



ระบบแจ้งเตือนการบุกรุกบ้านโดยใช้อุปกรณ์โซลาร์เทียชีวันร่วมกับกล้องราสเบอร์รี่พาย

Notification Home Invasion System Using Zolertia Z1 with Raspberry Pi Camera

นาย ชรัตน์ จำแสง

โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีไทย-ญี่ปุ่น

พ.ศ. 2559

ระบบแจ้งเตือนการบุกรุกบ้านโดยใช้อุปกรณ์โซลาร์เทียชีวันร่วมกับกล้องราสเบอร์รี่พาย
Notification Home Invasion System Using Zolertia Z1 with Raspberry Pi Camera

นาย ชรัตน์ จำแสง

โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิทยาศาสตรบัณฑิต สาขาเทคโนโลยีสารสนเทศ
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีไทย - ญี่ปุ่น
ปีการศึกษา 2559

คณะกรรมการสอบ

..... ประธานกรรมการสอบ

(อาจารย์ ดร.ภาสกร อภิรักษ์วรพินิต)

..... กรรมการสอบ

(อาจารย์ อมรพันธ์ ชมกลีน)

..... อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์ ดร.อรรณพ หมั่นสกุล)

..... ประธานโครงการศึกษาสาขาวิชา

(อาจารย์ อมรพันธ์ ชมกลีน)

ลิขสิทธิ์ของสถาบันเทคโนโลยีไทย – ญี่ปุ่น

ชื่อโครงการ/รายงาน	ระบบแจ้งเตือนการบุกรุกบ้านโดยใช้อุปกรณ์โซลาร์เทียชวันร่วมกับกล้อง ราสเบอร์รี่พาย	
	Notification Home Invasion System Using Raspberry Pi Camera	
ผู้เขียน	นาย ชรัตน์ จำแสง	
คณะวิชา	เทคโนโลยีสารสนเทศ	สาขาวิชา เทคโนโลยีสารสนเทศ
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร.อรรณพ หมั่นสกุล	

งานที่ปฏิบัติ

การนำเซ็นเซอร์ตรวจจับการเคลื่อนไหวและกล้องมาประยุกต์ใช้ในบ้านเรือน เพื่อป้องกันการบุกรุกบ้านจากผู้ไม่หวังดี ในเวลาที่ไม่มีการอยู่บ้าน โดยตัวเซ็นเซอร์จะทำการตรวจจับเวลาที่มีสิ่งที่เคลื่อนไหวภายในห้อง จากนั้นจะส่งข้อมูลไปที่เว็บไซต์ยูบิคอต แล้วทำการแจ้งเตือนในรูปแบบของอีเมลและเอสเอ็มเอสไปยังมือถือของเจ้าของบ้าน หลังจากนั้นกล้องก็จะถ่ายรูป แล้วส่งไปยังคลาวด์แพลตฟอร์ม เพื่อที่จะสามารถระบุได้ว่าเป็นใครหรือสัตว์ชนิดไหน

ผลที่ได้รับจากการดำเนินงานและประโยชน์ที่ได้รับ

สามารถแจ้งเตือนผ่านมือถือได้ เวลาที่มีผู้ไม่ประสงค์ดีหรือสัตว์อื่นเข้ามาในบ้านในเวลาที่เราไม่ได้อยู่บ้าน และสามารถระบุตัวตนได้ผ่านรูปภาพที่ถูกถ่ายผ่านระบบคลาวด์

กิตติกรรมประกาศ

รายงานฉบับนี้สำเร็จลุล่วงไปด้วยดี ด้วยความช่วยเหลือของอาจารย์ ผศ. ดร.อรรณพ หมั่นสกุล อาจารย์ที่ปรึกษา ซึ่งท่านได้ให้คำแนะนำและข้อคิดเห็นต่าง ๆ อันเป็นประโยชน์อย่างยิ่งในการทำวิจัย อีกทั้งยังช่วยแก้ปัญหาต่าง ๆ ที่เกิดขึ้นระหว่างการดำเนินงานอีกด้วย และขอขอบคุณต่อ คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีไทย-ญี่ปุ่น ที่เอื้อเฟื้อสถานที่ที่ใช้ในการวิจัยให้สำเร็จลุล่วงไปได้ด้วยดี

สุดท้ายนี้ ผู้จัดทำขอขอบพระคุณบิดามารดา และครอบครัว ซึ่งเปิดโอกาสให้ได้รับการศึกษาเล่าเรียน ตลอดจนคอยช่วยเหลือและให้กำลังใจผู้วิจัยเสมอมาจนสำเร็จการศึกษา

นาย ชรัตน์ จำแสง
ผู้จัดทำ

TNI

สารบัญ

งานที่ปฏิบัติ	ข
ผลที่ได้รับจากการดำเนินงานและประโยชน์ที่ได้รับ.....	ข
กิตติกรรมประกาศ	ค
สารบัญ.....	ง
สารบัญรูป.....	ช
สารบัญตาราง.....	ฅ
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์การศึกษา	1
1.3 ขอบเขตของการศึกษา.....	1
1.4 นิยามศัพท์เฉพาะ	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2 ทฤษฎีและเทคโนโลยีที่ใช้ในการปฏิบัติงาน	3
2.1 อุปกรณ์ Zolertia Z1.....	3
2.2 อุปกรณ์ Raspberry Pi.....	5
2.3 เว็บไซต์ TunnelBroker.net	6
2.4 โพรโทคอลอินเทอร์เน็ต (Internet Protocol version 4, IPv4).....	6
2.5 โพรโทคอลอินเทอร์เน็ต (Internet Protocol version 6, IPv6)	7
2.5.1 แนวคิดของ IPv6.....	7
2.5.2 6LoWPAN (IPv6 over Low power Wireless Personal Area Networks).....	8
2.5.3 IPv6 Transition Mechanism (6in4).....	9
2.6 Ubidots Cloud Platform.....	10
2.7 กล้อง Raspberry Pi NoIR Camera V2	11
2.8 เซ็นเซอร์ตรวจจับความเคลื่อนไหว (PIR motion sensor).....	12
2.9 ระบบปฏิบัติการ Contiki OS	14
2.10 คลาวด์แพลตฟอร์ม Amazon Simple Storage Service (Amazon S3).....	16

2.11 ชุดพัฒนาซอฟต์แวร์ Boto3	17
บทที่ 3 แผนงานการปฏิบัติงานและขั้นตอนการดำเนินงาน	18
3.1 แผนการดำเนินงาน	18
3.1.1 ปรึกษากับอาจารย์ที่ปรึกษาเกี่ยวกับการปรับหัวข้อใหม่	19
3.1.2 ศึกษาการทำงานของอุปกรณ์ Zolertia Z1	19
3.1.3 ศึกษาการทำโมดูลไอพีรุ่นที่หก	19
3.1.4 ศึกษาการทำงานของเซ็นเซอร์	19
3.1.5 ศึกษาการทำงานของอุปกรณ์ราสเบอรี่พาย และกล้องราสเบอรี่พายอินฟราเรด	19
3.1.6 ออกแบบและทดสอบการทำงาน	19
3.1.7 ปรับปรุงและพัฒนาให้ดีขึ้น	20
3.1.8 ทดลองและบันทึกผล	20
3.1.9 จัดทำรูปเล่ม	20
3.2 ขั้นตอนการดำเนินงาน	20
3.2.1 การตั้งค่าบนอุปกรณ์ Zolertia Z1	21
3.2.2 การตั้งค่าบนอุปกรณ์ Raspberry Pi	24
3.2.3 การตั้งค่าบนเว็บไซต์ Ubidots	33
3.2.4 การตั้งค่าบนเว็บไซต์ Amazon Simple Storage Services	35
บทที่ 4 สรุปผลการดำเนินงาน การวิเคราะห์และสรุปผลต่าง ๆ	38
4.1 ผลการดำเนินงาน	38
4.1.1 การติดตั้งในสถานที่จริง	38
4.1.2 สามารถแจ้งเตือนผ่านอีเมลและข้อความมือถือ	40
4.1.3 สามารถอัปโหลดรูปถ่ายขึ้น Amazon Simple Storage Services	41
4.1.4 แสดงการถ่ายรูปของกล้อง Raspberry Pi NoIR Camera	43
บทที่ 5 บทสรุปและข้อเสนอแนะ	45
5.1 สรุปผลการดำเนินงาน	45
5.2 แนวทางการแก้ไขปัญหา	45
5.3 ข้อเสนอแนะ	45
เอกสารอ้างอิง	46

ประวัติผู้จัดทำโครงการ	48
------------------------------	----



สารบัญรูป

รูป

หน้า

รูปที่ 2.1 อุปกรณ์ Zolertia Z1	3
รูปที่ 2.2 อุปกรณ์ Raspberry Pi	5
รูปที่ 2.3 โลโก้เว็บไซต์ tunnelbroker.net	6
รูปที่ 2.4 สแต็คของอินเทอร์เน็ตโปรโตคอล	7
รูปที่ 2.5 Network Diagram 6in4 Tunneling	9
รูปที่ 2.6 โลโก้ Ubidots	10
รูปที่ 2.7 กล้อง Raspberry Pi NoIR Camera v2.....	11
รูปที่ 2.8 PIR Motion Sensor.....	12
รูปที่ 2.9 การทำงานของ PIR Motion Sensor	13
รูปที่ 2.10 การทำงานของ PIR motion sensor เมื่อมีรังสีอินฟราเรดเปลี่ยนแปลง	14
รูปที่ 2.11 ระบบปฏิบัติการ Contiki OS	14
รูปที่ 2.12 โลโก้ Amazon Simple Storage Services	16
รูปที่ 2.13 ขั้นตอนการทำงานของ Amazon S3	16
รูปที่ 3.1 Network Diagram.....	20
รูปที่ 3.2 หน้าต่างที่ปรากฏหลังใช้คำสั่ง make connect-router	22
รูปที่ 3.3 หน้าต่างหลังแก้ไขไฟล์ project-conf.h	23
รูปที่ 3.4 การทำงานตั้งแต่อุปกรณ์ Raspberry Pi จนถึงเซิร์ฟเวอร์ของ tunnelbroker.net.....	24
รูปที่ 3.5 หน้าต่างการแก้ไขไฟล์ sysctl.conf	26
รูปที่ 3.6 หน้าต่างหลังสร้างอุโมงค์ที่เว็บไซต์ tunnelbroker.net สำเร็จ	29
รูปที่ 3.7 หน้าต่างการตั้งค่าของอุปกรณ์ Raspberry Pi	30
รูปที่ 3.8 หน้าต่างการตั้งค่าของ AWS CLI.....	31
รูปที่ 3.9 หน้าต่างการสร้างตัวแปรเพื่อรับค่าจากเซ็นเซอร์	33
รูปที่ 3.10 หน้าต่างการตั้งค่า Events ในเว็บไซต์ Ubidots	34
รูปที่ 3.11 หน้าต่างเมื่อสร้าง Events สำเร็จ	35

รูปที่ 3.12 ขั้นตอนการสร้าง Bucket	36
รูปที่ 3.13 หน้าต่างการขอ Access Key และ Secret Access Key	37
รูปที่ 4.1 ฟังก์ชันติดตั้งอุปกรณ์ภายในบ้าน	38
รูปที่ 4.2 หน้าต่างแสดงตารางการเชื่อมต่ออุปกรณ์ผ่าน RPL โพรโทคอล.....	39
รูปที่ 4.3 ข้อความมือถือที่ถูกส่งมาเมื่อมีการตรวจจับความเคลื่อนไหว.....	40
รูปที่ 4.4 อีเมลล์ที่ถูกส่งมาเมื่อมีการตรวจจับความเคลื่อนไหว.....	41
รูปที่ 4.5 รูปถ่ายที่ถูกอัปโหลดขึ้นไป Amazon Simple Storage Services	42
รูปที่ 4.6 ภาพที่ถูกถ่ายหลังจาก Motion2 ตรวจจับความเคลื่อนไหวได้.....	43
รูปที่ 4.7 ภาพที่ถูกถ่ายหลังจาก Motion1 ตรวจจับความเคลื่อนไหวได้.....	44



สารบัญตาราง

ตาราง

หน้า

ตารางที่ 2.1 ไอพีแอดเดรสและโฮสในแต่ละคลาส	7
ตารางที่ 2.2 ตารางโครงสร้างของระบบปฏิบัติการ Contiki OS.....	15



บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากตามบ้านเรือนที่อยู่อาศัยทั่วไปจะมีการป้องกันเพียงแค่ใช้กุญแจล็อกประตุนั้นๆ ถ้ามีผู้มิหวังดีบุกรุกเข้ามา เจ้าของบ้านอาจจะไม่รู้ตัวและไม่สามารถระบุได้ว่าคนร้ายเป็นใคร และในปัจจุบันนี้มีเทคโนโลยีใหม่ๆ เกิดขึ้นมากมาย ข้าพเจ้าจึงเล็งเห็นว่า การนำอุปกรณ์ Internet of Things มาประยุกต์ใช้ตามบ้านเรือน จะทำให้เกิดประโยชน์ สามารถเพิ่มประสิทธิภาพการป้องกันบ้านจากผู้มิหวังดีและสามารถเพิ่มคุณภาพชีวิตให้ดีขึ้นได้

1.2 วัตถุประสงค์การศึกษา

- 1) เพื่อให้ระบบสามารถตรวจจับความเคลื่อนไหวได้เวลาที่มีสิ่งเคลื่อนไหวภายในบ้าน
- 2) เพื่อให้ระบบสามารถถ่ายภาพสิ่งที่เคลื่อนไหวอยู่ได้
- 3) เพื่อให้ระบบสามารถแจ้งเตือนให้ผู้รู้ทราบได้ในทันทีที่มีความเคลื่อนไหวภายในบ้าน

1.3 ขอบเขตของการศึกษา

- 1) ใช้อุปกรณ์ Zolertia Z1 ในการรับค่าจากเซ็นเซอร์
- 2) ใช้ภาษาคอมพิวเตอร์ ภาษา C เขียนโปรแกรมบนอุปกรณ์ Zolertia Z1
- 3) ใช้อุปกรณ์ Raspberry Pi 3 ในการส่งข้อมูลจากอุปกรณ์ Zolertia Z1 เพื่อออกสู่อินเทอร์เน็ต
- 4) ใช้กล้อง Raspberry Pi NoIR Camera ในการถ่ายภาพ
- 5) ใช้ Contiki-OS พัฒนาโปรแกรมบนอุปกรณ์ Zolertia Z1
- 6) ใช้เว็บไซต์ Ubidots.com ในการแสดงผลของเซ็นเซอร์
- 7) ใช้เว็บไซต์ Tunnelbroker.net ในการสร้างอุโมงค์เปลี่ยนผ่านไอพี
- 8) ใช้ PIR motion sensor ในการตรวจจับการเคลื่อนไหวของวัตถุ
- 9) ใช้ Amazon Web Service S3 สำหรับการเก็บภาพถ่ายบนคลาวด์

1.4 นิยามศัพท์เฉพาะ

Tunneling คือการทำให้โมเมนต์เปลี่ยนผ่านไอพื้ร่นที่หก โดยการนำแพ็กเกจไอพื้ร่นที่หกห่อหุ้มด้วยแพ็กเกจของไอพื้ร่นที่สี่ แล้วส่งผ่านเครือข่ายไอพื้ร่นที่สี่จนถึงปลายทาง แล้วทำการคลายแพ็กเกจไอพื้ร่นที่หกออกแล้วส่งต่อไปยังปลายทางที่เป็นเครือข่ายไอพื้ร่นที่หก

เซ็นเซอร์ (Sensor) คืออุปกรณ์ที่ใช้ตรวจจับสภาพแวดล้อมตามลักษณะของเซ็นเซอร์ชนิดนั้นๆ แล้วแปลงค่าออกมาเป็นสัญญาณไฟฟ้าหรือสัญญาณแสงเพื่อทำการวัดค่าที่เกิดขึ้นได้

Zolertia Z1 คืออุปกรณ์ที่ทำหน้าที่รับข้อมูลจากเซ็นเซอร์แล้วส่งข้อมูลขึ้นไปยังอินเทอร์เน็ต โดยผ่านระบบเครือข่ายไร้สายพลังงานต่ำ สามารถเชื่อมต่อเครือข่ายได้ทั้งแบบ Star และแบบ Mesh

Raspberry Pi คืออุปกรณ์ที่ทำหน้าที่คล้ายกับคอมพิวเตอร์ สามารถประมวลผลงานที่ซับซ้อนได้ เหมาะกับงานที่ต้องใช้การประมวลผล แต่ใช้ทรัพยากรต่ำ

ระบบปฏิบัติการ Contiki คือชุดคำสั่งที่ใช้สั่งงานให้อุปกรณ์ Zolertia Z1 สามารถทำงานได้ โดยออกแบบมาสำหรับ Smart Home เพื่อการใช้พลังงานต่ำ รองรับไอพื้ร่นที่หก

Ubidots คือคลาวด์แพลตฟอร์มที่ให้บริการรับค่าจากเซ็นเซอร์แล้วแสดงผลที่หน้าเว็บไซต์ และสามารถส่งการแจ้งเตือนในรูปแบบของอีเมลและข้อความมือถือได้ สามารถสั่งงานอุปกรณ์ผ่านหน้าเว็บได้

Amazon S3 คือคลาวด์แพลตฟอร์มที่ทำหน้าที่เก็บข้อมูลไว้ในคลังที่อยู่บนอินเทอร์เน็ต สามารถจัดการได้ด้วยแอปพลิเคชัน third-party ได้

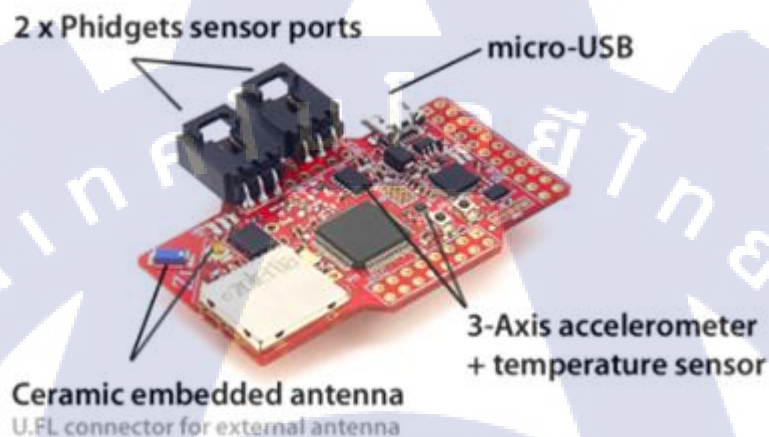
1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1) ผู้ใช้งานสามารถรับรู้การแจ้งเตือนต่าง ๆ ได้ตลอดเวลา เมื่ออยู่นอกบ้าน
- 2) ผู้ใช้งานสามารถลดค่าใช้จ่ายจากการใช้ระบบรักษาความปลอดภัยแบบเดิม
- 3) ผู้ใช้งานสามารถระบุได้ว่าเป็นคนหรือสัตว์ที่เข้ามาในบ้าน

บทที่ 2

ทฤษฎีและเทคโนโลยีที่ใช้ในการปฏิบัติงาน

2.1 อุปกรณ์ Zolertia Z1



<http://zolertia.sourceforge.net/wiki/index.php/Z1>

รูปที่ 2.1 อุปกรณ์ Zolertia Z1

อุปกรณ์ Zolertia Z1 เป็นเซ็นเซอร์เครือข่ายไร้สายพลังงานต่ำ (Wireless Sensor Network, WSN) เป็นโมดูลสำหรับการพัฒนางานแบบอนกประสงค์สำหรับนักพัฒนา ผู้ที่สนใจ และมีมือสมัครเล่น

อุปกรณ์ Zolertia Z1 เป็นอุปกรณ์ที่มีความยืดหยุ่นสูง สามารถเชื่อมต่อกับแหล่งพลังงาน เซนเซอร์ (sensor) และช่องทางการเชื่อมต่อเป็นจำนวนมาก และรองรับระบบปฏิบัติการแบบเปิดที่มีการทำงานของ WSN เช่น TinyOS และ Contiki โดยมีการรองรับสแต็กของระบบเครือข่าย ซึ่งรวมถึง ซิกโลวแพน (6LowPAN), Texas Instrument's SimplicTI, Z-Stack และในอนาคตจะรองรับสแต็กของ ซิกบี (Zigbee) แบบเปิด

การทำงานของอุปกรณ์ Zolertia Z1 จะเป็นโมดูลไร้สายประหยัดพลังงานที่ทำงานบนมาตรฐาน IEEE 802.15.4 และโปรโตคอล Zigbee ซึ่งจะช่วยให้นักพัฒนาเซ็นเซอร์เครือข่ายไร้สาย สามารถทดสอบ และพัฒนาโปรแกรมสำเร็จรูปของตัวเองและสร้างตัวต้นแบบที่ดีในระหว่างการพัฒนาและการรองรับ ฮาร์ดแวร์ได้อย่างยืดหยุ่น

ตัวอย่างโปรแกรมประยุกต์

- 1) การตรวจสอบสุขภาพส่วนบุคคล
- 2) การตรวจสอบด้านสิ่งแวดล้อม
- 3) เครื่องตรวจจับเวลาเกิดเหตุฉุกเฉิน
- 4) อุปกรณ์เพื่อความปลอดภัยและการกู้ภัย
- 5) การตรวจสอบแบบอัตโนมัติแบบระยะยาว
- 6) การวัดค่าพลังงาน
- 7) การตรวจสอบทางการเกษตร

ลักษณะของผลิตภัณฑ์

- 1) เป็นแพลตฟอร์มการพัฒนาตัวต้นแบบของเซ็นเซอร์เครือข่ายไร้สาย ได้อย่างรวดเร็ว
- 2) มีคุณภาพระดับอุตสาหกรรม รองรับอุณหภูมิตั้งแต่ -40 องศาเซลเซียส ถึง 85 องศาเซลเซียส
- 3) เพิ่มช่องทางการเชื่อมต่อเป็น 52 พิน
- 4) ใช้หน่วยประมวลผล MSP430 รุ่นที่สอง แบบประหยัดพลังงานมากเป็นพิเศษ
- 5) มีเซ็นเซอร์วัดความเอียงแบบ 3 ทิศทาง และระบบเน็ตเวิร์กใช้ 2.4 GHz มาตรฐาน IEEE 802.15.4, 6LowPAN และรองรับ Zigbee
- 6) มีเซ็นเซอร์วัดอุณหภูมิดิจิทัลที่ใช้พลังงานต่ำ (ความแม่นยำ ± 0.5 องศาเซลเซียส (ในช่วง -25 ถึง 85 องศาเซลเซียส))
- 7) มีช่องต่อเสาสัญญาณเสริม
- 8) มีช่องต่อ Micro-USB สำหรับการจ่ายพลังงานและการเขียนโปรแกรม

2.2 อุปกรณ์ Raspberry Pi



<https://www.raspberrypi.org/weekly/connected/>

รูปที่ 2.2 อุปกรณ์ Raspberry Pi

Raspberry Pi เป็นเครื่องคอมพิวเตอร์ขนาดเล็กที่สามารถทำงานได้เหมือนกับคอมพิวเตอร์เกือบทุกอย่าง รองรับ HDMI มีช่องต่อยูเอสบี สำหรับการนำเมาส์และคีย์บอร์ดมาต่อเพิ่ม และระบบการจ่ายไฟใช้หัว Mini-USB เหมือนกับสายชาร์จมือถือ และมีพอร์ต General-purpose input/output (GPIO) สำหรับการเชื่อมต่อเซ็นเซอร์และอุปกรณ์ควบคุมต่างๆ เช่น เซ็นเซอร์ตรวจจับความเคลื่อนไหว (Motion sensor)

ระบบปฏิบัติการที่ Raspberry Pi รองรับมีหลากหลาย เช่น Raspbian, Ubuntu MATE, Windows 10 IoT Core, OSMC, LIBREELEC, PiNET, Weather Station, Snappy Ubuntu, RISC OS เป็นต้น ซึ่งแต่ละระบบปฏิบัติการก็จะมีข้อดีแตกต่างกันไป ซึ่งที่ใช้ในงานนี้คือ Raspbian

Raspbian เป็นระบบปฏิบัติการที่มาจากผู้ผลิตเอง ซึ่งจะมาพร้อมกับโปรแกรมเขียนโค้ดภาษาต่างๆ เช่น ภาษา Python, ภาษา Java, โปรแกรม Scratch และโปรแกรม Sonic Pi เป็นต้น

2.3 เว็บไซต์ TunnelBroker.net



<https://www.tunnelbroker.net/>

รูปที่ 2.3 โลโก้เว็บไซต์ tunnelbroker.net

Hurricane Electric เป็นเว็บไซต์ที่ให้บริการในการทำอุโมงค์ไอพีรุ่นที่หก (IPv6 Tunnel Broker) สำหรับผู้ที่มีไอพีรุ่นที่สี่ (IPv4) อยู่ และต้องการจะใช้บริการไอพีรุ่นที่หก (IPv6) ด้วย โดยการทำอุโมงค์ (Tunnel) เป็นการทำให้ไอพีรุ่นที่หก สามารถวิ่งผ่านการเชื่อมต่อเครือข่ายไอพีรุ่นที่สี่ได้ ฟังก์ชันของ TunnelBroker.net นั้น จะมีเราเตอร์ (Router) ทำการแจกไอพีรุ่นที่หกมาให้ ส่วนทางฝั่งของเครื่องลูกข่ายก็ต้องมีเราเตอร์ ที่รองรับการทำงานของไอพีรุ่นที่หก และต้องเชื่อมต่ออินเทอร์เน็ตผ่านไอพีรุ่นที่สี่ จึงจะสามารถใช้งานอุโมงค์ไอพีรุ่นที่หกได้

2.4 ไอพีรุ่นที่สี่ (Internet Protocol version 4, IPv4)

ไอพีรุ่นที่สี่ เป็นอินเทอร์เน็ตโพรโทคอล (Internet Protocol, IP) รุ่นที่สี่ โดยไอพีจะอยู่ในระดับเน็ตเวิร์กเลเยอร์ (Network Layer) ทำหน้าที่จัดการเกี่ยวกับแอดเดรสและข้อมูล และควบคุมการส่งข้อมูลที่ใช้ในการหาเส้นทางของแพ็กเก็ต (Package) ซึ่งการหาเส้นทางของไอพี จะทำการหาเส้นทางที่ดีที่สุด และสามารถเปลี่ยนแปลงเส้นทางของข้อมูลในระดับดาต้าลิงก์เลเยอร์ (Data Link Layer)

การเชื่อมต่อไอพี เพื่อทำการส่งข้อมูลจะเป็นแบบ connectionless หรือเกิดการเชื่อมต่อข้อมูลในทุกๆ ครั้งของการส่งข้อมูล 1 ดาต้าแกรม (datagram) โดยจะไม่ทราบถึงข้อมูลดาต้าแกรมที่ส่งก่อนหน้า หรือส่งตามมา แต่การส่งข้อมูลใน 1 ดาต้าแกรม อาจเกิดการส่งได้หลายครั้งในกรณีที่มีการแบ่งข้อมูลออกเป็นส่วนย่อยๆ (fragmentation) และจะถูกนำไปรวมเป็นดาต้าแกรมเดิมเมื่อถึงปลายทาง

ไอพีแอดเดรสของไอพีรุ่นที่สี่ เป็นระบบ 32 บิต หรือสามารถระบุตัวเลขได้ตั้งแต่ 0.0.0.0 ถึง 255.255.255.255 ไอพีแอดเดรสของไอพีรุ่นที่สี่ จะเขียนโดยใช้เลขฐานสิบจำนวนจำนวนสี่หลักโดยใช้จุด

กันระหว่างแต่ละหลัก โดยไอพีรุ่นที่สี่ทั้งหมดจะถูกแบ่งออกเป็นคลาสชนิดต่างๆ เพื่อจุดประสงค์การใช้งานต่างๆ ดังต่อไปนี้

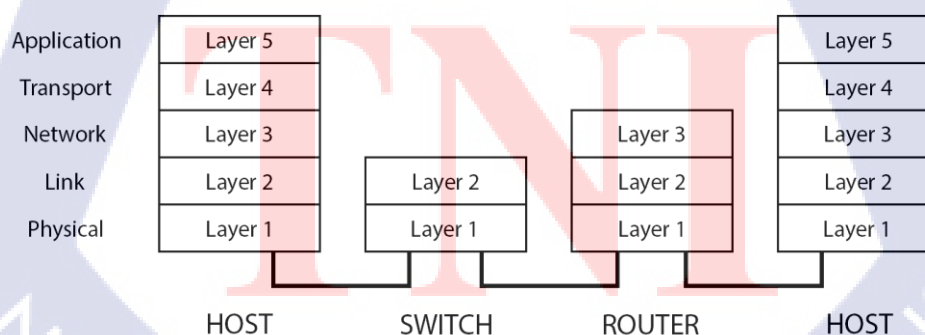
ตารางที่ 2.1 ไอพีแอดเดรสและโฮสในแต่ละคลาส

คลาส	ไอพีเริ่มต้น	ไอพีสิ้นสุด	เน็ตไอดี(บิต)	โฮสไอดี(บิต)
A	0.0.0.0	127.255.255.255	8	24 = 16777216
B	128.0.0.0	191.255.255.255	16	16 = 65536
C	192.0.0.0	223.255.255.255	24	8 = 256
D	224.0.0.0	239.255.255.255	-	มัลติแคส
E	240.0.0.0	247.255.255.255	-	สำรอง

2.5 ไอพีรุ่นที่หก (Internet Protocol version 6, IPv6)

2.5.1 แนวคิดของ IPv6

เราจำเป็นต้องรู้จักรุ่นล่าสุดของอินเทอร์เน็ตโพรโทคอล เพื่อที่จะเข้าใจว่าทำไมไอพีรุ่นที่หกถึงมีประโยชน์กับ Internet of Things และไอพีรุ่นที่หกเกี่ยวข้องกับโพรโทคอลอื่นๆอย่างไร เช่น 6LoWPAN ถ้าหากมีความคุ้นเคยมาก่อนเกี่ยวกับ บิต, ไบต์, สแต็กของเครือข่าย, แพ็กเกจ, ส่วนหัวขอไอพี และอื่นๆ คุณควรเข้าใจว่าไอพีรุ่นที่หก มีโพรโทคอลที่ต่างกัน และไม่เข้ากันกับไอพีรุ่นที่สี่



รูปที่ 2.4 สแต็กของอินเทอร์เน็ตโพรโทคอล

ไอพีรุ่นที่หกทำงานในชั้นที่สามที่เรียกว่า เน็ตเวิร์คเลเยอร์ (network layer) ส่วนของข้อมูลที่ถูกจัดการในชั้นที่สามเรียกว่า แพ็กเกจ อุปกรณ์ที่เชื่อมต่อกับเป็นได้ทั้งโฮส (host) และเราเตอร์

2.5.2 6LoWPAN (IPv6 over Low power Wireless Personal Area Networks)

6LoWPAN เป็นมาตรฐานในการสื่อสารของไอพีรุ่นที่หก บนมาตรฐานเทคโนโลยีการสื่อสารไร้สาย IEEE 802.15.4 โดยจะทำหน้าที่เป็นชั้นในการปรับเปลี่ยนระหว่างมาตรฐานไอพีรุ่นที่หก และการใช้พลังงานต่ำและการสื่อสารไร้สายแบบถูกลดทอน ที่ถูกนำเสนอโดย IEEE 802.15.4

Low power and Lossy networks (LLNs) เป็นคำที่ใช้อ้างอิงถึงระบบเครือข่ายที่สร้างจากอุปกรณ์ที่มีข้อจำกัดสูง (ข้อจำกัดของหน่วยประมวลผล, หน่วยความจำ และการใช้พลังงาน) ที่เชื่อมต่อกันด้วยความประหยัที่มีหลากหลายอย่างผ่านการเชื่อมต่อวิทยุแบบพลังงานต่ำ ซึ่งลักษณะความเร็วต่ำ, ลดคุณภาพลง, ประหยัดต้นทุน และการเชื่อมต่อที่เสถียรภาพต่ำ

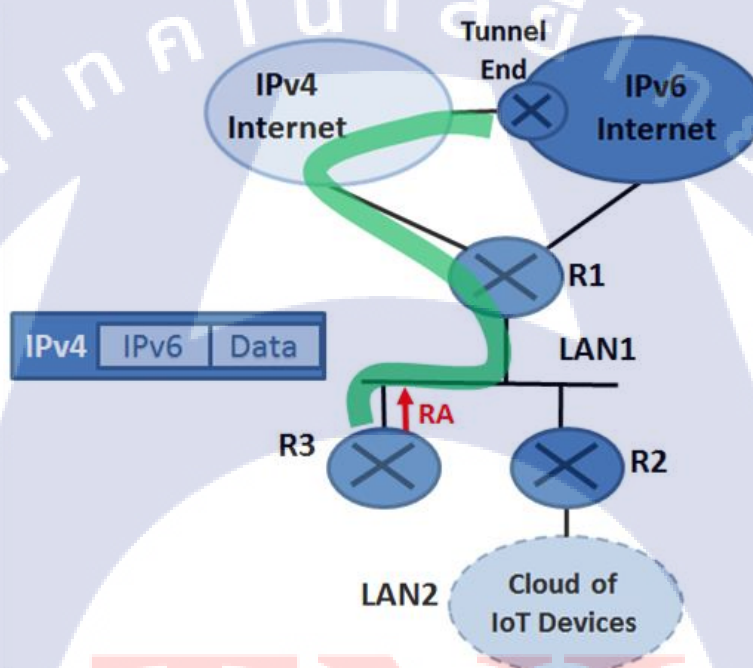
LoWPAN เป็นกรณีของ LLN ที่เกิดขึ้นจากอุปกรณ์ที่ใช้งานร่วมกับมาตรฐาน IEEE 802.15.4

ลักษณะที่สำคัญของอุปกรณ์ที่ใช้ LoWPAN คือ

- 1) การรองรับการประมวลผลที่จำกัด คือ รูปแบบที่ต่างกัน และรอบความเร็วที่ใช้ในการประมวลผล โดยจะเริ่มต้นที่ 8 บิต
- 2) การรองรับหน่วยความจำขนาดเล็ก คือ จากหน่วยความจำชั่วคราวไม่กี่กิโลไบต์ กับหน่วยความจำถาวรที่มีไม่กี่สิบกิโลไบต์ และคาดว่าจะมีการเพิ่มขึ้นอีกในอนาคต แต่ก็ยังคงความจำเป็นขั้นต่ำเอาไว้
- 3) ใช้พลังงานต่ำ ใช้พลังงานเพียงไม่กี่สิบบิลลิแอมแปร์
- 4) สื่อสารระยะสั้น The Personal Operating Space (POS) กำหนดให้มาตรฐาน IEEE 802.15.4 หมายถึงมีระยะการใช้งานได้ถึง 10 เมตร และสำหรับการใช้งานจริงจะสามารถไปไกลได้กว่า 100 เมตรในระยะสายตา
- 5) ประหยัดต้นทุน เพราะอุปกรณ์เหล่านี้ทำงานด้วยลักษณะเฉพาะ เช่น การประมวลผลต่ำ หน่วยความจำน้อย และอื่นๆ

2.5.3 IPv6 Transition Mechanism (6in4)

ถ้าไม่ได้เชื่อมต่อกับไอพีรุ่นที่หกและเราเตอร์ยังไม่รองรับไอพีรุ่นที่หก ในสถานการณ์นี้ยังหมายถึงการที่ไม่ได้รับไอพีรุ่นที่หกจากผู้ให้บริการอินเทอร์เน็ต (Internet Service Provider) เพื่อแก้ปัญหาการขาดแคลนไอพีรุ่นที่หกจากผู้ให้บริการ เราสามารถใช้ 6in4 ได้ และวิธีการนี้ยังสามารถแก้ปัญหาเราเตอร์ที่ไม่รองรับไอพีรุ่นที่หกได้อีกด้วย วิธีการก็คือการเพิ่มเราเตอร์ใหม่ที่รองรับทั้งไอพีรุ่นที่หกและไอพีรุ่นที่สี่ และสร้างอุโมงค์ 6in4 ขึ้นมาจากรouterนี้กับปลายอุโมงค์แห่งหนึ่งในอินเทอร์เน็ตของไอพีรุ่นที่หก ซึ่งเป็นวิธีที่นิยมและง่ายต่อการทำมากที่สุด โดยหลักการคือ สร้างอุโมงค์ดังกล่าวด้วยตัวเอง เชื่อมต่อแบบจุดต่อจุด ซึ่งอุโมงค์นี้จะทำการห่อหุ้มแพ็กเกจไอพีรุ่นที่หกไว้ในไอพีรุ่นที่สี่



รูปที่ 2.5 Network Diagram 6in4 Tunneling

ในสถานการณ์นี้สมมุติให้เราเตอร์ R3 ทำการสร้างอุโมงค์ 6in4 เชื่อมต่อกับปลายอุโมงค์ที่ทำหน้าที่เป็นเกตเวย์ (gateway) ของไอพีรุ่นที่หก ทำการส่ง RAs ไปยังอุปกรณ์ไอพีรุ่นที่หกทำการตั้งค่าอัตโนมัติในช่อง LAN1 และทำการห่อหุ้มแพ็กเกจที่เครื่องต้นทาง และจะไปคลายแพ็กเกจที่เครื่องปลายทาง ซึ่งอุโมงค์ 6in4 นี้จะต้องใช้ไอพีรุ่นที่สี่แบบสาธารณะ ดังนั้น R3 จะต้องมไอพีรุ่นที่แบบสาธารณะ

2.6 Ubidots Cloud Platform



<https://ubidots.com/>

รูปที่ 2.6 โลโก้ Ubidots

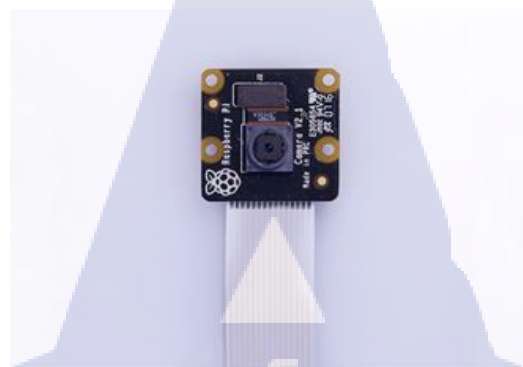
Ubidots เป็นการออกแบบคลาวด์แพลตฟอร์มโดยไม่ต้องเขียนโค้ดเอง เพื่อช่วยในการสร้างตัวต้นแบบและขยายโครงการไอโอทีเพื่อการเพิ่มผลผลิต โดยตัว Ubidots เองจะเป็นคลาวด์แพลตฟอร์มที่ให้บริการกับนักพัฒนาไอโอทีทั่วไป รวมถึงองค์กรที่ต้องการความน่าเชื่อถือในการใช้งาน และการจัดการลูกค้า

ขั้นตอนการใช้ Ubidots เบื้องต้น

- 1) เชื่อมต่ออุปกรณ์ไอโอทีไปยังคลาวด์แพลตฟอร์มโดยใช้คลังเฟิร์มแวร์ของเรา
- 2) สร้างการแสดงผลแบบกำหนดเองแบบเรียบง่าย มีการแสดงผลตลอดเวลาบนหน้าแดชบอร์ด
- 3) สามารถส่งเอสเอ็มเอส อีเมลล์ หรือทำให้เกิดคำสั่งต่อไปได้บนพื้นฐานข้อมูลของคุณ
- 4) ปรับใช้ ปรับแต่งกับงานระดับธุรกิจหรือสร้างแอปพลิเคชันของตัวเอง

TNI

2.7 กล้อง Raspberry Pi NoIR Camera V2



<https://www.raspberrypi.org/products/pi-noir-camera-v2/>

รูปที่ 2.7 กล้อง Raspberry Pi NoIR Camera v2

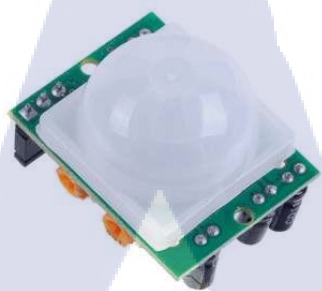
กล้องราสเบอร์รี่พายอินฟราเรดโมดูลรุ่นที่สองมาแทนที่กล้องราสเบอร์รี่พายอินฟราเรดรุ่นเริ่มต้น โดยใช้เซ็นเซอร์ โซนี่ IMX219 แพลตฟอร์มพิกเซล ในเดือนเมษายน ปี 2016 แทนรุ่นเก่า (เมื่อเทียบกับเซ็นเซอร์ OmniVision OV5647 หัวด้านพิกเซลของกล้องรุ่นเริ่มต้น).

กล้องราสเบอร์รี่พายอินฟราเรดสามารถทำได้ทุกอย่างเหมือนกับกล้องราสเบอร์รี่พายธรรมดา แต่มีหนึ่งสิ่งที่ต่างกันคือ กล้องราสเบอร์รี่พายอินฟราเรดจะไม่ได้ติดตั้งตัวกรองอินฟราเรด (NoIR = No Infrared.) ซึ่งหมายความว่า รูปที่ถ่ายในเวลากลางวันจะมีลักษณะซีด แต่ก็มีความสามารถในการมองเห็นในตอนกลางคืนแทน โดยใช้ร่วมกับแสงอินฟราเรด

ในแพ็คเกจของกล้องราสเบอร์รี่พายอินฟราเรด จะมีแผ่นเจลสีน้ำเงินที่หุ้มเล็กๆแถมมาพร้อมกับกล้องอินฟราเรด สำหรับใช้ในการตรวจสอบสุขภาพของพืช กล้องราสเบอร์รี่พายอินฟราเรดเป็นที่นิยมมากสำหรับมือสมัครเล่นที่ชอบในการส่องสัตว์ โดยใช้ร่วมกับหลอดไดโอดเปล่งแสงอินฟราเรด (LED Infrared) คุณสามารถตรวจสอบสัตว์ที่ออกหากินตอนกลางคืนได้ว่า พวกสัตว์มาทำอะไรในสวนของคุณโดยที่ไม่ต้องรบกวนพวกสัตว์เหล่านั้น

กล้องราสเบอร์รี่พายอินฟราเรดสามารถใช้งานได้กับราสเบอร์รี่พายทุกรุ่น มันสามารถเข้าถึงได้ด้วย MMAL และ V4L APIs และสามารถเข้าถึงได้โดย third-party libraries ที่ถูกสร้างสำหรับมัน รวมถึง PiCamera Python library

2.8 เซ็นเซอร์ตรวจจับความเคลื่อนไหว (PIR motion sensor)



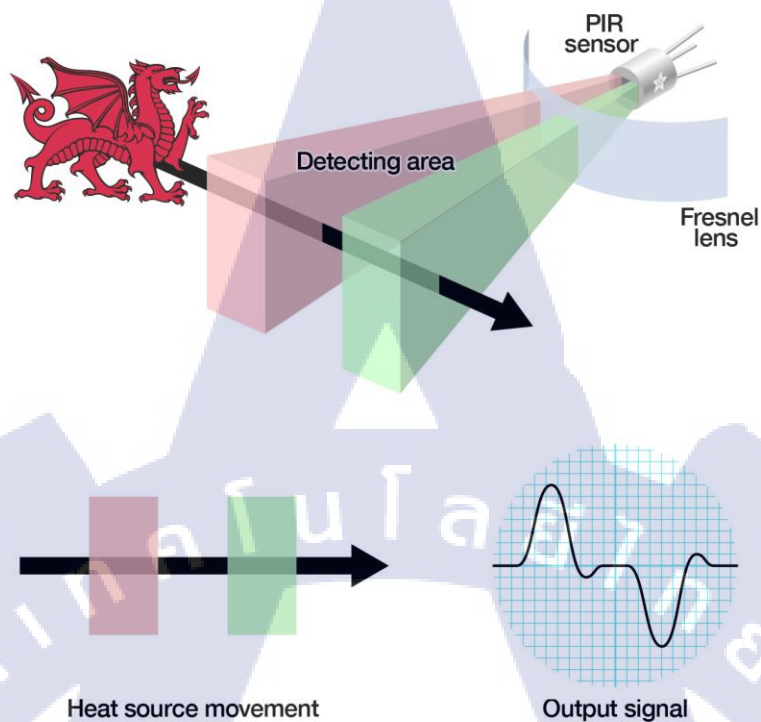
<http://thaisensormodule.com/index.php/light/pirsens1>

รูปที่ 2.8 PIR Motion Sensor

PIR ย่อมาจาก Passive Infrared เป็นอุปกรณ์ที่ใช้ตรวจจับความเคลื่อนไหวด้วยการวัดความร้อนในพื้นที่ที่ต้องการ โดยความร้อนสามารถวัดได้จากการเปลี่ยนแปลงระดับรังสีอินฟราเรดที่ปล่อยออกมาจากวัตถุ (สิ่งมีชีวิตทุกชนิดจะแผ่รังสีอินฟราเรดออกมาจากตัวเอง การแผ่รังสีเกิดจากการเคลื่อนของอิเล็กตรอนในอะตอม ปริมาณรังสีจะมีมากน้อยตามแต่โครงสร้างทางเคมีและอุณหภูมิของวัตถุหรือสิ่งมีชีวิตนั้นๆ) เมื่อวัตถุเคลื่อนที่จะทำให้แรงดันไฟฟ้ามีค่าสูงขึ้นและนำมาแปลงเป็นเสียงหรือแสง สำหรับใช้เป็นเซ็นเซอร์ได้ จึงมีการนำเอา PIR มาประยุกต์ใช้งานเพื่อเป็นตรวจจับการเคลื่อนไหวของสิ่งมีชีวิต หรือตรวจจับการบุกรุกในงานรักษาความปลอดภัย หรือที่เรียกกันว่า PIR motion sensor นั่นเอง

หลักการทำงานของ Passive Infrared sensor

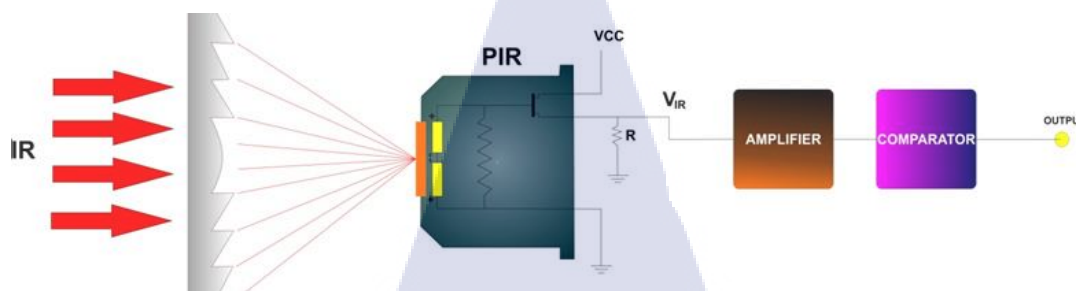
การทำงานของ PIR จะซับซ้อนกว่าเซ็นเซอร์อื่นๆ เพราะมีหลายตัวแปรที่มีผลต่อเอาต์พุตและอินพุตของเซ็นเซอร์ตัวนี้



<https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/how-pirs-work>

รูปที่ 2.9 การทำงานของ PIR Motion Sensor

ภายใน PIR จะมีอุปกรณ์ตรวจจับรังสีอินฟราเรดอยู่สองชุดด้วยกันดังรูป โดยในแต่ละชุดของอุปกรณ์จะทำจากวัสดุพิเศษที่ไวต่อรังสีอินฟราเรดมาก ส่วนเลนส์(Fresnel lens) ที่ครอบอยู่ทำหน้าที่คล้ายเลนส์นูนกระจายแสงจากจุดโฟกัสด้านหนึ่งเป็นลำแสงขนานออกไปในระยะไกลได้ เมื่อมีคนหรือสัตว์ที่มีความอบอุ่นในร่างกายเคลื่อนที่ผ่านเข้ามาในพื้นที่ตรวจจับ ซึ่งในขณะที่เคลื่อนไหวจะมีรังสีความร้อนแผ่ออกมารอบๆตัวในปริมาณที่แน่นอนอยู่จำนวนหนึ่ง ทำให้อุณหภูมิในบริเวณนั้นเกิดการเปลี่ยนแปลง PIR สามารถตรวจจับคลื่นรังสีอินฟราเรดที่แผ่ออกมาจากสิ่งมีชีวิตได้ PIR จะส่งคลื่นรังสีมายังตัว Pyro Electric ซึ่งจะเปลี่ยนพลังงานความร้อนจากรังสีอินฟราเรดเป็นพลังงานไฟฟ้า ตามการเคลื่อนไหวนั้นออกมาทางขาเอาท์พุท แล้วจะถูกป้อนเข้าสู่ไอซีเพื่อทำการขยายสัญญาณต่อ

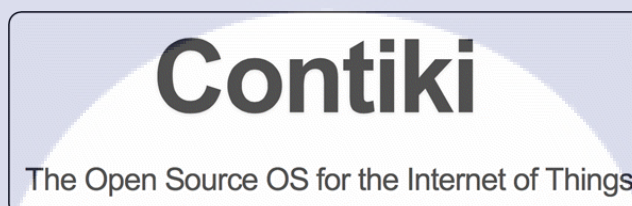


<http://www.thaieasyelec.com/article-wiki/review-product-article/pir-motion-sensor-getting-started.html>

รูปที่ 2.10 การทำงานของ PIR motion sensor เมื่อมีรังสีอินฟราเรดเปลี่ยนแปลง

ในตอนที่ยังไม่มีสิ่งมีชีวิตผ่านเซ็นเซอร์ไป ทั้งสองอุปกรณ์ตรวจจับคลื่นรังสีจะมีค่าแรงดันปกติ และเมื่อมีสิ่งมีชีวิตเคลื่อนที่ผ่าน อุปกรณ์ตรวจจับรังสีอินฟราเรดตัวที่หนึ่งจะได้สัญญาณเอาต์พุตออกมาสูงกว่าแรงดันปกติ เมื่อเทียบกับแรงดันจากอุปกรณ์อีกตัวหนึ่ง และเมื่อสิ่งมีชีวิตเคลื่อนออกจากพื้นที่ตรวจจับ จะเกิดการย้อนกลับของแรงดัน ทำให้แรงดันเอาต์พุตต่ำกว่าแรงดันปกติ แล้วจึงกลับมาเท่าปกติในภายหลัง นั่นหมายความว่าสามารถตรวจจับสิ่งมีชีวิตที่เคลื่อนไหวย้อนผ่านเซ็นเซอร์นี้ได้แล้ว

2.9 ระบบปฏิบัติการ Contiki OS



<http://www.thewindowsclub.com/contiki-os-vs-windows-10-internet-things>

รูปที่ 2.11 ระบบปฏิบัติการ Contiki OS

ระบบปฏิบัติการ Contiki เป็นระบบปฏิบัติการแบบสาธารณะ ถูกออกแบบมาสำหรับ Internet of Things เพื่อเชื่อมต่ออุปกรณ์ขนาดเล็ก ไมโครคอนโทรลเลอร์ที่ใช้พลังงานต่ำไปยังอินเทอร์เน็ต

ระบบปฏิบัติการ Contiki รองรับการใช้พลังงานต่ำและสามารถเชื่อมต่ออินเทอร์เน็ตได้ รองรับมาตรฐานไอพีรุ่นที่สี่และไอพีรุ่นที่หก พร้อมทั้งมาตรฐานของระบบเครือข่ายไร้สายพลังงานต่ำด้วย เช่น 6LoWPAN, RPL, CoAP, ContikiMAC ของ Contiki และ Sleepy Routers ถึงแม้ว่าจะทำงานด้วยระบบเครือข่ายไร้สายพลังงานต่ำก็ตาม ก็ยังสามารถใช้งานได้ดีในขณะที่ใช้พลังงานจากแบตเตอรี่

ระบบปฏิบัติการ Contiki ง่ายต่อการนำไปพัฒนาต่อ เพราะแอปพลิเคชันถูกเขียนด้วยภาษาซี และยังสามารถใช้โปรแกรมจำลอง Cooja Simulator ที่สามารถจำลองระบบเครือข่ายได้ ก่อนที่จะนำไปใช้งานบนอุปกรณ์และสถานที่จริง และมี Instant Contiki ที่เป็นระบบปฏิบัติการ Contiki แบบสำเร็จรูป สามารถดาวน์โหลดมาแล้วใช้งานได้ทันที

ส่วนประกอบของระบบปฏิบัติการ Contiki มีโครงสร้างดังนี้

ตารางที่ 2.2 ตารางโครงสร้างของระบบปฏิบัติการ Contiki OS

เพิ่มข้อมูล	รายละเอียด	ข้อมูลของอุปกรณ์ Zolertia
Examples	เตรียมการเพื่อสร้างตัวอย่าง	examples/zolertia, examples/cc2538-common
App	แอปพลิเคชันของ Contiki	-
Cpu	เพิ่มข้อมูลเฉพาะของ MCU	msp430, cc2538
Dev	หน่วยประมวลผลภายนอกและอุปกรณ์	cc2420, cc1200
Platform	เพิ่มข้อมูลเฉพาะและแพลตฟอร์มของตัวขับเคลื่อน	z1, zoul
Core	เพิ่มแกนหลักของ Contiki และคลังข้อมูล	-
Tools	เครื่องมือที่ใช้ในการล้าง, แก้ไข, ขอบกพร่อง, การจำลอง และอื่นๆ	zolertia, sky
Docs	เอกสารต่างๆที่จัดทำขึ้นมาด้วยตัวเอง	-
regression-tests	การทดสอบการถดถอย	-

ถ้าหากเข้าไปในเพิ่มข้อมูล [contiki/examples/zolertia/tutorial/](#) จะพบกับคู่มือการใช้งานระบบต่างๆของระบบปฏิบัติการ Contiki บนอุปกรณ์ Z1 และ Zoul โดยเนื้อหาจะแบ่งตามเพิ่มข้อมูลดังนี้

01-basics เกี่ยวกับพื้นฐานของ Contiki (เวลา, GPIO, หลอดไฟ, โดดแปลงแสง)

02-ipv6 เกี่ยวกับระบบเครือข่ายไร้สาย, พื้นฐานของสัญญาณวิทยุ, การใช้ IPv6/6LoWPAN และการใช้ขอบของเราเตอร์

03-coap	ตัวอย่างการใช้งานเซิร์ฟเวอร์ CoAP
04-mqtt	ตัวอย่างการใช้งาน MQTT บนเครื่องลูกข่าย

2.10 คลาวด์แพลตฟอร์ม Amazon Simple Storage Service (Amazon S3)



<http://www.signiant.com/fast-file-transfer-to-amazon-s3/>

รูปที่ 2.12 โลโก้ Amazon Simple Storage Services

Amazon Simple Storage Service (Amazon S3) คือที่เก็บข้อมูลที่อยู่บนอินเทอร์เน็ต คุณสามารถใช้ Amazon S3 ในการจัดเก็บและเรียกกลับข้อมูลขนาดใดก็ได้ตลอดเวลา จากที่ไหนก็ได้ที่มีการเชื่อมต่อกับอินเทอร์เน็ต คุณสามารถทำสิ่งเหล่านี้ได้ด้วยวิธีการที่เรียบง่ายและเข้าใจได้กับหน้าเว็บอินเทอร์เฟซหน้าตาการจัดการของ AWS และคู่มือนี้จะเป็นการทำความรู้จักกับ Amazon S3 และวิธีการใช้หน้าตาการจัดการของ AWS ให้สำเร็จด้วยการแสดงขั้นตอนการทำงานดังนี้



<http://docs.aws.amazon.com/AmazonS3/latest/gsg/GetStartedWithS3.html>

รูปที่ 2.13 ขั้นตอนการทำงานของ Amazon S3

- (1) สมัครเข้าใช้งาน Amazon S3
- (2) สร้างคลังเก็บข้อมูล
- (3) เพิ่มข้อมูลเข้าไปในคลัง
- (4) เรียกดูข้อมูลได้
- (5) สามารถเคลื่อนย้ายข้อมูลได้
- (6) สามารถข้อมูลหรือลบคลังออกได้

2.11 ชุดพัฒนาซอฟต์แวร์ Boto3

โบโทสาม(Boto3) เป็นชุดพัฒนาซอฟต์แวร์ของ Amazon Web Services (AWS) ที่ใช้ภาษาไพธอน (Python programming language) ซึ่งช่วยให้นักพัฒนาที่ใช้ภาษาไพธอนสามารถเขียนซอฟต์แวร์เรียกใช้บริการของ Amazon ได้ เช่น Simple Storage Services (S3), Elastic Compute Cloud (EC2) ชุดพัฒนาโบโทนั้น ง่ายต่อการใช้งาน ใช้ API ในงานเชิงวัตถุได้เช่นเดียวกับภาษาระดับล่าง สามารถเข้าถึงบริการได้โดยตรง

โบโทสามถูกเขียนขึ้นมาใหม่จากโบโท มันถูกใช้ในการส่งข้อมูลในการสร้างคลาสในระหว่างการทำงานโดยไฟล์ JSON ที่ถูกใช้ร่วมกันในระหว่างการพัฒนาซอฟต์แวร์ในภาษาต่างๆ ซึ่งรวมถึงการเขียนโปรแกรมในภาษาระดับสูง การเขียนในเชิงวัตถุที่มีอินเทอร์เฟซคล้ายกับหน้าตาของรุ่นก่อนๆ ของโบโท

โบโทสามประกอบด้วยคุณสมบัติที่สำคัญดังนี้

- 1) แหล่งข้อมูล - ใช้ภาษาระดับสูง, ออกแบบอินเทอร์เฟซในเชิงวัตถุ
- 2) คอลเลกชัน - ใช้เครื่องมือในการทำซ้ำและการจัดการกลุ่มของแหล่งข้อมูล
- 3) เครื่องลูกข่าย - เชื่อมต่อบริการในระดับล่าง
- 4) การจัดหน้า - การจัดหน้าอัตโนมัติทันทีที่มีการตอบสนอง
- 5) การให้บริการ - หาทางป้องกันจนกว่าจะมีสภาพที่แน่นอนที่จะไปถึงเป้าหมาย

พร้อมด้วยคุณสมบัติเหล่านี้ โบโทสามได้มีการเตรียมการต่างๆ และการทดสอบชุดไบนารี และการกำหนดค่าความปลอดภัยพื้นฐาน เช่น การยืนยันตัวตน พารามิเตอร์และการควบคุมการตอบสนอง มีระบบอีเว้นท์สำหรับการปรับแต่งและใช้ตรรกะในการแก้การร้องขอที่ล้มเหลว

TNI

บทที่ 3

แผนงานการปฏิบัติงานและขั้นตอนการดำเนินงาน

3.1 แผนการดำเนินงาน

ตารางที่ 3.1 แผนการดำเนินงาน

หัวข้อ	เดือน 1	เดือน 2	เดือน 3	เดือน 4
1.ปรึกษากับอาจารย์ที่ปรึกษาเกี่ยวกับการปรับหัวข้อใหม่				
2.ศึกษาการทำงานของอุปกรณ์ Zolertia Z1				
3.ศึกษาการทำอุโมงค์ไอฟีรูนที่หก				
4.ศึกษาการทำงานของเซ็นเซอร์				
5.ศึกษาการทำงานของอุปกรณ์ราสเบอร์รี่พาย และกล้องราสเบอร์รี่พายอินฟราเรด				
6.ออกแบบและทดสอบการทำงาน				
7.ปรับปรุงและพัฒนาให้ดีขึ้น				
8.ทดลองและบันทึกผล				
9.จัดทำรูปเล่ม				

3.1.1 ปรัชญาที่ปรึกษาเกี่ยวกับการปรับหัวข้อใหม่

จากเดิม ในวิชาโปรเจกต์ 1 ได้ศึกษาการทำงานของอุปกรณ์ Zolertia Z1 โดยจะเน้นไปทางด้านเทคนิคอล เกี่ยวกับการสื่อสารกันของตัวอุปกรณ์ด้วยตนเอง และการใช้งานเซ็นเซอร์ต่างๆ หลังจากนั้น ได้มีความคิดริเริ่มว่าควรจะนำความรู้ที่มีมาปรับใช้ในชีวิตประจำวัน จึงได้ทำการปรึกษากับอาจารย์ที่ปรึกษาถึง การนำอุปกรณ์มาทำให้เกิดประโยชน์ในชีวิตประจำวันมากขึ้น

3.1.2 ศึกษาการทำงานของอุปกรณ์ Zolertia Z1

ทำการศึกษาตัวอุปกรณ์ Zolertia Z1 ทั้งทางด้านกายภาพ ว่าสามารถเชื่อมต่อกับเซ็นเซอร์ใดได้บ้าง และทางด้านซอฟต์แวร์ว่าสามารถใช้ระบบปฏิบัติการใดได้บ้าง และการเขียนแอปพลิเคชันเพื่อใช้งานร่วมกับเซ็นเซอร์ต่างๆ รวมถึงวิธีการสื่อสารกันเองภายในเครือข่ายไอพีรุ่นที่หกแบบพลังงานต่ำ

3.1.3 ศึกษาการทำโมดูลไอพีรุ่นที่หก

ถ้าหากต้องการจะให้อุปกรณ์ Zolertia Z1 สามารถเชื่อมต่อกับอินเทอร์เน็ตได้ จะต้องใช้ไอพีรุ่นที่หกเท่านั้น ดังนั้น ในสถานที่ที่ไม่มีไอพีรุ่นที่หกจึงต้องหาทางเชื่อมต่อให้ได้ และถ้าสามารถเชื่อมต่อไอพีรุ่นที่สี่ได้ด้วยไอพีจริงก็จะสามารถทำการเชื่อมต่อไอพีรุ่นที่หกผ่านอุโมงค์ได้

3.1.4 ศึกษาการทำงานของเซ็นเซอร์

เซ็นเซอร์ชนิดต่างๆมีให้เลือกใช้หลากหลาย ดังนั้นเราจึงต้องศึกษาให้เข้าใจว่าเซ็นเซอร์แต่ละชนิดสามารถทำอะไรได้บ้าง ทั้งประเภทของเซ็นเซอร์ การใช้พลังงาน การเชื่อมต่อสาย การนำไปใช้กับงานอะไร

3.1.5 ศึกษาการทำงานของอุปกรณ์راسเบอร์พาย และกล้องراسเบอร์พายอินฟราเรด

อุปกรณ์راسเบอร์พายนั้นสามารถทำได้เหมือนคอมพิวเตอร์เครื่องหนึ่ง สามารถใช้งานได้หลากหลาย จึงต้องศึกษาว่าจะนำใช้งานแบบไหนถึงจะเหมาะสม และศึกษาทดลองการใ้กล้องراسเบอร์พายอินฟราเรด

3.1.6 ออกแบบและทดสอบการทำงาน

หลังจากได้ศึกษาอุปกรณ์ต่างๆแล้ว ก็ได้ทราบว่าอุปกรณ์แต่ละชนิดสามารถทำอะไรได้บ้าง ดังนั้นก็ถึงเวลาที่ต้องออกแบบสิ่งที่ต้องการจะทำ ตั้งแต่การเลือกวิธีการที่จะใช้ เซ็นเซอร์ที่จะใช้ เพื่อให้ได้ตัวผลงานออกมา

3.1.7 ปรับปรุงและพัฒนาให้ดีขึ้น

ตัวงานที่ทำขึ้นมานั้น ยังไม่ดีเท่าที่ควร ซึ่งส่วนมากจะอยู่ในส่วนของซอฟต์แวร์ ดังนั้นจึงต้องหาวิธีทำให้ผลลัพธ์ที่ออกมามีประสิทธิภาพให้มากที่สุด ทั้งในเรื่องของการวางตำแหน่งของอุปกรณ์, การเดินสายไฟ, ระยะทางของสัญญาณ ไร้สายที่จะไปถึง, วิธีการเชื่อมต่ออินเทอร์เน็ต, ความเร็วในการส่งข้อมูล และคุณภาพของรูปถ่าย เป็นต้น

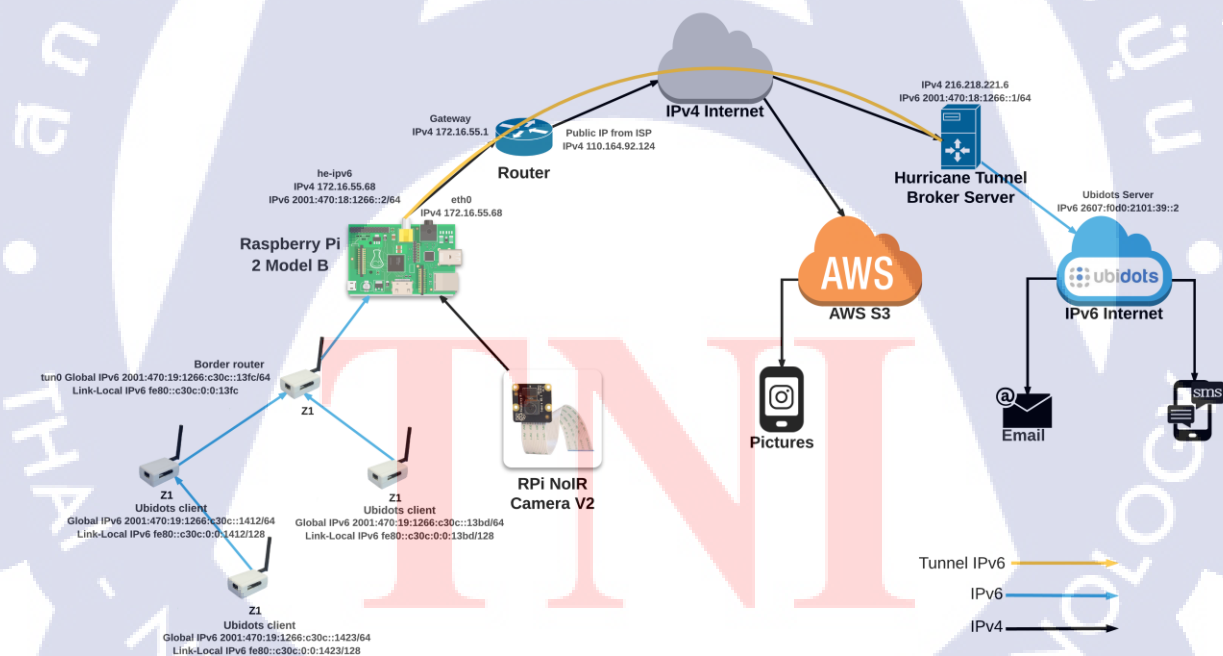
3.1.8 ทดลองและบันทึกผล

เมื่องานที่ทำอยู่ในจุดที่ต้องการแล้ว ก็ทำการทดลอง แล้วบันทึกผลที่เกิดขึ้นในขั้นตอนต่างๆ

3.1.9 จัดทำรูปเล่ม

รวบรวมงานที่ทำมาทั้งหมด และสรุปผลงานที่ทำออกมา แล้วนำมาเขียนเป็นรูปเล่ม

3.2 ขั้นตอนการดำเนินงาน



รูปที่ 3.1 Network Diagram

3.2.1 การตั้งค่าบนอุปกรณ์ Zolertia Z1

3.2.1.1 ส่วนของ Border Router

แอปพลิเคชันขอบเราเตอร์ประกอบไปด้วยเว็บเซิร์ฟเวอร์ การแสดงผลข้อมูลเครือข่าย เช่น การแสดงผลของอุปกรณ์ที่เชื่อมต่ออยู่ และเส้นทางการเชื่อมต่อของระบบเครือข่าย

วิธีการใช้งาน

- 1) เข้าไปที่เพิ่มข้อมูล /home/pi/contiki/zolertia/tutorial/02-ipv6/02-border-router
- 2) เขียนแอปพลิเคชัน border-router.c ลงบนอุปกรณ์ Zolertia Z1 โดยใช้บรรทัดคำสั่งดังนี้

```
make clean && make border-router.upload && make connect-router  
PREFIX=2001:470:19:1266::/64
```

คำอธิบายบรรทัดคำสั่ง

- 1) คำสั่ง make clean ใช้สำหรับลบการคอมไพล์โปรแกรมอ็อบเจกต์ก่อนหน้านี้
- 2) คำสั่ง make border-router.upload ใช้สำหรับเขียนแอปพลิเคชันขอบเราเตอร์ลงบนอุปกรณ์ Zolertia Z1
- 3) คำสั่ง make connect-router ใช้สำหรับตั้งค่าเลขชุดหน้าของไอพีรุ่นที่หก ถ้าไม่ได้ตั้งค่าในส่วนนี้ ค่าเดิมจะเป็น aaaa::1/64

```

pi@raspberrypi: ~
File Edit Tabs Help
rm border-router.ihex obj_z1/httpd-simple.o obj_z1/slip-bridge.o border-router.co obj_
z1/contiki-z1-main.o
using saved target 'z1'
sudo ./../../../../tools/tunslip6 2001:470:19:1266::/64
*****SLIP started on ``/dev/ttyUSB0''
opened tun device ``/dev/tun0''
ifconfig tun0 inet `hostname` mtu 1500 up
ifconfig tun0 add 2001:470:19:1266::/64
ifconfig tun0 add fe80::470:19:1266:0/64
ifconfig tun0

tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
inet addr:127.0.1.1  P-t-P:127.0.1.1  Mask:255.255.255.255
inet6 addr: 2001:470:19:1266::/64 Scope:Global
inet6 addr: fe80::470:19:1266:0/64 Scope:Link
UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:500
RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

*** Address:2001:470:19:1266:: => 2001:0470:0019:1266
Got configuration message of type P
Setting prefix 2001:470:19:1266::
Server IPv6 addresses:
 2001:470:19:1266:c30c::13fc
 fe80::c30c:0:0:13fc

```

รูปที่ 3.2 หน้าต่างที่ปรากฏหลังใช้คำสั่ง make connect-router

3.2.1.2 ส่วนของ Ubidots Client

แอปพลิเคชัน ubidots ถูกใช้โดยอุปกรณ์ Zolertia Z1 และทำการประกาศข้อมูลของตัวรับรู้ โดยใช้ไอพีรุ่นหกเชื่อมต่อไปที่ things.ubidots.com เพื่อแสดงผลของข้อมูลบนหน้าแดชบอร์ด

วิธีการใช้งาน

- 1) เข้าไปที่แฟ้มข้อมูล `/home/pi/contiki/examples/zolertia/tutorial/99-apps/project-ubidots`
- 2) เข้าไปแก้ไขไฟล์ `project-conf.h` ดังนี้

```

#define POST_PERIOD (CLOCK_SECOND)

#define UBIDOTS_CONF_AUTH_TOKEN “[ใส่ token key ของชื่อผู้ใช้จากเว็บไซต์ Ubidots]”

#define VARKEY_VARIABLE_ONE “[ใส่ variable key ที่สร้างขึ้นจากเว็บไซต์ Ubidots]”

#define UBIDOTS_CONF_REMOTE_HOST “2607:f0d0:2101:39::2”

```

เมื่อแก้ไขเสร็จ กด Save เพื่อบันทึก

- 3) เขียนแอปพลิเคชัน ubidots-client.c ลงบนอุปกรณ์ Zolertia Z1 โดยใช้บรรทัดคำสั่งดังนี้

```
make clean && make ubidots-client.upload
```

คำอธิบายบรรทัดคำสั่ง

- 1) คำสั่ง make clean ใช้สำหรับลบการคอมไพล์โปรแกรมอ็อบเจกต์ก่อนหน้านี้
- 2) คำสั่ง make ubidots-client.upload ใช้สำหรับเขียนแอปพลิเคชัน ubidots-client ลงบนอุปกรณ์ Zolertia Z1



```

/*-----*/
#ifndef PROJECT_CONF_H
#define PROJECT_CONF_H
/*-----*/

/* User configuration */
#define POST_PERIOD (CLOCK_SECOND)
#define VARIABLE_BUF_LEN 16
#define UBIDOTS_CONF_AUTH_TOKEN "yiAeGrBoXQdJOFsyXLaNnynN4fXoJM"
#define VARKEY_VARIABLE_ONE "5899865276254256212365d8" //Motion1
// #define VARKEY_VARIABLE_ONE "5899865d762542561e2ab48e" //Motion2
// #define VARKEY_VARIABLE_ONE "589986647625425620c3f2e5" //Motion3

#define UBIDOTS_CONF_IN_BUFFER_SIZE 64
/*-----*/
/* IPv6 address of things.ubidots.com is "2607:f0d0:2101:39::2", leave
 * commented to resolve the host name. The NAT64 address is "::ffff:3217:7c44"
 */
#define UBIDOTS_CONF_REMOTE_HOST "2607:f0d0:2101:39::2"
/*-----*/

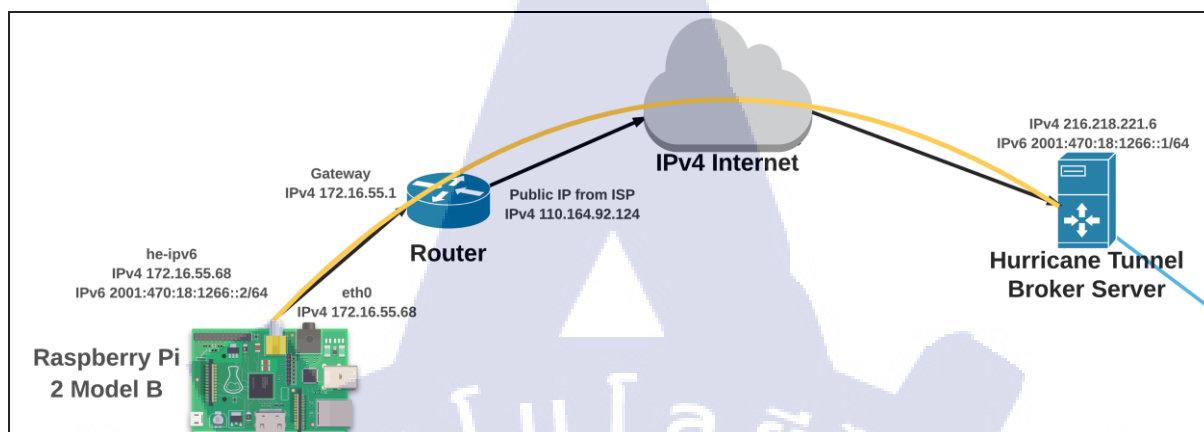
#undef NETSTACK_CONF_RADIO
#define NETSTACK_CONF_RADIO cc2538_rf_driver
#define ANTENNA_SW_SELECT_DEF_CONF ANTENNA_SW_SELECT_2_4GHZ
#define NETSTACK_CONF_RDC nullrdc_driver

#undef IEEE802154_CONF_PANID
#define IEEE802154_CONF_PANID 0xABCD
/*-----*/
/* The following are Z1 specific */
#undef RF_CHANNEL

```

รูปที่ 3.3 หน้าต่างหลังแก้ไขไฟล์ project-conf.h

3.2.2 การตั้งค่าบนอุปกรณ์ Raspberry Pi



รูปที่ 3.4 การทำงานตั้งแต่อุปกรณ์ Raspberry Pi จนถึงเซิร์ฟเวอร์ของ tunnelbroker.net

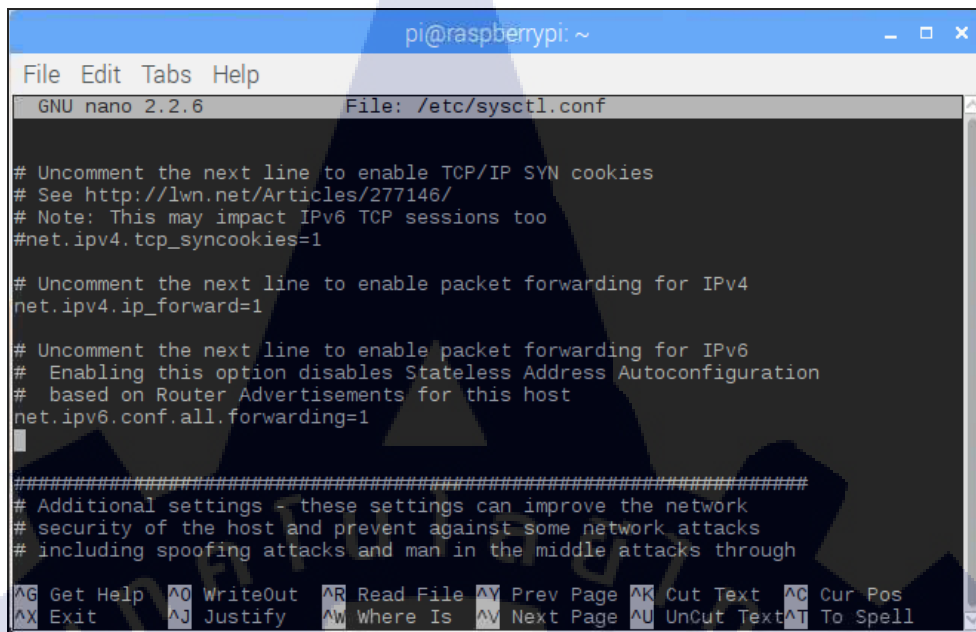
- 1) อินเทอร์เฟซ eth0 ใช้สำหรับรับไอพี ที่ทางสถาบันฯทำการ NAT กับไอพีรุ่นที่สี่แบบสาธารณะ เพื่อให้ออกสู่อินเทอร์เน็ต
- 2) อินเทอร์เฟซเสมือน tun0 ใช้สำหรับเชื่อมต่อเครือข่ายไอพีรุ่นที่หกบนเครือข่ายบุคคลไร้สายพลังงานต่ำของอุปกรณ์ Zolertia Z1 ที่ทำหน้าที่เป็นขอบเราเตอร์ เพื่อเชื่อมต่อกับอินเทอร์เฟซ eth0
- 3) อินเทอร์เฟซเสมือน he-ipv6 ใช้สำหรับเชื่อมโยงเครือข่ายไอพีรุ่นที่หกที่วิ่งผ่านเครือข่ายไอพีรุ่นที่สี่จากเว็บไซต์ tunnelbroker.net เพื่อให้ตัว Raspberry Pi 3 สามารถออกสู่อินเทอร์เน็ตผ่านระบบเครือข่ายไอพีรุ่นที่หกได้

ตั้งค่าตารางไอพีเพื่อให้การส่งกลุ่มข้อมูลสามารถวิ่งไปถึงปลายทางที่กำหนดได้ โดยการตั้งค่าให้ยอมรับ ในทุกช่องทาง ทั้งการนำเข้า การส่งต่อ และการนำออกของกลุ่มข้อมูล ตั้งค่าให้อินเทอร์เฟซ eth0 สามารถส่งต่อกลุ่มข้อมูลไปที่อินเทอร์เฟซเสมือน tun0 และตั้งค่าให้อินเทอร์เฟซเสมือน tun0 ส่งต่อกลุ่มข้อมูลกลับไปสู่อินเทอร์เฟซ eth0 ได้เช่นกัน ตั้งค่าให้อินเทอร์เฟซ eth0 ทำหน้าที่จัดหาเส้นทางให้กับเครือข่ายภายใน โดยการพิมพ์บรรทัดคำสั่งลงบนโปรแกรมคำสั่งงานดังนี้

```
iptables -I INPUT -j ACCEPT
iptables -F
iptables -X
iptables -t nat -F
iptables -t nat -X
iptables -t mangle -F
iptables -t mangle -X
iptables -P INPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -P OUTPUT ACCEPT
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
iptables -A FORWARD -i eth0 -o tun0 -m state --state RELATED,ESTABLISHED -j ACCEPT
iptables -A FORWARD -i tun0 -o eth0 -m state --state RELATED,ESTABLISHED -j ACCEPT
```

ทำการเปิดทางให้ไอพีรุ่นที่สี่และไอพีรุ่นที่หกสามารถส่งต่อกลุ่มข้อมูลได้ โดยการเข้าถึง
เพิ่มข้อมูล /etc/sysctl.conf และลบเครื่องหมาย หมายเหตุ (#) ที่อยู่ข้างหน้าบรรทัดด้านล่างนี้

```
net.ipv4.ip_forward = 1 และ
net.ipv6.conf.default.forwarding=1
```



```

pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 2.2.6 File: /etc/sysctl.conf

# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1

# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1

# Uncomment the next line to enable packet forwarding for IPv6
# Enabling this option disables Stateless Address Autoconfiguration
# based on Router Advertisements for this host
net.ipv6.conf.all.forwarding=1

#####
# Additional settings - these settings can improve the network
# security of the host and prevent against some network attacks
# including spoofing attacks and man in the middle attacks through
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

```

รูปที่ 3.5 หน้าต่างการแก้ไขไฟล์ sysctl.conf

ตั้งค่าอินเทอร์เฟซ eth0 ให้เป็นค่าคงที่เพื่อให้เส้นทางของกลุ่มข้อมูลสามารถส่งได้ถูกต้อง โดยการเข้าถึงเพิ่มข้อมูล /etc/dhcpd.conf และทำการเพิ่มบรรทัดคำสั่งด้านล่างนี้

```

interface eth0
static ip_address=172.16.55.68/24
static routers=172.16.55.1
static domain_name_servers=172.16.10.10

```

สร้างอินเทอร์เฟซเสมือน he-ipv6 เพื่อให้ Raspberry Pi สามารถใช้งานอินเทอร์เน็ตผ่าน ไอพีรุ่นที่หกได้ โดยการเพิ่มบรรทัดคำสั่งดังนี้

```
auto he-ipv6
iface he-ipv6 inet6 v4tunnel
    address 2001:470:18:126::2
    netmask 64
    endpoint 216.218.221.6
    local 172.16.55.68
    ttl 255
    gateway 2001:470:18:1266::1
```

3.2.2.2 วิธีการทำ Tunneling แบบ 6in4 โดยใช้ Hurricane Electric IPv6 Tunnel Broker

- 1) เข้าไปที่ www.tunnelbroker.net ทำการสมัครสมาชิก และ Login เพื่อเข้าใช้งาน
- 2) ให้กดที่ Create Regular Tunnel
- 3) ให้เลือก Server Tunnel ปลายทางที่อยู่ใกล้มากที่สุด แล้วใส่หมายเลข IPv4 ของเราลง (ดูได้จากบรรทัดด้านล่างตรง You are viewing from xxx.xxx.xxx) แล้วกดปุ่ม Create Tunnel ที่อยู่ด้านล่าง
- 4) เมื่อสร้างเสร็จแล้ว ก็จะขึ้นหน้าแสดงรายละเอียดต่าง ๆ ของ Tunnel จากนั้นก็ไปที่แท็บ Example Configurations
- 5) แล้วก็เลือกระบบปฏิบัติการที่เราใช้งานอยู่ ในที่นี้ใช้ Raspberry Pi ซึ่งระบบปฏิบัติการคือ Debian
- 6) ถ้าหาก IP เครื่องที่เราใช้อยู่ ไม่ใช่ Public IP จริง ก็ต้องเปลี่ยนตรงบรรทัด local ให้เป็น IP ของเครื่องเรา ในที่นี้คือเปลี่ยนจาก

```
local 160.18.2.109
```

เป็น

```
local 172.16.55.68
```

- 7) จากนั้นก็นำค่าที่ได้มาใส่ในเครื่อง เพื่อทำการตั้งค่า โดยการเปิด Terminal ขึ้นมา แล้วพิมพ์

```
sudo nano /etc/network/interfaces
```

แล้วกด Enter

- 8) เมื่อเปิดโปรแกรมขึ้นมาแล้ว ให้เลื่อนมาที่บรรทัดล่างสุด แล้วพิมพ์ตามคำสั่งในเว็บ

```
tunnelbroker.net
auto he-ipv6
iface he-ipv6 inet6 v4tunnel
    address [Client IPv6 Address]
    netmask 64
    endpoint [Server IPv4 Address]
    local [IP ของเครื่องเรา]
    ttl 255
    gateway [Server IPv6 Address]
```

แล้วกด Ctrl+X >>> กด Y >>> แล้วกด Enter

จากนั้นก็พิมพ์

```
sudo nano /etc/dhcpd.conf
```

แล้วกด Enter

- 9) เลื่อนลงมาที่บรรทัดล่างสุด ถ้าต่ออินเทอร์เน็ตผ่านสายแลนให้ใช้ eth0 แต่ถ้าต่อผ่านไวไฟให้ใช้ wlan0 แล้วใช้คำสั่งดังนี้

```
interface eth0
    static ip_address=[IP ของเครื่องเรา/netmask]
    static routers=[Default Gateway]
    static domain_name_servers=[DNS]
```

เมื่อพิมพ์เสร็จแล้วให้กด Ctrl+X >>> กด Y >>> แล้วกด Enter

จากนั้นลองใช้คำสั่ง ping6 ipv6.google.com ถ้าสามารถ ping ไปหา IPv6 ของ Google ได้ ก็แสดงว่าทำสำเร็จ

HURRICANE ELECTRIC
INTERNET SERVICES

Account Menu
Main Page
Account Info
Logout

User Functions
Create Regular Tunnel
Create BGP Tunnel
IPv6 Portscan

Tunnel Details
IPv6 Tunnel | Example Configurations | Advanced

Tunnel ID: 367917 Delete Tunnel
Creation Date: Oct 13, 2016
Description:

IPv6 Tunnel Endpoints
Server IPv4 Address: 216.218.221.6
Server IPv6 Address: 2001:470:18:1266::1/64
Client IPv4 Address: 110.164.92.124
Client IPv6 Address: 2001:470:18:1266::2/64

Routed IPv6 Prefixes
Routed /64: 2001:470:19:1266::/64
Routed /48: [Assign /48](#)

DNS Resolvers
Anycast IPv6 Caching Nameserver: 2001:470:20::2
Anycast IPv4 Caching Nameserver: 74.82.42.42

rDNS Delegations [Edit](#)
rDNS Delegated NS1:
rDNS Delegated NS2:
rDNS Delegated NS3:
rDNS Delegated NS4:
rDNS Delegated NS5:

Quick Links
[Certification](#)
[Tunnelbroker](#)
[Free DNS](#)
[BGP Toolkit](#)
[Net Tools App](#)
[Forums](#)
[FAQ](#)
[Video Presentations](#)
[Usage Statistics](#)
[Tunnel Server Status](#)
[Network Map](#)
[Looking Glass \(v4/v6\)](#)
[Route Server \(telnet\)](#)
[Global IPv6 Report](#)

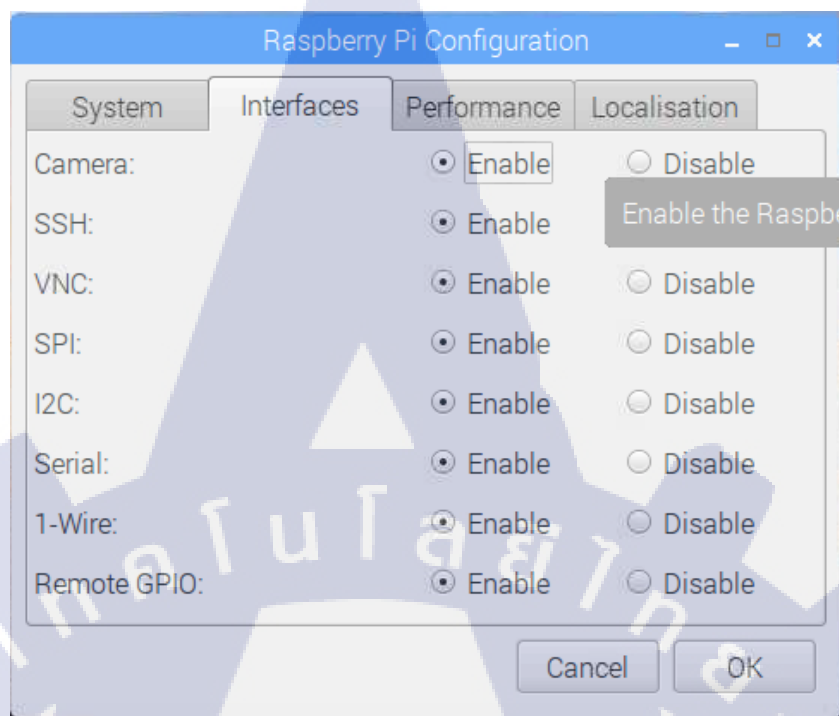
Services
[Transit](#)
[Colocation](#)
[Dedicated Servers](#)

v4 Exhaustion
IPv4 & IPv6 Statistics
RIR v4 IPs Left
AfriNIC 18,062,610
APNIC 6,891,757
ARIN 0
LACNIC 38,366
RIPE 12,549,131
v6 ASNs

รูปที่ 3.6 หน้าต่างหลังสร้างอุโมงค์ที่เว็บไซต์ tunnelbroker.net สำเร็จ

3.2.2.3 การต่อกล้อง Raspberry Pi NoIR Camera กับการเปิดใช้งาน

ในแพ็คเกจของกล้อง Raspberry Pi NoIR Camera V2 จะมีตัวกล้องติดมาพร้อมกับสายแพ ให้เรานำสายแพมาเสียบกับช่อง Camera ของอุปกรณ์ Raspberry Pi จากนั้นให้เข้าไปตั้งค่าที่ Raspberry Pi Configuration แล้วไปเลือก Enable ที่หัวข้อ Camera ที่แท็บ Interface



รูปที่ 3.7 หน้าต่างการตั้งค่าของอุปกรณ์ Raspberry Pi

3.2.2.4 การติดตั้ง Boto3 และ AWS CLI เพื่อใช้งาน Amazon Simple Storage Services

การติดตั้ง Boto3 ให้พิมพ์คำสั่งลงใน Command-line ดังนี้

```
sudo pip install boto3
```

การติดตั้ง AWS CLI ให้พิมพ์คำสั่งลงใน Command-line ดังนี้

```
sudo pip install awscli
```

จากนั้นพิมพ์คำสั่งเพื่อยืนยันตัวตนดังนี้

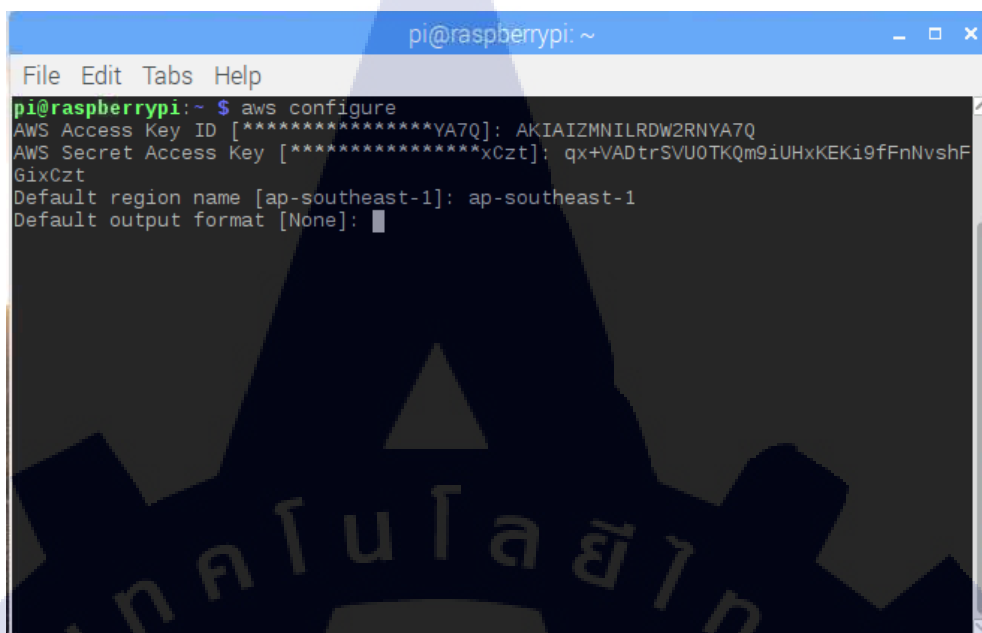
```
aws configure
```

AWS Access Key ID : [ใส่ key id ที่ได้จาก AWS]

AWS Secret Access Key : [ใส่ secret key ที่ได้จาก AWS]

Default Region Name : ap-southeast-1

Default output format : [เว้นว่าง]



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~$ aws configure  
AWS Access Key ID [*****YA7Q]: AKIAIZMNILRDW2RNYA7Q  
AWS Secret Access Key [*****xCzt]: qx+VADtrSVU0TKQm9iUHxKEKi9fFnNvshF  
GixCzt  
Default region name [ap-southeast-1]: ap-southeast-1  
Default output format [None]:
```

รูปที่ 3.8 หน้าต่างการตั้งค่าของ AWS CLI

3.2.2.5 เขียนโปรแกรมให้กล้องถ่ายรูปเมื่อตรวจจับความเคลื่อนไหวได้ และเมื่อถ่ายรูปแล้วให้อัปโหลดรูปขึ้น Amazon Simple Storage Services

ใช้คำสั่งเพื่อติดตั้งไลบรารีของ Ubidots เพื่อใช้ติดต่อกับเว็บไซต์ ดังนี้

```
sudo pip install ubidots
```

จากนั้น เขียนโปรแกรมด้วยภาษาไพธอน ดังนี้

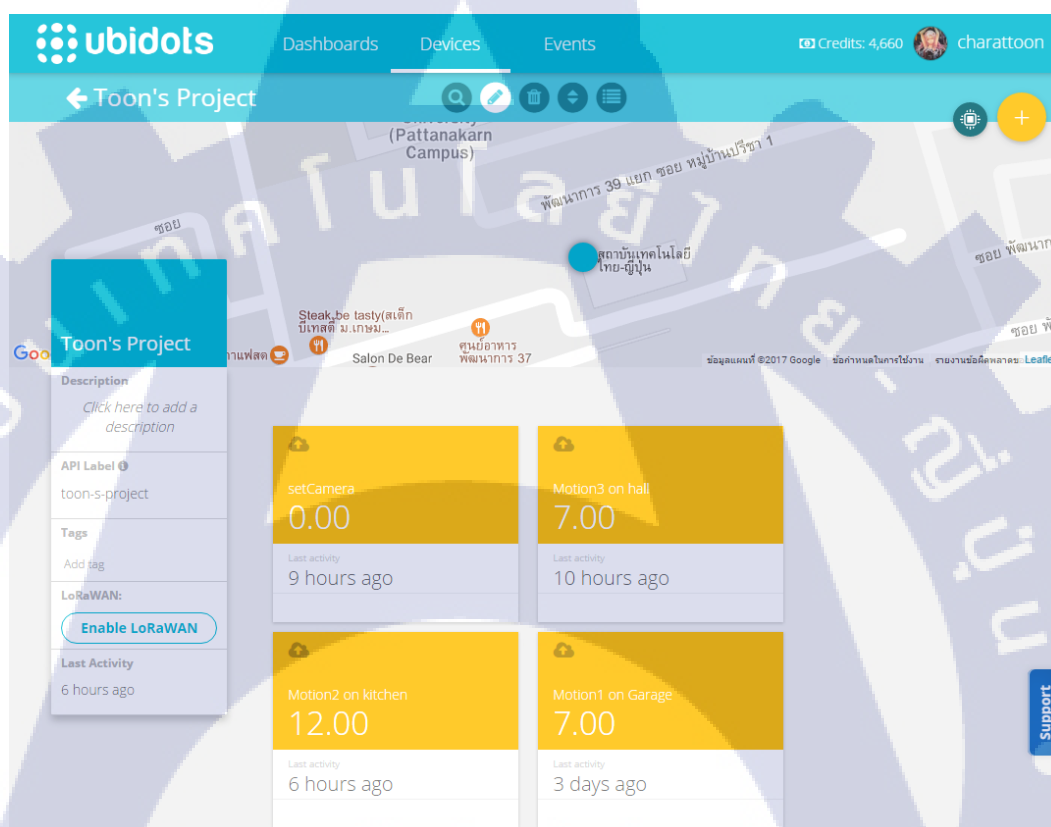
```
from ubidots import ApiClient
from picamera import PiCamera
from time import sleep
import boto3
import sys
import os

camera = PiCamera()
camera.rotation = 180
camera.resolution = (1920,1080)
api = ApiClient(token='[ใส่ token key ที่ได้จากเว็บไซต์ ubidots]')
variable = api.get_variable("[ใส่ variable key ที่สร้างจากเว็บไซต์ ubidots]")
arg = sys.argv
i = int(arg[1])
s3 = boto3.resource('s3')

while(1):
    last_value = variable.get_values(1)
    print last_value[0]['value']
    if(last_value[0]['value'] == 1):
        camera.capture('/home/pi/Pictures/image%s.jpg' %i)
        data=open('/home/pi/Pictures/image%s.jpg' %i, 'rb')
        obj=s3.bucket('homesecond').put_object(Key='pic/image%s.jpg'
            %i, Body=data)
        obj.Acl.put(ACL='public-read')
        os.remove('/home/pi/Pictures/image%s.jpg' %i)
    sleep(1)
```

3.2.3 การตั้งค่าบนเว็บไซต์ Ubidots

- 1) เข้าไปที่ <https://app.ubidots.com/> และทำการสมัครสมาชิก
- 2) ไปที่แท็บ Devices กดเครื่องหมาย + เพื่อสร้าง Device
- 3) เมื่อสร้างเสร็จ คลิกเข้าไปใน Device แล้วกด + เพื่อสร้าง Variable key สำหรับรับ-ส่งค่าจากเซ็นเซอร์



รูปที่ 3.9 หน้าต่างการสร้างตัวแปรเพื่อรับค่าจากเซ็นเซอร์

- 4) คลิกที่แท็บ Events กด + เพื่อสร้างการแจ้งเตือนเมื่อเกิดการเคลื่อนไหวขึ้น
 - เลือก Device
 - เลือก Variable
 - เลือก greater มากกว่า 1000

-จากนั้นเลือกให้ส่งทางอีเมลและข้อความมือถือ หรือจะให้ส่งค่ากลับไปยังอุปกรณ์ Raspberry Pi เพื่อทำการถ่ายรูป แล้วกด Finish

Select a Device

Filter Devices

Select a Variable

Filter Variables

if

Value

Motion1 on Garage

less greater less or equal greater or equal equal

than

1000

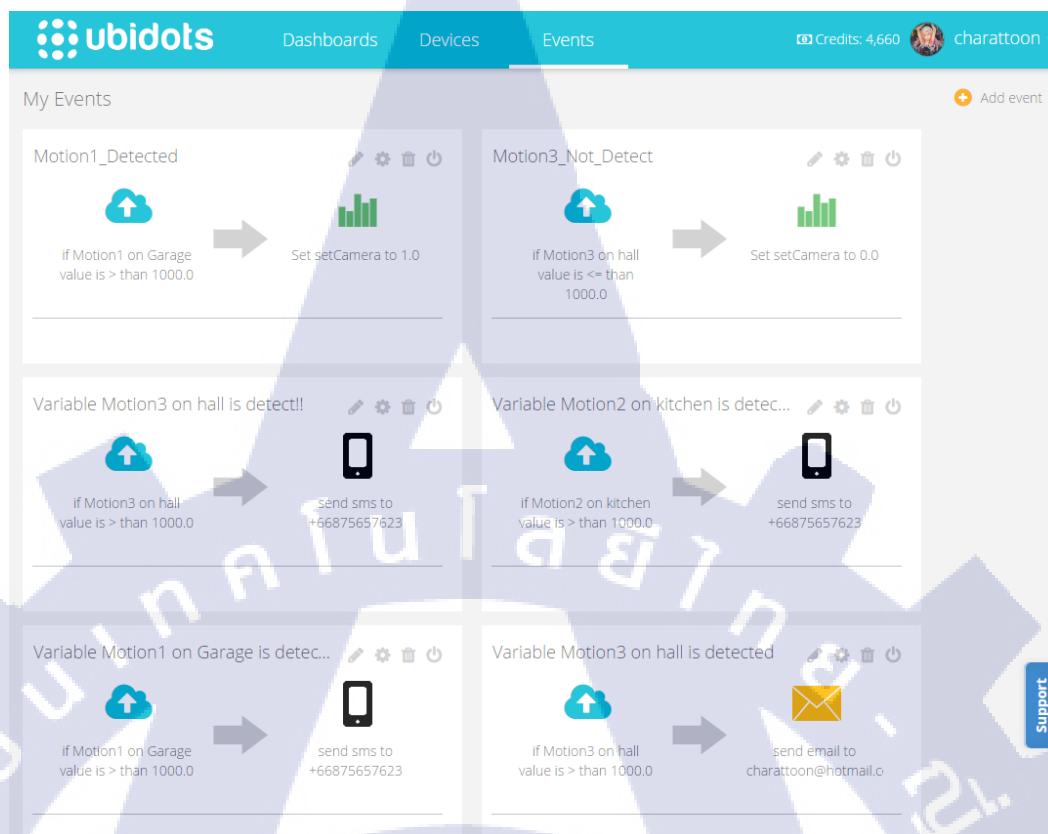
Continue

Send e-mail Send SMS Send Telegram WebHook Set a variable

Email charattoon@hotmail.com

Add email

รูปที่ 3.10 หน้าต่างการตั้งค่า Events ในเว็บไซต์ Ubidots

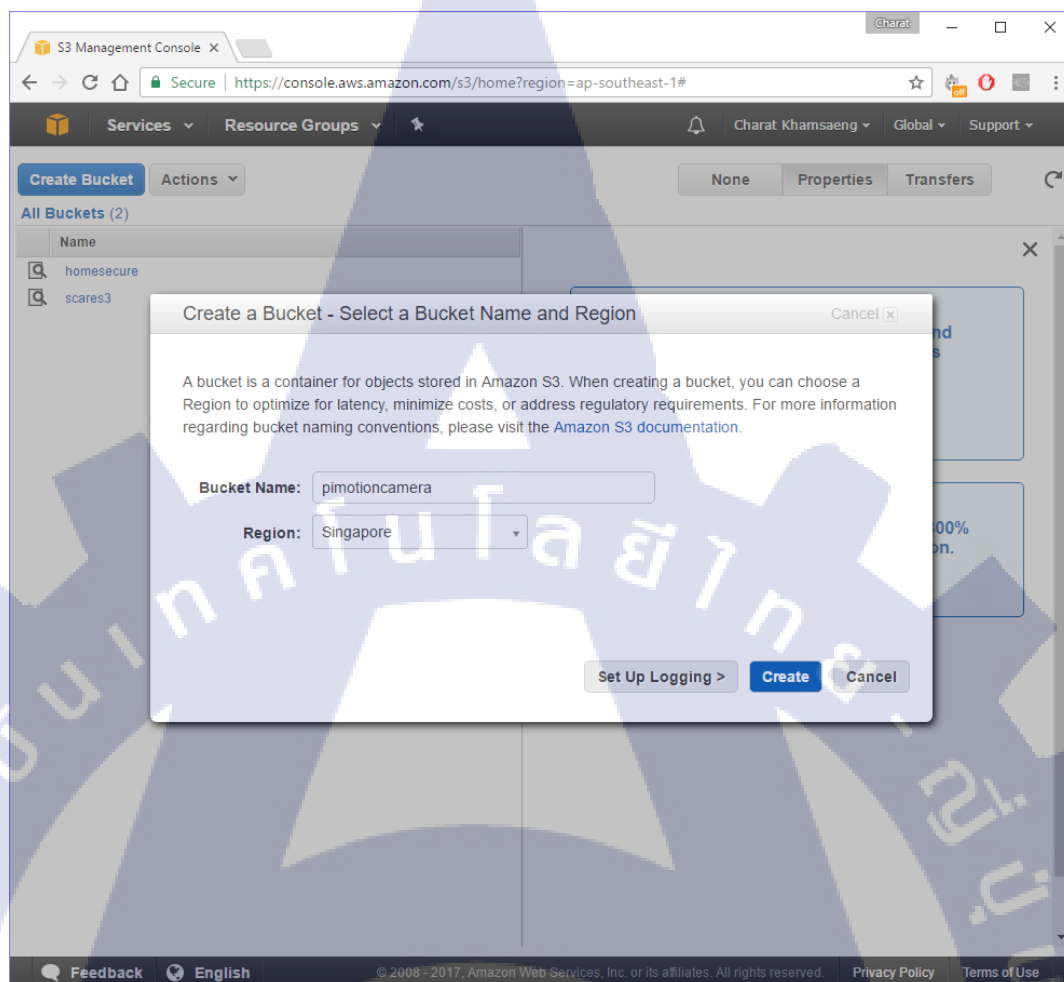


รูปที่ 3.11 หน้าต่างเมื่อสร้าง Events สำเร็จ

เพียงเท่านี้ เมื่อเกิดการเคลื่อนไหว ระบบก็จะทำการแจ้งเตือน ไปยังอีเมลและข้อความมือถือ

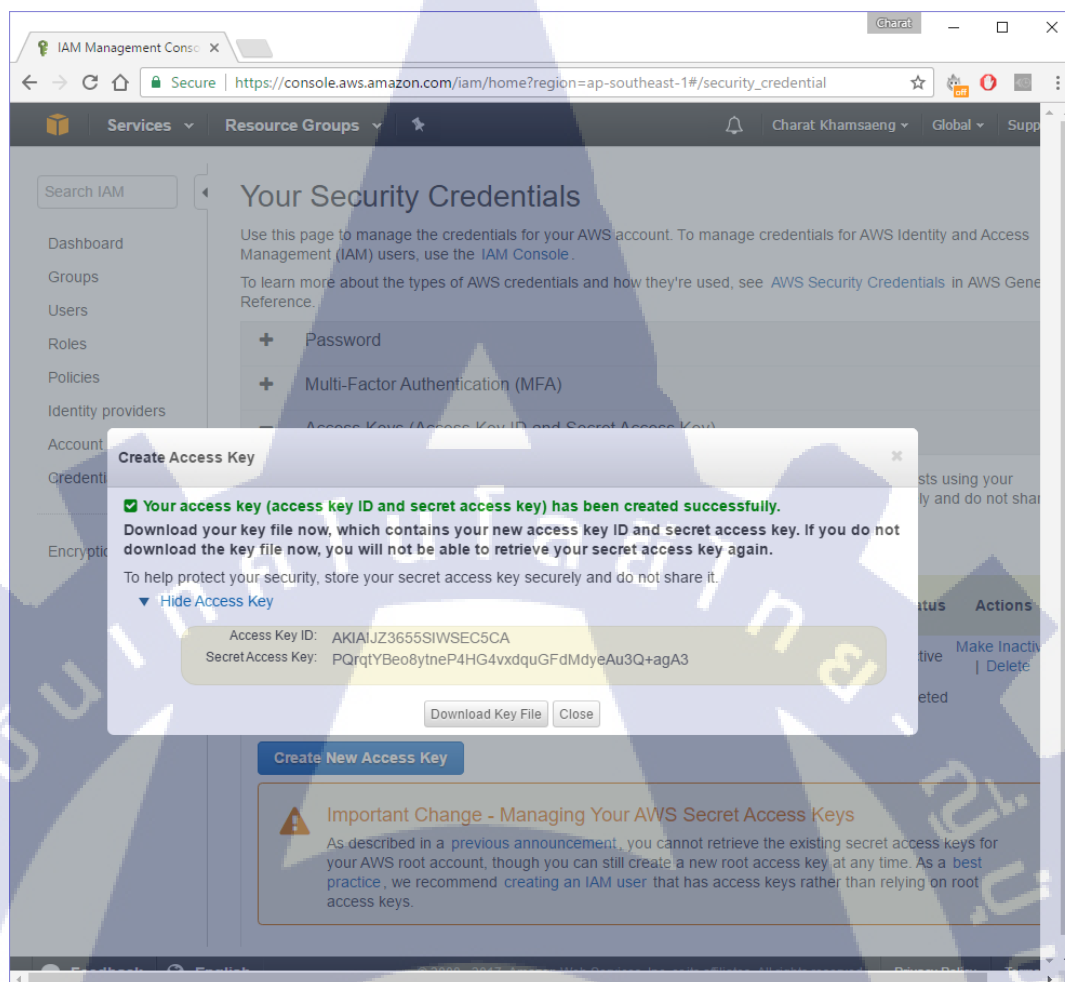
3.2.4 การตั้งค่าบนเว็บไซต์ Amazon Simple Storage Services

- 1) เข้าไปที่ <https://console.aws.amazon.com/console/home> แล้วทำการสมัครสมาชิก
- 2) เลือกบริการ Storage S3
- 3) กด Create Bucket > ใส่ชื่อของ Bucket แล้วเลือกภูมิภาค จากนั้นกด Create



รูปที่ 3.12 ขั้นตอนการสร้าง Bucket

- 4) กดที่ชื่อข้างบนขวา แล้วเลือก My Security Credentials
- 5) คลิกที่ Access Keys (Access Key ID and Secret Access Key) แล้วกด Create New Access Key



รูปที่ 3.13 หน้าต่างการขอ Access Key และ Secret Access Key

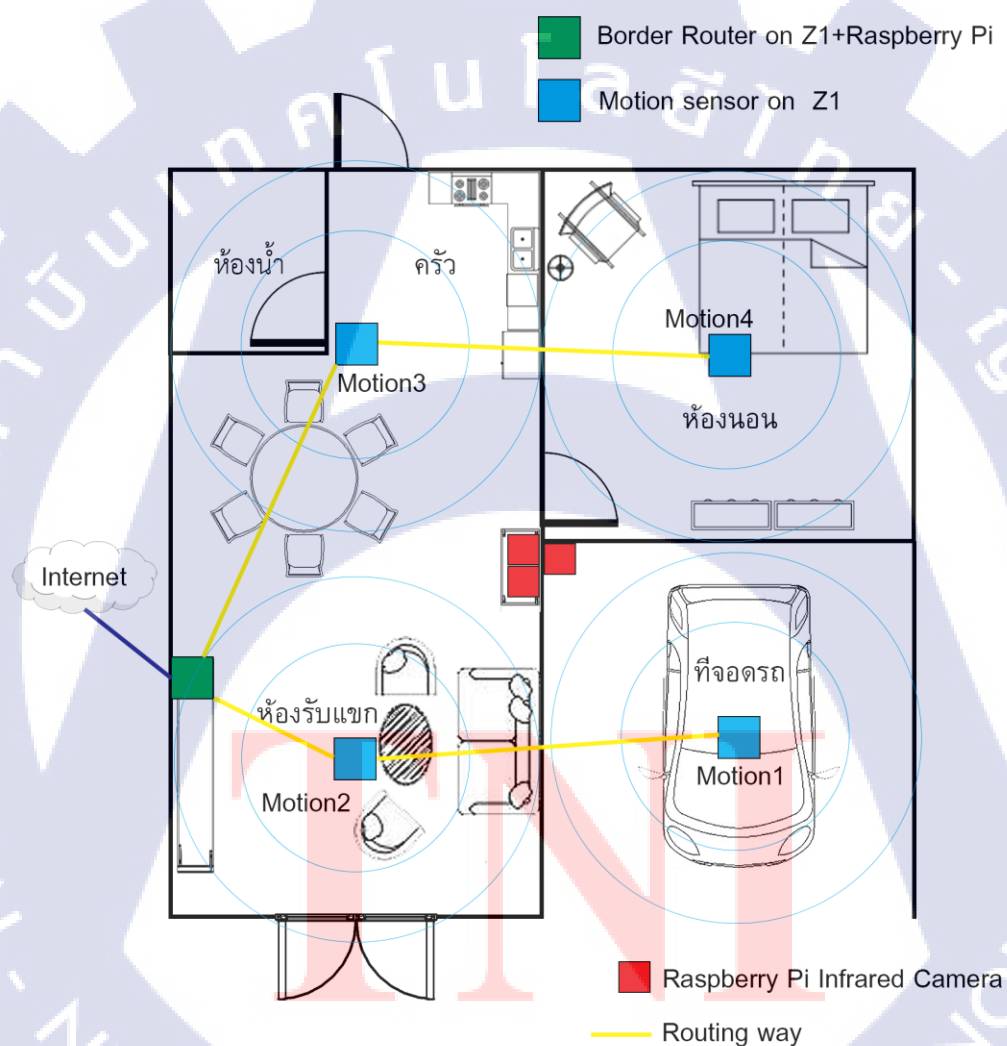
- 6) จะปรากฏหน้าต่าง Access Key และ Secret Access Key ขึ้นมา ให้กด Download Key File ไว้ใช้ยืนยันตัวตนบนอุปกรณ์ Raspberry Pi

บทที่ 4

สรุปผลการดำเนินงาน การวิเคราะห์และสรุปผลต่าง ๆ

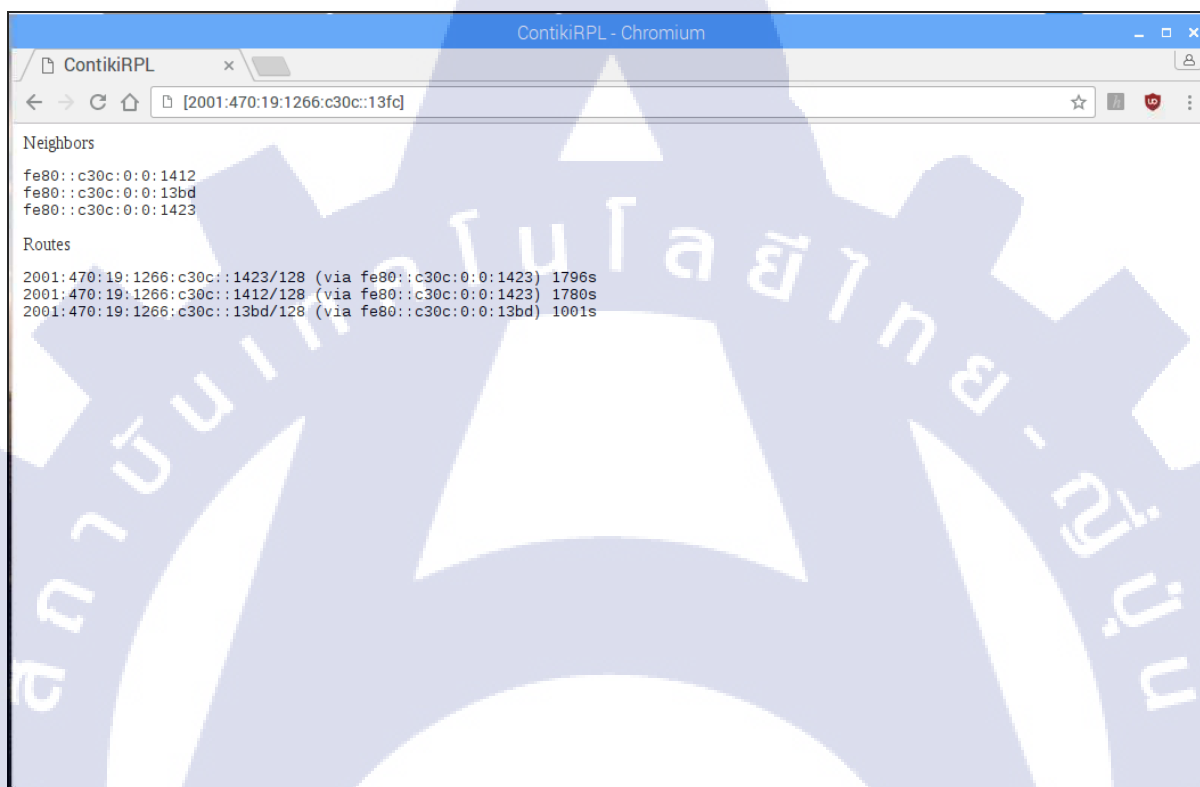
4.1 ผลการดำเนินงาน

4.1.1 การติดตั้งในสถานที่จริง



รูปที่ 4.1 ฟังก์ชันการติดตั้งอุปกรณ์ภายในบ้าน

การติดตั้งระบบป้องกันการบุกรุกบ้าน จากรูปที่ 4.1.1 จะเห็นได้ว่าการติดตั้ง Motion Sensor ไว้ที่ห้องรับแขก ห้องครัว ห้องนอน รวมถึงที่จอดรถด้วย โดยตัว Motion4 จะมีการส่งข้อมูล hop-by-hop ผ่านตัว Motion3 ไปยัง Border Router และตัว Motion1 ส่งข้อมูลผ่านตัว Motion2 ไปยัง Border Router เพื่อออกสู่อินเทอร์เน็ต และมีการติดตั้งกล้อง Raspberry Pi NoIR Camera ไว้ที่ห้องรับแขก ห้องครัว และที่จอดรถ



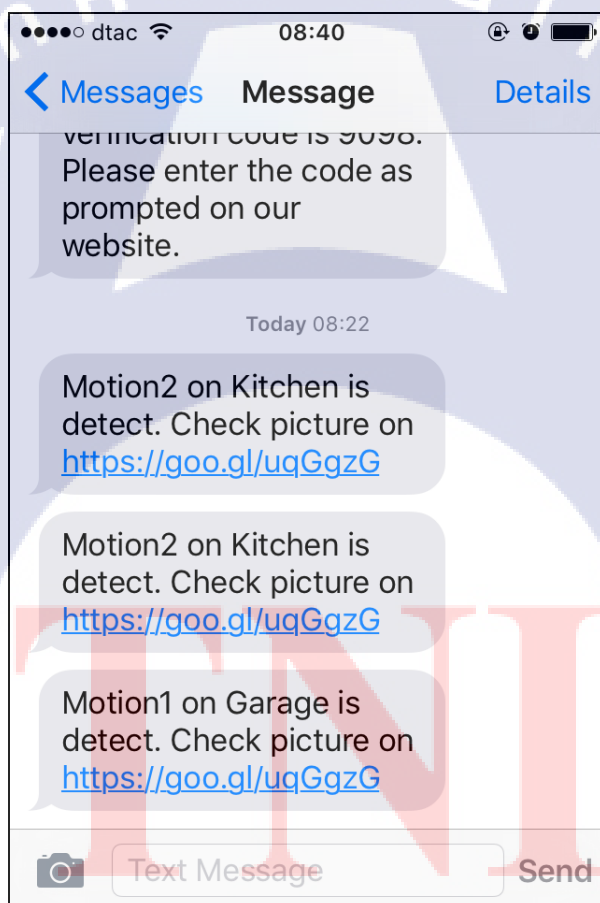
รูปที่ 4.2 หน้าต่างแสดงตารางการเชื่อมต่ออุปกรณ์ผ่าน RPL โพรโทคอล

โดยจะสามารถตรวจสอบการเชื่อมต่อได้จากการเปิดเว็บเบราว์เซอร์ขึ้นมา แล้วใส่ IPv6 ของตัว Border Router ก็จะขึ้นหน้าต่างแสดงตารางของอุปกรณ์ที่เชื่อมต่ออยู่ผ่าน RPL โพรโทคอล ดังรูปที่ 4.1.2 จะเห็นว่าอุปกรณ์ที่มีไอพีรุ่นที่หก หมายเลข ::1412 ได้เชื่อมต่อกับ Border Router ผ่านอุปกรณ์ที่มีไอพีรุ่นที่หก หมายเลข ::1423

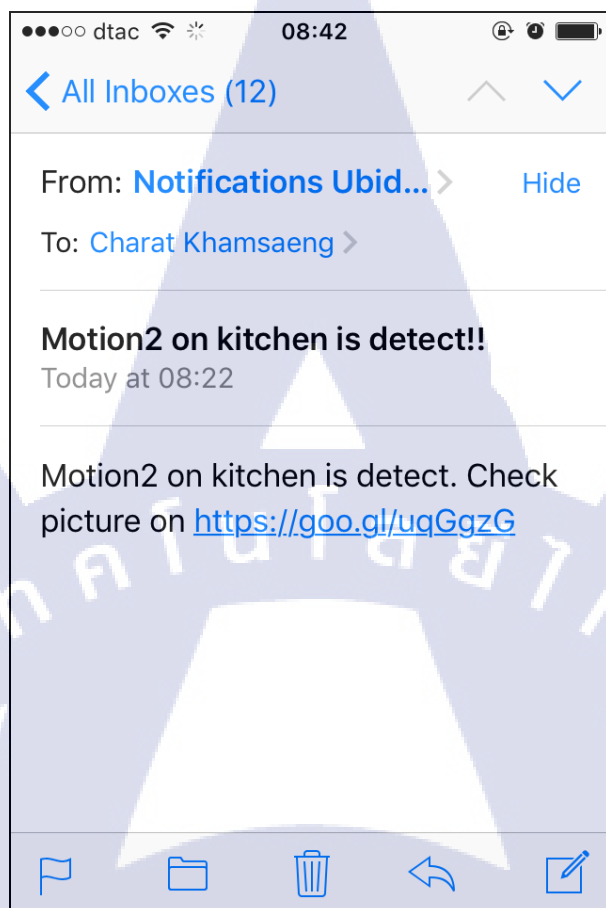
4.1.2 สามารถแจ้งเตือนผ่านอีเมลและข้อความมือถือ

การทดสอบอุปกรณ์ในสถานการณ์จำลอง โดยใช้ห้องวิจัย A604 ชั้น 6 อาคาร A สถาบันเทคโนโลยีไทย-ญี่ปุ่น เป็นสถานที่ทดสอบ ได้มีการติดตั้ง Motion Sensor ไว้ที่ประตูของห้องวิจัยทั้ง 2 บาน และใช้กล้อง Raspberry Pi NoIR Camera ในการถ่ายรูป โดยรูปที่ถ่ายได้จะเห็นประตูครบทั้ง 2 บาน โดยบานแรกใช้ Motion1 ในการตรวจจับ และบานที่ 2 ใช้ Motion2 ในการตรวจจับ

ทดสอบโดยเริ่มจากเดินผ่านประตูที่ 2 ตัว Motion2 ก็จับความเคลื่อนไหวได้ และส่งทั้งอีเมลและข้อความมือถือมายังผู้ใช้ และเมื่อเดินผ่านประตูที่ 1 ตัว Motion1 ก็จับความเคลื่อนไหวได้เช่นกัน และมีการส่งอีเมลและข้อความมือถือมายังผู้ใช้ด้วยเช่นกัน ดังรูปที่ 4.3 และรูปที่ 4.4



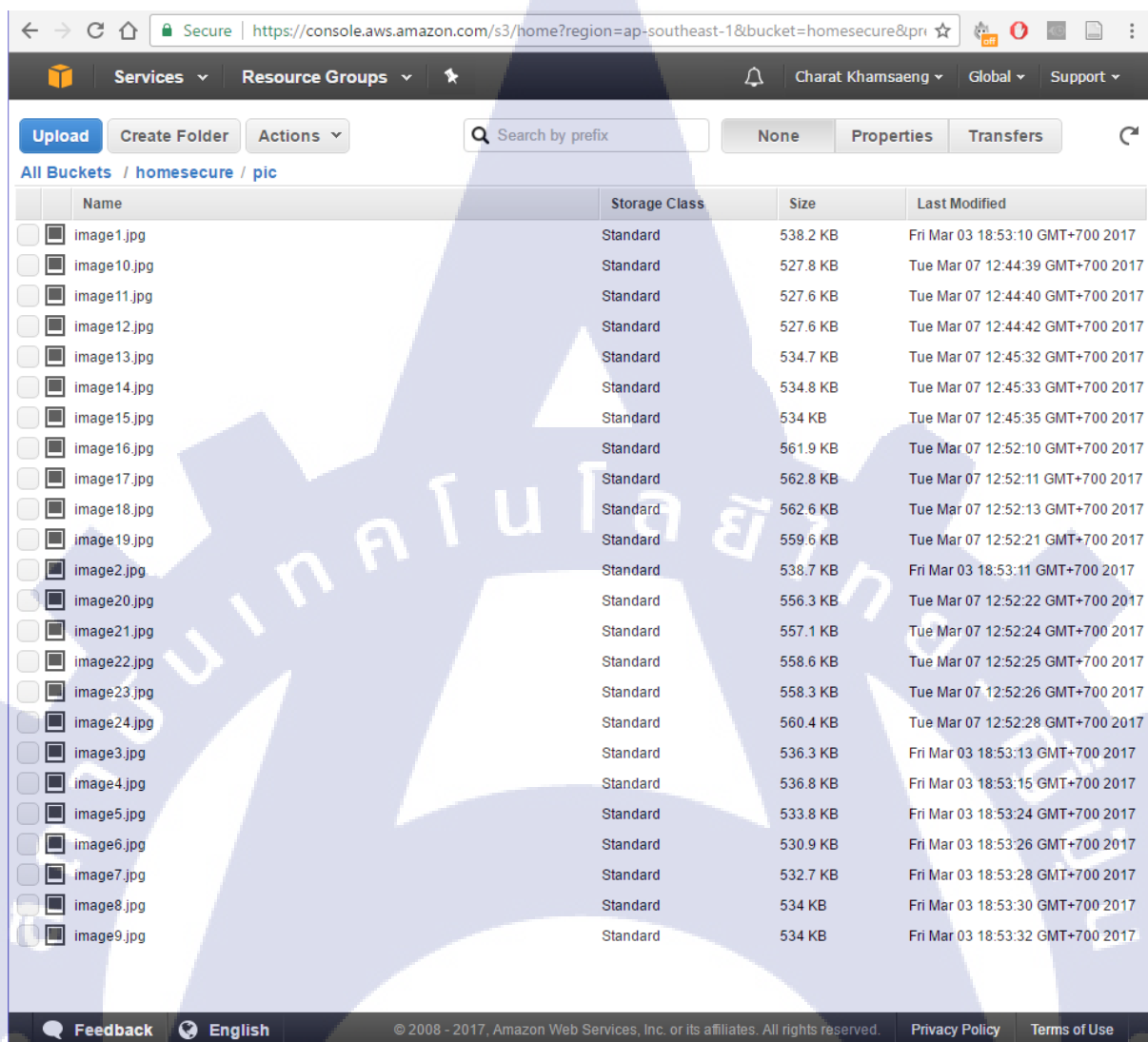
รูปที่ 4.3 ข้อความมือถือที่ถูกส่งมาเมื่อมีการตรวจจับความเคลื่อนไหว



รูปที่ 4.4 อีเมลที่ถูกส่งมาเมื่อมีการตรวจจับความเคลื่อนไหว

4.1.3 สามารถอัปโหลดรูปถ่ายขึ้น Amazon Simple Storage Services

เมื่อถ่ายรูปแล้ว ก็ทำการอัปโหลดรูปภาพขึ้นไปยัง Amazon Simple Storage Services เพื่อทำการเก็บไว้ แล้วจึงลบรูปภาพออกจากอุปกรณ์ Raspberry Pi ซึ่งสามารถอัปโหลดรูปถ่ายได้ และสามารถลบรูปภาพหลังจากอัปโหลดได้เช่นกัน



Secure | <https://console.aws.amazon.com/s3/home?region=ap-southeast-1&bucket=homesecond&pr>

Services Resource Groups Search by prefix None Properties Transfers

All Buckets / homesecond / pic

Name	Storage Class	Size	Last Modified
image1.jpg	Standard	538.2 KB	Fri Mar 03 18:53:10 GMT+700 2017
image10.jpg	Standard	527.8 KB	Tue Mar 07 12:44:39 GMT+700 2017
image11.jpg	Standard	527.6 KB	Tue Mar 07 12:44:40 GMT+700 2017
image12.jpg	Standard	527.6 KB	Tue Mar 07 12:44:42 GMT+700 2017
image13.jpg	Standard	534.7 KB	Tue Mar 07 12:45:32 GMT+700 2017
image14.jpg	Standard	534.8 KB	Tue Mar 07 12:45:33 GMT+700 2017
image15.jpg	Standard	534 KB	Tue Mar 07 12:45:35 GMT+700 2017
image16.jpg	Standard	561.9 KB	Tue Mar 07 12:52:10 GMT+700 2017
image17.jpg	Standard	562.8 KB	Tue Mar 07 12:52:11 GMT+700 2017
image18.jpg	Standard	562.6 KB	Tue Mar 07 12:52:13 GMT+700 2017
image19.jpg	Standard	559.6 KB	Tue Mar 07 12:52:21 GMT+700 2017
image2.jpg	Standard	538.7 KB	Fri Mar 03 18:53:11 GMT+700 2017
image20.jpg	Standard	556.3 KB	Tue Mar 07 12:52:22 GMT+700 2017
image21.jpg	Standard	557.1 KB	Tue Mar 07 12:52:24 GMT+700 2017
image22.jpg	Standard	558.6 KB	Tue Mar 07 12:52:25 GMT+700 2017
image23.jpg	Standard	558.3 KB	Tue Mar 07 12:52:26 GMT+700 2017
image24.jpg	Standard	560.4 KB	Tue Mar 07 12:52:28 GMT+700 2017
image3.jpg	Standard	536.3 KB	Fri Mar 03 18:53:13 GMT+700 2017
image4.jpg	Standard	536.8 KB	Fri Mar 03 18:53:15 GMT+700 2017
image5.jpg	Standard	533.8 KB	Fri Mar 03 18:53:24 GMT+700 2017
image6.jpg	Standard	530.9 KB	Fri Mar 03 18:53:26 GMT+700 2017
image7.jpg	Standard	532.7 KB	Fri Mar 03 18:53:28 GMT+700 2017
image8.jpg	Standard	534 KB	Fri Mar 03 18:53:30 GMT+700 2017
image9.jpg	Standard	534 KB	Fri Mar 03 18:53:32 GMT+700 2017

Feedback English © 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

รูปที่ 4.5 รูปถ่ายที่ถูกอัปโหลดขึ้นไป Amazon Simple Storage Services

4.1.4 แสดงการถ่ายรูปของกล้อง Raspberry Pi NoIR Camera

กล้อง Raspberry Pi NoIR Camera จะถ่ายรูปก็ต่อเมื่อ Motion Sensor ตรวจจับความเคลื่อนไหวได้ แล้วส่งค่าขึ้นไปเว็บไซต์ Ubidots จากนั้น เว็บไซต์ Ubidots จะส่งค่ากลับมายัง Raspberry Pi เพื่อสั่งให้กล้องถ่ายรูป โดยจากสถานการณ์จำลอง กล้องสามารถถ่ายรูปได้ตามที่ต้องการดังรูปที่ 4.6 และรูปที่ 4.7



รูปที่ 4.6 ภาพที่ถูกถ่ายหลังจาก Motion2 ตรวจจับความเคลื่อนไหวได้



รูปที่ 4.7 ภาพที่ถูกถ่ายหลังจาก Motion1 ตรวจจับความเคลื่อนไหวได้

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

อุปกรณ์ Zolertia Z1 สามารถทำงานร่วมกับเซ็นเซอร์ได้ดี สามารถตรวจจับความเคลื่อนไหวได้ สามารถใช้งานไอฟิรน์ที่หกดได้ โดยการทำอุโมงค์ไอฟิรน์ที่หกดให้วิ่งผ่านเครือข่ายไอฟิรน์ที่สี่ สามารถส่งค่าของเซ็นเซอร์ขึ้นไปยังเว็บไซต์ของ Ubidots ได้ สามารถทำการแจ้งเตือนด้วยอีเมลล์และข้อความมือถือได้ในทันทีที่ Motion Sensor ตรวจจับความเคลื่อนไหวได้ และสามารถใช้กล้อง Raspberry Pi NoIR Camera ถ่ายรูปเพื่อระบุตัวตนของผู้บุกรุกได้ แต่เนื่องจากข้อจำกัดของอุปกรณ์ Zolertia Z1 ที่ช่องต่อเซ็นเซอร์ที่ใช้ไฟ 5V จะต้องต่อไฟเลี้ยงตลอดเวลา ทำให้ไม่สามารถใช้งานแบบไร้สายได้ และสถานที่ที่ทำการทดลองยังไม่รองรับไอฟิรน์ที่หกดโดยตรง จึงจำเป็นต้องทำอุโมงค์ไอฟิรน์ที่หกด ซึ่งทำให้เกิดดีเลย์ในการเชื่อมต่อมากกว่าปกติ และสุดท้ายเรื่องของเว็บไซต์ Ubidots ที่มีข้อจำกัดในการรับค่าของเซ็นเซอร์ ที่สามารถรับได้เร็วสุดคือ ทุกๆ 5 วินาที จึงทำให้การจรวจจับไม่ทันทีเท่าที่ควร

5.2 แนวทางการแก้ไข้ปัญหา

- 1) ทำการขอไอฟิรน์ที่หกดจากผู้ให้บริการอินเทอร์เน็ต
- 2) ใช้แบตเตอรี่แทนการต่อสายไฟเลี้ยง
- 3) ค้นหาเพิ่มเติมหรือพัฒนาแพลตฟอร์มของตัวเองให้สามารถรับค่าได้ทุกๆวินาที

5.3 ข้อเสนอแนะ

เรื่องของ IPv6 และอุปกรณ์ IoT ยังเป็นเรื่องใหม่ที่ยังสามารถเรียนรู้ได้อีกมาก และสามารถนำความรู้เหล่านั้นมาประยุกต์ใช้ในชีวิตประจำวันให้เกิดประโยชน์ได้อีกมากมาย

เอกสารอ้างอิง

ควรปราโมทย์ ปิยะชัย, “Application of Embedded Linux Base: เซ็นเซอร์ตรวจจับความเคลื่อนไหว (Motion Sensor),” Application of Embedded Linux Base, 2553. .

“Boto 3 Documentation — Boto 3 Docs 1.4.4 documentation.” [Online]. Available: <http://boto3.readthedocs.io/en/latest/index.html>. [Accessed: 12-Jan-2017].

“Configuring the AWS Command Line Interface - AWS Command Line Interface.” [Online]. Available: <http://docs.aws.amazon.com/cli/latest/userguide/cli-chap-getting-started.html>. [Accessed: 12-Jan-2017].

“EdgeMAX - วิธีการสร้าง IPv6 Tunnel ด้วย Hurricane Electric TunnelBroker in Edgemax และตัวอย่างการตั้งค่าเบื้องต้น - Page 1.” [Online]. Available: <http://www.ubiquiticlub.com/edgemax/21/edgemax-ipv6-tunnel-hurricane-electric-tunnelbroker/206/>. [Accessed: 16-Dec-2016].

“Getting Started with Amazon Simple Storage Service - Amazon Simple Storage Service.” [Online]. Available: <http://docs.aws.amazon.com/AmazonS3/latest/gsg/GetStartedWithS3.html>. [Accessed: 12-Jan-2017].

“Getting started with picamera | Raspberry Pi Learning Resources.” [Online]. Available: <https://www.raspberrypi.org/learning/getting-started-with-picamera/>. [Accessed: 20-Jan-2017].

“How to give your Raspberry Pi a Static IP Address - UPDATE,” Cases for your Raspberry Pi. [Online]. Available: <https://www.modmypi.com/blog/how-to-give-your-raspberry-pi-a-static-ip-address-update>. [Accessed: 18-Dec-2016].

“Hurricane Electric Free IPv6 Tunnel Broker.” [Online]. Available: <https://tunnelbroker.net/>. [Accessed: 16-Dec-2016].

“IoT platform | Internet of Things | Ubidots.” [Online]. Available: <https://ubidots.com/>. [Accessed: 18-Dec-2016].

“IPv4,” Wikipedia. 16-Dec-2016.

“IPv6,” Wikipedia. 16-Dec-2016.

“networking - How to route traffic from a specific port on tun0 to eth0 (and back) - Raspberry Pi Stack Exchange.” [Online]. Available: <http://raspberrypi.stackexchange.com/questions/57380/how-to-route-traffic-from-a-specific-port-on-tun0-to-eth0-and-back>. [Accessed: 20-Dec-2016].

“PIR Motion Sensor Overview.” [Online]. Available: <http://irish7.exteen.com/>. [Accessed: 12-Jan-2017].

“RPi-Cam-Web-Interface - eLinux.org.” [Online]. Available: <http://elinux.org/RPi-Cam-Web-Interface>. [Accessed: 21-Jan-2017].

“Worksheet - Infrared Bird Box | Raspberry Pi Learning Resources.” [Online]. Available: <https://www.raspberrypi.org/learning/infrared-bird-box/worksheet/>. [Accessed: 21-Jan-2017].

“Zolertia/Resources,” GitHub. [Online]. Available: <https://github.com/Zolertia/Resources>. [Accessed: 18-Dec-2016].

“-IPv4 และ IPV6 แตกต่างกันอย่างไร - sawitree I’nong.” [Online]. Available: <https://sites.google.com/site/sawitree1008/bthkhwam/-ipv4-laea-ipv6-taek-tang-kan-xyangri>. [Accessed: 24-Nov-2016].

ประวัติผู้จัดทำโครงการ

ชื่อ – สกุล

ชรัตน์ จำแสง

วัน เดือน ปีเกิด

05 กันยายน พ.ศ. 2537

ประวัติการศึกษา

ระดับประถมศึกษา

ประถมศึกษาตอนต้น-ตอนปลาย พ.ศ. 2544

โรงเรียนวันทามรือา ราชบุรี

ระดับมัธยมศึกษา

มัธยมศึกษาตอนต้น-ตอนปลาย (สายวิทยาศาสตร์-คณิตศาสตร์) พ.ศ. 2550

โรงเรียนสายธรรมจันทร์

ระดับอุดมศึกษา

คณะเทคโนโลยีสารสนเทศ สาขาเทคโนโลยีสารสนเทศ พ.ศ. 2556

สถาบันเทคโนโลยีไทย-ญี่ปุ่น

ทุนการศึกษา

- ไม่มี -

ประวัติการฝึกอบรม

อบรมการทำเคเบิลได้น้ำของ CAT Telecom

ผลงานที่ได้รับการตีพิมพ์

- ไม่มี -

TNI