Ultrasonic sensor

จัดทำโดย

นายชินวัตร ศรีดามาตย์ 58011221027

นายบัญชา โทแก้ว 58011211081

นายปริวัฒน์ จันทร์สว่าง 58011211088

นางสาวพิชญาพร พูลสุข 58011211095

นายวรวุฒิ สารวงษ์ 58011211117

เสนอ

ผศ.ดร.โอฬาริก สุรินต๊ะ

วิชาการใช้งานคำสั่งลินุกซ์และการเขียนโปรแกรมด้วยเชลล์สคริปต์

(Linux Command Line and Shell Scripting)

สาขาเทคโนโลยีสารสนเทศ คณะวิทยาการสารสนเทศ

มหาวิทยาลัยมหาสารคาม

คำนำ

รายงานเล่มนี้เป็นส่วนหนึ่งของรายวิชาการใช้งานคำสั่งลินุกซ์และการเขียนโปรแกรมด้วยเชลล์สคริปต์ (Linux Command Line and Shell Scripting) รายงานเล่มนี้มีวัตถุประสงค์เพื่อการศึกษาวิธีการติดตั้งโปรแกรมขั้นต้น และการใช้งานของ Ultrasonic Sensor ทำให้เห็นผลลัพธ์อย่างไร และสามารถนำไปใช้ประโยชน์อย่างไรได้บ้าง

สุดท้ายนี้ ผู้จัดทำรายงายเล่มนี้หวังเป็นอย่างยิ่งว่าจะเกิดประโยชน์และความรู้ และหากพบเห็นข้อผิดพลาดประการใดขออภัยมา ณ ที่นี้

จัดทำโดย

คณะผู้จัดทำ

Ultrasonic sensor

1. Ultrasonic sensor

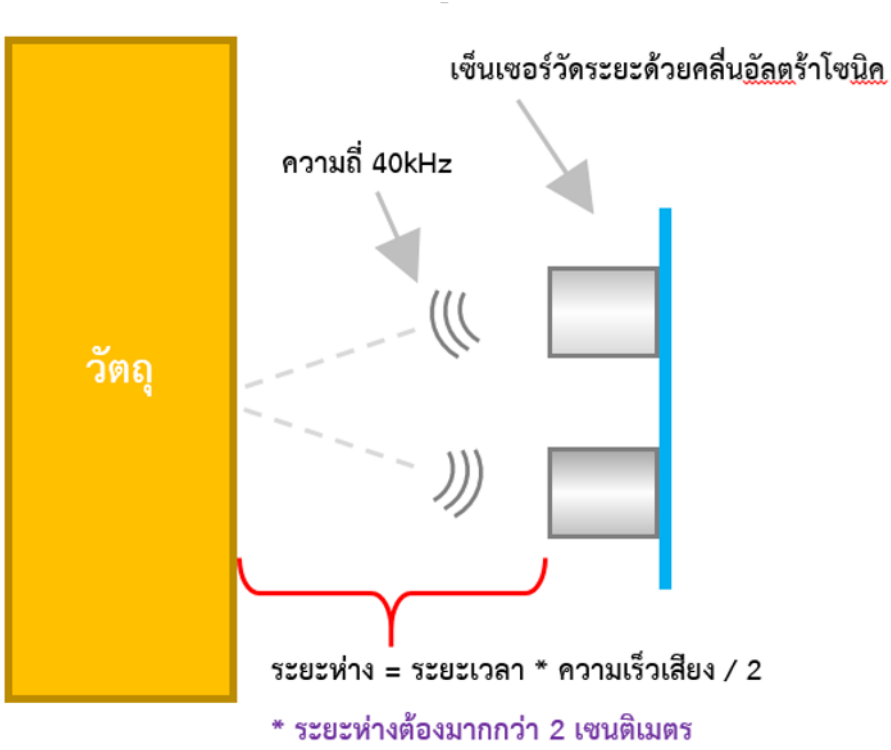
Ultrasonic sensor คือ คลื่นความถี่เสียงในช่วง Ultrasound ซึ่งเป็นคลื่นความถี่เสียงที่มนุษย์ไม่สามารถได้ยิน โดยมีย่านความถี่ตั้งแต่ 20 KHz ขึ้นไป ซึ่งข้อดีของการใช้ Ultrasonic Sensor ในการตรวจจับวัตถุนั้น คือ เรื่องของการเดินทางของคลื่น Ultrasound ที่สามารถเดินทางผ่านตัวกลางเช่น อากาศ ก๊าซ ของเหลว หรือ ของแข็งได้ ยกเว้นในสภาวะสูญญากาศ ทำให้สามารถใช้งานตรวจจับวัตถุได้หลากหลาย และสภาพแวดล้อมต่างๆ ได้ดี

คลื่นเสียงย่านอัลตร้าโซนิค นั้นเป็นคลื่นที่มีทิศทางที่แน่นอน ทำให้เราสามารถนำไปใช้งานได้หลายอย่าง เช่น นำไปใช้ในเครื่องควบคุมระยะไกล (Ultrasonic remote control) เครื่องล้างอุปกรณ์ (Ultrasonic cleaner) โดยการทำให้น้ำสั่นที่ความถี่สูง เครื่องวัดความหนาของวัตถุโดยสังเกตระยะเวลาที่คลื่นสะท้อนกลับมา เครื่องวัดความลึกและทำแผนที่ใต้ท้องทะเล โดยความถี่ที่นำมาใช้งานนั้นจะขึ้นอยู่กับตัวกลาง เช่น ถ้าคลื่นเสียงที่ต้องเดินทางผ่านอากาศความถี่ที่ใช้ก็มักจะจำกัดอยู่เพียงไม่เกิน 50 KHz เพราะที่ความถี่สูงขึ้นกว่านี้อากาศจะดูดกลืนคลื่นเสียงเพิ่มขึ้นมาก ทำให้ระดับความแรงของคลื่นเสียงที่ระยะห่างออกไปลดลงอย่างรวดเร็ว ส่วนการใช้งานด้านการแพทย์ซึ่งต้องการรัศมีทำการสั้น ๆ ก็อาจใช้ความถี่ในช่วง 1 MHz ถึง 10 MHz

รูปภาพที่ 1 อุปกรณ์วัดระยะทาง (Ultrasonic sensor)

1.1 หลักการวัดระยะด้วยคลื่นอัลตร้าโซนิค

หลักการที่สำคัญของการวัดระยะด้วยคลื่นอัลตร้าโซนิค คือการส่งคลื่นอัลตร้าโซนิคจำนวนหนึ่งออกไปจากตัวส่ง (Transmitter) เมื่อคลื่นวิ่งไปชนกับวัตถุ คลื่นจะมีการสะท้อนกลับมา แล้ววิ่งกลับไปชนตัวรับ (Receiver) ด้วยการเริ่มนับเวลาที่ส่งคลื่นออกไป จนถึงได้รับคลื่นกลับมานี้เอง ทำให้เราสามารถหาระยะห่างระหว่างวัตถุกับเซ็นเซอร์ได้



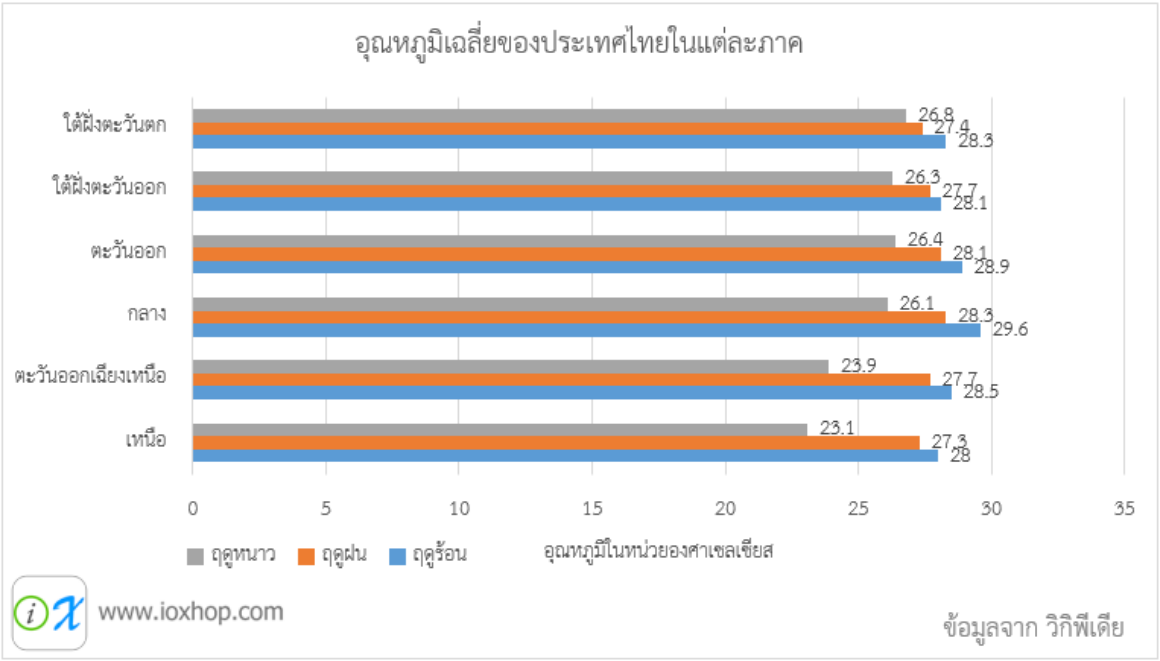
รูปที่ 2 การทำงานของเซ็นเซอร์วัดระยะด้วยคลื่นอัลตร้าโซนิค

ระยะเวลาที่ได้จากการวัดช่วงเวลาการเดินทางไปและกลับนี้ เราสามารถนำค่าที่ได้ไปเปรียบเทียบกับอัตราเร็วที่เสียงสามารถเดินทางได้ไปในอากาศได้เลย โดยอัตราเร็วเสียงที่เดินทางได้ในอากาศสามารถหาได้ตามสูตร

อัตราเร็วของเสียงในอากาศ = 331 + (0.606 * อุณหภูมิในหน่วยองศาเซลเซียส) m/s

สังเกตุว่า อัตราเร็วของเสียงที่เดินทางในอากาศนั้น จะขึ้นอยู่กับอุณหภูมิ ณ ขณะนั้นด้วย ดังนั้นในเซ็นเซอร์อัลตร้าโซนิคบางรุ่น จึงมีเซ็นเซอร์วัดอุณหภูมิมาด้วย ทำให้สามารถวัดระยะทางได้แม่นยำมากยิ่งขึ้น สำหรับในรุ่นที่ไม่มีเซ็นเซอร์วัดอุณหภูมิ ท่านสามารถนำเซ็นเซอร์วัดอุณหภูมิมาต่อเพื่อแก้ค่าความผิดพลาดเอง

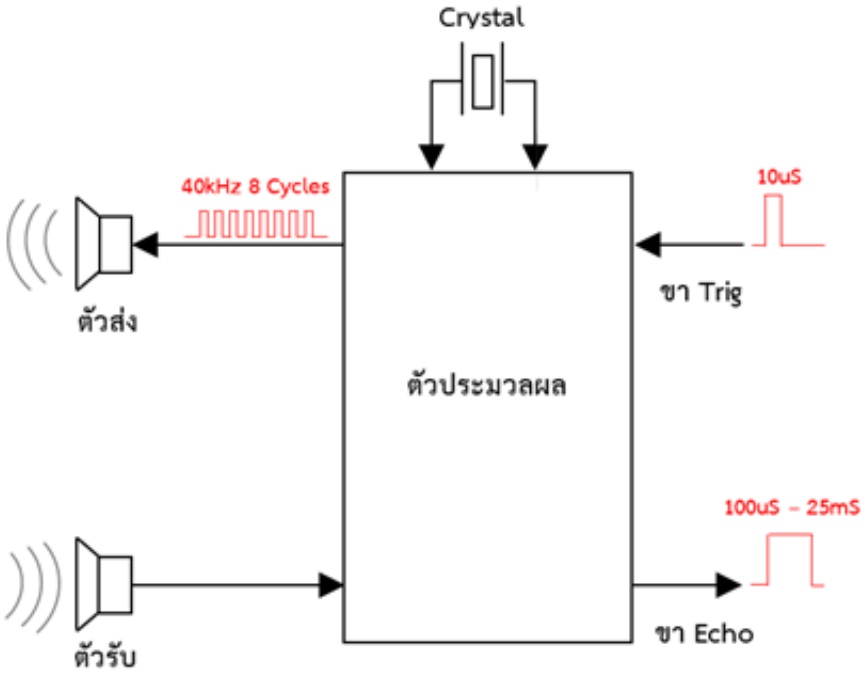
ได้ หรือใช้ค่าอุณหภูมิเฉลี่ยทั้งปีของประเทศไทยได้ โดยอุณหภูมิเฉลี่ยของประเทศไทยทั้งปีจะอยู่ที่ 27 องศาเซลเซียส



รูปที่ 3 กราฟแสดงค่าอุณหภูมิเฉลี่ยของประเทศไทยในแต่ละภาค

1.2 หลักการทำงานของเซ็นเซอร์วัดระยะด้วยคลื่นอัลตร้าโซนิค

ในโมดูลเซ็นเซอร์อัลตร้าโซนิคนั้น จะมีวงจรที่แตกต่างกัน เนื่องจากแต่ละรุ่นมีความสามารถที่แตกต่างกัน แต่ยังคงมีหลักการทำงานงานที่สำคัญที่เหมือนกัน



รูปที่ 4 ไดอะแกรมการทำงานของเซ็นเซอร์วัดระยะด้วยคลื่นอัลตร้าโซนิค

จากรูปที่ 4 จะเห็นว่า เมื่อมีการส่งสัญญาณเข้าไปที่ Trig วงจรภายในจะเริ่มสร้างความถี่ 40kHz จำนวน 8 ลูกคลื่นออกไป โดยใช้ความถี่จากคลิสตอลเป็นตัวอ้างอิง แล้วตัวส่งที่เปรียบเสมือนลำโพง จะส่งสัญญาณออกไป จากนั้นเมื่อคลื่นวิ่งกลับมาที่ตัวรับ ที่เปรียบเสมือนเป็นไมโครโฟน สัญญาณไฟฟ้าจะผ่านตัวประมวลผล แล้วให้ค่าเอาต์พุตออกมาทางขา Echo จะเห็นว่า แกนหลักของเซ็นเซอร์จะเป็นตัวประมวลผล ซึ่งตัวประมวนผลนี้ ในแต่ละรุ่นก็จะแตกต่างกัน อย่างในรุ่น HC-SR04 จะใช้ไอซีไมโครคอนโทรลเลอร์เบอร์ ATtiny24 ในรุ่น US-100 ก็จะใช้ไมโครคอนโทรลเลอร์แบบเดียวกัน แต่ไม่ทราบเบอร์ เนื่องจากเบอร์ไอซีบนโมดูลโดนลบ

1.3 หลักการใช้งานเซ็นเซอร์วัดระยะด้วยคลื่นอัลตร้าโซนิค

หลักการใช้งานจะแบ่งออกได้หลายแบบ ซึ่งจะขึ้นอยู่กับบัสที่ใช้สื่อสาร สามารถแบ่งได้ดังนี้

การทริกสัญญาณ

เซ็นเซอร์หลายรุ่น ใช้วิธีนี้ในการติดต่อสื่อสารกับไมโครคอนโทรลเลอร์ ซึ่งในแต่ละรุ่นจะใช้จำนวนสายไม่เท่ากัน ในบางรุ่นจะใช้สาย 2 เส้น คือ Trig สำหรับส่งสัญญาณ และ Echo สำหรับรับสัญญาณกลับมา และในบางรุ่นจะใช้เส้นเดียว คือทั้ง Trig และ Echo อยู่เส้นเดียวกันเลย และใช้วิธีแบ่งเวลารับ - ส่งข้อมูล (หลักการเหมือน 1-wire bus)

ในการสื่อสารแบบทริกสัญญาณ เริ่มต้นจะต้องให้สัญญาณขา Trig มีสถานะทางลอจิกเป็น LOW เสียก่อน จากนั้นจึงเริ่มทริกสัญญาณ โดยให้ขา Trig มีสถานะเป็น HIGH ค้างไว้อย่างน้อย 10uS แล้วจึงปรับสถานะเป็น LOW

จากนั้น ที่ขา Echo ให้เตรียมรับสัญญาณทริก HIGH กลับมา เมื่อมีการส่งสัญญาณ HIGH กลับมา ให้เริ่มนับเวลาที่สัญญาณเป็น HIGH และเมื่อสัญญาณขา Echo กลับเป็น LOW ให้สิ้นสุดการนับเวลา แล้วจึงนำค่าเวลาที่นับได้ ไปคำนวณอีกที ซึ่งในการคำนวณนั้น จะขึ้นอยู่กับรุ่น ในบางรุ่นสามารถใช้ค่าอัตราเร็วเสียงมาคำนวณได้เลย แต่ในบางรุ่น ต้องใช้สูตรคำนวณเฉพาะ

2. **Internet of Thing (IoT)**

Internet of Things (IoT) คือ "อินเตอร์เน็ตในทุกสรรพสิ่ง" หมายถึง การที่อุปกรณ์ต่างๆ สิ่งต่างๆ ได้ถูกเชื่อมโยงทุกสิ่งทุกอย่างสู่โลกอินเตอร์เน็ต ทำให้มนุษย์สามารถสั่งการควบคุมการใช้งานอุปกรณ์ต่างๆ ผ่านทางเครือข่ายอินเตอร์เน็ต เช่น การเปิด-ปิด อุปกรณ์เครื่องใช้ไฟฟ้า (การสั่งการเปิดไฟฟ้าภายในบ้านด้วยการเชื่อมต่ออุปกรณ์ควบคุม เช่น มือถือ ผ่านทางอินเตอร์เน็ต) รถยนต์ โทรศัพท์มือถือ เครื่องมือสื่อสาร เครื่องมือทางการเกษตร อาคาร บ้านเรือน เครื่องใช้ในชีวิตประจำวันต่างๆ ผ่านเครือข่ายอินเตอร์เน็ต เป็นต้น

รูปภาพที่ 5 Internet of Thing (IoT)

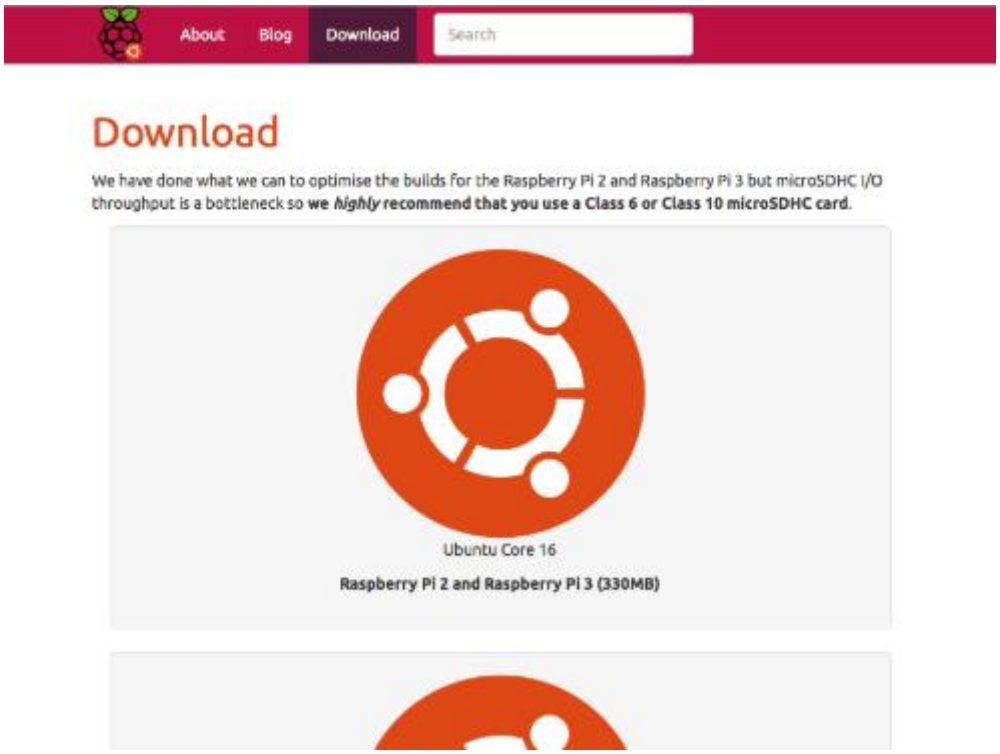
IoT มีชื่อเรียกอีกอย่างว่า M2M ย่อมาจาก Machine to Machine คือเทคโนโลยีอินเตอร์เน็ตที่เชื่อมต่ออุปกรณ์กับเครื่องมือต่างๆ เข้าไว้ด้วยกัน

เทคโนโลยี IoT มีความจำเป็นต้องทำงานร่วมกับอุปกรณ์ประเภท RFID และ Sensors ซึ่งเปรียบเสมือนการเติมสมองให้กับอุปกรณ์ต่างๆ ที่ขาดไม่คือการเชื่อมต่ออินเตอร์เน็ต เพื่อให้อุปกรณ์สามารถรับส่งข้อมูลถึงกันได้ เทคโนโลยี IoT มีประโยชน์ในหลายด้าน แต่ก็มาพร้อมกับความเสี่ยง เพราะหากระบบรักษาความปลอดภัยของอุปกรณ์ และเครือข่ายอินเตอร์เน็ตไม่ดีพอ ก็อาจทำให้มีผู้ไม่ประสงค์ดีเข้ามาขโมยข้อมูลหรือละเมิดความเป็นส่วนตัวของเราได้ ดังนั้นการพัฒนา IoT จึงจำเป็นต้องพัฒนามาตรการ และระบบรักษาความปลอดภัยไอทีควบคู่กันไปด้วย

3. ตัวอย่างการประยุกต์ใช้ Ultrasonic Sensor กับ Internet of Thing (IoT)

3.1 การติดตั้ง Raspberry pi

- ก่อนที่ Ubuntu จะใช้กับ Raspberry pi ได้ จะต้องสามารถทำงานได้บน ARM processor



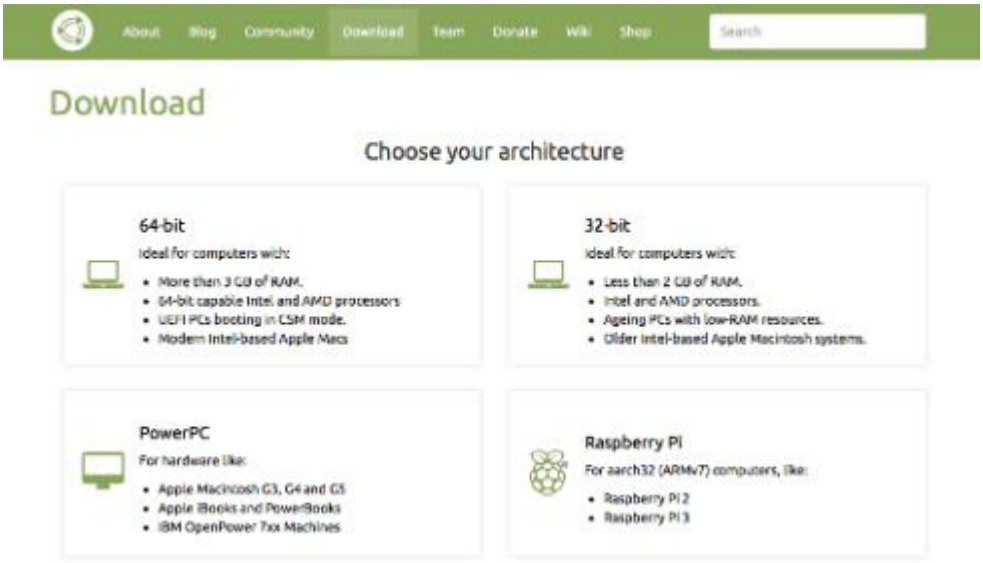
รูปภาพที่ 6 หน้าเว็บดาวน์โหลด Raspberry pi

กรณีการใช้ Ubuntu Mate สำหรับการติดตั้งลงบน Raspberry pi

- เข้าไปดาวน์โหลด Ubuntu Mate

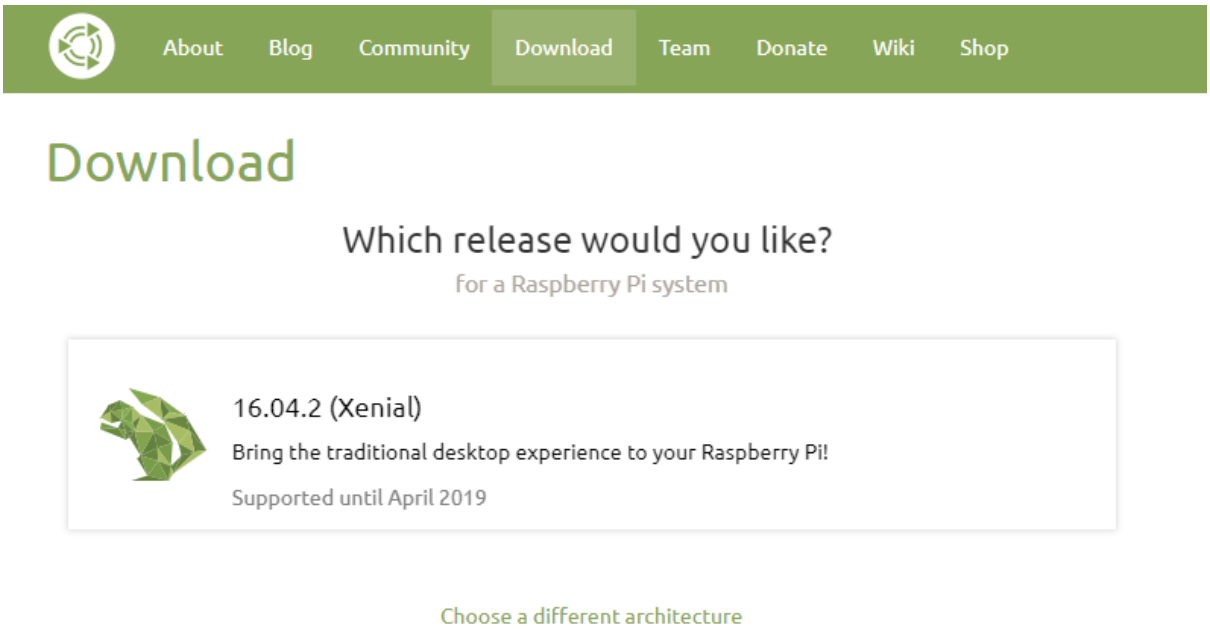
รูปภาพที่ 7 หน้าดาวน์โหลด Ubuntu Mate

- ไปที่แทบเมนูด้านบน แล้วเลือกดาวน์โหลด (Download) คลิกและเลือกไปที่ Raspberry Pi



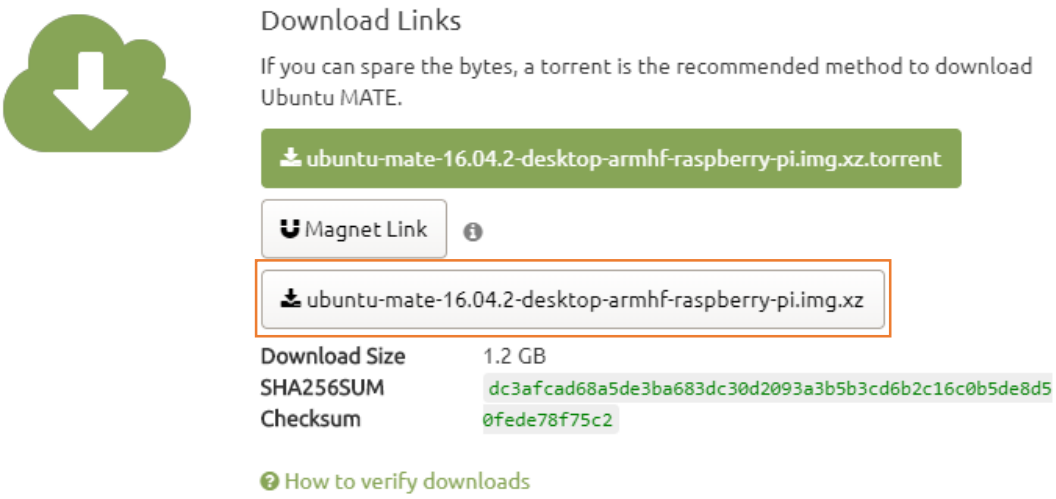
รูปภาพที่ 8 เลือกคลิกที่ Raspberry Pi

- เมื่อคลิกเข้าไปที่ Raspberry Pi จะเจอ Raspberry Pi เวอร์ชั่น 16.04.2 (Xenial) และคลิกที่ 16.04.2 (Xenial) เพื่อเข้าสู่หน้าดาวน์โหลด



รูปภาพที่ 9 เลือกคลิกที่ 16.04.2 (Xenial)

- เมื่อคลิกเข้ามาจะเจอหน้าดาวน์โหลด ให้สังเกตไปที่ Download Link คลิกที่ปุ่ม Ubuntu-mate-16.04.0-desktop-armhf-raspberry-pi.img.xz ที่เป็นปุ่มกดสีขาว

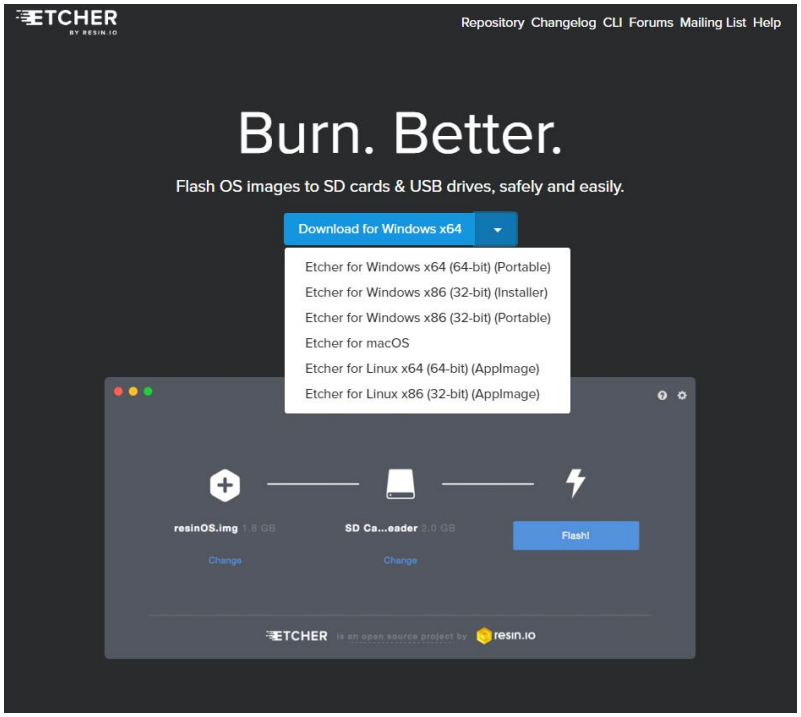


รูปภาพที่ 10 คลิกลิ้งค์ดาวน์โหลด

3.2 การติดตั้ง Etcher โปรแกรมสำหรับ Flash หรือ Burn ไฟล์อิมเมจของระบบปฏิบัติการ (OS image) ลง SD Card หรือ USB Drive

การดาวน์โหลดโปรแกรม Etcher

- เข้าไปดาวน์โหลดที่เว็บไซต์ https://etcher.io จากนั้นให้คลิกที่ปุ่มดาวน์โหลด (ปุ่มสีฟ้า) ซึ่งจะมีให้เลือกว่าจะดาวน์โหลดแบบ MacOS, Linux และ Windows ในกรณีนี้เราเลือกดาวน์โหลดแบบ MacOS



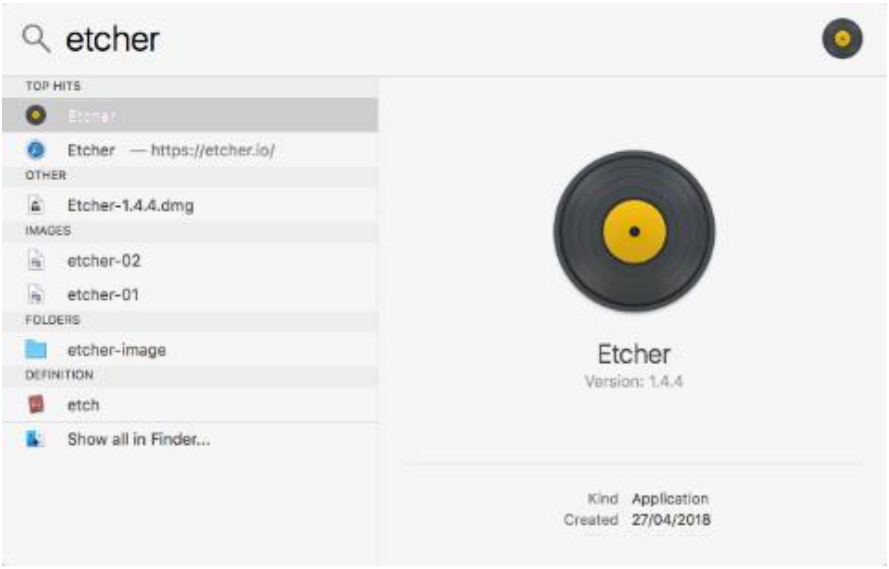
รูปที่ 11 หน้าดาวน์โหลดโปรแกรม Etcher

การติดตั้งโปรแกรม Etcher

- เมื่อดาวน์โหลดเสร็จแล้วให้เปิดหน้าโปรแกรม จะได้โปรแกรมดังนี้

รูปภาพที่ 11 หน้าโปรแกรม Etcher

- ให้ทำการเลือกคลิกและลากที่รูปไอคอน Etcher จากทางด้านซ้ายมือ ไปวางไว้ในรูปไอคอน Applications ทางด้านขวามือ ระบบจะทำการคัดลอกไฟล์ไปไว้ยัง Applications
- เมื่อเราทำการคัดลอกไฟล์ไปไว้ที่ Applications เรียบร้อยแล้ว จากนั้นสามารถเปิด Finder และเข้าไปที่ Applications และทำการค้นหาหน้าโปรแกรม Etcher เพื่อเรียกใช้โปรแกรม



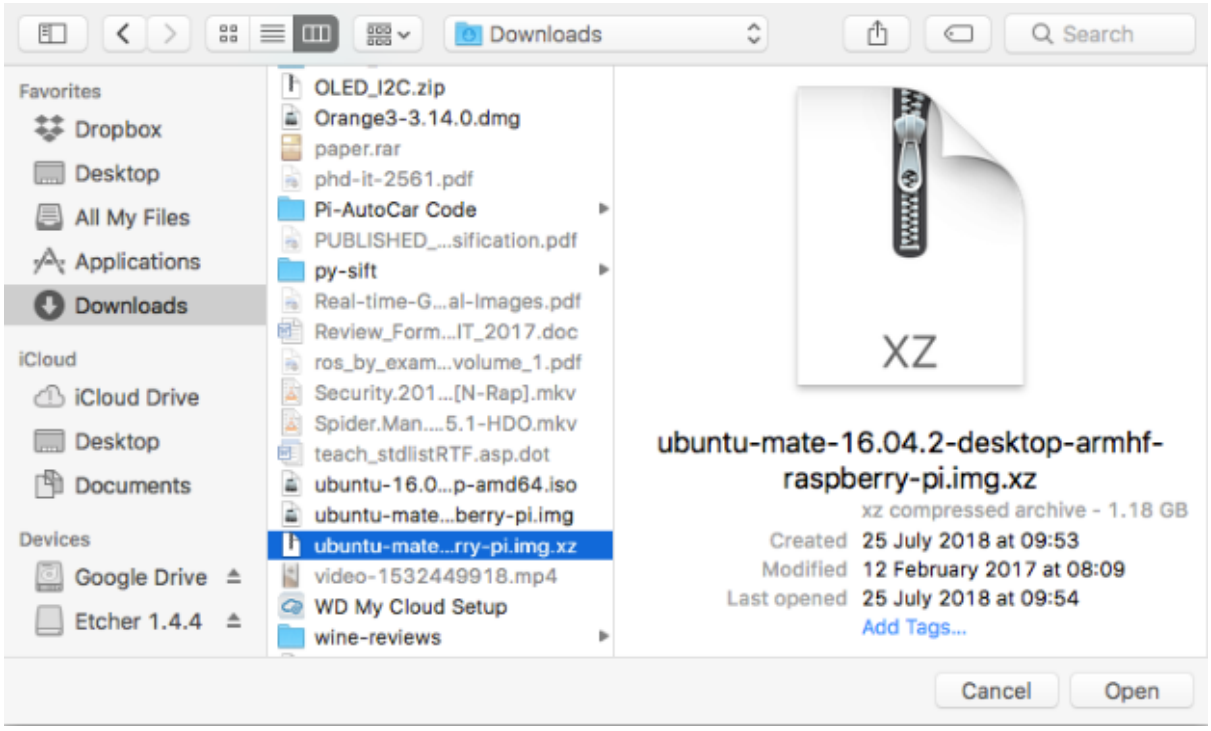
รูปภาพที่ 12 เมื่อคัดลอกไฟล์มาไว้ที่ Application

การใช้งานโปรแกรม Etcher เพื่อ Flash ระบบปฏิบัติการลงไปยัง SD Card

- หน้าจอโปรแกรม Etcher ที่จะแสดงต่อไปนี้

รูปภาพที่ 13 การแสดงหน้าจอโปรแกรม Etcher

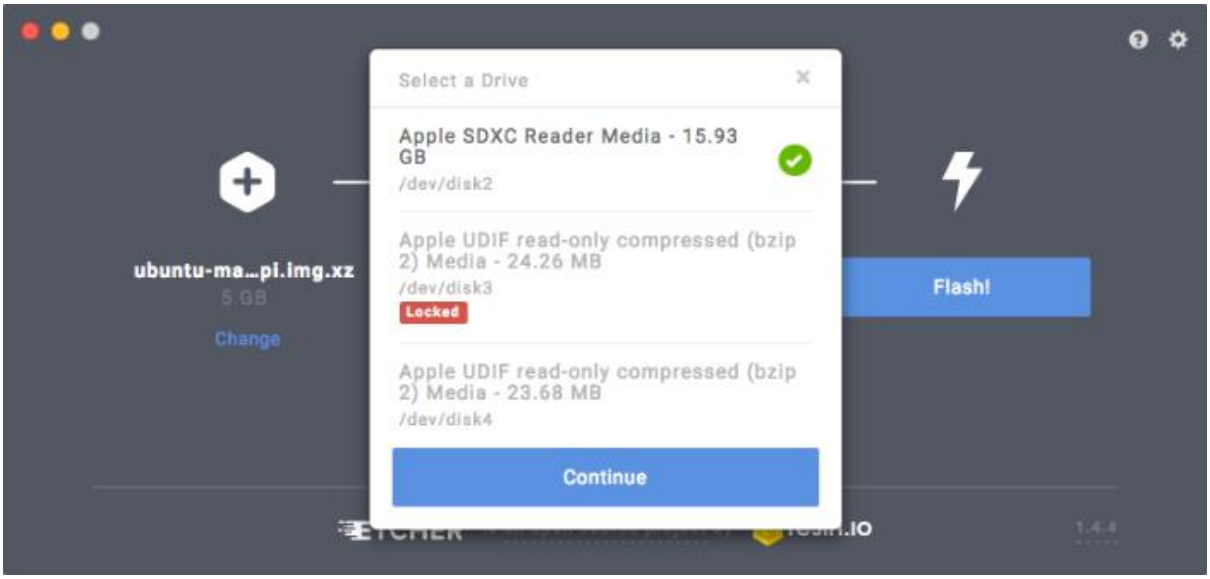
- เมื่อทำการดาวน์โหลด Ubuntu Mate ใน Raspberry Pi เสร็จเรียบร้อยแล้ว จากนั้นเปิดโปรแกรม Etcher และคลิกที่ Select Image (ปุ่มสีฟ้า) จะมีหน้าต่างขึ้นมาให้เลือกไฟล์อิมเมจที่ต้องการ



รูปภาพที่ 14 เลือกไฟล์อิมเมจ

โดยเราจะเลือกไฟล์ Ubuntu-mate-16.04.0-desktop-armhf-raspberry-pi.img.xz ที่เราดาวน์โหลดมา เมื่อกด Open จะกลับไปสู่หน้าจอหลักของโปรแกรม Etcher เหมือนเดิม

- เมื่อกลับมาที่หน้าหลักโปรแกรม Etcher เลือกกดที่ Select Drive และเลือก Drive ที่เป็น SD Card ที่นำมาใช้ หาก SD Card มีขนาดไม่เพียงพอระบบจะไม่แสดงข้อความเพื่อแจ้งเตือน ดังนั้นให้เปลี่ยน SD Card ที่มีขนาดที่เพียงพอ
- เมื่อเสียบ SD Card ที่มีขนาดเพียงพอจะแสดงตามภาพด้านล่างนี้ และให้คลิกที่ SD Card และกดที่ปุ่ม Continue



รูปภาพที่ 15 การเชื่อม SD Card ที่มีขนาดเพียงพอ

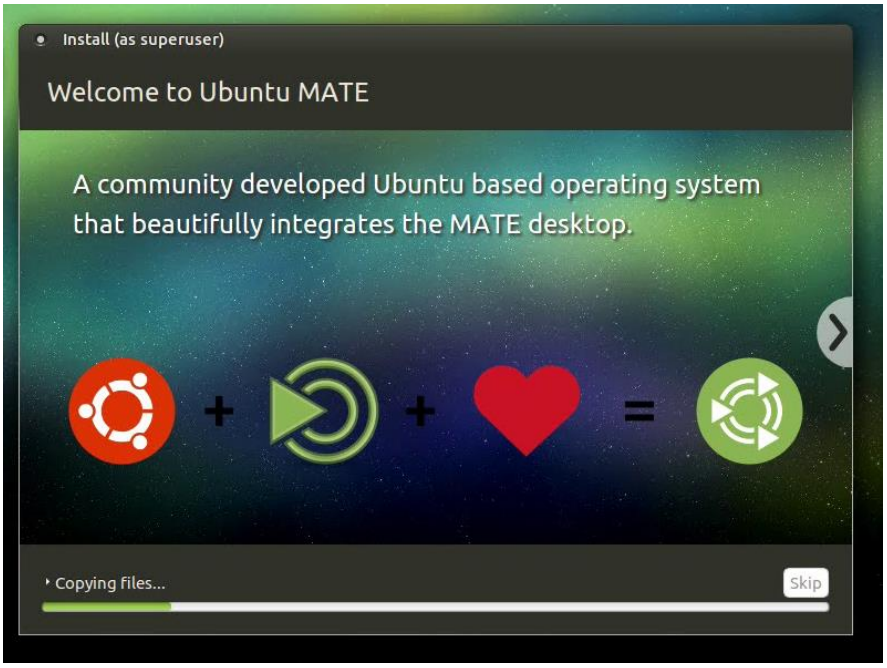
- เมื่อกดไปที่ปุ่ม Continue แล้ว จากนั้นก็จะกลับมาที่หน้าหลักโปรแกรมและคลิกที่ปุ่มสุดท้ายคือ Flash เพื่อทำการ Flash หรือ Burn ระบบจะทำการ Ubuntu Mate ลงบน SD Card

รูปภาพที่ 16 การ Flash ลง SD Card

- เมื่อทำการ Flash เรียบร้อยแล้ว จากนั้นสามารถนำ SD Card ที่ Flash เรียบร้อยแล้ว ไปเสียบที่เครื่อง Raspberry Pi เพื่อทำการติดตั้งในขั้นตอนต่อไป

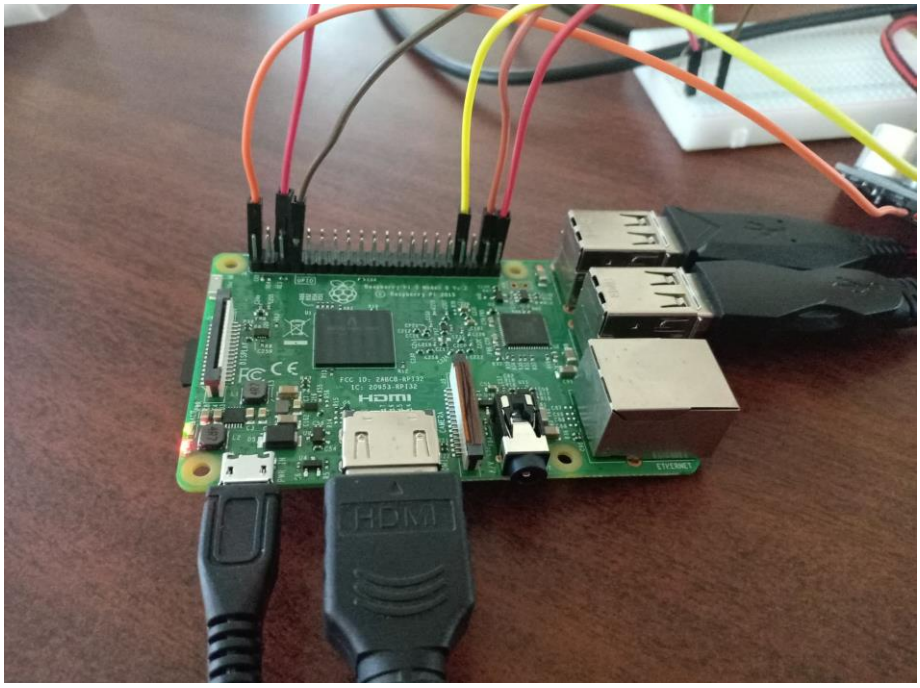
3.3 การติดตั้ง Ubuntu mate ใน Raspberry pi

- การติดตั้ง Ubuntu mate ใน Raspberry pi เพื่อจำลองให้ Raspberry pi เป็นเครื่องเซิร์ฟเวอร์



รูปภาพที่ 17 การติดตั้ง Ubuntu mate ใน Raspberry pi

- นำ SD Card ที่ทำการติดตั้ง Ubuntu Mate ด้วยโปรแกรม applePi-Baker มาเสียบที่ Raspberry Pi



รูปภาพที่ 18 การเชื่อมต่อ SD Card อุปกรณ์ Input

- เชื่อมต่อสาย HDMI จาก Raspberry Pi ไปยังหน้าจอคอมพิวเตอร์ หรือหน้าจอโทรทัศน์
- เชื่อมต่อ Mouse กับ Keyboard เข้ากับ USB ของ Raspberry Pi

รูปภาพที่ 19 การเชื่อมต่ออุปกรณ์ Input

- เสียบปลั๊กเข้าทางช่อง Micro USB เพื่อทำการจ่ายไฟไปยัง Raspberry Pi
- เมื่อติดตั้งเสร็จเรียบร้อยแล้ว ทำการเปิด Ubuntu Mate และเชื่อมต่อกับอินเทอร์เน็ต จากนั้นทำการ Update ระบบให้เป็นปัจจุบัน
- Code การอัพเดต Ubuntu Mate สามารถอัพเดตได้โดยการเปิด Terminal แล้วพิมพ์คำสั่ง

```
$ sudo apt-get update

$ sudo apt-get upgrade
```

- เมื่อติดตั้ง Ubuntu Mate เสร็จเรียบร้อยแล้วจะยังไม่สามารถ SSH เข้าไปเพื่อสั่งให้ Raspberry Pi ทำงานได้เลย จะต้องติดตั้ง OpenSSh Server ก่อน

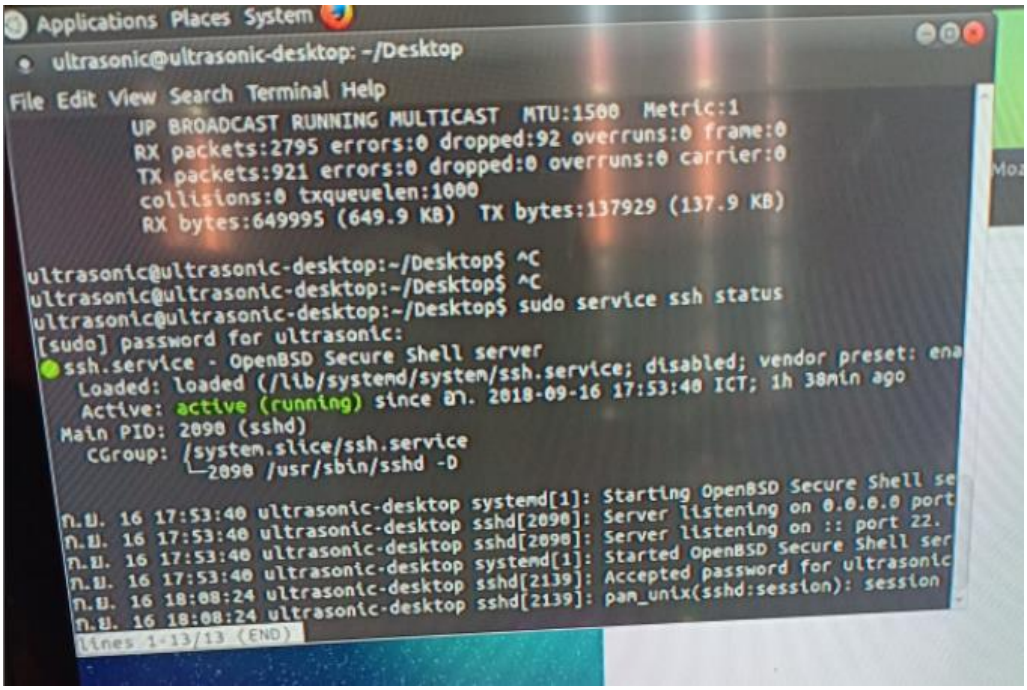
การติดตั้ง OpenSSH Server

- เชื่อมต่อกับอินเทอร์เน็ต เพราะการติดตั้งโปรแกรมจะต้องใช้อินเทอร์เน็ตทุกครั้ง
- เปิดหน้าโปรแกรม Terminal เพื่อพิมพ์คำสั่งต่อไปนี้ในการติดตั้งโปรแกรม OpenSSH Server

```
$ sudo apt-get install openssh-server
```

- เมื่อติดตั้งเสร็จเรียบร้อย ให้ทำการ Start server โดยพิมพ์คำสั่ง

```
$ sudo service ssh server
```



รูปภาพที่ 20 การ start server

การควบคุม Raspberry Pi ผ่าน ssh server

- ทำการค้นหา IP เครื่อง Raspberry Pi โดยพิมพ์คำสั่ง แล้วบันทึกไว้

```
$ ifconfig
```

- จากนั้นมาที่เครื่องที่เราต้องการ Connect ให้พิมพ์คำสั่ง

```
$ ssh <user>@ip address
```

User คือ ชื่อของเครื่อง Raspberry Pi

Ip คือ ip ของเครื่อง Raspberry Pi

ตัวอย่างโปรแกรมสำหรับ Ultrasonic Sensor

จากนั้นระบบจะทำการกรอกรหัสผ่าน หากถูกต้องก็สามารถเข้า Raspberry Pi ได้

ควบคุมการทำงาน

นำ code ด้านล่างไปใส่ file ที่จะรันคำสั่ง

```
import RPi.GPIO as GPIO

import time

import microgear.client as netpie

import base64, zlib, time


key = 'KblP2X5J0jHrgHm'

secret = 'uiH1aOoqb2ZpZkQdU1friztJn'

app = 'BangBangBang'
```

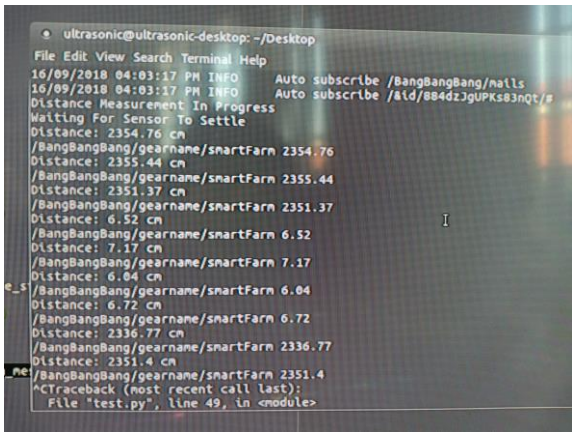
```
netpie.create(key,secret,app,{'debugmode': True})


def connection():
        print "Now I am connected with netpie"


def subscription(topic,message):
        print topic+" "+message


netpie.setname("smartFarm")
netpie.on_connect = connection
netpie.on_message = subscription
netpie.subscribe("/mails")
netpie.connect()
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)


TRIG = 21
ECHO = 20
led_pin = 4
print "Distance Measurement In Progress"
```

```
GPIO.setup(TRIG,GPIO.OUT)

GPIO.setup(ECHO,GPIO.IN)


GPIO.output(TRIG,False)

print "Waiting For Sensor To Settle"


time.sleep(2)

status =' '

while(True):

        GPIO.output(TRIG,True)

        time.sleep(0.00001)

        GPIO.output(TRIG,False)


        while GPIO.input(ECHO)==0:

                pulse_start = time.time()


        while GPIO.input(ECHO) ==1:

                pulse_end = time.time()


        pulse_duration = pulse_end - pulse_start


        distance = pulse_duration * 17150
```

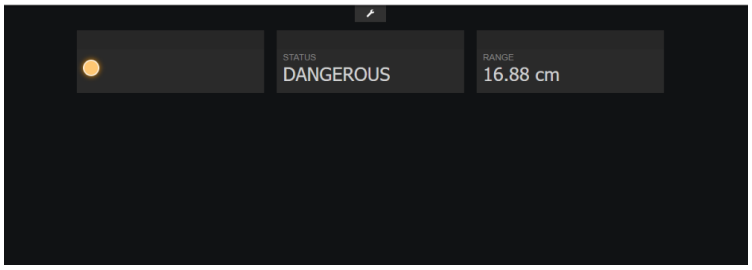
```
distance = round(distance, 2)


 if(distance < 10):
    GPIO.setup(led_pin,GPIO.OUT)
    GPIO.output(led_pin,GPIO.HIGH)
    status = 'VERY DANGEROUS'
   time.sleep(2)
elif(distance < 30):
    GPIO.setup(led_pin,GPIO.OUT)
    GPIO.output(led_pin,GPIO.HIGH)
    time.sleep(0.1)
    GPIO.output(led_pin,GPIO.LOW)
    status = 'DANGEROUS'
     time.sleep(0.1)
elif(distance < 100):
    GPIO.setup(led_pin,GPIO.OUT)
    GPIO.output(led_pin,GPIO.HIGH)
    time.sleep(0.7)
    GPIO.output(led_pin,GPIO.LOW)
    status = 'WARNING'
   time.sleep(0.7)
```

```
        else:
            GPIO.setup(led_pin,GPIO.OUT)
            GPIO.output(led_pin,GPIO.LOW)
            status = 'SAFE'
            time.sleep(2)




        smartFarm_message = str(distance) +","+ str(status)
        netpie.chat("smartFarm",smartFarm_message)


        print "Distance:", distance, "cm"
GPIO.cleanup()
```

เมื่อนำ code ข้างต้นไปใส่ ให้ใช้คำสั่ง python ตามด้วยชื่อ file.py รอดูผลการทำงาน



โดยโปรแกรมจะแสดงค่าระยะทางออกมาเป็นเซนติเมตร




```
import RPi.GPIO as GPIO

import time

import microgear.client as netpie

import base64, zlib, time
```

Code ส่วนนี้เป็นการ Import

```
key = 'KblP2X5J0jHrgHm'

secret = 'uiH1aOoqb2ZpZkQdU1friztJn'

app = 'BangBangBang'
```

Key,secret,app ใช้เพื่อจะติดต่อกับ Net pie ของเรา ขึ้นอยู่กับของแต่ละคนจะเป็นรหัสอะไร

```
netpie.create(key,secret,app,{'debugmode': True})

def connection():

        print "Now I am connected with netpie"

def subscription(topic,message):

        print topic+" "+message

netpie.setname("smartFarm")

netpie.on_connect = connection

netpie.on_message = subscription

netpie.subscribe("/mails")

netpie.connect()

GPIO.setmode(GPIO.BCM)

GPIO.setwarnings(False)
```

ตรงนี้เป็น Code การเชื่อมต่อของ Netpie

```
TRIG = 21

ECHO = 20

led_pin = 4

print "Distance Measurement In Progress"


GPIO.setup(TRIG,GPIO.OUT)

GPIO.setup(ECHO,GPIO.IN)


GPIO.output(TRIG,False)

print "Waiting For Sensor To Settle"


time.sleep(2)

status =' '
```

ตรงนี้เป็น Code set ตัวแปรต่างๆให้สอดคล้องกับ บอร์ด

```
while(True):
        GPIO.output(TRIG,True)
        time.sleep(0.00001)
        GPIO.output(TRIG,False)
while GPIO.input(ECHO)==0:
                pulse_start = time.time()
while GPIO.input(ECHO) ==1:
                pulse_end = time.time()


        pulse_duration = pulse_end - pulse_start


        distance = pulse_duration * 17150
        distance = round(distance, 2)
```

ตรงเป็นลูป การทำงานของ Sensor Ultrasonic หน่วยเป็น เซนติเมตร

```
 if(distance < 10):
    GPIO.setup(led_pin,GPIO.OUT)
    GPIO.output(led_pin,GPIO.HIGH)
    status = 'VERY DANGEROUS'
   time.sleep(2)
elif(distance < 30):
    GPIO.setup(led_pin,GPIO.OUT)
    GPIO.output(led_pin,GPIO.HIGH)
    time.sleep(0.1)
    GPIO.output(led_pin,GPIO.LOW)
    status = 'DANGEROUS'
     time.sleep(0.1)
 elif(distance < 100):
    GPIO.setup(led_pin,GPIO.OUT)
    GPIO.output(led_pin,GPIO.HIGH)
    time.sleep(0.7)
    GPIO.output(led_pin,GPIO.LOW)
    status = 'WARNING'
   time.sleep(0.7)
else:
    GPIO.setup(led_pin,GPIO.OUT)
    GPIO.output(led_pin,GPIO.LOW)
    status = 'SAFE'
    time.sleep(2)
```

จาก Code ข้างต้นเป็นการกำหนดเงื่อนไขให้ เกิดไฟกะพริบ

```
        smartFarm_message = str(distance) +","+ str(status)

        netpie.chat("smartFarm",smartFarm_message)



        print "Distance:", distance, "cm"

GPIO.cleanup()
```

ตรงนี้เป็นการส่งค่า ไปให้ Netpie