (GPIO), SPI, I2C, I2S และ UART



2. พอร์ตเชื่อมต่อสัญญาณภาพและเสียง Audio Video (AV) แบบ TSR สำหรับจอภาพที่มีระบบ Audio Video (AV)
3. LED แสดงสถานะของบอร์ด Raspberry Pi
4. คอนเน็กเตอร์ USB 2.0 จำนวน 4 พอร์ต
5. คอนเน็กเตอร์ LAN/100 Mbps สำหรับเชื่อมต่อเครือข่ายระบบอินเทอร์เน็ต และระบบอินทราเน็ต
6. คอนเน็กเตอร์ CSI สำหรับเชื่อมต่อกับโมดูลกล้อง
7. LAN Controller
8. คอนเน็กเตอร์ HDMI สำหรับต่อสัญญาณภาพและเสียง ใช้เชื่อมต่อสัญญาณภาพและเสียงยังจอภาพที่มีขั้วแบบ HDMI
9. ชิป SoC Broadcom BCM2837 โดยมีการรวม CPU, GPU และ SDRAM ไว้ในตัวถึงเดียวกัน สำหรับ CPU คือ ARM Cortex A53 ARMv8 แบบ Quad-Core ขนาด 64-bit ใช้ความถี่ที่ 1.2GHz
10. คอนเน็กเตอร์ DSI สำหรับเชื่อมต่อจอภาพ และเชื่อมต่อ GPIO Extension Board
11. คอนเน็กเตอร์ Micro USB สำหรับจ่ายไฟเลี้ยงบอร์ด
12. คอนเน็กเตอร์ SD Card (ด้านใต้บอร์ด)
13. ชิปสำหรับเชื่อมต่อ Wi-Fi IEEE 802.11n และ Bluetooth 4.1, Bluetooth Low-Energy (LE)

อุปกรณ์ที่ใช้กับบอร์ด Raspberry Pi

1. บอร์ด Raspberry Pi 3 B+



บอร์ด Raspberry Pi 3 B+ เป็นบอร์ดคอมพิวเตอร์ขนาดเล็กที่สามารถเชื่อมต่อกับจอมอนิเตอร์ คีย์บอร์ด และเมาส์ได้ สามารถนำมาประยุกต์ใช้ในการทำโครงการทางด้านอิเล็กทรอนิกส์ การเขียนโปรแกรม หรือเป็นเครื่องคอมพิวเตอร์ตั้งโต๊ะขนาดเล็ก ไม่ว่าจะเป็นการทำงาน Spreadsheet Word Processing ท่องอินเทอร์เน็ต ส่งอีเมลล์ หรือเล่นเกมส์ อีกทั้งยังสามารถเล่นไฟล์วิดีโอความละเอียดสูง (High-Definition) ได้อีกด้วย

บอร์ด Raspberry Pi รองรับระบบปฏิบัติการลินุกซ์ (Linux Operating System) ได้หลายระบบ เช่น Raspbian (Debian) Pidora (Fedora) และ Arch Linux เป็นต้น โดยติดตั้งบน SD Card บอร์ด Raspberry Pi นี้ ถูกออกแบบมาให้มี CPU GPU และ RAM อยู่ภายในชิปเดียวกัน มีจุดเชื่อมต่อ GPIO ให้ผู้ใช้สามารถนำไปใช้ร่วมกับอุปกรณ์อิเล็กทรอนิกส์อื่นๆ ได้อีกด้วย

2. บอร์ดขับเคลื่อนมอเตอร์ L298N



บอร์ดขับเคลื่อนมอเตอร์ L298N เป็นชุดขับเคลื่อนมอเตอร์ชนิด H-Bridge เป็นโมดูลที่ใช้ในการควบคุมความเร็วและทิศทางของมอเตอร์ และยังสามารถนำไปประยุกต์ใช้กับโปรเจกต์อื่นๆ ได้อีกด้วย เช่น ปรับค่าความสว่างของหลอดไฟ

H-Bridge เป็นวงจรที่สามารถใช้ควบคุมกระแสได้ทั้งชั่ววอกและลบด้วยการควบคุม pulse width modulation (PWM)

PWM หมายถึง การควบคุมช่วงจังหวะการทำงานของอิเล็กทรอนิกส์ ลองจินตนาการถึงแปรงขดลวดในมอเตอร์เป็นระหัดวิดน้ำและอิเล็กทรอนิกส์เป็นน้ำที่ตกลงมาจากระหัดวิดน้ำ ค่าแรงดันไฟฟ้าก็คล้ายกับกระแสน้ำที่ไหลผ่านระหัดวิดน้ำด้วยความเร็วคงที่ ยิ่งกระแสน้ำไหลเร็วเท่าไรก็จะหมายความว่าแรงดันไฟฟ้ายิ่งสูงขึ้น แต่มอเตอร์มีอัตราความเร็วคงที่และสามารถเสียหายได้หากมีแรงดันไฟฟ้าสูงไหลผ่านหรือหยุดทันทีเพื่อที่จะหยุดมอเตอร์ ดังนั้น PWM คล้ายกับการควบคุมระหัดวิดน้ำให้ตักน้ำในจังหวะคงที่ที่กระแสน้ำคงที่ ยิ่งระหัดวิดน้ำหมุนเร็วเท่าไรช่วงของ pulse ก็จะยาวขึ้น ในทางกลับกันถ้าระหัดวิดน้ำหมุนช้าช่วงของ pulse จะสั้นลง ดังนั้นเพื่อยืดอายุการใช้งานของมอเตอร์จึงควรที่จะควบคุมมอเตอร์ด้วย PWM

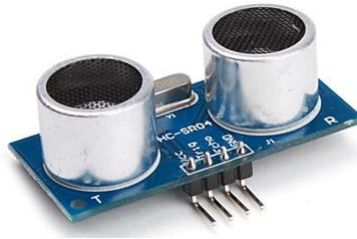
3. Servo 180 องศา



Servo Motor คือระบบควบคุมที่ประกอบด้วยไฟฟ้าคอนโทรลและเครื่องกล ใช้สำหรับงานที่ต้องการควบคุมตำแหน่งความเร็ว แรงบิด ความแม่นยำ และความรวดเร็ว เพื่อให้เครื่องกลและไฟฟ้าคอนโทรลทำงานสอดคล้องกันอย่างมีประสิทธิภาพ โดยใช้การควบคุมแบบป้อนกลับ (Feedback Controller) ซึ่งคือระบบควบคุมที่มีการวัดค่าเอาต์พุตของระบบ แล้วนำมาเปรียบเทียบกับค่าอินพุตเพื่อควบคุมและปรับแต่งให้ค่าเอาต์พุตของระบบให้มีค่า เท่ากับหรือใกล้เคียงกับค่าอินพุต

Servo คืออุปกรณ์มอเตอร์ ที่สามารถควบคุมการหมุนที่แม่นยำ เซอร์โว SG90 มีขนาดเล็กแรงบิด 1.2-1.4 kg/cm มีสาย 3 เส้นคือ สีน้ำตาลเป็นสายกราวด์ สีแดงเป็นไฟเข้า 4.8-7.2V สีส้มเป็นสัญญาณอินพุต หมุน 0-180 องศา

4. โมดูลวัดระยะทาง HC-SR04



HC-SR04 เป็นโมดูลวัดระยะทางที่ใช้หลักการสะท้อนของคลื่นอัลตราโซนิก โดยตัว HC-SR04 มีแหล่งกำเนิดคลื่นอัลตราโซนิกส่งไปสะท้อนกับวัตถุที่อยู่ข้างหน้ากลับมายังตัวรับสัญญาณ โดยระยะทางที่วัดได้จะสัมพันธ์กับระยะเวลาที่คลื่นอัลตราโซนิกเคลื่อนที่ไปกระทบวัตถุและสะท้อนกลับมายังตัวรับ เมื่อรู้ระยะเวลาที่คลื่นอัลตราโซนิกสะท้อนกลับมา จึงนำมาคำนวณหาเป็นระยะทางระหว่างโมดูล HC-SR04 กับวัตถุได้ โดยโมดูล HC-SR04 วัดระยะทางในช่วง 2 ถึง 500 ซม. (5 เมตร) มีความละเอียดอยู่ที่ 0.3 ซม. ใช้ไฟเลี้ยง +5V

5. IR Reflective Sensor



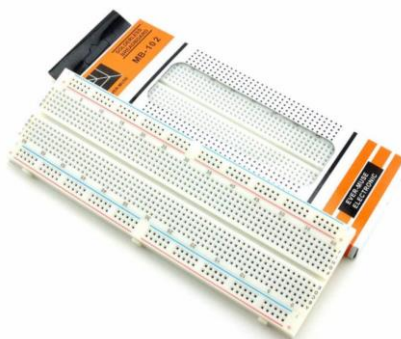
IR Reflective Sensor เป็นตัวรับและตัวส่ง infrared ในตัว ตัวสัญญาณ infrared จะส่งสัญญาณออกมาเมื่อมีวัตถุมาบัง คลื่นสัญญาณ infrared ที่ถูกส่งออกมาจะสะท้อนกลับไปเข้าตัวรับสัญญาณ สามารถนำมาใช้ตรวจจับวัตถุที่อยู่ตรงหน้าได้ และสามารถปรับความไว ระยะการตรวจจับ ใกล้หรือไกลได้

ภายในตัวเซ็นเซอร์นี้จะมีตัวส่ง Emitter และ ตัวรับ Receiver ติดตั้งภายในตัวเดียวกัน ทำให้ไม่จำเป็นต้องเดินสายไฟทั้งสองฝั่ง ทำให้การติดตั้งใช้งานได้ง่ายกว่า แต่อย่างไรก็ตามจำเป็นต้องติดตั้งตัวแผ่นสะท้อนหรือ Reflector ไว้ตรงข้ามกับตัวเซ็นเซอร์เอง โดยโฟโต้เซ็นเซอร์แบบที่ใช้แผ่นสะท้อนแบบนี้จะเหมาะสำหรับชิ้นงานที่มีลักษณะทึบแสงไม่เป็นมันวาว เนื่องจากอาจทำให้ตัวเซ็นเซอร์เข้าใจผิดว่าเป็นตัวแผ่นสะท้อน และ ทำให้ทำงานผิดพลาดได้

เซ็นเซอร์นี้จะมีช่วงในการทำงาน หรือ ระยะในการตรวจจับจะได้ใกล้กว่าแบบ Opposed mode ซึ่งในสภาวะการทำงานปกติตัวรับ Receiver จะสามารถรับสัญญาณแสงจากตัวส่ง Emitter ได้ตลอดเวลา เนื่องจากลำแสงจะสะท้อนกับแผ่นสะท้อน Reflector อยู่ตลอดเวลา จะแสดงค่า เป็น 1

หน้าที่หลักของเซ็นเซอร์ชนิดนี้ จะคอยตรวจจับวัตถุที่เคลื่อนที่ตัดผ่านหน้าเซ็นเซอร์ เมื่อวัตถุ หรือ ชิ้นงานผ่านเข้ามาที่หน้าเซ็นเซอร์ แล้วจะการขวางลำแสงที่ส่งจากตัวส่ง Emitter ที่ส่งไปยังแผ่นสะท้อน จึง ทำให้ตัวรับ Receiver ไม่สามารถรับลำแสงที่จะสะท้อนกลับมาได้ จะแสดงค่า เป็น 0

6. Breadboard 830 ช่อง



Breadboard เป็นอุปกรณ์ที่จะช่วยให้สามารถเชื่อมต่อวงจรเพื่อทดลองง่ายขึ้น ลักษณะของบอร์ดจะเป็นพลาสติกมีรูจำนวนมาก ภายใต้รูเหล่านั้นจะมีการเชื่อมต่อถึงกันอย่างมีรูปแบบ เมื่อนำอุปกรณ์อิเล็กทรอนิกส์มาเสียบ จะทำให้พลังงานไฟฟ้าสามารถไหลจากอุปกรณ์หนึ่ง ไปยังอุปกรณ์หนึ่งได้ ผ่านรูที่มีการเชื่อมต่อกันด้านล่าง พื้นที่การเชื่อมต่อกันของ Breadboard จะแบ่งได้เป็น 2 กลุ่มใหญ่ คือ

กลุ่มแนวตั้ง เป็นกลุ่มที่เป็นพื้นที่สำหรับการเชื่อมต่อวงจร วางอุปกรณ์ จะมีช่องเว้นกลางกลุ่มสำหรับเสียบไอซีตัวถังแบบ DIP และบ่งบอกการแบ่งเขตเชื่อมต่อ

กลุ่มแนวนอน เป็นกลุ่มที่มีการเชื่อมต่อกันในแนวนอน ใช้สำหรับพักไฟที่มาจากแหล่งจ่ายเพื่อใช้สำหรับเชื่อมต่อไฟจากแหล่งจ่ายเลี้ยงให้วงจรต่อไป และจะมีสี สัญลักษณ์สกรีน เพื่อบอกขั้วของแหล่งจ่ายที่ควรนำมาพักไว้ โดยสีแดง จะหมายถึงขั้วบวก และสีดำหรือสีน้ำเงิน จะหมายถึงขั้วลบ

7. LCD Display



LCD ย่อมาจาก Liquid Crystal Display ซึ่งเป็นจอแสดงผลแบบ (Digital) โดยภาพที่ปรากฏขึ้นเกิดจากแสงที่ถูกปล่อยออกมาจากหลอดไฟด้านหลังของจอภาพ (Back Light) ผ่านชั้นกรองแสง (Polarized

filter) แล้ววิ่งไปยัง คริสตัลเหลวที่เรียงตัวด้วยกัน 3 เซลล์คือ แสงสีแดง แสงสีเขียวและแสงสีน้ำเงิน กลายเป็นพิกเซล (Pixel) ที่สว่างสดใสเกิดขึ้น เทคโนโลยีที่พัฒนามาใช้กับ LCD นั้นแบ่งออกเป็น 2 ประเภทคือ

- Passive Matrix หรือที่เรียกว่า Super-Twisted Nematic (STN) เป็นเทคโนโลยีแบบเก่าที่ให้ความคมชัดและความสว่างน้อยกว่า ใช้ในจอโทรศัพท์มือถือทั่วไปหรือจอ Palm ขาวดำเป็นส่วนใหญ่

- Active Matrix หรือที่เรียกว่า Thin Film Transistors (TFT) สามารถแสดงภาพได้คมชัดและสว่างกว่าแบบแรก ใช้ในจอมอนิเตอร์หรือโน้ตบุ๊ก

8. Buzzer Module



Buzzer Module พาสซีฟบัสเซอร์ เป็นอุปกรณ์ทำหน้าที่คล้ายลำโพง แต่ออกแบบมาเพื่อใช้ในการส่งสัญญาณเสียงความถี่สูงๆ บนบอร์ดมีทรานซิสเตอร์สำหรับช่วยขับมาให้แล้ว สามารถต่อใช้งานกับไมโครคอนโทรลเลอร์ได้โดยตรง การใช้งานเพียงเขียนโค้ดสร้างสัญญาณ HIGH LOW สลับกันไปมา ข้อดีของบัสเซอร์แบบพาสซีฟคือสามารถกำหนดความถี่เสียงที่ต้องการได้เอง

9. Relay module HL-52R



Relay module HL-52R เป็นอุปกรณ์ไฟฟ้าชนิดหนึ่ง ซึ่งทำหน้าที่ตัดต่อวงจรแบบเดียวกับสวิตช์ โดยควบคุมการทำงานด้วยไฟฟ้า Relay มีหลายประเภท ตั้งแต่ Relay ขนาดเล็กที่ใช้ในงานอิเล็กทรอนิกส์ทั่วไป จนถึง Relay ขนาดใหญ่ที่ใช้ในงานไฟฟ้าแรงสูง โดยมีรูปร่างหน้าตาแตกต่างกันออกไป แต่มีหลักการ

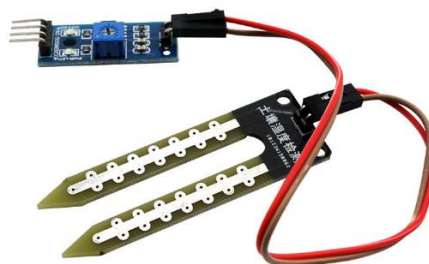
ทำงานที่คล้ายคลึงกัน สำหรับการนำ Relay ไปใช้งาน จะใช้ในการตัดต่อวงจร ทั้งนี้ Relay ยังสามารถเลือกใช้งานได้หลากหลายรูปแบบ

10. เซนเซอร์วัดอุณหภูมิและความชื้น DHT22



เซนเซอร์วัดอุณหภูมิและความชื้นสัมพัทธ์ เป็นเซนเซอร์วัดอุณหภูมิและความชื้นสัมพัทธ์แบบดิจิทัล และเชื่อมต่อด้วยสัญญาณเพียงเส้นเดียวแบบสองทิศทาง (bidirectional) ใช้แรงดันไฟเลี้ยงได้ในช่วง 3.3V ถึง 5.2V สามารถวัดค่าอุณหภูมิได้ในช่วง -40 ถึง 80°C ความละเอียดในการวัดอุณหภูมิและความชื้นคือ 0.5°C และ 0.1%RH

11. เซนเซอร์วัดความชื้นในดิน



เซนเซอร์วัดความชื้นในดิน ใช้วัดความชื้นในดิน หรือใช้เป็นเซ็นเซอร์น้ำ สามารถต่อใช้งานกับไมโครคอนโทรลเลอร์โดยใช้อินพุตอินพุตอ่านค่าความชื้น หรือเลือกใช้สัญญาณดิจิทัลที่ส่งมาจากโมดูลสามารถปรับความไวได้ด้วยการปรับ Trimpot ใช้หลักการตรวจสอบประจุของวัสดุ ถ้ามีค่าประจุมากแสดงว่าแสดงว่าชื้นมาก เนื่องจากใช้หลักการของการวัดประจุ ดังนั้นแผ่นเซนเซอร์จึงไม่ต้องสัมผัสกับดินหรือวัสดุโดยตรง จึงทนทานและแม่นยำกว่า โมดูลวัดความชื้นในดินนี้ ให้ค่าออกเป็นค่า Analog

12. Key Pad 4X4



Keypad เป็นหนึ่งในอุปกรณ์อินพุตที่นิยมใช้ในงานด้านระบบสมองกลฝังตัวมากที่สุด เราสามารถพบเห็นการใช้งาน Keypad ได้จากตู้อัตโนมัติต่าง ๆ เช่น ตู้เติมเงินมือถือ ตู้ซื้อขนมปัง ตู้ ATM ซึ่งแต่ละตู้นำมาใช้เพื่อให้ผู้ใช้บริการกรอกข้อมูลที่เป็นตัวเลขเข้าไป เช่น หมายเลขโทรศัพท์ จำนวนเงิน หมายเลขสินค้า หมายเลขบัญชี รหัสผ่านบัตร ATM แม้เราจะพบเห็นการใช้งาน Keypad ได้ทั่วไป แต่การนำ Keypad มาต่อใช้งานกลับยากจนน่าประหลาดใจ

Keypad มักจะบอกขนาดเป็นจำนวนแถว x จำนวนคอลัมน์ เช่น มี 1 แถว 4 คอลัมน์ มักจะเขียน 1x4 หรือมี 4 แถว 3 คอลัมน์ มักจะเขียน 4x3 และ 4 แถว 4 คอลัมน์ จะเขียนเป็น 4x4

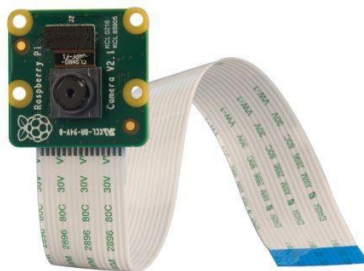
13. เซนเซอร์วัดแก๊สและควัน



เซนเซอร์วัดแก๊สและควัน เป็นโมดูลตรวจวัดแก๊ส ที่ไวต่อแก๊สไวไฟในกลุ่ม LPG, i-butane, propane, methane ,alcohol, Hydrogen รวมไปถึงควันไฟที่เกิดจากการเผาไหม้ด้วย จึงเป็นเซนเซอร์ที่นิยมนำมาใช้ในการตรวจจับการรั่วของแก๊สต่างๆ เพื่อป้องกันอันตรายที่จะเกิดขึ้นจากการรั่วไหลนั้นได้

- ใช้แรงดัน 5V
- ให้เอาที่พุดทั้งสัญญาณอนาล็อกซึ่งเป็นค่าที่วัดได้จริง และสัญญาณดิจิตอลสามารถปรับตั้งระดับแจ้งเตือนได้ (ใช้ LM393 เป็นวงจรเปรียบเทียบแรงดัน)
- เมื่อป้อนแรงดันให้แก่เซนเซอร์ ต้องรอการอุ่นชิพอย่างน้อย 20 วินาที ก่อนทำการวัดค่า

14. กล้อง Raspberry Pi Camera Module v2



Raspberry Pi Camera Board เป็นโมดูลกล้องที่ออกแบบมาใช้งานร่วมกับบอร์ด Raspberry Pi โดยสามารถเชื่อมต่อกับซอกเก็ต CAMERA บนบอร์ด ที่มีการเชื่อมต่อแบบ CSI bus ได้ทันที และยังเหมาะสำหรับงานวิดีโอความละเอียดสูงและการถ่ายภาพนิ่ง นอกจากนี้คุณยังสามารถถ่ายภาพแบบ time-lapse และ slow-motion

นอกจากนี้ยังมีฟังก์ชันการควบคุมอัตโนมัติเช่นการควบคุมการเปิดรับความสมดุลสีแสงขาว (white balance) และการตรวจจับความสว่างอีกด้วย

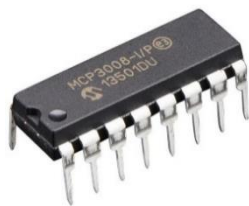
15. LDR Sensor



แอลดีอาร์(LDR) หรือชื่อเต็มๆคือ Light Dependent คือ ความต้านทานชนิดที่ไวต่อแสง ตัวต้านทานนี้สามารถเปลี่ยนสภาพทางความนำไฟฟ้า ได้เมื่อมีแสงมาตกกระทบ บางครั้งเรียกว่าโฟโตริซิสเตอร์ (Photo Resistor) หรือ โฟโตคอนดักเตอร์ (Photo Conductor) เป็นตัวต้านทานที่ทำมาจากสารกึ่งตัวนำ (Semiconductor) ประเภทแคดเมียมซัลไฟด์ (Cds : Cadmium Sulfide) หรือแคดเมียมซีลีไนด์ (CdSe : Cadmium Selenide) ซึ่งทั้งสองตัวนี้ก็เป็นสารประเภทกึ่งตัวนำ เอามาฉาบลงบนแผ่นเซรามิกที่ใช้เป็นฐานรองแล้วต่อขาจากสารที่ฉาบ ให้ออกมา

การทำงานของ LDR เพราะว่าเป็นสารกึ่งตัวนำ เวลาที่มีแสงตกกระทบลงไปก็จะถ่ายทอดพลังงานให้กับสาร ที่ฉาบอยู่ ทำให้เกิดโฮลกับอิเล็กตรอนวิ่งกันพล่าน. การที่มีโฮล กับอิเล็กตรอนอิสระนี้มากก็เท่ากับ ความต้านทานลดลงนั่นเอง ยิ่ง ความเข้มของแสงที่ตกกระทบมากเท่าไร ความต้านทานก็ยิ่งลดลงมากเท่านั้น

16. Chip แปลงสัญญาณ Analog to Digital (MCP3008)



MCP3008 (Microchip) เป็น Chip แปลงสัญญาณ Analog ให้เป็นสัญญาณ Digital ด้วยจำนวน 8 Channels 10 Bit ADC โดยต่อผ่าน SPI interface MCP3008 นี้สามารถต่อใช้งานร่วมกับบอร์ด microcontroller อาทิเช่น Raspberry Pi ซึ่งตัวบอร์ด Raspberry Pi สามารถรับได้แต่สัญญาณ Digital เท่านั้น แต่สัญญาณที่เข้ามาเป็น Analog จึงต้องใช้ MCP3008 แปลงสัญญาณให้ก่อน

17. Micro SD Card



สำหรับ SD Card ที่เหมาะสมนำมาใช้กับบอร์ด RPI จะต้องมีหน่วยความจำไม่น้อยกว่า 8GB (Raspberry Linux) โดยเฉพาะถ้าต้องการลงระบบปฏิบัติการ Windows 10 ควรใช้ SD Card ที่มีหน่วยความจำตั้งแต่ 16-32 GB ขึ้นไป จึงจะทำงานได้ดี “ไม่อืด”

18. สาย AC Adapter With USB Slot



สำหรับอะแดปเตอร์นี้ หัวที่ใช้เสียบป้อนไปเข้าบอร์ด RPI จะต้องเป็นแบบ Micro USB ซึ่งจะคล้ายกับสายชาร์จสมาร์ทโฟนบางรุ่นบางยี่ห้อ

19. HDMI Cable

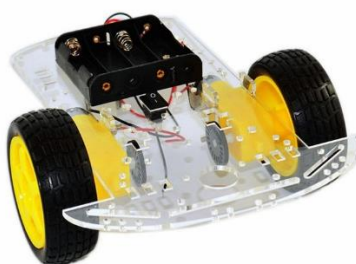


สำหรับสาย HDMI จะใช้ร่วมกับจอมอนิเตอร์ที่มีขั้วเสียบ HDMI ซึ่งจะสามารถส่งสัญญาณภาพและเสียงจากบอร์ด RPI ไปยังจอมอนิเตอร์ โดยหากจอมอนิเตอร์เป็นรุ่นเก่าที่ไม่มีขั้วเสียบ HDMI (มีแต่ VGA) อาจใช้หัวแปลง HDMI เป็น VGA มาใช้แทนได้ หรือหากจะใช้บอร์ด RPI ร่วมกับเครื่องรับโทรทัศน์ที่มีเพียงระบบ AV (Audio Video) ก็สามารถใช้ได้โดยการใช้สายแจ็ค TSR (เข้า RCA (ออก) หรือเข้า 1 (TSR) ออก 3 (RCA))

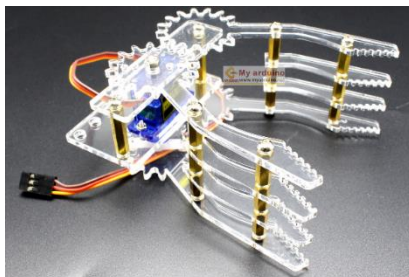
20. Case ใส่บอร์ด Raspberry Pi



21. ตัวรถ Smart Robot car พร้อมกระบอกน้ำและสวิทช์เปิด/ปิด



22. แขนกล



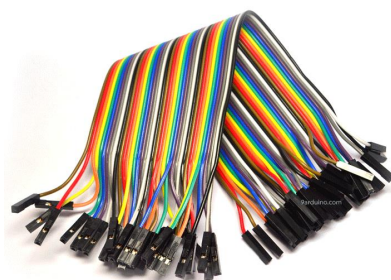
23. สาย Jumper ตัวผู้-ตัวผู้



24. สาย Jumper ตัวผู้-ตัวเมีย



25. สาย Jumper ตัวเมีย-ตัวเมีย



26. ตัวต้านทาน 330 โอห์ม



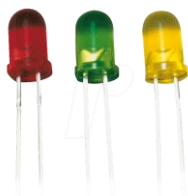
27. ตัวต้านทาน 4.7 กิโลโอห์ม



28. ปุ่มกด Pushbutton



29. หลอดไฟ LEDs



30. ถ่านขนาด AA Recharge



31. Power Bank ชาร์จไฟผ่าน USB



32. Remote Control



33. ที่ชาร์จถ่านขนาด 9V, AA และ AAA



34. ถ่าน 18650



หน่วยการเรียนรู้ที่ ๒ การติดตั้งระบบปฏิบัติการ Raspbian และการตั้งค่าเบื้องต้น

กลุ่มสาระการเรียนรู้วิทยาศาสตร์

สาระที่ ๔ เทคโนโลยี

มาตรฐาน ว ๔.๑ เข้าใจแนวคิดหลักของเทคโนโลยีเพื่อการดำรงชีวิตในสังคมที่มีการเปลี่ยนแปลงอย่างรวดเร็ว ใช้ความรู้และทักษะทางด้านวิทยาศาสตร์ คณิตศาสตร์ และศาสตร์อื่น ๆ เพื่อแก้ปัญหาหรือพัฒนางานอย่างมีความคิดสร้างสรรค์ด้วยกระบวนการออกแบบเชิงวิศวกรรม เลือกใช้เทคโนโลยีอย่างเหมาะสมโดยคำนึงถึงผลกระทบต่อชีวิต สังคม และสิ่งแวดล้อม

ตัวชี้วัด

- | | |
|-------------|--|
| ว ๔.๑ ม.๑/๕ | ใช้ความรู้และทักษะเกี่ยวกับวัสดุ อุปกรณ์ เครื่องมือ กลไก ไฟฟ้า หรืออิเล็กทรอนิกส์ เพื่อแก้ปัญหาได้อย่างถูกต้อง เหมาะสมและปลอดภัย |
| ว ๔.๑ ม.๒/๕ | ใช้ความรู้และทักษะเกี่ยวกับวัสดุ อุปกรณ์ เครื่องมือ กลไก ไฟฟ้า หรืออิเล็กทรอนิกส์ เพื่อแก้ปัญหาได้อย่างถูกต้อง เหมาะสมและปลอดภัย |
| ว ๔.๑ ม.๓/๕ | ใช้ความรู้และทักษะเกี่ยวกับวัสดุ อุปกรณ์ เครื่องมือ กลไก ไฟฟ้า หรืออิเล็กทรอนิกส์ เพื่อแก้ปัญหาได้อย่างถูกต้อง เหมาะสมและปลอดภัย |
| ว ๔.๑ ม.๔/๕ | ใช้ความรู้และทักษะเกี่ยวกับวัสดุ อุปกรณ์ เครื่องมือ กลไก ไฟฟ้า หรืออิเล็กทรอนิกส์ เพื่อแก้ปัญหาได้อย่างถูกต้อง เหมาะสมและปลอดภัย |

สาระสำคัญ

๑. การติดตั้งระบบปฏิบัติการ Raspbian
๒. การตั้งค่าพื้นฐานของบอร์ด Raspberry Pi

สาระการเรียนรู้

ความรู้

- ระบบปฏิบัติการ Raspbian และขั้นตอนการติดตั้ง
- ขั้นตอนการตั้งค่าพื้นฐานของบอร์ด Raspberry Pi

ทักษะ/กระบวนการ

- ปฏิบัติตามตัวอย่างที่แสดงให้เห็นทีละขั้นตอนจากขั้นพื้นฐานไปสู่ขั้นที่ซับซ้อนขึ้น

- ฝึกปฏิบัติตั้งแต่ต้นจนจบด้วยตนเอง
- กระบวนการกลุ่ม มุ่งให้ผู้เรียนทำงานร่วมกันเป็นทีม

คุณลักษณะที่พึงประสงค์

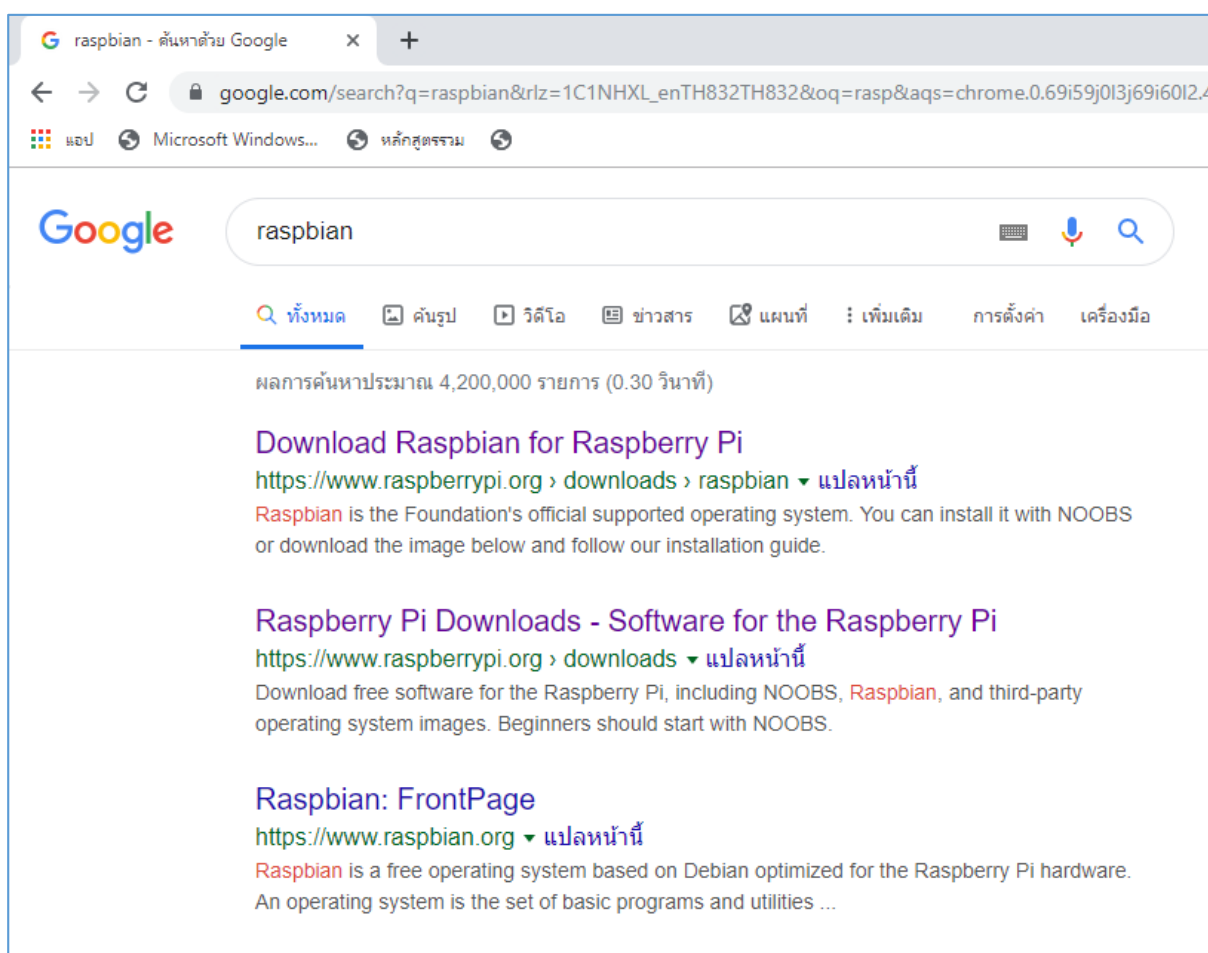
- รักษาติ ศาสน์ กษัตริย์
- ซื่อสัตย์
- มีวินัย
- ใฝ่เรียนรู้
- อยู่อย่างพอเพียง
- มุ่งมั่นในการทำงาน
- รักความเป็นไทย
- มีจิตสาธารณะ

หน่วยการเรียนรู้ที่ 2

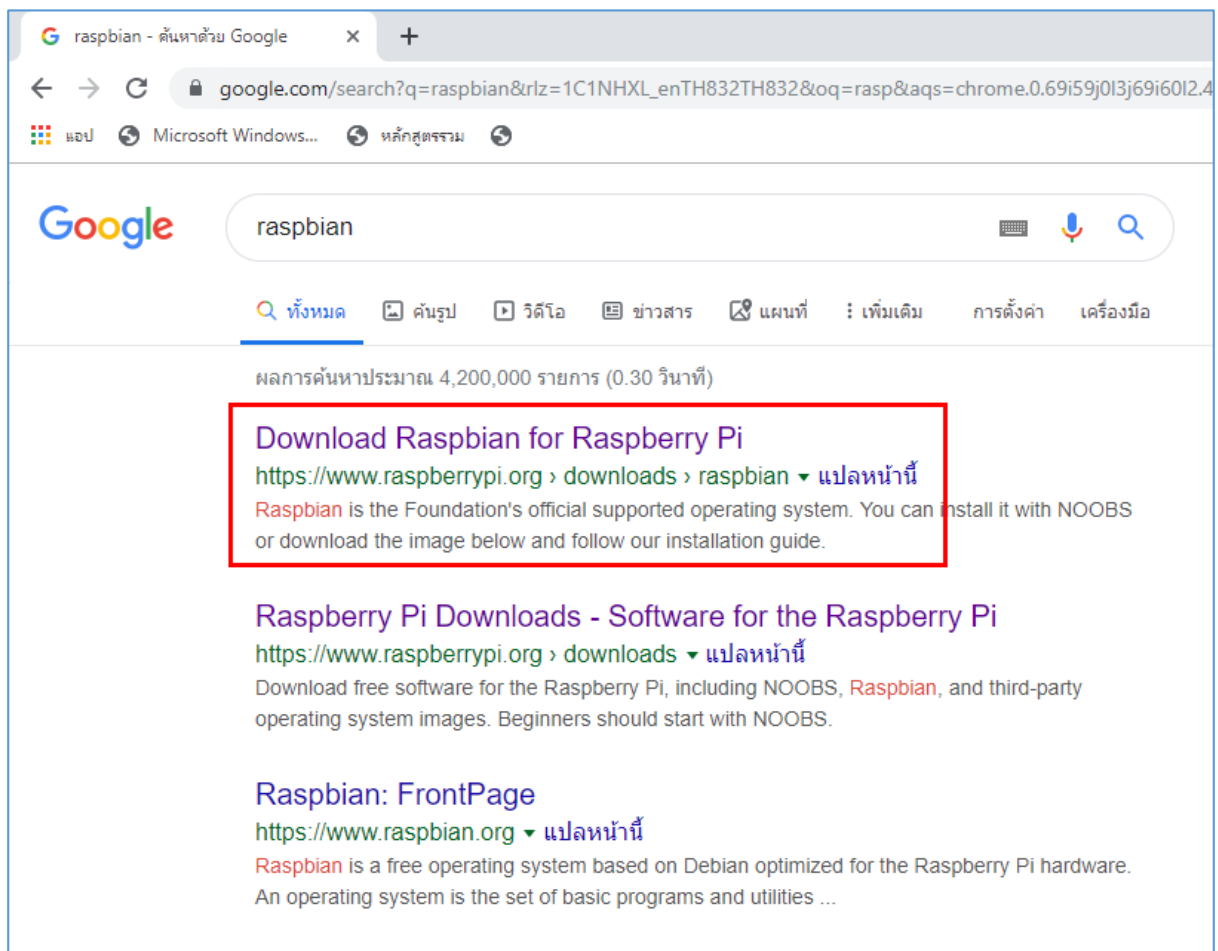
การติดตั้งระบบปฏิบัติการ Raspbian และการตั้งค่าเบื้องต้น

การดาวน์โหลดระบบปฏิบัติการ Raspbian

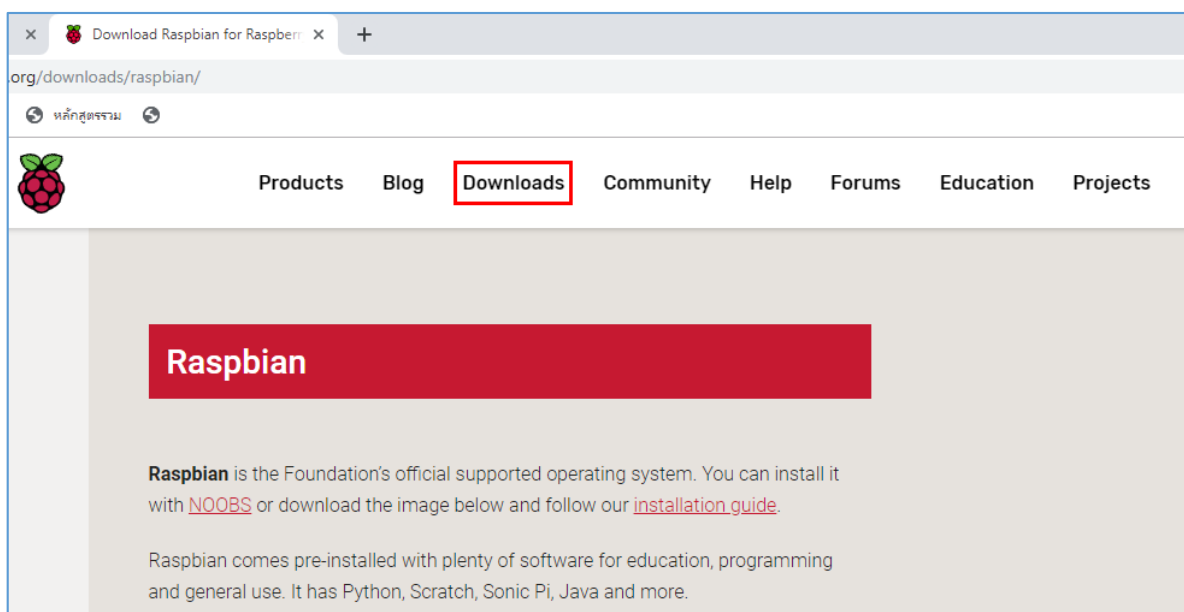
1. ใช้เครื่องคอมพิวเตอร์ในการดาวน์โหลดระบบปฏิบัติการ Raspbian
2. เสียบ SD Card ที่เครื่องคอมพิวเตอร์
3. เปิด Web Brower แล้วค้นหาคำว่า “raspbian”



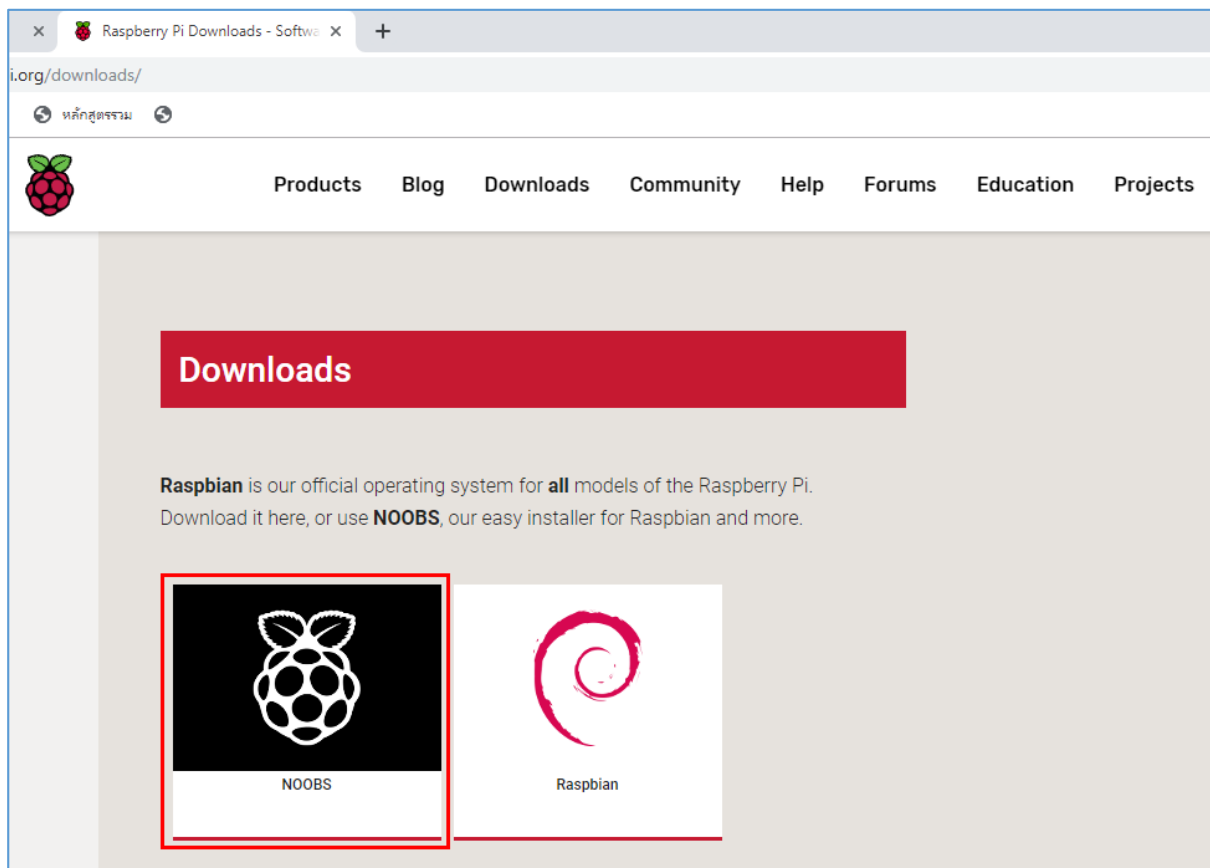
4. คลิกเข้าเว็บไซต์ raspberrypi.org เพื่อเข้าไปดาวน์โหลด Raspbian



5. เมื่อเข้ามาในเว็บไซต์ raspberrypi.org ให้คลิกเลือกเมนู Downloads



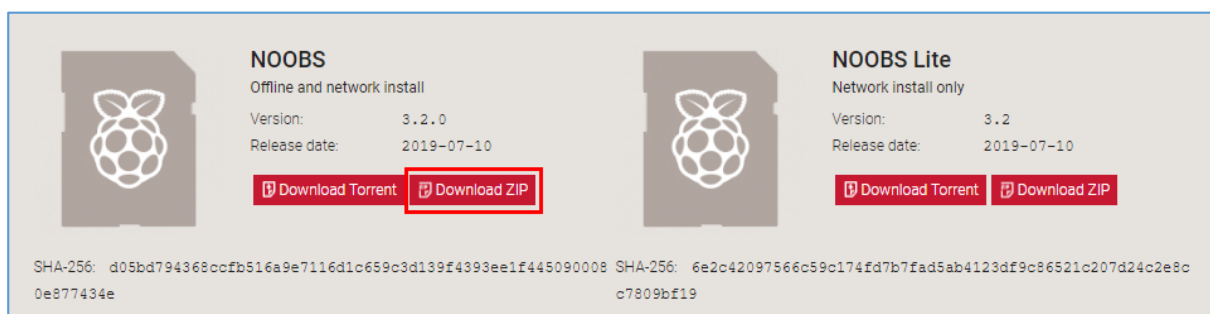
6. การดาวน์โหลดจะมี 2 แบบ คือ ดาวน์โหลด NOOBS และ Raspbian ให้ทำการคลิกเลือกดาวน์โหลดแบบ NOOBS ซึ่งจะมีขั้นตอนที่ง่ายกว่าสำหรับผู้เริ่มต้นใช้งานบอร์ด Raspberry Pi



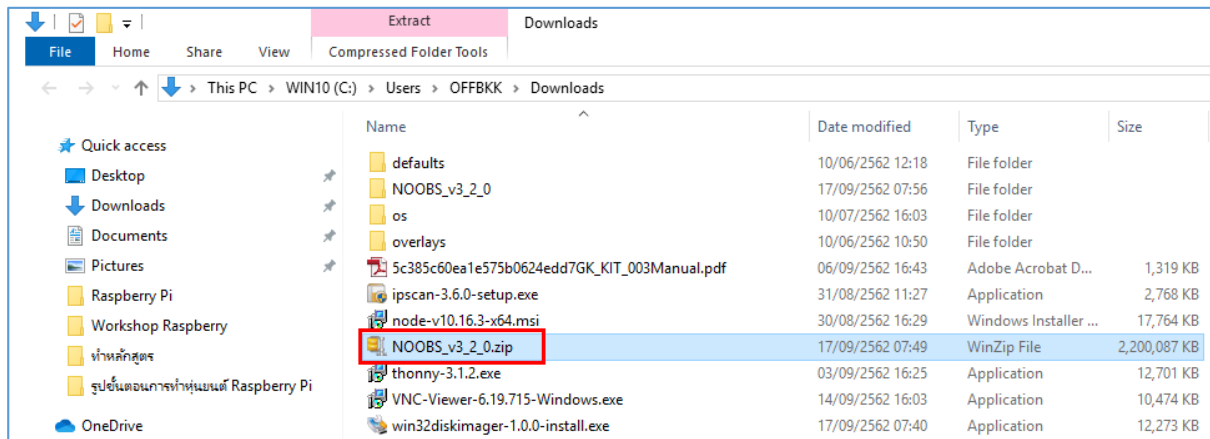
7. ไฟล์ที่จะดาวน์โหลดมี 2 แบบ คือ NOOBS และ NOOBS Lite

- NOOBS จะเป็นการติดตั้งแบบ Offline
- NOOBS Lite จะเป็นการติดตั้งแบบ Online

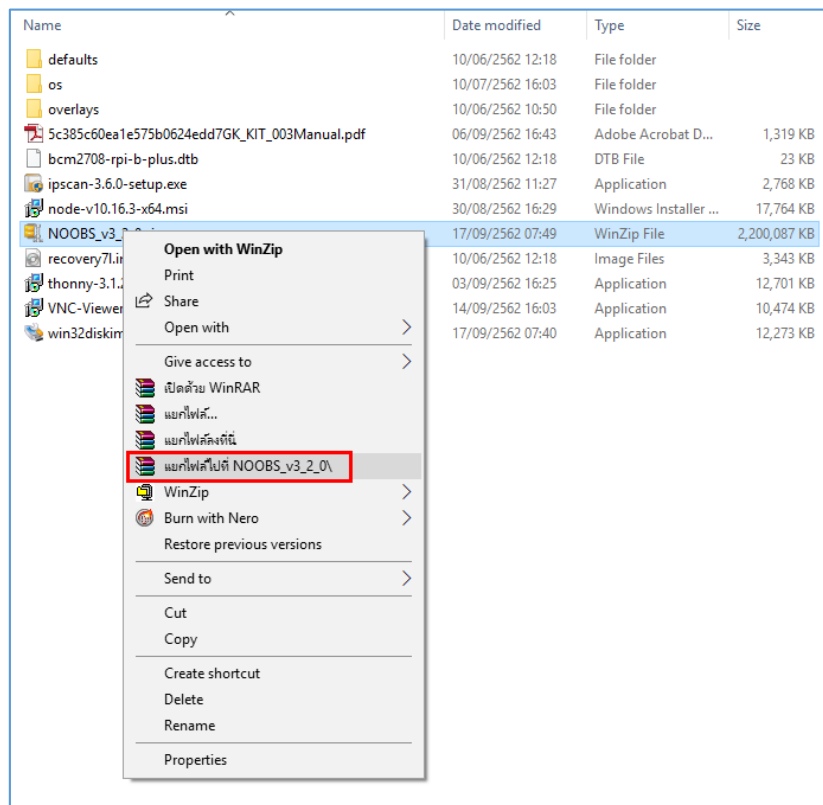
ให้เลือกดาวน์โหลดไฟล์ NOOBS เพื่อติดตั้งระบบปฏิบัติการแบบ Offline โดยดาวน์โหลดเป็นไฟล์ ZIP คลิกเลือกที่ Download ZIP



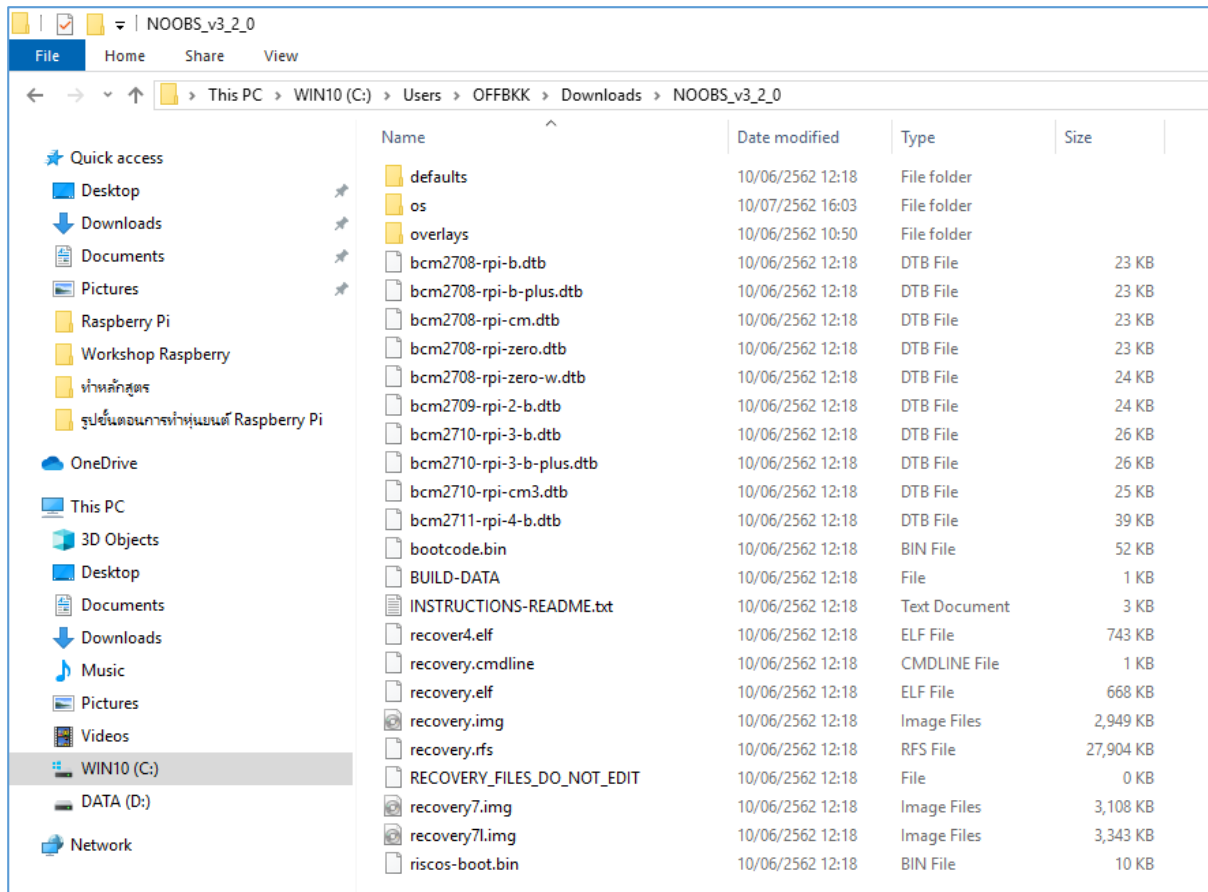
8. ไฟล์ที่ดาวน์โหลดมาจะมีขนาด 2.1 GB ซึ่งจะใช้เวลาในการดาวน์โหลดค่อนข้างนาน ขึ้นอยู่กับความเร็วอินเทอร์เน็ต โดยไฟล์ที่ดาวน์โหลดเสร็จแล้วจะถูกบันทึกในโฟลเดอร์ Downloads



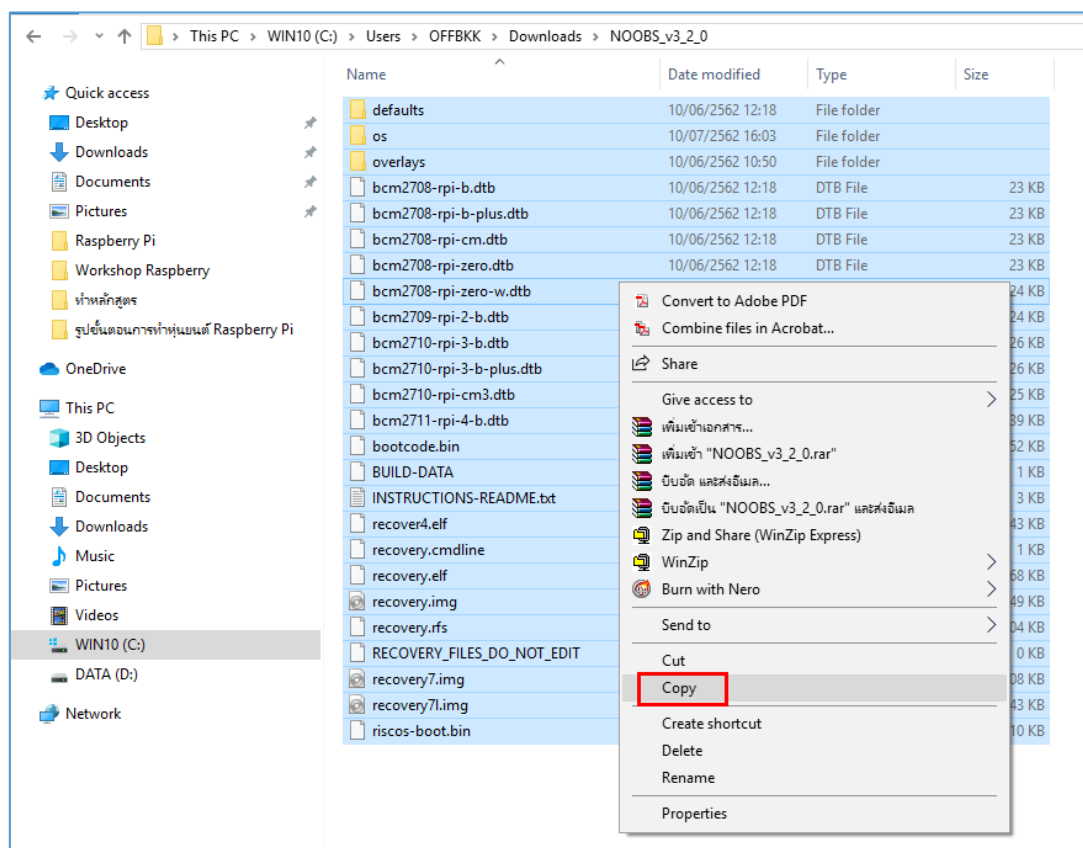
9. ทำการแตกไฟล์ ZIP โดยคลิกขวาเลือกคำสั่ง แยกไฟล์ไปที่.....

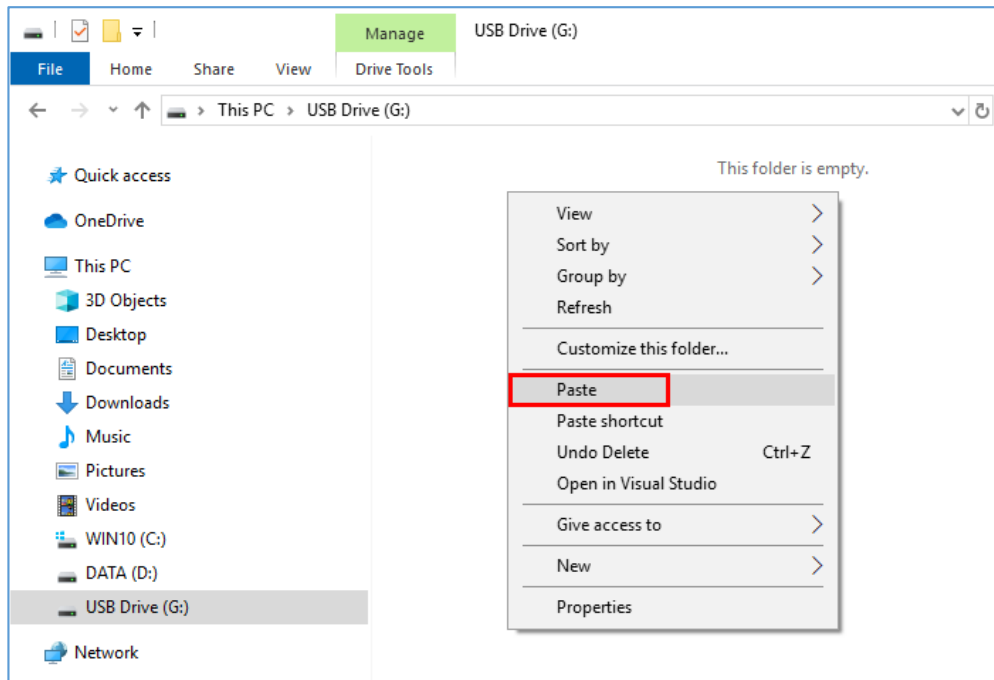


จะได้ไฟล์ต่างๆ ดังนี้



10. คัดลอกไฟล์ทั้งหมดไปวางใน SD Card





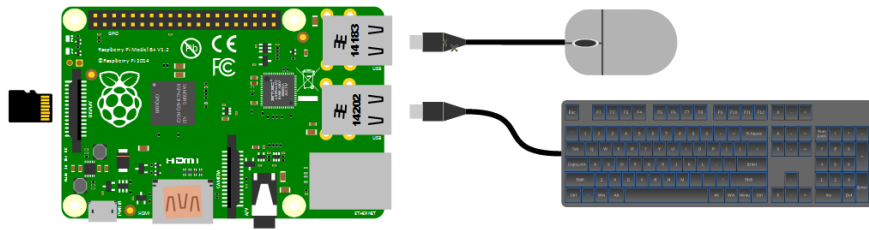
11. ถอด SD Card จากเครื่องคอมพิวเตอร์

การติดตั้งระบบปฏิบัติการ Raspbian

1. เสียบ SD Card เข้าในช่องเสียบ SD Card ที่อยู่ด้านหลังบอร์ด Raspberry Pi



2. เสียบเมาส์และคีย์บอร์ดเข้ากับพอร์ต USB ของบอร์ด Raspberry Pi



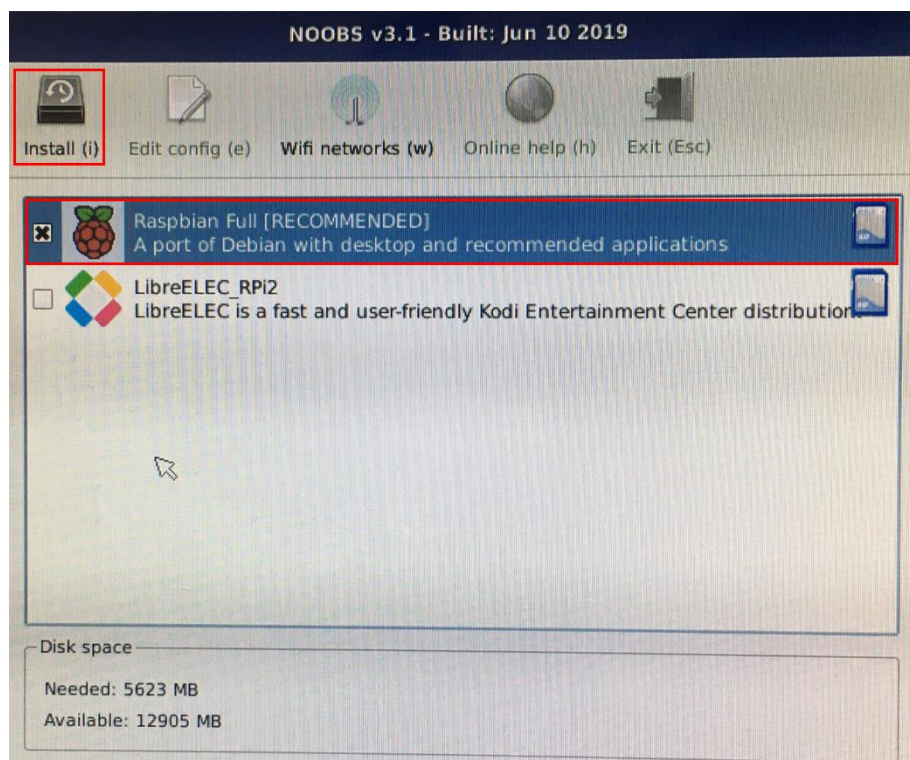
3. เสียบสาย HDMI เข้ากับพอร์ต HDMI ของบอร์ด Raspberry Pi ไปยังหน้าจอคอมพิวเตอร์ที่มีพอร์ต HDMI



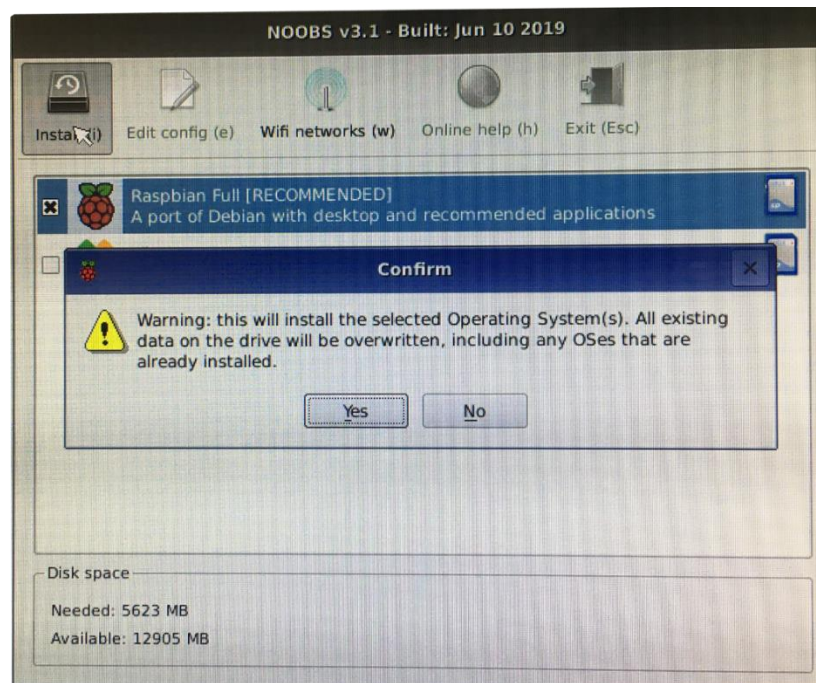
4. เสียบสาย Adapter เพื่อจ่ายไฟให้แกบอร์ด Raspberry Pi



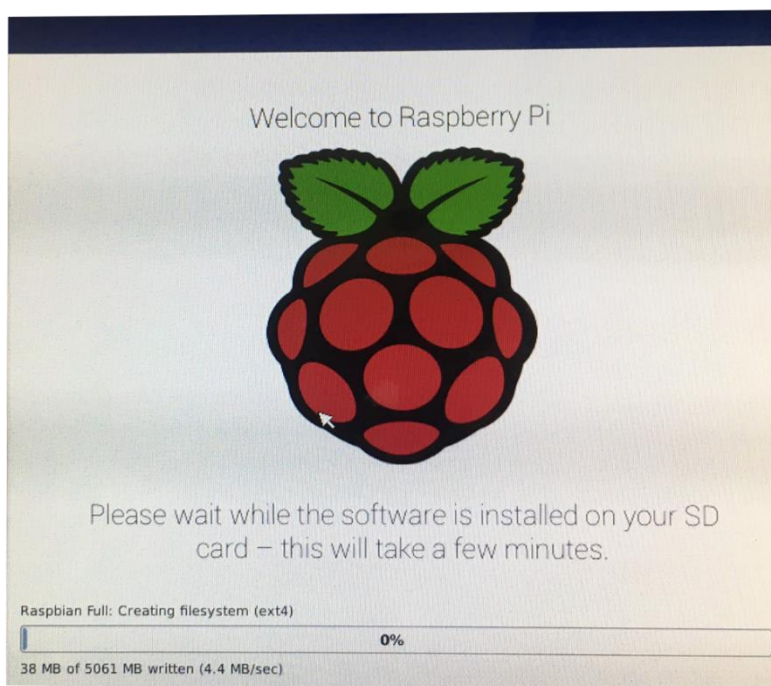
5. เมื่อมีกระแสไฟฟ้าเข้ามาเลี้ยงตัวบอร์ด บอร์ด Raspberry Pi จะถูกเปิดใช้งานมีลักษณะเช่นเดียวกับเครื่องคอมพิวเตอร์ หลังจากนั้นที่จอมอนิเตอร์จะปรากฏหน้าจอดังรูป ให้ทำการเริ่มต้นติดตั้งระบบปฏิบัติการ Raspbian โดยคลิกเลือก Raspbian Full แล้วคลิกคำสั่ง Install



6. ระบบจะถามว่า ข้อมูลจะถูกแทนที่ข้อมูลเดิมที่อยู่ใน SD Card ให้คลิกยืนยัน โดยคลิกปุ่ม Yes เพื่อยืนยันการติดตั้ง



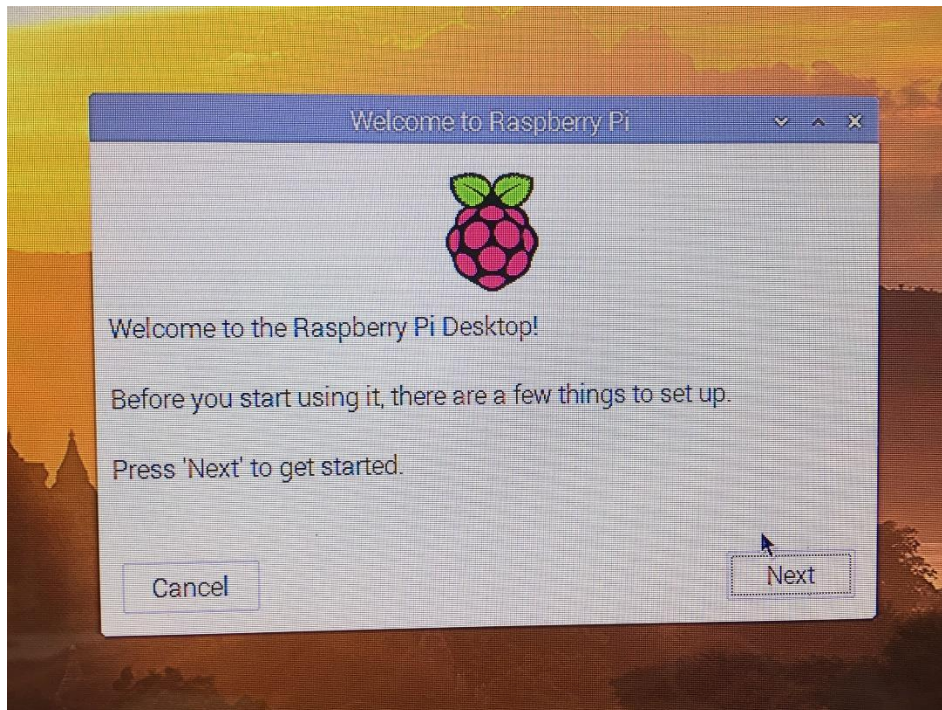
7. ในขั้นตอนการติดตั้งระบบปฏิบัติการ Raspbian จะใช้เวลาประมาณครึ่งชั่วโมง จึงจะติดตั้งครบ 100%



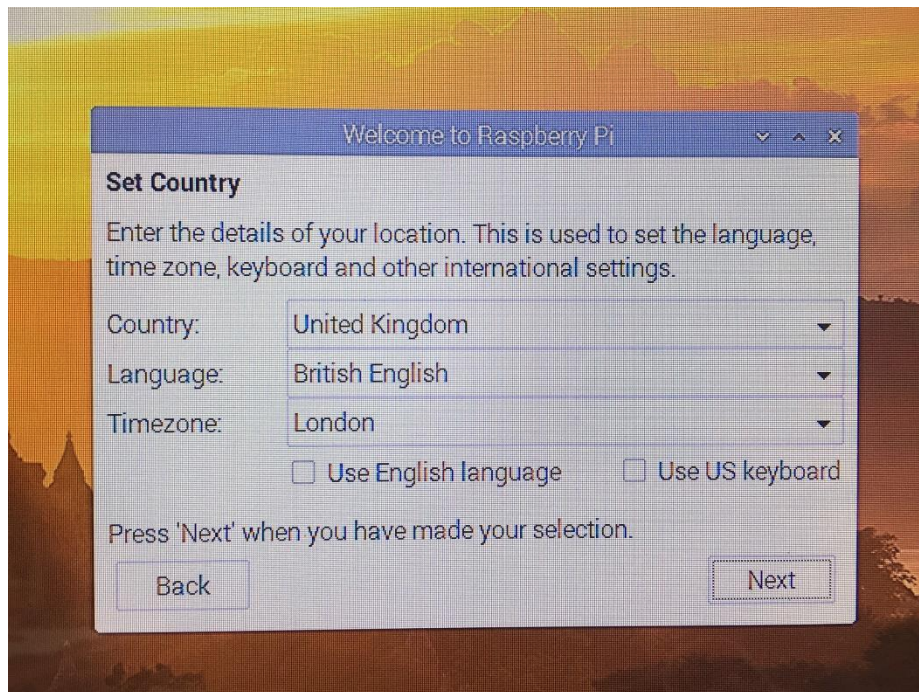
8. เมื่อติดตั้งครบ 100% แล้ว จะปรากฏหน้าต่างแจ้งเตือนขึ้นมาว่าการติดตั้งได้เสร็จสมบูรณ์แล้ว ให้คลิกปุ่ม OK บอร์ด Raspberry Pi จะทำการ Reboot 1 ครั้ง



9. เมื่อ raspberry เปิดขึ้นมาอีกครั้ง จะปรากฏหน้าต่าง Welcome to Raspberry Pi ให้คลิกปุ่ม Next



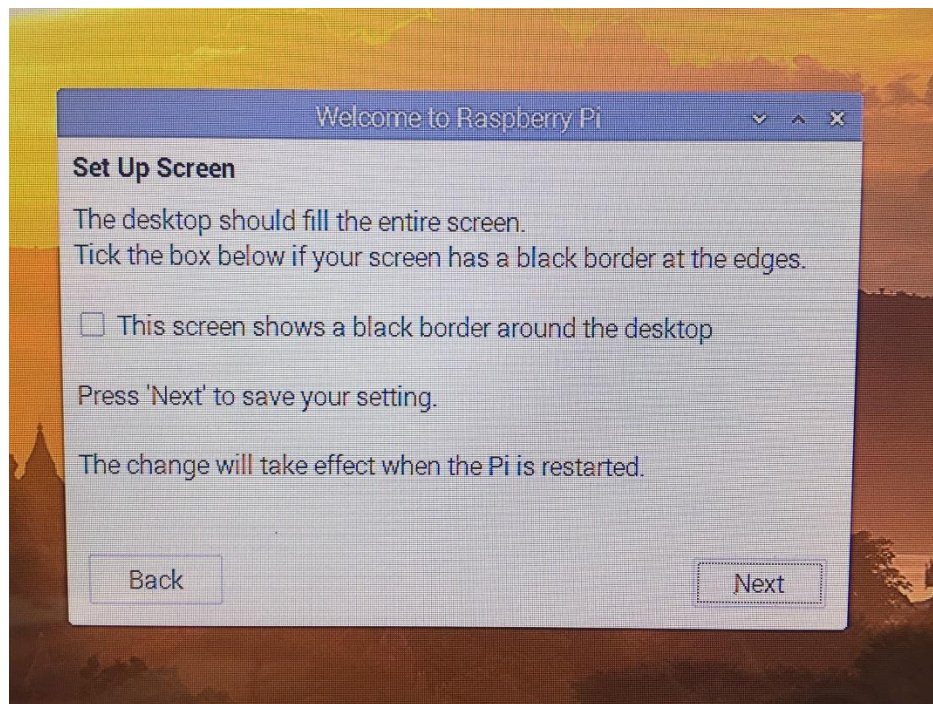
10. ระบบจะถามตำแหน่ง Location ของผู้ใช้งาน ในที่นี้ให้คลิก Next ไปเลย โดยไม่ต้องเลือกข้อมูลใดๆ ทั้งสิ้น (ในส่วนนี้สามารถตั้งค่าได้ที่หลัง)



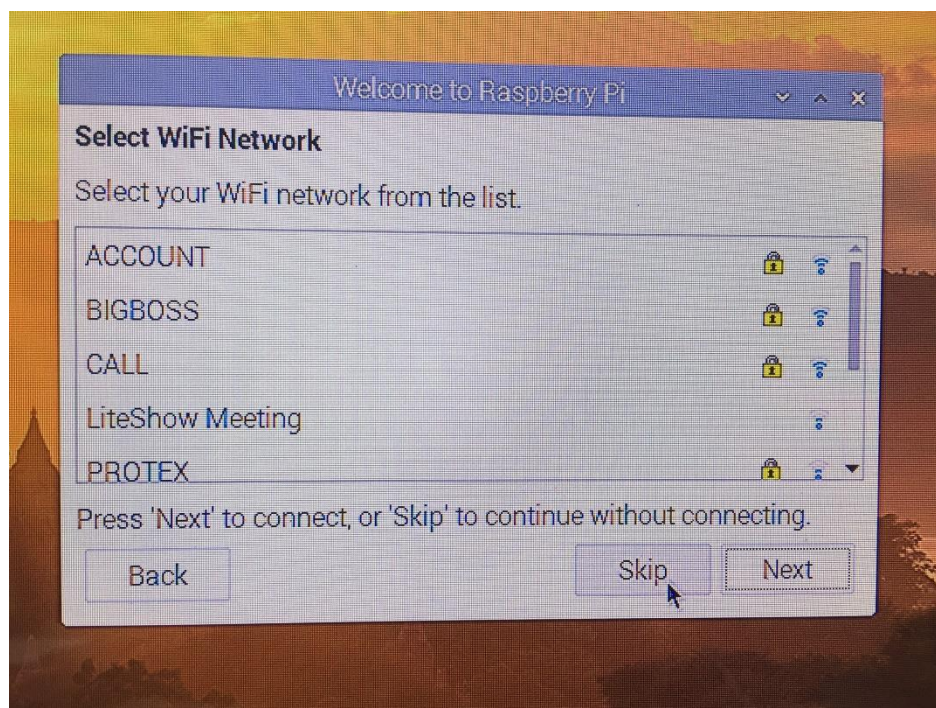
11. ระบบจะให้ทำการเปลี่ยน Password สำหรับผู้ใช้ในการเข้าใช้งานบอร์ด Raspberry Pi โดยจะมีค่าเริ่มต้น คือ user : pi และ password : raspberry ซึ่งเราสามารถที่จะทำการเปลี่ยน password ได้ โดยกรอกรหัสผ่านใหม่เข้าไปทั้งในช่อง Enter new password และยืนยันรหัสผ่านใหม่อีกครั้งในช่อง Confirm new password แล้วคลิกปุ่ม Next



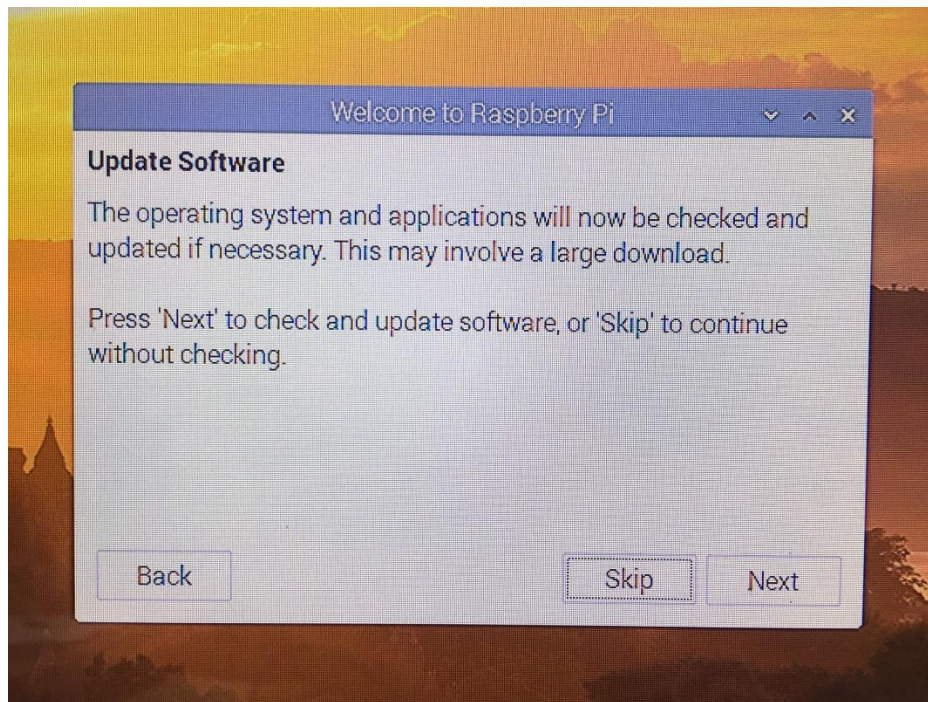
12. ต่อมาระบบจะถามว่า ต้องการให้มีเส้นกรอบสีดำล้อมรอบหน้าจอ Desktop หรือไม่ หากไม่ต้องการให้คลิกปุ่ม Next โดยไม่ต้องคลิกเลือกอะไรเลย



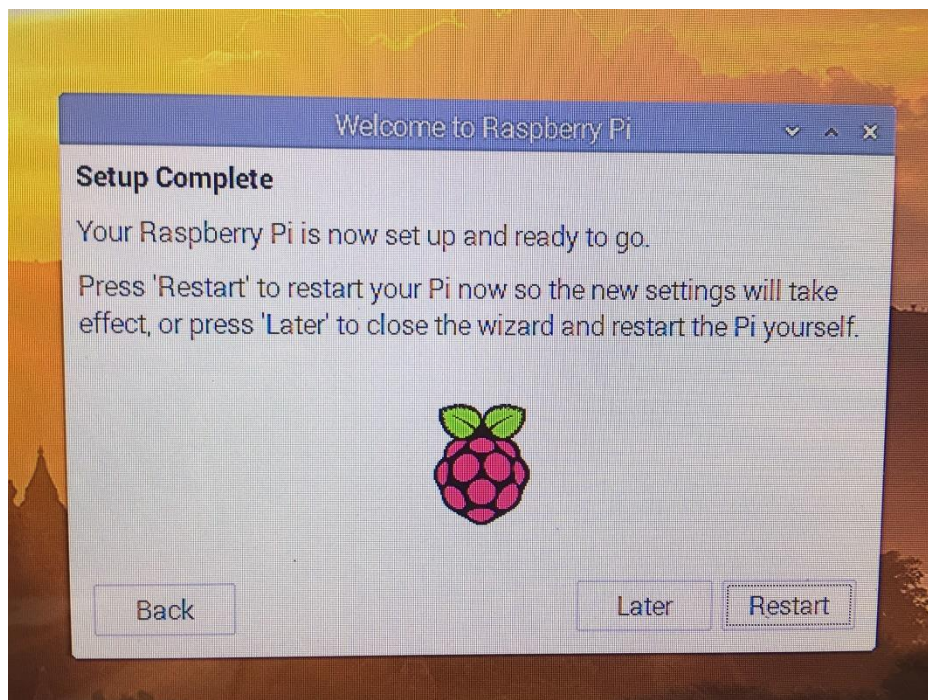
13. ในขั้นตอนการเลือก Wi-Fi ในส่วนนี้ให้คลิกปุ่ม Skip เพื่อข้ามขั้นตอนนี้ไปก่อน เนื่องจากสามารถเลือกได้ทีหลัง



14. ในขั้นตอนการอัปเดต Software ให้คลิกปุ่ม Skip เพื่อข้ามขั้นตอนนี้ไปก่อนเช่นกัน



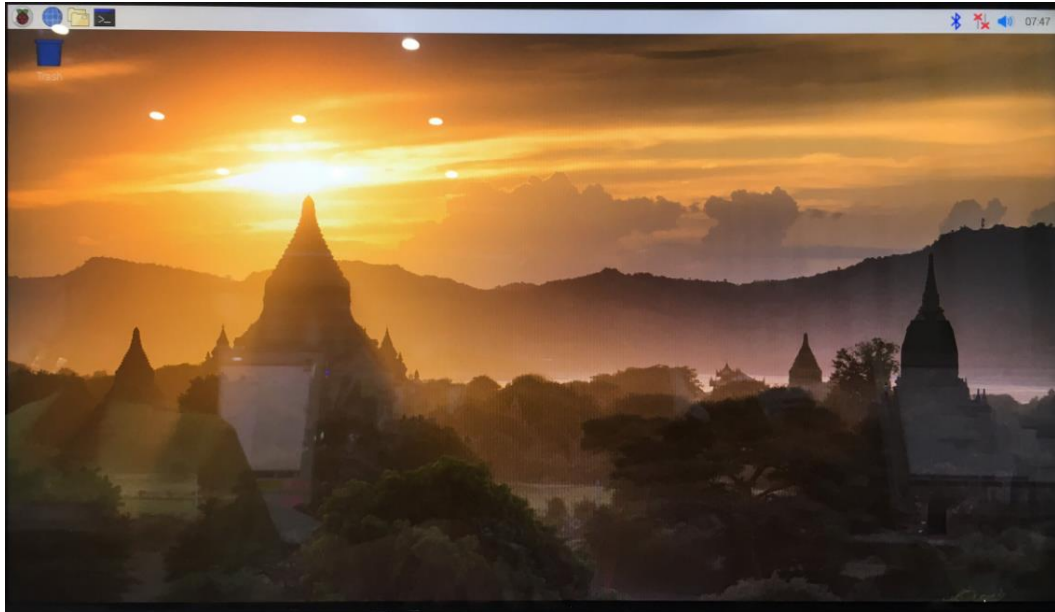
15. เมื่อมาสู่ขั้นตอนนี้ ให้คลิกปุ่ม Restart เพื่อ Reboot บอร์ด Raspberry Pi



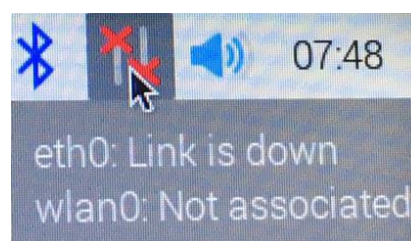
การตั้งค่าระบบ

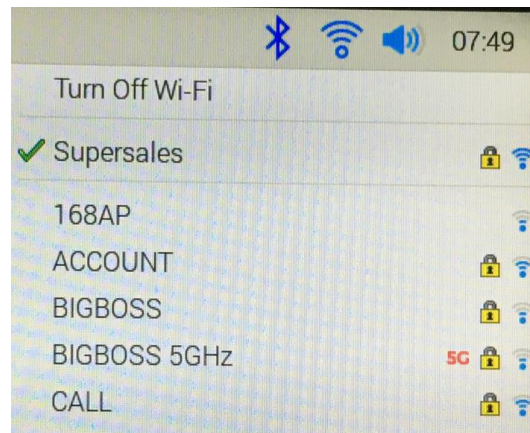
1. การตั้งค่า Wi-Fi

1.1 เมื่อ Raspberry Pi เปิดขึ้นมาอีกครั้ง จะปรากฏหน้าจอ Desktop ดังรูป

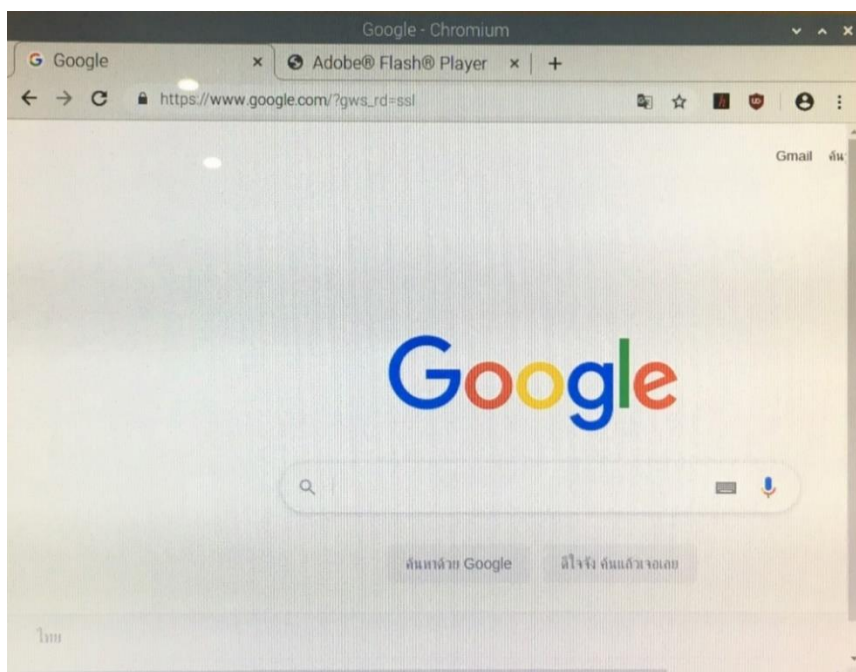
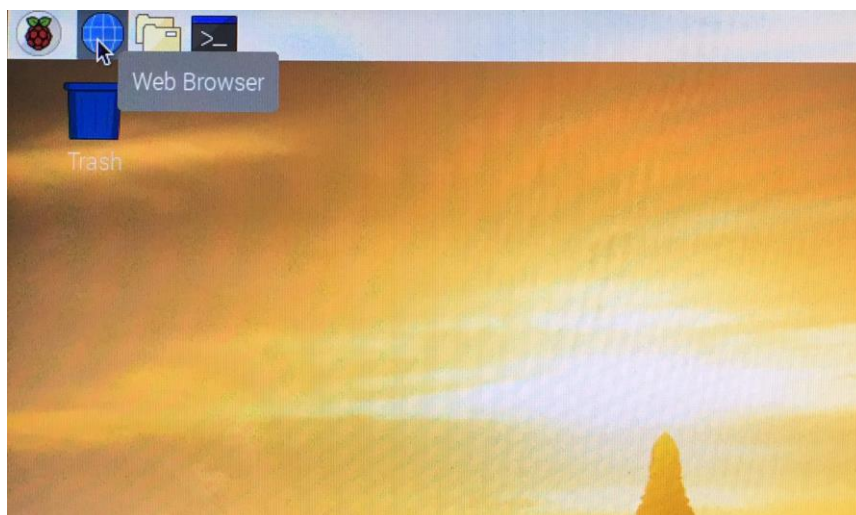


1.2 เชื่อมต่อ Wi-Fi โดยคลิกเลือกไอคอน Wi-Fi ที่แถบด้านบนขวา แล้วเลือก Wi-Fi ที่ต้องการ หาก Wi-Fi มีรหัสผ่าน จะต้องกรอกรหัสผ่านให้ถูกต้อง ระบบจะทำการเชื่อมต่อ Wi-Fi เช่นเดียวกับคอมพิวเตอร์ทั่วไป



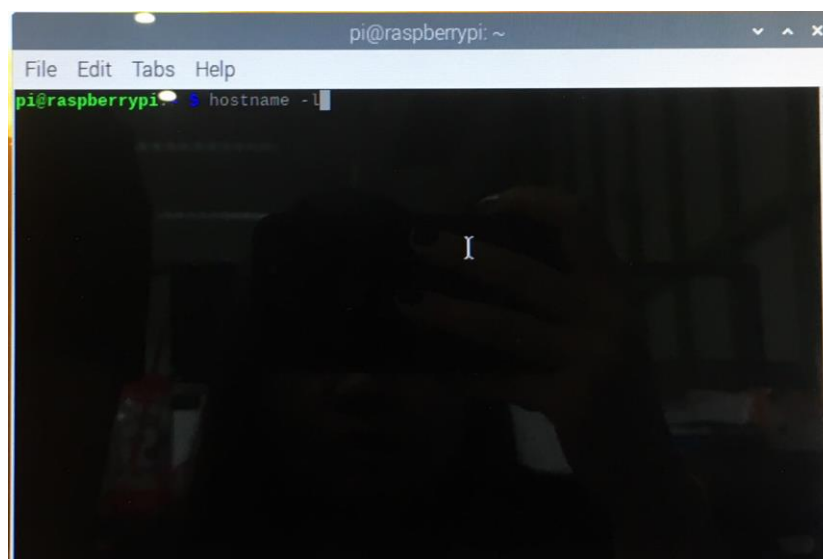
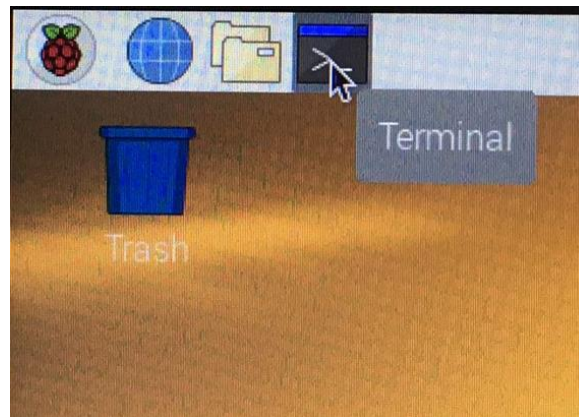


1.3 ทดลองการเชื่อมต่อ Wi-Fi โดยเปิด Web Browser และทดลองเข้าเว็บไซต์

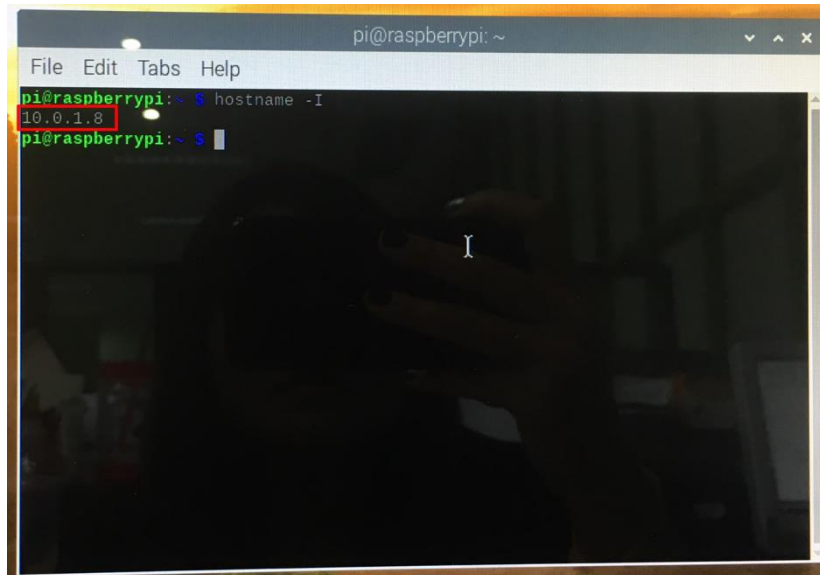


2. ตรวจสอบ IP Address

2.1 ตรวจสอบเลข IP Address ของ Raspberry Pi โดยคลิกเปิดโปรแกรม Terminal แล้วพิมพ์คำสั่ง `hostname -I` แล้วกด Enter

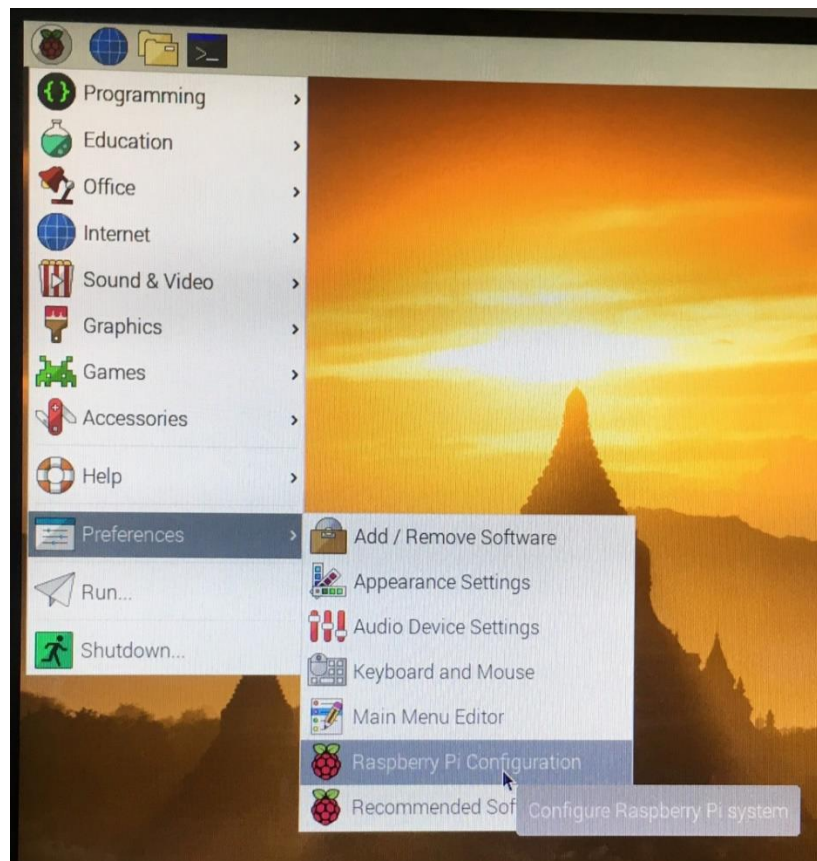


2.2 จะได้หมายเลข IP Address ของ Raspberry Pi ดังรูป เช่น 10.0.1.8 เป็นตัวเลขที่ใช้สำหรับ Remote จากเครื่องคอมพิวเตอร์มายังบอร์ด Raspberry Pi

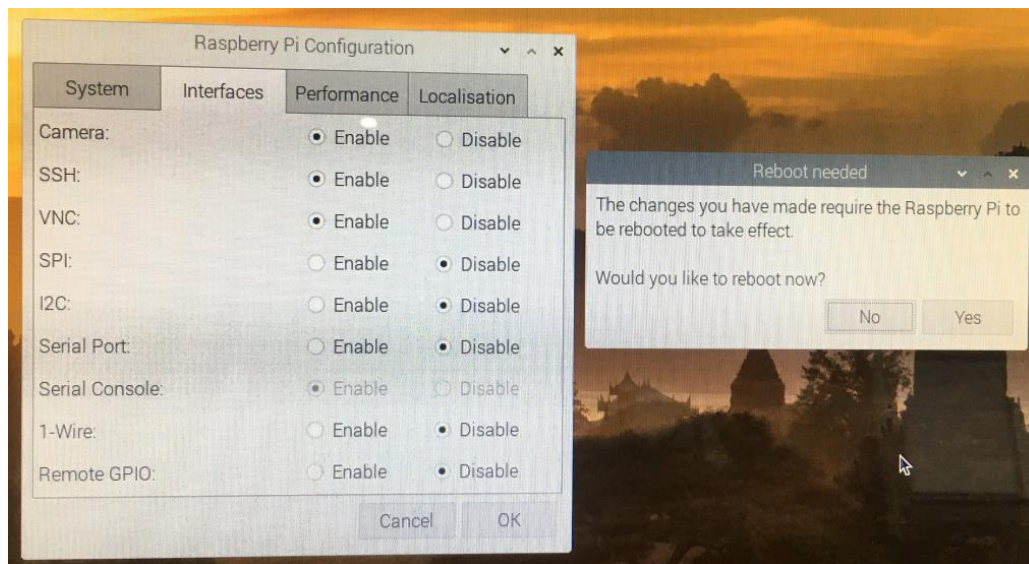


3. ตั้งค่า Config ให้เปิด (Enable) การใช้งานระบบ

3.1 คลิกเมนูรูป Pi > Preferences > Raspberry Pi Configuration



3.2 คลิกแท็บ Interfaces แล้วคลิกเลือก Enable เพื่อเปิดใช้งาน หรือคลิก Disable เพื่อปิดรายการที่ไม่ต้องการเปิดใช้งาน จากนั้นกด Yes ตามรูป



- Camera สำหรับการใช้งานกล้อง หากยังไม่มีใช้งาน ยังไม่ต้องเปิดไว้ก็ได้
- SSH สำหรับเชื่อมต่อควบคุม Rasp Pi แบบ Remote ผ่านโปรแกรม SSH ใน Text Mode
- VNC สำหรับเชื่อมต่อควบคุม Rasp Pi แบบ Remote ผ่านโปรแกรม VNC ในโหมดหน้าจอกราฟิก กล่าวคือจะเหมือนกับยกหน้าจอมาไว้ให้เห็นเลย
- หากต้องการเปิดส่วนอื่นๆ ก็สามารถเลือกเปิดไว้ตอนนี้ได้หรือจะกลับมาเปิดทีหลังก็ได้

4. การตั้งค่าภาษา

4.1 การตั้งค่า Location ในส่วนนี้จะมิตผลต่อหน้าเมนูของ Raspberry Pi

- เมนูเป็นภาษาไทย ให้คลิกแท็บ Localisation > Set Locale... แล้วเลือก

Language: th (Thai)

Country: TH (Thailand)

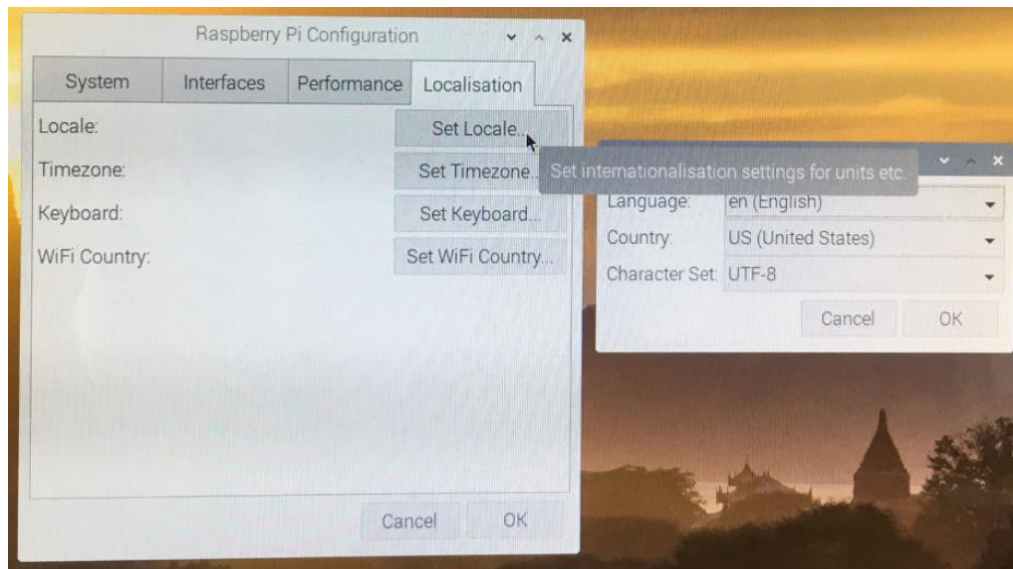
Character Set: UTF-8

- เมนูเป็นภาษาอังกฤษ ให้คลิกแท็บ Localisation > Set Locale... แล้วเลือก

Language: en (English)

Country: US (United States)

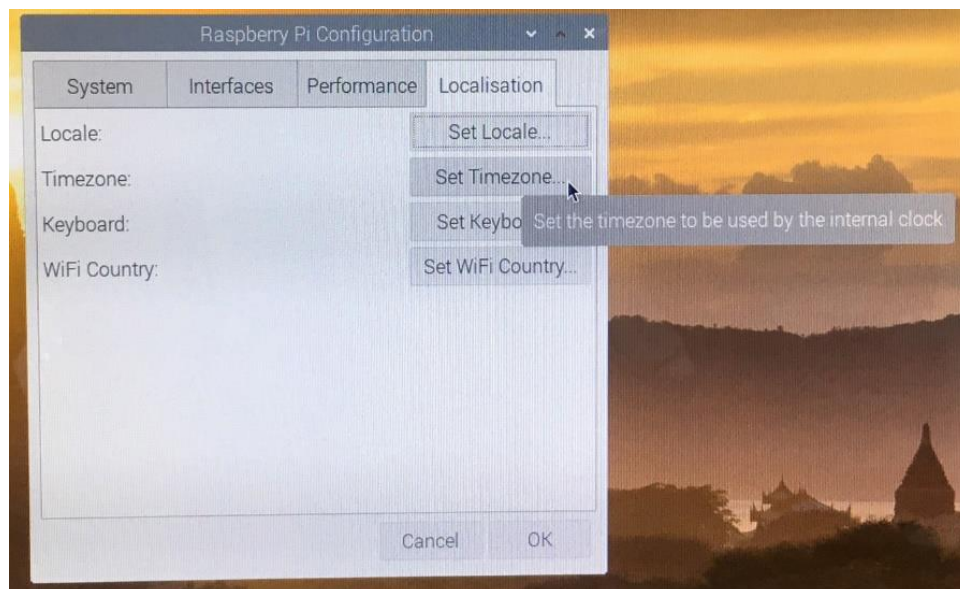
Character Set: UTF-8

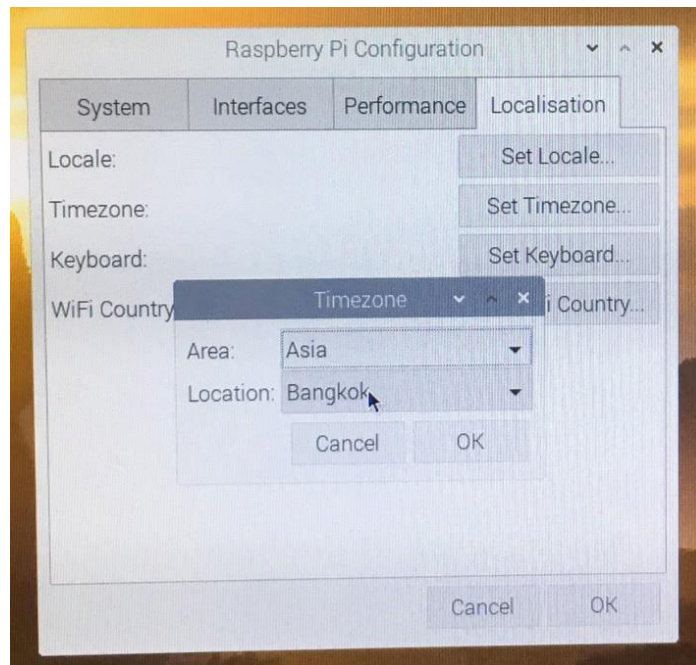


4.2 การตั้งค่าเวลา ที่แท็บ Localisation > Set Timezone... แล้วเลือก

Area: Asia

Location: Bangkok



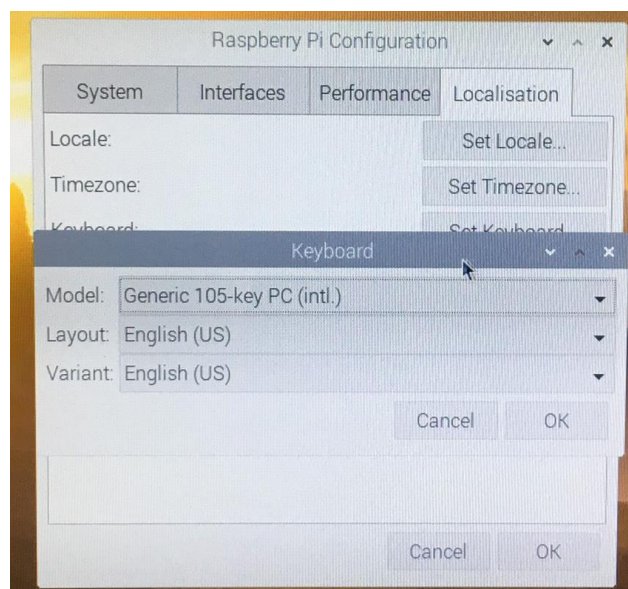


4.3 การตั้งค่าเป็นพิมพ์ คคลิก Set Keyboard.. ให้เลือก

Model: Generic 105-key PC (intl)

Layout: English (US)

Variant : English (US)



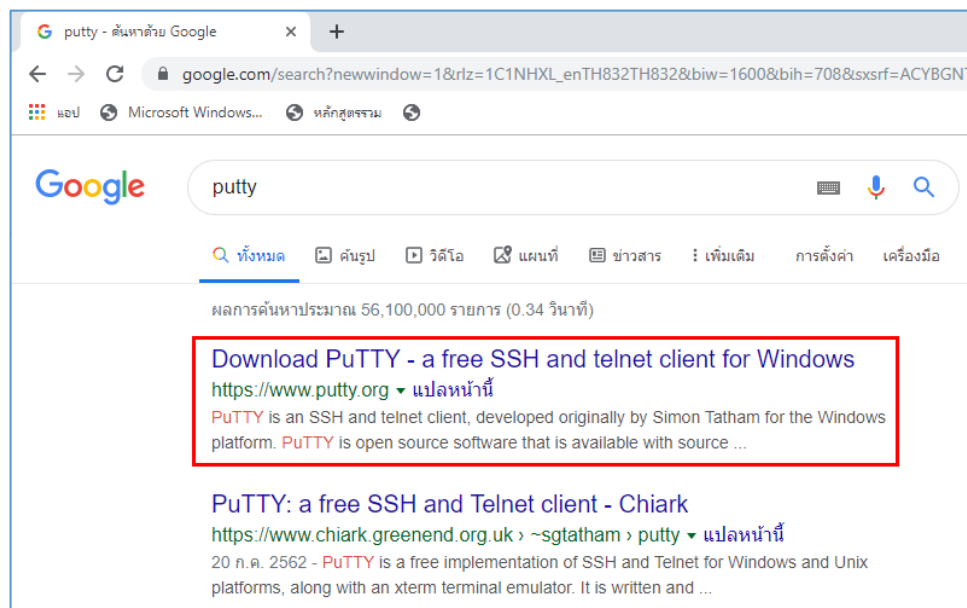
การรีโมทจากเครื่องคอมพิวเตอร์มายังบอร์ด Raspberry Pi

การใช้งาน Raspberry Pi ที่นิยมส่วนใหญ่ จะต่อจอภาพ เป็นพิมพ์ และเมาส์ เพื่อเซทอัพครั้งแรกครั้งเดียว จากนั้นก็ปลดออกหมด แล้วใช้วิธีที่เรียกว่า Headless กล่าวคือที่ Raspberry Pi จะไม่มีการต่อจอภาพ เป็นพิมพ์หรือเมาส์เอาไว้แต่อย่างใด จะใช้คอมพิวเตอร์เครื่องอื่นหรืออุปกรณ์สมาร์ตโฟนเชื่อมต่อ Remote เข้าไปควบคุม Raspberry Pi โดยการควบคุมรีโมทไปยังบอร์ด Raspberry Pi มี 2 แบบคือ

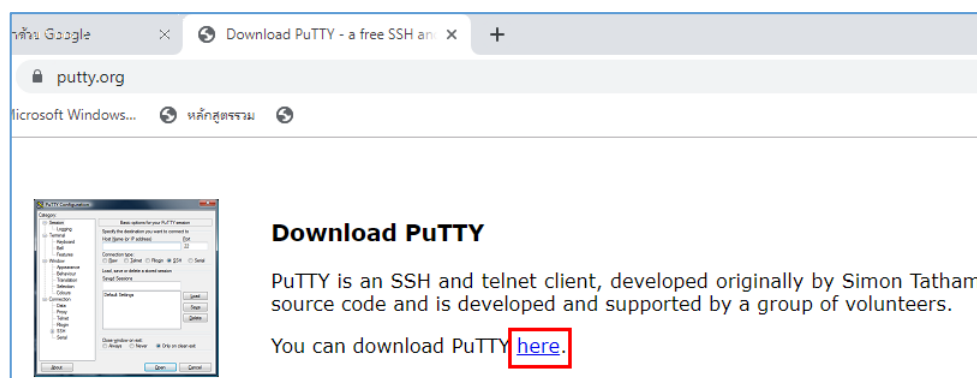
1. การรีโมทแบบ SSH Command line สำหรับพิมพ์คำสั่งควบคุม

สำหรับพิมพ์คำสั่ง Terminal ของ Raspberry Pi ผ่านทางคอมพิวเตอร์หรือสมาร์ตโฟนที่ใช้เป็น Remote ซึ่งทำงานในแบบ Text Mode โดยมีขั้นตอนดังนี้

1.1 ติดตั้งโปรแกรม Putty ที่เครื่องคอมพิวเตอร์ก่อน โดยดาวน์โหลดจากอินเทอร์เน็ต

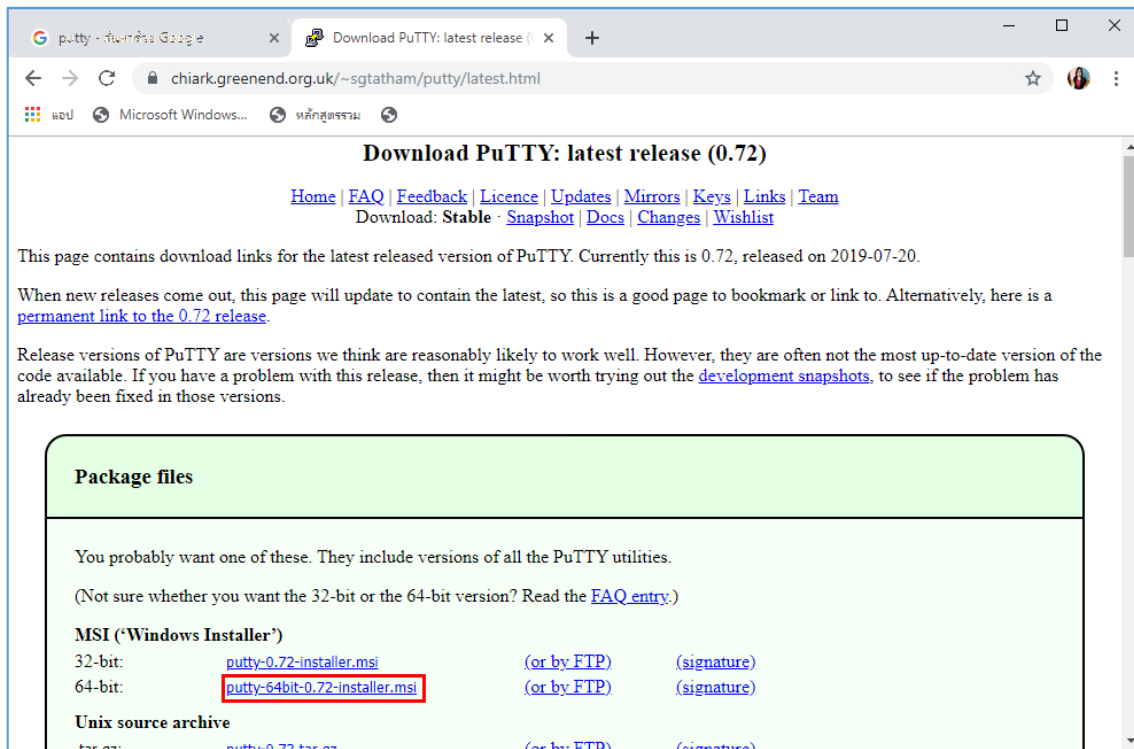


1.2 คลิกเข้าเว็บไซต์ putty.org และคลิกที่ here ตรงส่วนของ Download PuTTY



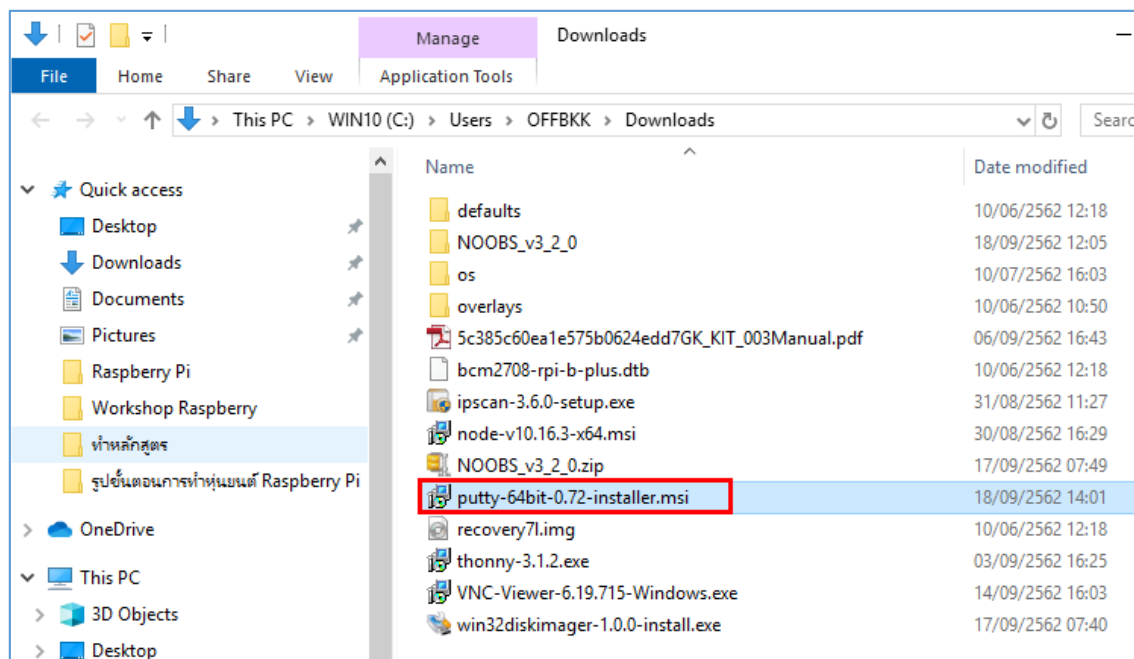
1.3 คลิกดาวน์โหลดโปรแกรมสำหรับติดตั้ง โดยเช็คว่าเครื่องคอมพิวเตอร์ที่ใช้อยู่เป็น 32 หรือ 64

บิต

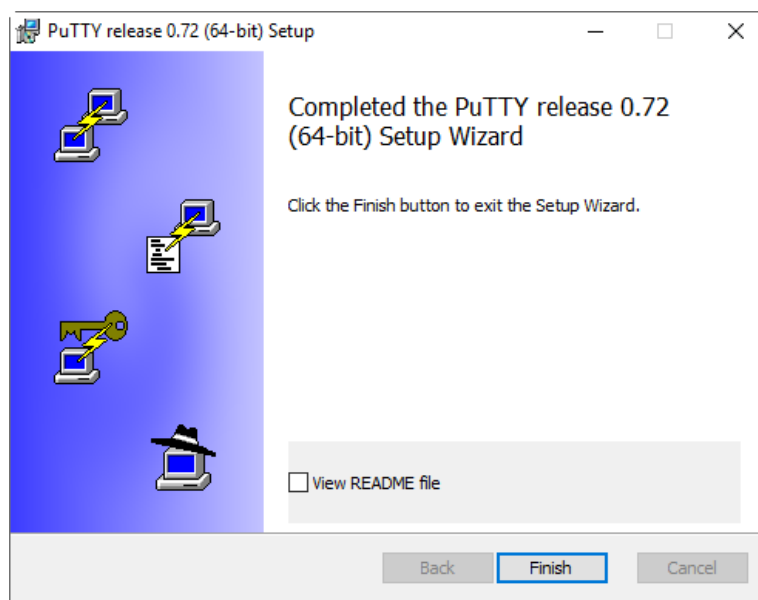


1.4 ไฟล์ที่ดาวน์โหลดเสร็จแล้วจะถูกบันทึกไว้ในโฟลเดอร์ Downloads ให้ดับเบิลคลิกเพื่อติดตั้ง

โปรแกรม putty

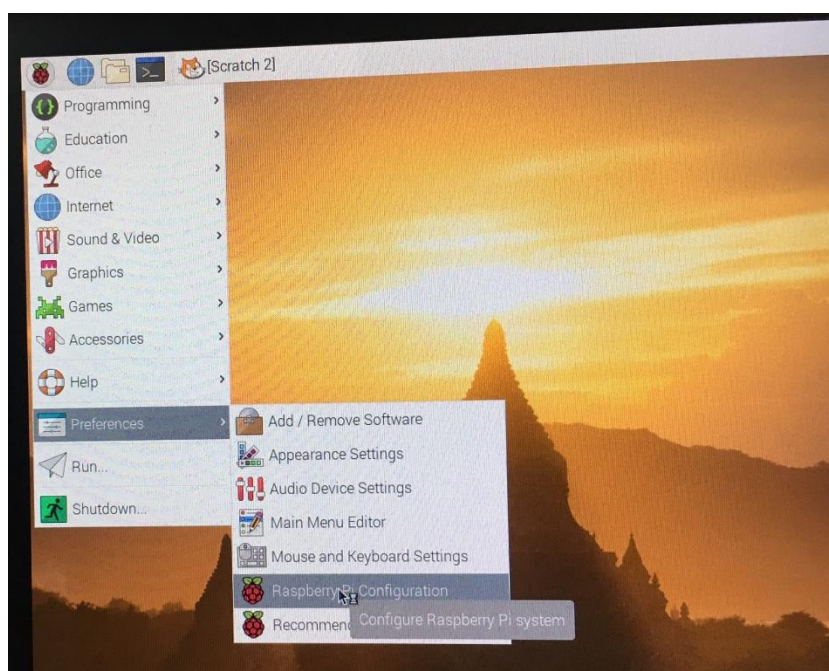


1.5 ทำการคลิกปุ่ม Next > Next > Install เสร็จแล้วคลิกปุ่ม Finish

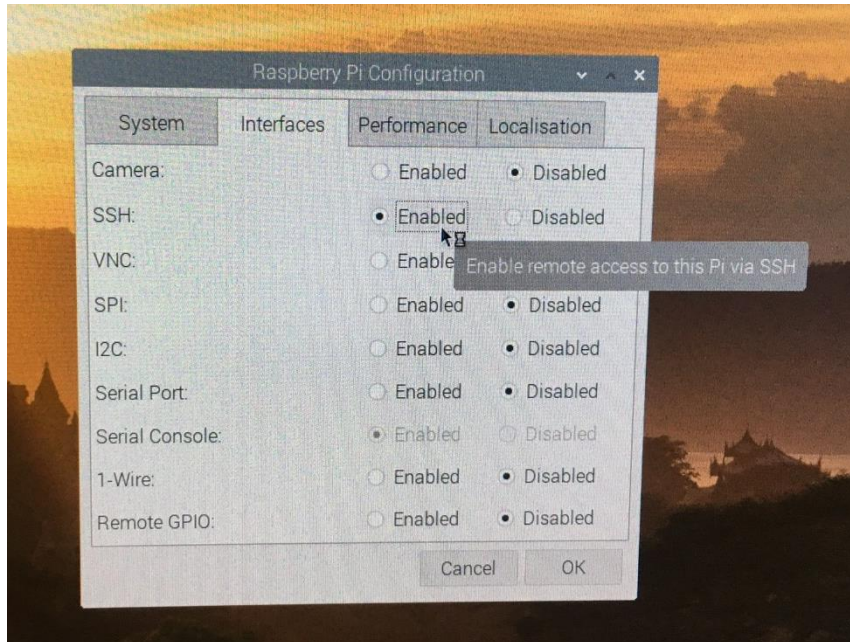


1.6 ตรวจสอบ IP Address ของคอมพิวเตอร์กับบอร์ด Raspberry Pi จะต้องอยู่ในวง LAN เดียวกัน หรือจับสัญญาณ Wi-Fi ตัวเดียวกัน

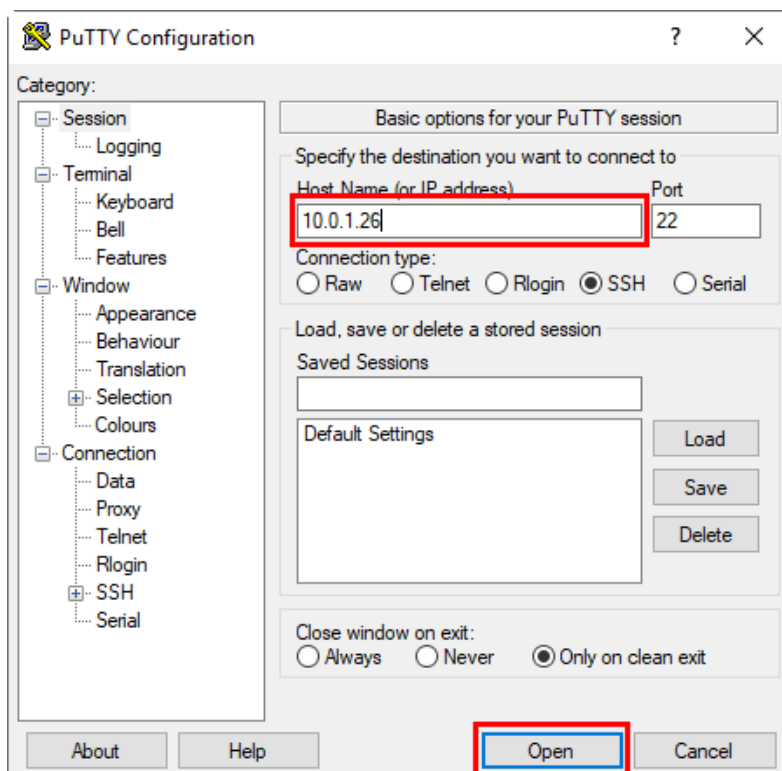
1.7 บอร์ด Raspberry Pi จะต้องเปิดการใช้งาน SSH ก่อน โดยไปที่ Pi > Preferences > RaspberryPi Configuration



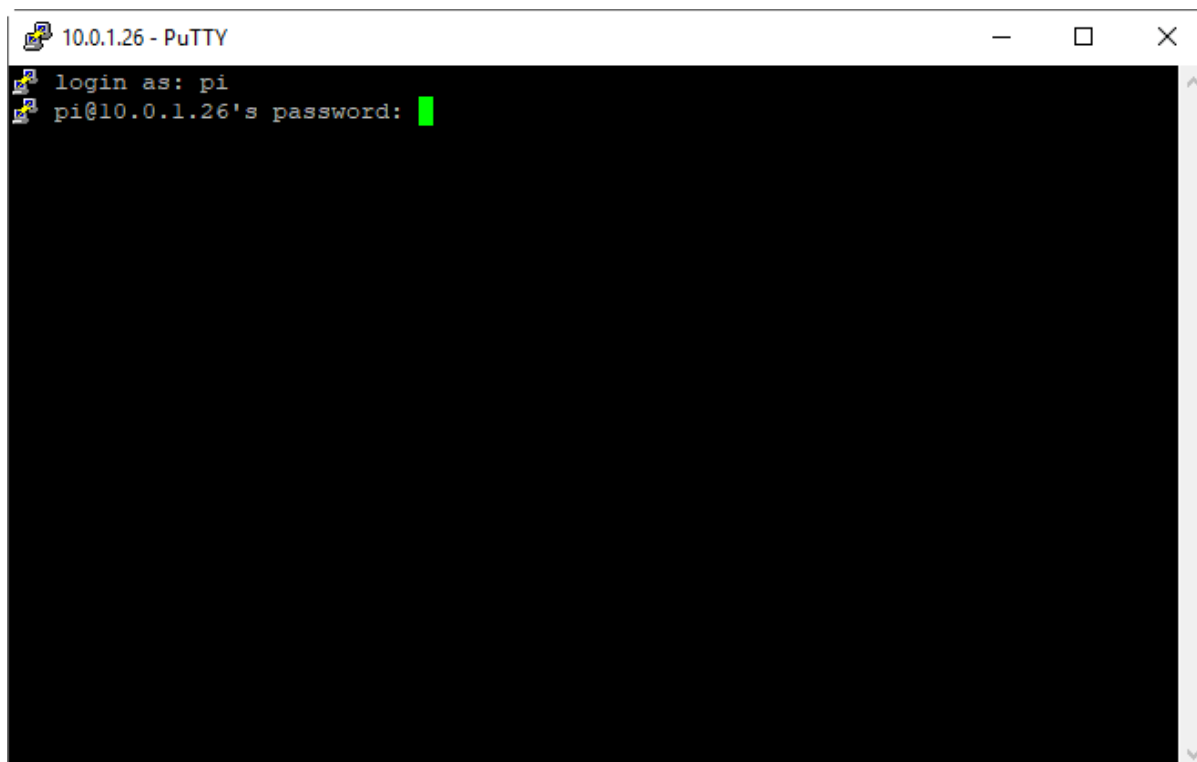
ไปที่แท็บ Interfaces เลือก SSH ให้เป็น Enabled แล้วคลิกปุ่ม OK



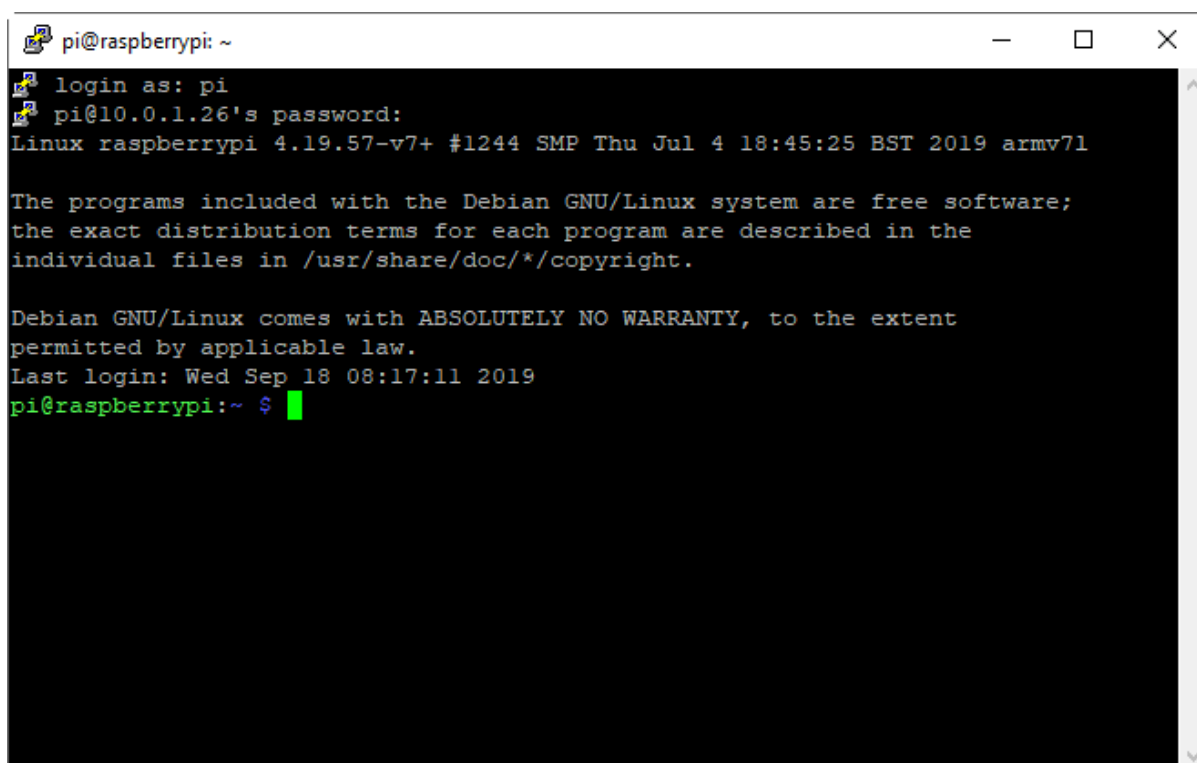
1.8 เปิดโปรแกรม Putty ขึ้นมา หลังจากนั้นให้กรอกหมายเลข IP Address ของบอร์ด Raspberry Pi ในช่อง Host Name (or IP Address) แล้วคลิกปุ่ม Open



1.9 จะเข้าสู่หน้าจอ Text Editor ให้ได้ User และ Password ของ Raspberry Pi



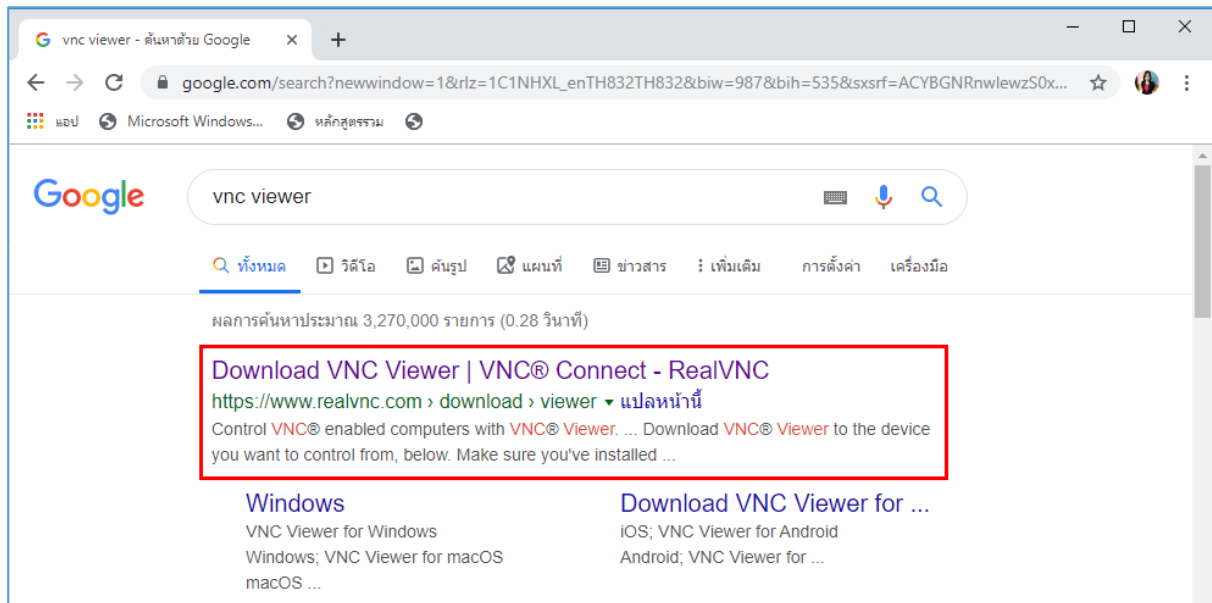
1.10 เมื่อรีโมทเข้าไปยังบอร์ด Raspberry Pi ได้แล้ว จะปรากฏคำสั่ง pi@raspberrypi: เพื่อรอรับคำสั่งในการควบคุมบอร์ด Raspberry Pi



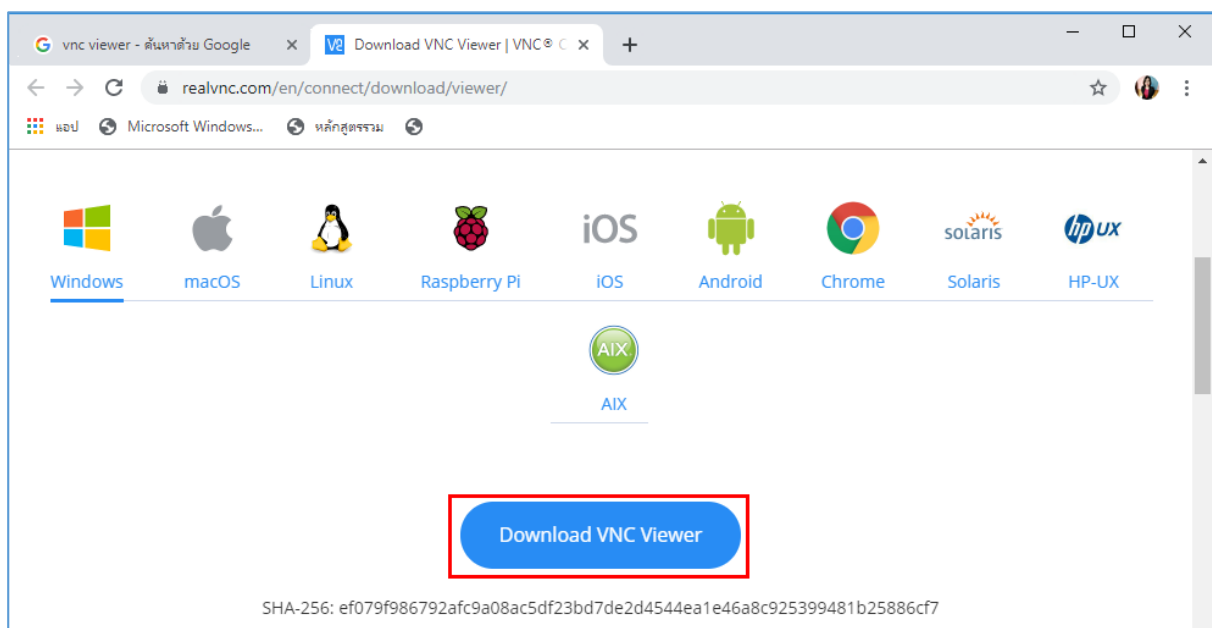
2. การรีโมทแบบกราฟิก VNC เห็นภาพจอ Raspberry Pi บนเครื่องที่ Remote

สำหรับให้คอมพิวเตอร์หรือสมาร์ทโฟนสามารถที่จะ Remote ไปยังบอร์ด Raspberry Pi ซึ่งทำงานในแบบหน้าจอ Desktop โดยมีขั้นตอนดังนี้

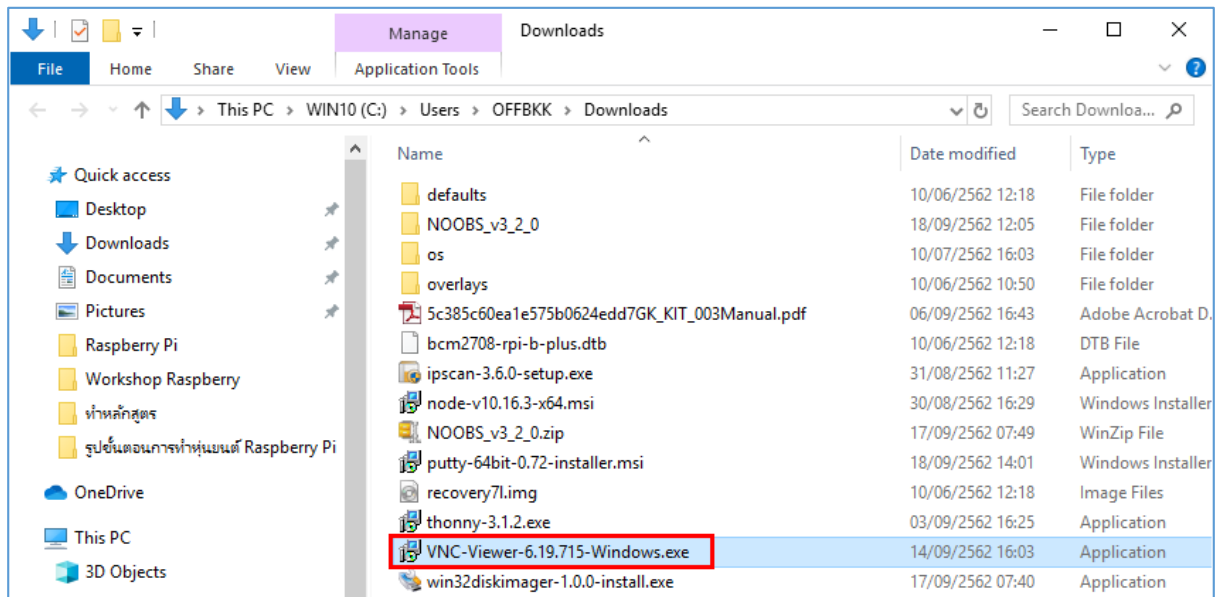
1.1 ติดตั้งโปรแกรม VNC Viewer ที่เครื่องคอมพิวเตอร์ก่อน โดยดาวน์โหลดจากอินเทอร์เน็ต



1.2 คลิกเข้าเว็บไซต์ realvnc.com และคลิกที่ Download VNC Viewer

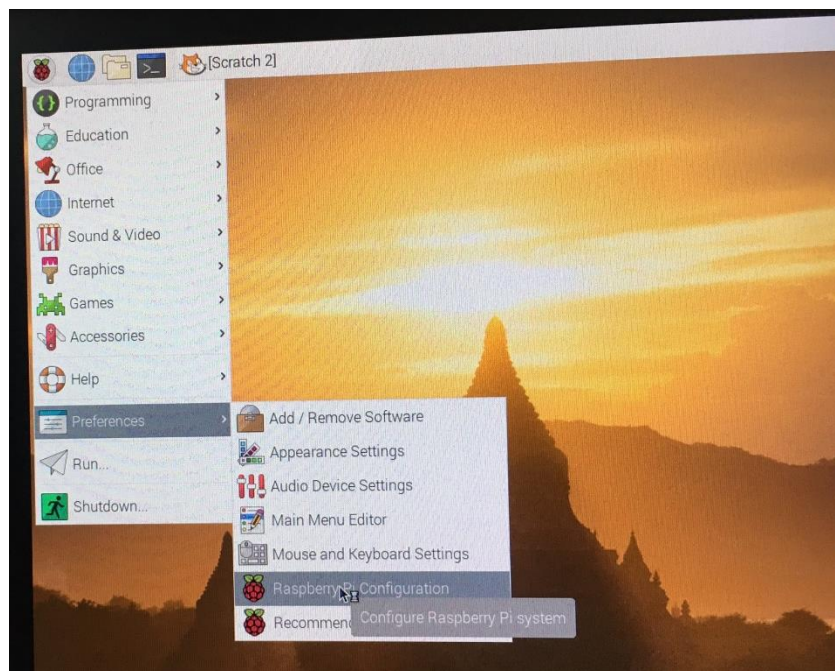


1.3 ไฟล์ที่ดาวน์โหลดเสร็จแล้วจะถูกบันทึกไว้ในโฟลเดอร์ Downloads ให้ดับเบิลคลิกเพื่อติดตั้งโปรแกรม VNC Viewer

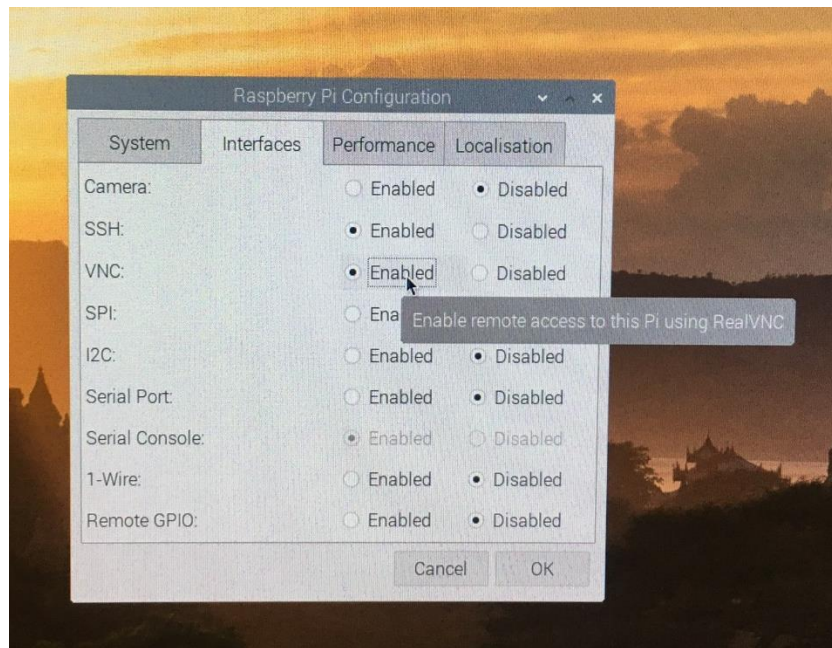


1.4 ตรวจสอบ IP Address ของคอมพิวเตอร์กับบอร์ด Raspberry Pi จะต้องอยู่ในวง LAN เดียวกัน หรือจับสัญญาณ Wi-Fi ตัวเดียวกัน

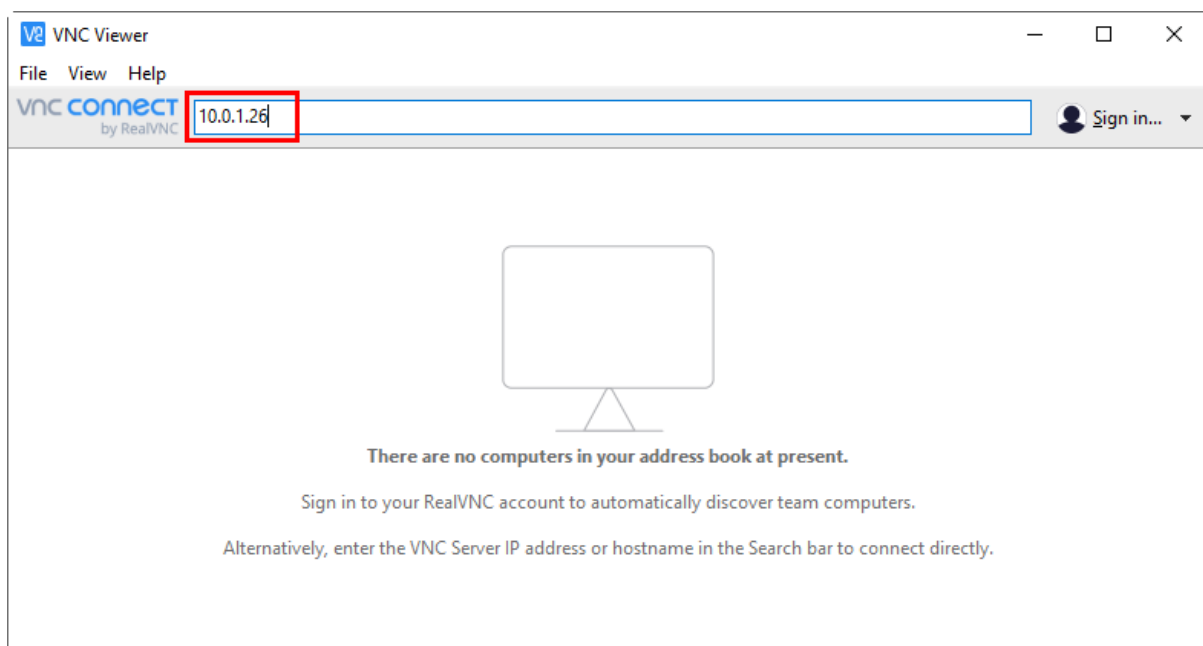
1.5 บอร์ด Raspberry Pi จะต้องเปิดการใช้งาน VNC ก่อน โดยไปที่ Pi > Preferences > RaspberryPi Configuration



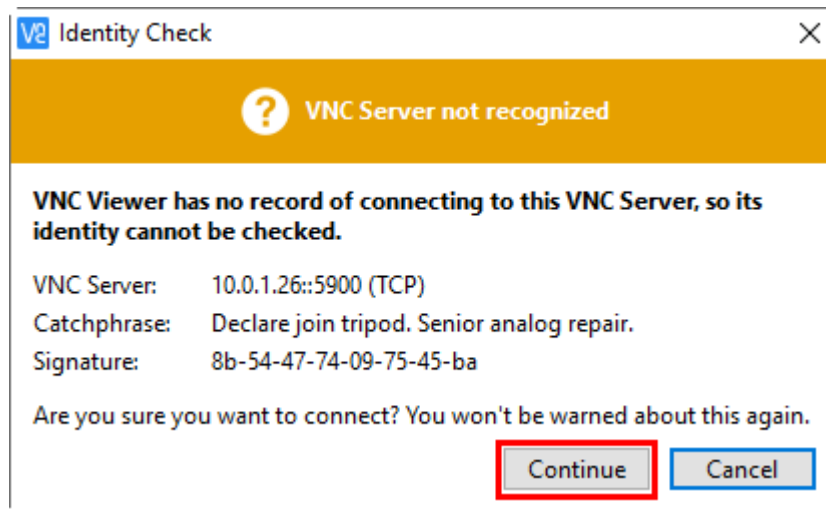
ไปที่แท็บ Interfaces เลือก VNC ให้เป็น Enabled แล้วคลิกปุ่ม OK



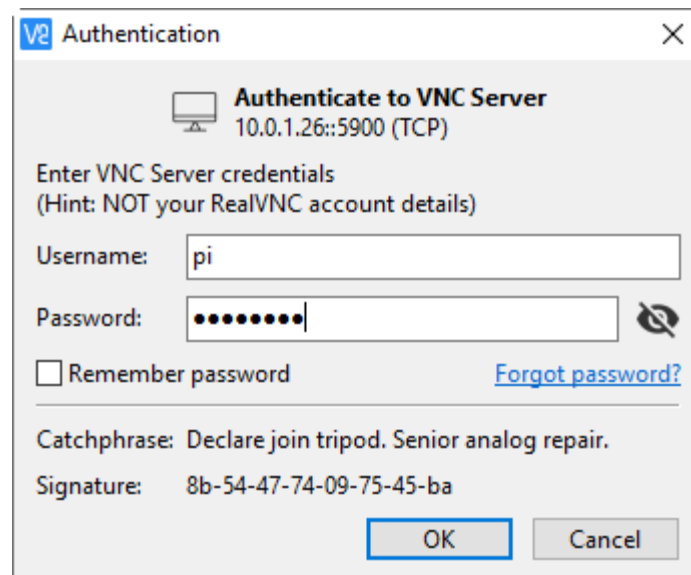
1.6 เปิดโปรแกรม VNC Viewer ขึ้นมา หลังจากนั้นให้กรอกหมายเลข IP Address ของบอร์ด Raspberry Pi แล้ว Enter



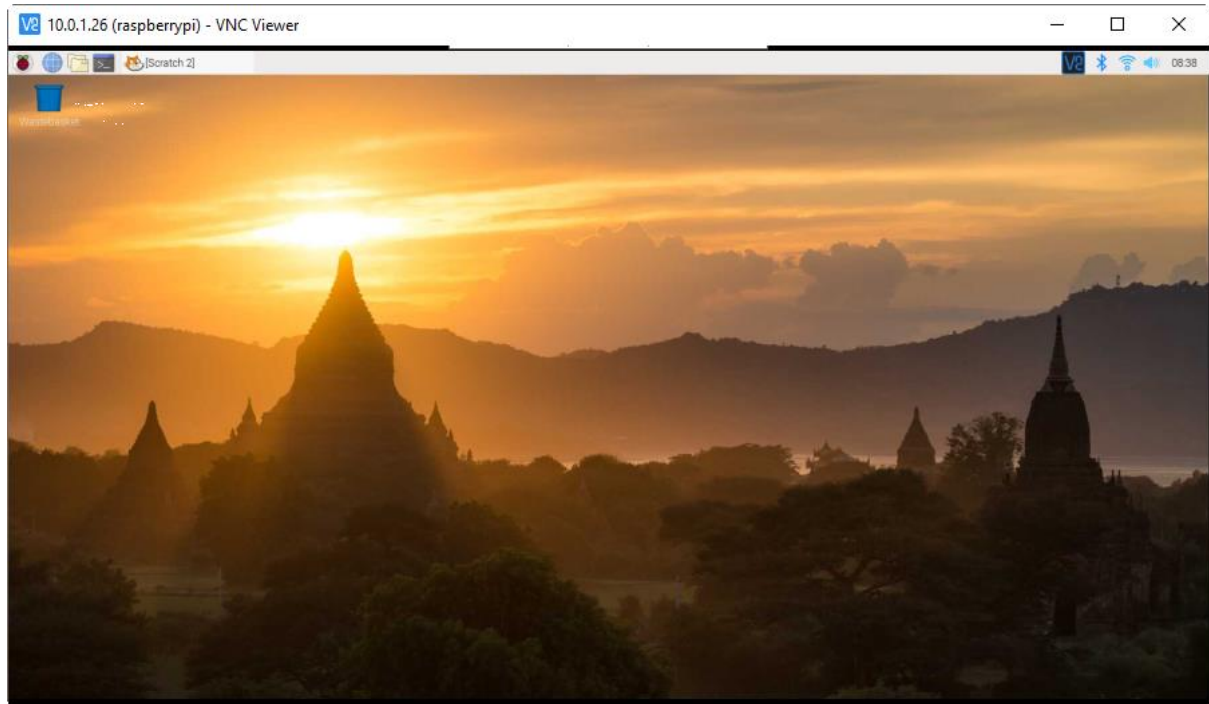
1.7 จะมีหน้าต่างปรากฏขึ้นมาเพื่อยืนยันการรีโมทเข้าใช้งาน ให้คลิกปุ่ม Continue



1.8 ใส่ User และ Password สำหรับเข้าใช้งาน Raspberry Pi แล้วคลิกปุ่ม OK

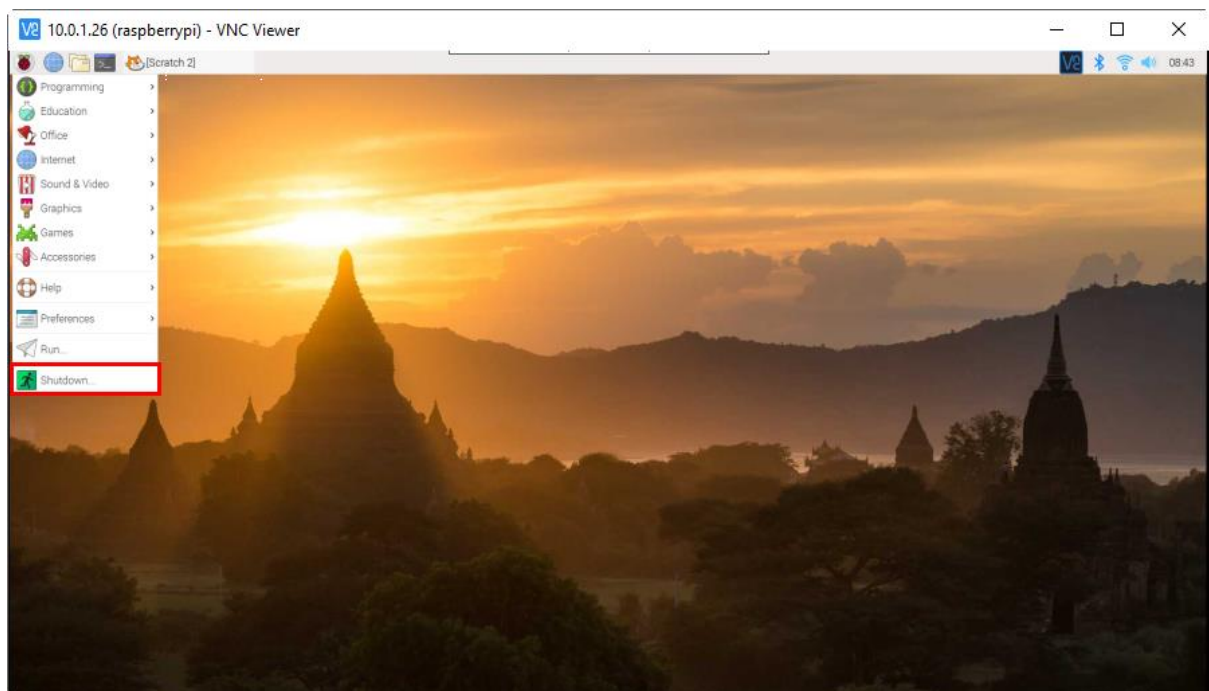


1.9 เมื่อล็อกอินเข้าใช้งานสำเร็จ จะเข้าสู่หน้าจอ Desktop ของบอร์ด Raspberry Pi

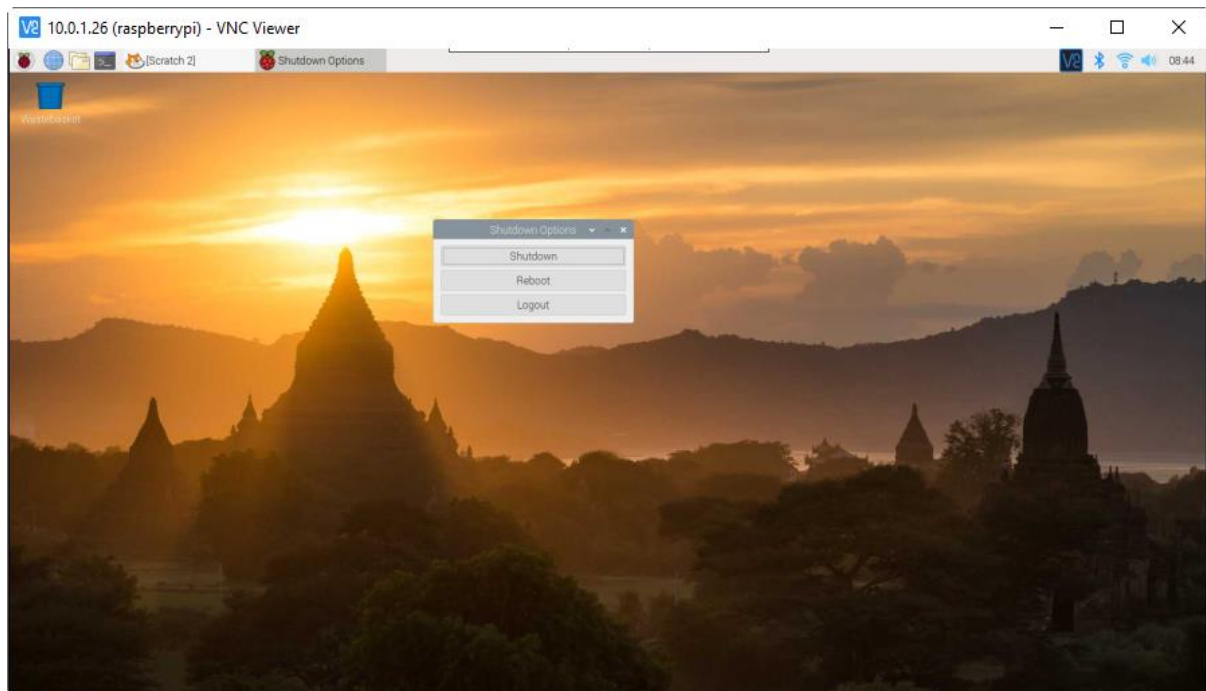


การเปิด/ปิดบอร์ด Raspberry Pi

1. ไปที่ Pi > Shutdown

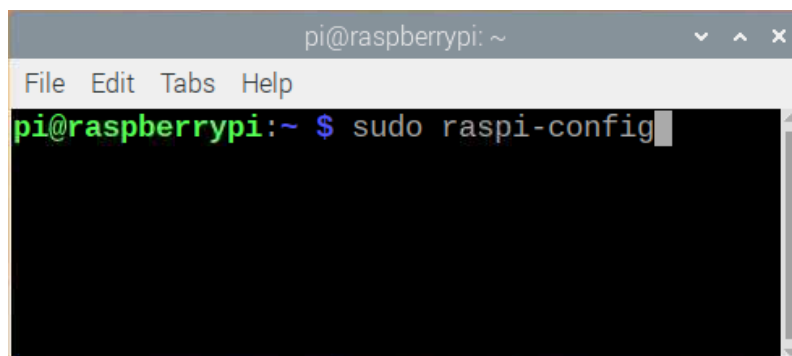


2. คลิกเลือก Option ที่ต้องการ ซึ่งได้แก่ Shutdown, Reboot หรือ Logout

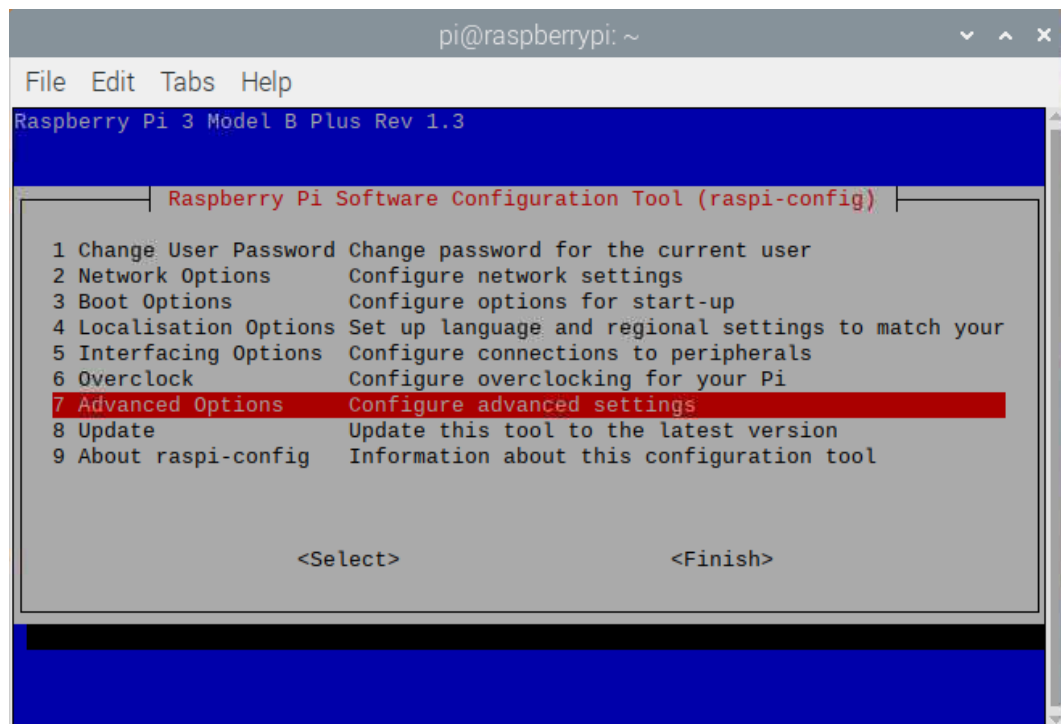


การเปิดใช้งาน Sound

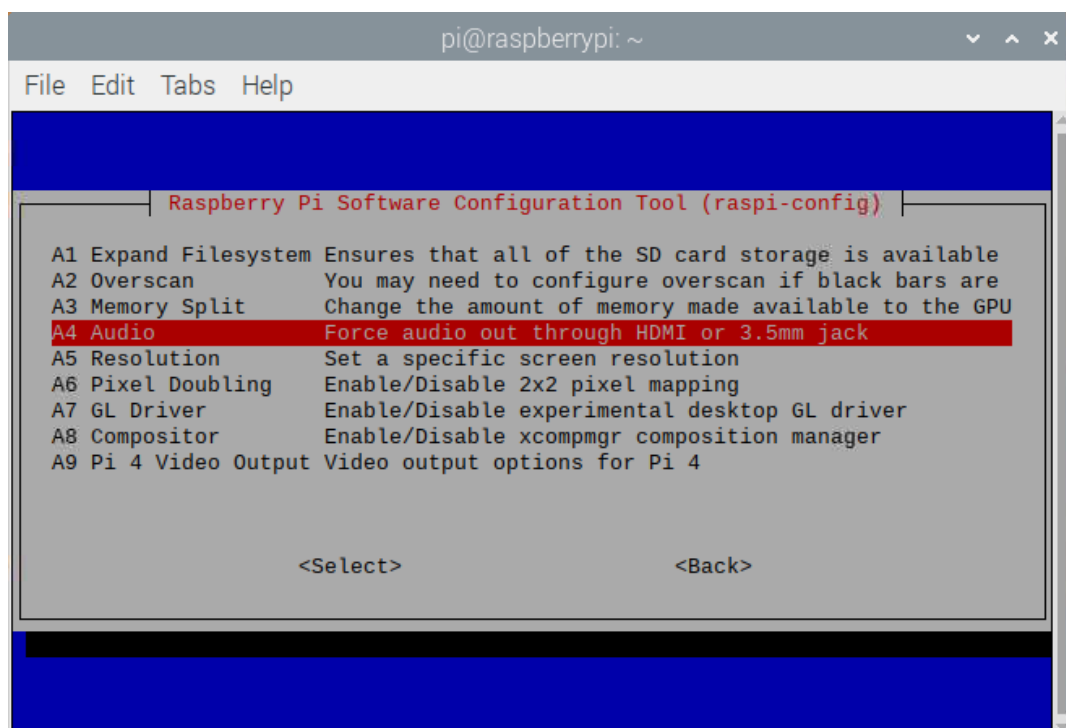
1. เปิดโปรแกรม Terminal ขึ้นมา แล้วใส่โค้ดคำสั่ง **sudo raspi-config**



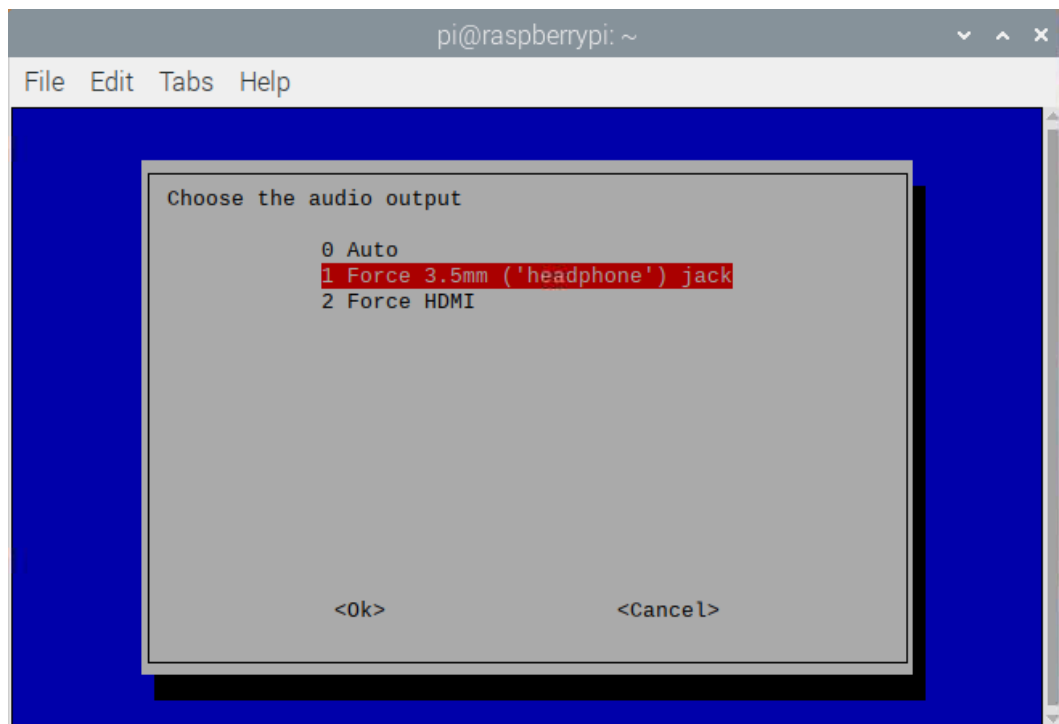
2. เลือกเมนู **Advanced Options** แล้วกดปุ่ม Enter



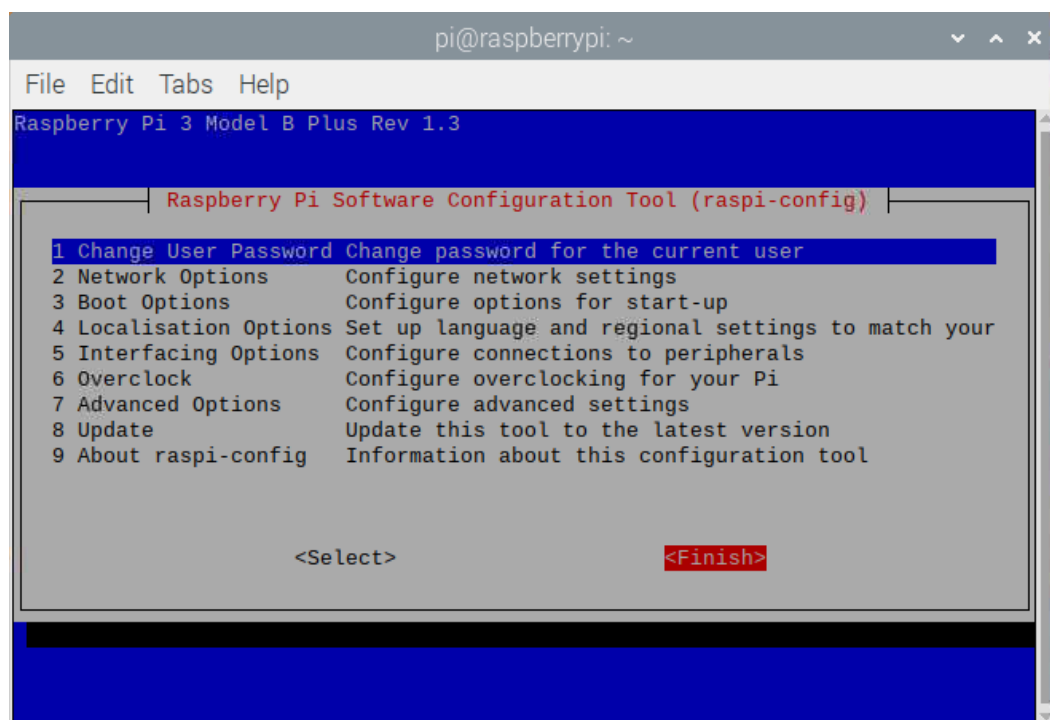
3. เลือกเมนู **Audio** แล้วกดปุ่ม Enter



4. เลือกเมนู **Force 3.5mm ('headphone') jack** แล้วกดปุ่ม Enter



5. กดปุ่ม Tab เพื่อเลื่อนมาที่ <Finish> แล้วกดปุ่ม Enter



6. ทำการรีสตาร์ทบอร์ด โดยพิมพ์คำสั่ง **reboot**

7. ทดสอบโดยการนำหูฟังเสียบเข้าไปในช่องเสียบหูฟังที่ตัวบอร์ด Raspberry Pi หลังจากนั้นลองเปิด Web Browser แล้วเข้าไปเว็บไซต์ Youtube เพื่อเปิดดูวิดีโอ เพื่อทดสอบเสียง

หน่วยการเรียนรู้ที่ ๓ แนะนำภาษา Python และ Scratch เบื้องต้น

กลุ่มสาระการเรียนรู้วิทยาศาสตร์

สาระที่ ๔ เทคโนโลยี

มาตรฐาน ว ๔.๒ เข้าใจและใช้แนวคิดเชิงคำนวณในการแก้ปัญหาที่พบในชีวิตจริงอย่างเป็นขั้นตอนและเป็นระบบ ใช้เทคโนโลยีสารสนเทศและการสื่อสารในการเรียนรู้ การทำงาน และการแก้ปัญหาได้อย่างมีประสิทธิภาพ รู้เท่าทัน และมีจริยธรรม

ตัวชี้วัด

ว ๔.๒ ม.๑/๒ ออกแบบและเขียนโปรแกรมอย่างง่ายเพื่อแก้ปัญหาทางคณิตศาสตร์หรือวิทยาศาสตร์

ว ๔.๒ ม.๒/๒ ออกแบบและเขียนโปรแกรมที่ใช้ตรรกะและฟังก์ชันในการแก้ปัญหา

สาระสำคัญ

๑. เรียนรู้และทำความเข้าใจกับภาษา Python

๒. เรียนรู้และทำความเข้าใจกับภาษา Scratch

สาระการเรียนรู้

ความรู้

- ความรู้พื้นฐานและตัวอย่างการเขียนโปรแกรมด้วยภาษา Python
- ความรู้พื้นฐานและตัวอย่างการเขียนโปรแกรมด้วยภาษา Scratch

ทักษะ/กระบวนการ

- ปฏิบัติตามตัวอย่างที่แสดงให้เห็นทีละขั้นตอนจากขั้นพื้นฐานไปสู่ขั้นที่ซับซ้อนขึ้น
- ฝึกปฏิบัติด้วยตนเองจนเกิดความชำนาญ หรือทำได้โดยอัตโนมัติ ซึ่งอาจเป็นงานชิ้นเดิมหรืองานที่คิดขึ้นใหม่

ที่คิดขึ้นใหม่

- ใช้กระบวนการคณิตศาสตร์ด้านการคิดคำนวณและการแก้ปัญหา
- กระบวนการกลุ่ม มุ่งให้ผู้เรียนทำงานร่วมกันเป็นทีม

คุณลักษณะที่พึงประสงค์

- รักชาติ ศาสน์ กษัตริย์
- ซื่อสัตย์
- มีวินัย
- ใฝ่เรียนรู้
- อยู่อย่างพอเพียง
- มุ่งมั่นในการทำงาน
- รักความเป็นไทย
- มีจิตสาธารณะ

หน่วยการเรียนรู้ที่ 3

แนะนำภาษา Python และ Scratch เบื้องต้น

ความรู้เบื้องต้นเบื้องต้นเกี่ยวกับ Python

Python เป็นภาษาเขียนโปรแกรมระดับสูงที่ใช้กันอย่างกว้างขวางในการเขียนโปรแกรมสำหรับวัตถุประสงค์ทั่วไป ภาษา Python นั้นสร้างโดย Guido van Rossum และถูกเผยแพร่ครั้งแรกในปี 1991 Python นั้นเป็นภาษาแบบ interpreter ที่ถูกออกแบบโดยมีปรัชญาที่จะทำให้โค้ดอ่านได้ง่ายขึ้น และโครงสร้างของภาษานั้นจะทำให้โปรแกรมเมอร์สามารถเข้าใจแนวคิดการเขียนโค้ดโดยใช้บรรทัดที่น้อยลงกว่าภาษาอย่าง C++ และ Java ซึ่งภาษานั้นถูกกำหนดให้มีโครงสร้างที่ตั้งใจให้การเขียนโค้ดเข้าใจง่ายทั้งในโปรแกรมเล็กไปจนถึงโปรแกรมขนาดใหญ่

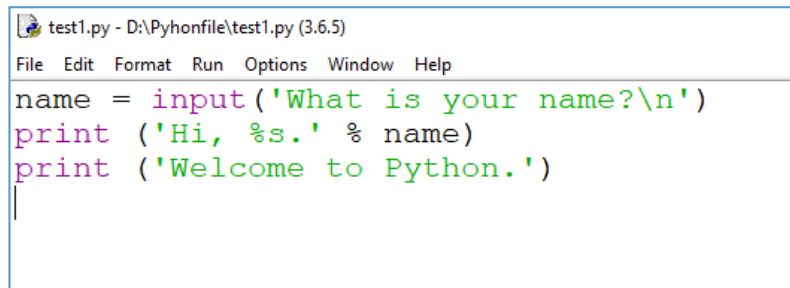
Python นั้นมีคุณสมบัติเป็นภาษาเขียนโปรแกรมแบบไดนามิกส์และมีระบบการจัดการหน่วยความจำอัตโนมัติและสนับสนุนการเขียนโปรแกรมหลายรูปแบบ ที่ประกอบไปด้วย การเขียนโปรแกรมเชิงวัตถุ imperative การเขียนโปรแกรมแบบฟังก์ชัน และการเขียนโปรแกรมแบบขั้นตอน มันมีไลบรารีที่ครอบคลุมการทำงานอย่างหลากหลาย ตัวแปรในภาษา Python นั้นมีให้ใช้ในหลายระบบปฏิบัติการ ทำให้โค้ดของภาษา Python สามารถรันในระบบต่างๆ ได้อย่างกว้างขวาง C Python นั้นเป็นการพัฒนาในขั้นต้นของ Python ซึ่งเป็นโปรแกรมแบบ open source และมีชุมชนสำหรับเป็นต้นแบบในการพัฒนา เนื่องจากได้มีการนำไปพัฒนากระจายไปอย่างหลากหลาย C Python นั้นจึงถูกจัดการโดยองค์กรไม่แสวงหาผลกำไรอย่าง Python Software Foundation

ภาษา Python นั้นกำเนิดขึ้นในปลายปี 1980 และการพัฒนาของมันนั้นเริ่มต้นใน December 1989 โดย Guido van Rossum ที่ Centrum Wiskunde & Informatica (CWI) ในประเทศเนเธอร์แลนด์ เนื่องในผู้ประสบความสำเร็จในการสร้างภาษา ABC ที่มีความสามารถสำหรับการ exception handling และการติดต่อผลานกับระบบปฏิบัติการ Amoeba ซึ่ง Van Rossum นั้นเป็นผู้เขียนหลักการของภาษา Python และเขาทำหน้าที่เป็นกลางในการตัดสินใจสำหรับทิศทางการพัฒนาของภาษา Python

โครงสร้างของภาษา Python

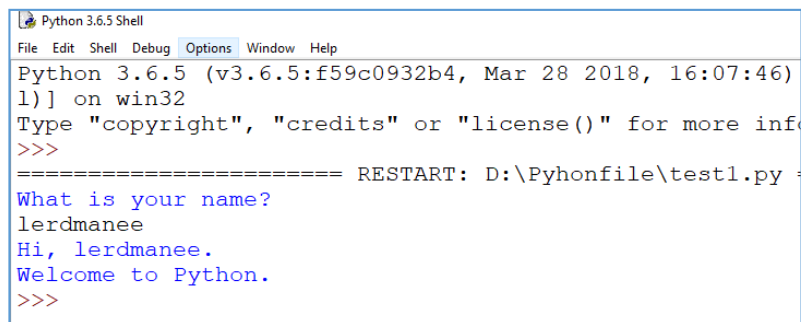
ภาษา Python ในภาษาคอมพิวเตอร์นั้นก็มีความโครงสร้างของภาษาเช่นเดียวกับภาษามนุษย์ ซึ่งสิ่งเหล่านี้ถูกกำหนดเพื่อเป็นรูปแบบและวิธีในการเขียนโปรแกรมในภาษา Python ใช้สำหรับควบคุมวิธีที่จะเขียนโค้ดเพื่อให้เข้าใจโดยตัวแปรภาษาหรือคอมพิวเตอร์

เริ่มต้นการเรียนรู้ในภาษา Python มาดูตัวอย่างของโปรแกรมอย่างง่าย โดยเป็นโปรแกรมที่ถามชื่อผู้ใช้และแสดงข้อความทักทายทางหน้าจอ มาเริ่มเขียนโปรแกรมแรกในภาษา Python โดยเขียนตามคำสั่งดังภาพ



```
test1.py - D:\Pyhonfile\test1.py (3.6.5)
File Edit Format Run Options Window Help
name = input('What is your name?\n')
print ('Hi, %s.' % name)
print ('Welcome to Python.')
|
```

ในตัวอย่างเป็นโปรแกรมในการรับชื่อและแสดงข้อความทักทายออกทางหน้าจอ ในการรันโปรแกรมสามารถรันได้หลายวิธี แต่ที่แนะนำคือการใช้ Python shell ให้เปิด Python shell ขึ้นมาแล้วกดสร้างไฟล์ใหม่โดยไปที่ File -> New File จะปรากฏกล่อง Text editor ของภาษา Python ขึ้นมา เพื่อรันโปรแกรม Run -> Run Module หรือกด F5 โปรแกรมจะเปลี่ยนกลับไปยัง Python shell และเริ่มดำเนินงาน



```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46)
l)] on win32
Type "copyright", "credits" or "license()" for more info
>>>
===== RESTART: D:\Pyhonfile\test1.py =====
What is your name?
lerdmanee
Hi, lerdmanee.
Welcome to Python.
>>>
```

ผลลัพธ์การทำงานในการรันโปรแกรม test1.py จาก Python shell ในตัวอย่างเราได้กรอกชื่อเป็น "lerdmanee" และหลังจากนั้นโปรแกรมได้แสดงข้อความทักทายและจบการทำงาน

โมดูล (Module)

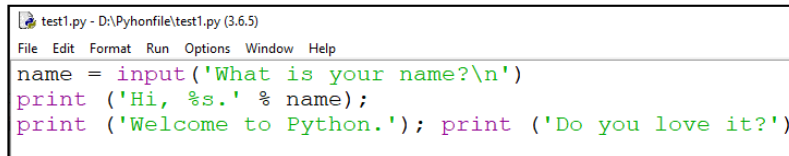
ในตัวอย่างโปรแกรมรับชื่อ เป็นโปรแกรมในบทเรียน Python และเราได้บันทึกเป็นไฟล์ที่ชื่อว่า test1.py ซึ่งไฟล์ของภาษา Python นั้นจะเรียกว่า Module ซึ่ง Module จะประกอบไปด้วยคลาส ฟังก์ชัน และตัวแปรต่างๆ และนอกจากนี้เรายังสามารถ import โมดูลอื่นเข้ามาในโปรแกรมได้ ซึ่งโมดูลอาจจะอยู่ภายใน package ซึ่งเป็นเหมือน directory ของ Module ในตัวอย่าง test1.py จึงเป็นโมดูลของโปรแกรมนั้น

คอมเมนต์ (Comment)

คอมเมนต์ในภาษา Python นั้นเริ่มต้นด้วยเครื่องหมาย # คอมเมนต์สามารถเริ่มต้นที่ตำแหน่งแรกของบรรทัดและหลังจากนั้นจะประกอบไปด้วย While space หรือโค้ดของโปรแกรม หรือคำอธิบาย ซึ่งโดยทั่วไปแล้วคอมเมนต์มักจะใช้สำหรับอธิบายข้อโค้ดที่เราเขียนขึ้นและไม่มีผลต่อการทำงานของโปรแกรม

Statement

Statement คือคำสั่งการทำงานของโปรแกรม แต่ละคำสั่งในภาษา Python นั้นจะแบ่งแยกด้วยการขึ้นบรรทัดใหม่ ซึ่งจะแตกต่างจากภาษา C และ Java ซึ่งใช้เครื่องหมายเซมิโคลอน สำหรับการจบคำสั่งการทำงาน แต่อย่างไรก็ตามในภาษา Python นั้น สามารถมีหลายคำสั่งในบรรทัดเดียวกันได้โดยการใช้เครื่องหมายเซมิโคลอน ;

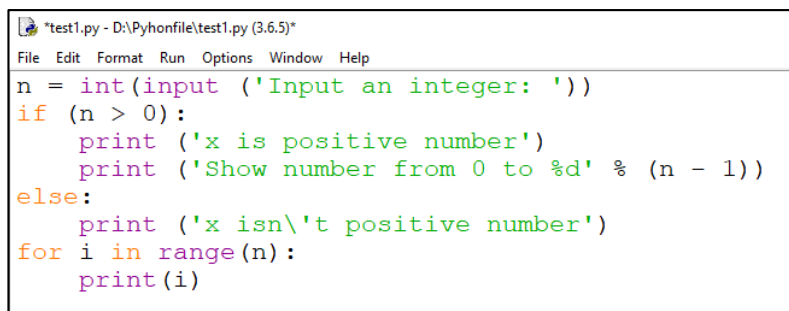


```
test1.py - D:\Pythonfile\test1.py (3.6.5)
File Edit Format Run Options Window Help
name = input('What is your name?\n')
print ('Hi, %s.' % name);
print ('Welcome to Python. '); print ('Do you love it?')
```

ในตัวอย่าง มี 4 คำสั่งในโปรแกรม สองบรรทัดแรกเป็นคำสั่งที่ใช้บรรทัดใหม่ในการจบคำสั่ง ซึ่งเป็นแบบปกติในภาษา Python และบรรทัดสุดท้ายมีสองคำสั่งในบรรทัดเดียวกันที่คั่นด้วยเครื่องหมาย ; สำหรับการจบคำสั่ง

Indentation and while space

ในภาษา Python นั้นใช้ While space และ Tab สำหรับกำหนดบล็อกของโปรแกรม เช่น คำสั่ง If Else For หรือการประกาศฟังก์ชัน ซึ่งคำสั่งเหล่านี้เป็นคำสั่งแบบบล็อก โดยจำนวนช่องว่างที่ใช้นั้นต้องเท่ากัน มาดูตัวอย่างของบล็อกคำสั่งในภาษา Python

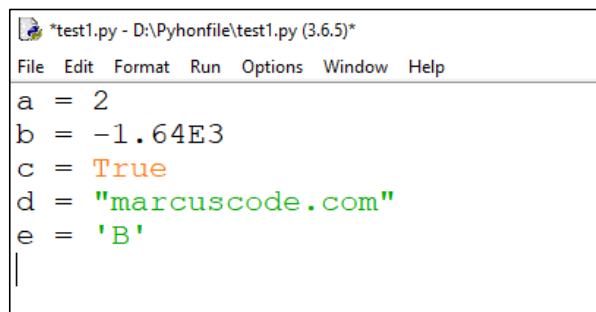


```
*test1.py - D:\Pythonfile\test1.py (3.6.5)*
File Edit Format Run Options Window Help
n = int(input('Input an integer: '))
if (n > 0):
    print ('x is positive number')
    print ('Show number from 0 to %d' % (n - 1))
else:
    print ('x isn\'t positive number')
for i in range(n):
    print(i)
```

ในตัวอย่าง เป็นบล็อกของโปรแกรมจาก 3 คำสั่ง ในคำสั่งแรกคือ If ในบล็อกนี้มีสองคำสั่งย่อยอยู่ภายใน ที่หัวของบล็อกนั้นจะต้องมีเครื่องหมาย : กำหนดหลังคำสั่งในการเริ่มต้นบล็อกเสมอ อีกสองบล็อกสุดท้ายนั้นเป็นคำสั่ง Else และ For ซึ่งมีหนึ่งคำสั่งย่อยอยู่ภายใน ในภาษา Python นี้เข้มงวดกับช่องว่างภายในบล็อกมาก นั้นหมายความว่าทุกคำสั่งย่อยภายในบล็อกนั้นต้องมีจำนวนช่องว่างเท่ากันเสมอ

Literals

ในการเขียนโปรแกรม Literal คือเครื่องหมายที่ใช้แสดงค่าของค่าคงที่ในโปรแกรม ในภาษา Python นั้นมี Literal ของข้อมูลประเภทต่างๆ เช่น Integer Floating-point number และ String หรือแม้กระทั่งตัวอักษร และ Boolean ตัวอย่างของการกำหนด Literal ให้กับตัวแปรในภาษา Python

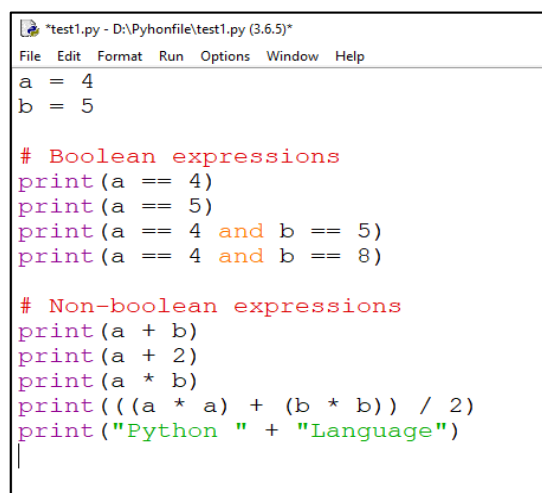


```
*test1.py - D:\Pyhonfile\test1.py (3.6.5)*
File Edit Format Run Options Window Help
a = 2
b = -1.64E3
c = True
d = "marcuscode.com"
e = 'B'
```

ในตัวอย่าง เป็นการกำหนด Literal ประเภทต่างๆ ให้กับตัวแปร ในค่าที่เป็นแบบตัวเลขนั้นสามารถกำหนดค่าลงไปโดยตรงได้ทันทีและสามารถกำหนดในรูปแบบสั้นได้อย่างในตัวแปร b และสำหรับ Boolean นั้นจะเป็น True ส่วน String หรือ Character นั้นจะต้องอยู่ภายในเครื่องหมาย double quote หรือ single quote เสมอ

Expressions

Expression คือการทำงานร่วมกันระหว่างค่าตั้งแต่หนึ่งไปจนถึงหลายค่า โดยค่าเหล่านี้จะมีตัวดำเนินการสำหรับควบคุมการทำงาน ในภาษา Python นั้น Expression จะมีสองแบบคือ Boolean expression เป็นการกระทำกันของตัวแปรและตัวดำเนินการและจะได้ผลลัพธ์เป็นค่า Boolean โดยทั่วไปแล้วมักจะเป็นตัวดำเนินการเปรียบเทียบค่าและตัวดำเนินการตรรกศาสตร์ และ Expression ทางคณิตศาสตร์ คือการกระทำกันกับตัวดำเนินการและได้ค่าใหม่ที่ไม่ใช่ Boolean นี่เป็นตัวอย่างของ Expressions ในภาษา Python



```
*test1.py - D:\Pyhonfile\test1.py (3.6.5)*
File Edit Format Run Options Window Help
a = 4
b = 5

# Boolean expressions
print(a == 4)
print(a == 5)
print(a == 4 and b == 5)
print(a == 4 and b == 8)

# Non-boolean expressions
print(a + b)
print(a + 2)
print(a * b)
print(((a * a) + (b * b)) / 2)
print("Python " + "Language")
```

ในตัวอย่าง เรามีตัวแปร a และ b และกำหนดค่าให้กับตัวแปรเหล่านี้และทำงานกับตัวดำเนินการประเภทต่างๆ ที่แสดง Expression ในรูปแบบของ Boolean expression ที่จะได้ผลลัพธ์สุดท้ายเป็นเพียงค่า True และ False เท่านั้น ส่วน Non-Boolean expression นั้นสามารถเป็นค่าใดๆ ที่ไม่ใช่ Boolean

Keywords

Keyword เป็นคำที่ถูกสงวนไว้ในการเขียนโปรแกรมภาษา Python เราไม่สามารถใช้คำสั่งเหล่านี้ในการตั้งชื่อตัวแปร ชื่อฟังก์ชัน คลาส หรือ identifier ใดๆ ที่กำหนดขึ้นโดยโปรแกรมเมอร์ นี่เป็นรายการของ

Keyword ในภาษา Python

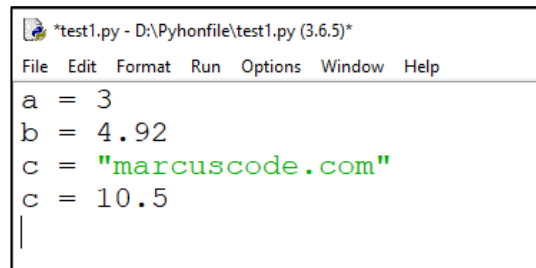
False	None	True	and
as	assert	break	class
continue	def	del	el if
else	except	finally	for
from	global	if	import
in	is	lambda	nonlocal
not	or	pass	raise
return	try	while	with
yield			

ตัวแปรและประเภทข้อมูล

ตัวแปร

ตัวแปร (variable) คือชื่อหรือเครื่องหมายที่กำหนดขึ้นสำหรับใช้เก็บค่าในหน่วยความจำ ตัวแปรจะมีชื่อ (identifier) สำหรับใช้ในการอ้างถึงข้อมูลของมัน ในการเขียนโปรแกรม ค่าของตัวแปรสามารถที่จะกำหนดได้ใน run-time หรือเปลี่ยนแปลงอยู่ตลอดเวลาในขณะที่โปรแกรมทำงาน (executing)

ในการเขียนโปรแกรมคอมพิวเตอร์นั้น ตัวแปรจะแตกต่างจากตัวแปรในทางคณิตศาสตร์ ค่าของตัวแปรนั้นไม่จำเป็นต้องประกอบไปด้วยสูตรหรือสมการที่สมบูรณ์เหมือนกับในคณิตศาสตร์ ในคอมพิวเตอร์ตัวแปรนั้นอาจจะมีการทำงานซ้ำๆ เช่น การกำหนดค่าในที่หนึ่ง และนำไปใช้อีกที่หนึ่งในโปรแกรม และนอกจากนี้ยังสามารถกำหนดค่าใหม่ให้กับตัวแปรได้ตลอดเวลา ต่อไปเป็นตัวอย่างของการประกาศตัวแปรในภาษา Python

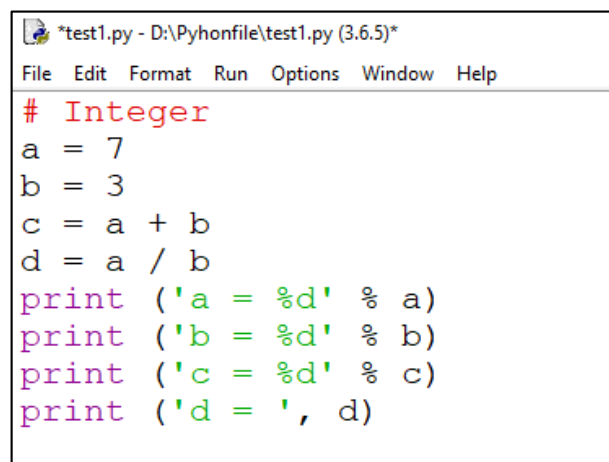


```
*test1.py - D:\Pyhonfile\test1.py (3.6.5)*
File Edit Format Run Options Window Help
a = 3
b = 4.92
c = "marcuscode.com"
c = 10.5
|
```

ในตัวอย่าง เราได้ทำการประกาศ 3 ตัวแปร ในการประกาศตัวแปรในภาษา Python คุณไม่จำเป็นต้องระบุประเภทของตัวแปรในตอนทีประกาศเหมือนในภาษา C ในตัวแปร a มีค่าเป็น 3 และเป็นประเภทเป็น Integer ตัวแปร b มีค่าเป็น 4.92 และเป็นประเภทเป็น Float และตัวแปร c มีค่าเป็น "marcuscode.com" และเป็นประเภท String ภายหลังเราได้เปลี่ยนค่าของตัวแปร c เป็น 10.5 ตัวแปรกลายเป็นประเภท Float

Numbers

ในภาษา Python นั้นสนับสนุนข้อมูลแบบตัวเลข ซึ่งข้อมูลประเภทนี้จะแบ่งออกเป็น Integer Float Decimal และ Complex อย่างไรก็ตามเราจะเน้นย้ำใน Integer ซึ่งเป็นการเก็บข้อมูลแบบจำนวนเต็ม และ Float เป็นข้อมูลแบบจำนวนจริง สำหรับประเภทแบบ Decimal นั้นแตกต่างไปจาก Float คือสามารถเก็บความละเอียดของจุดทศนิยมได้มากกว่า นอกจากนี้ Python ยังสนับสนุนตัวเลขในรูปแบบ Complex ที่แสดงในรูปแบบ $a+bj$ ต่อไปเป็นตัวอย่างในการประกาศและใช้งานตัวแปรแบบตัวเลขในภาษา Python

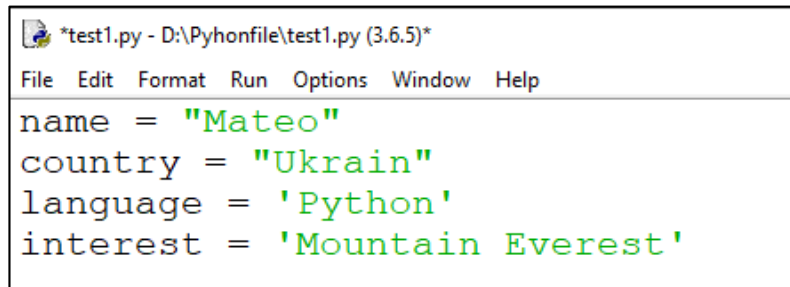


```
*test1.py - D:\Pyhonfile\test1.py (3.6.5)*
File Edit Format Run Options Window Help
# Integer
a = 7
b = 3
c = a + b
d = a / b
print ('a = %d' % a)
print ('b = %d' % b)
print ('c = %d' % c)
print ('d = ', d)
```

ในตัวอย่าง เป็นการประกาศและใช้งานตัวแปรประเภท Integer เราได้ทำการประกาศตัวแปรและกำหนดค่าให้กับ a และ b ในการแสดงผลในรูปแบบของ String format กับฟังก์ชัน print() นั้นจะใช้ specifier เป็น %d เราสามารถกำหนดค่าให้กับตัวแปรได้โดย Literal หรือ Expression และการหารตัวเลขในภาษา Python นั้นจะได้ค่าเป็น Float เสมอ ถึงแม้ตัวเลขทั้งสองจะเป็น Integer ก็ตาม เช่นในตัวแปร d

String

String นั้นเป็นประเภทข้อมูลที่สำคัญและใช้งานทั่วไปในการเขียนโปรแกรม ในภาษาเขียนโปรแกรมส่วนมากแล้วจะมีประเภทข้อมูลแบบ String และในภาษา Python เช่นกัน String เป็นลำดับของตัวอักษรหลายตัวเรียงต่อกัน ซึ่งในภาษา Python นั้น String จะอยู่ในเครื่องหมาย Double quote หรือ Single quote เท่านั้น นอกจากนี้ในภาษา Python ยังมีฟังก์ชันในการจัดการกับ String มากมายซึ่งเราจะพูดถึงอีกครั้งในบทของ String ในบทนี้มาทำความรู้จักกับ String เบื้องต้นกันก่อน



```
*test1.py - D:\Pythonfile\test1.py (3.6.5)*
File Edit Format Run Options Window Help
name = "Mateo"
country = "Ukrain"
language = 'Python'
interest = 'Mountain Everest'
```

ในตัวอย่าง เป็นการประกาศตัวแปรประเภท String สองตัวแปรแรกเป็นการประกาศโดยใช้ Double quote และสองตัวแปรต่อมาเป็นการใช้ Single quote ซึ่งคุณสามารถใช้แบบไหนก็ได้ แต่มีสิ่งที่แตกต่างกันเล็กน้อยคือเกี่ยวกับการกำหนดตัวอักขระพิเศษหรือเรียกว่า Escape character

List

List เป็นประเภทข้อมูลที่เก็บข้อมูลแบบเป็นชุดและลำดับ กล่าวคือมันสามารถเก็บข้อมูลได้หลายค่าในตัวแปรเดียว และมี Index สำหรับเข้าถึงข้อมูล ในภาษา Python นั้น List จะเป็นเหมือนอาร์เรย์ในภาษา C มันสามารถเก็บข้อมูลได้หลายตัวและยังสามารถเป็นประเภทข้อมูลที่แตกต่างกันได้อีกด้วย มาดูการประกาศและใช้งาน List ในเบื้องต้น

```

*test1.py - D:\Pythonfile\test1.py (3.6.5)*
File Edit Format Run Options Window Help
# Declare lists
numbers = [1, 2, 4, 6, 8, 19]
names = ["Mateo", "Danny", "James", "Thomas", "Luke"]
mixed = [-2, 5, 84.2, "Mountain", "Python"]
# Display lists
print(numbers)
print(names)
print(mixed)
# Display lists using the for loops
for n in numbers:
    print(n, end=" ")
print()

for n in names:
    print(n, end=" ")
print()

for n in mixed:
    print(n, end=" ")
print()

```

ในตัวอย่าง เราได้ทำการประกาศ 3 Lists โดยตัวแปรแรกนั้นเป็น List ของตัวเลข และตัวแปรที่สองเป็น List ของ String และตัวแปรสุดท้ายเป็น List แบบรวมกันของประเภทข้อมูล เราใช้ฟังก์ชัน print() ในการแสดงผลข้อมูลใน List และใช้คำสั่ง For loop ในการอ่านค่าในลิสต์และนำมาแสดงผลเช่นกัน

```

Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [
1)] on win32
Type "copyright", "credits" or "license()" for more infor
>>>
===== RESTART: D:\Pythonfile\test1.py =====
[1, 2, 4, 6, 8, 19]
['Mateo', 'Danny', 'James', 'Thomas', 'Luke']
[-2, 5, 84.2, 'Mountain', 'Python']
1 2 4 6 8 19
Mateo Danny James Thomas Luke
-2 5 84.2 Mountain Python
>>>

```

ฟังก์ชันที่ใช้กับตัวแปร

ในภาษา Python นั้นมีฟังก์ชันที่สร้างมาเพื่อให้ใช้งานกับตัวแปร เช่น ฟังก์ชันสำหรับหาขนาดของตัวแปร ฟังก์ชันในการหาประเภทของตัวแปร ฟังก์ชันลบตัวแปรออกไปในหน่วยความจำ และฟังก์ชันในการตรวจสอบว่าตัวแปรมีอยู่หรือไม่ ซึ่งในบางครั้งการเขียนโปรแกรมก็จำเป็นที่คุณอาจจะต้องมีการตรวจสอบสิ่งเหล่านี้ในขณะที่โปรแกรมทำงาน นี่เป็นตัวอย่างการใช้งาน

```

*test1.py - D:\Pyhonfile\test1.py (3.6.5)*
File Edit Format Run Options Window Help
import sys
a = 8
b = 13.4
c = "Python"
d = [1, 2, 3, 4]
print('Size of a = ', sys.getsizeof(a))
print('Type of a = ', type(a))
print('Size of b = ', sys.getsizeof(b))
print('Type of b = ', type(b))
print('Size of c = ', sys.getsizeof(c))
print('Type of c = ', type(c))
print('Size of d = ', sys.getsizeof(d))
print('Type of d = ', type(d))
del a
del b, c, d
if 'a' in locals():
    print("a is exist")
else:
    print("a is not exist")

```

ในตัวอย่าง เราได้ประกาศตัวแปรกับประเภทต่างๆ เราได้ฟังก์ชัน `getsizeof()` สำหรับหาขนาดของตัวแปรที่มีหน่วยเป็น Byte และฟังก์ชัน `type()` สำหรับประเภทของตัวแปรว่าอยู่ในคลาสไหน ฟังก์ชัน `del()` สำหรับยกเลิกหรือลบการประกาศตัวแปรออกไปจากหน่วยความจำ และสุดท้ายเป็นการตรวจสอบว่าตัวแปรถูกประกาศและหรือยังในฟังก์ชัน `locals()` สำหรับตรวจสอบตัวแปรในโมดูลปัจจุบัน หรือ `globals()` สำหรับตรวจสอบตัวแปรในโปรแกรมทั้งหมด

```

Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46)
1)] on win32
Type "copyright", "credits" or "license()" for more inf
>>>
===== RESTART: D:\Pyhonfile\test1.py
Size of a = 14
Type of a = <class 'int'>
Size of b = 16
Type of b = <class 'float'>
Size of c = 31
Type of c = <class 'str'>
Size of d = 52
Type of d = <class 'list'>
a is not exist
>>> |

```

การรับค่าและการแสดงผล

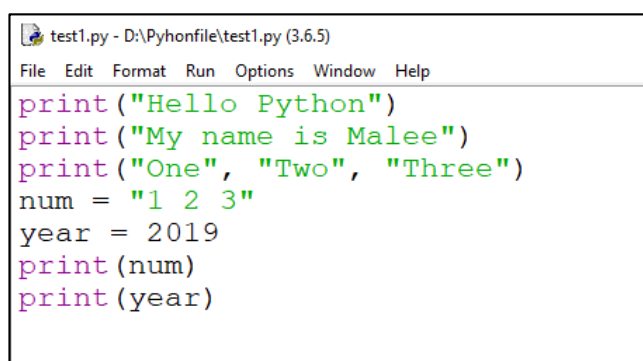
การรับค่าและการแสดงผลพื้นฐานในภาษา Python ซึ่งเป็นสิ่งสำคัญที่โปรแกรมต้องมีการติดต่อกับผู้ใช้ การรับค่าคือการรับข้อมูลจากภายนอกโดยทั่วไปแล้วมักจะเป็นการรับค่าทางคีย์บอร์ด ส่วนการแสดงผลนั้นจะเป็นบน Console ถ้าหากรัน Python บนเว็บเซิร์ฟเวอร์ การแสดงผลจะเป็นทางเว็บเบราว์เซอร์แทน และการรับค่าจะเป็นทาง URI หรือเว็บฟอร์ม

การแสดงผลด้วยฟังก์ชัน print()

ในการแสดงผลในภาษา Python นั้นจะใช้ฟังก์ชัน print() เพื่อแสดงผลข้อความ ตัวเลข หรือข้อมูลประเภทอื่นๆ ออกทางหน้าจอหรือสร้าง Http response นี่เป็นรูปแบบของการใช้งานฟังก์ชัน print() ในภาษา Python

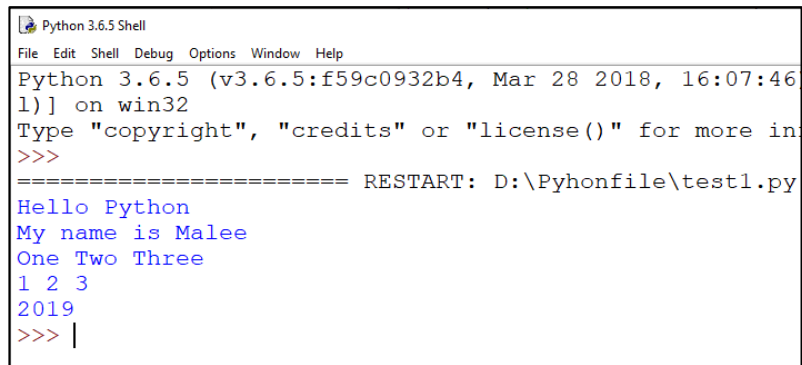
```
print(value, ..., sep = ' ', end = '\n');
```

ในรูปแบบการใช้งาน ฟังก์ชัน print() เราสามารถส่งอาร์กิวเมนต์ได้ตั้งแต่หนึ่งถึงหลายตัวเข้าไปในฟังก์ชัน นอกจากนี้ฟังก์ชันยังมี keyword อาร์กิวเมนต์ sep ซึ่งเป็นตัวแบ่งหากอาร์กิวเมนต์ที่ส่งเข้ามามีมากกว่า 1 ตัว ซึ่งมีค่า default เป็น whitespace และ keyword อาร์กิวเมนต์ end เป็นการแสดงผลในตอนท้ายของฟังก์ชัน ซึ่งมีค่า default เป็น \n หมายถึงการขึ้นบรรทัดใหม่ มาดูตัวอย่างการใช้งานฟังก์ชัน



```
test1.py - D:\Pythonfile\test1.py (3.6.5)
File Edit Format Run Options Window Help
print("Hello Python")
print("My name is Malee")
print("One", "Two", "Three")
num = "1 2 3"
year = 2019
print(num)
print(year)
```

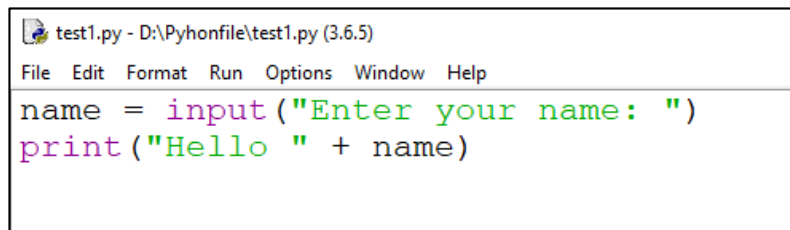
ในตัวอย่าง เป็นการแสดงผลในภาษา Python โดยในคำสั่งแรกและคำสั่งที่สองนั้นเป็นการแสดงข้อความ และในคำสั่งที่สามเป็นการส่งค่าแบบหลายอาร์กิวเมนต์ และในสองคำสั่งสุดท้ายเป็นการแสดงผลข้อมูลจากตัวแปร name และตัวแปร year



```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) on win32
Type "copyright", "credits" or "license()" for more in
>>>
===== RESTART: D:\Pyhonfile\test1.py
Hello Python
My name is Malee
One Two Three
1 2 3
2019
>>> |
```

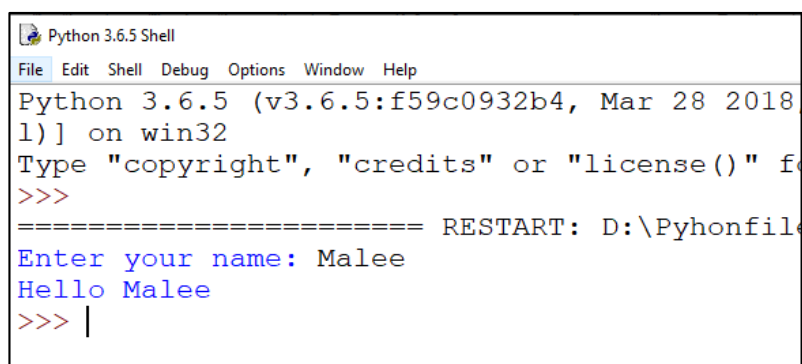
การรับค่าจาก Keyboard ด้วยฟังก์ชัน input()

นอกจากการแสดงผลแล้วนั้น การติดต่อกับผู้ใช้ในอีกรูปแบบหนึ่งคือการรับค่า โดยทั่วไปแล้ว มักจะเป็นการรับค่าทางคีย์บอร์ด ในภาษา Python เราใช้ฟังก์ชัน input() สำหรับการรับค่า String จากทางคีย์บอร์ด มาดูตัวอย่างการรับค่าจากผู้ใช้ในภาษา Python



```
test1.py - D:\Pyhonfile\test1.py (3.6.5)
File Edit Format Run Options Window Help
name = input("Enter your name: ")
print("Hello " + name)
```

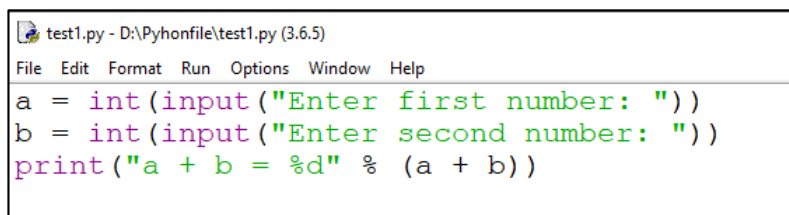
ในตัวอย่าง เป็นสำหรับการรับค่าชื่อจากคีย์บอร์ดและแสดงข้อความทักทายชื่อดังกล่าว ฟังก์ชัน input() เราได้ส่งอาร์กิวเมนต์เข้าไปในฟังก์ชันเพื่อเป็นข้อความบอกวิธีการใส่ค่ากับผู้ใช้ ฟังก์ชันจะส่งค่ากลับเป็น String ที่ผู้ใช้กรอกเข้ามาและจบการรับค่าด้วยการขึ้นบรรทัดใหม่ โดยที่ \n จะถูกตัดออกไป



```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) on win32
Type "copyright", "credits" or "license()" f
>>>
===== RESTART: D:\Pyhonfile
Enter your name: Malee
Hello Malee
>>> |
```

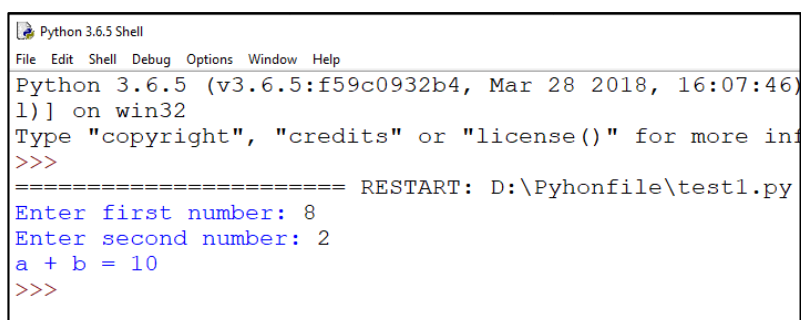
ผลลัพธ์การทำงานของโปรแกรม โดยเราได้กรอกชื่อ "Malee" และโปรแกรมแสดงข้อความทักทายทางหน้าจอ

ในการรับค่าด้วยฟังก์ชัน `input()` นั้นจะใช้กับการรับค่าที่เป็น String เท่านั้น และในการที่จะรับข้อมูลประเภทอื่นๆ เช่น ตัวเลข เราสามารถใช้ฟังก์ชันที่มากับภาษา Python ในการแปลงข้อมูลจาก String ไปเป็นข้อมูลประเภทอื่นได้ ต่อไปมาดูตัวอย่างการรับค่าตัวเลขในภาษา Python



```
test1.py - D:\Pyhonfile\test1.py (3.6.5)
File Edit Format Run Options Window Help
a = int(input("Enter first number: "))
b = int(input("Enter second number: "))
print("a + b = %d" % (a + b))
```

ในตัวอย่าง เป็นโปรแกรมในการรับค่าตัวเลขสองตัวจากคีย์บอร์ด เราใช้ฟังก์ชัน `int()` เพื่อแปลงข้อมูลแบบ String ให้เป็น Integer และนำมาใส่ในตัวแปร `a` และ `b` ตามลำดับ หลังจากนั้นเราแสดงผลบวกของตัวเลขทั้งสอง



```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46)
1] on win32
Type "copyright", "credits" or "license()" for more in
>>>
===== RESTART: D:\Pyhonfile\test1.py
Enter first number: 8
Enter second number: 2
a + b = 10
>>>
```

ผลลัพธ์การทำงานของโปรแกรม เราได้กรอกตัวเลขสองตัวและโปรแกรมแสดงผลรวมออกมา ในตัวอย่างคุณเห็นว่าเราสามารถใช้ฟังก์ชัน `input()` ในการรับค่าตัวเลขโดยการใช้ฟังก์ชัน `int()` ในการแปลง คุณสมารถรับค่าตัวเลขแบบอื่นได้เช่นกัน เช่น การใช้ฟังก์ชัน `float()` สำหรับแปลงข้อมูลแบบทศนิยม

ตัวดำเนินการ

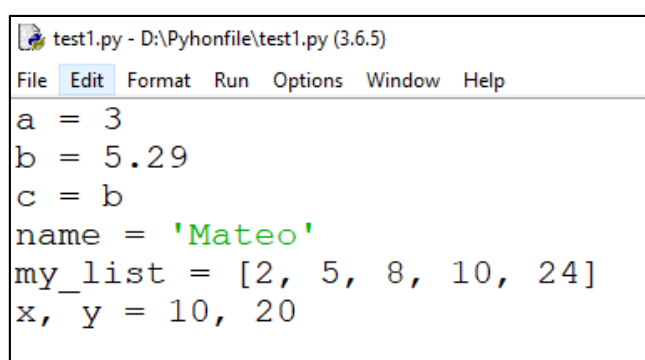
ตัวดำเนินการ (Operators) คือกลุ่มของเครื่องหมายหรือสัญลักษณ์ที่ใช้ทำงานเหมือนกับฟังก์ชัน แต่แตกต่างกันตรงไวยากรณ์หรือความหมายในการใช้งาน ในภาษา Python นั้นสนับสนุนตัวดำเนินการประเภทต่างๆ สำหรับการเขียนโปรแกรม เช่น ตัวดำเนินการ `+` เป็นตัวดำเนินการทางคณิตศาสตร์ที่ใช้สำหรับการบวกตัวเลขเข้าด้วยกัน หรือตัวดำเนินการ `>` เป็นตัวดำเนินการเพื่อให้เปรียบเทียบค่าสองค่า ตัวดำเนินการในภาษา Python

- ตัวดำเนินการกำหนดค่า Assignment operators
- ตัวดำเนินการทางคณิตศาสตร์ Arithmetic operators
- ตัวดำเนินการเปรียบเทียบ Comparison operators

- ตัวดำเนินการตรรกศาสตร์ Logical operators
- ตัวดำเนินการระดับบิต Bitwise operators
- ตัวดำเนินการแบบลำดับ Sequence Operators
- การตรวจสอบค่าความจริง Truth Value Testing

ตัวดำเนินการกำหนดค่า Assignment operator

ตัวดำเนินการที่เป็นพื้นฐานที่สุดสำหรับการเขียนโปรแกรมในทุกๆ ภาษาก็คือ ตัวดำเนินการกำหนดค่า (Assignment operator) ตัวดำเนินการนี้แสดงโดยใช้เครื่องหมายเท่ากับ (=) มันใช้สำหรับกำหนดค่าให้กับตัวแปร มาดูตัวอย่างการใช้งานในภาษา Python



```
test1.py - D:\Pythonfile\test1.py (3.6.5)
File Edit Format Run Options Window Help
a = 3
b = 5.29
c = b
name = 'Mateo'
my_list = [2, 5, 8, 10, 24]
x, y = 10, 20
```

ในตัวอย่าง เป็นการใช้งานตัวดำเนินการกำหนดค่าสำหรับกำหนดค่าให้กับตัวแปรประเภทต่างๆ โดยทั่วไปแล้ว ตัวดำเนินการกำหนดค่านั้นเกือบจะใช้ในหลายๆ ที่ในโปรแกรมและเป็นตัวดำเนินการที่ใช้บ่อยที่สุดของในบรรดาตัวดำเนินการทั้งหมด

ตัวดำเนินการทางคณิตศาสตร์ Arithmetic operators

ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic operators) คือตัวดำเนินการที่ใช้สำหรับการคำนวณทางคณิตศาสตร์ในพื้นฐาน เช่น การบวก การลบ การคูณ และการหาร มากไปกว่านั้น ในภาษา Python ยังมีตัวดำเนินการทางคณิตศาสตร์เพิ่มเติม เช่น การหารเอาเศษ (Modulo) การหารแบบเลขจำนวนเต็ม และการยกกำลัง เป็นต้น

Operator	Name	Example
+	Addition	a + b
-	Subtraction	a - b
*	Multiplication	a * b
/	Division	a / b

//	Division and floor	a // b
%	Modulo	a % b
**	Power	a ** b

ในตารางข้างบน เรามีตัวดำเนินการทางคณิตศาสตร์ประเภทต่างๆ สำหรับการคำนวณเกี่ยวกับคณิตศาสตร์เบื้องต้น อาจจะคุ้นเคยกับตัวดำเนินการบวก ลบ คูณ หาร ในการเรียนระดับมัธยมศึกษามาบ้างแล้ว ในภาษา Python นั้นสนับสนุนตัวดำเนินการสำหรับการหาเศษเช่นเดียวกับภาษาอื่นๆ และนอกจากนี้ ยังมีตัวดำเนินการแบบการหารที่ได้ผลลัพธ์เป็นจำนวนเต็ม และการหารเลขยกกำลังเพิ่มเข้ามา

ตัวอย่างการใช้ตัวดำเนินการประเภทต่างๆ ในภาษา Python

```

test1.py - D:\Pyhonfile\test1.py (3.6.5)
File Edit Format Run Options Window Help
a = 5
b = 3
print("a + b = ", a + b)
print("a - b = ", a - b)
print("a * b = ", a * b)
print("a / b = ", a / b)
print("a // b = ", a // b) # floor number to integer
print("a % b = ", a % b) # get division remainder
print("a ** b = ", a ** b) # power

```

ในตัวอย่าง ได้ประกาศตัวแปร a และ b และกำหนดค่าให้กับตัวแปรทั้งสองเป็น 5 และ 3 ตามลำดับ ในสี่ตัวดำเนินการแรกเป็นการดำเนินการทางคณิตศาสตร์พื้นฐาน สำหรับตัวดำเนินการ // เป็นการหารเช่นเดียวกัน แต่ผลลัพธ์ของการหารนั้นจะตัดส่วนที่เป็นทศนิยมทิ้งไป ส่วนตัวดำเนินการ % นั้นเป็นการหารโดยผลลัพธ์จะเป็นเศษของการหารแทน ส่วนสุดท้าย ** นั้นแทนการยกกำลัง

```

Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46)
1)] on win32
Type "copyright", "credits" or "license()" for more in
>>>
===== RESTART: D:\Pyhonfile\test1.py
a + b = 8
a - b = 2
a * b = 15
a / b = 1.6666666666666667
a // b = 1
a % b = 2
a ** b = 125
>>>

```


ตัวดำเนินการเปรียบเทียบ Comparison operators

ตัวดำเนินการเปรียบเทียบ (Comparison operators) คือตัวดำเนินการที่ใช้สำหรับเปรียบเทียบค่าหรือค่าในตัวแปร ซึ่งผลลัพธ์ของการเปรียบเทียบนั้นจะเป็น True หากเงื่อนไขเป็นจริง และเป็น False หากเงื่อนไขไม่เป็นจริง ตัวดำเนินการเปรียบเทียบมักจะใช้กับคำสั่งตรวจสอบเงื่อนไข if และคำสั่งวนซ้ำ for while เพื่อควบคุมการทำงานของโปรแกรม

Operator	Name	Example
<	Less than	a < b
<=	Less than or equal	a <= b
>	Greater than	a > b
>=	Greater than or equal	a >= b
==	Equal	a == b
!=	Not equal	a != b
is	Object identity	a is b
is not	Negated object identity	a is not b

แสดงให้เห็นถึงตัวดำเนินการเปรียบเทียบประเภทต่างๆ เช่น การเปรียบเทียบความเท่ากัน โดยคุณสามารถใช้ตัวดำเนินการเปรียบเทียบเพื่อเปรียบเทียบว่าค่าในตัวแปรนั้นเท่ากันหรือไม่ หรือการเปรียบเทียบค่ามากกว่าหรือน้อยกว่า ต่อไปมาดูตัวอย่างการใช้งานตัวดำเนินการเปรียบเทียบในภาษา Python

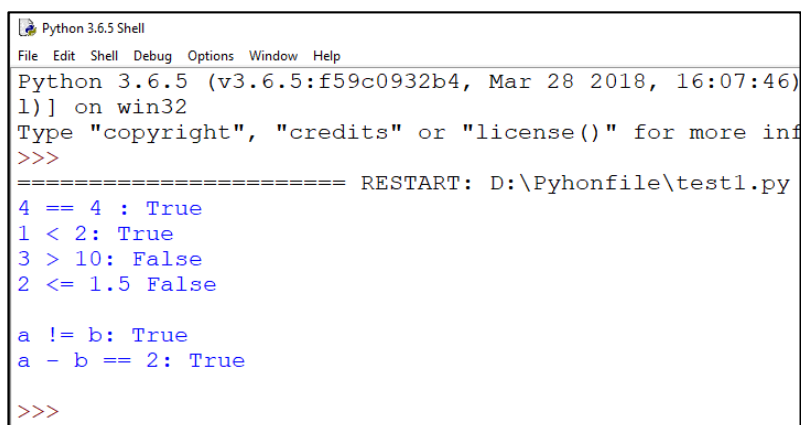
```

test1.py - D:\Pyhonfile\test1.py (3.6.5)
File Edit Format Run Options Window Help
# Constant comparison
print('4 == 4 :', 4 == 4)
print('1 < 2:', 1 < 2)
print('3 > 10:', 3 > 10)
print('2 <= 1.5', 2 <= 1.5)
print()

# Variable comparison
a = 10
b = 8
print('a != b:', a != b)
print('a - b == 2:', a - b == 2)
print()

```

ในตัวอย่าง เป็นการเปรียบเทียบค่าประเภทต่างๆ ในคำสั่งกลุ่มแรกนั้นเป็นการใช้ตัวดำเนินการเปรียบเทียบค่าคงที่ ในกลุ่มที่สองเป็นการใช้งานกับตัวแปร ซึ่งถ้าหากเงื่อนไขเป็นจริงจะได้ผลลัพธ์เป็น True และถ้าหากไม่จริงจะได้ผลลัพธ์เป็น False



```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46)
1)] on win32
Type "copyright", "credits" or "license()" for more inf
>>>
===== RESTART: D:\Pyhonfile\test1.py
4 == 4 : True
1 < 2: True
3 > 10: False
2 <= 1.5 False

a != b: True
a - b == 2: True

>>>
```

ตัวดำเนินการตรรกศาสตร์ Logical operators

ตัวดำเนินการตรรกศาสตร์ (Logical operators) คือตัวดำเนินการที่ใช้สำหรับประเมินค่าทางตรรกศาสตร์ ซึ่งเป็นค่าที่มีเพียงจริง (True) และเท็จ (False) เท่านั้น โดยทั่วไปแล้วเรามักใช้ตัวดำเนินการตรรกศาสตร์ในการเชื่อม Boolean expression ตั้งแต่หนึ่ง expression ขึ้นไปและผลลัพธ์สุดท้ายที่ได้นั้นจะเป็น Boolean

ตารางของตัวดำเนินการตรรกศาสตร์ในภาษา Python

Operator	Example	Result
and	a and b	True if a and b are true, else False
or	a or b	True if a or b are true, else False
not	not a	True if a is False, else True

ในภาษา Python นั้นมีตัวดำเนินการทางตรรกศาสตร์ 3 ชนิด คือ ตัวดำเนินการ and เป็นตัวดำเนินการที่ใช้เชื่อมสอง Expression และได้ผลลัพธ์เป็น True หาก Expression ทั้งสองเป็น True ไม่เช่นนั้นจะได้ผลลัพธ์เป็น False ตัวดำเนินการ or เป็นตัวดำเนินการที่ใช้เชื่อมสอง Expression และได้ผลลัพธ์เป็น True หากมีอย่างน้อยหนึ่ง Expression ที่เป็น True ไม่เช่นนั้นได้ผลลัพธ์เป็น False และตัวดำเนินการ not ใช้ในการกลับค่าจาก True เป็น False และในทางกลับกัน มาดูตัวอย่างการใช้งาน

```

test1.py - D:\Pyhonfile\test1.py (3.6.5)
File Edit Format Run Options Window Help
print('Log in page')
username = input('Username: ')
password = input('Password: ')

if (username == 'admin' and password == '3456'):
    print('Welcome')
else:
    print('Login incorrect')

```

ในตัวอย่าง เราได้สร้างโปรแกรมจำลองในการเข้าสู่ระบบของเว็บไซต์ ในการที่จะเข้าสู่ระบบ ผู้ใช้ต้องกรอกชื่อผู้ใช้และรหัสผ่านให้ถูกต้อง ดังนั้นเราจึงใช้ตัวดำเนินการ and เพื่อตรวจสอบว่าทั้งชื่อผู้ใช้และรหัสผ่านนั้นถูกต้อง ทำให้เงื่อนไขเป็นจริงและในบล็อกคำสั่ง if จะทำงาน

```

Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) on win32
Type "copyright", "credits" or "license()" for more in
>>>
===== RESTART: D:\Pyhonfile\test1.py
Log in page
Username: admin
Password: 3456
Welcome
>>> |

```

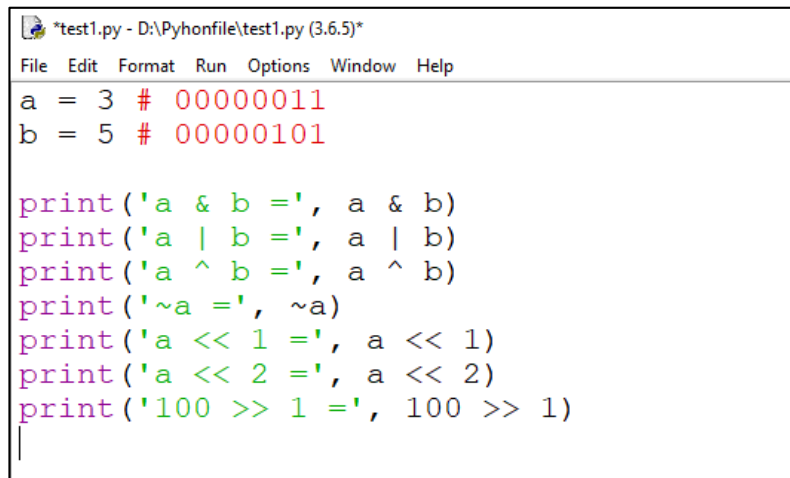
ผลลัพธ์การทำงานของโปรแกรม เมื่อเรากรอกชื่อผู้ใช้เป็น "admin" และรหัสผ่านเป็น "3456" ซึ่งถูกต้องทั้งสองอย่างทำให้สามารถเข้าสู่ระบบได้สำเร็จ

ตัวดำเนินการระดับบิต Bitwise operators

ตัวดำเนินการระดับบิต (Bitwise operators) เป็นตัวดำเนินการที่ทำงานในระดับบิตของข้อมูล หรือจัดการข้อมูลในระบบเลขฐานสอง โดยทั่วไปแล้วตัวดำเนินการระดับบิตมักจะใช้กับการเขียนโปรแกรมระดับต่ำ เช่น การเขียนโปรแกรมเพื่อควบคุมฮาร์ดแวร์ อย่างไรก็ตาม ในภาษา Python นั้นสนับสนุนตัวดำเนินการเพื่อให้เราสามารถจัดการกับบิตของข้อมูลโดยตรงได้

Operator	Name	Result
&	Bitwise and	a & b
	Bitwise or	a b
^	Bitwise exclusive or	a ^ b
<<	Bitwise shifted left	a << b
>>	Bitwise shifted right	a >> b
~	Bitwise invert	~a

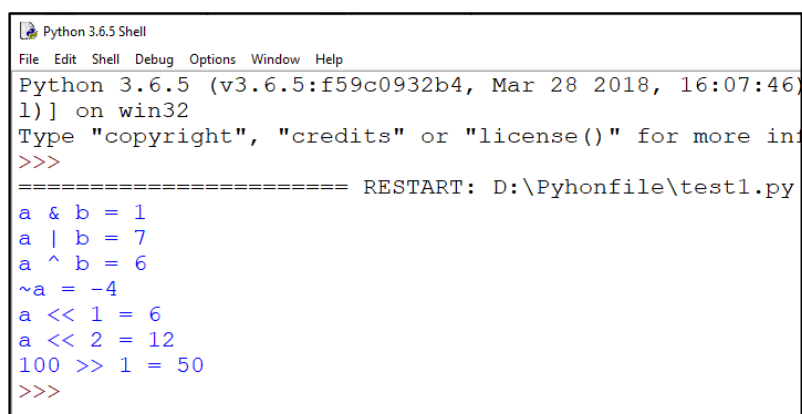
ตัวดำเนินการระดับบิตใช้จัดการกับบิตของข้อมูลที่เป็นตัวเลข โดยปกติแล้วเมื่อเรากำหนดค่าให้กับตัวแปรนั้น คอมพิวเตอร์จะเก็บค่าเหล่านี้ในหน่วยความจำในรูปแบบของตัวเลขฐานสอง (binary form) ซึ่งประกอบไปด้วยเพียง 1 และ 0 เท่านั้น ดังนั้นเราใช้ตัวดำเนินการเหล่านี้ในการจัดการกับข้อมูลได้โดยตรง มาดูตัวอย่าง



```
*test1.py - D:\Pyhonfile\test1.py (3.6.5)*
File Edit Format Run Options Window Help
a = 3 # 00000011
b = 5 # 00000101

print('a & b =', a & b)
print('a | b =', a | b)
print('a ^ b =', a ^ b)
print('~a =', ~a)
print('a << 1 =', a << 1)
print('a << 2 =', a << 2)
print('100 >> 1 =', 100 >> 1)
```

ในตัวอย่าง เป็นการใช้ตัวดำเนินการระดับบิตประเภทต่างๆ ในภาษา Python เรามีตัวแปร a และตัวแปร b และกำหนดค่า 3 และ 5 ให้กับตัวแปรตามลำดับ เราได้คอมเมนต์ค่าในฐานสองไว้ด้วย ในการทำงานนั้นโปรแกรมจะทำงานทีละคู่ของบิต



```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46)
1)] on win32
Type "copyright", "credits" or "license()" for more in
>>>
===== RESTART: D:\Pyhonfile\test1.py
a & b = 1
a | b = 7
a ^ b = 6
~a = -4
a << 1 = 6
a << 2 = 12
100 >> 1 = 50
>>>
```

ตัวดำเนินการแบบลำดับ Sequence Operators

ในภาษา Python มีตัวดำเนินการในการตรวจสอบการเป็นสมาชิกในออบเจกต์ประเภท List Tuple และ Dictionary ตัวดำเนินการ in ใช้ในการตรวจสอบถ้าหากค่านั้นมีอยู่ในออบเจกต์ ถ้าหากพบจะได้ผลลัพธ์เป็น True และหากไม่พบจะได้ผลลัพธ์เป็น False และตัวดำเนินการ not in นั้นจะทำงานตรงกันข้าม หากไม่พบจะได้ผลลัพธ์เป็น True แทน

ตารางของตัวดำเนินการในการตรวจสอบการเป็นสมาชิกในออบเจกต์ ในภาษา Python

Operator	Name	Example
in	Object memberships	a in b
not in	Negated object memberships	a not in b

ตัวอย่างการใช้งานของตัวดำเนินการเหล่านี้ เราจะใช้ในการตรวจสอบการมีอยู่ของข้อมูลใน List และ Dictionary

```

test1.py - D:\Pythonfile\test1.py (3.6.5)
File Edit Format Run Options Window Help
names = ['Mateo', 'David', 'Andrill', 'Joshep']

if 'Mateo' in names:
    print('\nMateo\ exist in the list')
else:
    print('\nMateo\ not exist in the list')

if 'Jonathan' in names:
    print('\nJonathan\ exist in the list')
else:
    print('\nJonathan\ not exist in the list')

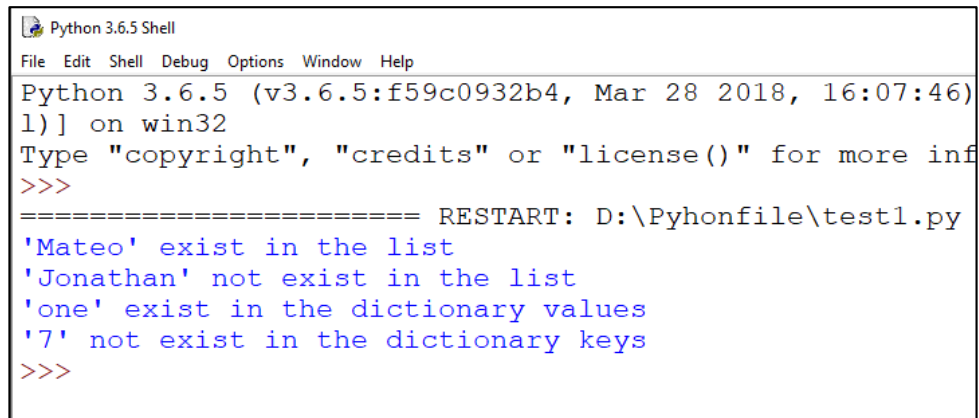
numbers = {'1': 'one', '3': 'three', '2': 'two', '5': 'five'}

if 'one' in numbers.values():
    print('\n'one\ exist in the dictionary values')
else:
    print('\n'one\ not exist in the dictionary values')

if '7' in numbers.keys():
    print('\n'7\ exist in the dictionary keys')
else:
    print('\n'7\ not exist in the dictionary keys')

```

ในตัวอย่าง เป็นการตรวจสอบข้อมูลใน List และ Dictionary ในโปรแกรมของเรามีตัวแปร List names ซึ่งมีรายชื่ออยู่ภายใน เราใช้คำสั่ง if เพื่อตรวจสอบว่า "Mateo" นั้นมีอยู่ใน List หรือไม่ ผลลัพธ์ที่ได้ นั้นจะเป็นจริงเพราะชื่อมีอยู่ และต่อมาตรวจสอบ "Jonathan" นั้นไม่พบชื่อดังกล่าวใน List ต่อมาเป็นการตรวจสอบการมีอยู่ของข้อมูลใน Dictionary เนื่องจาก Dictionary นั้นเป็นข้อมูลที่เก็บในคู่ของ Key และ Values เพื่อตรวจสอบกับ Key เราต้องใช้เมธอด keys() และเมธอด values() สำหรับ Value



```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46)
[1] on win32
Type "copyright", "credits" or "license()" for more info
>>>
==== RESTART: D:\Pyhonfile\test1.py
'Mateo' exist in the list
'Jonathan' not exist in the list
'one' exist in the dictionary values
'7' not exist in the dictionary keys
>>>
```

การตรวจสอบค่าความจริง Truth Value Testing

เนื่องจากตัวแปรในภาษา Python นั้นเป็นประเภทข้อมูลแบบไดนามิกส์ ดังนั้นออบเจ็กต์ต่างๆ นั้นสามารถที่จะทำมาประเมินสำหรับค่าความจริง โดยใช้คำสั่งตรวจสอบเงื่อนไขเช่น if หรือ while หรือการกระทำเพื่อตรวจหาค่า Boolean โดยค่าข้างล่างนี้เป็นค่าที่ถูกประเมินเป็น False

-None

-False

-ค่าศูนย์ของข้อมูลประเภทตัวเลขใดๆ เช่น 0, 0L, 0.0, 0j

-ข้อมูลแบบลำดับที่ว่างเปล่า เช่น "", [], {}

-ข้อมูลแบบ map ที่ว่างเปล่า {}

ตัวแปรจากคลาสที่ผู้สร้างขึ้น และคลาสดังกล่าวถูกกำหนดเมธอด `__nonzero__()` หรือ `__len__()` และเมธอดเหล่านี้ส่งค่ากลับเป็นศูนย์หรือค่า Boolean False

ส่วนค่าอื่นๆ ที่นอกเหนือจากที่ได้กล่าวไปนั้นจะถูกประเมินเป็น True ทั้งหมด และออบเจ็กต์ของประเภทใดๆ ก็เป็น True เช่นกัน

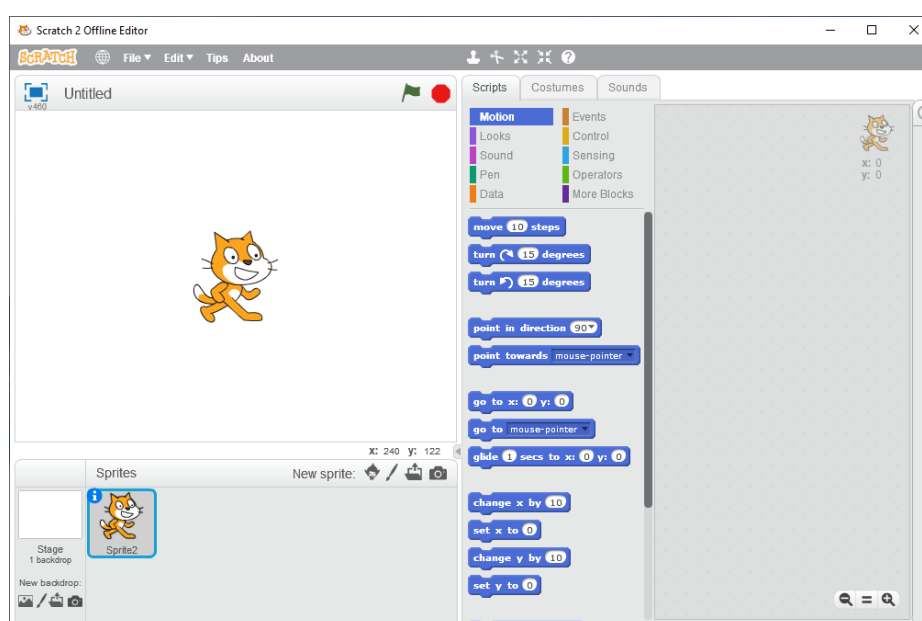
ความรู้เบื้องต้นเบื้องต้นเกี่ยวกับ Scratch

โปรแกรม Scratch เป็นโปรแกรมภาษาคอมพิวเตอร์ภาษาหนึ่งที่พัฒนาขึ้นมาเพื่อให้ นักเรียนได้เรียนรู้ การเขียนโปรแกรมคอมพิวเตอร์ ในรูปแบบของกราฟิก ลากแล้ววาง อาจมีการกำหนดค่าบ้างเล็กน้อย นักเรียนสามารถเรียนรู้ได้ง่าย สนุกสนานกับการเรียนรู้ ฝึกกระบวนการคิดอย่างเป็นขั้นเป็นตอนอย่างสร้างสรรค์ สามารถสร้างชิ้นงานได้หลากหลายตามความต้องการหรือตามจินตนาการ เช่น การสร้างการเคลื่อนไหว สร้างเป็นเรื่องราว สร้างเกม หรือด้านดนตรีตลอดจนงานศิลปะ ฯลฯ

เหมาะสำหรับการเริ่มต้นในการหัดเขียนโปรแกรมเป็นอย่างยิ่ง เมื่อเด็กเกิดความคิดสร้างสรรค์ขึ้นมาสักอย่าง ก็มีหลายวิธีที่จะแปรเปลี่ยนความคิดให้เป็นผลงานที่จับต้องได้ และสะท้อนความคิดของเขาได้ ในยุคปัจจุบัน คอมพิวเตอร์เป็นอุปกรณ์ที่ใช้กันแพร่หลายแล้ว และเป็นเครื่องมือชิ้นหนึ่งที่ใช้สอนในโรงเรียน โปรแกรมสำเร็จรูปสามารถสร้างผลงานอะไรได้มากมาย แต่ไม่ยืดหยุ่นพอที่จะให้สะท้อนความคิดความต้องการ ของแต่ละคนได้

ส่วนประกอบหลักของโปรแกรม

หน้าต่างการทำงานของโปรแกรม Scratch มีส่วนประกอบหลักดังนี้



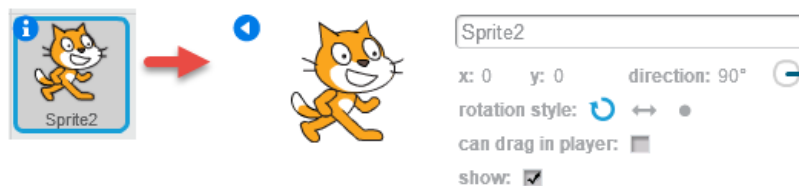
1. แถบเมนูเครื่องมือ (Toolbar)

	ปุ่มเปลี่ยนภาษา
File ▼	เมนู เปิด-บันทึกโปรเจกต์ บันทึกวิดีโอ แชร์เว็บไซต์ เช็คอัปเดต และปิดโปรแกรม
Edit ▼	เมนู แก้ไขการตั้งค่า
Tips	เมนู แนะนำโปรแกรมตัวอย่าง
About	เมนู เว็บไซต์ http://scratch.mit.edu/

2. เครื่องมือเวที (Stage Toolbar)

	ปุ่มประทับตราตัวละคร
	ปุ่มลบตัวละคร
	ปุ่มเพิ่มขนาดตัวละคร
	ปุ่มลดขนาดตัวละคร
	ปุ่มช่วยเหลือให้คำแนะนำ

3. ข้อมูลของตัวละครที่ถูกเลือก



4. กลุ่มบล็อก (Block Palette)

5. บล็อกในกลุ่มที่เลือก

6. พื้นที่ทำงาน (Script Area)

7. เวที (Stage)

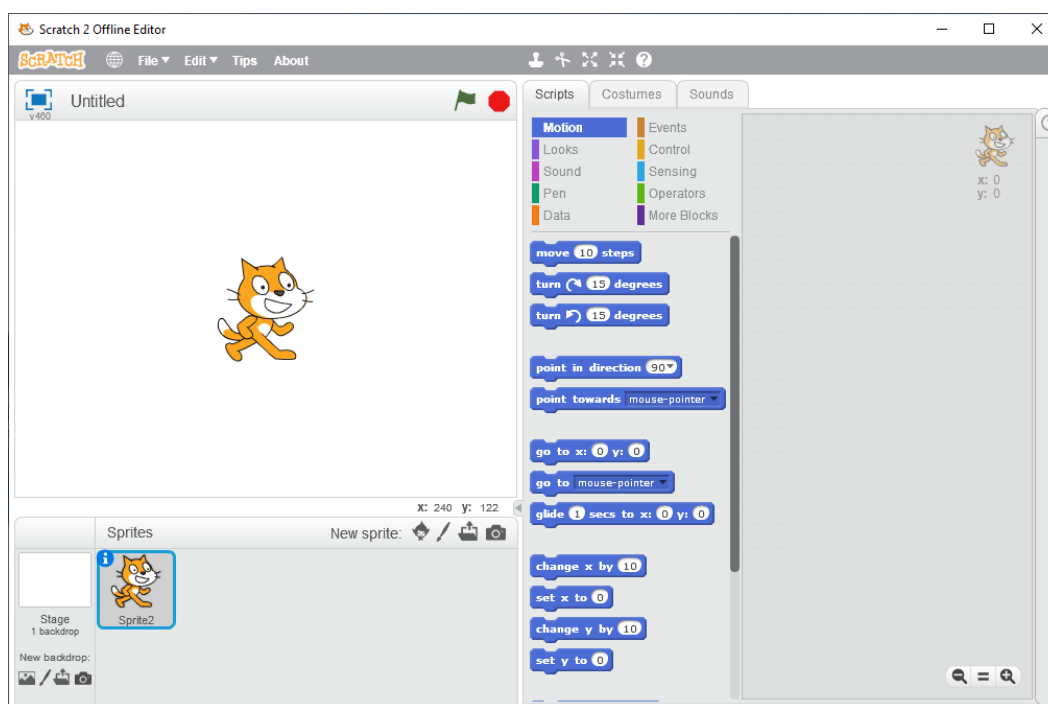
8. รายการตัวละคร และเวทีที่ใช้ในโปรเจกต์ปัจจุบัน (Sprites Pane)

9. แถบเมนูแสดงข้อมูลสคริปต์ costumes และเสียงของตัวละครหรือเวที

10. พื้นที่การแสดงผลของการทำงานของโปรแกรมที่มีขนาดที่เปลี่ยนแปลง

รู้จักโปรเจกต์

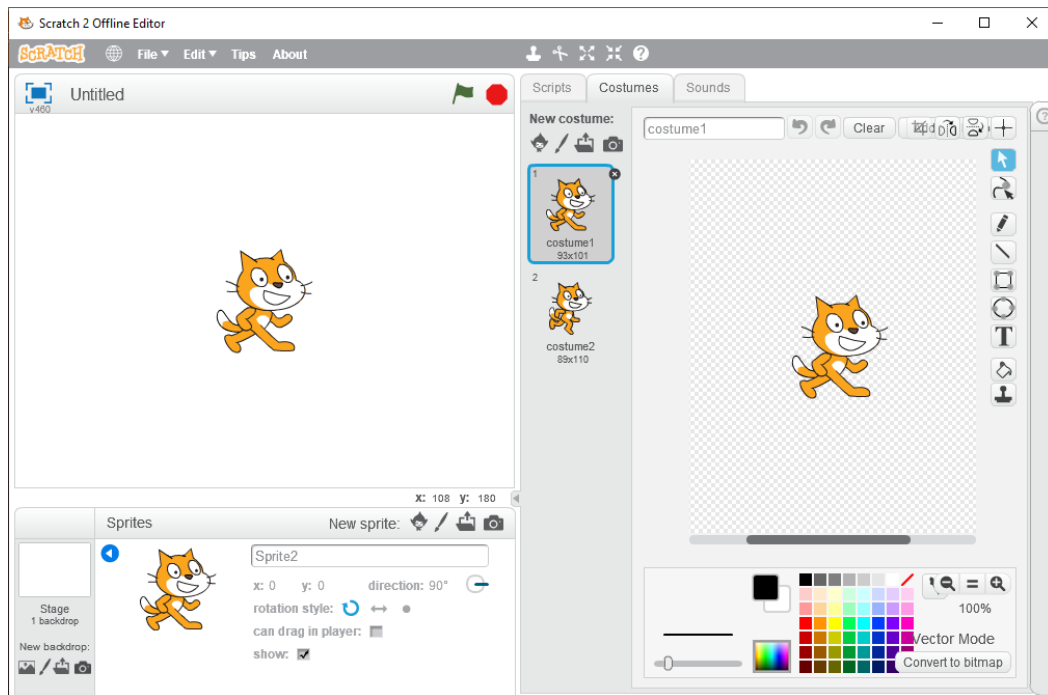
โปรเจกต์ใน Scratch มีโครงสร้าง 3 ส่วนประกอบด้วย



1. เวที (Stage)
2. ตัวละคร (Sprite)
3. สคริปต์ (Script)

เวที

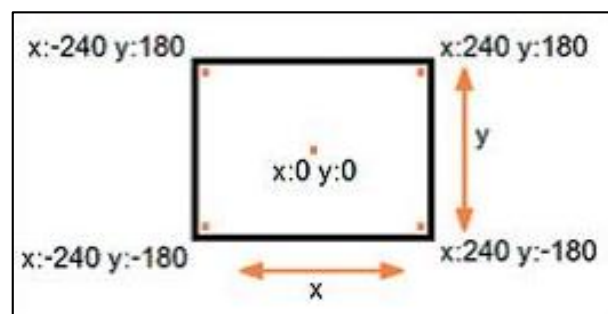
เวทีมีขนาดกว้าง 480 หน่วย สูง 360 หน่วย ในแต่ละโปรเจกต์มีเวทีเดียว จึงมีชื่อเดียวและ ไม่สามารถเปลี่ยนชื่อได้ เวทีใช้แสดงผลการทำงานของสคริปต์ (script) เสียง (sound) หรือฉาก (background) ได้ และฉากที่จะแสดงบนเวทีจะต้องมีขนาดไม่เกินขนาดของเวที (480 X 360) ถ้าพื้นหลังที่ใช้มีขนาดใหญ่กว่าโปรแกรม Scratch จะลดขนาดพื้นหลังนั้นอัตโนมัติเพื่อให้พอดีกับขนาดของเวที

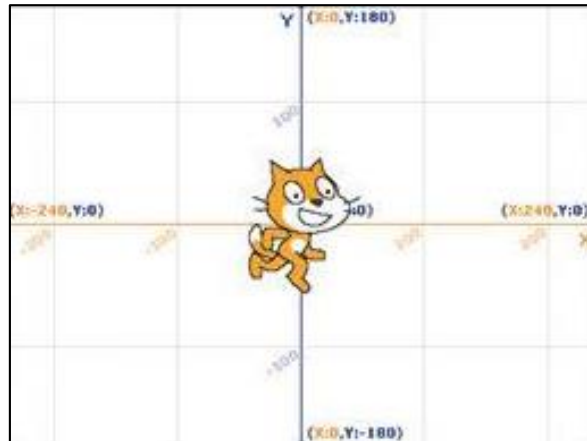


รายละเอียดของเวที

1. คลิก Stage เพื่อดูรายละเอียดของเวที
2. แท็บ Scripts ของเวที
3. แท็บ Backdrops
4. แท็บ Sounds
5. เพิ่มฉากใหม่ New Backdrop
6. ฉากทั้งหมดที่มีอยู่บนเวที

การบอกตำแหน่งใดๆ บนเวทีจะบอกโดยใช้ค่า (x, y) โดยค่า x และ y ที่ตำแหน่ง (0, 0) จะอยู่ตรงกลาง



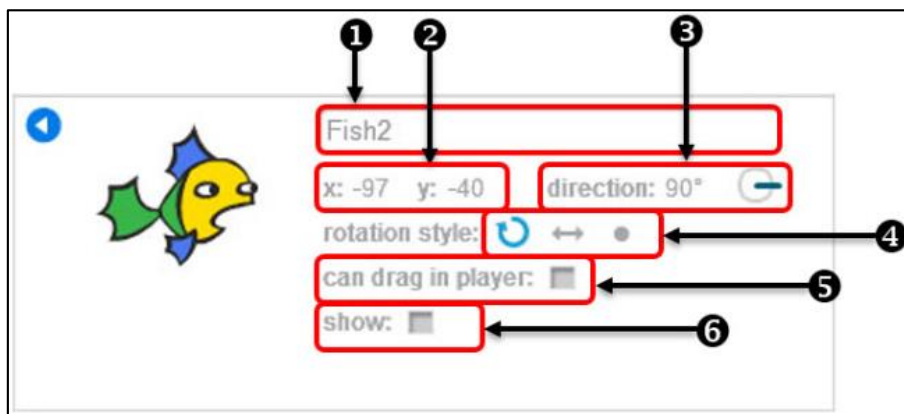


ตัวละคร

ตัวละครแต่ละตัวจะมีข้อมูลแตกต่างกัน โดยสามารถคลิก ที่ตัวละคร เพื่อดูข้อมูลของตัวละครนั้น เช่น ตัวละครปลา มีข้อมูลดังตาราง

1) ชื่อตัวละคร

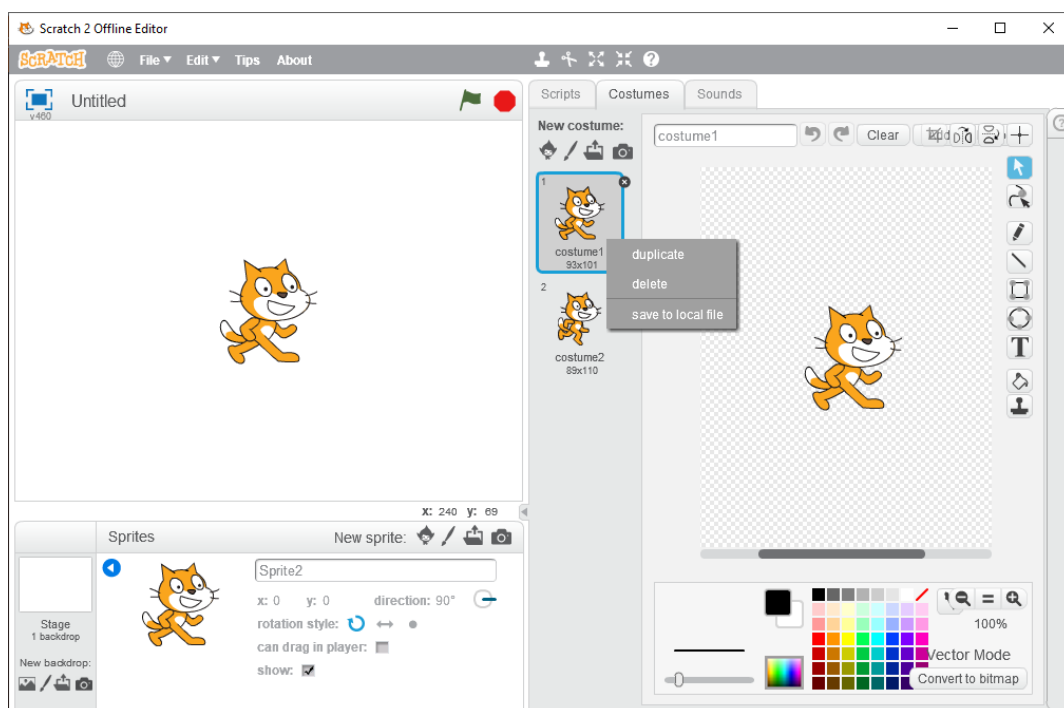
โปรแกรมจะตั้งชื่อตัวละครให้เป็น Sprite1, Sprite2, Sprite3... ตามลำดับที่สร้างขึ้น โดยอัตโนมัติถ้าต้องการเปลี่ยนชื่อตัวละครให้พิมพ์ชื่อใหม่บนแถบชื่อหมายเลข 1 ตามภาพ



หมายเลข	ข้อมูล	รายละเอียดข้อมูลตัวละคร
①	ชื่อตัวละคร	Fish2
②	ตำแหน่งบนเวที	x: -7 y: 24
③	ทิศทางการเคลื่อนที่ (direction)	90 องศา
④	รูปแบบการหมุน มี 3 ลักษณะ  หมุนได้รอบทิศทาง  หันได้เฉพาะซ้ายหรือขวา  ห้ามหมุน	 หมุนได้รอบทิศทาง
⑤	can drag in player: ☒ การลากตัวละครในโหมดนำเสนอ หรือบนเว็บไซต์	can drag in player: ☐ ลากตัวละครไม่ได้
⑥	show: ☒ การแสดงตัวละครบนเวทีในขณะที่ย่อแบบ	show: ☐ ตัวละครบนเวทีไม่ปรากฏใน ขณะที่ย่อแบบ

2) ชุดตัวละคร

ชุดตัวละคร (Costumes) เป็นภาพของตัวละคร ซึ่งสามารถเปลี่ยนแปลง ภาพเดิม หรือเพิ่ม ภาพใหม่และอาจเขียนสคริปต์เพิ่มให้กับตัวละครเปลี่ยนชุด หรือให้มองเห็นเป็นการ เคลื่อนไหว ใน รูปแบบต่างๆ ตามต้องการ



รายละเอียดชุดตัวละคร

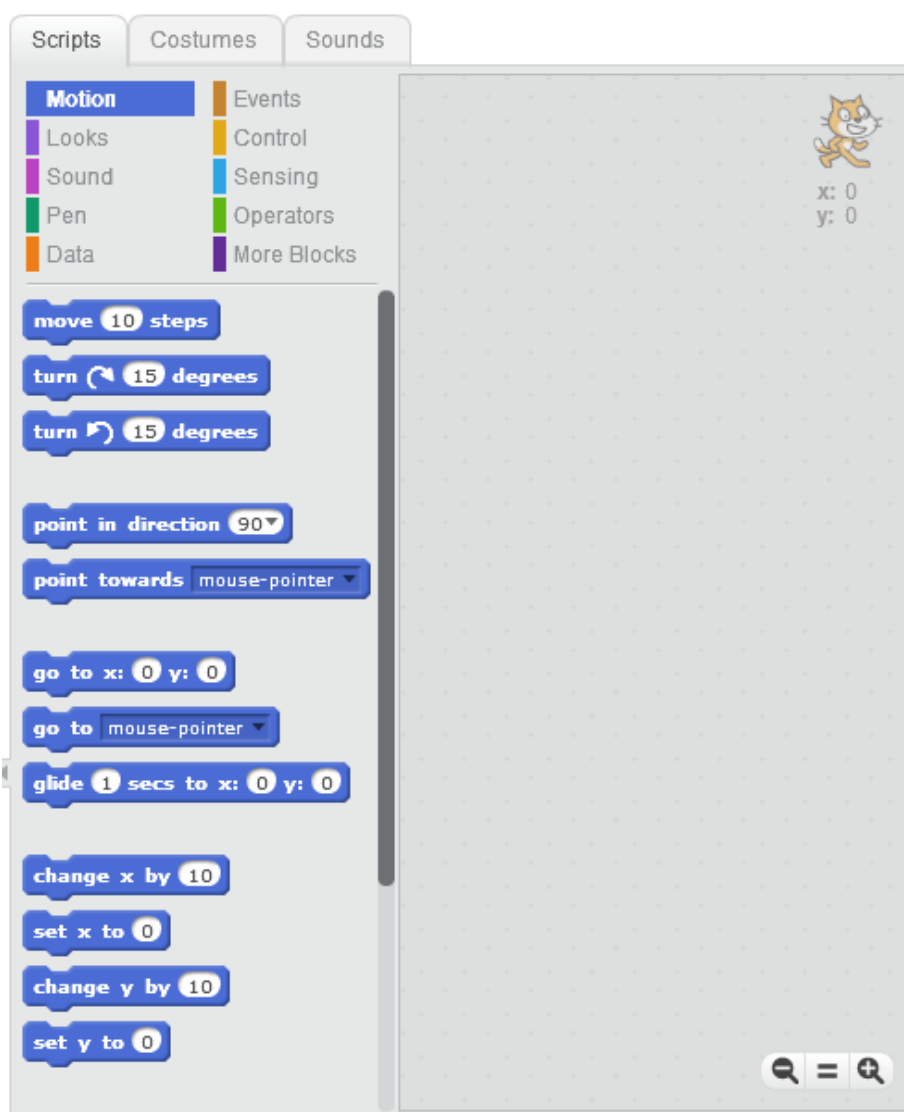
1. คลิกที่ตัวละคร Monkey2
2. คลิกแท็บ Costumes
3. คลิกขวาที่ชุดตัวละครที่ต้องการ จะปรากฏเมนูคัดลอก ลบ และบันทึกชุดตัวละคร
4. ตัวละครในตัวอย่างมีชื่อว่า monkey2 ประกอบด้วยชุดตัวละคร 2 ชุด
ชื่อชุดตัวละครที่ 1 ชื่อว่า monkey2-a มีลักษณะยกมือขึ้น 2 ข้าง
ชื่อชุดตัวละครที่ 2 ชื่อว่า monkey2-b มีลักษณะมือลงทั้ง 2 ข้าง
5. พื้นที่ออกแบบและแก้ไขชุดตัวละครที่เลือก

สคริปต์

สคริปต์คือชุดคำสั่งสำหรับตัวละครหรือเวที เพื่อสั่งให้ตัวละครหรือเวทีทำงานตามวัตถุประสงค์ที่ต้องการ โดยการเลือกสคริปต์จากกลุ่มบล็อก ซึ่งแบ่งเป็นกลุ่ม ดังนี้

กลุ่มบล็อก	การทำงาน
Motion	การเคลื่อนไหว เช่น เคลื่อนที่ไปข้างหน้า หมุนไปทางซ้ายหรือขวา
Events	เหตุการณ์ เช่น เมื่อคลิก
Looks	การแสดง เช่น พูด คิด เปลี่ยนขนาด
Control	การควบคุม เช่น การวนซ้ำ การตรวจสอบเงื่อนไข
Sound	การแสดงเสียง เช่น เล่นเสียงสัตว์ กลอง โน้ต
Sensing	การรับรู้ เช่น สัมผัสกับขอบ คลิกเมาส์ จับเวลา
Pen	ปากกา เช่น ยกปากกาขึ้น ตั้งค่าสีปากกา
Operators	ตัวดำเนินการ เช่น บวก และ หรือ
Data	ตัวแปร เช่น สร้างค่าตัวแปร
More Blocks	บล็อกอื่นๆ เช่น การสร้างฟังก์ชัน

เมื่อคลิกที่กลุ่มบล็อกใด จะปรากฏบล็อกในกลุ่มนั้น บล็อกสำหรับตัวละครและเวทีอาจมีความแตกต่างกันบ้างเช่น กลุ่มบล็อก Motion ของตัวละครจะมีบล็อกดังรูปด้านซ้าย ส่วนรูปด้านขวาเป็นของเวทีซึ่งไม่มีบล็อก Motion เนื่องจากเวทีเคลื่อนที่ไม่ได้



สคริปต์หนึ่งๆ ประกอบไปด้วยบล็อกมาเรียงต่อกันเป็นกลุ่ม บางบล็อกสามารถอยู่ข้างในหรือซ้อนอยู่บนบล็อกอื่นได้

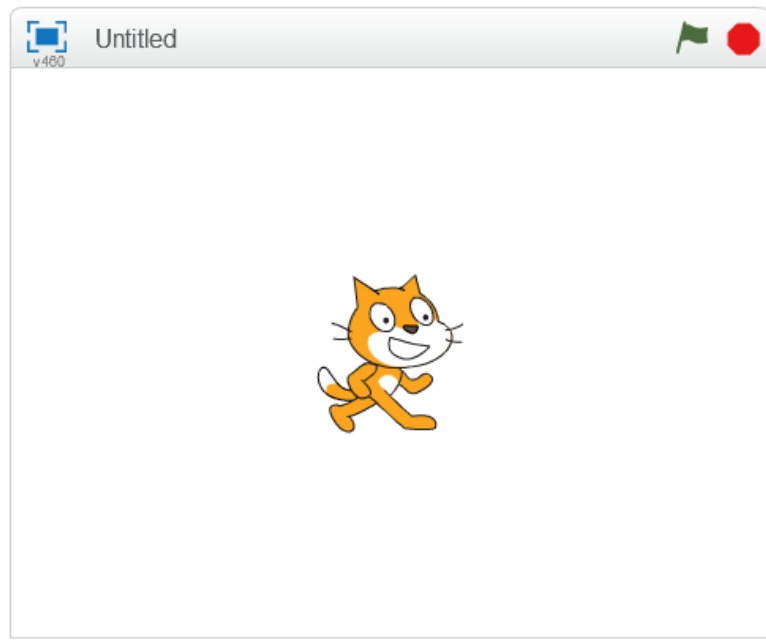
การสั่งให้โปรเจกต์เริ่มทำงานและหยุดทำงาน

การสั่งให้โปรเจกต์เริ่มทำงาน ทำได้โดยคลิก  ซึ่งอยู่มุมขวาบนของเวที โดยทุกสคริปต์ของ ทุกตัวละครและเวที ที่เริ่มต้นสคริปต์ด้วยบล็อก



จะเริ่มทำงานพร้อมกันและถ้าต้องการหยุด

การทำงานทั้งโปรเจกต์ ให้คลิก 



ภาษา Python กับ บอร์ด Raspberry Pi

Python เป็นภาษาที่ใช้สำหรับพัฒนาโปรแกรมคอมพิวเตอร์ภาษาหนึ่ง เช่นเดียวกับพวก C, Java, PHP โดย Python เป็นภาษาที่ทำงานได้กับทุกระบบ ไม่ว่าจะเป็น Windows Linux และ Mac OS ประกอบกับมีไลบรารี (Libraries) สนับสนุนงานด้านต่างๆ จำนวนมาก ทั้ง IOT, Data, Science, Machine Learning ดังนั้น Python ดังนั้น Python จึงเป็นภาษาที่ได้รับความนิยมอย่างสูง

การทำงานของ Python เป็นลักษณะตัวแปลภาษา (interpret) ไม่ได้คอมไพล์ (Compile) ดังนั้น การทำงานก็จะช้ากว่าภาษาอื่นที่ใช้วิธีการคอมไพล์อยู่บ้าง แต่ Python มีความได้เปรียบตรงที่เขียนโปรแกรมง่าย และมีไลบรารีสนับสนุนจำนวนมาก Python ถูกติดตั้งมาให้เรียบตอนที่ติดตั้งระบบปฏิบัติการ Raspbian ดังนั้นสามารถใช้งานได้เลย

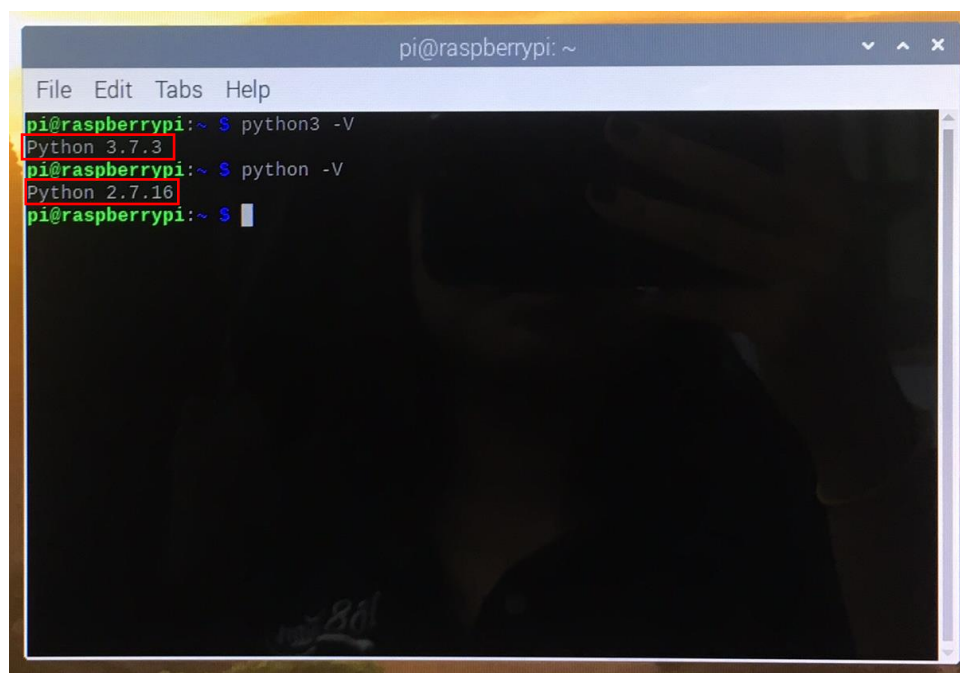
การตรวจสอบเวอร์ชัน Python

การตรวจสอบเวอร์ชัน Python ด้วย Common Line โดยไปที่ Terminal แล้วพิมพ์คำสั่ง

\$ python3 -V คือ Python3

\$ python2 -V คือ Python2

จะมีเลขเวอร์ชันบอกหลังจากที่เราพิมพ์คำสั่งไปดังรูป



```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ python3 -V
Python 3.7.3
pi@raspberrypi:~ $ python -V
Python 2.7.16
pi@raspberrypi:~ $
  
```

การรัน Python การรันคำสั่งและโปรแกรมสคริปต์ Python มีดังนี้

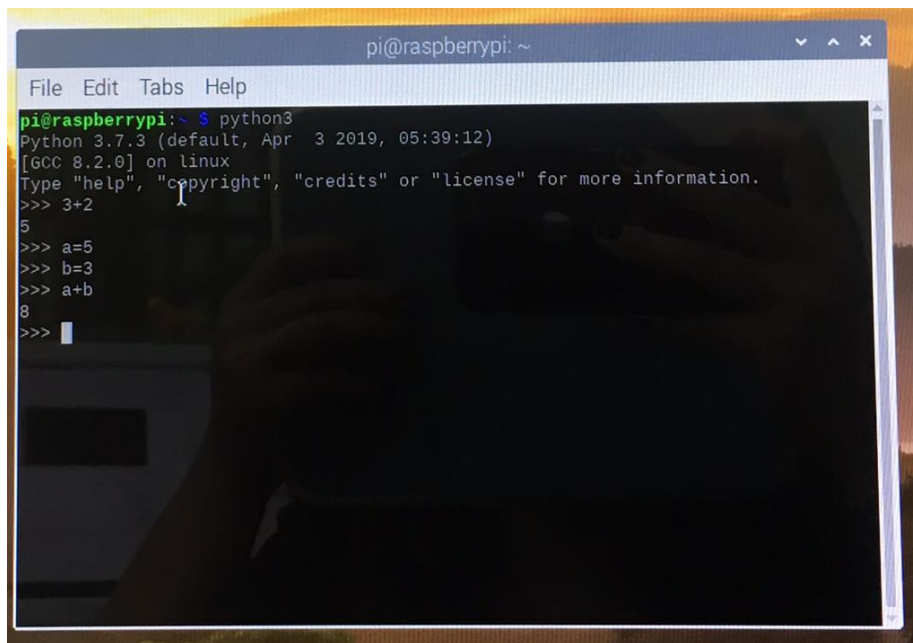
- รันทีละคำสั่งที่ Shell
- รันที่โปรแกรม IDE

- รันสคริปต์โปรแกรมที่ Terminal

การรันทีละคำสั่งที่ Shell

เป็นการรันคำสั่งทีละบรรทัด เหมาะสำหรับการทดลองคำสั่ง หรือ โปรแกรมที่ไม่ยาว วิธีการคือ เปิด Terminal แล้วเข้าสู่ Python Shell โดยพิมพ์คำสั่ง `$ python3` (`python3` คือเวอร์ชันของ Python)

จากนั้นหน้าบรรทัดจะมีเครื่องหมาย `>>>` แสดงว่าเข้าสู่ Python แล้ว สามารถพิมพ์คำสั่งแต่ละบรรทัดเข้าไปได้เลย โดยเมื่อกด Enter จะมีการประมวลผลที่



```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~$ python3
Python 3.7.3 (default, Apr  3 2019, 05:39:12)
[GCC 8.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> 3+2
5
>>> a=5
>>> b=3
>>> a+b
8
>>>

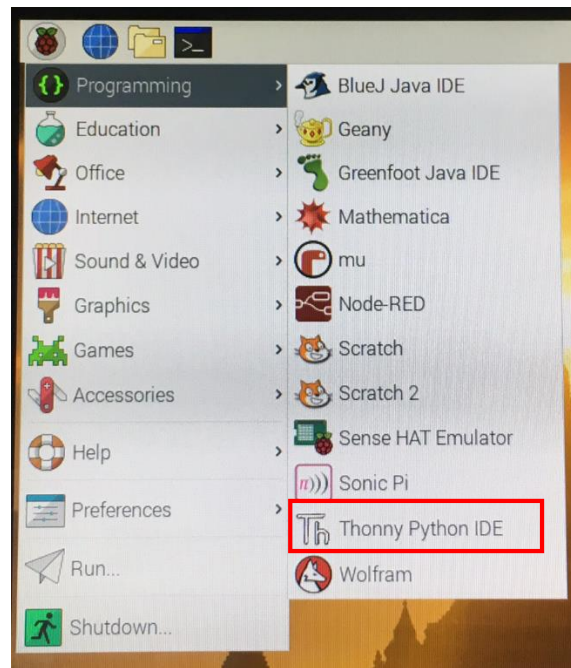
```

เมื่อต้องการออกจาก Python Shell ให้กดแป้น `Ctrl + d` หรือพิมพ์ `exit()` แล้วกด Enter

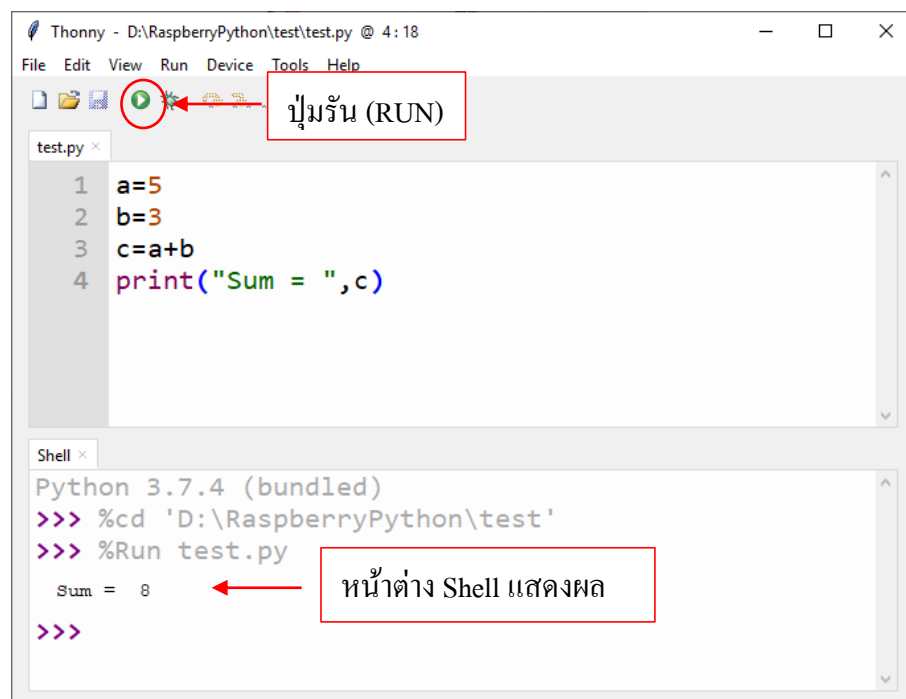
การรันกับโปรแกรม IDE

ในกรณีที่โปรแกรมมีหลายคำสั่ง ต้องมีการทดลอง แก้ไข หรือต้องการเรียกรันหลายๆ ครั้งก็ควรจะเขียนเก็บเป็นไฟล์หรือที่เรียกว่าสคริปต์ โดยเราสามารถใช่ IDE หรือ เอดิเตอร์อะไรเขียนก็ได้ โดย Raspbian ใน Raspberry Pi ได้ติดตั้งโปรแกรม IDE มาให้หลายตัว ได้แก่ IDLE Geany Thonny ซึ่งล้วนแล้วแต่สามารถใช้เขียนโปรแกรมและรัน Python ได้ทั้งสิ้น

วิธีการใช้งานคือ คลิกเปิดโปรแกรม IDE ดังกล่าว โดยสามารถเลือก Thonny Python IDE



เมื่อ IDE เปิดขึ้นมาแล้ว ให้พิมพ์คำสั่งเข้าไป เหมือนกับการเขียนโปรแกรมทั่วไป จากนั้นทำการบันทึก (นามสกุลไฟล์เป็น .py) ทำการรันโดยกดปุ่ม Play หรือ กด F5 ที่คีย์บอร์ด ผลการรันจะแสดงที่หน้าต่างเอาต์พุต



การรันไฟล์สคริปต์ที่ Terminal

หลักการคือเขียนโปรแกรมหรือสคริปต์ โดยใช้เอดิเตอร์ตัวใดก็ได้ซึ่งจะเป็นโปรแกรมที่อยู่บน Rasp Pi โดยใช้ NotePad EditPlus Sublime Visual Studio แล้วส่งไฟล์เข้ามาที่ Rasp Pi (ใช้ FTP หรือคัดลอกวาง) ก็ได้ จากนั้นสั่งรันโดยเรียกชื่อไฟล์ดังกล่าว เช่น ไฟล์สคริปต์ชื่อ test1.py

เปิดหน้าต่าง Terminal

```
$ python3 test1.py
```

หากสคริปต์โปรแกรมอยู่ใดเรกทอรีอื่น ก็ให้ cd เข้าไปก่อน หรือระบุ path เต็มเช่น

```
$ python3 /home/pi/python/test1.py
```

```
$ python3 project/test1.py
```

ตัวแปรและการ Output เบื้องต้น

ความง่ายของการใช้ตัวแปร Python คือ ไม่ต้องประกาศชนิดของตัวแปร สามารถเรียกใช้ได้เลย ส่วนการ output ก็คล้ายๆ กับภาษาอื่น เช่น

a = 5	กำหนดให้ a มีค่าเท่ากับ 5 (ชนิดเลขจำนวนเต็ม Integer)
num = 56.5	กำหนดให้ num มีค่าเท่ากับ 56.5 (ชนิดเลขทศนิยม Float)
name = "Sara"	กำหนดให้ name ที่ค่าเท่ากับ "Sara" (ชนิดข้อมูล String)
print (b)	พิมพ์ค่า b

เงื่อนไข (Condition)

เงื่อนไขการรันโปรแกรมใช้ if ตรวจสอบ

```
test.py x
1 score = 80
2 if score > 60:
3     print ("Pass")
```

โปรแกรมจะทำการปรี้นข้อมูล Pass

score > 60 จะเป็นเงื่อนไขในการรันชุดคำสั่ง

print("Pass") เป็นคำสั่งที่จะถูกเรียกใช้ หากเงื่อนไขที่กำหนดไว้เป็นจริง (ในกรณีนี้คือต้องมีคะแนนมากกว่า 60) โดยการระบุกลุ่มคำสั่งจะแบ่งแยกจากตัวเงื่อนไขโดยอาศัยการย่อหน้า เพื่อกำหนดขอบเขตของกลุ่มคำสั่ง

```
test.py x
1 score = 50
2 if score > 60:
3     print ("Pass")
4 else:
5     print ("Fail")
```

โปรแกรมจะทำการปรี้นข้อมูล Fail เนื่องจากคะแนนน้อยกว่าเงื่อนไขที่กำหนดคือ 60

score > 60 จะเป็นเงื่อนไขในการตัดสินใจรันชุดคำสั่ง

print("Pass") จะเป็นคำสั่งที่ถูกดำเนินการหากเงื่อนไขเป็นจริง บอกขอบเขตของกลุ่มคำสั่งที่ดำเนินการด้วยการย่อหน้า

print("Fail") จะเป็นคำสั่งที่ถูกดำเนินการหากเงื่อนไขเป็นเท็จ บอกขอบเขตของกลุ่มคำสั่งที่ดำเนินการด้วยการย่อหน้า

```
test.py x
1 score = 75
2 if score < 70 and score >= 60:
3     print ("Grade C")
4 elif score < 80 and score >= 70:
5     print ("Grade B")
6 elif score <= 100 and score >= 80:
7     print ("Grade A")
8 else:
9     print ("Fail")
```

โปรแกรมจะทำการแสดงค่า Grade B

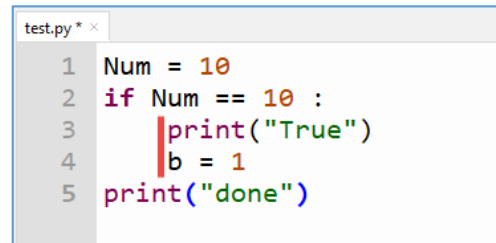
หากคะแนนอยู่ระหว่าง $60 \leq \text{คะแนน} < 70$ จะทำการพิมพ์ค่า Grade C

หากคะแนนอยู่ระหว่าง $70 \leq \text{คะแนน} < 80$ จะทำการพิมพ์ค่า Grade B

หากคะแนนอยู่ระหว่าง $80 \leq \text{คะแนน} \leq 100$ จะทำการพิมพ์ค่า Grade A

Indent

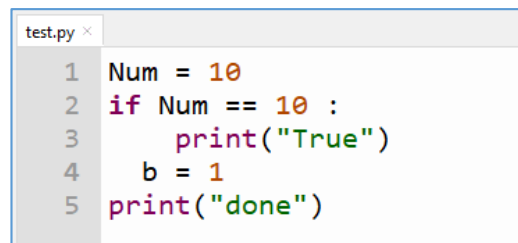
การแบ่งส่วนโค้ดเป็นบล็อกๆ ของภาษาอื่นๆ จะใช้เครื่องหมายครอบไว้ เช่น C, Java, PHP จะใช้ {...} แต่สำหรับ Python แตกต่างออกไป โดยใช้วิธีการคือ บล็อกจะเริ่มต้นด้วย : และบรรทัดต่อไปให้เว้นระยะต้นบรรทัด (ที่เรียกว่า Indent) 1Tab ให้เป็นแนวเดียวกันลงไปจนหมดบล็อก เช่น



```
test.py * x
1 Num = 10
2 if Num == 10 :
3     print("True")
4     b = 1
5 print("done")
```

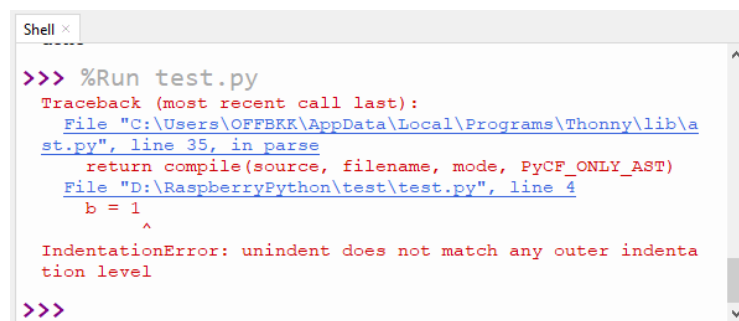
แสดงว่า บล็อกคำสั่งหลัง if คือ print("True") และ print("Num") ส่วน print("done") อยู่นอกบล็อก ถือว่าไม่เกี่ยวกับคำสั่ง if

แต่ถ้ามีการเว้นไม่ถูกต้อง เช่น



```
test.py x
1 Num = 10
2 if Num == 10 :
3     print("True")
4     b = 1
5 print("done")
```

จะเกิด Error ขึ้นว่า



```
Shell x
>>> %Run test.py
Traceback (most recent call last):
  File "C:\Users\OFFBKK\AppData\Local\Programs\Thonny\lib\ast.py", line 35, in parse
    return compile(source, filename, mode, PyCF_ONLY_AST)
  File "D:\RaspberryPython\test\test.py", line 4
    b = 1
    ^
IndentationError: unindent does not match any outer indentation level
>>>
```

การวนรอบ

การวนรอบ (Loop) ใช้ for และ while โดยบล็อกคำสั่งในการวนรอบจะเริ่มต้นด้วย : และบรรทัดต่อไปที่อยู่ในการวนรอบจะต้องมีการเว้นระยะต้นบรรทัด (Indent) ให้เป็นแนวเดียวกัน

ใช้ for โดยกำหนดให้ x มีลำดับใน range(3) แล้วแสดงลำดับออกมา ต่อมาใช้ else แสดงลำดับสุดท้ายออกมา นั่นคือ 2

```
test.py x
1 for x in range(3):
2     print (x)
3 else:
4     print ('Final x = %d' % (x))
```

ใช้ while โดยโปรแกรมจะแสดงตัวเลข 1 ถึง 10 โดยการใช้คำสั่ง while loop ในตอนแรก เราได้ประกาศตัวแปร i และกำหนดค่าให้กับตัวแปรเป็น 1 หลังจากนั้นสร้างเงื่อนไขสำหรับ while loop เป็น $i \leq 10$ นั้นหมายความว่าโปรแกรมจะทำงานในขณะที่ค่าในตัวแปร i ยังคงน้อยกว่าหรือเท่ากับ 10 และแสดงผลค่าของ i ในบล็อกของคำสั่ง while และเพิ่มค่าของตัวแปรขึ้นทุกครั้งหลังจากที่แสดงผลเสร็จ ถ้าไม่เพิ่มค่าของ i โปรแกรมจะทำงานไม่มีวันหยุดหรือเรียกว่า Infinite loop

```
test.py x
1 i = 1
2 while i <= 10:
3     print(i, end = ', ')
4     i = i + 1
```

การแสดงผลข้อมูลด้วย print และจัดรูปแบบ

การ output แสดงข้อมูลด้วย print เป็นคำสั่งหนึ่งถือว่าใช้งานบ่อยและมีความสำคัญ เนื่องจากมักใช้แสดงค่า แสดงผลการทำงาน หรือแสดงสถานะต่าง ๆ

ตัวอย่างนี้เป็นการ print ข้อมูลชนิด String ในกรณีที่มีข้อมูลแสดงต่อเนื่อง สามารถใช้ , คั่นระหว่างข้อมูล หรือใช้ .format() หรือใช้ %s ได้

```
test.py x 1.py x
1 name = "Sara"
2 print("Hi,",name)
3 print("Hi, {}".format(name))
4 print("Hi, %s" %name)
```

การ print ข้อมูลชนิดตัวเลข สามารถใช้ , คั่นระหว่างข้อมูลได้ เช่น

```
test.py x 1.py x
1 a = 5
2 b = 53.2
3 name = "Sara"
4 print("Hi,", name, "a =", a, ",", b)
```

หน่วยการเรียนรู้ที่ ๔ การเขียนโปรแกรมเบื้องต้นเพื่อควบคุม Input – Output ของขา GPIO

กลุ่มสาระการเรียนรู้วิทยาศาสตร์

สาระที่ ๔ เทคโนโลยี

มาตรฐาน ว ๔.๒ เข้าใจและใช้แนวคิดเชิงคำนวณในการแก้ปัญหาที่พบในชีวิตจริงอย่างเป็นขั้นตอนและเป็นระบบ ใช้เทคโนโลยีสารสนเทศและการสื่อสารในการเรียนรู้ การทำงาน และการแก้ปัญหาได้อย่างมีประสิทธิภาพ รู้เท่าทัน และมีจริยธรรม

ตัวชี้วัด

ว ๔.๒ ม.๑/๒ ออกแบบและเขียนโปรแกรมอย่างง่ายเพื่อแก้ปัญหาทางคณิตศาสตร์หรือวิทยาศาสตร์

ว ๔.๒ ม.๒/๒ ออกแบบและเขียนโปรแกรมที่ใช้ตรรกะและฟังก์ชันในการแก้ปัญหา

สาระสำคัญ

๑. เรียนรู้การต่อวงจรด้วยอุปกรณ์ Input / Output เข้ากับขา GPIO ของบอร์ด Raspberry Pi และเขียนโปรแกรมด้วยภาษา Python

๒. เรียนรู้การต่อวงจรด้วยอุปกรณ์ Input / Output เข้ากับขา GPIO ของบอร์ด Raspberry Pi และเขียนโปรแกรมด้วยภาษา Scratch

สาระการเรียนรู้

ความรู้

- ความรู้เกี่ยวกับขา GPIO ของบอร์ด Raspberry Pi
- ความรู้เกี่ยวกับอุปกรณ์ Input / Output
- ความรู้พื้นฐานในการต่อวงจรไฟฟ้า
- การเขียนโปรแกรมควบคุมขา GPIO ด้วยภาษา Python และ Scratch

ทักษะ/กระบวนการ

- ปฏิบัติตามตัวอย่างที่แสดงให้เห็นทีละขั้นตอนจากขั้นพื้นฐานไปสู่ขั้นที่ซับซ้อนขึ้น
- ฝึกปฏิบัติด้วยตนเองจนเกิดความชำนาญ หรือทำได้โดยอัตโนมัติ ซึ่งอาจเป็นงานชิ้นเดิมหรืองานที่คิดขึ้นใหม่

- ใช้กระบวนการคณิตศาสตร์ด้านการคิดคำนวณและการแก้ปัญหา
- กระบวนการกลุ่ม มุ่งให้ผู้เรียนทำงานร่วมกันเป็นทีม

คุณลักษณะที่พึงประสงค์

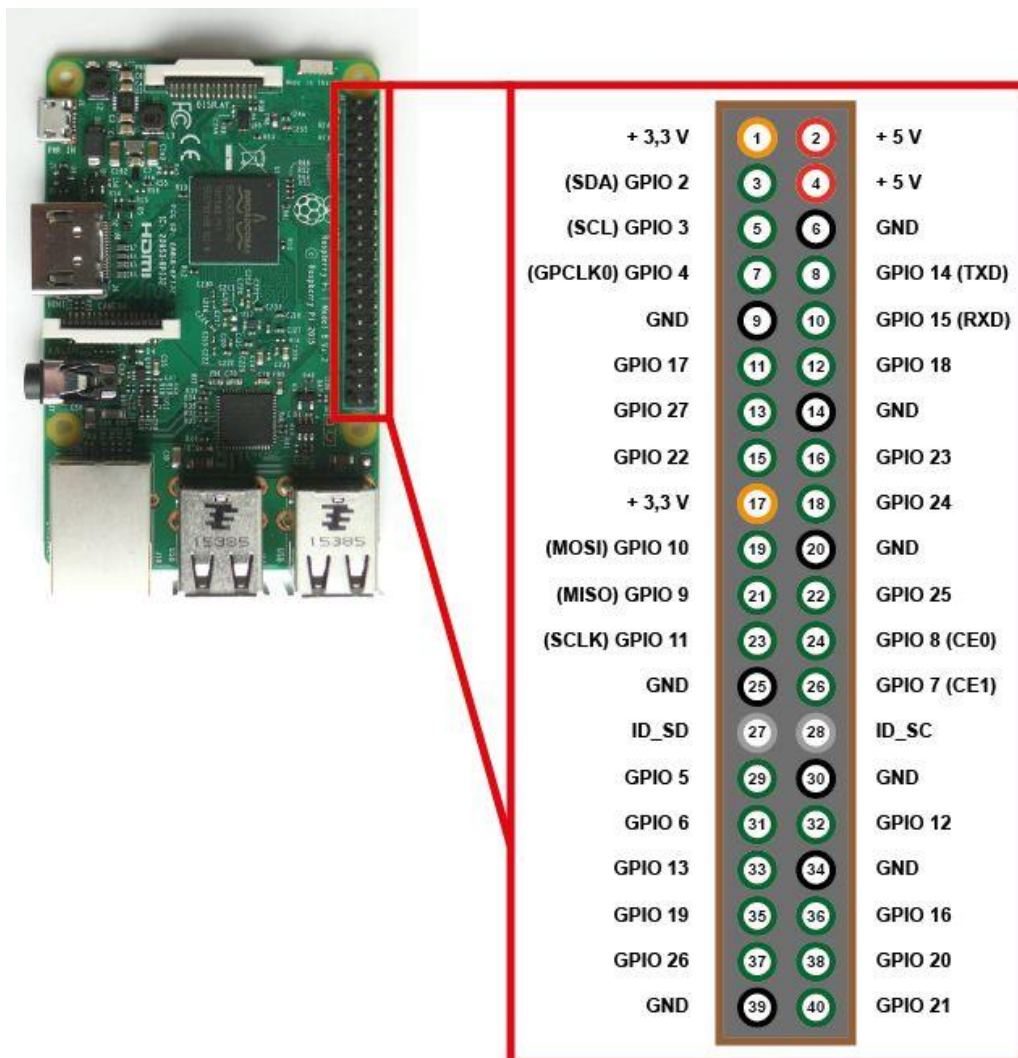
- รักษาติ ศาสน์ กษัตริย์
- ซื่อสัตย์
- มีวินัย
- ใฝ่เรียนรู้
- อยู่อย่างพอเพียง
- มุ่งมั่นในการทำงาน
- รักความเป็นไทย
- มีจิตสาธารณะ

หน่วยการเรียนรู้ที่ 4

การเขียนโปรแกรมเบื้องต้นเพื่อควบคุม Input – Output ของขา GPIO

การเขียนโปรแกรมสำหรับผู้เริ่มต้นใช้งานบอร์ด RPI พื้นฐานที่สำคัญ ก็คือ การฝึกหัดควบคุม Input – Output ของขา GPIO บนบอร์ด Raspberry Pi ซึ่งจะทำให้เราเรียนรู้และทราบถึงหน้าที่ของขา GPIO ในแต่ละขาบนบอร์ด RPI “มีหน้าที่อะไรบ้างและมีคุณสมบัติใดบ้าง” โดยขา GPIO บนบอร์ด Raspberry Pi จะมีทั้งหมด 40 ขา 20 คู่ ซึ่งจะมีหน้าที่และคุณสมบัติที่แตกต่างกันดังนี้

- 1 และ 17 คือ ขาแรงดันไฟฟ้า 3.3 V
- 2 และ 4 คือ ขาแรงดันไฟฟ้า 5 V
- 6, 9, 14, 20, 25, 30, 34, และ 39 คือขา Ground
- 3, 5, 7, 8, 10, 11, 12, 13, 15, 16, 18, 19, 21, 22, 23, 26, 27, 28, 29, 31, 32, 33, 35, 36, 37, 38 และ 40 คือขา GPIO เป็นหมายเลข และสมบัติอื่นๆ ได้ดังนี้



- ขา 3 คือ GPIO2 และเป็น (I Square C)
- ขา 5 คือ GPIO3 และเป็น (I Square C)
- ขา 7 คือ GPIO4
- ขา 8 คือ GPIO14 และเป็น (Serial Port)
- ขา 10 คือ GPIO15 และเป็น (Serial Port)
- ขา 11 คือ GPIO17
- ขา 13 คือ GPIO27
- ขา 15 คือ GPIO22
- ขา 16 คือ GPIO23
- ขา 18 คือ GPIO24
- ขา 19 คือ GPIO10 และเป็น (SPI)
- ขา 21 คือ GPIO9 และเป็น (SPI)
- ขา 22 คือ GPIO25
- ขา 23 คือ GPIO11 และเป็น (SPI)
- ขา 24 คือ GPIO8 และเป็น (SPI)
- ขา 26 คือ GPIO7 และเป็น (SPI)
- ขา 27 คือ GPIO, ID_SD (SDA.1 – SCL.1)
- ขา 28 คือ GPIO, ID_SD (SDA.0 – SCL.0)
- ขา 29 คือ GPIO5
- ขา 31 คือ GPIO6
- ขา 32 คือ GPIO12
- ขา 33 คือ GPIO13
- ขา 35 คือ GPIO19
- ขา 36 คือ GPIO16
- ขา 37 คือ GPIO26
- ขา 38 คือ GPIO20
- ขา 40 คือ GPIO21

สำหรับการเชื่อมต่อใช้งานขา GPIO จะใช้ Jumper Wires Cable “Female Cable” สวมเข้ากับขา GPIO เพื่อส่งและรับสัญญาณ Digital เช่นนั้น จะต้องมีการเขียนโปรแกรมเพื่อควบคุมขา GPIO เพื่อส่งและ

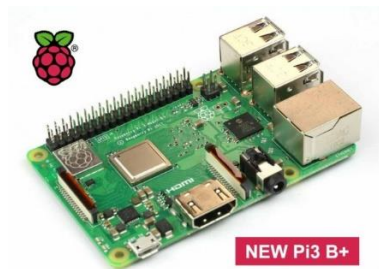
รับสัญญาณในคำสั่งต่าง ๆ โดยการเขียนโปรแกรมนั้น สำหรับผู้เริ่มเขียนโปรแกรม ควรใช้โปรแกรมที่ใช้งานง่ายไม่ซับซ้อน เพื่อฝึกหัดควบคุมขา GPIO บนบอร์ด RPI ซึ่งโปรแกรมการสร้างโปรแกรมควบคุมขา GPIO ก็คือโปรแกรม Python ที่มากับระบบปฏิบัติการ Raspbian

ตัวอย่างการเขียนโปรแกรมเบื้องต้นเพื่อควบคุม Input – Output ของขา GPIO ด้วยภาษา Python

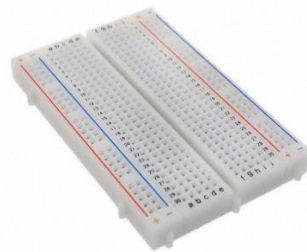
1. เขียนโปรแกรมควบคุมไฟกระพริบ

อุปกรณ์

1. บอร์ด Raspberry Pi



2. บอร์ดทดลอง (Breadboard)



3. ไดโอดเปล่งแสง (Diode LED)



4. ตัวต้านทาน (Resistor) 330 โอห์ม

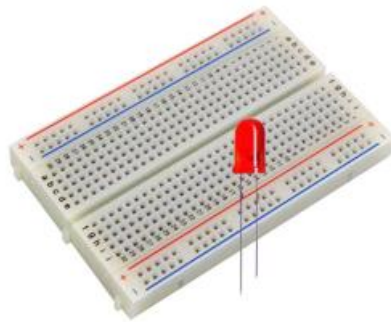


5. สาย Jumper

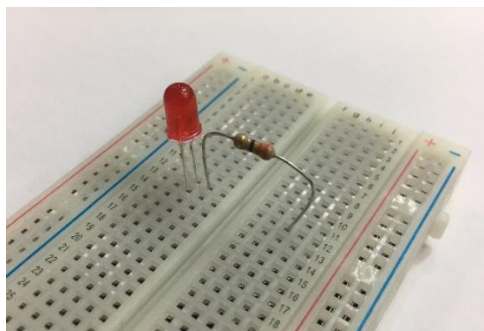


ขั้นตอนการต่อวงจร

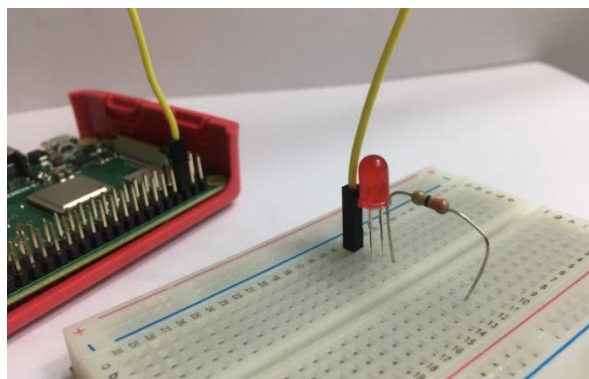
1. เสียบขา LED (ไดโอดเปล่งแสง) ก่อนอื่นจะต้องหาขาขั้วบวก (+) ของ LED โดยการหาว่าขา LED ข้างใดยาวกว่า ขาอื่น นั่น คือ ขั้วบวก (+) ส่วนขา LED ที่สั้นกว่า คือ ขั้วลบ (-)



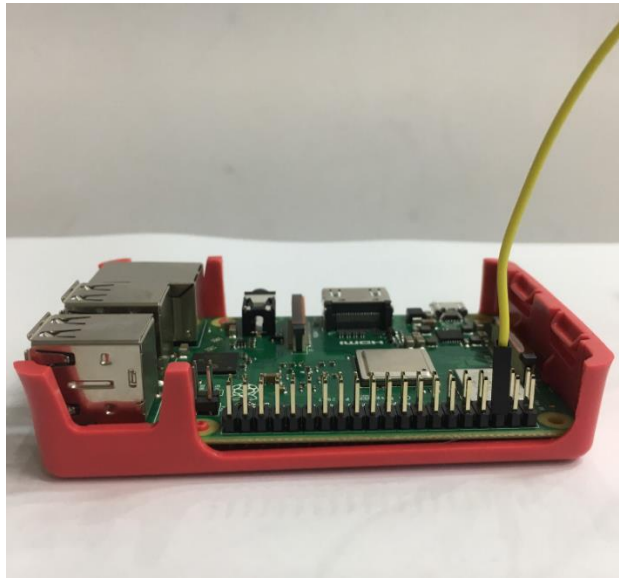
2. นำตัวต้านทาน (Resistor) 330 โอห์ม เสียบเข้ากับขาบวก (+) ของ LED



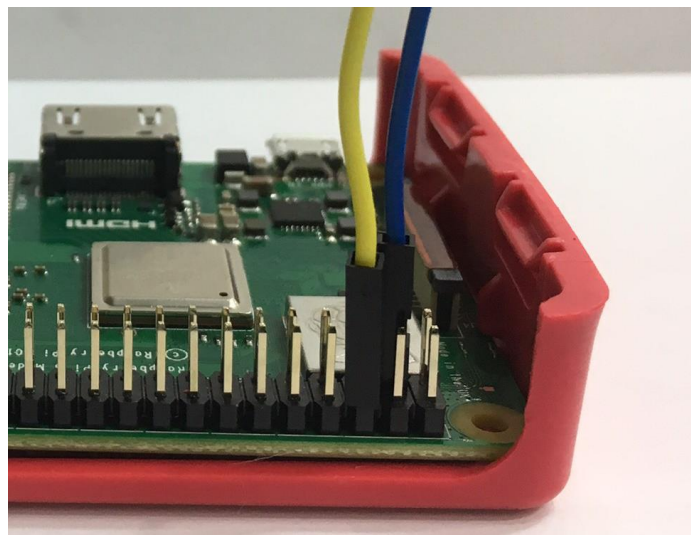
3. ต่อ Ground ให้กับวงจรไฟกะพริบ โดยใช้ Male Cable (หัวเสียบ) ต่อเข้ากับขาลบ (-) ของ LED



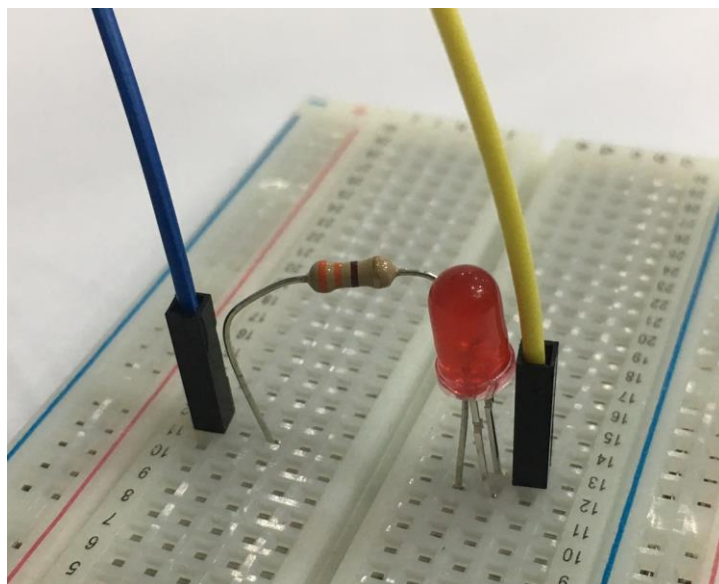
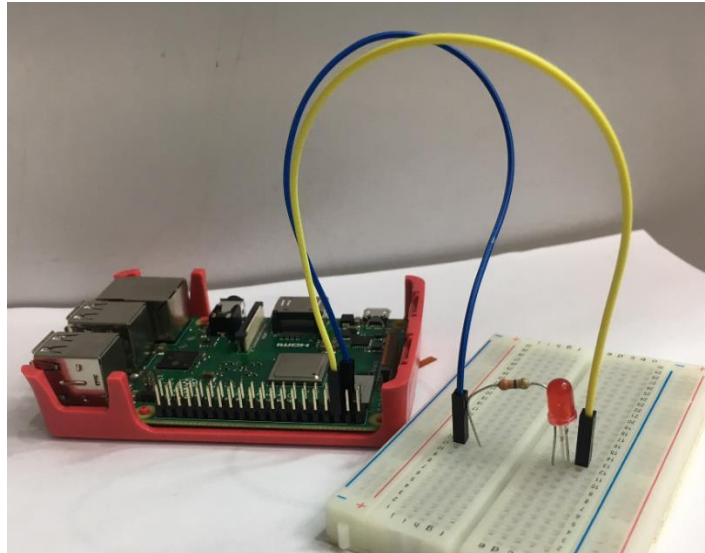
4. นำ Female Cable (หัวสวม) เสียบเข้ากับขา PIN6 (GROUND) ของขา GPIO



5. ป้อนสัญญาณ Output (สัญญาณดิจิทัล ค่า 0 หรือ 1) โดยใช้ Female Cable สวมเข้ากับขา PIN3 (GPIO2)

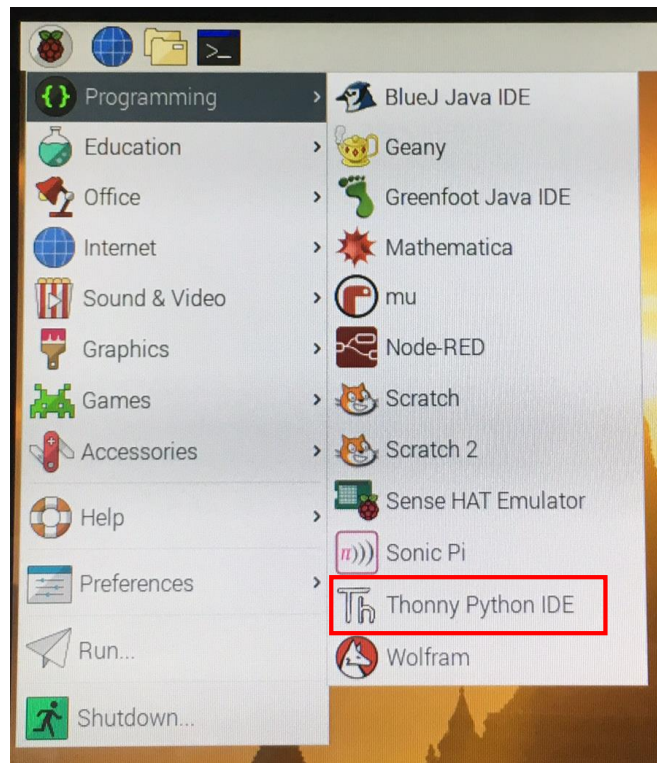


6. เสียบ Male Cable ที่เชื่อมต่อขา PIN3 (GPIO2) ต่อเข้ากับขาตัวต้านทาน 330 โอห์ม ถีว่วงจรไฟกะพริบ

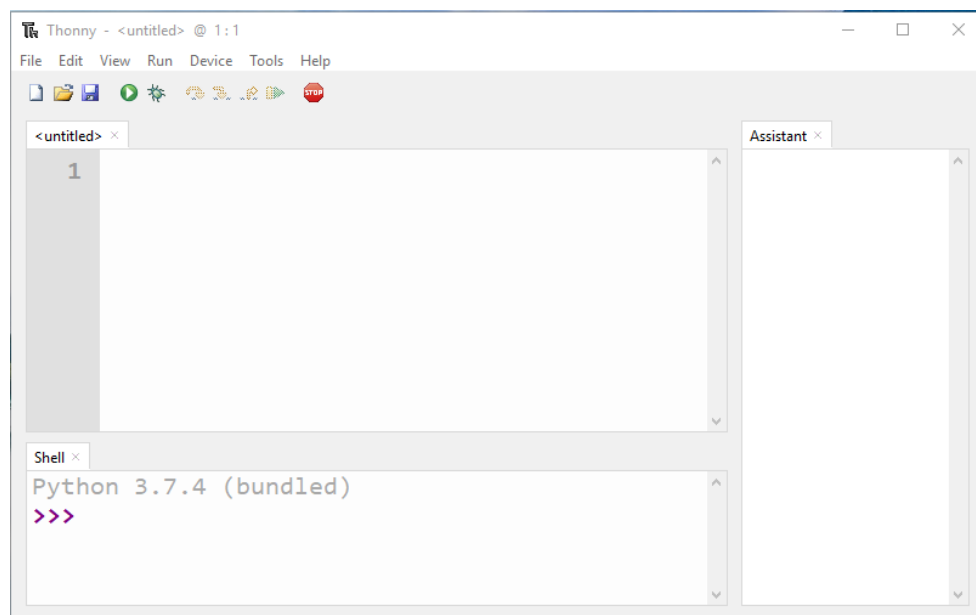


ขั้นตอนการเขียนโปรแกรมด้วยภาษา Python

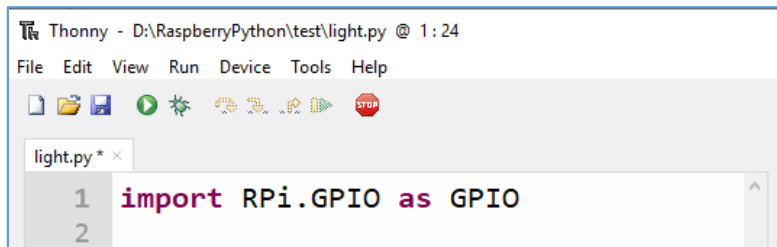
เมื่อเปิดระบบปฏิบัติการ Raspbian ของบอร์ด Raspberry Pi แล้ว ให้คลิกที่ Pi --> Programming --> Thonny Python IDE ซึ่งเป็นโปรแกรมสำหรับเขียนโค้ดคำสั่งด้วยภาษา Python



ระบบจะเปิดหน้าต่าง Thonny Python IDE ขึ้นมาให้เราป้อนคำสั่งหรือพิมพ์ Code เพื่อสร้างโปรแกรมควบคุมวงจร

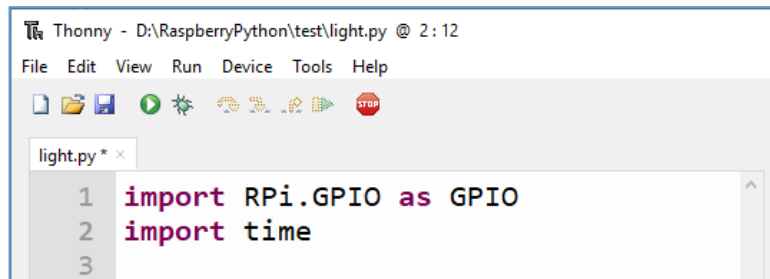


เริ่มต้นเขียนโปรแกรมเพื่อควบคุมขา GPIO โดยจะต้องทำการเรียกใช้ Library GPIO ก่อน โดยพิมพ์คำสั่ง `import RPi.GPIO as GPIO`



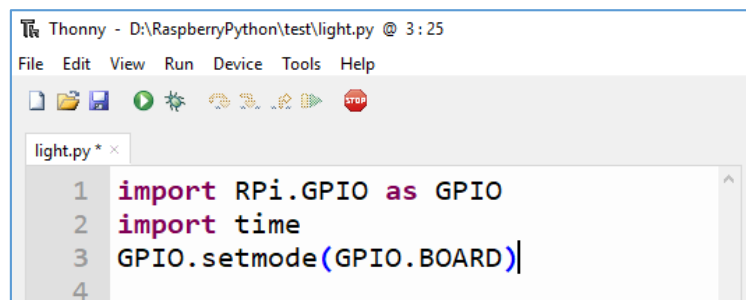
```
Thonny - D:\RaspberryPython\test\light.py @ 1:24
File Edit View Run Device Tools Help
light.py *
1 import RPi.GPIO as GPIO
2
```

จากนั้นให้เรียกใช้ Library time เพื่อใช้ในการหน่วงเวลา โดยให้พิมพ์คำสั่ง `import time`



```
Thonny - D:\RaspberryPython\test\light.py @ 2:12
File Edit View Run Device Tools Help
light.py *
1 import RPi.GPIO as GPIO
2 import time
3
```

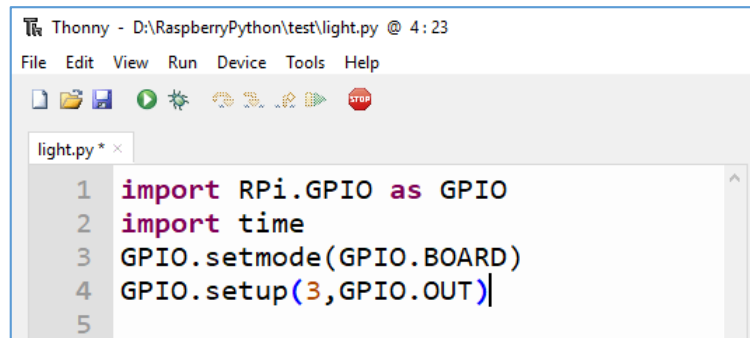
ตั้งค่าโหมดสำหรับการเรียกใช้งาน PIN ของ GPIO โดยสามารถเรียกใช้งานตามลำดับของขา (PIN) ที่อยู่ในบอร์ด RPI (GPIO.BOARD) หรือเรียกใช้งานตามชื่อของขาเหล่านั้น (GPIO.BCM) แต่ในที่นี้จะเรียกใช้งานตามลำดับของขา GPIO (GPIO.BOARD) ซึ่งเป็นวิธีที่ง่ายกว่าและเหมาะสมกับผู้เริ่มต้นใช้งานอีกด้วย โดยพิมพ์คำสั่ง `GPIO.setmode(GPIO.BOARD)` แต่ถ้าจะใช้วิธีเรียกใช้ตามชื่อของขา GPIO ก็เพียงเปลี่ยนจากคำว่า BOARD เป็น BCM



```
Thonny - D:\RaspberryPython\test\light.py @ 3:25
File Edit View Run Device Tools Help
light.py *
1 import RPi.GPIO as GPIO
2 import time
3 GPIO.setmode(GPIO.BOARD)|
4
```

ขั้นตอนต่อไปเป็นการเลือกขาที่จะใช้เป็น Digital Output โดยพิมพ์คำสั่ง `GPIO.setup(3GPIO.OUT)`

โดย 3 หมายถึง ขา PIN3 ที่เราได้เชื่อมต่อกับตัวต้านทาน (Resistor) และ `GPIO.OUT` หมายถึง ให้ขา PIN3 ทำหน้าที่เป็น Digital Output เพื่อส่งสัญญาณดิจิทัลไปยังวงจรไฟกะพริบ



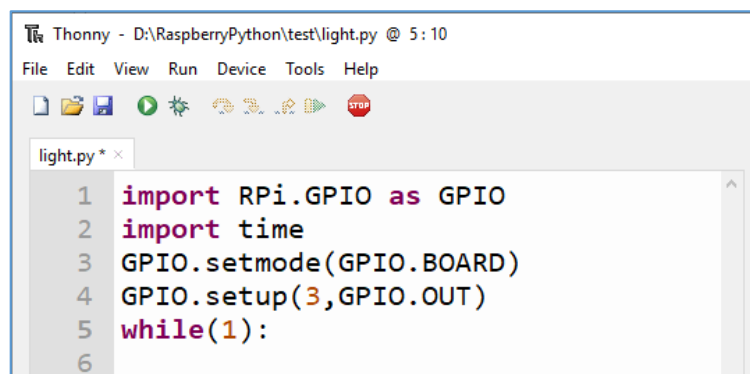
```

Thonny - D:\RaspberryPython\test\light.py @ 4:23
File Edit View Run Device Tools Help

light.py * x
1 import RPi.GPIO as GPIO
2 import time
3 GPIO.setmode(GPIO.BOARD)
4 GPIO.setup(3,GPIO.OUT)
5

```

หากต้องการให้วงจรไฟกะพริบทำงานในลักษณะติดและดับวนไปเรื่อยๆ จะต้องป้อนคำสั่งในการวนลูปไปเรื่อยๆ โดยพิมพ์คำสั่ง `while(1):` หรือ `while 1:` หรือ `while True:`



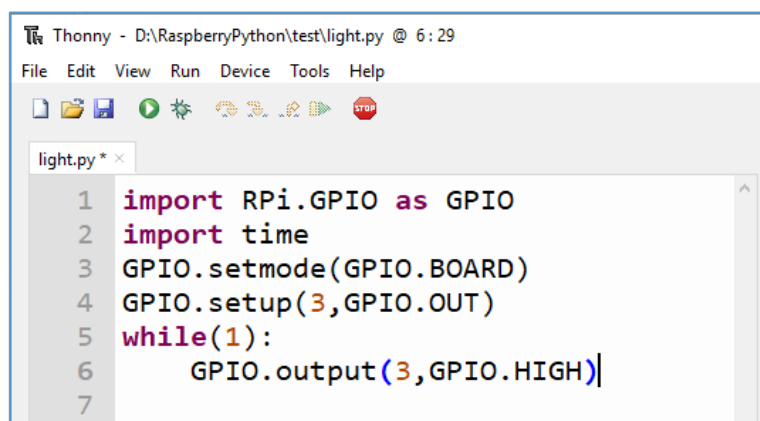
```

Thonny - D:\RaspberryPython\test\light.py @ 5:10
File Edit View Run Device Tools Help

light.py * x
1 import RPi.GPIO as GPIO
2 import time
3 GPIO.setmode(GPIO.BOARD)
4 GPIO.setup(3,GPIO.OUT)
5 while(1):
6

```

ภายในคำสั่ง `while` จะพิมพ์คำสั่งว่า `GPIO.output(3,GPIO.HIGH)` เพื่อสั่งให้ไฟ LED ติด



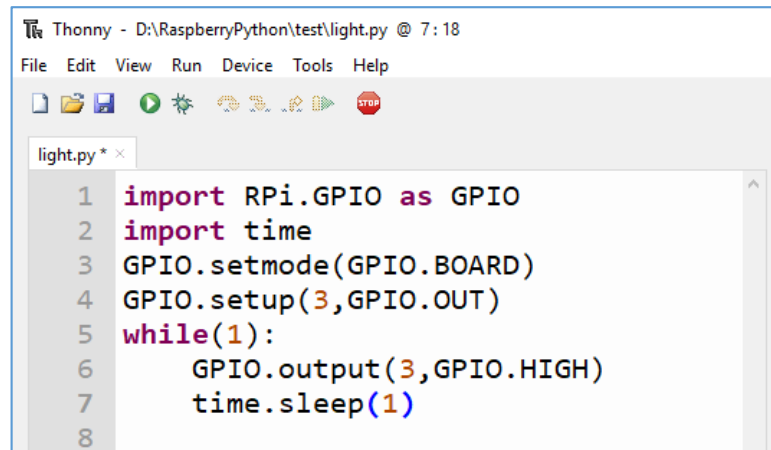
```

Thonny - D:\RaspberryPython\test\light.py @ 6:29
File Edit View Run Device Tools Help

light.py * x
1 import RPi.GPIO as GPIO
2 import time
3 GPIO.setmode(GPIO.BOARD)
4 GPIO.setup(3,GPIO.OUT)
5 while(1):
6     GPIO.output(3,GPIO.HIGH)
7

```

จากนั้นพิมพ์คำสั่งหน่วงเวลาเพื่อให้ไฟติดเป็นระยะเวลาที่วินาที โดยพิมพ์คำสั่งว่า `time.sleep(1)` ตัวเลขในวงเล็บคือจำนวนวินาทีที่หน่วงเวลา



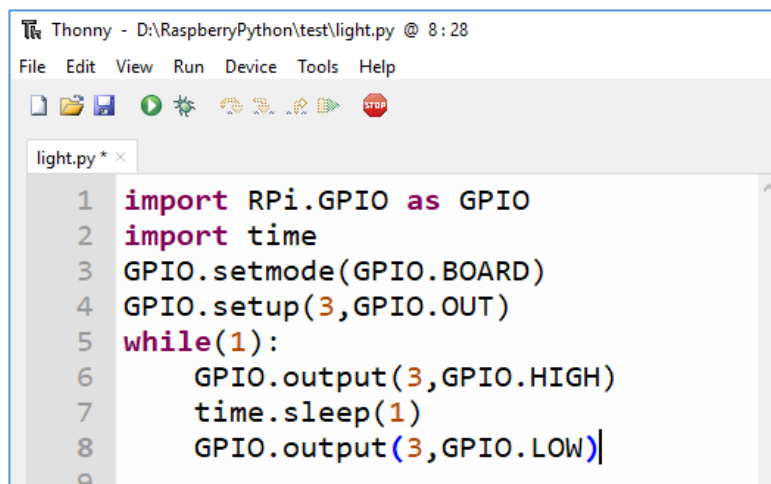
```

Thonny - D:\RaspberryPython\test\light.py @ 7:18
File Edit View Run Device Tools Help

light.py * x
1 import RPi.GPIO as GPIO
2 import time
3 GPIO.setmode(GPIO.BOARD)
4 GPIO.setup(3,GPIO.OUT)
5 while(1):
6     GPIO.output(3,GPIO.HIGH)
7     time.sleep(1)
8

```

ต่อไปจะพิมพ์คำสั่งให้ไฟดับ โดยพิมพ์คำสั่งว่า `GPIO.output(3,GPIO.LOW)`



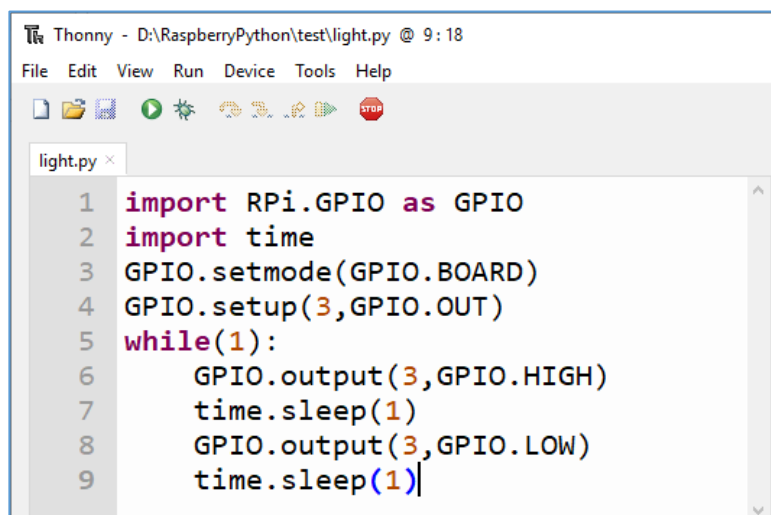
```

Thonny - D:\RaspberryPython\test\light.py @ 8:28
File Edit View Run Device Tools Help

light.py * x
1 import RPi.GPIO as GPIO
2 import time
3 GPIO.setmode(GPIO.BOARD)
4 GPIO.setup(3,GPIO.OUT)
5 while(1):
6     GPIO.output(3,GPIO.HIGH)
7     time.sleep(1)
8     GPIO.output(3,GPIO.LOW)|
9

```

และหน่วงเวลาให้ไฟดับก็วินาที โดยพิมพ์คำสั่งว่า `time.sleep(1)` ให้หน่วงเวลา 1 วินาที



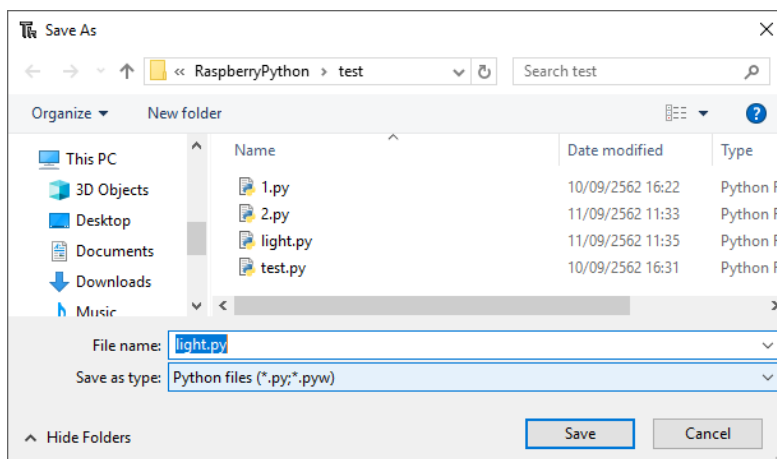
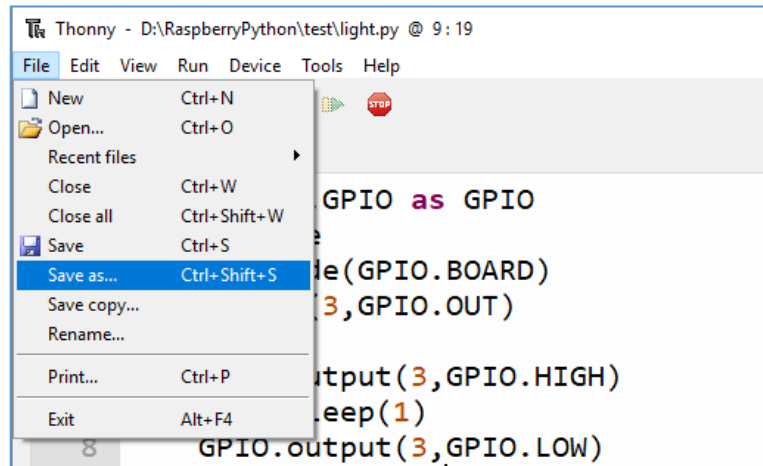
```

Thonny - D:\RaspberryPython\test\light.py @ 9:18
File Edit View Run Device Tools Help

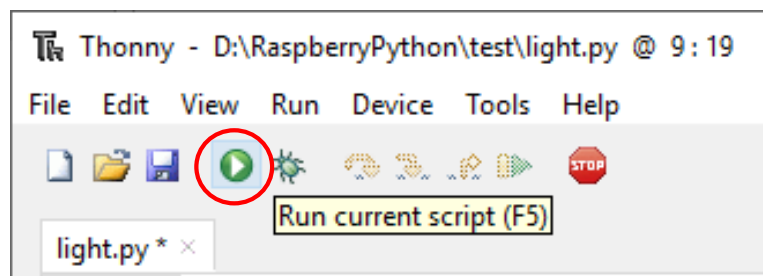
light.py x
1 import RPi.GPIO as GPIO
2 import time
3 GPIO.setmode(GPIO.BOARD)
4 GPIO.setup(3,GPIO.OUT)
5 while(1):
6     GPIO.output(3,GPIO.HIGH)
7     time.sleep(1)
8     GPIO.output(3,GPIO.LOW)
9     time.sleep(1)|

```

เมื่อพิมพ์คำสั่งเสร็จแล้วให้ทำการ Save โดยเลือกที่ File --> Save แล้วทำการตั้งชื่อไฟล์ในช่อง File Name : จากนั้นคลิกปุ่ม Save โดยไฟล์ที่บันทึกจะอยู่ในโฟลเดอร์ /home/pi

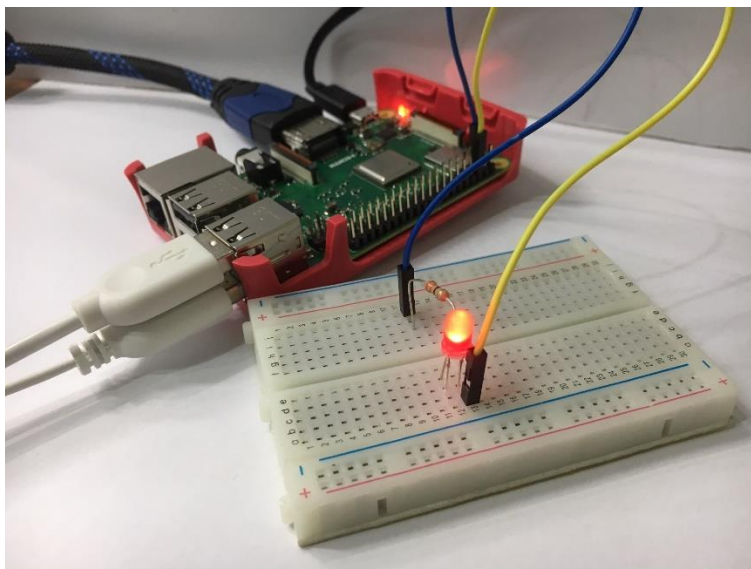


เมื่อการเขียนโปรแกรมเสร็จเรียบร้อยแล้ว ต่อไปจะเป็นการ Run โปรแกรมเพื่อควบคุมการทำงานของวงจรไฟกระพริบ โดยคลิกปุ่ม Run



ผลลัพธ์ที่ได้

ไดโอดเปล่งแสง LED จะติดเป็นเวลา 1 วินาที และดับเป็นเวลา 1 วินาที สลับกันไปเรื่อย ๆ จะเห็นได้ว่าอุปกรณ์ OUTPUT ในที่นี้คือหลอดไฟ LED ซึ่งจะส่งค่า HIGH (ไฟติด) และ LOW (ไฟดับ)



2. โปรแกรมเปิดและปิดหลอดไฟ LED ด้วยปุ่ม Push Button อุปกรณ์

1. บอร์ด Raspberry Pi



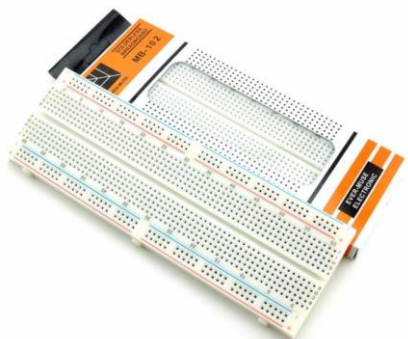
2. หลอด LED



3. ตัวต้านทาน 330 โอห์ม



4. Breadboard



5. สาย Jumper

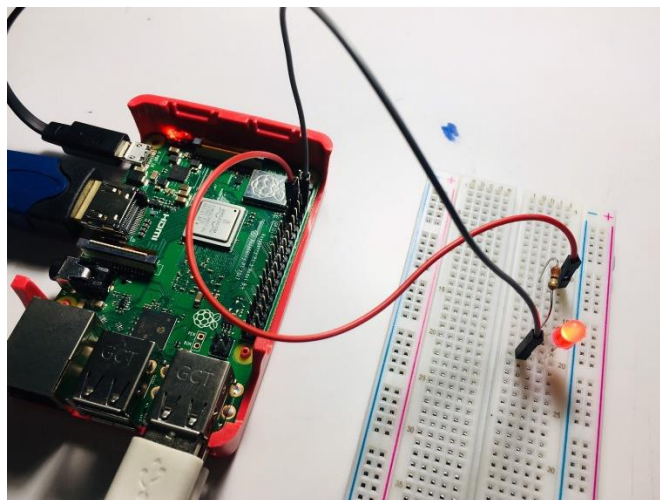


6. Push Button

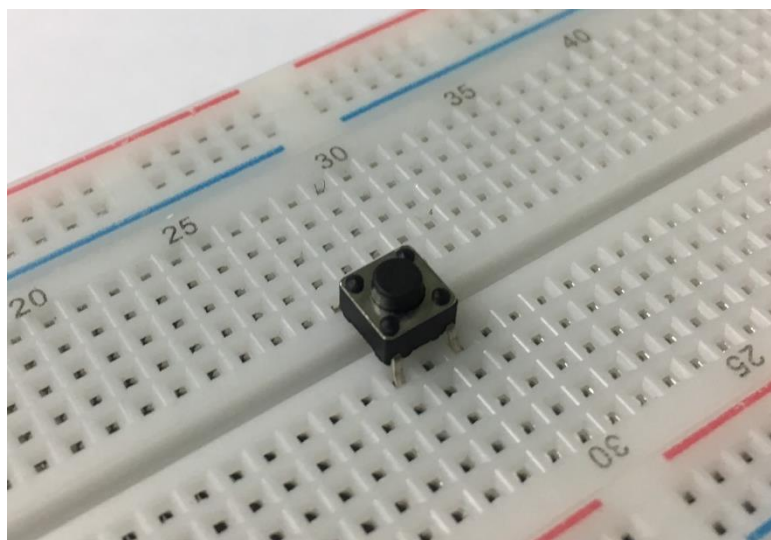


การต่อวงจร

1. เสียบ LED บน Breadboard
2. ขา - (สั้น) ของหลอดไฟ LED เสียบเข้ากับช่อง PIN6 (GROUND) ของขา GPIO
3. ขา + ของ LED ต่อเข้ากับตัวต้านทาน 330 โอห์ม
4. ขาอีกข้างของตัวต้านทาน 330 โอห์มเสียบเข้ากับช่อง PIN7 (GPIO4) ของขา GPIO

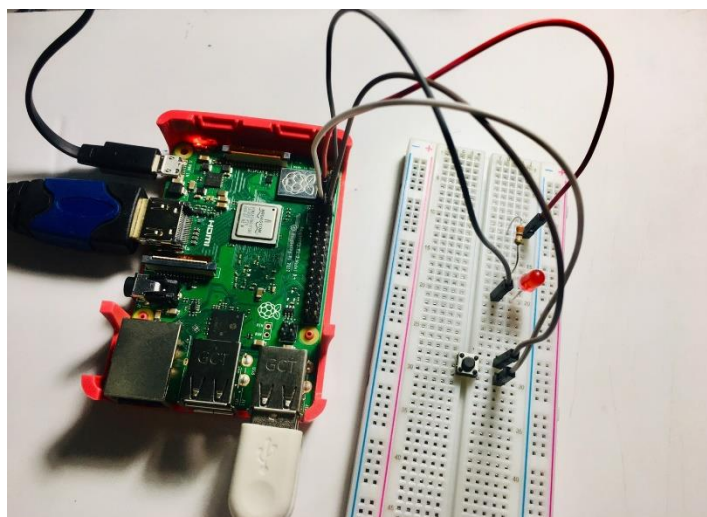


2. เสียบ Pushbutton บน Breadboard



3. เสียบสาย Jumper ที่ขาข้างหนึ่งของ Pushbutton ต่อเข้ากับขา PIN14 (GROUND) ของขา GPIO

4. เสียบสาย Jumper ที่ขาอีกข้างหนึ่งของ Pushbutton ต่อเข้ากับขา PIN11 (GPIO17) ของขา GPIO



การเขียนโค้ดคำสั่ง

```

1 import RPi.GPIO as GPIO
2 import time
3 GPIO.setwarnings(False)
4
5 GPIO.setmode(GPIO.BOARD)
6 GPIO.setup(7,GPIO.OUT)
7 GPIO.output(7,GPIO.LOW)
8 GPIO.setup(11,GPIO.IN,pull_up_down=GPIO.PUD_UP)
9
10 while True:
11     #Don't Push ==> LED ON
12     if GPIO.input(11):
13         GPIO.output(7,GPIO.LOW)
14
15     #Push ==> LED OFF
16     else:
17         GPIO.output(7,GPIO.HIGH)
18         time.sleep(0.1)
19 GPIO.cleanup()

```

ผลลัพธ์

เมื่อกดปุ่ม Pushbutton ไฟ LED จะติด (HIGH) แต่ถ้าไม่กดปุ่ม Pushbutton ไฟ LED จะดับ (LOW)
 ซึ่งอุปกรณ์ที่เป็น INPUT คือ Pushbutton และอุปกรณ์ที่เป็น OUTPUT คือ หลอดไฟ LED

ตัวอย่างการเขียนโปรแกรมเบื้องต้นเพื่อควบคุม Input – Output ของขา GPIO ด้วยภาษา Scratch

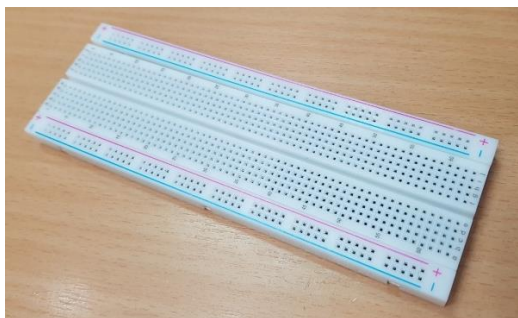
การเปิดหลอด LED

อุปกรณ์ที่ต้องใช้

1. บอร์ด Raspberry Pi



2. Breadboard 830 ช่อง



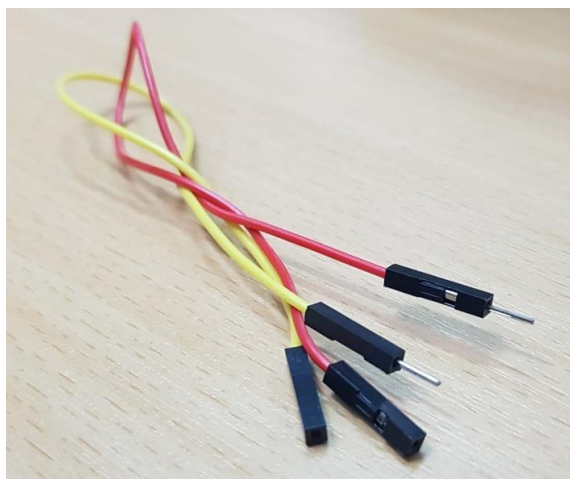
3. ตัวต้านทาน 330 โอห์ม



4. หลอดไฟ LEDs

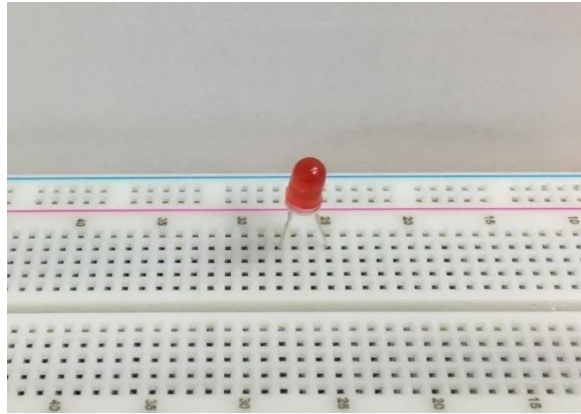


5. สาย Jumper ตัวผู้-ตัวเมีย

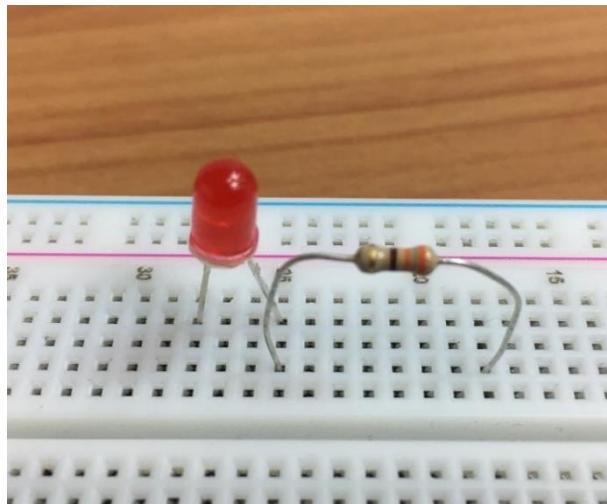


ขั้นตอนการต่อกับบอร์ด

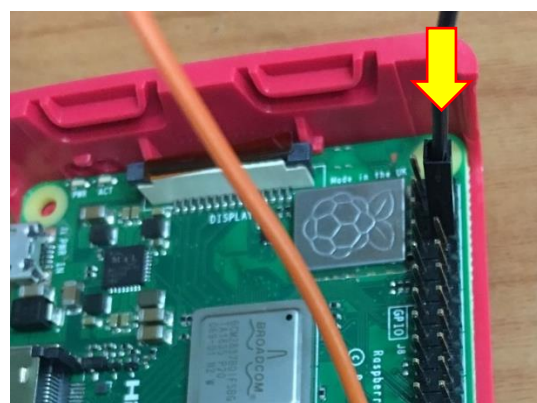
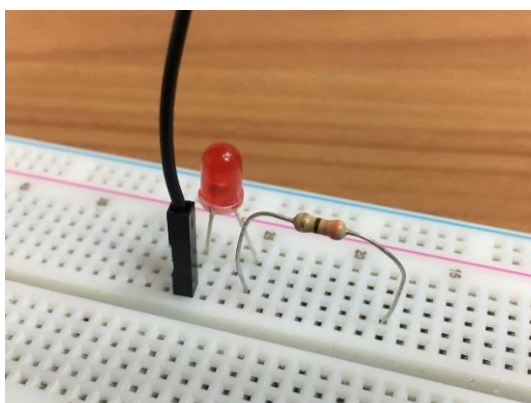
1. นำหลอด LED มาเสียบเข้าที่ Breadboard โดยขั้วบวกขั้วลบจะอยู่คนละแถว เราจะสังเกตได้จาก ถ้า ขาวาวจะเป็นขั้วบวก ขาสั้นจะเป็นขั้วลบ



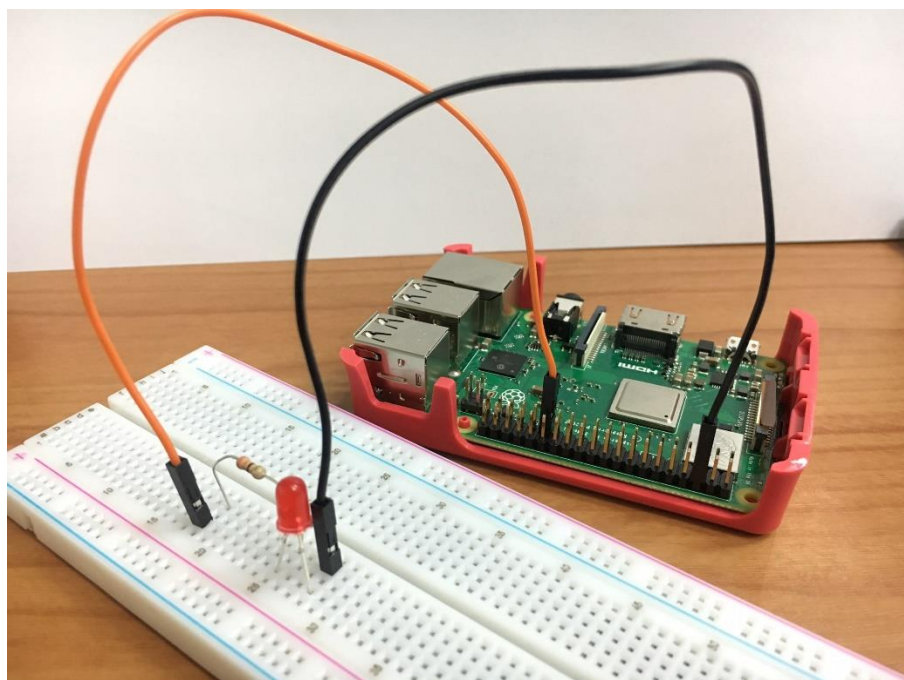
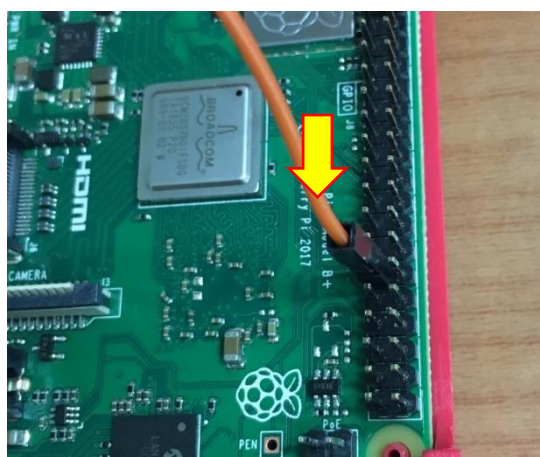
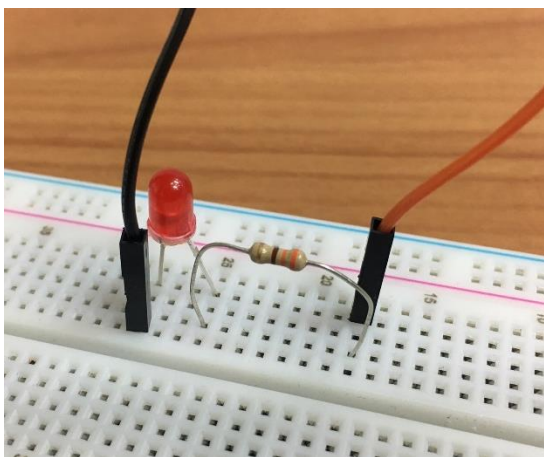
2. นำตัวต้านทานมาเสียบเข้าแถวเดียวกันกับขั้วบวก



3. นำสาย Jumper ตัวผู้-ตัวเมีย โดยเอาด้านตัวผู้มาเสียบเข้าที่ช่องแถวเดียวกับขั้วลบของหลอด LED จากนั้นเอาด้านตัวเมียเสียบเข้ากับขา PIN6 (Ground) ของบอร์ด Raspberry Pi

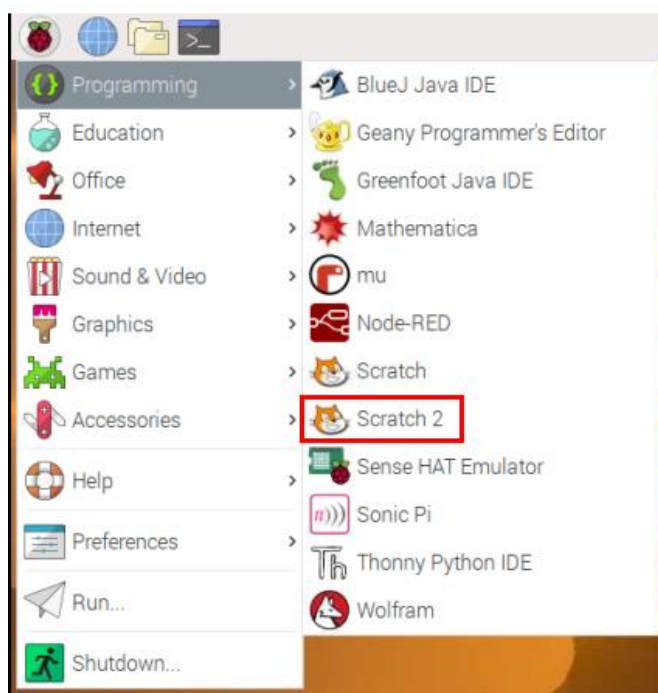


4. นำสาย Jumper ตัวผู้-ตัวเมีย โดยเอาด้านตัวผู้เสียบเข้าช่องแถวเดียวกับตัวด้านทาน เพื่อรับไฟเข้า แล้วเอาด้านตัวเมียเสียบเข้าที่ช่อง GPIO Input/Output ในที่นี้จะเสียบเข้ากับขา PIN29 (GPIO5)

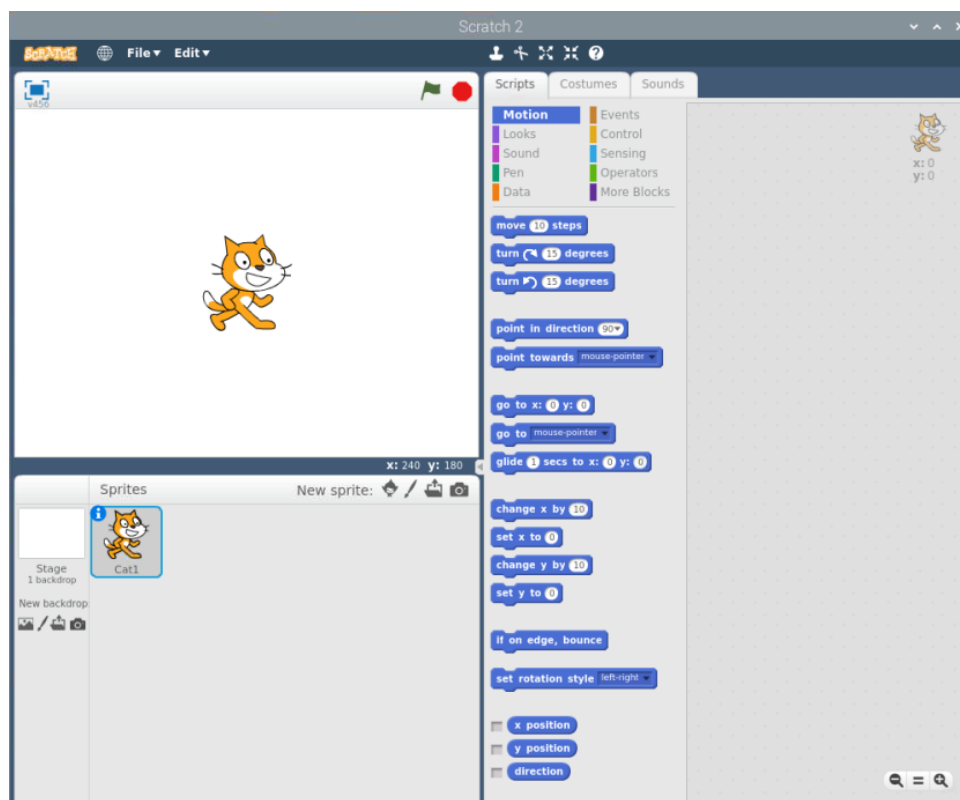


เขียนโปรแกรม

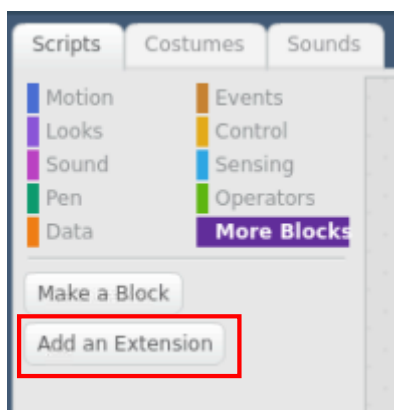
1. เข้าสู่โปรแกรม Scratch 2 โดยที่ Raspberry Pi => Programming => Scratch 2



จะได้หน้าต่างดังนี้



2. เริ่มเขียนโปรแกรม โดยเรียกใช้ไลบรารี GPIO ไปที่ Scripts => More Blocks => Add an Extension



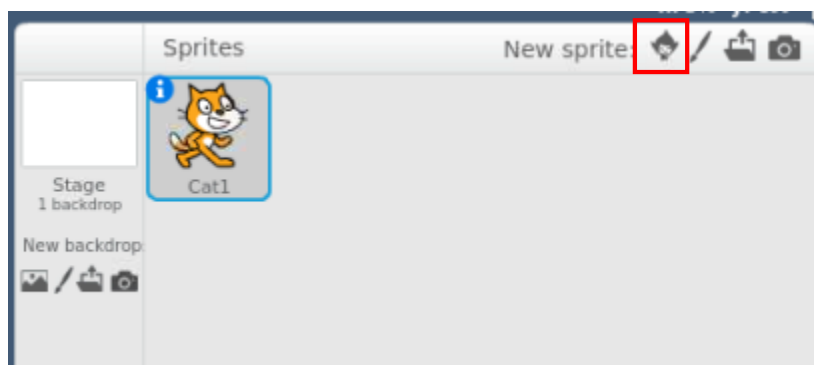
เลือก Pi GPIO แล้วกดปุ่ม OK



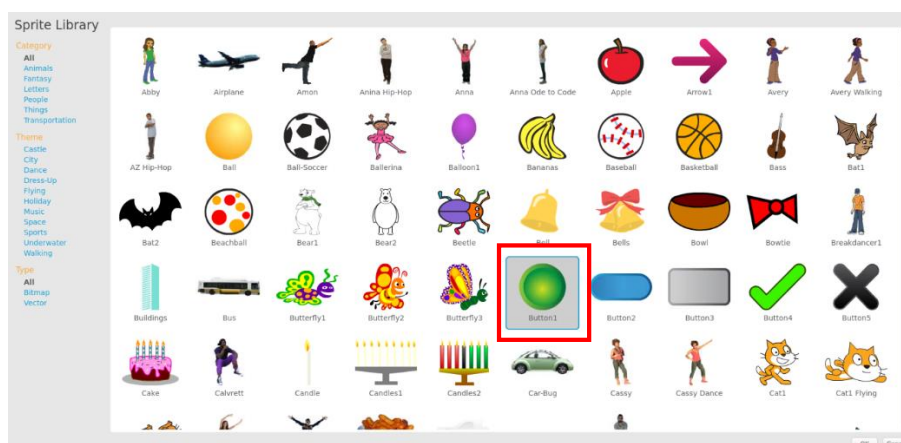
จะได้ คำสั่ง GPIO ขึ้นมาดังภาพ



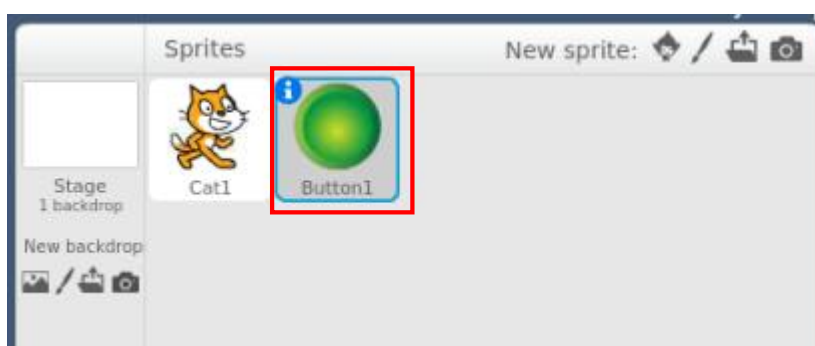
3. เพิ่ม Sprites เพื่อสร้างปุ่มกด โดยไปที่



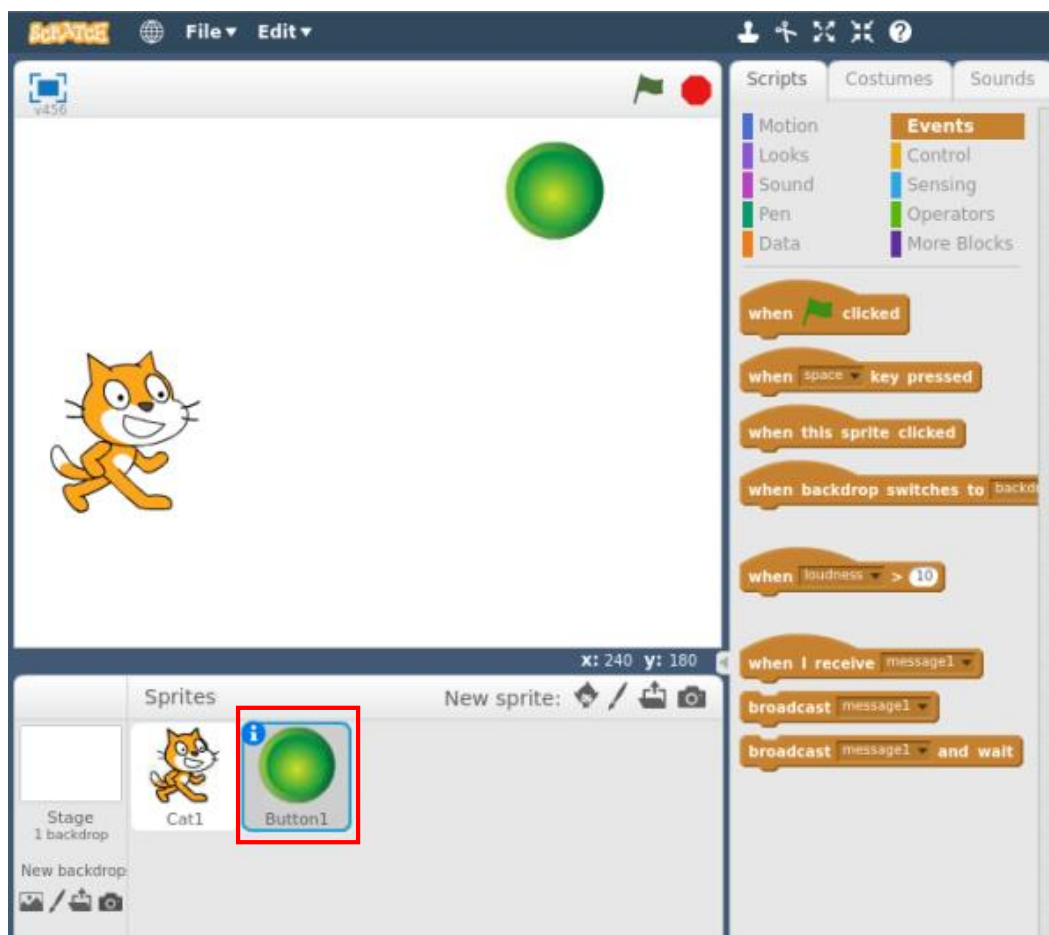
เลือกรูปปุ่มที่ต้องการ แล้วกด OK



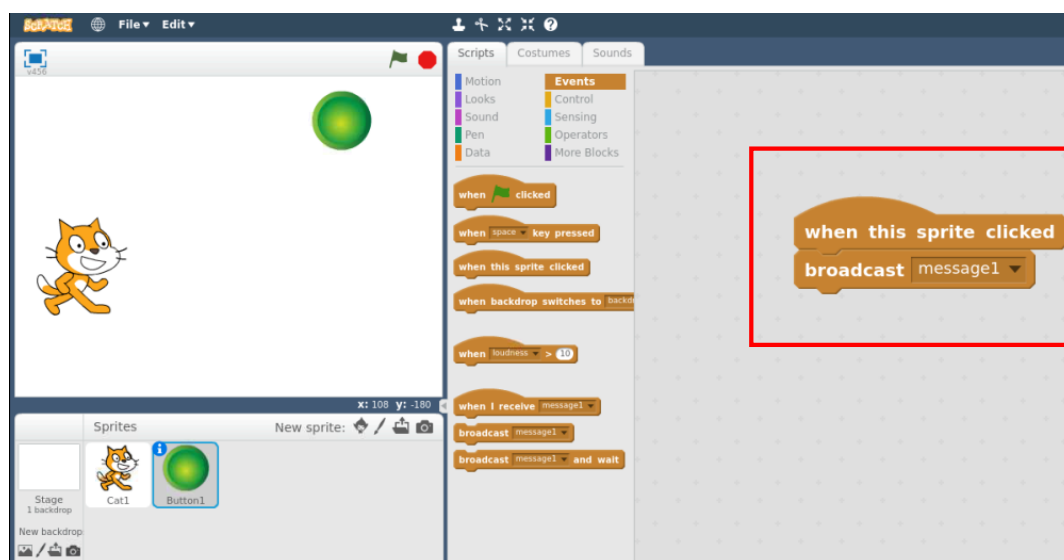
จะได้ Sprites เพิ่มขึ้นมา 1



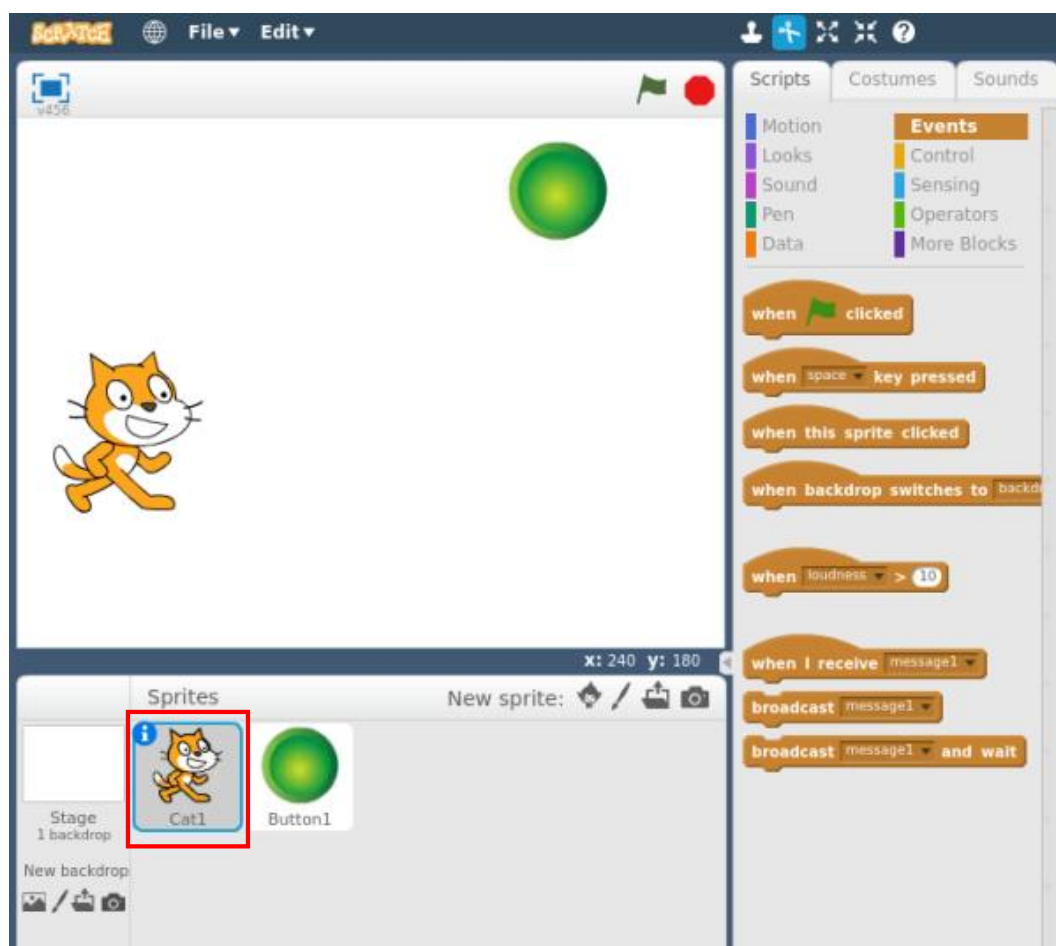
4. เลือก Sprites button1 เพื่อเขียนคำสั่งเข้าไป



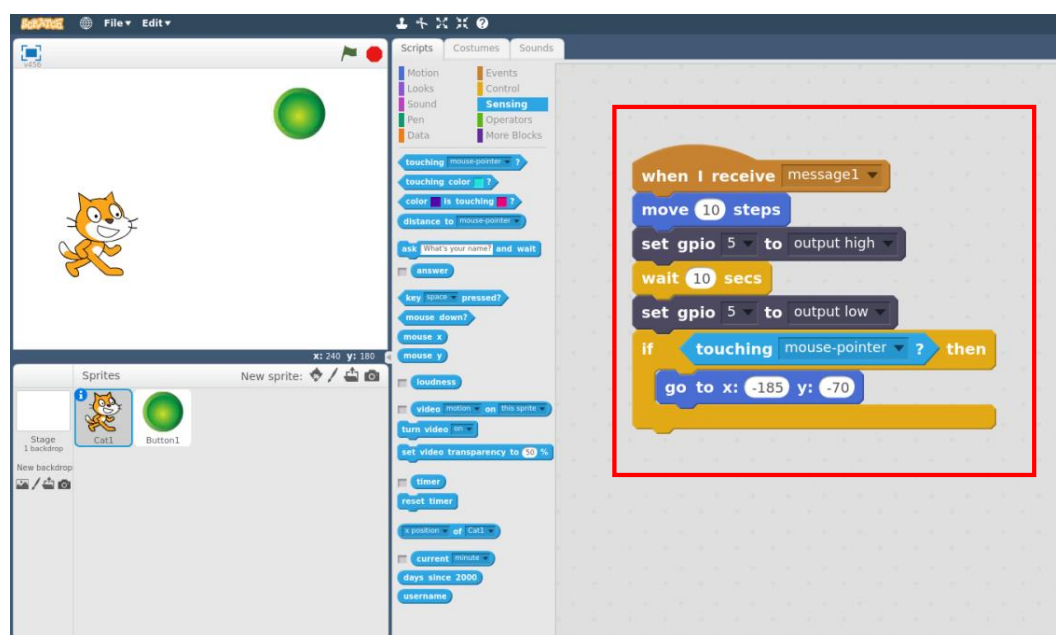
เขียนคำสั่งโดยลากบล็อก



5. เลือก Sprites Cat1 เพื่อเขียนคำสั่งเข้าไป



6. เขียนคำสั่งโดยลากบล็อก



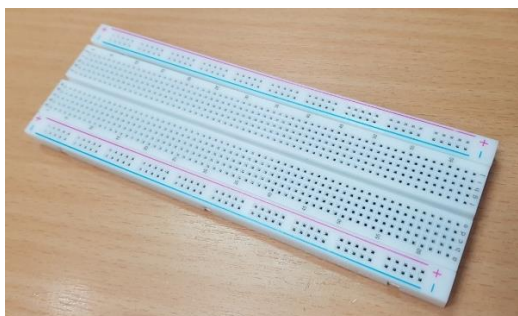
การทำสัญญาณไฟจราจร

อุปกรณ์

1. บอร์ด Raspberry Pi



2. Breadboard 830 ช่อง



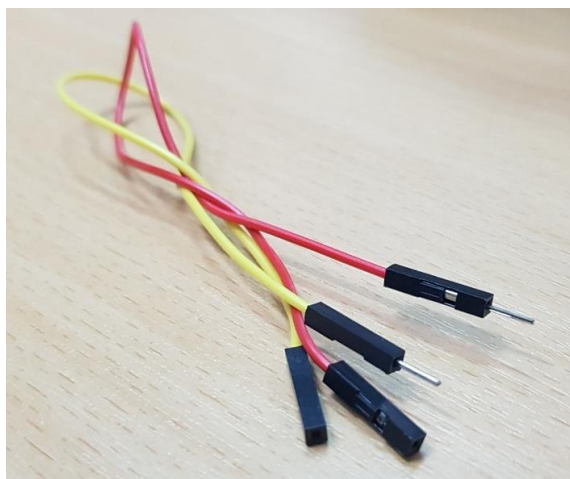
3. ตัวต้านทาน 4.7 กิโลโอห์ม



4. หลอดไฟ LEDs

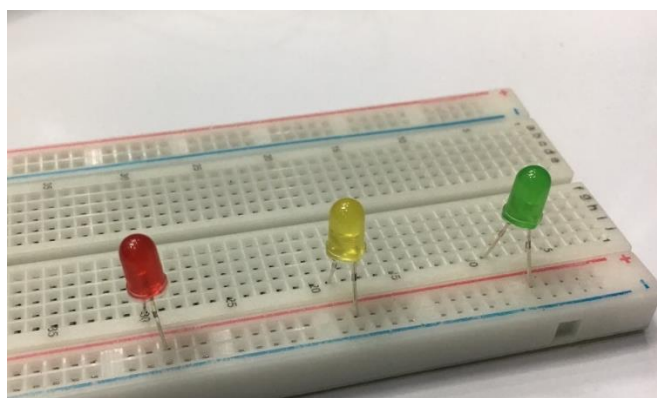


5. สาย Jumper ตัวผู้-ตัวเมีย

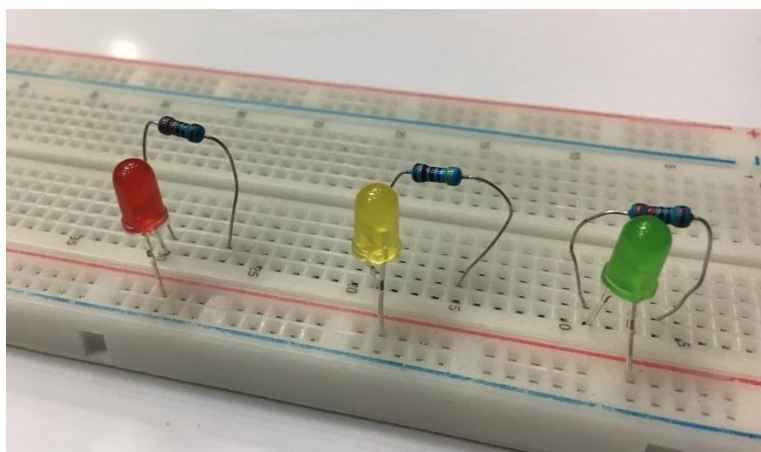


ขั้นตอนต่อวงจร

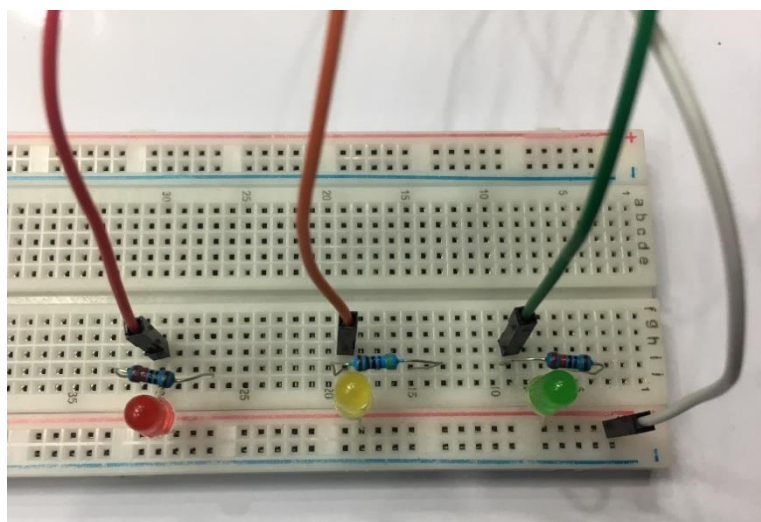
1. นำหลอด LED ทั้ง 3 ดวง มาเสียบเข้าที่ Breadboard โดยขั้วบวกขั้วลบจะอยู่คนละแถว เราจะสังเกตได้จาก ถ้าขายาวจะเป็นขั้วบวก ขาสั้นจะเป็นขั้วลบ

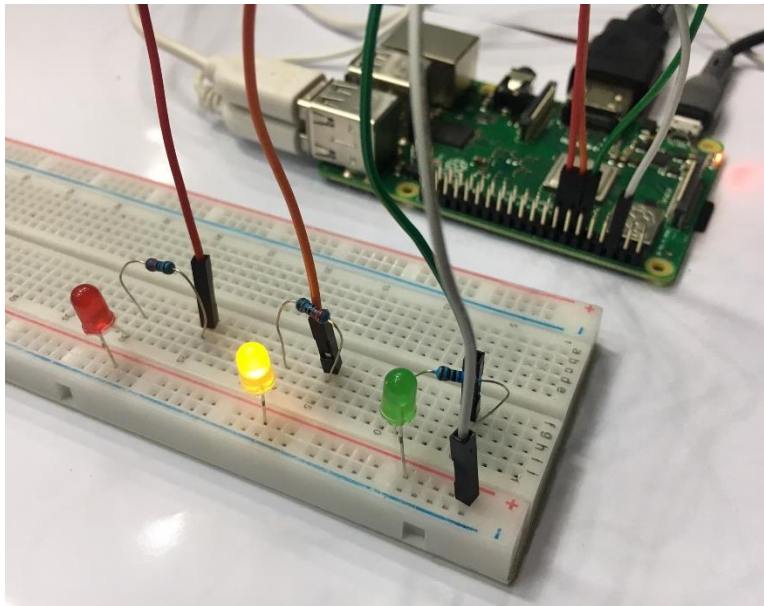
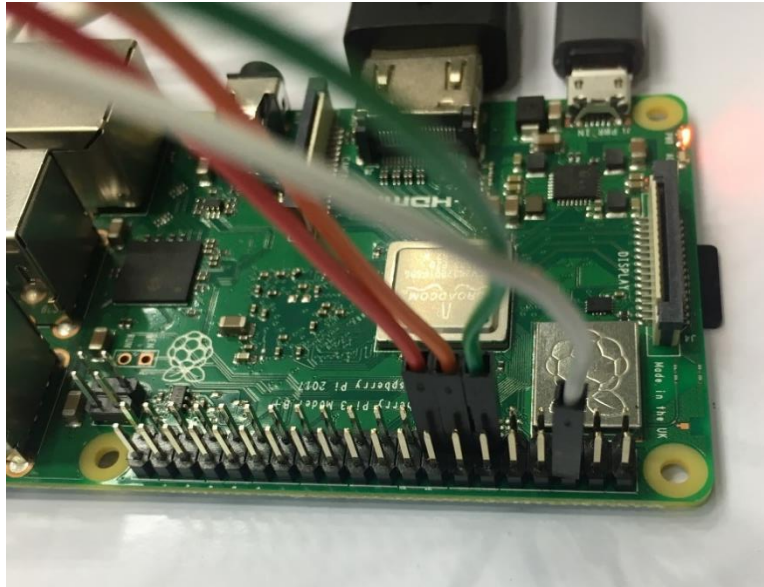


2. นำตัวต้านทานมาเสียบเข้าแถวเดียวกันกับขั้วบวก



3. นำสาย Jumper ตัวผู้-ตัวเมีย โดยเอาด้านตัวผู้มาเสียบเข้าที่ช่องแถวเดียวกับขั้วลบของหลอด LED จากนั้นเอาด้านตัวเมียเสียบเข้า Input/Output (ในที่นี้จะเสียบเข้ากับขา PIN11 (GPIO17), PIN13 (GPIO27), PIN15 (GPIO22) ตามลำดับ)





เขียนคำสั่ง



จากบล็อกคำสั่งเริ่มการทำงานโดยประกาศตัวแปร 3 ตัวแปร เพื่อตั้งค่า GPIO ให้ตัวแปร **Green** เท่ากับ GPIO 17 ตัวแปร **yello** เท่ากับ GPIO 27 **red** เท่ากับ GPIO 22 จากนั้นกำหนดค่าให้ทำงาน ถ้า GPIO 17 สว่าง GPIO 27 ดับ GPIO 22 ดับ แล้วรอ 1 วินาที ถ้า GPIO 17 ดับ GPIO 27 ดับ GPIO 22 สว่าง แล้วรอ 1 วินาที ถ้า GPIO 17 ดับ GPIO 27 สว่าง GPIO 22 ดับ รอ 1 วินาที ทำงานวนซ้ำไปเรื่อยๆ

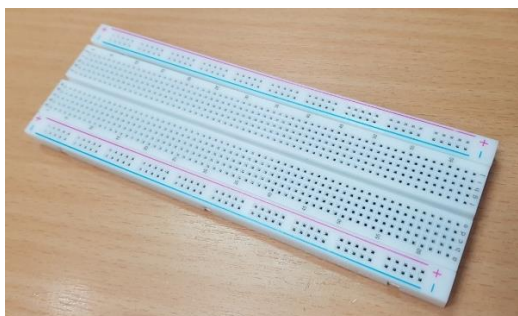
การสร้างเกมส่ถึงเก็บกล้วย

อุปกรณ์ที่ใช้

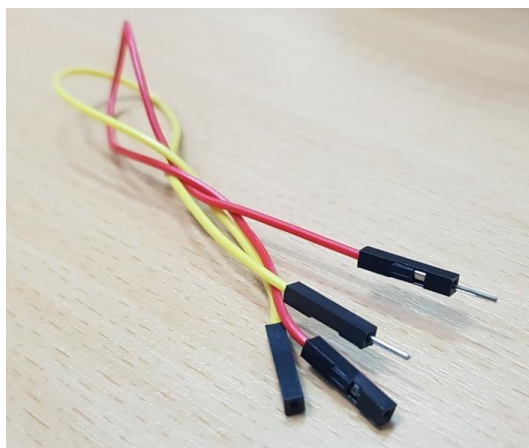
1. บอร์ด Raspberry Pi



2. Breadboard 830 ช่อง



3. สาย Jumper ตัวผู้-ตัวเมีย



4. สาย Jumper ตัวเมีย – ตัวเมีย

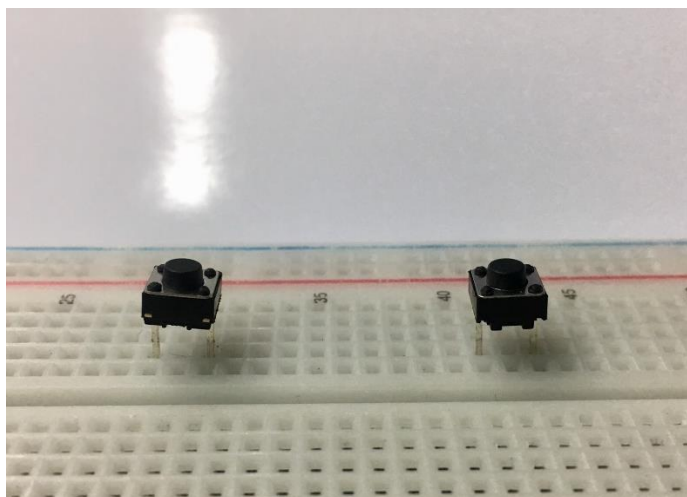


5. ปุ่มกด Pushbutton

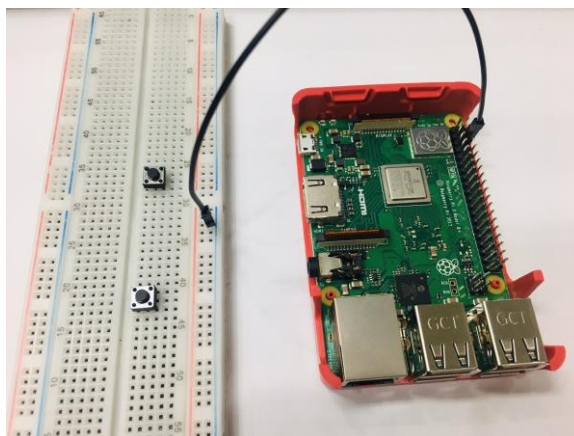


การต่อวงจร

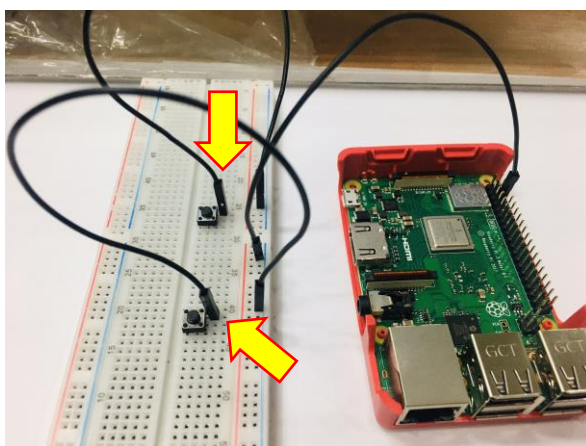
1. นำปุ่มกด Pushbutton มาเสียบเข้าไปใน Breadboard



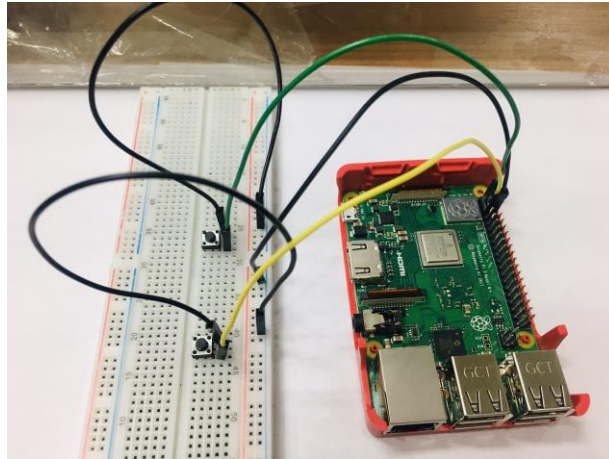
2. ใช้สาย Jumper แบบตัวผู้-ตัวเมีย 1 เส้น เสียบจาก PIN6 (Ground) ของบอร์ด Raspberry Pi มายังช่องใดก็ได้ในแถว – ของ Breadboard



3. ใช้สาย Jumper แบบตัวผู้-ตัวผู้ เสียบจากขาต้านหนึ่งของปุ่ม Push Button ไปยังช่องใดก็ได้ในแถว – ของ Breadboard (ทำเหมือนกันทั้ง 2 ปุ่ม)

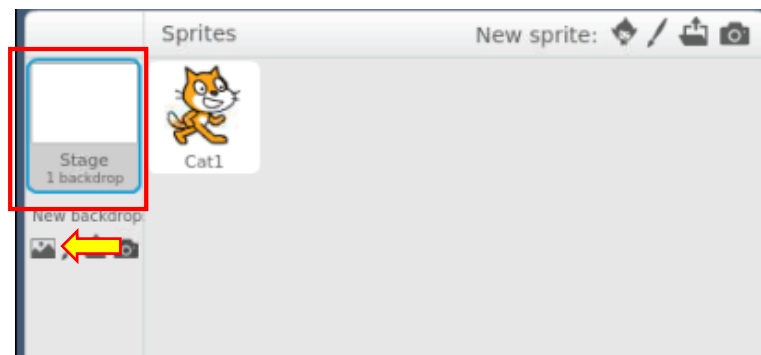


4. นำสาย Jumper ตัวผู้+ตัวเมีย มาเสียบแถวเดียวกับปุ่มกด Pushbutton อีกขาที่ยังว่างอยู่ เชื่อมต่อไปยังขา PIN3 (GPIO2) และ PIN5 (GPIO3) ของบอร์ด Raspberry Pi ตามลำดับ

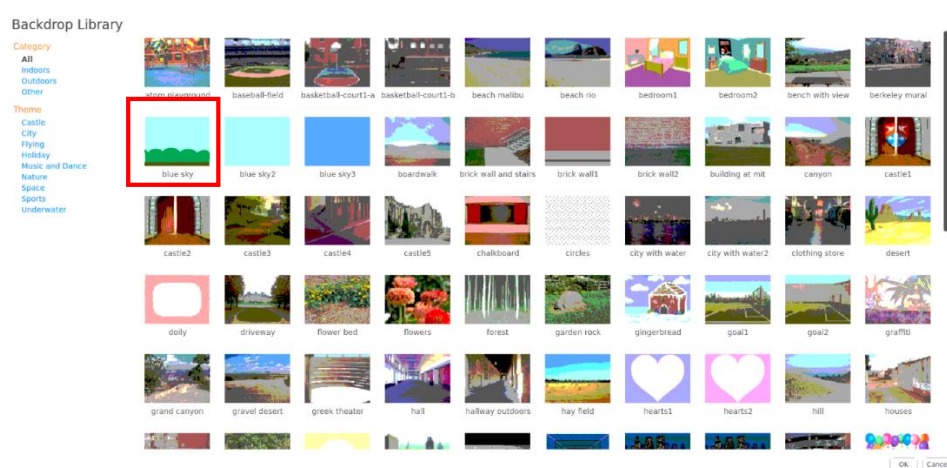


ขั้นตอนการเขียนคำสั่ง

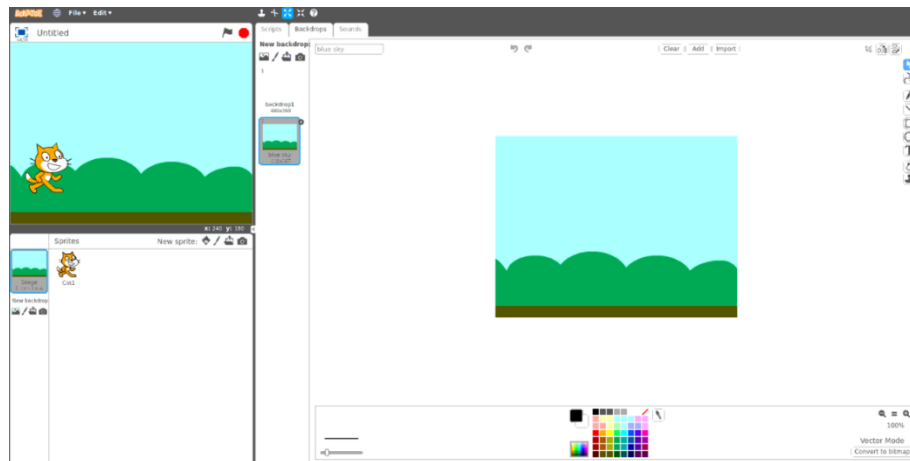
1. สร้างฉากพื้นหลังให้กับเกม โดยไปที่ Stage แล้วคลิกที่ปุ่ม Choose backdrop from library



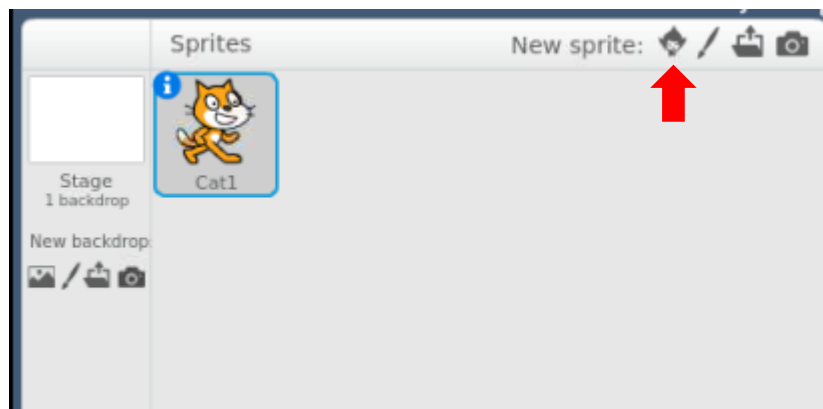
เลือกพื้นหลังที่ต้องการ แล้วกด OK



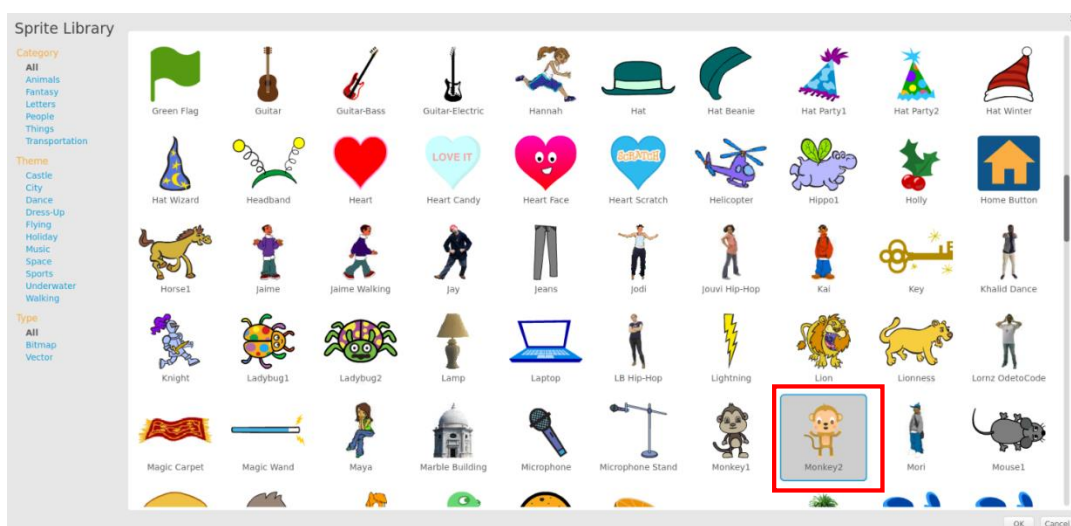
จะได้ภาพพื้นหลัง ดังรูป



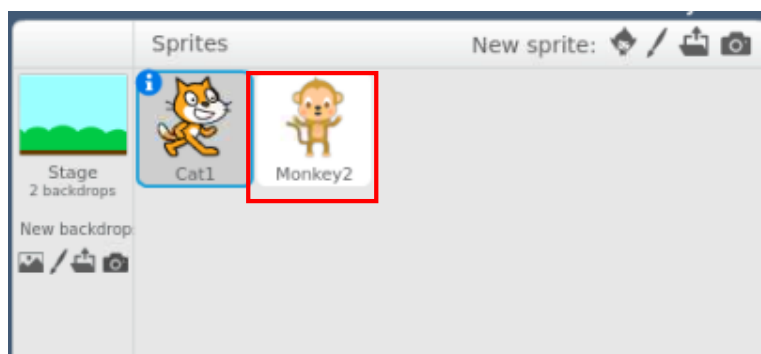
2. สร้างตัวละครใหม่ โดยคลิกที่ New sprite



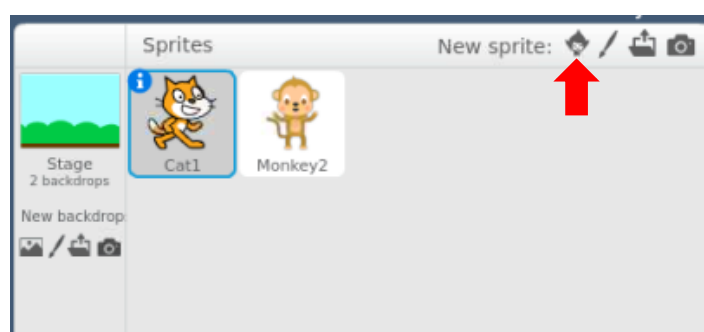
เลือกตัวละครที่เป็นรูปลิงที่ชื่อว่า Monkey2 แล้วกด OK



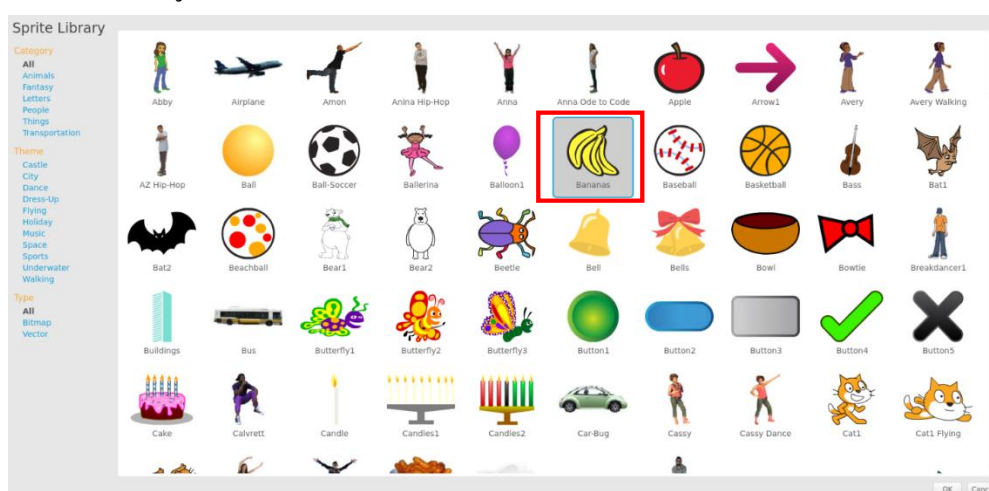
จะได้ตัวละครเพิ่มขึ้นมา จากนั้นให้ลบตัวละคร Cat1 ทิ้งไป



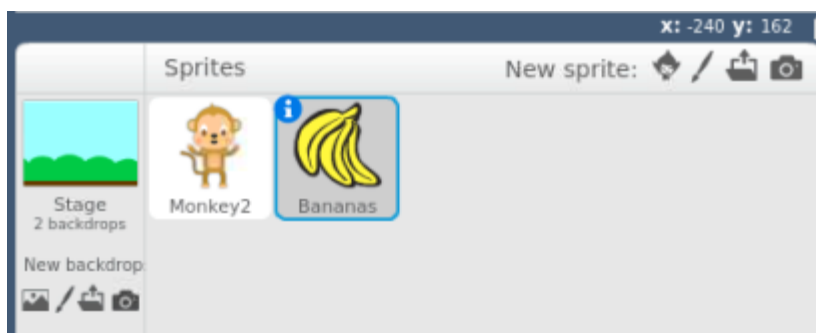
3. สร้างตัวละครขึ้นมาอีก 1 ตัว โดยไปที่ New sprite



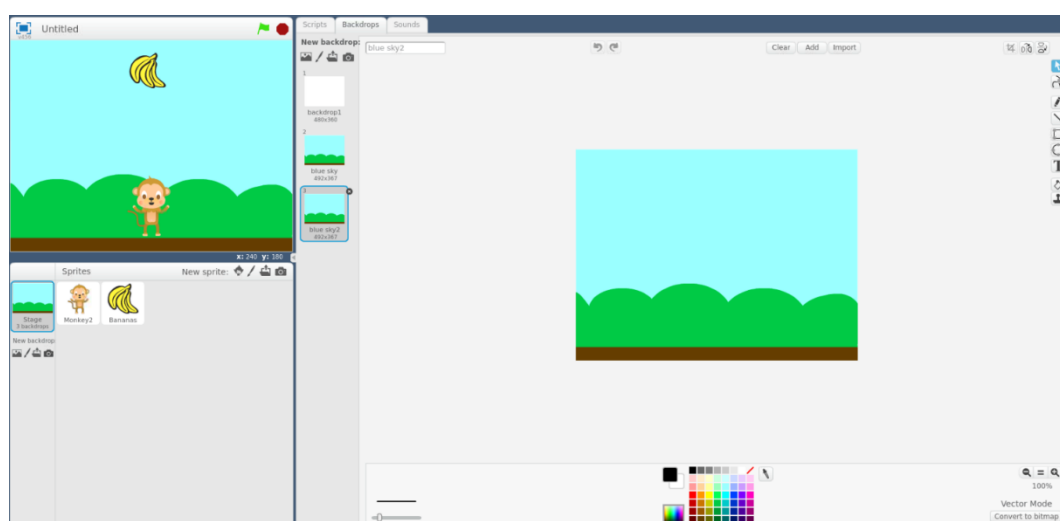
เลือกตัวละครที่เป็นรูปกล้วย แล้วกด OK



จะได้ตัวละคร 2 ตัว ดังรูป



4. สร้างฉากพื้นหลังใหม่ขึ้นมาอีก 1 ฉาก เพื่อสร้างเป็นฉากจบเกม โดยไปที่ Stage แล้วคลิกที่ปุ่ม
Choose backdrop from library แล้วเลือกภาพพื้นหลังที่ต้องการ



5. พิมพ์ข้อความเข้าไปในฉากพื้นหลังตอนจบว่า “Game over” โดยไปที่เครื่องมือ Text

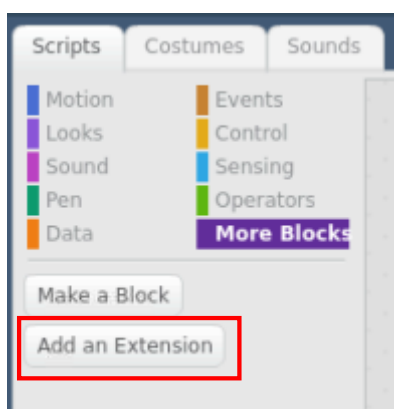


6. เลื่อนตำแหน่งข้อความมาไว้ตรงกลางฉาก โดยคลิกที่เครื่องมือ Select แล้วเลือกกล่องข้อความ



7. เริ่มเขียนโปรแกรม โดยเรียกใช้ไลบรารี GPIO ขึ้นมาใช้งาน โดยไปที่ Scripts => More Blocks

=> Add an Extension



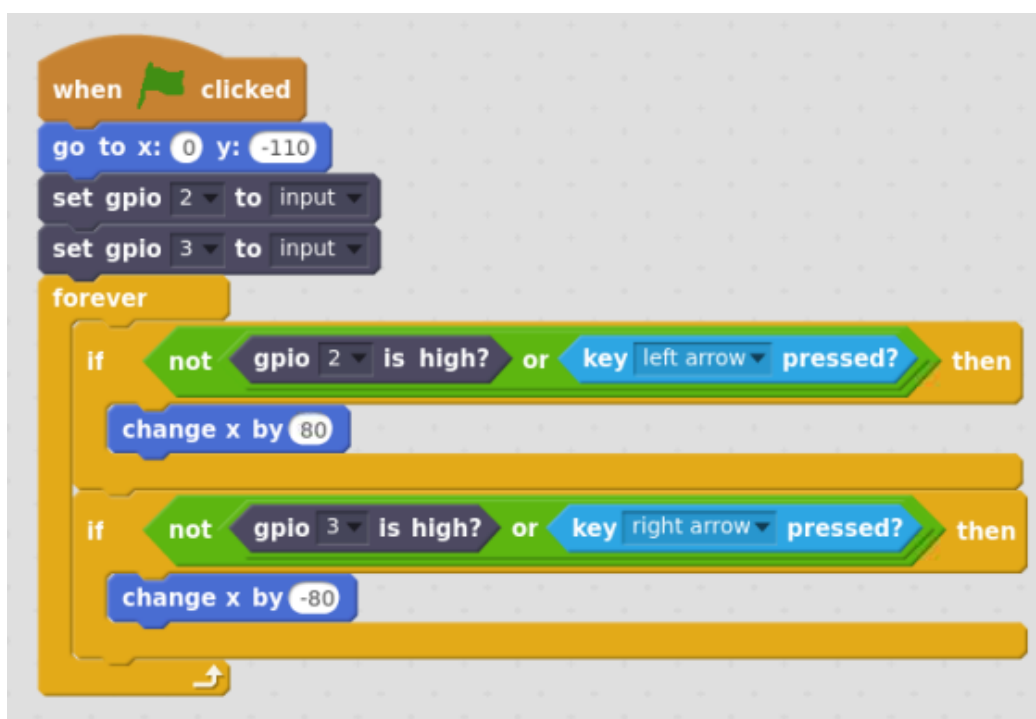
8. เลือก Pi GPIO แล้วกดปุ่ม OK



จะได้ คำสั่ง GPIO ขึ้นมาดังรูป

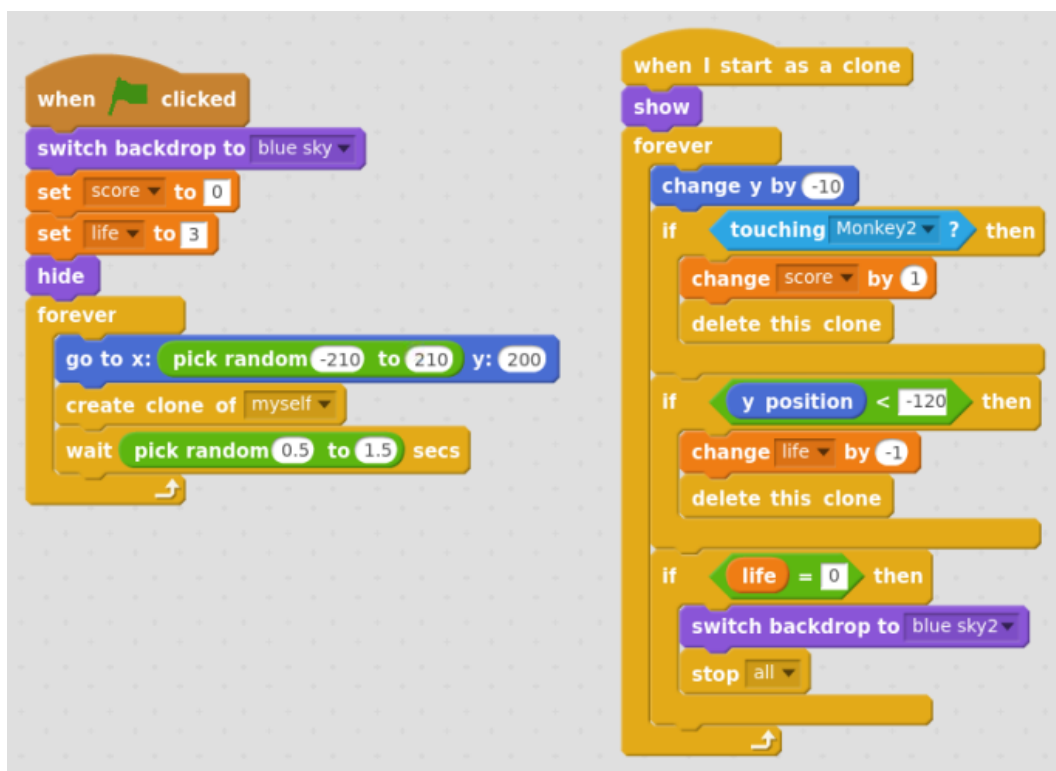


9. คลิกที่ตัวละคร Monkey2 แล้วลากบล็อกคำสั่ง ดังรูป



จากบล็อกคำสั่ง เป็นการเขียนคำสั่งเชื่อมต่อกับขา GPIO กับบอร์ด Raspberry Pi เพื่อรับค่าจากปุ่ม pushbutton 2 ปุ่ม คือ ปุ่มที่ 1 คือ GPIO2 ปุ่มที่ 2 คือ GPIO3 ถ้ากดปุ่มที่เชื่อมต่อกับ GPIO2 จะเคลื่อนที่ไปทางซ้าย ถ้ากดปุ่มที่เชื่อมต่อกับ GPIO3 จะวิ่งไปทางขวา โดยจะเคลื่อนที่วนไปเรื่อยๆ เมื่อมีการกดปุ่ม

10. คลิกที่ตัวละคร Bananas แล้วลากบล็อกคำสั่ง ดังรูป



จากบล็อกคำสั่ง เป็นการเขียนคำสั่ง โดยเมื่อเรากดตรงเขียวเพื่อเริ่มเกมจะให้เราคลิกพื้นหลัง Blue sky จากนั้นสร้างตัวแปร 2 ตัว ตัวแรก Score เพื่อเก็บคะแนน โดยให้เริ่มจาก 0 คะแนน อีกหนึ่งตัวแปรชื่อ life เพื่อเก็บค่าชีวิตและกำหนดค่าชีวิตเท่ากับ 3 จากนั้นทำการซ่อนตัวละครก่อน ถัดมาเราจะกำหนดตำแหน่งให้กับตัวละครโดยกำหนดให้ตัวละครนั้นสุ่มค่าในแกน x ระหว่าง -210 ถึง 210 ในแกน y กำหนดตำแหน่งที่ $y = 200$ แล้วให้ตัวละครทำการโคลนนิ่งตัวเองซึ่งก่อนที่จะตัวละครจะโคลนนิ่งนั้นให้รอ และสุ่มเวลาในการรอประมาณ 0.5 ถึง 1.5 วินาที โดยให้ทำงานวนซ้ำตลอด หลังจากทีตัวละครโคลนนิ่งแล้วให้แสดงตัวละครและให้ตัวละครเปลี่ยนตำแหน่ง y ทีละ -10 เพื่อให้ตัวละครเลื่อนลงมาเรื่อยๆตามแกน y หลังจากนั้นเราจะสร้างเงื่อนไข 3 เงื่อนไข เงื่อนไขที่หนึ่งคือ ถ้าตัวละครกล้วสัมผัสกับตัวละครลิง ให้เปลี่ยนค่า score ทีละ 1 แล้วลบตัวเองทิ้ง เงื่อนไขที่สอง คือถ้าตัวละครกล้วไม่สัมผัสตัวละครลิงแล้วเลื่อนลงมาจนถึงตำแหน่งที่ $y < -120$ ให้ลดค่า life ลงทีละ 1 แล้วลบตัวเองทิ้ง เงื่อนไขที่สาม คือ ถ้าค่า $life = 0$ ให้สลับฉากหลังเป็น Blue sky2 แล้วหยุดการทำงานทั้งหมด

หน่วยการเรียนรู้ที่ ๕ การใช้งาน Sensor โมดูล และอุปกรณ์ที่เชื่อมต่อบอร์ด Raspberry Pi

กลุ่มสาระการเรียนรู้วิทยาศาสตร์

สาระที่ ๔ เทคโนโลยี

มาตรฐาน ว ๔.๒ เข้าใจและใช้แนวคิดเชิงคำนวณในการแก้ปัญหาที่พบในชีวิตจริงอย่างเป็นขั้นตอนและเป็นระบบ ใช้เทคโนโลยีสารสนเทศและการสื่อสารในการเรียนรู้ การทำงาน และการแก้ปัญหาได้อย่างมีประสิทธิภาพ รู้เท่าทัน และมีจริยธรรม

ตัวชี้วัด

ว ๔.๒ ม.๑/๒ ออกแบบและเขียนโปรแกรมอย่างง่ายเพื่อแก้ปัญหาทางคณิตศาสตร์หรือวิทยาศาสตร์

ว ๔.๒ ม.๒/๒ ออกแบบและเขียนโปรแกรมที่ใช้ตรรกะและฟังก์ชันในการแก้ปัญหา

สาระสำคัญ

๑. เรียนรู้และเข้าใจลักษณะการใช้งานและการต่อวงจรเซนเซอร์ โมดูล และอุปกรณ์อื่นๆ เข้ากับบอร์ด Raspberry Pi

๒. เรียนรู้การเขียนโปรแกรมเพื่อใช้งานเซนเซอร์ โมดูล และอุปกรณ์ต่างๆ ด้วยภาษา Python

สาระการเรียนรู้

ความรู้

- ความรู้เกี่ยวกับเซนเซอร์ โมดูล และอุปกรณ์ต่างๆ ที่เชื่อมต่อบอร์ด
- ความรู้เกี่ยวกับการต่อวงจรเซนเซอร์ โมดูล และอุปกรณ์ต่างๆ เข้ากับบอร์ด Raspberry Pi
- การเขียนโปรแกรมเพื่อใช้งานเซนเซอร์ โมดูล และอุปกรณ์ต่างๆ ด้วยภาษา Python

ทักษะ/กระบวนการ

- ปฏิบัติตามตัวอย่างที่แสดงให้เห็นทีละขั้นตอนจากขั้นพื้นฐานไปสู่ขั้นที่ซับซ้อนขึ้น
- ฝึกปฏิบัติด้วยตนเองจนเกิดความชำนาญ หรือทำได้โดยอัตโนมัติ ซึ่งอาจเป็นงานชิ้นเดิมหรืองานที่คิดขึ้นใหม่

ที่คิดขึ้นใหม่

- ใช้กระบวนการคิดศาสตร์ด้านการคิดคำนวณและการแก้ปัญหา
- กระบวนการกลุ่ม มุ่งให้ผู้เรียนทำงานร่วมกันเป็นทีม

คุณลักษณะที่พึงประสงค์

- รักชาติ ศาสน์ กษัตริย์
- ซื่อสัตย์
- มีวินัย
- ใฝ่เรียนรู้
- อยู่อย่างพอเพียง
- มุ่งมั่นในการทำงาน
- รักความเป็นไทย
- มีจิตสาธารณะ

หน่วยการเรียนรู้ที่ 5

การใช้งาน Sensor โมดูล และอุปกรณ์ที่เชื่อมต่อบอร์ด Raspberry Pi

ก่อนที่จะเริ่มต้นศึกษาการใช้งานอุปกรณ์ต่างๆ ที่ใช้งานร่วมกับบอร์ด Raspberry Pi ให้ทำขั้นตอนดังต่อไปนี้

1. ต่อเมาส์และคีย์บอร์ดเข้ากับพอร์ต USB ของบอร์ด Raspberry Pi
2. เสียบสาย HDMI ที่จอ 모니터เข้ากับพอร์ต HDMI ของบอร์ด Raspberry Pi
3. เสียบสาย Adapter เพื่อจ่ายไฟให้กับบอร์ด Raspberry Pi เพื่อเปิดการใช้งาน ก็จะเริ่มต้นเข้าสู่

หน้าจอ Desktop ของ Raspberry Pi



Sensor ที่ใช้กับบอร์ด Raspberry Pi

1. เซนเซอร์อินฟราเรดตรวจจับวัตถุ IR Reflective Sensor



IR Reflective Sensor เป็นตัวรับและตัวส่ง infrared ในตัวสัญญาณ infrared จะส่งสัญญาณออกมาเมื่อมีวัตถุมาบัง โดยคลื่นสัญญาณ infrared ที่ถูกส่งออกมาจะสะท้อนกลับไปเข้าตัวรับสัญญาณ สามารถนำมาใช้ตรวจจับวัตถุที่อยู่ตรงหน้าได้ และสามารถปรับความไว ระยะเวลาตรวจจับ และระยะใกล้หรือไกลได้

ภายในตัวเซ็นเซอร์จะมีตัวส่ง Emitter และตัวรับ Receiver ติดตั้งภายในตัวเดียวกัน ทำให้ไม่จำเป็นต้องเดินสายไฟทั้งสองฝั่ง ทำให้การติดตั้งใช้งานได้ง่ายกว่า แต่อย่างไรก็ตามจำเป็นต้องติดตั้งตัวแผ่นสะท้อนหรือ Reflector ไว้ตรงข้ามกับตัวเซ็นเซอร์เอง โดยโฟโต้เซ็นเซอร์แบบที่ใช้แผ่นสะท้อนแบบนี้จะเหมาะสำหรับชิ้นงานที่มีลักษณะทึบแสงไม่เป็นมันวาว เนื่องจากอาจทำให้ตัวเซ็นเซอร์เข้าใจผิดว่าเป็นตัวแผ่นสะท้อน และทำให้ทำงานผิดพลาดได้

เซ็นเซอร์นี้จะมีช่วงในการทำงานหรือระยะในการตรวจจับได้ไกลกว่าแบบ Opposed mode ซึ่งในสภาวะการทำงานปกติตัวรับ Receiver จะสามารถรับสัญญาณแสงจากตัวส่ง Emitter ได้ตลอดเวลา เนื่องจากลำแสงจะสะท้อนกับแผ่นสะท้อน Reflector อยู่ตลอดเวลา จะแสดงค่าเป็น 1

หน้าที่หลักของเซ็นเซอร์ชนิดนี้ จะคอยตรวจจับวัตถุที่เคลื่อนที่ตัดผ่านหน้าเซ็นเซอร์ เมื่อวัตถุผ่านเข้ามาขวางที่หน้าเซ็นเซอร์ จะขวางลำแสงที่ส่งจากตัวส่ง Emitter ที่ส่งไปยังแผ่นสะท้อน จึงทำให้ตัวรับ Receiver ไม่สามารถรับลำแสงที่จะสะท้อนกลับมาได้ จะแสดงค่าเป็น 0

ฝึกทักษะการทดลองกิจกรรมที่ 1 เรียนรู้การใช้งาน IR Reflective Sensor

อุปกรณ์ที่ใช้สำหรับการทดลอง

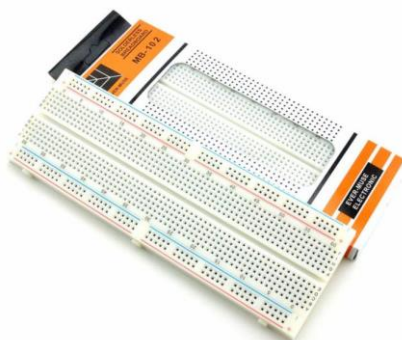
1. บอร์ด Raspberry Pi



2. IR Reflective Sensor



3. Breadboard

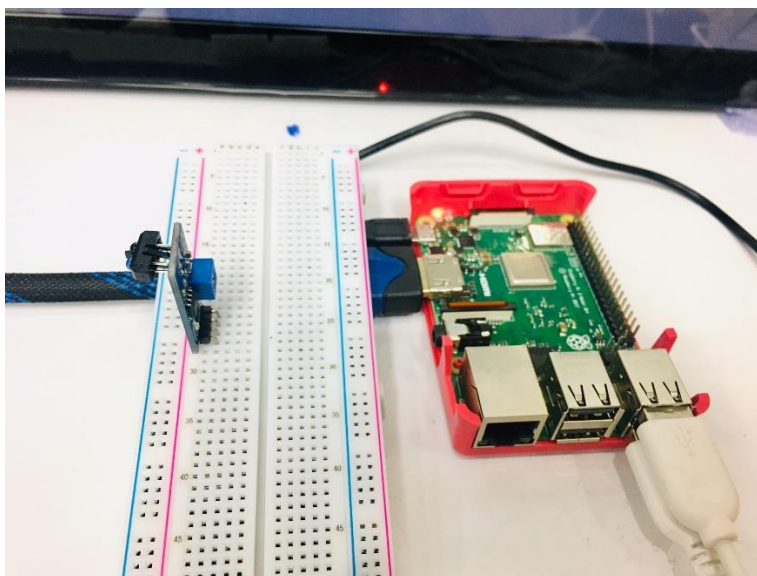


4. สาย Jumper

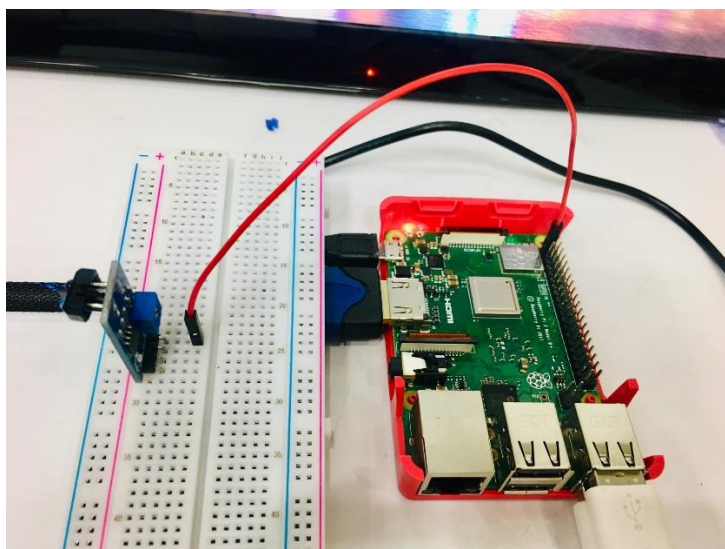


การต่อวงจร

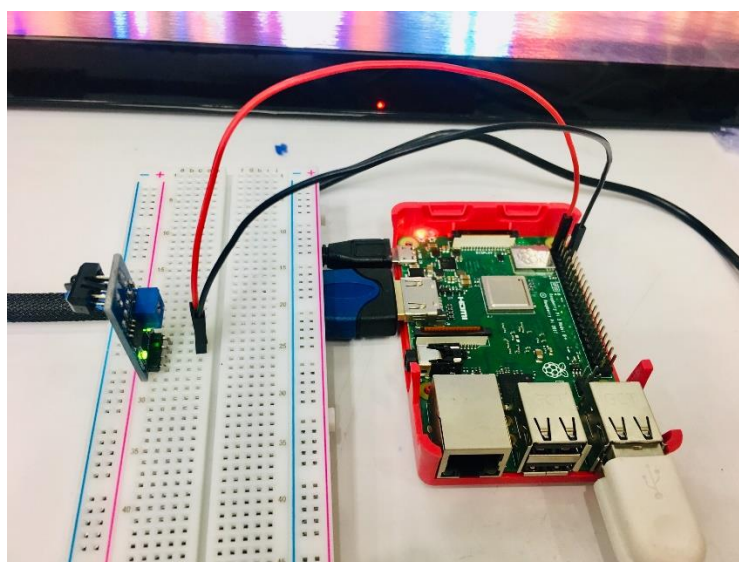
1. เสียบ IR Reflective Sensor บน Breadboard



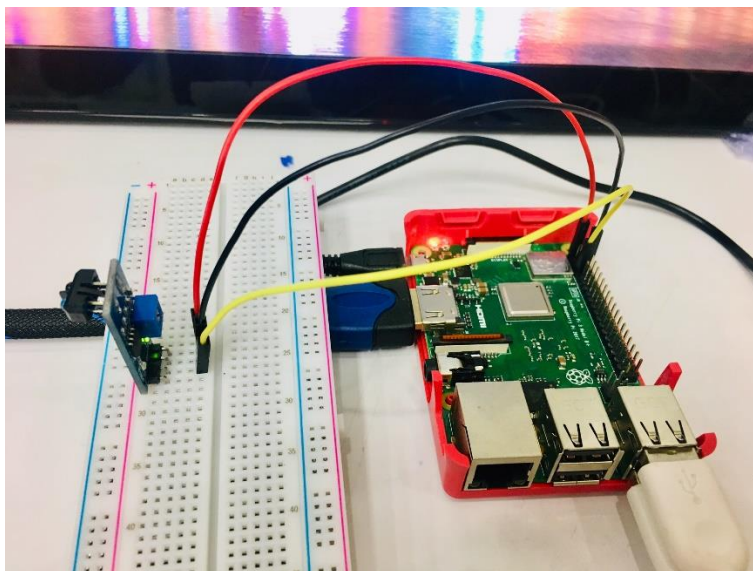
2. ใช้สาย Jumper เสียบลงบน Breadboard ในแถวเดียวกับขา VCC ของ IR Reflective Sensor ส่วนปลายสาย Jumper เสียบเข้ากับขา PIN1 (3V3) ของขา GPIO ของบอร์ด Raspberry Pi



3. ใช้สาย Jumper เสียบบน Breadboard แถวเดียวกับขา GND ของ IR Reflective Sensor ส่วนปลายสายของ Jumper เสียบเข้ากับขา PIN6 (GROUND) ของขา GPIO ของบอร์ด Raspberry Pi

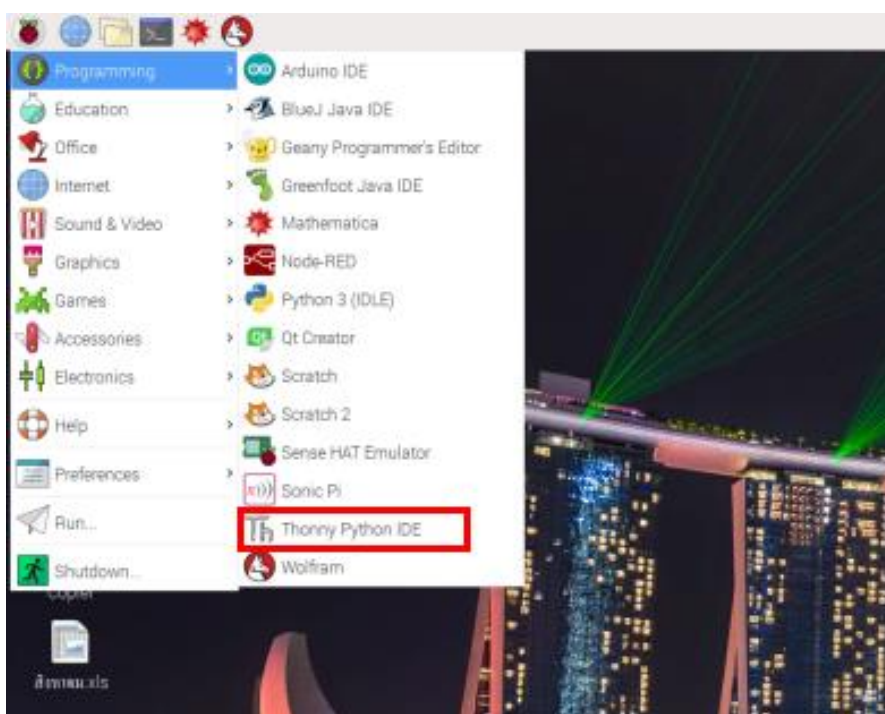


4. ใช้สาย Jumper เสียบบน Breadboard แถวเดียวกับขา D0 (Digital) ของ IR Reflective Sensor ส่วนปลายสาย Jumper เสียบเข้ากับขา PIN7 (GPIO4) ของขา GPIO ของบอร์ด Raspberry Pi



การเขียนโปรแกรมด้วยภาษาไพทอนเพื่อทดสอบอุปกรณ์

1. เปิดโปรแกรม Thonny Python IDE โดยไปที่ Pi > Programming > Thonny Python IDE

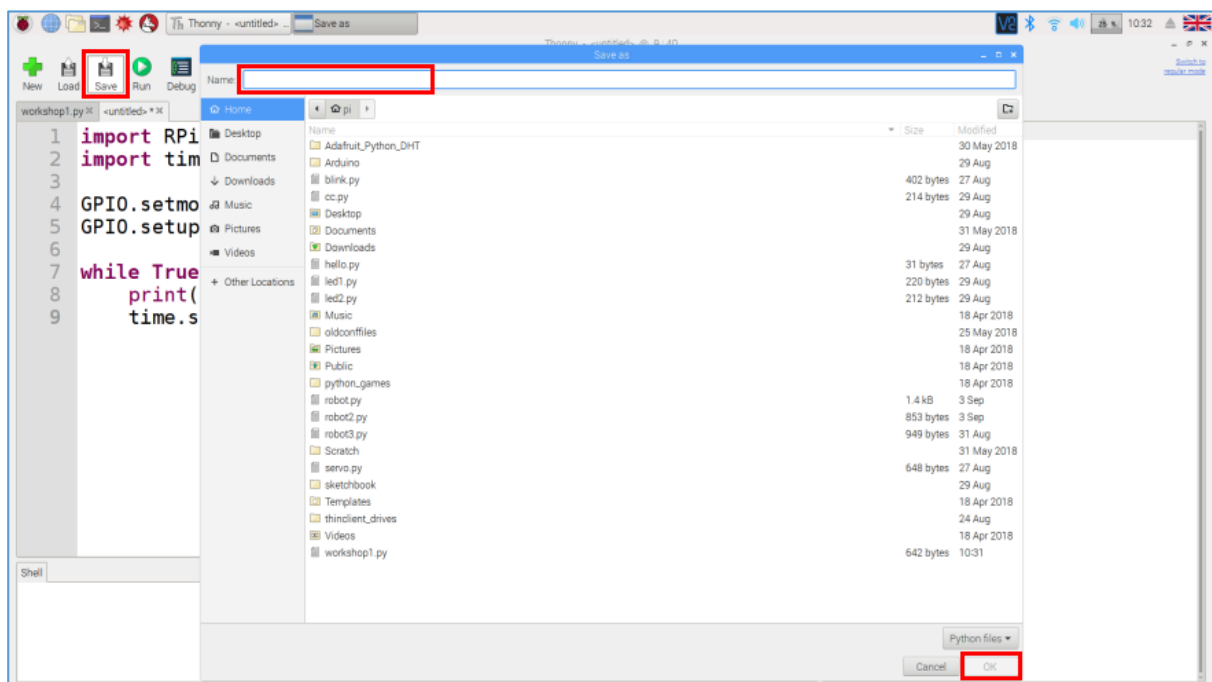


2. เขียนโค้ดคำสั่ง ดังนี้

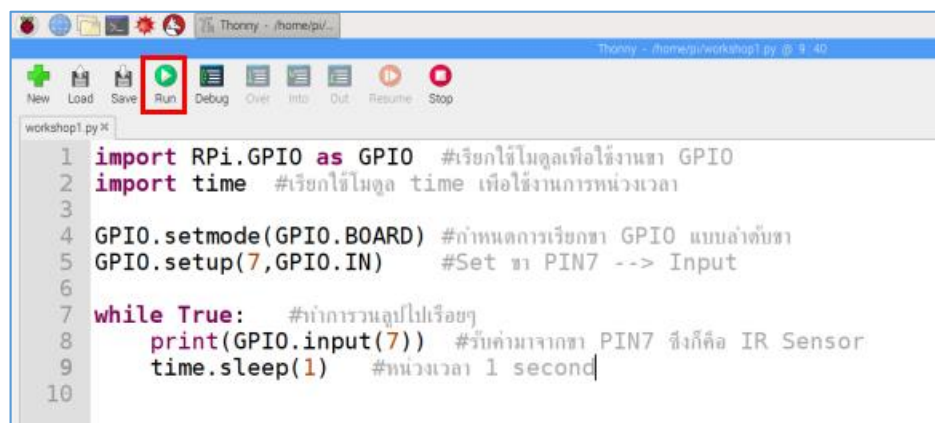
```
workshop1.py *
1 import RPi.GPIO as GPIO #เรียกใช้โมดูลเพื่อใช้งาน GPIO
2 import time #เรียกใช้โมดูล time เพื่อใช้งานการหน่วงเวลา
3
4 GPIO.setmode(GPIO.BOARD) #กำหนดการเรียก GPIO แบบลำดับ
5 GPIO.setup(7,GPIO.IN) #Set พิน PIN7 --> Input
6
7 while True: #ทำการวนลูปไปเรื่อย ๆ
8     print(GPIO.input(7)) #รับค่ามาจากพิน PIN7 ซึ่งก็คือ IR Sensor
9     time.sleep(1) #หน่วงเวลา 1 second
10
```

3. บันทึกไฟล์ โดยคลิกที่ Save ไฟล์จะบันทึกไว้ที่ /home/pi ให้ตั้งชื่อไฟล์ในช่อง Name แล้วคลิกปุ่ม

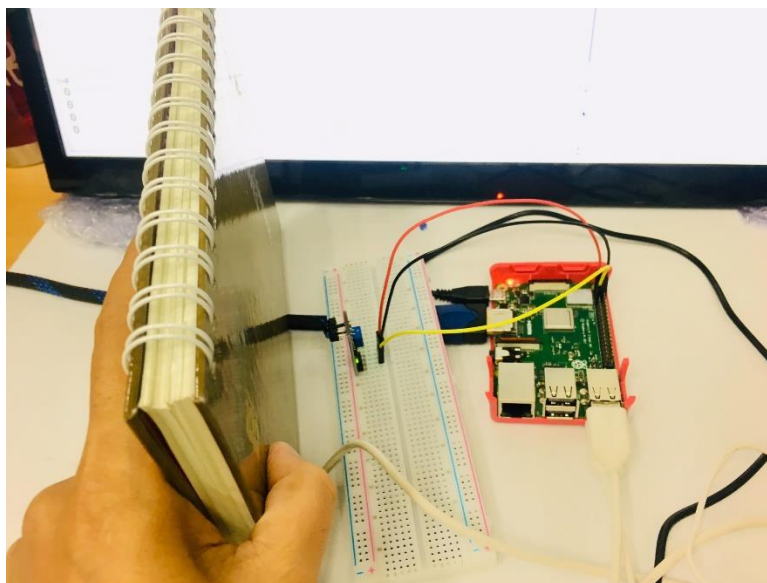
OK



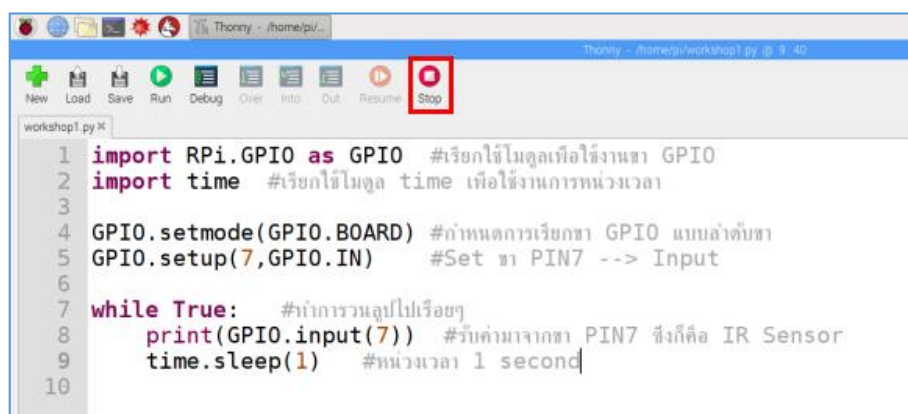
4. รันโปรแกรมเพื่อดูผลลัพธ์ โดยคลิกที่ปุ่ม Run



5. ทดลองใช้มือหรือวัตถุใดก็ได้บังด้านหน้า IR Reflective Sensor แล้วสังเกตค่าที่ได้ในช่อง Shell ของโปรแกรม Thonny Python IDE หลังจากนั้นเอามือหรือวัตถุออกจากด้านหน้า IR Reflective Sensor แล้วสังเกตค่าที่ได้



6. เมื่อทดสอบเสร็จแล้ว ต้องการหยุดการรันโปรแกรม ให้คลิกปุ่ม Stop



ผลลัพธ์ที่ได้

เมื่อมีวัตถุมาบังหน้า IR Reflective Sensor ไฟสถานะที่อยู่ด้านหลังเซนเซอร์จะติด ค่าที่แสดงออกมาจะมีค่าเท่ากับ 0 ถ้าไม่มีวัตถุมาบังหน้า IR Reflective Sensor ไฟสถานะที่อยู่ด้านหลังเซนเซอร์จะดับ ค่าที่แสดงออกมามีค่าเท่ากับ 1 ซึ่งสามารถนำ IR Reflective Sensor ไปประยุกต์ใช้ในการตรวจจับวัตถุได้

```

workshop1.py x
1 import RPi.GPIO as GPIO #เรียกใช้โมดูลเพื่อใช้งาน GPIO
2 import time #เรียกใช้โมดูล time เพื่อใช้งานการหน่วงเวลา
3
4 GPIO.setmode(GPIO.BOARD) #กำหนดการเรียก GPIO แบบลำดับขา
5 GPIO.setup(7,GPIO.IN) #Set ขา PIN7 --> Input
6
7 while True: #ทำการวนลูปไปเรื่อย ๆ
8     print(GPIO.input(7)) #รับค่ามาจากขา PIN7 ซึ่งก็คือ IR Sensor
9     time.sleep(1) #หน่วงเวลา 1 second
10

```

Shell

```

0
0
0
0
0
0
1
1
1
1
1
1
1

```

← เจอวัตถุ

← ไม่เจอวัตถุ

2. เซนเซอร์วัดอุณหภูมิและความชื้น Temperature and Humidity Sensor (DHT22)



เซนเซอร์วัดอุณหภูมิและความชื้นสัมผัส เป็นเซนเซอร์วัดอุณหภูมิและความชื้นสัมผัสแบบดิจิทัล และเชื่อมต่อด้วยสัญญาณเพียงเส้นเดียวแบบสองทิศทาง (bidirectional) ใช้แรงดันไฟเลี้ยงได้ในช่วง 3.3V ถึง 5.2V สามารถวัดค่าอุณหภูมิได้ในช่วง -40 ถึง 80°C ความละเอียดในการวัดอุณหภูมิและความชื้น คือ 0.5°C และ 0.1%RH

คุณสมบัติของ DHT22

- แรงดันไฟฟ้า: 3.3V – 6V DC
- เอาต์พุต: Single Bus
- ย่านการวัด: ความชื้น 0-100% RH / อุณหภูมิ -40 °C ถึง 125 °C
- ระยะเวลาในการส่งข้อมูล: ประมาณ 2 วินาที

ฝึกทักษะการทดลองกิจกรรมที่ 2 เรียนรู้การใช้งาน DHT22

อุปกรณ์ที่ใช้สำหรับการทดลอง

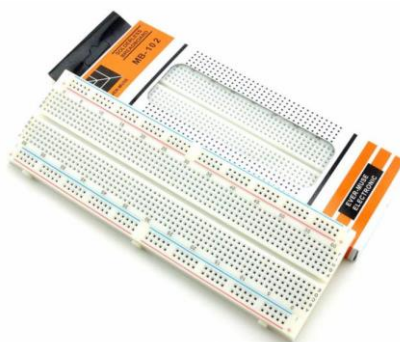
1. บอร์ด Raspberry Pi



2. เซนเซอร์ DHT22



3. Breadboard

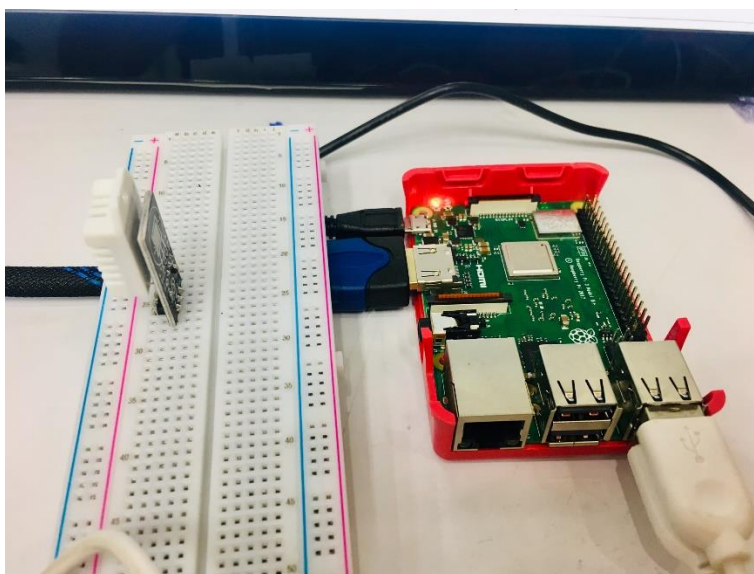


4. สาย Jumper

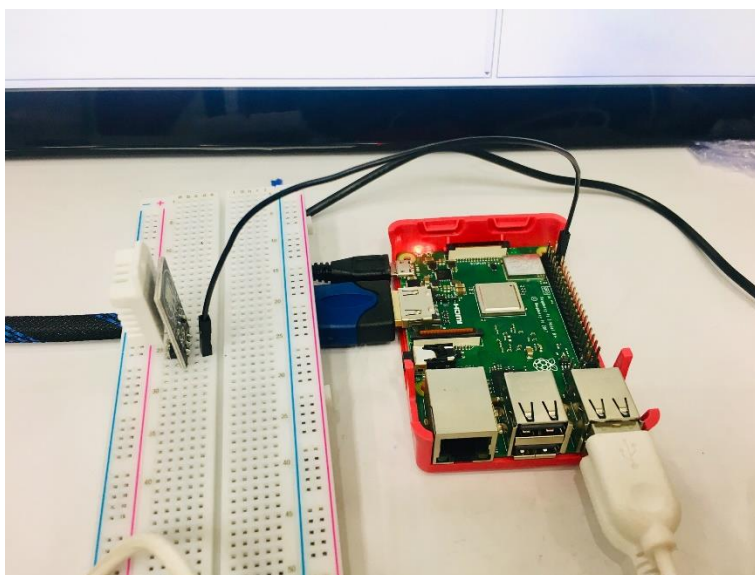


การต่อวงจร

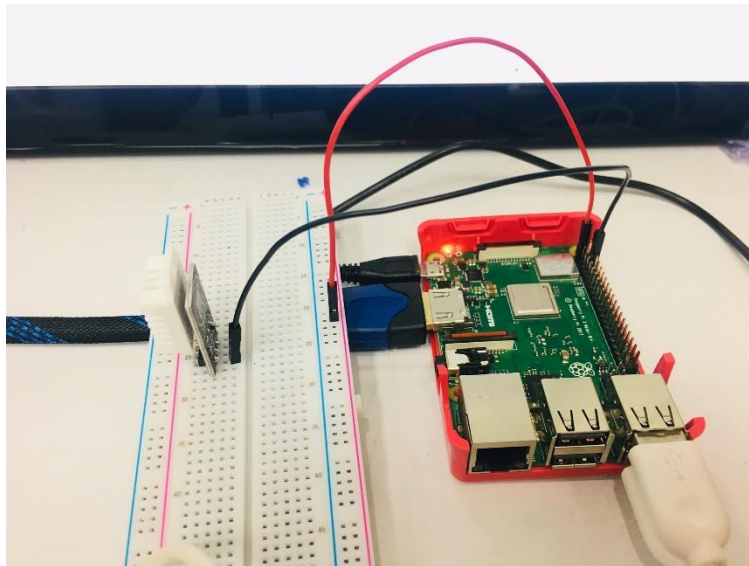
1. เสียบเซนเซอร์ DHT22 บน Breadboard



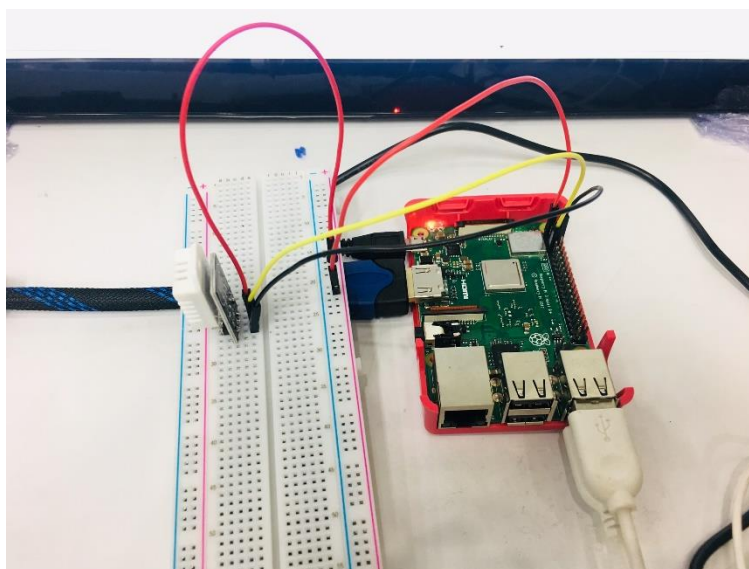
2. เสียบสาย Jumper ต่อจากขาลบ – ของ DHT22 ไปยัง PIN6 (GROUND) ของขา GPIO



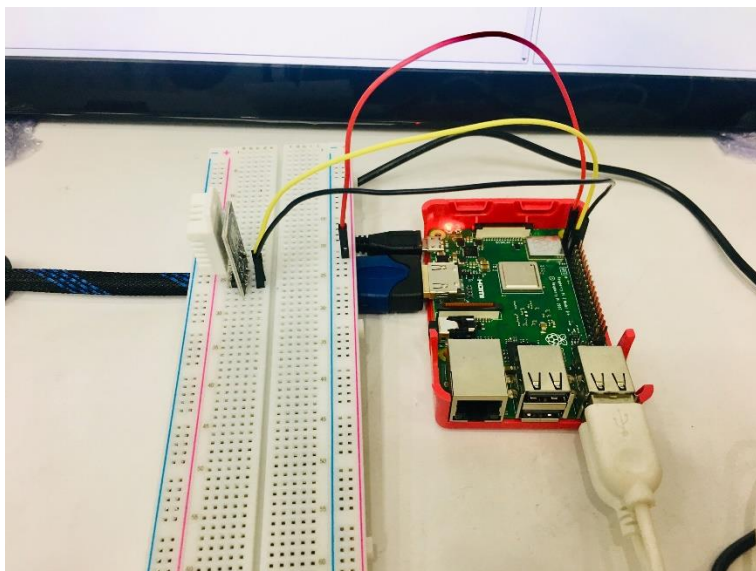
3. เสียบสาย Jumper จากขา PIN1 (3V3) ของขา GPIO ของบอร์ด Raspberry Pi ต่อมายังช่อง + (แถวแนวตั้งสีแดง) บน Breadboard



4. เสียบสาย Jumper ต่อจากขา + ของ DHT22 ต่อมายังช่อง + (แถวแนวตั้งสีแดง) บน Breadboard

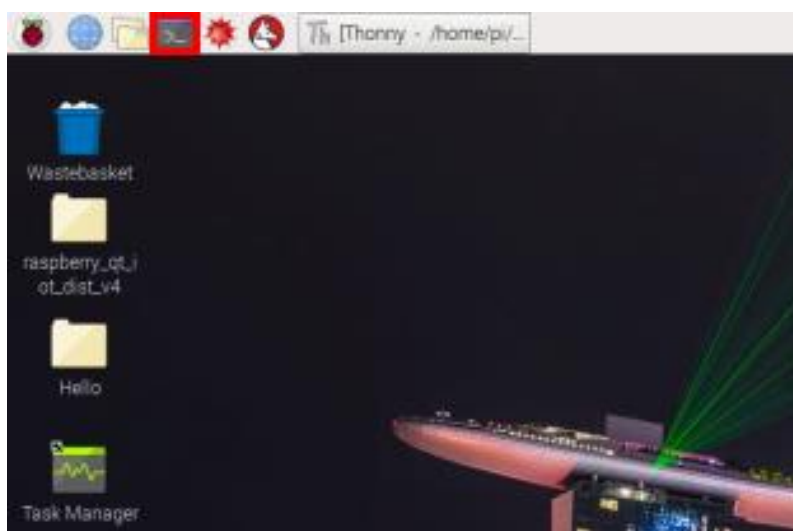


5. เสียบสาย Jumper ต่อจากขา out ของ DHT22 ไปยัง PIN7 (GPIO4) ของขา GPIO

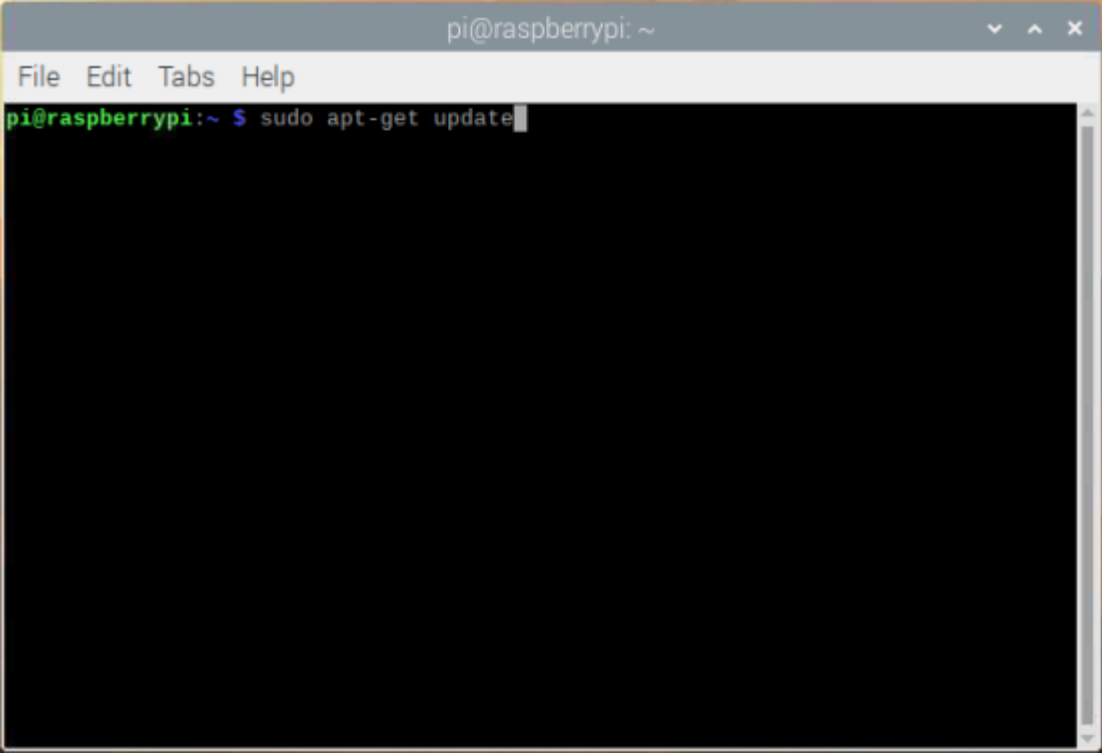


การเขียนโปรแกรมด้วยภาษาไพทอนเพื่อทดสอบอุปกรณ์

1. ก่อนการเขียนโปรแกรมเพื่อใช้งาน DHT22 จะต้องทำการดาวน์โหลดและติดตั้งไลบรารี Adafruit_DHT เสียก่อน โดยเปิดโปรแกรม Terminal ขึ้นมา



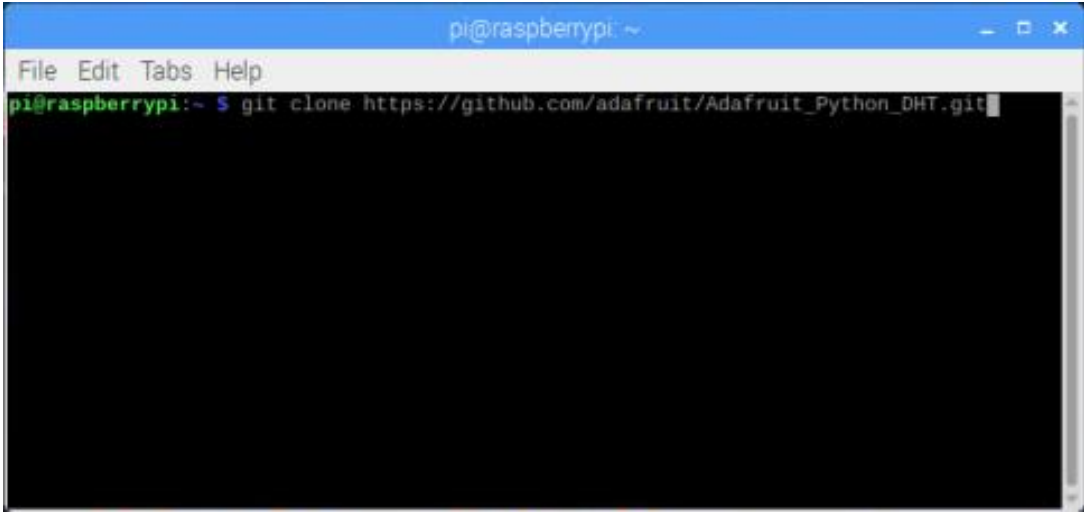
2. ก่อนจะดาวน์โหลดไลบรารี ให้ทำการอัปเดตซอฟต์แวร์ทั้งระบบก่อน โดยพิมพ์คำสั่ง `sudo apt-get update`



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ sudo apt-get update
```

3. ดาวน์โหลดไลบรารี Adafruit_DHT โดยพิมพ์คำสั่ง

`git clone https://github.com/adafruit/Adafruit_Python_DHT.git`



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ git clone https://github.com/adafruit/Adafruit_Python_DHT.git
```

4. เข้าไปในโฟลเดอร์ Adafruit_Python_DHT โดยพิมพ์คำสั่ง `cd Adafruit_Python_DHT`

```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ sudo apt-get update
Get:1 http://archive.raspberrypi.org/debian buster InRelease [25.2 kB]
Get:2 http://raspbian.raspberrypi.org/raspbian buster InRelease [15.0 kB]
Get:3 http://archive.raspberrypi.org/debian buster/main armhf Packages [229 kB]
Get:4 http://raspbian.raspberrypi.org/raspbian buster/main armhf Packages [13.0 MB]
Get:5 http://raspbian.raspberrypi.org/raspbian buster/non-free armhf Packages [103 kB]
Fetched 13.4 MB in 12s (1,093 kB/s)
Reading package lists... Done
pi@raspberrypi:~ $ git clone https://github.com/adafruit/Adafruit_Python_DHT.git
Cloning into 'Adafruit_Python_DHT'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 322 (delta 0), reused 0 (delta 0), pack-reused 317
Receiving objects: 100% (322/322), 99.37 KiB | 225.00 KiB/s, done.
Resolving deltas: 100% (171/171), done.
pi@raspberrypi:~ $ cd Adafruit_Python_DHT

```

5. ติดตั้งไลบรารี โดยพิมพ์คำสั่ง `sudo python setup.py install`

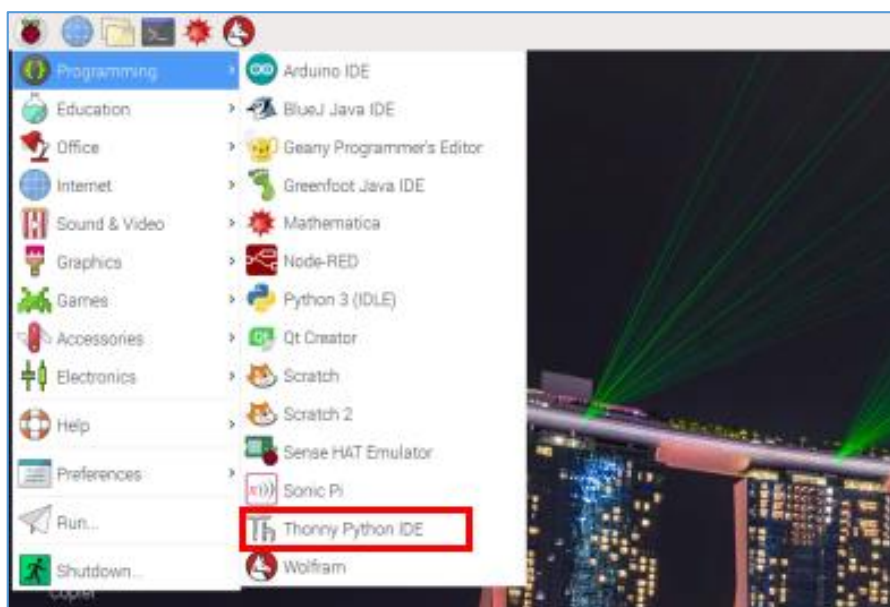
```

pi@raspberrypi: ~/Adafruit_Python_DHT
File Edit Tabs Help
pi@raspberrypi:~ $ sudo apt-get update
Get:1 http://archive.raspberrypi.org/debian buster InRelease [25.2 kB]
Get:2 http://raspbian.raspberrypi.org/raspbian buster InRelease [15.0 kB]
Get:3 http://archive.raspberrypi.org/debian buster/main armhf Packages [229 kB]
Get:4 http://raspbian.raspberrypi.org/raspbian buster/main armhf Packages [13.0 MB]
Get:5 http://raspbian.raspberrypi.org/raspbian buster/non-free armhf Packages [103 kB]
Fetched 13.4 MB in 12s (1,093 kB/s)
Reading package lists... Done
pi@raspberrypi:~ $ git clone https://github.com/adafruit/Adafruit_Python_DHT.git
Cloning into 'Adafruit_Python_DHT'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 322 (delta 0), reused 0 (delta 0), pack-reused 317
Receiving objects: 100% (322/322), 99.37 KiB | 225.00 KiB/s, done.
Resolving deltas: 100% (171/171), done.
pi@raspberrypi:~ $ cd Adafruit_Python_DHT
pi@raspberrypi:~/Adafruit_Python_DHT $ sudo python setup.py install

```

6. ติดตั้งเสร็จแล้ว ให้ปิดโปรแกรม Terminal

7. เปิดโปรแกรม Thonny Python IDE โดยไปที่ Pi > Programming > Thonny Python IDE



8. เขียนโค้ดคำสั่ง ดังนี้

```

1 import Adafruit_DHT
2 import time
3
4 while True:
5     h,t = Adafruit_DHT.read_retry(22,4) #22 is version of DHT & 4 is GPIO4 at GPIO Pin
6     print("Humidity = %.2f && Temperature = %.2f" %(h,t))
7     time.sleep(2)
8

```

9. บันทึกไฟล์และรันโปรแกรม โดยคลิกปุ่ม Run



ผลลัพธ์ที่ได้

ในแต่ละครั้งเซนเซอร์วัดอุณหภูมิและความชื้นจะใช้เวลาในการตรวจวัด 2 วินาที การเขียนโปรแกรมจึงจำเป็นต้องทำการหน่วงเวลาไว้ทุกๆ 2 วินาที จากการรันโค้ดคำสั่งจะเห็นได้ว่าในทุกๆ 2 วินาทีนั้นเซนเซอร์จะวัดค่าและแสดงค่าออกมาเป็น “Temperature = ค่าอุณหภูมิที่วัดได้ humidity = ค่าความชื้นที่วัดได้” ซึ่งจะแสดงค่าเป็นทศนิยม 2 ตำแหน่ง

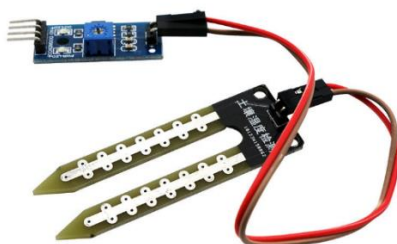

```

Shell
Python 3.5.3 (/usr/bin/python3)
>>> %Run workshop2.py

Humidity = 52.20  &&  Temperature = 25.90
Humidity = 52.60  &&  Temperature = 25.90
Humidity = 52.70  &&  Temperature = 25.90
Humidity = 52.70  &&  Temperature = 25.90
Humidity = 52.70  &&  Temperature = 25.90
Humidity = 52.60  &&  Temperature = 25.90
Humidity = 52.60  &&  Temperature = 25.90
Humidity = 52.50  &&  Temperature = 25.90
Humidity = 52.50  &&  Temperature = 25.90
Humidity = 52.50  &&  Temperature = 25.90
Humidity = 52.40  &&  Temperature = 25.90
Humidity = 52.40  &&  Temperature = 25.90

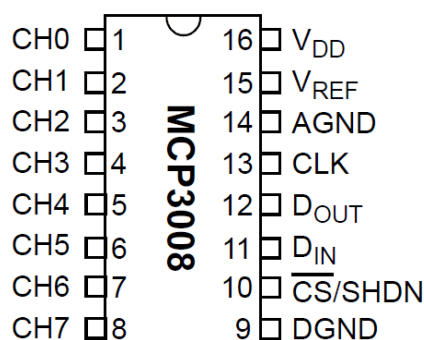
```

3. เซนเซอร์วัดความชื้นในดิน Soil Moisture Sensor



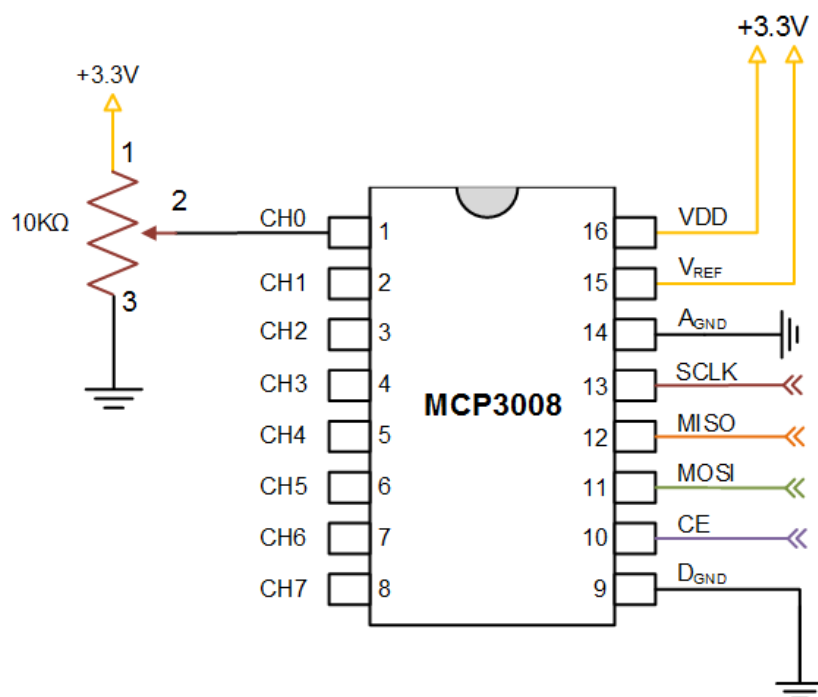
เซนเซอร์วัดความชื้นในดิน ใช้วัดความชื้นในดินหรือใช้เป็นเซนเซอร์น้ำ สามารถต่อใช้งานกับไมโครคอนโทรลเลอร์โดยใช้อินพุตแอนาล็อกอ่านค่าความชื้นหรือเลือกใช้สัญญาณดิจิตอลที่ส่งมาจากโมดูลสามารถปรับความไวได้ด้วยการปรับ Trimpot ใช้หลักการตรวจสอบประจุของวัสดุ ถ้ามีค่าประจุมากแสดงว่าชื้นมาก เนื่องจากใช้หลักการของการวัดประจุ ดังนั้นแผ่นเซนเซอร์จึงไม่ต้องสัมผัสกับดินหรือวัสดุโดยตรง จึงทนทานและแม่นยำกว่า โมดูลวัดความชื้นในดินนี้จะให้ค่าออกเป็นค่า Analog จึงต้องใช้คู่กับชิป MCP3008 ซึ่งเป็นชิปแปลงสัญญาณ Analog เป็น Digital

ลักษณะของชิป MCP3008



ขาของ MCP3008

การต่อขาวงจรกับขา GPIO ของบอร์ด



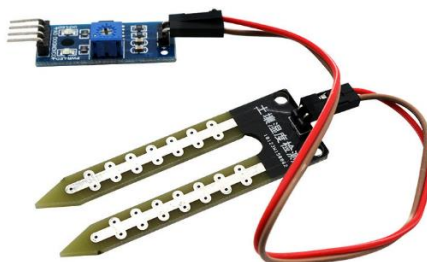
ฝึกทักษะการทดลองกิจกรรมที่ 3 เรียนรู้การใช้งาน Soil Moisture Sensor

อุปกรณ์ที่ใช้สำหรับการทดลอง

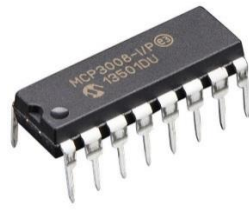
1. บอร์ด Raspberry Pi



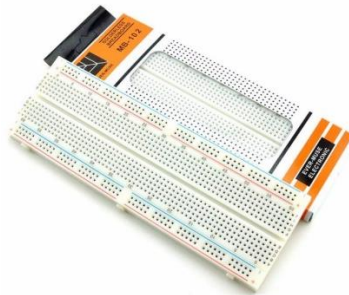
2. เซนเซอร์วัดความชื้นในดิน



3. ชิป MCP3008



4. Breadboard

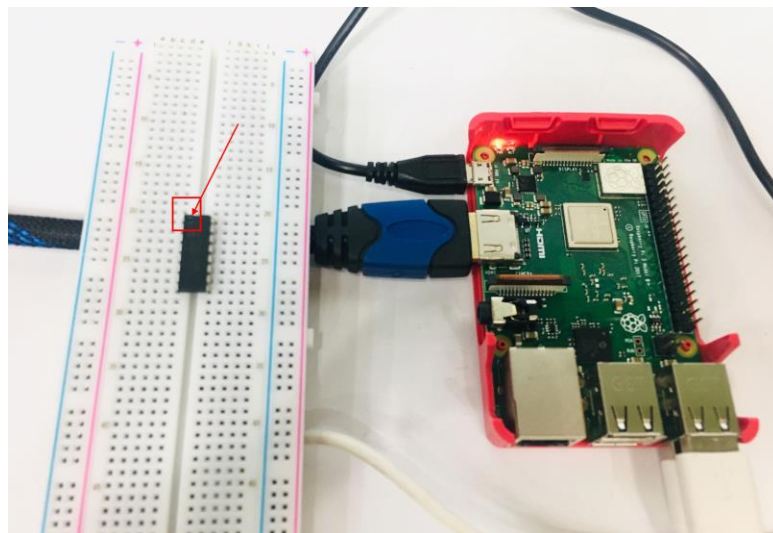


5. สาย Jumper

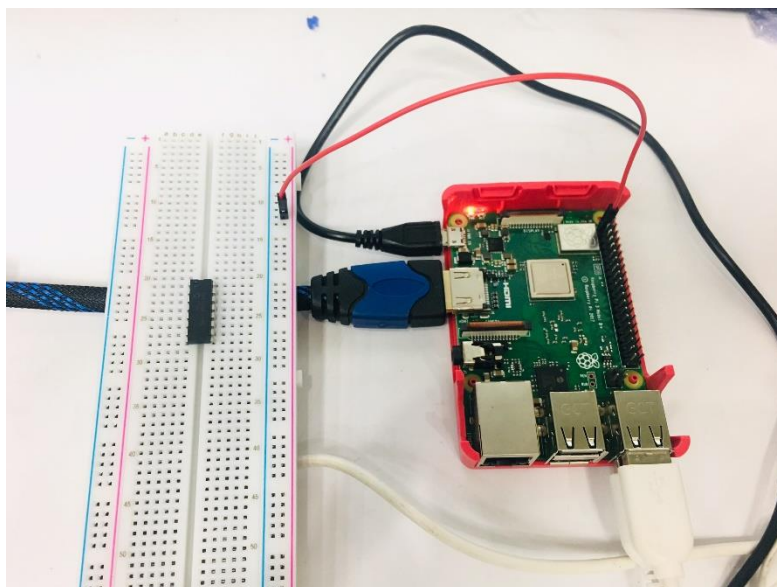


การต่อวงจร

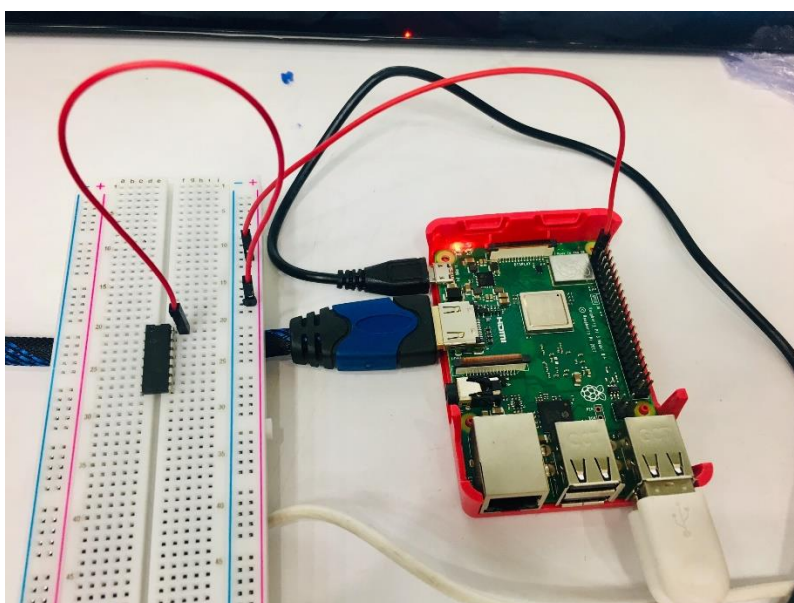
1. เสียบชิป MCP3008 ไว้ตรงกลาง Breadboard โดยสังเกตที่ตัวชิปจะมีจุดกลมๆ บนชิป ให้วางอยู่ด้านบน



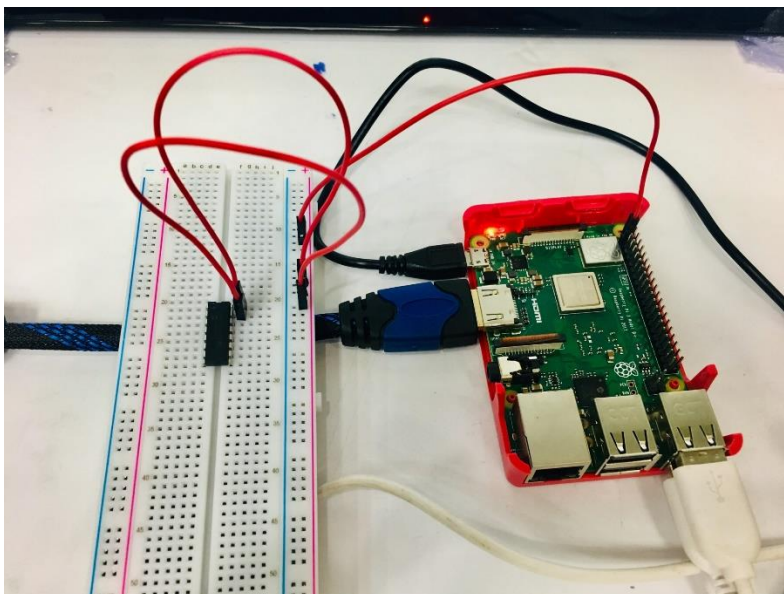
2. เสียบสาย Jumper จากขา PIN1 (3V3) ของขา GPIO ไปยังช่องใดก็ได้ในแถว + (สีแดง) ของ Breadboard



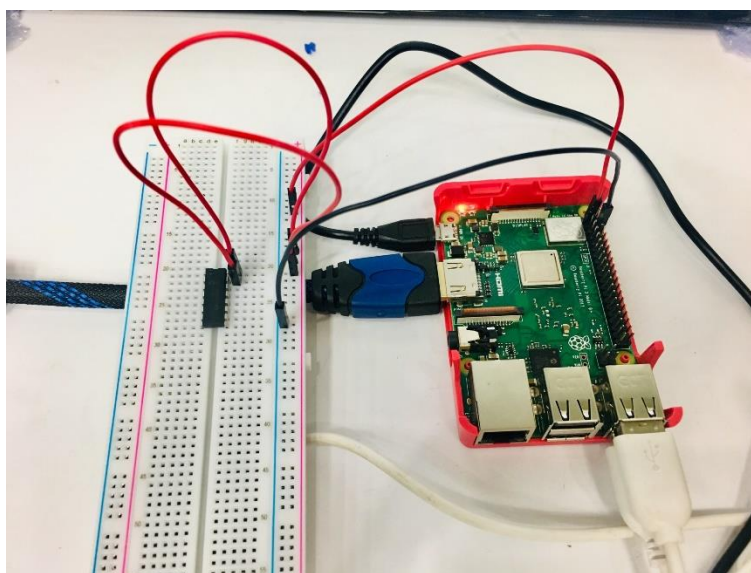
3. เสียบสาย Jumper จากขา VDD (ขาที่ 1 ขวามือ) ของชิป MCP3008 ไปยังแถว + (สีแดง) ของ Breadboard



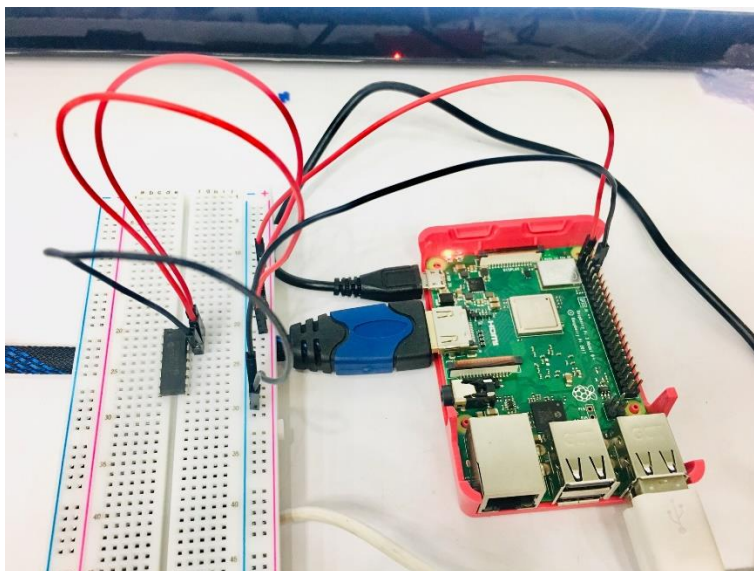
4. เสียบสาย Jumper จากขา VREF (ขาที่ 2 ขวามือ) ของชิป MCP3008 ไปยังแถว + (สีแดง) ของ Breadboard



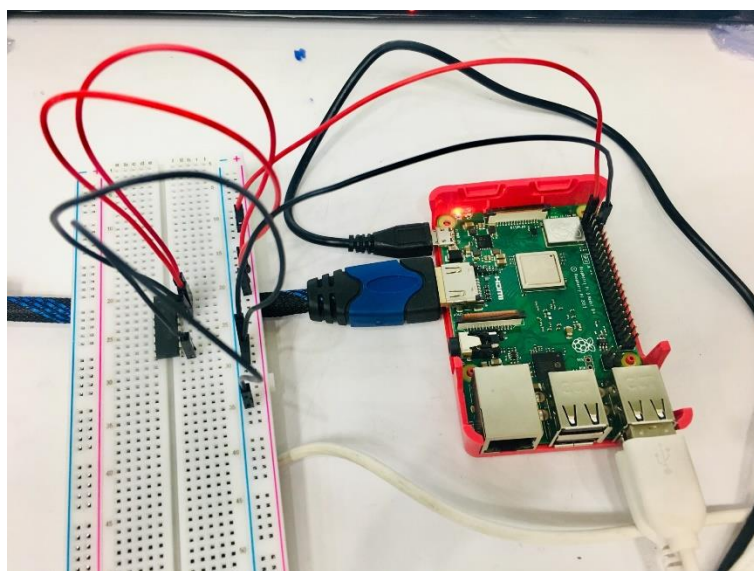
5. เสียบสาย Jumper จากขา PIN6 (Ground) ของขา GPIO ไปยังแถว - (สีฟ้า) ของ Breadboard



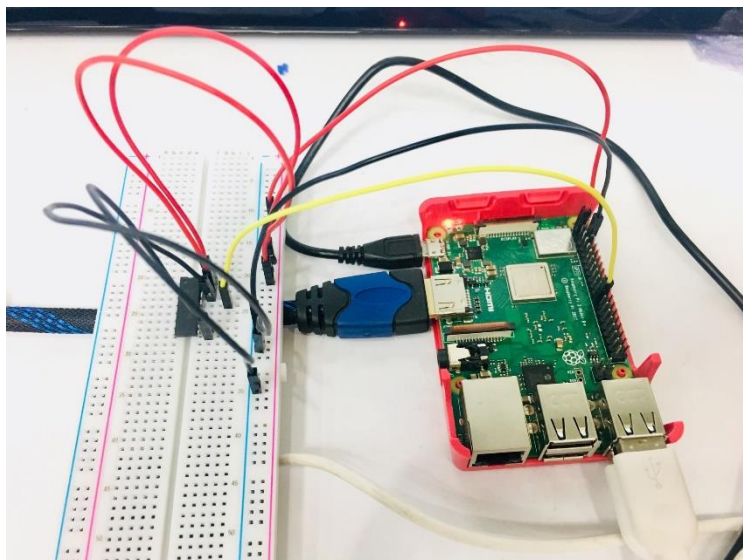
6. เสียบสาย Jumper จากขา AGND (ขาที่ 3 ขาวมือ) ของชิป MCP3008 ไปยังแถว - (สีฟ้า) ของ Breadboard



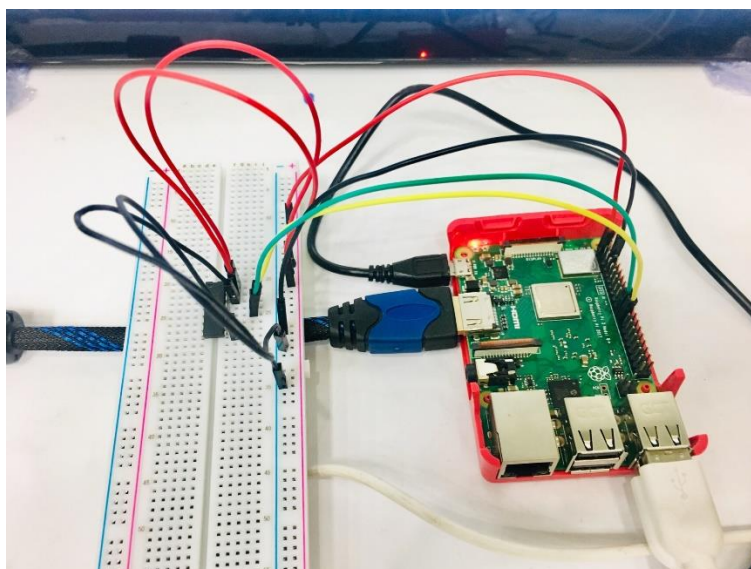
7. เสียบสาย Jumper จากขา DGND (ขาที่ 8 ขวามือ) ของชิป MCP3008 ไปยังแถว - (สีฟ้า) ของ Breadboard



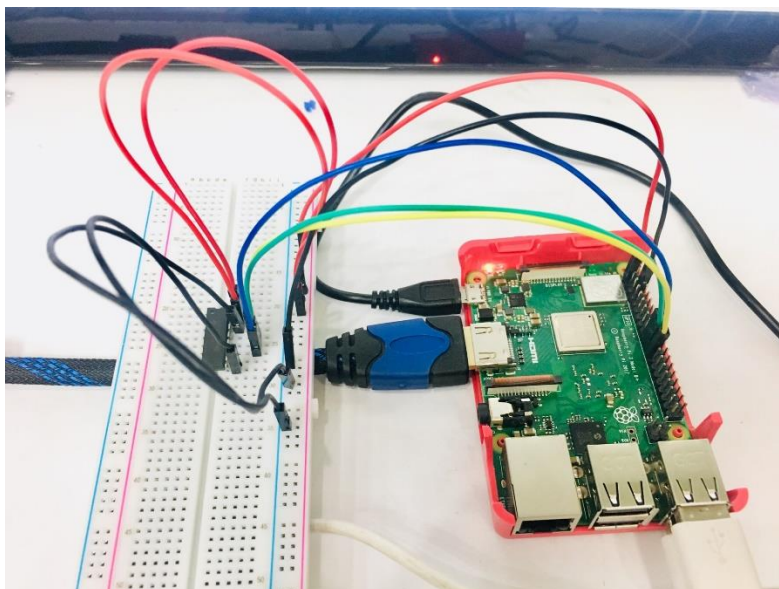
8. เสียบสาย Jumper จากขา CLK (ขาที่ 4 ขวามือ) ของชิป MCP3008 ไปยัง PIN23 (GPIO11 SP10_SCLK) ของขา GPIO



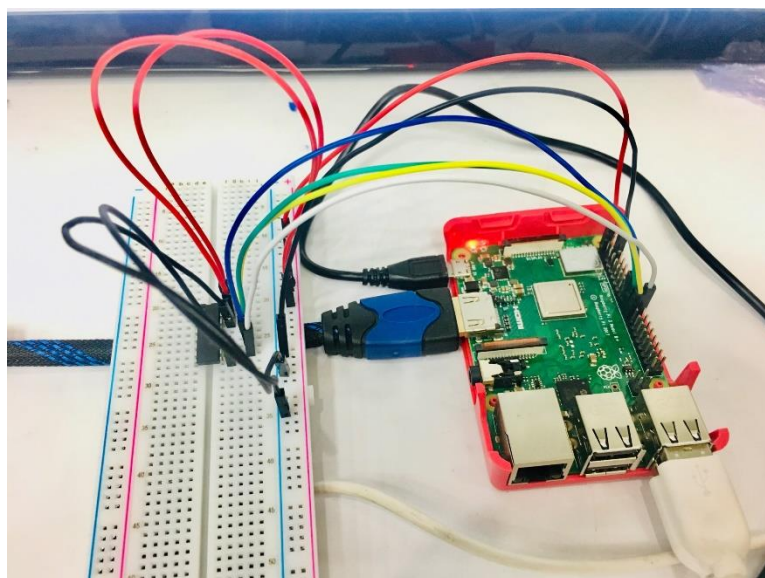
9. เสียบสาย Jumper จากขา DOUT (ขาที่ 5 ขวามือ) ของชิป MCP3008 ไปยัง PIN21 (GPIO9 SP10_MISO) ของขา GPIO



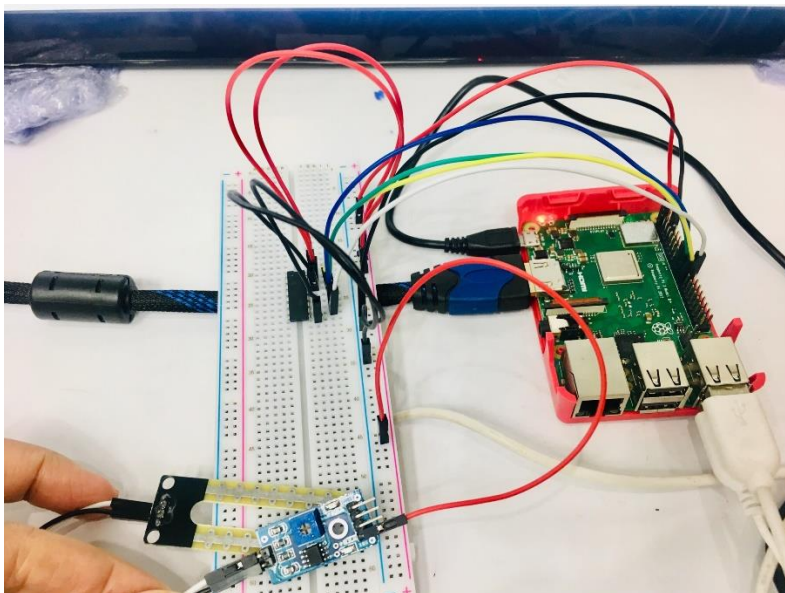
10. เสียบสาย Jumper จากขา DIN (ขาที่ 6 ขวามือ) ของชิป MCP3008 ไปยัง PIN19 (GPIO10 SP10_MOSI) ของขา GPIO



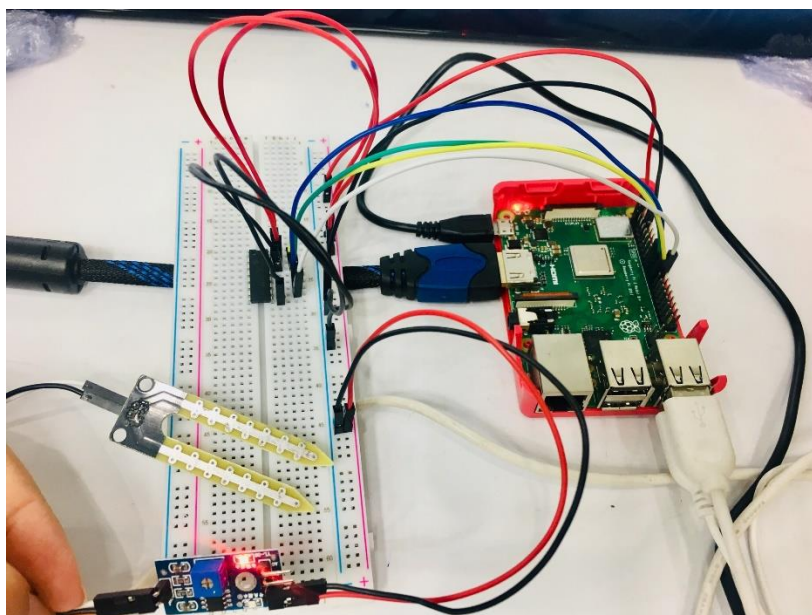
11. เสียบสาย Jumper จากขา CS/SHDN (ขาที่ 7 ขวามือ) ของชิป MCP3008 ไปยัง PIN24 (GPIO8 SP10_CE0_N) ของขา GPIO



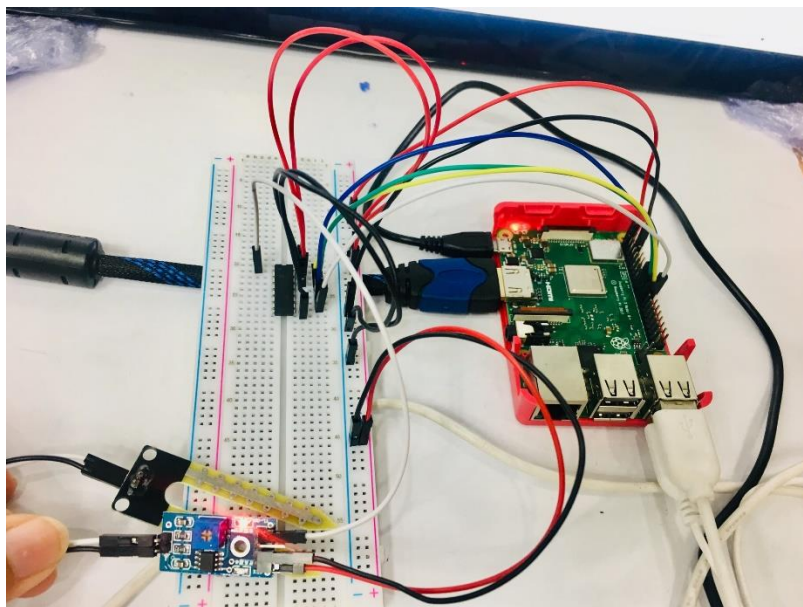
12. เสียบสาย Jumper จากขา VCC ของเซนเซอร์วัดความชื้นในดิน ไปยังแถว + (สีแดง) ของ Breadboard



13. เสียบสาย Jumper จากขา GND ของเซนเซอร์วัดความชื้นในดิน ไปยังแถว - (สีฟ้า) ของ Breadboard (เมื่อเสียบ Ground จะเห็นว่าไฟสถานะของเซนเซอร์จะติด)



14. เสียบสาย Jumper จากขา A0 ของเซนเซอร์วัดความชื้นในดิน ไปยังขา CH0 ของชิป MCP3008



การเขียนโปรแกรมด้วยภาษาไพทอนเพื่อทดสอบอุปกรณ์

เนื่องจาก Raspberry Pi ไม่มีความสามารถในการรับค่าแบบ Analog โดยตรง (รับได้เฉพาะ Digital In/Out ซึ่งมีค่า 0 และ 1 เท่านั้น) ซึ่งไม่เหมือนกับ Arduino Board ซึ่งรับค่าได้ทั้งแบบ Digital และ Analog ดังนั้นจึงมีนาชิป MCP3008 มาใช้ในการแปลงสัญญาณ Analog ให้เป็น Digital ค่าที่อ่านได้จะมีค่าระหว่าง 0 – 1023

1. ในการใช้งานชิป MCP3008 จะต้องใช้ไลบรารีเพิ่มเติม ซึ่งจะต้องดาวน์โหลดและติดตั้งไลบรารี gpiozero โดยเปิด Terminal ขึ้นมา แล้วพิมพ์คำสั่งดังนี้

```
sudo apt-get update
```

```
sudo apt-get install python-pip python3-pip
```

```
sudo pip install gpiozero
```

```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ sudo apt-get update
```

```
pi@raspberrypi: ~  
File Edit Tabs Help  
Get:1 http://archive.raspberrypi.org/debian stretch InRelease [25.4 kB]  
Get:2 http://raspbian.raspberrypi.org/raspbian stretch InRelease [15.0 kB]  
Get:3 http://archive.raspberrypi.org/debian stretch/main armhf Packages [221 kB]  
Get:4 http://raspbian.raspberrypi.org/raspbian stretch/main armhf Packages [11.7  
MB]  
Get:5 http://raspbian.raspberrypi.org/raspbian stretch/contrib armhf Packages [5  
6.9 kB]  
Fetched 12.0 MB in 28s (422 kB/s)  
Reading package lists... Done  
pi@raspberrypi:~ $ sudo apt-get install python-pip python3-pip  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
python-pip is already the newest version (9.0.1-2+rpt2).  
python3-pip is already the newest version (9.0.1-2+rpt2).  
The following packages were automatically installed and are no longer required:  
  arduino-core avr-libc avrdude binutils-avr coinor-libipopt1v5  
  extra-xdg-menus gcc-avr libdirectfb-1.2-9 libftdi1 libiso9660-8 libjna-java  
  libjna-jni libmumps-seq-4.10.0 libraw15 librx-tx-java libvcdinfo0 libvllcore8  
  lxkeymap python-cairo python-gobject python-gobject-2 python-gtk2  
  python-xklavier realpath  
Use 'sudo apt autoremove' to remove them.  
0 upgraded, 0 newly installed, 0 to remove and 68 not upgraded.  
pi@raspberrypi:~ $
```



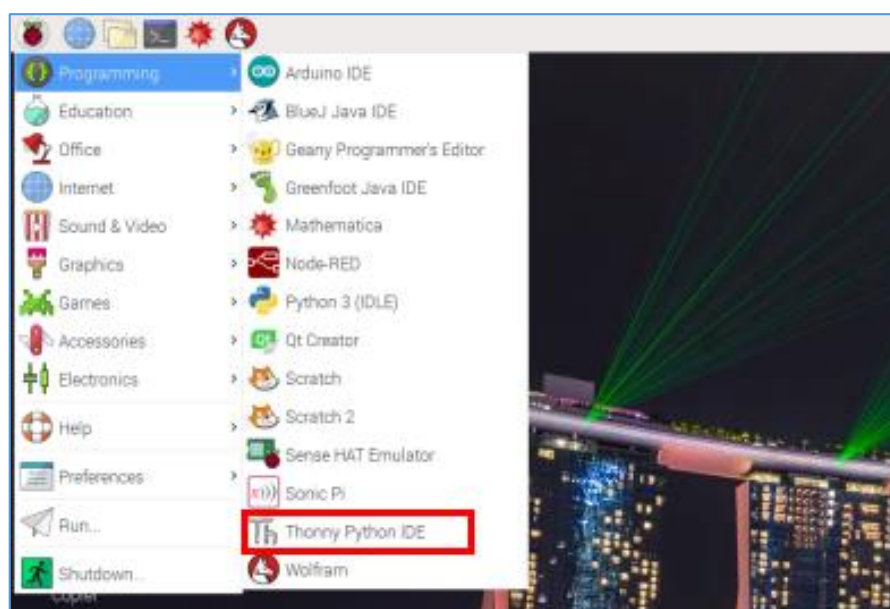
```

pi@raspberrypi:~ $ sudo apt-get update
Get:1 http://archive.raspberrypi.org/debian stretch InRelease [25.4 kB]
Get:2 http://rasbian.raspberrypi.org/raspbian stretch InRelease [15.0 kB]
Get:3 http://archive.raspberrypi.org/debian stretch/main armhf Packages [221 kB]
Get:4 http://rasbian.raspberrypi.org/raspbian stretch/main armhf Packages [11.7 MB]
Get:5 http://rasbian.raspberrypi.org/raspbian stretch/contrib armhf Packages [56.9 kB]
Fetched 12.0 MB in 28s (422 kB/s)
Reading package lists... Done
pi@raspberrypi:~ $ sudo apt-get install python-pip python3-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
python-pip is already the newest version (9.0.1-2+rpt2).
python3-pip is already the newest version (9.0.1-2+rpt2).
The following packages were automatically installed and are no longer required:
  arduino-core avr-libc avrdude binutils-avr coinor-libipopt1v5
  extra-xdg-menus gcc-avr libdirectfb-1.2-9 libftdi1 libiso9660-8 libjna-java
  libjna-jni libmumps-seq-4.10.0 libraw15 librtx-java libvcdinfo0 libvltcore8
  lxkeymap python-cairo python-gobject python-gobject-2 python-gtk2
  python-xklavier realpath
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 68 not upgraded.
pi@raspberrypi:~ $ sudo pip install gpiozero

```

2. ติดตั้งเสร็จแล้ว ให้ปิดโปรแกรม Terminal

3. เปิดโปรแกรม Thonny Python IDE โดยไปที่ Pi > Programming > Thonny Python IDE



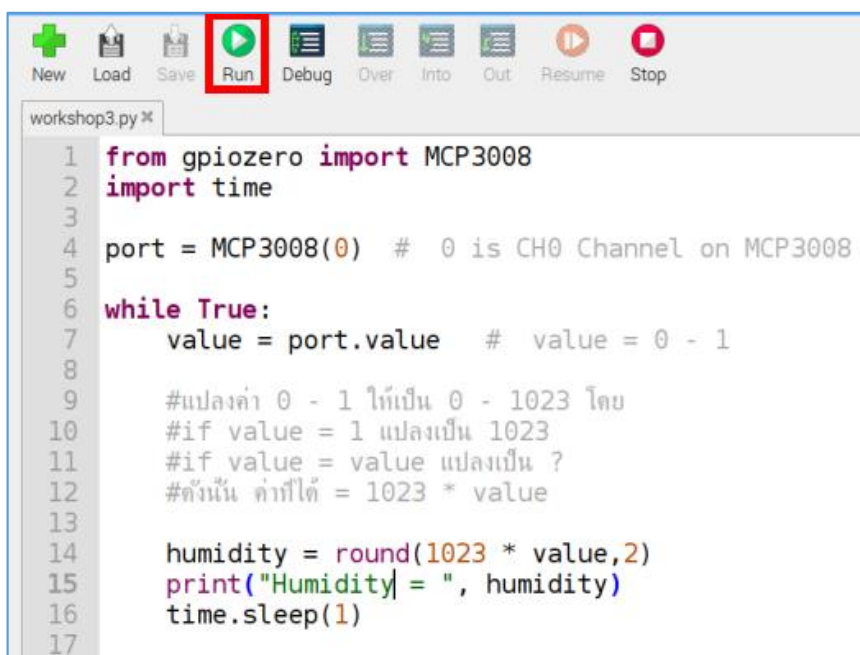
4. เขียนโปรแกรม ดังนี้

```

1 from gpiozero import MCP3008
2 import time
3
4 port = MCP3008(0) # 0 is CH0 Channel on MCP3008
5
6 while True:
7     value = port.value # value = 0 - 1
8
9     #แปลงค่า 0 - 1 ให้เป็น 0 - 1023 โดย
10    #if value = 1 แปลงเป็น 1023
11    #if value = value แปลงเป็น ?
12    #ดังนั้น ค่าที่ได้ = 1023 * value
13
14    humidity = round(1023 * value,2)
15    print("Humidity = ", humidity)
16    time.sleep(1)

```

5. บันทึกไฟล์และรันโปรแกรม โดยคลิกปุ่ม Run

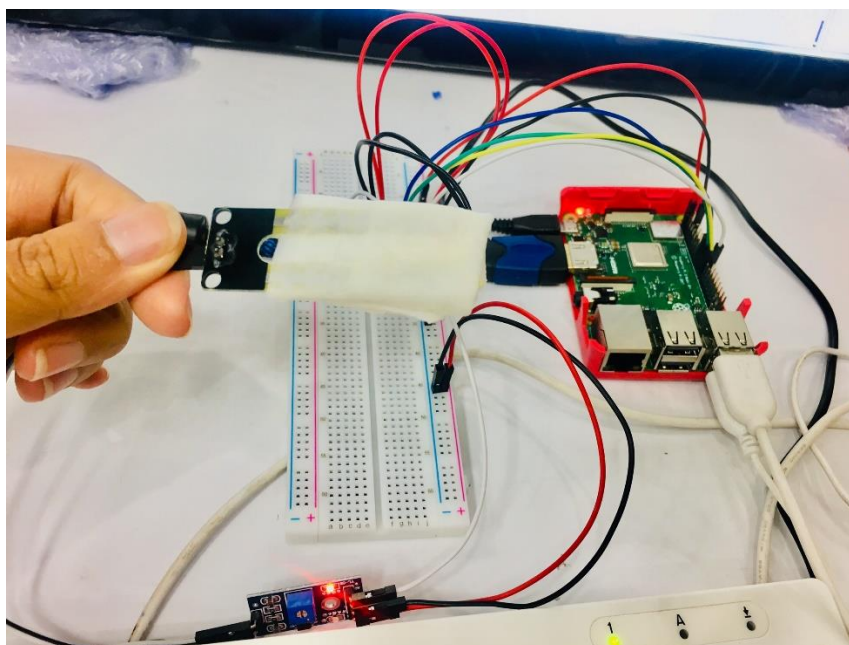


ผลลัพธ์ที่ได้

โดยปกติแล้วตัวเซนเซอร์วัดความชื้นในดินจะวัดค่าออกมาเป็น Digital คือ มีค่าอยู่ระหว่าง 0 – 1 โดยถ้าไม่มีความชื้นจะวัดค่าออกมาได้เท่ากับ 1 หากมีความชื้นจะส่งค่าออกมาเป็น 0 แต่ในการนำไปใช้จริง จำเป็นที่จะต้องนำค่าข้อมูลที่มากกว่า 0 และ 1 จึงจำเป็นตัวแปลงสัญญาณ นั่นก็คือ ชิพ MCP3008 ซึ่งในที่นี่ได้ต่อขา A0 จากเซนเซอร์ไปยังขา CH0 ของชิพ MCP3008 เพื่อส่งค่าข้อมูลจากเซนเซอร์ไปแปลงที่ชิพ ก่อนจะส่งข้อมูลไปยังขา GPIO

เมื่อทำการแปลงข้อมูลผ่านชิป MCP3008 จะเห็นว่าค่าที่ได้จะอยู่ระหว่าง 0.00 – 1.00 ซึ่งได้ค่าตัวเลขเป็นทศนิยมหลายตำแหน่ง หากต้องการเปลี่ยนจากค่าเลขทศนิยม 0.00 – 1.00 ไปเป็นค่า 0 – 1023 จึงจะต้องเขียนโค้ดคำสั่งในการแปลง โดยใช้หลักการเทียบบัญญัติไตรยางศ์ ซึ่งทำให้ได้สูตรออกมา คือ $1023 * \text{value}$

จากการเขียนโค้ดคำสั่ง จะเห็นว่าเมื่อขาของเซนเซอร์ไม่มีความชื้น จะส่งค่าออกมาเป็น 1023 และหากทดลองนำทิชชูเปียกมาหุ้มขาของเซนเซอร์ จะทำให้ค่าที่ได้ลดลง ยิ่งชื้นมาก ค่าที่ได้ยิ่งน้อยลง



```

Shell
Humidity = 1023.0
Humidity = 1023.0
Humidity = 1023.0
Humidity = 1023.0
Humidity = 1023.0
Humidity = 1022.0
Humidity = 713.15
Humidity = 655.18
Humidity = 600.21
Humidity = 568.22
Humidity = 574.22
Humidity = 574.22
Humidity = 585.21
Humidity = 606.2
  
```

← ไม่ชื้น

← ชื้น

4. เซนเซอร์วัดความเข้มแสง LDR Sensor (Light Dependent Resistor)



LDR คือ ความต้านทานชนิดที่ไวต่อแสง ตัวต้านทานนี้สามารถเปลี่ยนสภาพทางความนำไฟฟ้าได้เมื่อมีแสงมาตกกระทบ บางครั้งเรียกว่า “โฟโตริซิสเตอร์” (Photo Resistor) หรือ “โฟโตคอนดักเตอร์” (Photo Conductor) เป็นตัวต้านทานที่ทำมาจากสารกึ่งตัวนำ (Semiconductor) ประเภทแคดเมียมซัลไฟด์หรือแคดเมียมซีลีไนด์ ซึ่งทั้งสองตัวนี้ก็เป็นสารประเภทกึ่งตัวนำ เอามาฉาบลงบนแผ่นเซรามิกที่ใช้เป็นฐานรอง แล้วต่อขาจากสารที่ฉาบไว้ออกมา

การทำงานของ LDR หากมีแสงมาตกกระทบน้อย จะทำให้มีความต้านทานมาก และหากมีแสงมาตกกระทบบ่อย ความต้านทานจะน้อยลง

ฝึกทักษะการทดลองกิจกรรมที่ 4 เรียนรู้การใช้งาน LDR Sensor

อุปกรณ์ที่ใช้สำหรับการทดลอง

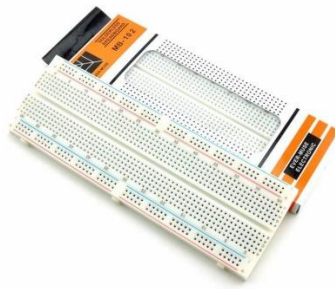
1. บอร์ด Raspberry Pi



2. LDR Sensor



4. Breadboard



5. สาย Jumper

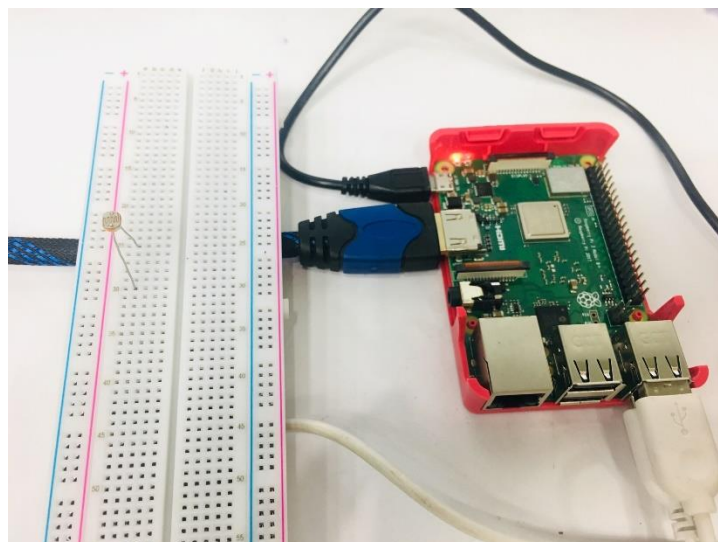


6. ตัวต้านทาน 4.7 กิโลโอห์ม

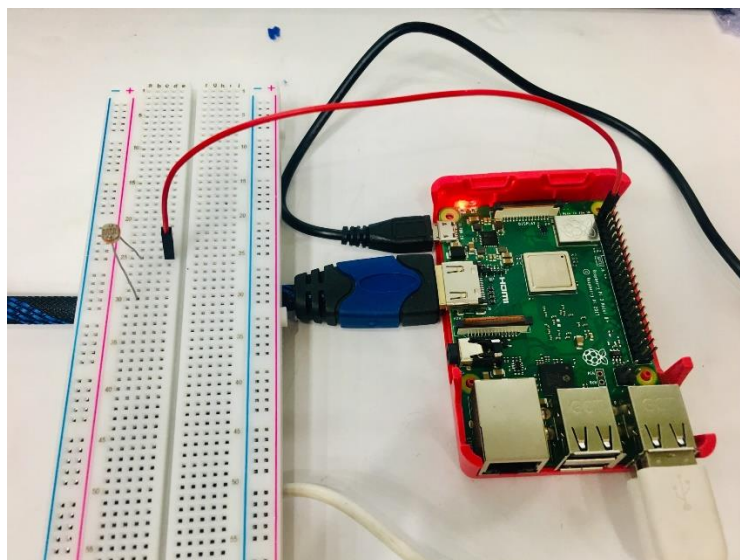


การต่อวงจร

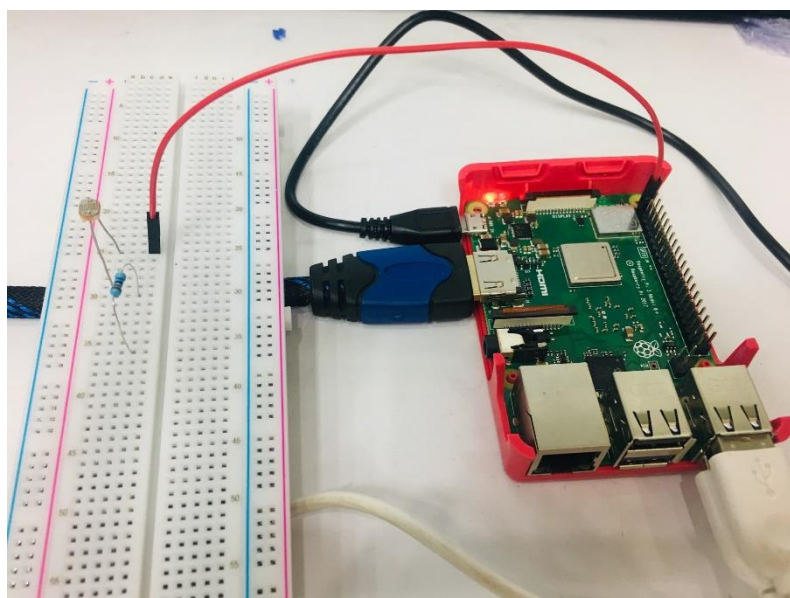
1. เชื่อม LDR Sensor บน Breadboard



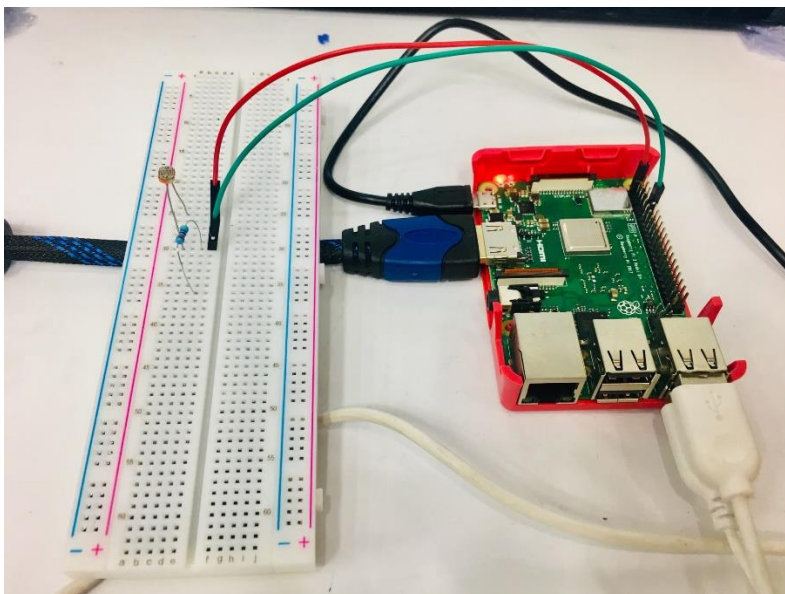
2. เสียบสาย Jumper บน Breadboard โดยต่อแถวเดียวกับขาด้านหนึ่งของ LDR Sensor ส่วนปลายสาย Jumper อีกด้านต่อไปยังขา PIN1 (3V3)



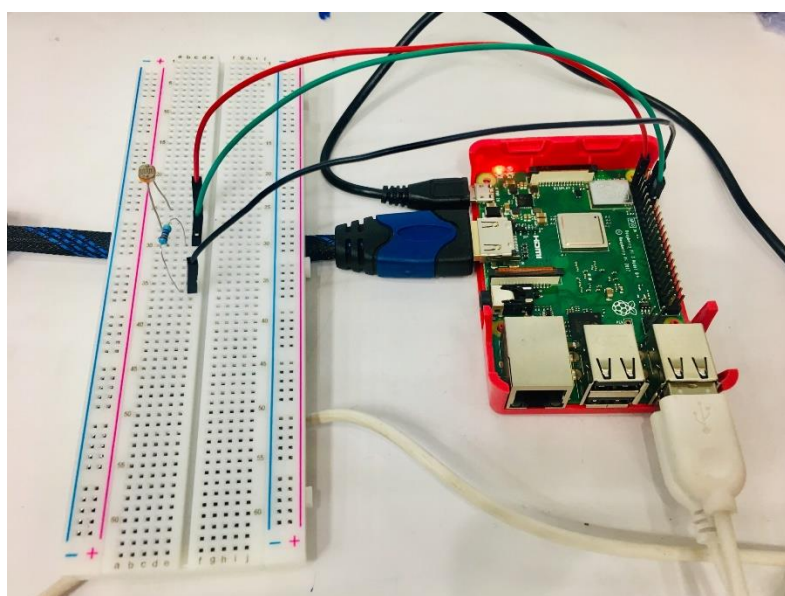
3. เสียบตัวต้านทาน 4.7 กิโลโอห์มเข้ากับ Breadboard ในแถวเดียวกับขาอีกข้างหนึ่งของ LDR Sensor



4. ใช้สาย Jumper เสียบบน Breadboard ในแถวเดียวกับขา LDR และตัวต้านทาน 4.7 กิโลโอห์ม ส่วนปลายสาย Jumper ต่อไปยังขา PIN10 (GPIO15)

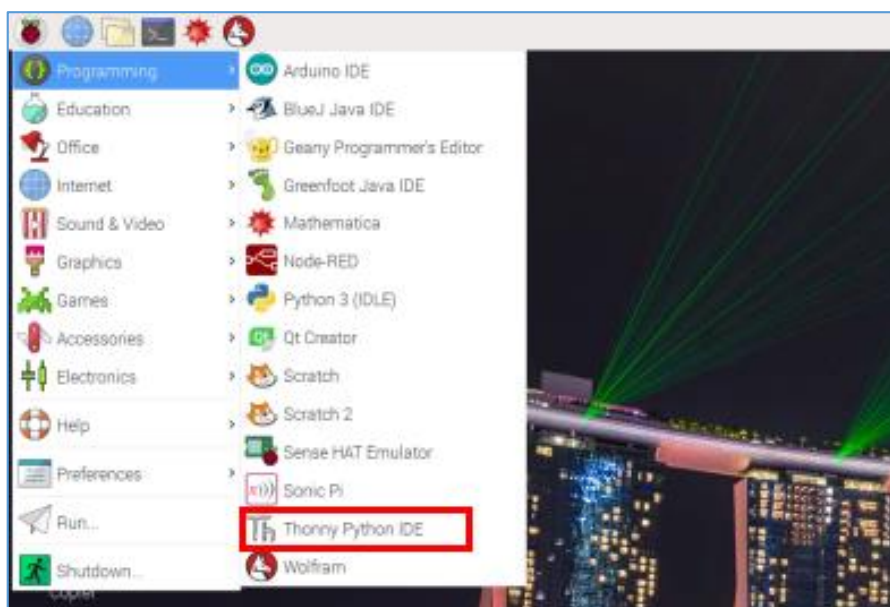


5. เสียบสาย Jumper ที่ขาอีกด้านหนึ่งของตัวต้านทาน 4.7 กิโลโอห์ม แล้วต่อไปยังขา PIN6 (Ground) ของขา GPIO



การเขียนโปรแกรมด้วยภาษาไพทอนเพื่อทดสอบอุปกรณ์

1. เปิดโปรแกรม Thonny Python IDE โดยไปที่ Pi > Programming > Thonny Python IDE

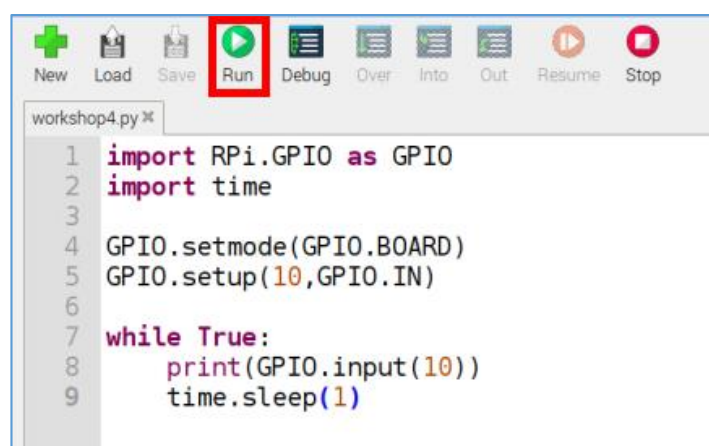


2. เขียนโปรแกรม ดังนี้

```

1 import RPi.GPIO as GPIO
2 import time
3
4 GPIO.setmode(GPIO.BOARD)
5 GPIO.setup(10,GPIO.IN)
6
7 while True:
8     print(GPIO.input(10))
9     time.sleep(1)
  
```

3. บันทึกและรันโปรแกรม โดยคลิกปุ่ม Run



ผลลัพธ์ที่ได้

LDR Sensor จะวัดค่าแสงทุกๆ 1 วินาที เมื่อ LDR Sensor วัดค่าแสงในที่ที่มีแสงสว่างน้อย (กลางคืน) ซึ่งอาจทดสอบโดยการเอามือปิดตัวเซนเซอร์ จะได้ค่าออกมาเป็นค่า 0 ถ้าแสงสว่างมาก (กลางวัน) ซึ่งอาจทดสอบโดยการเอามือที่ปิดตัวเซนเซอร์ออก จะได้ค่าออกมาเป็นค่า 1

```

Shell
>>> %Run workshop4.py
1
1
1
0
0
1
  
```

แสงมาก

แสงน้อย

5. เซนเซอร์วัดแก๊สและควัน MQ-2



เซนเซอร์วัดแก๊สและควัน เป็นโมดูลตรวจวัดแก๊ส ที่ไวต่อแก๊สไวไฟในกลุ่ม LPG, i-butane, propane, methane ,alcohol, Hydrogen รวมไปถึงควันไฟที่เกิดจากการเผาไหม้ด้วย จึงเป็นเซ็นเซอร์ที่นิยมนำมาใช้ในการตรวจจับการรั่วของแก๊สต่างๆ เพื่อป้องกันอันตรายที่จะเกิดขึ้นจากการรั่วไหลนั้นได้

- ใช้แรงดัน 5V
- ให้เอาที่พุดทั้งสัญญาณนอกซึ่งเป็นค่าที่วัดได้จริง และสัญญาณดิจิตอลสามารถปรับตั้งระดับแรงดันได้ (ใช้ LM393 เป็นวงจรเปรียบเทียบแรงดัน)
- เมื่อป้อนแรงดันให้แก่เซ็นเซอร์ ต้องรอการอุ่นชิพอย่างน้อย 20 วินาที ก่อนทำการวัดค่า

ฝึกทักษะการทดลองกิจกรรมที่ 5 เรียนรู้การใช้งาน MQ-2 Sensor

อุปกรณ์ที่ใช้สำหรับการทดลอง

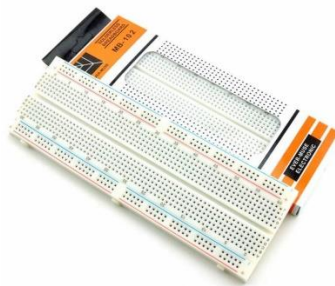
1. บอร์ด Raspberry Pi



2. MQ-2 Sensor



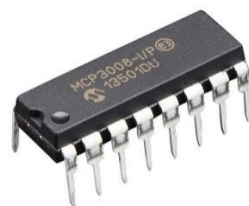
4. Breadboard



5. สาย Jumper



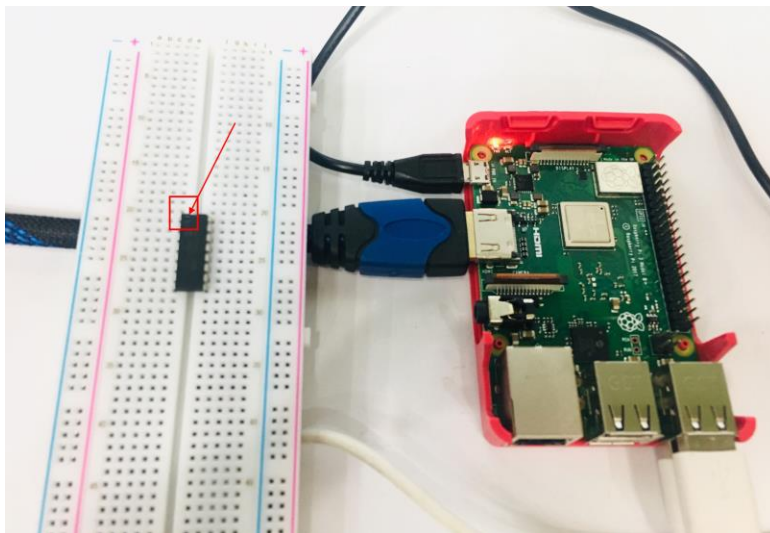
6. ชิป MCP3008



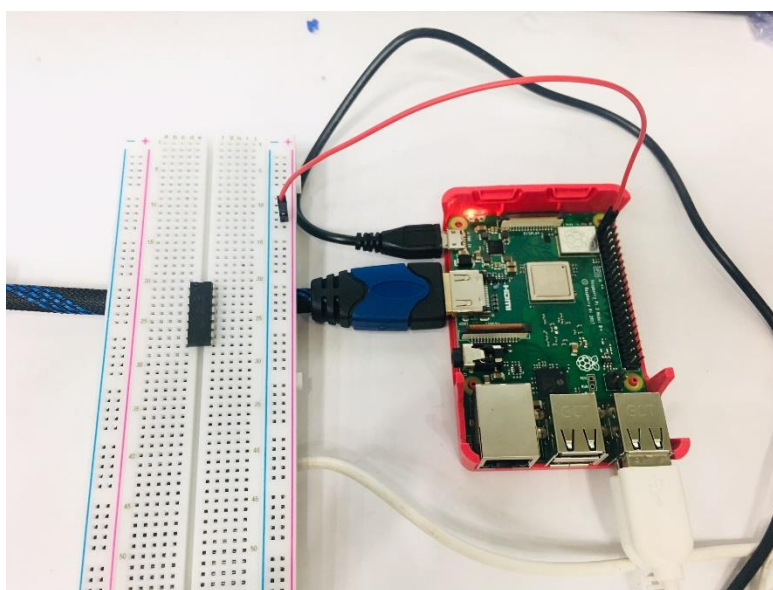
การต่อวงจร

ติดตั้ง MCP3008

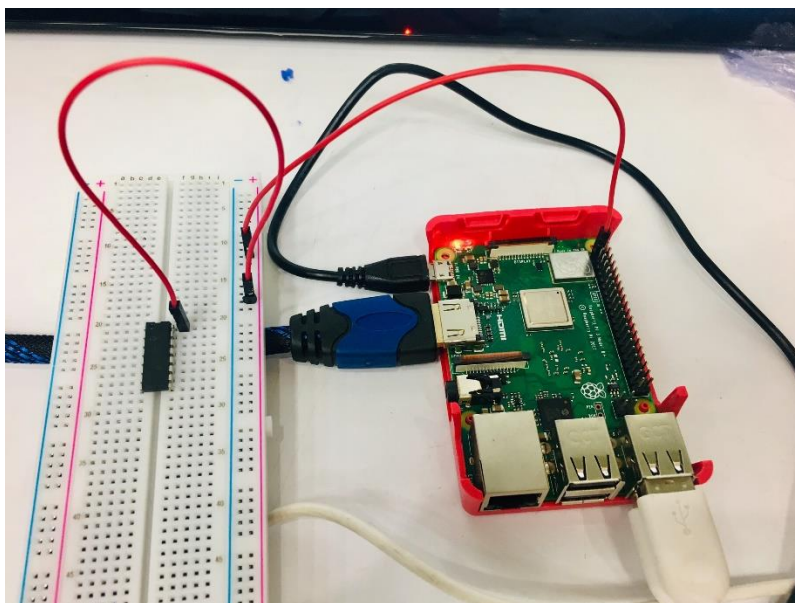
1. เสียบชิป MCP3008 ไว้ตรงกลาง Breadboard โดยสังเกตที่ตัวชิปจะมีจุดกลมๆ บนชิป ให้วางอยู่ด้านบน



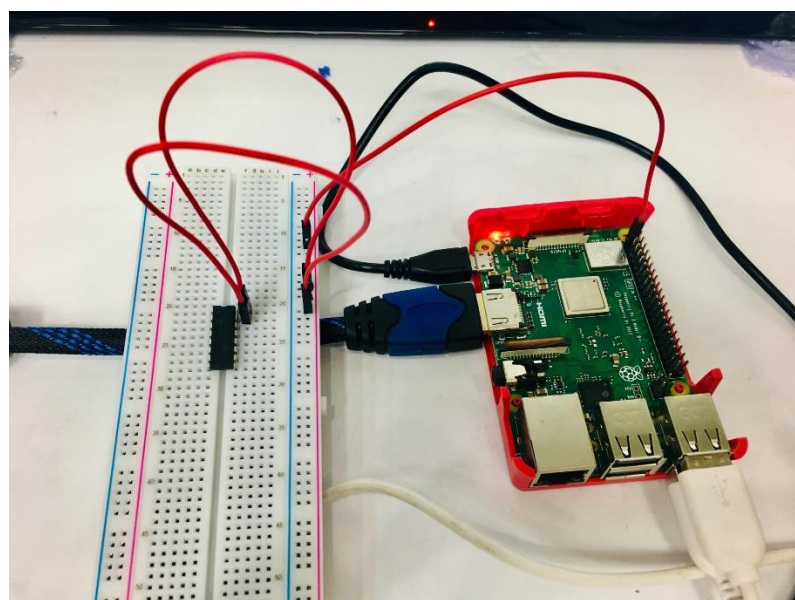
2. เสียบสาย Jumper จากขา PIN1 (3V3) ของขา GPIO ไปยังช่องใดก็ได้ในแถว + (สีแดง) ของ Breadboard



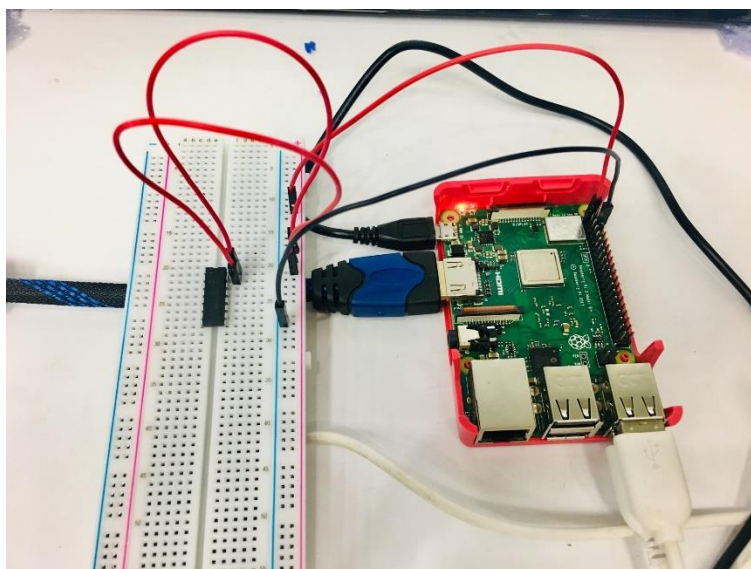
3. เสียบสาย Jumper จากขา VDD (ขาที่ 1 ขวามือ) ของชิป MCP3008 ไปยังแถว + (สีแดง) ของ Breadboard



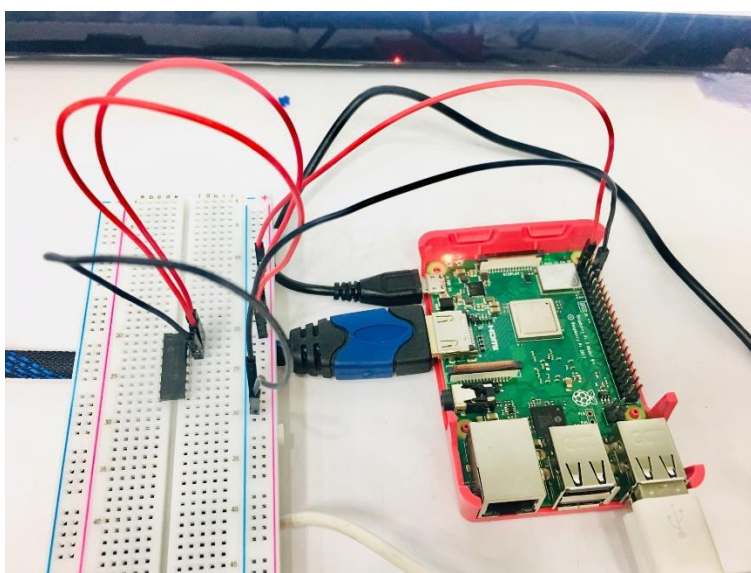
4. เสียบสาย Jumper จากขา VREF (ขาที่ 2 ขวามือ) ของชิป MCP3008 ไปยังแถว + (สีแดง) ของ Breadboard



5. เสียบสาย Jumper จากขา PIN6 (Ground) ของขา GPIO ไปยังแถว - (สีฟ้า) ของ Breadboard

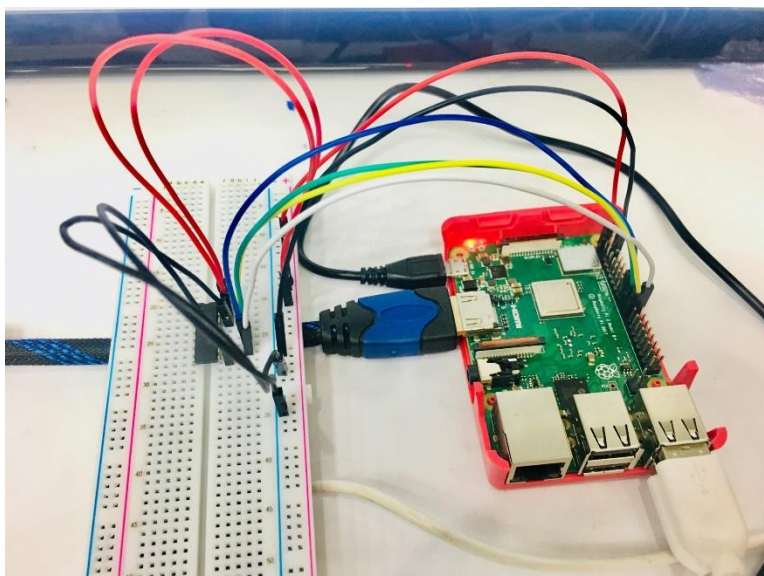


6. เสียบสาย Jumper จากขา AGND (ขาที่ 3 ขาวมือ) ของชิป MCP3008 ไปยังแถว - (สีฟ้า) ของ Breadboard



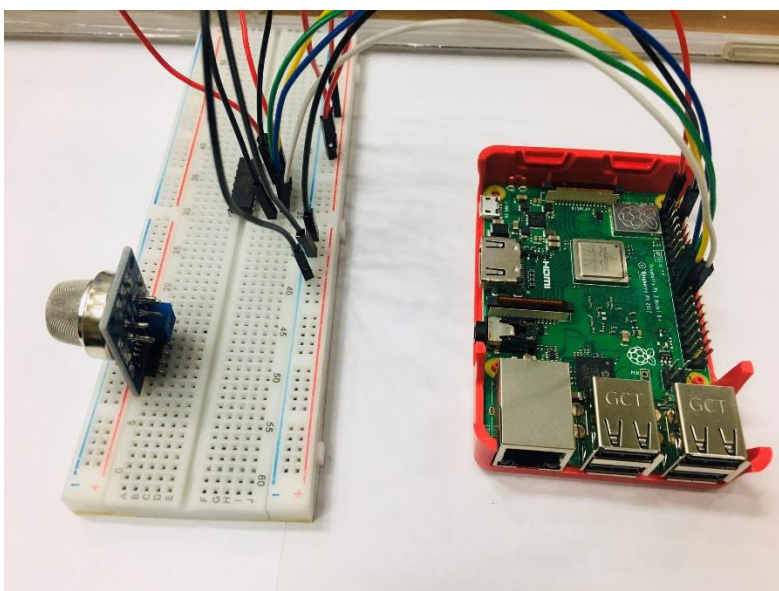
7. เสียบสาย Jumper จากขา DGND (ขาที่ 8 ขาวมือ) ของชิป MCP3008 ไปยังแถว - (สีฟ้า) ของ Breadboard

11. เสียบสาย Jumper จากขา CS/SHDN (ขาที่ 7 ขวามือ) ของชิป MCP3008 ไปยัง PIN24 (GPIO8 SP10_CE0_N) ของขา GPIO

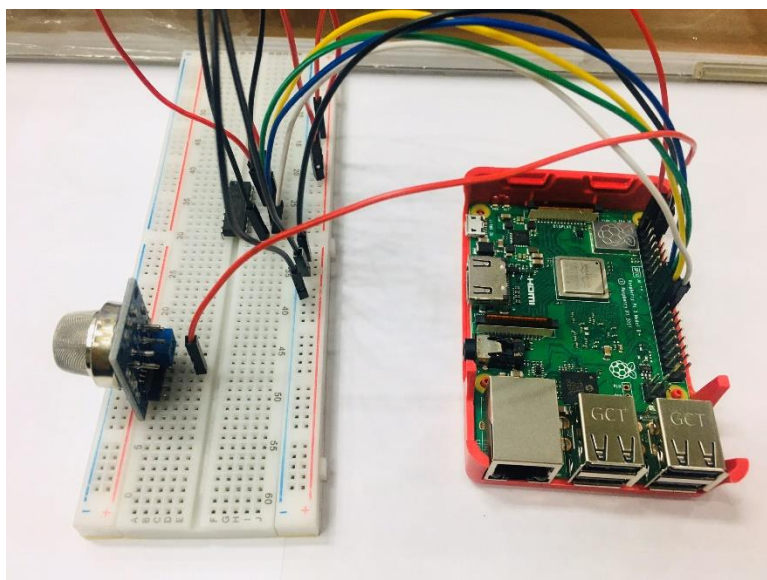


ติดตั้ง MQ-2 Sensor

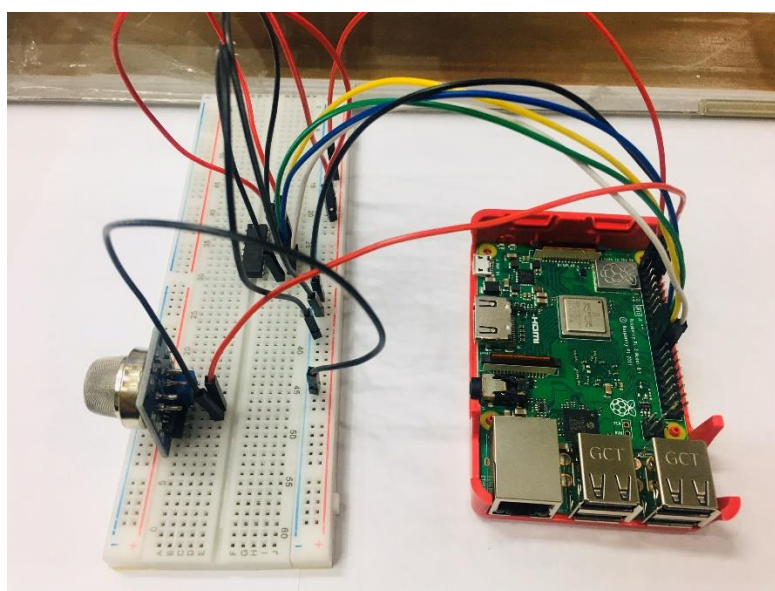
12. นำเซนเซอร์ MQ-2 เสียบบน Breadboard



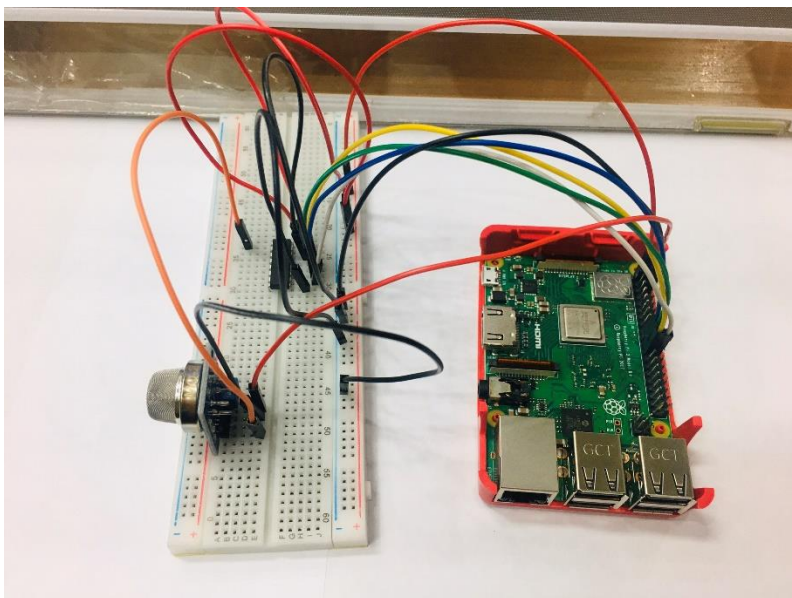
13. นำสาย Jumper แบบตัวผู้-ตัวเมีย เสียบเข้าไปในแถวเดียวกับขา VCC ของ MQ-2 ปลายสายต่อเข้ากับขา PIN2 (5V) ของบอร์ด Raspberry Pi



14. นำสาย Jumper แบบตัวผู้-ตัวผู้ เสียบเข้าไปในแถวเดียวกับขา GND ของ MQ-2 ปลายสายต่อเข้ากับช่อง – ของ Breadboard

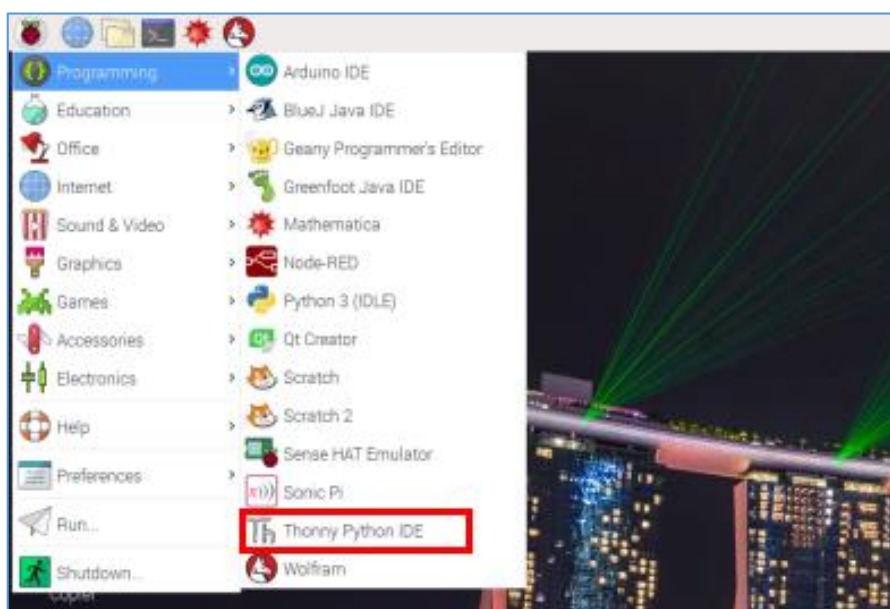


15. นำสาย Jumper แบบตัวผู้-ตัวผู้ เสียบเข้าไปในแถวเดียวกับขา A0 ของ MQ-2 ปลายสายต่อเข้ากับขา CH0 ของชิป MCP3008



การเขียนโปรแกรมด้วยภาษาไพทอนเพื่อทดสอบอุปกรณ์

1. เปิดโปรแกรม Thonny Python IDE โดยไปที่ Pi > Programming > Thonny Python IDE



2. เขียนโปรแกรม ดังนี้

```

gas.py x
1  from gpiozero import MCP3008
2  import time
3
4  port = MCP3008(0)
5
6  while True:
7      gas = port.value
8
9      x = round(1000 * gas,2)
10     print("Gas Value : %.2f" %x)
11     time.sleep(1)

```

ผลลัพธ์ที่ได้

MQ-2 Sensor จะวัดค่าแสงทุกๆ 1 วินาที ซึ่งค่าที่วัดได้จากเซนเซอร์จะอยู่ในช่วง 0 - 1000 ถ้าวัดค่าแก๊สในสถานะที่ไม่มีแก๊ส ค่าที่ได้จะมีค่าน้อย ถ้าวัดค่าแก๊สในสถานะที่มีแก๊ส ค่าที่ได้จะมีค่ามาก ซึ่งอาจทดสอบโดยการนำไฟแช็กที่เพิ่งจุดไฟมาวางด้านหน้าเซนเซอร์ เพื่อจำลองสถานการณ์ว่ามีแก๊สรั่วไหล

```

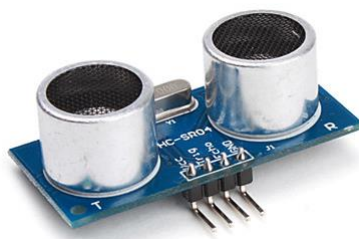
Shell
>>> %Run gas.py
Gas Value : 110.89
Gas Value : 110.89
Gas Value : 110.89
Gas Value : 110.89

```

โมดูล Module ต่างๆ ที่นำมาใช้กับบอร์ด Raspberry Pi

module โมดูลหรือส่วนจำเพาะ เป็นส่วนประกอบของระบบที่ใช้ในการเชื่อมต่อกับระบบอื่น โดยตัวโมดูลเองจะมีการออกแบบและควบคุมคัดแปลงภายในตัว

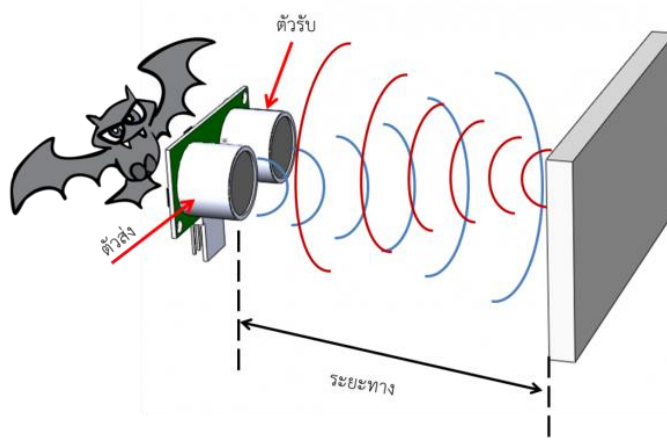
1. โมดูลวัดระยะทาง HC-SR04



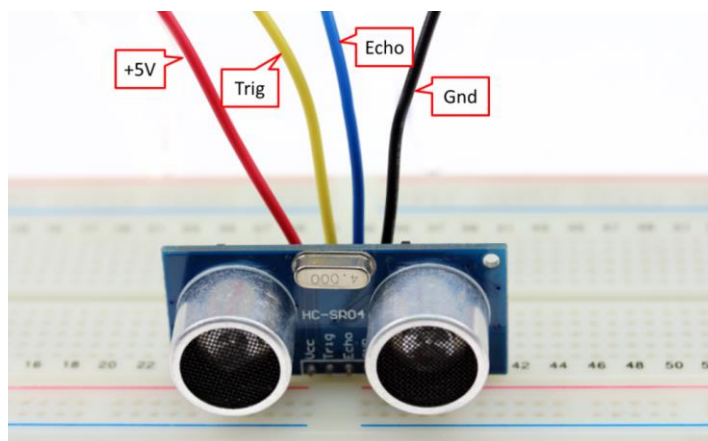
หลักการทำงาน

HC-SR04 เป็นเซนเซอร์โมดูลสำหรับตรวจจับวัตถุและวัดระยะทางแบบไม่สัมผัส โดยใช้คลื่นอัลตราโซนิก ซึ่งเป็นคลื่นเสียงความถี่สูงเกินกว่าการได้ยินของมนุษย์ วัดระยะได้ตั้งแต่ 2 – 400 เซนติเมตรหรือ 1 – 156 นิ้ว สามารถต่อใช้งานกับไมโครคอนโทรลเลอร์ได้ง่าย ใช้พลังงานต่ำ เหมาะกับการนำไปประยุกต์ใช้งานด้านระบบควบคุมอัตโนมัติหรืองานด้านหุ่นยนต์ หลักการทำงานจะเหมือนกันกับการตรวจจับวัตถุด้วยเสียงของค้างคาว โดยจะประกอบไปด้วยตัวรับ-ส่งอัลตราโซนิก ตัวส่งจะส่งคลื่นความถี่ 40 kHz ออกไปในอากาศด้วยความเร็วประมาณ 346 เมตรต่อวินาที และตัวรับจะคอยรับสัญญาณที่สะท้อนกลับจากวัตถุ เมื่อทราบความเร็วในการเคลื่อนที่ของคลื่น, เวลาที่ใช้ในการเดินทางไป-กลับ (t) ก็จะสามารควัดระยะห่างของวัตถุ (S) ได้จากสูตร

$$S = 346 \times 0.5t$$

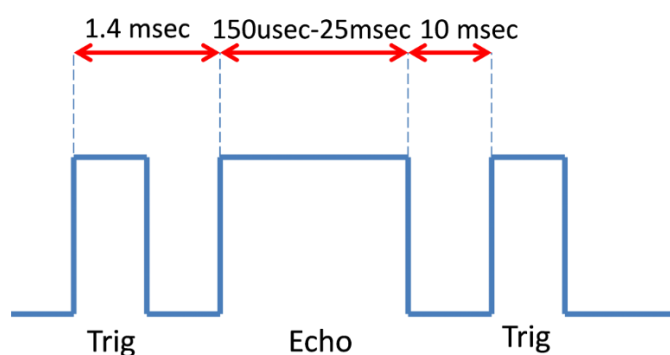


รูปที่ 1 หลักการตรวจจับและวัดระยะห่างระหว่างวัตถุด้วยคลื่นเสียง



รูปที่ 2 การต่อใช้งานโมดูลอัลตราโซนิก HC-SR04

ตามคุณลักษณะของเซนเซอร์ จะต้องสร้างสัญญาณพัลส์ความกว้างไม่น้อยกว่า 10 msec ป้อนเข้าที่ขา Trig หลังจากนั้นอีกประมาณ 1.4 msec จึงจะเริ่มมีสัญญาณพัลส์เกิดขึ้นที่ขา Echo มีความกว้างของสัญญาณตั้งแต่ 150 usec – 25 msec ซึ่งถ้าหากกว้างกว่านี้จะถือว่าตรวจไม่พบวัตถุ หลังจากนั้นควรหน่วงเวลาออกไปอีก 10 msec จึงจะส่งสัญญาณ Trig ออกไปอีกรอบ ตามรูปที่ 3



รูปที่ 3 สัญญาณที่ขา Trig และขา Echo ของโมดูลเซนเซอร์อัลตราโซนิก HC-SR04

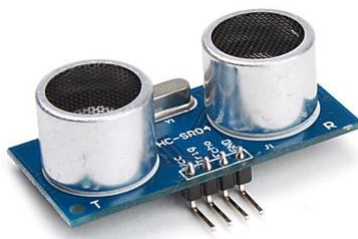
ฝึกทักษะการทดลองกิจกรรมที่ 6 เรียนรู้การใช้งาน HC-SR04

อุปกรณ์ที่ใช้สำหรับการทดลอง

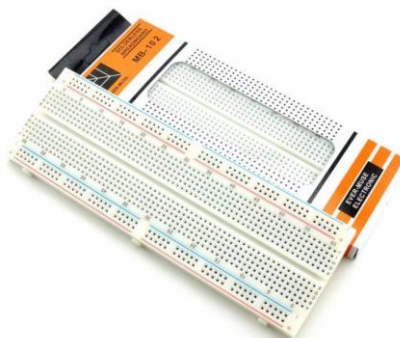
1. บอร์ด Raspberry Pi



2. โมดูลวัดระยะทาง HC-SR04



3. Breadboard



4. สาย Jumper

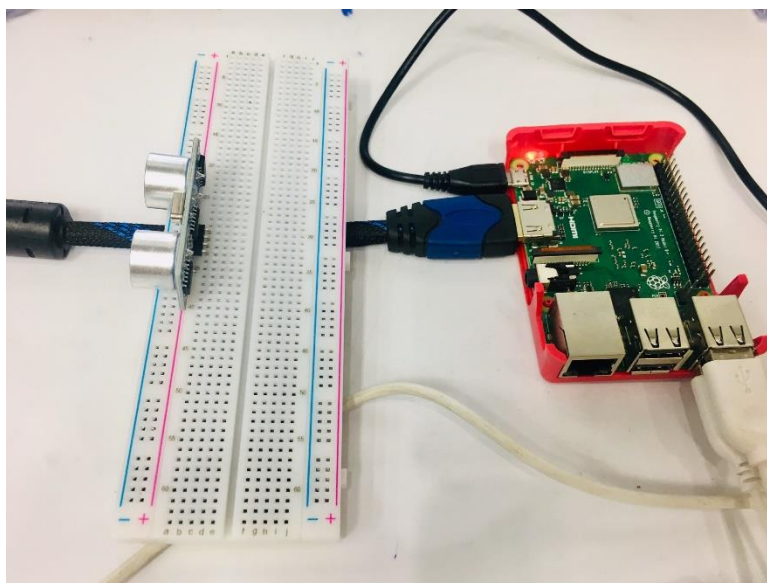


5. ตัวต้านทาน 4.7 กิโลโอห์ม



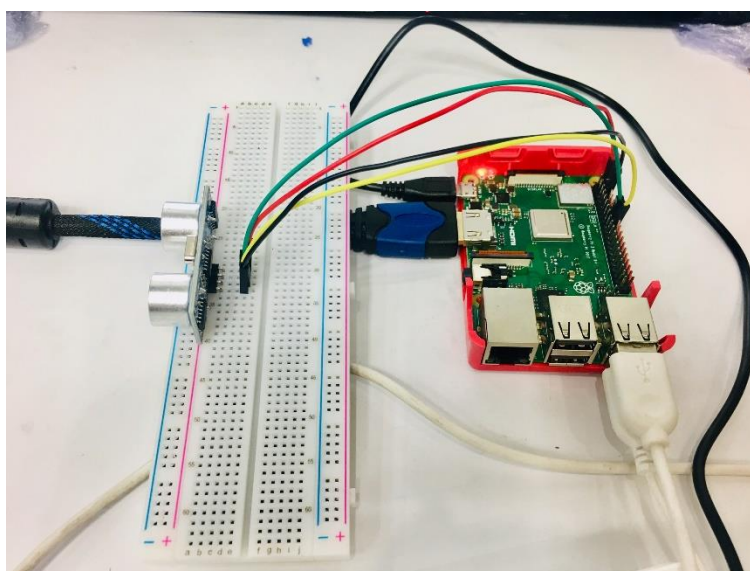
การต่อวงจร

1. เสียบโมดูลวัดระยะทาง HC-SR04 เข้ากับ Breadboard



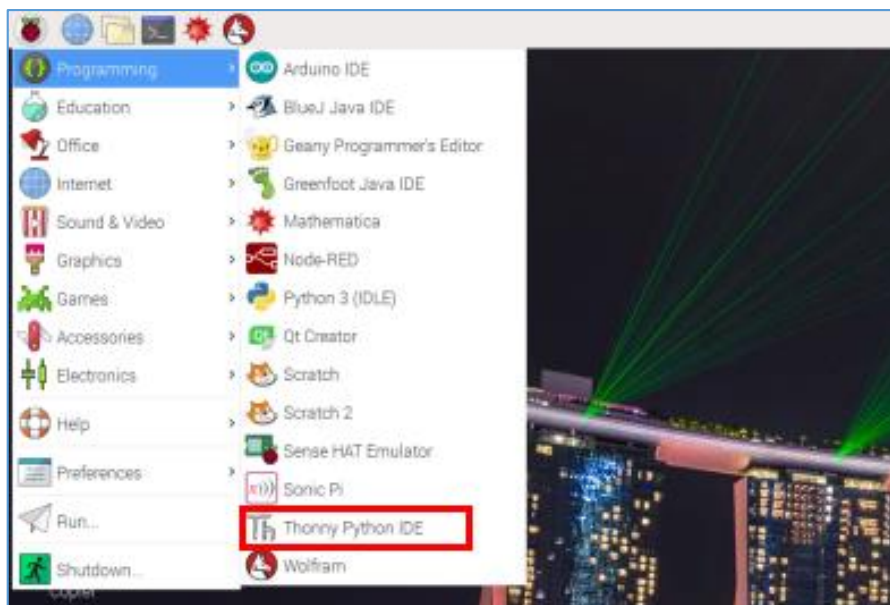
2. ใช้สาย Jumper 4 เส้นเสียบต่อจากขาของโมดูล HC-SR04 ซึ่งมี 4 ขา ไปยังขา GPIO โดยต่อดังนี้

- GND จะเสียบเข้ากับขา PIN6 (GROUND) ของ GPIO
- ECHO จะเสียบเข้ากับขา PIN18 (GPIO24) ของ GPIO
- TRIG จะเสียบเข้ากับขา PIN16 (GPIO23) ของ GPIO
- VCC จะเสียบเข้ากับขา PIN2 (5V) ของ GPIO



การเขียนโปรแกรมด้วยภาษาไพทอนเพื่อทดสอบอุปกรณ์

1. เปิดโปรแกรม Thonny Python IDE โดยไปที่ Pi > Programming > Thonny Python IDE



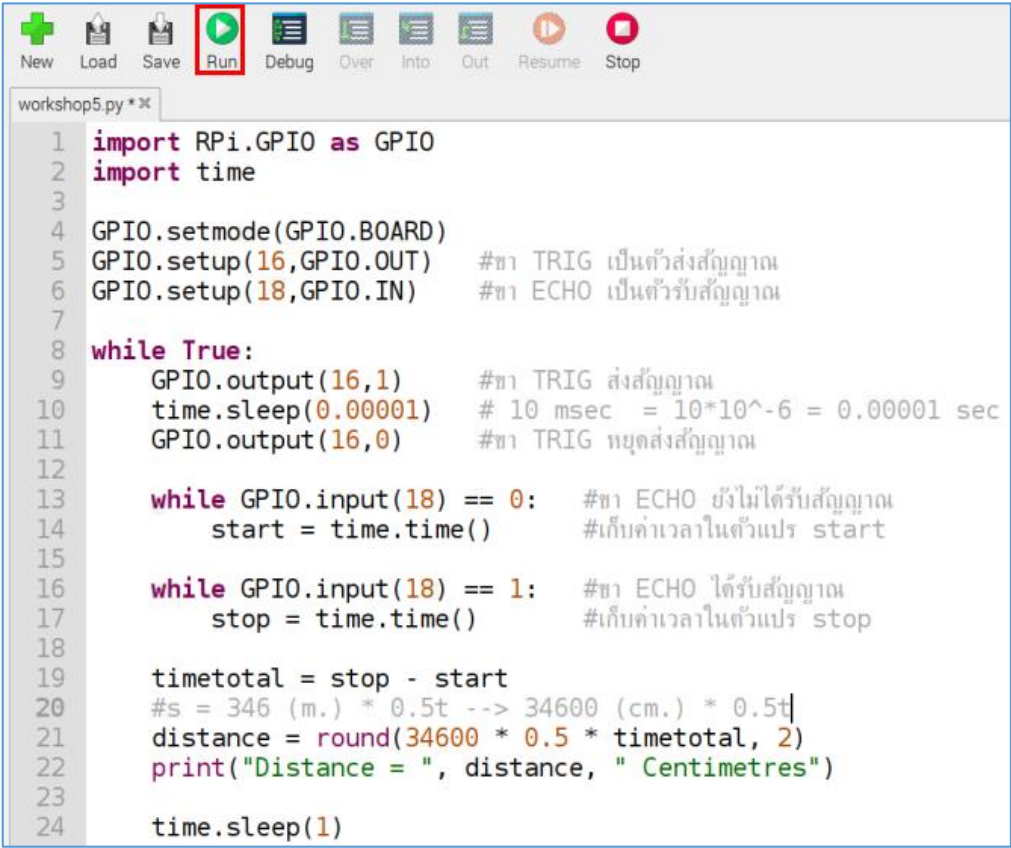
2. เขียนโปรแกรม ดังนี้

```

1 import RPi.GPIO as GPIO
2 import time
3
4 GPIO.setmode(GPIO.BOARD)
5 GPIO.setup(16,GPIO.OUT)    #ขา TRIG เป็นตัวส่งสัญญาณ
6 GPIO.setup(18,GPIO.IN)    #ขา ECHO เป็นตัวรับสัญญาณ
7
8 while True:
9     GPIO.output(16,1)      #ขา TRIG ส่งสัญญาณ
10    time.sleep(0.00001)    # 10 msec = 10*10^-6 = 0.00001 sec
11    GPIO.output(16,0)      #ขา TRIG หยุดส่งสัญญาณ
12
13    while GPIO.input(18) == 0:    #ขา ECHO ยังไม่ได้รับสัญญาณ
14        start = time.time()      #เก็บค่าเวลาในตัวแปร start
15
16    while GPIO.input(18) == 1:    #ขา ECHO ได้รับสัญญาณ
17        stop = time.time()       #เก็บค่าเวลาในตัวแปร stop
18
19    timetotal = stop - start
20    #s = 346 (m.) * 0.5t --> 34600 (cm.) * 0.5t
21    distance = round(34600 * 0.5 * timetotal, 2)
22    print("Distance = ", distance, " Centimetres")
23
24    time.sleep(1)

```

3. บันทึกและรันโปรแกรม โดยคลิกปุ่ม Run

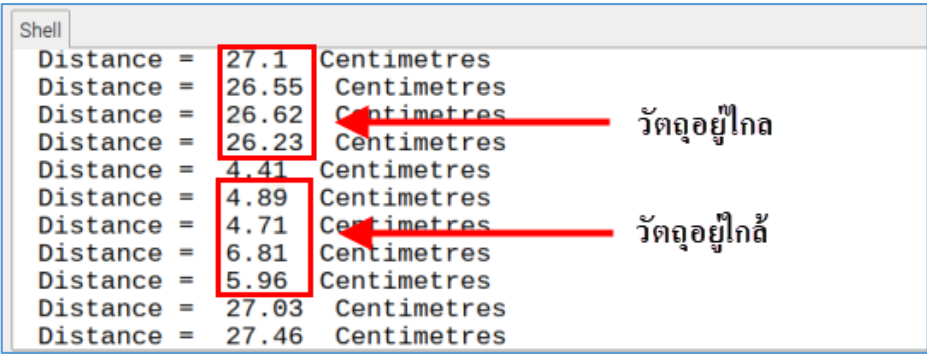


```

1 import RPi.GPIO as GPIO
2 import time
3
4 GPIO.setmode(GPIO.BOARD)
5 GPIO.setup(16,GPIO.OUT)    #ขา TRIG เป็นตัวส่งสัญญาณ
6 GPIO.setup(18,GPIO.IN)    #ขา ECHO เป็นตัวรับสัญญาณ
7
8 while True:
9     GPIO.output(16,1)      #ขา TRIG ส่งสัญญาณ
10    time.sleep(0.00001)    # 10 msec = 10*10^-6 = 0.00001 sec
11    GPIO.output(16,0)      #ขา TRIG หยุดส่งสัญญาณ
12
13    while GPIO.input(18) == 0: #ขา ECHO ยังไม่ได้รับสัญญาณ
14        start = time.time()  #เก็บค่าเวลาในตัวแปร start
15
16    while GPIO.input(18) == 1: #ขา ECHO ได้รับสัญญาณ
17        stop = time.time()   #เก็บค่าเวลาในตัวแปร stop
18
19    timetotal = stop - start
20    #s = 346 (m.) * 0.5t --> 34600 (cm.) * 0.5t
21    distance = round(34600 * 0.5 * timetotal, 2)
22    print("Distance = ", distance, " Centimetres")
23
24    time.sleep(1)
  
```

ผลลัพธ์ที่ได้

เมื่อมีวัตถุมาวางด้านหน้าของโมดูล HC-SR04 จะแสดงค่าระยะห่างระหว่างโมดูล HC-SR04 กับวัตถุออกมาทางจอภาพทุกๆ 1 นาที โดยจะแสดงตัวเลขทศนิยม 2 ตำแหน่ง จากนั้นให้ลองขยับระยะห่างให้น้อยลงหรือเพิ่มขึ้น แล้วสังเกตตัวเลขที่แสดงออกมา (ระยะห่างที่ได้จะมีหน่วยเป็นเซนติเมตร)



```

Shell
Distance = 27.1 Centimetres
Distance = 26.55 Centimetres
Distance = 26.62 Centimetres
Distance = 26.23 Centimetres
Distance = 4.41 Centimetres
Distance = 4.89 Centimetres
Distance = 4.71 Centimetres
Distance = 6.81 Centimetres
Distance = 5.96 Centimetres
Distance = 27.03 Centimetres
Distance = 27.46 Centimetres
  
```

วัตถุอยู่ไกล

วัตถุอยู่ใกล้

2. ลำโพง Buzzer Module



Buzzer Module เป็นอุปกรณ์ทำหน้าที่คล้ายลำโพง แต่ออกแบบมาเพื่อใช้ในการส่งสัญญาณเสียงความถี่สูงๆ บนบอร์ด มีทรานซิสเตอร์สำหรับช่วยขับมาไว้แล้วสามารถต่อใช้งานกับไมโครคอนโทรลเลอร์ได้โดยตรง การใช้งานเพียงเขียนโค้ดสร้างสัญญาณ HIGH และสัญญาณ LOW โดยสามารถกำหนดความถี่เสียงที่ต้องการได้เอง

ฝึกทักษะการทดลองกิจกรรมที่ 7 เรียนรู้การใช้งาน Buzzer Module

อุปกรณ์ที่ใช้สำหรับการทดลอง

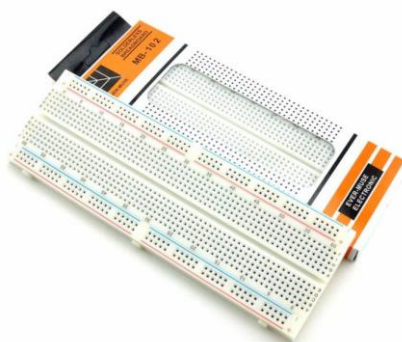
1. บอร์ด Raspberry Pi



2. Buzzer Module



3. Breadboard

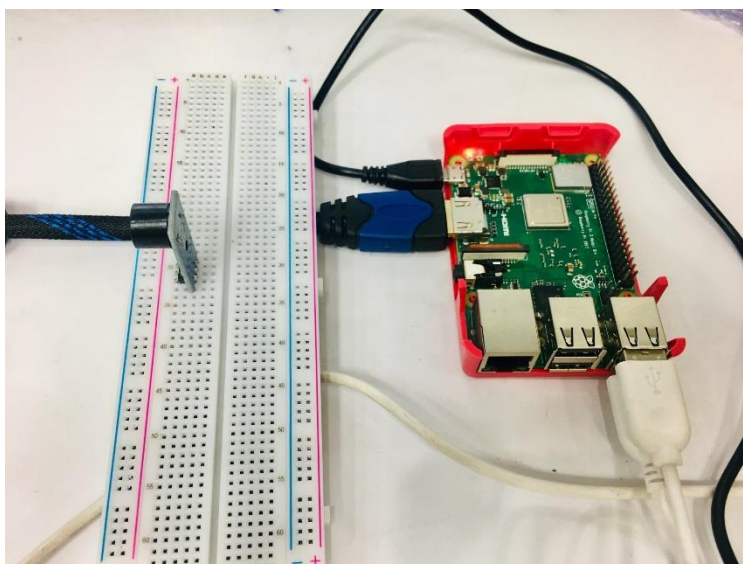


4. สาย Jumper

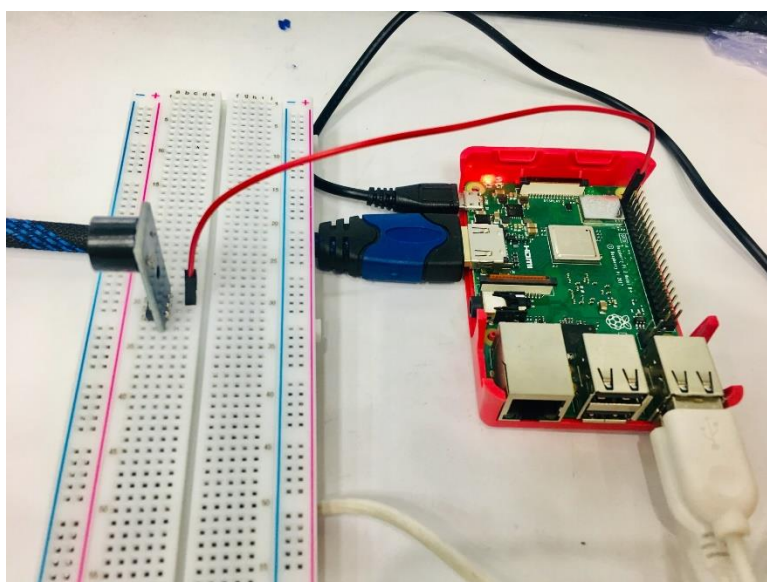


การต่อวงจร

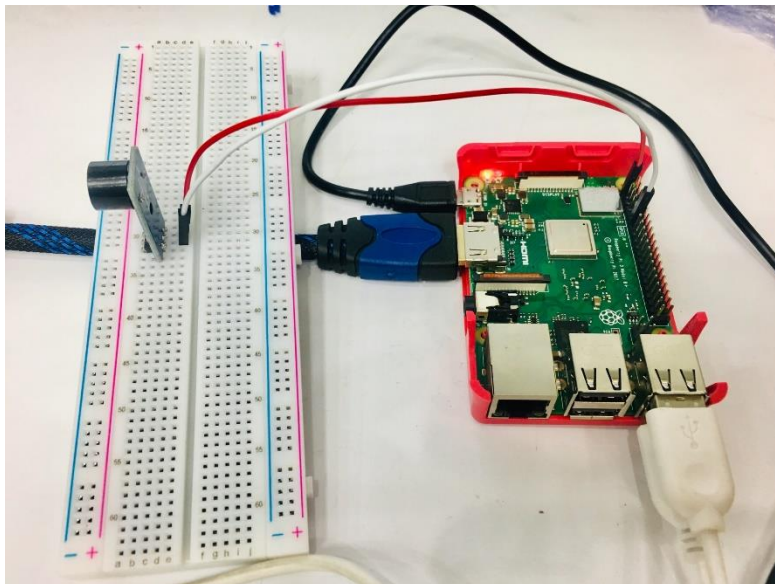
1. เสียบ Buzzer บน Breadboard



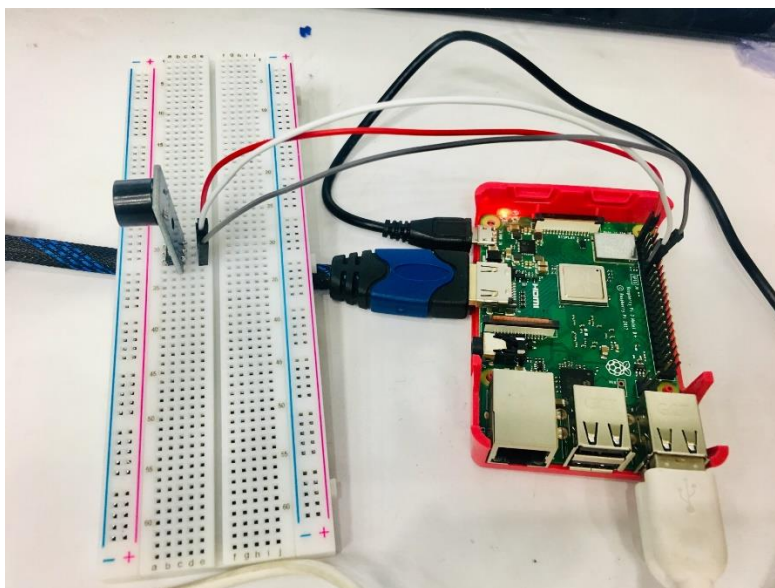
2. ใช้สาย Jumper เสียบบน Breadboard ในแถวเดียวกับขา VCC ของ Buzzer ส่วนปลายอีกด้านของสาย Jumper เสียบเข้ากับช่อง PIN1 (3V3) ของขา GPIO



3. ใช้สาย Jumper เสียบบน Breadboard ในแถวเดียวกับขา I/O ของ Buzzer ส่วนปลายอีกด้านของสาย Jumper เสียบเข้ากับช่อง PIN11 (GPIO17) ของขา GPIO

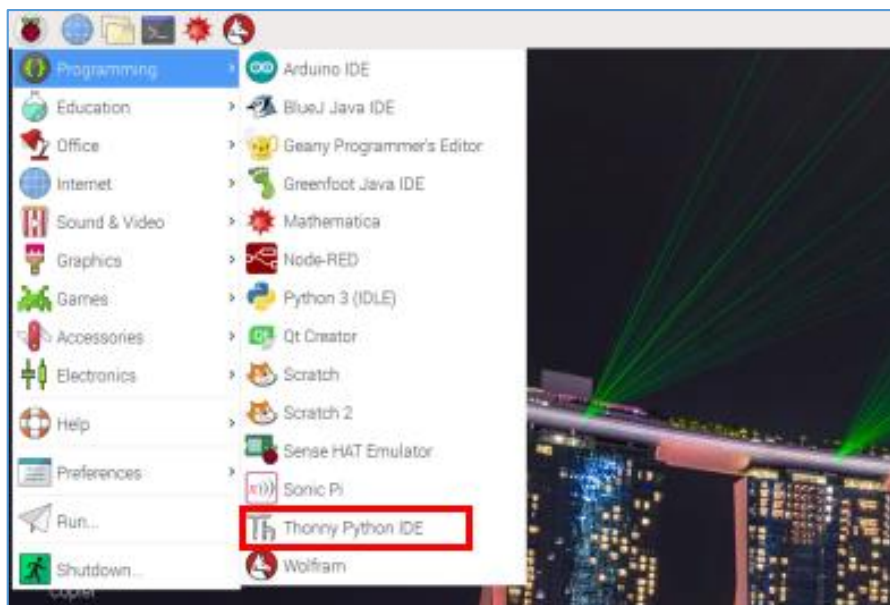


4. ใช้สาย Jumper เสียบบน Breadboard ในแถวเดียวกับขา GND ของ Buzzer ส่วนปลายอีกด้านของสาย Jumper เสียบเข้ากับช่อง PIN6 (Ground) ของขา GPIO (เมื่อเสียบสาย Ground เสียงคำโพงจะดังต่อเนื่องให้ถอดสาย Ground ออกชั่วคราว แล้วค่อยเสียบใหม่หลังรันโปรแกรม)



การเขียนโปรแกรมด้วยภาษาไพทอนเพื่อทดสอบอุปกรณ์

1. เปิดโปรแกรม Thonny Python IDE โดยไปที่ Pi > Programming > Thonny Python IDE



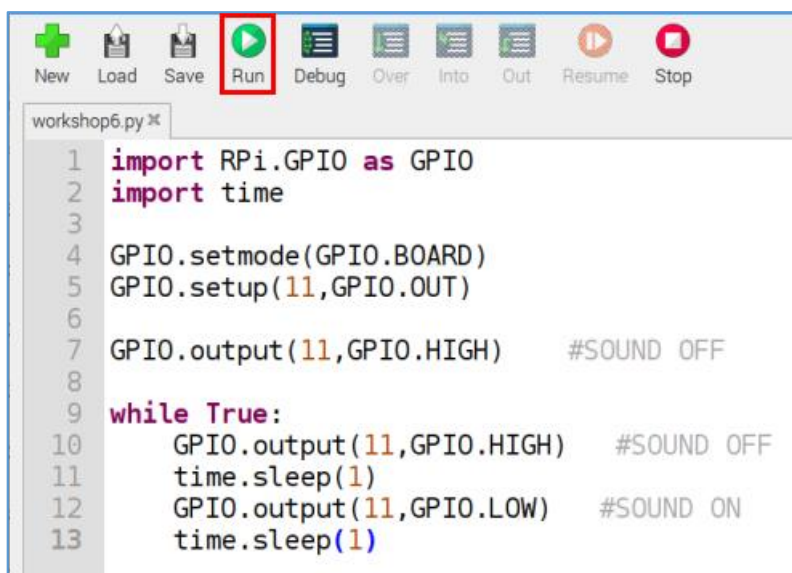
2. เขียนโปรแกรม ดังนี้

```

1 import RPi.GPIO as GPIO
2 import time
3
4 GPIO.setmode(GPIO.BOARD)
5 GPIO.setup(11,GPIO.OUT)
6 GPIO.output(11,GPIO.HIGH) #ปิดเสียงไว้ก่อน
7
8 while True:
9     GPIO.output(11,GPIO.HIGH) #ปิดเสียง
10    time.sleep(0.5)
11    GPIO.output(11,GPIO.LOW) #เปิดเสียง
12    time.sleep(0.5)

```

3. บันทึกและรันโปรแกรม โดยคลิกปุ่ม Run



```

workshop6.py
1 import RPi.GPIO as GPIO
2 import time
3
4 GPIO.setmode(GPIO.BOARD)
5 GPIO.setup(11,GPIO.OUT)
6
7 GPIO.output(11,GPIO.HIGH)    #SOUND OFF
8
9 while True:
10     GPIO.output(11,GPIO.HIGH)    #SOUND OFF
11     time.sleep(1)
12     GPIO.output(11,GPIO.LOW)    #SOUND ON
13     time.sleep(1)

```

ผลลัพธ์ที่ได้

เสียงลำโพงจะดังและหยุดสลับกันทุกๆ 1 วินาที

3. Relay module แบบ 2 Channel (HL-52R)



Relay module HL-52R เป็นอุปกรณ์ไฟฟ้าชนิดหนึ่ง ซึ่งทำหน้าที่ตัดต่อวงจรแบบเดียวกับสวิตช์ โดยควบคุมการทำงานด้วยไฟฟ้า Relay มีหลายประเภท ตั้งแต่ Relay ขนาดเล็กที่ใช้ในงานอิเล็กทรอนิกส์ทั่วไป จนถึง Relay ขนาดใหญ่ที่ใช้ในงานไฟฟ้าแรงสูง โดยมีรูปร่างหน้าตาแตกต่างกันออกไป แต่มีหลักการทำงานที่คล้ายคลึงกัน สำหรับการนำ Relay ไปใช้งาน จะใช้ในการตัดต่อวงจร ทั้งนี้ Relay ยังสามารถเลือกใช้งานได้หลากหลายรูปแบบ

ฝึกทักษะการทดลองกิจกรรมที่ 8 เรียนรู้การใช้งาน Relay Module อุปกรณ์ที่ใช้สำหรับการทดลอง

1. บอร์ด Raspberry Pi



2. Relay Module แบบ 2 Channel



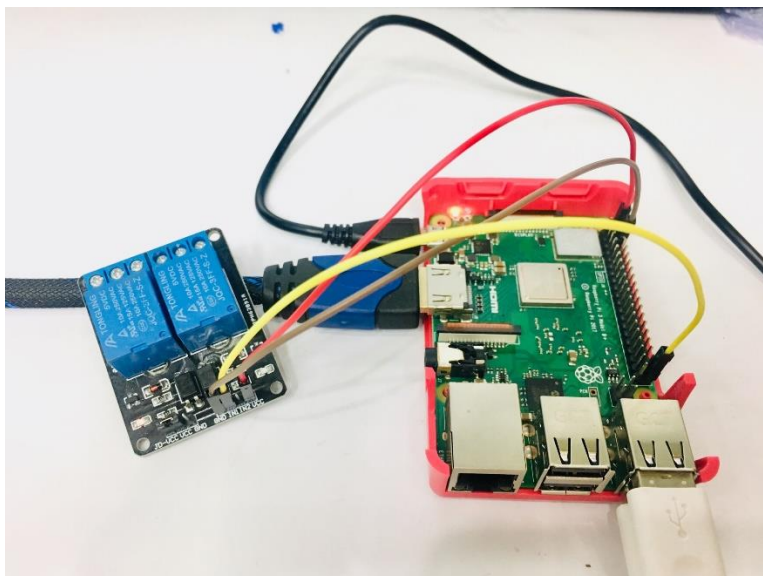
3. สาย Jumper



การต่อวงจร

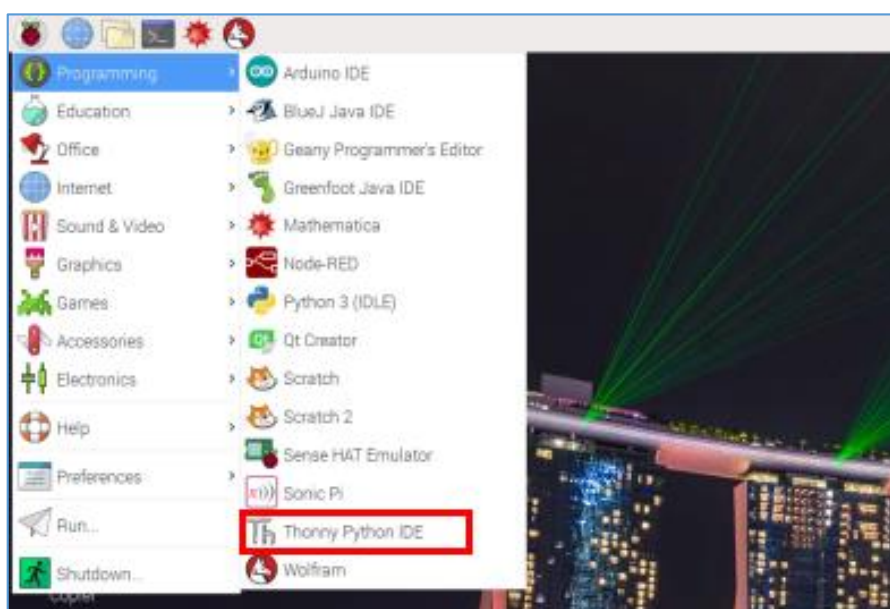
1. เสียบสาย Jumper เข้ากับขาของ Relay Channel ที่ 1 โดยต่อสายดังนี้

- ช่อง VCC ของ Relay เสียบเข้ากับ PIN2 (5V) ของขา GPIO
- ช่อง GND ของ Relay เสียบเข้ากับ PIN6 (Ground) ของขา GPIO
- ช่อง IN1 ของ Relay เสียบเข้ากับ PIN40 (GPIO21) ของขา GPIO



การเขียนโปรแกรมด้วยภาษาไพทอนเพื่อทดสอบอุปกรณ์

1. เปิดโปรแกรม Thonny Python IDE โดยไปที่ Pi > Programming > Thonny Python IDE



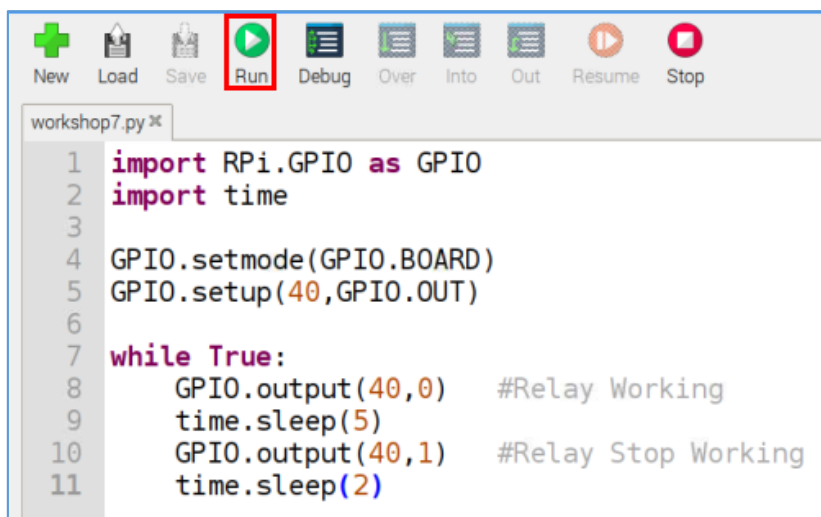
2. เขียนโปรแกรม ดังนี้

```

1 import RPi.GPIO as GPIO
2 import time
3
4 GPIO.setmode(GPIO.BOARD)
5 GPIO.setup(40,GPIO.OUT)
6
7 while True:
8     GPIO.output(40,0)    #Relay Working
9     time.sleep(5)
10    GPIO.output(40,1)    #Relay Stop Working
11    time.sleep(2)

```

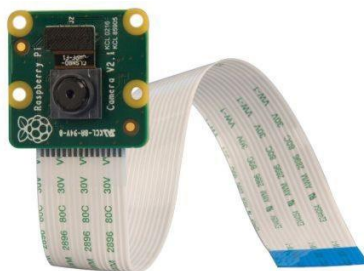
3. บันทึกไฟล์ และรันโปรแกรม โดยคลิกปุ่ม Run



ผลลัพธ์ที่ได้

Relay Channel ที่ 1 ซึ่งเปรียบเสมือนเป็นสวิตช์ไฟจะทำงาน โดยจะมีเสียงติ๊ด 1 ครั้ง และไฟสถานะจะติดเป็นเวลา 5 วินาที หลังจากนั้น Relay Channel ที่ 1 จะหยุดทำงาน โดยจะมีเสียงติ๊ด 1 ครั้ง และไฟสถานะจะดับเป็นเวลา 2 วินาที สลับกันไปเรื่อยๆ

4. กล้อง Raspberry Pi Camera Module v2



Raspberry Pi Camera Board เป็นโมดูลกล้องที่ออกแบบมาใช้งานร่วมกับบอร์ด Raspberry Pi โดยสามารถเชื่อมต่อกับ Socket CAMERA บนบอร์ดที่มีการเชื่อมต่อแบบ CSI Bus ได้ทันที และยังเหมาะสำหรับงานวิดีโอความละเอียดสูงและการถ่ายภาพนิ่ง นอกจากนี้ยังสามารถถ่ายภาพแบบ time-lapse และ slow-motion ได้

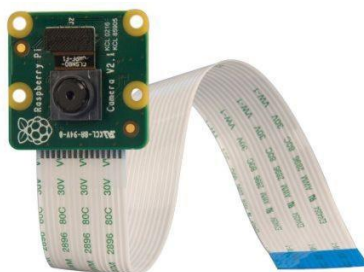
ฝึกทักษะการทดลองกิจกรรมที่ 9 เรียนรู้การใช้งาน Pi Camera สำหรับถ่ายภาพนิ่ง 1 ภาพ

อุปกรณ์ที่ใช้สำหรับการทดลอง

1. บอร์ด Raspberry Pi

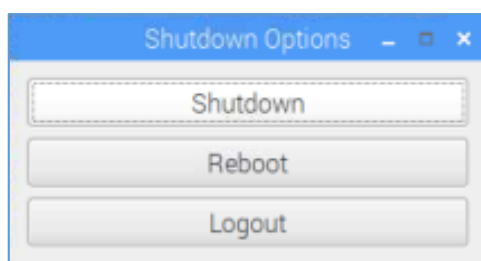
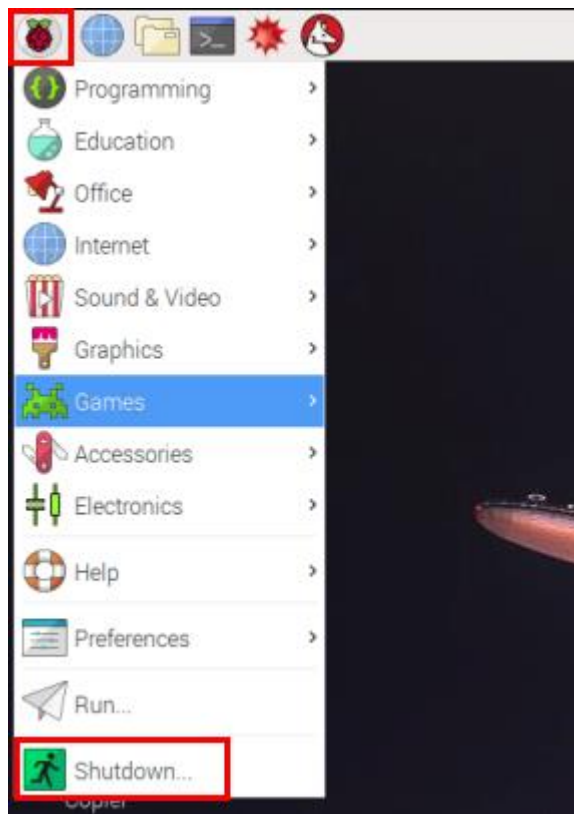


2. Pi Camera



การเชื่อมต่อ

1. ปิด Raspberry Pi โดยไปที่ Pi > Shutdown > Shutdown



2. ถอดสาย Adapter ออกจากบอร์ด Raspberry Pi
3. เสียบสายแพ Pi Camera เข้าช่อง CSI โดยดึงเขี้ยวล็อกขึ้นก่อน จากนั้นเสียบสายแพลงไป โดยเอาด้านสีฟ้าหันมาทางพอร์ต LAN จากนั้นกดเขี้ยวล็อกลงไป

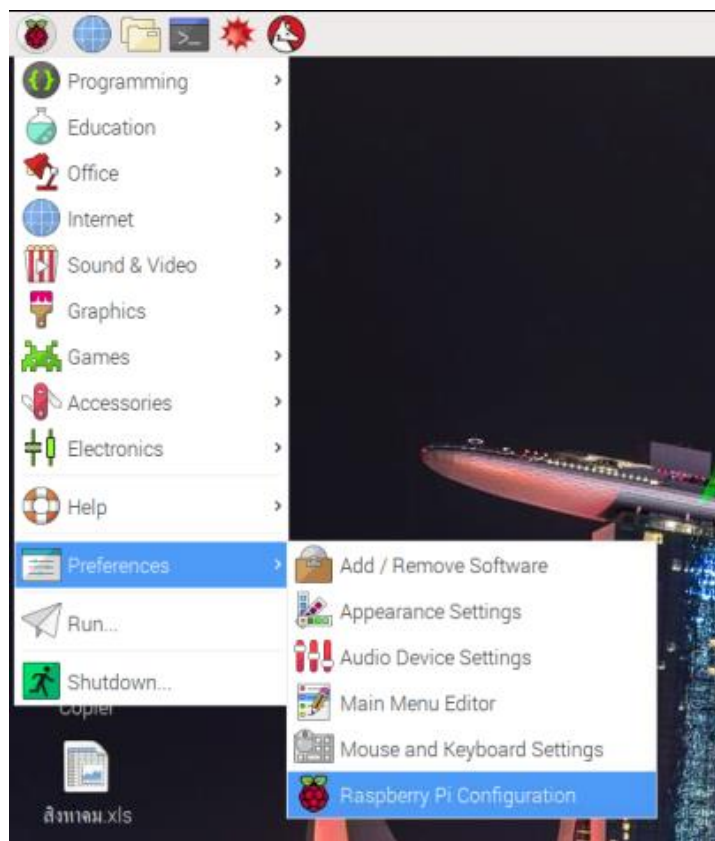


4. เสียบสาย Adapter เพื่อเปิดเครื่องอีกครั้ง

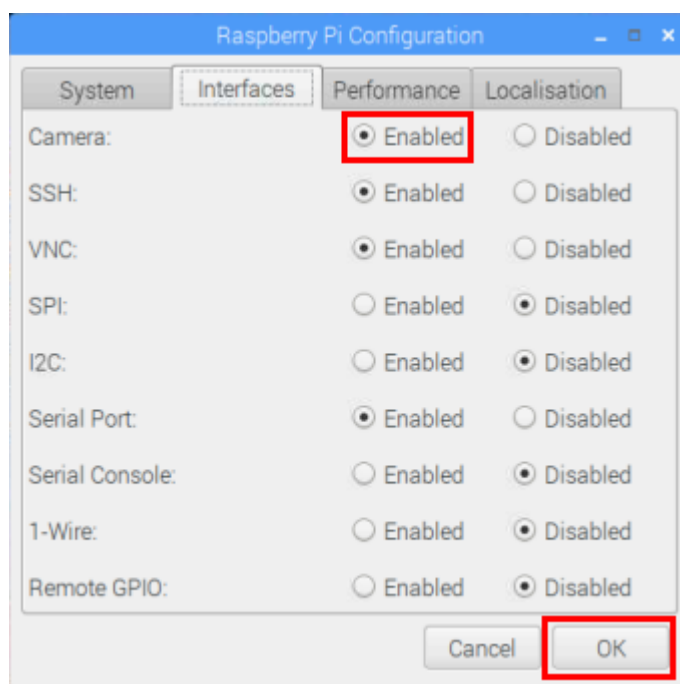
ตั้งค่าระบบ

1. เปิดการใช้งานกล้อง Pi Camera โดยคลิกเมนูไอคอน Pi --> Preferences --> Raspberry Pi

Configuration

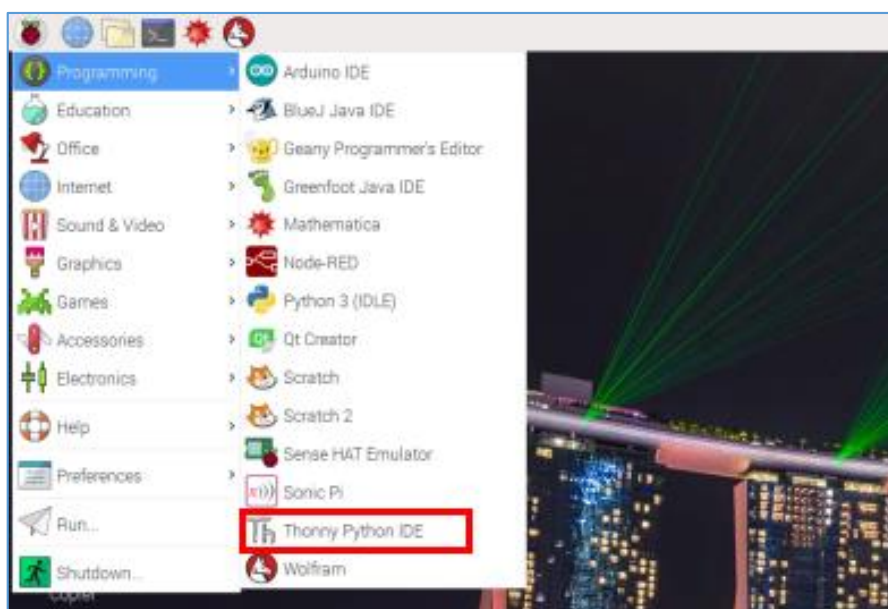


2. คลิกแท็บ Interface แล้วคลิกที่ Camera ให้เป็น Enabled แล้วคลิกปุ่ม OK



การเขียนโปรแกรมด้วยภาษาไพทอนเพื่อทดสอบอุปกรณ์

1. เปิดโปรแกรม Thonny Python IDE โดยไปที่ Pi > Programming > Thonny Python IDE



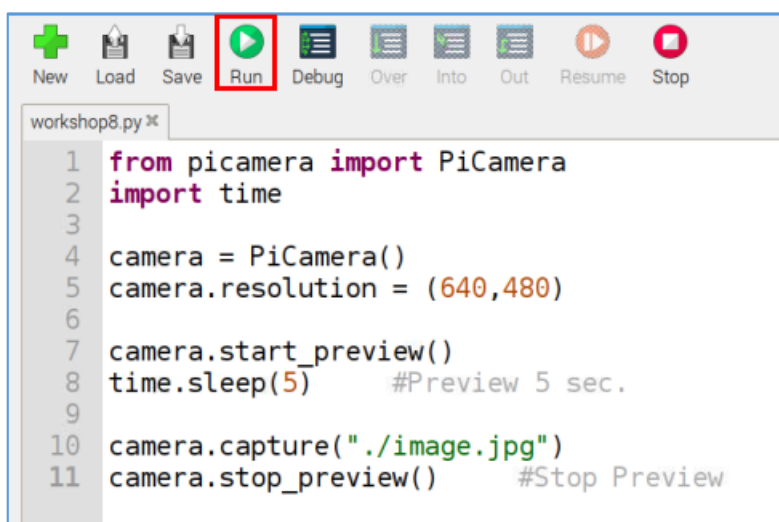
2. เขียนโปรแกรม ดังนี้

```

1 from picamera import PiCamera
2 import time
3
4 camera = PiCamera()
5 camera.resolution = (640,480)
6
7 camera.start_preview()
8 time.sleep(5)      #Preview 5 sec.
9
10 camera.capture("./image.jpg")
11 camera.stop_preview()    #Stop Preview

```

3. บันทึกไฟล์และรันโปรแกรม โดยคลิกปุ่ม Run

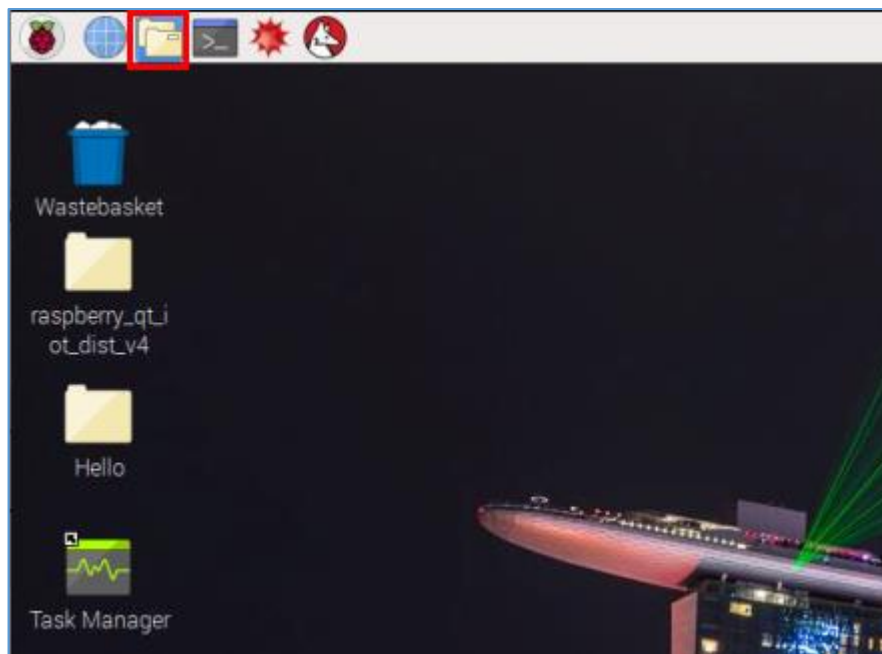


ผลลัพธ์ที่ได้

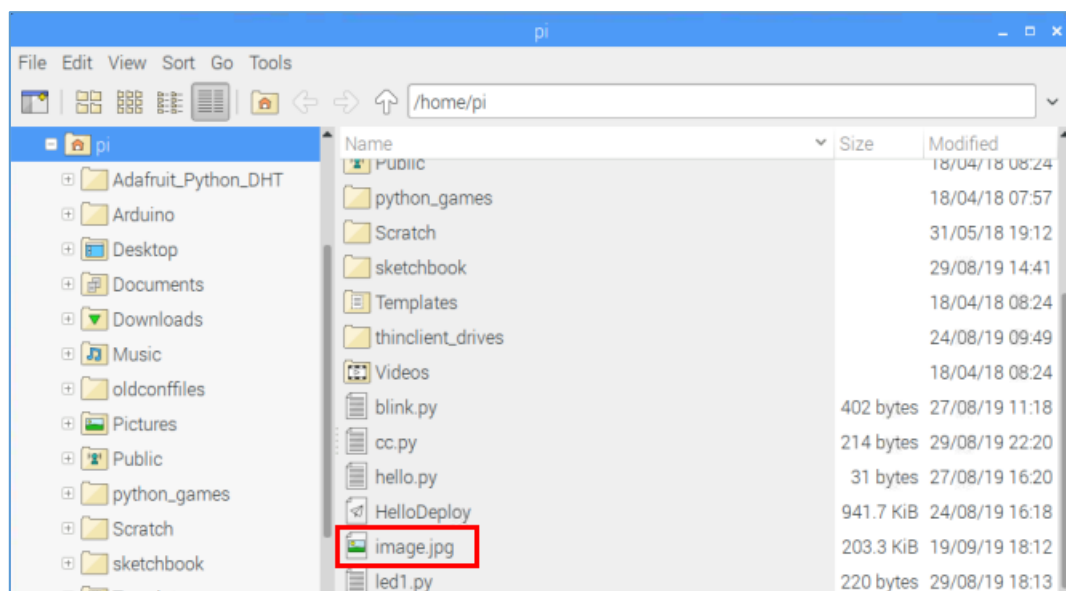
กล้องจะเปิดหน้า Preview ขึ้นมา 5 วินาที หลังจากนั้นกล้องจะถ่ายภาพ และบันทึกภาพที่มีขนาด 640 x 480 pixels ไฟล์รูปภาพที่ได้จะถูกบันทึกไว้ในตำแหน่งเดียวกับไฟล์โค้ดคำสั่ง โดยมีชื่อว่า image1.jpg



คลิกเข้า File Manager



จะเจอรูปภาพที่ได้ถ่ายเอาไว้ ชื่อว่า image.jpg

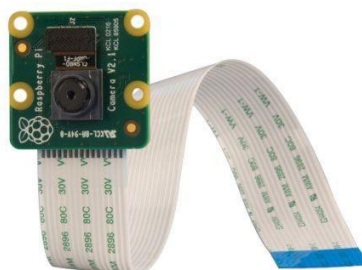


ฝึกทักษะการทดลองกิจกรรมที่ 10 เรียนรู้การใช้งาน Pi Camera สำหรับถ่ายภาพนิ่งหลาย Shot
อุปกรณ์ที่ใช้สำหรับการทดลอง

1. บอร์ด Raspberry Pi

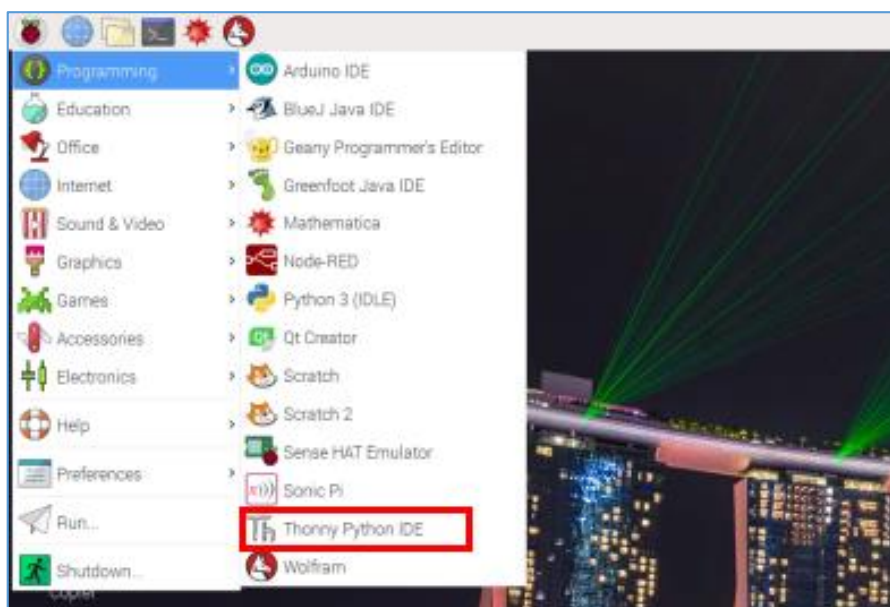


2. Pi Camera



การเขียนโปรแกรมด้วยภาษาไพทอนเพื่อทดสอบอุปกรณ์

1. เปิดโปรแกรม Thonny Python IDE โดยไปที่ Pi > Programming > Thonny Python IDE



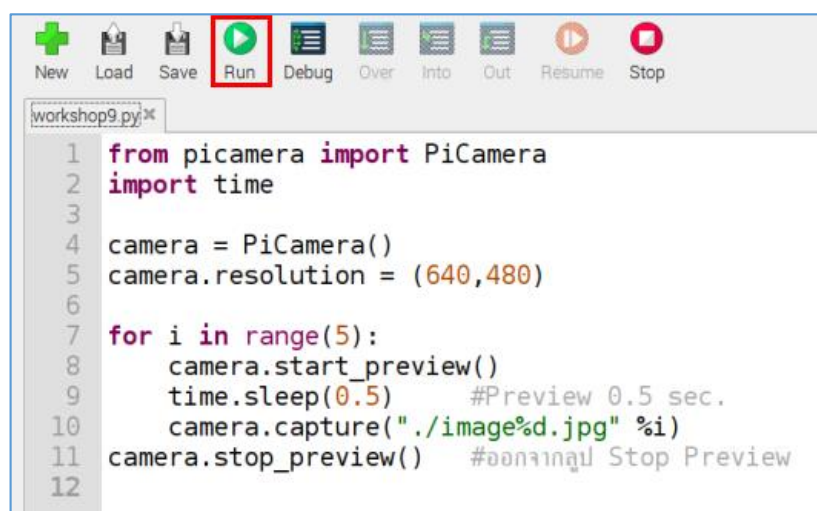
2. เขียนโปรแกรม ดังนี้

```

1  from picamera import PiCamera
2  import time
3
4  camera = PiCamera()
5  camera.resolution = (640,480)
6
7  for i in range(5):
8      camera.start_preview()
9      time.sleep(0.5)      #Preview 0.5 sec.
10     camera.capture("./image%d.jpg" %i)
11     camera.stop_preview()  #ออกจากลูป Stop Preview
12

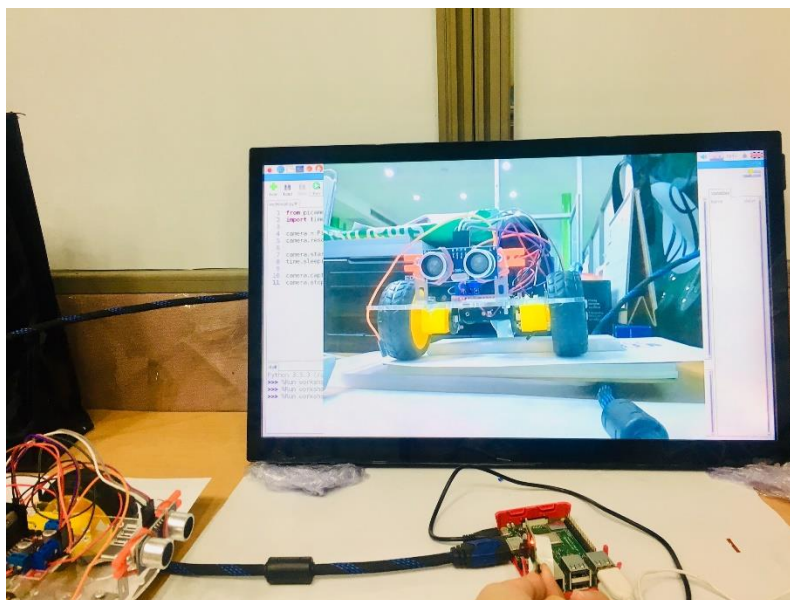
```

3. บันทึกไฟล์และรันโปรแกรม โดยคลิกปุ่ม Run

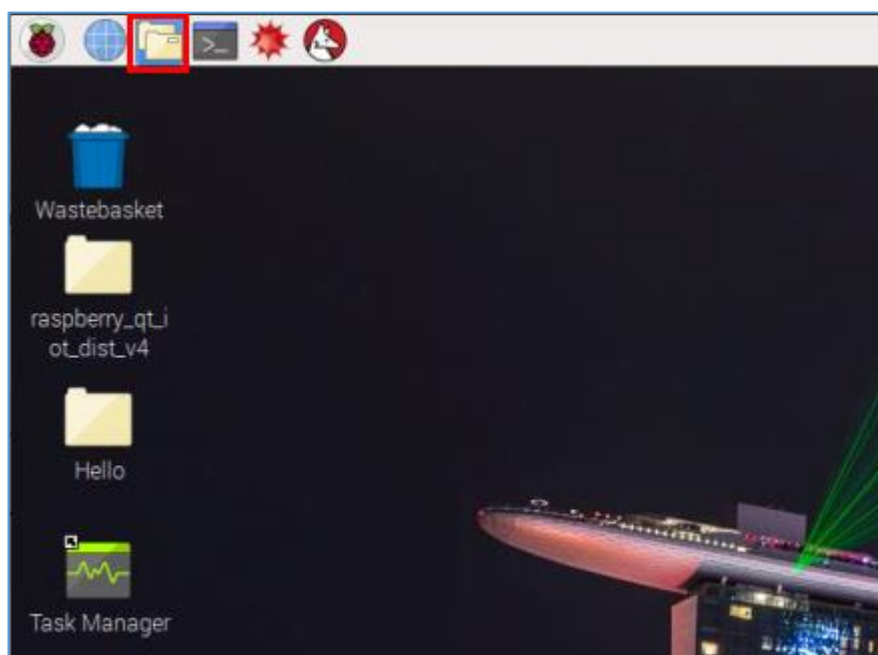


ผลลัพธ์ที่ได้

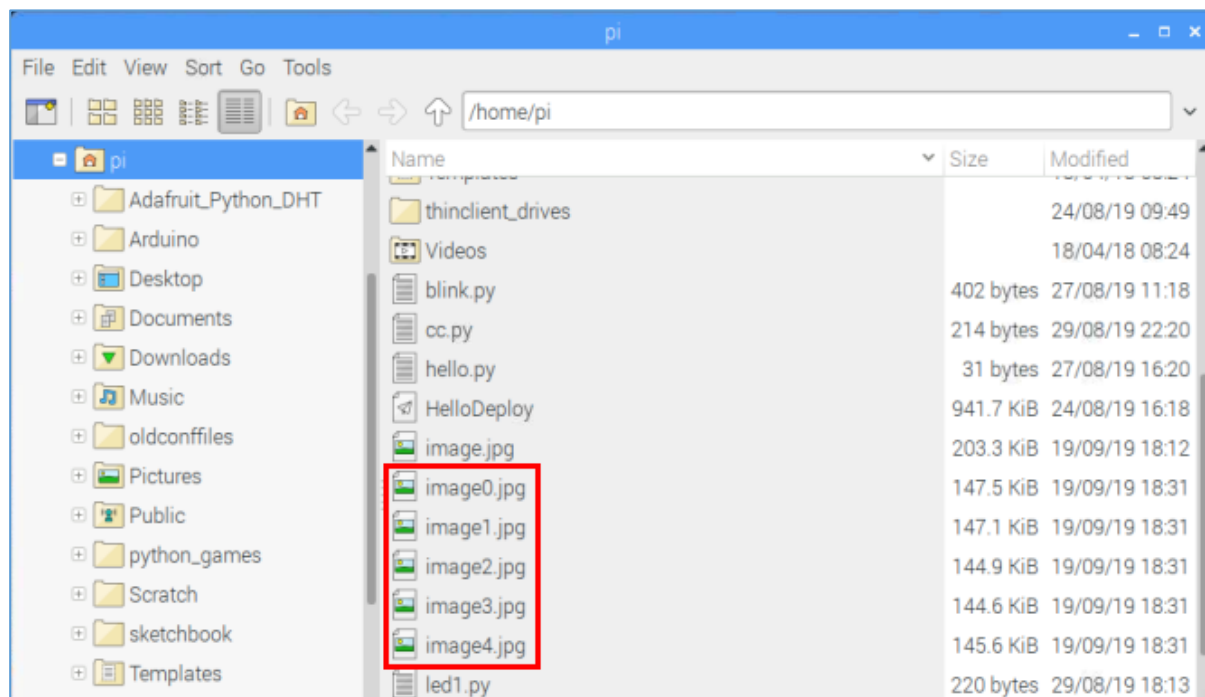
กล้องจะเปิดหน้า Preview ขึ้นมา 0.5 วินาที หลังจากนั้นกล้องจะถ่ายภาพ และจะวนซ้ำถ่ายภาพต่อเนื่อง 5 รอบ โดยบันทึกภาพที่มีขนาด 640 x 480 pixels และไฟล์รูปภาพที่ได้จะถูกบันทึกไว้ในตำแหน่งเดียวกับไฟล์โค้ดคำสั่ง โดยมีชื่อว่า image0.jpg, image1.jpg, image2.jpg, image3.jpg และ image4.jpg ตามลำดับ



คลิกเข้า File Manager



จะเจอรูปภาพที่ได้ถ่ายเอาไว้ ชื่อว่า image0.jpg, image1.jpg, image2.jpg, image3.jpg และ image4.jpg



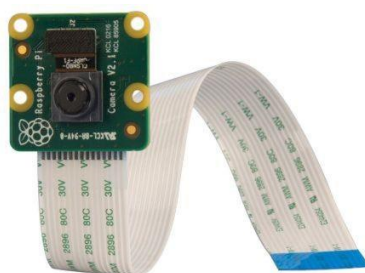
ฝึกทักษะการทดลองกิจกรรมที่ 11 เรียนรู้การใช้งาน Pi Camera แสดงภาพนิ่งบนหน้าจอ

อุปกรณ์ที่ใช้สำหรับการทดลอง

1. บอร์ด Raspberry Pi

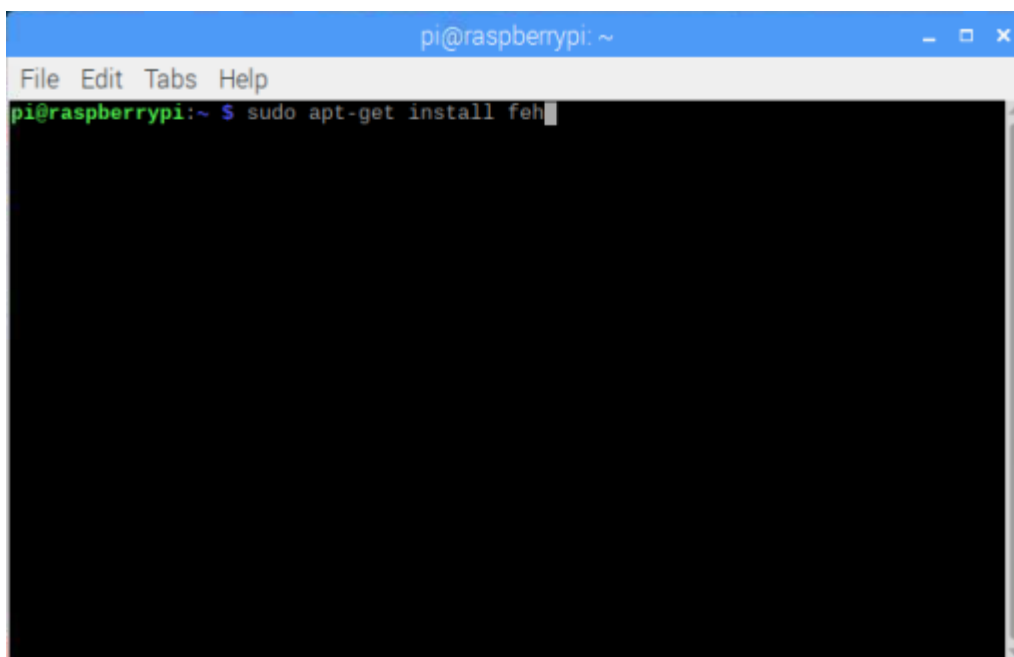


2. Pi Camera

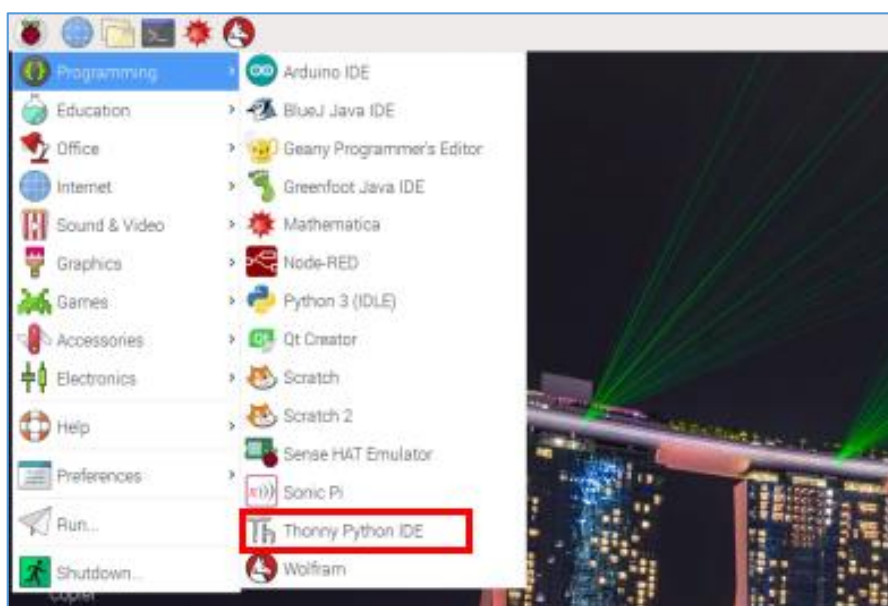


การเขียนโปรแกรมด้วยภาษาไพทอนเพื่อทดสอบอุปกรณ์

1. เชื่อมต่ออินเทอร์เน็ต และติดตั้งโปรแกรมที่ใช้แสดงภาพนิ่ง ในที่นี้จะติดตั้งโปรแกรม feh โดยเปิด Terminal แล้วพิมพ์คำสั่ง `sudo apt-get install feh`



2. เปิดโปรแกรม Thonny Python IDE โดยไปที่ Pi > Programming > Thonny Python IDE



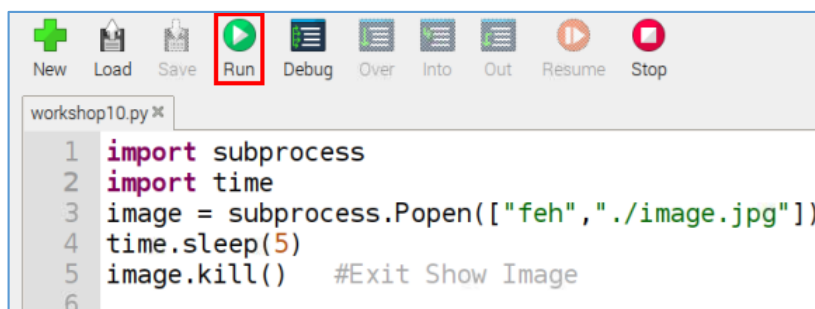
3. เขียนโปรแกรม ดังนี้

```

1 import subprocess
2 import time
3 image = subprocess.Popen(["feh", "./image.jpg"])
4 time.sleep(5)
5 image.kill() #Exit Show Image
6

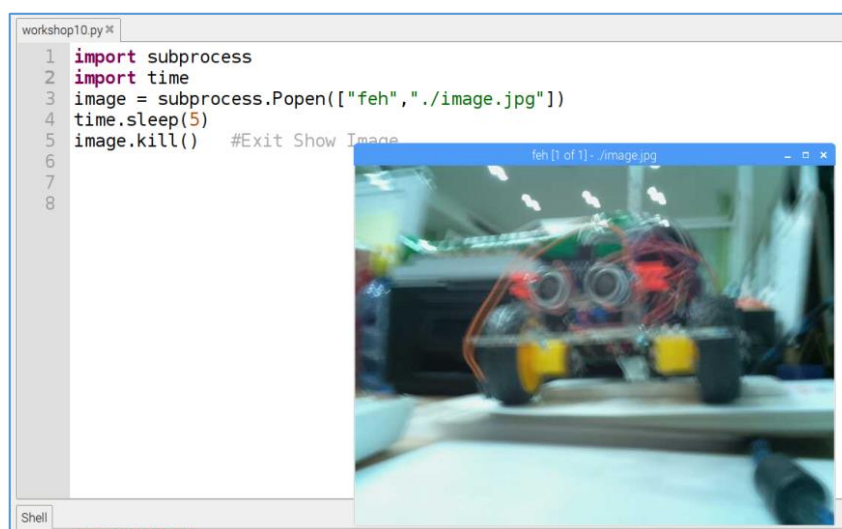
```

4. บันทึกไฟล์และรันโปรแกรม โดยคลิกปุ่ม Run

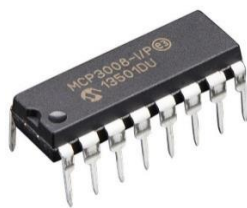


ผลลัพธ์ที่ได้

โปรแกรม feh จะแสดงรูปภาพที่ชื่อว่า image.jpg ขึ้นมาให้คุณเป็นเวลา 5 วินาที แล้วปิดหน้าต่างแสดงภาพโดยอัตโนมัติ

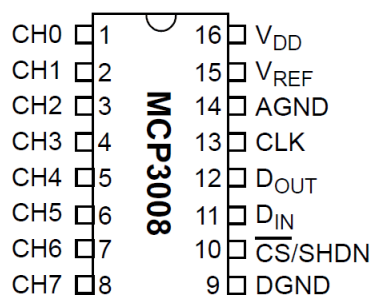


5. Chip แปลงสัญญาณ Analog to Digital (MCP3008)

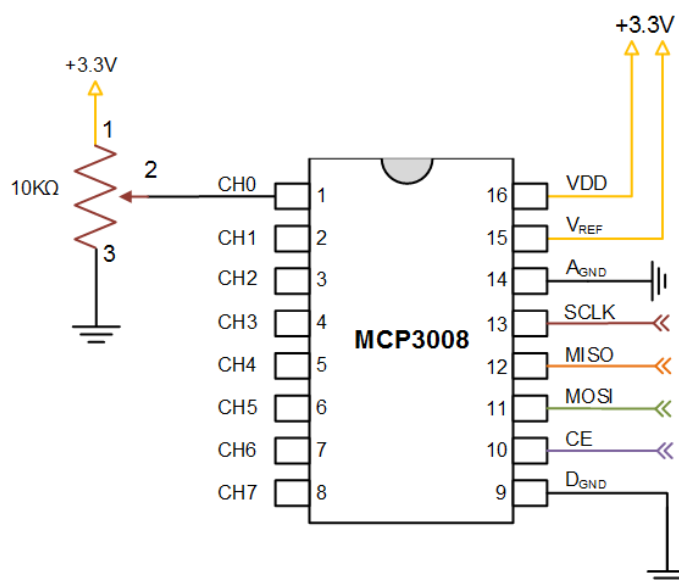


MCP3008 (Microchip) เป็น Chip แปลงสัญญาณ Analog ให้เป็นสัญญาณ Digital ด้วยจำนวน 8 Channels 10 Bit ADC โดยต่อผ่าน SPI interface MCP3008 นี้สามารถต่อใช้งานร่วมกับบอร์ด microcontroller อาทิเช่น Raspberry Pi ซึ่งตัวบอร์ด Raspberry Pi สามารถรับได้แต่สัญญาณ Digital เท่านั้น แต่สัญญาณที่เข้ามาเป็น Analog จึงต้องใช้ MCP3008 แปลงสัญญาณให้ก่อน

ขาของ MCP3008



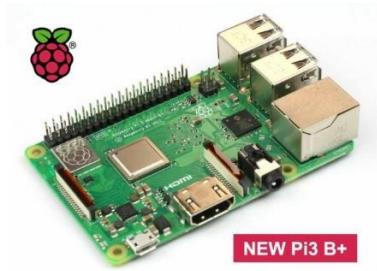
การต่อขาวงจรกับขา GPIO ของบอร์ด



ฝึกทักษะการทดลองกิจกรรมที่ 12 เรียนรู้การใช้งาน MCP3008

อุปกรณ์ที่ใช้สำหรับการทดลอง

1. บอร์ด Raspberry Pi



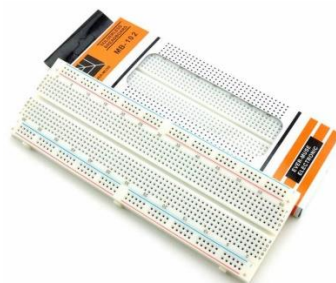
2. ชิป MCP3008



3. LDR Sensor



4. Breadboard



5. สาย Jumper

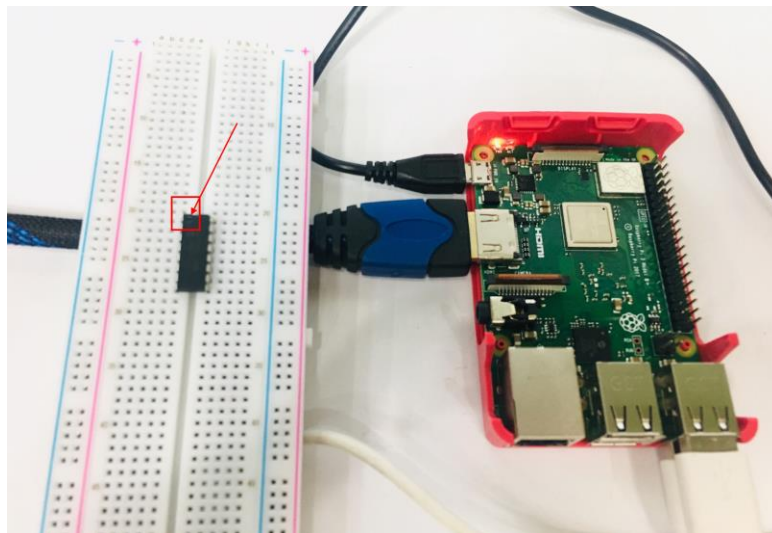


6. ตัวต้านทาน 4.7 กิโลโอห์ม

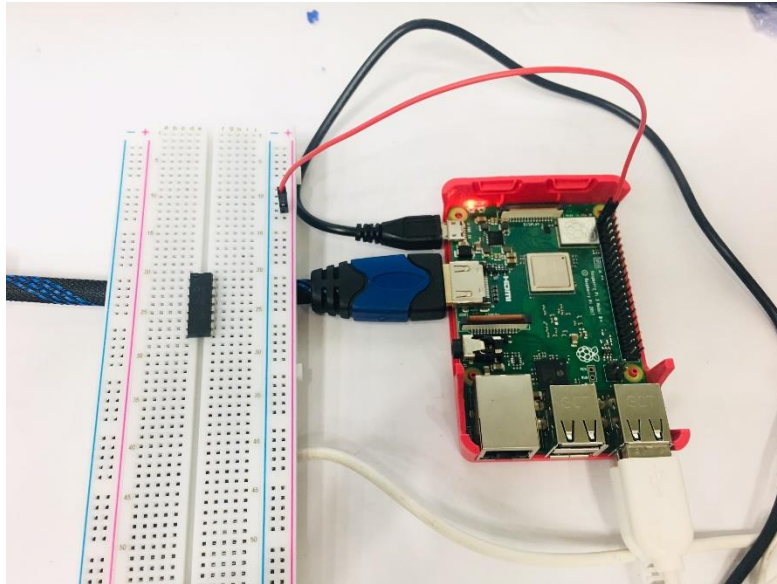


การต่อวงจร

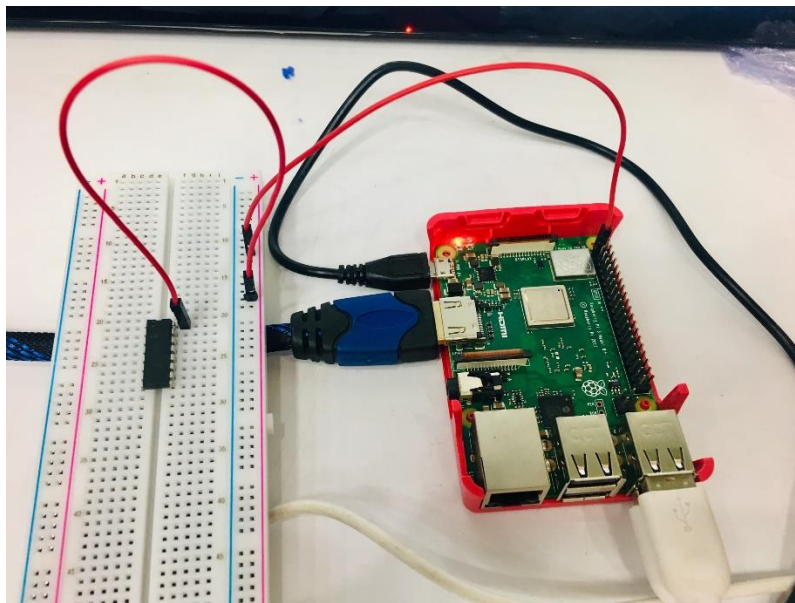
1. เสียบชิป MCP3008 ไว้ตรงกลาง Breadboard โดยสังเกตที่ตัวชิปจะมีจุดกลมๆ บนชิป ให้วางอยู่ด้านบน



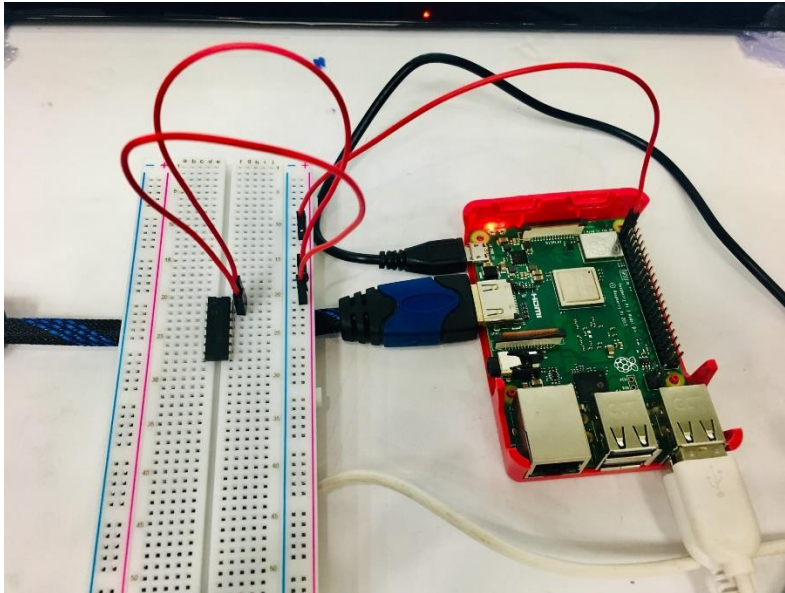
2. เสียบสาย Jumper จากขา PIN1 (3V3) ของขา GPIO ไปยังช่องใดก็ได้ในแถว + (สีแดง) ของ Breadboard



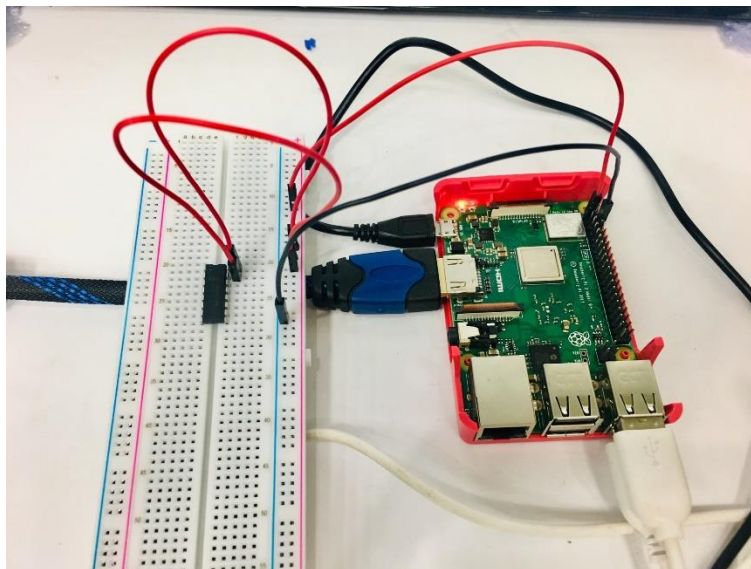
3. เสียบสาย Jumper จากขา VDD (ขาที่ 1 ขวามือ) ของชิป MCP3008 ไปยังแถว + (สีแดง) ของ Breadboard



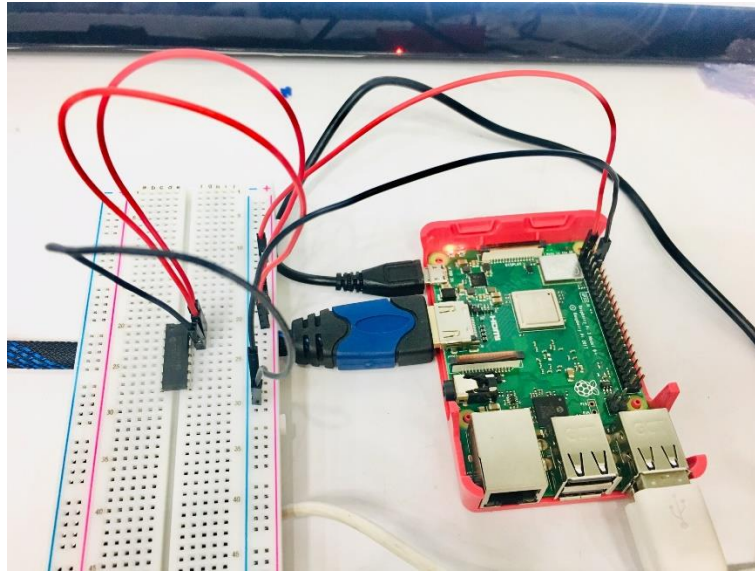
4. เสียบสาย Jumper จากขา VREF (ขาที่ 2 ขวามือ) ของชิป MCP3008 ไปยังแถว + (สีแดง) ของ Breadboard



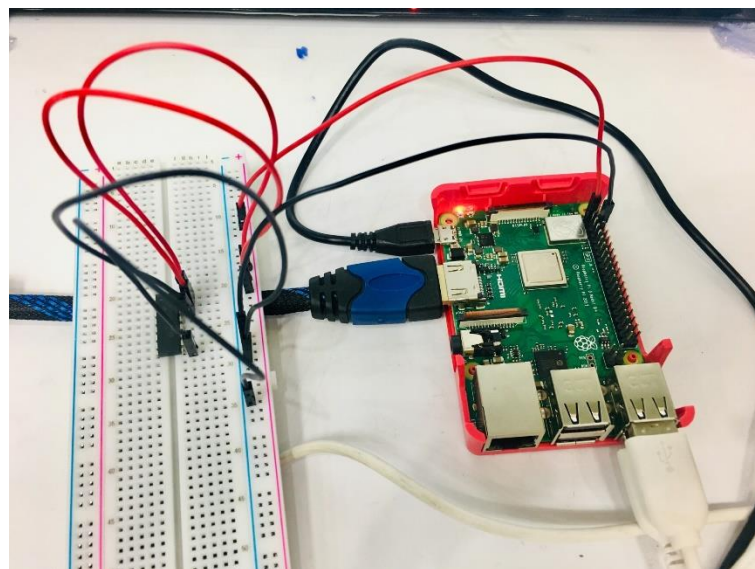
5. เสียบสาย Jumper จากขา PIN6 (Ground) ของขา GPIO ไปยังแถว - (สีฟ้า) ของ Breadboard



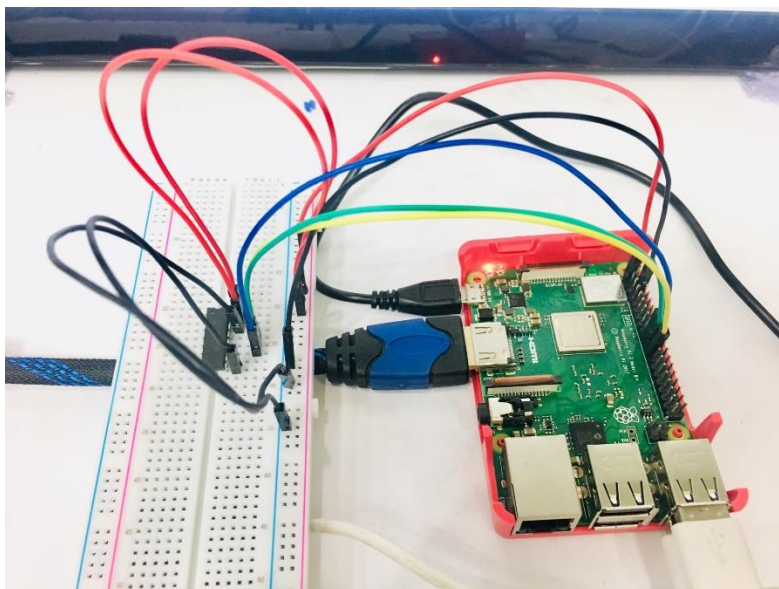
6. เสียบสาย Jumper จากขา AGND (ขาที่ 3 ขาวมือ) ของชิป MCP3008 ไปยังแถว - (สีฟ้า) ของ Breadboard



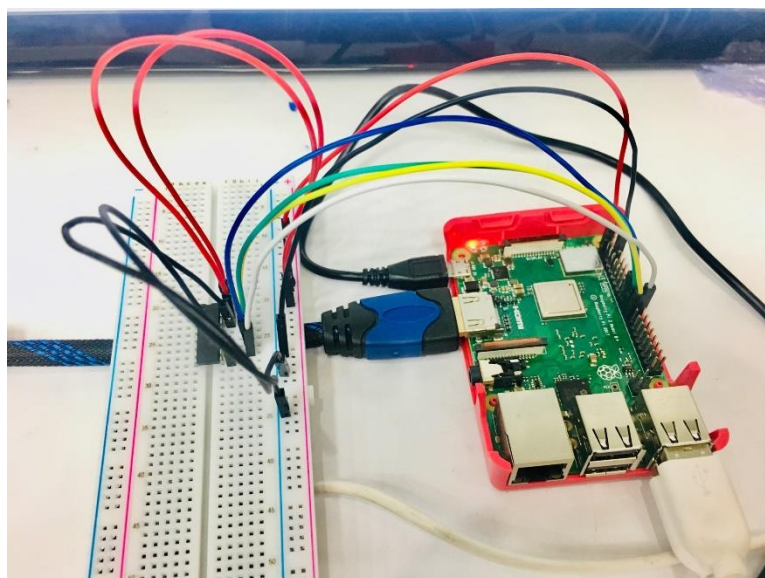
7. เสียบสาย Jumper จากขา DGND (ขาที่ 8 ขวามือ) ของชิป MCP3008 ไปยังแถว - (สีฟ้า) ของ Breadboard



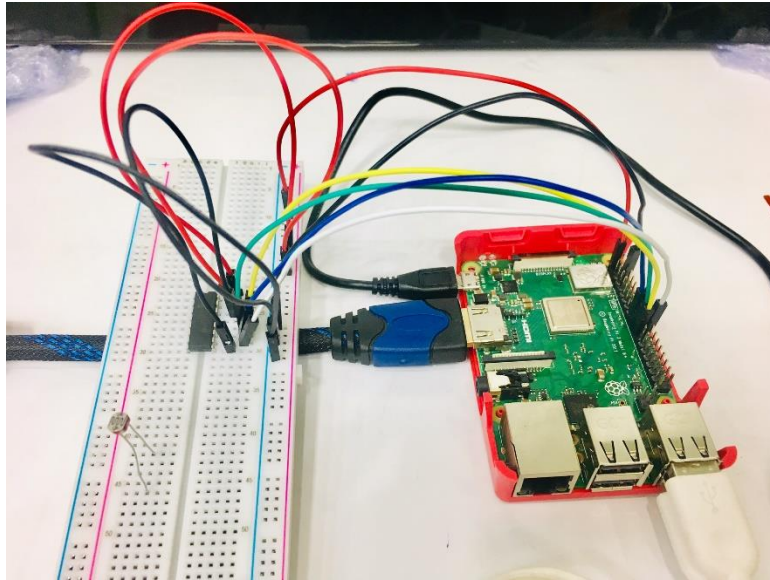
8. เสียบสาย Jumper จากขา CLK (ขาที่ 4 ขวามือ) ของชิป MCP3008 ไปยัง PIN23 (GPIO11 SP10_SCLK) ของขา GPIO



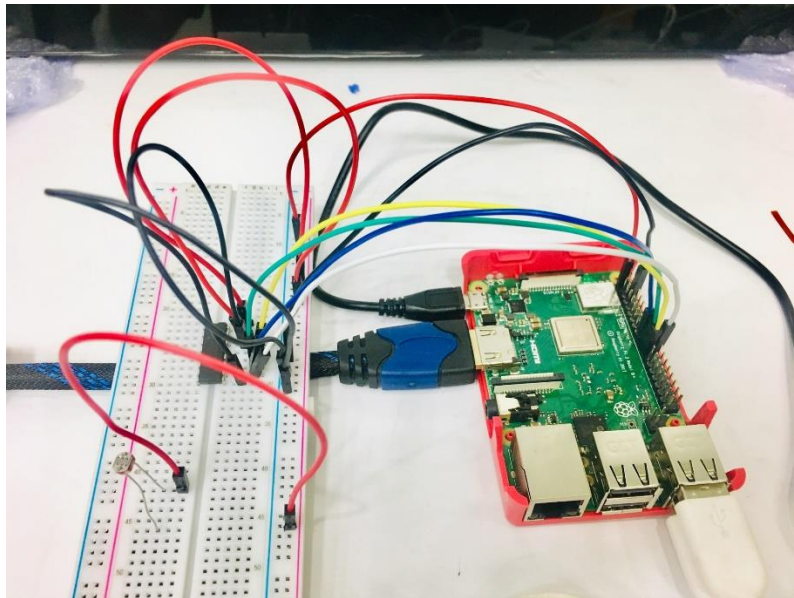
11. เสียบสาย Jumper จากขา CS/SHDN (ขาที่ 7 ขวามือ) ของชิป MCP3008 ไปยัง PIN24 (GPIO8 SP10_CE0_N) ของขา GPIO



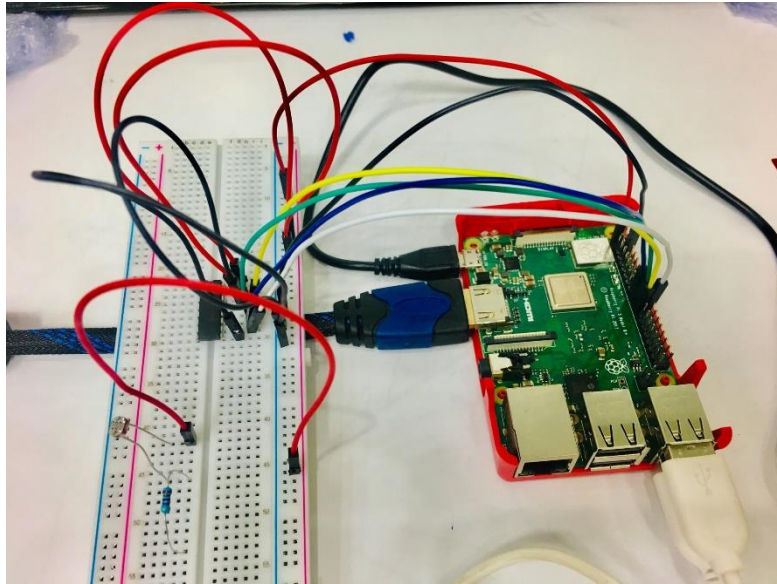
12. เสียบ LDR Sensor บน Breadboard



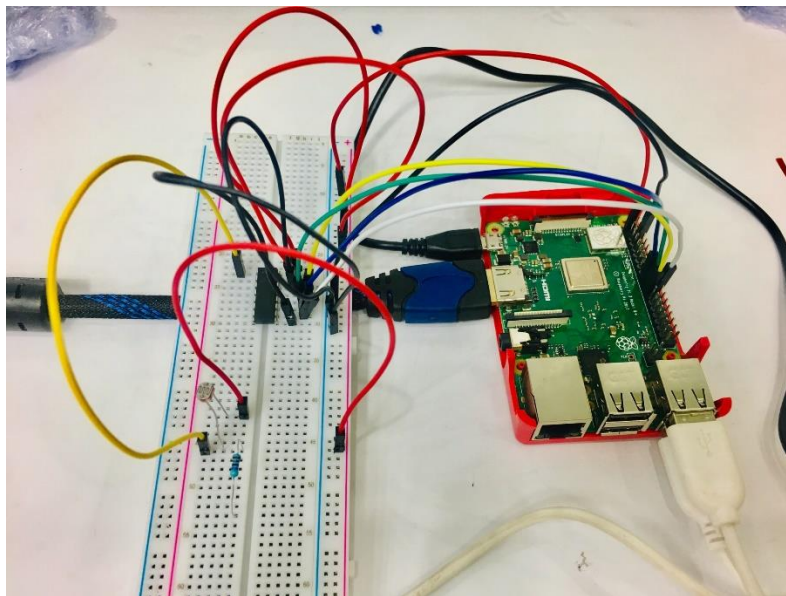
13. เสียบสาย Jumper บน Breadboard โดยต่อแถวเดียวกับขาหนึ่งของ LDR Sensor ไปยังแถว + (สีแดง) ของ Breadboard



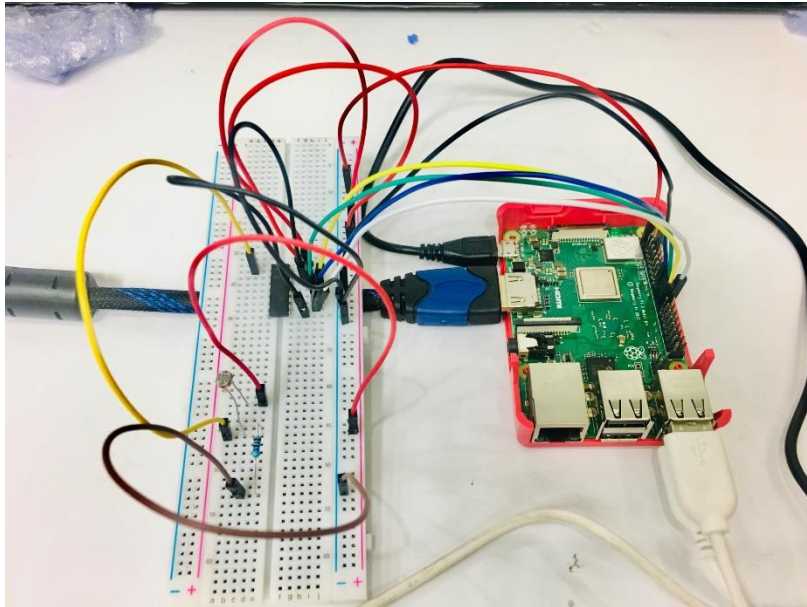
14. เสียบตัวต้านทาน 4.7 กิโลโอห์มเข้ากับ Breadboard ในแถวเดียวกับขาอีกข้างหนึ่งของ LDR Sensor



15. ใช้สาย Jumper เสียบบน Breadboard ในแถวเดียวกับขา LDR และตัวต้านทาน 4.7 กิโลโอห์ม ส่วนปลายสาย Jumper ต่อไปยังขา CH0 ของชิป MCP3008



16. เสียบสาย Jumper ที่ขาอีกด้านหนึ่งของตัวต้านทาน 4.7 กิโลโอห์ม แล้วต่อไปยังแถว - (สีฟ้า) ของ Breadboard



การเขียนโปรแกรมด้วยภาษาไพทอนเพื่อทดสอบอุปกรณ์

เนื่องจาก Raspberry Pi ไม่มีความสามารถในการรับค่าแบบ Analog โดยตรง (รับได้เฉพาะ Digital In/Out ซึ่งมีค่า 0 และ 1 เท่านั้น) ซึ่งไม่เหมือนกับ Arduino Board ซึ่งรับค่าได้ทั้งแบบ Digital และ Analog ดังนั้นจึงมีนำชิป MCP3008 มาใช้ในการแปลงสัญญาณ Analog ให้เป็น Digital ค่าที่อ่านได้จะมีค่าระหว่าง 0 – 1023

1. ในการใช้งานชิป MCP3008 จะต้องใช้ไลบรารีเพิ่มเติม ซึ่งจะต้องดาวน์โหลดและติดตั้งไลบรารี gpiozero โดยเปิด Terminal ขึ้นมา แล้วพิมพ์คำสั่งดังนี้

```
sudo apt-get update
```

```
sudo apt-get install python-pip python3-pip
```

```
sudo pip install gpiozero
```



```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ sudo apt-get update

```

```

pi@raspberrypi: ~
File Edit Tabs Help
Get:1 http://archive.raspberrypi.org/debian stretch InRelease [25.4 kB]
Get:2 http://raspbian.raspberrypi.org/raspbian stretch InRelease [15.0 kB]
Get:3 http://archive.raspberrypi.org/debian stretch/main armhf Packages [221 kB]
Get:4 http://raspbian.raspberrypi.org/raspbian stretch/main armhf Packages [11.7
  MB]
Get:5 http://raspbian.raspberrypi.org/raspbian stretch/contrib armhf Packages [5
  6.9 kB]
Fetched 12.0 MB in 28s (422 kB/s)
Reading package lists... Done
pi@raspberrypi:~ $ sudo apt-get install python-pip python3-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
python-pip is already the newest version (9.0.1-2+rpt2).
python3-pip is already the newest version (9.0.1-2+rpt2).
The following packages were automatically installed and are no longer required:
  arduino-core avr-libc avrdude binutils-avr coinor-libipopt1v5
  extra-xdg-menus gcc-avr libdirectfb-1.2-9 libftdi1 libiso9660-8 libjna-java
  libjna-jni libmumps-seq-4.10.0 libraw15 librx-tx-java libvcdinfo0 libvllcore8
  lxkeymap python-cairo python-gobject python-gobject-2 python-gtk2
  python-xklavier realpath
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 68 not upgraded.
pi@raspberrypi:~ $

```

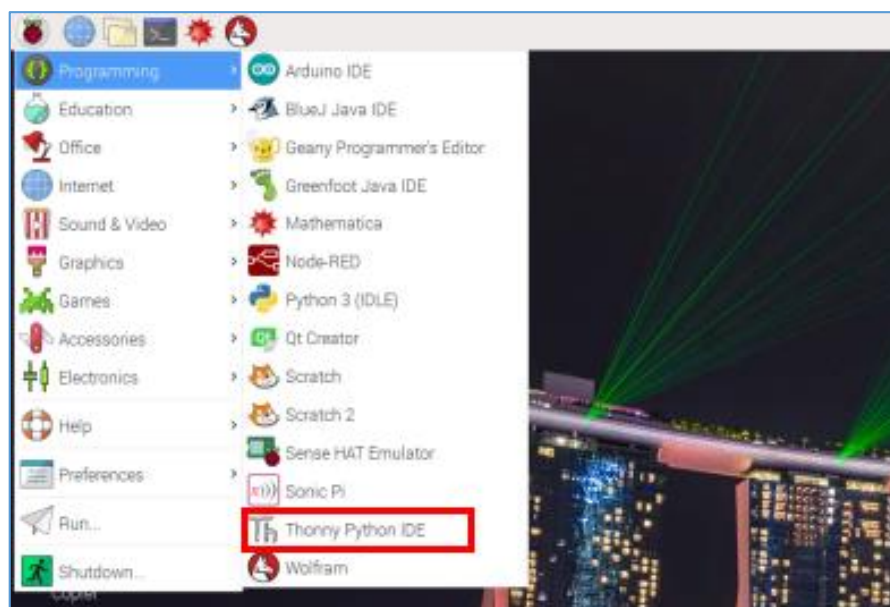
```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ sudo apt-get update
Get:1 http://archive.raspberrypi.org/debian stretch InRelease [25.4 kB]
Get:2 http://rasbian.raspberrypi.org/raspbian stretch InRelease [15.0 kB]
Get:3 http://archive.raspberrypi.org/debian stretch/main armhf Packages [221 kB]
Get:4 http://rasbian.raspberrypi.org/raspbian stretch/main armhf Packages [11.7 MB]
Get:5 http://rasbian.raspberrypi.org/raspbian stretch/contrib armhf Packages [56.9 kB]
Fetched 12.0 MB in 28s (422 kB/s)
Reading package lists... Done
pi@raspberrypi:~ $ sudo apt-get install python-pip python3-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
python-pip is already the newest version (9.0.1-2+rpt2).
python3-pip is already the newest version (9.0.1-2+rpt2).
The following packages were automatically installed and are no longer required:
  arduino-core avr-libc avrdude binutils-avr coinor-libipopt1v5
  extra-xdg-menus gcc-avr libdirectfb-1.2-9 libftdi1 libiso9660-8 libjna-java
  libjna-jni libmumps-seq-4.10.0 libraw15 librtx-java libvcdinfo0 libvllcore8
  lxkeymap python-cairo python-gobject python-gobject-2 python-gtk2
  python-xklavier realpath
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 68 not upgraded.
pi@raspberrypi:~ $ sudo pip install gpiozero

```

2. ติดตั้งเสร็จแล้ว ให้ปิดโปรแกรม Terminal

3. เปิดโปรแกรม Thonny Python IDE โดยไปที่ Pi > Programming > Thonny Python IDE



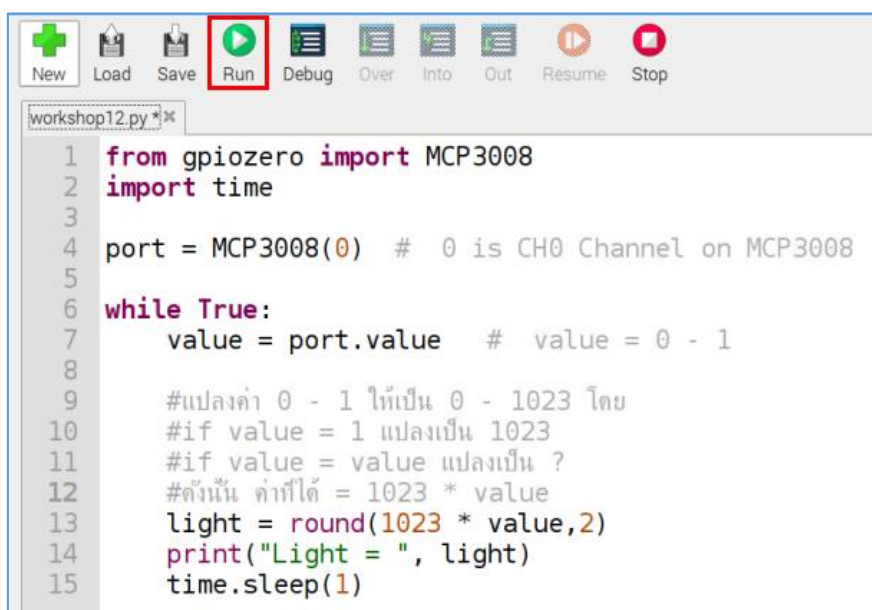
4. เขียนโปรแกรม ดังนี้

```

1 from gpiozero import MCP3008
2 import time
3
4 port = MCP3008(0) # 0 is CH0 Channel on MCP3008
5
6 while True:
7     value = port.value # value = 0 - 1
8
9     #แปลงค่า 0 - 1 ให้เป็น 0 - 1023 โดย
10    #if value = 1 แปลงเป็น 1023
11    #if value = value แปลงเป็น ?
12    #ดังนั้น ค่าที่ได้ = 1023 * value
13    light = round(1023 * value,2)
14    print("Light = ", light)
15    time.sleep(1)

```

5. บันทึกและรันโปรแกรม โดยคลิกปุ่ม Run



ผลลัพธ์ที่ได้

โดยปกติแล้วตัวเซนเซอร์เข็มของแสงจะวัดค่าออกมาเป็น Digital คือ มีค่าอยู่ระหว่าง 0 – 1 โดยถ้าวัดค่าแสงในที่ที่มีแสงสว่างน้อย (กลางคืน) ซึ่งอาจทดสอบโดยการเอามือปิดตัวเซนเซอร์ จะได้ค่าออกมาเป็นค่า 0 ถ้าแสงสว่างมาก (กลางวัน) ซึ่งอาจทดสอบโดยการเอามือที่ปิดตัวเซนเซอร์ออก จะได้ค่าออกมาเป็นค่า 1 แต่ในการนำไปใช้จริงจำเป็นที่จะต้องใช้ค่าข้อมูลที่มากกว่า 0 และ 1 จึงจำเป็นตัวแปลงสัญญาณ นั่นก็คือชิป MCP3008 ซึ่งในที่นี้ได้ขอหาเซนเซอร์ไปยังขา CH0 ของชิป MCP3008 เพื่อส่งค่าข้อมูลจากเซนเซอร์ไปแปลงที่ชิป ก่อนจะส่งข้อมูลไปยังขา GPIO

เมื่อทำการแปลงข้อมูลผ่านชิป MCP3008 จะเห็นว่าค่าที่ได้จะอยู่ระหว่าง 0.00 – 1.00 ซึ่งได้ค่าตัวเลขเป็นทศนิยมหลายตำแหน่ง หากต้องการเปลี่ยนจากค่าเลขทศนิยม 0.00 – 1.00 ไปเป็นค่า 0 – 1023 จึงจะต้องเขียนโค้ดคำสั่งในการแปลง โดยใช้หลักการเทียบบัญญัติไตรยางศ์ ซึ่งทำให้ได้สูตรออกมา คือ $1023 * \text{value}$

จากการเขียนโค้ดคำสั่ง จะเห็นว่า LDR Sensor จะวัดค่าแสงทุกๆ 1 วินาที เมื่อ LDR Sensor วัดค่าแสงในที่ที่มีแสงสว่างน้อย จะได้ค่าออกมาเป็นค่าน้อย ถ้าแสงสว่างมาก จะได้ค่าออกมาเป็นค่ามาก

```

Shell
Light = 334.34
Light = 0.5
Light = 0.5
Light = 89.46
Light = 81.46
Light = 328.34
Light = 336.34
Light = 538.24
Light = 671.17
Light = 705.16
Light = 743.14
Light = 756.13
  
```

อุปกรณ์อื่นๆ ที่ใช้กับบอร์ด Raspberry Pi

1. LCD Display

LCD ย่อมาจาก Liquid Crystal Display ซึ่งเป็นจอแสดงผลแบบ (Digital) โดยภาพที่ปรากฏขึ้นเกิดจากแสงที่ถูกปล่อยออกมาจากหลอดไฟด้านหลังของจอภาพ (Back Light) ผ่านชั้นกรองแสง (Polarized filter) แล้ววิ่งไปยัง คริสตัลเหลวที่เรียงตัวด้วยกัน 3 เซลล์คือ แสงสีแดง แสงสีเขียวและแสงสีน้ำเงิน กลายเป็น พิกเซล (Pixel) ที่สว่างสดใสเกิดขึ้น

หมายเหตุ ชิปลิ์ค่าที่อยู่ด้านหลัง LCD Display คือ I2C Bus ย่อมาจาก Inter Integrate Circuit Bus (IIC) (ไอ-สแควร์-ซี-บัส) เป็นการสื่อสารอนุกรมแบบซิงโครนัส (Synchronous) เพื่อใช้ติดต่อสื่อสารระหว่าง ไมโครคอนโทรลเลอร์กับอุปกรณ์ภายนอก ซึ่งถูกพัฒนาขึ้นโดยบริษัท Philips Semiconductors โดยใช้สายสัญญาณเพียง 2 เส้นเท่านั้น คือ serial data (SDA) และสาย serial clock (SCL) ซึ่งสามารถเชื่อมต่ออุปกรณ์จำนวนหลายๆ ตัวเข้าด้วยกันได้ ทำให้ MCU ใช้พอร์ตเพียง 2 พอร์ตเท่านั้น

ฝึกทักษะการทดลองกิจกรรมที่ 13 เรียนรู้การใช้งาน LCD Display แบบที่ 1

อุปกรณ์ที่ใช้สำหรับการทดลอง

1. บอร์ด Raspberry Pi



2. สาย Jumper

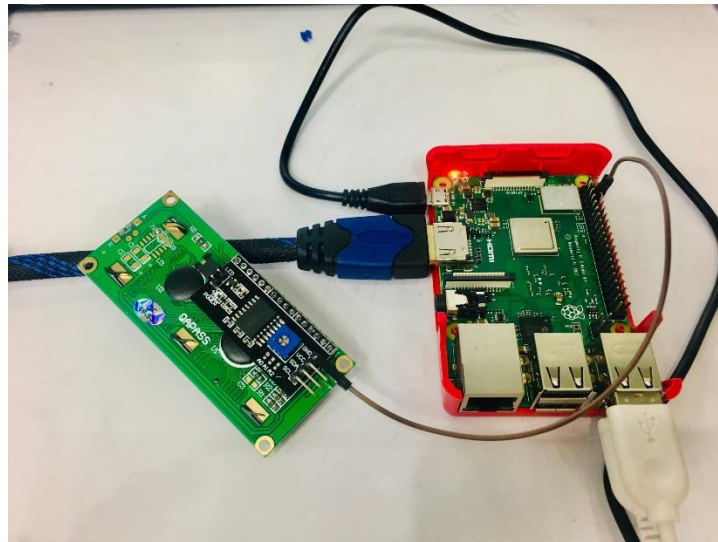


3. LCD Display 16x2 via I2C

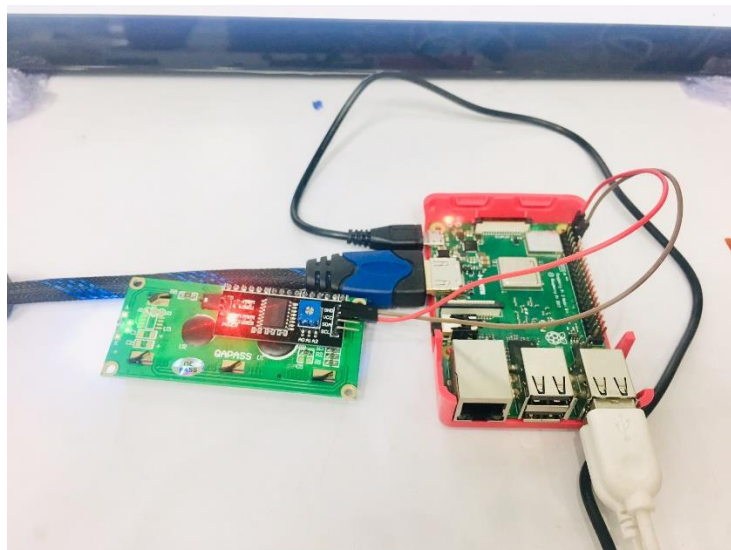


การต่อวงจร

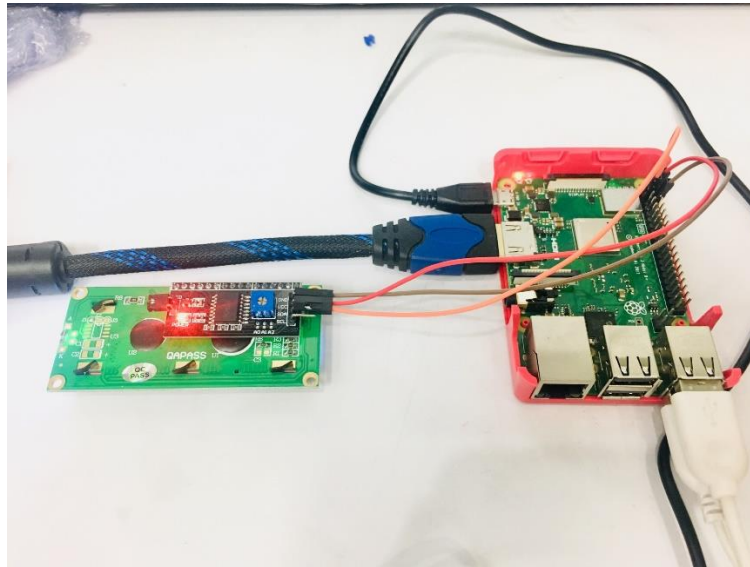
1. เสียบสาย Jumper เข้ากับขา GND ของ LCD Display ปลายสาย Jumper เสียบเข้ากับขา PIN6 (GROUND) ของ GPIO



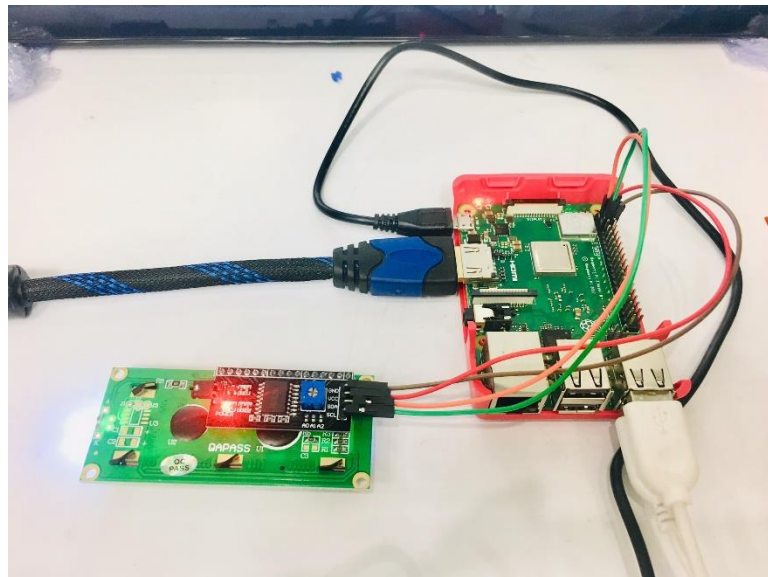
2. เสียบสาย Jumper เข้ากับขา VCC ของ LCD Display ปลายสาย Jumper เสียบเข้ากับขา PIN2 (5V) ของ GPIO



3. เสียบสาย Jumper เข้ากับขา SDA ของ LCD Display ปลายสาย Jumper เสียบเข้ากับขา PIN3 (GPIO2 SDA1) ของ GPIO



4. เสียบสาย Jumper เข้ากับขา SCL ของ LCD Display ปลายสาย Jumper เสียบเข้ากับขา PIN5 (GPIO3 SCL1) ของ GPIO



การเขียนโปรแกรมด้วยภาษาไพทอนเพื่อทดสอบอุปกรณ์

1. ติดตั้ง LCD i2c Driver โดยเปิด Terminal แล้วพิมพ์คำสั่ง `git clone https://github.com/the-raspberry-pi-guy/lcd`


```
pi@raspberrypi:~ $ git clone https://github.com/the-raspberry-pi-guy/lcd
```

2. เรียกดูไฟล์ในโฟลเดอร์ที่ดาวน์โหลดเข้ามา โดยพิมพ์คำสั่ง **ls**

```
pi@raspberrypi:~ $ git clone https://github.com/the-raspberry-pi-guy/lcd
Cloning into 'lcd'...
remote: Enumerating objects: 78, done.
remote: Total 78 (delta 0), reused 0 (delta 0), pack-reused 78
Unpacking objects: 100% (78/78), done.
pi@raspberrypi:~ $ ls
Desktop  Downloads  MagPi  Pictures  Templates  lcd
Documents LCD-show  Music  Public   Videos
```

3. เข้าไปในโฟลเดอร์ lcd โดยพิมพ์คำสั่ง **cd lcd**

```
pi@raspberrypi:~ $ git clone https://github.com/the-raspberry-pi-guy/lcd
Cloning into 'lcd'...
remote: Enumerating objects: 78, done.
remote: Total 78 (delta 0), reused 0 (delta 0), pack-reused 78
Unpacking objects: 100% (78/78), done.
pi@raspberrypi:~ $ ls
Desktop  Downloads  MagPi  Pictures  Templates  lcd
Documents LCD-show  Music  Public   Videos
pi@raspberrypi:~ $ cd lcd
pi@raspberrypi:~/lcd $
```

4. เรียกดูไฟล์ในโฟลเดอร์ lcd โดยพิมพ์คำสั่ง **ls**

```
pi@raspberrypi:~ $ git clone https://github.com/the-raspberry-pi-guy/lcd
Cloning into 'lcd'...
remote: Enumerating objects: 78, done.
remote: Total 78 (delta 0), reused 0 (delta 0), pack-reused 78
Unpacking objects: 100% (78/78), done.
pi@raspberrypi:~ $ ls
Desktop  Downloads  MagPi  Pictures  Templates  lcd
Documents LCD-show  Music  Public   Videos
pi@raspberrypi:~ $ cd lcd
pi@raspberrypi:~/lcd $ ls
README.md      demo_lcd.py          i2c_lib.py  installConfigs
demo_clock.py  demo_scrolling_text.py  install.sh  lcddriver.py
pi@raspberrypi:~/lcd $
```

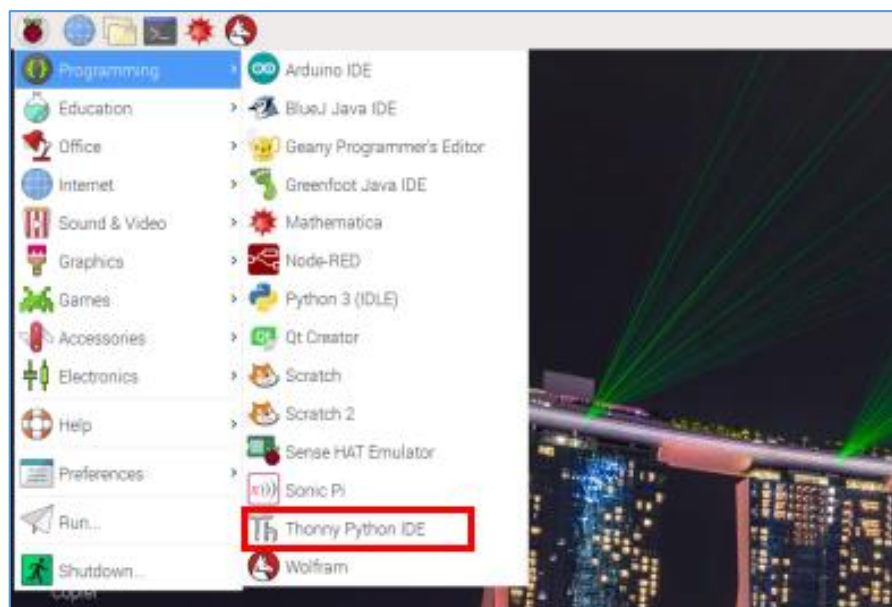
5. ทำการติดตั้ง โดยพิมพ์คำสั่ง **sudo sh install.sh**

```

pi@raspberrypi:~ $ git clone https://github.com/the-raspberry-pi-guy/lcd
Cloning into 'lcd'...
remote: Enumerating objects: 78, done.
remote: Total 78 (delta 0), reused 0 (delta 0), pack-reused 78
Unpacking objects: 100% (78/78), done.
pi@raspberrypi:~ $ ls
Desktop  Downloads  MagPi  Pictures  Templates  lcd
Documents  LCD-show  Music  Public  Videos
pi@raspberrypi:~ $ cd lcd
pi@raspberrypi:~/lcd $ ls
README.md      demo_lcd.py      i2c_lib.py  installConfigs
demo_clock.py  demo_scrolling_text.py  install.sh  lcddriver.py
pi@raspberrypi:~/lcd $ sudo sh install.sh

```

6. เมื่อติดตั้งเสร็จแล้ว เครื่อง Raspberry Pi จะทำการ Reboot โดยอัตโนมัติ
7. เปิดโปรแกรม Thonny Python IDE โดยไปที่ Pi > Programming > Thonny Python IDE



8. เขียนโปรแกรม ดังนี้

```

1  import lcddriver
2  import time
3
4  display = lcddriver.lcd()
5
6  while True:
7      display.lcd_display_string("Hello",1)
8      display.lcd_display_string("Raspberry Pi",2)

```

10. บันทึกไฟล์โค้ดคำสั่งไว้ในโฟลเดอร์ lcd ซึ่งอยู่ในโฟลเดอร์ /home/pi/

ผลลัพธ์ที่ได้

หน้าจอ LCD จะปรากฏข้อความบนแถวแรกว่า Hello ส่วนแถวล่าง จะปรากฏข้อความว่า Raspberry Pi

ฝึกทักษะการทดลองกิจกรรมที่ 14 เรียนรู้การใช้งาน LCD Display แบบที่ 2 อุปกรณ์ที่ใช้สำหรับการทดลอง

1. บอร์ด Raspberry Pi



2. สาย Jumper

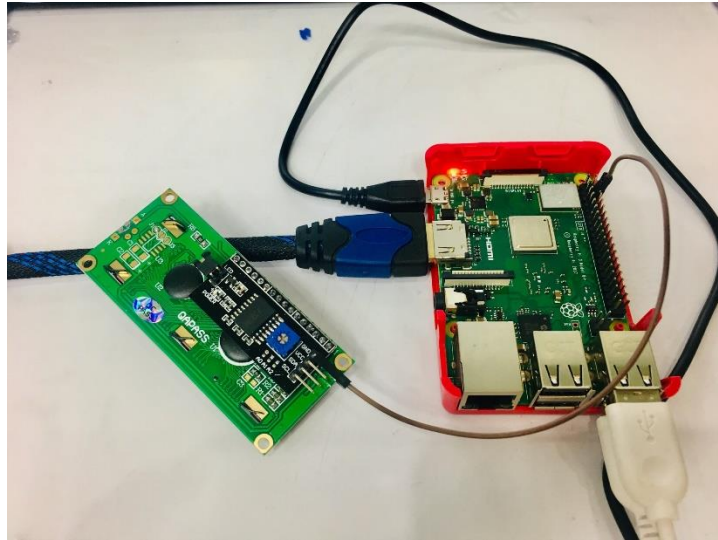


3. LCD Display 16x2 via I2C

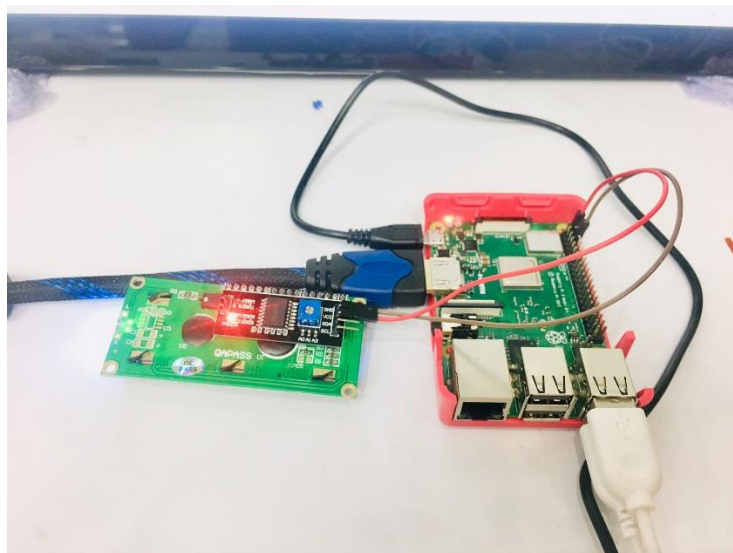


การต่อวงจร

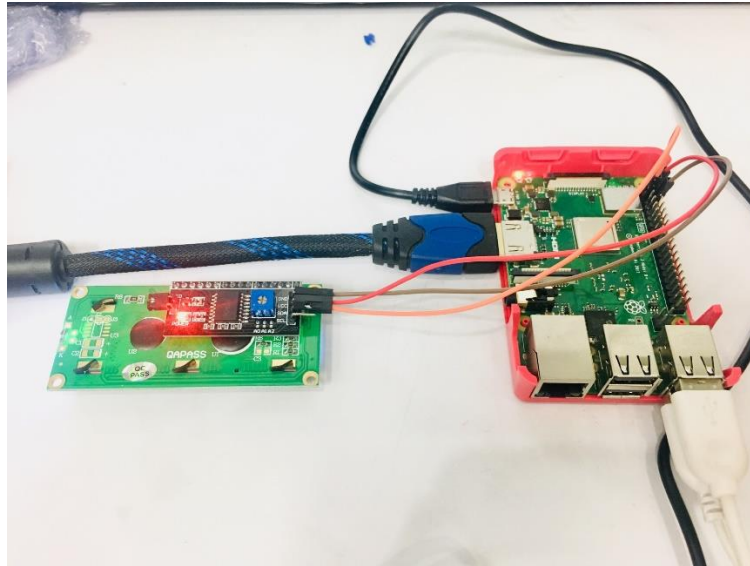
1. เสียบสาย Jumper เข้ากับขา GND ของ LCD Display ปลายสาย Jumper เสียบเข้ากับขา PIN6 (GROUND) ของ GPIO



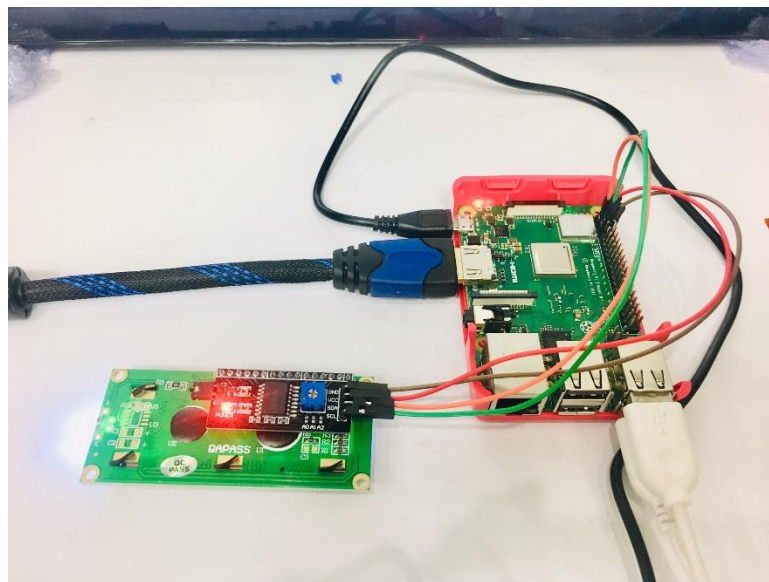
2. เสียบสาย Jumper เข้ากับขา VCC ของ LCD Display ปลายสาย Jumper เสียบเข้ากับขา PIN2 (5V) ของ GPIO



3. เสียบสาย Jumper เข้ากับขา SDA ของ LCD Display ปลายสาย Jumper เสียบเข้ากับขา PIN3 (GPIO2 SDA1) ของ GPIO



4. เสียบสาย Jumper เข้ากับขา SCL ของ LCD Display ปลายสาย Jumper เสียบเข้ากับขา PIN5 (GPIO3 SCL1) ของ GPIO



การเขียนโปรแกรมด้วยภาษาไพทอนเพื่อทดสอบอุปกรณ์

1. ทำการ Enable I2C Module โดยเปิดโปรแกรม Terminal แล้วพิมพ์คำสั่ง `sudo raspi-config`


```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ sudo raspi-config

```

2. เลือก 5 Interfacing Options และเลือก P5 I2C (Enable/Disable automatic loading...) แล้วเลือก

Yes

```

Raspberry Pi Software Configuration Tool (raspi-config)

1 Change User Password Change password for the current user
2 Network Options       Configure network settings
3 Boot Options          Configure options for start-up
4 Localisation Options  Set up language and regional settings to match your
5 Interfacing Options   Configure connections to peripherals
6 Overclock             Configure overclocking for your Pi
7 Advanced Options      Configure advanced settings
8 Update                Update this tool to the latest version
9 About raspi-config    Information about this configuration tool

<Select> <Finish>

```

```

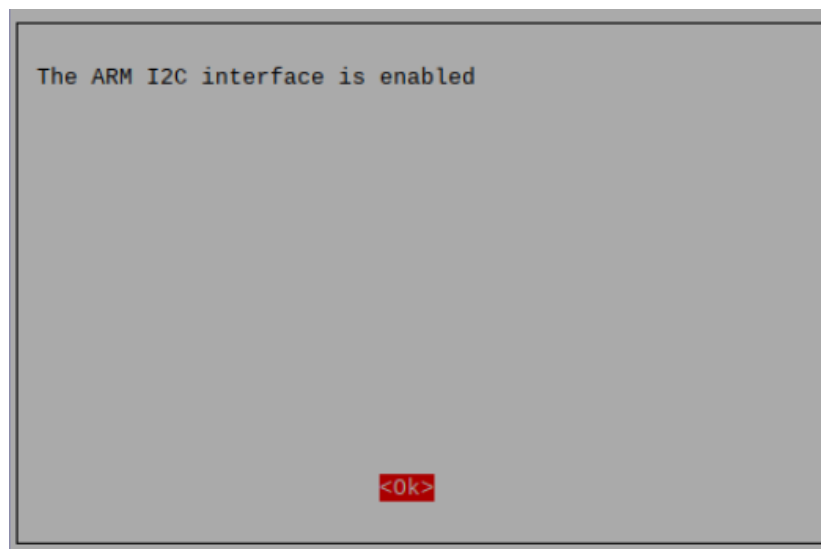
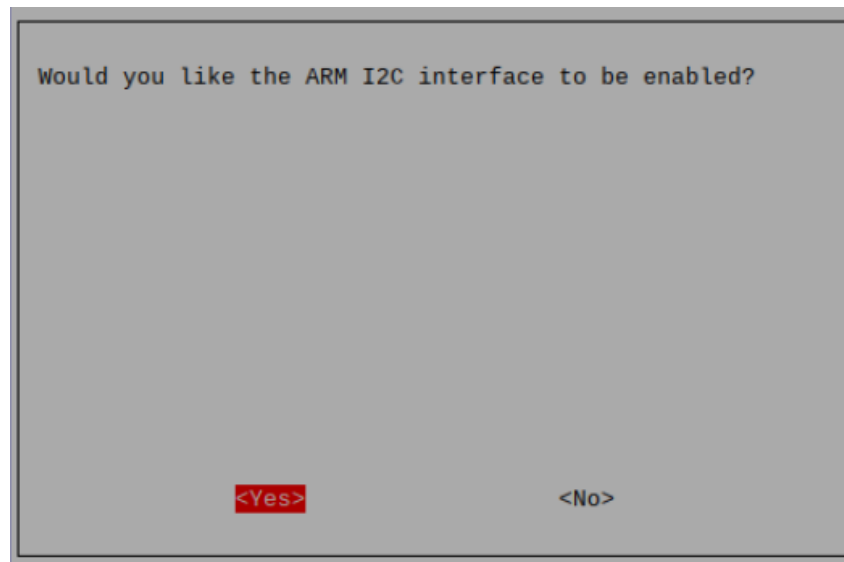
Raspberry Pi Software Configuration Tool (raspi-config)

P1 Camera      Enable/Disable connection to the Raspberry Pi Camera
P2 SSH          Enable/Disable remote command line access to your Pi using
P3 VNC          Enable/Disable graphical remote access to your Pi using Rea
P4 SPI          Enable/Disable automatic loading of SPI kernel module
P5 I2C          Enable/Disable automatic loading of I2C kernel module
P6 Serial       Enable/Disable shell and kernel messages on the serial conn
P7 1-Wire       Enable/Disable one-wire interface
P8 Remote GPIO  Enable/Disable remote access to GPIO pins

<Select> <Back>

```

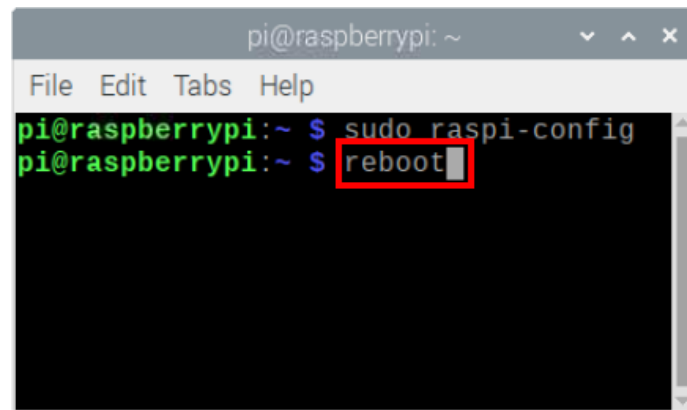
3. ยืนยันด้วยการตอบ Yes จากนั้นคลิกที่ OK



4. กดปุ่ม Tab แล้วเลือก **Finish**

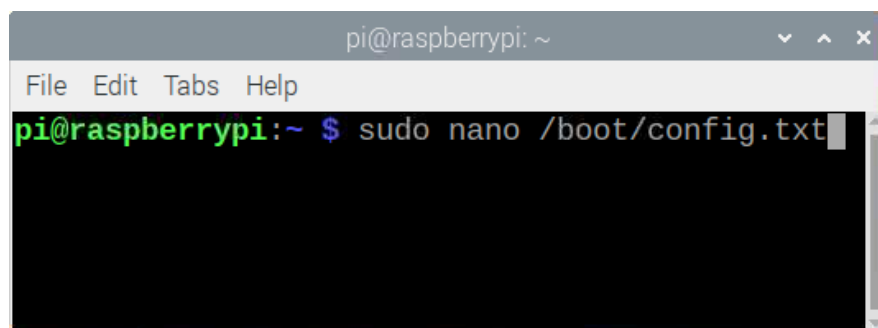


5. Reboot 1 ครั้ง โดยพิมพ์คำสั่ง **reboot**



```
pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ sudo raspi-config
pi@raspberrypi:~ $ reboot
```

6. ทำการอัปเดตไฟล์ `/boot/config.txt` โดยเปิดโปรแกรม Terminal แล้วพิมพ์คำสั่ง **sudo nano**
`/boot/config.txt`

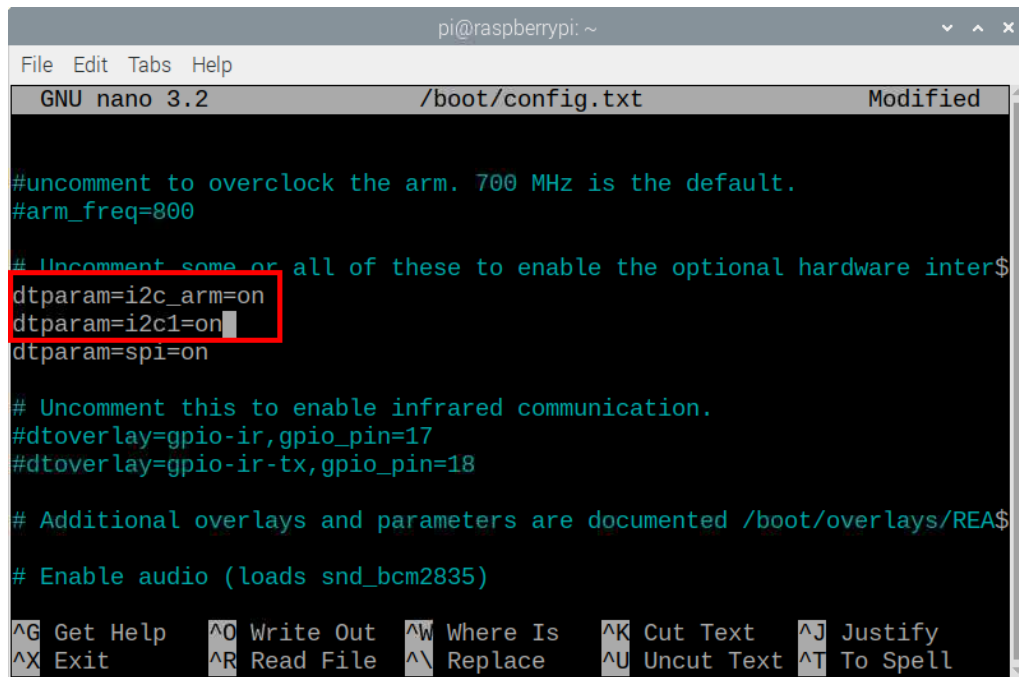


```
pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ sudo nano /boot/config.txt
```

7. ใส่ข้อความ (หรือ uncomment) ต่อไปนี้

dtoverlay=i2c1=on

dtoverlay=i2c_arm=on



```

pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 3.2 /boot/config.txt Modified

#uncomment to overclock the arm. 700 MHz is the default.
#arm_freq=800

# Uncomment some or all of these to enable the optional hardware inter$
dtparam=i2c_arm=on
dtparam=i2c1=on
dtparam=spi=on

# Uncomment this to enable infrared communication.
#dtoverlay=gpio-ir,gpio_pin=17
#dtoverlay=gpio-ir-tx,gpio_pin=18

# Additional overlays and parameters are documented /boot/overlays/REA$

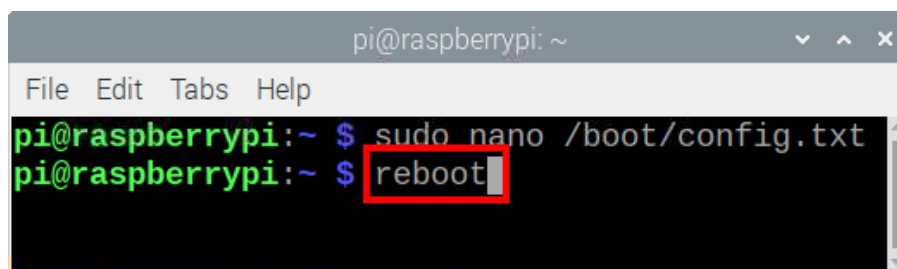
# Enable audio (loads snd_bcm2835)

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Spell

```

8. บันทึกการแก้ไขไฟล์ โดยกดปุ่ม **Ctrl + X** แล้วพิมพ์ **Y** เพื่อยืนยัน หลังจากนั้นกดปุ่ม Enter

9. Reboot 1 ครั้ง โดยพิมพ์คำสั่ง **reboot**



```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~$ sudo nano /boot/config.txt
pi@raspberrypi:~$ reboot

```

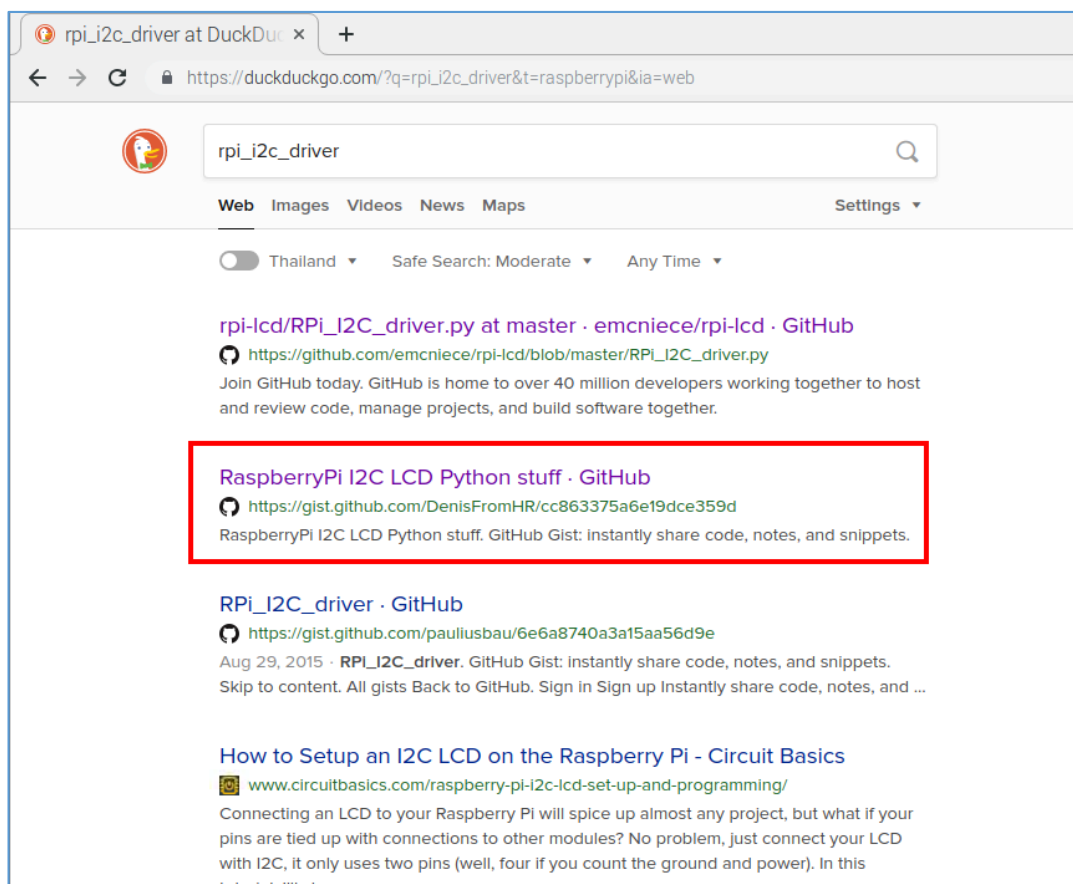
10. ลองใช้คำสั่ง **sudo i2cdetect -y 1** เพื่อดูว่าเจอ I2C Module หรือไม่ ซึ่งผลที่ได้คือ address ของอุปกรณ์ (อาจแตกต่างกันไปในแต่ละเครื่อง ซึ่งในที่นี้คือ 0x27)

```

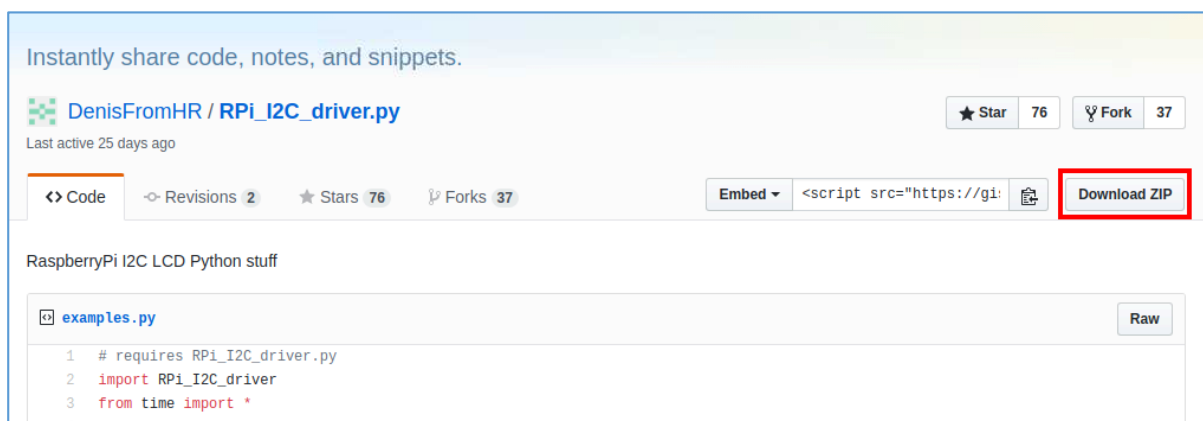
pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ sudo i2cdetect -y 1
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  27  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
pi@raspberrypi:~ $

```

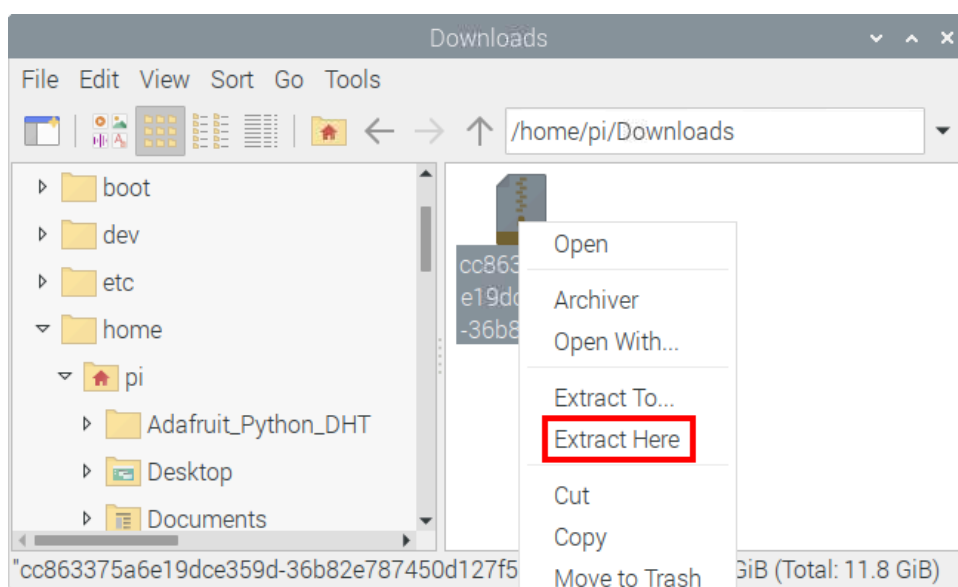
11. ดาวน์โหลด library RPi_I2C_Driver มาใช้งาน โดยเปิด Web Browser ขึ้นมา แล้วค้นหาคำว่า “RPi_I2C_Driver” แล้วเลือกคลิกเว็บไซต์ดังรูป



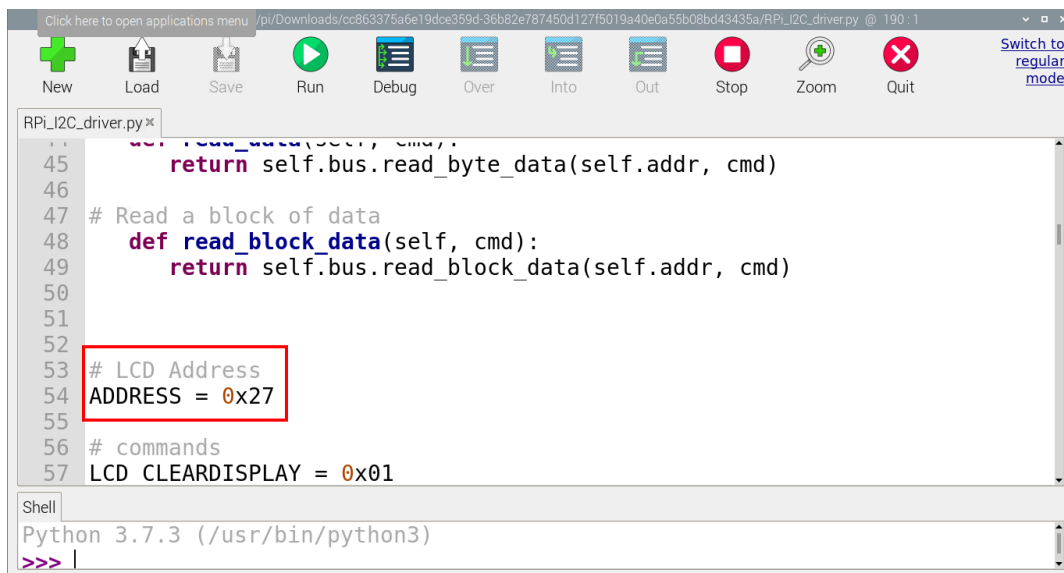
12. ทำการดาวน์โหลด โดยคลิกที่ Download ZIP



13. ไฟล์ที่ดาวน์โหลดจะอยู่ในโฟลเดอร์ /home/pi/Downloads ให้คลิกขวาที่ไฟล์แล้วเลือกคำสั่ง Extract Here เพื่อแตกไฟล์



14. เข้าไปในโฟลเดอร์ แล้วดับเบิลคลิกไฟล์ RPi_I2C_driver แล้วตรวจสอบ Address ของ LCD จะต้องตรงกับตัวเลขที่ตรวจสอบได้จากข้อ 10.



```

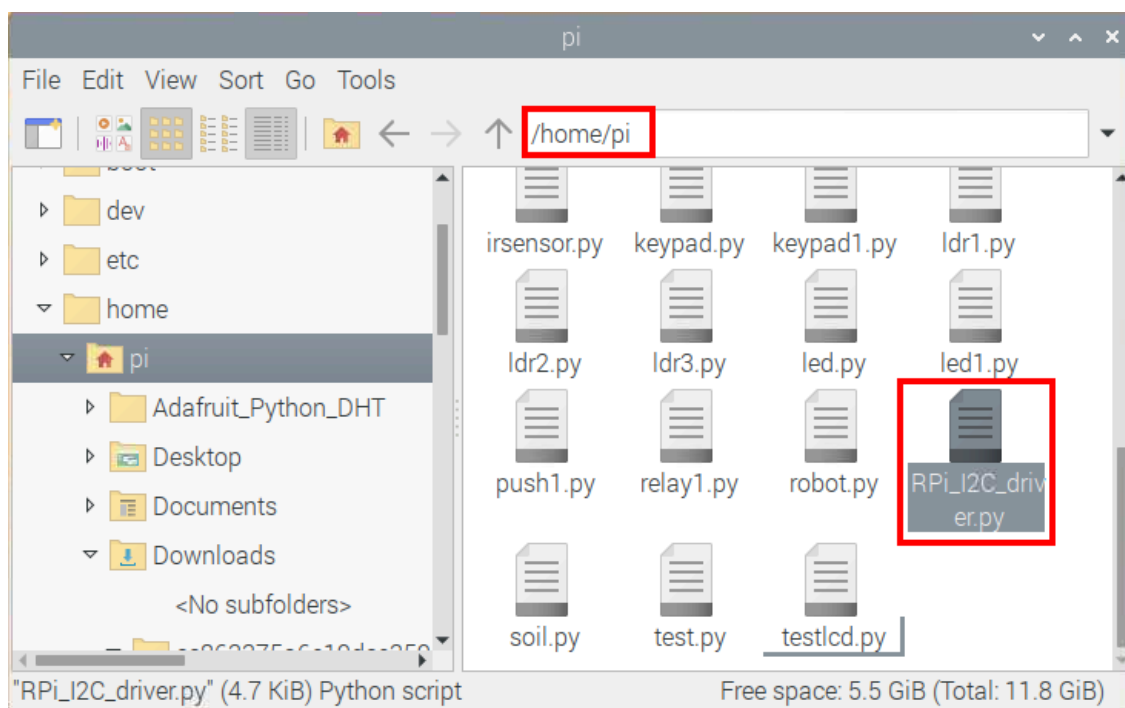
Click here to open applications menu /pi/Downloads/cc863375a6e19dce359d-36b82e787450d127f5019a40e0a55b08bd43435a/RPi_I2C_driver.py @ 190 : 1
New Load Save Run Debug Over Into Out Stop Zoom Quit Switch to regular mode

RPi_I2C_driver.py
45     return self.bus.read_byte_data(self.addr, cmd)
46
47 # Read a block of data
48 def read_block_data(self, cmd):
49     return self.bus.read_block_data(self.addr, cmd)
50
51
52
53 # LCD Address
54 ADDRESS = 0x27
55
56 # commands
57 LCD_CLEARDISPLAY = 0x01

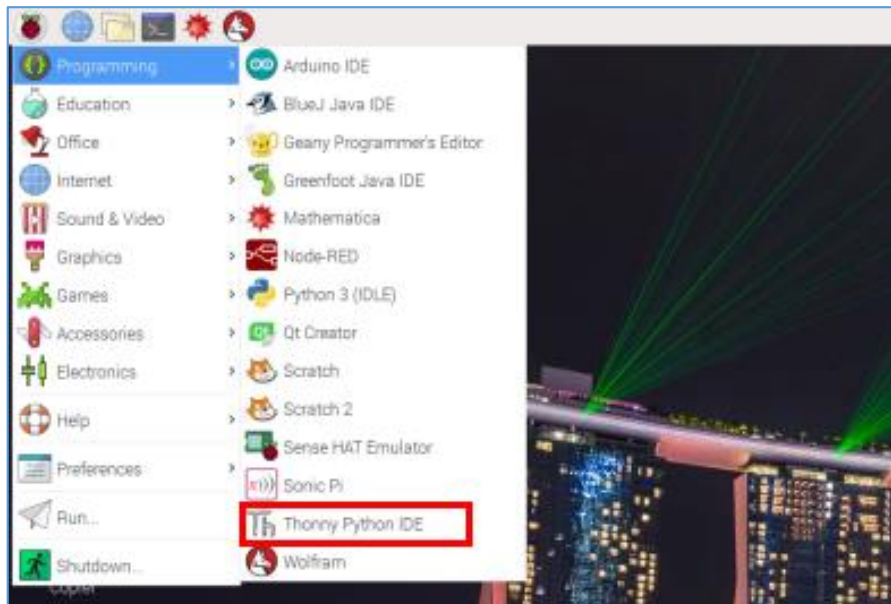
Shell
Python 3.7.3 (/usr/bin/python3)
>>>

```

15. คัดลอกไฟล์ RPi_I2C_driver.py มาไว้ที่ /home/pi/



16. เปิดโปรแกรม Thonny Python IDE โดยไปที่ Pi > Programming > Thonny Python IDE



17. เขียนโปรแกรมดังนี้ แล้วบันทึกไฟล์ไว้ใน /home/pi/

 A screenshot of the Thonny Python IDE interface. The main window shows a Python script named 'testlcd.py' with the following code:


```
1 import RPi_I2C_driver
2
3 mylcd = RPi_I2C_driver.lcd()
4
5 mylcd.lcd_display_string("Hello", 1)
6 mylcd.lcd_display_string("Raspberrry Pi", 2)
```

 Below the code editor is a 'Shell' window showing the command prompt:


```
Python 3.7.3 (/usr/bin/python3)
>>> %Run testlcd.py
```

 The top toolbar includes icons for New, Load, Save, Run, Debug, Over, Into, Out, and Stop.

ผลลัพธ์ที่ได้

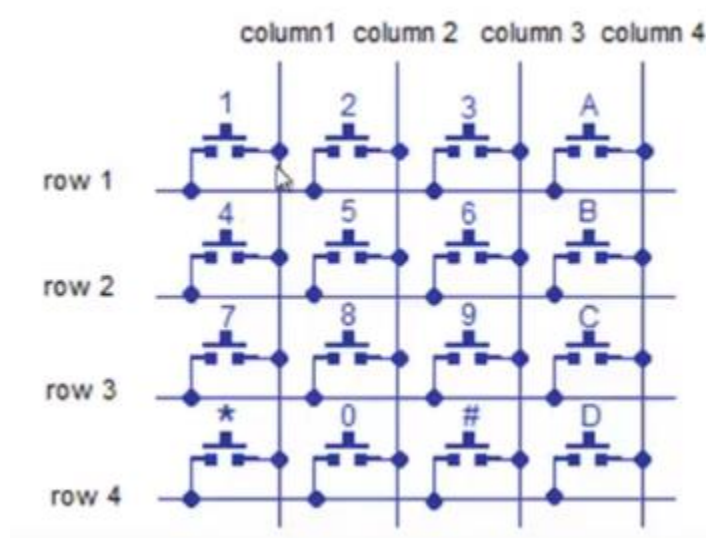
หน้าจอ LCD จะปรากฏข้อความบนแถวแรกว่า Hello ส่วนแถวล่าง จะปรากฏข้อความว่า Raspberrry Pi

2. Key Pad 4X4

Keypad เป็นหนึ่งในอุปกรณ์อินพุตที่นิยมใช้ในงานด้านระบบสมองกลฝังตัวมากที่สุด เราสามารถพบเห็นการใช้งาน Keypad ได้จากตู้อัตโนมัติต่าง ๆ เช่น ตู้เติมเงินมือถือ ตู้ซื้อขนมปัง ตู้ ATM ซึ่งแต่ละตู้นำมาใช้เพื่อให้ผู้ใช้บริการกรอกข้อมูลที่เป็นตัวเลขเข้าไป เช่น หมายเลขโทรศัพท์ จำนวนเงิน หมายเลขสินค้า หมายเลขบัญชี รหัสผ่านบัตร ATM แม้เราจะพบเห็นการใช้งาน Keypad ได้ทั่วไป แต่การนำ Keypad มาต่อใช้งานกลับยากจนน่าประหลาดใจ

Keypad มักจะบอกขนาดเป็นจำนวนแถว x จำนวนคอลัมน์ เช่น มี 1 แถว 4 คอลัมน์ มักจะเขียน 1x4 หรือมี 4 แถว 3 คอลัมน์ มักจะเขียน 4x3 และ 4 แถว 4 คอลัมน์ จะเขียนเป็น 4x4

ลักษณะวงจรภายใน Key Pad



ฝึกทักษะการทดลองกิจกรรมที่ 15 เรียนรู้การใช้งาน Key Pad แสดงที่ละ 1 ตัวอักษร
อุปกรณ์ที่ใช้สำหรับการทดลอง

1. บอร์ด Raspberry Pi



2. สาย Jumper



3. Key Pad 4X4

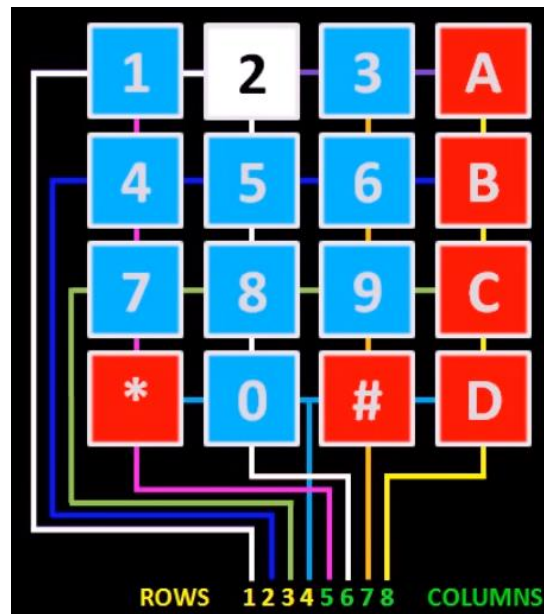


การต่อวงจร

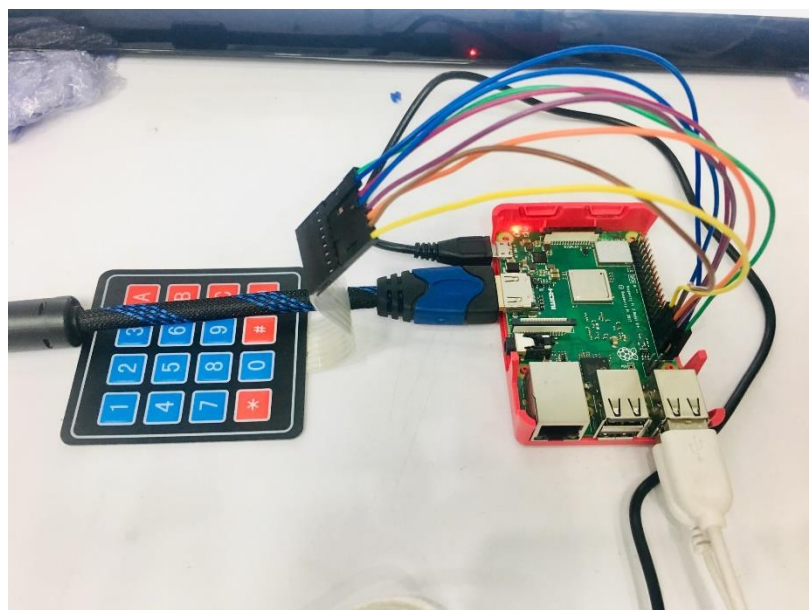
1. เสียบสาย Jumper แบบตัวผู้-ตัวเมีย เข้ากับช่องทั้ง 8 ช่องของ Key Pad



2. นำสาย Jumper อีกด้าน ไปเสียบเข้ากับ PIN ขา GPIO โดยเรียงจากเส้นซ้ายไปขวา ดังนี้

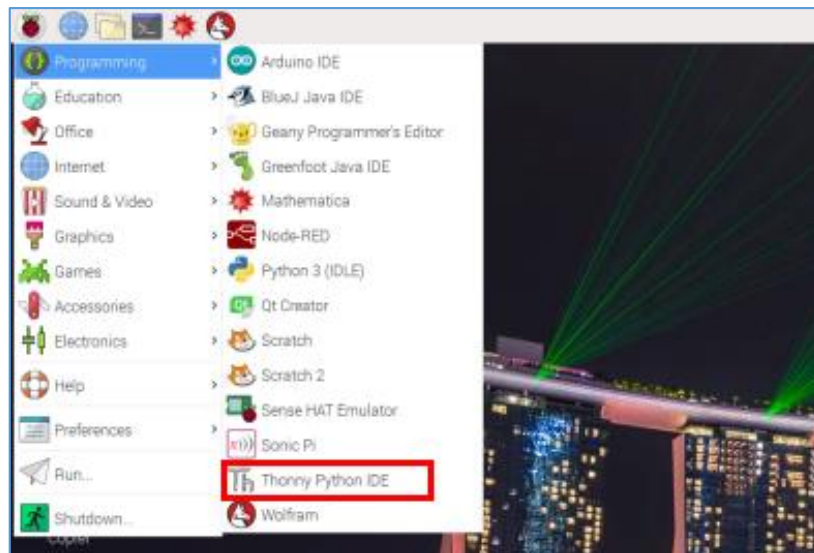


- เส้นที่ 1 เสียบเข้า PIN31 (GPIO6)
- เส้นที่ 2 เสียบเข้า PIN33 (GPIO13)
- เส้นที่ 3 เสียบเข้า PIN35 (GPIO19)
- เส้นที่ 4 เสียบเข้า PIN37 (GPIO26)
- เส้นที่ 5 เสียบเข้า PIN32 (GPIO12)
- เส้นที่ 6 เสียบเข้า PIN36 (GPIO16)
- เส้นที่ 7 เสียบเข้า PIN38 (GPIO20)
- เส้นที่ 8 เสียบเข้า PIN40 (GPIO21)



การเขียนโปรแกรมด้วยภาษาไพทอนเพื่อทดสอบอุปกรณ์

1. เปิดโปรแกรม Thonny Python IDE โดยไปที่ Pi > Programming > Thonny Python IDE



2. เขียนโปรแกรม ดังนี้

```

keypad1.py ✕
1 import RPi.GPIO as GPIO
2 import time
3 GPIO.setwarnings(False)
4 GPIO.setmode(GPIO.BOARD)
5
6 pad = [[1,2,3,'A'],
7         [4,5,6,'B'],
8         [7,8,9,'C'],
9         ['*',0,'#','D']]
10
11 row = [31,33,35,37]
12 column = [32,36,38,40]
13
14 for i in range(4):
15     GPIO.setup(column[i],GPIO.OUT)
16     GPIO.setup(row[i],GPIO.IN,pull_up_down = GPIO.PUD_UP)
17     GPIO.output(column[i],1)
18
19 while True:
20     for i in range(4):
21         GPIO.output(column[i],0)
22         for j in range(4):
23             if GPIO.input(row[j]) == 0:
24                 print(pad[j][i])
25                 time.sleep(0.5)
26         GPIO.output(column[i],1)

```

3. บันทึกไฟล์และรันโปรแกรม โดยคลิกปุ่ม Run

ผลลัพธ์ที่ได้

เมื่อกดปุ่มใดบน Key Pad ก็จะแสดงข้อความบนปุ่มนั้นออกมาทางหน้าจอ

ฝึกทักษะการทดลองกิจกรรมที่ 16 เรียนรู้การใช้งาน Key Pad แสดงที่ละหลายตัวอักษร
อุปกรณ์ที่ใช้สำหรับการทดลอง

1. บอร์ด Raspberry Pi



2. สาย Jumper



3. Key Pad 4X4

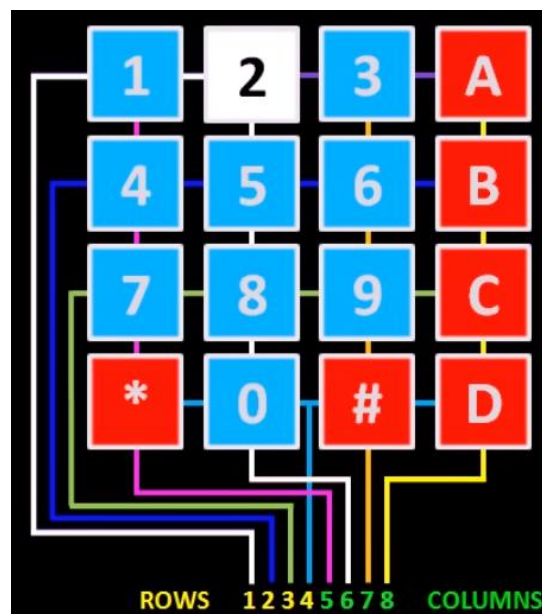


การต่อวงจร

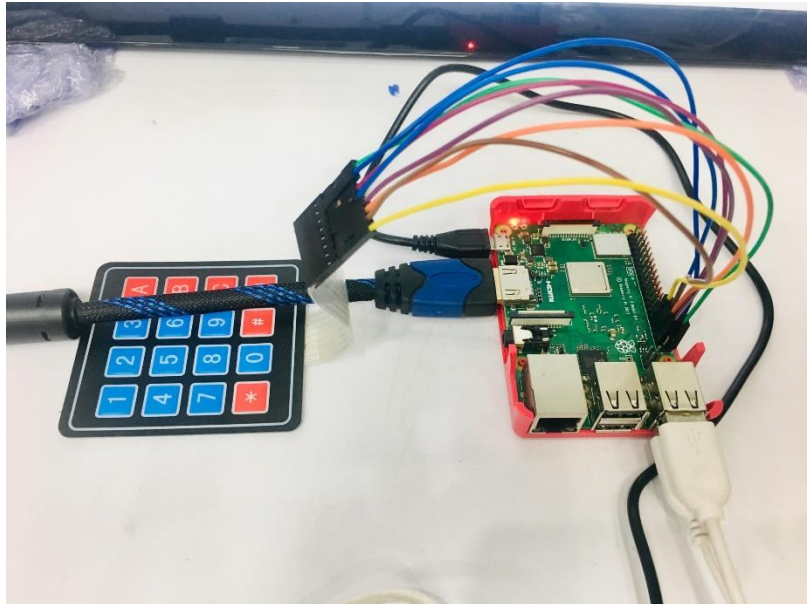
1. เสียบสาย Jumper แบบตัวผู้-ตัวเมีย เข้ากับช่องทั้ง 8 ช่องของ Key Pad



2. นำสาย Jumper อีกด้านไปเสียบเข้ากับ PIN ขา GPIO โดยเรียงจากเส้นซ้ายไปขวา ดังนี้

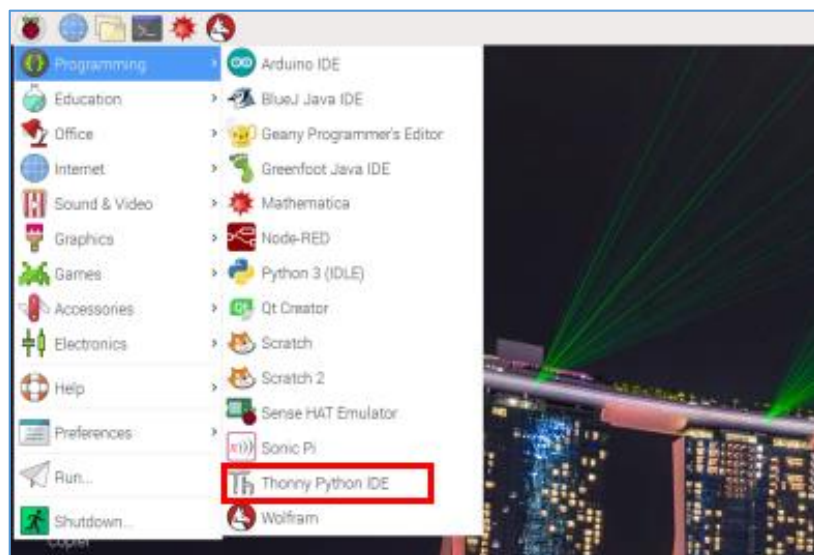


- เส้นที่ 1 เสียบเข้า PIN31 (GPIO6)
- เส้นที่ 2 เสียบเข้า PIN33 (GPIO13)
- เส้นที่ 3 เสียบเข้า PIN35 (GPIO19)
- เส้นที่ 4 เสียบเข้า PIN37 (GPIO26)
- เส้นที่ 5 เสียบเข้า PIN32 (GPIO12)
- เส้นที่ 6 เสียบเข้า PIN36 (GPIO16)
- เส้นที่ 7 เสียบเข้า PIN38 (GPIO20)
- เส้นที่ 8 เสียบเข้า PIN40 (GPIO21)



การเขียนโปรแกรมด้วยภาษาไพทอนเพื่อทดสอบอุปกรณ์

1. เปิดโปรแกรม Thonny Python IDE โดยไปที่ Pi > Programming > Thonny Python IDE



2. เขียนโปรแกรม ดังนี้


```

keypad.py ✕
1 import RPi.GPIO as GPIO
2 import time
3 GPIO.setwarnings(False)
4 GPIO.setmode(GPIO.BOARD)
5
6 pad = [[1,2,3,'A'],
7         [4,5,6,'B'],
8         [7,8,9,'C'],
9         ['*',0,'#','D']]
10
11 row = [31,33,35,37]
12 column = [32,36,38,40]
13
14 for i in range(4):
15     GPIO.setup(column[i],GPIO.OUT)
16     GPIO.setup(row[i],GPIO.IN,pull_up_down = GPIO.PUD_UP)
17     GPIO.output(column[i],1)
18
19 mytext = ''
20 while True:
21     for i in range(4):
22         GPIO.output(column[i],0)
23         for j in range(4):
24             if GPIO.input(row[j]) == 0:
25                 if pad[j][i] != '#':
26                     mytext += str(pad[j][i])
27                     time.sleep(0.5)
28             else:
29                 print(mytext)
30                 time.sleep(1)
31                 mytext = ''
32                 break
33         GPIO.output(column[i],1)

```

3. บันทึกไฟล์และรันโปรแกรม โดยคลิกปุ่ม Run

ผลลัพธ์ที่ได้

เมื่อกดปุ่มใดบน Key Pad ที่ไม่ใช่ปุ่ม # ก็จะไม่มีความใดๆ ปรากฏออกมาทางหน้าจอ แต่จะมีการเก็บทุกตัวเลขและตัวอักษรไว้ในตัวแปร mytext เมื่อมีการกดปุ่ม # ก็จะแสดงตัวอักษรและตัวเลขทั้งหมดที่กดไปก่อนหน้านี้ แสดงออกมาทางหน้าจอแสดงผล

3. LED

LED (Light-emitting diode) หรือเรียกว่า ไดโอดเปล่งแสง ภายในตัวหลอดไฟ LED เมื่อได้รับแรงดันไฟฟ้าจะปล่อยคลื่นแสงออกมาด้วยความถี่ที่ต่างกัน ทำให้เราสามารถมองเห็นเป็นสีต่างๆ กัน สำหรับหลอด LED ที่ใช้กับบอร์ด Raspberry Pi จะใช้ LED 3 สี ได้แก่ สีแดง สีเหลือง และสีเขียว

หลอดไฟ LED จะมี 2 ขา ได้แก่ ขาแอนโธ โดยขาหนึ่งจะเป็นขาที่ยาวกว่า จะต้องป้อนไฟบวก (+) เท่านั้น ส่วนอีกขาเรียกว่า แคโทด โดยขาหนึ่งจะเป็นขาสั้นกว่า จะต้องป้อนไฟลบ Ground (-) เท่านั้น

การต่อหลอดไฟ LED กับตัวต้านทาน

แหล่งจ่าย	ค่าความต้านทาน (โอห์ม)
3V	100 - 200
5V	150 - 250
9V	350 - 450
12V	500 - 1K

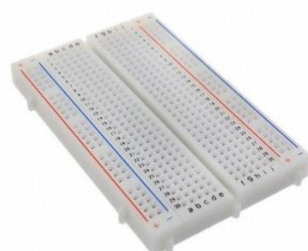
ฝึกทักษะการทดลองกิจกรรมที่ 17 เรียนรู้การใช้งาน LED

อุปกรณ์ที่ใช้สำหรับการทดลอง

1. บอร์ด Raspberry Pi



2. บอร์ดทดลอง (Breadboard)



3. ไดโอดเปล่งแสง (Diode LED)



4. ตัวต้านทาน (Resistor) 330 โอห์ม

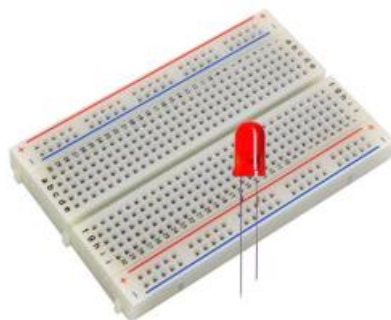


5. สาย Jumper

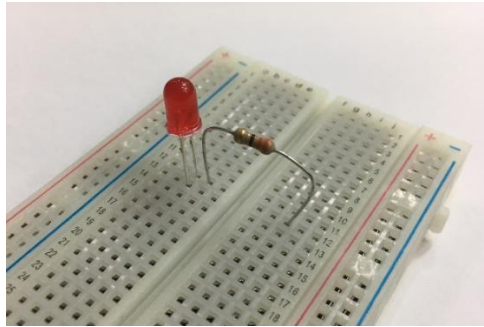


การต่อวงจร

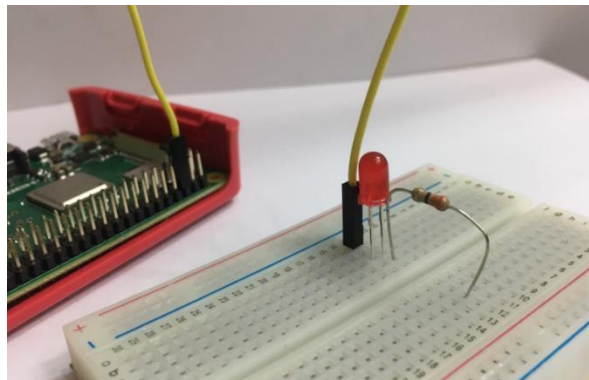
1. เสียบขา LED (ไดโอดเปล่งแสง) ก่อนอื่นจะต้องหาขาขั้วบวก (+) ของ LED โดยการหาว่าขา LED ข้างใดยาวกว่า ขานั้น คือ ขั้วบวก (+) ส่วนขา LED ที่สั้นกว่า คือ ขั้วลบ (-)



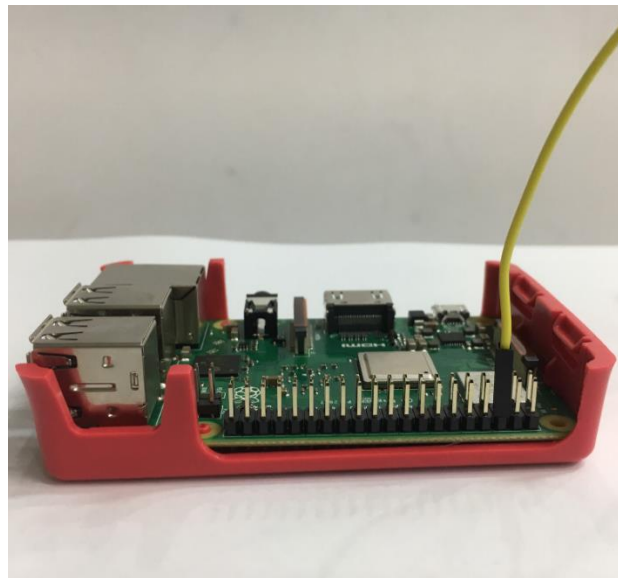
2. นำตัวต้านทาน (Resistor) 330 โอห์ม เสียบเข้ากับขาบวก (+) ของ LED



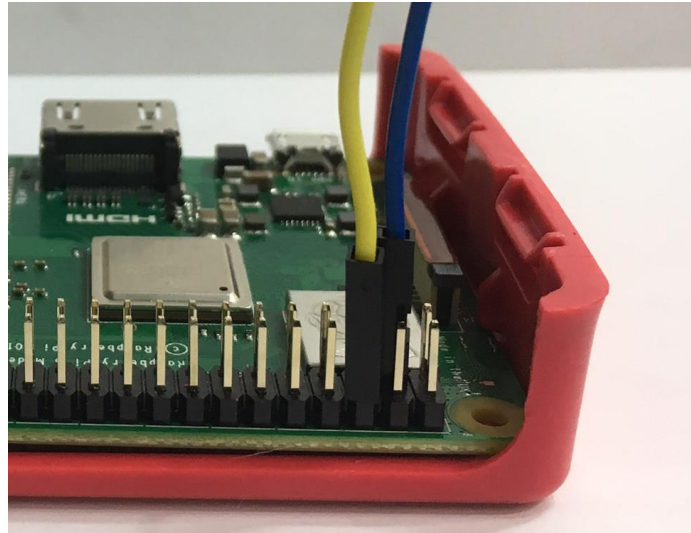
3. ต่อ Ground ให้กับวงจรไฟกะพริบ โดยใช้ Male Cable (หัวเสียบ) ต่อเข้ากับขาลบ (-) ของ LED



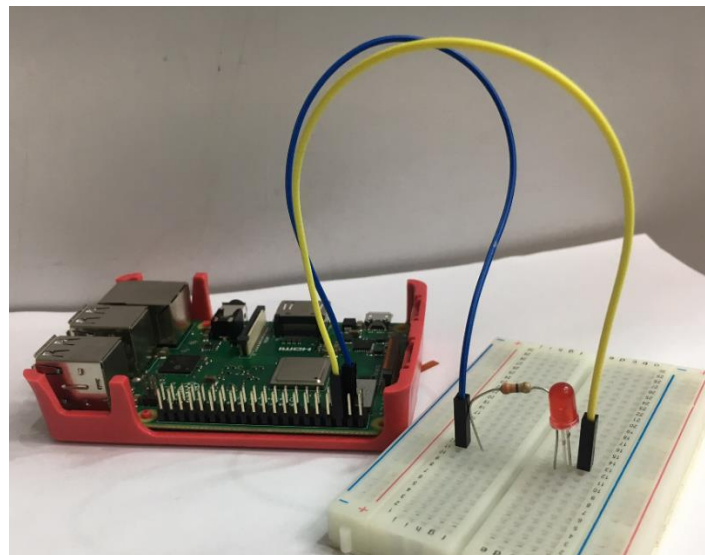
4. นำ Female Cable (หัวสวม) เสียบเข้ากับขา PIN6 (GROUND) ของขา GPIO

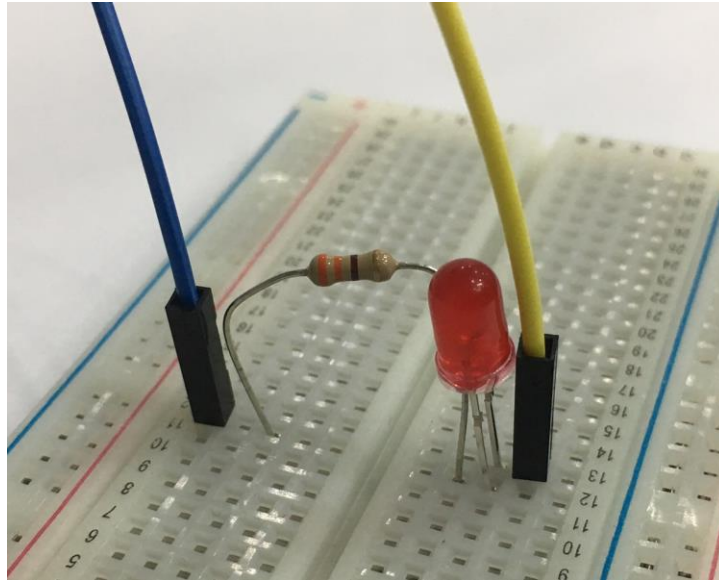


5. ป้อนสัญญาณ Output (สัญญาณดิจิทัล ค่า 0 หรือ 1) โดยใช้ Female Cable สวมเข้ากับขา PIN3 (GPIO2)



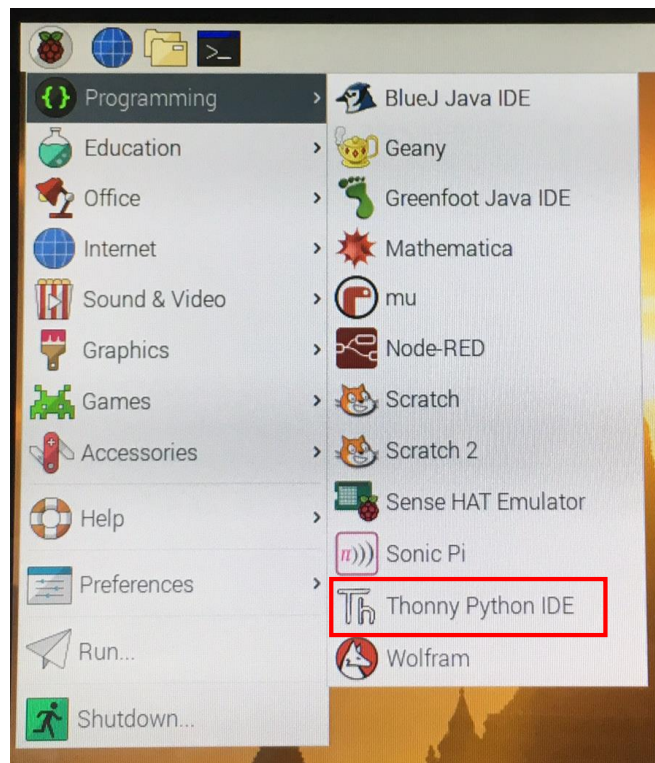
6. เสียบ Male Cable ที่เชื่อมต่อขา PIN3 (GPIO2) ต่อเข้ากับขาตัวต้านทาน 330 โอห์ม ถือว่าครบวงจรไฟกะพริบ



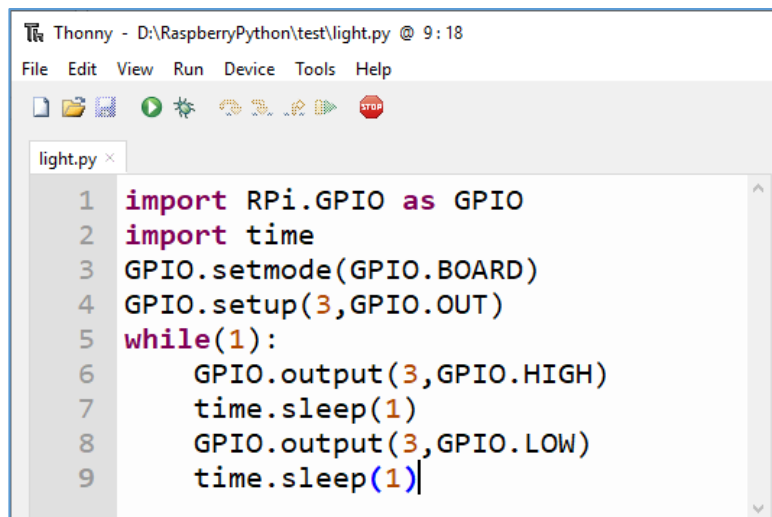


ขั้นตอนการเขียนโปรแกรมด้วยภาษา Python

1. เปิดโปรแกรม Thonny Python IDE โดยคลิกที่ Pi --> Programming --> Thonny Python IDE



2. เขียนโปรแกรม ดังนี้

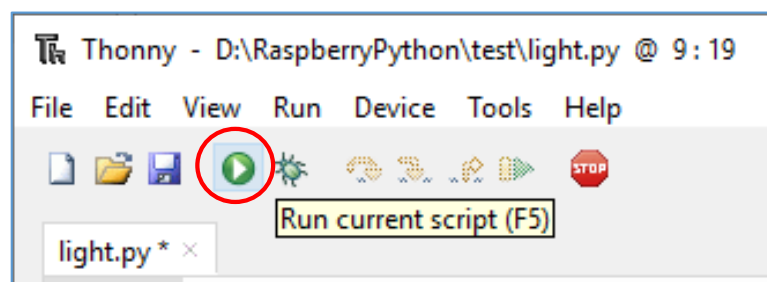


```

Thonny - D:\RaspberryPython\test\light.py @ 9:18
File Edit View Run Device Tools Help

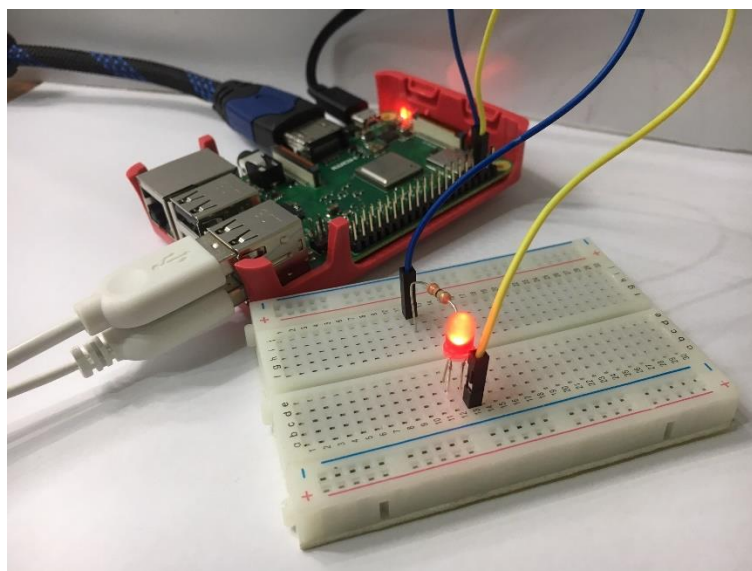
light.py x
1 import RPi.GPIO as GPIO
2 import time
3 GPIO.setmode(GPIO.BOARD)
4 GPIO.setup(3,GPIO.OUT)
5 while(1):
6     GPIO.output(3,GPIO.HIGH)
7     time.sleep(1)
8     GPIO.output(3,GPIO.LOW)
9     time.sleep(1)
  
```

3. บันทึกไฟล์และรันและรัน โปรแกรม โดยคลิกปุ่ม Run



ผลลัพธ์ที่ได้

ไดโอดเปล่งแสง LED จะติดเป็นเวลา 1 วินาที และดับเป็นเวลา 1 วินาที สลับกันไปเรื่อย ๆ จะเห็นได้ว่าอุปกรณ์ OUTPUT ในที่นี้คือหลอดไฟ LED ซึ่งจะส่งค่า HIGH (ไฟติด) และ LOW (ไฟดับ)



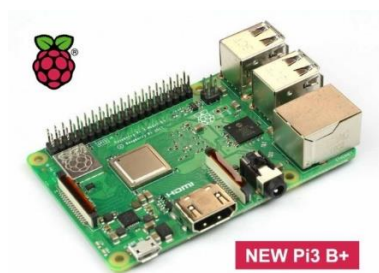
4. Push Button

Push Button หรือที่เรียกกันว่าสวิตช์ปุ่มกด เป็นอุปกรณ์ทางไฟฟ้า ซึ่งทำหน้าที่ตัดและต่อวงจรทางไฟฟ้าและ ใช้ในการควบคุมการทำงานของมอเตอร์หรือการทำงานของเครื่องจักรต่างๆ เป็นเหมือนอุปกรณ์พื้นฐาน ใช้ได้กับอุตสาหกรรมทั่วไป มีทั้งแบบมีไฟและทึบแสง

ฝึกทักษะการทดลองกิจกรรมที่ 18 เรียนรู้การใช้งาน Push Button

อุปกรณ์ที่ใช้สำหรับการทดลอง

1. บอร์ด Raspberry Pi



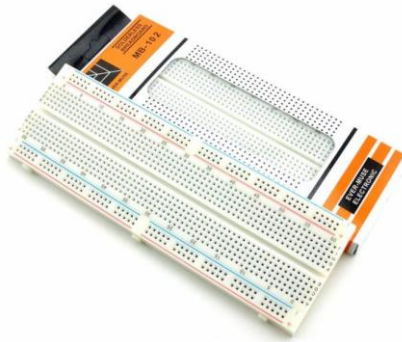
2. หลอด LED



3. ตัวต้านทาน 330 โอห์ม



4. Breadboard



5. สาย Jumper

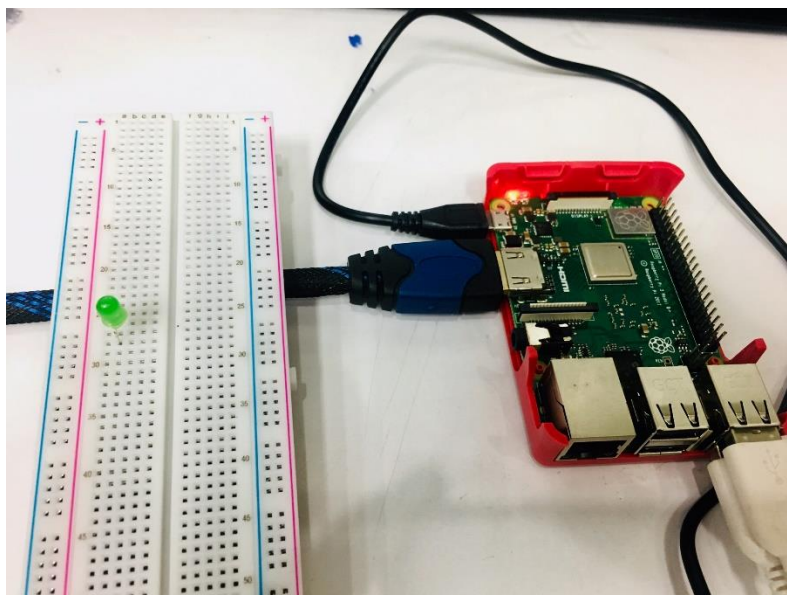


6. Push Button

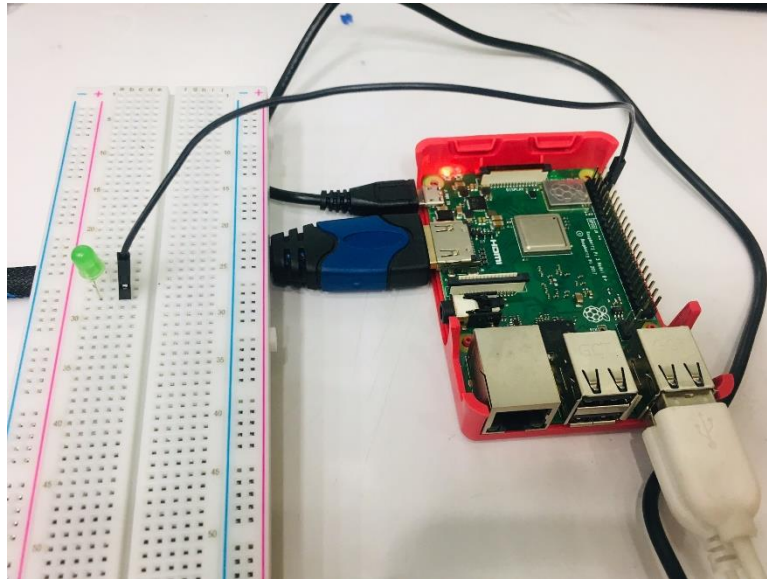


การต่อวงจร

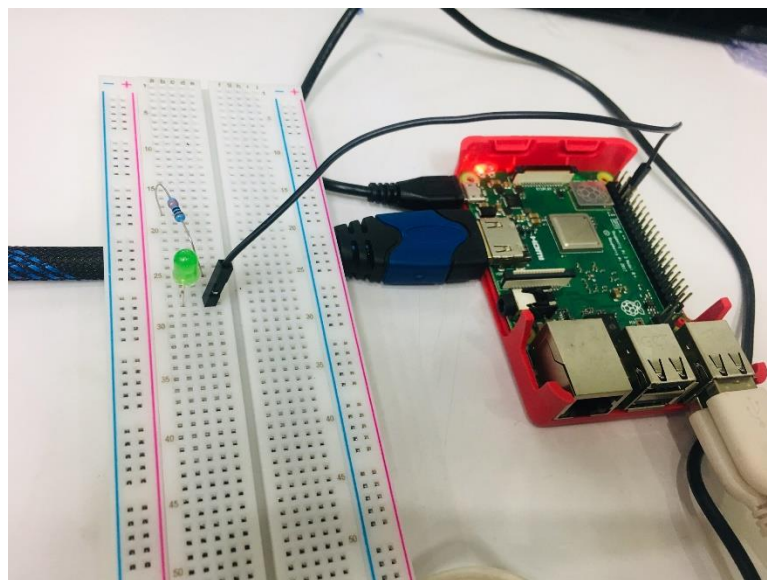
1. เชื่อม LED บน Breadboard



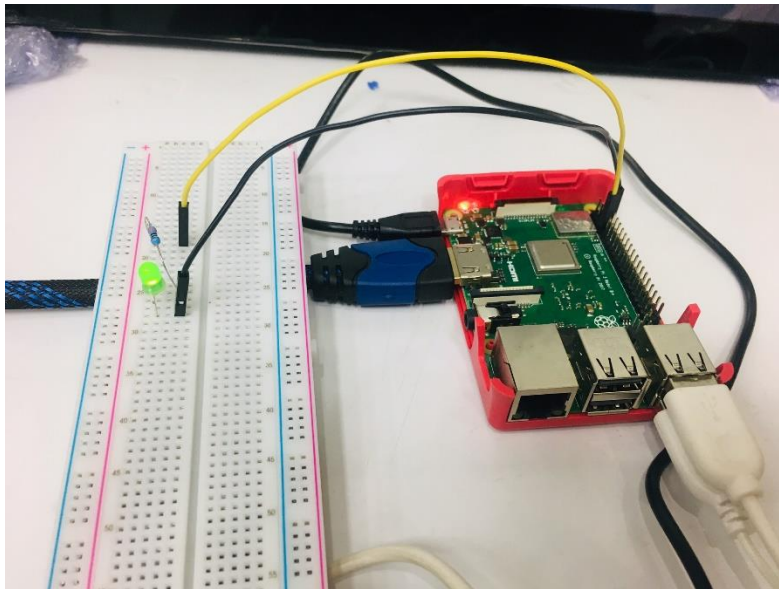
2. เสียบสาย Jumper ในแถวเดียวกับขา – (สั้น) ของหลอดไฟ LED ปลายสายเสียบเข้ากับช่อง PIN6 (GROUND) ของขา GPIO



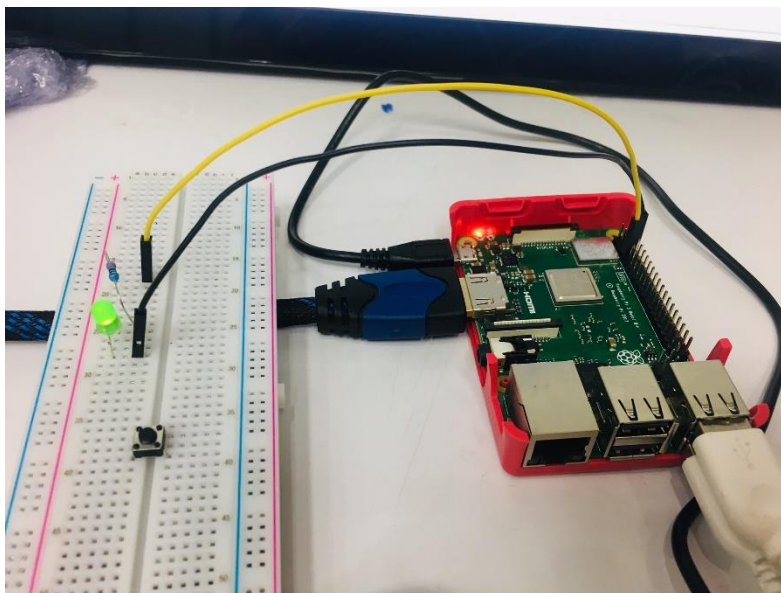
3. เสียบตัวต้านทาน 330 โอห์มเข้าไปในแถวเดียวกับขา + ของ LED



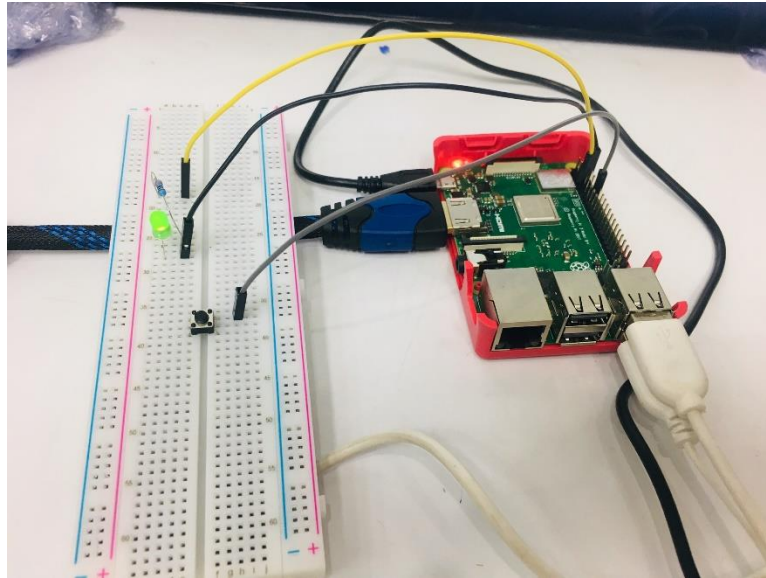
4. ใช้สาย Jumper เสียบในแถวเดียวกับตัวต้านทาน 330 โอห์มอีกด้านหนึ่ง ปลายสายเสียบเข้ากับช่อง PIN7 (GPIO4) ของขา GPIO



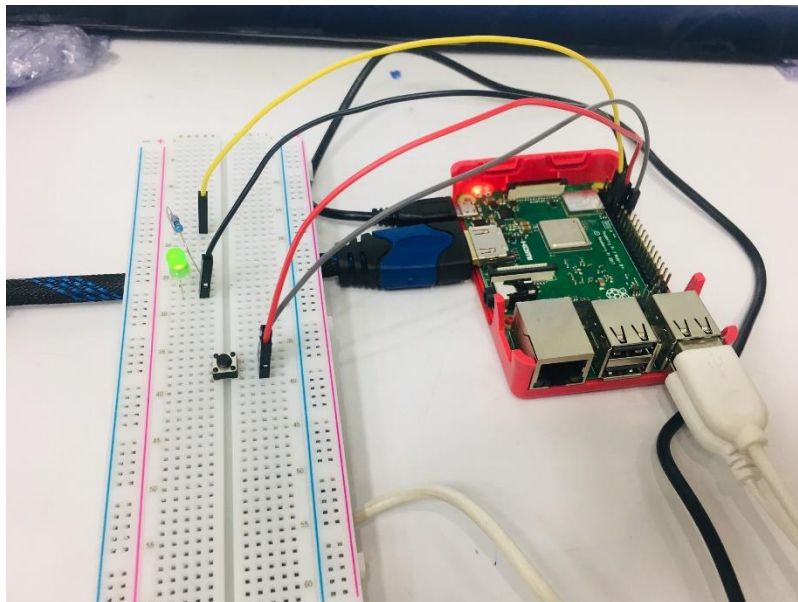
5. เสียบ Pushbutton บน Breadboard



6. เสียบสาย Jumper ที่ขาข้างหนึ่งของ Pushbutton ต่อเข้ากับขา PIN14 (GROUND) ของขา GPIO

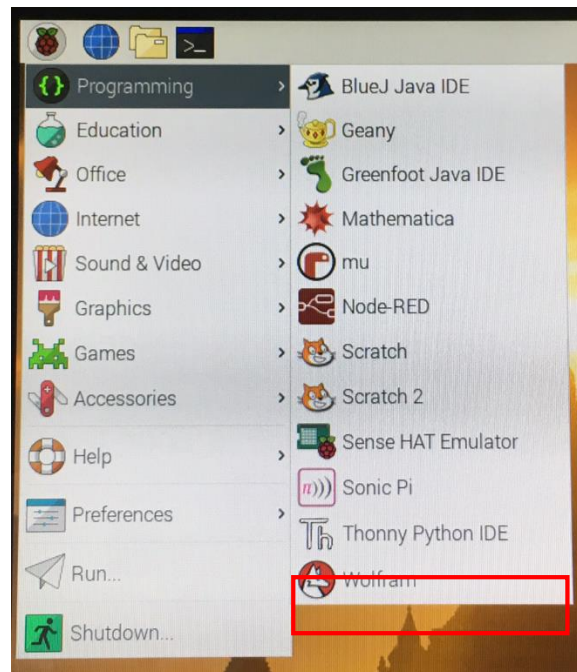


7. เสียบสาย Jumper ที่ขาอีกข้างหนึ่งของ Pushbutton ต่อเข้ากับขา PIN11 (GPIO17) ของขา GPIO



ขั้นตอนการเขียนโปรแกรมด้วยภาษา Python

1. เปิดโปรแกรม Thonny Python IDE โดยคลิกที่ Pi --> Programming --> Thonny Python IDE



2. เขียนโปรแกรม ดังนี้

```

1 import RPi.GPIO as GPIO
2 import time
3 GPIO.setwarnings(False)
4
5 GPIO.setmode(GPIO.BOARD)
6 GPIO.setup(7,GPIO.OUT)
7 GPIO.output(7,GPIO.LOW)
8 GPIO.setup(11,GPIO.IN,pull_up_down=GPIO.PUD_UP)
9
10 while True:
11     #Don't Push ==> LED ON
12     if GPIO.input(11):
13         GPIO.output(7,GPIO.LOW)
14
15     #Push ==> LED OFF
16     else:
17         GPIO.output(7,GPIO.HIGH)
18         time.sleep(0.1)
19 GPIO.cleanup()

```

3. บันทึกไฟล์และรันโปรแกรม โดยคลิกปุ่ม Run

ผลลัพธ์ที่ได้

เมื่อกดปุ่ม Pushbutton ไฟ LED จะติด (HIGH) แต่ถ้าไม่กดปุ่ม Pushbutton ไฟ LED จะดับ (LOW)

หน่วยการเรียนรู้ที่ ๖ พัฒนาระบบ Internet Of Things จากบอร์ด Raspberry Pi

กลุ่มสาระการเรียนรู้วิทยาศาสตร์

สาระที่ ๔ เทคโนโลยี

มาตรฐาน ว ๔.๒ เข้าใจและใช้แนวคิดเชิงคำนวณในการแก้ปัญหาที่พบในชีวิตจริงอย่างเป็นขั้นตอนและเป็นระบบ ใช้เทคโนโลยีสารสนเทศและการสื่อสารในการเรียนรู้ การทำงาน และการแก้ปัญหาได้อย่างมีประสิทธิภาพ รู้เท่าทัน และมีจริยธรรม

ตัวชี้วัด

- ว ๔.๒ ม.๑/๒ ออกแบบและเขียนโปรแกรมอย่างง่ายเพื่อแก้ปัญหาทางคณิตศาสตร์หรือวิทยาศาสตร์
- ว ๔.๒ ม.๒/๒ ออกแบบและเขียนโปรแกรมที่ใช้ตรรกะและฟังก์ชันในการแก้ปัญหา
- ว ๔.๒ ม.๓/๑ พัฒนาแอปพลิเคชันที่มีการบูรณาการกับวิชาอื่นอย่างสร้างสรรค์

สาระสำคัญ

- ๑. เรียนรู้ระบบ Internet of Things
- ๒. เรียนรู้การใช้งานโปรแกรมและแอปพลิเคชันในการสร้างระบบ IOT
- ๓. เรียนรู้การเขียนโปรแกรมเพื่อสร้างแอปพลิเคชันในการใช้งานระบบ IOT

สาระการเรียนรู้

ความรู้

- ความรู้เกี่ยวกับระบบ Internet of Things (IOT)
- ความรู้ในการเขียนโปรแกรมเพื่อพัฒนาระบบ IOT

ทักษะ/กระบวนการ

- ปฏิบัติตามตัวอย่างที่แสดงให้เห็นทีละขั้นตอนจากขั้นพื้นฐานไปสู่ขั้นที่ซับซ้อนขึ้น
- ฝึกปฏิบัติด้วยตนเองจนเกิดความชำนาญ หรือทำได้โดยอัตโนมัติ ซึ่งอาจเป็นงานชิ้นเดิมหรืองาน

ที่คิดค้นใหม่

- ใช้กระบวนการคณิตศาสตร์ด้านการคิดคำนวณและการแก้ปัญหา
- กระบวนการกลุ่ม มุ่งให้ผู้เรียนทำงานร่วมกันเป็นทีม

คุณลักษณะที่พึงประสงค์

- รักชาติ ศาสน์ กษัตริย์
- ซื่อสัตย์
- มีวินัย
- ใฝ่เรียนรู้
- อยู่อย่างพอเพียง
- มุ่งมั่นในการทำงาน
- รักความเป็นไทย
- มีจิตสาธารณะ

หน่วยการเรียนรู้ที่ 6

พัฒนาระบบ Internet Of Things จากบอร์ด Raspberry Pi

แนวคิดและที่มาของ Internet of Things

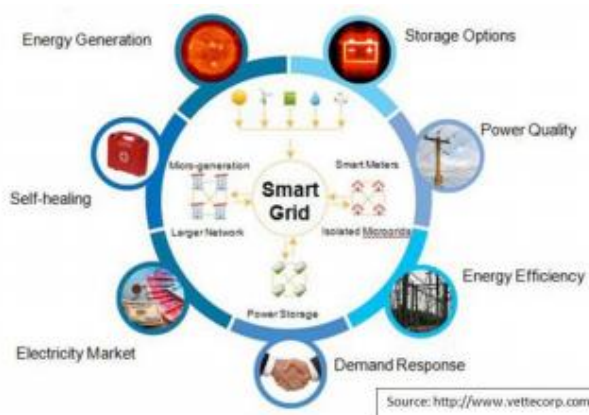
เทคโนโลยีเข้ามามีบทบาทในการใช้ชีวิตประจำวันของเรามากขึ้น ไม่ว่าจะเป็น โทรศัพท์มือถือ สมาร์ทโฟน แท็บเล็ต ยานพาหนะ หรือแม้กระทั่งเครื่องใช้ไฟฟ้าในบ้านของเราเอง ที่มีการพัฒนาให้ฉลาดและอำนวยความสะดวกต่อผู้ใช้งานมากขึ้น แนวคิด IOT ที่ย่อมาจาก Internet of Things จึงเป็นส่วนหนึ่งที่จะช่วยให้อุปกรณ์รอบตัวของเรานั้น สามารถทำงานเชื่อมต่อและควบคุมได้อย่างอัจฉริยะ เสมือนว่าอุปกรณ์สามารถเชื่อมต่อและพูดคุยกันเอง ทำให้การใช้ชีวิตประจำวันของเราสะดวกสบายและปลอดภัยมากขึ้น

แนวคิด Internet of Things เดิมมาจาก Kevin Ashton บิดาแห่ง Internet of Things ในปี 1999 ในขณะที่ทำงานวิจัยอยู่ที่มหาวิทยาลัย Massachusetts Institute of Technology หรือ MIT เขาได้ถูกเชิญให้ไปบรรยายเรื่องนี้ให้กับบริษัท Procter & Gamble (P&G) เขาได้นำเสนอโครงการที่ชื่อว่า Auto-ID Center ต่อยอดมาจากเทคโนโลยี RFID ที่ในขณะนั้นถือเป็นมาตรฐานโลกสำหรับการจับสัญญาณเซ็นเซอร์ต่างๆ (RFID Sensors) ว่าตัวเซ็นเซอร์เหล่านั้นสามารถทำให้มันพูดคุยเชื่อมต่อกันได้ผ่านระบบ Auto-ID ของเขา โดยการบรรยายให้กับ P&G ในครั้งนั้น Kevin ก็ได้ใช้คำว่า Internet of Things ในสไลด์การบรรยายของเขาเป็นครั้งแรก โดย Kevin นิยามเอาไว้ตอนนั้นว่าอุปกรณ์อิเล็กทรอนิกส์ใดๆก็ตามที่สามารถสื่อสารกันได้ก็คือเป็น “internet-like” หรือพูดง่ายก็คืออุปกรณ์อิเล็กทรอนิกส์ที่สื่อสารแบบเดียวกันกับระบบอินเทอร์เน็ตนั่นเอง โดยคำว่า “Things” ก็คือคำใช้แทนอุปกรณ์อิเล็กทรอนิกส์ต่างๆเหล่านั้น



Kevin Ashton

ต่อมาในยุคหลังปี 2000 มีอุปกรณ์อิเล็กทรอนิกส์ถูกผลิตออกจัดจำหน่ายเป็นจำนวนมากทั่วโลก จึงเริ่มมีการใช้คำว่า Smart ซึ่งในที่นี้คือ Smart Device, Smart Grid, Smart Home, Smart Network, Smart Intelligent Transportation ต่างๆเหล่านี้ ล้วนถูกฝัง RFID Sensors เสมือนกับการเติม ID และสมอง ทำให้มันสามารถเชื่อมต่อกับโลกอินเทอร์เน็ตได้ ซึ่งการเชื่อมต่อเหล่านั้นเองก็เลยมาเป็นแนวคิดที่ว่าอุปกรณ์เหล่านั้นก็ย่อมสามารถสื่อสารกันได้ด้วยเช่นกัน โดยอาศัยตัว Sensor ในการสื่อสารถึงกัน นั่นแปลว่านอกจาก Smart Device ต่างๆจะเชื่อมต่ออินเทอร์เน็ตได้แล้ว ยังสามารถเชื่อมต่อไปยังอุปกรณ์ตัวอื่นได้ด้วย



Internet of Things (IOT) คือ การที่อุปกรณ์อิเล็กทรอนิกส์ต่างๆ สามารถเชื่อมโยงหรือส่งข้อมูลถึงกันได้ด้วยอินเทอร์เน็ต โดยไม่ต้องป้อนข้อมูล การเชื่อมโยงนี้ง่ายจนทำให้เราสามารถสั่งการควบคุมการใช้งานอุปกรณ์อิเล็กทรอนิกส์ต่างๆ ผ่านทางเครือข่ายอินเทอร์เน็ตได้ ไปจนถึงการเชื่อมโยงการใช้งานอุปกรณ์อิเล็กทรอนิกส์ต่างๆ ผ่านทางเครือข่ายอินเทอร์เน็ตเข้ากับการใช้งานอื่นๆ จนเกิดเป็นบรรดา Smart ต่างๆ ได้แก่ Smart Device, Smart Grid, Smart Home, Smart Network, Smart Intelligent Transportationทั้งหลายที่

เราเคยได้ยินนั่นเอง ซึ่งแตกต่างจากในอดีตที่อุปกรณ์อิเล็กทรอนิกส์เป็นเพียงสื่อกลางในการส่งและแสดงข้อมูลเท่านั้น

Internet of Things นี้ได้แก่การเชื่อมโยงของอุปกรณ์อัจฉริยะทั้งหลายผ่านอินเทอร์เน็ตที่เราคุ้นเคยกัน เช่น แอปพลิเคชัน แว่นตาอัจฉริยะ กล้องวงจรปิดที่สามารถเชื่อมต่อข้อมูลการวิ่ง ทั้งความเร็ว ระยะทาง สถานที่ และสถิติได้

นอกจากนั้น Cloud Storage หรือ บริการรับฝากไฟล์และประมวลผลข้อมูลของคุณผ่านทางออนไลน์ หรือเราเรียกอีกอย่างว่า แคล้งเก็บข้อมูลบนก้อนเมฆ เป็นอีกสิ่งหนึ่งที่เรารู้จักกันเป็นอย่างดีแต่ไม่รู้ว่าเป็นหนึ่งในรูปแบบของ Internet of Things สมัยนี้ผู้ใช้นิยมเก็บข้อมูลไว้ในก้อนเมฆมากขึ้น เนื่องจากมีข้อดีหลายประการ คือ ไม่ต้องกลัวข้อมูลสูญหายหรือถูกโจรกรรม ทั้งยังสามารถกำหนดให้เป็นแบบส่วนตัวหรือสาธารณะก็ได้ เข้าถึงข้อมูลได้ทุกที่ทุกเวลาด้วยอุปกรณ์อิเล็กทรอนิกส์ใดๆผ่านเครือข่ายอินเทอร์เน็ต แล้วยังมีพื้นที่ใช้สอยมาก มีให้เลือกหลากหลาย ช่วยเราประหยัดค่าใช้จ่ายได้อีกด้วย เนื่องจากเราไม่ต้องเสียเงินซื้ออุปกรณ์จัดเก็บข้อมูล เช่น ฮาร์ดไดรฟ์ หรือ Flash drive ต่างๆ

การแบ่งกลุ่ม Internet of Things

ปัจจุบันมีการแบ่งกลุ่ม Internet of Things ออกตามตลาดการใช้งานเป็น 2 กลุ่มได้แก่

1. Industrial IOT คือ แบ่งจาก local network ที่มีหลายเทคโนโลยีที่แตกต่างกันในโครงข่าย Sensor nodes โดยตัวอุปกรณ์ IOT Device ในกลุ่มนี้จะเชื่อมต่อแบบ IP network เพื่อเข้าสู่อินเทอร์เน็ต
2. Commercial IOT คือ แบ่งจาก local communication ที่เป็น Bluetooth หรือ Ethernet (wired or wireless) โดยตัวอุปกรณ์ IOT Device ในกลุ่มนี้จะสื่อสารภายในกลุ่ม Sensor nodes เดียวกันเท่านั้นหรือเป็นแบบ local devices เพียงอย่างเดียวอาจไม่ได้เชื่อมสู่อินเทอร์เน็ต

การใช้ประโยชน์จาก Internet of Things เพื่อทำให้หลายๆ อย่างง่ายขึ้น

การที่เทคโนโลยีเป็นที่แพร่หลายนั้นไม่ได้อยู่ที่ปัจจัยด้านราคาอย่างเดียว แต่เทคโนโลยีนั่นต้องส่งมอบประโยชน์ต่อชีวิตของพวกเราด้วย ซึ่ง Internet of Things ในปัจจุบัน ก่อให้เกิดประโยชน์ในด้านต่างๆ

1. รับส่งข้อมูลรูปแบบดิจิทัล ปัจจุบัน ข้อมูลดิจิทัลมีความจำเป็นมาก เพราะสามารถนำไปใช้กับเทคโนโลยีอื่นๆ ได้ทันที ซึ่ง IOT มีคุณสมบัติด้านการเก็บข้อมูลทางภาพถ่ายให้อยู่ในรูปแบบดิจิทัลได้อย่างง่ายดายและรวดเร็ว จึงนับเป็นประโยชน์อย่างมากในยุค Digital Transformation

2. แม่นยำ ใช้ได้ตลอดเวลา และส่งข้อมูลแบบ Real-Time ข้อมูลจาก IOT ไม่เพียงแต่เป็นดิจิทัลเท่านั้น แต่ยังสามารถแลกเปลี่ยนได้อย่างรวดเร็วระดับ Real-Time มีความแม่นยำ และสามารถใช้งานได้ตลอดเวลา ช่วยให้มีข้อมูลในการตัดสินใจได้ทันทุกที่

3. ลดภาระงานของบุคลากร ในอดีตการเก็บข้อมูลอาจต้องใช้คนเดินทางเข้าไปสอดส่องที่เครื่องมือเพื่อหาความผิดปกติ แต่ปัจจุบัน IOT ไม่เพียงแต่สอดส่องให้เราผ่าน Dashboard เท่านั้น แต่ยังสามารถเรียนรู้การหาความผิดปกติด้วยเทคโนโลยีอื่นๆ เช่น Artificial Intelligence ได้

4. ทำงานตรวจสอบในจุดที่คนเข้าไม่ถึง เราสามารถออกแบบ Smart Device ให้มีขนาดเล็กและทนทานเพื่อติดตั้งตามจุดที่คนเข้าถึงยากหรือในจุดที่มีอันตรายระหว่างดำเนินการได้ เช่น ภายในท่อส่งน้ำมัน หรือบ่อน้ำบาดาลเสีย ช่วยลดความเสี่ยงต่อชีวิตและทรัพย์สินจากการต้องเข้าพื้นที่อันตรายเป็นประจำได้

ต้องถือว่า Internet of Things เป็นอีกหนึ่งเทคโนโลยีที่สำคัญมากในยุค Digital Transformation โดยปัจจุบันเทคโนโลยีนี้ก็เริ่มเข้าไปเกี่ยวข้องกับแวดวงต่างๆ ทั้งภาครัฐกิจและสังคมบ้างแล้ว จะเป็นอย่างไรบ้างและเติบโตไปแค่ไหนแล้ว รอติดตามได้ในบทความต่อไปของเราได้

ตัวอย่างสำคัญของการใช้ IOT

เทคโนโลยีของ Internet of Things นั้นได้สร้างพรมแดนใหม่ของการใช้ข้อมูลและการสร้างประสบการณ์ใหม่ๆ เลยกลายเป็นเรื่องที่น่าตื่นเต้นสำหรับหลายๆ คนมากมายที่กำลังพยายามลองหาวิธีการใหม่ๆ ในการเข้าถึงกลุ่มเป้าหมายที่ต่างไปจากเดิม

1. Smart home หรือบ้านอัจฉริยะ คือการใช้เทคโนโลยีมาควบคุมอุปกรณ์ต่างๆ ภายในบ้าน เพื่ออำนวยความสะดวกแก่ผู้อยู่อาศัย, มีระบบการจัดการพลังงาน ระบบรักษาความปลอดภัยอัตโนมัติทั้งภายในและรอบตัวบ้าน ส่วนใหญ่จะควบคุมด้วยระบบคอมพิวเตอร์ เช่น ระบบควบคุมไฟฟ้าแสงสว่าง เช่น เปิด/ปิด หรือปรับระดับความสว่าง ระบบควบคุมอุปกรณ์ไฟฟ้าภายในบ้าน เช่น สั่งงานเครื่องปรับอากาศ หรือการเปิดปิดม่าน เป็นต้น



2. Wearable เป็นอุปกรณ์สวมใส่ เป็นได้ทั้งแฟชั่นและไอทีที่ช่วยเพื่ออำนวยความสะดวกให้กับผู้ใช้ ที่ให้คุณมากกว่าแค่เครื่องประดับร่างกาย เช่น การ Sync ข้อมูลระหว่างนาฬิกา กับ smartphone วัตถุประสงค์การก้าวเดิน การแจ้งเตือน sms โทรเข้า รวมถึงสั่งเตือนในกรณีนาฬิกาอยู่ห่างจาก smartphone เชื่อมต่อผ่านทาง Bluetooth, internet เป็นต้น



3. Smart City “เมืองอัจฉริยะ” หรือสมาร์ทซิตี (smart city) เป็นแนวคิดที่หมายถึงระบบที่เชื่อมโยงถึงกันซึ่งเป็นส่วนหนึ่งของโครงสร้างพื้นฐานของเมือง แนวคิดเบื้องหลังของเมืองอัจฉริยะคือการที่สภาพแวดล้อมสามารถรับรู้และปรับเปลี่ยนตัวเองเพื่อส่งมอบบริการที่มีประสิทธิภาพสูงสุดให้กับผู้อยู่อาศัย ระบบไฟจราจรที่เชื่อมโยงกัน ทำให้รู้ว่าควรหยุดตอนไหนและคนเดินถนนรู้ว่าควรข้ามถนนเมื่อไร ในที่สุดระบบสัญญาณจราจรก็พัฒนาเซ็นเซอร์ (sensor) เพื่อตรวจจับการเคลื่อนไหวของรถยนต์ แทนที่จะใช้เพียงการจับเวลา

ระบบอย่างการสอดแนมด้วยกล้องวงจรปิด (CCTV) ระบบวิเคราะห์ใบหน้า ระบบอ่านป้ายทะเบียนรถยนต์ ระบบเหล่านี้สร้างข้อมูลขึ้นมาแม้ว่าเราจะได้กำลังพกพาอุปกรณ์อิเล็กทรอนิกส์ใดๆ แต่สัญญาณอยู่บนท้องถนนเราก็ป้อนข้อมูลบางอย่างให้กับระบบแล้ว



4. Smart farming การทำไร่ทำสวนเป็นการปฏิบัติงานในพื้นที่ที่ค่อนข้างห่างไกล ฉะนั้นการนำ Internet of Things มาประยุกต์ใช้เพื่อทำการมอนิเตอร์จึงเป็นอะไรที่สามารถปฏิวัติวงการการทำเกษตรได้เลยทีเดียว



การพัฒนาระบบ Internet Of Things จากบอร์ด Raspberry Pi

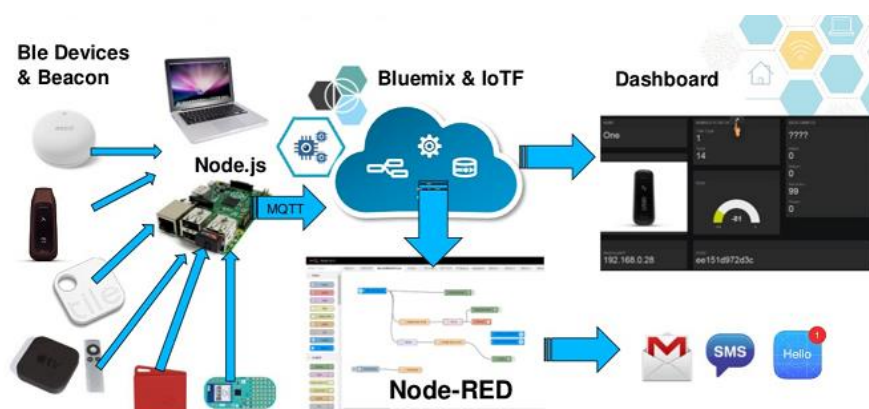
6.1 การใช้งาน Node-RED บน Raspberry Pi เพื่อใช้งาน IOT ผ่าน Web Browser

Node-RED เป็นเครื่องมือสำหรับนักพัฒนาโปรแกรมในการเชื่อมต่ออุปกรณ์ฮาร์ดแวร์เข้ากับ APIs (Application Programming Interface) ซึ่งเป็นการพัฒนาโปรแกรมแบบ Flow-Based Programming ที่มีหน้า UI สำหรับนักพัฒนาให้ใช้งานผ่าน Web Browser ทำให้การเชื่อมต่อเส้นทางการไหลของข้อมูลนั้นเป็นเรื่องง่าย



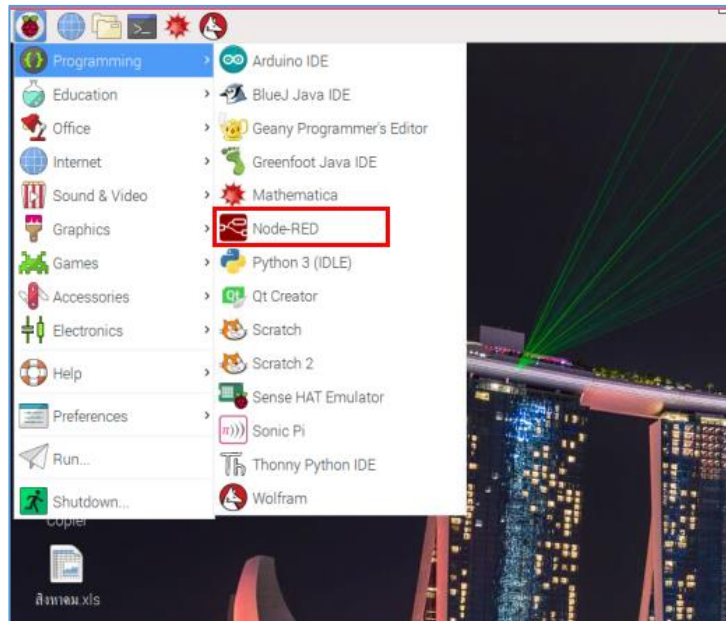
เนื่องจาก Node-RED เป็น Flow-Based Programming นั้นทำให้เราแทบจะไม่ต้องเขียน Code ในการพัฒนาโปรแกรมเลย แค่เพียงเลือก Node มาวางแล้วเชื่อมต่อก็สามารถควบคุม I/O ได้ โดย Node-RED จะมี Node ให้เลือกใช้งานอย่างหลากหลาย สามารถสร้างฟังก์ชัน JavaScript ได้โดยใช้ Text Editor ที่มีอยู่ใน Node-RED และยังสามารถบันทึก Function, Templates, Flows เพื่อไปใช้งานกับงานอื่นต่อไป

Node-RED นั้นทำงานบน Node.js ทำให้เหมาะสำหรับการใช้งานกับ Raspberry Pi เนื่องจากใช้ทรัพยากรน้อย ขนาดไฟล์ไม่ใหญ่และ Node.js ยังทำหน้าที่เป็นตัวกลางให้ Raspberry Pi สามารถติดต่อกับ Web Browser และอุปกรณ์อื่นๆ ได้

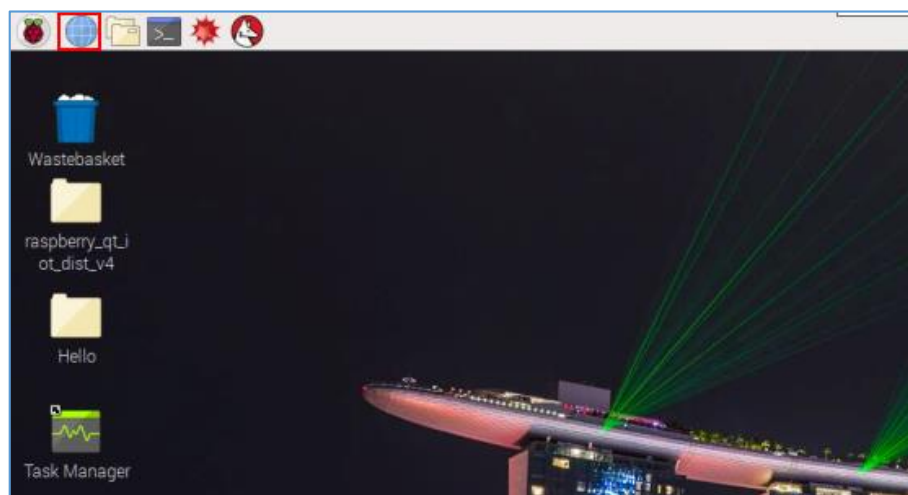


เริ่มต้นการใช้งาน Node-RED

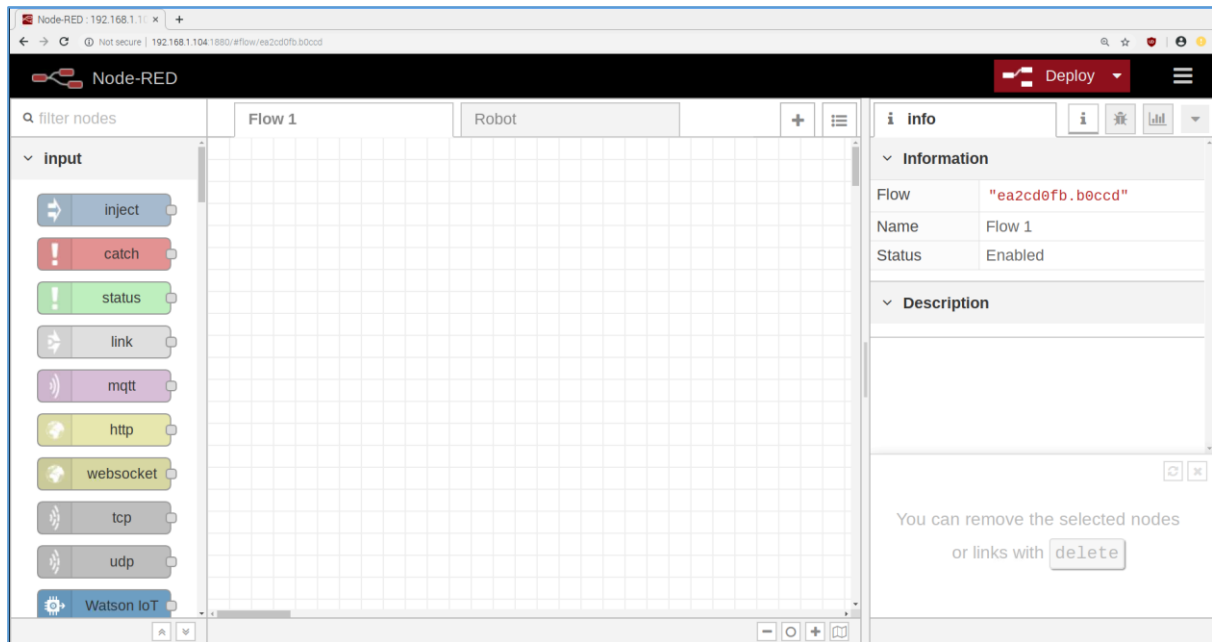
1. บอร์ด Raspberry Pi จับสัญญาณ Wi-Fi หรือเชื่อมต่อ LAN
2. เปิดโปรแกรม Node-RED โดยไปที่ Pi > Programming > Node-RED เพื่อเปิดใช้งาน Node-RED



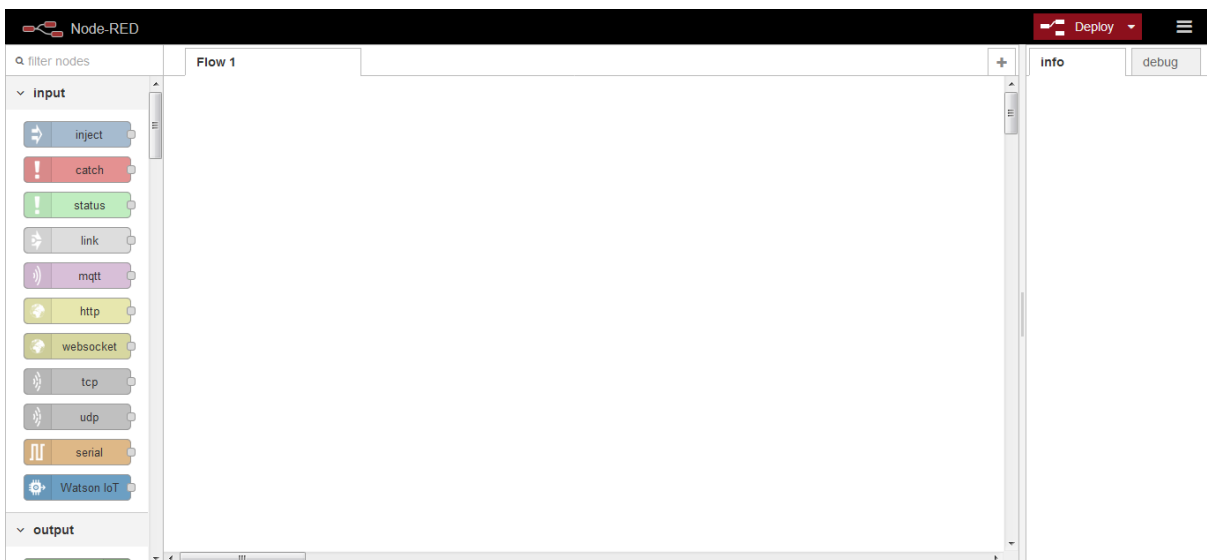
3. คลิก Web Browser เพื่อใช้งาน Node-RED



4. ใส่ IP Address ของ Raspberry ตามด้วย :1880 เช่น 192.168.1.104:1880 แล้วกดปุ่ม Enter

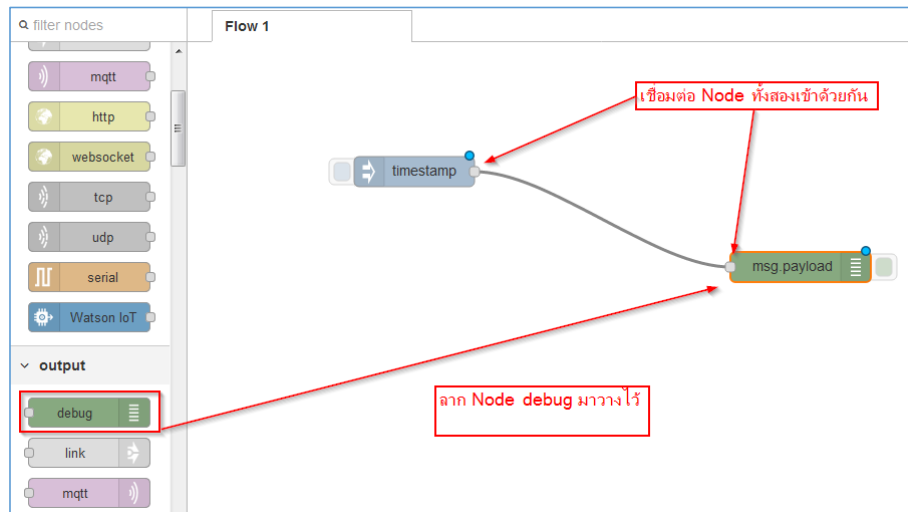


5. จะปรากฏหน้าจอ UI ของ Node-RED ดังรูป



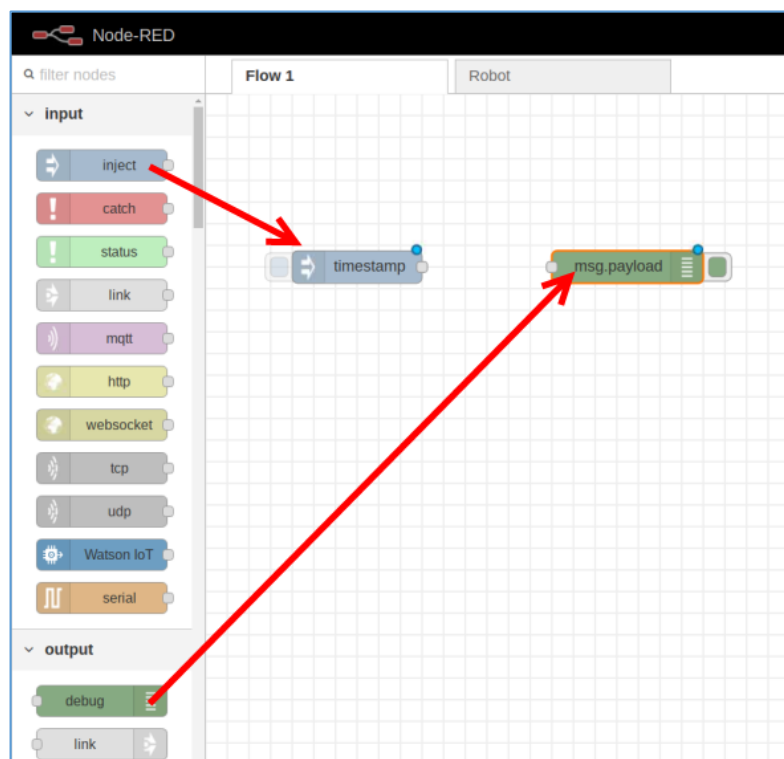
การเชื่อมต่อ Node

1. การใช้งาน Node-RED นั้น จะเป็นการใช้งานแบบ Flow-based programming โดยเราสามารถเลือก node ที่ต้องการจะนำมาวางบน Workspace ได้เลย
2. ให้เลือก inject และ debug แล้วลากมาวางบน Workspace

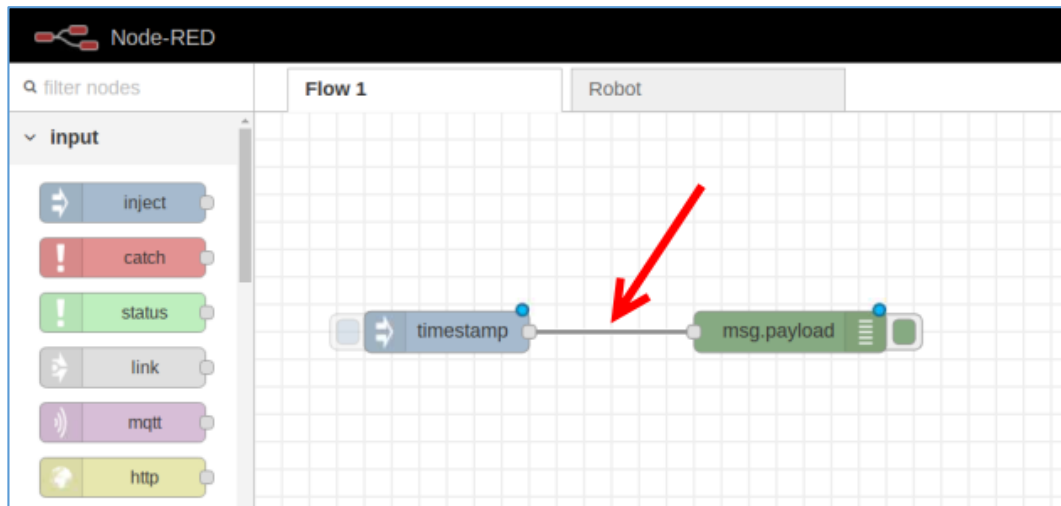


ฝึกทักษะการทดลองกิจกรรมที่ 1 เรียนรู้การใช้งาน Node-RED : สังพรั้นข้อความ

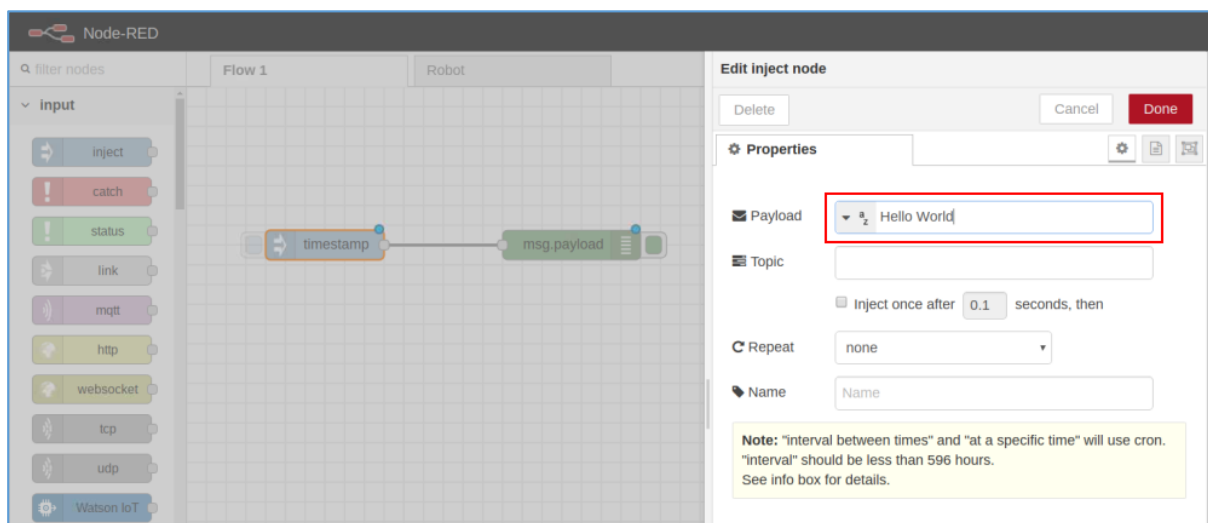
1. เริ่มต้นวาด Flow โดยลากโหนด inject (input) และ โหนด debug (output) มาวางบน Workspace



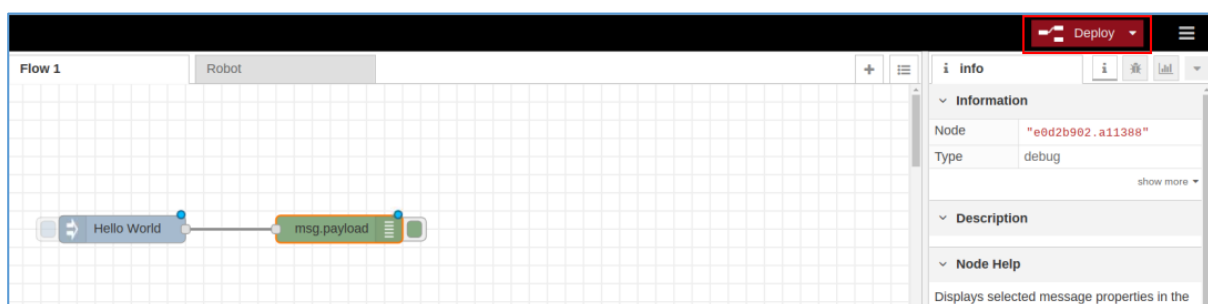
2. จากนั้นให้ลากเส้นเชื่อมโยงจากจุดกึ่งกลางของโหนดแรกไปยังจุดกึ่งกลางของโหนดที่สอง



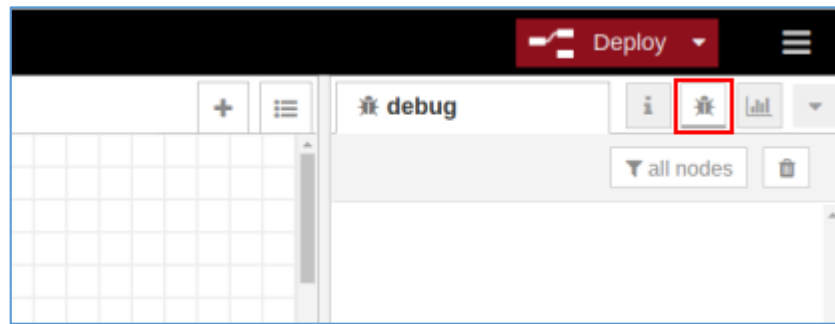
3. ดับเบิลคลิกที่โหนด timestamp ในช่อง Payload ให้เปลี่ยนจาก timestamp เป็น String แล้วพิมพ์ “Hello World” แล้วคลิกปุ่ม Done



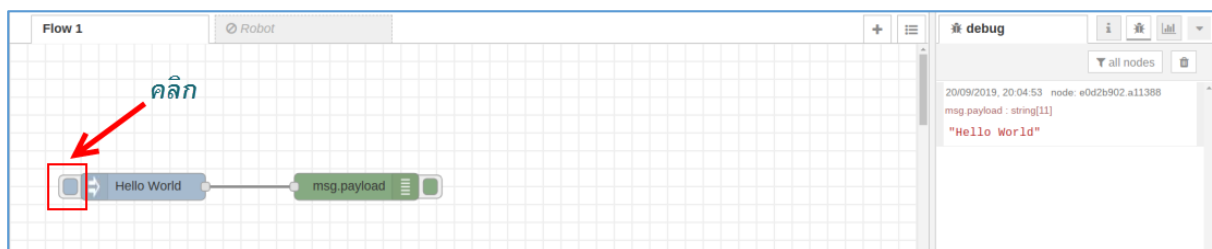
4. เมื่อเขียน Flow เรียบร้อยแล้ว ให้คลิกปุ่ม Deploy เพื่อสั่งรันโปรแกรม



5. เปิดมุมมอง Debug messages เพื่อแสดงผลลัพธ์ (Output)



6. ทดสอบโปรแกรมโดยคลิกที่ปุ่มด้านหน้าโหนด Hello World จะปรากฏข้อความที่หน้าต่าง Debug Messages ว่า “Hello World”



ฝึกทักษะการทดลองกิจกรรมที่ 2 เรียนรู้การใช้งาน Node-RED : เปิด-ปิดไฟด้วยสวิตช์ Push Button
อุปกรณ์ที่ใช้สำหรับการทดลอง

1. บอร์ด Raspberry Pi



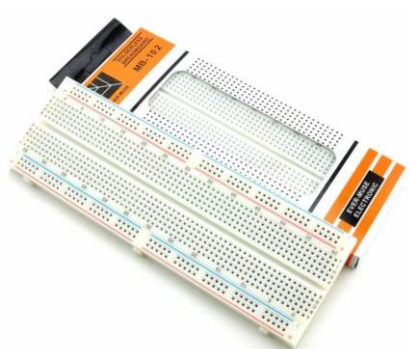
2. หลอด LED



3. ตัวต้านทาน 330 โอห์ม



4. Breadboard



5. สาย Jumper

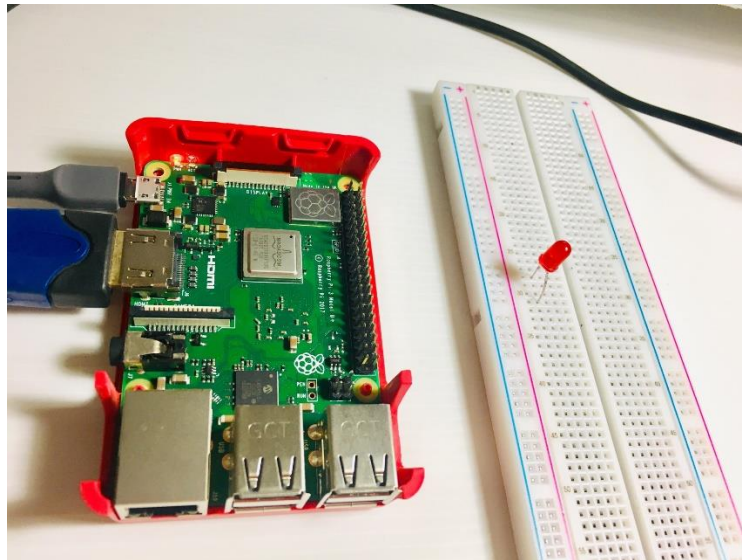


6. Push Button

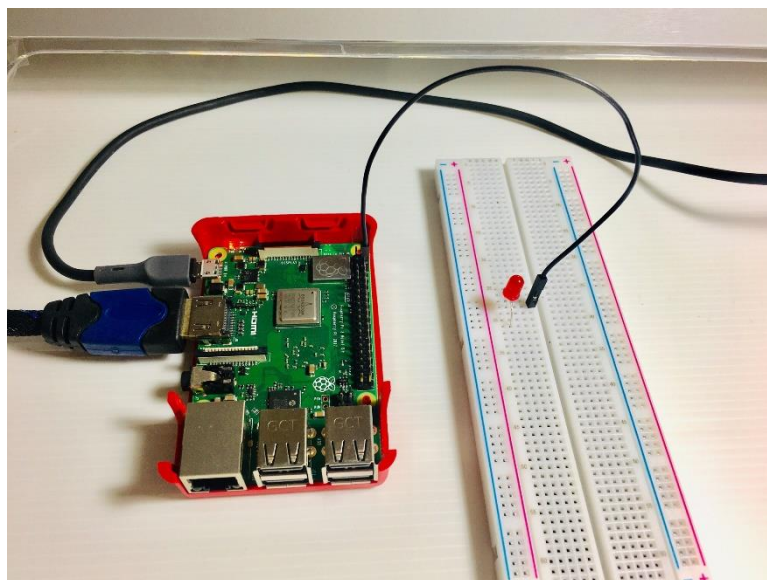


การต่อวงจร

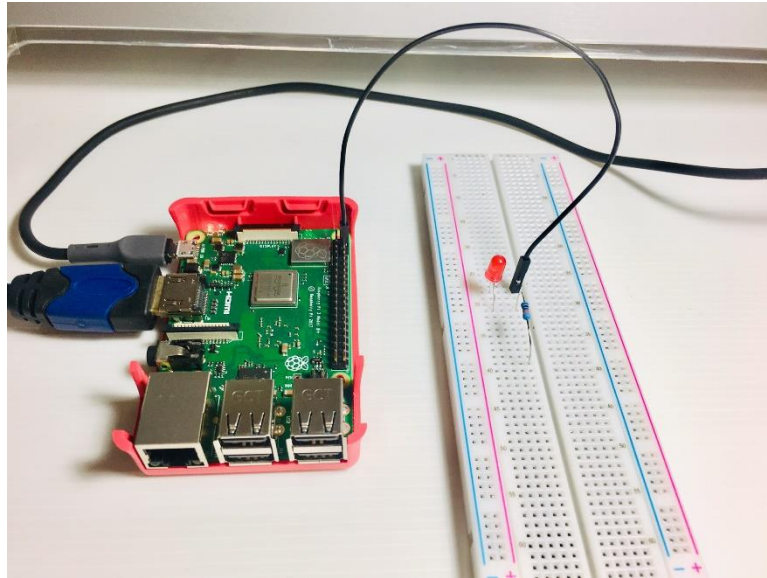
1. เสียบ LED บน Breadboard



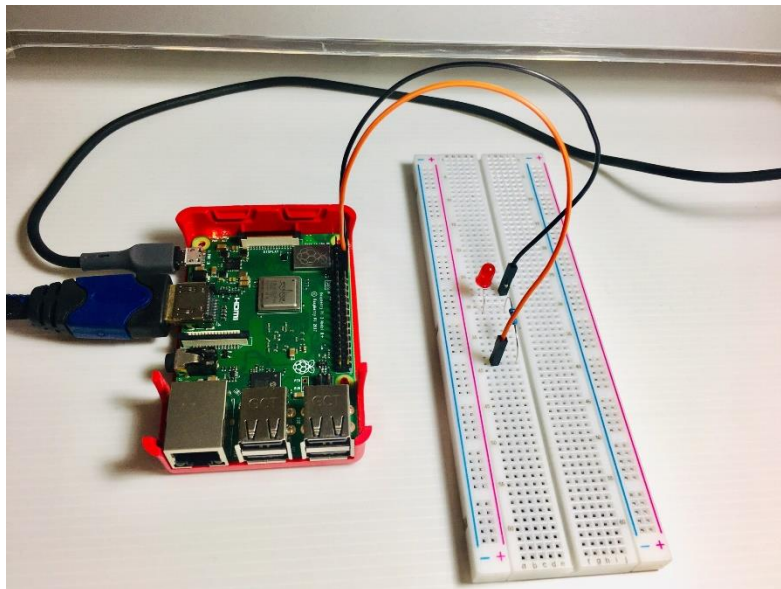
2. เสียบสาย Jumper ในแถวเดียวกับขา – (สั้น) ของหลอดไฟ LED ปลายสายเสียบเข้ากับช่อง PIN6 (GROUND) ของขา GPIO



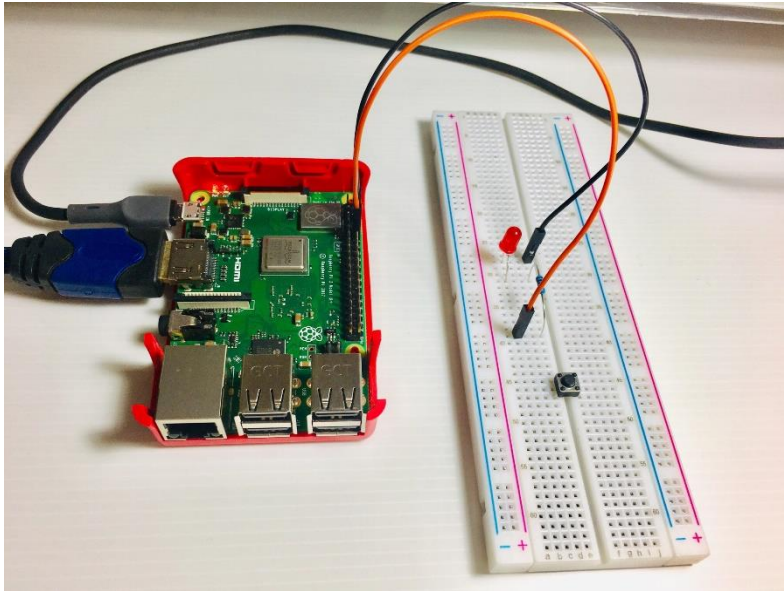
3. เสียบตัวต้านทาน 330 โอห์มเข้าไปในแถวเดียวกับขา + ของ LED



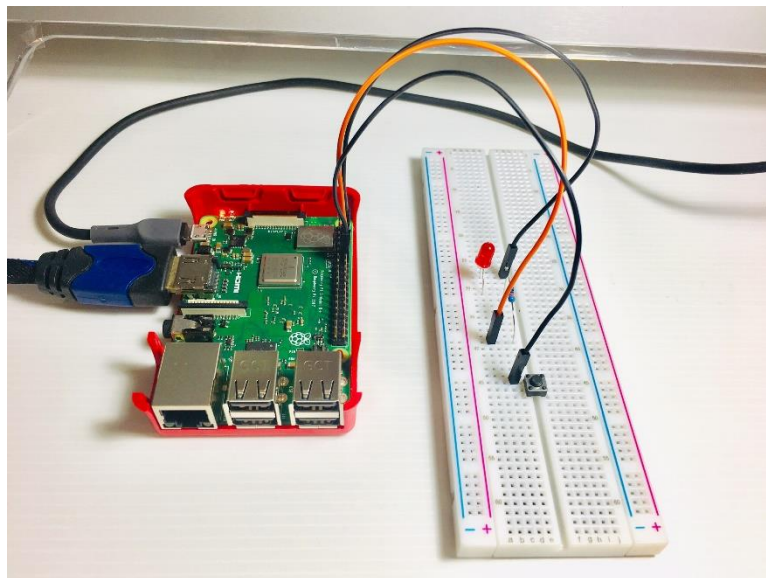
4. ใช้สาย Jumper เสียบในแถวเดียวกับตัวต้านทาน 330 โอห์มอีกด้านหนึ่ง ปลายสายเสียบเข้ากับช่อง PIN12 (GPIO18) ของขา GPIO



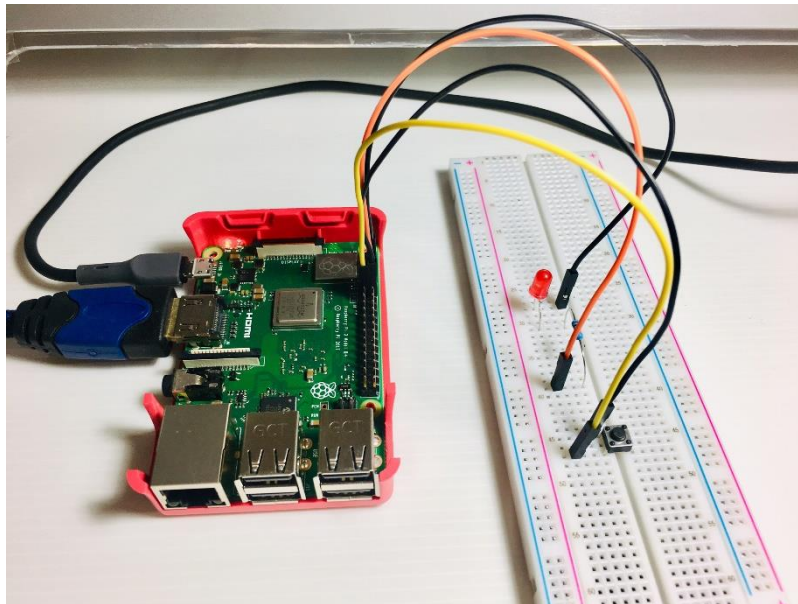
5. เสียบ Pushbutton บน Breadboard



6. เสียบสาย Jumper ที่ขาข้างหนึ่งของ Pushbutton ต่อเข้ากับขา PIN14 (GROUND) ของขา GPIO

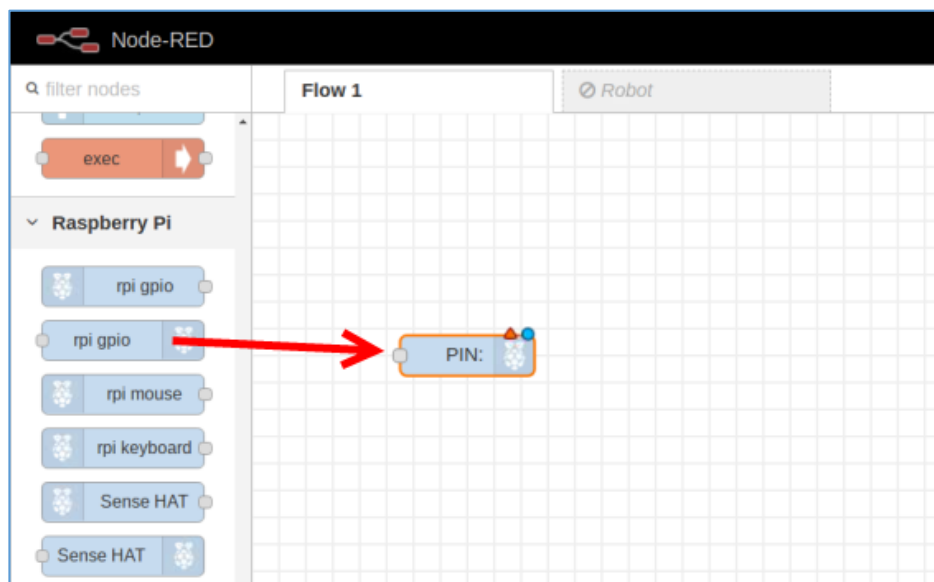


7. เสียบสาย Jumper ที่ขาอีกข้างหนึ่งของ Pushbutton ต่อเข้ากับขา PIN13 (GPIO27) ของขา GPIO

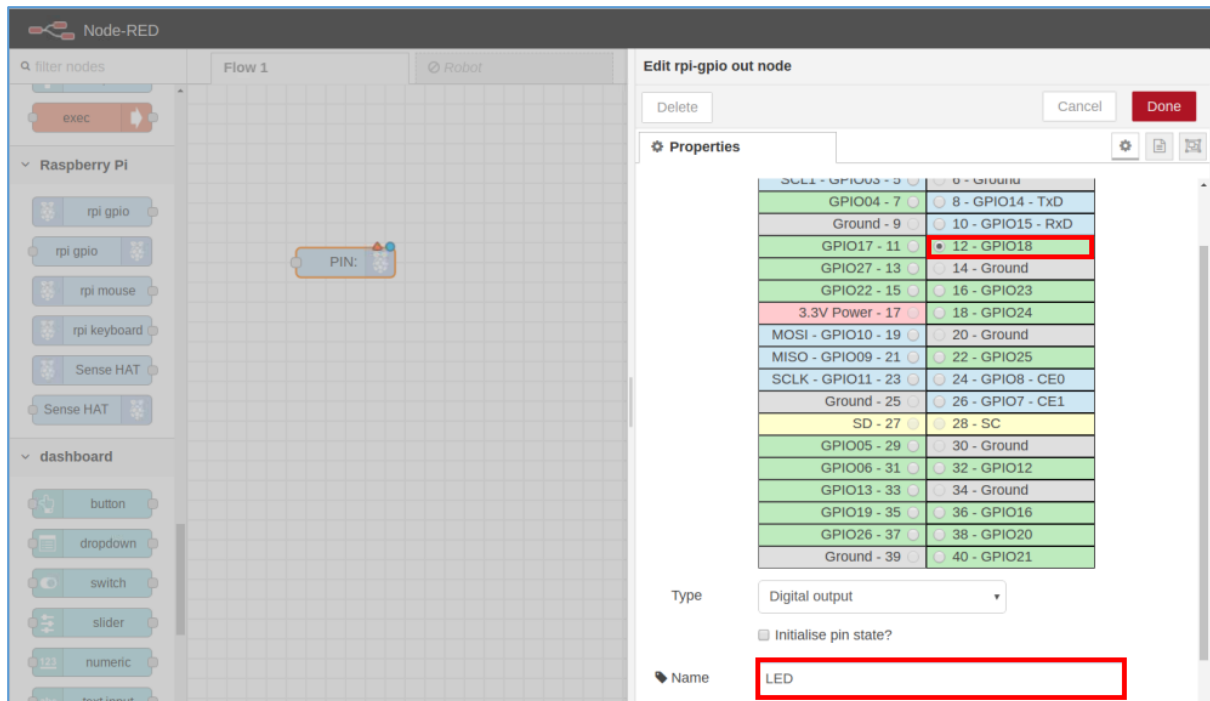


การเขียนผังวงจร

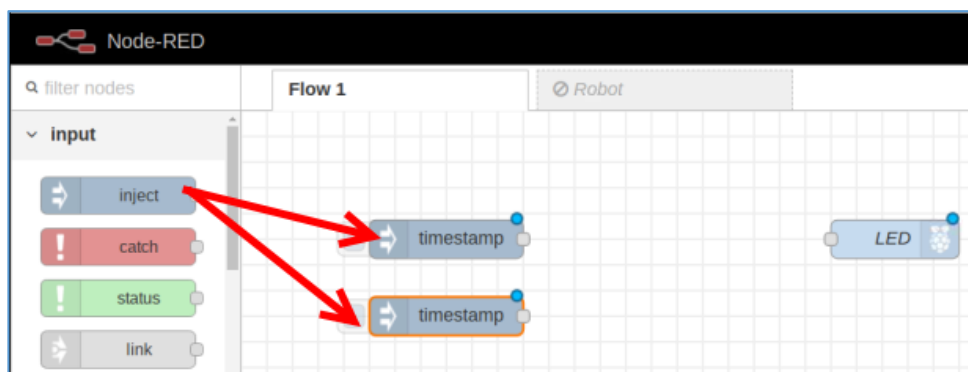
1. ลากโหนด rpi gpio ดังรูป มาวางบน Workspace



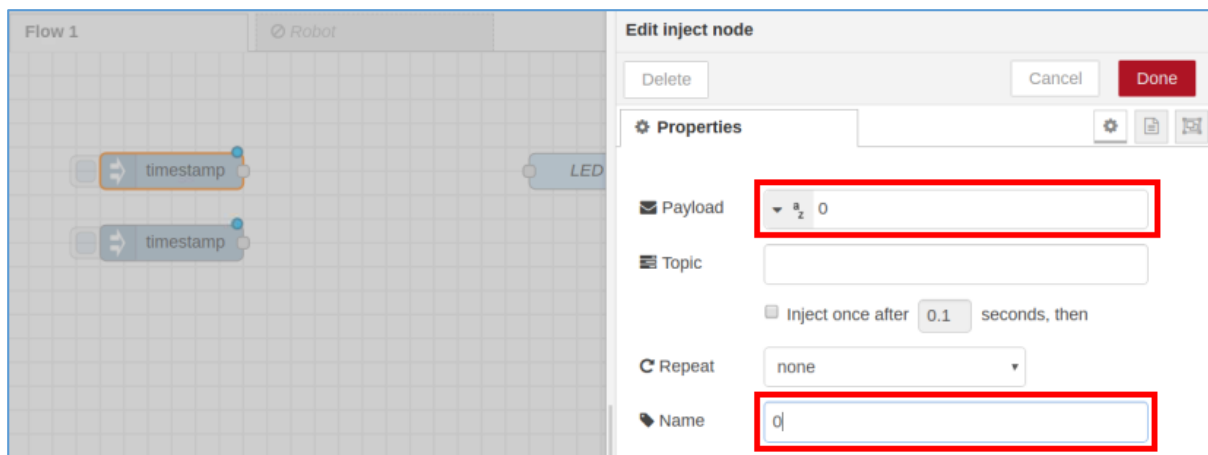
2. ดับเบิลคลิกที่โหนด PIN: เพื่อกำหนด PIN ที่เชื่อมต่อกับ LED โดยกำหนดค่าดังนี้ เสร็จแล้วคลิกปุ่ม Done



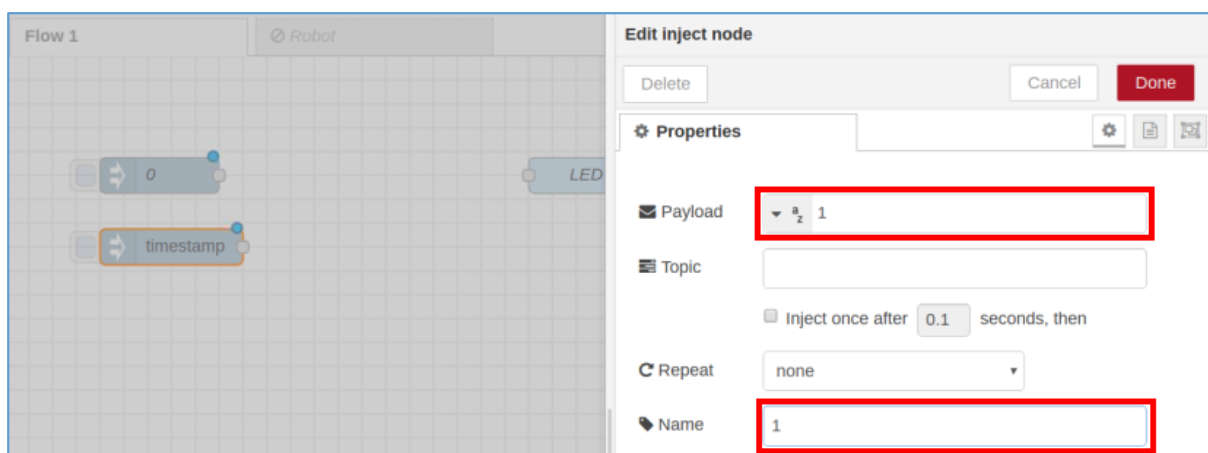
3. ลากโหนด inject มาวางบน Workspace 2 บล็อก



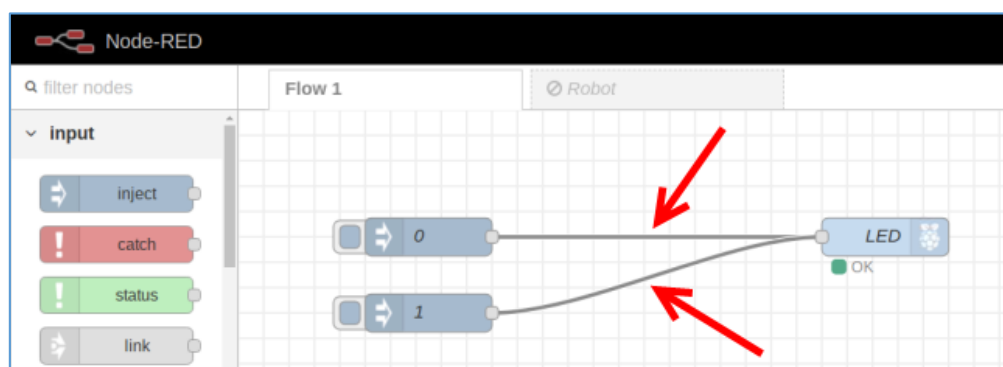
4. ดับเบิ้ลคลิกที่โหนด timestamp บล็อกแรก หลังจากนั้นกำหนดค่าดังนี้ แล้วคลิกปุ่ม Done



5. ดับเบิลคลิกที่โหนด timestamp อีกอัน หลังจากนั้นกำหนดค่าดังนี้ แล้วคลิกปุ่ม Done

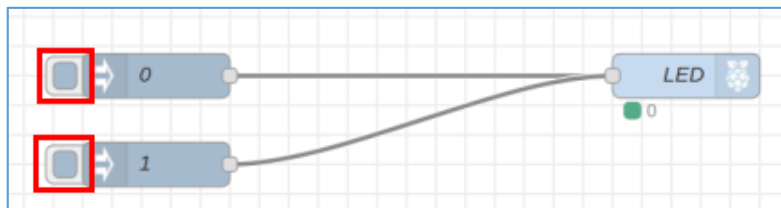


6. เชื่อมต่อโหนด 0 และ 1 ไปยังโหนด LED

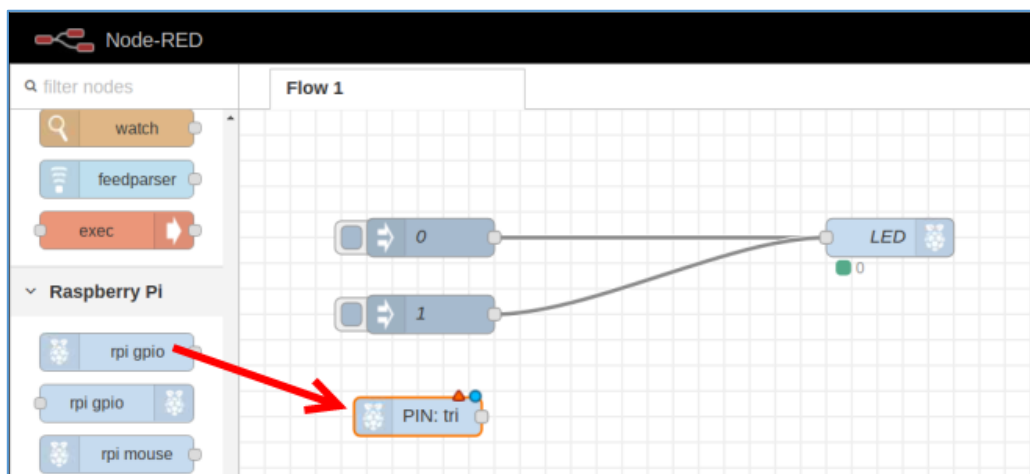


7. เสร็จแล้วคลิกปุ่ม Deploy เพื่อรันโปรแกรม

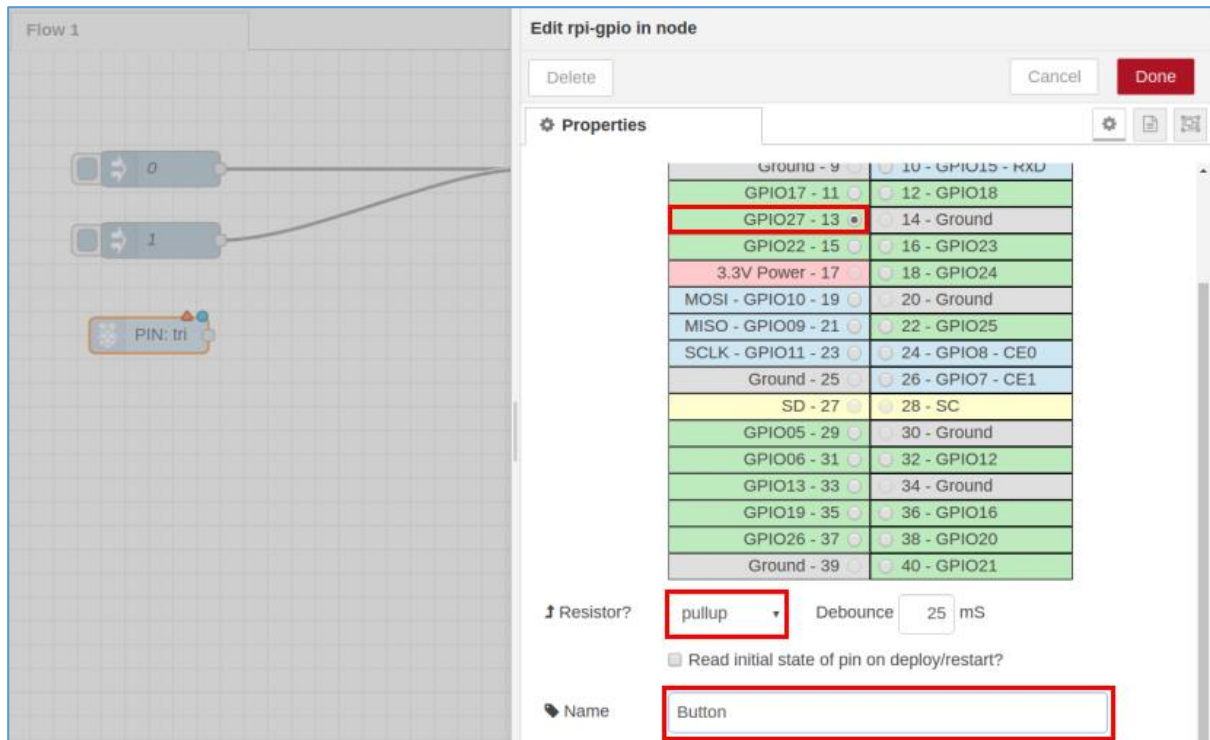
8. ทดสอบโปรแกรม โดยคลิกปุ่มหน้าโหนด 0 และปุ่มหน้าโหนด 1 เพื่อส่งค่า 0 หรือ 1 ไปยังโหนด LED ซึ่งเชื่อมต่ออยู่ที่ PIN12 (GPIO18) แล้วสังเกตไฟ LED



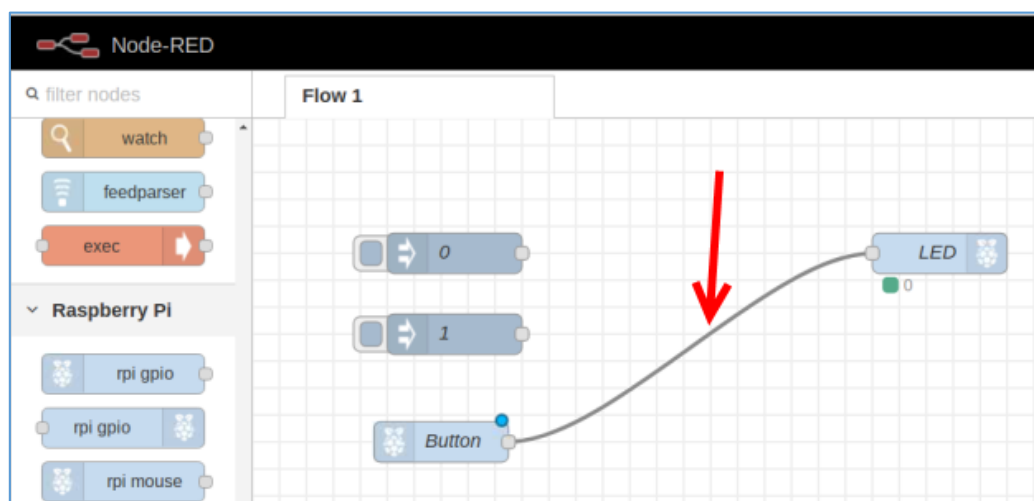
9. ลากโหนด rpi gpio ดังรูป มาวางบน Workspace



10. ดับเบิลคลิกที่โหนด PIN: tri เพื่อกำหนดค่า PIN ของ Push button แล้วกำหนดค่าดังนี้ แล้วคลิกปุ่ม Done



11. ลบเส้นที่เชื่อมโยงระหว่างโหนด 0 และ 1 ที่เชื่อมโยงไปยังโหนด LED ออก แล้วลากเส้นเชื่อมโยงระหว่างโหนด Button ไปยังโหนด LED

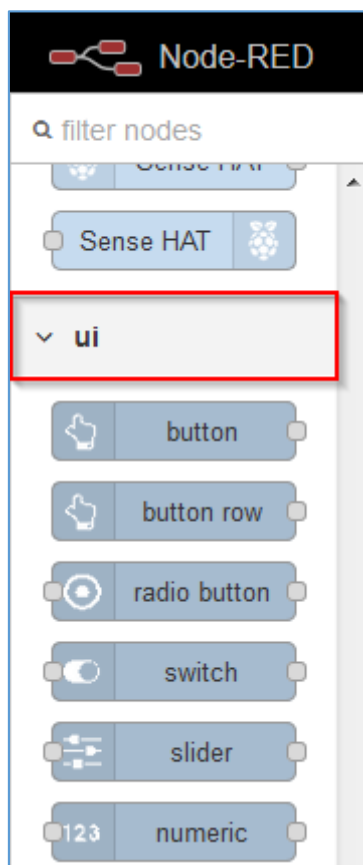


12. คลิกปุ่ม Deploy เพื่อรันโปรแกรม แล้วลองทดสอบโปรแกรมโดยคลิกปุ่ม Push Button จะเห็นได้ว่าเมื่อกดปุ่ม Push Button ไฟ LED จะดับ ถ้าไม่กดปุ่ม Push Button ไฟจะติด

การใช้งาน Node-RED UI

Node-RED UI จะเป็นส่วนเสริมของโปรแกรม Node-RED เพื่อให้โปรแกรม Node-RED สามารถสร้าง UI ไว้สำหรับการแสดงผลผ่าน Web Browser ของคอมพิวเตอร์หรือมือถือได้

ในการวาด Flow ตรงแถบเครื่องมือก็จะมี node สำหรับการสร้าง UI



เมื่อวาด Flow และได้โค้ดคำสั่งเสร็จเรียบร้อยแล้ว แล้วคลิกปุ่ม Deploy เพื่อรันโปรแกรม ในการเข้าดูผลลัพธ์หน้า Node-RED UI จะต้องเข้าใช้งานผ่าน web Browser โดยใช้ IP Address port 1880 ตามด้วย /ui เช่น 192.168.1.104:1880/ui ก็จะแสดงหน้าของ Node-RED UI ขึ้นมา

ฝึกทักษะการทดลองกิจกรรมที่ 3 เรียนรู้การใช้งาน Node-RED UI : เปิด-ปิดไฟ LED

อุปกรณ์ที่ใช้สำหรับการทดลอง

1. บอร์ด Raspberry Pi



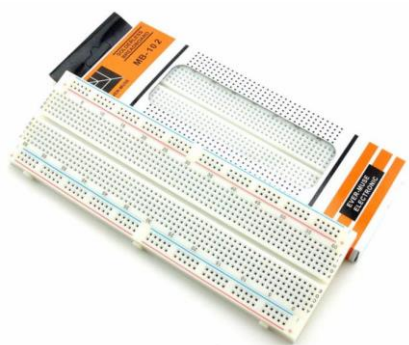
2. หลอด LED



3. ตัวต้านทาน 330 โอห์ม



4. Breadboard

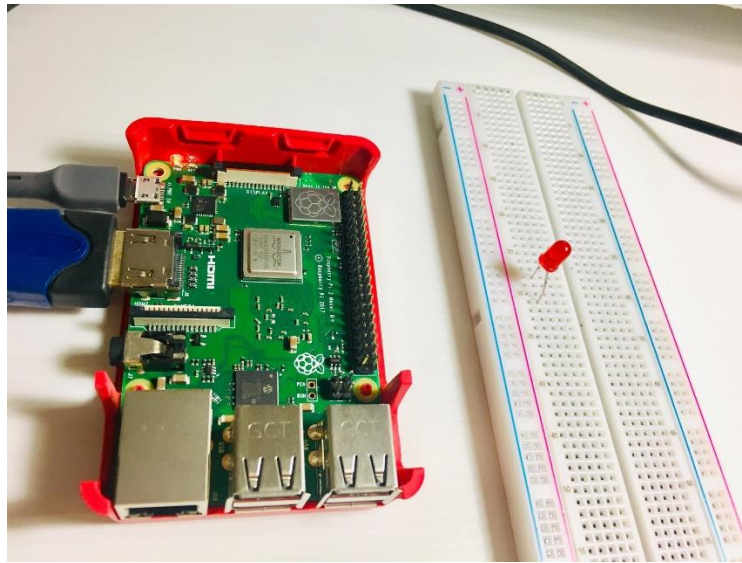


5. สาย Jumper

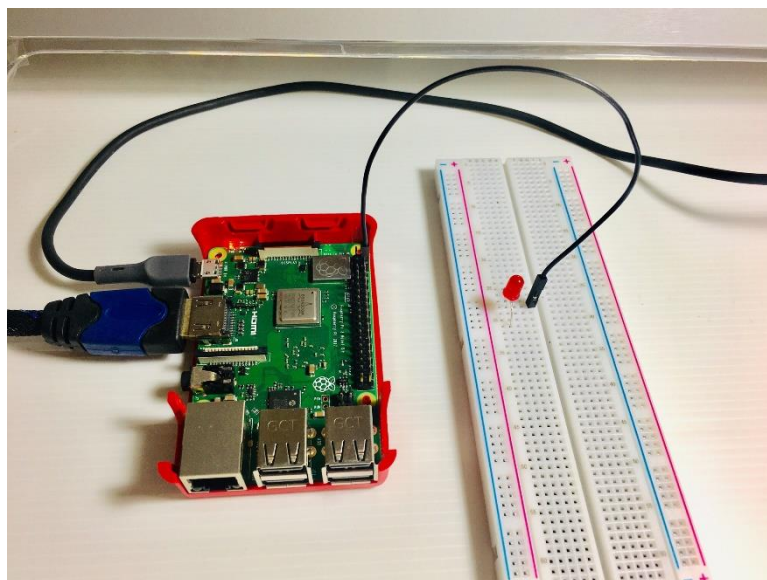


การต่อวงจร

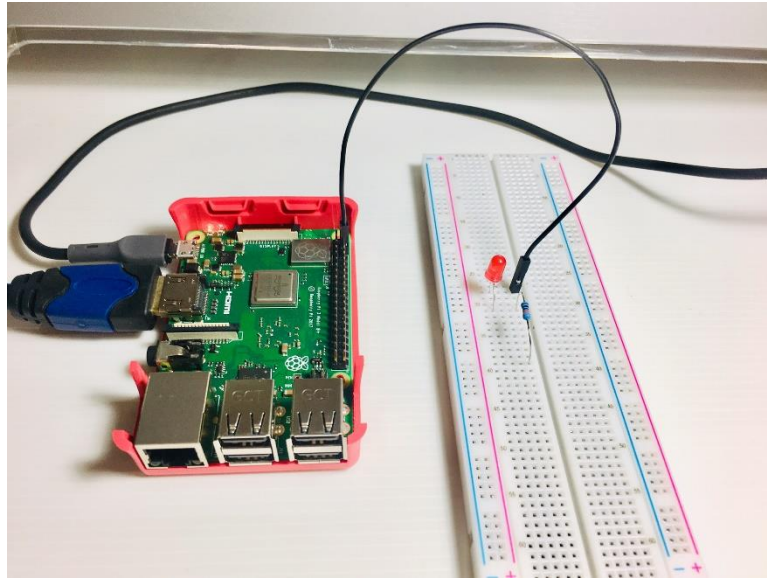
1. เสียบ LED บน Breadboard



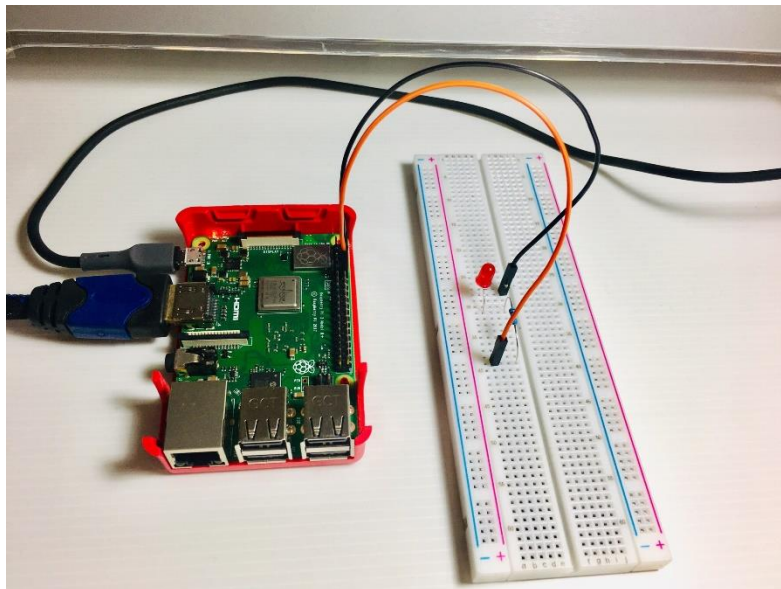
2. เสียบสาย Jumper ในแถวเดียวกับขา – (สั้น) ของหลอดไฟ LED ปลายสายเสียบเข้ากับช่อง PIN6 (GROUND) ของขา GPIO



3. เสียบตัวต้านทาน 330 โอห์มเข้าไปในแถวเดียวกับขา + ของ LED

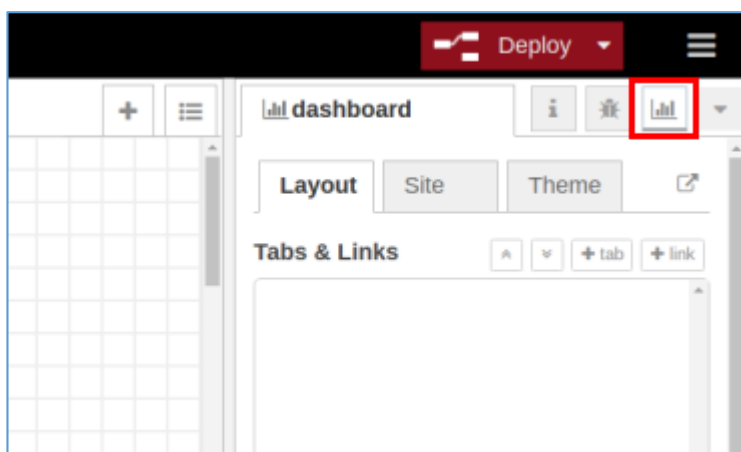


4. ใช้สาย Jumper เสียบในแถวเดียวกับตัวต้านทาน 330 โอห์มอีกด้านหนึ่ง ปลายสายเสียบเข้ากับช่อง PIN12 (GPIO18) ของขา GPIO

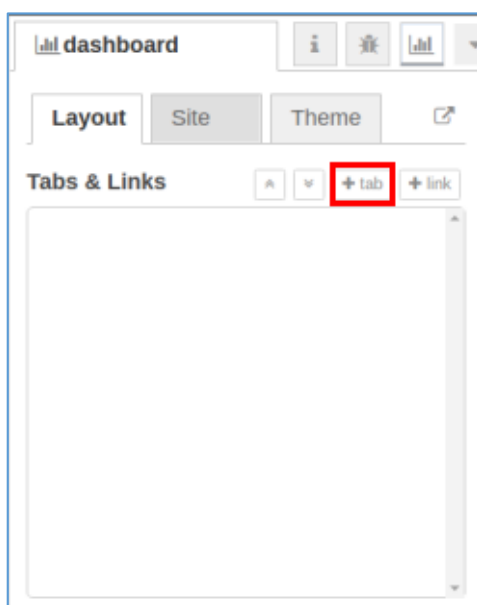


การออกแบบ Flow

1. คลิกไปที่มุมมอง Dashboard



2. คลิกสร้างแท็บใหม่ขึ้นมา โดยคลิกที่ + tab



3. คลิกที่ปุ่ม edit ด้านหลัง Tab1 โดยกำหนดค่าดังนี้ แล้วคลิกปุ่ม Update

Edit dashboard tab node

Delete Cancel Update

Properties

Name **Control LED**

Icon dashboard

State ☒ Enabled

Nav. Menu ☒ Visible

4. คลิกที่ปุ่ม + group ด้านหลังแท็บ Control LED จะได้ Group 1 ขึ้นมา

dashboard

Layout Site Theme

Tabs & Links

Control LED

> Group 1 + spacer edit

5. คลิกปุ่ม edit ด้านหลัง Group 1 แล้วกำหนดค่าดังนี้ แล้วคลิกปุ่ม Update

Edit dashboard group node

Delete Cancel Update

Properties

Name **ON/OFF**

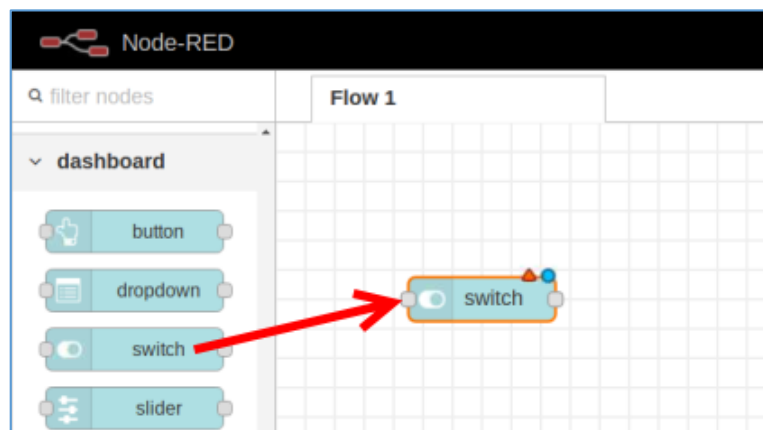
Tab Control LED

Width 6

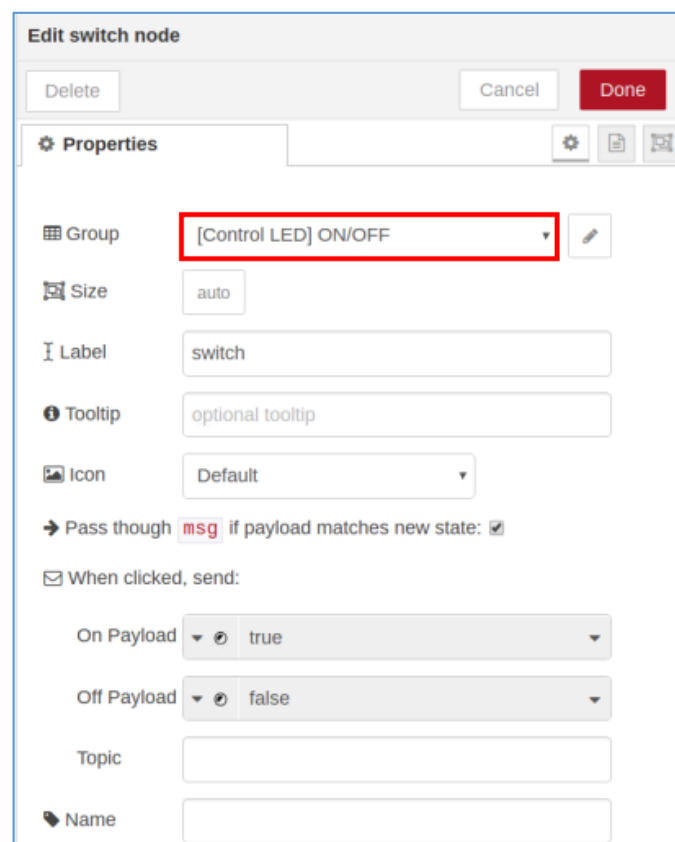
☒ Display group name

☐ Allow group to be collapsed

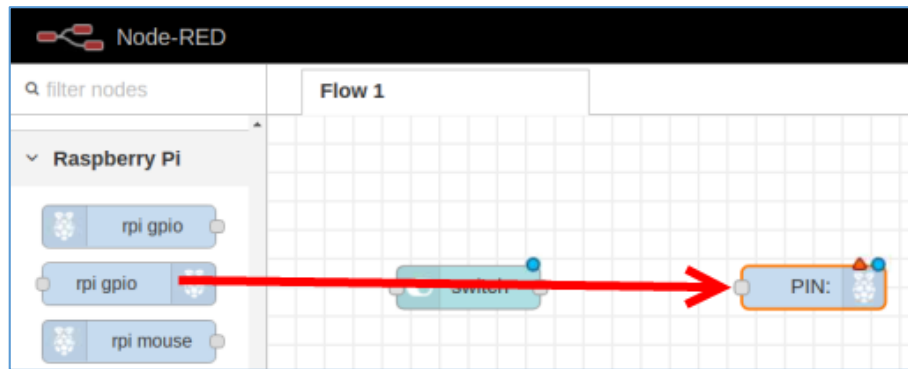
6. ลากโหนด switch มาวางบน Workspace



7. ดับเบิ้ลคลิกที่โหนด switch เพื่อกำหนดให้แสดงปุ่ม switch ในกลุ่มใด แท็บใด โดยกำหนดค่าดังนี้ แล้วคลิกปุ่ม Done



8. ลากโหนด rpi gpio มาวางบน Workspace



9. ดับเบิลคลิกที่โหนด PIN: เพื่อกำหนดขา PIN ของ LED โดยกำหนดค่าดังนี้ แล้วคลิกปุ่ม Done

The image shows the 'Edit rpi-gpio out node' dialog box. It has a 'Delete' button, 'Cancel' and 'Done' buttons, and a 'Properties' tab. The main area is a table of GPIO pins with radio buttons for selection. The pin '12 - GPIO18' is selected and highlighted with a red box. Below the table, the 'Type' is set to 'Digital output' and the 'Name' is set to 'LED', both highlighted with red boxes.

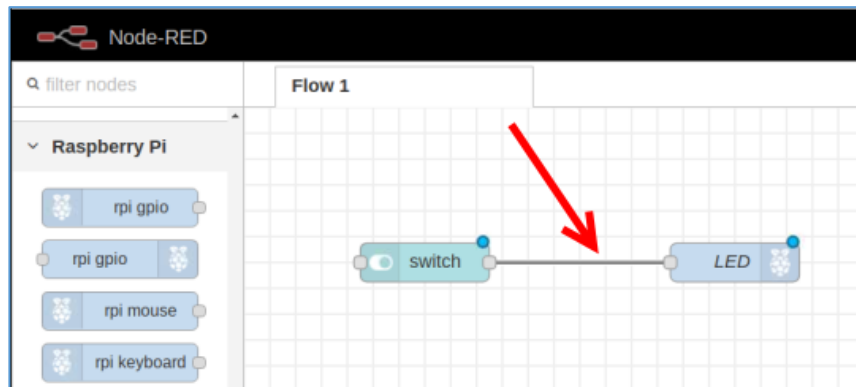
SCL1 - GPIO03 - 5	8 - Ground
GPIO04 - 7	8 - GPIO14 - TxD
Ground - 9	10 - GPIO15 - RxD
GPIO17 - 11	12 - GPIO18
GPIO27 - 13	14 - Ground
GPIO22 - 15	16 - GPIO23
3.3V Power - 17	18 - GPIO24
MOSI - GPIO10 - 19	20 - Ground
MISO - GPIO09 - 21	22 - GPIO25
SCLK - GPIO11 - 23	24 - GPIO8 - CE0
Ground - 25	26 - GPIO7 - CE1
SD - 27	28 - SC
GPIO05 - 29	30 - Ground
GPIO06 - 31	32 - GPIO12
GPIO13 - 33	34 - Ground
GPIO19 - 35	36 - GPIO16
GPIO26 - 37	38 - GPIO20
Ground - 39	40 - GPIO21

Type: Digital output

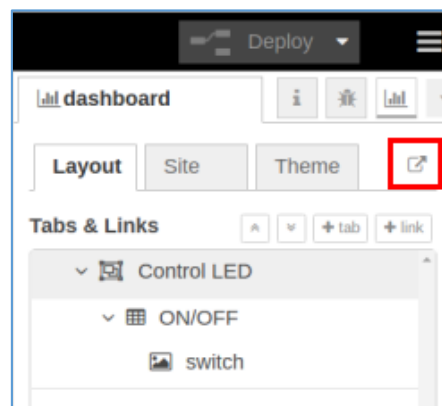
☐ Initialise pin state?

Name: LED

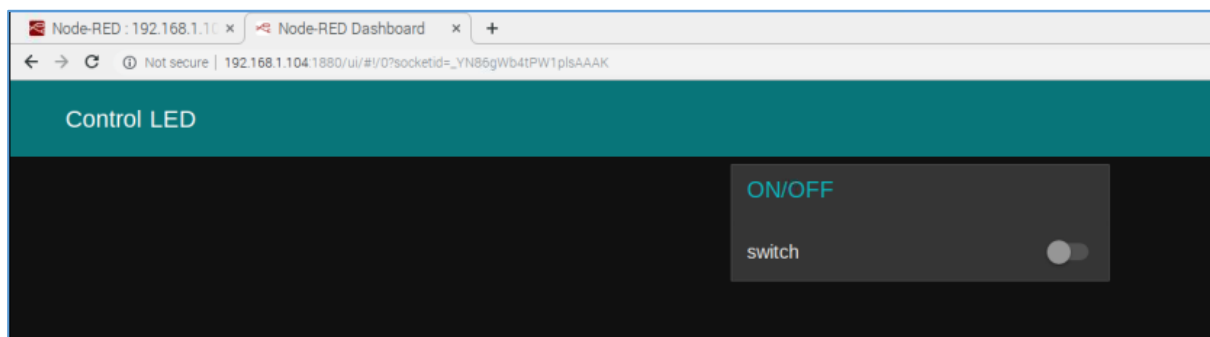
10. ลากเส้นเชื่อมโยงจากโหนด switch ไปยังโหนด LED



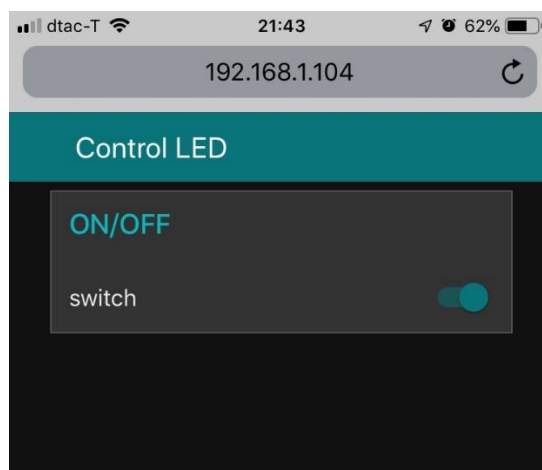
11. คลิกปุ่ม Deploy แล้วคลิกที่ปุ่มดังรูป เพื่อเปิดหน้า UI ขึ้นมาให้ใช้งาน



12. Web Browser จะเปิดแท็บใหม่ขึ้นมา ซึ่งเป็นหน้าเพจสำหรับเปิด-ปิดสวิตช์ควบคุมไฟ LED ตามที่ได้ออกแบบ Flow เอาไว้



13. ทดลองใช้มือถือหรือแท็บเล็ตจับสัญญาณ Wi-Fi ตัวเดียวกับ Raspberry Pi หลังจากนั้นเปิด Web Brower ขึ้นมา แล้วใส่ IP Address ของ Raspberry Pi ตามด้วย :1880/ui เช่น 192.168.1.104:1880/ui เมื่อแตะเปิดสวิตช์ ไฟ LED จะติด เมื่อแตะปิดสวิตช์ ไฟ LED จะดับ



ฝึกทักษะการทดลองกิจกรรมที่ 4 เรียนรู้การใช้งาน Node-RED UI : วัดค่าความชื้นและอุณหภูมิในอากาศ อุปกรณ์ที่ใช้สำหรับการทดลอง

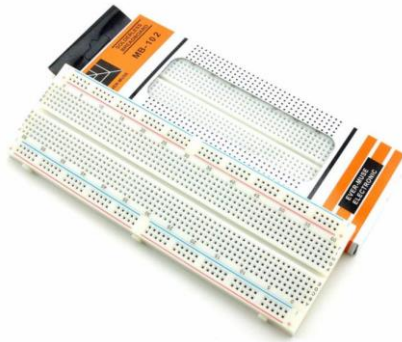
1. บอร์ด Raspberry Pi



2. เซนเซอร์วัดความชื้นและอุณหภูมิ DHT22



3. Breadboard

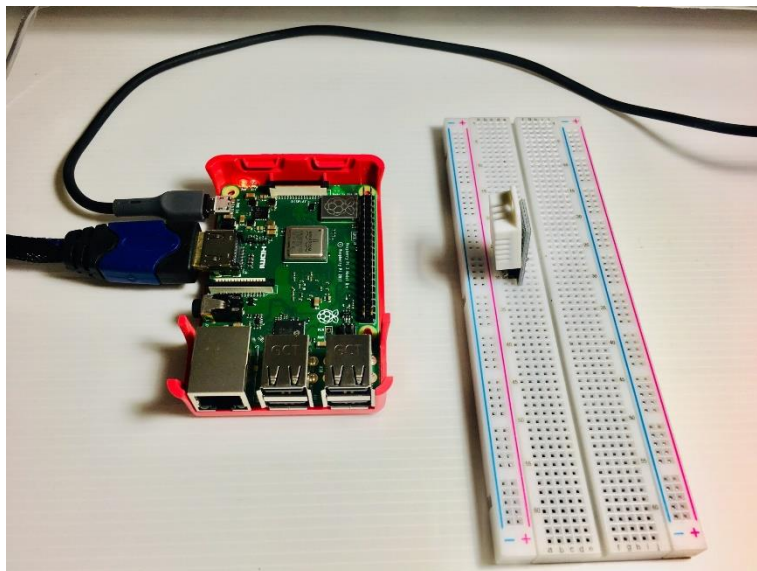


4. สาย Jumper

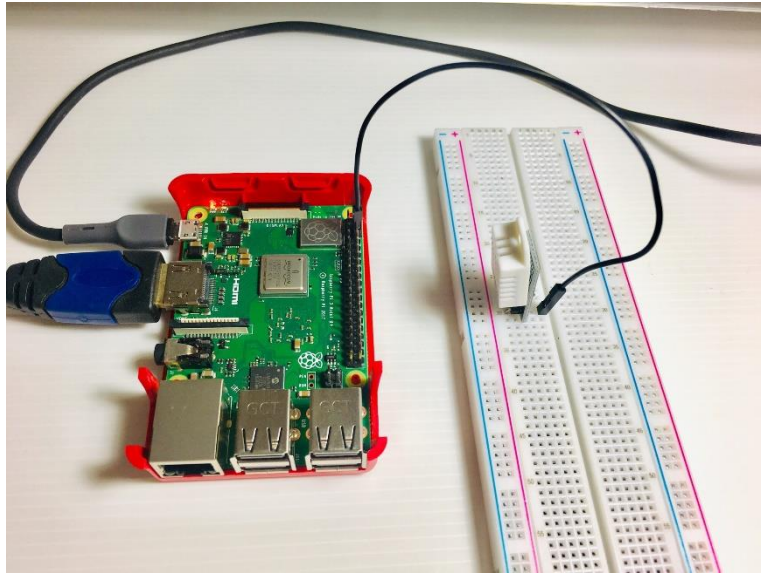


การต่อวงจร

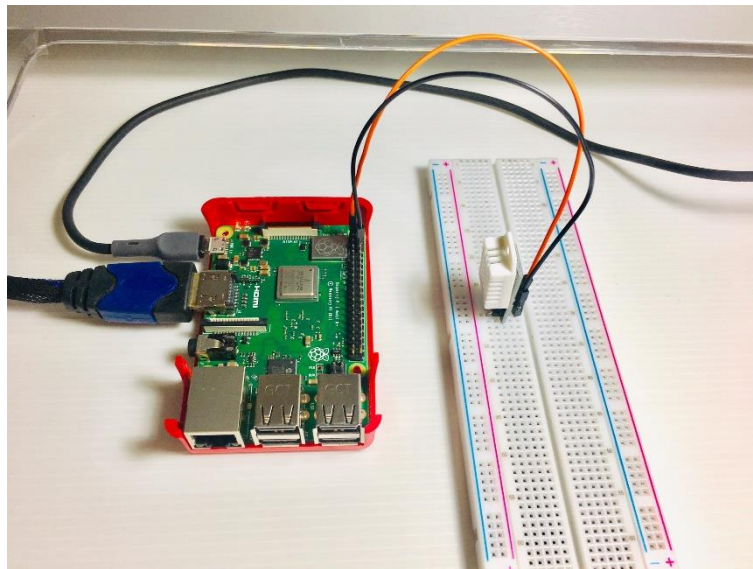
1. เสียบเซนเซอร์ DHT22 บน Breadboard



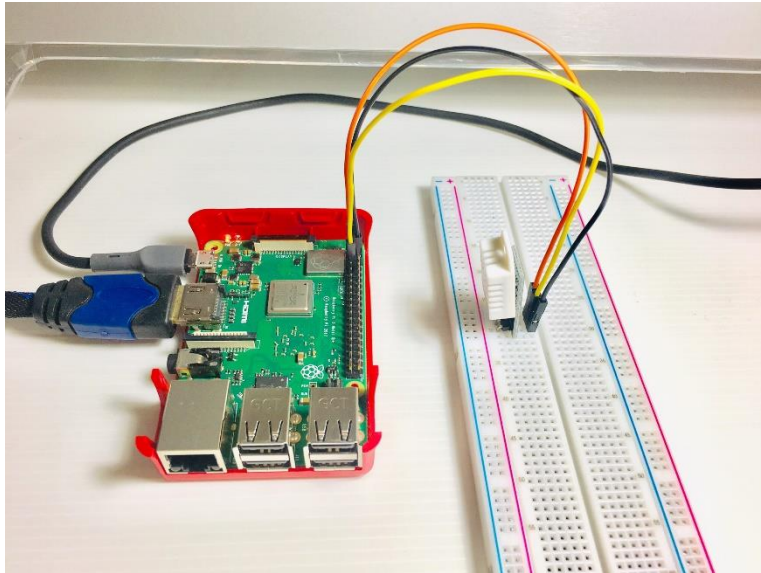
2. เสียบสาย Jumper ต่อจากขาลบ – ของ DHT22 ไปยัง PIN6 (GROUND) ของขา GPIO



3. เสียบสาย Jumper ต่อจากขาลบ + ของ DHT22 ไปยังขา PIN1 (3V3) ของขา GPIO



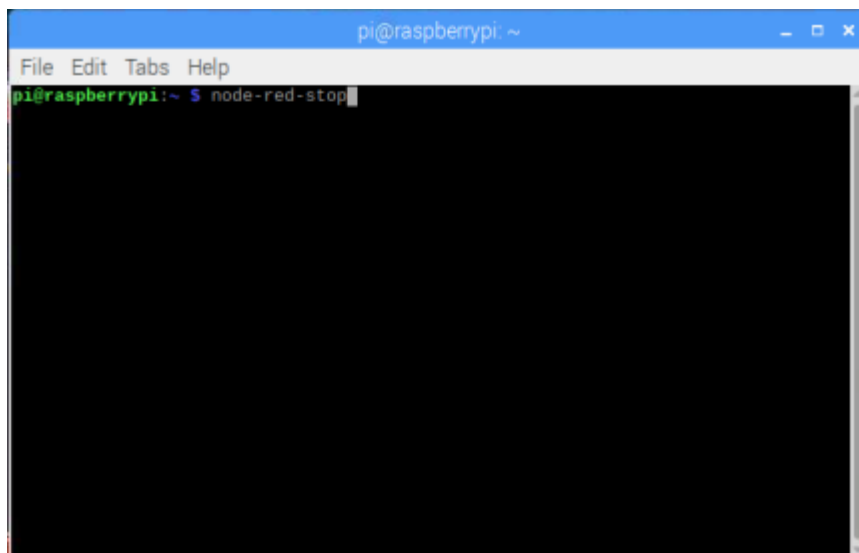
4. เสียบสาย Jumper ต่อจากขา out ของ DHT22 ไปยัง PIN7 (GPIO4) ของขา GPIO



การออกแบบ Flow และการเขียนโปรแกรม

ก่อนการใช้งาน DHT22 กับ Node-RED จะต้องทำการดาวน์โหลดและติดตั้งโหนด DHT และไลบรารี BCM2835 ก่อน โดยทำตามขั้นตอน ต่อไปนี้

1. เปิดโปรแกรม Terminal แล้วพิมพ์คำสั่ง `node-red-stop` เพื่อปิดการใช้งาน node-red ชั่วคราว



2. พิมพ์คำสั่ง `cd ~/.node-red/` เพื่อเข้าไปในโฟลเดอร์ node-red

```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ node-red-stop
Stop Node-RED
Use node-red-start to start Node-RED again
^[[A
pi@raspberrypi:~ $ cd ~/.node-red/

```

3. ก่อนที่จะดาวน์โหลดและติดตั้งโปรแกรมอะไรก็แล้วแต่ จะต้องอัปเดตซอฟต์แวร์ระบบเสียก่อน โดยพิมพ์คำสั่ง `sudo apt-get update`

```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ node-red-stop
Stop Node-RED
Use node-red-start to start Node-RED again
^[[A
pi@raspberrypi:~ $ cd ~/.node-red/
pi@raspberrypi:~/.node-red $ sudo apt-get update
Get:1 http://raspbian.raspberrypi.org/raspbian stretch InRelease [15.0 kB]
Get:2 http://archive.raspberrypi.org/debian stretch InRelease [25.4 kB]
Get:3 http://raspbian.raspberrypi.org/raspbian stretch/main armhf Packages [11.7 MB]
75% [3 Packages 8,047 kB/11.7 MB 69%]

```

4. ติดตั้ง node-dht-sensor โดยพิมพ์คำสั่ง `npm install node-red-contrib-dht-sensor`

```

pi@raspberrypi: ~/node-red
File Edit Tabs Help
^[[A
pi@raspberrypi:~ $ cd ~/.node-red/
pi@raspberrypi:~/node-red $ sudo apt-get update
Get:1 http://raspbian.raspberrypi.org/raspbian stretch InRelease [15.0 kB]
Get:2 http://archive.raspberrypi.org/debian stretch InRelease [25.4 kB]
Get:3 http://raspbian.raspberrypi.org/raspbian stretch/main armhf Packages [11.7
  MB]
Fetched 11.7 MB in 13s (859 kB/s)
Reading package lists... Done
pi@raspberrypi:~/node-red $ sudo apt-get install npm
Reading package lists... Done
Building dependency tree
Reading state information... Done
npm is already the newest version (1.4.21+ds-2)..
The following packages were automatically installed and are no longer required:
  arduino-core avr-libc avrdude binutils-avr coinor-libipoptlv5
  extra-xdg-menus gcc-avr libdirectfb-1.2-0 libftdi1 libiso9660-8 libjna-java
  libjna-jni libmumps-seq-4.10.0 libraw15 librx-tx-java libvcdinfo0 libvllcore8
  lxkeymap python-cairo python-gobject python-gobject-2 python-gtk2
  python-xklavier realpath
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 68 not upgraded.
pi@raspberrypi:~/node-red $ npm install node-red-contrib-dht-sensor
[.....] | fetchMetadata:

```

5. การติดตั้งโหนด dht ต้องทำอีก 3 ขั้นตอน ได้แก่

5.1 พิมพ์คำสั่ง `sudo npm install --unsafe-perm -g node-dht-sensor`

```

pi@raspberrypi: ~/node-red
File Edit Tabs Help
> node-gyp build
make: Entering directory '/home/pi/.node-red/node_modules/node-dht-sensor/build'
CC(target) Release/obj.target/nothing/./node-addon-api/src/nothing.o
AR(target) Release/obj.target/./node-addon-api/src/nothing.a
COPY Release/nothing.a
CC(target) Release/obj.target/node_dht_sensor/src/bcm2835/bcm2835.o
CXX(target) Release/obj.target/node_dht_sensor/src/node-dht-sensor.o
CXX(target) Release/obj.target/node_dht_sensor/src/dht-sensor.o
CXX(target) Release/obj.target/node_dht_sensor/src/util.o
SOLINK_MODULE(target) Release/obj.target/node_dht_sensor.node
COPY Release/node_dht_sensor.node
make: Leaving directory '/home/pi/.node-red/node_modules/node-dht-sensor/build'
node-red-project@0.0.1 /home/pi/.node-red
├─ node-red-contrib-dht-sensor@1.0.4
│ └─ node-dht-sensor@0.3.0
│   └─ node-addon-api@1.7.1
pi@raspberrypi:~/node-red $ sudo npm install --unsafe-perm -g node-dht-sensor
> node-dht-sensor@0.3.0 install /usr/lib/node_modules/node-dht-sensor
> node-gyp configure

```

5.2 พิมพ์คำสั่ง `sudo npm install --unsafe-perm -g node-red-contrib-dht-sensor`


```

pi@raspberrypi: ~/node-red
File Edit Tabs Help
> node-dht-sensor@0.3.0 install /usr/lib/node_modules/node-red-contrib-dht-sensor/node_modules/node-d
ht-sensor
> node-gyp configure

> node-dht-sensor@0.3.0 postinstall /usr/lib/node_modules/node-red-contrib-dht-sensor/node_modules/node-dht-s
ensor
> node-gyp build

make: Entering directory '/usr/lib/node_modules/node-red-contrib-dht-sensor/node_modules/node-dht-sensor/build
d'
  CC(target) Release/obj.target/nothing/../node-addon-api/src/nothing.o
  AR(target) Release/obj.target/nothing/../node-addon-api/src/nothing.a
  COPY Release/nothing.a
  CC(target) Release/obj.target/node_dht_sensor/src/bcm2835/bcm2835.o
  CXX(target) Release/obj.target/node_dht_sensor/src/node-dht-sensor.o
  CXX(target) Release/obj.target/node_dht_sensor/src/dht-sensor.o
  CXX(target) Release/obj.target/node_dht_sensor/src/util.o
  SOLINK_MODULE(target) Release/obj.target/node_dht_sensor.node
  COPY Release/node_dht_sensor.node
make: Leaving directory '/usr/lib/node_modules/node-red-contrib-dht-sensor/node_modules/node-dht-sensor/build
d'

/usr/lib
├── node-red-contrib-dht-sensor@1.0.4
├── node-dht-sensor@0.3.0
└── node-addon-api@1.7.1

pi@raspberrypi:~/node-red $ sudo npm install --unsafe-perm -g node-red-contrib-dht-sensor

```

5.3 ดาวน์โหลดและติดตั้งไลบรารี BCM2835 โดยพิมพ์คำสั่ง

wget <http://www.airspayce.com/mikem/bcm2835/bcm2835-1.5.tar.gz>

```

pi@raspberrypi: ~/node-red
File Edit Tabs Help
pi@raspberrypi:~/node-red $ cd ~/node-red/
pi@raspberrypi:~/node-red $ wget http://www.airspayce.com/mikem/bcm2835/bcm2835-1.5.tar.gz
--2019-09-21 16:20:20-- http://www.airspayce.com/mikem/bcm2835/bcm2835-1.5.tar.gz
Resolving www.airspayce.com (www.airspayce.com)... 192.185.48.187
Connecting to www.airspayce.com (www.airspayce.com)[192.185.48.187]:80... connect
ed.
HTTP request sent, awaiting response... 200 OK
Length: 220573 (215K) [application/x-gzip]
Saving to: 'bcm2835-1.5.tar.gz'

bcm2835-1.5.tar.gz 100%[=====] 215.40K 1.46MB/s in 0.2s

2019-09-21 16:20:21 (1.40 MB/s) - 'bcm2835-1.5.tar.gz' saved [220573/220573]

pi@raspberrypi:~/node-red $

```

5.4 แยกไฟล์ Zip โดยพิมพ์คำสั่ง tar zxvf bcm2835-1.5.tar.gz

```

pi@raspberrypi: ~/node-red
File Edit Tabs Help
Resolving www.airspayce.com (www.airspayce.com)... 192.185.48.187
Connecting to www.airspayce.com (www.airspayce.com)[192.185.48.187]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 220573 (215K) [application/x-gzip]
Saving to: 'bcm2835-1.5.tar.gz'

bcm2835-1.5.tar.gz 100%[=====] 215.40K 1.40MB/s in 0.2s

2019-09-21 16:20:21 (1.40 MB/s) - 'bcm2835-1.5.tar.gz' saved [220573/220573]

pi@raspberrypi:~/node-red $ tar zxvf bcm2835-1.55.tar.gz
tar (child): bcm2835-1.55.tar.gz: Cannot open: No such file or directory
tar (child): Error is not recoverable: exiting now
tar: Child returned status 2
tar: Error is not recoverable: exiting now
pi@raspberrypi:~/node-red $ ls
bcm2835-1.5.tar.gz  lib  package.json  wget-log
flows_raspberrypi.json  node_modules  settings.js
pi@raspberrypi:~/node-red $ tar zxvf bcm2835-1.5.tar.gz
bcm2835-1.5/
bcm2835-1.5/config.sub
bcm2835-1.5/ChangeLog
bcm2835-1.5/doc/

```

5.5 เข้าไปในโฟลเดอร์ bcm2835-1.5 โดยพิมพ์คำสั่ง `cd bcm2835-1.5`

```

pi@raspberrypi: ~/node-red/bcm2835-1.5
File Edit Tabs Help
bcm2835-1.5/config.guess
bcm2835-1.5/configure.in
bcm2835-1.5/NEWS
bcm2835-1.5/COPYING
bcm2835-1.5/Makefile.am
bcm2835-1.5/ltmain.sh
bcm2835-1.5/configure
bcm2835-1.5/missing
bcm2835-1.5/install.sh
bcm2835-1.5/Makefile.in
bcm2835-1.5/AUTHORS
bcm2835-1.5/examples/
bcm2835-1.5/examples/blink/
bcm2835-1.5/examples/blink/blink.c
bcm2835-1.5/examples/input/
bcm2835-1.5/examples/input/input.c
bcm2835-1.5/examples/spin/
bcm2835-1.5/examples/spin/spin.c
bcm2835-1.5/examples/event/
bcm2835-1.5/examples/event/event.c
bcm2835-1.5/examples/spi/
bcm2835-1.5/examples/spi/spi.c
pi@raspberrypi:~/node-red $ cd bcm2835-1.5
pi@raspberrypi:~/node-red/bcm2835-1.5

```

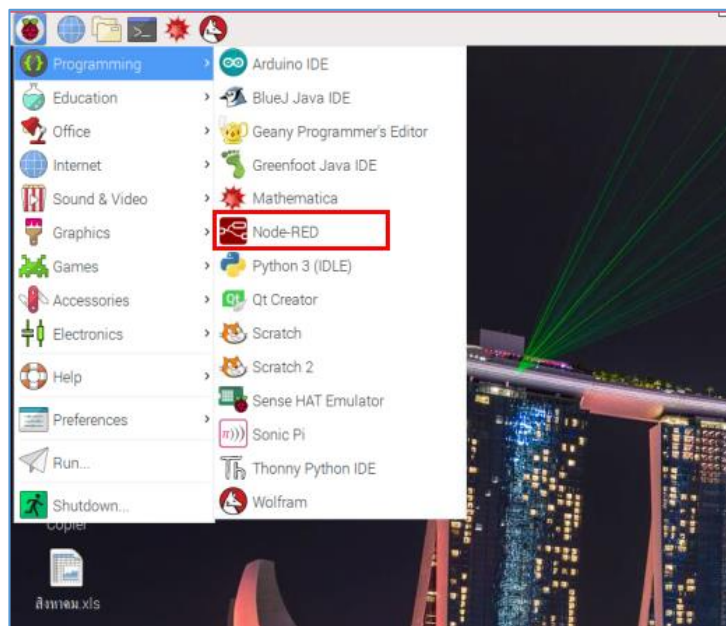
5.6 พิมพ์คำสั่ง 3 คำสั่งตามลำดับ

`./configure`

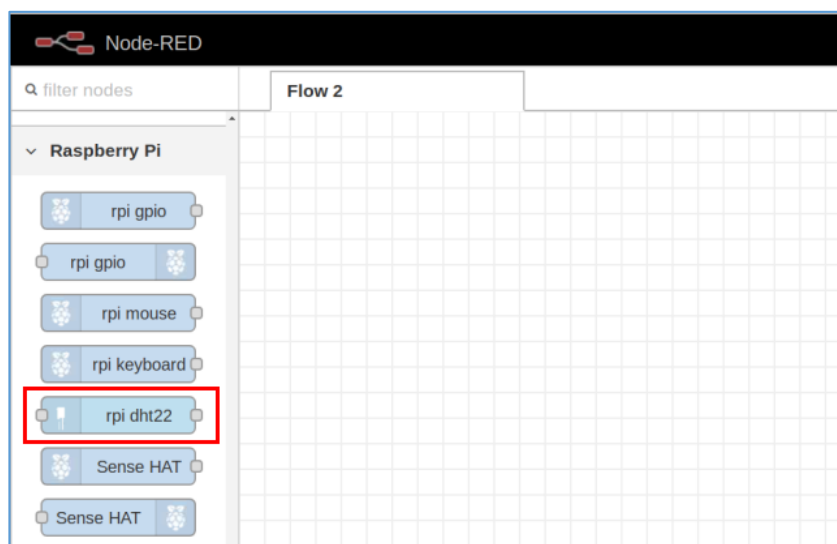
`make`

`sudo make install`

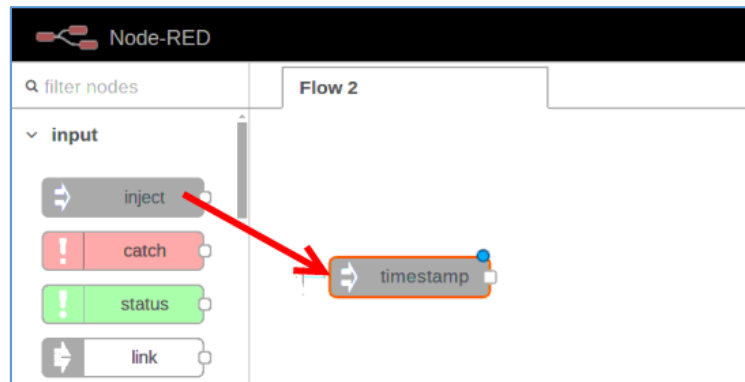
6. ปิดหน้าต่าง Terminal แล้วไปที่ Pi > Programming > Node-RED เพื่อเปิดใช้งาน Node-RED อีกครั้ง



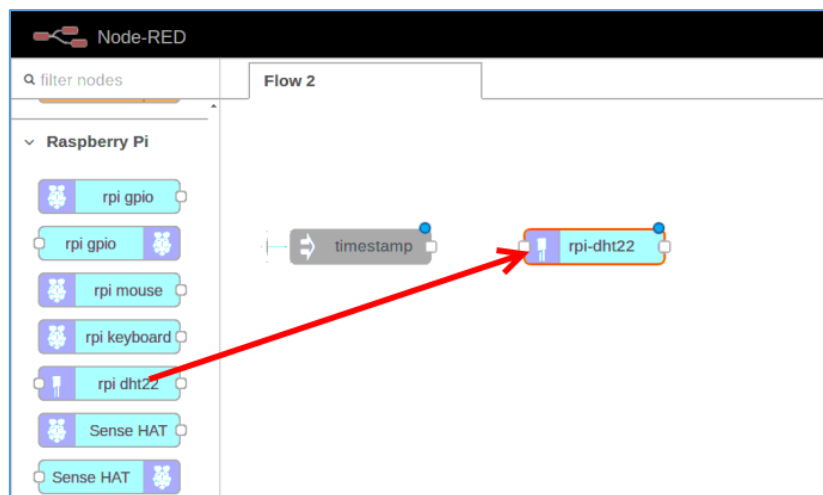
7. เปิด Web Browser เพื่อเข้าหน้าจอ Node-RED จะปรากฏโหนด rpi dht22 เพิ่มขึ้นมาอีก 1 โหนด



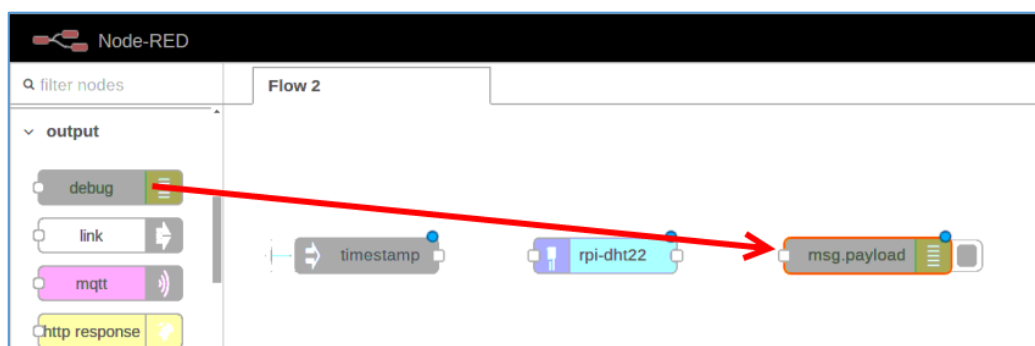
8. ลากโหนด inject มาวางใน Workspace



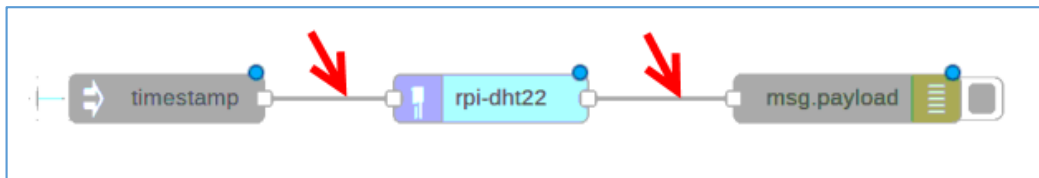
9. ลากโหนด rpi dht22 มาวางใน Workspace



10. ลากโหนด debug มาวางใน Workspace

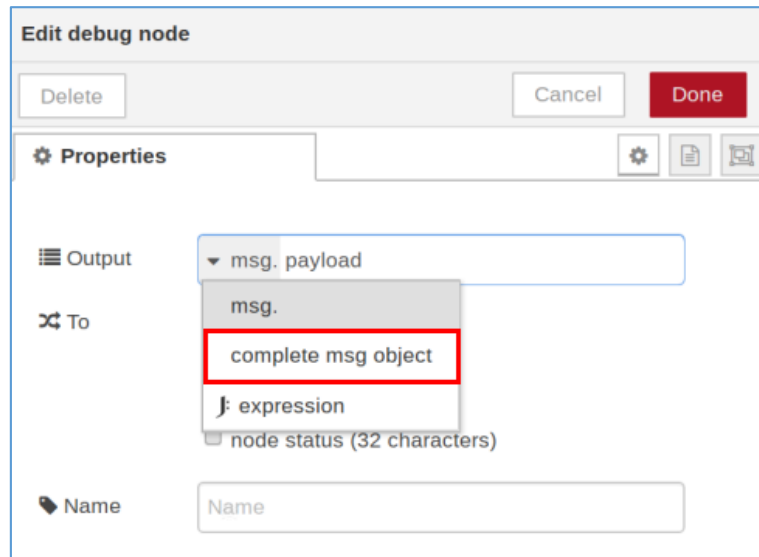


11. ลากเส้นเชื่อมโยงทั้ง 3 โหนด ดังรูป

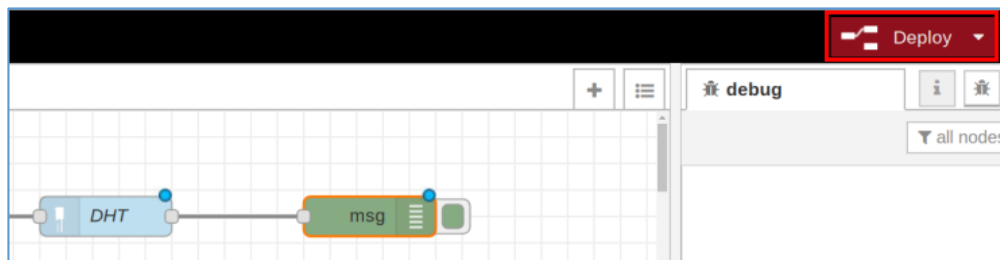


12. ดับเบิลคลิกที่โหนด rpi-dht22 เพื่อกำหนดรุ่นและขา PIN ของเซนเซอร์ DHT (ซึ่งมี 2 รุ่นคือ DHT11 และ DHT22) โดยกำหนดค่าดังรูป แล้วคลิกปุ่ม Done

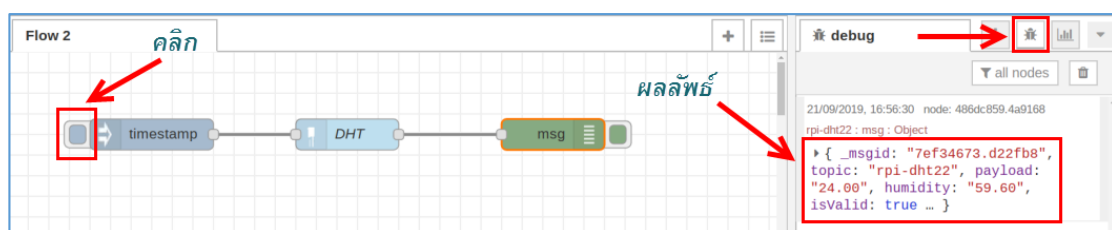
13. ดับเบิลคลิกที่โหนด msg.payload แล้วกำหนดค่าดังรูป แล้วคลิกปุ่ม Done



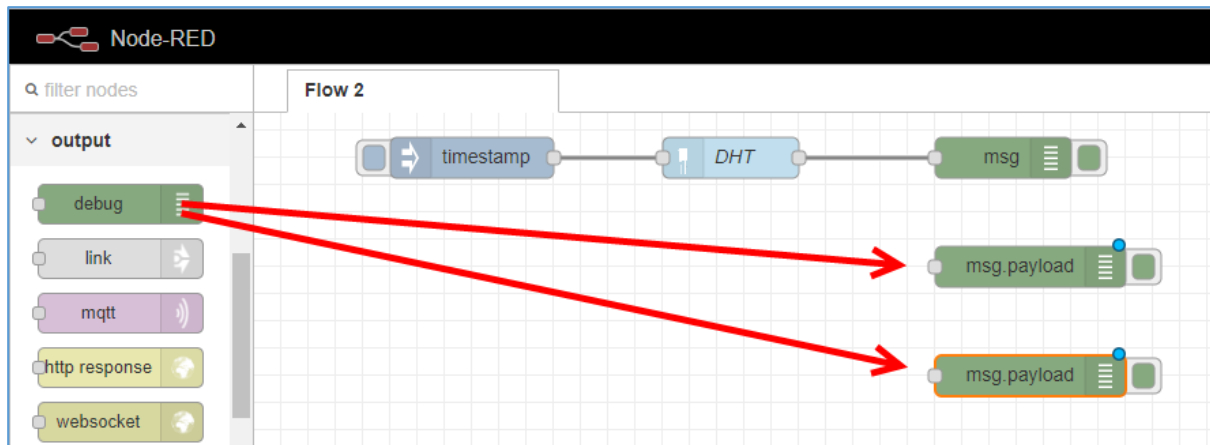
13. คลิกปุ่ม Deploy เพื่อรันโปรแกรม



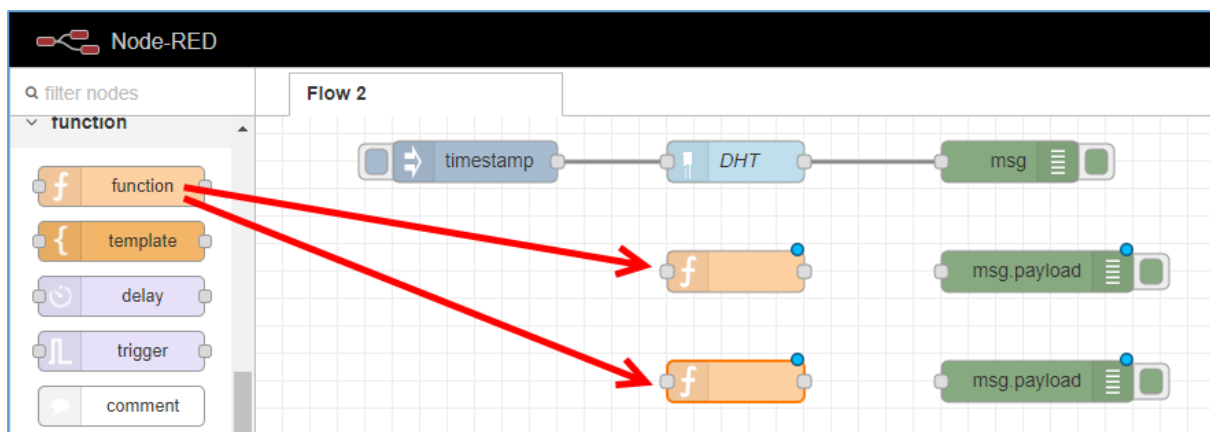
14. ลองคลิกที่ปุ่มด้านหน้าโหนด timestamp จะแสดงค่าตัวเลขออกมา 2 ค่า คือค่าอุณหภูมิและความชื้นสัมพัทธ์ในอากาศ โดยจะแสดงผลออกมาในหน้าต่าง Debug Messages



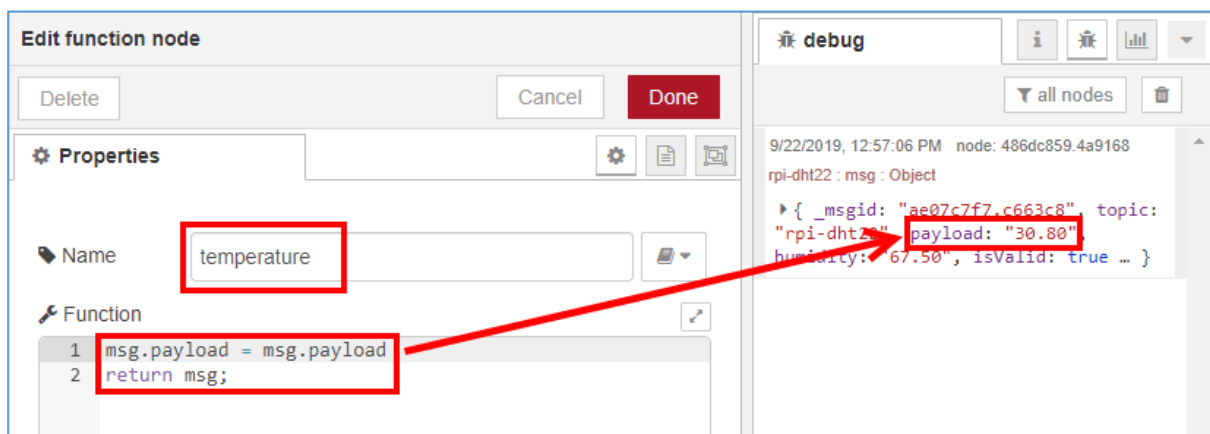
15. ลากโหนด debug ออกมาอีก 2 อัน



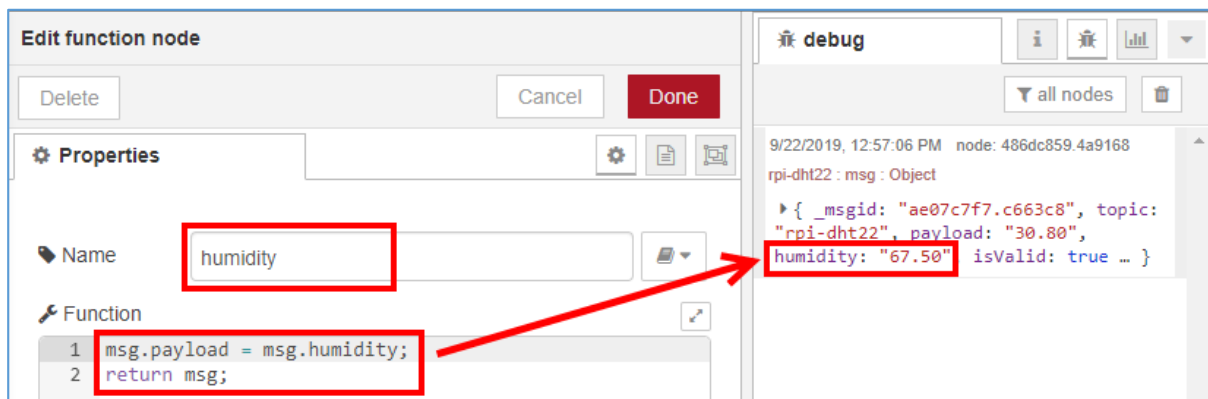
16. ลากโหนด function ออกมา 2 อัน



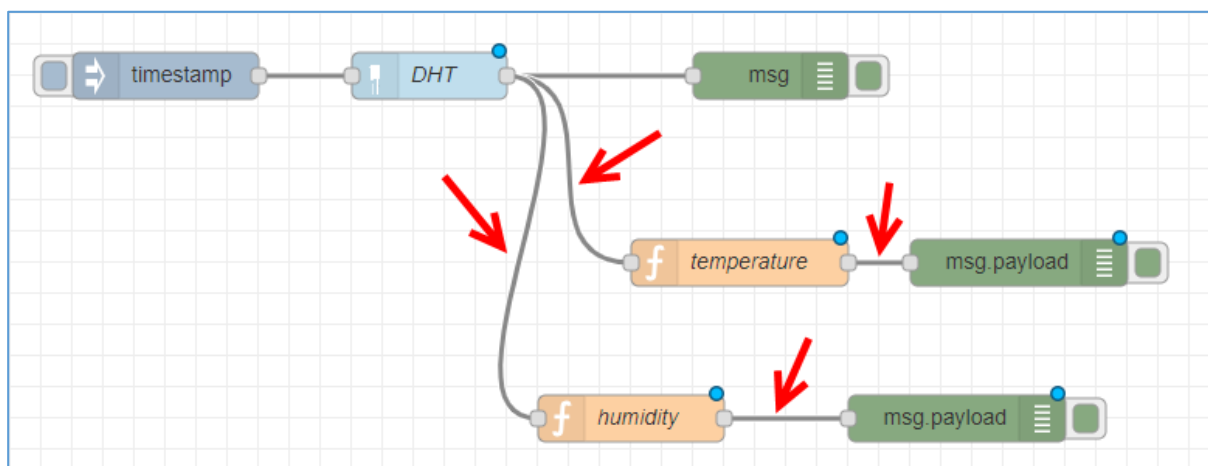
17. ดับเบิ้ลคลิกโหนด function อันแรก แล้วใส่โค้ด JavaScript ดังรูป โดยอ้างอิงค่าอุณหภูมิซึ่งอยู่ในส่วน payload: แล้วคลิกปุ่ม Done



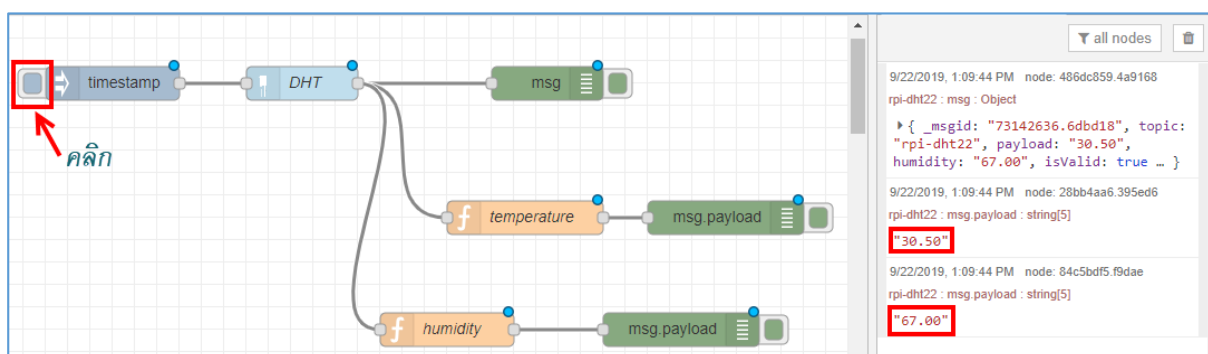
18. ดับเบิลคลิกโหนด function อันที่สอง แล้วใส่โค้ด JavaScript ดังรูป โดยอ้างอิงค่าความชื้นซึ่งอยู่ในส่วน humidity: แล้วคลิกปุ่ม Done



19. ลากเส้นเชื่อมโยงระหว่างโหนด ดังรูป

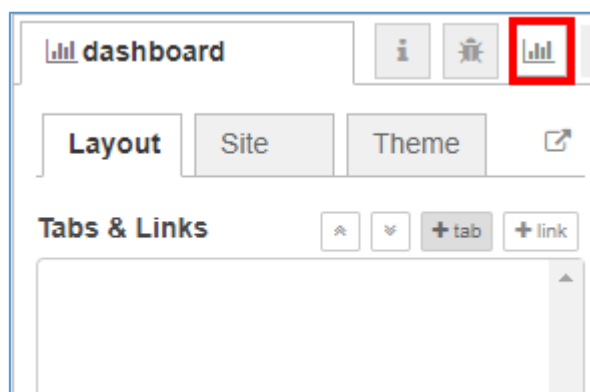


20. คลิกปุ่ม Deploy เพื่อรันโปรแกรม แล้วลองคลิกปุ่มด้านหน้าโหนด timestamp จะเห็นว่าจะได้ค่าออกมา 2 ค่าแยกออกจากกัน คือ ค่าอุณหภูมิและความชื้น

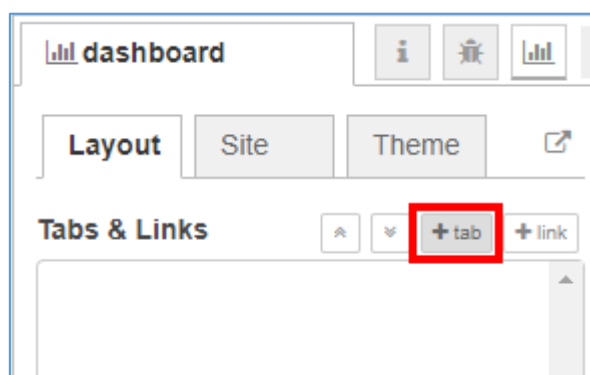


สร้างหน้าจอ UI

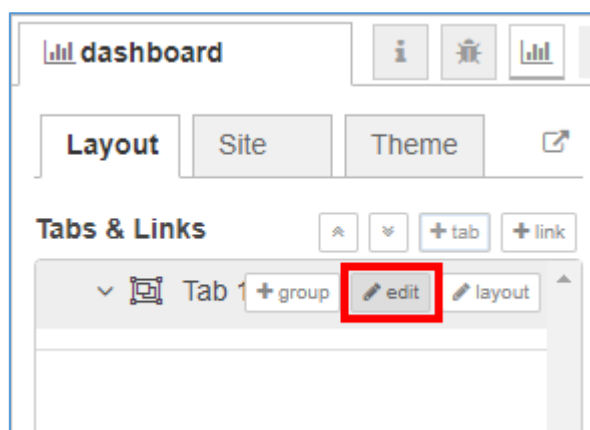
1. คลิกเปิดหน้าต่าง dashboard



2. สร้างแท็บแสดงบนหน้าจอ โดยคลิกที่ + tab



3. เปลี่ยนชื่อแท็บใหม่ โดยคลิกที่ปุ่ม edit ด้านหลัง Tab 1



4. เปลี่ยนชื่อแท็บในช่อง Name แล้วคลิกปุ่ม Update

Edit dashboard tab node

Delete Cancel Update

Properties

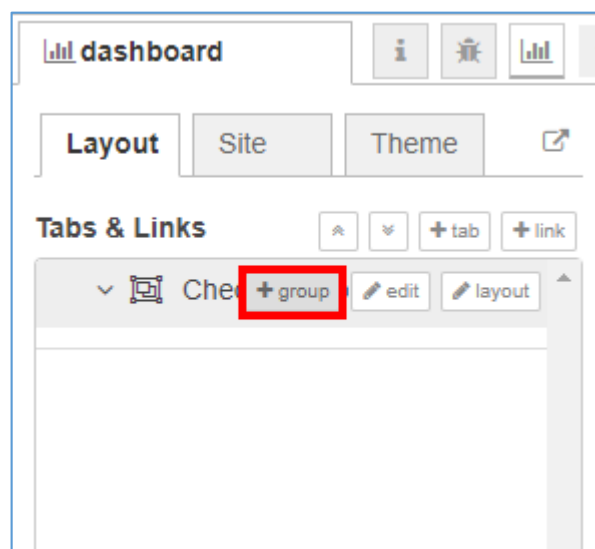
Name Check Environment

Icon dashboard

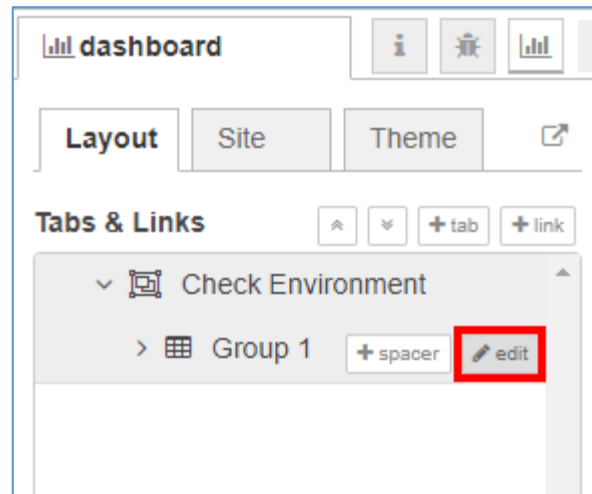
State ☒ Enabled

Nav. Menu ☒ Visible

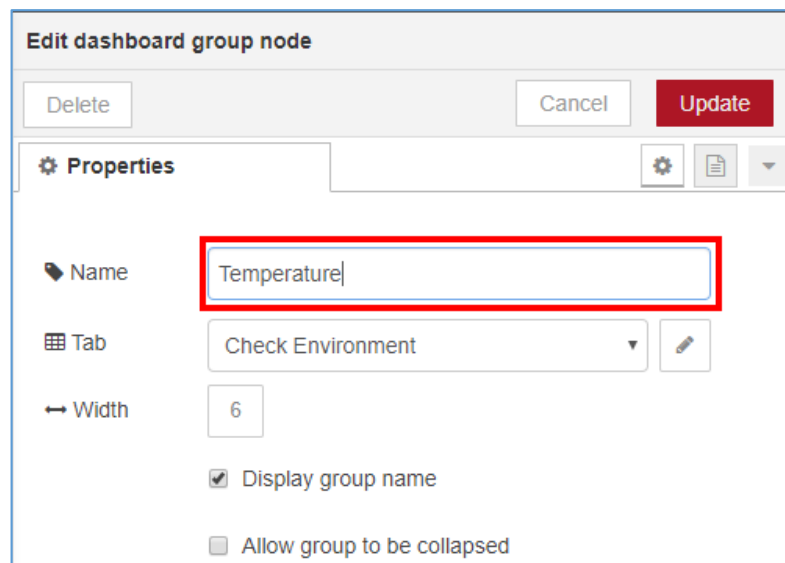
5. สร้าง Group เข้ามาในแท็บ โดยคลิกปุ่ม + Group ด้านหลังชื่อแท็บ



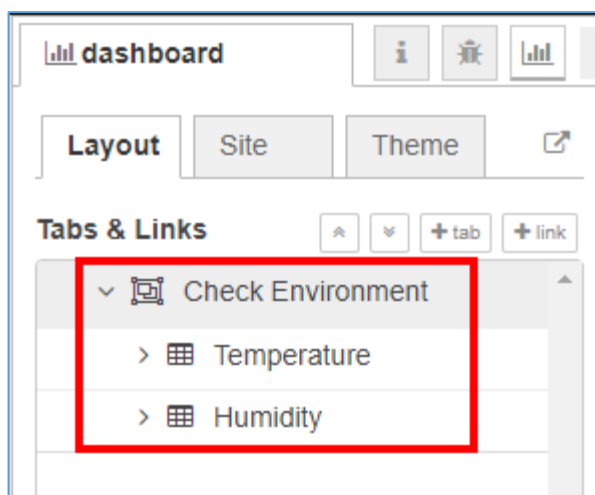
6. แก้ไขชื่อ Group โดยคลิกที่ปุ่ม edit ด้านหลัง Group 1



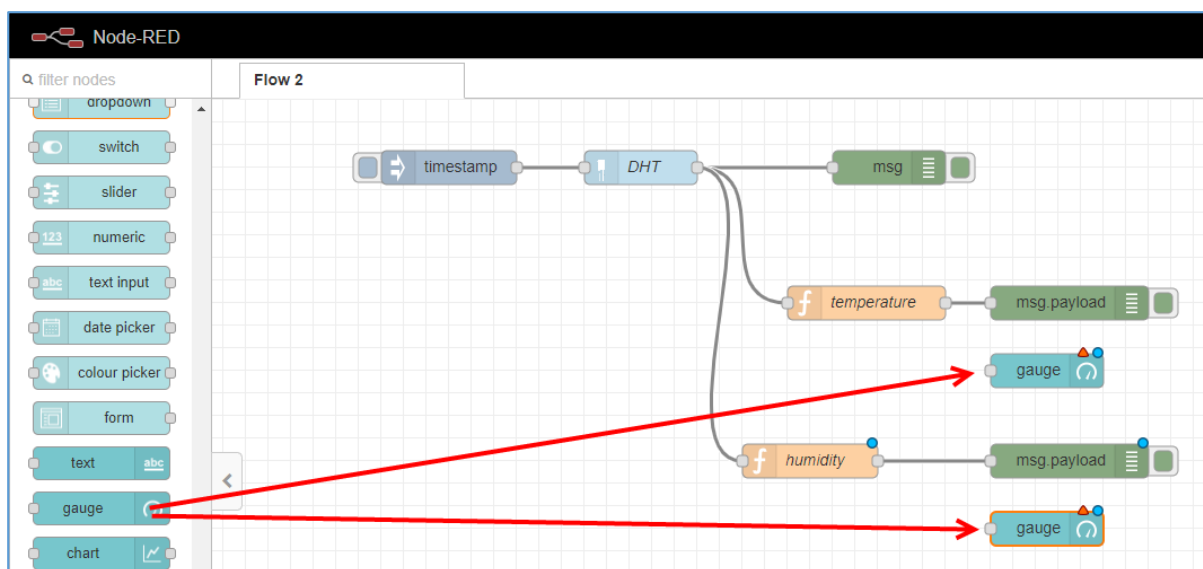
7. เปลี่ยนชื่อ Group ในช่อง Name แล้วคลิกปุ่ม Update



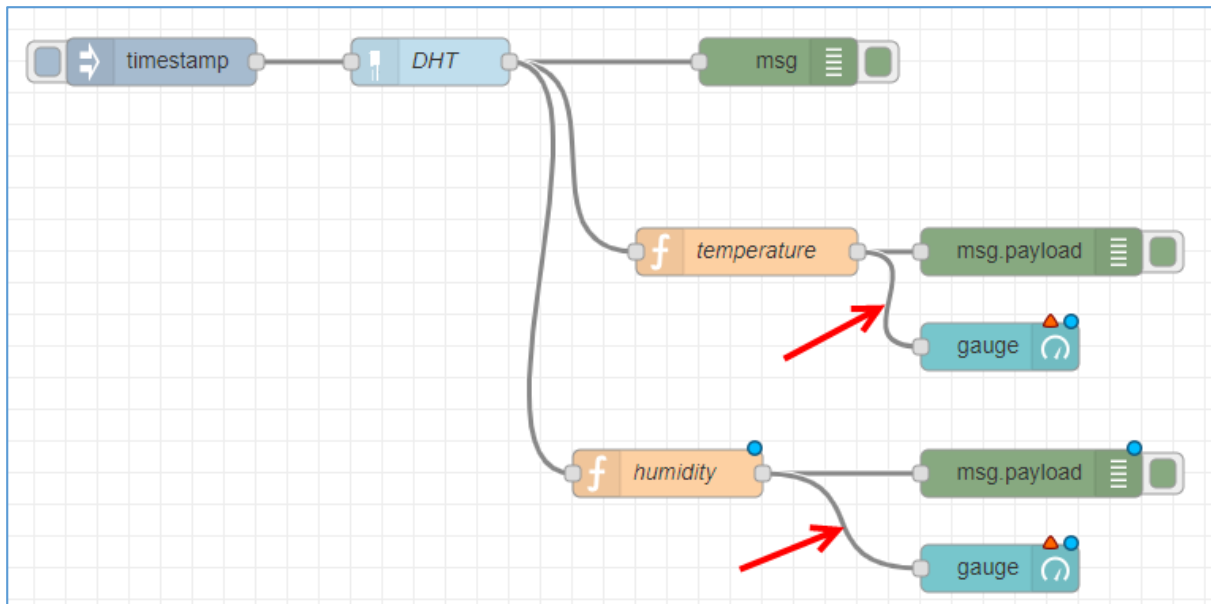
8. สร้าง Group ใหม่ขึ้นมาอีก 1 Group (ทำเหมือนขั้นตอน 5-7) โดยตั้งชื่อ Group ว่า Humidity ก็จะ
ได้แท็บที่มี 2 Group ดังนี้



9. ลากโหนด gauge 2 อันมาวางบน Workspace



10. ลากเส้นเชื่อมต่อโยง ดังรูป



11. ดับเบิลคลิกที่โหนด gauge ที่เชื่อมต่อมาจากฟังก์ชัน temperature โดยกำหนดค่าดังนี้ แล้วคลิกปุ่ม Done

Edit gauge node
Delete
Cancel
Done

Properties

Group
[Check Environment] Temperature

Size
auto

Type
Gauge

Label
gauge

Value format
{{value}}

Units
°C

Range
min -40 max 125

Colour gradient

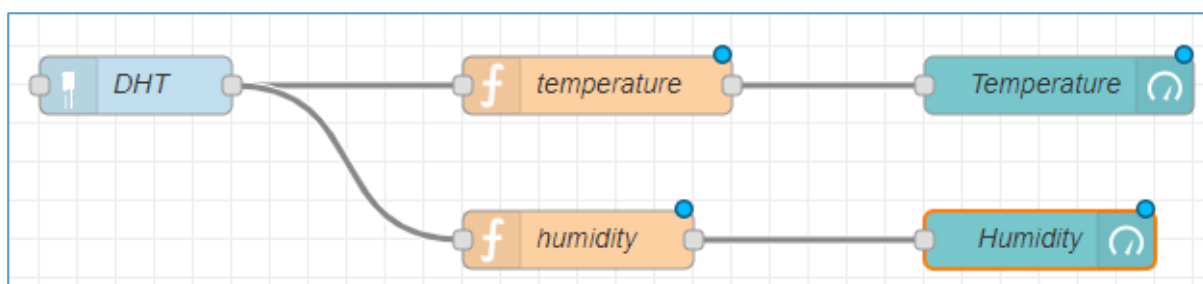
Sectors
-40 ... optional ... optional ... 125

Name
Temperature

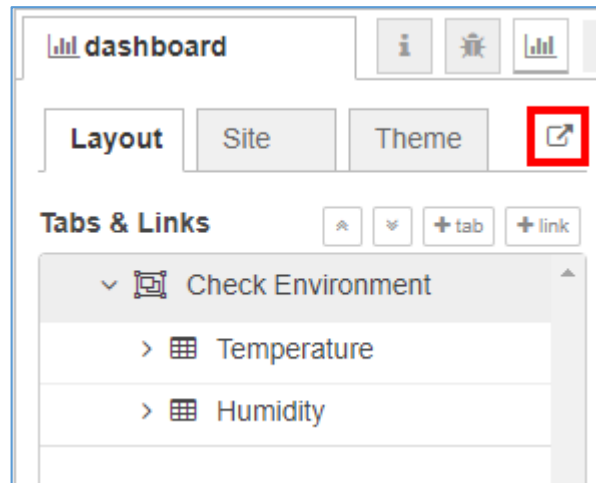
12. ดับเบิลคลิกที่โหนด gauge ที่เชื่อมต่อมาจากฟังก์ชัน humidity โดยกำหนดค่าดังนี้ แล้วคลิกปุ่ม

Done

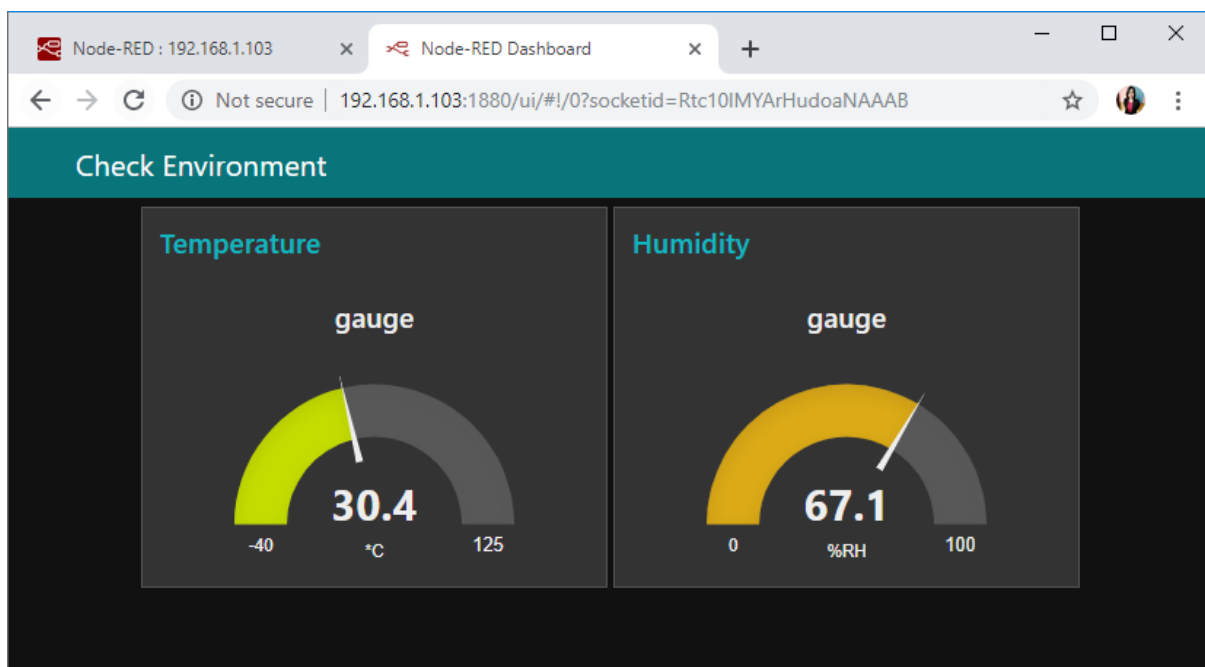
13. ลบโหนด timestamp ออกไป เพื่อให้เซนเซอร์วัดค่าอัตโนมัติ โดยไม่ต้องคลิกปุ่ม และลบโหนดที่ใช้แสดงข้อความทั้ง 3 โหนดออก ให้เหลือแค่โหนดดังรูป



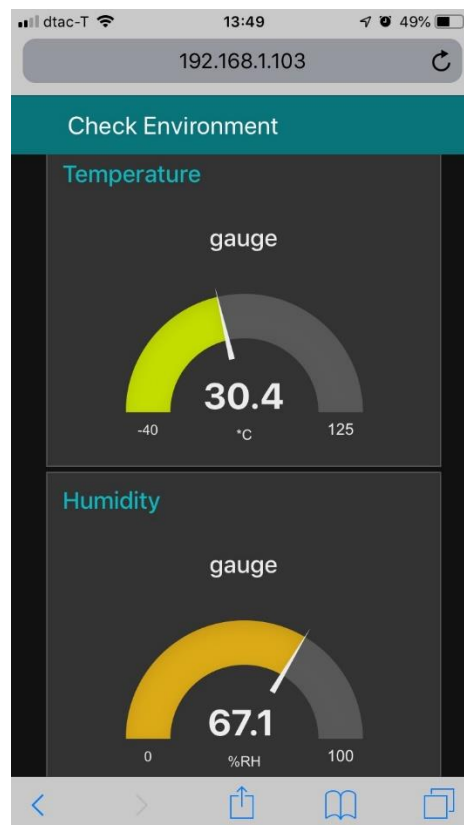
14. คลิกปุ่ม Deploy เพื่อรันโปรแกรม แล้วคลิกปุ่มดังรูป เพื่อเปิดไปยังหน้า UI



14. เซนเซอร์ DHT22 จะวัดค่าอุณหภูมิและความชื้นออกมา โดยแสดงออกมาเป็นเกจวัดค่าตามที่ได้ทำการเลือกโหมด gauge เอาไว้



13. ทดลองใช้มือถือหรือแท็บเล็ตจับสัญญาณ Wi-Fi ตัวเดียวกับ Raspberry Pi หลังจากนั้นเปิด Web Brower ขึ้นมา แล้วใส่ IP Address ของ Raspberry Pi ตามด้วย :1880/ui เช่น 192.168.1.104:1880/ui



6.1 การทำ IOT ผ่านแอปพลิเคชัน Blynk

อุปกรณ์ที่ใช้

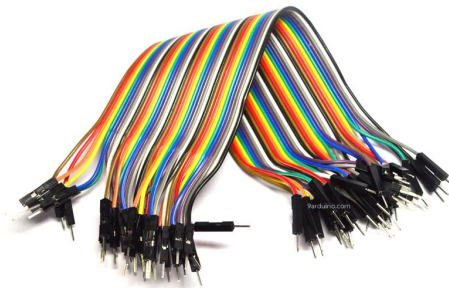
1. บอร์ด Raspberry Pi



2. เซนเซอร์วัดอุณหภูมิและความชื้น (DHT22)



3. สาย Jumper



4. Relay Module



การต่อวงจร

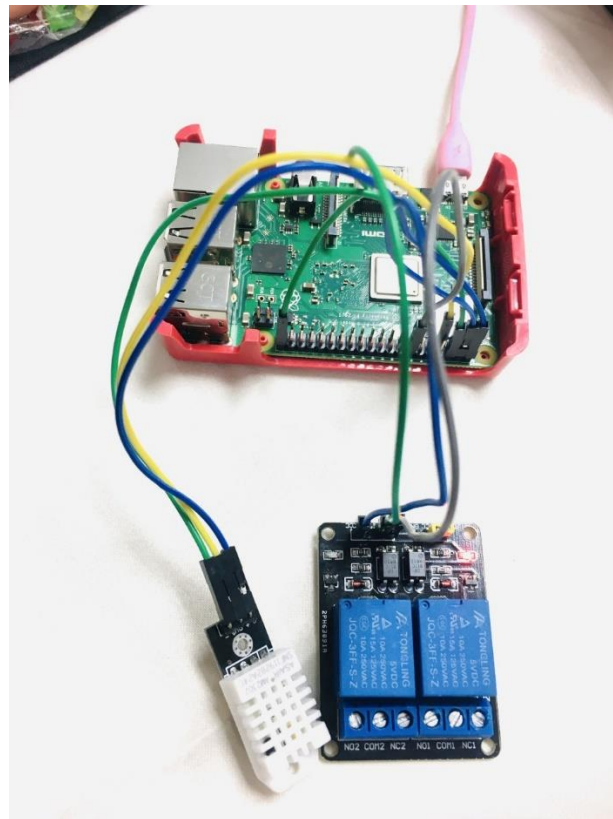
เซนเซอร์ DHT22

1. นำสาย Jumper เสียบเข้ากับขา +, out และ - ของเซนเซอร์ DHT22
2. เสียบสาย Jumper ที่ต่อจากขา + ของเซนเซอร์ DHT22 ไปยังขา PIN2 (5V)
3. เสียบสาย Jumper ที่ต่อจากขา out ของเซนเซอร์ DHT22 ไปยังขา PIN7 (GPIO4)
4. เสียบสาย Jumper ที่ต่อจากขา - ของเซนเซอร์ DHT22 ไปยังขา PIN6 (GND)

Relay Module

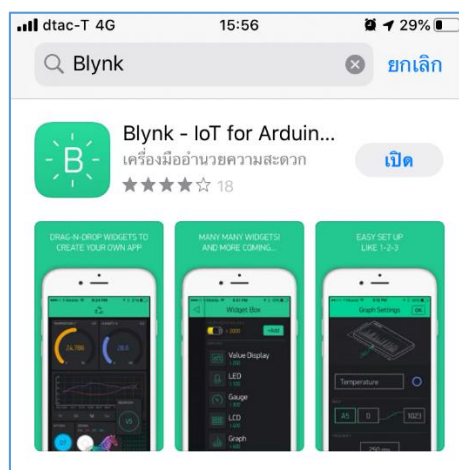
1. นำสาย Jumper เสียบเข้ากับขา VCC, GND และ IN1 ของ Relay Module
2. เสียบสาย Jumper ที่ต่อจากขา VCC ของ Relay Module ไปยังขา PIN4 (5V)

3. เสียบสาย Jumper ที่ต่อจากขา GND ของ Relay Module ไปยังขา PIN14 (GND)
4. เสียบสาย Jumper ที่ต่อจากขา IN1 ของ Relay Module ไปยังขา PIN40 (GPIO21)

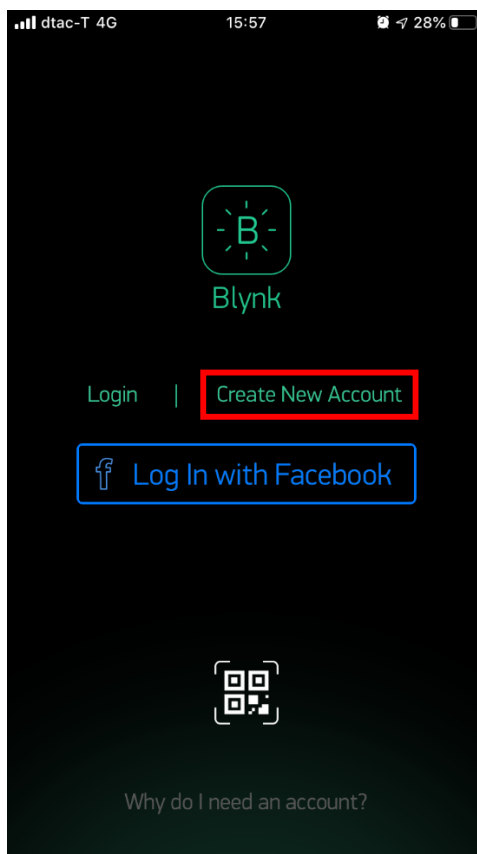


การติดตั้งและใช้งานแอปพลิเคชัน Blynk

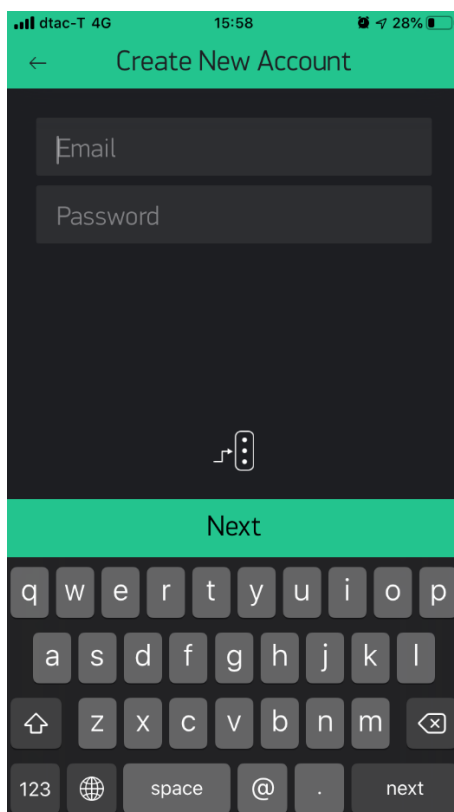
1. ติดตั้งแอปพลิเคชัน Blynk บนมือถือ



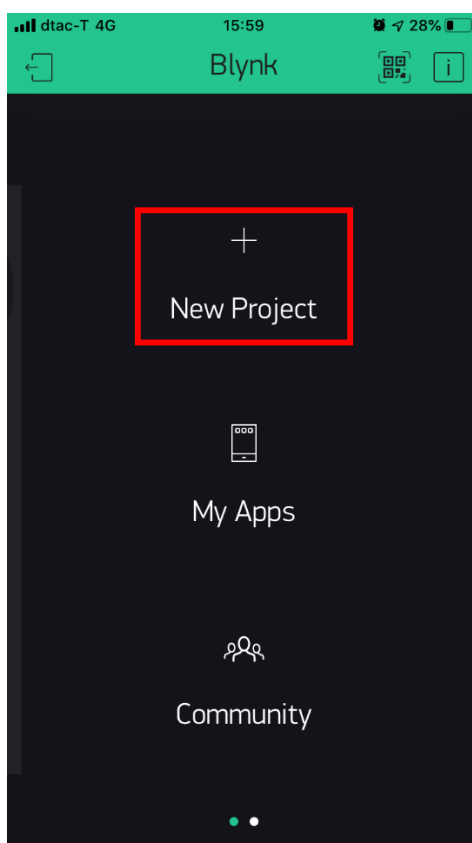
2. เปิดแอปพลิเคชันขึ้นมา หลังจากนั้นให้สร้างบัญชีในการใช้งานขึ้นมาใหม่ โดยแตะที่ Create New Account



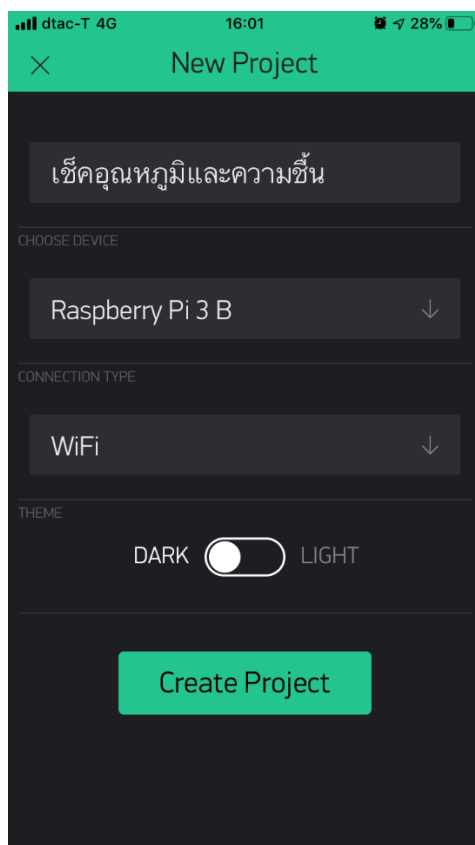
3. กรอกข้อมูลในช่อง Email และ Password หลังจากนั้นให้แตะที่ Next



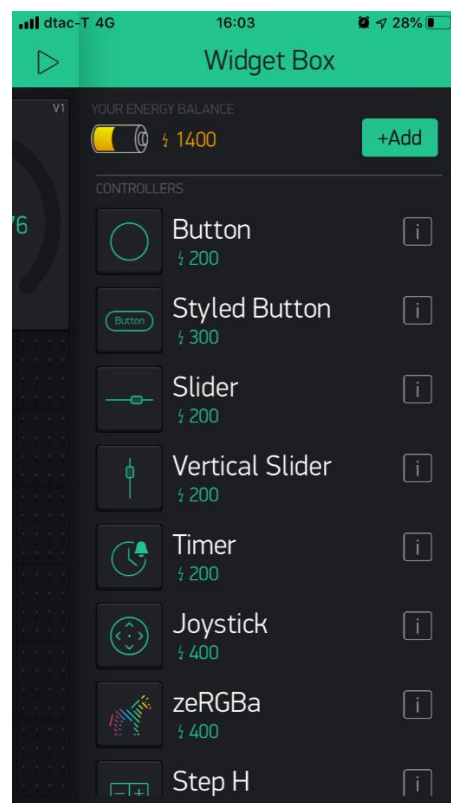
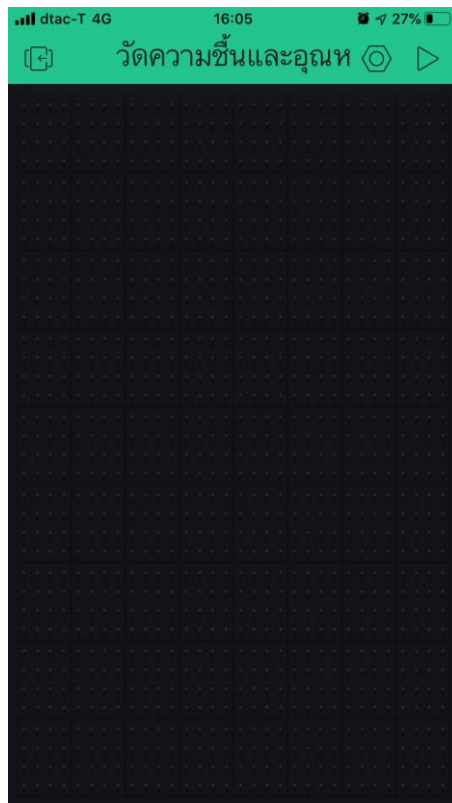
4. สร้างโปรเจกต์ใหม่ โดยแตะที่ New Project



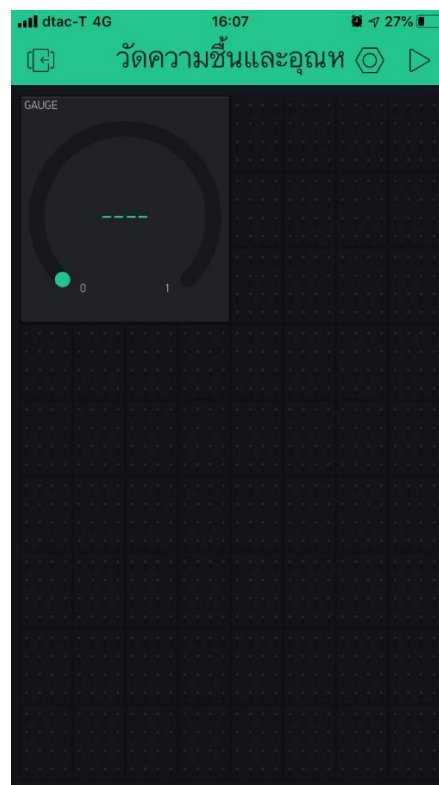
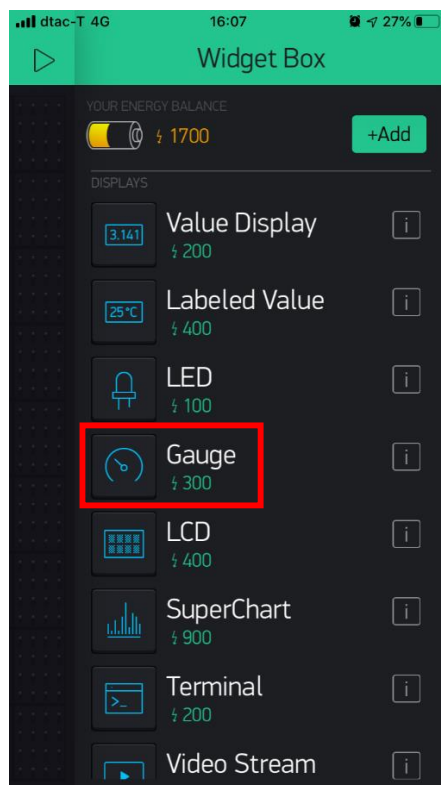
5. กำหนดค่าในโปรเจกต์ที่สร้างขึ้นใหม่ ดังนี้



6. จะเข้าสู่หน้าจอในการออกแบบ ให้แตะที่หน้าจอ 1 ครั้ง จะปรากฏหน้าต่าง Widget Box ขึ้นมา โดยในการใช้งาน Blynk ระบบจะให้ใช้งานฟรี 2000 Energy หากต้องการซื้อเพิ่มให้แตะที่ปุ่ม +Add เพื่อซื้อ Energy เพิ่ม

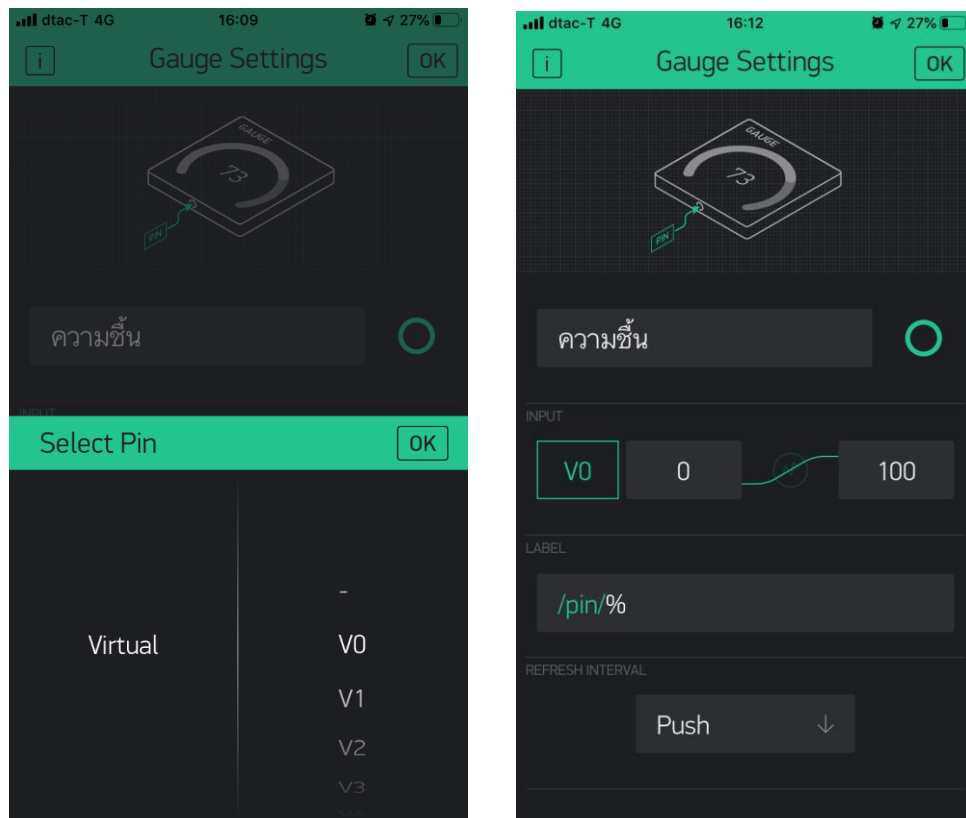


7. แตะที่ Gauge ซึ่งใช้ 300 Energy จะปรากฏหน้าจอ Gauge ขึ้นมา 1 ชิ้น



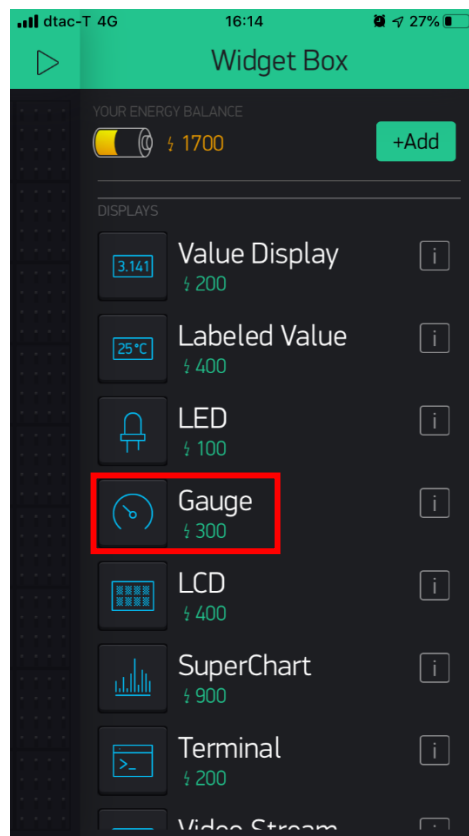
8. แตะที่ Gauge 1 ครั้ง จะปรากฏหน้าจอ Gauge Settings ให้กำหนดค่าดังนี้

- Gauge : กำหนดข้อความบนหน้าจอ Gauge เป็น “ความชื้น”
- PIN : เลือกเป็น V0
- กำหนดช่วงข้อมูลเป็น 0-100
- ช่อง LABEL : กำหนดหน่วยให้เป็น “%”



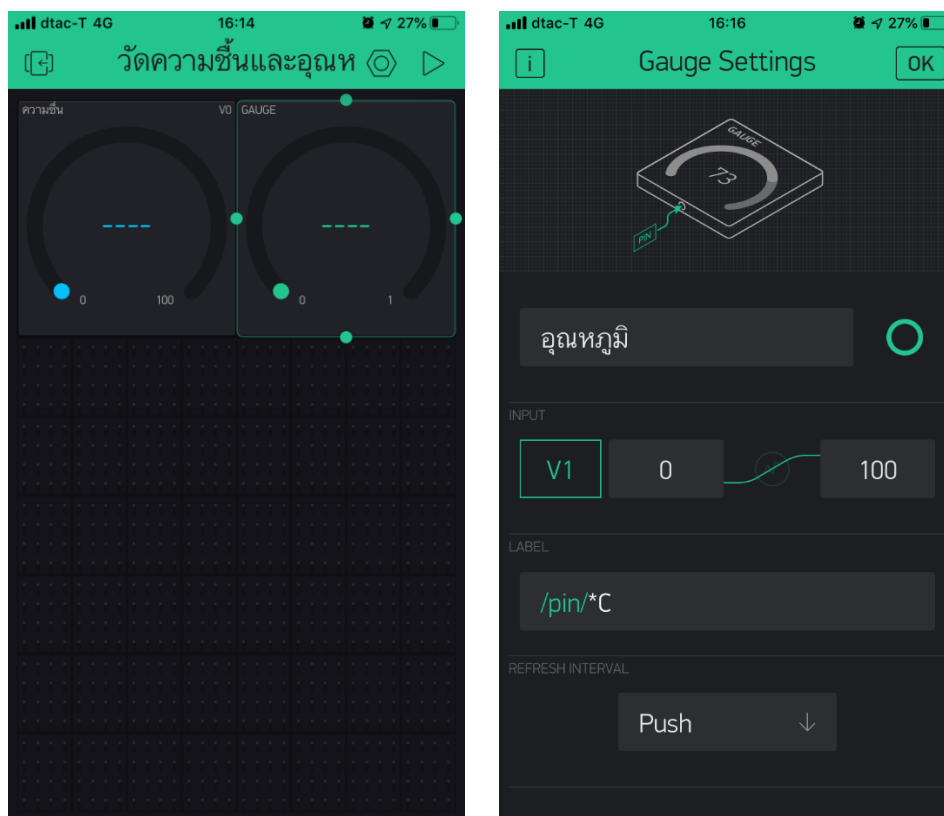
9. เมื่อเสร็จแล้วให้แตะที่ OK

10. แตะที่หน้าจอออกแบบ 1 ครั้ง หลังจากนั้นให้แตะที่ Gauge อีกครั้งเพื่อสร้างหน้าจอ Gauge ขึ้นมาใหม่



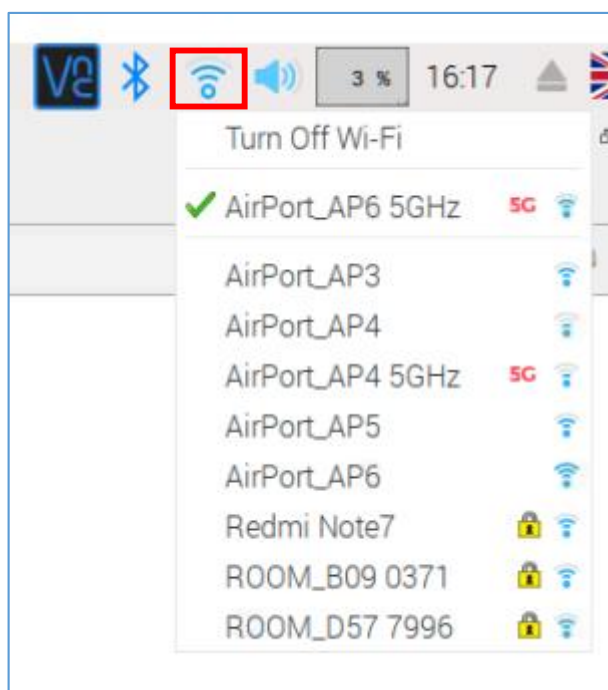
11. และที่ Gauge ที่สร้างขึ้นใหม่อีก 1 ครั้ง จะปรากฏหน้าจอ Gauge Settings ให้กำหนดค่าดังนี้

- Gauge : กำหนดข้อความบนหน้าจอ Gauge เป็น “อุณหภูมิ”
- PIN : เลือกเป็น V1
- กำหนดช่วงข้อมูลเป็น 0-100
- ช่อง LABEL : กำหนดหน่วยให้เป็น “*C”

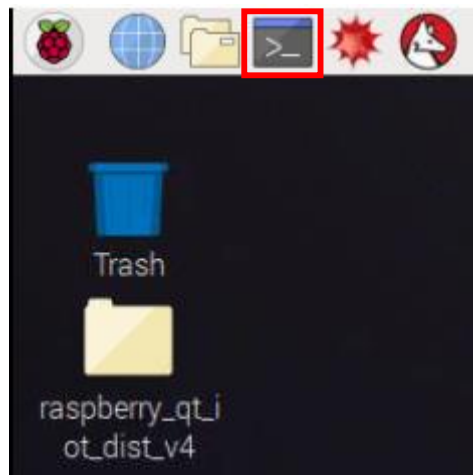


เขียนโปรแกรมคำสั่ง

1. มาที่หน้าจอ Desktop ของ Raspberry Pi ทำการเชื่อมต่อ Wi-Fi



2. เปิดโปรแกรม Terminal ขึ้นมา แล้วตรวจสอบเวอร์ชันของไพทอน 3 โดยพิมพ์คำสั่ง `python3 --version`

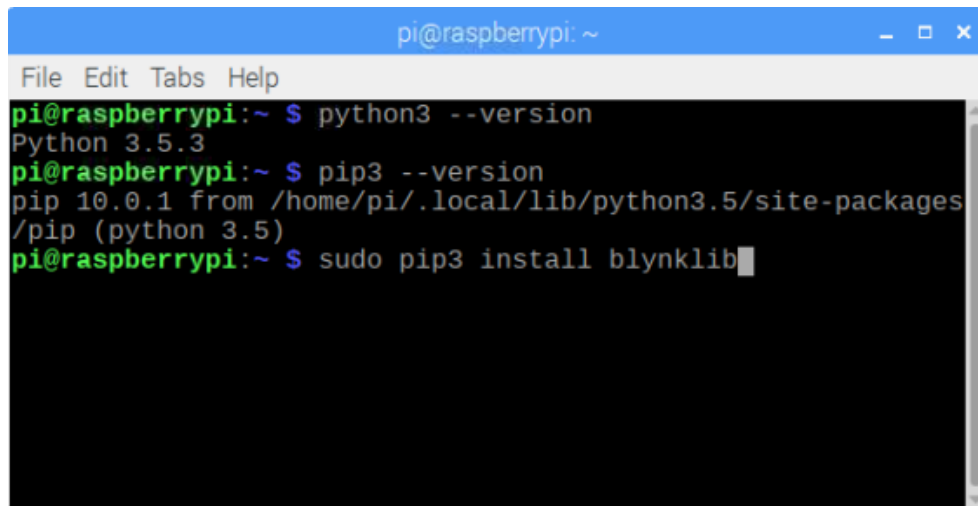


```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ python3 --version
```

3. ตรวจสอบเช็คเวอร์ชันของ pip โดยพิมพ์คำสั่ง `pip3 --version`

```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ python3 --version  
Python 3.5.3  
pi@raspberrypi:~ $ pip3 --version
```

4. ติดตั้ง Blynk Python Library โดยพิมพ์คำสั่ง `sudo pip3 install blynklib`

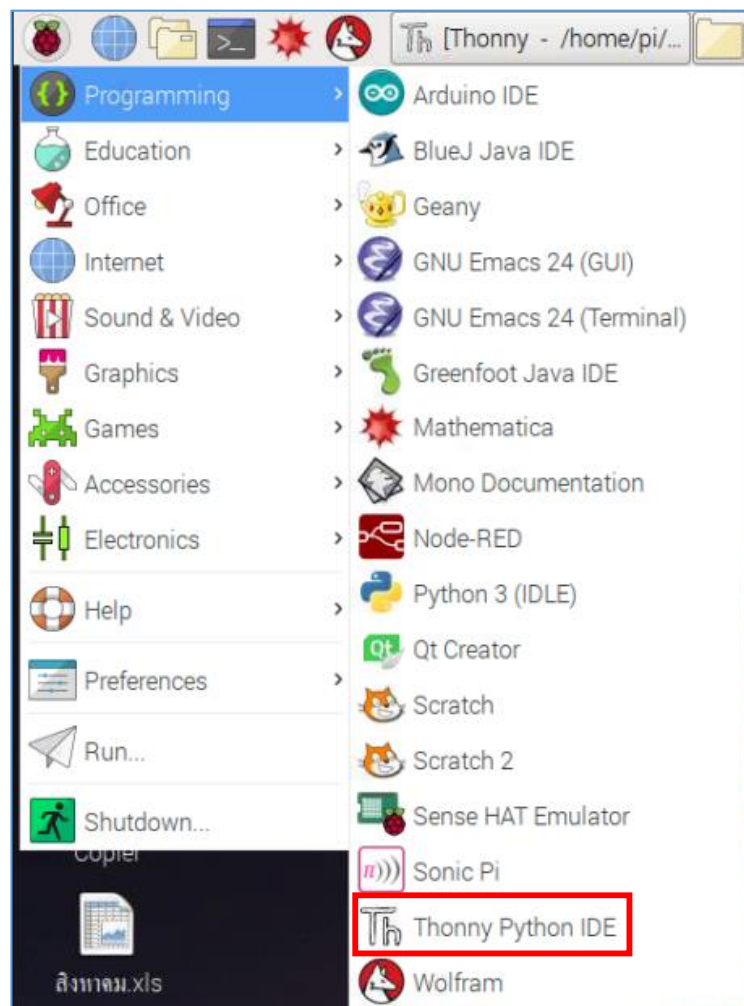


```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ python3 --version
Python 3.5.3
pi@raspberrypi:~ $ pip3 --version
pip 10.0.1 from /home/pi/.local/lib/python3.5/site-packages/pip (python 3.5)
pi@raspberrypi:~ $ sudo pip3 install blynklib

```

5. เปิดโปรแกรม Thonny Python IDE ขึ้นมาเพื่อเขียนโปรแกรม



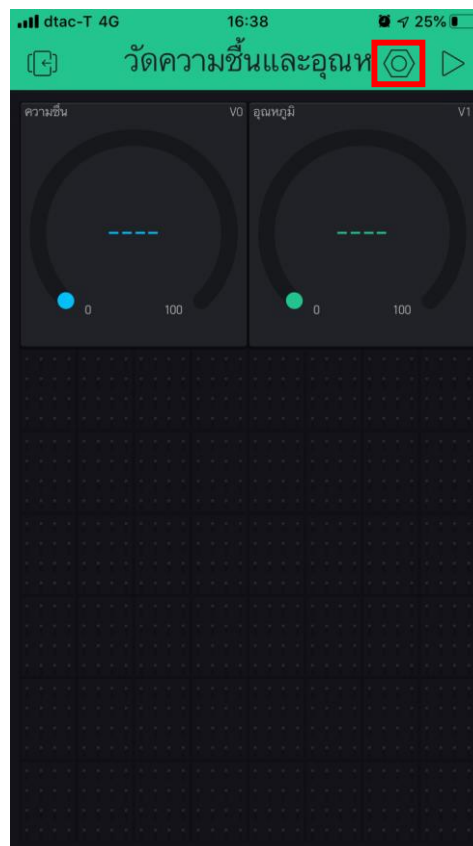
6. เขียนโค้ดคำสั่งดังนี้

```

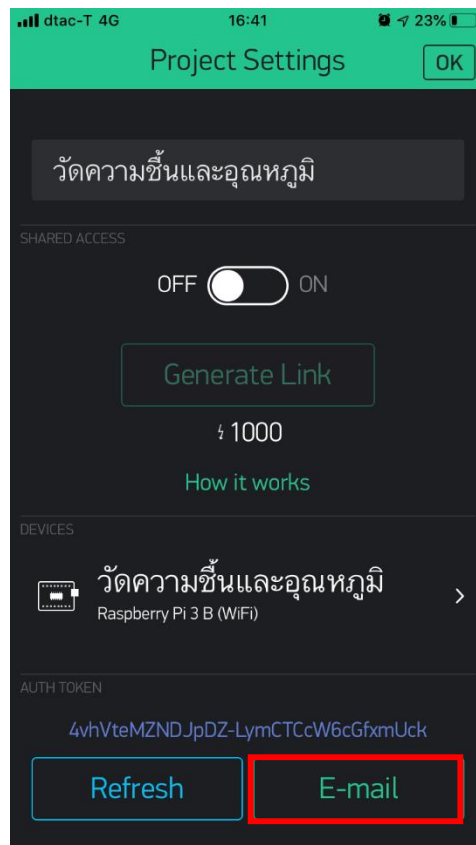
8_DHT22.py *
1  import blynklib
2  import Adafruit_DHT
3  import time
4
5  BLYNK_AUTH = '<YourAuthToken>'
6  blynk = blynklib.Blynk(BLYNK_AUTH,server='blynk-cloud.com',port=8442, heartbeat=15)
7
8  while True :
9      blynk.run()
10     humidity,temperature = Adafruit_DHT.read_retry(22,4)
11     print("Temperature = %.2f    humidity = %.2f" %(temperature,humidity))
12     blynk.virtual_write(0,humidity)
13     blynk.virtual_write(1,temperature)
14     time.sleep(3)
15

```

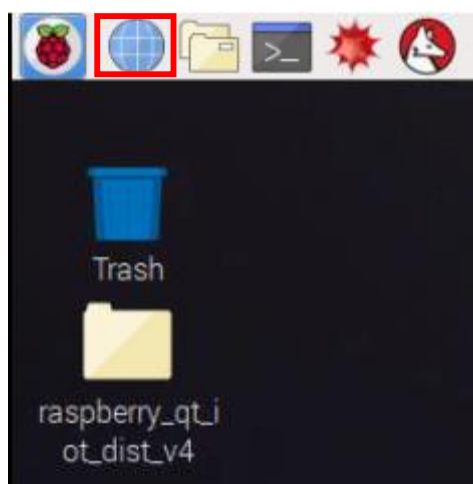
7. ในช่อง <YourAuthToken> จะต้องนำ AUTH TOKEN ของโปรเจกต์ของตนเองในแอปพลิเคชัน Blynk มาใส่แทน ซึ่งจะเป็นรหัสที่ไม่ซ้ำกัน โดยให้แต่ละเครื่องหมายหกล้มด้านบนขวาของหน้าจอโปรเจกต์



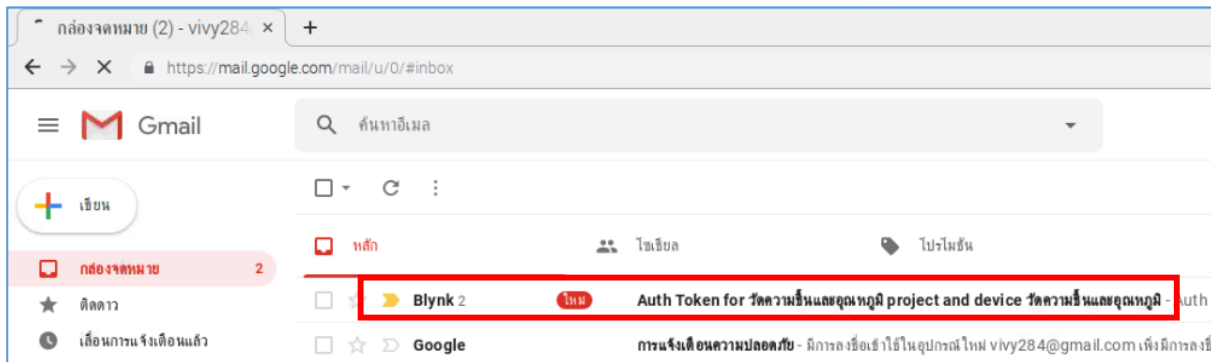
8. แตะเมนู E-Mail เพื่อส่งรหัสในช่อง AUTH TOKEN ด้านล่างจอไปยังอีเมลที่ได้ลงทะเบียนบัญชีไว้



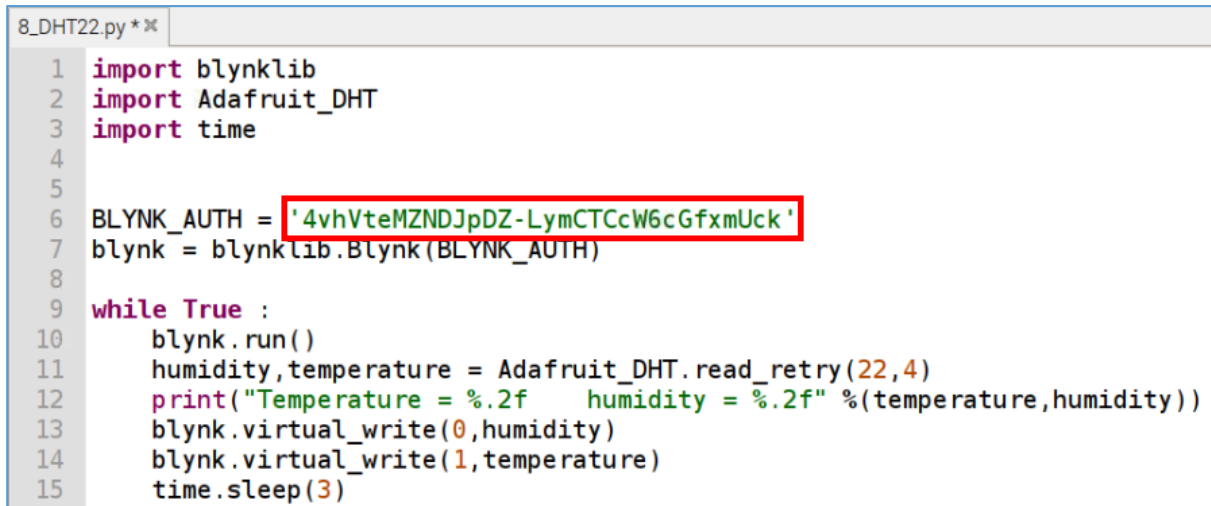
9. กลับมาที่หน้าจอ Desktop Raspberry Pi แล้วเปิด Web Browser ขึ้นมาเพื่อเข้าอีเมล



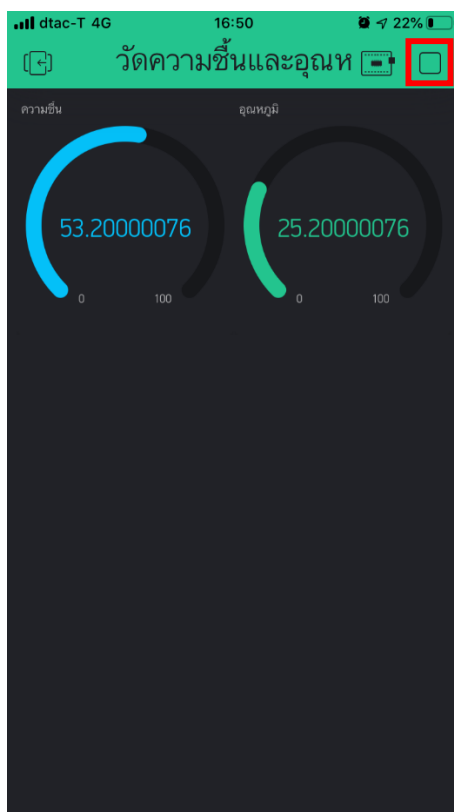
10. เปิดอีเมลที่ได้ลงทะเบียนบัญชี Blynk เอาไว้ จะมีอีเมลของ Blynk เข้ามา



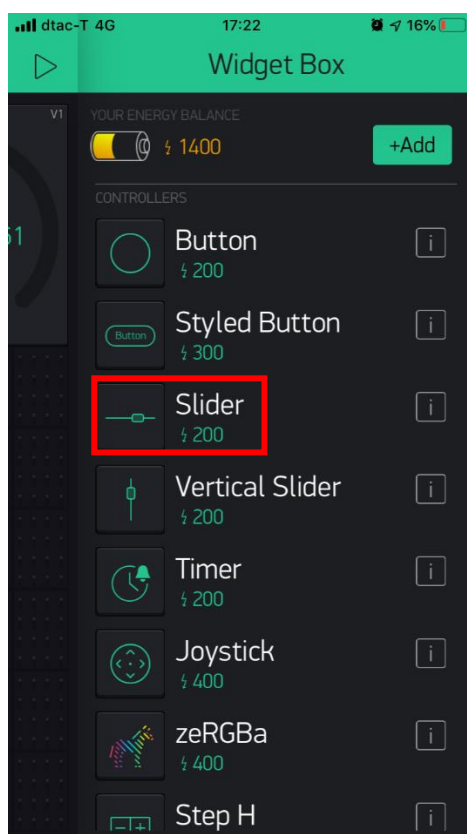
11. คลิกเปิดอีเมล แล้วคัดลอกรหัส Auth Token มาแทนที่คำสั่ง <YourAuthToken> ในโค้ดคำสั่งที่เขียน



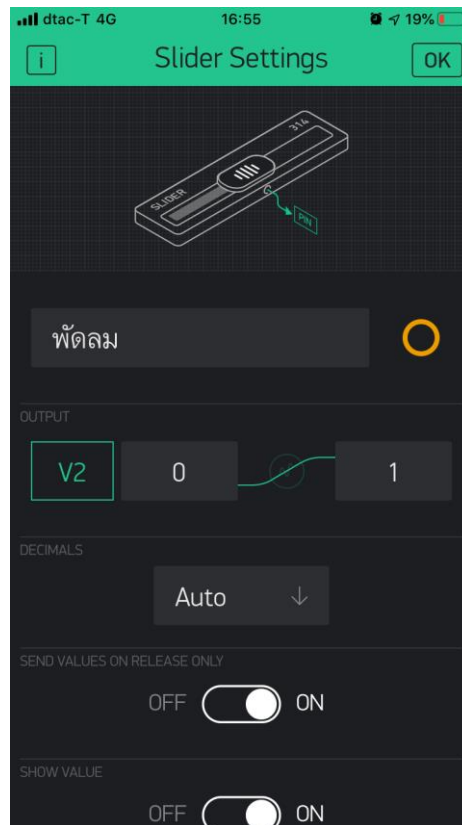
12. บันทึกไฟล์และรันโปรแกรม โดยคลิกที่ปุ่ม Run จะเห็นว่าในหน้าต่าง Shell ด้านล่างจะทำการเชื่อมต่อกับแอปพลิเคชัน Blynk



15. เพิ่ม Slider ขึ้นมา 1 อัน เพื่อใช้แสดงสถานะการเปิด-ปิดพัดลม



16. แตะที่ Slider แล้วตั้งค่าดังนี้



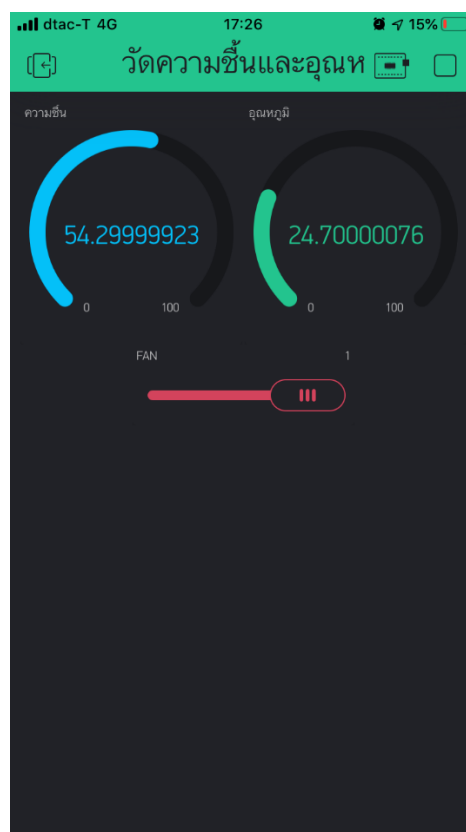
17. กลับมาที่โปรแกรม Thonny Python IDE เพื่อกำหนดเงื่อนไขว่า ถ้าอุณหภูมิสูงกว่า 25 องศาเซลเซียส ให้ Relay ทำงาน โดยการปล่อยกระแสไฟฟ้าไปยังพัลลุม ทำให้พัลลุมหมุน และส่งค่า 1 ไปยังแถบ Slider นอกนั้นให้พัลลุมหยุดหมุน และส่งค่า 0 ไปยังแถบ Slider โดยเขียนคำสั่งเพิ่มเติมดังนี้

```

8_DHT22.py *
1 import blynklib
2 import Adafruit_DHT
3 import time
4 import RPi.GPIO as GPIO
5
6 GPIO.setmode(GPIO.BOARD)
7 GPIO.setup(40,GPIO.OUT)
8
9
10 BLYNK_AUTH = '4vhVteMZNDJpDZ-LymCTCcW6cGfxmUck'
11 blynk = blynklib.Blynk(BLYNK_AUTH)
12
13 while True :
14     blynk.run()
15     humidity,temperature = Adafruit_DHT.read_retry(22,4)
16     print("Temperature = %.2f    humidity = %.2f" %(temperature,humidity))
17     blynk.virtual_write(0,humidity)
18     blynk.virtual_write(1,temperature)
19
20     if temperature > 24:
21         GPIO.output(40,0)
22         blynk.virtual_write(2,1)
23     else:
24         GPIO.output(40,1)
25         blynk.virtual_write(2,0)
26     time.sleep(3)

```

18. คลิกปุ่ม Run โปรแกรม และแตะปุ่ม Play ที่แอปพลิเคชัน Blynk จะเห็นว่าเมื่ออุณหภูมิมากกว่า 25 องศาเซลเซียส Relay จะทำงาน แถบ Slider จะปรับค่าเป็น 1 เมื่ออุณหภูมिन้อยกว่า 25 องศาเซลเซียส Relay จะหยุดทำงาน แถบ Slider จะปรับค่าเป็น 0



หน่วยการเรียนรู้ที่ ๗ สร้างหุ่นยนต์เคลื่อนที่ด้วยบอร์ด Raspberry Pi

กลุ่มสาระการเรียนรู้วิทยาศาสตร์

สาระที่ ๔ เทคโนโลยี

มาตรฐาน ว ๔.๒ เข้าใจและใช้แนวคิดเชิงคำนวณในการแก้ปัญหาที่พบในชีวิตจริงอย่างเป็นขั้นตอนและเป็นระบบ ใช้เทคโนโลยีสารสนเทศและการสื่อสารในการเรียนรู้ การทำงาน และการแก้ปัญหาได้อย่างมีประสิทธิภาพ รู้เท่าทัน และมีจริยธรรม

ตัวชี้วัด

- ว ๔.๒ ม.๑/๒ ออกแบบและเขียนโปรแกรมอย่างง่ายเพื่อแก้ปัญหาทางคณิตศาสตร์หรือวิทยาศาสตร์
- ว ๔.๒ ม.๒/๒ ออกแบบและเขียนโปรแกรมที่ใช้ตรรกะและฟังก์ชันในการแก้ปัญหา
- ว ๔.๒ ม.๓/๑ พัฒนาแอปพลิเคชันที่มีการบูรณาการกับวิชาอื่นอย่างสร้างสรรค์

สาระสำคัญ

- ๑. เรียนรู้หลักการทำงานของมอเตอร์
- ๒. เรียนรู้การประกอบหุ่นยนต์ในรูปแบบต่างๆ
- ๓. เรียนรู้การเขียนโปรแกรมเพื่อควบคุมการทำงานของหุ่นยนต์

สาระการเรียนรู้

ความรู้

- ความรู้เกี่ยวกับมอเตอร์
- ความรู้การประกอบชุดหุ่นยนต์ในแต่ละรูปแบบ
- ความรู้เกี่ยวกับการเขียนโปรแกรมเพื่อควบคุมการทำงานของหุ่นยนต์

ทักษะ/กระบวนการ

- ปฏิบัติตามตัวอย่างที่แสดงให้เห็นทีละขั้นตอนจากขั้นพื้นฐานไปสู่ขั้นที่ซับซ้อนขึ้น
- ฝึกปฏิบัติด้วยตนเองจนเกิดความชำนาญ หรือทำได้โดยอัตโนมัติ ซึ่งอาจเป็นงานชิ้นเดิมหรืองาน

ที่คิดขั้นใหม่

- ใช้กระบวนการคิดศาสตร์ด้านการคิดคำนวณและการแก้ปัญหา
- กระบวนการกลุ่ม มุ่งให้ผู้เรียนทำงานร่วมกันเป็นทีม

คุณลักษณะที่พึงประสงค์

- รักชาติ ศาสน์ กษัตริย์
- ซื่อสัตย์
- มีวินัย
- ใฝ่เรียนรู้
- อยู่อย่างพอเพียง
- มุ่งมั่นในการทำงาน
- รักความเป็นไทย
- มีจิตสาธารณะ

หน่วยการเรียนรู้ที่ 7 สร้างหุ่นยนต์เคลื่อนที่ด้วยบอร์ด Raspberry Pi

1. รถหุ่นยนต์ Raspberry Pi คอนโทรลด้วยมือถือ

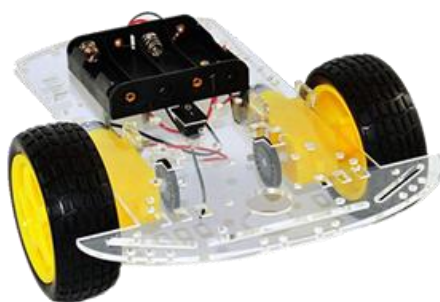
สิ่งที่ต้องมี

1. สัญญาณ Wi-Fi โดยบอร์ด Raspberry Pi และมือถือจะต้องทำการเชื่อมต่อ Wi-Fi ตัวเดียวกัน ซึ่งภายในตัวบอร์ดนั้นจะมี Wi-Fi อยู่ในตัวบอร์ดแล้ว

2. บอร์ด Raspberry Pi ใช้บอร์ดที่มี GPIO 40 ขา



3. ชุดประกอบรถหุ่นยนต์ ซึ่งประกอบด้วยตัวรถ ล้อ มอเตอร์ กระบะใส่ถ่าน



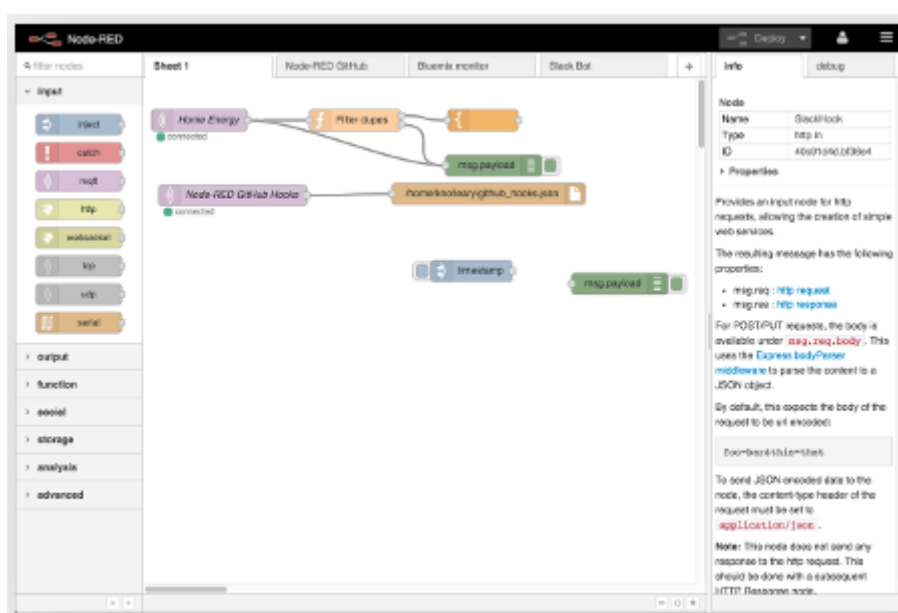
4. บอร์ดขับเคลื่อนมอเตอร์ ในที่นี้จะใช้บอร์ด L298N



5. **Power bank** ที่มีขนาดความจุ 2,200 mAh. ขึ้นไป เพื่อจ่ายไฟให้แก่บอร์ด Raspberry Pi เพื่อให้หุ่นยนต์เคลื่อนที่ได้อย่างอิสระ

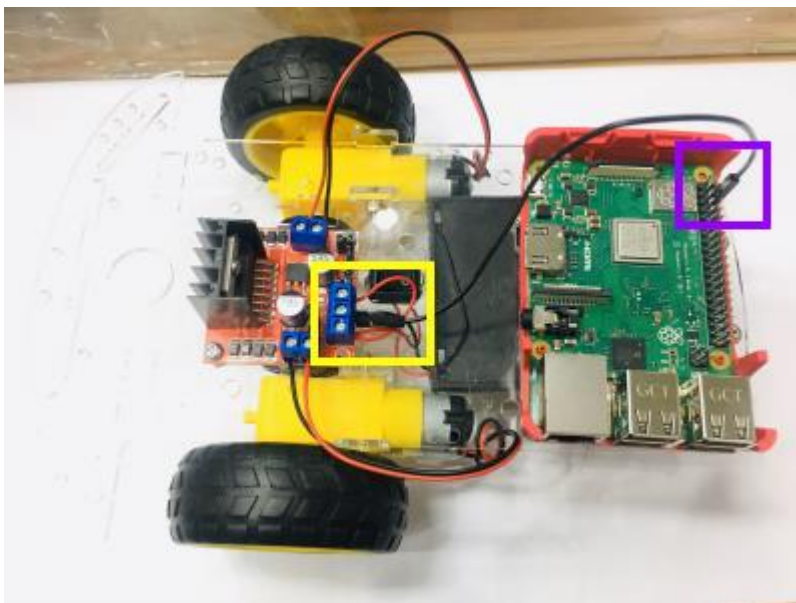


6. โปรแกรม **Node-RED** ใช้ในการสร้างปุ่มคำสั่งควบคุมการเคลื่อนที่ของหุ่นยนต์ ไม่ว่าจะเป็นเดินหน้า ถอยหลัง เลี้ยวซ้าย เลี้ยวขวา และหยุดเดิน โดยจะต้องเพิ่มปุ่มปิดลงในโปรแกรมเพื่อให้สามารถปิด Raspberry Pi จากกระยะไกลได้

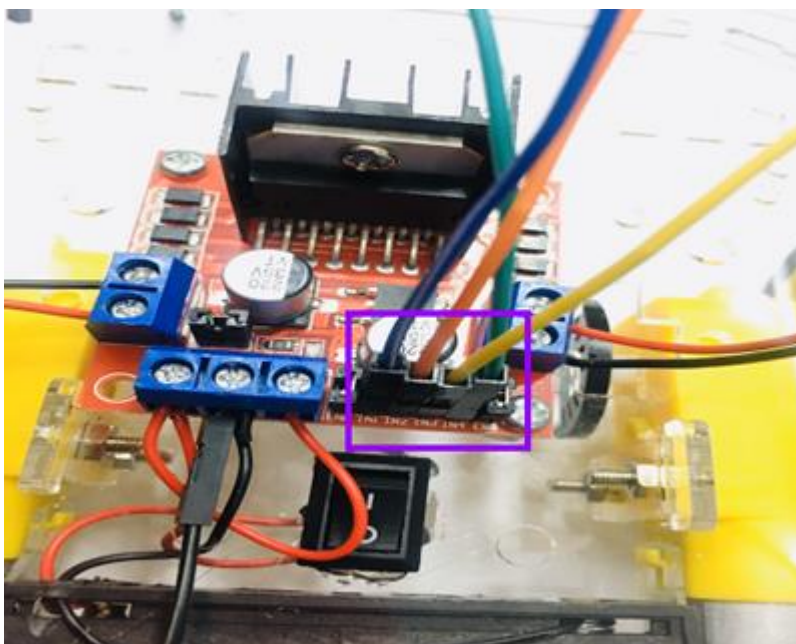


ขั้นตอนการเชื่อมต่อบอร์ด L298N กับบอร์ด Raspberry Pi

1. ใช้ไขควงคลายน็อตในช่อง GROUND (ช่องกลาง) ของบอร์ด L298N ให้เปิดออก แล้วนำขาเสียบฝั่งตัวผู้ของสาย Jumper แบบตัวผู้-ตัวเมีย เสียบเข้าไปในช่อง แล้วขันน็อตให้แน่น ส่วนปลายสายฝั่งตัวเมียของสาย Jumper ให้เสียบเข้าไปที่ขา PIN6 (GND) ของขา GPIO ที่บอร์ด Raspberry Pi



2. ใช้สาย Jumper ตัวเมีย-ตัวเมีย 4 เส้น เสียบเข้าไปในช่อง IN1 – IN4 ของบอร์ด L298N



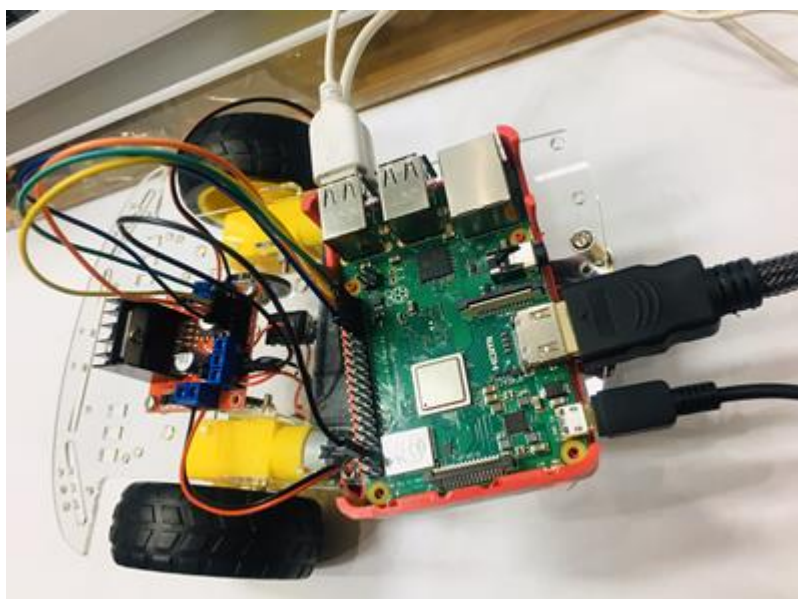
3. ปลายสาย Jumper อีกด้านเสียบเข้ากับขา GPIO ที่เป็นขา Input/Output ของบอร์ด Raspberry Pi (ในที่นี้จะเสียบเข้ากับขา GPIO ที่บอร์ด Raspberry Pi ตามลำดับคือ

- ขา IN1 ของ L298N ต่อเข้ากับขา PIN31 ของบอร์ด Raspberry Pi
- ขา IN2 ของ L298N ต่อเข้ากับขา PIN33 ของบอร์ด Raspberry Pi

- ขา IN3 ของ L298N ต่อเข้ากับขา PIN35 ของบอร์ด Raspberry Pi
- ขา IN4 ของ L298N ต่อเข้ากับขา PIN37 ของบอร์ด Raspberry Pi



4. นำบอร์ด Raspberry Pi มาเสียบสาย HDMI ต่อเข้ากับจอภาพหรือเครื่องคอมพิวเตอร์ที่มีพอร์ต HDMI IN และเชื่อมต่อเมาส์และคีย์บอร์ดเข้ากับบอร์ด Raspberry Pi หลังจากนั้นเชื่อมต่อสาย Adapter เข้ากับ Power Bank เชื่อมต่อกับบอร์ด เพื่อจ่ายไฟแก่บอร์ด Raspberry Pi จะปรากฏไปสถานะสี่เหลี่ยมที่ตัวบอร์ด ระบบ จะทำการบู๊ตเข้าระบบปฏิบัติการ Raspbian โดยอัตโนมัติ

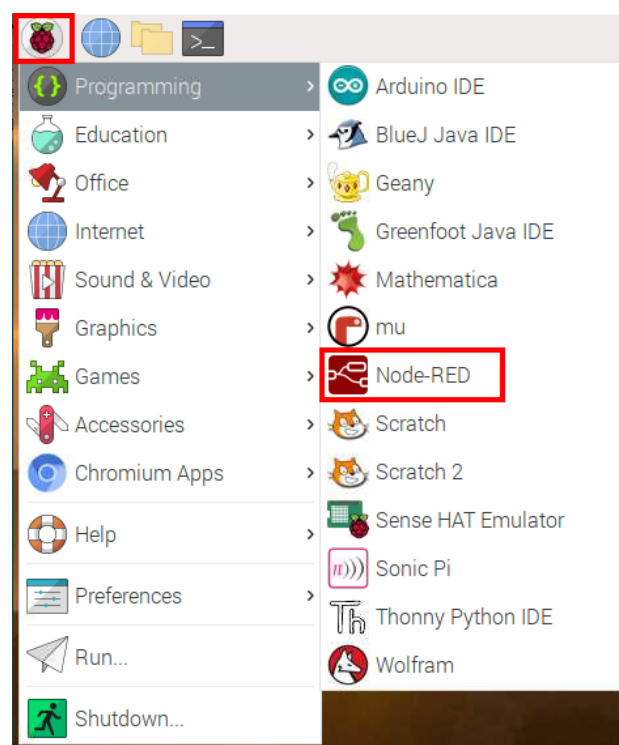


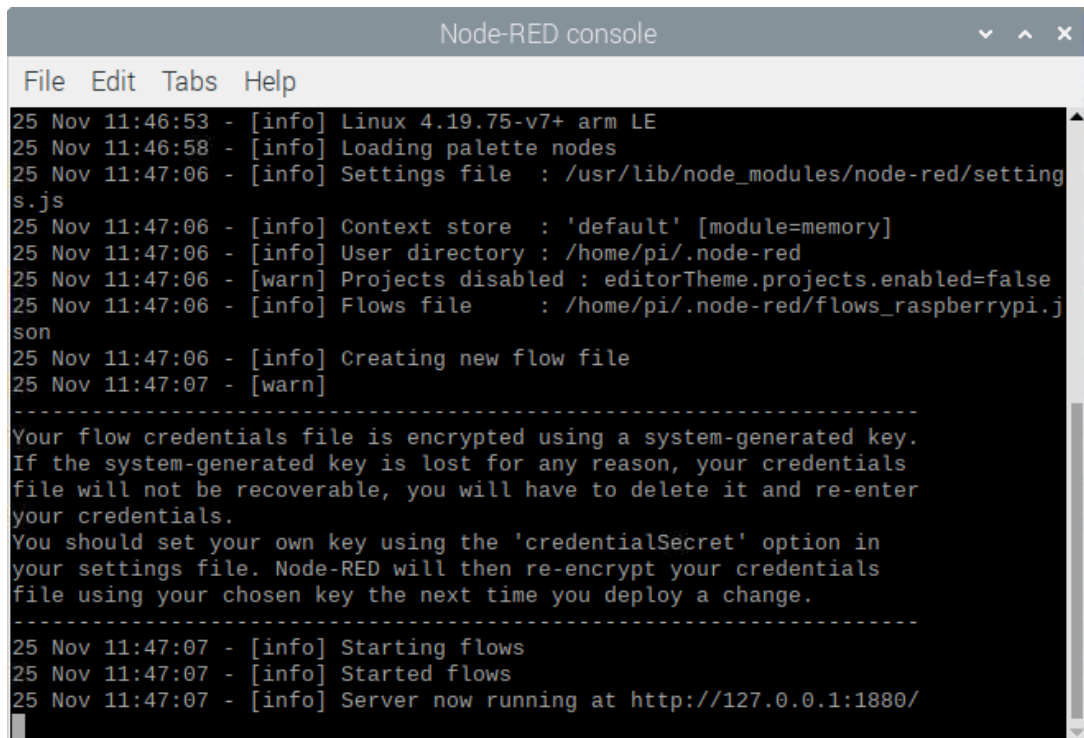
สร้างปุ่มควบคุมและเขียนโปรแกรมคำสั่งด้วยภาษา JavaScript จากโปรแกรม Node-RED

1. ทำการเชื่อมต่อ Wi-Fi โดยคลิกที่เมนู Wi-Fi ที่มุมบนขวาของหน้าจอและคลิก Turn On Wi-Fi หลังจากนั้นคลิกที่ Wi-Fi อีกครั้ง คลิกเลือก Wi-Fi ที่ต้องการเชื่อมต่อ หากมีรหัส Wi-Fi ก็ต้องกรอกรหัส และคลิกปุ่ม OK



2. เปิดใช้งาน Node-RED โดยคลิกที่ RasPi --> Programming --> Node-RED และรอนจนกว่าจะเปิดใช้งานเสร็จเรียบร้อย





```

Node-RED console
File Edit Tabs Help
25 Nov 11:46:53 - [info] Linux 4.19.75-v7+ arm LE
25 Nov 11:46:58 - [info] Loading palette nodes
25 Nov 11:47:06 - [info] Settings file : /usr/lib/node_modules/node-red/settings.js
25 Nov 11:47:06 - [info] Context store : 'default' [module=memory]
25 Nov 11:47:06 - [info] User directory : /home/pi/.node-red
25 Nov 11:47:06 - [warn] Projects disabled : editorTheme.projects.enabled=false
25 Nov 11:47:06 - [info] Flows file : /home/pi/.node-red/flows_raspberrypi.json
25 Nov 11:47:06 - [info] Creating new flow file
25 Nov 11:47:07 - [warn]
-----
Your flow credentials file is encrypted using a system-generated key.
If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.
You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----
25 Nov 11:47:07 - [info] Starting flows
25 Nov 11:47:07 - [info] Started flows
25 Nov 11:47:07 - [info] Server now running at http://127.0.0.1:1880/

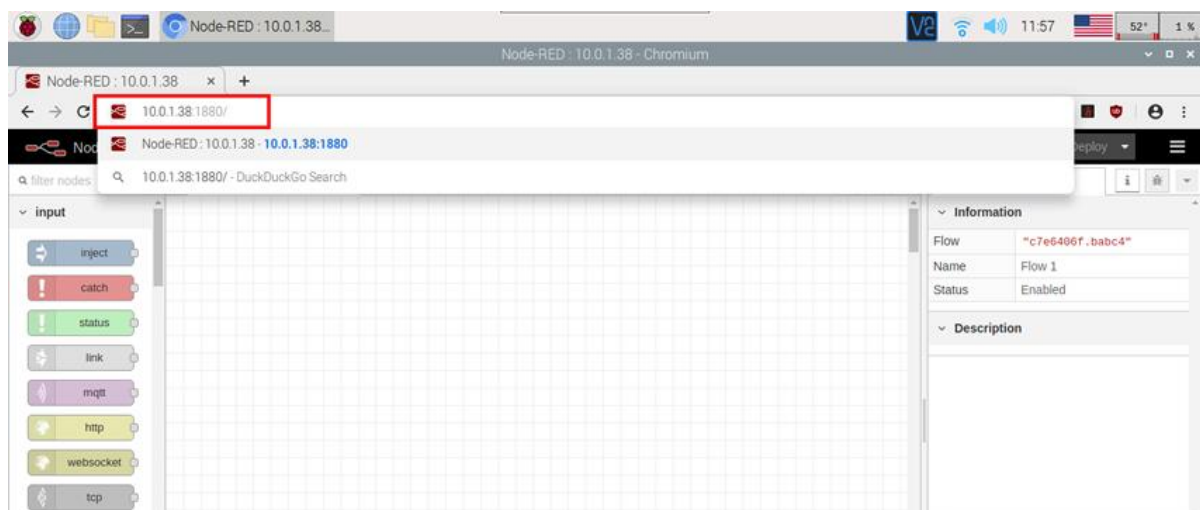
```

3. ตรวจสอบ IP Address ของบอร์ด Raspberry Pi โดยใช้เมาส์ชี้ที่ Wi-Fi จะปรากฏหมายเลข IP Address ของเครื่อง

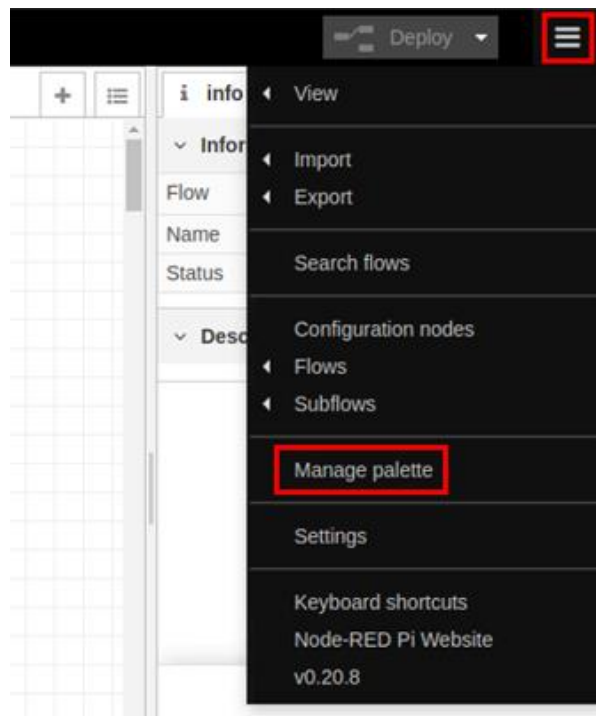
4. คลิกที่ Web Browser แล้วพิมพ์ url ตามรูปแบบด้านล่าง แล้วกดปุ่ม Enter

<http://<ใส่ IP Address ของบอร์ด Raspberry Pi>:1880>

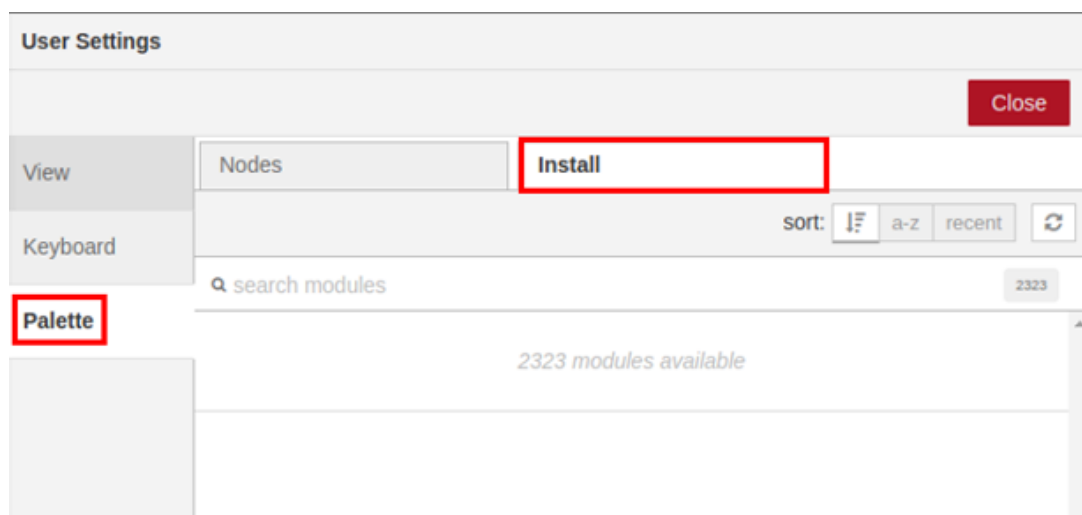
เช่น <http://10.0.1.38:1880>



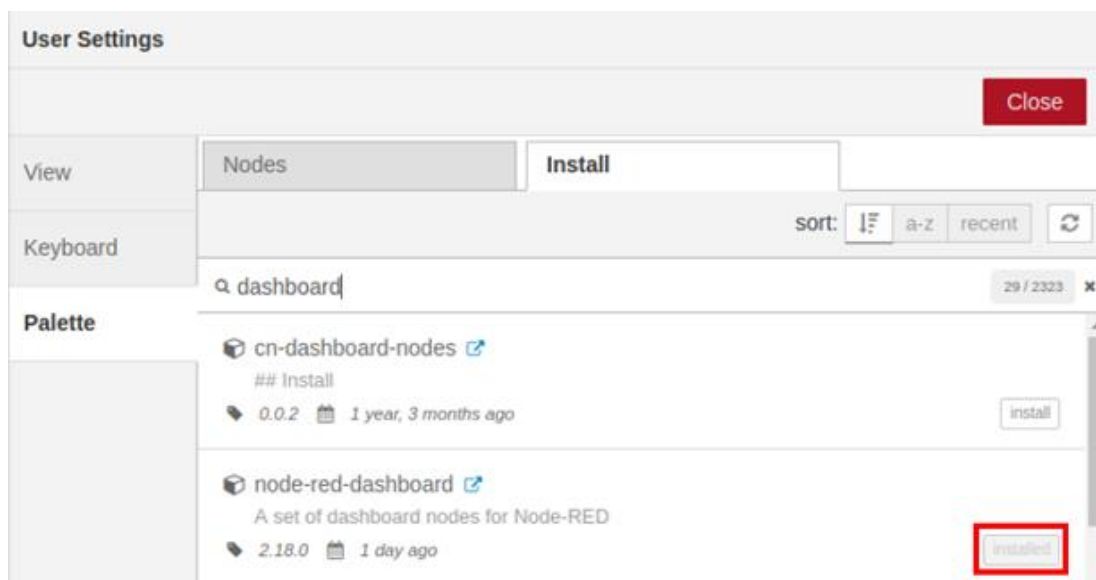
5. คาว์โหนด Node dashboard โดยไปที่เมนู แล้วคลิกที่  แล้วคลิกเมนู Manage palette



6. คลิกแท็บ Palette แล้วคลิกที่แท็บ Install



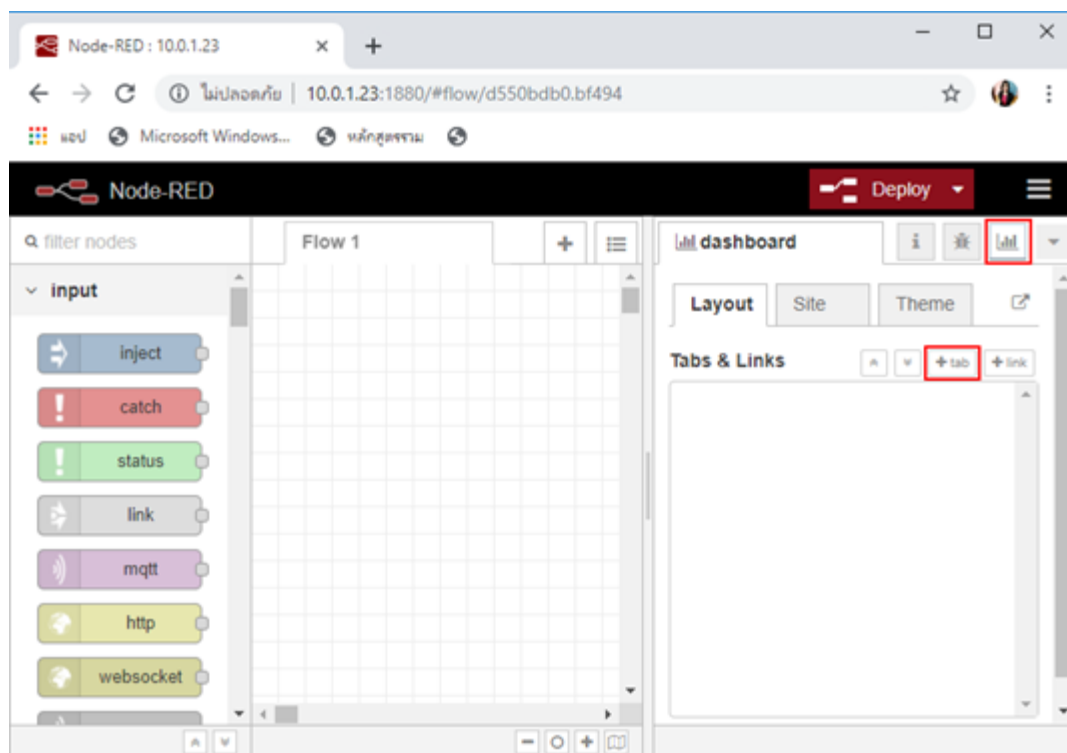
7. คลิกในช่อง search modules แล้วพิมพ์คำว่า "dashboard" จะเจอโหนดที่มีชื่อว่า node-red-dashboard ให้คลิกปุ่ม install



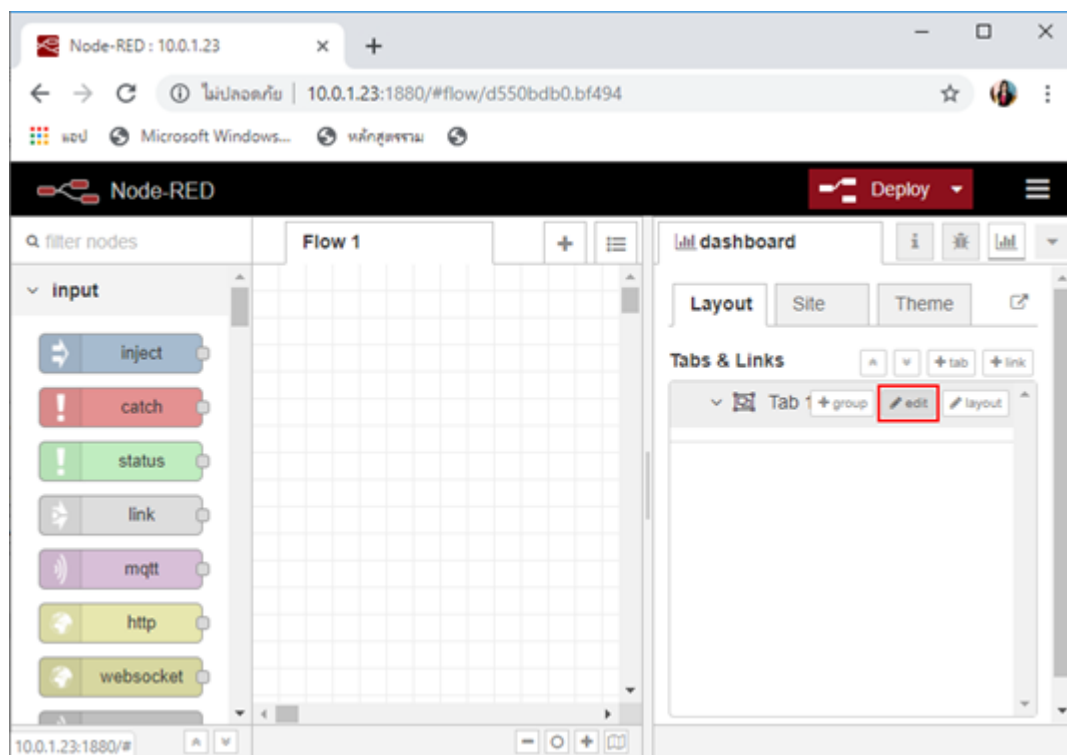
8. จะปรากฏโหนด dashboard ขึ้นมาด้านซ้ายของโปรแกรม ดังรูป



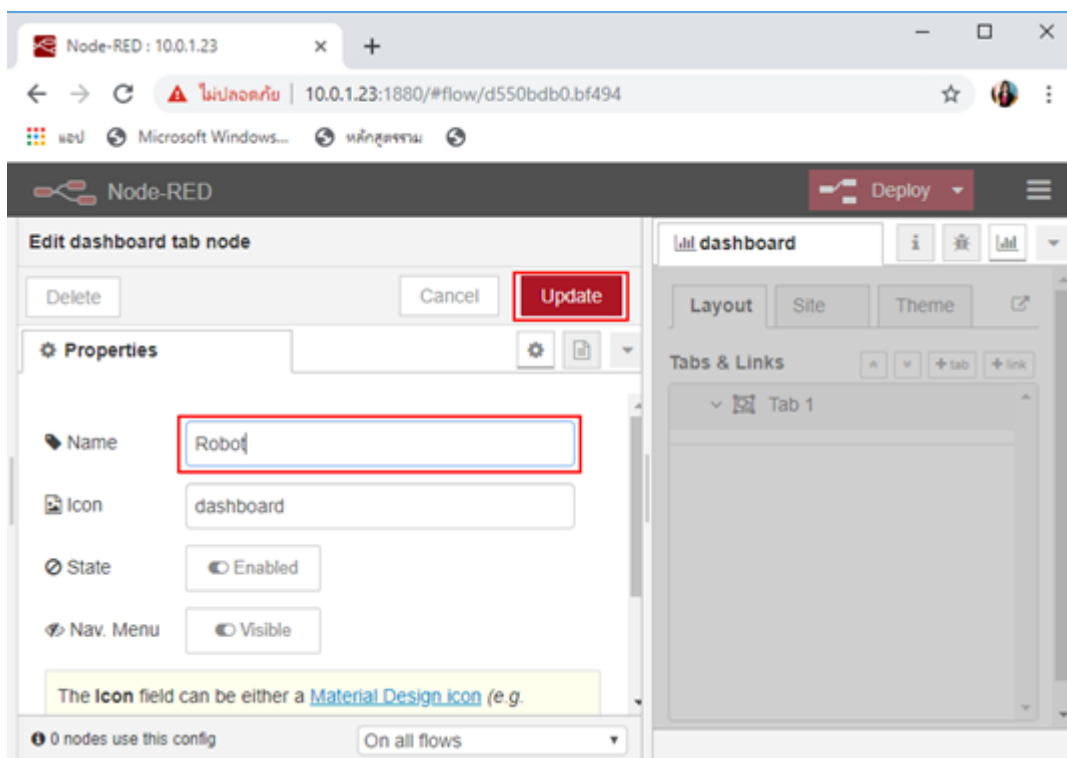
9. คลิกที่เมนู dashboard ที่มุมด้านบนขวา ในแท็บ Layout แล้วคลิกที่ +tab เพื่อทำการสร้างแท็บใหม่ขึ้นมา



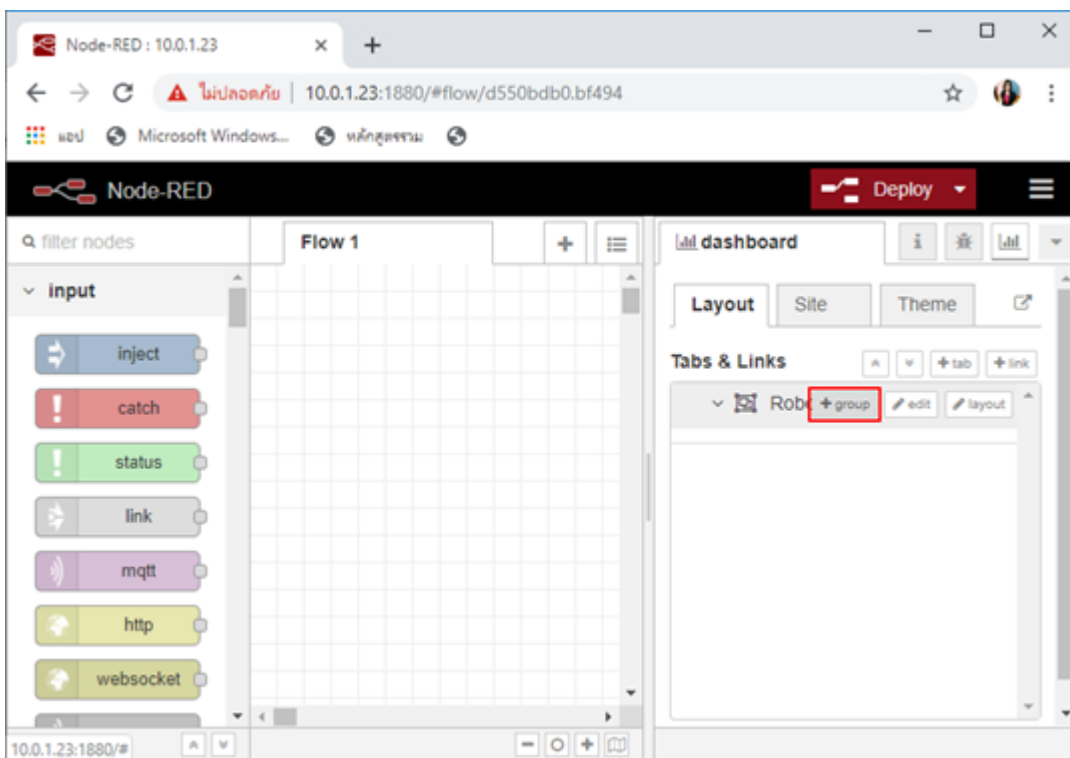
แท็บที่สร้างขึ้นใหม่จะชื่อว่า Tab1 ให้เปลี่ยนชื่อแท็บโดยคลิกที่คำสั่ง edit ที่อยู่ด้านหลังแท็บ



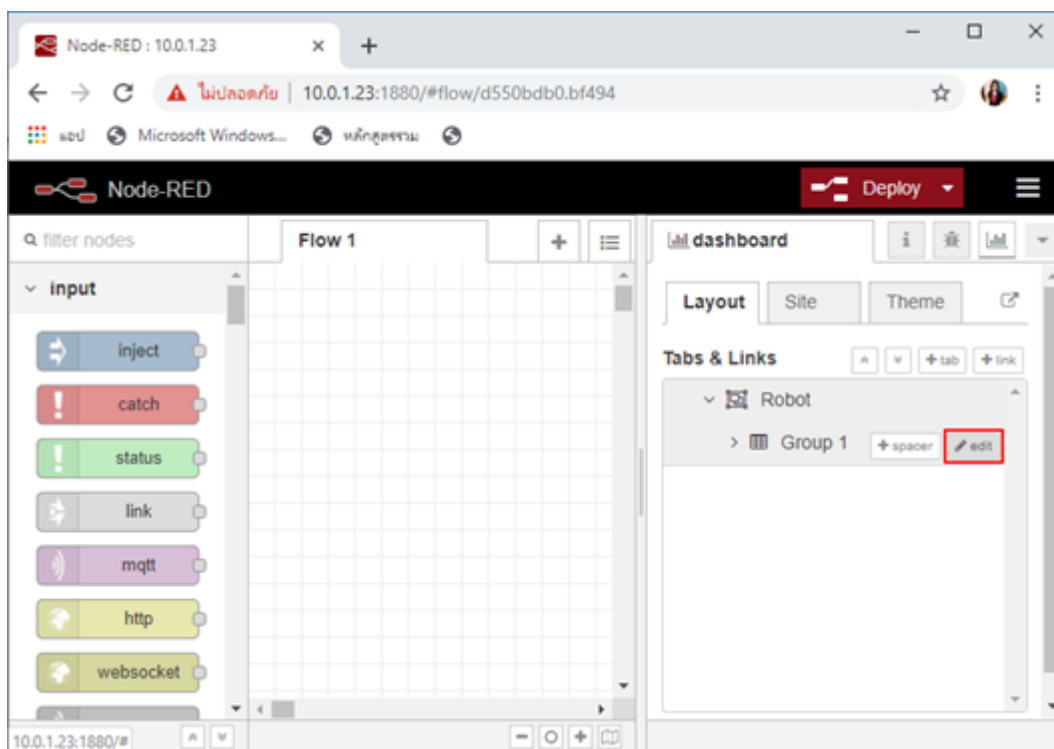
เปลี่ยนชื่อแท็บ ในที่นี้จะตั้งชื่อแท็บว่า Robot แล้วคลิกที่ปุ่ม Update



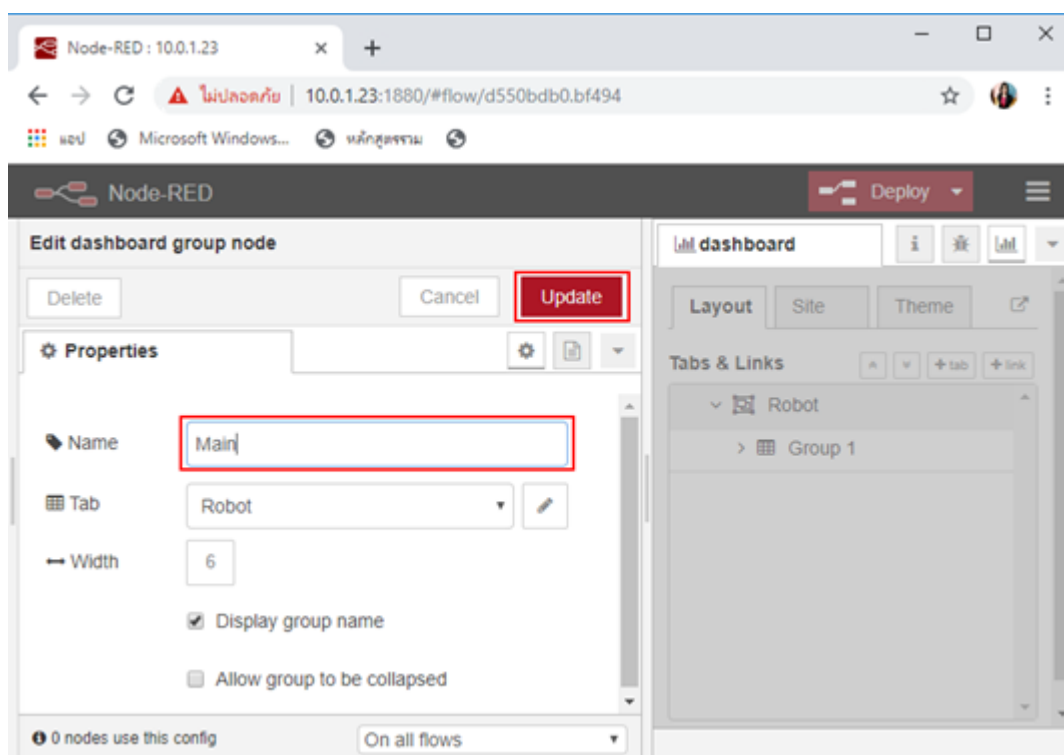
สร้าง Group ในแท็บ Robot โดยคลิกที่ +group ด้านหลังแท็บ



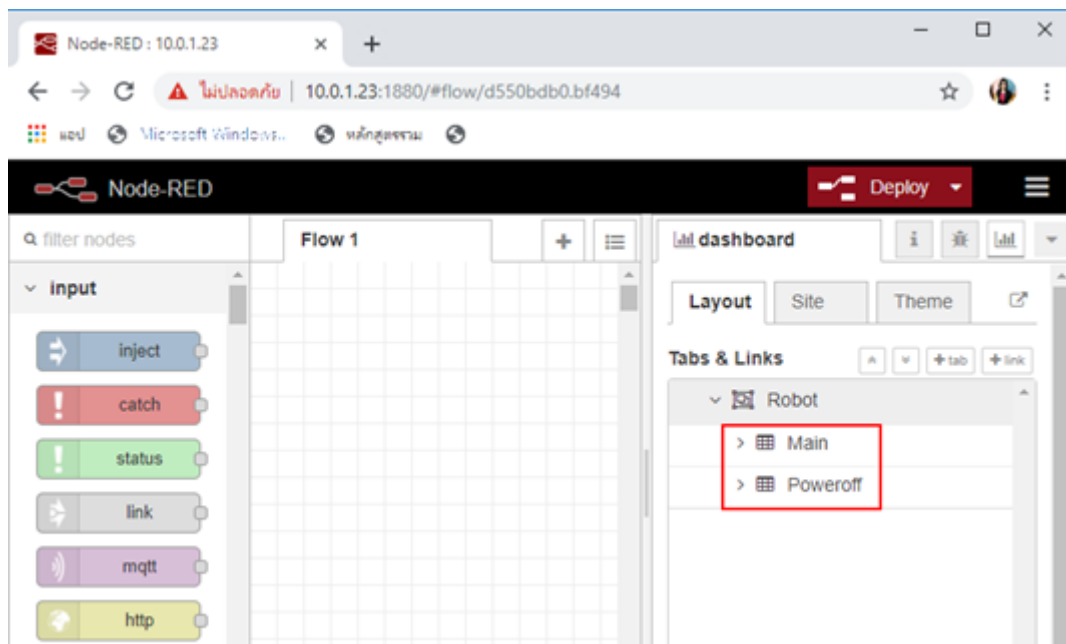
เมื่อสร้าง Group ขึ้นมา จะมีชื่อว่า Group1 ให้เปลี่ยนชื่อ Group โดยคลิกที่ปุ่ม edit



เปลี่ยนชื่อ Group เป็น Main และคลิกปุ่ม Update ดังรูป

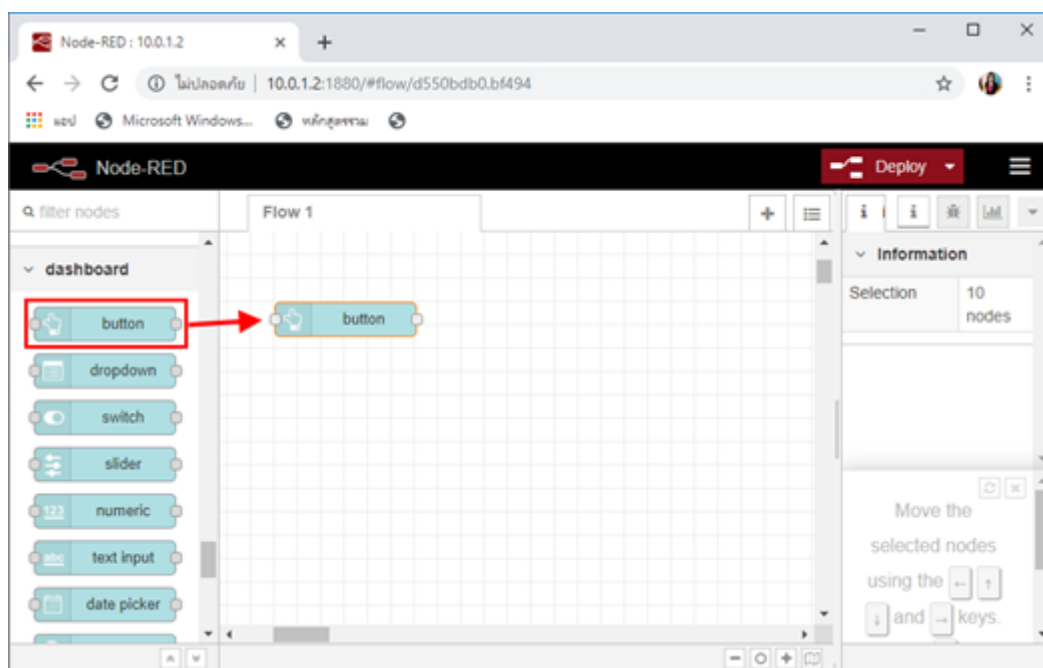


สร้าง Group เพิ่มขึ้นมาอีก 1 Group โดยตั้งชื่อว่า Poweroff โดยทำตามขั้นตอนเดียวกับ Group Main



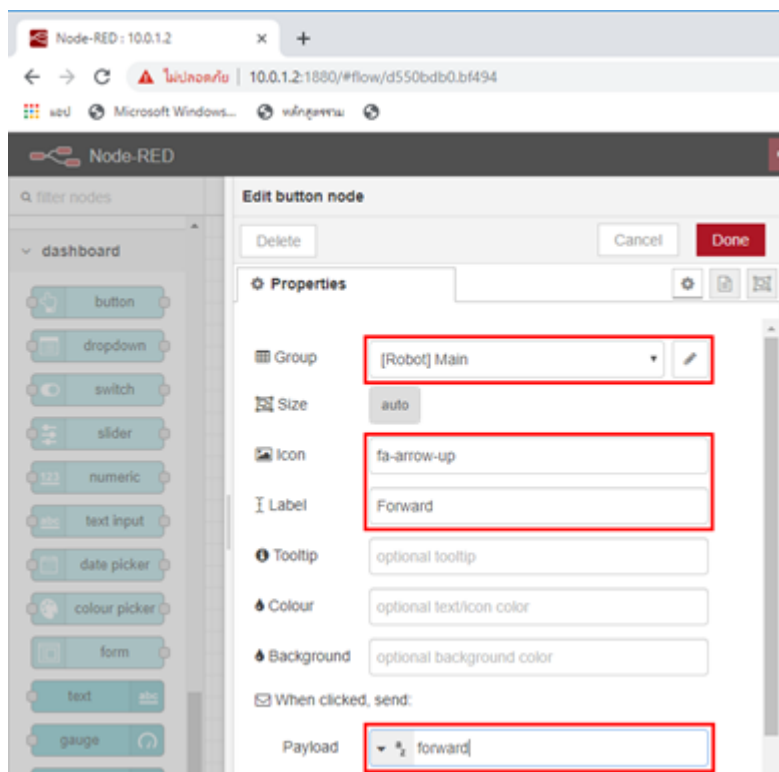
10. สร้างปุ่มควบคุมขึ้นมา 5 ปุ่ม คือ ปุ่ม Forward, Backward, Turn Left, Turn Right และ Stop ใน Group : Main

10.1 ลากปุ่มคำสั่ง button มาวางบนพื้นที่วาด



10.2 ดับเบิลคลิกปุ่มที่เพิ่งลากมาวาง แล้วกำหนดค่าต่างๆ ดังรูป แล้วคลิกปุ่ม Done

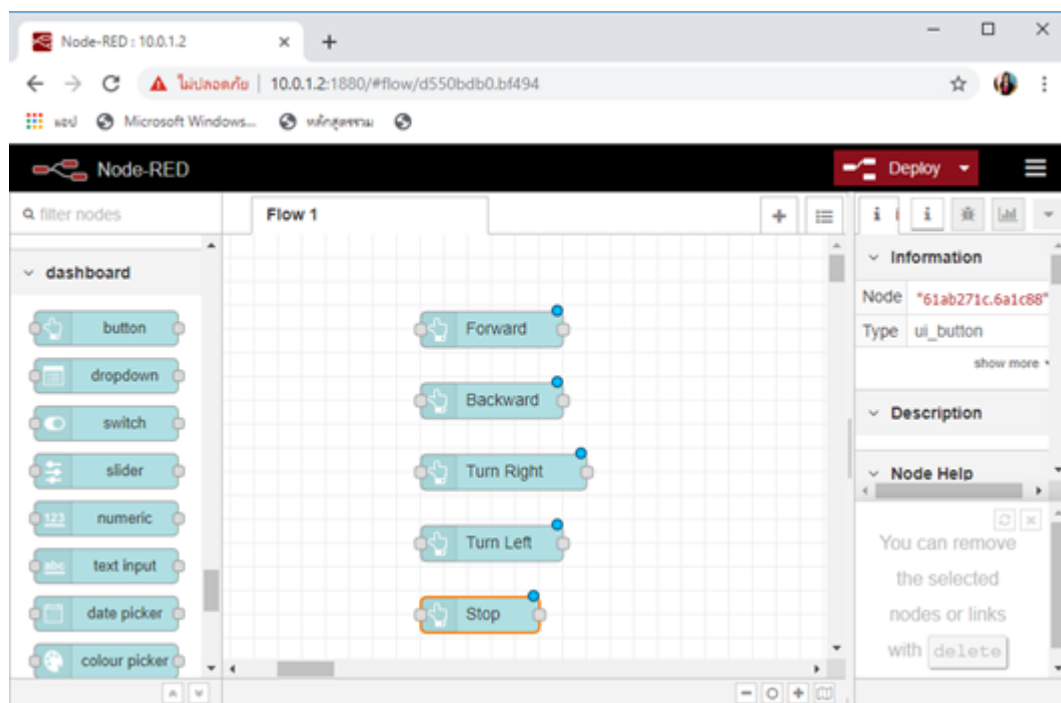
- Group : Main
- Icon : fa-arrow-up- Label : Forward
- Payload : forward



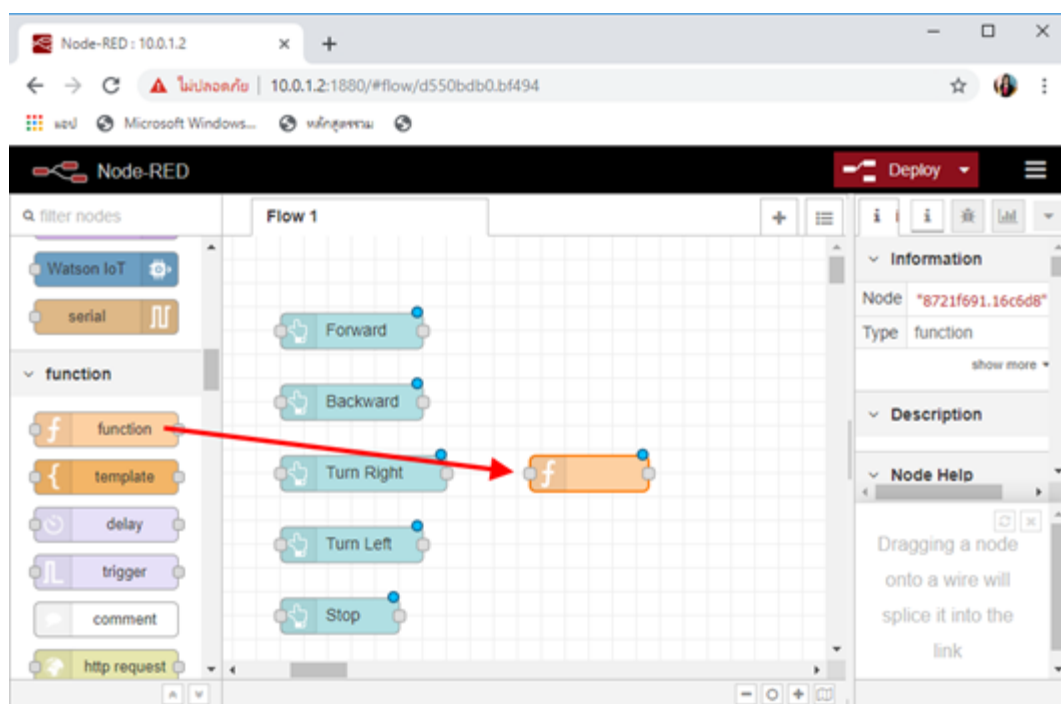
10.3 สร้างปุ่ม Backward, Turn Left, Turn Right และ Stop ขึ้นมาตามลำดับ โดยทำซ้ำตามขั้นตอน 10.1 และ 10.2 โดยกำหนดค่าดังนี้

NODE	การกำหนดคุณสมบัติ
Forward	Group : Main[Robot] Size : auto Icon : fa-arrow-up Label : Forward Payload : forward

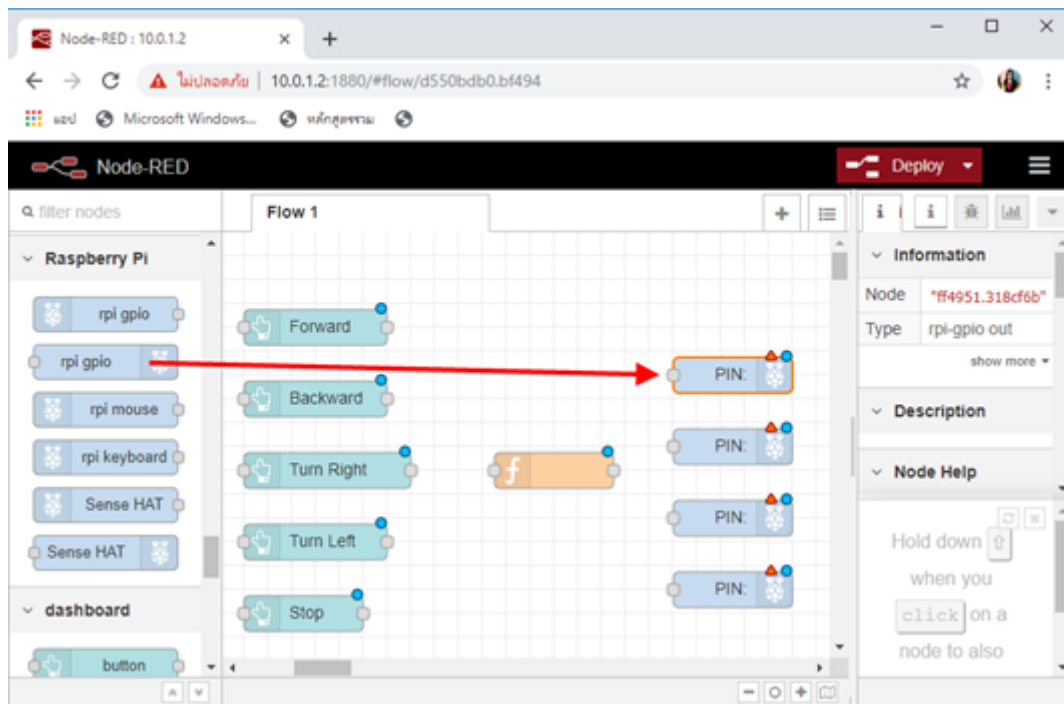
Backward	Group : Main[Robot] Size : auto Icon : fa-arrow-down Label : Backward Payload : backward
Turn Left	Group : Main[Robot] Size : auto Icon : fa-arrow-left Label : Turn Left Payload : left
Turn Right	Group : Main[Robot] Size : auto Icon : fa-arrow-right Label : Turn Right Payload : right
Stop	Group : Main[Robot] Size : auto Icon : fa-hand-paper-o Label : Stop Payload : stop



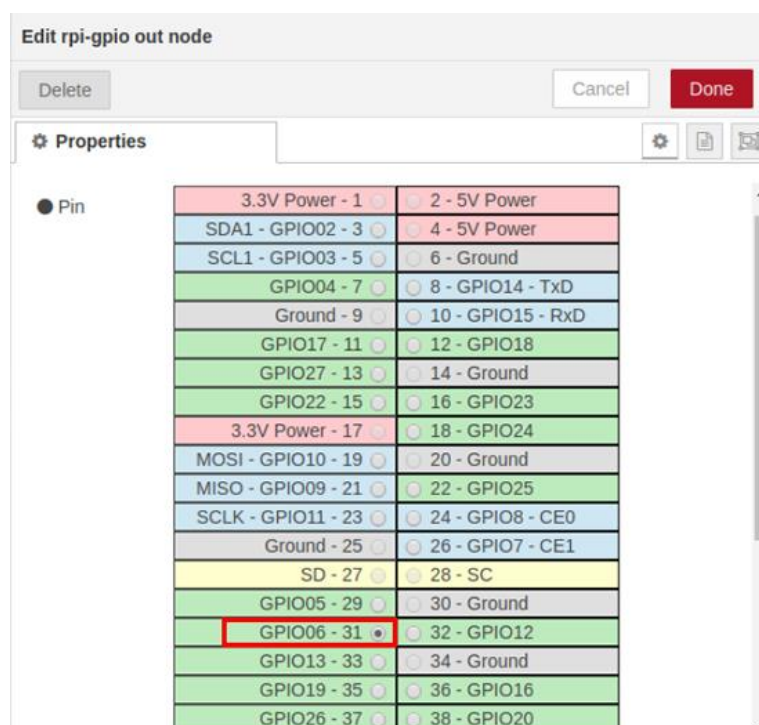
11. ลากปุ่ม Function เข้ามาเพื่อเขียนโปรแกรมด้วยภาษา JavaScript ควบคุมการเคลื่อนที่ของรถหุ่นยนต์ เมื่อผู้ใช้กดปุ่มคำสั่ง

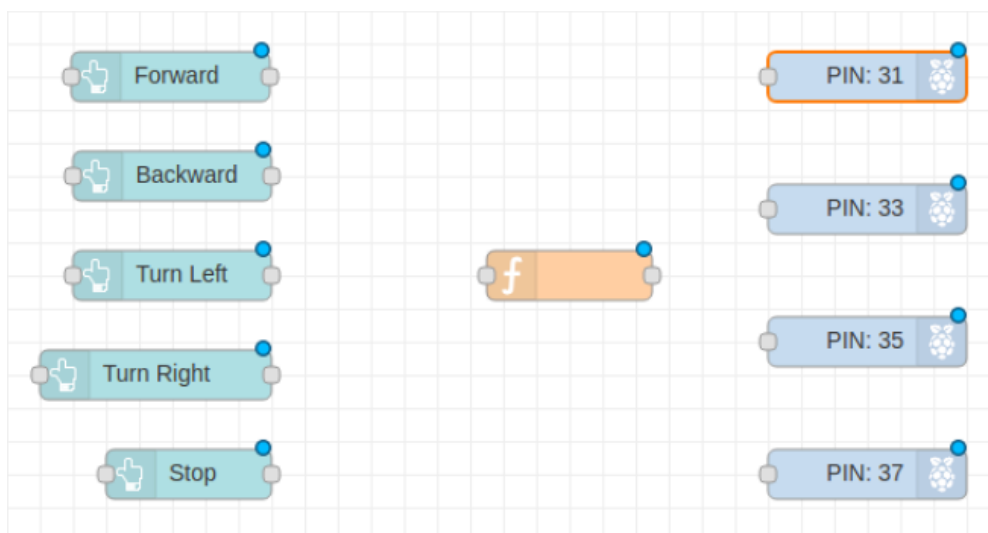


12. ลากบล็อก rpi gpio ออกมา 4 อัน ซึ่งจะเป็นส่วน Output ของโปรแกรม ซึ่งในที่นี้คือขา GPIO ที่เชื่อมต่อกับมอเตอร์ของหุ่นยนต์นั่นเอง

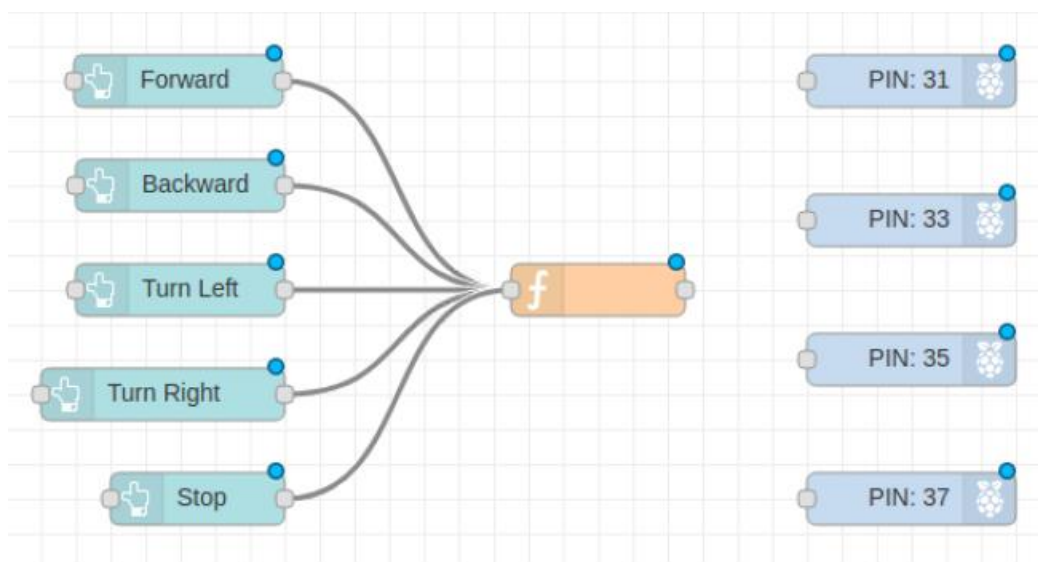


13. ดับเบิลคลิกที่บล็อก PIN : แต่ละบล็อกแล้วกำหนดขา GPIO ที่เชื่อมต่อกับมอเตอร์ ซึ่งก่อนหน้านี้เราได้ทำการเชื่อมต่อเข้ากับขา GPIO ลำดับที่ 31, 33, 35 และ 37 ตามลำดับ

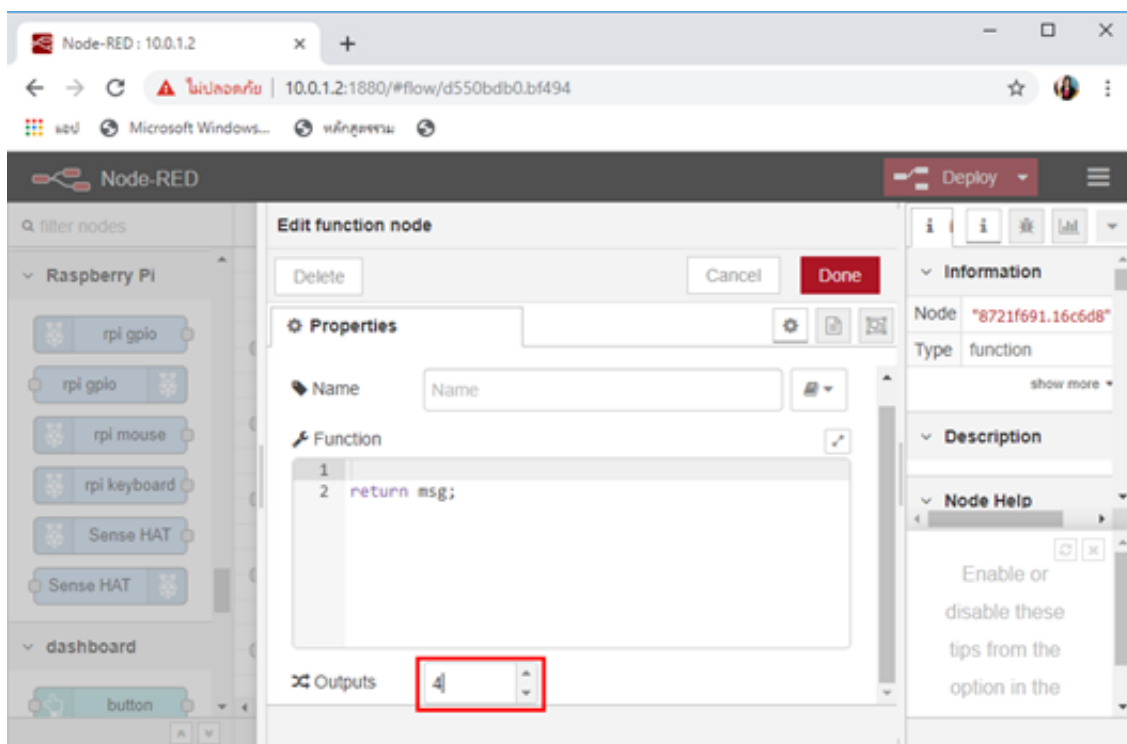




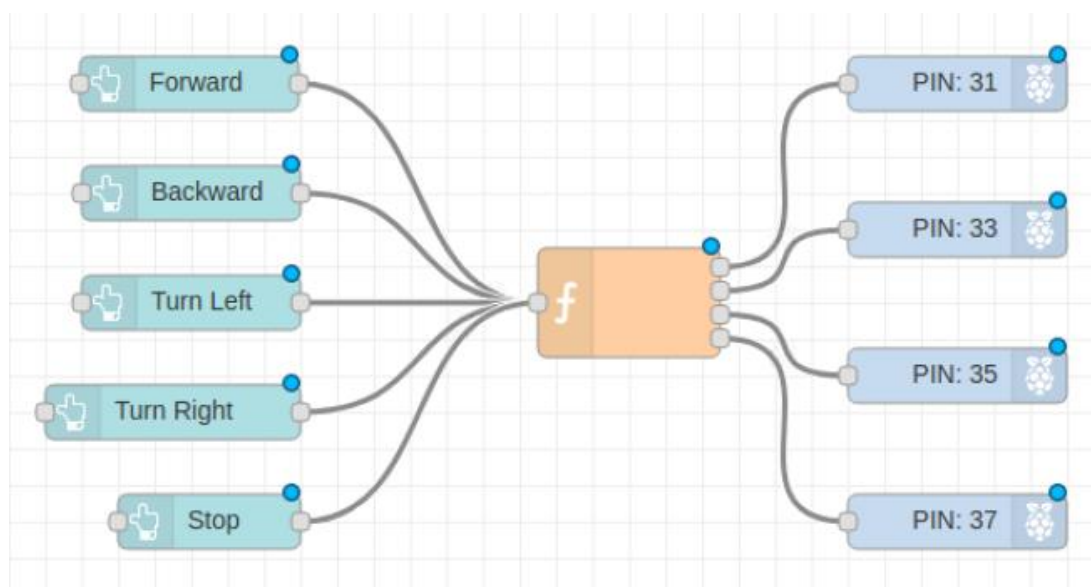
14. ลากเส้นเชื่อมต่อส่วน Input (ปุ่ม) เข้ากับฟังก์ชัน (โปรแกรม) ให้ครบทุกปุ่ม



15. ดับเบิลคลิกเข้าไปในบล็อกฟังก์ชัน แล้วกำหนดว่ามีกี่ Output แล้วคลิกปุ่ม Done



16. ลากเส้นเชื่อมต่อบล็อกฟังก์ชันเข้ากับ Output ให้ครบทุกจุด



17. ดับเบิ้ลคลิกเข้าไปในบล็อกฟังก์ชัน แล้วใส่โค้ดภาษา JavaScript ดังนี้

```
var msg1 = {payload: 0};
var msg2 = {payload: 0};
var msg3 = {payload: 0};
var msg4 = {payload: 0};
if(msg.payload === "forward"){
    msg1.payload = 1;
    msg2.payload = 0;
    msg3.payload = 0;
    msg4.payload = 1;
}
if(msg.payload === "backward"){
    msg1.payload = 0;
    msg2.payload = 1;
    msg3.payload = 1;
    msg4.payload = 0;
}
if(msg.payload === "right"){
    msg1.payload = 1;
    msg2.payload = 0;
    msg3.payload = 0;
    msg4.payload = 0;
}
if(msg.payload === "left"){
    msg1.payload = 0;
    msg2.payload = 0;
    msg3.payload = 0;
    msg4.payload = 1;
}
if(msg.payload === "stop"){
    msg1.payload = 0;
```

```

    msg2.payload = 0;
    msg3.payload = 0;
    msg4.payload = 0;
}
return [msg1,msg2,msg3,msg4];

```

```

1  var msg1 = {payload: 0};
2  var msg2 = {payload: 0};
3  var msg3 = {payload: 0};
4  var msg4 = {payload: 0};
5  if(msg.payload === "forward"){
6      msg1.payload = 1;
7      msg2.payload = 0;
8      msg3.payload = 0;
9      msg4.payload = 1;
10 }
11 if(msg.payload === "backward"){
12     msg1.payload = 0;
13     msg2.payload = 1;
14     msg3.payload = 1;
15     msg4.payload = 0;
16 }

17 if(msg.payload === "right"){
18     msg1.payload = 1;
19     msg2.payload = 0;
20     msg3.payload = 0;
21     msg4.payload = 0;
22 }
23 if(msg.payload === "left"){
24     msg1.payload = 0;
25     msg2.payload = 0;
26     msg3.payload = 0;
27     msg4.payload = 1;
28 }

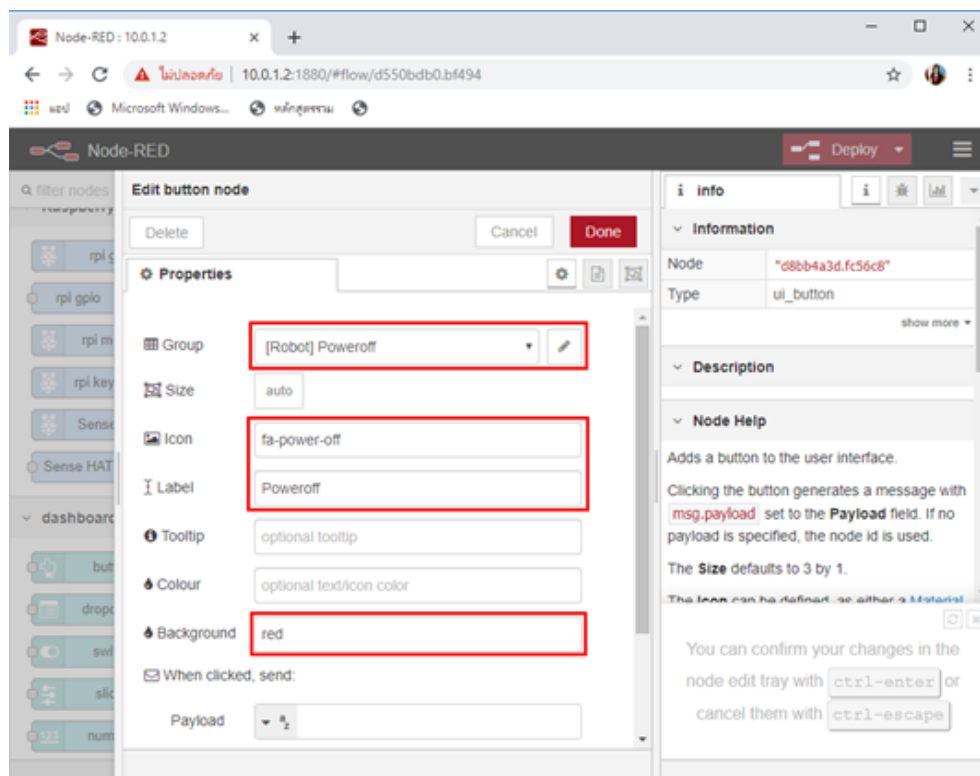
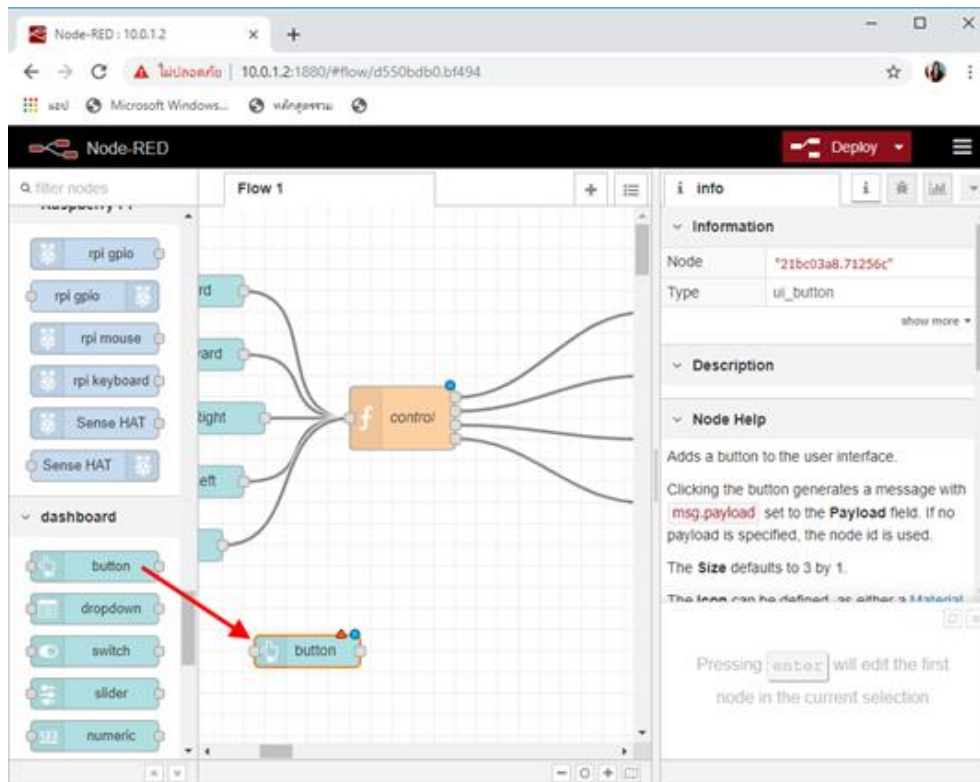
29 if(msg.payload === "stop"){
30     msg1.payload = 0;
31     msg2.payload = 0;
32     msg3.payload = 0;
33     msg4.payload = 0;
34 }
35
36 return[msg1,msg2,msg3,msg4];

```

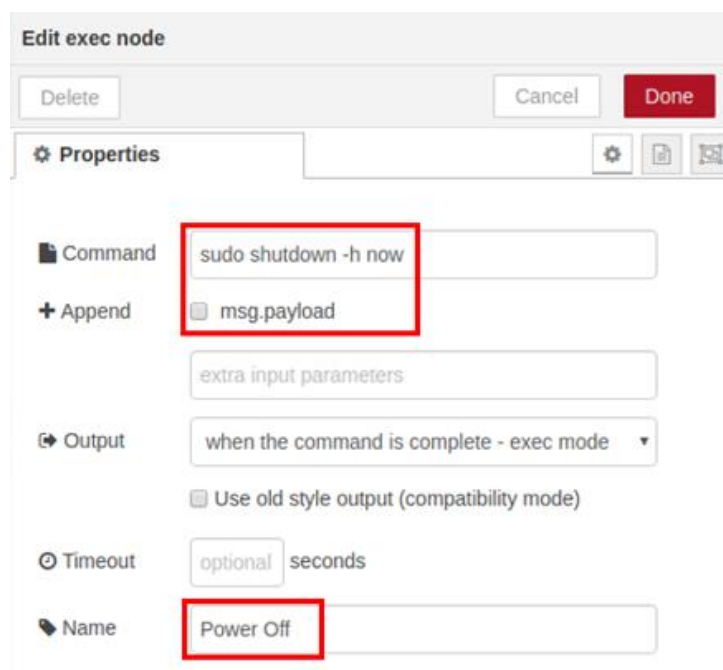
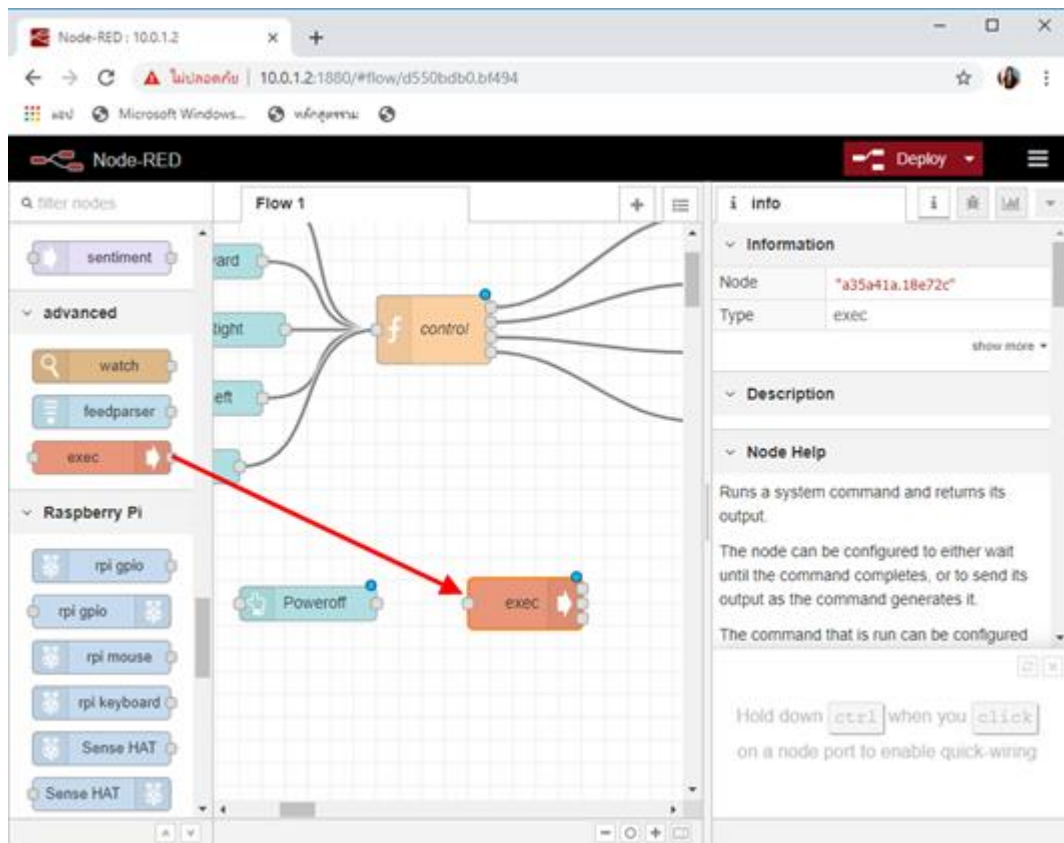

18. สร้างปุ่มปิดใน Group : PowerOff

18.1 ลากบล็อก button เข้ามาวางในพื้นที่การวาด แล้วดับเบิลคลิกเข้าไปตั้งค่าดังนี้ แล้วคลิก

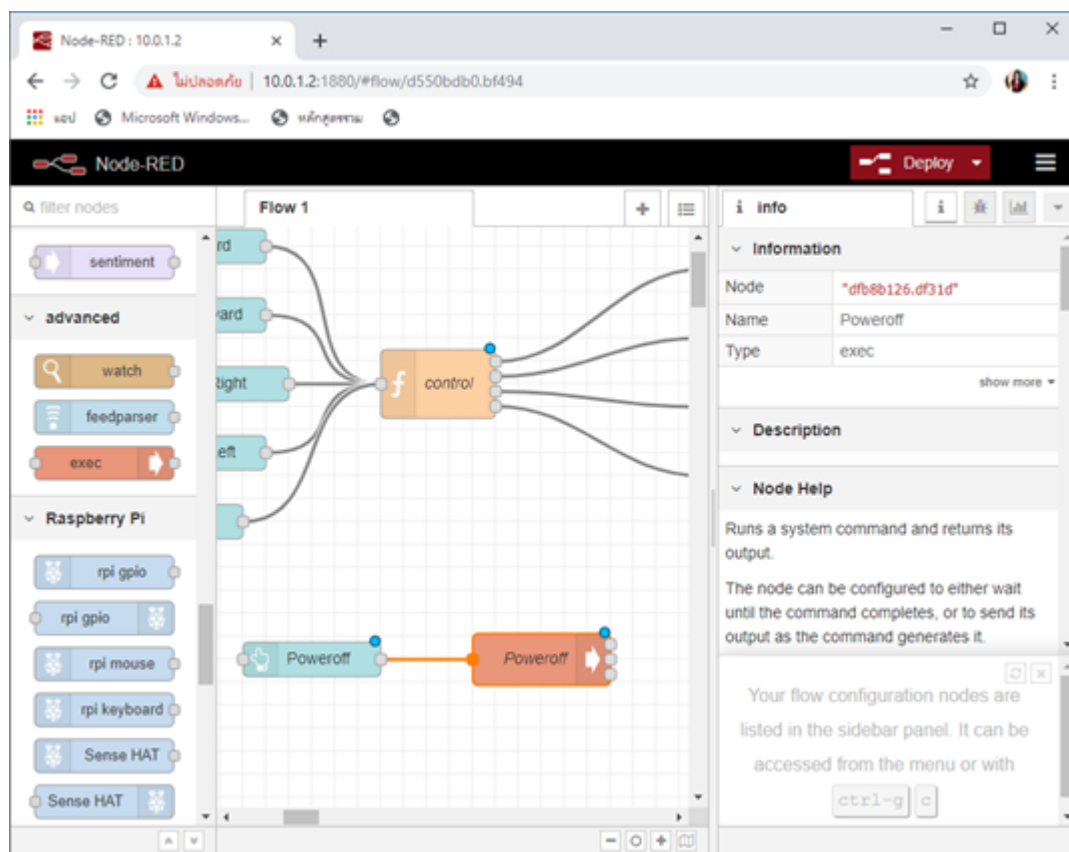
ปุ่ม Done



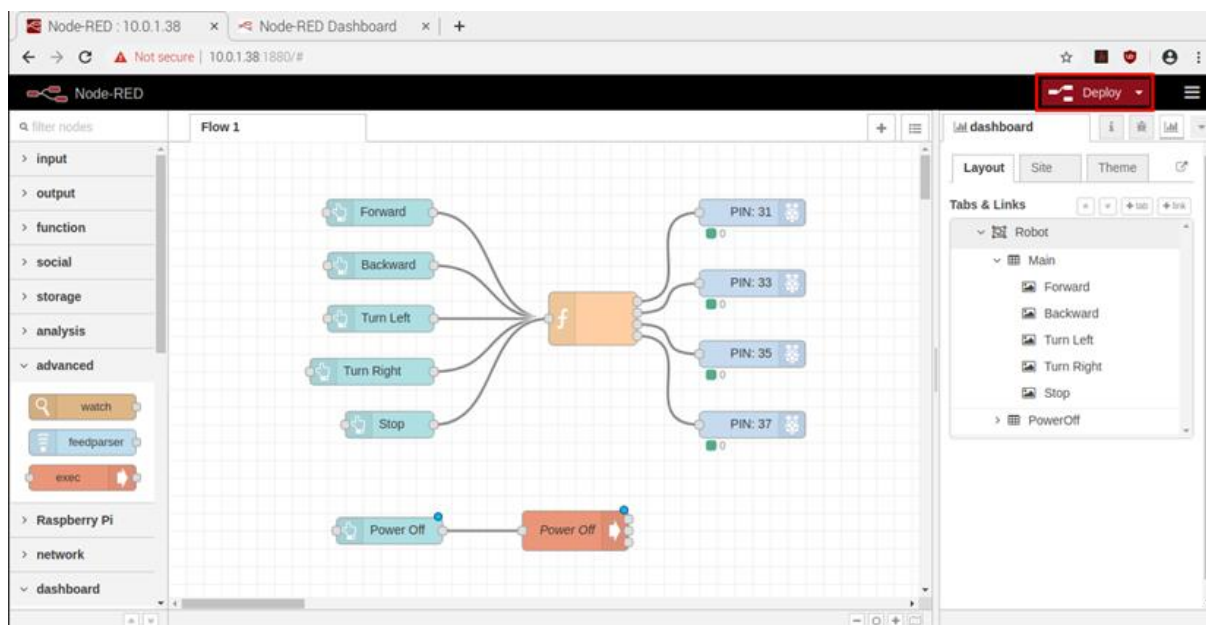
18.2 ลากบล็อก exec มาวาง เพื่อให้ได้คำสั่งในการปิดบอร์ด Raspberry Pi โดยดับเบิลคลิกเข้าไปในบล็อก และตั้งค่าดังนี้



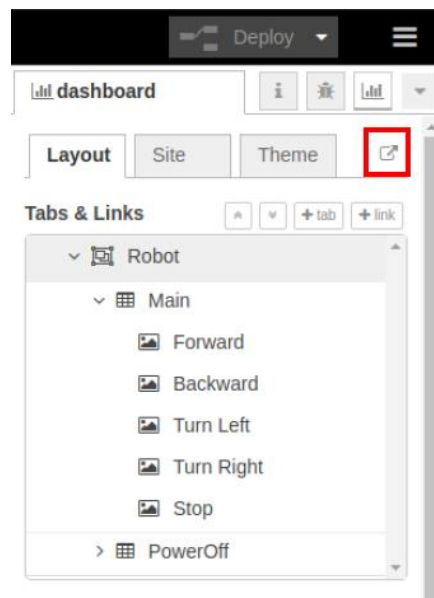
18.3 ลากเส้นเชื่อมโยงระหว่าง 2 บล็อก ดังรูป



19. เมื่อเขียนโปรแกรมเสร็จเรียบร้อยแล้ว ให้กดปุ่ม Deploy ด้านบนขวาของโปรแกรม Node-RED



20. คลิกเมนู Preview เพื่อเปิดหน้าจอผลลัพธ์ ดังรูป

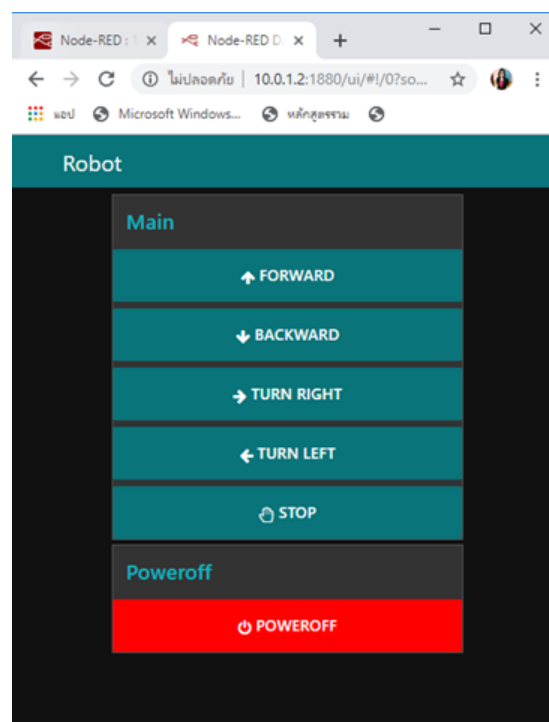


ซึ่งจะเห็นว่าหน้าแสดงผลจะมีรูปแบบ URL ดังนี้

`http://<ใส่ IP Address ของบอร์ด Raspberry Pi>:1880/ui`

เช่น <http://10.0.1.23:1880/ui>

จะได้หน้าจอโปรแกรม ดังรูป



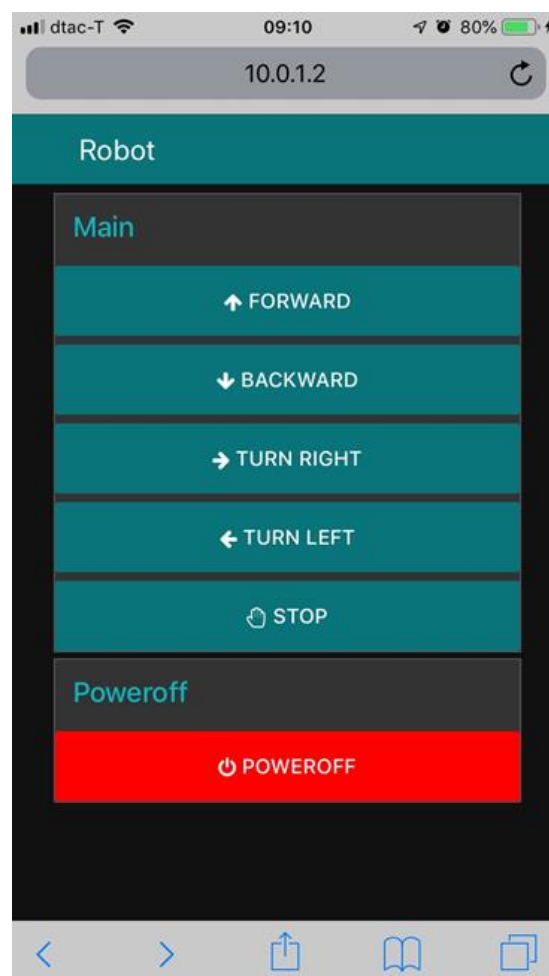
21. นำถ่านขนาด 2A จำนวน 4 ก้อนใส่ในกระเบถ่าน หลังจากนั้นให้ทดลองกดปุ่มต่างๆ แล้วดูการเคลื่อนที่ของหุ่นยนต์ หากผลลัพธ์ไม่ถูกต้อง ก็กลับไปแก้ไขในโปรแกรม Node-RED หลังการแก้ไขให้คลิกปุ่ม Deploy ใหม่อีกครั้ง

22. เมื่อได้ผลลัพธ์ตามที่ต้องการแล้ว ให้ถอดสาย HDMI เมาส์ และคีย์บอร์ดออกจากบอร์ด Raspberry Pi เพื่อให้หุ่นยนต์เคลื่อนที่ได้อย่างอิสระ

23. ใช้มือถือ แท็บเล็ต หรือคอมพิวเตอร์จับ Wi-Fi ตัวเดียวกับ Raspberry Pi แล้วเปิดเว็บเบราว์เซอร์ขึ้นมา แล้วพิมพ์ url

<http://<ใส่ IP Address ของบอร์ด Raspberry Pi>:1880/ui>

จะได้หน้าจอโปรแกรมควบคุมหุ่นยนต์ ดังรูป



2. หุ่นยนต์หลบหลีกสิ่งกีดขวาง

สิ่งที่ต้องมี

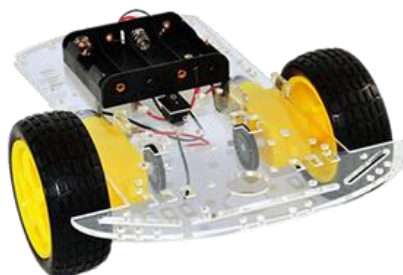
1. โมดูลวัดระยะทาง HC-SR04 (เซนเซอร์อัลตราโซนิก)



2. บอร์ด Raspberry Pi ใช้บอร์ดที่มี GPIO 40 ขา



3. ชุดประกอบรถหุ่นยนต์ ซึ่งประกอบด้วยตัวรถ ล้อ มอเตอร์ กระจับปะใส่ถ่าน



4. บอร์ดขับเคลื่อนมอเตอร์ ในที่นี้จะใช้บอร์ด L298N

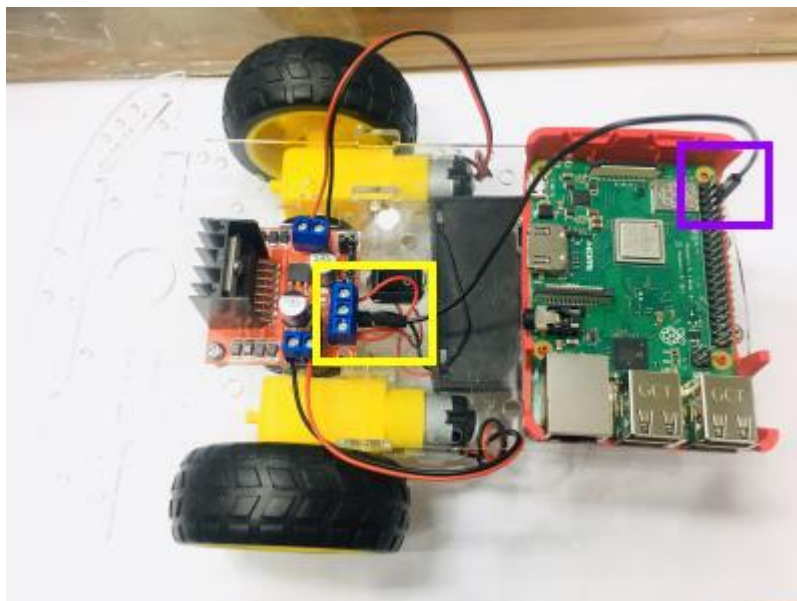


5. Power bank ที่มีขนาดความจุ 2,200 mAh. ขึ้นไป เพื่อจ่ายไฟให้แก่บอร์ด Raspberry Pi เพื่อให้หุ่นยนต์เคลื่อนที่ได้อย่างอิสระ

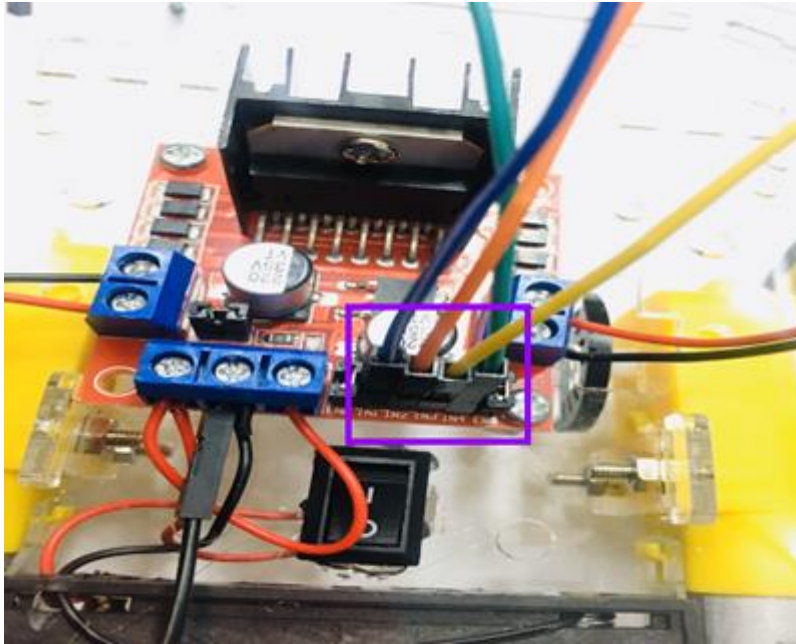


ขั้นตอนการเชื่อมต่อบอร์ด L298N กับบอร์ด Raspberry Pi

1. ใช้ไขควงคลายน็อตในช่อง GROUND (ช่องกลาง) ของบอร์ด L298N ให้เปิดออก แล้วนำขาเสียบฝั่งตัวผู้ของสาย Jumper แบบตัวผู้-ตัวเมีย เสียบเข้าไปในช่อง แล้วขันน็อตให้แน่น ส่วนปลายสายฝั่งตัวเมียของสาย Jumper ให้เสียบเข้าไปที่ขา PIN6 (GND) ของขา GPIO ที่บอร์ด Raspberry Pi



2. ใช้สาย Jumper ตัวเมีย-ตัวเมีย 4 เส้น เสียบเข้าไปในช่อง IN1 – IN4 ของบอร์ด L298N



3. ปลายสาย Jumper อีกด้านเสียบเข้ากับขา GPIO ที่เป็นขา Input/Output ของบอร์ด Raspberry Pi (ในที่นี้จะเสียบเข้ากับขา GPIO ที่บอร์ด Raspberry Pi ตามลำดับคือ

- ขา IN1 ของ L298N ต่อเข้ากับขา PIN31 ของบอร์ด Raspberry Pi
- ขา IN2 ของ L298N ต่อเข้ากับขา PIN33 ของบอร์ด Raspberry Pi
- ขา IN3 ของ L298N ต่อเข้ากับขา PIN35 ของบอร์ด Raspberry P
- ขา IN4 ของ L298N ต่อเข้ากับขา PIN37 ของบอร์ด Raspberry Pi

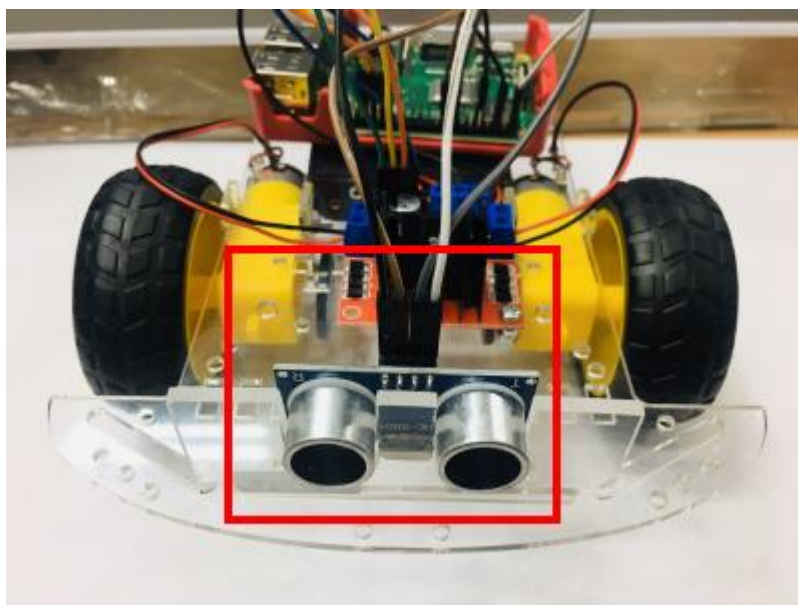


ขั้นตอนการเชื่อมต่อโมดูลวัดระยะทาง HC-SR04 กับบอร์ด Raspberry Pi

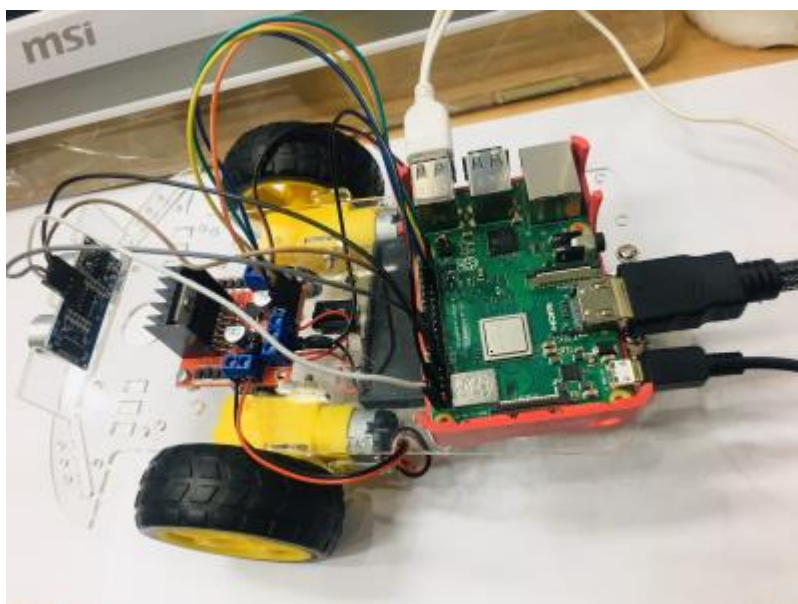
1. ใช้สาย Jumper ตัวเมีย-ตัวเมีย 4 เส้นเสียบเข้ากับขาของโมดูล HC-SR04 ซึ่งมี 4 ขา ดังนี้

- GND จะเสียบเข้ากับขา PIN9 (GROUND) ของบอร์ด Raspberry Pi
- ECHO จะเสียบเข้ากับขา PIN18 ของบอร์ด Raspberry Pi
- TRIG จะเสียบเข้ากับขา PIN16 ของบอร์ด Raspberry Pi
- VCC จะเสียบเข้ากับขา PIN2 (5V) ของบอร์ด Raspberry Pi

หลังจากนั้นให้นำโมดูล HC-SR04 ติดตั้งไว้ด้านหลังหน้าของรถหุ่นยนต์



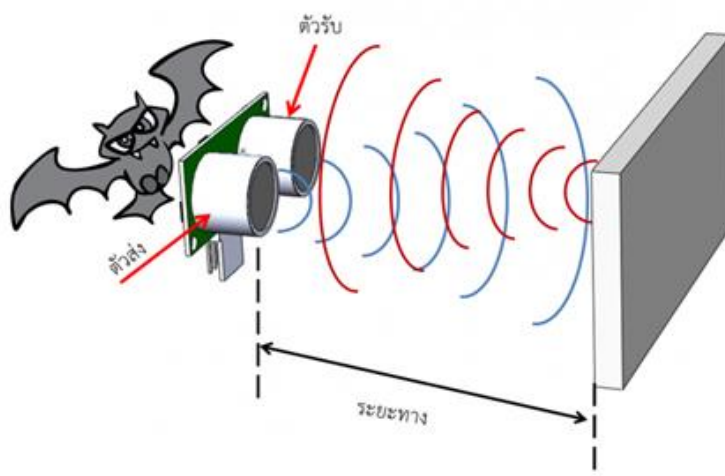
2. นำบอร์ด Raspberry Pi มาเสียบสาย HDMI เข้ากับจอภาพหรือเครื่องคอมพิวเตอร์ที่มีพอร์ต HDMI IN เชื่อมต่อเมาส์และคีย์บอร์ด และหลังจากนั้นเสียบสาย Power Bank เพื่อจ่ายไฟแก่บอร์ด Raspberry Pi จะปรากฏไปสถานะสีส้มที่ตัวบอร์ด ระบบจะทำการบู๊ตเข้าระบบปฏิบัติการ Raspbian โดยอัตโนมัติ



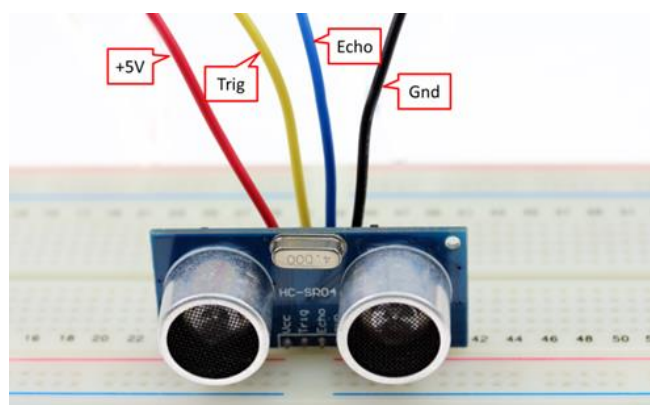
หลักการทำงานของโมดูลวัดระยะทาง HC-SR04

HC-SR04 เป็นเซนเซอร์โมดูลสำหรับตรวจจับวัตถุและวัดระยะทางแบบไม่สัมผัส โดยใช้คลื่นอัลตราโซนิก ซึ่งเป็นคลื่นเสียงความถี่สูงเกินกว่าการได้ยินของมนุษย์ วัดระยะได้ตั้งแต่ 2 – 400 เซนติเมตรหรือ 1 – 156 นิ้ว สามารถต่อใช้งานกับไมโครคอนโทรลเลอร์ได้ง่าย ใช้พลังงานต่ำเหมาะกับการนำไปประยุกต์ใช้งานด้านระบบควบคุมอัตโนมัติหรืองานด้านหุ่นยนต์ หลักการทำงานจะเหมือนกันกับการตรวจจับวัตถุด้วยเสียงของค้างคาว โดยจะประกอบไปด้วยตัวรับ-ส่งอัลตราโซนิก ตัวส่งจะส่งคลื่นความถี่ 40 kHz ออกไปในอากาศด้วยความเร็วประมาณ 346 เมตรต่อวินาที และตัวรับจะคอยรับสัญญาณที่สะท้อนกลับจากวัตถุ เมื่อทราบความเร็วในการเคลื่อนที่ของคลื่น, เวลาที่ใช้ในการเดินทางไป-กลับ (t)

คำนวณหาระยะห่างของวัตถุ (S) ได้จากสูตร $S = 346 \times 0.5t$ (ได้ระยะทางหน่วยเป็นเมตร)

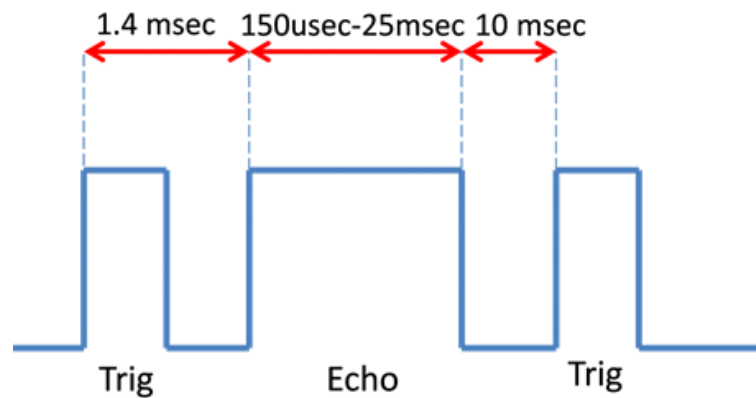


รูปที่ 1 หลักการตรวจจับและวัดระยะห่างระหว่างวัตถุด้วยคลื่นเสียง



รูปที่ 2 การต่อใช้งานโมดูลอัลตราโซนิก HC-SR04

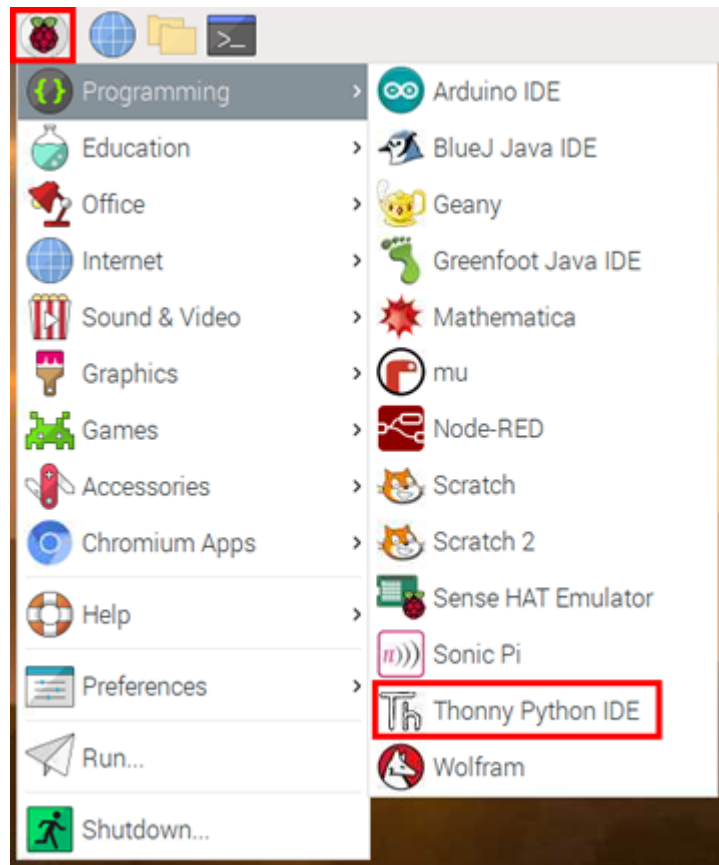
ตามคุณลักษณะของเซนเซอร์ จะต้องสร้างสัญญาณพัลส์ความกว้างไม่น้อยกว่า 10 msec ป้อนเข้าที่ขา Trig หลังจากนั้นอีกประมาณ 1.4 msec จึงจะเริ่มมีสัญญาณพัลส์เกิดขึ้นที่ขา Echo มีความกว้างของสัญญาณตั้งแต่ 150 usec – 25 msec ซึ่งถ้าหากกว้างกว่านี้จะถือว่าตรวจไม่พบวัตถุ หลังจากนั้นควรหน่วงเวลาออกไปอีก 10 msec จึงจะส่งสัญญาณ Trig ออกไปอีกรอบ ตามรูปที่ 3



รูปที่ 3 สัญญาณที่ขา Trig และขา Echo ของโมดูลเซนเซอร์อัลตราโซนิก HC-SR04

เขียนโปรแกรมด้วยภาษา Python

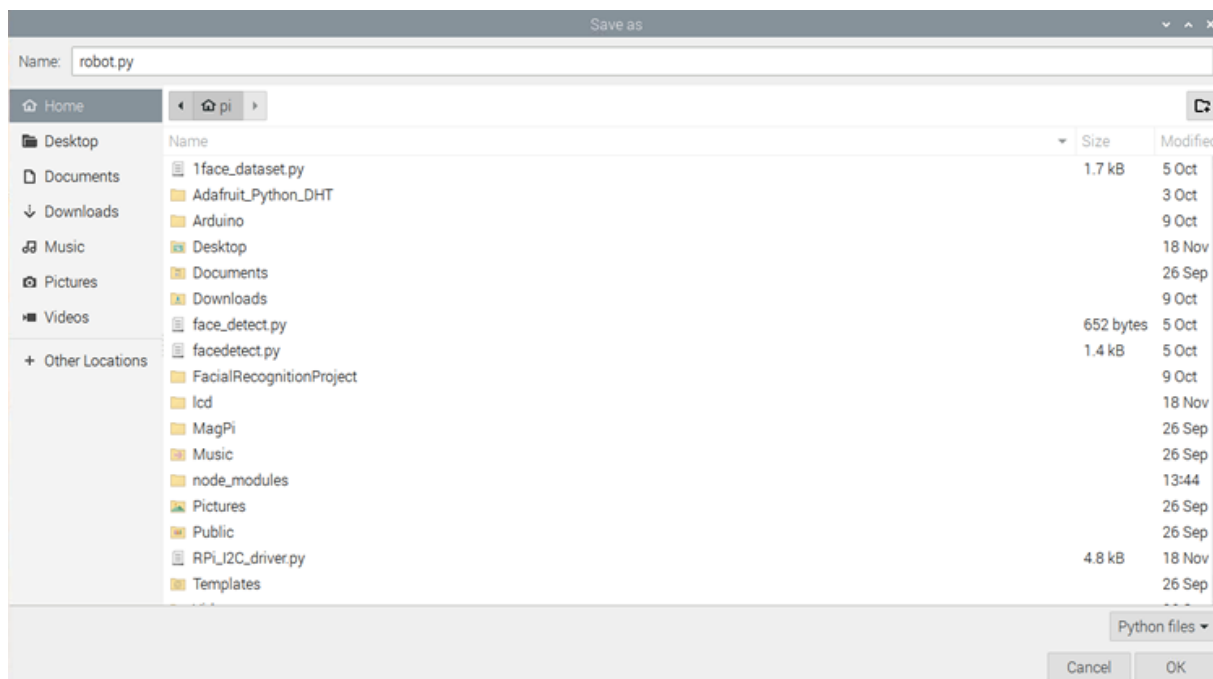
1. เปิดโปรแกรม Python โดยคลิกที่สัญลักษณ์ Raspberry Pi ที่อยู่บนด้านบนซ้ายของหน้าจอ แล้วเลือกเมนู Programming เลือกโปรแกรม Python 3 (IDLE) หรือโปรแกรม Thonny Python IDE (ในที่นี้จะเลือกใช้โปรแกรม Thonny Python IDE)



2. เขียนโค้ดคำสั่ง ดังนี้

```
robot.py ✕
1 import RPi.GPIO as GPIO
2 import time
3 GPIO.setwarnings(False)
4
5 GPIO.setmode(GPIO.BOARD)
6 GPIO.setup(31, GPIO.OUT)
7 GPIO.setup(33, GPIO.OUT)
8 GPIO.setup(35, GPIO.OUT)
9 GPIO.setup(37, GPIO.OUT)
10
11 GPIO.setup(16, GPIO.OUT)
12 GPIO.setup(18, GPIO.IN)
13
14 while True:
15     GPIO.output(16,1)
16     time.sleep(0.00001)
17     GPIO.output(16,0)
18
19     while GPIO.input(18) == 0:
20         timestart = time.time()
21
22     while GPIO.input(18) == 1:
23         timestop = time.time()
24
25
26     timetotal = timestop - timestart
27     distance = round(timetotal * 34600 * 0.5,2)
28     print(distance)
29
30     if distance > 20.00:
31         #Forward
32         GPIO.output(31,1)
33         GPIO.output(33,0)
34         GPIO.output(35,0)
35         GPIO.output(37,1)
36         time.sleep(0.1)
37     else :
38         #Stop
39         GPIO.output(31,0)
40         GPIO.output(33,0)
41         GPIO.output(35,0)
42         GPIO.output(37,0)
43         time.sleep(0.1)
44         #Turn Right
45         GPIO.output(31,1)
46         GPIO.output(33,0)
47         GPIO.output(35,0)
48         GPIO.output(37,0)
49         time.sleep(0.5)
```

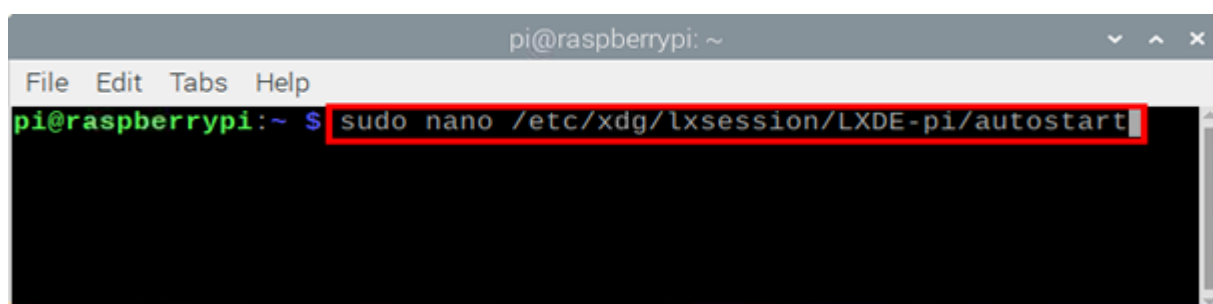

3. เมื่อเขียนโค้ดคำสั่งเสร็จแล้ว ให้คลิกปุ่ม Run เพื่อทดสอบโปรแกรม ระบบจะให้ทำการบันทึกไฟล์ .py ในที่นี้จะตั้งชื่อไฟล์ว่า robot.py โดยโฟลเดอร์เริ่มต้นในการบันทึกไฟล์จะอยู่ที่ /home/pi/ และหากต้องการหยุดการรันโปรแกรม ให้คลิกปุ่ม Stop หากรถหุ่นยนต์ยังไม่หยุดหมุน ให้ปิดสวิตช์บนตัวรถหุ่นยนต์



4. หากต้องการให้รถหุ่นยนต์เคลื่อนที่ตามโค้ดคำสั่งโดยอัตโนมัติ เมื่อเปิดบอร์ด Raspberry Pi (Autostart) จะต้องเข้าไปตั้งค่าดังนี้

4.1 เปิดโปรแกรม Terminal แล้วพิมพ์คำสั่งดังนี้

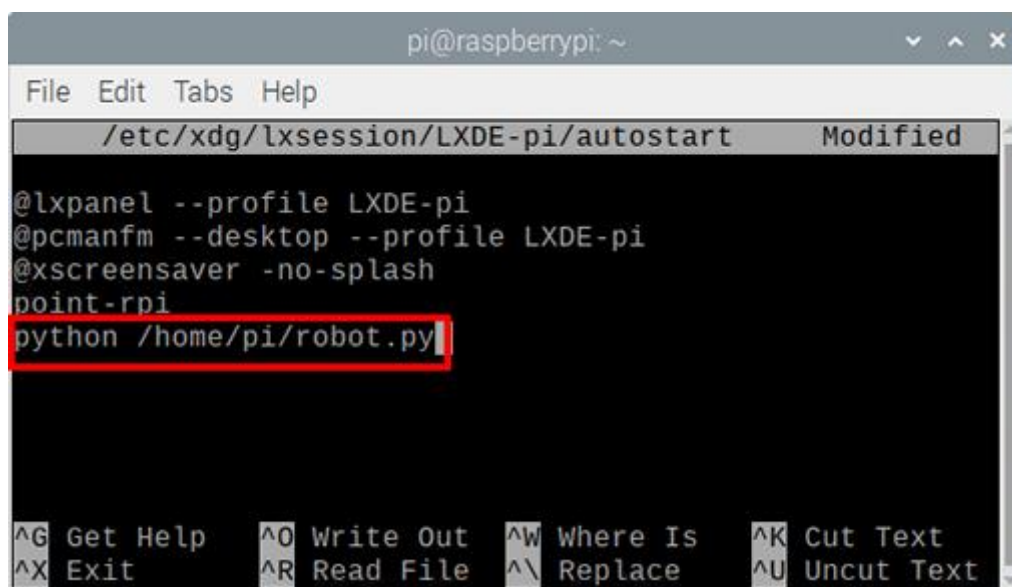
`sudo nano /etc/xdg/lxsession/LXDE-pi/autostart`



4.2 เลื่อนมาที่บรรทัดสุดท้าย เพิ่มคำสั่งในการรัน โปรแกรม ดังนี้

`python /home/pi/ชื่อไฟล์.py`

ในที่นี้จะต้องพิมพ์คำสั่ง `python /home/pi/robot.py` เป็นต้น



```

pi@raspberrypi: ~
File Edit Tabs Help
/etc/xdg/lxsession/LXDE-pi/autostart Modified
@lxpanel --profile LXDE-pi
@pcmanfm --desktop --profile LXDE-pi
@xscreensaver -no-splash
point-rpi
python /home/pi/robot.py
^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text

```

4.3 ทำการบันทึก โดยกดปุ่ม Ctrl + X ระบบจะถามว่าต้องการบันทึกการเปลี่ยนแปลงหรือไม่ ให้พิมพ์ Y แล้วกด Enter ออกจากหน้าจอ โดยกด Enter อีกครั้ง

5. Reboot บอร์ด Raspberry Pi โดยคลิกที่สัญลักษณ์ Raspberry แล้วคลิกเมนู Shutdown แล้วคลิกเลือกคำสั่ง Reboot รอสักครู่เมื่อบอร์ด Raspberry เปิดขึ้นมาอีกครั้ง รถหุ่นยนต์จะเคลื่อนที่ตามโค้ดคำสั่งในไฟล์ .py ที่ได้ทำการตั้งค่า autostart ขึ้นโดยอัตโนมัติ