

ระบบรักษาความปลอดภัยภายในบ้านโดยอาศัย IoT
IoT based Home Security System



ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561

ระบบรักษาความปลอดภัยภายในบ้านโดยอาศัย IoT

IoT based Home Security System



โดย
นายอัฐพงศ์ ภูฤกษ์เงิน 56011448

อาจารย์ที่ปรึกษา
รศ.ดร. ยุทธพงษ์ รังสรรค์เสรี

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561

ผ่านการตรวจรูปเล่มแล้ว
(.....)
อาจารย์ที่ปรึกษา
24/5/62
วิศวกรรมโทรคมนาคม
Telecommunications Engineering

ผ่านการตรวจชิ้นงานแล้ว
(.....)
กรรมการผู้ตรวจชิ้นงาน
24/5/62
วิศวกรรมโทรคมนาคม
Telecommunications Engineering

ปริญญาโทปีการศึกษา 2561

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบรักษาความปลอดภัยภายในบ้านโดยอาศัย IOT

IOT BASED HOME SECURITY SYSTEM

ผู้จัดทำ

1. นายอัฐพงศ์

ภูถุกเงิน

56011448

(รศ.ดร. ยุทธพงษ์ รัชสรค์เสรี)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงการเรื่อง ระบบรักษาความปลอดภัยภายในบ้านโดยอาศัย IoT ฉบับนี้สำเร็จไปได้ด้วยดี ด้วยความอนุเคราะห์ของบุคคลหลายท่าน ซึ่งไม่อาจจะนำมากล่าวได้ทั้งหมด ซึ่งผู้มีพระคุณท่านแรกที่คุณศึกษาใคร่ขอกราบพระคุณคือ รศ.ดร. ยุทธพงษ์ รั้งสรรค์เสรี อาจารย์ผู้สอนที่ได้ให้ความรู้ สนับสนุนงบประมาณ คำแนะนำ ด้วยความเอาใจใส่ทุกขั้นตอน และเทคนิคการนำเสนอรายงานปากเปล่า เพื่อให้การค้นคว้าฉบับนี้สมบูรณ์ที่สุด ผู้ศึกษาใคร่ขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้ รวมถึงเพื่อนๆ ครอบครัวทุกคนที่คอยให้คำแนะนำและกำลังใจตลอดการทำโครงการที่ผ่านมา

ผู้จัดทำขอขอบพระคุณทุกๆ ท่านไว้ ณ ที่นี้ด้วย ที่ได้ให้คำแนะนำในการจัดทำโครงการนี้ เสร็จลุล่วงไปได้ด้วยดี

นาย อัฐพงษ์

ภูฏกเงิน

ผู้จัดทำ

ระบบรักษาความปลอดภัยภายในบ้านโดยอาศัย IoT

IoT based Home Security System

โดย อัฐพงศ์ ภูฎกเงิน

56011448

อาจารย์ที่ปรึกษา รศ.ดร. ยุทธพงษ์ รังสรรค์เสรี

บทคัดย่อ

ระบบรักษาความปลอดภัยเป็นสิ่งสำคัญอย่างมากในการสร้างความอุ่นใจให้กับผู้พักอาศัย บางบริษัทที่มีนวัตกรรมที่ครอบคลุมการรักษาความปลอดภัยแบบครบวงจร แต่ก็มีราคาสูง จึงได้นำเทคโนโลยีสมองกลฝังตัวมาช่วยให้อุปกรณ์อิเล็กทรอนิกส์ทำงานร่วมกับเซ็นเซอร์ต่าง ๆ เพื่อสร้างให้เป็นระบบเตือนภัยใช้เองได้อย่างมีประสิทธิภาพและราคาประหยัดได้ โดยได้ใช้อุปกรณ์อิเล็กทรอนิกส์สมองกลฝังตัว ESP8266 มาสร้างเป็นระบบรักษาความปลอดภัยในที่พักอาศัยที่มีเซ็นเซอร์ตรวจจับแก๊ส เพื่อตรวจจับแก๊สหากมีการรั่วไหลและเซ็นเซอร์ตรวจจับการเคลื่อนไหวหากมีสิ่งมีชีวิตเข้ามาในรัศมีก็จะมีการถ่ายภาพเอาไว้ โดยใช้ IoT ในการในการทำให้ระบบเชื่อมต่อและแจ้งเตือนไปยังสังคมออนไลน์ต่าง ๆ ได้

ABSTRACT

Security systems are very important in creating peace of mind for residents. Some companies have an innovation that covers comprehensive security, but it has a high price. Therefore, bringing embedded technology to help electronic devices work with various sensors. In order to create an alarm system that can be used effectively and economically by using embedded electronic devices ESP8266 to

create a security system in the residence with a gas detection sensor. In order to detect gas if there is a leak and motion detection sensor, if there is life in the radius, it will be photographed. Using IoT in order to connect and alert to online society.



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
สารบัญ	IV
สารบัญรูป	VIII
บทที่ 1	
บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของโครงการ	1
บทที่ 2	
ทฤษฎีและหลักการที่เกี่ยวข้อง	2
2.1 Internet of Things	2
2.1.1 การแบ่งกลุ่ม Internet of Things	3
2.1.1.1 Industrial	3
2.1.1.2 Commercial IoT	3
2.2 NodeMCU ESP8266	3
2.2.1 การติดตั้ง Arduino IDE ลงบน NodeMUC ESP8266	5
2.3 โมดูลเซ็นเซอร์ (Sensor Module)	7
2.3.1 เซ็นเซอร์ตรวจจับควัน	7
2.3.2 เซ็นเซอร์ตรวจจับความเคลื่อนไหว	8
2.4 ไอเอฟทีทีทีแอปพลิเคชัน	9
2.4.1 ตัวอย่าง IFTTT ที่มีคนสร้างไว้	10

สารบัญ (ต่อ)

	หน้า
2.4.1.1 เปิด/ปิด Wi-fi อัตโนมัติเมื่อออกจากบ้าน	10
2.4.1.1 แจ้งเตือนสภาพอากาศเมื่อมีฝนจะตก	10
2.5 เน็ตพายและพรีบอร์ดไอโอ	11
2.5.1 การสร้าง Application	12
2.5.2 วิธีดูผลการทำงาน	15
2.6 โพรโตคอล MQTT	20
2.7 Yun Shield	21
2.7.1 Specification	22
2.7.2 Features	22
2.8 Micro SD Card Module	23
บทที่ 3 การออกแบบและการจัดทำปริญญาานิพนธ์	25
3.1 การออกแบบ	26
3.1.2 โมดูลเซ็นเซอร์แก๊ส MQ2	26
3.1.2.1 การออกแบบวงจร	29
3.1.3 โมดูลกล้องและโมดูล PIR	30
3.1.3.1 การออกแบบวงจร	31
3.2 เครื่องมือที่ใช้ในการทดลอง	32
3.3 การจัดเก็บผลการทดลอง	32
3.3.1 ผลการทดลองของโมดูลเซ็นเซอร์แก๊ส MQ-2	32
3.3.2 ผลการทดลองของโมดูลตรวจจับความเคลื่อนไหว	34
บทที่ 4 ผลการทดลอง	35
4.1 ผลการทดลองเซ็นเซอร์แก๊ส MQ-2	35

สารบัญ (ต่อ)

	หน้า
บทที่ 5	
4.2 ผลการทดลอง โมดูลตรวจจัดการเคลื่อนไหว	40
สรุปผลและข้อเสนอแนะ	42
5.1 สรุปผล	42
5.2 ข้อเสนอแนะ	42
บรรณานุกรม	43
ภาคผนวก	
โค้ดที่ใช้ในการเขียนโปรแกรม	44



สารบัญรูป

รูปที่	หน้า	
2.1	Internet of Things	2
2.2	NodeMCU ESP8266	4
2.3	ชิพ ESP8266	4
2.4	การติดตั้ง Arduino IDE ลง NodeMCU (1)	6
2.5	การติดตั้ง Arduino IDE ลง NodeMCU (2)	6
2.6	MQ-2	7
2.7	PIR Sensor	8
2.8	ตัวอย่างการเชื่อมโยงเครือข่ายทางสังคมผ่านบริการ IFTTT	9
2.9	ตัวอย่างการใช้งาน IFTTT (1)	10
2.10	ตัวอย่างการใช้งาน IFTTT (2)	10
2.11	รูปแบบการใช้บริการของ NETPIE	12
2.12	การสร้าง Application (1)	13
2.13	การสร้าง Application (2)	13
2.14	การสร้าง Application (3)	14
2.15	การสร้าง Application (4)	15
2.16	การสร้างการใช้งาน Freeboard (1)	16
2.17	การสร้างการใช้งาน Freeboard (2)	16
2.18	การสร้างการใช้งาน Freeboard (3)	17
2.19	การสร้างการใช้งาน Freeboard (4)	18
2.20	การสร้างการใช้งาน Freeboard (5)	19
2.21	การสร้างการใช้งาน Freeboard (6)	19
2.22	หลักการส่งข้อมูลของ MQTT โพรโทคอลผ่านตัวกลาง Broker	20

สารบัญรูป (ต่อ)

รูปที่	หน้า
2.23 Yun Shield	21
2.24 Micro SD Card Module	23
2.25 การต่อวงจรการใช้งาน	24
3.1 ภาพรวมของระบบที่ใช้ในการออกแบบ	25
3.2 Block diagram รวม	26
3.3 แผนผังลำดับการเกิดการเตือนภัยของเซ็นเซอร์ แก๊ส MQ-2	27
3.4 การทำงานของระบบขณะเชื่อมต่ออินเทอร์เน็ต	28
3.5 วงจรของส่วน MQ-2	29
3.6 MQ-2 ในกล่องอุปกรณ์	29
3.7 ขั้นตอนการทำงานของโมดูลกล้องโมดูล PIR	30
3.8 วงจรส่วน PIR และ webcam	31
3.9 PIR และ webcam ในกล่องอุปกรณ์	32
3.10 การแจ้งเตือนผ่านทางอีเมล	33
3.11 การแจ้งเตือนผ่านทางข้อความไลน์	34
4.1 monitor แสดงค่าที่อ่านได้จากเซ็นเซอร์	35
4.2 การแจ้งเตือนผ่านทางอีเมล	36
4.3 ข้อความปรากฏขึ้นในโทรศัพท์หลังจากติดตั้ง APP IFTTT	37
4.4 ข้อความที่ปรากฏใน Line notify	38
4.5 ไฟสีเขียวติด แสดงสถานะปกติ	39
4.6 ไฟสีแดงติด แสดงสถานะผิดปกติ	39
4.7 โค้ดการทำงานของ PIR	40
4.8 PIR sensor on serial monitor	41

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

บ้าน คือ ที่ให้ความอบอุ่นแก่ผู้อยู่อาศัย แต่ในปัจจุบันนอกจากการสร้างบ้านให้มีความแข็งแรงแล้ว ระบบรักษาความปลอดภัยที่ดีก็เป็นอีกปัจจัยหนึ่งที่สร้างความอุ่นใจให้ต่อผู้พักอาศัยได้ เพราะอาจเกิดปัญหาต่างเช่น ไฟฟ้าลัดวงจร อุปกรณ์เครื่องใช้ไฟฟ้าชำรุดและอาจส่งผลให้เกิดไฟไหม้ขึ้นได้ การที่มีระบบเตือนภัยหรือฝ้าระวังที่สามารถแจ้งเตือนแก่สรวหรือตรวจจับควันไฟได้ ย่อมมีประโยชน์เป็นอย่างมาก ถึงในปัจจุบันบางบริษัทได้มีการจัดทำอุปกรณ์หรือระบบรักษาความปลอดภัยขึ้นมาแต่ก็มีค่าใช้จ่ายที่สูง จึงได้นำอุปกรณ์สมองกลฝังตัวมีราคาถูกลงใช้งานง่ายและสามารถเชื่อมต่อสัญญาณแบบไร้สายได้ ซึ่งสามารถช่วยให้อุปกรณ์อิเล็กทรอนิกส์สมองกลฝังตัวทำงานร่วมกับเซ็นเซอร์ต่างๆ เพื่อสร้างให้เป็นระบบเตือนภัยใช้เองได้อย่างมีประสิทธิภาพและราคาประหยัดได้

1.2 วัตถุประสงค์

1. เพื่อศึกษาการทำงานของไมโครคอนโทรลเลอร์ร่วมกับเซ็นเซอร์ต่างๆ
2. เพื่อสร้างอุปกรณ์แจ้งเตือนต่างๆ ภายในบ้าน

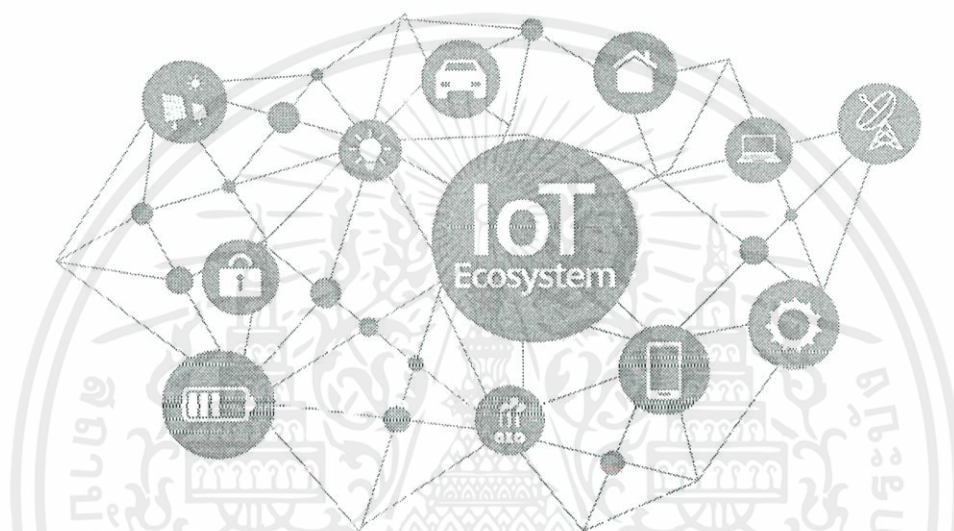
1.3 ขอบเขตของโครงการ

สร้างอุปกรณ์โดยเขียนโปรแกรมสั่งให้ไมโครคอนโทรลเลอร์ควบคุมเซ็นเซอร์ต่างๆ คือ เซ็นเซอร์ตรวจจับแก๊ส MQ-2 กรณีที่มีแก๊สรั่วไหลภายในบ้าน เซ็นเซอร์ตรวจจับการเคลื่อนไหว PIR เพื่อตรวจจับการเคลื่อนไหวของบุคคล สามารถตรวจสอบสถานะของแก๊สได้ผ่าน NETPIE และยังสามารถแจ้งเตือนกรณีฉุกเฉินได้ผ่านทางเสียงจาก Buzzer การแจ้งเตือนผ่านทางโซเชียลต่างๆ

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 Internet of Things



รูปที่ 2.1 Internet of Things

IoT หมายถึง เครื่องข่ายของวัตถุ อุปกรณ์ พาหนะ สิ่งปลูกสร้าง และสิ่งของอื่นๆ ที่มีวงจรอิเล็กทรอนิกส์ซอฟต์แวร์ เซ็นเซอร์ และการเชื่อมต่อกับเครือข่ายฝังตัวอยู่ และทำให้วัตถุเหล่านั้นสามารถเก็บบันทึกและแลกเปลี่ยนข้อมูลได้ IoT ทำให้วัตถุสามารถรับรู้สภาพแวดล้อมและถูกควบคุมได้จากระยะไกลผ่านโดยผ่านเครือข่ายอินเทอร์เน็ตที่มีอยู่แล้ว ทำให้สามารถผสานโลกกายภาพกับระบบคอมพิวเตอร์ได้อย่างแนบแน่นมากยิ่งขึ้น ผลที่ตามมาคือประสิทธิภาพ ความแม่นยำ การเก็บข้อมูลและประมวลผลแบบเวลาจริง (real time) ซึ่งก่อให้เกิดประโยชน์ทางเศรษฐกิจที่เพิ่มมากขึ้น เมื่อ IoT สามารถเชื่อมโยงทุกสิ่งทุกอย่างเข้าสู่โลกอินเทอร์เน็ต ทำให้มนุษย์สามารถควบคุมสิ่งการ อุปกรณ์เหล่านั้น ผ่านทางเครือข่ายอินเทอร์เน็ตและระบบคลาวด์ได้ เช่น

การเปิด-ปิด อุปกรณ์เครื่องใช้ไฟฟ้าภายในบ้านจากระยะไกล การมอนิเตอร์ระบบรักษาความปลอดภัย เมืองต้นแบบอัจฉริยะ (smart city) การควบคุมเครื่องจักรอุตสาหกรรมจากระยะไกล การบริหารจัดการระบบการทำเกษตรสมัยใหม่ การเพิ่มประสิทธิภาพของการสื่อสารคมนาคมและการขนส่ง ฯลฯ ด้วยที่กล่าวมาข้างต้นนี้ล้วนทำได้โดยเทคโนโลยี IoT

เทคโนโลยี IoT มีความจำเป็นต้องทำงานร่วมกับอุปกรณ์ประเภท RFID และ Sensors ซึ่งเปรียบเสมือนการเติมสมองให้กับอุปกรณ์ต่างๆ ที่ขาดไม่ได้คือการเชื่อมต่ออินเทอร์เน็ต เพื่อให้อุปกรณ์สามารถรับส่งข้อมูลถึงกันได้ เทคโนโลยี IoT มีประโยชน์ในหลายด้าน แต่ก็มาพร้อมกับความเสี่ยง เพราะหากระบบรักษาความปลอดภัยของอุปกรณ์ และเครือข่ายอินเทอร์เน็ตไม่ดีพอ ก็อาจทำให้มีผู้ไม่ประสงค์ดีเข้ามาขโมยข้อมูลหรือละเมิดความเป็นส่วนตัวของเราได้ ดังนั้นการพัฒนา IoT จึงจำเป็นต้องพัฒนามาตรการ และระบบรักษาความปลอดภัยไอทีควบคู่กันไปด้วย

2.1.1 การแบ่งกลุ่ม Internet of Things

2.1.1.1 Industrial

คือ แบ่งจาก local network ที่มีหลายเทคโนโลยีที่แตกต่างกัน ในโครงข่าย Sensor node โดยตัวอุปกรณ์ IoT Device ในกลุ่มนี้จะเชื่อมต่อแบบ IP network เพื่อเข้าสู่อินเทอร์เน็ต

2.1.1.2 Commercial IoT

คือ แบ่งจาก local communication ที่เป็น Bluetooth หรือ Ethernet (wired or wireless) โดยตัวอุปกรณ์ IoT Device ในกลุ่มนี้จะสื่อสารภายใน Sensor node เดียวกันเท่านั้นหรือเป็นแบบ local devices เพียงอย่างเดียวอาจไม่ได้เชื่อมสู่อินเทอร์เน็ต

2.2 NodeMCU ESP8266

ESP8266 NodeMCU คือ แพลตฟอร์มหนึ่งที่ใช้ช่วยในการสร้างโปรเจกต์ Internet of Things (IoT) ที่ประกอบไปด้วย Development Kit (ตัวบอร์ด) และ Firmware (Software บน

บอร์ด) ที่เป็นแบบ open source ซึ่งสามารถเขียนโปรแกรมด้วยภาษา Lau ได้ ทำให้ใช้งานได้ง่าย
 ขึ้น มาพร้อมกับโมดูล WiFi (FSP8266) แสดงดังรูปที่ 2.2 – 2.3 ซึ่งเป็นหัวใจสำคัญในการใช้
 เชื่อมต่อกับอินเทอร์เน็ตนั่นเอง



รูปที่ 2.2 NodeMCU ESP8266

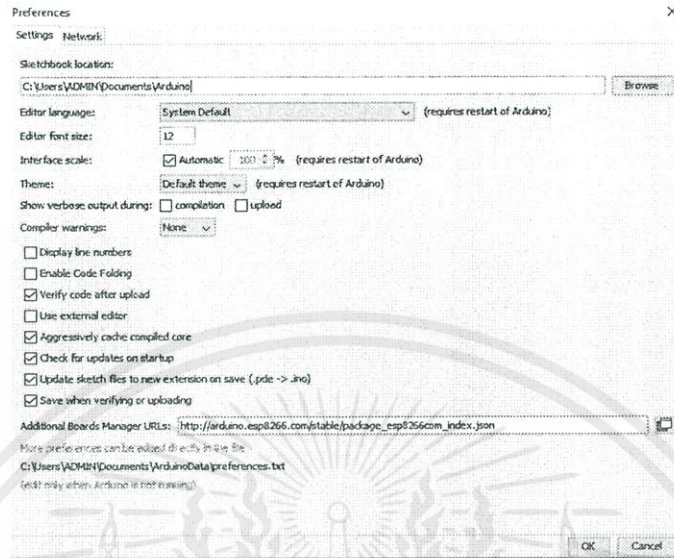
รูปที่ 2.3 ชิป ESP8266

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ESP8266 ถูกพัฒนามาจากบริษัท Espressif Systems โดยชาวสิงคโปร์ ชื่อ Mr. Teo Swee Ann ESP8266 นั้นถูกพัฒนาตั้งแต่ ESP-01 ถึงจน ESP-14 ในยุคปัจจุบัน ซึ่งคุณสมบัติโดยทั่วไปสำหรับ ชิพ ESP8266 นอกจากจะเป็นชิพไวไฟโมดูลที่มีราคาถูกลงแล้ว ยังเป็นไมโครคอนโทรลเลอร์ขนาด 32 บิต สามารถทำงานร่วมกับชุดคำสั่งมาตรฐาน (AT Command) ที่สามารถใช้ติดต่อสื่อสารกับอุปกรณ์สื่อสารแบบอนุกรม (Serial) ทำหน้าที่ได้ทั้งเป็นผู้ให้บริการ (Server) และ ผู้ร้องขอใช้บริการ (Client) หรือที่เรียกว่า “ Soft AP” ตามมาตรฐาน IEEE 802.11 b/g/n มีพอร์ตอินพุต-พอร์ตเอาต์พุตให้ใช้งาน พร้อมทั้งฟังก์ชันรูปแบบการสื่อสาร แบบ I2C, SPI, UART, ADC ให้ใช้งาน พร้อมมีเครื่องมือและพัฒนาชุดภาษา เช่น ภาษาซีพลัสพลัส (C++) , ไพธอน (python), จาวา (Java), ลูอาสคริปต์ (LUA Script) ให้ใช้งานมากมาย

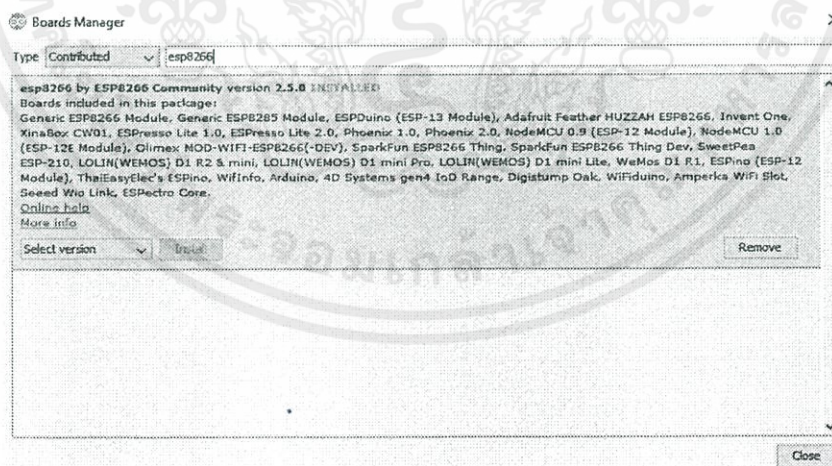
2.2.1 การติดตั้ง Arduino IDE ลงบน NodeMUC ESP8266

ติดตั้ง Arduino IDE เวอร์ชัน 1.6.4 หรือใหม่กว่า เมื่อทำการติดตั้งเสร็จเรียบร้อยแล้ว ให้เปิดโปรแกรมขึ้นมา จากนั้นไปที่ menu File >> Preferences ใส่ URL >> ลงในตรงช่อง Addition Board Manager URLs: ดัง แสดง ดังนี้ http://arduino.esp8266.com/stable/package_esp8266com_index.json จากนั้นกด OK ดังรูปที่ 2.4



รูปที่ 2.4 การติดตั้ง Arduino IDE ลง NodeMCU (1)

จากนั้นไปที่ Menu Tools >> Board:"xxxxx" >> Board Manager เลือก Type เป็น Contributed ไปที่ ESP8266 และกด Install ดังรูปที่ 2.5



รูปที่ 2.5 การติดตั้ง Arduino IDE ลง NodeMCU (2)

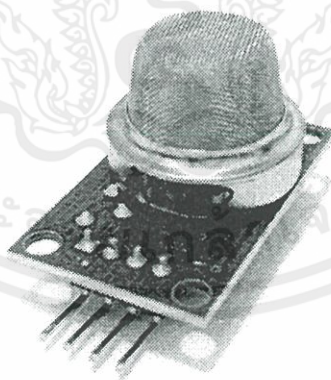
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อติดตั้ง ESP8266 เสร็จเรียบร้อยแล้ว ให้ปิดโปรแกรม Arduino IDE ก่อนแล้วจึงเปิดขึ้นมาใหม่ เมื่อเปิดโปรแกรมแล้วให้ไปที่ Menu Tools >> Board: "xxxxxx" จะพบว่า มี Menu สำหรับเลือกใช้งาน ESP8266 กับ Arduino IDE ขึ้นมาให้เลือกใช้งานแล้ว

2.3 โมดูลเซ็นเซอร์ (Sensor Module)

2.3.1 เซ็นเซอร์ตรวจจับควัน MQ-2

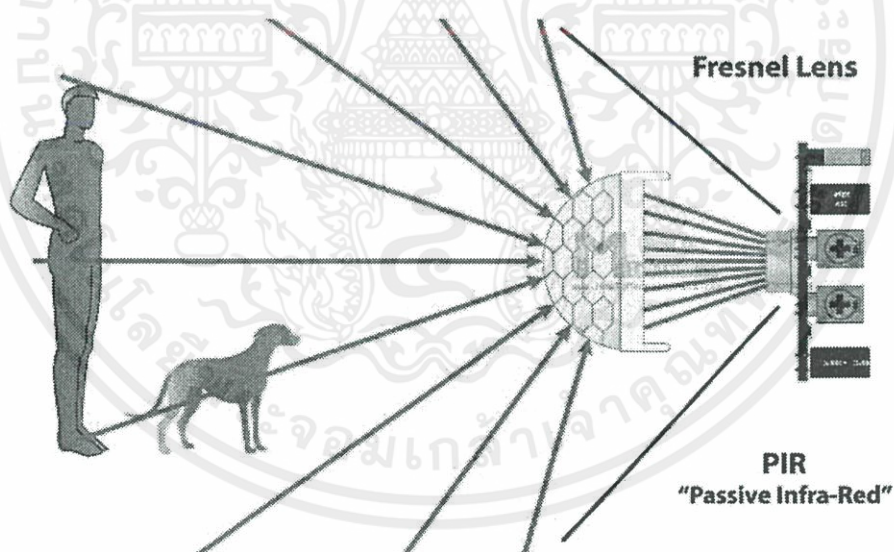
MQ-2 เป็นเซ็นเซอร์ตรวจจับความเข้มข้นของแก๊สที่ติดไฟในอากาศ ซึ่งได้แก่ แก๊สประเภท แอลพีจี (LPG) โพรเพน (Propane) ไฮโดรเจน (Hydrogen) มีเทน (Methane) และบิวเทน (Butane) เป็นต้น รวมถึงควันไฟ (Smoke) ซึ่งให้แรงดันเอาต์พุตออกมาเป็นตัวเลขแบบอนาล็อก เซ็นเซอร์ชนิดนี้สามารถวัดความเข้มข้นของแก๊สได้ในระหว่างช่วง 200 ถึง 10,000 ppm และทำงานได้ในอุณหภูมิ -20 องศาเซลเซียส ถึง 50 องศาเซลเซียส โดยกินกระแสไฟฟ้าประมาณ 150 mA ที่แหล่งจ่ายแรงดัน 3.3 โวลต์ -5 โวลต์ แสดงดังรูปที่ 2.3



รูปที่ 2.6 MQ-2

2.3.2 เซ็นเซอร์ตรวจจับความเคลื่อนไหว (Motioned sensor)

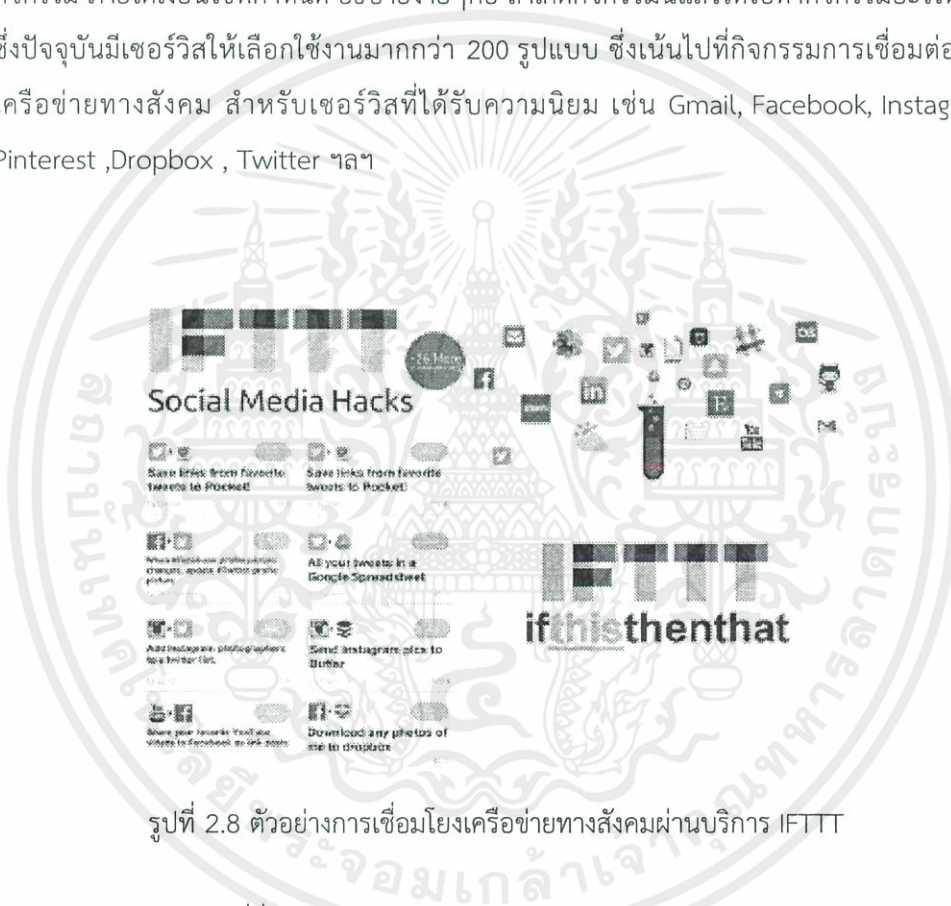
เมื่อกล่าวถึงเซ็นเซอร์ตรวจจับความเคลื่อนไหว มักนึกถึงพาสซีฟอินฟราเรดเซ็นเซอร์ (Passive infrared sensor: PIR) เป็นส่วนมาก แต่ในความเป็นจริงแล้วเซ็นเซอร์ตรวจจับความเคลื่อนไหว ยังมีอีก 2 ชนิดใหญ่ๆ คือ ชนิดอัลตราโซนิกเซ็นเซอร์ (Ultrasonic sensor) เป็นเซ็นเซอร์ที่มีการปล่อยคลื่นอัลตราโซนิกออกไปและตรวจจับโดยการวัดการสะท้อนกลับของคลื่นกลับมา ส่วนอีกชนิดหนึ่ง คือไมโครเวฟเซ็นเซอร์ (Microwave sensor) ซึ่งเป็นเซ็นเซอร์ที่ปล่อยคลื่นไมโครเวฟออกมาและตรวจจับวัตถุจากการสะท้อนของคลื่นเช่นกัน ในโครงงานนี้จะกล่าวถึงเซ็นเซอร์ตรวจจับความเคลื่อนไหวชนิดพาสซีฟอินฟราเรดเซ็นเซอร์ (พีไออาร์เซ็นเซอร์) เท่านั้น พีไออาร์ (PIR) เป็นอุปกรณ์เซ็นเซอร์ที่ตรวจจับความร้อนจากสิ่งมีชีวิต ซึ่งต้องอยู่ในบริเวณพื้นที่รัศมีองศาที่จำกัดในการตรวจจับ โดยทั่วไปรัศมีการตรวจจับได้ไกลถึงประมาณ 6-7 เมตร ซึ่งหลักการทำงานของพีไออาร์คือ เซ็นเซอร์ปล่อยรังสีอินฟราเรดออกมารอการเคลื่อนที่ของวัตถุหรือสิ่งมีชีวิตเข้ามาในรัศมี ทำให้เกิดการตรวจจับระดับของรังสีอินฟราเรดที่เปลี่ยนแปลงไปดังรูปที่ 3



รูปที่ 2.7 PIR sensor

2.4 ไอเอฟทีทีทีแอปพลิเคชัน

ไอเอฟทีทีที (IFTTT: IF This Then That) เป็นการให้บริการฟรีบนพื้นฐานของเว็บ-เบส (การทำงานผ่านโปรแกรม web browser) โดยอนุญาตให้ผู้ใช้งานทำการกำหนดสร้างเงื่อนไขกิจกรรมของตัวเองขึ้นมา หรือเรียกว่า “รีซิป” (recipe) โดย recipe นั้นเป็นการลิงค์กิจกรรม 2 กิจกรรม ภายใต้เงื่อนไขที่กำหนด อธิบายง่ายๆคือ ถ้าเกิดกิจกรรมนี้แล้วให้ไปทำกิจกรรมอะไรต่อไป ซึ่งปัจจุบันมีเซิร์ฟเวอร์ให้เลือกใช้งานมากกว่า 200 รูปแบบ ซึ่งเน้นไปที่กิจกรรมการเชื่อมต่อทางเครือข่ายทางสังคม สำหรับเซิร์ฟเวอร์ที่ได้รับความนิยม เช่น Gmail, Facebook, Instagram, Pinterest ,Dropbox , Twitter ฯลฯ

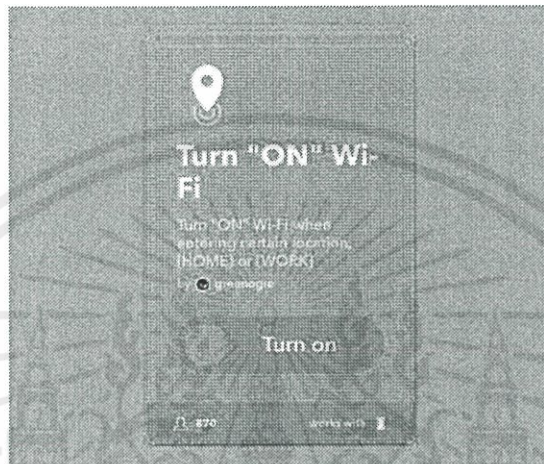


รูปที่ 2.8 ตัวอย่างการเชื่อมโยงเครือข่ายทางสังคมผ่านบริการ IFTTT

การเขียน ที่ชื่อว่า IF This Then That เพราะลักษณะการทำงานจะเป็น Flow เมื่อเกิดสิ่งใดสิ่งหนึ่งเกิดขึ้นตามเงื่อนไขที่เราสร้างไว้ แล้วจะเกิดสิ่งต่อไปตามมา ยกตัวอย่างง่ายๆ เช่น ถ้าผมเล่นทั้ง Facebook และ Twitter เมื่อผมเปลี่ยนรูป Profile บน Facebook แล้วอยากให้ Twitter เปลี่ยนเป็นรูปเดียวกันด้วย จะได้ไม่ต้องมานั่งกดเปลี่ยนทีละอัน หรือเมื่อผมโพสต์กดให้มี Notification แจ้งผ่านทางอีเมล เป็นต้น

2.4.1 ตัวอย่าง IFTTT ที่มีคนสร้างไว้

2.4.1.1 เปิด/ปิด wifi อัตโนมัติเมื่อออกจากบ้าน



รูปที่ 2.9 ตัวอย่างการใช้งาน IFTTT (1)

2.4.1.2 แจ้งเตือนพยากรณ์อากาศว่ามีฝนจะตก

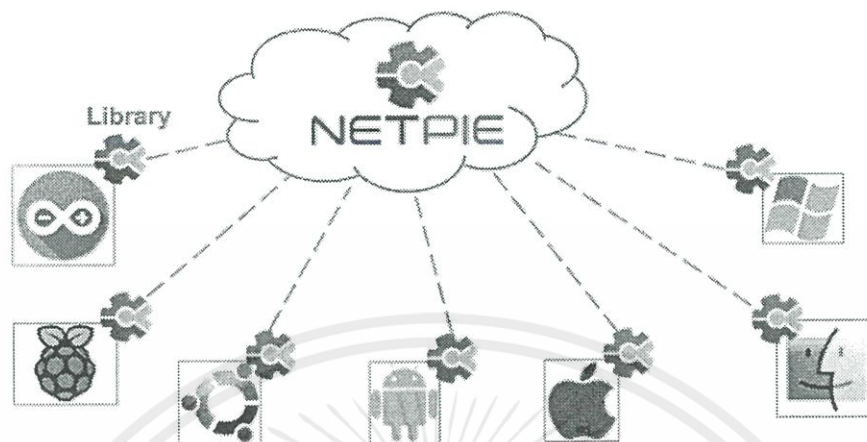


รูปที่ 2.10 ตัวอย่างการใช้งาน IFTTT (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 เน็ตพายและฟิร์บอร์ดีไอโอ

NETPIE ถือกำเนิดขึ้นจากศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (NECTEC) เพื่อวัตถุประสงค์ “ให้เกิดการพัฒนาอุตสาหกรรมไอทีประยุกต์ขึ้นในประเทศ ตอบสนองให้นักพัฒนา หรือผู้ประกอบการเอสเอ็มอีไทยได้เข้ามาร่วมใช้แพลตฟอร์ม IoT ในการพัฒนานวัตกรรมเพื่อนำไปสู่การเกิดธุรกิจ บริการและผลิตภัณฑ์ใหม่ ๆ ซึ่งในระยะยาว นักพัฒนาและผู้ประกอบการกลุ่มนี้ จะเป็นแรงขับเคลื่อนให้เกิดการนา เทคโนโลยี IoT ไปประยุกต์ใช้และสร้างมูลค่าเพิ่มในภาคอุตสาหกรรมมากขึ้น” NETPIE คือการให้บริการคลาวด์เซิร์ฟเวอร์ (cloud server) ในรูปแบบเป็น Platform as a Service เป็นบริการการเชื่อมต่อข้อมูลและแลกเปลี่ยนข้อมูลระหว่างอุปกรณ์ต่างๆ ได้เหมาะสำหรับ Internet of Things ซึ่งรูปแบบการใช้งานจะเหมือนกับ MQTT พ่วงความสามารถด้าน Authentication และ Token เพิ่มความปลอดภัยในการสื่อสาร ซึ่งสามารถเชื่อมต่อกับ NETPIE ได้กับอุปกรณ์สมองกลฝังตัวประเภท บอร์ด Arduino บอร์ด Raspberry Pi บอร์ด ESP8266 ฯลฯ นอกจากนี้ ยังสามารถพัฒนาร่วมกับการใช้ภาษา HTML5 หรือแม้กระทั่ง Node.js เป็นต้น NETPIE เน้นการใช้งานที่ง่าย สะดวก ยืดหยุ่น และคล่องตัวสูงในการขยายตัว ซึ่ง NETPIE ช่วยให้อุปกรณ์สามารถคุยกันได้โดยผู้พัฒนาไม่ต้องกังวลว่า อุปกรณ์นั้นจะอยู่ที่ใด เพียงนำ NETPIE library ไปติดตั้งในอุปกรณ์ NETPIE จะรับหน้าที่ดูแลการเชื่อมต่อให้ทั้งหมด ไม่ว่าจะอุปกรณ์นั้นจะอยู่ในเครือข่ายชนิดใด ลักษณะใด หรือแม้กระทั่งเคลื่อนย้ายไปอยู่ที่ใด ทำให้จัดการปัญหาการออกแบบการเข้าถึงอุปกรณ์จากระยะไกลไปได้ ซึ่งทางเนคเทคจะเปิด NETPIE library ในรูปแบบโอเพ่นซอร์ส (open source) เพื่อให้ นักพัฒนาสามารถนำไปปรับปรุงต่อให้ตรงกับความต้องการใช้งานและโดยเปิดโอกาสให้นำไปใช้ในเชิงพาณิชย์ได้

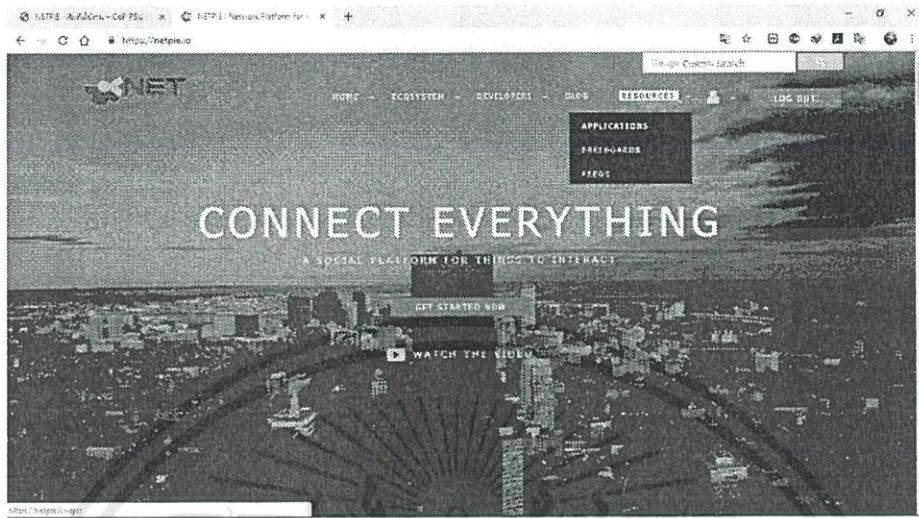


รูปที่ 2.11 รูปแบบการใช้บริการของ Netpie

นอกจากนี้ NETPIE ได้นำเอาฟรีบอร์ดไอโอ (FreeboardIO) ไว้ด้วยทำให้สะดวกต่อการสร้างแอปพลิเคชันแสดงแผงหน้าปัดแสดงต่างๆ เป็นอย่างมากสำหรับการใช้งานใดๆ ทั้งนี้การใช้งาน NETPIE จะจำเป็นต้องติดตั้งไลบรารี “microgear” เสียก่อน สำหรับการสื่อสารระหว่างอุปกรณ์ต่างๆ NETPIE จะใช้โปรโตคอล MQTT (Message Queuing Telemetry Transport) เช่นเดียวกับผู้ใช้บริการรายอื่นๆ เช่น Thingspeak, SparkFun, IoT-playground เป็นต้น

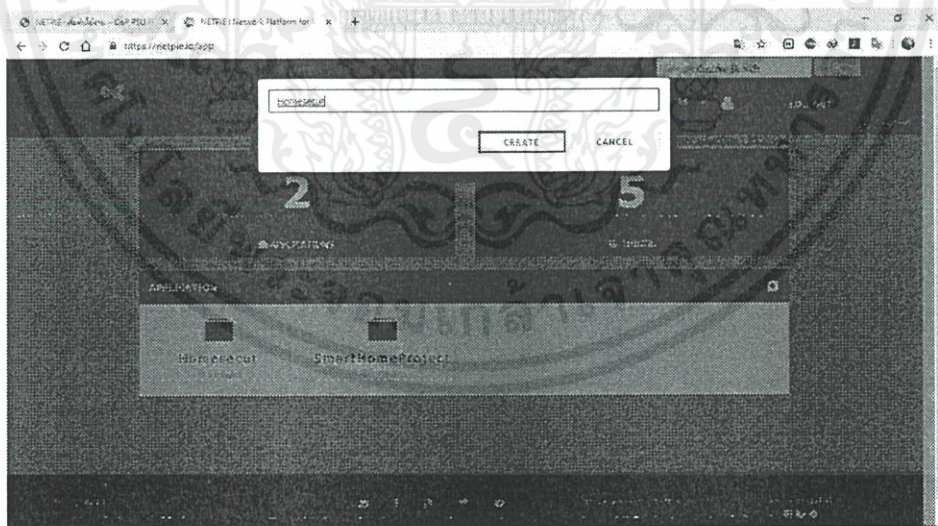
2.5.1 การสร้าง Application

ไปที่เมนู Resources > Applications ดังแสดงในรูปที่ 2.12



รูปที่ 2.12 การสร้าง Application (1)

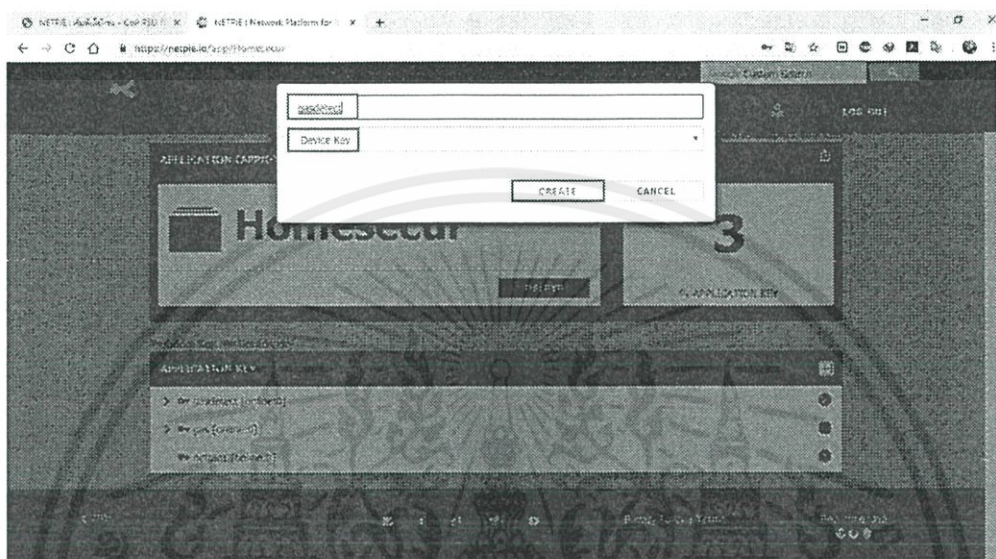
เมื่อกดแล้วจะจะมีหน้าต่างให้สร้าง Application ให้กดที่เครื่องหมายบวก (+) แล้วตั้งชื่อตามที่ต้องการ โดยในโครงการนี้ใช้ชื่อว่า “Homesecur” จากนั้นคลิก Create ดังแสดงในรูปที่ 2.13



รูปที่ 2.13 การสร้าง Application (2)

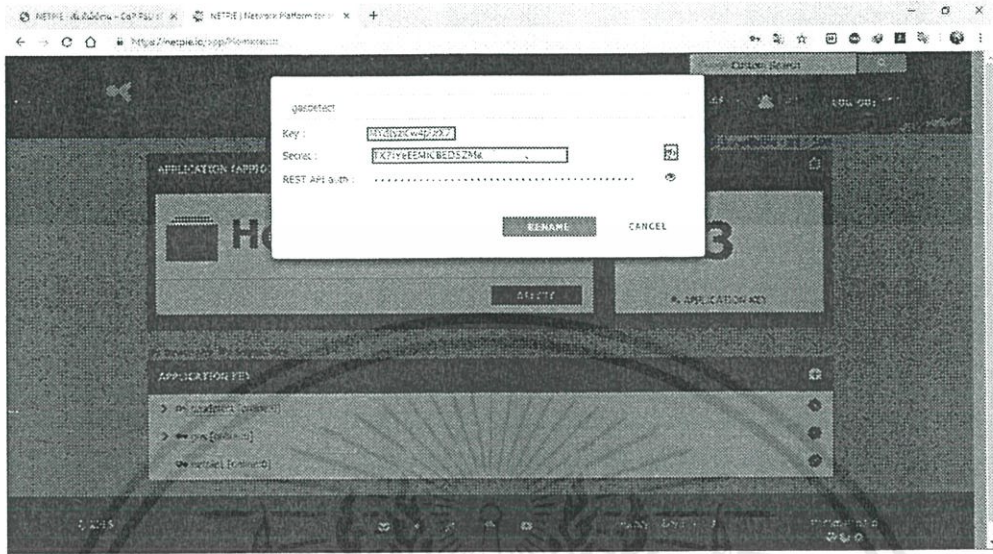
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใน Application ที่สร้างเสร็จให้คลิกที่เครื่องหมายบวก (+) ใส่ชื่ออุปกรณ์ลงในที่นี้ใช้ชื่อว่า “gasdetect” เลือกชนิดเป็น Device Key แล้วคลิก Create ดังแสดงในรูปที่ 2.14



รูปที่ 2.14 การสร้าง Application (3)

ต่อไปเป็นการเอาค่า Key และ Secret ไปใช้งาน โดย Key ในที่นี้คือ MYdlyziCw4pUrX7 และ Secret คือ TX7lYeEEMICBED5ZMaUxxxxxx โดยสามารถดูได้จากการคลิกที่ Application KEY ที่สร้างขึ้น แล้วคลิกรูปลูกตาหลังบรรทัด Secret แล้วให้ copy ค่าที่ได้ไว้ เพราะต้องนำไปใช้งานต่อไป ดังแสดงในรูปที่ 2.15

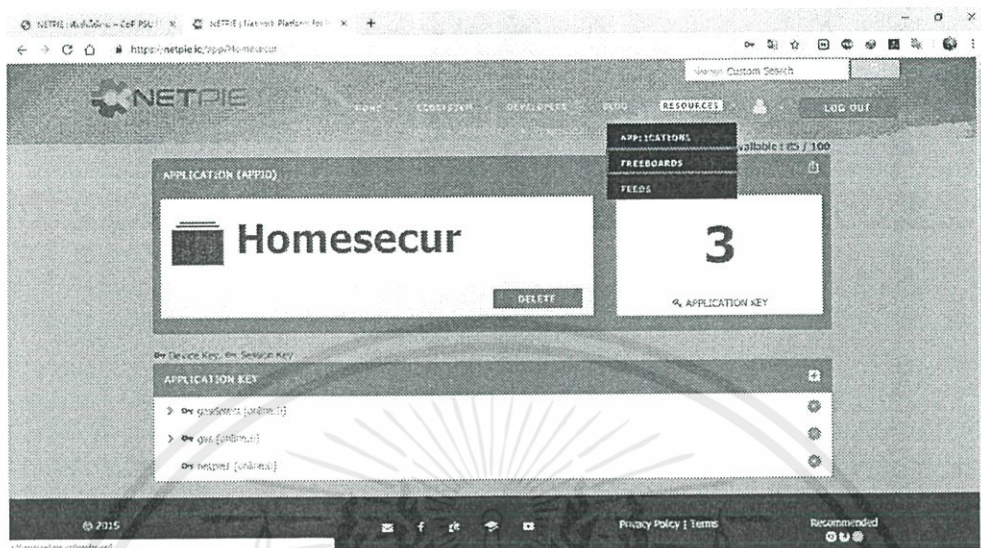


รูปที่ 2.15 การสร้าง Application (4)

2.5.2 วิธีดูผลการทำงาน

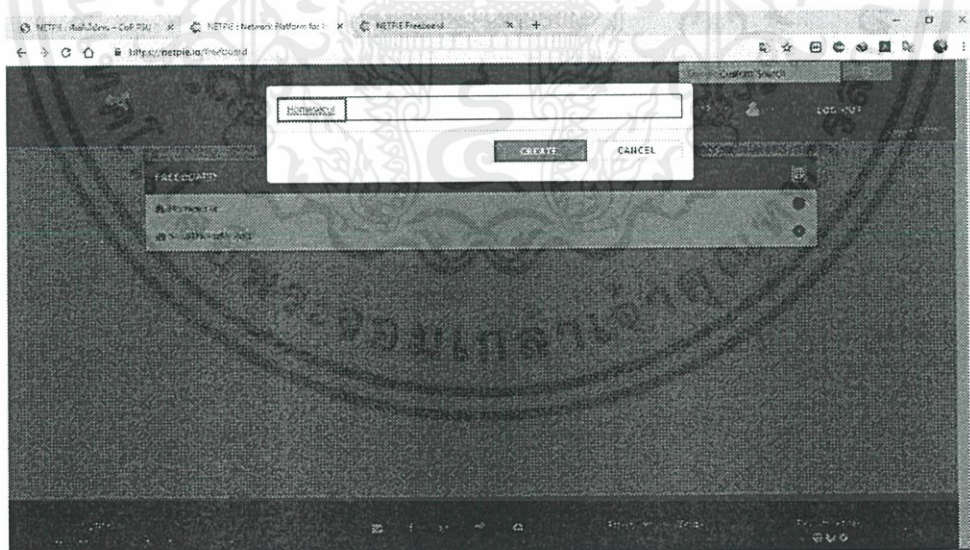
คลิกที่ Resources > Freeboards ดังแสดงในรูปที่ 2.16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.16 การสร้างการใช้งาน Freeboard (1)

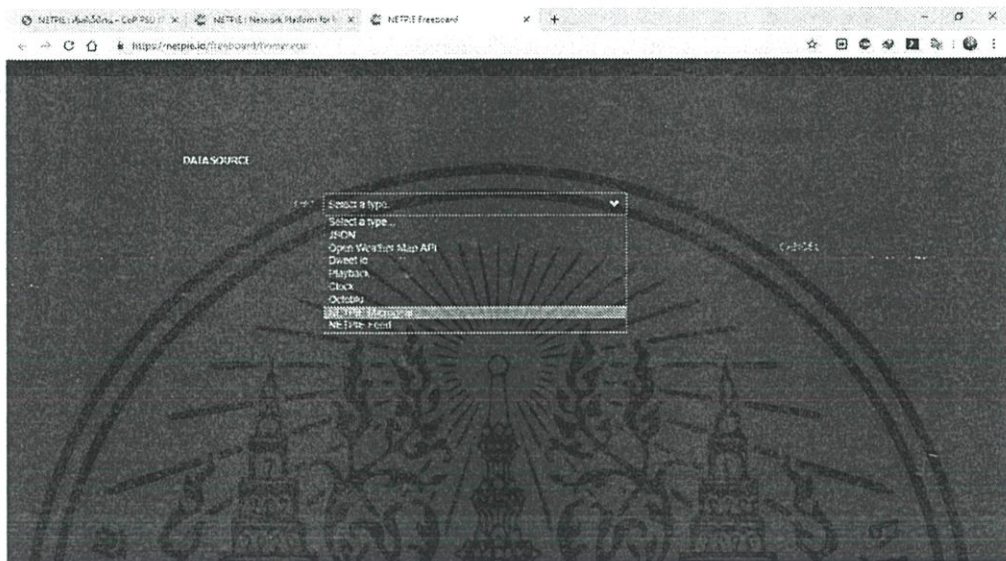
จากนั้นให้คลิกที่เครื่องหมายบวก (+) ตั้งชื่อตามที่ต้องการ โดยในที่นี้ตั้งชื่อว่า “Homesecur” แล้วคลิก Create ดังแสดงในรูปที่ 2.17



รูปที่ 2.17 การสร้างการใช้งาน Freeboard (2)

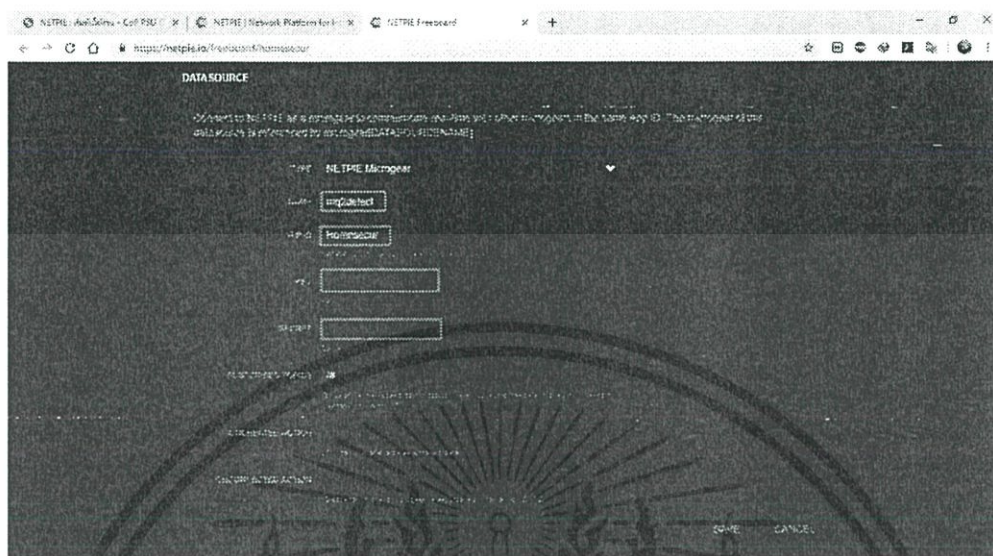
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในหน้า Freeboard ที่สร้างขึ้นมานั้นให้คลิกที่ Add ในส่วนของ Data Sources เลือกชนิดที่จะสร้างเป็น NETPIE Microgear ดังแสดงในรูปที่ 2.18



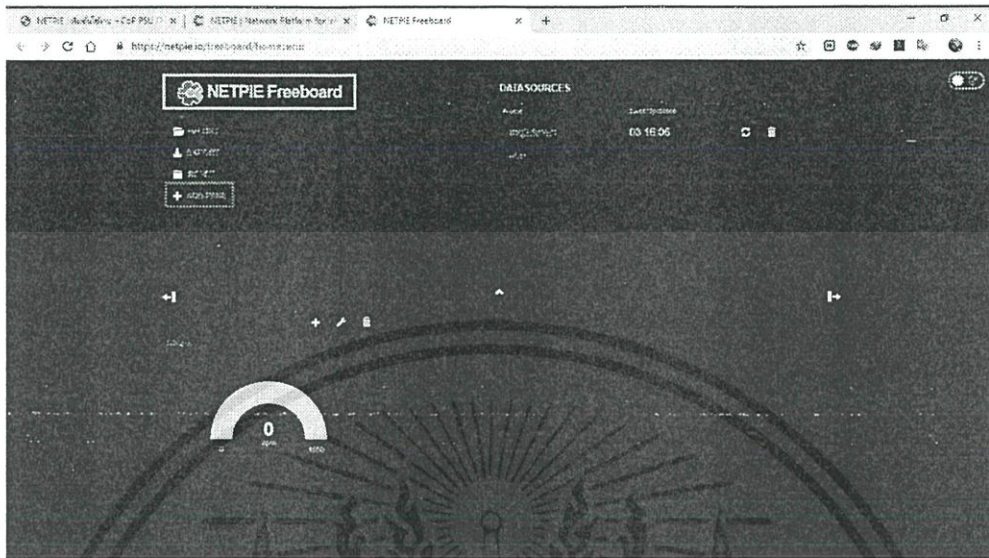
รูปที่ 2.18 การสร้างการใช้งาน Freeboard (3)

จากนั้นกรอกข้อมูลต่างลงไป Name ที่ต้องการ App ID ในที่นี้คือชื่อที่เราตั้งไว้ในตอนแรกคือ “Homese Secur” ส่วน Key และ Secret ก็เป็นค่าที่เราได้มาจากขั้นตอนในก่อนหน้านั้น เมื่อกรอกตามช่องที่วงสีแดงไว้ครบถ้วนแล้วก็คลิก Save ได้เลย ดังแสดงในรูปที่ 2.19

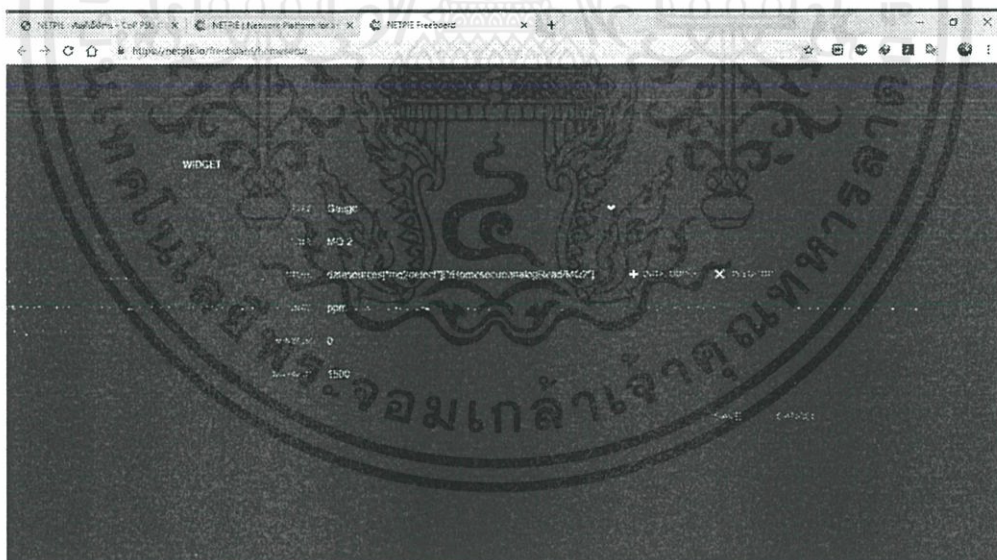


รูปที่ 2.19 การสร้างการใช้งาน Freeboard (4)

ต่อไปคลิกที่ Add Pane จากนั้นจะมีหน้าต่างวางๆขึ้นมาให้คลิกที่เครื่องหมายบวก (+) เลือกชนิดที่ต้องการใช้งาน โยโนโครงานนี้เลือกใช้ชนิดเป็น Gauge ถัดมา TITLE ใส่เป็น MQ-2 เป็นชื่อแสดงให้ทราบว่าหน้าต่างนี้แสดงค่าของอะไร ถัดมา VALUE เป็นส่วนที่จะให้หน้าต่างการทำงานนี้ทำงานร่วมกับอุปกรณ์ของเราได้ โดยค่าส่วนตรงนี้เป็นค่าที่เราได้เขียนไว้ในส่วนของการเขียนโปรแกรมใน Arduino IDE ถัดมา UNITS เป็นหน่วยที่ใช้ ซึ่งจะใช้เป็น ppm ถัดมาก็เป็นช่วงค่าต่ำสุดถึงสูงสุดที่เราจะกำหนด ดังแสดงในรูปที่ 2.20 – 2.21



รูปที่ 2.20 การสร้างการใช้งาน Freeboard (5)

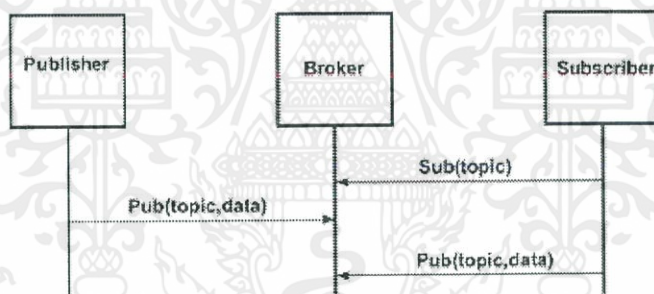


รูปที่ 2.21 การสร้างการใช้งาน Freeboard (6)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 โพรโทคอล MQTT

MQTT เป็นโพรโทคอลระหว่างอุปกรณ์ 2 อุปกรณ์ หรือบางครั้งเรียกว่า เครื่องจักรคุยกับเครื่องจักร (machine-to-machine (M2M) สนับสนุนเทคโนโลยีอินเทอร์เน็ตออฟริงส์ โพรโทคอลนี้ใช้วิธีการเหมือนกับ Message Queue ปกติ แต่พัฒนาให้รองรับงานด้านอินเทอร์เน็ตออฟริงส์ นอกจากจะรอรับและอ่านค่าแล้ว ยังสามารถส่งงานอุปกรณ์ที่ใช้ MQTT ได้ด้วย เนื่องจากโพรโทคอลตัวนี้มีน้ำหนักเบา ออกแบบมาเพื่อใช้งานกับอุปกรณ์อิเล็กทรอนิกส์ขนาดเล็กการรับส่งข้อมูลในเครือข่ายที่มีขนาดเล็ก แบนด์วิดท์ต่ำ ใช้หลักการแบบ publisher / subscriber คล้ายกับหลักการที่ใช้ในเว็บเซอร์วิส (Web Service) ที่ต้องใช้เว็บเซิร์ฟเวอร์ (Web Server) เป็นตัวกลางระหว่างคอมพิวเตอร์ของผู้ใช้ แต่ MQTT จะใช้ตัวกลางที่เรียกว่า Broker เพื่อทำหน้าที่ จัดการคิวรับ - ส่งข้อมูลระหว่างอุปกรณ์และทั้งในส่วนที่เป็น Publisher และ Subscriber แสดงดังรูปที่ 2.7



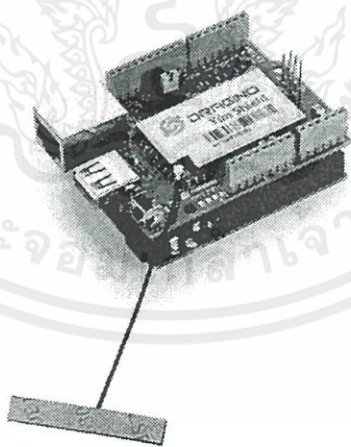
รูปที่ 2.22 หลักการส่งข้อมูลของ MQTT โพรโทคอลผ่านตัวกลาง Broker

จากรูปที่ 2.7 จะเห็นได้ว่า Topic จะเป็นตัวอ้างอิงหลักข้อมูลที่จะ Publisher ออกไปยัง Broker ซึ่งจะต้องมี topic กำกับไว้เสมอ ส่วนทางฝ่าย subscriber ก็จะต้องอ้างอิงถึง topic เพื่อเรียกข้อมูลที่ต้องการ แต่การกระทำใดๆ ต้องผ่าน Broker ทุกตัว ตัวอย่างที่เห็นได้อย่างชัดเจน คือ Facebook Messenger ก็ใช้โพรโทคอลนี้เช่นกัน ดังนั้นจะเห็นได้ว่าการใช้งานโพรโทคอลนี้จะต้องมีอุปกรณ์ที่ฝัง MQTT Broker มาเพื่อทำหน้าที่รับส่งข้อมูลไปยังอุปกรณ์อื่นที่ใช้ MQTT เหมือนกัน MQTT โพรโทคอลจะประกอบไปด้วย Broker, Publisher และ Subscriber ซึ่งแต่ละตัวมีหน้าที่ดังนี้ Broker จะทำหน้าที่เป็นตัวกลางคอยจัดข้อความการส่ง message โดยอ้างอิงด้วย

หัวข้อ topic ซึ่ง โปรแกรมที่ใช้ทำ MQTT Broker มีหลายค่ายให้ใช้งาน หนึ่งในนั้นคือโปรแกรม Mosquitto เป็น OpenSource MQTT Broker สนับสนุน MQTT Broker v3.1/3.1.1 Subscriber จะทำหน้าที่คอยดูการเปลี่ยนแปลงของข้อความการส่ง message ที่อ้างอิงด้วยหัวข้อ topic ยกตัวอย่างเช่น ถ้าสมมติว่าหัวข้อ topic ที่สนใจมีการเปลี่ยนแปลงก็จะดึงข้อมูล (data) มาใช้งาน Publisher จะทำหน้าที่คอยส่งข้อมูลไปยังหัวข้อ topic นั้นๆ Subscriber จะทำหน้าที่คอยดูการเปลี่ยนแปลงของการส่ง message ที่อ้างอิงด้วยหัวข้อ Topic เช่นถ้ามีหัวข้อที่น่าสนใจและมีการเปลี่ยนแปลงก็จะทำการดึงข้อมูลนั้น ๆ มาใช้งาน

2.7 Yun Shield

Yun Shield เป็นระบบ Open Source OpenWrt (ระบบเดียวกับที่รันใน Arduino Yun) และสามารถทำงานเข้ากันได้กับ Arduino IDE v1.5.4 หรือใหม่กว่าอย่างเต็มที่ดังรูปที่ 2.13 Yun Shield เป็นทางเลือกที่ยอดเยี่ยมสำหรับโครงการ Arduino ที่ต้องการการเชื่อมต่ออินเทอร์เน็ตที่หลากหลาย โดยพื้นฐานแล้ว Yun Shield + Leonardo เท่ากันกับ Arduino Yun อย่างเป็นทางการ แต่ Yun Shield นั้นมีความยืดหยุ่นมากกว่าเพราะสามารถทำงานร่วมกับบอร์ด Arduino อื่น ๆ เช่น Uno, Duemilanove, Mega ฯลฯ และ Yun Shield ใช้เสาอากาศ wifi ภายนอกซึ่งให้เสถียรภาพและความเป็นไปได้ที่หลากหลายในหลายสภาพแวดล้อม



รูปที่ 2.23 Yun Shield

2.7.1 Specification

- Processor: 400MHz, 24K MIPS
- Flash: 16Mbytes
- RAM: 64Mbytes
- Power Input: 4.75v ~ 23v via Arduino VIN pin
- 1 x 10M/100M RJ45 connector
- 150M WiFi 802.11 b/g/n
- External Antenna via I-Pex connector
- 1 x USB 2.0 host connector, used for USB storage or 3G connection
- 1 x Reset button
- Compatible with 3.3v or 5v I/O Arduino.

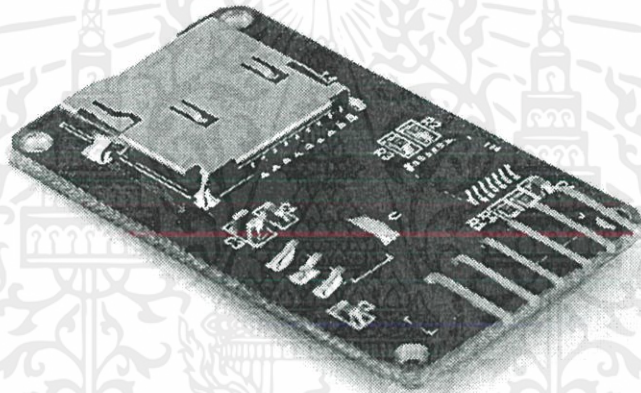
2.7.2 Features

- Open source Linux (OpenWrt) onboard
- Low power consumption
- Compatible with Arduino IDE 1.5.4 or later, user can program, debug or upload sketches to the Arduino board via the Arduino IDE.
- Managed by Web GUI, SSH via LAN or WiFi
- Software upgradable via network
- Built-in web server
- Supports internet connection via LAN port, WiFi or 3G dongle.
- Supports USB flash to provide storage for Arduino projects.
- Failsafe design provides robust system.
- Compatible with Arduino Leonardo, Uno , Duemilanove, Diecimila, Mega

2.8 Micro SD Card module

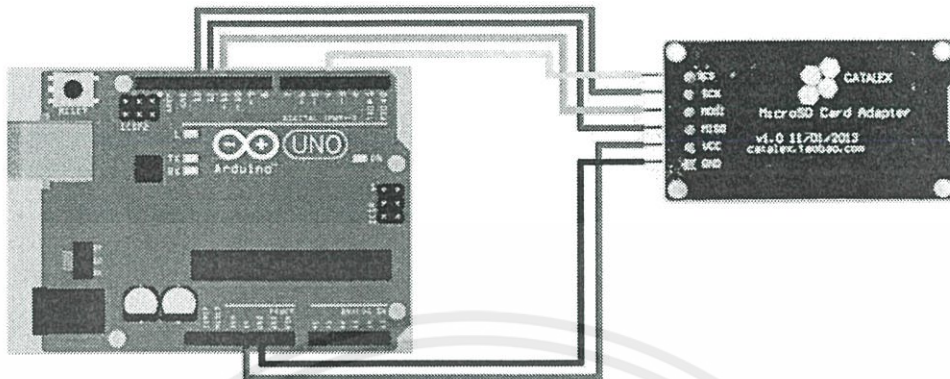
โมดูล Micro SD Card เป็นโมดูลสำหรับอ่านการ์ด Micro SD และอินเทอร์เฟซ SPI ผ่าน ไตรฟ์เวอร์ระบบไฟล์ระบบไมโครคอนโทรลเลอร์เพื่อให้สามารถอ่านและเขียนไฟล์ Micro SD card ได้ ทำหน้าที่ บันทึกข้อมูลลง SD Card เก็บ Data logger โดยใช้ร่วมกับ Arduino มี library มาตรฐาน พร้อมใช้งาน สามารถใช้งานได้ง่าย

รูปที่ 2.24 แสดงลักษณะอุปกรณ์ของ Micro SD card Module ที่ใช้ โดยที่รองรับ Micro SD Card ($\leq 2G$), การ์ด Micro SDHC ($\leq 32G$) ซึ่งเป็นการเพิ่มความเร็วสูง บอร์ดวงจรการแปลงระดับที่สามารถเชื่อมต่อกับระดับที่รองรับได้คือ 5V หรือ 3.3V แหล่งจ่ายไฟคือ 4.5V ~ 5.5V, 3.3V วงจรควบคุมแรงดันไฟฟ้าบอร์ด อินเทอร์เฟซการสื่อสารเป็นอินเทอร์เฟซ SPI มาตรฐาน 4 M2 มีรูเจาะตำแหน่งสกรูสำหรับติดตั้งและมีขนาดเล็ก สามารถใช้งานได้ง่าย โดยมีขนาดเพียงแค่ว่า 4.1×2.4 ซม



รูปที่ 2.24 Micro SD Card Module

ตัวอย่างการใช้งานการเชื่อมต่อระหว่าง Arduino กับ Micro SD Card ดังแสดงในรูปที่ 2.25 โดยที่ขาต่างๆ ของ Micro SD Card มีการเชื่อมต่อดังนี้ CS -> 4, SCK -> 13, MOSI -> 11, MISO -> 12, VCC -> 5V และ GND -> GND



2.25 การต่อวงจรการใช้งาน

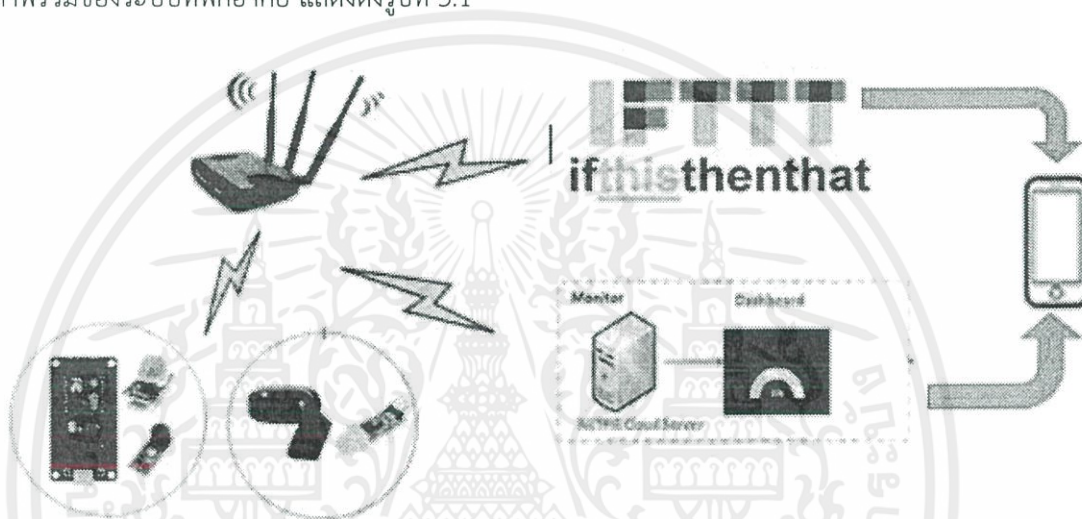


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและจัดทำปฏิญญาฉบับ

ในการออกแบบระบบรักษาความปลอดภัยในโครงการนี้จะแบ่งเป็น 2 ส่วน คือ ส่วนที่เป็นโมดูลเซ็นเซอร์แก๊ส MQ-2 และส่วนที่เป็นโมดูลกล้องร่วมกับโมดูลตรวจจับความเคลื่อนไหว ซึ่งภาพรวมของระบบที่พิกอาศัย แสดงดังรูปที่ 3.1



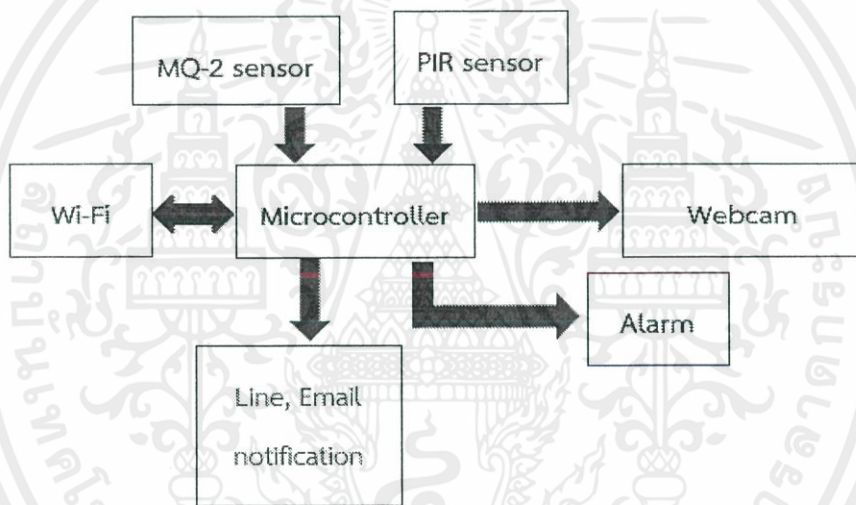
รูปที่ 3.1 ภาพรวมของระบบที่ใช้ในการออกแบบ

ทำการติดตั้งเซ็นเซอร์ MQ-2 ไว้ที่ห้องนั่งเล่น 1 จุด ขณะที่บริเวณหน้าบ้านทำการติดตั้งโมดูลกล้องร่วมกับโมดูลตรวจจับความเคลื่อนไหว 1 จุด ในสถานการณ์ปกติ ผู้ใช้งานระบบสามารถล็อกอินผ่านเว็บแอปพลิเคชันของ NETPIE ซึ่งเป็น cloud service ช่วย monitor เพื่อดูการทำงานของแต่ละอุปกรณ์เช่น สถานะของ alarm ผ่านแดชบอร์ด และในกรณีที่เกิดการเตือนขึ้นจากอุปกรณ์เนื่องจากสามารถตรวจจับกลิ่นแก๊สหรือควันไฟได้ ระบบจะสั่งให้ลำโพงดัง และขณะเดียวกันส่งข้อความเตือนไปยังอีเมลของเจ้าของ อีกทั้งเจ้าของสามารถรับรู้ข้อความเตือนได้ทันที เมื่อมีอีเมลนั้นเข้ามา โดยผ่านทางไลน์แอปพลิเคชันของเจ้าของอีกด้วย สำหรับกรณีเมื่อสามารถตรวจจับความเคลื่อนไหวจากอุปกรณ์ PIR ระบบจะทำการบันทึกภาพลงใน SD การ์ด แล้วส่งภาพเหล่านั้นต่อไป

ยังเฟสบุ๊คของเจ้าของด้วย ซึ่งถ้าเจ้าของได้ทำการติดตั้งแอปพลิเคชันของ IFTTT ไว้ในโทรศัพท์มือถือด้วยแล้ว ก็สามารถที่จะเห็นข้อความเตือนนั้นและรูปภาพได้อย่างทันทีทันใด

3.1 การออกแบบ

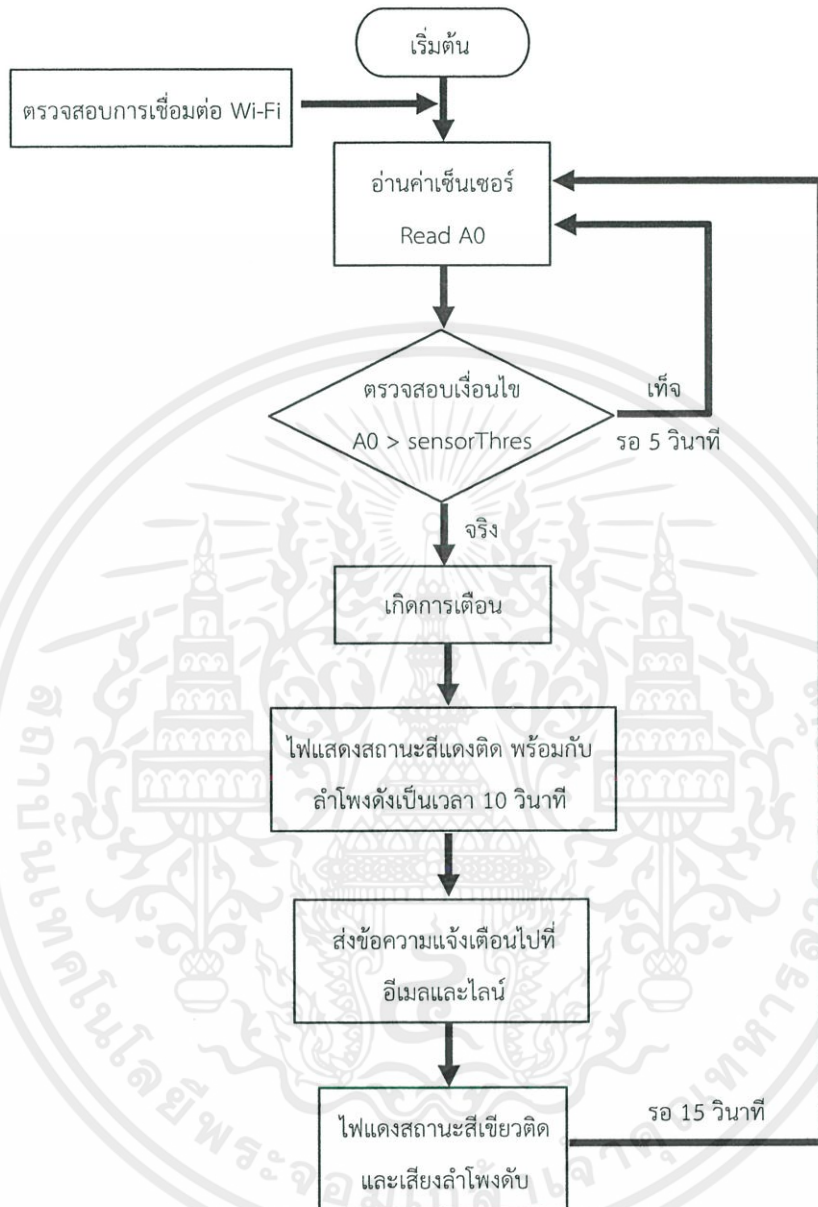
ในการทำโปรเจกต์นี้ได้ออกแบบไว้โดยใช้ตัวไมโครคอนโทรลเลอร์เชื่อมต่อ wi-fi เป็นตัวรับข้อมูลจากเซ็นเซอร์ต่าง ๆ เข้ามาแล้วแสดงค่าเป็นการแจ้งเตือนต่าง ๆ ดังแสดงในรูป 3.2 ที่เป็น Block diagram รวมของระบบ



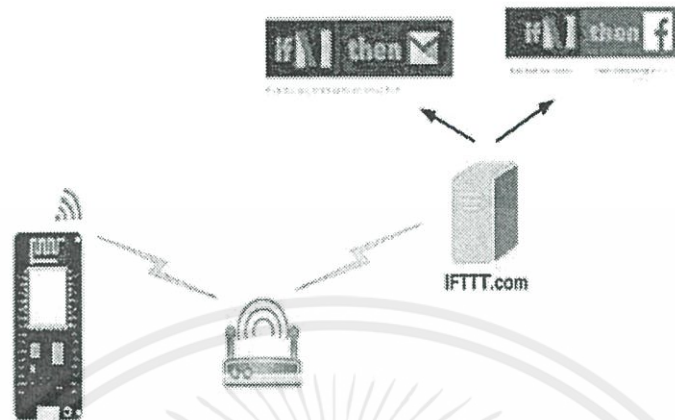
รูปที่ 3.2 Block diagram รวม

3.1.2 โมดูลเซ็นเซอร์แก๊ส MQ-2

แผนผังลำดับขั้นตอนของการเกิดการเตือนภัยของระบบที่ได้ทำการออกแบบไว้ของระบบเซ็นเซอร์ MQ-2 แสดงดังรูปที่ 3.2 ขณะที่รูปที่ 3.3 แสดงการทำงานของระบบขณะเชื่อมต่ออินเทอร์เน็ต



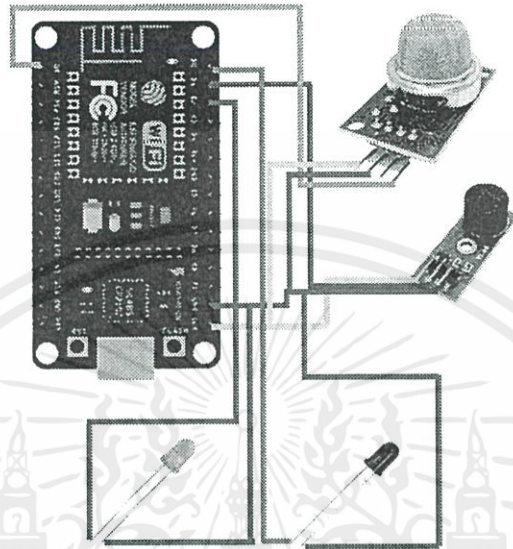
รูปที่ 3.3 แผนผังลำดับการเกิดการเตือนภัยของเซ็นเซอร์ แก๊ส MQ-2



รูปที่ 3.4 การทำงานของระบบขณะเชื่อมต่ออินเทอร์เน็ต

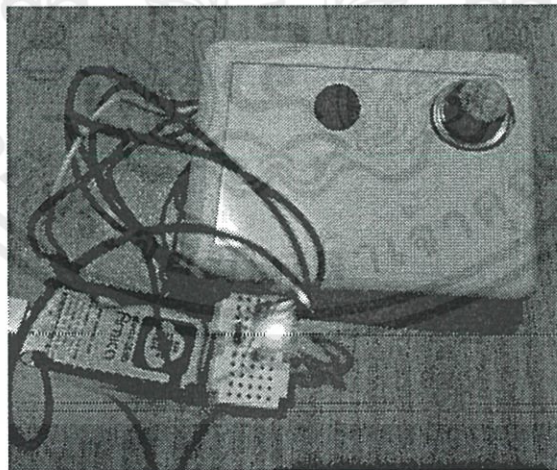
เมื่อเริ่มต้นระบบจะทำการเชื่อมต่อกับสัญญาณไวไฟก่อน เป็นลำดับแรกเพื่อเตรียมความพร้อมในการเชื่อมต่อสัญญาณ อินเทอร์เน็ต จากนั้นระบบจะทำการอ่านค่าสัญญาณเซ็นเซอร์จากพอร์ต A0 ของวงจร ESP8266 เปรียบเทียบกับค่า sensorThres ที่เรากำหนดขึ้น ถ้าค่าที่อ่านได้มีค่ามากกว่าค่า threshold นั้นแสดงว่าเซ็นเซอร์กำลังตรวจจับควันไฟหรือได้กลิ่นแก๊สรั่ว ณ ขณะนั้น ๆ ซึ่งสถานะ LED จะติดพร้อมกับเกิดเสียงคำโพงดังขึ้น เวลานั้นจะทำการส่งข้อความ เตือนภัยไปที่อีเมลของเจ้าของ รวมทั้งไลน์ไอดีของเจ้าของเช่นกัน ในกรณีที่สถานการณ์ปกติ ระบบก็จะวนอ่านค่าเซ็นเซอร์ไปเรื่อย ๆ เมื่อส่งทั้งข้อความเตือนภัยไปที่อีเมลแล้ว ระบบจะทำการรอก่อนที่จะกลับไปอ่านค่าเซ็นเซอร์อีกครั้ง และทำการตรวจเงื่อนไขอีกครั้งหนึ่ง แต่ถ้าเงื่อนไขเป็นเท็จ ก็ให้วนไปอ่านค่าของเซ็นเซอร์ใหม่เป็นอย่างนี้เรื่อย ๆ ในการแจ้งเตือน ไปยังอีเมลเมื่อมีสัญญาณเราจะใช้บริการ IFTTT ที่เป็น Web Application ในการรองรับการทริกเกอร์ (trigger) ไปสู่การโพสข้อความขึ้นอีเมล

3.1.2.1 การออกแบบวงจร



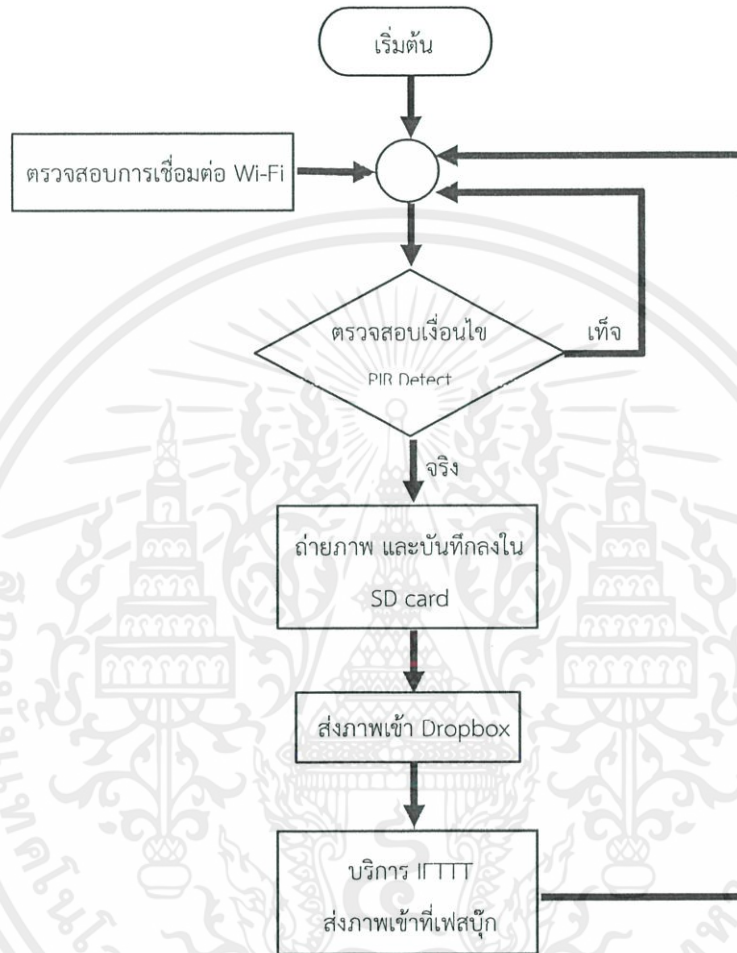
รูปที่ 3.5 วงจรของส่วน MQ-2

ในส่วนของเซ็นเซอร์ MQ-2 จะมีการต่อเซ็นเซอร์เข้ากับ NodeMCU และมีการต่อบuzzer กับ LED สีเขียวและแดงเพื่อเอาไว้เป็นการเตือนของระบบนี้ดังแสดงในรูปที่ 3.5 และรูปที่ 3.6 แสดงการนำวงจรที่ต่อลงบรรจุในกล่องอุปกรณ์



รูปที่ 3.6 MQ-2 ในกล่องอุปกรณ์

3.1.3 โมดูลกล้องและโมดูล PIR

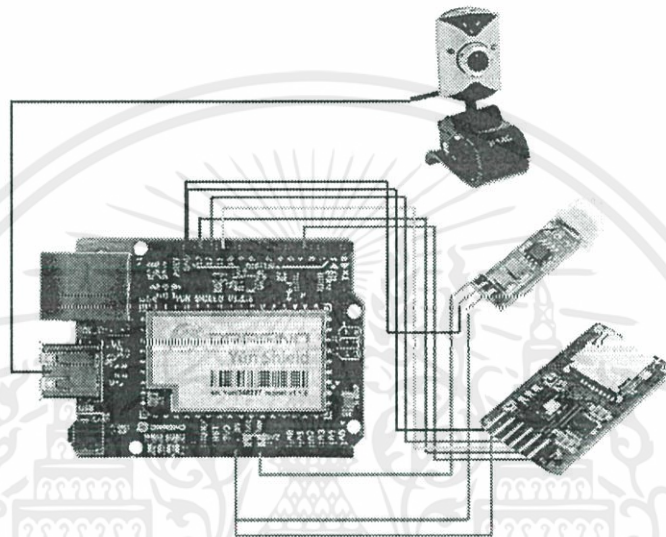


รูปที่ 3.7 ขั้นตอนการทำงานของโมดูลกล้องโมดูล PIR

ในรูปที่ 3.7 แสดงแผนผังลำดับขั้นตอนของการเกิดการเตือนภัยในส่วนที่ใช้ส่วนที่ใช้กล้อง HD แคม และโมดูลตรวจจับความเคลื่อนไหว คือ เมื่อเริ่มต้นระบบจะทำการเชื่อมต่อกับสัญญาณไวไฟก่อนเป็นลำดับแรกเพื่อเตรียมความพร้อมในการเชื่อมต่อสัญญาณอินเทอร์เน็ต จากนั้นระบบทำการอ่านค่าเซ็นเซอร์ตรวจจับความเคลื่อนไหว ซึ่งถ้ายังตรวจจับไม่เจอให้วนกลับไปอ่านค่าใหม่ ถ้าตรวจสอบเจอ (เป็นจริง) ระบบจะทำการถ่ายภาพ และบันทึกลงใน SD card จากนั้นส่งภาพเข้า Dropbox และจากนั้นด้วยบริการ IFTTT จะทำให้รูปภาพที่ถูกเก็บไว้ใน Dropbox folder ถูกทริก

เกอร์ จะส่งผ่านไปยังเฟสบุ๊คหน้า wall photo page รวมทั้งสามารถได้รับข้อความเตือนบอกและเห็นรูปขึ้นบนมือถือ ผ่านแอปพลิเคชัน IFTTT ด้วย

3.1.3.1 การออกแบบวงจร



รูปที่ 3.8 วงจรส่วน PIR และ webcam

ในส่วนของ PIR กับ webcam จะต่อเข้ากับ Yun Shield และมี Micro SD Card ต่อร่วมด้วยเพื่อเอาไว้บันทึกภาพที่ถ่ายได้จากตัว webcam ดังรูปที่ 3.8 และรูปที่ 3.9 คือนำวงจรที่ต่อบรรจุลงในกล่องอุปกรณ์



รูปที่ 3.9 PIR และ webcam ในกล่องอุปกรณ์

3.2 เครื่องมือที่ใช้ในการทดลอง

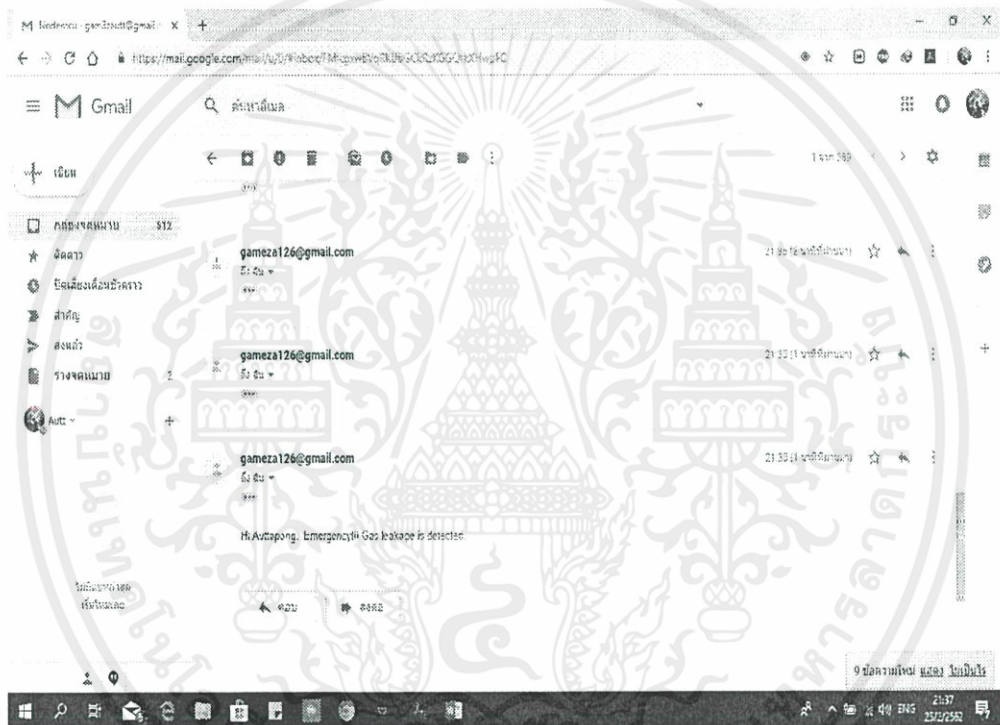
- 3.2.1 NodeMCU ESP8266
- 3.2.2 MQ-2 sensor
- 3.2.3 PIR sensor
- 3.2.4 Buzzer
- 3.2.5 Webcam
- 3.2.6 LED
- 3.2.7 Yun shield
- 3.2.8 Micro SD card

3.3 การจัดเก็บผลการทดลอง

3.3.1 ผลการทดลองของโมดูลเซ็นเซอร์แก๊ส MQ-2

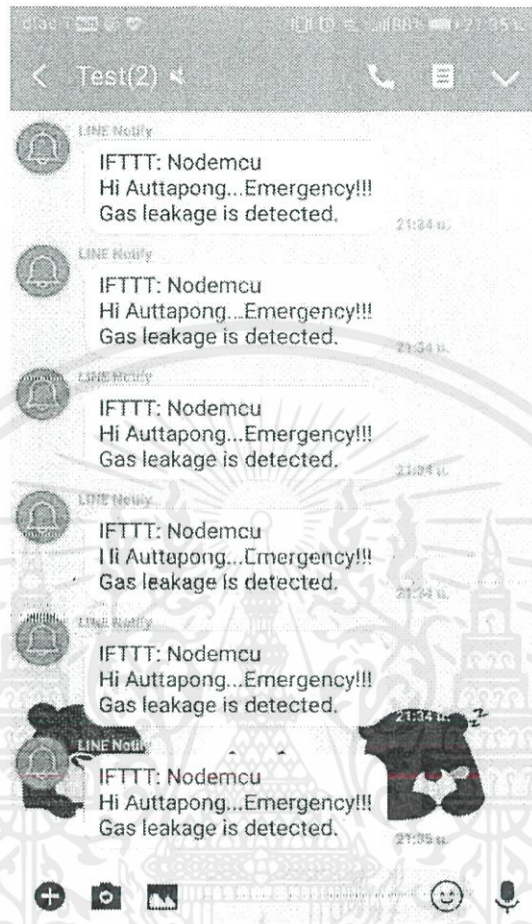
หลังจากที่ได้ทำการติดตั้งแล้วโมดูลเซ็นเซอร์แก๊ส MQ2 แล้ว จากนั้นจะทำการจำลองสถานการณ์จริงเพื่อเป็นการทดสอบระบบ โดยจะทำการเผาเศษกระดาษเพื่อให้เกิดควันไฟจริงขึ้น แล้วนำไปจ่อใกล้ๆกับอุปกรณ์ ซึ่งขั้นตอนที่สำคัญคือตั้งเปิดสัญญาณไวฟายขณะที่ทำการทดสอบ

ระบบด้วย เมื่อระบบเชื่อมกับสัญญาณไวไฟแล้ว ในการมอนิเตอร์หรือดูการทำงานของอุปกรณ์นั้น จะผ่านทางเว็บแอปพลิเคชันของ NETPIE ซึ่งสามารถใช้โทรศัพท์มือถือหรือเปิดดูผ่านเว็บเบราว์เซอร์ ได้ โดยจะเกิดการเตือนขึ้นเมื่อค่าที่ monitor มีค่ามากกว่าค่าขีดเริ่ม (sensorThres) ซึ่งในที่นี้ กำหนดค่าขีดเริ่มในโปรแกรมเท่ากับ 450 เพราะฉะนั้น เมื่อเซ็นเซอร์ตรวจจับค่าได้มากกว่า 500 จะเกิดการเตือนขึ้น จากนั้นผู้ใช้ได้รับข้อความการแจ้งเตือนผ่านแอปพลิเคชันของ IFTTT ข้อความที่ระบุอยู่ใน APP IFTTT และจะมีการส่งข้อความไปที่อีเมลร่วมด้วยดังรูปที่ 3.10



รูปที่ 3.10 การแจ้งเตือนผ่านทางอีเมล

นอกจากการเตือนในส่วนเหล่านี้แล้วได้เชื่อมต่อการเตือนผ่าน “Line notify” เข้าไปด้วย โดยใช้บริการของ IFTTT เช่นกันดังรูปที่ 3.11



รูปที่ 3.11 การแจ้งเตือนผ่านทางข้อความไลน์

3.3.2 ผลการทดลองของโมดูลตรวจจับความเคลื่อนไหว

หลังจากที่ได้ทำการติดตั้งอุปกรณ์ต่าง ๆ แล้ว จากนั้นจะทำการจำลองสถานการณ์จริง เพื่อเป็นการทดสอบระบบโดยทำการเดินเข้ามาภายในบ้าน เมื่อโมดูลมินิ PIR สามารถตรวจจับความเคลื่อนไหว จากนั้นก็ทำการถ่ายภาพ และบันทึกไปยัง SD การ์ด และส่งข้อมูลภาพไปยัง Dropbox และ Facebook ตามลำดับ ผลการทดสอบระบบเมื่อมีบุคคลเดินเข้ามาในบริเวณบ้านซึ่งโมดูล PIR เกิดการตรวจจับความเคลื่อนไหว ซึ่งภาพถูกส่งมายัง Dropbox ด้วยบริการ IFTTT และรูปภาพถูกส่งไปยัง Facebook

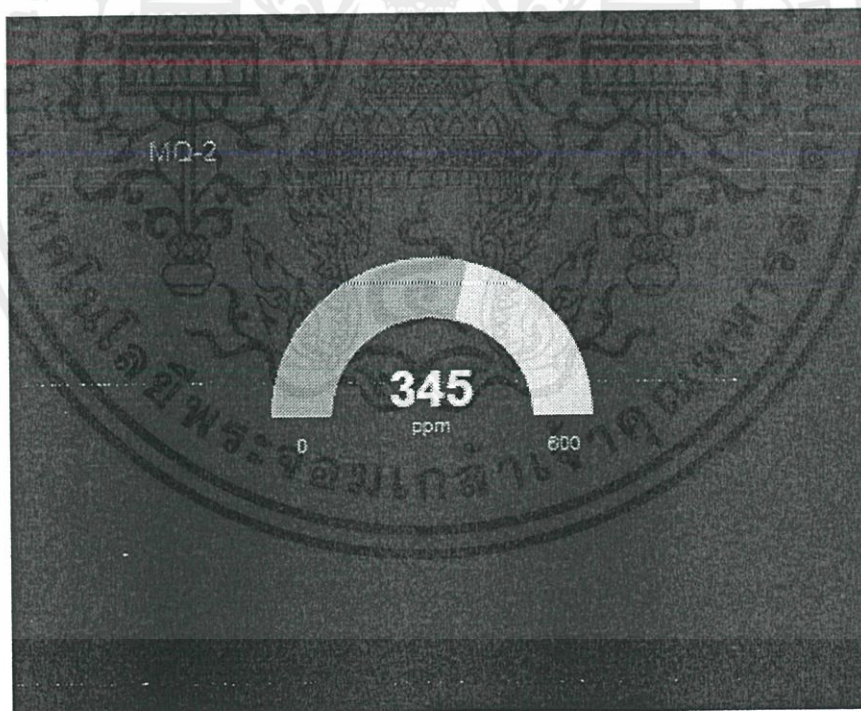
บทที่ 4

ผลการทดลอง

จากการทดลอง ได้ออกแบบการทดลองไว้เป็นสองส่วนคือ ส่วนที่เป็นโมดูลเซ็นเซอร์แก๊ส MQ-2 ซึ่งจะทำการติดตั้งไว้ภายในบ้าน และส่วนที่เป็นโมดูลตรวจจับการเคลื่อนไหวจะทำการติดตั้งไว้ที่นอกบ้าน

4.1 ผลการทดลองเซ็นเซอร์แก๊ส MQ-2

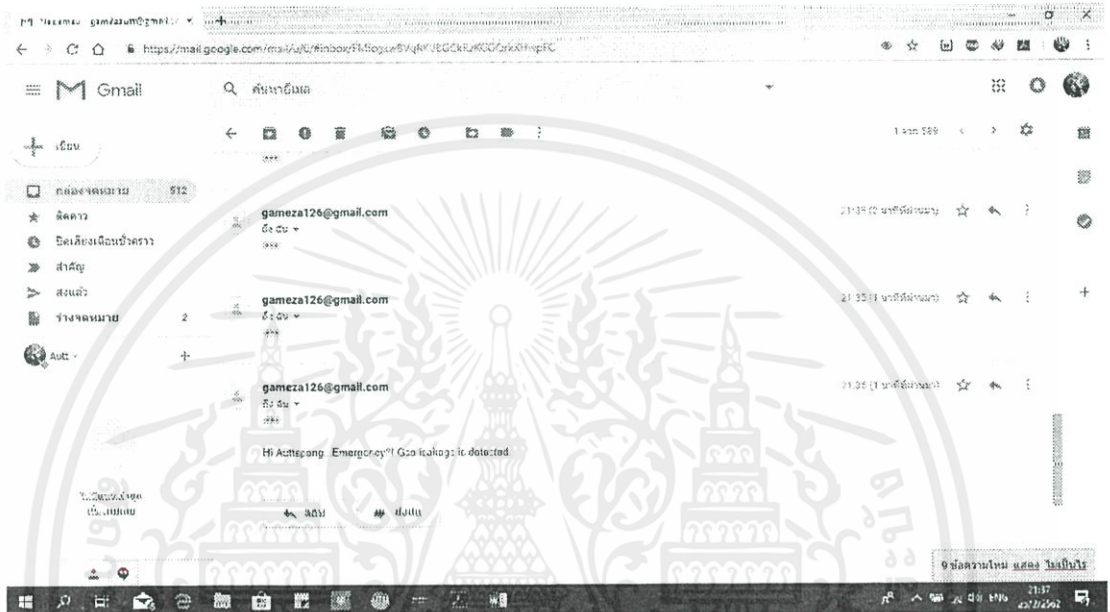
หลังจากติดตั้งและได้ทำการทดลอง เมื่อระบบทำงานอุปกรณ์จะเชื่อมต่อกับสัญญาณไวไฟยัดโนมิตี ซึ่งในเวลาปกติสามารถดูสถานการณ์ทำงานของอุปกรณ์ได้ผ่านทางเว็บแอปพลิเคชันของ NETPIE ซึ่งสามารถใช้โทรศัพท์หรือเปิดดูผ่านเว็บเบราว์เซอร์ก็ได้ โดยที่จะมีค่าขึ้นที่มอนิเตอร์ดังแสดงในรูปที่ 4.1



รูปที่ 4.1 monitor แสดงค่าที่อ่านได้จากเซ็นเซอร์

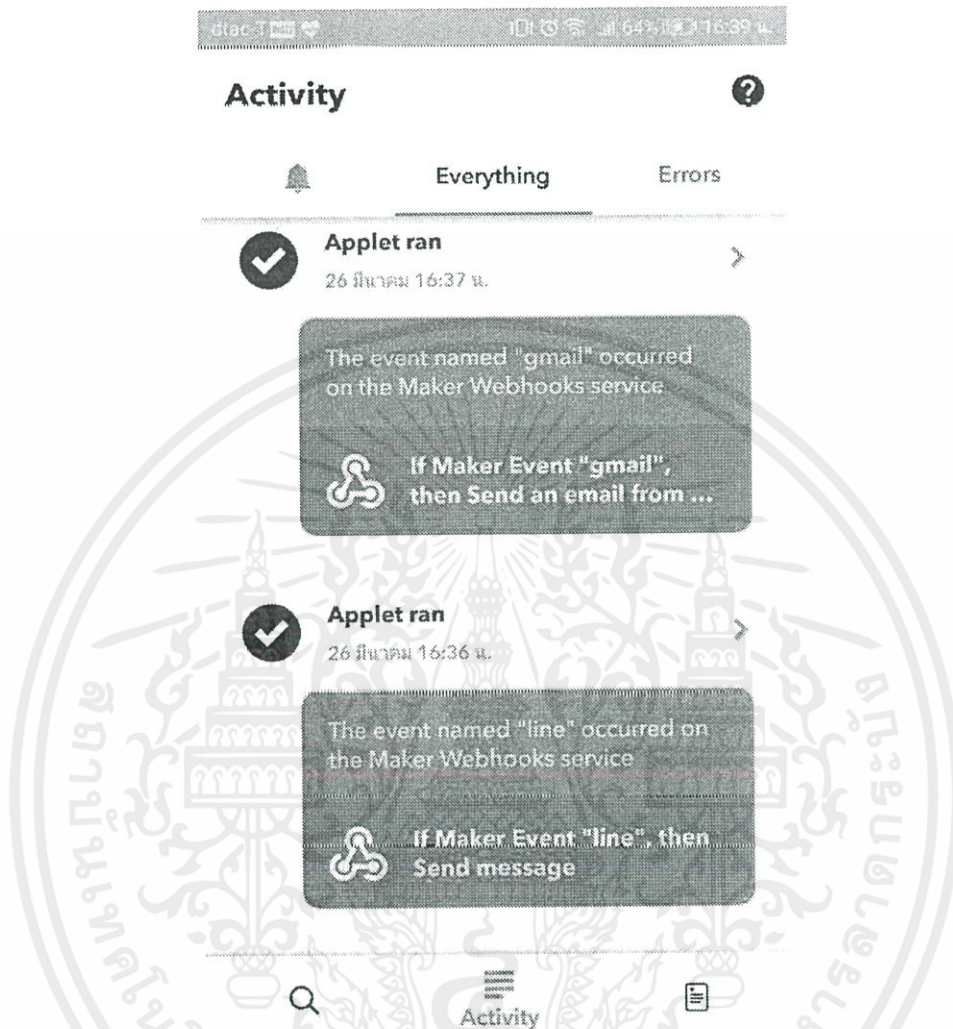
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีการกำหนดค่าไว้ (sensorThres) ซึ่งได้กำหนดไว้ที่ 500 เมื่อเซ็นเซอร์เกิดการรับค่าได้มากกว่าที่กำหนดไว้ก็จะเกิดการแจ้งเตือนผ่านแอปพลิเคชันของ IFTTT ข้อความที่ระบุไว้ใน APP IFTTT และจะมีการส่งข้อความไปที่อีเมลร่วมด้วยดังรูปที่ 4.2 – 4.3



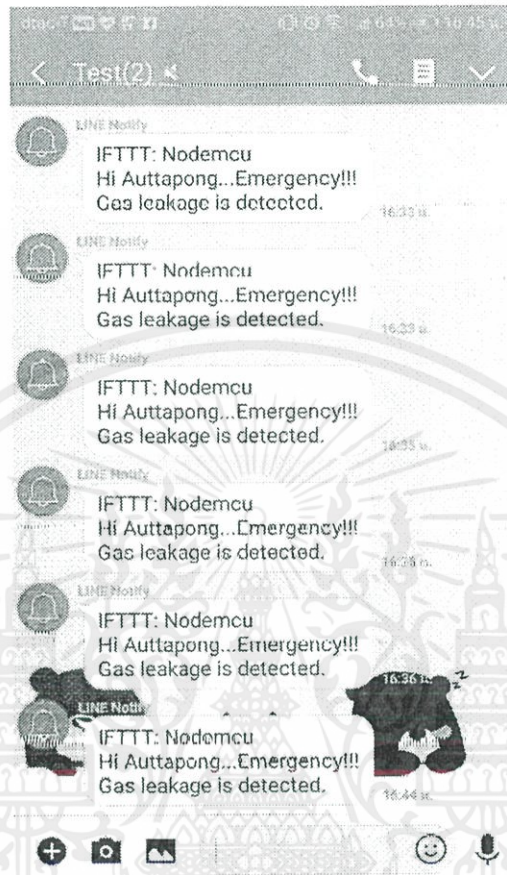
รูปที่ 4.2 การแจ้งเตือนผ่านทางอีเมล

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



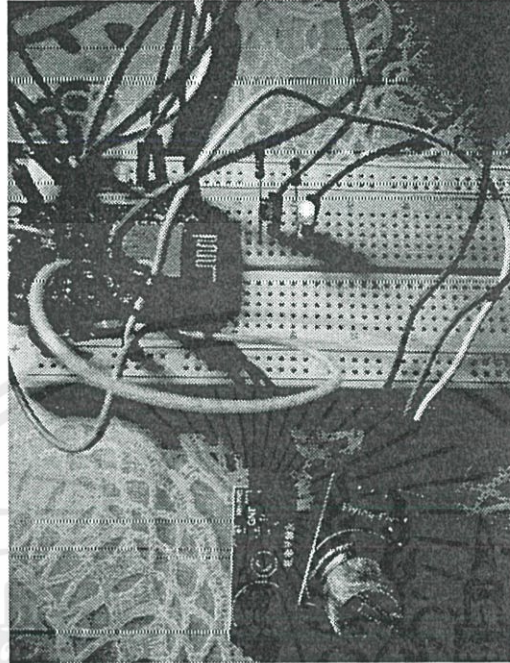
รูปที่ 4.3 ข้อความปรากฏขึ้นในโทรศัพท์หลังจากติดตั้ง APP IFTTT

นอกจากนี้ยังได้เชื่อมต่อการแจ้งเตือนผ่านทาง “Line notify” เข้าไปด้วย โดยใช้บริการของ IFTTT เช่นกัน ทำให้เห็นข้อความเตือนมายัง Line notify ดังรูปที่ 4.4

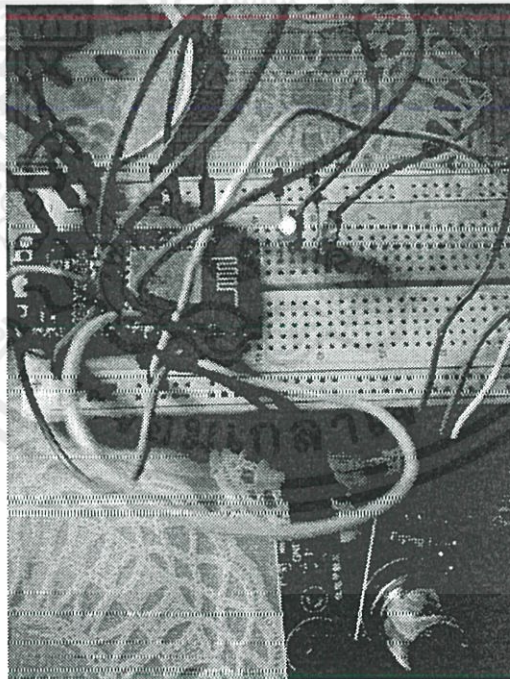


รูปที่ 4.4 ข้อความที่ปรากฏใน Line notify

รูปที่ 4.5 ซึ่ง LED สีเขียวติดอยู่เป็นการบ่งบอกว่าเซ็นเซอร์อ่านค่าได้น้อยกว่าที่ตั้งไว้ หรือก็คืออยู่ในสภาวะปกติ แต่รูปที่ 4.6 LED สีแดงติด ซึ่งหมายถึงว่าเซ็นเซอร์อ่านค่าได้มากกว่าค่าที่กำหนดไว้ อยู่ในสภาวะผิดปกติ ซึ่งอาจจะมีเหตุฉุกเฉินเกิดขึ้น



รูปที่ 4.5 ไฟสีเขียวติด แสดงสถานะปกติ



รูปที่ 4.6 ไฟสีแดงติด แสดงสถานะผิดปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ผลการทดลอง โมดูลตรวจจับการเคลื่อนไหว

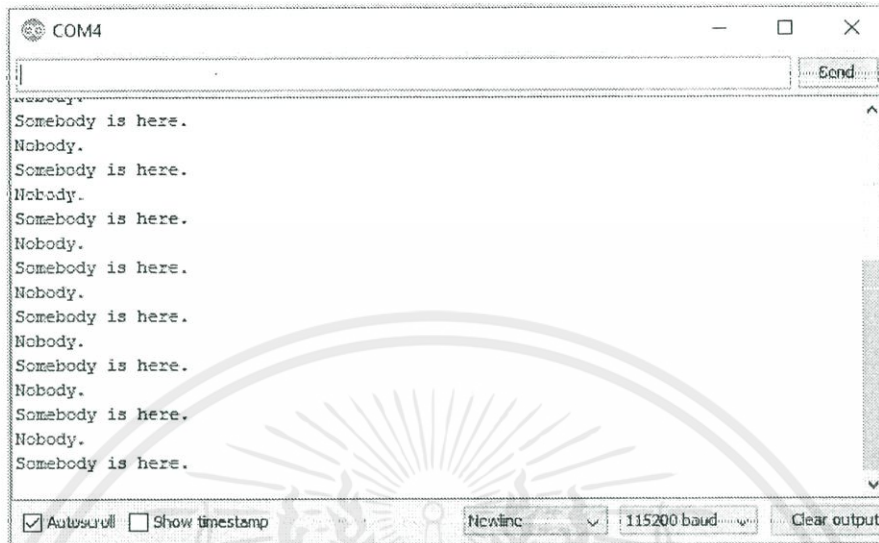
```

void loop()
{
  val = digitalRead(inputPin);
  if(val == HIGH)
  {
    digitalWrite(ledPin, HIGH);
    if (pirState == LOW)
    {
      Serial.println("Somebody is here.");
      pirState = HIGH;
    }
  }
  else |
  {
    digitalWrite(ledPin, LOW);
    if (pirState == HIGH)
    {
      Serial.println("Motion ended!");
      pirState = LOW;
    }
  }
}

```

รูปที่ 4.7 โค้ดการทำงานของ PIR

จากรูปที่ 4.7 คือโค้ดส่วนการทำงานของ PIR เซ็นเซอร์ โดยเริ่มแรกสมมติให้ไม่มีการตรวจจับการเคลื่อนไหวใดๆ เมื่อเข้าสู่การทำงานมีการรับค่าเข้ามาเป็น HIGH PIR เซ็นเซอร์ทำงานพร้อมกับแสดงข้อความบน Serial monitor ว่า “Somebody is here.” ดังรูปที่ 4.8 แต่หากไม่มีการรับค่าเข้ามาก็เกิดการเปลี่ยนจาก HIGH เป็น LOW พร้อมกับข้อความแสดงขึ้นว่า “Nobody.”



รูปที่ 4.8 PIR sensor on serial monitor

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

การจัดทำปฏิญานี้พันธนี้เป็น การประยุกต์องค์ความรู้ IoT (Internet of Things) เพื่อสร้างความปลอดภัยพื้นฐานให้กับบ้านพักอาศัยโดยมีต้นทุนของอุปกรณ์ไม่สูงจนเกินไป และสามารถติดตั้งใช้งานได้เอง และด้วยการบริการแบบไม่มีค่าใช้จ่ายจากทั้ง NETPIE (cloud service) , Dropbox และ IFTTT ทำให้โครงการนี้เปลี่ยนระบบเตือนภัยแบบเดิมๆ ให้มีความทันสมัย สามารถเชื่อมโยงอินเทอร์เน็ตสรรพสิ่งและการเตือนภัยเข้าหากันได้โดยไม่ต้องได้ยื่นเตือนภัยดังเหมือนในอดีต สำหรับการใช้งานในส่วนของเซ็นเซอร์แก๊สทั้ง MQ-2 หลังจากการติดตั้งและทดลองใช้เป็นที่น่าพอใจ ในการสร้างความอุ่นใจให้กับเจ้าของบ้านพักอาศัย พร้อมทั้งมีแดชบอร์ดช่วยมอนิเตอร์ว่าระบบยังคงทำงานได้ดีอยู่หรือไม่ ในกรณีที่ต้องการเพิ่มส่วนขยายเซ็นเซอร์แก๊สหรือเซ็นเซอร์ตรวจจับควันไฟไปยังบริเวณอื่นๆ ของบ้าน สามารถทำได้ง่าย เพียงแค่เพิ่มอุปกรณ์ปลั๊กอินในแต่ละโมดูลเข้าไปตามจุดต่างๆ ของบ้านที่ต้องการได้เลย

5.2 ข้อเสนอแนะ

กรณีที่ไปต่างจังหวัด ไม่อยู่บ้านและต้องการปิดระบบไฟฟ้าทั้งบ้าน แต่หากยังต้องการให้ระบบเซนเซอร์แก๊สเซ็นเซอร์ และเซ็นเซอร์ตรวจจับการเคลื่อนไหวสามารถทำงานอยู่ได้ จำเป็นต้องติดตั้งแบตเตอรี่แห้งให้กับอุปกรณ์ access point หรือ router เพื่อให้สามารถเชื่อมต่อกับอินเทอร์เน็ตได้ต่อ ขณะที่อุปกรณ์บอร์ด ESP8266 จำเป็นต้องติดตั้งแบตเตอรี่เพื่อให้ทำงานได้เช่นกัน ซึ่งอาจจะใช้ เซลล์แบตเตอรี่ลิเทียมโพลีเมอร์ หรือถ่านชาร์จลิเทียม ก็ได้เป็นต้น

บรรณานุกรม

- [1] “TECHNICAL DATA MQ-2 GAS SENSOR.” <http://gas-sensor.ru/pdf/combustible-gas-sensor.pdf>.
- [2] ธีรวิฑูร จิตรพธมา และ ชัยวัฒน์ ลิ้มพรจิตรวิไล, “เริ่มต้นใช้งาน NETPIE กับ ESP8266/NodeMCU-12E.” บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด. 2016
- [3] “MQ2/MQ5 โมดูลเซ็นเซอร์ตรวจจับควันไฟ.”
<http://www.arduinoecia.com.br/2015/01/alarme-Sensor-de-gas-modulo-mq-2.htm>.
- [4] “แหล่งที่มาของการให้บริการ IFTTT.” <https://ifttt.com/>.
- [5] “การให้บริการของฟรีบอร์ดไอโอ.” <https://freeboard.io/>.
- [6] “Node MCU.” <https://surepornce.wordpress.com/category/บทเรียน/>.
- [7] “Internet Of Things (IoT) คืออะไร.”
<https://blog.sogoodweb.com/Article/Detail/59554>.
- [8] Nokkaew. “IFTTT เมื่อจับ Service ทุกอย่างมาয়ারวมกัน แล้วสั่งให้มันทำงานอัตโนมัติ.” If this then that. The Zero Publishing Co., Ltd, 2016.
- [9] NETPIE. [Online].Available : <https://www.nectec.or.th/innovation/innovation-software/netpie.html>
- [10] “โมดูลบันทึกข้อมูล Micro SD Card Module.”
<https://www.thaielecs.com/product/208/โมดูลบันทึกข้อมูล-micro-sd-card-module>.



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โค้ดการทำงานในส่วนของเซ็นเซอร์ MQ-2

```

#include <AuthClient.h>
#include <debug.h>
#include <MicroGear.h>
#include <MQTTClient.h>
#include <PubSubClient.h>
#include <SHAL.h>
#include <MQ2.h>
#include <ESP8266WiFi.h>

const char* ssid      = "11328 2.4GHz";           //ชื่อเครือข่าย wifi ที่ NodeMCU เชื่อมต่อ
const char* password  = "ap0918720441";         //รหัสผ่านเครือข่าย wifi ที่ NodeMCU เชื่อมต่อ

int sensorThres = 500;    //ค่าเกณฑ์ค่าของแก๊ส
#define LEDred D0
#define LEDgreen D2
#define BUZZ D1
int GAS = A0;
MQ2 mq2(GAS);

#define APPID      "HomeSecur"           //ชื่อ Application ID ที่สร้างบน NETPIE
#define KEY        "MYdlyziCw4pUzX7"    //Key ของอุปกรณ์ ที่ได้จากการสร้างบน NETPIE
#define SECRET     "TX71YeEEMICREDSZMaUd9X08K" //Secret ของอุปกรณ์ ที่ได้จากการสร้างบน NETPIE
#define ALIAS      "mq2"                //ตั้งชื่อ Alias ให้กับ Things นี้ เป็นชื่อเรียกเวลา chat มาหา
#define SCOPE      ""

WiFiClient client;

String MakerIFTTT_Key ;
String MakerIFTTT_Event;
char 'append_str(char *here, String s) { int i=0; while (*here++ = s[i]){i++;}return here-1;}
char 'append_ul(char *here, unsigned long u) { char buf[20]; return append_str(here, ultoa(u, buf, 10));}
char post_reqst[256];char *p;char *content_length_here;char *json_start;int compl;

AuthClient *authclient;
MicroGear microgear(client);

void onConnected(char *attribute, uint8_t* msg, unsigned int msglen)
{
  Serial.println("Connected to NETPIE...");
  microgear.setAlias(ALIAS);
}

void onMsgHandler(char *topic, uint8_t* msg, unsigned int msglen)
{
  Serial.print("Incoming message --> ");
  Serial.print(topic);
  Serial.print(" : ");
  char strState[msglen];
  for (int i = 0; i < msglen; i++)
  {
    strState[i] = (char)msg[i];
    Serial.print((char)msg[i]);
  }
  Serial.println();
}

```

```

void setup()
{
  // put your setup code here, to run once:
  microgear.on(MESSAGE, onMsgHandler);
  microgear.on(CONNECTED, onConnected);

  Serial.begin(115200);
  Serial.println("Starting...");
  pinMode(D0, OUTPUT);
  pinMode(D1, OUTPUT);
  pinMode(D2, OUTPUT);
  pinMode(GAS, INPUT);

  if (WiFi.begin(ssid, password))
  {
    while (WiFi.status() != WL_CONNECTED)
    {
      delay(500);
      Serial.print(".");
    }

    Serial.println("WiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
    microgear.init(KEY, SECRET, ALIAS);
    microgear.connect(APPID);
    mq2.begin();
  }

  void loop()
  {
    if (microgear.connected())
    {
      microgear.loop();
      //delay(300);
      Serial.println("connect...");

      int value = analogRead(GAS);
      microgear.publish("/analogRead/MQ2", value);
    }
    else
    {
      Serial.println("connection lost, reconnect...");
      microgear.connect(APPID);
    }

    int sensorValue = analogRead(GAS);
    Serial.print("Valule ");
    Serial.println(sensorValue);
    if (sensorValue > sensorThres)
    {
      if (client.connect("maker.ifttt.com", 80))

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//ส่วนการแจ้งเตือนทาง gmail
MakerIFTTT_Key = "lnQmC8rgICy7KzNztp6DZ";
MakerIFTTT_Event = "gmail";
p = post_rqst;
p = append_str(p, "POST /trigger/");
p = append_str(p, MakerIFTTT_Event);
p = append_str(p, "/with/key/");
p = append_str(p, MakerIFTTT_Key);
p = append_str(p, " HTTP/1.1\r\n");
p = append_str(p, "Host: maker.ifttt.com\r\n");
p = append_str(p, "Content-Type: application/json\r\n");
p = append_str(p, "Content-Length: ");
content_length_here = p;
p = append_str(p, "NN\r\n");
p = append_str(p, "\r\n");
json_start = p;
p = append_str(p, "{\"value1\":\");
p = append_str(p, "Nodemcu");
p = append_str(p, "\",\"value2\":\");
p = append_str(p, "Hi Auttapong...Emergency!!!");
p = append_str(p, "\",\"value3\":\");
p = append_str(p, "Gas leakage is detected.");
p = append_str(p, "\");

compi= strlen(json_start);
content_length_here[0] = '0' + (compi/10);
content_length_here[1] = '0' + (compi%10);
client.print(post_rqst);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//ส่วนการส่งข้อมูลทาง line
if (client.connect("maker.ifttt.com",80))
{
  MakerIFTTT_Key ="lnQmC8rgICg7KzNztp6DZ";
  MakerIFTTT_Event ="line";
  p = post_rqst;
  p = append_str(p, "POST /trigger/");
  p = append_str(p, MakerIFTTT_Event);
  p = append_str(p, "/with/key/");
  p = append_str(p, MakerIFTTT_Key);
  p = append_str(p, " HTTP/1.1\r\n");
  p = append_str(p, "Host: maker.ifttt.com\r\n");
  p = append_str(p, "Content-Type: application/json\r\n");
  p = append_str(p, "Content-Length: ");
  content_length_here = p;
  p = append_str(p, "NN\r\n");
  p = append_str(p, "\r\n");
  json_start = p;
  p = append_str(p, "{\"value1\":\");
  p = append_str(p, "Nodemcu");
  p = append_str(p, "\",\"value2\":\");
  p = append_str(p, "Hi Auttapong...Emergency!!!");
  p = append_str(p, "\",\"value3\":\");
  p = append_str(p, "Gas leakage is detected.");
  p = append_str(p, "\"");

  compi= strlen(json_start);
  content_length_here[0] = '0' + (compi/10);
  content_length_here[1] = '0' + (compi%10);
  client.print(post_rqst);

  digitalWrite(LEDred, HIGH);
  digitalWrite(LEDgreen, LOW);
  tone(BUZZ, 1000, 10000);
  Serial.println("Your message send.....");
  Serial.println("Gas leakage is detected");
  }
  delay(15000);
  }

else
{
  digitalWrite(LEDred, LOW);
  digitalWrite(LEDgreen, HIGH);
  noTone(BUZZ);
  }
  delay(5000);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้