ระบบดูแลความปลอดภัยสำหรับหมู่บ้านจัดสรร
SECURITY SYSTEM FOR VILLAGE

โดย
นายวีรพงษ์ แก้วมณี
นายศรัณญ์ เกษศิลป์
นายอัญวุฒิ สินธพเลิศชัยกุล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2560

ระบบดูแลความปลอดภัยสำหรับหมู่บ้านจัดสรร
Security system for village

โดย

นายวีรพงษ์	แก้วมณี	57011197
นายศรัณญ์	เกษศิลป์	57011215
นายอัญวุฒิ	สินธพเลิศชัยกุล	57011521

อาจารย์ที่ปรึกษา

ผศ.ดร. พิเชฐ ม่วงนวล

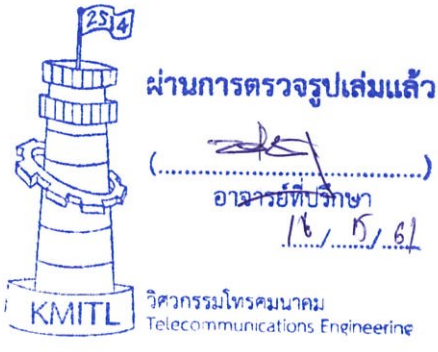
ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560



ปริญญานิพนธ์ปีการศึกษา 2560

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบดูแลความปลอดภัยสำหรับหมู่บ้านจัดสรร

SECURITY SYSTEM FOR VILLAGE

ผู้จัดทำ

1. นายวีรพงษ์ แก้วมณี 57011197
2. นายศรัณญ์ เกษศิลป์ 57011215
3. นายอัญวุฒิ สินธพเลิศชัยกุล 57011521

..... อาจารย์ที่ปรึกษา

(ผศ.ดร. พิเชฐ ม่วงนวล)

กิตติกรรมประกาศ

โครงงานนี้จะไม่สามารถสำเร็จลุล่วงไปได้ด้วยดี หากปราศจากคำแนะนำและความช่วยเหลือจากหลายๆภาคส่วน ซึ่งคนแรกที่ต้องขอกล่าวถึง คือ ผศ.ดร. พิเชฐ ม่วงนวล อาจารย์ที่ปรึกษาโครงงาน ที่คอยให้คำปรึกษา ให้คำแนะนำ คอยย้ำเตือนกวดขันให้งานคืบหน้า และคอยสนับสนุนทุกๆอย่าง อย่างดีเสมอมา

และขอขอบพระคุณคณะอาจารย์ ภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่คอยอบรมสั่งสอน ให้ความรู้เสมอมา

นอกจากนี้ยังต้องขอขอบคุณ พี่ๆเพื่อนๆและน้องๆทุกคนที่คอยช่วยเหลือ สนับสนุนและให้กำลังใจต่อกันเรื่อยมา

และท้ายที่สุด ต้องกราบขอบพระคุณไปยังคุณพ่อและคุณแม่ที่ได้เลี้ยงดูเอาใจใส่ และคอยสนับสนุน ให้โอกาสในด้านการศึกษาอย่างเต็มที่ ทำให้ข้าพเจ้าได้มีความรู้ และเป็นคนดีได้อย่างทุกวันนี้ ทั้งยังคอยเป็นกำลังใจให้ตลอดมา

นายวีรพงษ์ แก้วมณี
นายศรัณญ์ เกษศิลป์
นายอัญวุฒิ สินธพเลิศชัยกุล
ผู้จัดทำ

ระบบดูแลความปลอดภัยสำหรับหมู่บ้านจัดสรร
SECURITY SYSTEM FOR VILLAGE

โดย นายวีรพงษ์ แก้วมณี 57011197
นายศรัณญ์ เกษศิลป์ 57011215
นายอัญวุฒิ สินธพเลิศชัยกุล 57011521

อาจารย์ที่ปรึกษา ผศ.ดร. พิเชฐ ม่วงนวล

บทคัดย่อ

ปริญญานิพนธ์นี้เป็นการนำเสนอระบบดูแลความปลอดภัยสำหรับหมู่บ้านจัดสรร โดยภายในบ้านแต่ละหลังจะทำการติดตั้งเซนเซอร์ต่างๆเพื่อตรวจจับเหตุการณ์ผิดปกติเช่นเหตุเพลิงไหม้และเหตุโจรกรรม เมื่อมีเหตุเพลิงไหม้หรือเหตุโจรกรรมเซนเซอร์จะส่งสัญญาณแบบไร้สายไปยังไมโครคอนโทรลเลอร์โดยใช้โมดูลสื่อสารจากนั้นไมโครคอนโทรลเลอร์จะจ่ายแรงดันให้ Buzzer ดังขึ้นเพื่อแจ้งเตือนบ้านที่อยู่บริเวณรอบๆและนอกจากนั้นไมโครคอนโทรลเลอร์ยังส่งข้อมูลความผิดปกติไปยังศูนย์ควบคุมส่วนกลางผ่านทางเครือข่ายอินเตอร์เน็ตเพื่อเตือนเจ้าหน้าที่รักษาความปลอดภัยและแจ้งเตือนเจ้าของบ้านถึงเหตุการณ์ผิดปกติผ่านทางแอพพลิเคชั่น Line ดังนั้นเจ้าของบ้านและเจ้าหน้าที่รักษาความปลอดภัยประจำหมู่บ้านจึงสามารถรับรู้ถึงเหตุการณ์ที่เกิดขึ้นและสามารถเข้าไปแก้ไขปัญหาที่เกิดขึ้นได้ทันท่วงที

ABSTRACT

This project presents security system for village. Many types of sensor have to be installed in each house to detect fire and thief. When sensors can detect abnormal event, it'll send signal to microcontroller by using wireless module. Microcontroller will make three way to alarm. First, Buzzer is used to alarm neighborhood. Second, Line application is used to notify house's owner what problems have been happened. Third, Microcontroller send data to the central control center through internet for notifying security guard. As the result, security guard and house's owner can immediately manage that situation.

สารบัญ

		หน้า
กิตติกรรมประกาศ		I
บทคัดย่อ		II
สารบัญ		III
สารบัญรูป		V
บทที่ 1	**บทนำ**	**1**
	1.1 ความเป็นมาและความสำคัญของปัญหา	1
	1.2 วัตถุประสงค์	1
	1.3 ขอบเขตของปริญญานิพนธ์	1
บทที่ 2	**ทฤษฎีและหลักการที่เกี่ยวข้อง**	**3**
	2.1 ไมโครคอนโทรลเลอร์	3
	2.2 เซนเซอร์ตรวจจับการเคลื่อนไหว	8
	2.3 เซนเซอร์ตรวจจับเปลวไฟ	9
	2.4 โมดูลสื่อสารไร้สาย NRF24L01	11
	2.5 บัซเซอร์	13
	2.6 การสร้างฐานข้อมูล	13
	2.7 การสร้างหน้าเว็บเพจแสดงผลการแจ้งเตือน	16
	2.8 โปรแกรม XAMPP	19
	2.9 โปรแกรม APACHE	21
	2.10 โปรแกรม PHPMYADMIN	21
	2.11 ภาษา PYTHON	22
	2.13 กล้องเว็บแคม	23

สารบัญ (ต่อ)

		หน้า
บทที่ 3	**การออกแบบและการจัดทำปริญญานิพนธ์**	**25**
	3.1 การออกแบบ	25
	3.2 เครื่องมือที่ใช้ในการทดลอง	42
	3.3 การจัดเก็บผลการทดลอง	46
บทที่ 4	**ผลการทดลอง**	**48**
	4.1 ผลการทดสอบเซนเซอร์ตรวจจับเปลวไฟเบื้องต้น	48
	4.2 ผลการทดสอบเซนเซอร์ตรวจจับการเคลื่อนไหว	50
	4.3 ผลการทดสอบในส่วนของระบบการรับส่งข้อมูลไร้สายในรูปแบบดิจิตอล	52
	4.4 ผลการทดสอบระบบในส่วนของราสป์เบอร์รีไพ	54
	4.5 ผลการทดสอบในส่วนของการใช้งานกล้องเว็บแคม	59
	4.6 ผลการทดสอบในส่วนของหน้าจอมอนิเตอร์	61
	4.7 ผลการออกแบบฐานข้อมูล	69
	4.8 ผลการออกแบบในส่วนของการแจ้งเตือนผ่านแอปพลิเคชั่นไลน์	71
บทที่ 5	**สรุปผลและข้อเสนอแนะ**	**73**
	5.1 สรุปผล	73
	5.2 ข้อเสนอแนะ	73
บรรณานุกรม		**75**
ภาคผนวก		**77**

สารบัญรูป

รูปที่		หน้า
1.1	บล็อกไดอะแกรมของโครงงาน	2
2.1	โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์	4
2.2	ราสป์เบอร์รีไพ 3 โมเดลบี	5
2.3	คุณสมบัติของขา GPIO	6
2.4	ARDUINO NANO 3.0	6
2.5	ขาต่างๆของ ARDUINO NANO	7
2.6	เซนเซอร์ตรวจจับการเคลื่อนไหว	8
2.7	วงจรภายใน PIR SENSOR MODULE	9
2.8	เซนเซอร์ตรวจจับเปลวไฟ	9
2.9	การเชื่อมต่อเซนเซอร์ตรวจจับเปลวไฟกับ ARDUINO	10
2.10	โมดูลสื่อสารไร้สาย NRF24L01	11
2.11	การต่อขาโมดูลสื่อสารไร้สาย NRF24L01 กับ ARDUINO	12
2.12	บัซเซอร์	13
2.13	โปรแกรม MYSQL	14
2.14	ระบบจัดการฐานข้อมูลแบบข้อมูลเชิงสัมพันธ์	14
2.15	ภาษา SQL	15
2.16	ภาษา HTML	16
2.17	ตัวอย่างภาษา HTML	17
2.18	ภาษา PHP	17
2.19	ตัวอย่างภาษา PHP	18
2.20	โปรแกรม XAMPP	19
2.21	หน้าต่างเริ่มต้นโปรแกรม XAMPP	20
2.22	หน้าต่างการทำงานโปรแกรม XAMPP	20
2.23	โปรแกรม APACHE	21
2.24	ตัวอย่างโครงสร้างฐานข้อมูล	22
2.25	โปรแกรม PYTHON	23

สารบัญรูป (ต่อ)

รูปที่		หน้า
2.26	ตัวอย่างคำสั่ง PYTHON	23
2.27	กล้องเว็บแคมยี่ห้อ OKER รุ่น 088	24
3.1	บล็อกไดอะแกรมของระบบ	26
3.2	วงจรรวมของระบบตรวจจับอัคคีภัยฝั่งเครื่องส่ง	27
3.3	วงจรรวมของระบบฝั่งเครื่องรับและระบบกล้องวงจรปิด	28
3.4	วงจรของรีโมทคอนโทรล	29
3.5	ลายแผ่น PCB ของวงจรรีโมทคอนโทรล	29
3.6	ผังการทำงานของไมโครคอนโทรลเลอร์ที่ต่อกับ IR SENSOR	30
3.7	ผังการทำงานของไมโครคอนโทรลเลอร์ส่วนที่ 1	32
3.8	ผังการทำงานของไมโครคอนโทรลเลอร์ส่วนที่ 2	33
3.9	ผังการทำงานของไมโครคอนโทรลเลอร์ที่ต่อกับสวิตช์ต่างๆ	35
3.10	ผังการทำงานของหน้าจอมอนิเตอร์	36
3.11	ผังการทำงานของหน้าต่างแสดงประวัติ	38
3.12	ผังการทำงานของหน้าต่างแสดงสถานะอุปกรณ์	39
3.13	ตัวแปรต่างๆที่จะเก็บในฐานข้อมูลที่มีชื่อว่า “SECURITY”	41
3.14	ตัวแปรต่างๆที่จะเก็บในฐานข้อมูลที่ชื่อว่า “LOGFILE”	42
3.15	ราสป์เบอร์รีไพ 3 โมเดลบี	42
3.16	เซนเซอร์ตรวจจับการเคลื่อนไหว	43
3.17	เซนเซอร์ตรวจจับเปลวไฟ	43
3.18	บัซเซอร์	44
3.19	โมดูลสื่อสารไร้สาย NRF24L01	44
3.20	ไมโครคอนโทรลเลอร์อาดูโนนาโน	45
3.21	กล้องเว็บแคมยี่ห้อ OKER รุ่น 088	45
4.1	สัญญาณเอาต์พุตในกรณีที่ไม่มีการตรวจจับเปลวไฟ	48
4.2	สัญญาณเอาต์พุตในกรณีที่ตรวจจับเปลวไฟได้	49
4.3	สัญญาณเอาต์พุตของ IR FLAME SENSOR 2 ตัว	49

สารบัญรูป (ต่อ)

รูปที่		หน้า
4.4	สัญญาณเอาต์พุตของ PIR SENSOR MODULE เมื่อไม่มีการตรวจจับมนุษย์หรือสิ่งมีชีวิต	50
4.5	สัญญาณเอาต์พุตของ PIR SENSOR MODULE เมื่อมีการตรวจจับสิ่งมีชีวิตในโหมด L	51
4.6	สัญญาณเอาต์พุตของ PIR SENSOR MODULE เมื่อมีการตรวจจับสิ่งมีชีวิตในโหมด H	51
4.7	การทดสอบโมดูล NRF24L01 ด้านส่ง	52
4.8	SERIAL MONITOR ของด้านส่ง	53
4.9	การทดสอบโมดูล NRF24L01 ด้านรับ	53
4.10	SERIAL MONITOR ของด้านรับ	54
4.11	หน้าต่าง PYTHON SHELL ของราสป์เบอร์รีไพเมื่อไม่มีเหตุการณ์ผิดปกติ	54
4.12	หน้าต่าง PYTHON SHELL ของราสป์เบอร์รีไพ เมื่อกดรีโมทคอนโทรลเพื่อเปิดระบบป้องกันการโจรกรรม	55
4.13	หน้าต่าง PYTHON SHELL ของราสป์เบอร์รีไพ เมื่อมีเหตุเพลิงไหม้	56
4.14	หน้าต่าง PYTHON SHELL ของราสป์เบอร์รีไพ เมื่อมีเหตุโจรกรรม	56
4.15	หน้าต่าง PYTHON SHELL ของราสป์เบอร์รีไพ เมื่อกดรีโมทคอนโทรลเพื่อส่งสัญญาณขอความช่วยเหลือ (SOS)	57
4.16	หน้าต่าง PYTHON SHELL ของราสป์เบอร์รีไพ เมื่อกดรีโมทคอนโทรลเพื่อหยุดเสียงเตือน	58
4.17	หน้าต่าง PYTHON SHELL ของราสป์เบอร์รีไพ เมื่อกดรีโมทคอนโทรลเพื่อปิดระบบป้องกันการโจรกรรม	58
4.18	เชื่อมต่อราสป์เบอร์รีไพกับกล้องเว็บแคม	59
4.19	ภาพถ่ายจากกล้องเว็บแคม	60
4.20	ไฟล์วีดีโอจากกล้องเว็บแคม	60
4.21	หน้าจอแสดงผลเมื่อเหตุการณ์ปกติ	61
4.22	ข้อมูลในฐานข้อมูลเมื่อเหตุการณ์ปกติ	61
4.23	ฐานข้อมูลเมื่อมีเหตุการณ์ไฟไหม้บ้านของ ALICE	62
4.24	หน้าจอมอนิเตอร์เมื่อบ้านเลขที่ “195” มีเหตุการณ์ไฟไหม้	62

สารบัญรูป (ต่อ)

รูปที่		หน้า
4.25	ฐานข้อมูลเมื่อมีเหตุการณ์ไฟไหม้บ้านของ ALICE และ SUSA	62
4.26	หน้าจอมอนิเตอร์เมื่อมีเหตุการณ์ไฟไหม้ 2 ที่พร้อมกัน	63
4.27	ฐานข้อมูลเมื่อมีผู้บุกรุกบ้านของ RICH	63
4.28	หน้าจอมอนิเตอร์เมื่อมีการโจรกรรมบ้านของ RICH	64
4.29	หน้าจอมอนิเตอร์หลังจากการกดเข้าไปดูข้อมูลภาพของเหตุการณ์	64
4.30	ฐานข้อมูลเมื่อมีการโจรกรรม 2 ที่พร้อมกัน	64
4.31	หน้าจอมอนิเตอร์เมื่อมีการโจรกรรม 2 ที่พร้อมกัน	65
4.32	ฐานข้อมูลเมื่อมีการโจรกรรมและเหตุการณ์ไฟไหม้พร้อมกัน	65
4.33	หน้าจอมอนิเตอร์เมื่อมีการโจรกรรมและเกิดเหตุไฟไหมพร้อมกัน	65
4.34	หน้าจอมอนิเตอร์เมื่อมีผู้ขอความช่วยเหลือ	66
4.35	หน้าต่างแสดงประวัติการเกิดเหตุการณ์ผิดปกติทั้งหมด	66
4.36	หน้าต่างแสดงประวัติของเหตุการณ์ไฟไหม้เพียงอย่างเดียว	67
4.37	หน้าต่างแสดงประวัติของเหตุการณ์โจรกรรมเพียงอย่างเดียว	67
4.38	หน้าต่างแสดงประวัติของเหตุการณ์ขอความช่วยเหลือเพียงอย่างเดียว	68
4.39	หน้าต่างแสดงการค้นหาจากบ้านเลขที่	68
4.40	หน้าต่างจัดการระบบ	69
4.41	ฐานข้อมูลที่ชื่อว่า SECURITY	70
4.42	ฐานข้อมูลที่มีชื่อว่า LOGFILE	71
4.43	ข้อความแจ้งเตือนผ่านแอปพลิเคชั่นไลน์เมื่อเกิดเหตุไฟไหม้	71
4.44	ข้อความแจ้งเตือนผ่านแอปพลิเคชั่นไลน์เมื่อเกิดเหตุโจรกรรม	72
4.45	ข้อความแจ้งเตือนผ่านแอปพลิเคชั่นไลน์เมื่อปัญหายุติแล้ว	72

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันที่พักอาศัยส่วนใหญ่มักจะอยู่รวมกันเป็นกลุ่ม เช่นหมู่บ้านจัดสรร เป็นต้น ปัญหาที่พบบ่อยคือปัญหาอัคคีภัยและปัญหาโจรกรรมทรัพย์สิน ซึ่งอาจทำให้เกิดความเสียหายต่อชีวิตและทรัพย์สินได้ กรณีที่เกิดอัคคีภัยอาจจะลุกลามเป็นวงกว้างไปยังบริเวณใกล้เคียง ซึ่งวิธีการแก้ไขปัญหาทั่วไปกรณีการเกิดอัคคีภัยผู้พบเห็นเหตุการณ์เพลิงไหม้ จะรับรู้ได้โดยการสังเกต และสัมผัสได้ถึงกลิ่นควัน ณ สถานการณ์นั้นเปลวเพลิงอาจจะลุกลามไปมากเกินกว่าที่จะควบคุมสถานการณ์ได้ และกรณีของการโจรกรรมจะรับรู้ได้ก็ต่อเมื่อทรัพย์สินได้สูญหายไปแล้ว

จึงได้มีการออกแบบระบบดูแลความปลอดภัยสำหรับหมู่บ้านจัดสรร ซึ่งใช้อุปกรณ์ตรวจจับบุคคลและอุปกรณ์ตรวจจับเปลวไฟพร้อมทั้งระบบกล้องจับภาพ ติดตั้งไว้ภายในบ้านแต่ละบ้าน และให้อุปกรณ์ทุกตัว ส่งข้อมูลความผิดปกติที่ตรวจพบไปยังศูนย์ควบคุมส่วนกลาง เพื่อให้เจ้าหน้าที่รักษาความปลอดภัยใช้เฝ้าระวังผ่านโครงข่ายอินเตอร์เน็ต และแจ้งเตือนบุคคลบริเวณใกล้ที่เกิดเหตุโดยใช้สัญญาณเสียงจากบัซเซอร์ (Buzzer) เพื่อจะได้แก้ไขสถานการณ์ได้ทันเวลาและลดความสูญเสียให้เกิดขึ้นน้อยที่สุด

1.2 วัตถุประสงค์

1) เพื่อศึกษาและสร้างระบบรักษาความปลอดภัยภายในหมู่บ้านจัดสรร
2) เพื่อให้ระบบสามารถแก้ปัญหาอัคคีภัยและการโจรกรรมในหมู่บ้านได้อย่างทันท่วงที
3) เพื่อลดความเสียหายที่จะเกิดขึ้นกับชีวิตและทรัพย์สิน

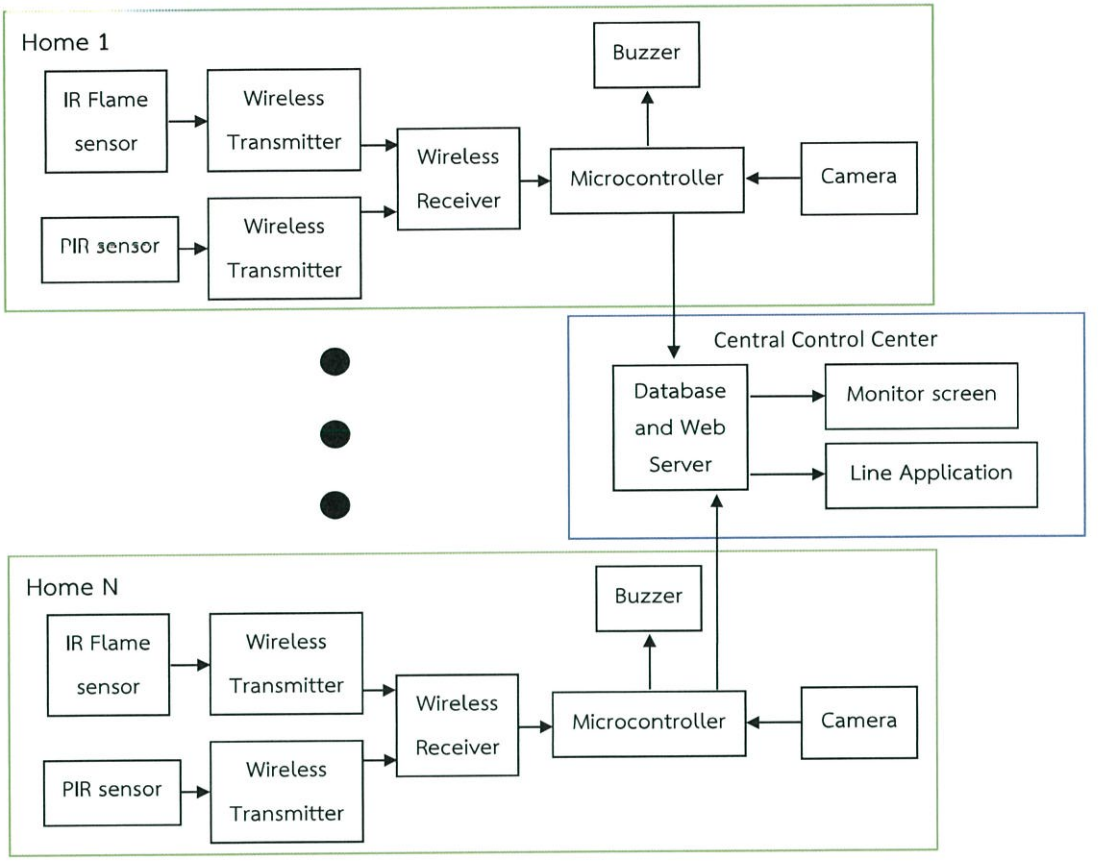
1.3 ขอบเขตของปริญญานิพนธ์

1) ออกแบบระบบตรวจจับผู้บุกรุกและตรวจจับอัคคีภัยในบ้าน แสดงผลผ่านเครือข่ายอินเตอร์เน็ต

2) ออกแบบระบบการมอนิเตอร์ (Monitoring) เพื่อให้เจ้าหน้าที่รักษาความปลอดภัยใช้เฝ้าระวังเหตุอัคคีภัยและการโจรกรรมภายในหมู่บ้านจัดสรร

3) สร้างการแจ้งเตือนบริเวณที่เกิดเหตุโดยใช้สัญญาณเสียงจากบัซเซอร์

บล็อกไดอะแกรมของโครงงานที่นำเสนอ



รูปที่ 1.1 บล็อกไดอะแกรมของโครงงาน

ระบบรักษาความปลอดภัยในหมู่บ้านจัดสรร บล็อกไดอาแกรมเป็นไปตามรูปที่ 1.1 ประกอบด้วยเซนเซอร์ 2 ชนิดคือเซนเซอร์ตรวจจับเปลวไฟ (IR flame sensor) ใช้ในการตรวจจับเปลวไฟที่เกิดขึ้นและเซนเซอร์ตรวจจับการเคลื่อนไหว (PIR sensor) ใช้ในการตรวจจับผู้บุกรุก ซึ่งเซนเซอร์หนึ่งตัวจะเชื่อมต่อกับไมโครคอนโทรลเลอร์ (Arduino Nano) และโมดูลสื่อสารไร้สาย (NRF24L01 Module) ทั้ง 3 ส่วนนี้จะเรียกรวมเป็นหนึ่งโหนด เมื่อเซนเซอร์ตรวจจับเปลวไฟหรือผู้บุกรุกได้ เซนเซอร์จะส่งสัญญาณแรงดันไฟฟ้าไปที่อาดูโน่นาโน ข้อมูลจะถูกส่งจากอาดูโน่นาโนแบบไร้สายไปยังราสป์เบอร์รีไพ ราสป์เบอร์รีไพจะจ่ายแรงดันไฟฟ้าไปที่พอร์ตที่ต่อกับบัซเซอร์ เพื่อแจ้งเตือนเพื่อนบ้านในบริเวณใกลเคียงว่ามีอันตรายเกิดขึ้น และมีข้อความแจ้งเตือนเจ้าของบ้านผ่านแอพพลิเคชั่นไลน์ (LINE Application) นอกจากนี้ราสป์เบอร์รีไพจะส่งข้อมูลความผิดปกติและข้อมูลภาพไปยังศูนย์ควบคุมส่วนกลางผ่านเครือข่ายอินเตอร์เน็ตเพื่อให้เจ้าหน้าที่รักษาความปลอดภัยประจำหมู่บ้านรับรู้ถึงเหตุการณ์ที่เกิด และสามารถเข้าไปแก้ไขปัญหาที่เกิดขึ้นได้ทันท่วงที

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 ไมโครคอนโทรลเลอร์ (Microcontroller)

ไมโครคอนโทรลเลอร์มีการทำงานคล้ายคอมพิวเตอร์แต่ขนาดเล็กกว่าและรวบรวมวงจรต่างๆ ไว้ในชิปเดียว ภายในชิปจะประกอบไปด้วย

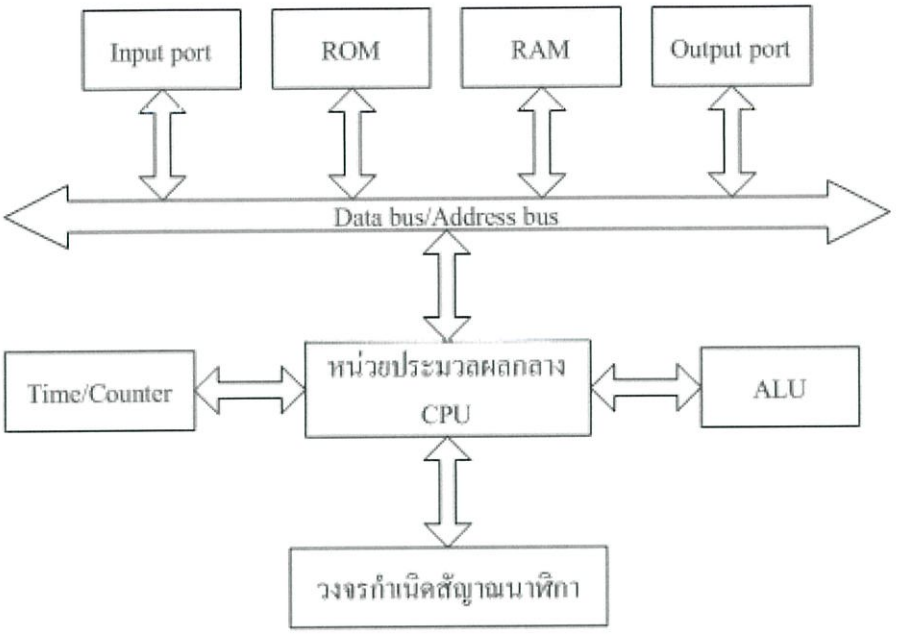
1. หน่วยประมวลผลกลาง (Central Processing Unit : CPU) เปรียบเสมือนสมองของคอมพิวเตอร์ในการทำหน้าที่ตัดสินใจหรือคำนวณ จากคำสั่งที่ได้รับมา ถือเป็นอุปกรณ์หลักในการประมวลผลของคอมพิวเตอร์ โดยพื้นฐานแล้วหน่วยประมวลผลกลางทำหน้าที่ประมวลผลข้อมูลทางคณิตศาสตร์และข้อมูลเชิงตรรกะ

2. หน่วยความจำ (Memory) คือส่วนที่เก็บข้อมูลต่างๆ ไว้ซึ่งแบ่งเป็น 2 ส่วน ส่วนแรกคือหน่วยความจำที่ไม่จำเป็นต้องมีไฟเลี้ยง จะใช้เป็นส่วนของการเก็บโปรแกรมหลัก (Program Memory) ส่วนที่สองคือหน่วยความจำข้อมูล (Data Memory) เป็นหน่วยความจำที่จำเป็นต้องมีไฟเลี้ยงถึงสามารถเก็บข้อมูลได้ เมื่อไม่มีไฟเลี้ยงข้อมูลก็จะหายทำหน้าที่เป็นที่พักข้อมูลชั่วคราวของหน่วยประมวลผลกลางตัวอย่างเช่น RAM

3. ส่วนติดต่อกับอุปกรณ์ภายนอกหรือพอร์ตอินพุต/เอาต์พุต ใช้สำหรับรับข้อมูลจากอุปกรณ์ภายนอกหรือส่งข้อมูลไปยังอุปกรณ์ภายนอก

4. ช่องทางเดินของสัญญาณหรือบัส (BUS) คือเส้นทางการแลกเปลี่ยนสัญญาณข้อมูลระหว่าง หน่วยประมวลผลกลาง หน่วยความจำและพอร์ต เป็นลักษณะของสายสัญญาณจำนวนมากอยู่ภายในตัวไมโครคอนโทรลเลอร์ โดยแบ่งเป็นบัสข้อมูล (Data Bus), บัสแอดเดรส (Address Bus) และบัสควบคุม (Control Bus)

5. วงจรกำเนิดสัญญาณเป็นส่วนที่ควบคุมจังหวะการทำงานของไมโครคอนโทรลเลอร์ ซึ่งแปลความหมายแบบตรงตัวคือ ระบบควบคุมขนาดเล็กที่สามารถนำมาประยุกต์ใช้งานได้หลากหลาย โดยผ่านการออกแบบวงจรให้เหมาะกับงานต่างๆ และยังสามารถเขียนโปรแกรมเพื่อควบคุมพอร์ตอินพุต/เอาต์พุต เพื่อสั่งงานให้ไปควบคุมอุปกรณ์ต่างๆ ได้อีกด้วย ซึ่งก็นับว่าเป็นระบบที่สามารถนำมาประยุกต์ใช้งานได้หลากหลาย ทั้งทางด้านดิจิตอลและแอนาล็อก ยกตัวอย่างเช่น การควบคุมเวลาเปิดปิดของพัดลม, ระบบแจ้งเตือนไฟไหม้ในอาคารและอื่นๆ ยิ่งไปกว่านั้นระบบไมโครคอนโทรลเลอร์ในยุคปัจจุบันสามารถทำการเชื่อมต่อกับเครือข่ายอินเตอร์เน็ตได้ ดังนั้นการสั่งงานจึงไม่ใช่แค่หน้าแผงวงจร แต่อาจจะเป็นการสั่งงานทางไกลผ่านเครือข่ายอินเตอร์เน็ต แสดงการทำงานในรูปที่ 2.1

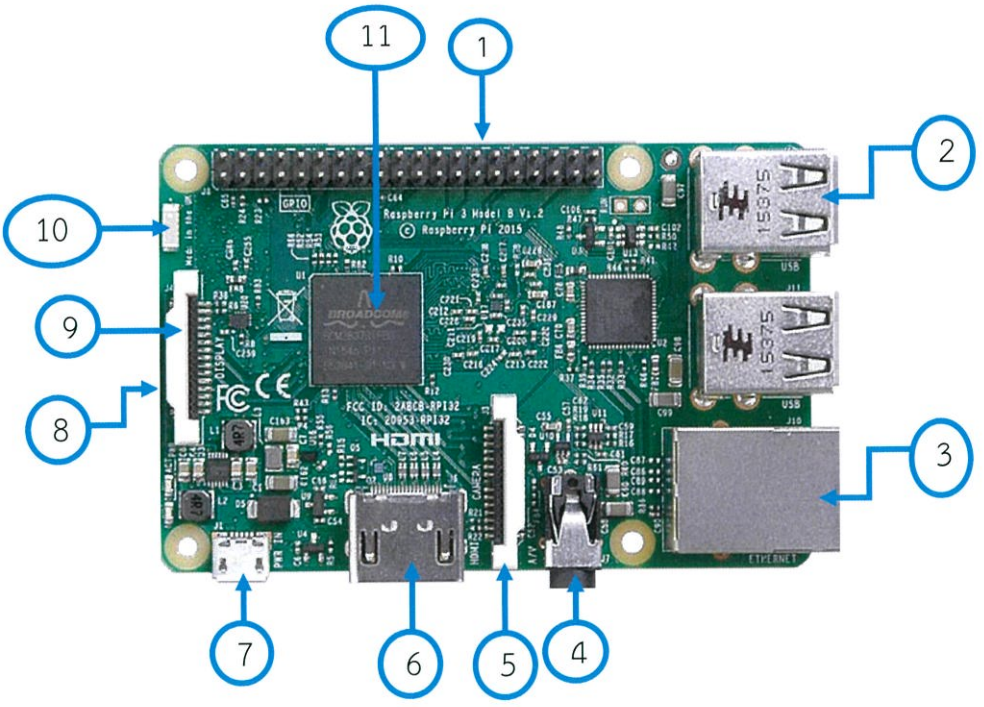


รูปที่ 2.1 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์

2.1.1 ราสป์เบอร์รีไพ (Raspberry pi)

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

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



รูปที่ 2.2 ราสป์เบอร์รีไพ 3 โมเดลบี

พอร์ตต่างๆ ของราสป์เบอร์รีไพมี ดังนี้

1. พอร์ต GPIO
2. พอร์ต USB 2.0 จำนวน 2 พอร์ต
3. พอร์ต RJ-45 Ethernet LAN 10/100Mbps
4. จุดเชื่อมต่อสัญญาณเสียงขนาด 3.5 มิลลิเมตร
5. พอร์ต CSI (Camera Serial Interface) สำหรับเชื่อมต่อโมดูลกล้อง
6. พอร์ต HDMI สำหรับเชื่อมต่อสัญญาณภาพและเสียง
7. พอร์ต Micro USB Power สำหรับเป็นไฟเลี้ยงวงจรบอร์ดราสป์เบอร์รีไพ
8. ช่องเสียบ SD Card อยู่บริเวณด้านล่างของบอร์ด
9. พอร์ต DSI (Display Serial Interface) ใช้สำหรับต่อจอแสดงผล
10. สายอากาศของการเชื่อมต่อไร้สาย Bluetooth และ Wi-Fi
11. ชิพ Broadcom BCM2837 64bit Quad core CPU at 1.2GHz, 1GB RAM

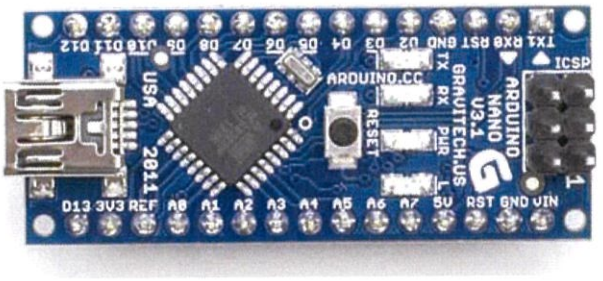
ขาต่างๆ ของราสป์เบอร์รีไพมีดังนี้

Peripherals	GPIO	Particle	Pin #			Pin #	Particle	GPIO	Peripherals
3.3V			1	X	X	2	5V		
I2C	GPIO2	SDA	3	X	X	4	5V		
	GPIO3	SCL	5	X	X	6	GND		
Digital I/O	GPIO4	D0	7	X	X	8	TX	GPIO14	UART
GND			9	X	X	10	RX	GPIO15	Serial 1
Digital I/O	GPIO17	D1	11	X	X	12	D9/A0	GPIO18	PWM 1
Digital I/O	GPIO27	D2	13	X	X	14	GND		
Digital I/O	GPIO22	D3	15	X	X	16	D10/A1	GPIO23	Digital I/O
3.3V			17	X	X	18	D11/A2	GPIO24	Digital I/O
SPI	GPIO10	MOSI	19	X	X	20	GND		
	GPIO9	MISO	21	X	X	22	D12/A3	GPIO25	Digital I/O
	GPIO11	SCK	23	X	X	24	CE0	GPIO8	SPI (chip enable)
GND			25	X	X	26	CE1	GPIO7	
DO NOT USE	ID_SD	DO NOT USE	27	X	X	28	DO NOT USE	ID_SC	DO NOT USE
Digital I/O	GPIO5	D4	29	X	X	30	GND		
Digital I/O	GPIO6	D5	31	X	X	32	D13/A4	GPIO12	Digital I/O
PWM 2	GPIO13	D6	33	X	X	34	GND		
PWM 2	GPIO19	D7	35	X	X	36	D14/A5	GPIO16	PWM 1
Digital I/O	GPIO26	D8	37	X	X	38	D15/A6	GPIO20	Digital I/O
GND			39	X	X	40	D16/A7	GPIO21	Digital I/O

รูปที่ 2.3 คุณสมบัติของขา GPIO

มีทั้งหมด 40 ขา โดยแบ่งเป็น GPIO ทั้งหมด 21 ขา ซึ่งสามารถเป็นขาที่ใช้สื่อสารข้อมูลแบบ SPI 4 ขา , ขา I2C 2 ขา , ขา PWM 4 ขา , UART 1ขา และ Serial 1ขา , ไฟเลี้ยง 5 โวลต์ 2 ขา , ไฟเลี้ยง 3.3 โวลต์ 2 ขา , กราวด์ 7 ขา

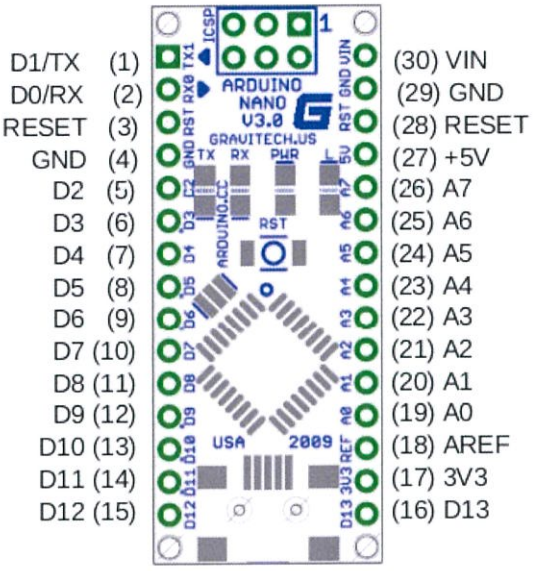
2.1.3 อาดูโนนาโน (Arduino Nano)



รูปที่ 2.4 Arduino Nano 3.0

อาดูโนนาโน (Arduino Nano) ดังรูปที่ 2.4 เป็นไมโครคอนโทรลเลอร์ที่ใช้ชิปประมวลผล ATmega328 การใช้งานทำได้เหมือนกัน แต่ต่างกันตรงที่ออกแบบให้มีขนาดเล็ก

กว่าเดิม และตัดบางฟังก์ชั่นที่ไม่จำเป็นออกไป ใช้ Mini USB เชื่อมต่อกับคอมพิวเตอร์ มีพอร์ตดิจิตอลอินพุตเอาต์พุต 14 พอร์ต มีพอร์ตอนาล็อกอินพุต 8 พอร์ต ดังรูปที่ 2.5



รูปที่ 2.5 ขาต่างๆของ Arduino Nano

2.1.4 คุณสมบัติของอาดูโนนาโน

1. ใช้แรงดันไฟฟ้า 5 โวลต์
2. การจ่ายแรงดันไฟฟ้าที่แนะนำ 7 - 12 โวลต์ จำกัดที่ 6 – 20 โวลต์
3. หน่วยความจำภายใน 32000 ไบต์
4. หน่วยความจำถาวร (EEPROM) 1000 ไบต์
5. หัวเชื่อมต่อเป็นแบบ Mini-USB
6. ความถี่คริสตัล 16 MHz
7. มีขาดิจิตอล I/O 14 ขา, ขาแอนะล็อก 8 ขา

2.2 เซนเซอร์ตรวจจับการเคลื่อนไหว (Passive Infrared Sensor module)

รูปที่ 2.6 เซนเซอร์ตรวจจับการเคลื่อนไหว

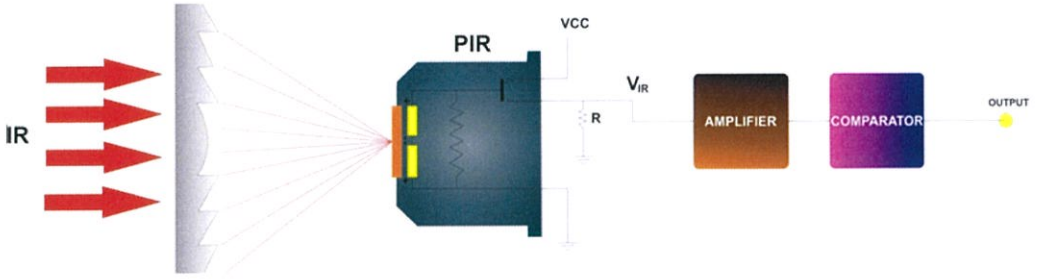
เซนเซอร์ตรวจจับการเคลื่อนไหว (Passive Infrared Sensor) หรือ PIR sensor คือ อุปกรณ์ที่ใช้สำหรับตรวจจับการมีอยู่ของมนุษย์หรือสัตว์โดยใช้หลักการการตรวจจับรังสีอินฟราเรด จากวัตถุผ่านยังเลนส์เพื่อรวมแสงแล้วโฟกัสมายังอุปกรณ์ตรวจจับความเคลื่อนไหว ซึ่งเป็นตัวแปลง พลังงานความร้อนจากรังสีอินฟราเรดเป็นพลังงานไฟฟ้า ดังรูปที่ 2.6

2.2.1 คุณสมบัติของเซนเซอร์ตรวจจับการเคลื่อนไหว

1. ชื่อรุ่น PIR HC-SR501
2. ขนาด 3.2 เซนติเมตร × 2.4 เซนติเมตร × 1.8 เซนติเมตร
3. ค่าหน่วยเวลา 2 - 200 วินาที
4. แหล่งจ่ายไฟ 3 – 5 โวลต์
5. กินกระแสน้อยกว่า 60 ไมโครแอมป์
6. แรงดันเอาต์พุตในขณะที่ตรวจจับได้ 3.3 โวลต์ และตรวจจับไม่ได้จะเป็น 0 โวลต์
7. มุมในการตรวจจับสูงสุด 140°
8. อุณหภูมิที่สามารถทำงานได้ -20 ถึง 80 °C
9. สามารถปรับค่า sensitivity และ ค่าหน่วงเวลาได้
10. ระยะตรวจจับสูงสุด 7 เมตร

2.2.2 หลักการทำงานของเซนเซอร์ตรวจจับการเคลื่อนไหว

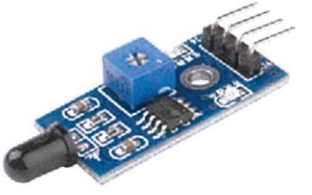
เมื่อมีสิ่งมีชีวิตที่มีรังสีอินฟราเรด (Infrared) เคลื่อนผ่านเข้ามารังสีอินฟราเรดก็จะถูกรวมโดยเลนส์ แล้วโฟกัสไปยังไพโรอิเล็กตริก (pyroelectric) จากนั้นจะเปลี่ยนรังสีอินฟราเรดให้เป็นแรงดันไฟฟ้า แม้ว่าจะมีรังสีอินฟราเรดเพียงเล็กน้อยก็สามารถตรวจจับได้เพราะเลนส์ทำหน้าที่รวบรวมให้รังสีเข้มขึ้น



รูปที่ 2.7 วงจรภายใน PIR sensor module

เมื่อไพโรอิเล็กตริกส่งแรงดันออกมาจะผ่านวงจรขยายและเปรียบเทียบ จากนั้นแรงดันไฟฟ้าที่โหนดเอาต์พุตก็จะนำไปผ่านวงจรต่างๆ เพื่อจัดเรียงแรงดันให้สามารถใช้งานกับไมโครคอนโทรลเลอร์ได้อย่างสะดวกมากขึ้น ดังรูปที่ 2.7

2.3 เซนเซอร์ตรวจจับเปลวไฟ (Infrared Flame Detector Sensor Module)



รูปที่ 2.8 เซนเซอร์ตรวจจับเปลวไฟ

เซนเซอร์ตรวจจับเปลวไฟ (Infrared Flame Detector sensor Module) เป็นโมดูลเซ็นเซอร์ที่ใช้ในการตรวจจับรังสีรังสีอินฟราเรดที่เกิดจากแหล่งกำเนิดความร้อนต่างๆเช่นเปลวเพลิง สามารถให้เอาต์พุตทั้งเป็นแบบแอนะล็อกแรงดันเอาต์พุต 0 - 5 โวลต์ และแบบดิจิตอลถ้าไม่มีการตรวจจับเปลวไฟจะให้แรงดัน 0 โวลต์ แต่ถ้ามีการตรวจจับเปลวเพลิงก็จะจ่ายแรงดัน 5 โวลต์ สามารถปรับค่า Threshold ได้โดยการปรับ potentiometer บนโมดูล ดังรูปที่ 2.8

2.3.1 คุณสมบัติของ Infrared Flame Detector Sensor Module

1. สามารถตรวจจับรังสีอินฟราเรด ที่ความยาวคลื่น 760 - 1100 นาโนเมตร
2. แรงดันทำงาน 3.3 - 5 โวลต์
3. ขนาดของบอร์ด 1.4 - 3.2 เซนติเมตร
4. สามารถปรับค่า sensitivity ได้
5. มุมในการตรวจจับเปลวไฟ 60°
6. สามารถใช้งานได้ทั้งแบบแอนะล็อกและดิจิตอล

2.3.2 การใช้งานของเซนเซอร์

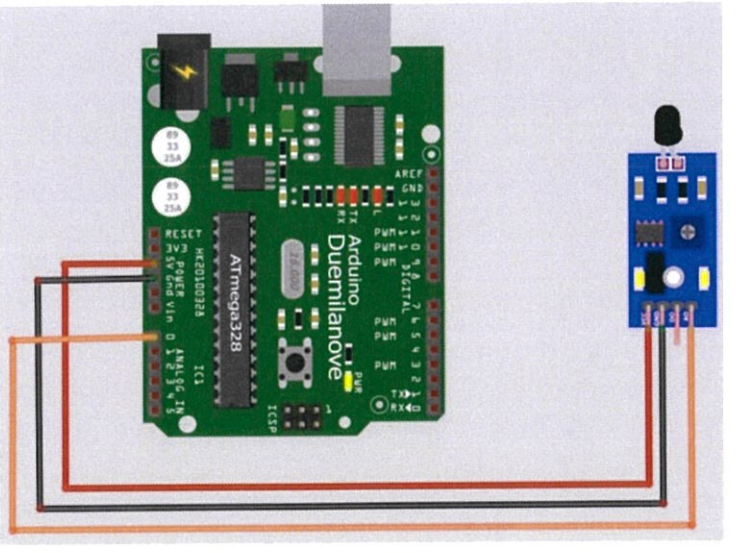
1. เซนเซอร์มีความไวสูงต่อการตรวจจับเปลวไฟ สามารถตรวจจับเปลวไฟได้ดี เหมาะต่อการใช้เป็นเซนเซอร์ตรวจจับไฟไหม้ เพื่อแจ้งเตือน

2. สามารถเชื่อมต่อโดยตรงไปยังพอร์ตไมโครคอนโทรลเลอร์

3. การติดตั้งเซนเซอร์ควรมีระยะห่างจากความร้อน หรือเปลวไฟ ประมาณ 80 เซนติเมตร เพื่อที่จะไม่ให้เกิดความเสียหายต่อเซนเซอร์

4. การแสดงผลในรูปแบบแอนะล็อก สามารถผ่านขั้นตอนการแปลงสัญญาณจากแอนะล็อกเป็นดิจิตอล (AD) ได้โดยไม่มีการสูญเสีย มีความถูกต้องและแม่นยำสูง

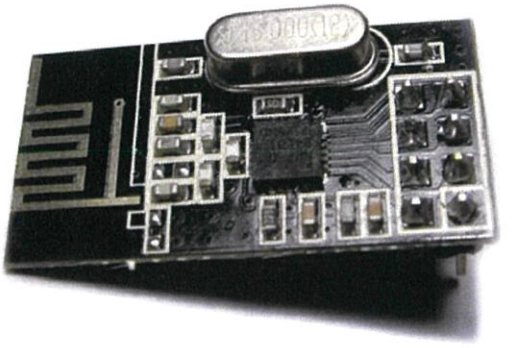
5. มีการติดตั้งและใช้งานง่าย ดังรูปที่ 2.9



รูปที่ 2.9 การเชื่อมต่อเซนเซอร์ตรวจจับเปลวไฟกับ Arduino

2.4 โมดูลสื่อสารไร้สาย NRF24L01 (NRF24L01 Module)

เป็นโมดูลสื่อสารชนิดไร้สาย (NRF24L01 Module) ซึ่งสามารถใช้เป็นได้ทั้งตัวรับและตัวส่งขึ้นอยู่กับการเขียนโปรแกรมบน Arduino IDE สามารถใช้งานกับไมโครคอนโทรลเลอร์ให้ติดต่อสื่อสารพร้อมกันหลายๆ ตัวได้ ระยะทำงานอยู่ที่ 15 – 500 เมตร ขึ้นอยู่กับความเร็วที่ใช้ในการส่ง ใช้พลังงานต่ำจึงสามารถใช้งานได้นาน นำมาประยุกต์ใช้ในการส่งข้อมูลสถานะที่ตรวจจับได้จากเซนเซอร์ตรวจจับการเคลื่อนไหวและตรวจจับเปลวไฟ ไปที่ศูนย์ควบคุมส่วนกลาง ดังรูปที่ 2.10

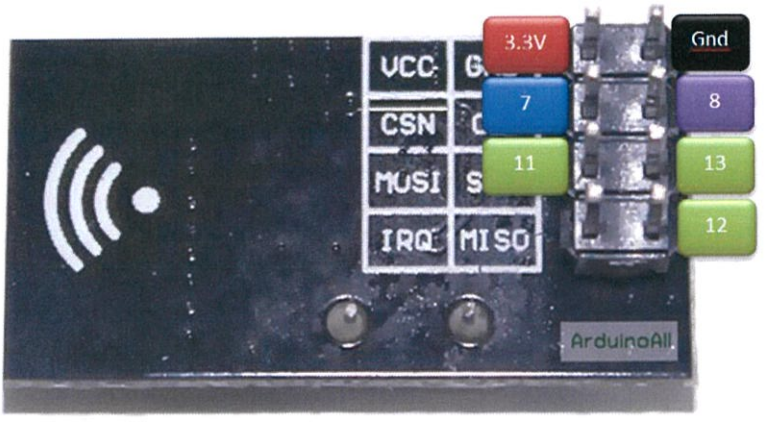


รูปที่ 2.10 โมดูลสื่อสารไร้สาย NRF24L01

2.4.1 การใช้งานของโมดูล NRF24L01

การต่อขาโมดูล NRF24L01 กับ Arduino เป็นดังรูปที่ 2.11

1. VCC ต่อกับ 3.3 โวลต์
2. GND ต่อกับ กราวด์
3. CSN ต่อกับ ขา 7
4. CE ต่อกับ ขา 8
5. MOSI ต่อกับ ขา 11
6. MISO ต่อกับ ขา 12
8. SCK ต่อกับ ขา 13



รูปที่ 2.11 การต่อขาโมดูลสื่อสารไร้สาย NRF24L01 กับ Arduino

2.4.2 การสื่อสารข้อมูลด้วยระบบ SPI Bus

SPI Bus (Serial Peripheral Interface Bus) เป็นหนึ่งในรูปแบบของการสื่อสารข้อมูลระหว่างอุปกรณ์แบบดิจิตอลที่เจอได้ทั่วไป ใช้สื่อสารระหว่างอุปกรณ์ได้มากกว่าสองอุปกรณ์ขึ้นไปซึ่งจะนำมาต่อกันเป็นบัส (Bus) บัส SPI จะรับและส่งข้อมูลทีละบิต (Bit Serial) และใช้สัญญาณนาฬิกา (Clock) กำหนดจังหวะการทำงาน การทำงานของอุปกรณ์ในระบบบัสจะแบ่งออกเป็น 2 แบบ นั่นคือ SPI Master และ SPI Slave ซึ่ง SPI Master จะเป็นด้านเริ่มการรับส่งข้อมูล โดยจะสร้างสัญญาณนาฬิกา เพื่อใช้เป็นตัวกำหนดจังหวะรับส่งของการสื่อสาร ส่วนฝั่ง SPI Slave จะทำหน้าที่ในการคอยตอบสนองฝั่ง SPI Slave ซึ่งการรับส่งแบบนี้ หนึ่ง SPI Master สามารถมีได้หลาย SPI Slave (Single-Master, Multi-Slave)

SPI นั้นจะใช้สายสัญญาณ 4 เส้น ได้แก่

1. SCK (Serial Clock) เป็นสัญญาณนาฬิกา ที่ถูกสร้างโดย SPI Master ใช้ในการกำหนดจังหวะรับส่งข้อมูลกับ SPI Slave

2. MOSI (Master-Out Slave-In) เป็นสัญญาณที่ใช้ในการส่งข้อมูลบิตจาก SPI Master ไป SPI Slave

3. MISO (Master-In Slave-Out) เป็นสัญญาณที่ใช้ในการส่งข้อมูลบิตจาก SPI Slave ไปยัง SPI Master

4. SS (Slave Select, Active-Low) เป็นสัญญาณที่สร้างโดย SPI Master เพื่อใช้ระบุว่า ต้องการสื่อสารกับ SPI Slave หรือไม่ในกรณีที่มีอุปกรณ์ SPI Slave มากกว่าหนึ่งชุด จะต้องมีสัญญาณ Slave Select มากกว่าหนึ่งเส้น และแยกสำหรับแต่ละอุปกรณ์

เมื่อเริ่มการรับส่งข้อมูล สัญญาณ Slave Select (SS) จะต้องเปลี่ยนจากสถานะ HIGH เป็น LOW จากนั้นข้อมูลขนาดหนึ่งไบต์แต่ละไบต์จะถูกส่งออกแบบเลื่อนไปทีละบิตจาก SPI Master ตามจังหวะสัญญาณนาฬิกาที่สร้างขึ้น สามารถเลือกได้ว่าจะส่ง MSB (Most-Significant Bit) หรือ LSB (Least-Significant Bit) ไปก่อน ซึ่งขณะเดียวกันนั้นก็จะรับข้อมูลจากฝั่ง SPI Slave ไปด้วย ซึ่งจะเป็นการแลกเปลี่ยนข้อมูลขนาดเท่าๆ กัน ก่อนที่ SS จะกลับมาเป็นสถานะ LOW อีกครั้ง อาจมีการรับส่งข้อมูลได้มากกว่า 1 ไบต์

2.5 บัซเซอร์ (Buzzer)

เป็นลำโพงแบบแม่เหล็กไฟฟ้า เสียงที่ได้เกิดจากการจ่ายแรงดันไฟฟ้ากระแสตรงขนาดแรงดัน 3 – 24 โวลต์ เข้าไปที่ตัวลำโพง วงจรกำเนิดความถี่ในตัวของลำโพงจะให้กำเนิดเสียง แต่เสียงจะมีความถี่เดียว ไม่สามารถเปลี่ยนความถี่ของเสียงได้ ตัวอย่างของ buzzer ดังแสดงในรูปที่ 2.12

รูปที่ 2.12 บัซเซอร์

2.6 การสร้างฐานข้อมูล

ฐานข้อมูล (Database) คือเครื่องมือที่ใช้เก็บรวบรวมกลุ่มของข้อมูล ที่มีความสัมพันธ์กันนำมารวมไว้ด้วยกันอย่างมีระเบียบ เพื่อง่ายต่อการใช้งานและลดความซับซ้อนของข้อมูล ข้อมูลที่นำมาเก็บ เช่น ข้อมูลเกี่ยวกับบุคคล สิ่งของ ราคา สถานที่ หรือเหตุการณ์สำคัญที่สนใจก็สามารถเก็บได้

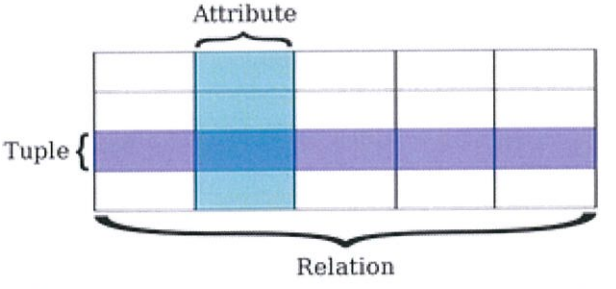
ระบบการจัดการฐานข้อมูล (Database Management System) คือกลุ่มของซอฟต์แวร์ที่ทำหน้าที่จัดการและบริหารฐานข้อมูลให้มีประสิทธิภาพในการใช้งานสูงที่สุด ทำให้ผู้ใช้งานเข้าถึงข้อมูลได้ง่ายและสะดวก ตัวอย่างของระบบจัดการฐานข้อมูลที่นิยมใช้ในปัจจุบันได้แก่ SQL Server, Oracle เป็นต้น ดังรูปที่ 2.13

2.6.1 โปรแกรมฐานข้อมูล MySQL

รูปที่ 2.13 โปรแกรม MySQL

MySQL เป็นระบบจัดการฐานข้อมูลแบบข้อมูลเชิงสัมพันธ์ (Relational Database Management System) เก็บข้อมูลเป็นตารางในตารางจะแบ่งเป็นแถว (Row) และคอลัมน์ (Column) มีการเชื่อมโยงข้อมูลในตารางโดยอ้างอิงจากแถวและคอลัมน์ที่กำหนดเอาไว้ มีภาษาที่ใช้เขียนเฉพาะคือ ภาษา SQL (Structures Query Language) เป็นซอฟแวร์ที่เปิดให้ใช้งานแบบโอเพนซอร์ส (open source) โดยมีลิขสิทธิ์การใช้งาน 2 แบบมีทั้งแบบผู้ดูแลระบบไม่เสียค่าใช้จ่ายใดๆ และแบบที่มีลิขสิทธิ์ทางการค้าของ MySQL AB ที่เป็นผู้ผลิต ดังรูปที่ 2.14

2.6.2 คุณสมบัติของโปรแกรม MySQL



รูปที่ 2.14 ระบบจัดการฐานข้อมูลแบบข้อมูลเชิงสัมพันธ์

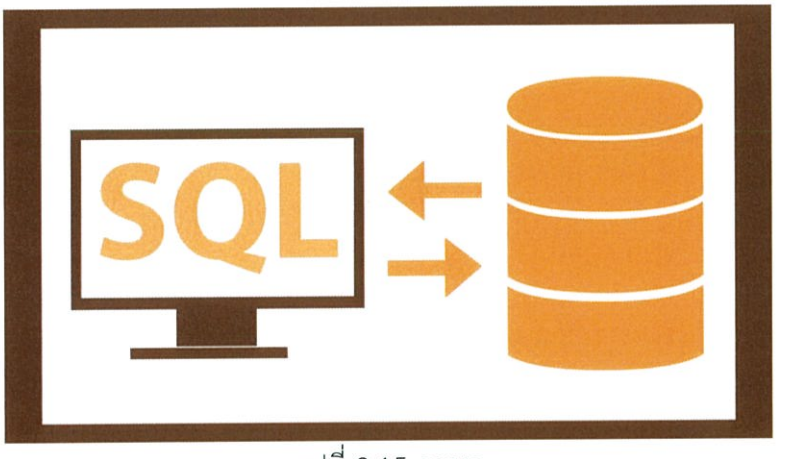
1. MySQL เป็นซอฟแวร์แบบโอเพนซอร์สใช้งานฟรีและมีประสิทธิภาพสูง
2. สามารถรองรับผู้ใช้งานได้จำนวนมากและได้รับการยอมรับความเร็วจากนักพัฒนาฐานข้อมูล

3. รองรับการใช้งานบนระบบปฏิบัติการมากมายเช่น UNIX, MAC OS, Window เป็นต้น

4 .สามารถใช้งานร่วมกับ platform ที่ใช้ในการพัฒนาเว็บเพจ (Web page) ได้อย่างหลากหลาย เช่น C, C++, Java, Perl, PHP, Python

5. ได้รับความนิยมและมีแนวโน้มที่สูงขึ้นเรื่อย ๆ

2.6.3 ภาษา SQL (Standard Query Language)



รูปที่ 2.15 ภาษา

เป็นภาษามาตรฐานที่ใช้จัดการฐานข้อมูลบนระบบฐานข้อมูลเชิงสัมพันธ์สามารถใช้คำสั่งของ SQL กับฐานข้อมูลชนิดใดก็ได้ โครงสร้างของภาษาไม่ซับซ้อนและเข้าใจง่าย ทำงานได้ซับซ้อนโดยใช้เพียงไม่กี่คำสั่ง ดังรูปที่ 2.15 สามารถแบ่งการทำงานได้ 4 ประเภท

1. Select query ใช้ดึงข้อมูลที่ต้องการ

```
SELECT column1, column2, ...
FROM table_name;
```

2. Update query ใช้สำหรับแก้ไขข้อมูล

```
UPDATE table_name
SET column1 = value1, column2 = value2, ...
WHERE condition;
```

3. Insert query ใช้สำหรับการเพิ่มข้อมูล

```
INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);
```

4. Delete query ใช้สำหรับลบข้อมูล

```
DELETE FROM table_name
WHERE condition;
```

2.6.4 ประเภทของคำสั่ง SQL

1. ภาษานิยามข้อมูล (Data Definition Language: DDL) เป็นกลุ่มคำสั่งที่ใช้สร้างฐานข้อมูล กำหนดโครงสร้างข้อมูล กำหนด Attribute ของข้อมูล

2. ภาษาจัดการข้อมูล (Data Manipulation Language: DML) เป็นกลุ่มของคำสั่งที่ใช้ในการเพิ่ม ลบและเปลี่ยนแปลงข้อมูลในตาราง

3. ภาษาควบคุมข้อมูล (Data Control Language: DCL) เป็นกลุ่มคำสั่งที่กำหนดสิทธิในการเข้าถึงฐานข้อมูลเพื่อให้ฐานข้อมูลมีความปลอดภัย

2.7 การสร้างหน้าเว็บเพจแสดงผลการแจ้งเตือน

2.7.1 ภาษา HTML

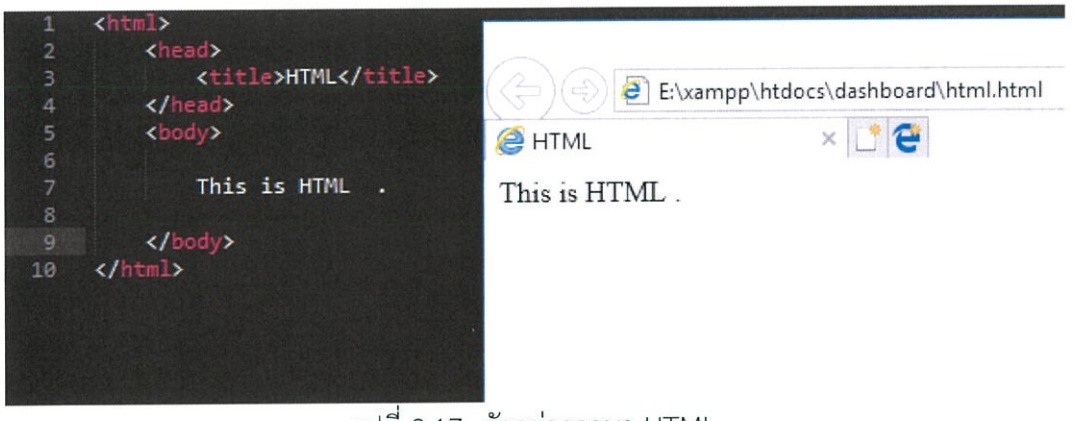
รูปที่ 2.16 ภาษา HTML

Hypertext Markup Language (HTML) เป็นภาษาคอมพิวเตอร์ประเภท Markup ที่ถูกออกแบบมาเพื่อใช้ในการสร้างเว็บเพจ (Web page) หรือที่เรียกกันว่าเว็บไซต์ (Web site) โครงสร้างของ HTML จะใช้ Tag ในการกำหนดการแสดงผลต่างๆ บนเว็บเพจเช่น ข้อความ ภาพนิ่ง ภาพเคลื่อนไหว เสียงและอื่นๆ ที่สามารถแสดงผลผ่าน Google Chrome, Mozilla, Firefox,

Safari, Opera, IE Microsoft Internet ซึ่งแอพพลิเคชั่นเหล่านี้เรียกว่าเว็บเบราว์เซอร์ (Web Browser) และยังสามารถเชื่อมโยงระหว่างเว็บเพจที่สร้างกับเว็บเพจอื่น ๆ ดังรูปที่ 2.16

HTML ได้ถูกพัฒนามาเรื่อย ๆตั้งแต่เวอร์ชั่นแรกจนมาถึงเวอร์ชั่นปัจจุบัน HTML 5 ได้รับการยอมรับและถูกจัดให้เป็นมาตรฐานแล้วโดยองค์กร World Wide Web consortium (W3C)

ในการสร้างเว็บเพจโดยใช้ภาษา HTML สามารถทำได้โดยการใช้ Text Editor หรืออาศัยเครื่องมือช่วยสร้างเช่น Dreamweaver, Microsoft Frontpage เพื่อช่วยให้สะดวกในการเขียนและออกแบบ ดังรูปที่ 2.17



รูปที่ 2.17 ตัวอย่างภาษา HTML

2.7.2 ภาษา PHP

รูปที่ 2.18 ภาษา PHP

ภาษาพีเอชพีคำว่า PHP มาจาก Hypertext Preprocessor หรือชื่อเดิม Personal Home Page เป็นภาษาคอมพิวเตอร์ประเภท scripting language คำสั่งต่างๆ จะเก็บอยู่ในไฟล์ มีรากฐานโครงสร้างคำสั่งมาจากภาษาซี ภาษาจาวา PHP ได้รับการพัฒนาและออกแบบมาให้ที่ใช้

ร่วมกับภาษา HTML โดยสามารถเพิ่มหรือแก้ไขเนื้อหาได้โดยอัตโนมัติซึ่งจะไม่แสดงคำสั่งที่เขียนทำให้แตกต่างจากภาษาอื่นที่มีลักษณะเป็น ไคลแอนต์-ไซด์ (client-site) สคริปต์เช่น ภาษาจาวาสคริปต์ที่ผู้เข้าชมเว็บไซต์สามารถคัดลอกคำสั่งไปใช้ได้ ดังรูปที่ 2.18

2.7.3 ลักษณะเด่นของภาษา PHP

ตัวอย่างของภาษา PHP ดังรูปที่ 2.19 มีลักษณะเด่น ดังนี้

1. เปิดให้ใช้งานแบบโอเพนซอร์ส ผู้ใช้สามารถนำเอาโค้ดของ PHP ไปใช้ได้โดยไม่เสียค่าใช้จ่าย
2. เป็นโปรแกรมที่ฝังอยู่ที่เซิร์ฟเวอร์จึงไม่มีผลต่อการทำงานของ client
3. สามารถใช้ได้บนระบบปฏิบัติการหลายชนิด เช่น UNIX ,Linux ,Window, Mac OS
4. สามารถทำงานได้ในเว็บเซิร์ฟเวอร์หลายชนิด เช่น Personal Web Server, Apache, Internet Information Service (IIS) เป็นต้น
5. สนับสนุนการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming)
6. สามารถใช้งานร่วมกับ XML ได้ทันที
7. สามารถใช้งานกันระบบฐานข้อมูลที่หลากหลายเช่น MySQL, FilePro, Solid ,Oracle
8. สามารถใช้งานกับข้อมูลตัวอักษรและการประมวลผลภาพได้อย่างมีประสิทธิภาพ
9. สามารถใช้งานร่วมกับโครงสร้างข้อมูลแบบ Scalar, Array, Associative array

```php
<?php
   $host = "localhost";
   $username = "root";
   $password = "";
   $dbName     = "data";
   $objconncet= mysqli_connect($host,$username,$password,$dbName) or die("Error Connect to Database");
   mysqli_select_db($objconncet,"security");
   mysqli_set_charset($objconncet, 'utf8');
   $strSQL = "SELECT * FROM security WHERE fire='1' OR thief='1';";

   $objQuery = mysqli_query($objconncet,$strSQL) or die ("Error Query [".$strSQL."]");

?>
```

รูปที่ 2.19 ตัวอย่างภาษา PHP

2.7.4 หลักการทำงานของ PHP

หลักการทำงานของ PHP มีดังนี้

1. ฝั่งไคลแอนต์ (Clicnt) จะทำการร้องขอหรือเรียกใช้งานไฟล์ ที่เก็บอยู่ในเครื่องเซิร์ฟเวอร์ (Server)

2. ฝั่งเซิร์ฟเวอร์จะทำการค้นหาไฟล์ PHP แล้วทำการประมวลผลไฟล์ PHP ตามที่ไคลแอนต์ทำการร้องขอมา

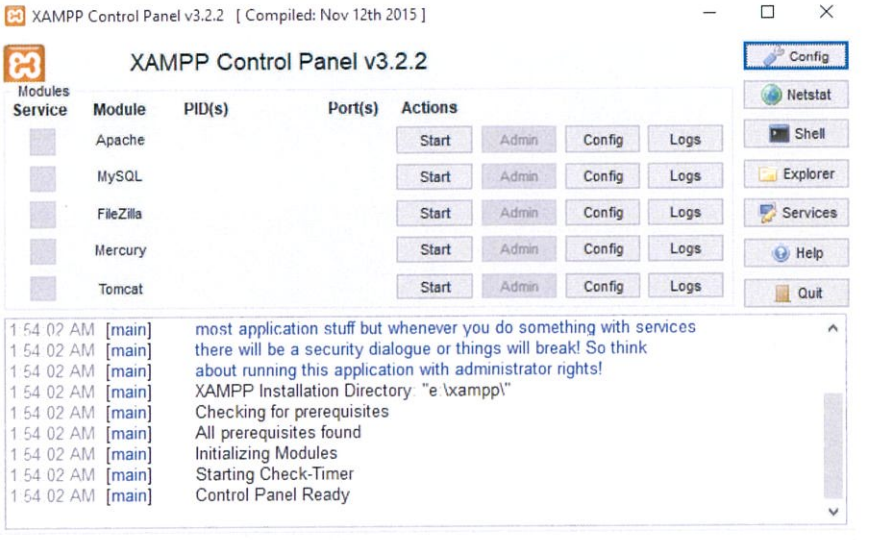
3. เป็นการติดต่อกับฐานข้อมูลและนำข้อมูลในฐานข้อมูลมาใช้ร่วมกับการประมวลผลส่งผลลัพธ์สุดท้ายจากการประมวลผลไปให้เครื่องไคลแอนต์

2.8 โปรแกรม XAMPP

รูปที่ 2.20 โปรแกรม XAMPP

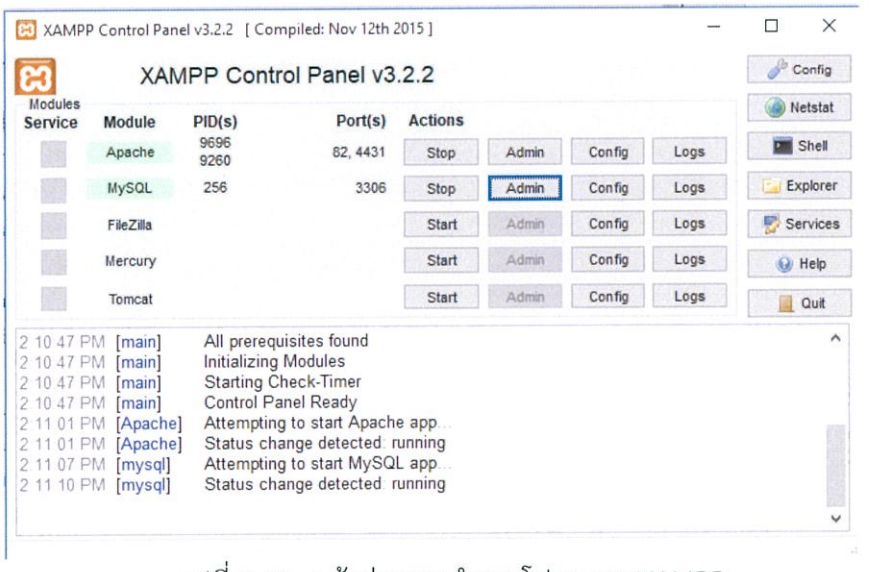
Xampp คือโปรแกรมใช้สำหรับจำลองให้เครื่องคอมพิวเตอร์ส่วนบุคคล ให้ทำงานในลักษณะของ Webserver คือเครื่องคอมพิวเตอร์จะเป็นทั้งเซิร์ฟเวอร์ที่ให้บริการเว็บเพจ และ ไคลแอนต์ผู้ร้องขอบริการในเครื่องเดียวกัน ทำให้ไม่จำเป็นต้องเชื่อมต่อกับเครือข่ายอินเตอร์เน็ต ก็สามารถทดสอบเว็บเพจที่สร้างขึ้น ได้ทุกเวลา ดังรูปที่ 2.20

Xampp ประกอบด้วย Apache, PHP, MySQL, PHP MyAdmin, Perl ซึ่งเป็นโปรแกรมพื้นฐานที่รองรับการทำงาน CMS ซึ่งเป็นชุดโปรแกรม สำหรับออกแบบเว็บไซต์ที่ได้รับความนิยมใน XAMPP นั้นรองรับระบบปฏิบัติการหลายตัว เช่น Windows, Linux, Apple ทำงานได้ทั้งบนระบบปฏิบัติการแบบ 32 บิต และ 64 บิต ดังรูปที่ 2.21 มีความโดดเด่นกว่าโปรแกรมอื่นคือมีตัวช่วยติดตั้ง CMS ที่เรียกว่า BitNami ซึ่งช่วยให้สามารถติดตั้ง CMS รุ่นใหม่ ๆ ได้



รูปที่ 2.21 หน้าต่างเริ่มต้นโปรแกรม XAMPP

จากหน้าต่างโปรแกรม XAMPP Control Panel ให้คลิกปุ่ม Start ของโปรแกรม Apache ต่อด้วยการสั่งให้โปรแกรมฐานข้อมูลทำงาน โดยคลิกปุ่ม Start ของ MySQL ดังรูปที่ 2.22



รูปที่ 2.22 หน้าต่างการทำงานโปรแกรม XAMPP

การยุติการทำงานของโปรแกรมก็ทำได้โดยการคลิกปุ่ม Stop การเข้าสู่หน้าเว็บทำได้โดยคลิกปุ่ม Admin และการปรับแต่งระบบทำได้โดยคลิกปุ่ม Config

ข้อจำกัด

1. เครื่องคอมพิวเตอร์ควรมี RAM ไม่ต่ำกว่า 128 MB
2. Hard disk ควรมีพื้นที่มากกว่า 320 MB

2.9 โปรแกรม Apache

โปรแกรม Apache เป็นโปรแกรมที่ใช้ในการทำเครื่องคอมพิวเตอร์นั้นๆ ให้เป็นเว็บเซิร์ฟเวอร์ (Web server) โดยการเปิดพอร์ตรอรับการเชื่อมต่อจากเว็บไคลแอนต์ (Web client) และทำหน้าที่จัดเก็บโฮมเพจ (Homepage) และส่งโฮมเพจไปยังเว็บบราวเซอร์ที่มีการร้องขอโฮมเพจจากเว็บเซิร์ฟเวอร์ ซึ่ง Apache นั้นเป็นซอฟแวร์แบบโอเพนซอร์ส ที่เปิดให้บุคคลภายนอกสามารถพัฒนาส่วนต่างๆได้ทำให้สามารถทำงานร่วมกับภาษาอื่นๆได้มากมาย นอกจากนั้นโปรแกรม Apache ยังมีความสามารถอีกหลายอย่างเช่น การทำการยืนยันตัวบุคคล เพิ่มความปลอดภัยในการส่งข้อมูลโดยใช้โปรโตคอล https สร้างโฮสเสมือนภายในเครื่องเดียวกันได้ และการทำให้ url นั้นอ่านได้ง่ายขึ้นเช่น จาก https://th.wikipedia.org/wiki/%E0%B8%AD%E เปลี่ยนไปเป็น https://th.wikipedia.org/wifi/อะแพชี_เว็บเซิร์ฟเวอร์

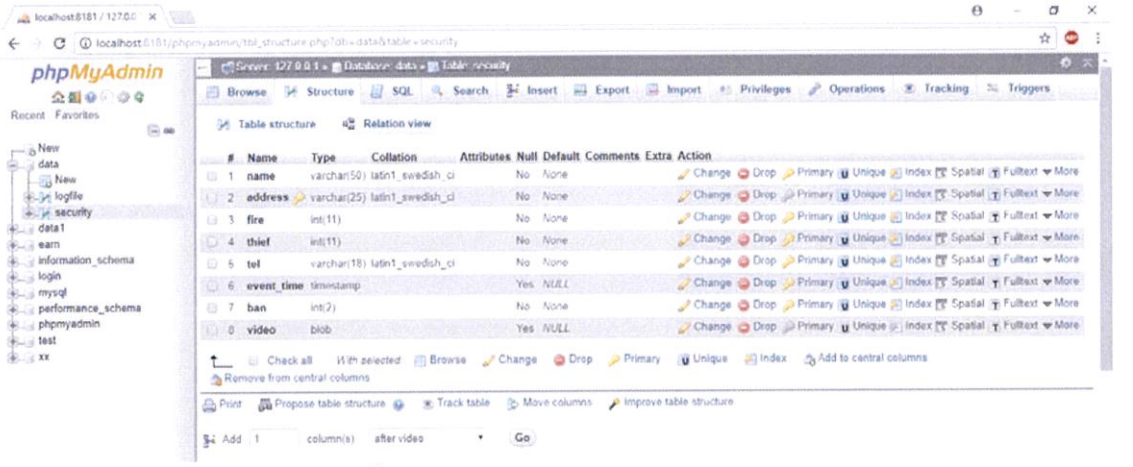
รูปที่ 2.23 โปรแกรม Apache

2.10 โปรแกรม phpMyAdmin

phpMyAdmin เป็นซอฟแวร์ที่พัฒนาขึ้นโดยใช้ภาษา PHP ใช้ในการบริหารจัดการฐานข้อมูล MySQL และ MariaDB โดยการใช้คำสั่งผ่านภาษา PHP หรือจัดการผ่านทางหน้าเว็บซึ่งจะมีเครื่องมือให้ใช้มากมายทำให้สามารถจัดการกับฐานข้อมูลได้ง่ายขึ้นเช่นการสร้าง ลบ แทรกแถวหรือหลัก อัพโหลดไฟล์ การแสดงผลข้อมูล เวลาของข้อมูลและอื่นๆอีกมากมาย

2.10.1 ลักษณะเด่นของ phpMyAdmin

1. สามารถใช้งานง่ายผ่านหน้าเว็บ
2. รองรับหลากหลายฟังชันการทำงานของ MySQL
3. สามารถนำเข้าข้อมูลได้ทั้ง SQL และ CSV
4. สามารถส่งข้อมูลออกได้หลายรูปแบบเช่น CSV SQL XML PDF ISO/IEC 26300 และอื่นๆ
5. สามารถจัดการได้เซิร์ฟเวอร์ได้หลากหลาย
6. สามารถสร้างกราฟฟิคของฐานข้อมูลได้หลากหลาย



รูปที่ 2.24 ตัวอย่างโครงสร้างฐานข้อมูล

2.11 ภาษา Python

ภาษา python นั้นสร้างขึ้นมาจากภาษาซี การประมวลผลเป็นแบบ interpreted ซึ่งจะประมวลคำสั่งทีละบรรทัดและเป็นภาษาชั้นสูงที่ถูกสร้างขึ้นมาให้สามารถใช้กับระบบได้หลายแบบ เช่น Window NP, Window XP ,Linux, Unix และภาษานี้ยังเป็นซอฟแวร์แบบโอเพนซอร์สซึ่งทำให้ผู้พัฒนาสามารถพัฒนาได้หลายหลายทำให้มีความสามารถสูงขึ้นอย่างรวดเร็ว โดยภาษานี้สามารถใช้กับราสป์เบอร์รีไพที่ใช้ระบบปฏิบัติการ Raspbian ซึ่งเป็นระบบปฏิบัติการที่ตั้งอยู่บนพื้นฐานของ Linux และยิ่งไปกว่านั้นภาษานี้ยังใช้ง่ายโดยใช้การเว้นช่องไฟในการแบ่งว่าคำสั่งใดอยู่ภายใต้ของชุดคำสั่งใดแทนที่จะใช้เครื่องหมาย { } หรือ ( ) เหมือนโปรแกรมอื่น

รูปที่ 2.25 โปรแกรม python

```
1  from wsgiref.simple_server import make_server
2  from pyramid.config import Configurator
3  from pyramid.view import view_config
4  from pyramid.response import Response
5
6  @view_config(route_name='hello', renderer='string')
7  def hello_world(request):
8      return 'Hello World'
9
10 if __name__ == '__main__':
11     config = Configurator()
12     config.add_route('hello', '/hello')
13     config.scan()
14     app = config.make_wsgi_app()
15     server = make_server('0.0.0.0', 8080, app)
16     server.serve_forever()
```

รูปที่ 2.26 ตัวอย่างคำสั่ง python

จากรูปที่ 2.26 จะเห็นได้ว่าในคำสั่งที่อยู่ภายใต้คำสั่ง if (บรรทัดที่ 11-16) นั้นจะมีการเว้นช่องไฟออกมาเพื่อบ่งบอกว่าคำสั่งนั้นเป็นคำสั่งที่อยู่ภายใต้ของคำสั่ง if

2.12 กล้องเว็บแคม (Webcam)

กล้องเว็บแคม รุ่น 088 ยี่ห้อ OKER ใช้การเชื่อมต่อด้วยพอร์ต Universal Serial Bus (USB) ซึ่งเป็นการส่งข้อมูลสื่อสารระหว่างอุปกรณ์กับคอมพิวเตอร์แบบอนุกรม โดยใช้ต่อกับพอร์ต USB ของราสป์เบอร์รีไพ เพื่อใช้เป็นอุปกรณ์บันทึกภาพขณะเกิดเหตุการณ์โจรกรรมภายในบ้าน ดังรูปที่ 2.27

รูปที่ 2.27 กล้องเว็บแคมยี่ห้อ OKER รุ่น 088

2.12.1 รายละเอียดของสินค้า OKER กล้องเว็บแคม รุ่น 088

1. ความละเอียด 10 ล้าน พิกเซล
2. มีอินฟราเรด ช่วยการมองเห็นในที่มืดสนิท (อินฟราเรดแท้)
3. สามารถปิด-เปิด ระบบอินฟราเรดได้ด้วยสวิทช์
4. ปรับแสงเองอัตโนมัติ
5. มีไมโครโฟนในตัว (USB Microphone)
6. มีปุ่มกดถ่ายรูป
7. สามารถอัดวีดีโอ และถ่ายภาพนิ่งได้
8. เซนเซอร์ CMOS 2 ล้าน - 10 ล้าน พิกเซล
9. เฟรมเรท (Frame rate) 30 เฟรมต่อวินาที

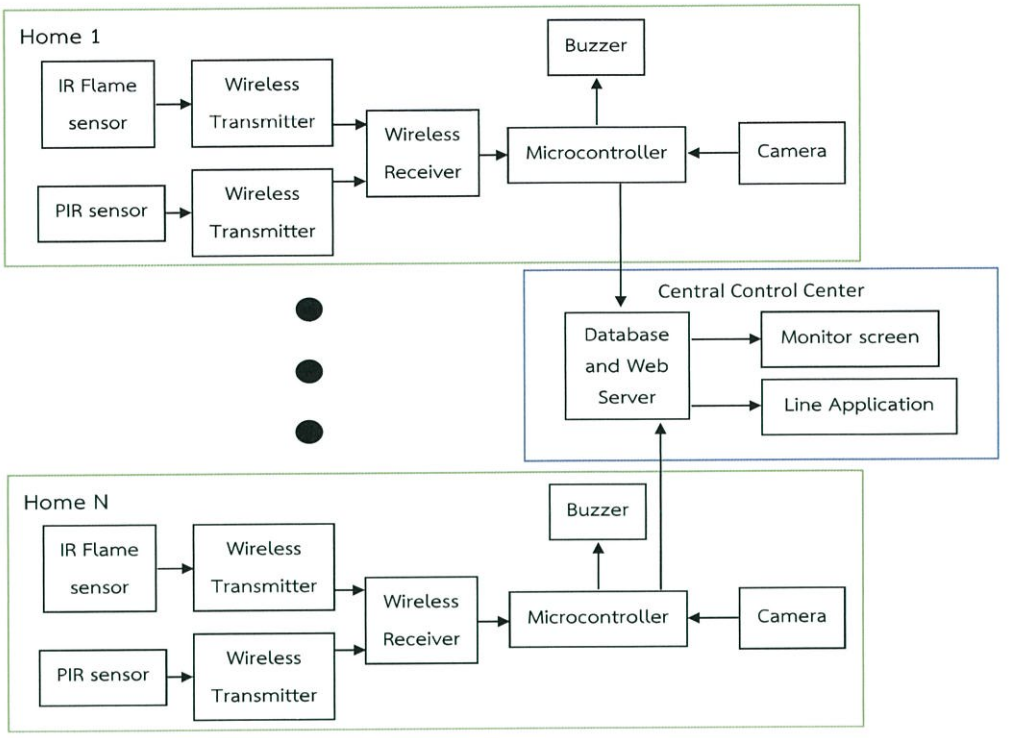
บทที่ 3

การออกแบบและการจัดทำปริญญานิพนธ์

3.1 การออกแบบ

3.1.1 การออกแบบบล็อกไดอะแกรมของระบบ

บล็อกไดอาแกรมดังรูปที่ 3.1 ระบบรักษาความปลอดภัยในหมู่บ้านจัดสรร ประกอบด้วยเซนเซอร์ 2 ชนิดคือเซนเซอร์ตรวจจับเปลวไฟ (IR flame sensor) ใช้ในการตรวจจับเปลวไฟที่เกิดขึ้นและเซนเซอร์ตรวจจับการเคลื่อนไหว (PIR sensor) ใช้ในการตรวจจับผู้บุกรุกที่เข้ามาบุกรุกภายในบ้านโดยไม่ได้รับอนุญาต ซึ่งเซนเซอร์หนึ่งตัวจะเชื่อมต่อกับไมโครคอนโทรลเลอร์ (Arduino Nano) และโมดูลสื่อสารไร้สาย (NRF24L01) ทั้ง 3 ส่วนนี้จะเรียกรวมเป็นหนึ่งโหนด เมื่อเซนเซอร์ตรวจจับเปลวไฟหรือผู้บุกรุกได้ เซนเซอร์ก็จะส่งสัญญาณแรงดันไฟฟ้าไปที่อาดูโนนาโน ข้อมูลจะถูกส่งจากอาดูโนนาโนแบบไร้สายไปยังราสป์เบอร์รีไพ ราสป์เบอร์รีไพก็จะจ่ายแรงดันไฟฟ้าออกมาทางพอร์ตที่ต่อกับบัซเซอร์ เพื่อแจ้งเตือนเพื่อนบ้านในบริเวณนั้นว่ามีอันตรายเกิดขึ้นและมีข้อความแจ้งเตือนไปยังเจ้าของบ้านผ่านแอพพลิเคชั่นไลน์ (LINE Application) นอกจากนี้ราสป์เบอร์รีไพก็จะส่งข้อมูลความผิดปกติและข้อมูลภาพไปยังศูนย์ควบคุมส่วนกลางผ่านเครือข่ายอินเตอร์เน็ตเพื่อให้เจ้าหน้าที่รักษาความปลอดภัยประจำหมู่บ้านจัดสรรรับรู้ถึงเหตุการณ์ที่เกิดแล้วสามารถเข้าไปแก้ไขปัญหาที่เกิดขึ้นได้ทันท่วงที

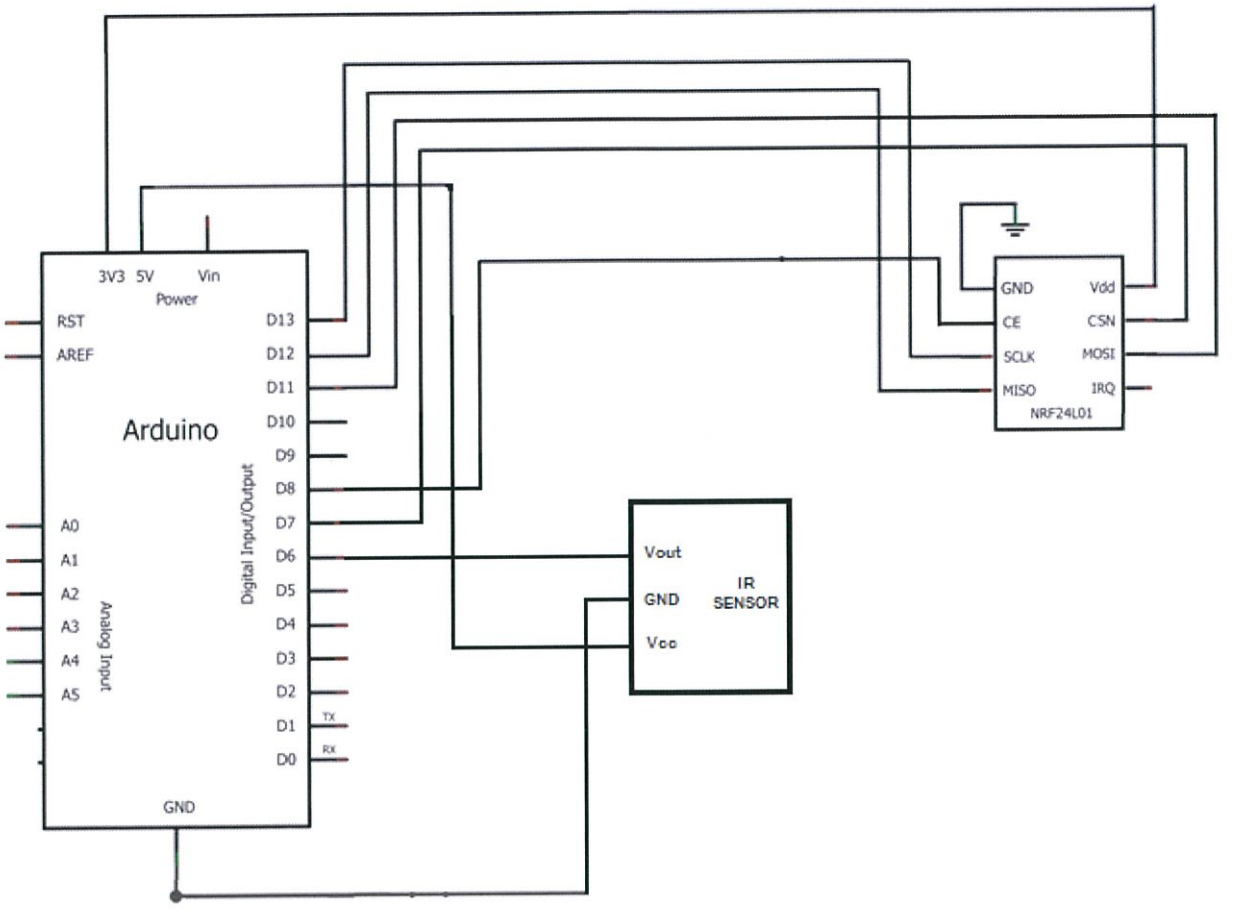


รูปที่ 3.1 บล็อกไดอะแกรมของระบบ

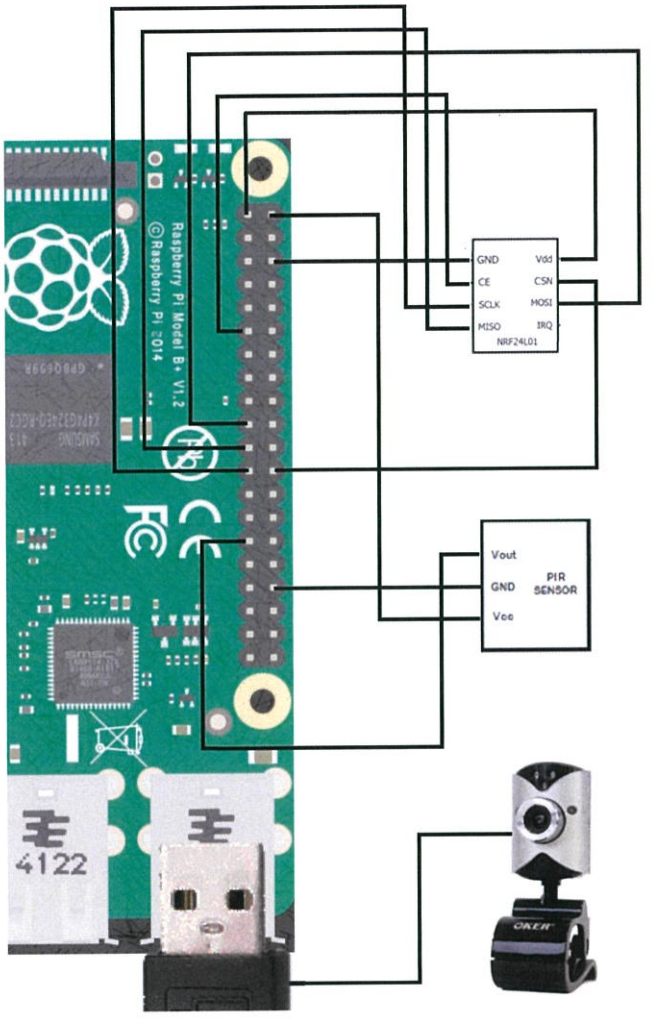
3.1.2 การออกแบบวงจร

3.1.2.1 วงจรรวมของระบบตรวจจับอัคคีภัยฝั่งเครื่องส่ง

อุปกรณ์เซนเซอร์ตรวจจับเปลวไฟมีขาใช้งาน 4 ขาแต่ใช้งานเพียง 3 ขาคือ ขาแรงดันไฟเลี้ยงขั้วลบ (Ground) ขาแรงดันไฟเลี้ยงขั้นบวก (Vcc) และขาแรงดันไฟออก (Vout) ซึ่งต่อกับขา D6 ของอาดูโนนาโน 3.0 ดังรูปที่ 3.2



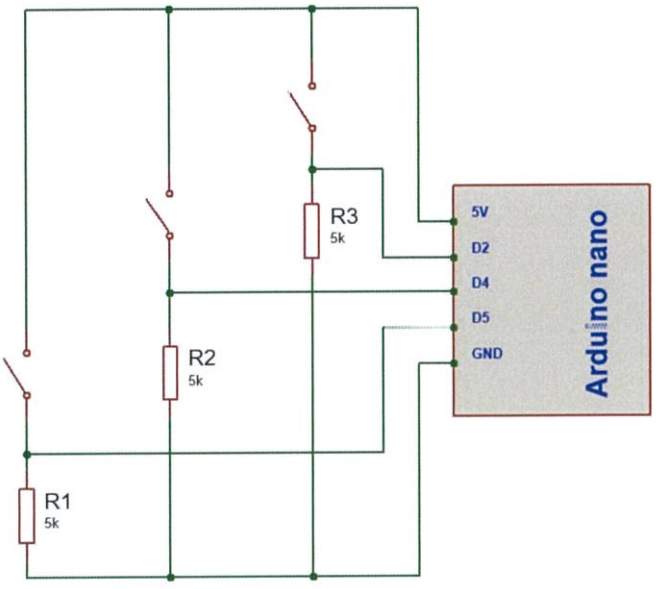
รูปที่ 3.2 วงจรรวมของระบบตรวจจับอัคคีภัยฝั่งเครื่องส่ง



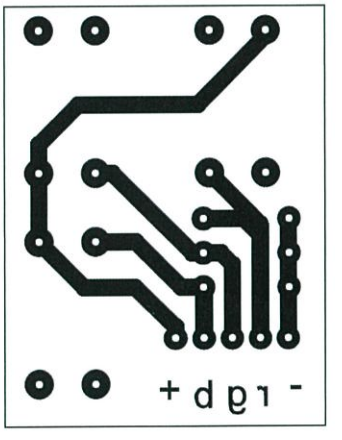
รูปที่ 3.3 วงจรรวมของระบบฝั่งเครื่องรับและระบบกล้องวงจรปิด

3.1.2.2 วงจรของรีโมทคอนโทรล

ในส่วนของรีโมทคอนโทรลนั้นทำขึ้นเพื่อให้ผู้ใช้งานสามารถเปิด-ปิดระบบรักษาความปลอดภัยได้ด้วยตัวเองโดยมีปุ่มที่ใช้ในการกดเพื่อส่งสัญญาณ SOS ไปที่ศูนย์รักษาความปลอดภัยเพื่อแจ้งเหตุฉุกเฉินได้อีกด้วย โดยวงจรประกอบด้วยอาดูโนนาโน 3.0 ตัวต้านทาน 20,000 โอห์ม และสวิตช์เปิด-ปิด มีวงจรดังรูปที่ 3.4 - 3.5



รูปที่ 3.4 วงจรของรีโมทคอนโทรล

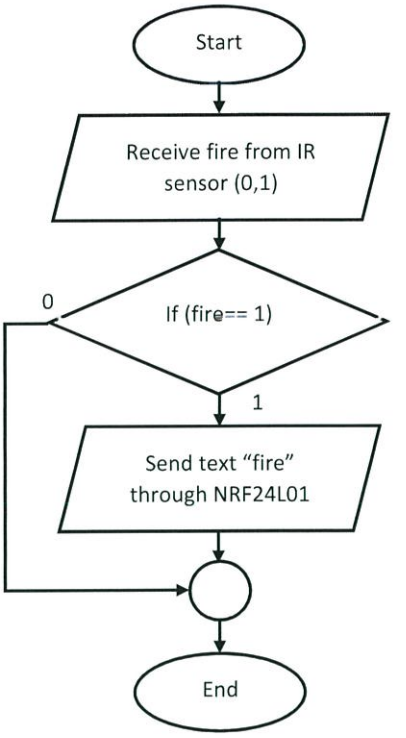


รูปที่ 3.5 ลายแผ่น PCB ของวงจรรีโมทคอนโทรล

3.1.3 ผังการทำงานของไมโครคอนโทรลเลอร์

3.1.3.1 ผังการทำงานของไมโครคอนโทรลเลอร์ที่ต่อกับเซนเซอร์ตรวจจับเปลวไฟ

เมื่อเซนเซอร์ตรวจจับเปลวไฟได้จะทำการส่งข้อมูลตัวหนังสือว่า “fire” ผ่าน NRF24L01 ไปยังตัวรับข้อมูล มีผังการทำงานตามรูปที่ 3.6



รูปที่ 3.6 ผังการทำงานของไมโครคอนโทรลเลอร์ที่ต่อกับ IR sensor

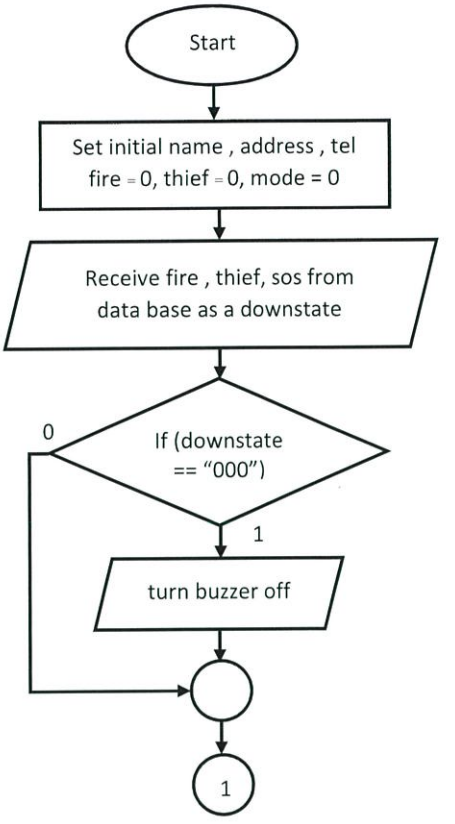
3.1.3.2 ผังการทำงานของไมโครคอนโทรลเลอร์ที่ต่อกับเซนเซอร์ตรวจจับการเคลื่อนไหวและเว็บแคม

แรกเริ่มจะต้องกำหนดค่า “name” , “address” , “tel” โดยที่ “name” คือชื่อของเจ้าของบ้าน “address” คือเลขที่บ้านและ “tel” คือเบอร์โทรศัพท์ของเจ้าของบ้าน การทำงานจะแบ่งเป็น 2 ส่วนคือการทำงานของการตรวจจับอัคคีภัยและการตรวจจับผู้บุกรุก ส่วนที่หนึ่งการตรวจจับอัคคีภัยราสป์เบอร์รีไพจะรอรับข้อมูลจากเครื่องรับสัญญาณไร้สาย ถ้าได้รับเป็นข้อมูลว่า “fire” หมายถึงเซนเซอร์ตรวจจับเปลวไฟได้แล้วจึงส่งข้อมูลแบบไร้สายว่า “fire” มายังฝั่งรับ เมื่อราสป์เบอร์รีไพ ได้รับข้อมูล ข้อมูลนี้จะทำการส่งข้อมูลสถานะความผิดปกติขึ้นไปยังฐานข้อมูลว่าเกิดเหตุอัคคีภัยโดยใช้ GET Request ไปยังฐานข้อมูลและเว็บเซิร์ฟเวอร์เพื่อแก้ไขสถานะ fire ในฐานข้อมูลจาก 0 ให้มีค่าเป็น 1 และจ่ายแรงดันให้กับขาที่ต่อกับบัซเซอร์เพื่อให้บัซเซอร์ดังแจ้งเตือนเจ้าของบ้านและคนในหมู่บ้านผ่านแอพพลิเคชั่นไลน์

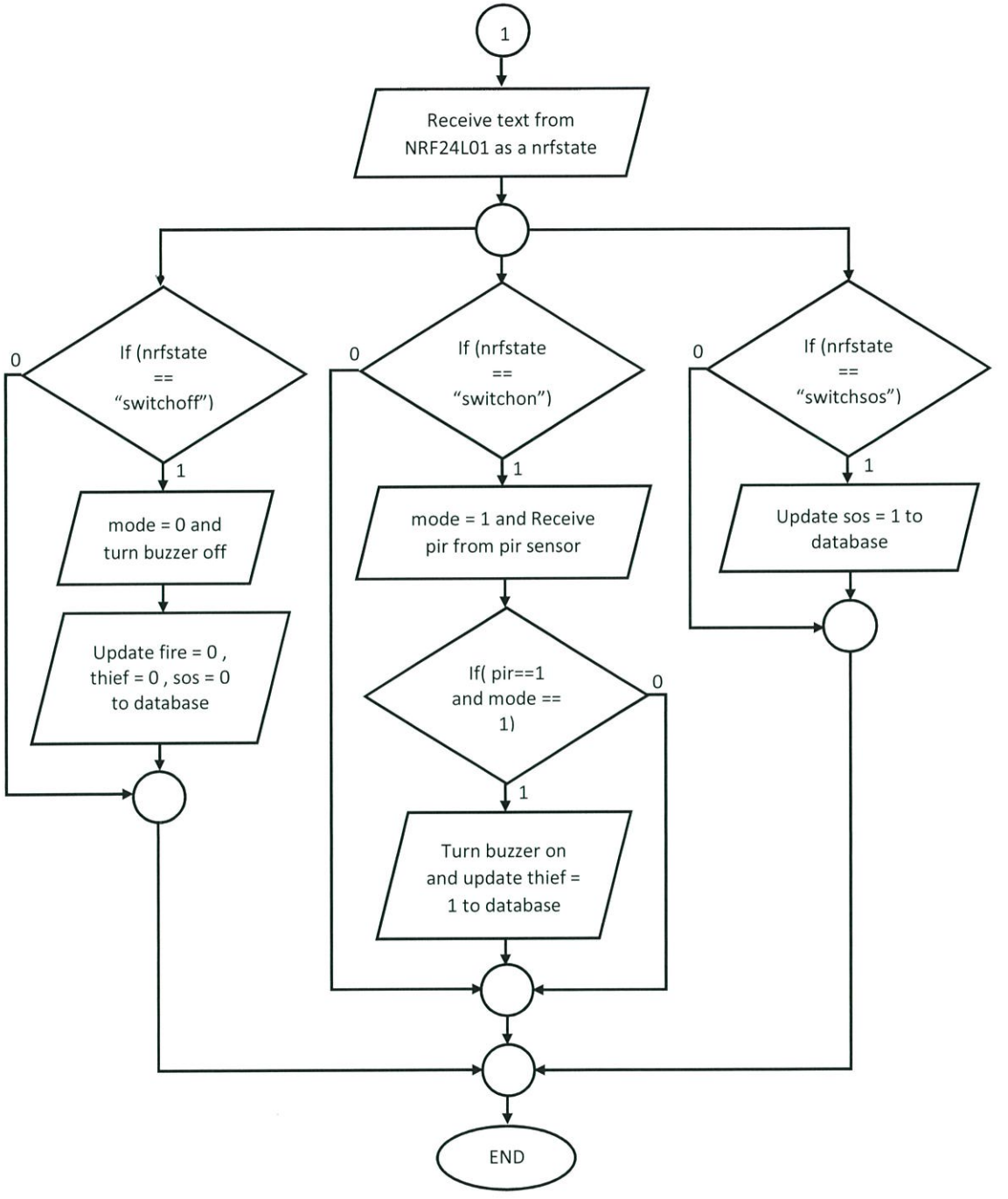
ส่วนที่สองการตรวจจับผู้บุกรุก ราสป์เบอร์รีไพจะตรวจสอบโหมดการทำงานโดยการรับคำสั่งจากรีโมทสวิตช์ โหมดการทำงานแรกคือให้เซนเซอร์ตรวจจับการเคลื่อนไหวไม่ทำงานใช้ใน

กรณีที่ต้องการปิดการทำงานเซนเซอร์ตรวจจับการเคลื่อนไหว เช่นในกรณีที่เจ้าของบ้านอยู่บ้าน โหมดสองคือเปิดการทำงานเซนเซอร์ตรวจจับการเคลื่อนไหวใช้ในกรณีที่ต้องการให้เปิดการตรวจจับการเคลื่อนไหวภายในบ้านเช่นในกรณีไม่อยู่บ้านหรือนอนหลับ เมื่ออาดูโนนาโนได้รับค่าแรงดันสูงจากขาที่ต่อกับเซนเซอร์ตรวจจับการเคลื่อนไหวแสดงว่าตรวจจับการเคลื่อนไหวได้จึงทำการส่งข้อมูลสถานะความผิดปกติโดยใช้ GET Request ไปยังฐานข้อมูลและเว็บเซิร์ฟเวอร์เพื่อแก้ไขสถานะ thief ในฐานข้อมูลจากค่าสถานะ 0 ให้มีค่าสถานะเป็น 1 และจ่ายแรงดันให้กับขาที่ต่อกับบัซเซอร์เพื่อให้บัซเซอร์ดังขึ้นต่อจากนั้นทำการแจ้งเตือนเจ้าของบ้านและคนในหมู่บ้านผ่านแอพพลิเคชั่นไลน์ นอกจากนี้ยังส่งข้อมูลภาพไปแสดงผลบนหน้าจอมอนิเตอร์ที่ศูนย์ควบคุมส่วนกลางเพื่อให้เจ้าหน้าที่รักษาความปลอดภัยเห็นเหตุการณ์ที่เกิดขึ้นและแก้ไขสถานการณ์ได้ทันท่วงที

กรณีที่เจ้าของไม่อยู่บ้านแล้วเกิดเหตุไฟไหม้หรือการโจรกรรมทรัพย์สินเจ้าหน้าที่รักษาความปลอดภัยจะเข้าไปแก้ไขปัญหาได้อย่างทันท่วงทีและสามารถติดต่อเจ้าของบ้านได้ตามเบอร์โทรศัพท์ที่ขึ้นบนจอมอนิเตอร์ เมื่อสามารถแก้ไขสถานการณ์ที่เกิดขึ้นได้แล้วเจ้าหน้าที่รักษาความปลอดภัยสามารถปิดเสียงบัซเซอร์ผ่านหน้าจอมอนิเตอร์ได้ เมื่อกดปิดบัซเซอร์ผ่านหน้าจอมอนิเตอร์ก็จะส่งค่า “fire = 0” , “thief = 0” และ “sos = 0” ไปที่ฐานข้อมูลจากนั้นไมโครคอนโทรลเลอร์ส่วนกลางของบ้านก็จะรับค่า “fire = 0” , “thief = 0” และ “sos = 0” จากฐานข้อมูลแล้วไมโครคอนโทรลเลอร์ส่วนกลางของบ้านก็จ่ายแรงดันศูนย์ออกไปที่พอร์ตบัซเซอร์เพื่อปิดเสียงแจ้งเตือน มีผังการทำงานตามรูปที่ 3.7-3.8



รูปที่ 3.7 ผังการทำงานของฝั่งเซนเซอร์ระบบตรวจจับผู้บุกรุกส่วนที่ 1

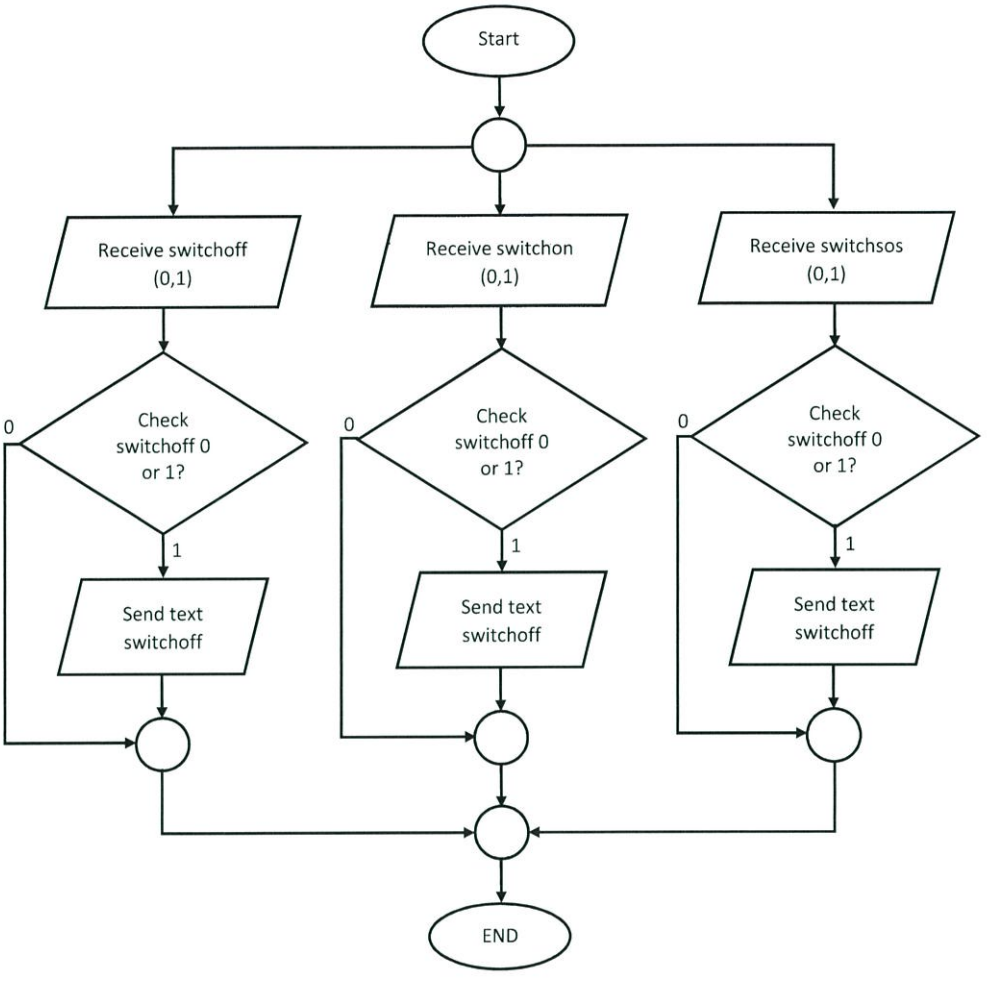


รูปที่ 3.8 ผังการทำงานของฝั่งเซนเซอร์ระบบตรวจจับผู้บุกรุกส่วนที่ 2

3.1.2.3 ผังการทำงานของไมโครคอนโทรลเลอร์ของรีโมท

กรณีที่เหตุการณ์ผิดปกติไม่ได้เกิดขึ้นจากเหตุอัคคีภัยหรือการบุกรุกจากบุคคลภายนอกได้อาทิเช่นกรณีที่เกิดข้อผิดพลาดตรวจจับคนในบ้านแทนที่จะเป็นผู้บุกรุก สัญญาณบัซเซอร์ จึงดังขึ้นโดยไม่ได้เจตนาก็สามารถกดสวิตช์เพื่อปิดเสียงบัซเซอร์ได้ เมื่อกดปิดบัซเซอร์ราสป์เบอร์รีไพก็จะ ใช้ GET Request ไปยังฐานข้อมูลและเว็บเซิร์ฟเวอร์เพื่อแก้ไขสถานะ “fire” , “thief” และ “sos” ในฐานข้อมูลให้มีค่าสถานะเป็น “0” แล้วจึงทำการแจ้งเตือนเจ้าของบ้านและคนในหมู่บ้านผ่านแอพพลิเคชั่นไลน์

ในกรณีที่เจ้าของบ้านต้องการความช่วยเหลือฉุกเฉินก็สามารถกดสวิตช์บนรีโหมดเพื่อขอความช่วยเหลือจากเจ้าหน้าที่รักษาความปลอดภัยได้ เมื่อกดขอความช่วยเหลือราสป์เบอร์รีไพจะใช้ GET Request ไปยังฐานข้อมูลและเว็บเซิร์ฟเวอร์เพื่อแก้ไขสถานะ “sos” ในฐานข้อมูลให้มีค่าสถานะเป็น “1” แล้วจึงทำการแจ้งเตือนเจ้าของบ้านและคนในหมู่บ้านผ่านแอพพลิเคชั่นไลน์ ดังรูปที่ 3.9



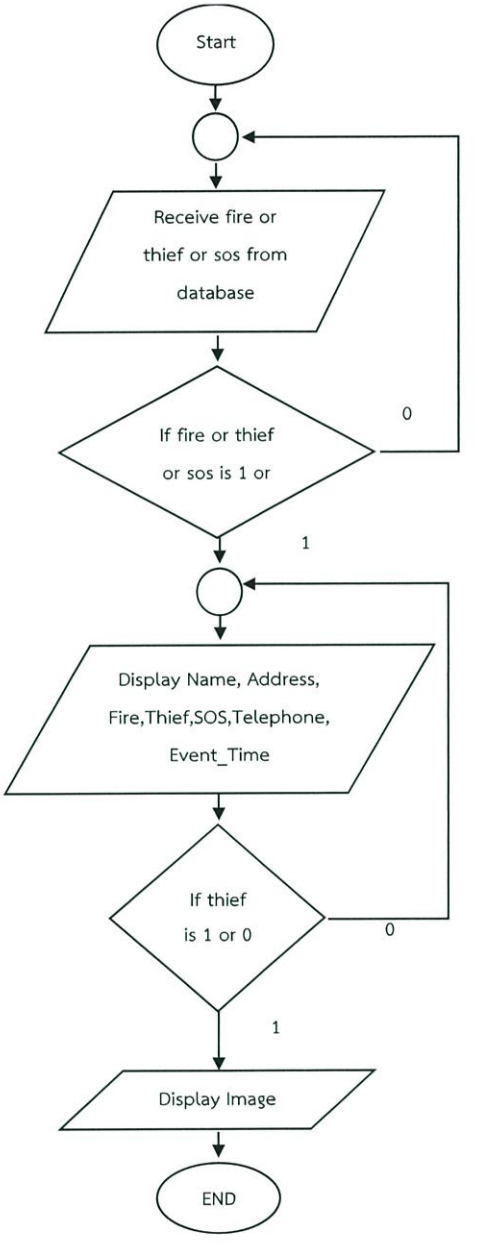
รูปที่ 3.9 ผังการทำงานของไมโครคอนโทรลเลอร์ที่ต่อกับสวิตช์ต่างๆ

3.1.3 ผังการทำงานของส่วนแสดงผล

3.1.3.1 ผังการทำงานของหน้าจอมอนิเตอร์

ในส่วนของการแสดงผลนั้นระบบจะทำการตรวจสอบข้อมูล “fire”, “thief”, “sos” ที่อยู่ในฐานข้อมูล ถ้าข้อมูลที่ตรวจสอบมีค่าเปลี่ยนจาก “0” ไปเป็น “1” ระบบจะนำข้อมูลทั้งแถวซึ่งประกอบไปด้วย Name, Address, Fire, Thief, SOS, Telephone, Event_Time มาแสดงผลบนหน้าจอมอนิเตอร์และจะมีปุ่มกดรูปกล้องอยู่ท้ายแถวนั้น ในกรณีที่เกิดเหตุการณ์ผิดปกติภายในบ้านเนื่องมาจากเซนเซอร์ตรวจจับการเคลื่อนไหวได้ ระบบจะเปิดใช้งานกล้องเพื่อบันทึก

ข้อมูลภาพ ณ ขณะนั้นและจะส่งข้อมูลภาพมาแสดงบนหน้าจอมอนิเตอร์เพื่อให้เจ้าหน้าที่รักษาความปลอดภัยภายในหมู่บ้านนั้นได้ทำการตรวจสอบภาพของเหตุการณ์ที่เกิดขึ้น ดังรูปที่ 3.10



รูปที่ 3.10 ผังการทำงานของหน้าจอมอนิเตอร์

3.1.3.2 ผังการทำงานของหน้าต่างแสดงประวัติ

ในส่วนของหน้าต่างแสดงประวัติระบบจะทำการตรวจสอบข้อมูลภายในฐานข้อมูลที่ใช้ในการเก็บประวัติ จะใช้ฐานข้อมูลชื่อ “logfile” ฐานข้อมูลนี้จะมีการเพิ่มข้อมูลในฐานข้อมูลทุกครั้งเมื่อมีเหตุการณ์ผิดปกติมาแสดงซึ่งประกอบไปด้วย Name, Address, Fire, Thief, SOS, Telephone, Event_Time และเมื่อเหตุการณ์เป็นปกติแล้ว จะมีอีกหนึ่งคอลัมน์ที่มีชื่อว่า stop_time เพิ่มเข้ามาจะเก็บข้อมูลของเวลาเมื่อเหตุการณ์ได้กลับไปเป็นปกติ

นอกจากนี้ภายในหน้าต่างนี้จะมีฟังก์ชันการค้นหาแบบเฉพาะอีก 4 แบบคือ

1) แสดงเฉพาะเหตุการณ์ที่เกิดไฟไหม้เพียงอย่างเดียวจะนำข้อมูลของทุกๆบ้านที่เกิดเหตุการณ์ไฟไหม้มาแสดงบนหน้าต่างประวัติ

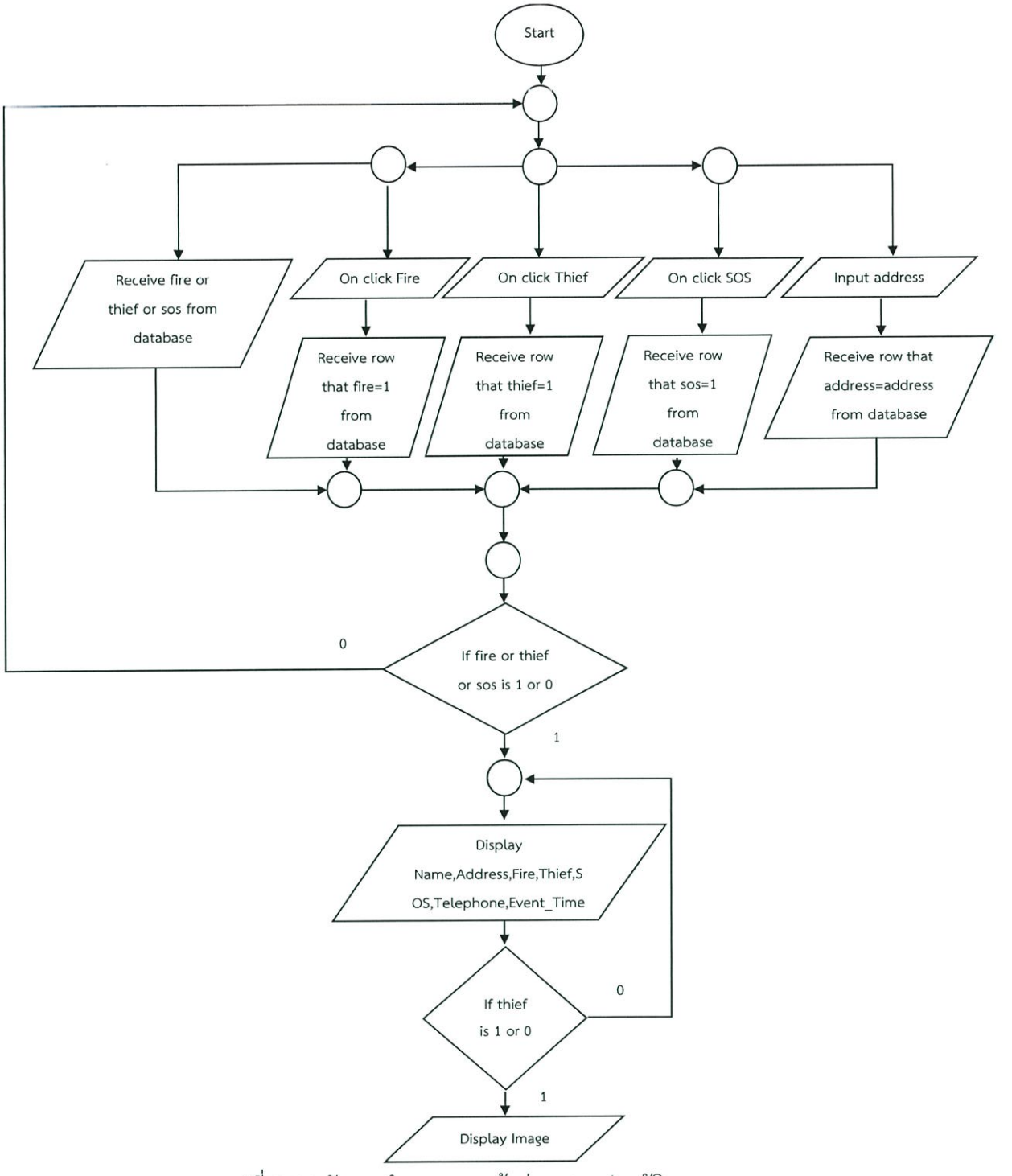
2) แสดงเฉพาะเหตุการณ์ที่เกิดโจรกรรมเพียงอย่างเดียวจะนำข้อมูลของทุกๆบ้านที่เกิดเหตุการณ์โจรกรรมมาแสดงบนหน้าต่างประวัติ

3) แสดงเฉพาะเหตุการณ์ที่มีการร้องขอความช่วยเหลือจากเจ้าของบ้านเพียงอย่างเดียวจะนำข้อมูลของทุกๆบ้านที่มีการร้องขอความช่วยเหลือจากเจ้าหน้าที่รักษาความปลอดภัยมาแสดงบนหน้าต่างประวัติ

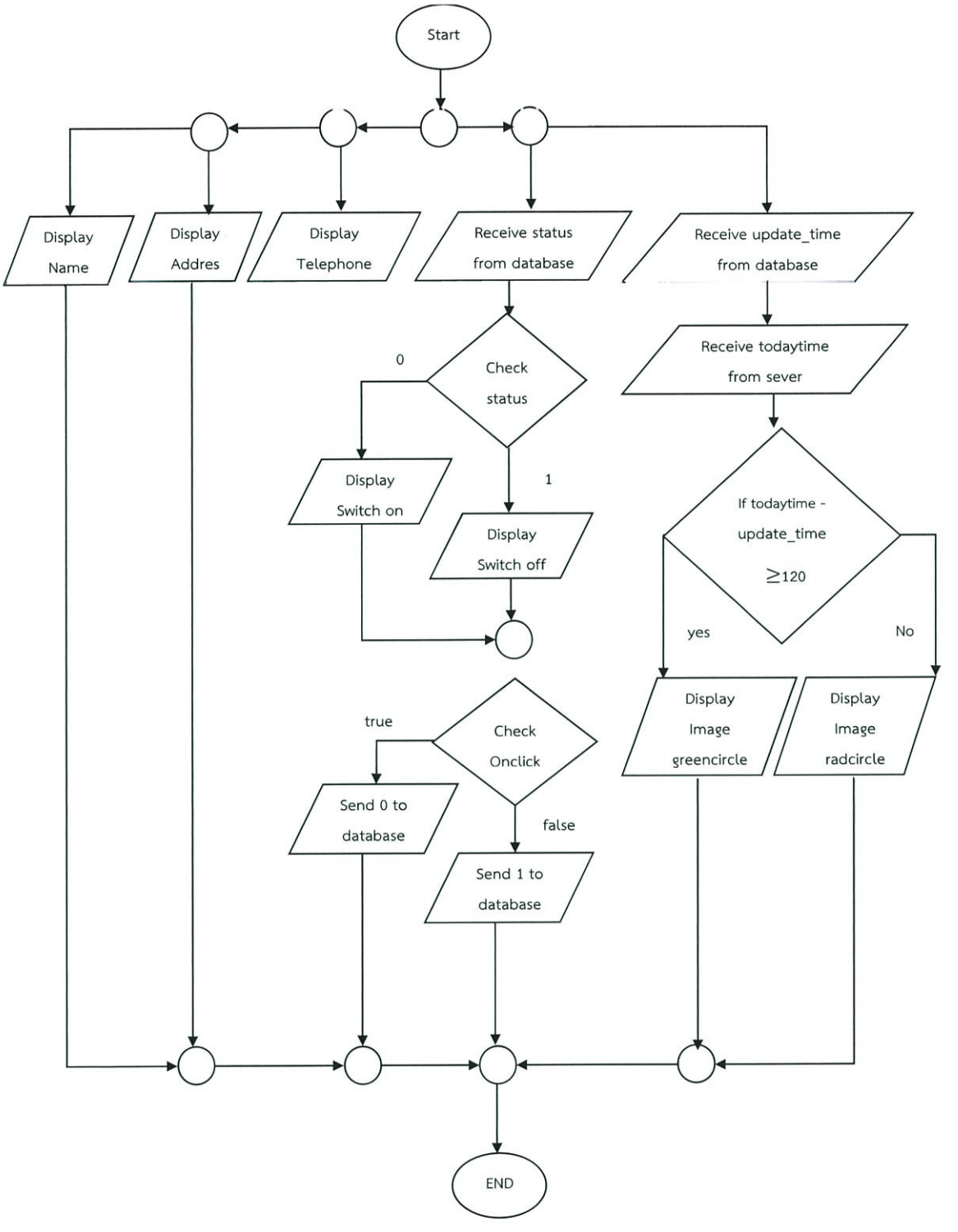
4) แสดงเฉพาะเหตุการณ์ที่เกิดกับบ้านเลขที่เดียว จะค้นหาเหตุการณ์ที่เกิดขึ้นภายในบ้านที่ต้องการเข้าดูแบบเจาะจงไปที่บ้านเลขที่เดียวและนำมาแสดงบนหน้าต่างประวัติ การทำงานของหน้าต่างแสดงประวัติมีผังการทำงานตามรูปที่ 3.11

3.1.3.3 ผังการทำงานของหน้าต่างแสดงสถานะอุปกรณ์

หน้าต่างนี้จะใช้ในการความคุมการทำงานโหมดออนไลน์หรือโหมดออฟไลน์ของราสป์เบอร์รีไพ ใช้ในกรณีที่ภายในบ้านนั้นมีเซนเซอร์ตัวใดตัวหนึ่งเสียทำให้เกิดการแจ้งเตือนซ้ำๆ เจ้าหน้าที่รักษาความปลอดภัยสามารถที่จะสั่งปิดการการใช้งานของราสป์เบอร์รีไพบ้านหลังนั้นให้อยู่ในโหมดออฟไลน์ได้ อีกหนึ่งการทำงานคือสามารถตรวจสอบการเชื่อมต่อระหว่างราสป์เบอร์รีไพกับฐานข้อมูลได้เพื่อที่จะได้มั่นใจได้ว่าราสป์เบอร์รีไพยังสามารถส่งข้อมูลมายังฐานข้อมูลได้และยังใช้งานได้ดี มีผังการทำงานของหน้าต่างแสดงสถานะอุปกรณ์ ดังรูปที่ 3.12



รูปที่ 3.11 ผังการทำงานของหน้าต่างแสดงประวัติ



รูปที่ 3.12 ผังการทำงานของหน้าต่างแสดงสถานะอุปกรณ์

3.1.4 การออกแบบในส่วนของการแจ้งเตือนผ่านแอปพลิเคชั่นไลน์

3.1.4.1 การแจ้งเตือนผ่านแอปพลิเคชั่นไลน์ เมื่อเกิดอัคคีภัย

เมื่อตรวจจับเหตุการณ์ผิดปกติต่างๆได้ ระบบจะส่งข้อความผ่านแอปพลิเคชั่นไลน์ ไปที่เจ้าของบ้านที่เกิดเหตุ เพื่อแจ้งให้เจ้าของบ้านทราบ โดยข้อความแจ้งเตือนจะระบุถึง Address คือ บ้านเลขที่ ที่เกิดเหตุ และ Problem คือ ปัญหาที่เกิดขึ้น เช่น เมื่อเกิดเหตุการณ์อัคคีภัยจะส่งข้อความว่า Security Alert : Address : "122" Problem : "FIRE" PS : Security guard is working on it

3.1.4.2 การแจ้งเตือนผ่านแอปพลิเคชั่นไลน์ เมื่อเกิดการโจรกรรม

การแจ้งเตือนผ่านแอปพลิเคชั่นไลน์เมื่อเกิดการโจรกรรม จะคล้ายกับการแจ้งเตือนเมื่อเกิดอัคคีภัย แต่จะแตกต่างกันเล็กน้อย โดยเมื่อเกิดเหตุการณ์อัคคีภัยจะเปลี่ยนเป็นแจ้งเตือนในช่อง Problem ว่า “Robbery”

3.1.4.3 การแจ้งเตือนผ่านแอปพลิเคชั่นไลน์เมื่อสามารถยุติเหตุการณ์ต่างๆที่เกิดขึ้นได้แล้ว

เมื่อสามารถยุติเหตุการณ์ที่เกิดขึ้นได้แล้ว เจ้าหน้าที่รักษาความปลอดภัยของหมู่บ้านจะหยุดการแจ้งเตือนสถานการณ์ผิดปกติผ่านทางหน้าเว็บเพจ ซึ่งจะมีการแจ้งเตือนไปที่เจ้าของบ้านที่เกิดเหตุอีกครั้ง ผ่านแอปพลิเคชั่นไลน์เพื่อแจ้งเตือนว่า เหตุการณ์ได้กลับมาเป็นปกติแล้วโดยจะแจ้งบ้านเลขที่ที่สามารถยุติเหตุการณ์ได้แล้ว พร้อมกับข้อความแสดงว่า ปัญหาที่เกิดขึ้นได้ถูกยุติลงแล้ว

3.1.5 การออกแบบฐานข้อมูล

3.1.5.1 การออกแบบฐานข้อมูลที่ชื่อว่า “security”

ฐานข้อมูลที่ชื่อว่า “security” เป็นฐานข้อมูลที่ใช้ในการแสดงผลในหน้าของจอมอนิเตอร์มีทั้งหมด 9 แถว ดังรูปที่ 3.13 คือ

1. “name” คือชื่อเจ้าของบ้าน
2. “address” คือบ้านเลขที่
3. “fire” สถานะของอัคคีภัย

4. “thief” สถานะของการโจรกรรม

5. “sos” สถานการณ์ขอความช่วยเหลือ

6. “tel” คือเบอร์โทรศัพท์ของเจ้าของบ้าน

7. “event_time” คือเวลาที่เกิดเหตุขึ้น

8. “status” คือสถานะที่ใช้เปิดปิดการรับส่งข้อมูลของไมโครคอนโทรลเลอร์กับศูนย์ควบคุมส่วนกลางถ้ามีค่าเป็น “1” คือปิด และ “0” คือเปิด

9. “update_time” คือเวลาที่อุปกรณ์ในแต่ละบ้านติดต่อมาล่าสุดใช้ในการดูสถานการณ์ออนไลน์และออฟไลน์

รูปที่ 3.13 ตัวแปรต่างๆที่จะเก็บในฐานข้อมูลที่มีชื่อว่า “security”

3.1.5.2 การออกแบบฐานข้อมูลที่ชื่อว่า “logfile”

ฐานข้อมูลที่มีชื่อว่า “logfile” ใช้ในการเก็บข้อมูลของเหตุการณ์ที่เกิดทั้งหมด เช่น เหตุการณ์ไฟไหม้ เหตุการณ์โจรกรรม เหตุการณ์ขอความช่วยเหลือ มีทั้งหมด 9 แถว ดังรูปที่ 3.14 คือ

1. “name” คือชื่อเจ้าของบ้าน
2. “address” คือบ้านเลขที่
3. “fire” สถานะของอัคคีภัย
4. “thief” สถานะของการโจรกรรม
5. “sos” สถานการณ์ขอความช่วยเหลือ
6. “tel” คือเบอร์โทรศัพท์ของเจ้าของบ้าน
7. “event_time” คือเวลาที่เกิดเหตุขึ้น
8. “stop_time” คือเวลาที่เหตุการณ์จบลง
9. “photo” คือชื่อไฟล์รูปภาพที่บันทึกในบ้านที่เกิดเหตุโจรกรรม

name	address	fire	thief	sos	tel	event_time	stop_time	photo

รูปที่ 3.14 ตัวแปรต่างๆที่จะเก็บในฐานข้อมูลที่ชื่อว่า “logfile”

3.2 เครื่องมือที่ใช้ในการทดลอง

3.2.1 ราสป์เบอร์รีไพ 3 โมเดลบี (Raspberry pi 3 model B)

บอร์ดคอมพิวเตอร์ขนาดเล็กที่สามารถเชื่อมต่อกับจอมอนิเตอร์ คีย์บอร์ด และเมาส์ได้ สามารถนำมาประยุกต์ใช้ในการทำโครงงานทางด้านอิเล็กทรอนิกส์ การเขียนโปรแกรม อีกทั้งยังสามารถเล่นไฟล์วีดีโอความละเอียดสูง (High-Definition) ได้อีกด้วย มีจุดเชื่อมต่อ GPIO ให้ผู้ใช้สามารถนำไปใช้ร่วมกับอุปกรณ์อิเล็กทรอนิกส์อื่นๆได้ ซึ่งนำมาประยุกต์ใช้เป็นหน่วยประมวลผลส่วนกลางของแต่ละบ้าน ใช้รับค่าจากเซนเซอร์ต่างๆในบ้านเพื่อส่งไปที่ฐานข้อมูล นอกจากนี้ยังใช้เชื่อมต่อกับกล้องเว็บแคมเพื่อใช้เป็นกล้องวงจรปิด ดังรูปที่ 3.15

รูปที่ 3.15 Raspberry pi 3 model B

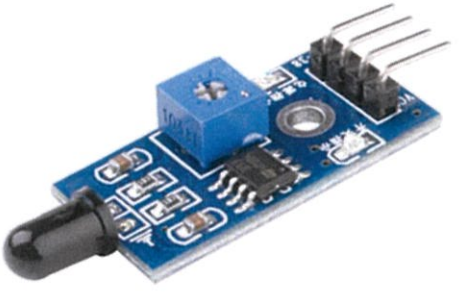
3.2.2 เซนเซอร์ตรวจจับการเคลื่อนไหว (Passive Infrared Sensor module)

เซนเซอร์ตรวจจับการเคลื่อนไหว (PIR sensor) คืออุปกรณ์ที่ใช้สำหรับตรวจจับการมีอยู่ของมนุษย์หรือสัตว์โดยใช้หลักการการตรวจจับอินฟราเรด จากวัตถุผ่านยังเลนส์เพื่อรวมแสงแล้วโฟกัสมายังไพโรอิเล็กตริก (pyroelectric) ซึ่งเป็นตัวแปลงพลังงานความร้อนจากรังสีอินฟราเรด เป็นพลังงานไฟฟ้า เมื่อมีสิ่งมีชีวิตที่มีรังสีอินฟราเรดเคลื่อนผ่านเข้ามารังสีอินฟราเรดก็จะถูกรวมโดยเลนส์แล้วโฟกัสไปยังไพโรอิเล็กตริกแล้วเปลี่ยนอินฟราเรดให้เป็นแรงดันไฟฟ้าแม้ว่าจะมีอินฟราเรด เพียงเล็กน้อยก็สามารถตรวจจับได้เพราะเลนส์ทำหน้าที่รวบรวมให้รังสีเข้มขึ้น ดังรูปที่ 3.16

รูปที่ 3.16 Passive Infrared Sensor module

3.2.3 เซนเซอร์ตรวจจับเปลวไฟ (Infrared IR Flame Detector Sensor Module)

เซนเซอร์ตรวจจับเปลวไฟเป็นโมดูลเซนเซอร์ที่ใช้ในการตรวจจับรังสี Infrared ที่เกิดจากแหล่งกำเนิดความร้อนต่างๆเช่นเปลวเพลิง สามารถให้เอาต์พุตทั้งเป็นแบบ แอนาล็อก แรงดันเอาต์พุต 0 - 5 โวลต์ และแบบดิจิตอลถ้าไม่มีการตรวจจับเปลวไฟจะให้แรงดัน 0 โวลต์ แต่ถ้ามีการตรวจจับเปลวเพลิงก็จะจ่ายแรงดัน 5 โวลต์ สามารถปรับค่า Threshold ได้โดยการปรับ potentiometer บนโมดูล ดังรูปที่ 3.17



รูปที่ 3.17 Infrared IR Flame Detector Sensor Module

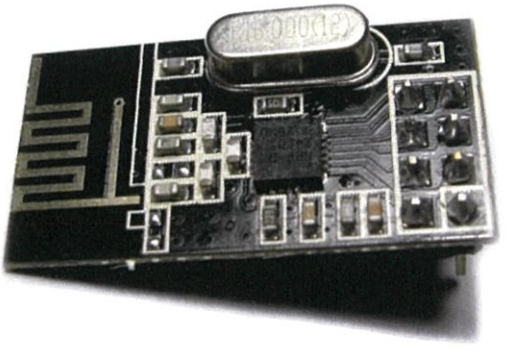
3.2.4 บัซเซอร์ (Buzzer)

เป็นลำโพงแบบแม่เหล็กไฟฟ้า เสียงที่ได้เกิดจากการจ่ายแรงดันไฟฟ้ากระแสตรงขนาดความต่างศักย์ 3 – 24 โวลต์ เข้าไปที่ตัวลำโพง วงจรกำเนิดความถี่ในตัวของลำโพงจะให้กำเนิดเสียงแต่เสียงจะมีความถี่เดียว ไม่สามารถเปลี่ยนความถี่ของเสียงได้ ตัวอย่างของบัซเซอร์ดังรูปที่ 3.18

รูปที่ 3.18 Buzzer

3.2.5 โมดูลสื่อสารไร้สาย (NRF24L01 Module)

เป็นโมดูลสื่อสารชนิดไร้สาย ซึ่งสามารถใช้เป็นได้ทั้งตัวรับและตัวส่งขึ้นอยู่กับการเขียนโปรแกรมบนอาดูโนไอดีอีสามารถใช้งานกับไมโครคอนโทรลเลอร์ให้ติดต่อสื่อสารพร้อมกันหลายๆ ตัวได้ ระยะทำงานอยู่ที่ 15 – 500 เมตร ขึ้นอยู่กับความเร็วที่ใช้ในการส่งใช้พลังงานต่ำจึงสามารถใช้งานได้นาน นำมาประยุกต์ใช้ในการส่งข้อมูลสถานะที่ตรวจจับได้จากเซนเซอร์ตรวจจับผู้บุกรุกและตรวจจับเปลวไฟ ดังรูปที่ 3.19

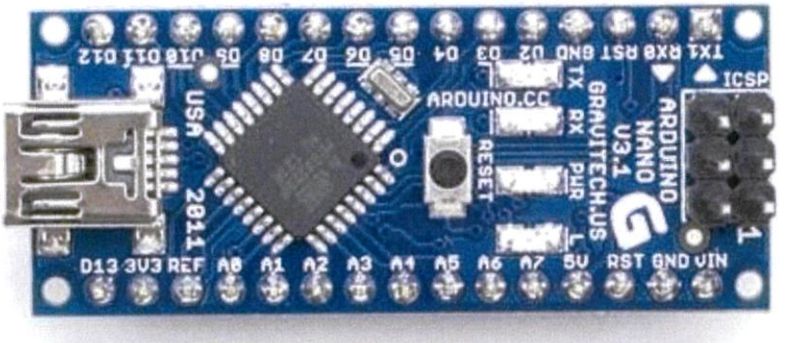


รูปที่ 3.19 NRF24L01 Module

3.2.6 ไมโครคอนโทรลเลอร์อาดูโนนาโน (Arduino Nano)

เป็นไมโครคอนโทรลเลอร์ที่มีความสามารถคล้ายกับอาดูโนอูโน (Arduino Uno) คือใช้ชิปประมวลผล ATmega328 หรือ ATmega168 เหมือนกัน การใช้งานทำได้เหมือนกัน แต่ต่างกันตรงที่ออกแบบให้มีขนาดเล็กกว่าเดิม และตัดบางฟังก์ชั่นที่ไม่จำเป็นออก ใช้มินิยูเอสบี

(Mini USB) เชื่อมต่อกับคอมพิวเตอร์ มีพอร์ตดิจิตอลอินพุตเอาต์พุต 14 พอร์ต มีพอร์ตแอนาล็อกอินพุต 8 พอร์ต ดังรูปที่ 3.20



รูปที่ 3.20 ไมโครคอนโทรลเลอร์ Arduino Nano

3.2.7 กล้องเว็บแคมยี่ห้อ OKER รุ่น 088

กล้องเว็บแคมรุ่น 088 ยี่ห้อ OKER ใช้การเชื่อมต่อแบบยูนิเวอร์แซลซีเรียลบัส (Universal Serial Bus) หรือ USB ซึ่งเป็นการส่งข้อมูลสื่อสารระหว่างอุปกรณ์กับคอมพิวเตอร์แบบอนุกรม โดยใช้ต่อกับพอร์ต USB ของราสป์เบอร์รีไพ 3 โมเดลบีเพื่อใช้เป็นอุปกรณ์บันทึกภาพขณะเกิดเหตุการณ์โจรกรรมภายในบ้าน ดังรูปที่ 3.21

รูปที่ 3.21 กล้องเว็บแคม ยี่ห้อ OKER รุ่น 088

3.3 การจัดเก็บผลการทดลอง

3.3.1 ผลการทดสอบเซนเซอร์ตรวจจับเปลวไฟเบื้องต้น

1) ผลการทดสอบเซนเซอร์ตรวจจับเปลวไฟเมื่อไม่มีการตรวจจับเปลวไฟ

2) ผลการทดสอบเซนเซอร์ตรวจจับเปลวไฟเมื่อมีการตรวจจับเปลวไฟได้

3) ผลการทดสอบเซนเซอร์ตรวจจับเปลวไฟ 2 ตัวพร้อมกัน

3.3.2 ผลการทดสอบเซนเซอร์ตรวจจับการเคลื่อนไหว

1) ผลการทดสอบเซนเซอร์ตรวจจับการเคลื่อนไหว เมื่อไม่มีการตรวจจับการเคลื่อนไหว

2) ผลการทดสอบเซนเซอร์ตรวจจับการเคลื่อนไหว เมื่อมีการตรวจจับการเคลื่อนไหว

3) ผลการทดสอบระยะการตรวจจับของเซนเซอร์ตรวจจับการเคลื่อนไหว

3.3.3 ผลการทดสอบในส่วนของระบบการรับส่งข้อมูลไร้สายในรูปแบบดิจิตอล

1) ผลการทดสอบโมดูลสื่อสารไร้ NRF24L01 ด้านส่ง

2) ผลการทดสอบโมดูลสื่อสารไร้ NRF24L01 ด้านรับ

3.3.4 ผลการทดสอบระบบในส่วนของราสป์เบอร์รีไพ

1) ผลการทดสอบระบบในส่วนของราสป์เบอร์รีไพเมื่อไม่มีเหตุการณ์ผิดปกติ

2) ผลการทดสอบระบบในส่วนของราสป์เบอร์รีไพเมื่อกดรีโมทคอนโทรลเพื่อเปิดระบบป้องกันการโจรกรรม

3) ผลการทดสอบระบบในส่วนของราสป์เบอร์รีไพเมื่อมีเหตุพลิงไหม้

4) ผลการทดสอบระบบในส่วนของราสป์เบอร์รีไพเมื่อมีเหตุโจรกรรม

5) ผลการทดสอบระบบในส่วนของราสป์เบอร์รีไพเมื่อกดรีโมทคอนโทรลเพื่อส่งสัญญาณขอความช่วยเหลือ

6) ผลการทดสอบระบบในส่วนของราสป์เบอร์รีไพเมื่อกดรีโมทคอนโทรลเพื่อหยุดเสียงเตือน

7) ผลการทดสอบระบบในส่วนของราสป์เบอร์รีไพเมื่อกดรีโมทคอนโทรลเพื่อปิดระบบป้องกันการโจรกรรม

3.3.5 ผลการทดสอบในส่วนของการใช้งานกล้องเว็บแคม

1) ผลการทดสอบกล้องเว็บแคมจากการใช้ภาษา python เพื่อสั่งการกล้องให้ถ่ายภาพ

2) ผลการทดสอบกล้องเว็บแคมจากการใช้ภาษา python เพื่อสั่งการกล้องให้ถ่ายวีดีโอ

3.3.6 ผลการทดสอบในส่วนของหน้าจอมอนิเตอร์

1) ผลการทดสอบระบบเมื่อเหตุการณ์เป็นปกติ

2) ผลการทดสอบระบบเมื่อมีเหตุการณ์ไฟไหม้

3) ผลการทดสอบระบบเมื่อมีการโจรกรรม

4) ผลการทดสอบระบบเมื่อมีการโจรกรรมและเกิดเหตุไฟไหม้พร้อมกัน

5) ผลการทดสอบสัญญาณขอความช่วยเหลือ

6) ผลการทดสอบหน้าต่างเก็บประวัติเหตุการณ์ผิดปกติทั้งหมดที่เกิดขึ้น

7) ผลการทดสอบหน้าต่างแสดงสถานะของอุปกรณ์

3.3.7 ผลการออกแบบฐานข้อมูล

1) ผลการออกแบบฐานข้อมูลที่ชื่อว่า “security”

2) ผลการออกแบบฐานข้อมูลที่ชื่อว่า “logfile”

3.3.8 การออกแบบในส่วนของการแจ้งเตือนผ่านแอปพลิเคชั่นไลน์

1) ผลการแจ้งเตือนผ่านแอพพลิเคชั่นไลน์เมื่อเกิดอัคคีภัย

2) ผลการแจ้งเตือนผ่านแอพพลิเคชั่นไลน์เมื่อเกิดการโจรกรรม

3) ผลการแจ้งเตือนผ่านแอพพลิเคชั่นไลน์เมื่อสามารถยุติเหตุการณ์ต่างๆที่เกิดขึ้นได้แล้ว

บทที่ 4

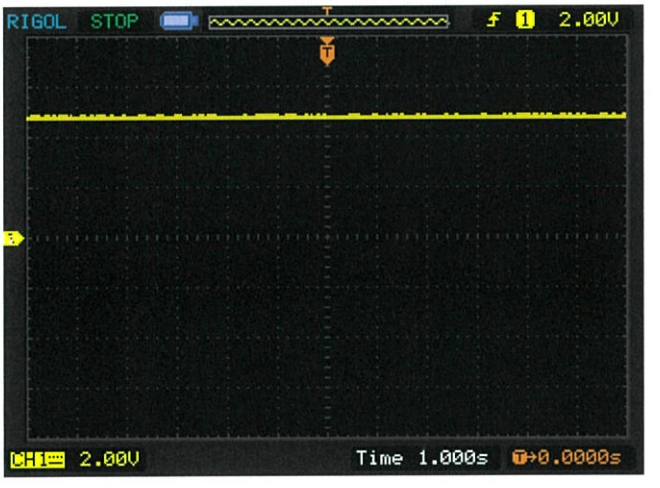
ผลการทดลอง

4.1 ผลการทดสอบเซนเซอร์ตรวจจับเปลวไฟเบื้องต้น

เซนเซอร์ตรวจจับเปลวไฟใช้ในการตรวจจับรังสีอินฟราเรดจากแหล่งกำเนิดความร้อนต่างๆ เช่น เปลวไฟ โมดูลเซนเซอร์ชนิดนี้ใช้ไฟเลี้ยง 3.3 - 5 โวลต์

4.1.1 ผลการทดสอบเซนเซอร์ตรวจจับเปลวไฟเมื่อไม่มีการตรวจจับเปลวไฟ

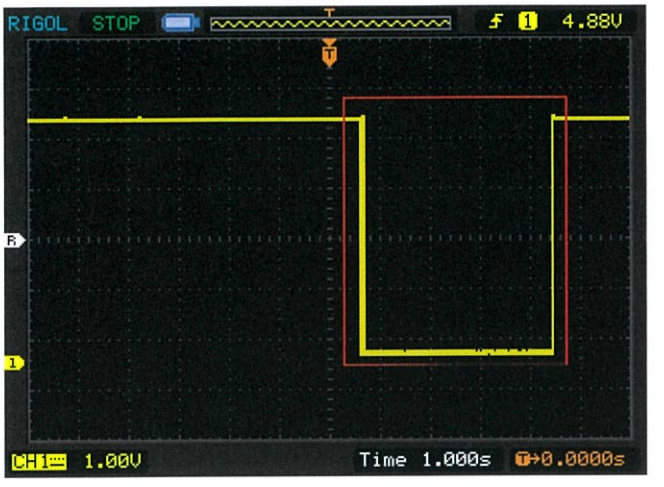
เมื่อไม่มีการตรวจจับเปลวไฟได้ขาเอาต์พุตแบบดิจิตอลจะมีแรงดันไฟฟ้าเท่ากับไฟเลี้ยง ซึ่งมีค่าเท่ากับ 4.8 โวลต์ ดังรูปที่ 4.1



รูปที่ 4.1 สัญญาณเอาต์พุตในกรณีที่ไม่มีการตรวจจับเปลวไฟ

4.1.2 ผลการทดสอบเซนเซอร์ตรวจจับเปลวไฟเมื่อมีการตรวจจับเปลวไฟได้

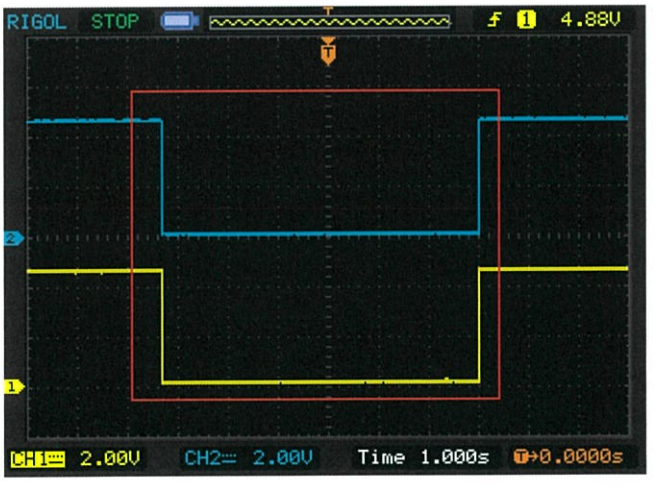
เซนเซอร์ตรวจจับเปลวไฟใช้ไฟเลี้ยง 3.3 – 5 โวลต์ เมื่อไม่มีการตรวจจับเปลวไฟได้ที่ขาเอาต์พุตแบบดิจิตอลจะมีแรงดันไฟฟ้าเท่ากับศูนย์ ดังกรอบสีแดงในรูปที่ 4.2 จะเห็นว่าเมื่อตรวจจับเปลวไฟได้จะมีแรงดันเอาต์พุตเท่ากับศูนย์ แต่เมื่อตรวจไม่พบเปลวไฟก็จะกลับไปสู่สภาวะที่ไม่มีการตรวจพบเปลวไฟคือแรงดันมีค่าท่ากับไฟเลี้ยงหรือ 4.8 โวลต์



รูปที่ 4.2 สัญญาณเอาต์พุตในกรณีที่ตรวจจับเปลวไฟได้

4.1.3 ผลการทดสอบเซนเซอร์ตรวจจับเปลวไฟ 2 ตัวพร้อมกัน

เมื่อทดสอบเซนเซอร์ตรวจจับเปลวไฟ 2 ตัวพร้อมกันโดยให้ตรวจจับเปลวไฟพร้อมกัน ปรากฏว่าเซนเซอร์ตรวจจับเปลวไฟมีความเสถียรและมีความไวต่อการเปลี่ยนแปลงต่อเปลวไฟเท่าๆ กันดังกรอบสีแดง ดังในรูปที่ 4.3



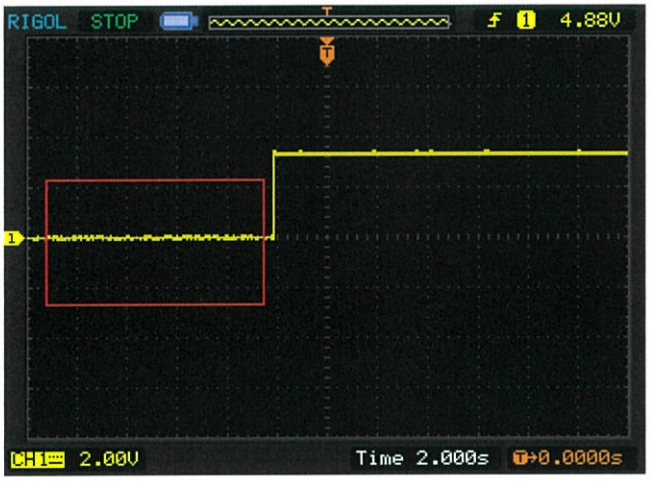
รูปที่ 4.3 สัญญาณเอาต์พุตของเซนเซอร์ตรวจจับเปลวไฟ 2 ตัว

4.2 ผลการทดสอบเซนเซอร์ตรวจจับการเคลื่อนไหว

เซนเซอร์ตรวจจับการเคลื่อนไหวใช้ไฟเลี้ยง 3 - 5 โวลต์ ใช้สำหรับตรวจจับการเคลื่อนไหวของมนุษย์หรือสิ่งมีชีวิต โดยใช้หลักการการตรวจจับอินฟราเรดจากวัตถุผ่านยังเลนส์เพื่อรวมแสงแล้วโฟกัสมายังไพโรอิเล็กตริก ซึ่งเป็นตัวแปลงพลังงานความร้อนจากรังสีอินฟราเรดเป็นพลังงานไฟฟ้า

4.2.1 ผลการทดสอบเซนเซอร์ตรวจจับการเคลื่อนไหวเมื่อไม่มีการตรวจจับการเคลื่อนไหว

เมื่อไม่มีการตรวจจับการเคลื่อนไหว เซนเซอร์ตรวจจับการเคลื่อนไหวจะจ่ายแรงดันออกมาที่ขาเอาต์พุตเท่ากับศูนย์ดังกรอบสีแดง ดังในรูปที่ 4.4



รูปที่ 4.4 สัญญาณเอาต์พุตของเซนเซอร์ตรวจจับการเคลื่อนไหวเมื่อไม่มีการตรวจจับการเคลื่อนไหว

4.2.2 ผลการทดสอบเซนเซอร์ตรวจจับการเคลื่อนไหวเมื่อมีการตรวจจับเคลื่อนไหว

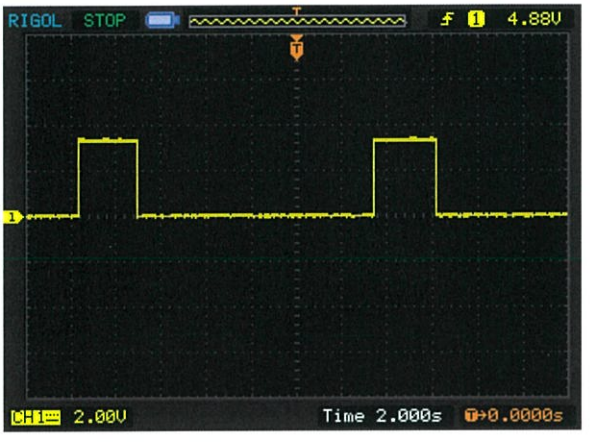
เซนเซอร์ตรวจจับการเคลื่อนไหวมี 2 โหมดการทำงานคือ โหมด L และ โหมด H

1. โหมด L (Non - Retriggering) เมื่อตรวจจับการเคลื่อนไหวได้ เซนเซอร์ตรวจจับการเคลื่อนไหวจะส่งแรงดันเอาต์พุตออกมาเท่ากับไฟเลี้ยงและจะคงค่าไว้ช่วงเวลาหนึ่งและจะเปลี่ยนสถานะกลับตามเดิม (แรงดันเอาต์พุตเป็นศูนย์) แม้ว่าจะมีการเคลื่อนไหวอยู่ก็ตาม และจะเข้าสู่สถานะไม่รับรู้การเคลื่อนไหวชั่วขณะหนึ่งจึงจะกลับมาสภาวะปกติ

2. โหมด H (Retriggering) เมื่อเซนเซอร์ตรวจจับการเคลื่อนไหวสามารถตรวจจับการเคลื่อนไหวได้จะส่งแรงดันเอาต์พุตออกมาเท่ากับไฟเลี้ยงและค้างสถานะเอาไว้จนกว่าจะไม่มีการเคลื่อนไหวจึงจะเปลี่ยนสถานะกลับไปยังสถานะเดิม (แรงดันเอาต์พุตเป็นศูนย์) และจะเข้าสู่ช่วงไม่ตอบสนองการเคลื่อนไหวใดๆ ชั่วขณะหนึ่ง

4.2.2.1 การทดสอบเซนเซอร์ตรวจจับการเคลื่อนไหวเมื่อมีการตรวจจับการเคลื่อนไหวในโหมด L

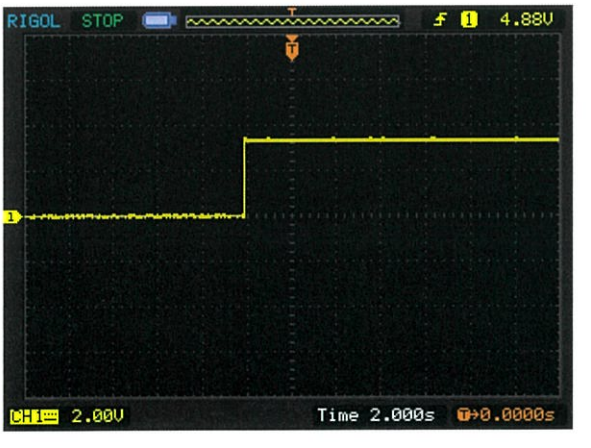
เมื่อตรวจจับการเคลื่อนไหวได้ขาเอาต์พุตจะคงค่าแรงดันเท่ากับ 4.8 โวลต์ เป็นระยะเวลา 2.6 วินาที แล้วจึงจ่ายแรงดันศูนย์ออกมาเป็นช่วงเวลาหนึ่งจึงสามารถตรวจจับการเคลื่อนไหวได้อีกครั้งหนึ่งดังรูปที่ 4.5



รูปที่ 4.5 สัญญาณเอาต์พุตของเซนเซอร์ตรวจจับการเคลื่อนไหว
เมื่อมีการตรวจจับการเคลื่อนไหวในโหมด L

4.2.2.2 การทดสอบเซนเซอร์ตรวจจับการเคลื่อนไหวเมื่อมีการตรวจจับเคลื่อนไหวในโหมด H

เมื่อตรวจจับการเคลื่อนไหวได้ขาเอาต์พุตจะค้างค่าแรงดันเท่ากับ 4.8V จนกว่าจะไม่มีการตรวจจับการเคลื่อนไหวแล้วจึงจ่ายแรงดันศูนย์ออกมาดังรูปที่ 4.6



รูปที่ 4.6 สัญญาณเอาต์พุตของเซนเซอร์ตรวจจับการเคลื่อนไหว
เมื่อมีการตรวจจับเคลื่อนไหวในโหมด H

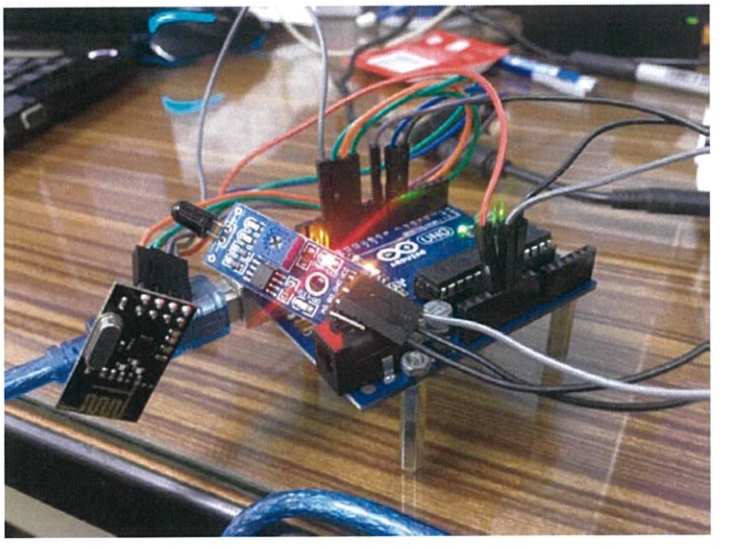
4.2.3 ผลการทดสอบระยะการตรวจจับของเซนเซอร์ตรวจจับการเคลื่อนไหว

จากการทดลองวัดระยะการตรวจจับและทดสอบมุมในการตรวจจับพบว่าระยะทางสูงสุดที่ตรวจจับได้มีค่าเฉลี่ยเท่ากับ 15.73 เมตร และมุมที่สามารถตรวจจับการเคลื่อนไหวสูงสุดด้านหน้าเซนเซอร์มีค่าเฉลี่ยเท่ากับ 143 องศา

4.3 ผลการทดสอบในส่วนของระบบการรับส่งข้อมูลไร้สายในรูปแบบดิจิตอล

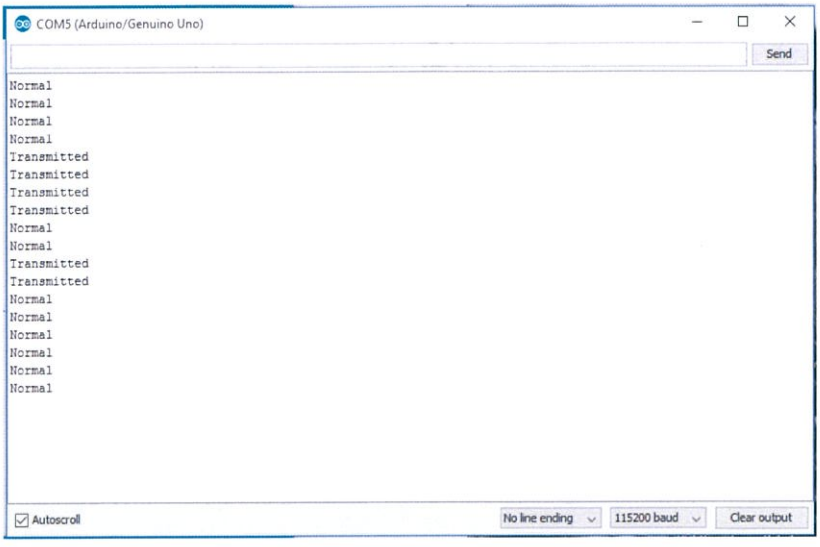
4.3.1 ผลการทดสอบโมดูลสื่อสารไร้สาย NRF24L01 ด้านส่ง

ที่ด้านส่งคือด้านที่รับค่าจากเซนเซอร์เพื่อส่งต่อไปที่หน่วยประมวลผลกลาง จะต่อตัวเซนเซอร์ตรวจจับเปลวไฟและโมดูลสื่อสารไร้สาย NRF24L01 เข้ากับไมโครคอนโทรลเลอร์ ดังรูปที่ 4.7



รูปที่ 4.7 การทดสอบโมดูลสื่อสารไร้สาย NRF24L01 ด้านส่ง

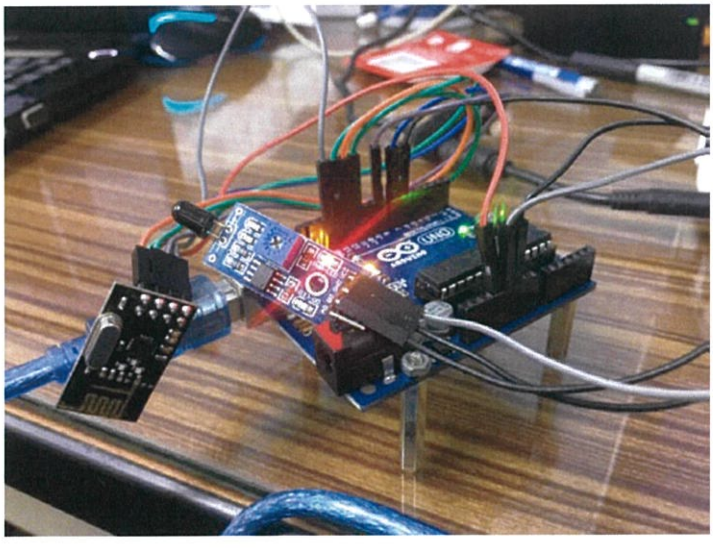
เมื่อเหตุการณ์ปกติจะให้ส่งค่า “0” ไปที่หน่วยประมวลผลกลางและแสดงผลบนหน้า Serial Monitor ว่า “Normal” ส่วนเมื่อสามารถตรวจจับเหตุการณ์ผิดปกติได้ จะส่งค่า “1” ไปที่หน่วยประมวลผลกลางและแสดงผลบนหน้า Serial Monitor ว่า “Transmitted” ดังรูปที่ 4.8



รูปที่ 4.8 Serial Monitor ของด้านส่ง

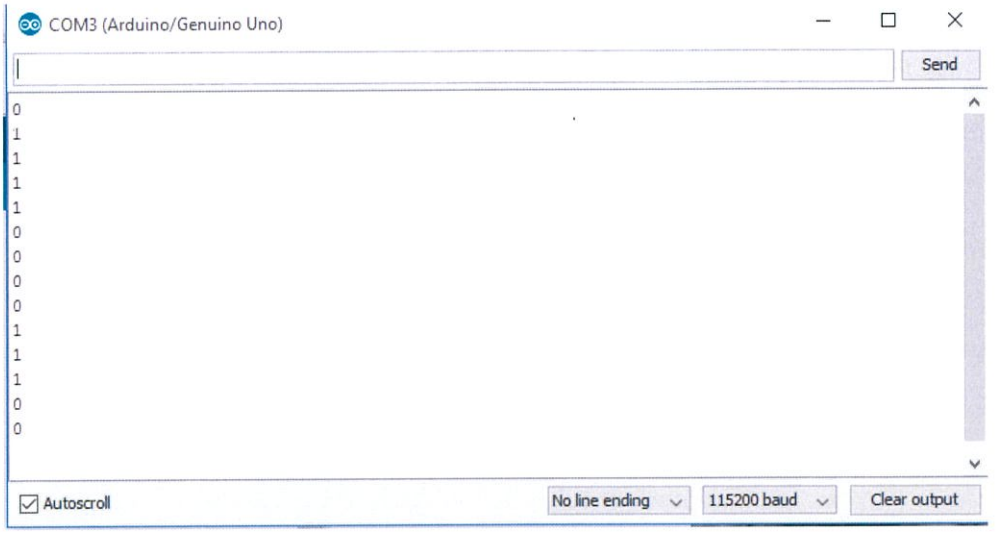
4.3.2 ผลการทดสอบโมดูลสื่อสารไร้สาย NRF24L0 ด้านรับ

ที่ด้านรับนำโมดูลสื่อสารไร้สาย NRF24L01 มาต่อเข้ากับไมโครคอนโทรลเลอร์แล้วดูค่าที่รับได้ผ่าน Serial Monitor ดังรูปที่ 4.9



รูปที่ 4.9 การทดสอบโมดูล NRF24L01 ด้านรับ

เมื่อแสดงค่าที่รับได้จากด้านส่งบน Serial Monitor พบว่า จะแสดงค่าเป็น “0” , “1” ซึ่งหมายถึงเหตุการณ์ปกติและเหตุการณ์ที่สามารถตรวจจับเปลวไฟได้ตามลำดับดังรูปที่ 4.10

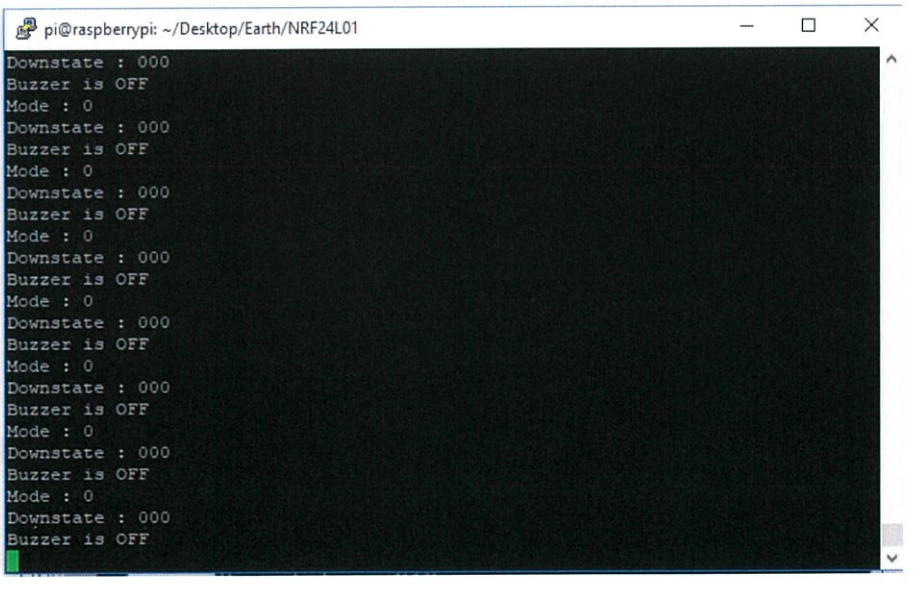


รูปที่ 4.10 Serial Monitor ของด้านรับ

4.4 ผลการทดสอบระบบในส่วนของราสป์เบอร์รีไพ

4.4.1 ผลการทดสอบระบบในส่วนของราสป์เบอร์รีไพเมื่อไม่มีเหตุการณ์ผิดปกติ

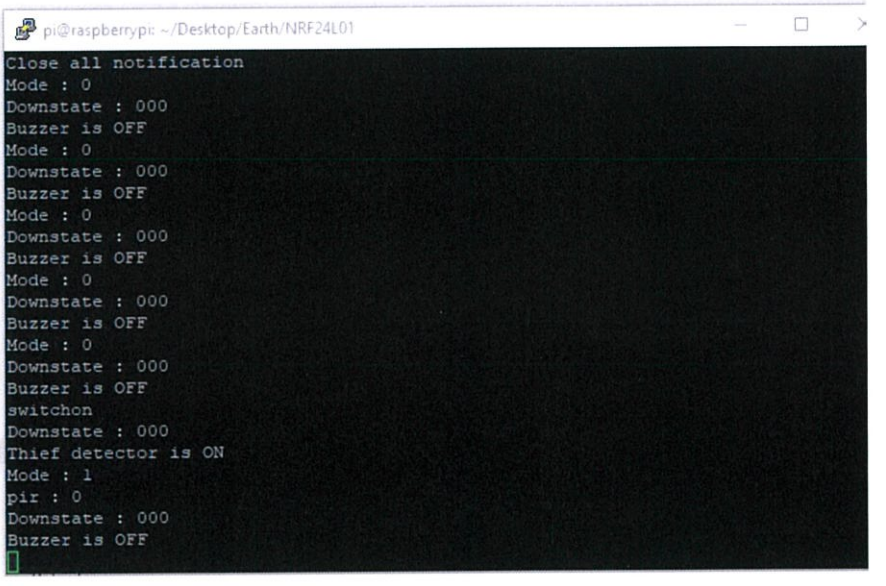
เมื่อไม่มีเหตุเพลิงไหม้หรือเหตุโจรกรรมจะแสดงคำว่า “Buzzer is OFF” ซึ่งหมายถึงไม่มีการแจ้งเตือนของสัญญาณเสียง ดังในรูปที่ 4.11



รูปที่ 4.11 หน้าต่าง Python Shell ของราสป์เบอร์รีไพ เมื่อไม่มีเหตุการณ์ผิดปกติ

4.4.2 ผลการทดสอบระบบในส่วนของราสป์เบอร์รีไพเมื่อกดรีโมทคอนโทรลเพื่อเปิดระบบป้องกันการโจรกรรม

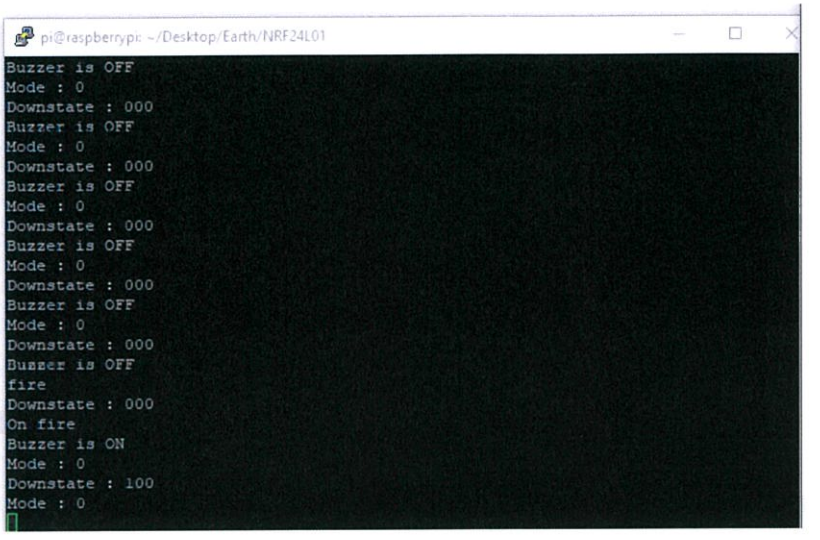
เมื่อไม่มีผู้อาศัยอยู่ภายในบ้าน และต้องการให้ระบบป้องกันการโจรกรรมทำงาน จะต้องทำการกดสวิตช์ที่รีโมทคอนโทรลเพื่อสั่งการให้ระบบป้องกันการโจรกรรมเริ่มทำงาน ซึ่งเมื่อกดสวิตช์ที่รีโมท จะมีการแสดงค่า “switchon” บนหน้าต่าง Python Shell ซึ่งหมายถึงการสั่งการให้ระบบป้องกันการโจรกรรมเริ่มทำงาน จากนั้นบนหน้าต่าง Python Shell จะแสดงคำว่า “Thief detector is ON” ซึ่งแสดงว่าระบบป้องกันการโจรกรรมเริ่มทำงานแล้ว ดังในรูปที่ 4.12



รูปที่ 4.12 หน้าต่าง Python Shell ของ ราสป์เบอร์รีไพ
เมื่อกดรีโมทเพื่อเปิดระบบป้องกันการโจรกรรม

4.4.3 ผลการทดสอบระบบในส่วนของราสป์เบอร์รีไพเมื่อมีเหตุเพลิงไหม้

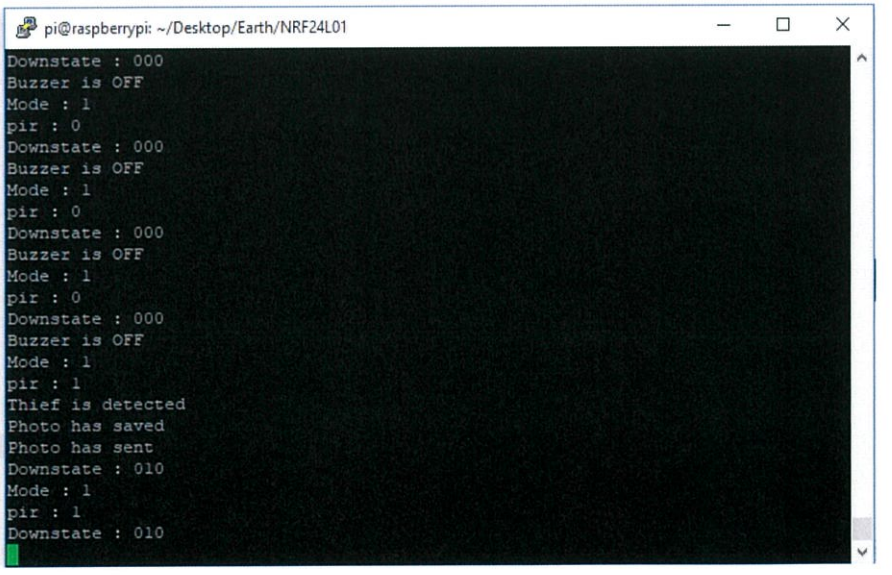
เมื่อมีเหตุเพลิงไหมจะส่งข้อมูลไร้สายผ่านโมดูลสื่อสารไร้สาย NRF24L01 ซึ่งเมื่อได้รับจะแสดงคำว่า fire บนหน้าต่าง Python Shell หลังจากนั้นจึงมีการจ่ายแรงดันออกทางพอร์ตที่ต่อกับ Buzzer ทำให้มีเสียงแจ้งเตือนดังออกมาและส่งสถานะตรวจจับเหตุเพลิงไหม้ได้ผ่านอินเตอร์เน็ตไปที่ฐานข้อมูล ดังในรูปที่ 4.13



รูปที่ 4.13 หน้าต่าง Python Shell ของราสป์เบอร์รีไพเมื่อมีเหตุเพลิงไหม้

4.4.4 ผลการทดสอบระบบในส่วนของราสป์เบอร์รีไพเมื่อมีเหตุโจรกรรม

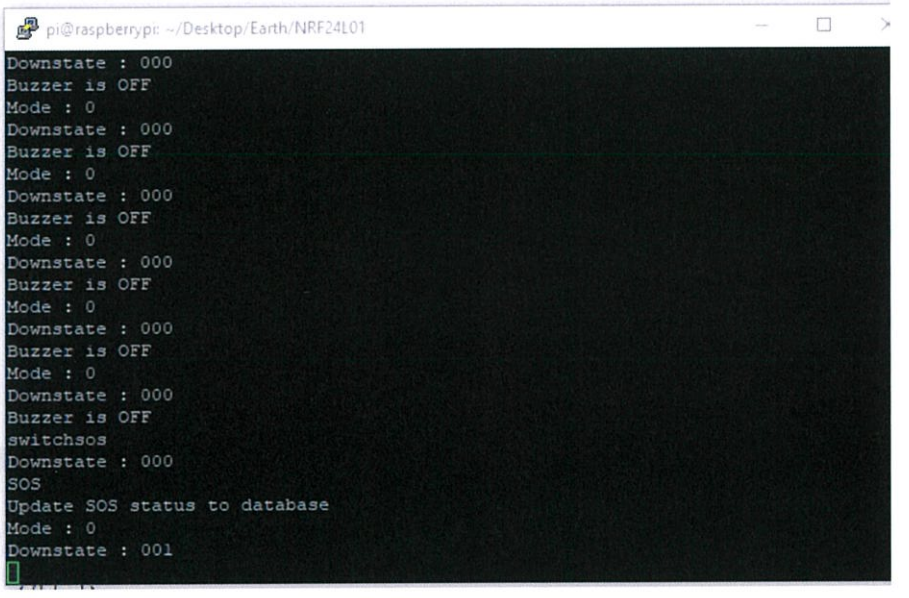
เมื่อมีเหตุโจรกรรมเซนเซอร์ตรวจจับการเคลื่อนไหวจะส่งข้อมูลไร้สายผ่านโมดูลสื่อสารไร้สาย NRF24L01 เมื่อราสป์เบอร์รีไพรับค่าได้จะแสดงคำว่า “Thief is detected” บนหน้าต่าง Python Shell จากนั้นราสป์เบอร์รีไพจะสั่งการให้กล้องบันทึกข้อมูลภาพในขณะนั้น เมื่อบันทึกข้อมูลเสร็จบนหน้าต่าง Python Shell จะแสดงคำว่า “Photo has saved” และถูกส่งออกไป เมื่อส่งเสร็จจะแสดงคำว่า “Photo has sent” บนหน้าต่าง Python Shell หลังจากนั้นจึงมีการจ่ายแรงดันออกทางพอร์ตที่ต่อกับ Buzzer ทำให้มีเสียงแจ้งเตือนดังออกมาและส่งสถานะตรวจจับเหตุเพลิงไหม้ได้ผ่านอินเตอร์เน็ตไปที่ฐานข้อมูล ดังในรูปที่ 4.14



รูปที่ 4.14 Python Shell ของราสป์เบอร์รีไพเมื่อมีเหตุโจรกรรม

4.4.5 ผลการทดสอบระบบในส่วนของราสป์เบอร์รีไพเมื่อกดรีโมทคอนโทรลเพื่อส่งสัญญาณขอความช่วยเหลือ

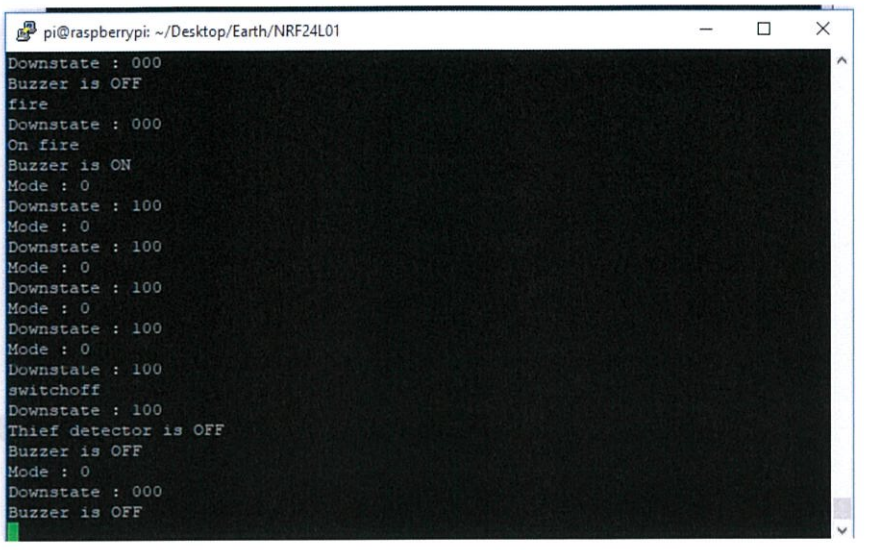
เมื่อต้องการความช่วยเหลือและกดสวิตช์บนรีโมทจะส่งข้อมูลไร้สายผ่านโมดูล NRF24l01 ไปที่ราสป์เบอร์รีไพ บนหน้าต่าง Python Shell จะแสดงคำว่า “switchsos” หลังจากนั้นจะมีการอัพเดตค่า “SOS” ไปที่ฐานข้อมูล เพื่อแจ้งเตือนให้เจ้าหน้าที่รักษาความปลอดภัยได้รับรู้และเข้ามาช่วยเหลือทันเวลา ดังในรูปที่ 4.15



รูปที่ 4.15 หน้าต่าง Python Shell ของราสป์เบอร์รีไพเมื่อกดรีโมทเพื่อส่งสัญญาณขอความช่วยเหลือ

4.4.6 ผลการทดสอบระบบในส่วนของราสป์เบอร์รีไพเมื่อกดรีโมทคอนโทรลเพื่อหยุดเสียงเตือน

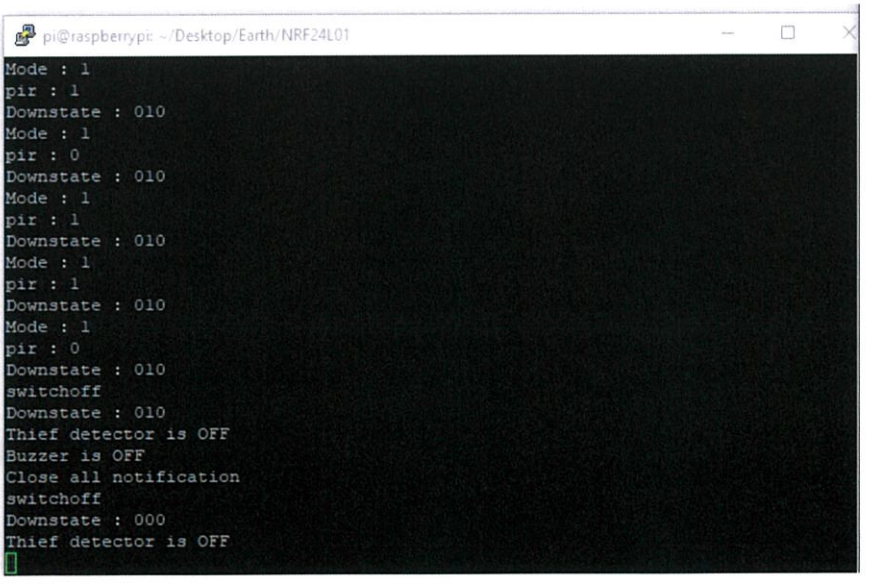
เมื่อเกิดเหตุการณ์ที่ผิดปกติขึ้น และสามารถยุติเหตุการณ์นั้นๆได้แล้ว จะใช้การกดสวิตช์บนรีโมทเพื่อหยุดเสียงเตือน ซึ่งเมื่อกดสวิตช์ราสป์เบอร์รีไพจะแสดงคำว่า “switchoff” จากนั้นระบบจะทำการส่งค่าไปยังฐานข้อมูลและปิดเสียงแจ้งเตือนลง ดังรูปที่ 4.16



รูปที่ 4.16 หน้าต่าง Python Shell ของราสป์เบอร์รีไพเมื่อกดรีโมทคอนโทรลเพื่อหยุดเสียงเตือน

4.4.7 ผลการทดสอบระบบในส่วนของราสป์เบอร์รีไพเมื่อกดรีโมท เพื่อปิดระบบป้องกันการโจรกรรม

เมื่อมีคนอยู่บ้านและไม่ต้องการให้ระบบป้องกันการโจรกรรมทำงานจะใช้การกดสวิตช์บนรีโมทเพื่อสั่งการให้ระบบป้องกันการโจรกรรมนั้น หยุดทำงานลงชั่วคราว เมื่อราสป์เบอร์รีไพได้รับค่า จะแสดงคำว่า “Thief detector is OFF” และ “Buzzer is OFF” บนหน้าต่าง Python Shell จากนั้นจะแสดงคำว่า “Close all notification” เป็นการบอกว่าระบบป้องกันการโจรกรรมได้หยุดทำงานอย่างชั่วคราวแล้ว ซึ่งสามารถทำการเปิดใช้งานระบบได้ใหม่เมื่อกดสวิตช์เปิดการทำงานระบบป้องกันการโจรกรรมบนรีโมท ดังรูปที่ 4.17



รูปที่ 4.17 หน้าต่าง Python Shell ของราสป์เบอร์รีไพเมื่อกดรีโมทเพื่อปิดระบบป้องกันการโจรกรรม

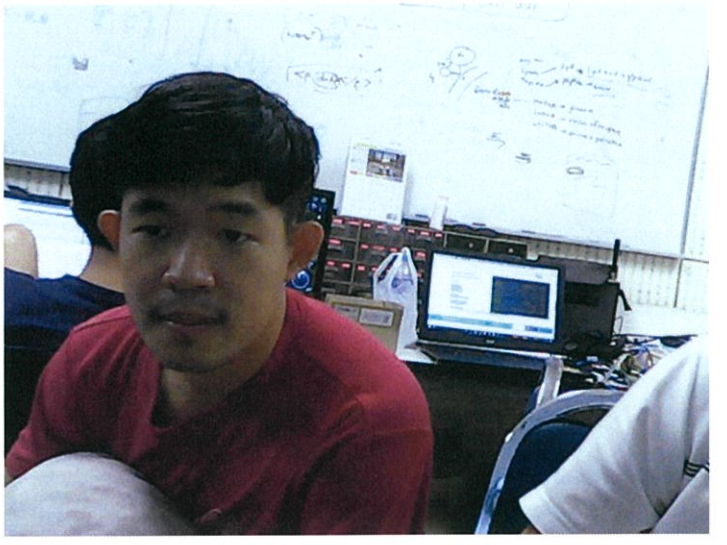
4.5 ผลการทดสอบในส่วนของการใช้งานกล้องเว็บแคม

4.5.1 ผลการทดสอบกล้องเว็บแคม โดยใช้ภาษา python สั่งการกล้องให้ถ่ายภาพ

เมื่อนำกล้องเว็บแคม มาเชื่อมต่อกับราสป์เบอร์รีไพจะสามารถเขียนโปรแกรมโดยใช้ภาษา python เพื่อใช้ในการควบคุมกล้องเว็บแคม ให้ถ่ายภาพได้ตามคุณสมบัติที่เรากำหนดได้ ดังรูปที่ 4.18

รูปที่ 4.18 เชื่อมต่อราสป์เบอร์รีไพกับกล้องเว็บแคม

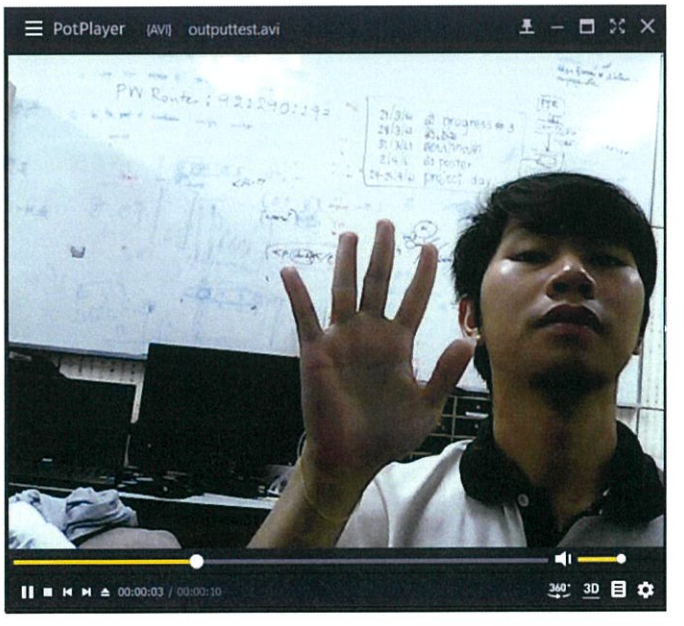
เมื่อสั่งการให้กล้องถ่ายรูป โดยกำหนดความละเอียดของภาพ ชนิดของไฟล์ภาพ ซึ่งทดลองให้ความละเอียดของภาพอยู่ที่ 640x480 พิกเซล ชนิดของนามสกุลของไฟล์เป็น .jpg จะได้ไฟล์ภาพ ดังรูปที่ 4.19



รูปที่ 4.19 ภาพถ่ายจากกล้องเว็บแคม

4.5.2 ผลการทดสอบกล้องเว็บแคม โดยใช้ภาษา python สั่งการกล้องให้ถ่ายวีดีโอ

นอกจากจะใช้ภาษา python ในการเขียนโปรแกรมสั่งการให้กล้องถ่ายรูปภาพได้แล้ว ยังสามารถเขียนโปรแกรมให้บันทึกวีดีโอได้อีกด้วย ซึ่งสามารถกำหนดความละเอียดของไฟล์วีดีโอ ชนิดของไฟล์วีดีโอและระยะเวลาที่ใช้อัดได้ โดยให้ความละเอียดของวีดีโอเป็น 640x480 พิกเซล ชนิดของวีดีโอเป็น .avi ความยาวของวีดีโอคือ 10 วินาที ซึ่งไฟล์วีดีโอที่ได้สามารถเปิดใช้ได้กับโปรแกรมเล่นวีดีโอทั่วไปได้ ดังในรูปที่ 4.20



รูปที่ 4.20 ไฟล์วีดีโอจากกล้องเว็บแคม

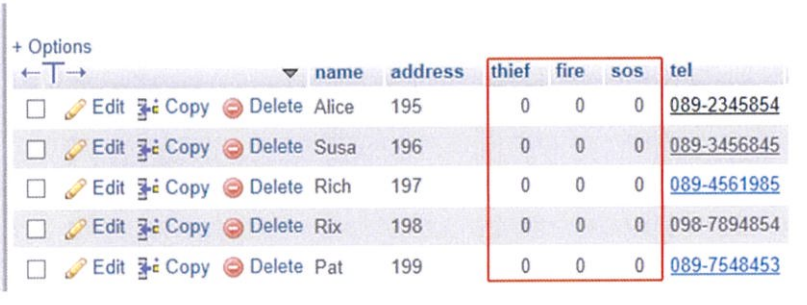
4.6 ผลการทดสอบในส่วนของหน้าจอมอนิเตอร์

4.6.1 ผลการทดสอบระบบเมื่อเหตุการณ์เป็นปกติ

เมื่อนำระบบมาทำการทดสอบเพื่อแสดงผลปรากฏว่าหน้าจอมอนิเตอร์ไม่แสดงข้อมูลของบ้านที่เกิดเหตุการณ์ผิดปกติแสดงดังรูปที่ 4.21

รูปที่ 4.21 หน้าจอแสดงผลเมื่อเหตุการณ์ปกติ

กรณีที่เหตุการณ์เป็นปกติข้อมูลที่อยู่ในฐานข้อมูลจะมีค่าเป็น “0” ในคอลัมน์ที่มีชื่อเป็น “fire” , “thief” และ “sos” ดังที่แสดงในรูปที่ 4.22

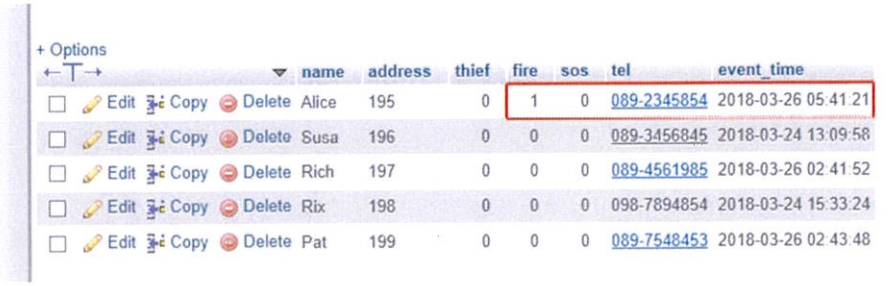


รูปที่ 4.22 ข้อมูลในฐานข้อมูลเมื่อเหตุการณ์ปกติ

4.6.2 ผลการทดสอบระบบเมื่อมีเหตุการณ์ไฟไหม้

4.6.2.1 ผลการทดสอบระบบเมื่อมีเหตุการณ์ไฟไหม้บ้าน 1 หลัง

ถ้าในกรณีที่ข้อมูลในฐานข้อมูลของคอลัมน์ที่มีชื่อว่า “fire” เกิดมีการเปลี่ยนแปลงข้อมูลจาก “0” ไปเป็น “1” แสดงว่าเซนเซอร์ตรวจจับเปลวไฟที่ถูกติดตั้งอยู่ที่บ้านหลังนั้นสามารถตรวจจับเปลวไฟได้ข้อมูลการเปลี่ยนแปลงจะถูกส่งเข้าไปยังฐานข้อมูลโดยไมโครคอนโทรลเลอร์ พร้อมทั้งจะส่งค่าเวลาที่เกิดเหตุการณ์นั้นไปด้วยด้วยดังตัวอย่างที่แสดงในรูปที่ 4.23



+ Options

	name	address	thief	fire	sos	tel	event_time
Edit Copy Delete	Alice	195	0	1	0	089-2345854	2018-03-26 05:41:21
Edit Copy Delete	Susa	196	0	0	0	089-3456845	2018-03-24 13:09:58
Edit Copy Delete	Rich	197	0	0	0	089-4561985	2018-03-26 02:41:52
Edit Copy Delete	Rix	198	0	0	0	098-7894854	2018-03-24 15:33:24
Edit Copy Delete	Pat	199	0	0	0	089-7548453	2018-03-26 02:43:48

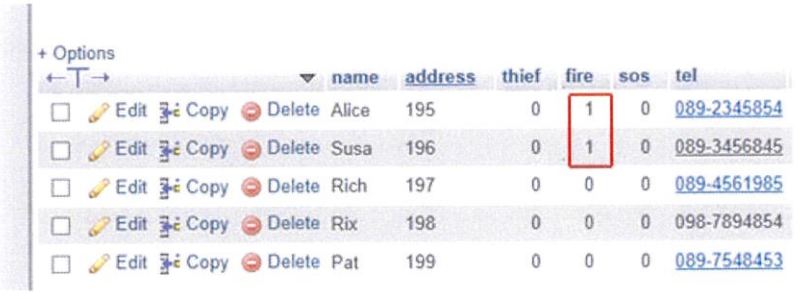
รูปที่ 4.23 ฐานข้อมูลเมื่อมีเหตุการณ์ไฟไหม้บ้านของ Alice

จากรูปที่ 4.23 จะเห็นได้ว่าบ้านของ “Alice” ที่ค่าของคอลัมน์ “fire” เปลี่ยนจาก “0” เป็น “1” และมีข้อมูลของวันที่และเวลาที่เกิดเหตุการณ์ถูกส่งขึ้นไปยังฐานข้อมูล ข้อมูลนี้จะถูกนำไปแสดงผลต่อที่หน้าจอมอนิเตอร์และแจ้งเตือนด้วยเสียงดังตัวอย่างในรูปที่ 4.24

รูปที่ 4.24 หน้าจอมอนิเตอร์เมื่อบ้านเลขที่ “195” มีเหตุการณ์ไฟไหม้

4.6.2.2 ผลการทดสอบระบบเมื่อมีเหตุการณ์ไฟไหม้ 2 ที่พร้อมกัน

กรณีที่ข้อมูลในฐานข้อมูลคอลัมน์ที่มีชื่อว่า “fire” เกิดมีการเปลี่ยนแปลงค่าจาก “0” ไปเป็น “1” ทั้งสองแถวแสดงว่ามีเหตุการณ์ผิดปกติอันเนื่องมาจากตรวจพบเปลวไฟ 2 บ้านดังตัวอย่างที่แสดงในรูปที่ 4.25

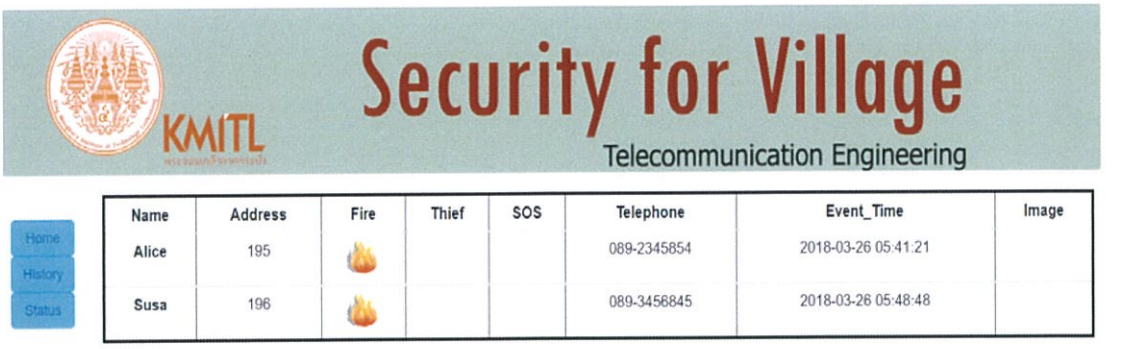


+ Options

	name	address	thief	fire	sos	tel
Edit Copy Delete	Alice	195	0	1	0	089-2345854
Edit Copy Delete	Susa	196	0	1	0	089-3456845
Edit Copy Delete	Rich	197	0	0	0	089-4561985
Edit Copy Delete	Rix	198	0	0	0	098-7894854
Edit Copy Delete	Pat	199	0	0	0	089-7548453

รูปที่ 4.25 ฐานข้อมูลเมื่อมีเหตุการณ์ไฟไหม้บ้านของ “Alice” และ “Susa”

จากรูปที่ 4.25 จะเห็นได้ว่าทั้งบ้านของ “Alice” และบ้านของ “Susa” มีข้อมูลเปลี่ยนจาก “0” เป็น “1” ในคอลัมน์ “fire” ทั้ง 2 แถวและข้อมูลนี้จะถูกนำไปแสดงผลต่อที่หน้ามอนิเตอร์ทั้ง 2 บ้านพร้อมกันและแจ้งเตือนด้วยเสียงดังตัวอย่างในรูปที่ 4.26



รูปที่ 4.26 หน้าจอมอนิเตอร์เมื่อมีเหตุการณ์ไฟไหม้บ้าน 2 หลังพร้อมกัน

4.6.3 ผลการทดสอบระบบเมื่อมีการโจรกรรม

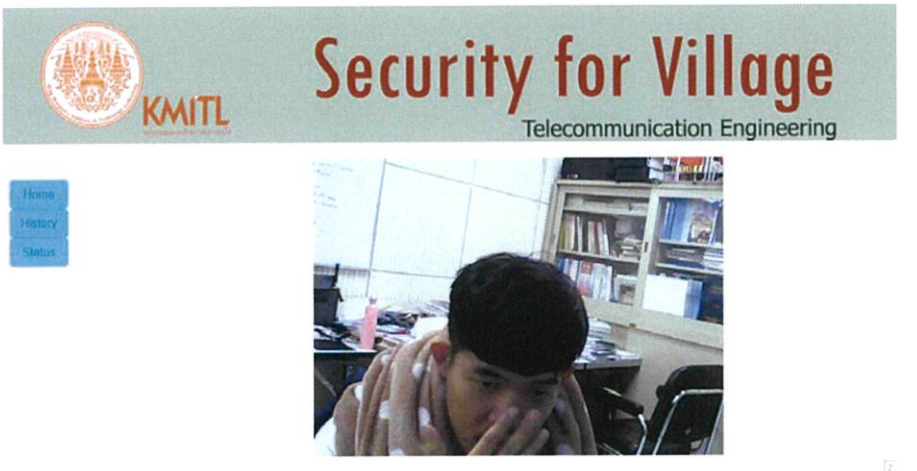
4.6.3.1 ผลการทดสอบระบบเมื่อมีการโจรกรรมบ้าน 1 หลัง

ในกรณีที่เกิดเหตุการณ์ผิดปกติเนื่องมาจาก เซนเซอร์ตรวจจับการเคลื่อนไหว สามารถตรวจจับการเคลื่อนไหวได้ ระบบจะทำการส่งข้อมูลการเปลี่ยนแปลงไปยังในตารางของฐานข้อมูลที่มีชื่อว่า “thief” โดยข้อมูลที่เปลี่ยนแปลงจะเปลี่ยนจาก “0” ไปเป็น “1” และจะส่งเวลาที่เกิดเหตุการณ์ไปยังฐานข้อมูล นอกจากนี้ระบบจะใช้งานกล้องให้บันทึกรูปภาพเหตุการณ์ในขณะนั้นและจัดเก็บชื่อของไฟล์ของข้อมูลภาพเพื่อนำมาแสดงหน้าจอมอนิเตอร์ต่อไป ดังตัวอย่างที่แสดงในรูปที่ 4.27

รูปที่ 4.27 ฐานข้อมูลเมื่อมีผู้บุกรุกบ้านของ “Rich”

จะเห็นได้ว่าข้อมูลในฐานข้อมูลของบ้าน “Rich” คอลัมน์ที่ชื่อว่า “thief” จะมีการเปลี่ยนแปลงค่าพื้นฐานจากที่เป็น “0” กลายเป็น “1” จากนั้นกล้องที่ติดตั้งไว้จะทำการบันทึกภาพ ณ ขณะนั้นๆไว้ในราสป์เบอร์รีไพโดยจะส่งไฟล์ข้อมูลภาพไปที่เซิร์ฟเวอร์และส่งชื่อของข้อมูลภาพไปยังฐานข้อมูลซึ่งที่หน้าจอมอนิเตอร์ จะมีการแจ้งเตือน พร้อมทั้งมีปุ่มสำหรับใช้กดเรียกดูข้อมูลภาพบนหน้าจอมอนิเตอร์ ดังตัวอย่างรูปที่ 4.28 – 4.29

รูปที่ 4.28 หน้าจอมอนิเตอร์เมื่อมีการโจรกรรมบ้านของ “Rich”



รูปที่ 4.29 หน้าจอมอนิเตอร์หลังจากการกดเข้าไปดูข้อมูลภาพของเหตุการณ์

4.6.3.2 ผลการทดสอบระบบเมื่อมีการโจรกรรม 2 ที่พร้อมกัน

ถ้าในกรณีที่ข้อมูลภายในหลักของฐานข้อมูลที่มีชื่อว่า “thief” เกิดมีการเปลี่ยนแปลงค่าจาก “0” ไปเป็น “1” ทั้งสองแถวแสดงว่ามีเหตุการณ์ผิดปกติที่เกิดจากตรวจพบการเคลื่อนไหวพร้อมกัน 2 บ้านดังตัวอย่างที่แสดงในรูปที่ 4.30

รูปที่ 4.30 ฐานข้อมูลเมื่อมีการโจรกรรม 2 ที่พร้อมกัน

จากฐานข้อมูลแสดงว่าบ้านของ “Rich” และ “Rix” มีการโจรกรรมพร้อมกัน หน้าจอมอนิเตอร์จะแสดงผลและแจ้งเตือนด้วยเสียงทั้ง 2 บ้านดังตัวอย่างในรูปที่ 4.31

รูปที่ 4.31 หน้าจอมอนิเตอร์เมื่อมีการโจรกรรม 2 ที่พร้อมกัน

4.6.4 ผลการทดสอบระบบเมื่อมีการโจรกรรมและเกิดเหตุไฟไหมพร้อมกัน

กรณีที่เกิดเหตุการณ์ผิดปกติ 2 แบบในเวลาเดียวกันคือเซนเซอร์ตรวจจับเปลวไฟตรวจจับเปลวไฟได้ และเซนเซอร์ตรวจจับการเคลื่อนไหวตรวจจับการเคลื่อนไหวได้ระบบจะส่งค่าการเปลี่ยนแปลงของข้อมูลของเซนเซอร์ทั้ง 2 ไปยังฐานข้อมูล ข้อมูลจะเปลี่ยนจากค่า “0” เป็น “1” ทั้งหลักของ “fire” และ “Thief” ดังตัวอย่างรูปที่ 4.32

รูปที่ 4.32 ฐานข้อมูลเมื่อมีการโจรกรรมและเหตุการณ์ไฟไหม้พร้อมกัน

ระบบจะนำข้อมูลจากฐานข้อมูลมาแสดงผลบนมอนิเตอร์จะเห็นว่าจะมีรูปเปลวไฟและรูปโจรทั้งในคอลัมน์ของ “fire” และ “thief” ดังรูปที่ 4.33 แล้วมีการแจ้งเตือนด้วยเสียง

รูปที่ 4.33 หน้าจอมอนิเตอร์เมื่อมีการโจรกรรมและเกิดเหตุไฟไหมพร้อมกัน

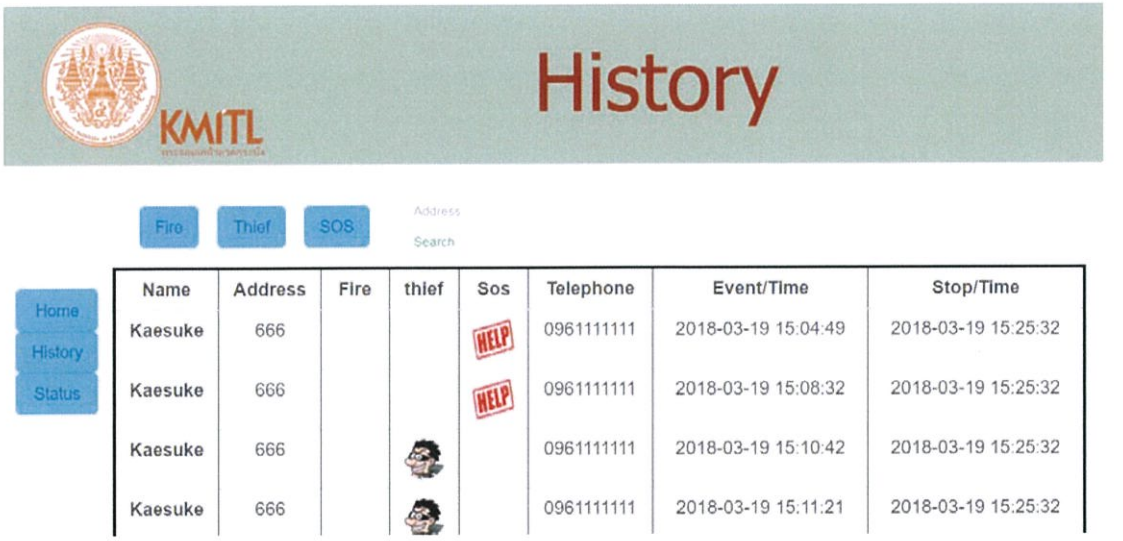
4.6.5 ผลการทดสอบสัญญาณขอความช่วยเหลือ

เมื่อเจ้าของบ้านพบกับเหตุการณ์อันตรายหรือต้องการความช่วยเหลือจากเจ้าหน้าที่รักษาความปลอดภัย เจ้าของบ้านสามารถกดปุ่มเพื่อร้องขอความช่วยเหลือจากเจ้าหน้าที่ได้ สัญญาณขอความช่วยเหลือจะไปแสดงอยู่ที่หน้ามอนิเตอร์ดังรูปที่ 4.34

รูปที่ 4.34 หน้าจอมอนิเตอร์เมื่อมีผู้ขอความช่วยเหลือ

4.6.6 ผลการทดสอบหน้าต่างเก็บประวัติเหตุการณ์ผิดปกติทั้งหมดที่เกิดขึ้น

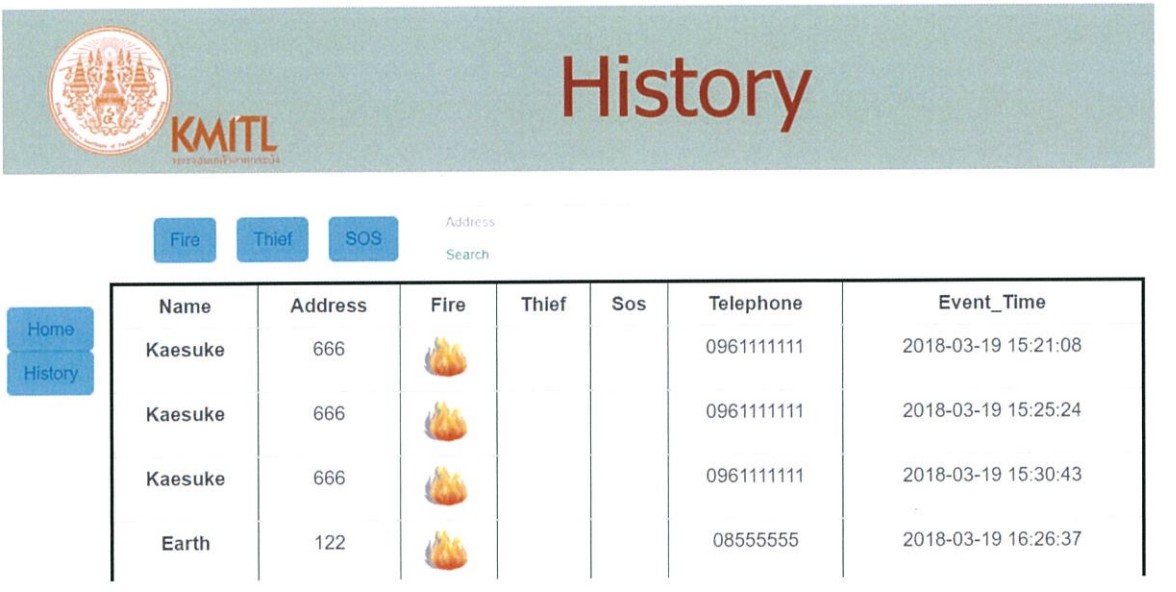
เมื่อมีการเกิดเหตุการณ์ผิดปกติทั้งหมดไม่ว่าจะเป็นไฟไหม้ โจรกรรม หรือขอความช่วยเหลือจะมีการเก็บประวัติของการเกิดเหตุการณ์ทั้งหมด ซึ่งจะเก็บทั้งเวลาที่เกิดเหตุการณ์และเวลาที่ยุติเหตุการณ์นั้นลงเพื่อสามารถนำมาเป็นข้อมูลอ้างอิงได้ หน้าจอแสดงประวัติเป็นดังรูปที่ 4.35



รูปที่ 4.35 หน้าต่างแสดงประวัติการเกิดเหตุการณ์ผิดปกติทั้งหมด

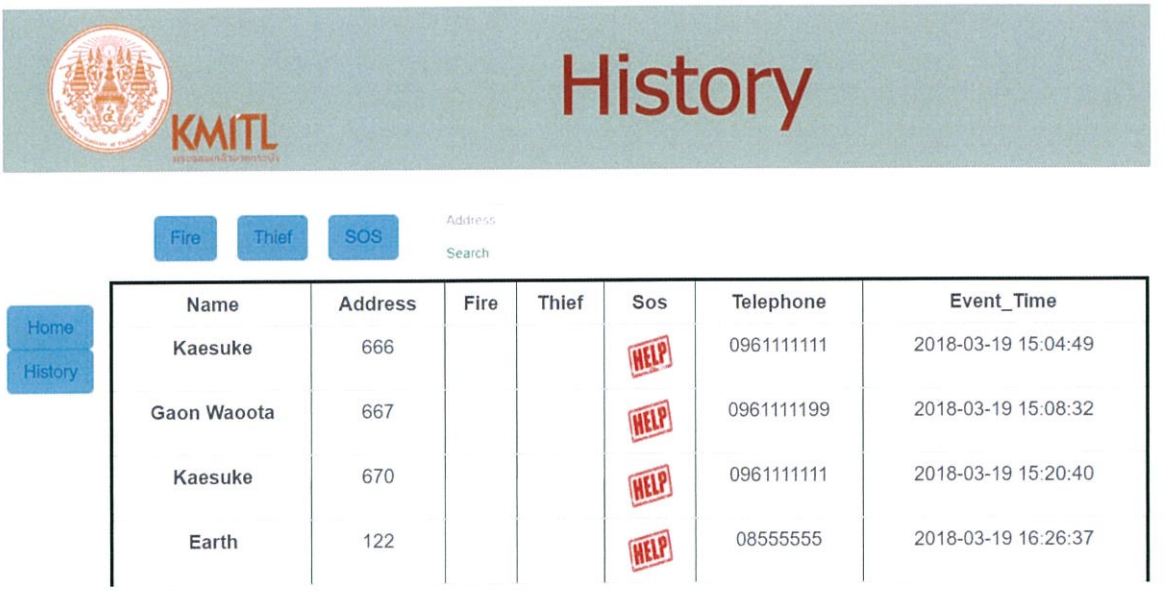
ในหน้าการเก็บประวัตินั้นสามารถกดเลือกได้ว่าต้องการจะดูข้อมูลเหตุการณ์ผิดปกติบางเหตุการณ์เพียงอย่างเดียวเช่นต้องการดูเหตุการณ์ไฟไหม้เพียงอย่างเดียวจะเป็นดังรูปที่ 4.36 หรือ

ต้องการดูเหตุการณ์โจรกรรมเพียงอย่างเดียวจะเป็นดังรูปที่ 4.37 และต้องการดูเหตุการณ์ขอความช่วยเหลือเพียงอย่างเดียวจะเป็นดังรูปที่ 4.38



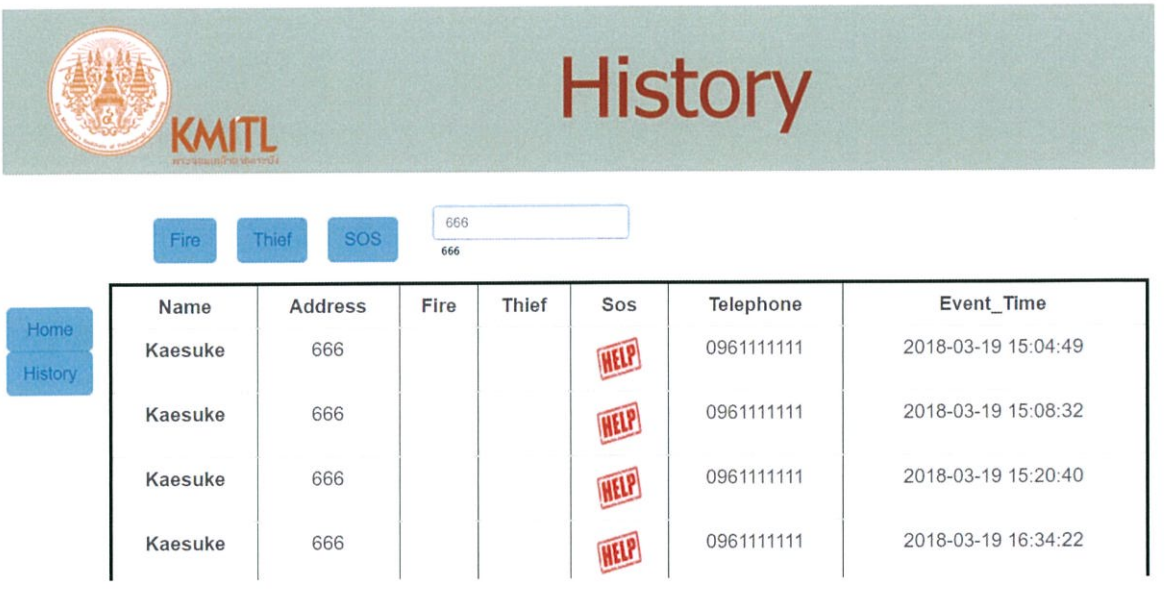
รูปที่ 4.36 หน้าต่างแสดงประวัติของเหตุการณ์ไฟไหม้เพียงอย่างเดียว

รูปที่ 4.37 หน้าต่างแสดงประวัติของเหตุการณ์โจรกรรมเพียงอย่างเดียว



รูปที่ 4.38 หน้าต่างแสดงประวัติของเหตุการณ์ขอความช่วยเหลือเพียงอย่างเดียว

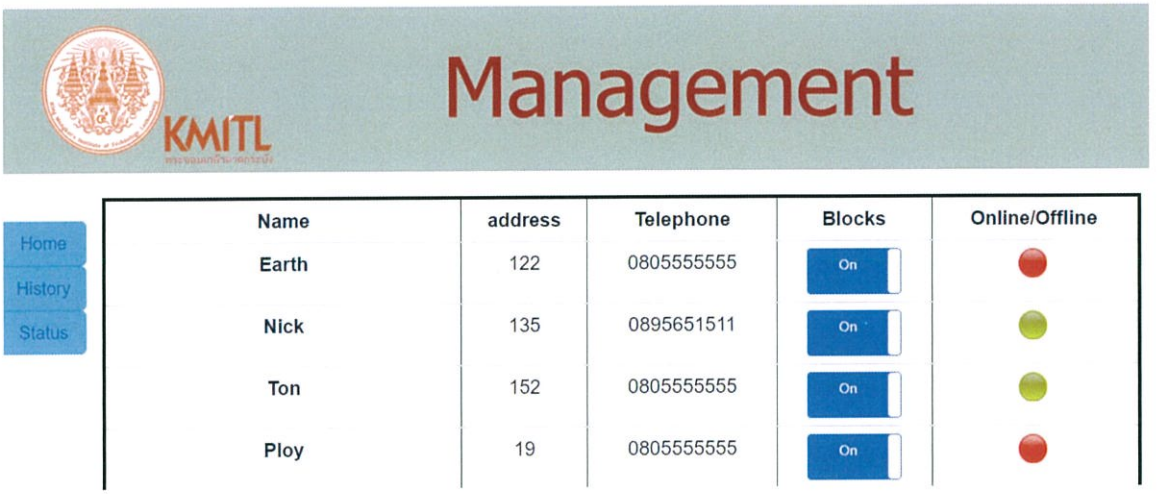
หน้าต่างแสดงประวัติสามารถค้นหาบ้านเลขที่หรือชื่อเจ้าของบ้านได้เพื่อให้สะดวกในการค้นหาประวัติของแต่ละบ้านดังรูปที่ 4.39



รูป 4.39 หน้าต่างแสดงการค้นหาจากบ้านเลขที่

4.6.7 ผลการทดสอบหน้าต่างแสดงสถานะของอุปกรณ์

เจ้าหน้าที่รักษาความปลอดภัยสามารถดูสถานะอุปกรณ์ภายในบ้านได้และถ้าหากมีอุปกรณ์ทำงานผิดพลาดหรือไมโครคอนโทลเลอร์ไม่สามารถติดต่อกับฐานข้อมูลได้ซึ่งระบบจะเช็คให้เองเจ้าหน้าที่รักษาความปลอดภัยสามารถปิดการแจ้งเตือนของแต่ละบ้านได้ในหน้าต่างจัดการระบบ ดังรูปที่ 4.40



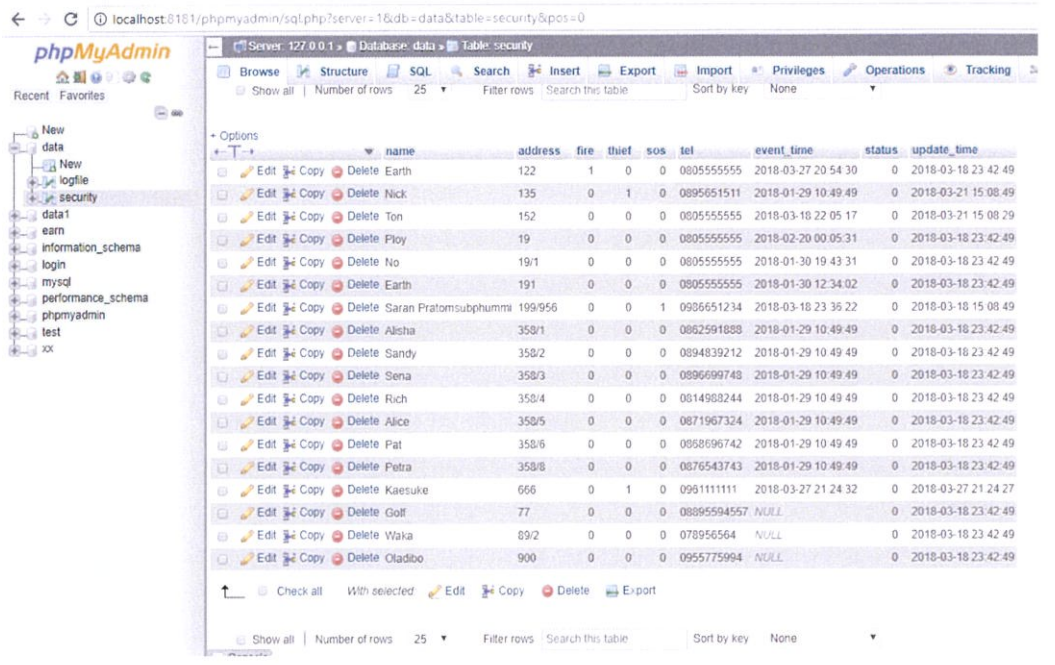
รูปที่ 4.40 หน้าต่างการจัดการระบบ

4.7 ผลการออกแบบฐานข้อมูล

4.7.1 ผลการออกแบบฐานข้อมูลที่มีชื่อว่า “security”

ฐานข้อมูลที่ชื่อว่า “security” เป็นฐานข้อมูลที่ใช้ในการแสดงผลในหน้าของจอมอนิเตอร์มีทั้งหมด 9 แถว ดังรูปที่ 4.41 คือ

1. “name” คือชื่อเจ้าของบ้าน
2. “address” คือบ้านเลขที่
3. “fire” สถานะของอัคคีภัย
4. “thief” สถานะของการโจรกรรม
5. “sos” สถานการณ์ขอความช่วยเหลือ
6. “tel” คือเบอร์โทรศัพท์ของเจ้าของบ้าน
7. “event_time” คือเวลาที่เกิดเหตุขึ้น
8. “status” คือสถานะที่ใช้เปิดปิดการรับส่งข้อมูลของไมโครคอนโทรลเลอร์กับศูนย์ควบคุมส่วนกลางถ้ามีค่าเป็น 1 คือปิด และ 0 คือเปิด
9. “update_time” คือเวลาที่อุปกรณ์ในแต่ละบ้านติดต่อมาล่าสุดใช้ในการดูสถานะออนไลน์และออฟไลน์

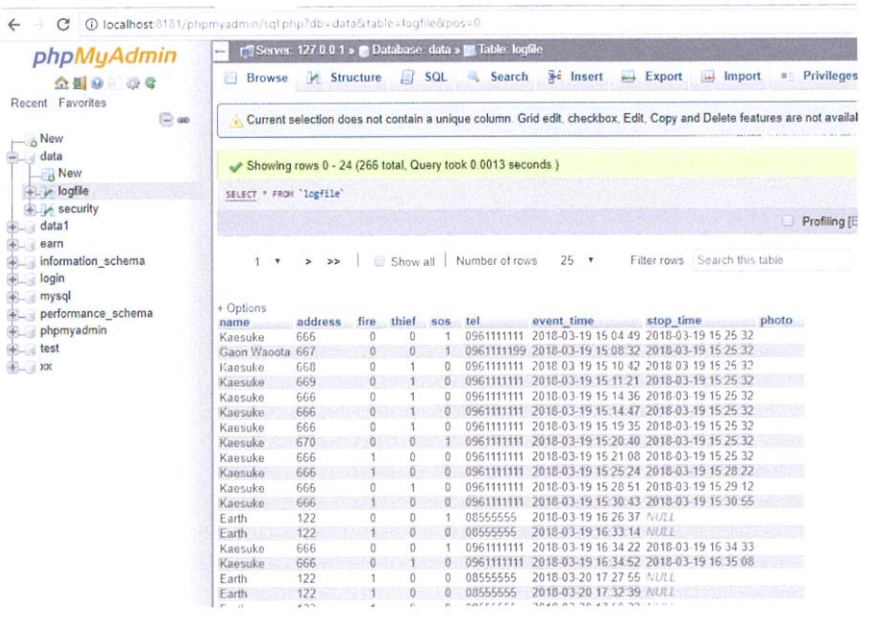


รูปที่ 4.41 ฐานข้อมูลที่ชื่อว่า “security”

4.7.2 ผลการออกแบบฐานข้อมูลที่มีชื่อว่า “logfile”

ฐานข้อมูลที่มีชื่อว่า “logfile” ใช้ในการเก็บเหตุการณ์ที่เกิดทั้งหมดเช่นเหตุการณ์ไฟไหม้ เหตุการณ์โจรกรรม เหตุการณ์ขอความช่วยเหลือ มีทั้งหมด 9 แถว ดังรูปที่ 4.42 คือ

1. “name” คือชื่อเจ้าของบ้าน
2. “address” คือบ้านเลขที่
3. “fire” สถานะของอัคคีภัย
4. “thief” สถานะของการโจรกรรม
5. “sos” สถานการณ์ขอความช่วยเหลือ
6. “tel” คือเบอร์โทรศัพท์ของเจ้าของบ้าน
7. “event_time” คือเวลาที่เกิดเหตุขึ้น
8. “stop_time” คือเวลาที่เหตุการณ์จบลง
9. “photo” คือชื่อไฟล์รูปภาพที่บันทึกในบ้านที่เกิดเหตุโจรกรรม

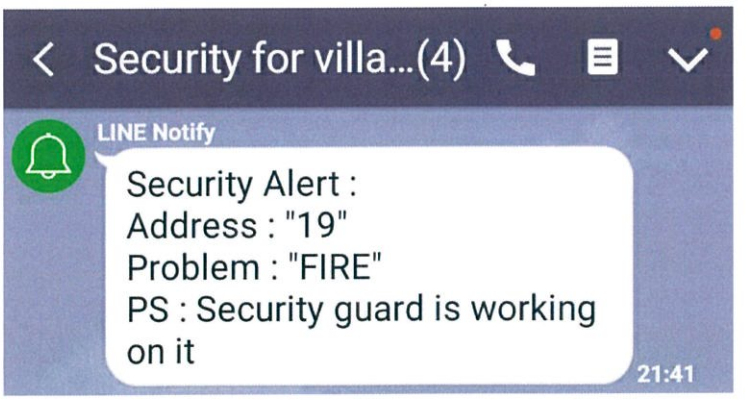


รูปที่ 4.42 ฐานข้อมูลที่มีชื่อว่า “logfile”

4.8 ผลการออกแบบในส่วนของการแจ้งเตือนผ่านแอปพลิเคชั่นไลน์

4.8.1 การแจ้งเตือนผ่านแอปพลิเคชั่นไลน์เมื่อเกิดอัคคีภัย

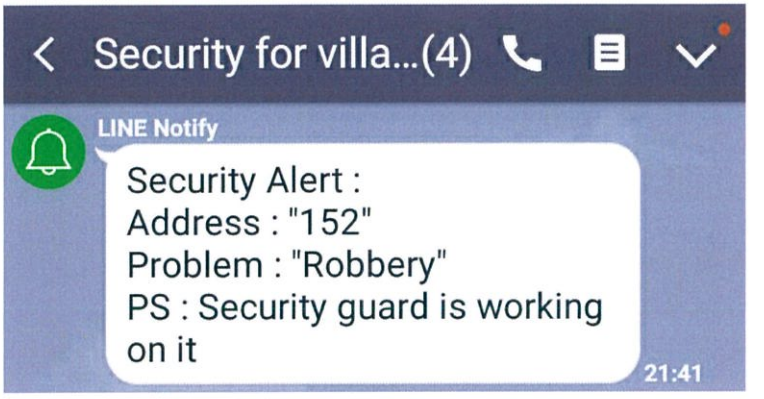
เมื่อตรวจจับเหตุการณ์ผิดปกติต่างๆได้ ระบบจะส่งข้อความผ่านแอปพลิเคชั่นไลน์ ไปที่เจ้าของบ้านที่เกิดเหตุ เพื่อแจ้งให้เจ้าของบ้านทราบโดยข้อความแจ้งเตือนจะระบุถึง “Address” คือ บ้านเลขที่ที่เกิดเหตุ และ “Problem” คือ ปัญหาที่เกิดขึ้น เช่น เมื่อเกิดเหตุการณ์อัคคีภัยจะแจ้งเตือนในช่อง “Problem” ว่า “FIRE” ดังรูปที่ 4.43



รูปที่ 4.43 ข้อความแจ้งเตือนผ่านแอปพลิเคชั่นไลน์เมื่อเกิดเหตุไฟไหม้

4.8.2 การแจ้งเตือนผ่านแอปพลิเคชั่นไลน์เมื่อเกิดการโจรกรรม

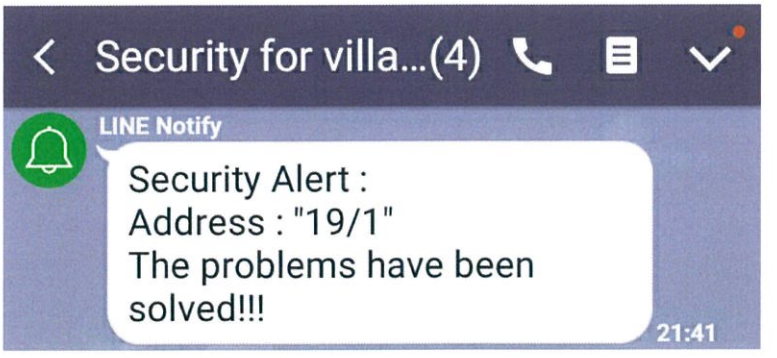
การแจ้งเตือนผ่านแอปพลิเคชั่นไลน์เมื่อเกิดการโจรกรรมจะเหมือนกับการแจ้งเตือนเมื่อเกิดอัคคีภัย แต่จะเปลี่ยนตรงช่อง Problem ว่า “Robbery” ดังรูปที่ 4.44



รูปที่ 4.44 ข้อความแจ้งเตือนผ่านแอปพลิเคชั่นไลน์เมื่อเกิดเหตุโจรกรรม

4.8.3 การแจ้งเตือนผ่านแอปพลิเคชั่นไลน์เมื่อสามารถยุติเหตุการณ์ต่างๆที่เกิดขึ้นได้แล้ว

เมื่อสามารถยุติเหตุการณ์ที่เกิดขึ้นได้แล้ว เจ้าหน้าที่รักษาความปลอดภัยของหมู่บ้านจะหยุดการแจ้งเตือนสถานการณ์ผิดปกติผ่านทางหน้า webpage ซึ่งจะมีการแจ้งเตือนไปที่เจ้าของบ้านที่เกิดเหตุอีกครั้ง ผ่านแอปพลิเคชั่นไลน์เพื่อแจ้งเตือนว่า เหตุการณ์สงบลงแล้ว โดยจะแจ้งบ้านเลขที่ที่สามารถยุติเหตุการณ์ได้แล้ว พร้อมกับข้อความแสดงว่า ปัญหาที่เกิดขึ้นได้ถูกยุติลงแล้ว ดังรูปที่ 4.45



รูปที่ 4.45 ข้อความแจ้งเตือนผ่านแอปพลิเคชั่นไลน์เมื่อปัญหายุติแล้ว

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

การทดสอบระบบรักษาความปลอดภัยในหมู่บ้านจัดสรร ขั้นตอนแรกจะต้องมีการติดตั้งเซนเซอร์ไว้แต่ละบ้าน เช่น เซนเซอร์ตรวจจับเปลวไฟ (IR flame sensor) ใช้ตรวจจับการเกิดอัคคีภัยและเซนเซอร์ตรวจจับการเคลื่อนไหว (PIR sensor) ใช้ในการตรวจจับผู้บุกรุกที่เข้ามาภายในบ้านโดยไม่ได้รับอนุญาต แล้วตั้งค่า SSID ของ WiFi , Password ของ WiFi , เลขที่บ้าน , ชื่อเจ้าของบ้าน และ เบอร์โทรศัพท์ของเจ้าของบ้านที่โหนดไมโครคอนโทรลเลอร์ เมื่อไมโครคอนโทรลเลอร์ได้รับค่าว่าเซนเซอร์ตรวจจับเปลวไฟ ตรวจจับเปลวไฟได้ หรือ เซนเซอร์ตรวจจับการเคลื่อนไหว ตรวจจับผู้ดบุกรุกได้ ไมโครคอนโทรลเลอร์จะจ่ายแรงดันไฟฟ้าออกไปที่พอร์ตที่ต่อกับบัซเซอร์ ทำให้เกิดเสียงแจ้งเตือนขึ้นเพื่อที่จะเตือนเพื่อนบ้านในบริเวณนั้นว่ามีอันตรายเกิดขึ้น นอกจากนั้นไมโครคอนโทรลเลอร์จะส่งค่าความผิดปกติไปที่ศูนย์ควบคุมส่วนกลางแสดงผลผ่านหน้าจอมอนิเตอร์ โดยที่หน้าจอมอนิเตอร์จะแสดงข้อมูลของบ้านที่เกิดเหตุเช่น บ้านเลขที่ ชื่อของเจ้าของบ้าน เบอร์โทรเจ้าของบ้าน และสถานการณ์ผิดปกติที่เกิดขึ้น ในกรณีที่ PIR sensor ทำงาน ระบบจะทำการเปิดใช้งานกล้องเพื่อจับภาพและส่งข้อมูลภาพในขณะนั้นมาแสดงที่หน้าจอมอนิเตอร์ เพื่อให้เจ้าหน้าที่รักษาความปลอดภัยประจำหมู่บ้านรับรู้ถึงเหตุการณ์ที่เกิดและสามารถเข้าแก้ไขปัญหาที่เกิดขึ้นได้อย่างทันท่วงที ในส่วนของหน้าจอแสดงผลยังสามารถแสดงประวัติการเกิดเหตุการณ์ที่เคยเกิดขึ้นในอดีตได้โดยข้อมูลที่นำมาแสดง คือ ชื่อ , บ้านเลขที่ , ประเภทของเหตุการณ์ , เบอร์โทรศัพท์ , เวลาที่เกิดเหตุและเวลาที่เหตุการณ์ยุติ เพื่อที่จะได้เป็นประโยชน์ในการเฝ้าระวังได้มากขึ้น

ในกรณีที่เกิดข้อผิดพลาดของระบบเช่นเซนเซอร์ทำงานผิดปกติทำให้เกิดการแจ้งเตือนซ้ำๆ หรือ ไมโครคอนโทรลเลอร์ไม่สามารถเชื่อมต่อกับฐานข้อมูลได้ ส่วนของหน้าจอแสดงผลจะมีฟังก์ชั่นสั่งปิดการทำงานของไมโครคอนโทรลเลอร์ให้เข้าสู่โหมดออฟไลน์ เพื่อที่จะสามารถเข้าตรวจสอบและแก้ไขระบบให้กลับมาทำงานได้ปกติอีกครั้ง

5.2 ข้อเสนอแนะ

เพื่อความแม่นยำในการตรวจจับเปลวไฟ หากบ้านมีขนาดใหญ่ควรเพิ่มจำนวนเซนเซอร์ตรวจจับเปลวไฟให้เหมาะสม และติดตั้งในสถานที่ที่สุ่มเสี่ยงต่อการเกิดเหตุเพลิงไหม้ได้ เช่น ห้องพระ หรือบริเวณใกล้ๆ ห้องครัว โดยหันทิศทางไปที่ที่อาจจะเกิดเพลิงไหม้ เช่น ถ้าเราติดตั้งไว้บนเพดานควรหันทิศทางเซนเซอร์ลงสู่พื้น

การตรวจจับผู้บุกรุกควรติดตั้งเซนเซอร์ตรวจจับการเคลื่อนไหว ในสถานที่ที่เหมาะสม และหันทิศของเซนเซอร์ไปทางที่ที่สามารถครอบคลุมพื้นที่ที่ต้องการเฝ้าระวัง เช่น ในกรณีที่ห้องนอนอยู่ที่ชั้นสอง เราสามารถติดเซนเซอร์ตรวจจับการเคลื่อนไหวกับก้าแพงที่ชั้นหนึ่ง และหันทิศทางเซนเซอร์ไปที่บริเวณกลางบ้าน เพื่อป้องกันการบุกรุกที่ชั้นหนึ่ง และครอบคลุมพื้นที่ที่ต้องการ

กรณีที่มีผู้คนอยู่ภายในบ้านควรจะปิดระบบการตรวจจับผู้บุกรุกทุกครั้ง เพราะระบบอาจตรวจจับผู้คนภายในบ้านที่ไม่ใช่ผู้บุกรุกได้

บรรณานุกรม

[1] รศ.ดร. สมยศ จุณณะปิยะ. *การประยุกต์ใช้งานไมโครคอนโทรลเลอร์*. พิมพ์ครั้งที่ 6. กรุงเทพฯ : สจล., 2550

[2] ผศ.ธีรวัฒน์ ประกอบผล. *ระบบคอมพิวเตอร์และภาษาซี*. พิมพ์ครั้งที่ 7 กรุงเทพมฯ : ส.ส.ท., 2537

[3] ศุภชัย สมพานิช. *"สร้างระบบงานฐานข้อมูลด้วย Visual Basic.NET ฉบับโปรแกรมเมอร์"* นนทบุรี : สำนักพิมพ์ ไอดีซี อินโฟ ดิสทริบิวเตอร์ เซ็นเตอร์ 2546

[4] Md Iftekharul Mobin, Md Abid-Ar-Rafi, Md Neamul Islam, and Md Rifat Hasan, "An Intelligent Fire Detection and Mitigation System Safe from Fire (SFF)", Department of Computer Science and Engineering University of Liberal Arts Bangladesh, (2010)

[5] Kausik Sen, Jeet Sarkar, Sutapa Saha, Anukrishna Roy, Dipsetu Dey, Sumit Baitalik,Chandra Sekhar Nandi. "Automated Fire Detection and Controlling System", Students of Applied Electronics and Instrumentation Engineering Department, University Institute Of Technology, Burdwan University,W.B. India, (2008)

[6] Yan Ge, Xianghong Sun, and Li Wang. "Presenting a Fire Alarm Using Natural Language": The Communication of Temporal Information, Thesis, Institute of Psychology, (2015)

[7] Thai Easy Elec. "รู้จักกับบอร์ด Raspberry Pi" http://thaieasyelec.com /article-wiki/embedded-electronics-application/ บทความการพัฒนาโปรแกรมบน-raspberry-pi-ด้วย-qt.html.

[8] MindPHP. "Python คืออะไร" http://www.mindphp.com/คู่มือ/73-คืออะไร/2417-python-คืออะไร.html.

[9] Lazada. "OKER กล้อง webcam รุ่น 088"
https://www.lazada.co.th/ products/oker-webcam-088-i120337014-s125181778.html.

[10] phpmyadmin. "Bringing MySQL to the web"
https://www.phpmyadmin.net/.

[11] Nirav Patel. "Camera Module Introduction"
http://www.pygame.org/ docs/tut/CameraIntro.html

ภาคผนวก ก

ไมโครคอนโทรลเลอร์

1. ชุดคำสั่งที่ใช้กับ Arduino nano 3.0 ที่ต่ออยู่กับ IR flame sensor

PitoArduinoNRF24-Fire §

```
#include<SPI.h>
#include<RF24.h>
RF24 radio(7,8); //CE,CSN
int firepin = 6;
void setup()
{
  pinMode(firepin,INPUT);
  Serial.begin(115200);
  radio.begin();
  radio.setPALevel(RF24_PA_MAX);
  radio.setChannel(0x76);
  radio.openWritingPipe(0xF0F0F0F0E1LL);
  radio.enableDynamicPayloads();
  radio.powerUp();
}
void loop() {
int  fire = digitalRead(firepin);
  fire = abs(fire-1);
  if(fire == true)
  {
  const char text[] = "fire";
  Serial.println(text);
  radio.write(&text,sizeof(text));
  delay(1000);
  }
  fire = 0;
}
```

2. ชุดคำสั่งที่ใช้กับ Arduino nano 3.0ที่ใช้เป็นรีโมท

PitoArduinoNRF24-switch1

```
#include<SPI.h>
#include<RF24.h>
RF24 radio(7,8); //CE,CSN
int switchoffpin = 2;
int switchonpin = 4;
int switchsospin = 5;
void setup()
{
  pinMode(switchoffpin,INPUT);
  pinMode(switchonpin,INPUT);
  pinMode(switchsospin,INPUT);
  Serial.begin(115200);
  radio.begin();
  radio.setPALevel(RF24_PA_MAX);
  radio.setChannel(0x76);
  radio.openWritingPipe(0xF0F0F0F0E1LL);
  radio.enableDynamicPayloads();
  radio.powerUp();
}

void loop() {
int  switchoff = digitalRead(switchoffpin);
int  switchon = digitalRead(switchonpin);
int  switchsos = digitalRead(switchsospin);
Serial.println(switchoff);
Serial.println(switchon);
Serial.println(switchsos);
```

PitoArduinoNRF24-switch1

```
Serial.println(switchoff);
Serial.println(switchon);
Serial.println(switchsos);
  if(switchoff == true)
  {
  const char text[] = "switchoff";
  Serial.println(text);
  radio.write(&text,sizeof(text));
  delay(1000);
  }
  else if(switchon == true)
  {
  const char text[] = "switchon";
  Serial.println(text);
  radio.write(&text,sizeof(text));
  delay(1000);
  }
  else if(switchsos == true)
  {
  const char text[] = "switchsos";
  Serial.println(text);
  radio.write(&text,sizeof(text));
  delay(1000);
  }
  switchoff = 0;
  switchon = 0;
  switchsos = 0;
}
```

3. ชุดคำสั่งที่ใช้กับ Raspberry pi 3 model B ที่ต่อกับ PIR sensor และ Webcam

```
import RPi.GPIO as GPIO
from lib_nrf24 import NRF24
import time
import calendar
import spidev
import requests
import pygame
import pygame.camera
from socket import *

ip = "192.168.1.99"
port = "8181"
name = "Kaesuke"
address = "666"
tel = "0961111111"

urladd  = "http://"+ip+":"+port+"/uploadtodatabase/add8.php?address="+address+"&name="+name+"&tel="+tel
urladdthief = "http://"+ip+":"+port+"/uploadtodatabase/addthief5.php?address="+address+"&name="+name+"&tel="+tel
urladdfire = "http://"+ip+":"+port+"/uploadtodatabase/addfire4.php?address="+address+"&name="+name+"&tel="+tel
urldown = "http://"+ip+":"+port+"/uploadtodatabase/down6.php?address="+address
urlstoptime = "http://"+ip+":"+port+"/uploadtodatabase/stoptime5.php?address="+address
urladdsos = "http://"+ip+":"+port+"/uploadtodatabase/addsos4.php?address="+address+"&name="+name+"&tel="+tel

pygame.init()
pygame.camera.init()
cam = pygame.camera.Camera("/dev/video0", (640, 480))


buzzer = 26
pirpin = 5
mode = 0
GPIO.setmode(GPIO.BCM)
GPIO.setup(buzzer,GPIO.OUT)
GPIO.setup(pirpin,GPIO.IN)

client = socket(AF_INET,SOCK_STREAM)

bufsize = 1024000
portpython = 8000
ADDR = (ip,portpython)
```

```
pipes = [[0xE8, 0xE8, 0xF0, 0xF0, 0xE1], [0xF0, 0xF0, 0xF0 , 0xF0, 0xE1]]
radio = NRF24(GPIO,spidev.SpiDev())
radio.begin(0,25)
radio.setPayloadSize(32)
radio.setChannel(0x76)
radio.setDataRate(NRF24.BR_1MBPS)
radio.setPALevel(NRF24.PA_MIN)
radio.setAutoAck(True)
radio.enableDynamicPayloads()
radio.enableAckPayload()
radio.openReadingPipe(1, pipes[1])
radio.printDetails()
radio.startListening()

requests.get(urladd)
while True:
        time.sleep(1)
        while not radio.available(0):
                time.sleep(1)
                r = requests.get(urldown)
                downstate = r.text
                print("Downstate : "+downstate)
                print("Mode : "+str(mode))
                if(mode==1):
                        pir = GPIO.input(pirpin)
                        print("pir : "+str(pir))
                        if(pir==1 and downstate[1] != "1"):
                                print("Thief is detected")
                                GPIO.output(buzzer,True)
                                print("Buzzer is ON")
                                ts = calendar.timegm(time.gmtime())
                                photoname = address+'_'+str(ts)
                                filename = 'Snaps/'+photoname+'.jpg'
                                cam.start()
                                image = cam.get_image()
                                pygame.image.save(image, filename)
                                print('Photo has saved')
                                cam.stop()
                                addthiefphoto = urladdthief+'&photo='+photoname
                                requests.get(addthiefphoto)
                                data = open(filename,'rb').read()
                                client.connect(ADDR)
                                client.send(str.encode(photoname))
                                time.sleep(0.1)
                                client.send(data)
                                client.close()
                                print('Photo has sent')
        receivedMessage = []
        radio.read(receivedMessage, radio.getDynamicPayloadSize())
        nrfstate = ""
        for n in receivedMessage:
            if(n >= 32 and n<= 126):
                nrfstate += chr(n)
        print(nrfstate)
        if(nrfstate=="switchoff"):
                mode = 0
                print("Thief detector is OFF")
                GPIO.output(buzzer,True)
                time.sleep(0.7)
                GPIO.output(buzzer,False)
                if(downstate!="000"):
                        requests.get(urlstoptime)
                        print("Buzzer is OFF")
        elif(downstate[0]!="1" and nrfstate=="fire"):
                print("On fire")
                requests.get(urladdfire)
                GPIO.output(buzzer,True)
                print("Buzzer is ON")
        elif(nrfstate=="switchon"):
                print("Thief detector is ON")
                mode = 1
                if(downstate=="000"):
                        GPIO.output(buzzer,True)
                        time.sleep(0.3)
                        GPIO.output(buzzer,False)
                        time.sleep(0.3)
                        GPIO.output(buzzer,True)
                        time.sleep(0.3)
                        GPIO.output(buzzer,False)
        elif(downstate[2]!="1" and nrfstate=="switchsos"):
                print("SOS")
                requests.get(urladdsos)
                print("Update SOS status to database")
                if(downstate=="000"):
                        GPIO.output(buzzer,True)
                        time.sleep(0.2)
                        GPIO.output(buzzer,False)
                        time.sleep(0.2)
                        GPIO.output(buzzer,True)
                        time.sleep(0.2)
                        GPIO.output(buzzer,False)
                        time.sleep(0.2)
                        GPIO.output(buzzer,True)
                        time.sleep(0.2)
                        GPIO.output(buzzer,False)
                        time.sleep(0.2)
                        GPIO.output(buzzer,True)
                        time.sleep(0.2)
                        GPIO.output(buzzer,False)
```

ภาคผนวก ข

การเชื่อมต่อฐานข้อมูล

1.ส่วนของการติดต่อฐานข้อมูลเพื่อส่งค่าข้อมูลต่างๆไปยังฐานข้อมูลเมื่อมีการใช้งานใหม่

```php
<?php
    $serverName     = "localhost";
    $userName       = "root";
    $userPassword         = "";
    $dbName       = "data";
    $conn = mysqli_connect($serverName,$userName,$userPassword,$dbName);
    if ($conn)
    {
        echo "connect : success<br>";
    }
    else
    {
        echo "bad connection";
    }
$name = $_GET['name'];
$address = $_GET['address'];
$fire = 0;
$thief = 0;
$tel = $_GET['tel'];
$sql1 = "DELETE FROM security WHERE address = '$address'";
$sql2 = "INSERT INTO security(name,address,fire,thief,tel)VALUES('$name','$address','$fire','$thief','$tel')";
if($conn->query($sql1)&&$conn->query($sql2) == TRUE){
    echo "save OK";
}else{
    echo "Error:" . $sql1 . "</br>" . $conn->error;
}
$conn->close();
?>
```

2.ส่วนของการติดต่อฐานข้อมูลค่าเมื่อเกิดเหตุอัคคีภัย

```php
<?php
include("connectline3.php");
    $serverName     = "localhost";
    $userName       = "root";
    $userPassword         = "";
    $dbName       = "data";
    $conn = mysqli_connect($serverName,$userName,$userPassword,$dbName);
    if ($conn)
    {
        echo "connect : success<br>";
    }
    else
    {
        echo "bad connection";
    }
$name = $_GET['name'];
$address = $_GET['address'];
$fire = 1;
$thief = 0;
$sos = 0;
$tel = $_GET['tel'];
$message = '
Address : "'.$address.'"'.'
Problem : "FIRE"'.'
PS : Security guard is working on it';
$line = send_line_notify($message, $token);
echo $line;
$sql1 = "UPDATE `security` SET `fire`=$fire , event_time = CURRENT_TIMESTAMP WHERE address = '$address';";
$sql2 = "INSERT INTO logfile(name,address,fire,thief,tel,event_time)VALUES('$name','$address','$fire','$thief','$tel',CURRENT_TIMESTAMP)";
if($conn->query($sql1)&&$conn->query($sql2) == TRUE){
    echo "save OK";
}else{
    echo "Error:" . $sql1 . "</br>" . $conn->error;
}
$conn->close();
?>
```

3.ส่วนของการติดต่อฐานข้อมูลค่าเมื่อเกิดเหตุการณ์โจรกรรม

```php
<?php
include("connectline3.php");
    $serverName     = "localhost";
    $userName       = "root";
    $userPassword       = "";
    $dbName         = "data";
    $conn = mysqli_connect($serverName,$userName,$userPassword,$dbName);
    if ($conn)
    {
        echo "connect : success<br>";
    }
    else
    {
        echo "bad connection";
    }
$name = $_GET['name'];
$address = $_GET['address'];
$fire = 0;
$thief = 1;
$sos = 0;
$tel = $_GET['tel'];
$photo = $_GET['photo'];
$message = '
Address : "'.$address.'"'.'
Problem : "Robbery"'.'
PS : Security guard is working on it';
$line = send_line_notify($message, $token);
echo $line;
$sql1 = "UPDATE `security` SET `thief`=$thief , event_time = CURRENT_TIMESTAMP WHERE address = '$address';";
$sql2 = "INSERT INTO logfile(name,address,fire,thief,sos,tel,event_time,photo)VALUES('$name','$address','$fire','$thief','$sos','$tel',
    CURRENT_TIMESTAMP,'$photo')";
if($conn->query($sql1)&&$conn->query($sql2) == TRUE){
    echo "save OK";
}else{
    echo "Error:" . $sql1 . "</br>" . $conn->error;
}
$conn->close();
?>
```

4.ส่วนของการติดต่อฐานข้อมูลค่าเมื่อเกิดกดปุ่มร้องขอความช่วยเหลือ

```php
<?php
include("connectline3.php");
    $serverName     = "localhost";
    $userName       = "root";
    $userPassword       = "";
    $dbName         = "data";
    $conn = mysqli_connect($serverName,$userName,$userPassword,$dbName);
    if ($conn)
    {
        echo "connect : success<br>";
    }
    else
    {
        echo "bad connection";
    }
$name = $_GET['name'];
$address = $_GET['address'];
$sos = 1;
$fire = 0;
$thief = 0;
$tel = $_GET['tel'];
$message = '
Address : "'.$address.'"'.'
Problem : "SOS"'.'
PS : Security guard is working on it';
$line = send_line_notify($message, $token);
echo $line;
$sql1 = "UPDATE `security` SET `sos`=$sos , event_time = CURRENT_TIMESTAMP WHERE address = '$address';";
$sql2 = "INSERT INTO logfile(name,address,fire,thief,sos,tel,event_time)VALUES('$name','$address','$fire','$thief','$sos','$tel',
    CURRENT_TIMESTAMP)";
if($conn->query($sql1)&&$conn->query($sql2) == TRUE){
    echo "save OK";
}else{
    echo "Error:" . $sql1 . "</br>" . $conn->error;
}
$conn->close();
?>
```

5.ส่วนของการติดต่อฐานข้อมูลเพื่อตรวจสอบสถานะ fire thief และ sos และส่งค่าเวลาล่าสุดที่ติดต่อกับฐานข้อมูล

```php
<?php
    $serverName     = "localhost";
    $userName      = "root";
    $userPassword        = "";
    $dbName      = "data";
    $conn = mysqli_connect($serverName,$userName,$userPassword,$dbName);
    mysqli_select_db($conn,"security");
    $address = $_GET['address'];
$sql1 = "UPDATE `security` SET update_time = CURRENT_TIMESTAMP WHERE address = '$address';";
$conn->query($sql1);
$sql2 = "SELECT fire,thief,sos FROM security WHERE address = '$address';";
$objQuery = mysqli_query($conn,$sql2);
while($objResult = mysqli_fetch_array($objQuery))
{
    $firedown = $objResult["fire"];
    $thiefdown = $objResult["thief"];
    $sos = $objResult["sos"];
}
$o = $firedown.$thiefdown.$sos;
echo $o;
$conn->close();
?>
```

6.ส่วนของการติดต่อฐานข้อมูลเมื่อกดปิดการแจ้งเตือนทั้งหมด

```php
<?php
include("connectline3.php");
    $serverName     = "localhost";
    $userName      = "root";
    $userPassword        = "";
    $dbName      = "data";
    $conn = mysqli_connect($serverName,$userName,$userPassword,$dbName);
    if ($conn)
    {
        echo "connect : success<br>";
    }
    else
    {
        echo "bad connection";
    }
$address = $_GET['address'];
$message = '
Address : "' $address.'"
The problems have been solved!!!';
$line = send_line_notify($message, $token);
echo $line;
$sql2 = "UPDATE `logfile` SET `stop_time`= CURRENT_TIMESTAMP WHERE address = '$address' ORDER BY event_time DESC
    LIMIT 1;";
$sql3 = "UPDATE `security` SET `fire`= 0,`thief`= 0 ,sos=0 WHERE address = '$address'";
if($conn->query($sql2)&&$conn->query($sql3) == TRUE){
    echo "save OK";
}else{
    echo "Error:" . $sql2 . "</br>" . $conn->error;
}
$conn->close();
?>
```

ภาคผนวก ค

การแสดงผล

1.ส่วนของหน้าจอมอนิเตอร์เหตุผิดปกติต่างๆภายในหมู่บ้าน

1.1 การเชื่อมต่อฐานข้อมูลและการเลือกข้อมูลจากฐานข้อมูล โดยเจาะจงฐานข้อมูลที่มีชื่อว่า security และคอลัมน์ที่มีชื่อ fire, thief และ sos ที่มีค่าเป็น 1

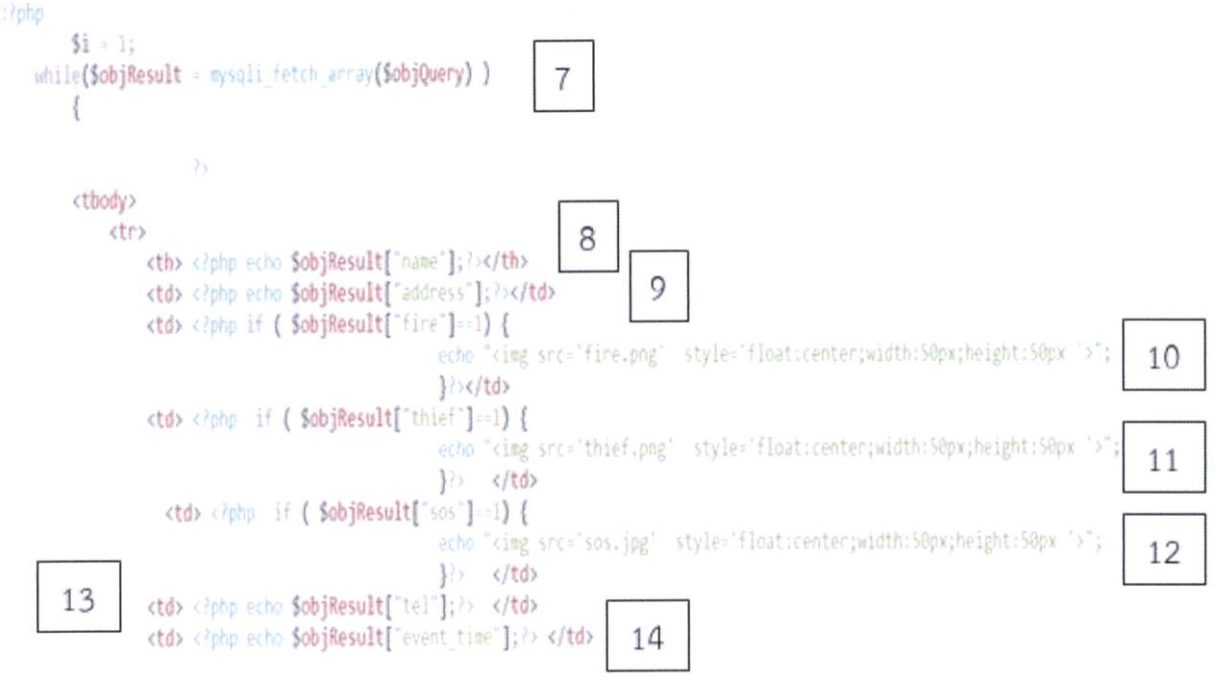
```php
<?php
  $host = "localhost";
  $username = "root";
  $password = "";
  $dbName     = "data";
  $objconncet= mysqli_connect($host,$username,$password,$dbName) or die("Error Connect to Database");

    mysqli_select_db($objconncet,"security");
  $strSQL = "SELECT * FROM security WHERE fire='1' OR thief='1' OR sos='1';";
              $objQuery =mysqli_query ($objconncet,$strSQL) or die ("Error Query [".$strSQL."]");

  $num = mysqli_num_rows($objQuery);
      if ($num>=1) {
        echo "<embed src='bellPart.mp3' width='0' height='0' ></embed>";
      }elseif ($num==0) {
        # code...
      }
```

1. เก็บค่า data ซึ่งเป็นชื่อของฐานข้อมูลไว้ในตัวแปล $dbName
2. เชื่อมต่อฐานข้อมูลโดยใช้คำสั่ง mysqli_connect() ถ้าเชื่อมต่อได้จะมีค่า TRUE เก็บไว้ในตัวแปล $objconnect
3. เชื่อมต่อฐานข้อมูลโดยเจาะจงฐานข้อมูลที่มีชื่อว่า security
4. ค้นหาข้อมูลในฐานข้อมูล เจาะจงคอลัมน์ fire, thief และ sos ที่มีข้อมูลเป็น 1
5. นับแถวของข้อมูลเมื่อค้นหาแบบเจาะจงแล้ว ถ้านับได้ 1 แถวแสดงว่าจะต้องมีแถวที่มีข้อมูลของคอลัมน์ fire หรือ thief หรือ sos เปลี่ยนจาก 0 เป็น 1 จำนวนที่นับได้จะถูกเก็บไว้ใน $num
6. กรณีที่ $num มีค่ามากกว่าหรือเท่ากับ 1 จะแจ้งเตือนโดยใช้ไฟล์เสียงชื่อ bellpart.mp3

1.2. การนำข้อมูลในฐานข้อมูลมาแสดงที่หน้าจอมอนิเตอร์



$objQuery ที่เก็บข้อมูลของฐานข้อมูลที่ถูกเลือกหลัก fire หรือ thief หรือ sos แล้วจะนำข้อมูลที่ได้มาแสดงผลบนหน้าจอมอนิเตอร์เป็นตารางโดยจะเริ่มจากแถวแรกเรียงคอลัมน์ไปเรื่อย ๆ จนถึงแถวสุดท้ายคอลัมน์สุดท้ายจะมีขั้นตอนดังนี้

7. คืนค่าข้อมูลในฐานข้อมูลในลักษณะเป็นแถวโดยใช้คำสั่ง mysqil_fetch_array() ข้อมูลจะถูกเก็บไว้ใน $objResult จากนั้นใช้คำสั่ง While เริ่มแถวที่ $i=1 จนถึง $i=$i+1 เพื่อแสดงข้อมูล ข้อมูลจะแสดงทีละคอลัมน์เริ่มจากข้อ 8,9,10,11,12,13,14,15 ตามลำดับ

8. แสดงข้อมูลที่เก็บไว้ในคอลัมน์ชื่อ name ของแถวที่ $i

9. แสดงข้อมูลที่เก็บไว้ในคอลัมน์ชื่อ address ของแถวที่ $i

10. แสดงข้อมูลที่เก็บไว้ในคอลัมน์ชื่อ fire ของแถวที่ $i โดยตรวจสอบก่อนว่าถ้าข้อมูลเป็น 1 ให้แสดงข้อมูลบนเว็บเพจเป็นรูป fire.png

11. แสดงข้อมูลที่เก็บไว้ในคอลัมน์ชื่อ thief ของแถวที่ $i โดยตรวจสอบก่อนว่าถ้าข้อมูลเป็น 1 ให้แสดงข้อมูลบนเว็บเพจเป็นรูป thief.png

12. แสดงข้อมูลที่เก็บไว้ในคอลัมน์ชื่อ sos ของแถวที่ $i โดยตรวจสอบก่อนว่าถ้าข้อมูลเป็น 1 ให้แสดงข้อมูลบนเว็บเพจเป็นรูป sos.jpg

13. แสดงข้อมูลที่เก็บไว้ในคอลัมน์ชื่อ tel ของแถวที่ $i

14. แสดงข้อมูลแสดงข้อมูลที่เก็บไว้ในคอลัมน์ชื่อ event_time แสดงค่าเวลาที่เกิดเหตุการณ์

```
<td> <?php
                if ( $objResult["thief"]==1) {
                 $ad[$i] = $objResult["address"];
               echo"<form name='frmMain' method='POST' action='showpig.php' >
                                <input type='hidden' name='address' value=$ad[$i]>

                                <button type='submit' style='width: 50px; height: 40px'><img
                                   src='CCTV.png' width='35' height='35' align=absmiddle>
                                </form>";                                                    }
 ?> </td>
```

15

15. เลือกเหตุการณ์ที่มีค่า thief เท่ากับ 1 และ $ad[$i] = $objResult[“address”] จะเป็นการจับคู่ address เข้ากับแถว $i ไว้ที่ ตัวแปล $ad จากนั้นสร้างปุ่ม submit โดยให้แสดงผลเป็นไฟล์ภาพ CCTV.png ถ้ามีการกดจะทำการส่งข้อมูล address ไปที่ showpig.php

1.3 ไฟล์ showpig.php

```
<?php
   $address = $_POST['address'];
   $host = "localhost";
   $username = "root";
   $password = "";
   $dbName     = "data";
   $objconncet= mysqli_connect($host,$username,$password,$dbName) or die("Error Connect to Database");

    mysqli_select_db($objconncet,"security");
   $spcSQL = "SELECT * FROM security WHERE address='$address'";
   $objQuery =mysqli_query ($objconncet,$spcSQL) or die ("Error Query [".$spcSQL."]");
        $objResult = mysqli_fetch_array($objQuery);
$namepig= $objResult["photo"];
   ?>

      <?php  echo  echo "<img src='$namepig.jpg'  style='float:center;width:50px;height:50px '>";
```

รับข้อมูล address ที่ถูกส่งมาเก็บไว้ในตัวแปล $address จากนั้นเชื่อมต่อฐานข้อมูลและเลือกข้อมูลที่เจาะจงกับ address เฉพาะคอลัมน์ที่มีชื่อว่า photo ซึ่งคอลัมน์นี้จะเก็บชื่อของไฟล์ข้อมูลภาพไว้จากนั้นนำไฟล์ข้อมูลภาพที่ได้มาแสดงผ่านหน้าจอมอนิเตอร์

2. ในส่วนของหน้าต่างแสดงประวัติเหตุการณ์ที่เคยเกิดขึ้น

2.1 การเชื่อมต่อกับฐานข้อมูลจะเหมือนกับของหน้าจอมอนิเตอร์แต่จะแตกต่างกันตรงที่จะใช้อีกฐานข้อมูลในการเก็บประวัติ ในส่วนนี้ใช้ฐานข้อมูลชื่อ logfile ซึ่งเมื่อเกิดเหตุการณ์ผิดปกติขึ้นไมโครคอลโทลเลอร์จะส่งข้อมูลทั้งหมดไปยังฐานข้อมูล logfile เป็นการเพิ่มแถวภายในตารางและจะเพิ่มเรื่อย ๆ เมื่อมีเหตุการณ์ใหม่เกิดขึ้นมาอีก

```php
<?php
  $host = "localhost";
  $username = "root";
  $password = "";
  $dbName     = "data";
  $objconncet= mysqli_connect($host,$username,$password,$dbName) or die("Error Connect to Database");
    mysqli_select_db($objconncet,"logfile");
  $strSQL = "SELECT * FROM logfile WHERE fire='1' OR thief='1' OR sos='1';";//
  $objQuery = mysqli_query($objconncet,$strSQL) or die ("Error Query [".$strSQL."]");
```

2.2 การนำข้อมูลจากฐานข้อมูลมาแสดงจะเหมือนหน้าจอมอนิเตอร์แต่จะมีข้อมูลของเวลาเมื่อเหตุการณ์สิ้นสุดเพิ่มเข้ามาซึ่งจะใช้ชื่อคอลัมน์ชื่อว่า stop_time

```php
    <tr>
            <th scope="row"><?php echo $objResult["name"];?></th>
            <td><?php echo $objResult["address"];?></td>
            <td><?php if ( $objResult["fire"]=1) {
                                        echo "<img src='fire.png'  style='float:center;width:50px;height:50px
                                        }?></td>
            <td><?php  if ( $objResult["thief"]=1) {
                                        echo "<img src='thief.png'  style='float:center;width:50px;height:50p
                                        }?></td>
             <td><?php  if ( $objResult["sos"]=1) {
                                        echo "<img src='sos.jpg'  style='float:center;width:50px;height:50px
                                        }?></td>
            <td><?php echo $objResult["tel"];?></td>
            <td><?php echo $objResult["event_time"];?></td>
            <td><?php echo $objResult["stop_time"];?></td>
            <td> <?php
                        if ( $objResult["thief"]=1) {
                         $ad[$i] = $objResult["address"];
                    echo"<form name='frmMain' method='POST' action='showvideo.php' >
                                                <input type='hidden' name='address' value=$ad[$i]>

                                                <button type='submit' style='width: 50px; height: 40px'><img
                                                  src='CCTV.png' width='35' height='35' align=absmiddle>
                                                </form>";                                                    }?>


    </tr>
<?php
$i = $i+1;
}
```

2.3 หน้าต่างแสดงประวัติเหตุการณ์จะมีการค้นหาเหตุการณ์แบบเฉพาะเจาะจง 4 แบบ

2.3.1 ค้นหาข้อมูลของเหตุการณ์บ้านที่เกิดเหตุการณ์ไฟไหม้อย่างเดียว เมื่อกดที่ปุ่มค้นหาเหตุการณ์ที่เกิดไฟไหม้หน้าจอแสดงประวัติจะสลับไปใช้หน้าของไฟล์ searchfire.php

```
<?php
    $serverName     = "localhost";
    $userName       = "root";
    $userPassword         = "";
    $dbName       = "data";
    $conn = mysqli_connect($serverName,$userName,$userPassword,$dbName)or die("Error Connect to Database");

    mysqli_select_db($conn,"logfile");

    $spcSQL = "SELECT * FROM logfile WHERE fire='1'";
    $result  =mysqli_query ($conn,$spcSQL) or die ("Error Query [".$spcSQL."]");
?>
```

ในการเชื่อมต่อฐานข้อมูลของ searchfire.php จะทำการเชื่อมต่อไปที่ฐานข้อมูลที่มีชื่อว่า logfile และเลือกเฉพาะแถวที่มีค่าในคอลัมน์ของ fire เป็น 1 ในส่วนของการนำข้อมูลต่างๆมาแสดงผลนั้นจะทำงานดังเช่นหน้าต่างแสดงประวัติเหตุการณ์

2.3.2 ค้นหาข้อมูลของเหตุการณ์บ้านที่เกิดเหตุการณ์โจรกรรมอย่างเดียว เมื่อกดที่ปุ่มค้นหาเหตุการณ์ที่เกิดไฟไหม้หน้าจอแสดงประวัติจะสลับไปใช้หน้าของไฟล์ searchthief.php

```
<?php
    $serverName     = "localhost";
    $userName       = "root";
    $userPassword         = "";
    $dbName       = "data";
    $conn = mysqli_connect($serverName,$userName,$userPassword,$dbName)or die("Error Connect to Database");
    mysqli_select_db($conn,"logfile");
    $spcSQL = "SELECT * FROM logfile WHERE thief='1'";
    $result  =mysqli_query ($conn,$spcSQL) or die ("Error Query [".$spcSQL."]");
?>
```

ในการเชื่อมต่อฐานข้อมูลของ searchthief.php จะทำการเชื่อมต่อไปที่ฐานข้อมูลที่มีชื่อว่า logfile และเลือกเฉพาะแถวที่มีค่าในคอลัมน์ของ thief เป็น 1 ในส่วนของการนำข้อมูลต่างๆมาแสดงผลนั้นจะทำงานดังเช่นหน้าต่างแสดงประวัติเหตุการณ์

2.2.3 ค้นหาข้อมูลของเหตุการณ์บ้านที่มีการร้องขอความช่วยเหลือ เมื่อกดที่ปุ่มค้นหาเหตุการณ์ที่เกิดไฟไหม้หน้าต่างแสดงประวัติจะสลับไปใช้หน้าของไฟล์ searchsos.php

```
<?php
    $serverName     = "localhost";
    $userName       = "root";
    $userPassword         = "";
    $dbName       = "data";
    $conn = mysqli_connect($serverName,$userName,$userPassword,$dbName)or die("Error Connect to Database");
    mysqli_select_db($conn,"logfile");
    $spcSQL = "SELECT * FROM logfile WHERE sos='1'";
    $result  =mysqli_query ($conn,$spcSQL) or die ("Error Query [".$spcSQL."]");
?>
```

ในการเชื่อมต่อฐานข้อมูลของ searchsos.php จะทำการเชื่อมต่อไปที่ฐานข้อมูลที่มีชื่อว่า logfile และเลือกเฉพาะแถวที่มีค่าในคอลัมน์ของ sos เป็น 1 ในส่วนของการนำข้อมูลต่างๆมาแสดงผลนั้นจะทำงานดังเช่นหน้าต่างแสดงประวัติเหตุการณ์

2.2.4 การค้นหาเหตุการณ์ที่เกิดขึ้นทั้งหมดโดยการค้นหาบ้านเลขที่

```
<td>
        <button type="button" class="btn btn-info btn-lg"><?php echo
          "<a href='searchfire.php'>Fire</a>"?></button></td>

<td>
        <button type="button" class="btn btn-info btn-lg"><?php echo
          "<a href='searchthief.php'>Thief</a>"?></button></td>

<td>
        <button type="button" class="btn btn-info btn-lg"><?php echo
          "<a href='searchsos.php'>SOS</a>"?></button> </td>

    <td><?php

          echo "<div class='collapse navbar-collapse'
            id='navbarSupportedContent'>
<form class='form-inline my-2 my-lg-0' method='POST' action='bot.php'>
<input class='form-control mr-sm-2' type='search'
  placeholder='Address' aria-label='Search' name='address'>
          <button class='btn btn-outline-success my-2 my-sm-0'
            type='submit'>Search</button>
              </form></div> "; ?> </td>
```

จากรูปเป็นการสร้างอินพุตที่ใช้สำหรับใส่ข้อมูลบ้านเลขที่เมื่อกรอกข้อมูลที่ต้องการแล้วจะต้องกดปุ่ม Search เพื่อที่จะส่งข้อมูลบ้านเลขที่ไปยังไฟล์ bot.php

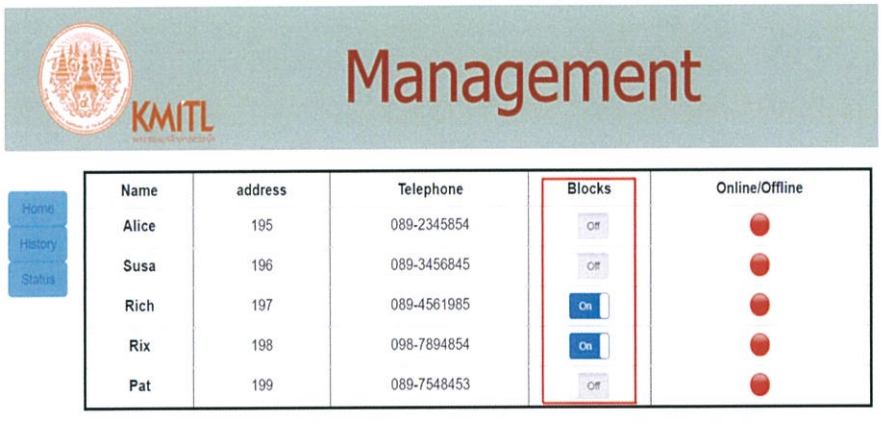
```
<?php
    $serverName     = "localhost";
    $userName       = "root";
    $userPassword         = "";
    $dbName      = "data";
    $conn = mysqli_connect($serverName,$userName,$userPassword,$dbName)or die("Error Connect to Database");
    mysqli_select_db($conn,"logfile");
    $address = $_POST['address'];
    $spcSQL = "SELECT * FROM logfile WHERE address='$address'";
    $result  =mysqli_query ($conn,$spcSQL) or die ("Error Query [".$spcSQL."]");
?>
```

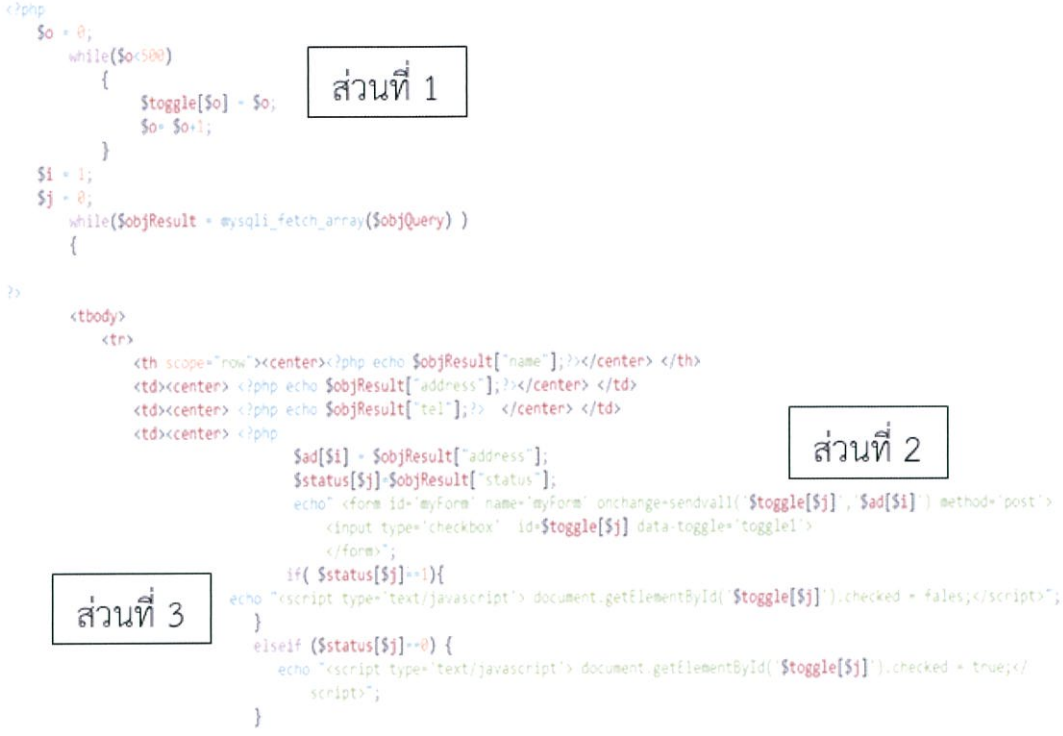
ไฟล์ bot.php จะเชื่อมต่อกับฐานข้อมูล logfile และจะรับค่า address ที่ถูกส่งมาเก็บไว้ในตัวแปล $address จากนั้นก็จะใช้ตัวแปลที่เก็บค่าบ้านเลขที่ ใช้เลือกทั้งแถวภายในฐานข้อมูลที่มีที่มีคอลัมน์ address=$address ในส่วนของการนำข้อมูลต่างๆมาแสดงผลนั้นจะทำงานดังเช่นหน้าต่างแสดงประวัติเหตุการณ์

3. ในส่วนของหน้าต่างแสดงสถานะของอุปกรณ์

ในส่วนนี้จะมีหน้าที่สำคัญอยู่ 2 ส่วนคือ

3.1 หน้าต่างแสดงสวิตช์เปิด/ปิดการใช้งานไมโครคอลโทลเลอร์ที่ควบคุมการส่งข้อมูลภายในบ้าน






```
<?php
    $o = 0;
        while($o<500)
            {
                $toggle[$o] = $o;
                $o= $o+1;
            }
    $i = 1;
    $j = 0;
        while($objResult = mysqli_fetch_array($objQuery) )
        {

?>
        <tbody>
            <tr>
                <th scope="row"><center><?php echo $objResult["name"];?></center> </th>
                <td><center> <?php echo $objResult["address"];?></center> </td>
                <td><center> <?php echo $objResult["tel"];?>  </center> </td>
                <td><center> <?php
                                $ad[$i] = $objResult["address"];
                                $status[$j]=$objResult["status"];
                                echo" <form id='myForm' name='myForm' onchange=sendval1('$toggle[$j]','$ad[$i]') method='post'>
                                    <input type='checkbox'  id=$toggle[$j] data-toggle='toggle1'>
                                    </form>";
                               if( $status[$j]==1){
                            echo "<script type='text/javascript'> document.getElementById('$toggle[$j]').checked = false;</script>";
                              }
                              elseif ($status[$j]==0) {
                                 echo "<script type='text/javascript'> document.getElementById('$toggle[$j]').checked = true;</
                                    script>";
                              }
```

ในการใช้งานสวิตช์เปิด/ปิดนั้นจะใช้การอ้างอิงข้อมูลภายในฐานข้อมูลเพื่อเชื่อมโยงการทำงานของหน้าจอมอนิเตอร์กับไมโครคอนโทลเลอร์โดยคอลัมน์ที่ใช้ในการอ้างอิงนั้นจะมีชื่อว่า status ถ้าต้องการให้ไมโครคอลโทลเลอร์ทำงาน ข้อมูลภายในฐานข้อมูลจะมีค่าเท่ากับ 0
ถ้าต้องการให้ไมโครคอลโทลเลอร์หยุดทำงาน ข้อมูลภายในฐานข้อมูลจะมีค่าเท่ากับ 1

ส่วนที่ 1 การเขียนโปรแกรมในส่วนที่ 1 นี้ทำขึ้นเพื่อสร้างตัวแปล $toggle0, $toggle1, $toggle3, ...จนถึง$toggle499 ขึ้นมาเก็บไว้

ส่วนที่ 2 เป็นส่วนที่ใช้สร้างสวิตช์เปิด/ปิดโดยจะนำตัวแปล $toggle[$o] ที่สร้างไว้มากำหนด id เฉพาะของสวิตช์แต่ละตัว จะเรียกมาใช้งานในรูปแบบ $toggle[$i] เมื่อมีการเปลี่ยนแปลงค่าของสวิตช์จะให้เรียกใช้ฟังก์ชั่น sendval1() พร้อมกับส่งตัวแปล id ของสวิตช์ ($toggle[$i]) และบ้านเลขที่ ($ad[$i]) ไปยัง sendval1()

```php
<?php

echo "<script>
                function sendval1(x,k){
                    var y = document.getElementById(x).checked;
                alert(y+k)
                 var add;
                 var mode;

                  $.ajax({
                      url : 'offlinesys.php',
                      data : {'add':k,'mode':y},
                      type : 'GET',
                      datatype : 'json',
                      success : function(data) {
                                  if(data == true){
                                     console.log('Success');
                                                  }

                                                  }
        });

            }
</script>";
?>
```

จากส่วนที่ 2 ที่ส่ง id ของสวิตช์ ($toggle[$i]) และบ้านเลขที่ ($ad[$i]) มา sendval() จะเปลี่ยนค่าของตัวแปลให้อยู่ในรูป x=$toggle[$i] และ k=$ad[$i] นำ id ของสวิตช์ซึ่งถูกเก็บไว้ในตัวแปล x มาเช็คสถานะของสวิตช์ว่าเป็น ture หรือ fales และเก็บไว้ในตัวแปล y จากนั้นส่งบ้านเลขที่ (k) และสถานะ (y) ไปยังไฟล์ offlinesys.php

```php
<?php
$mode=$_GET['mode'];
$address=$_GET['add'];
            $serverName     = "localhost";
            $userName       = "root";
            $userPassword         = "";
            $dbName       = "data";
            $conn = mysqli_connect($serverName,$userName,$userPassword,$dbName);
if ($mode=='true'){
    $sql1 = "UPDATE `security` SET `status`=0 ,event_time = CURRENT_TIMESTAMP WHERE address = '$address';";
        if($conn->query($sql1)==TRUE){
            echo "Success";
            $conn->close();

        }
}
else if ($mode=='false'){
    $sql1 = "UPDATE `security` SET `status`=1,`thief`=0,`fire`=0,`sos`=0  WHERE address = '$address';";
        if($conn->query($sql1)==TRUE){
            echo "Success";
            $conn->close();
        }
}
 ?>
```

ไฟล์ offlinesys.php จะรับข้อมูลสถานะของสวิตช์เก็บไว้ในตัวแปล $mode และบ้านเลขที่เก็บไว้ในตัวแปล $address หลังจากนั้นทำการเชื่อมต่อฐานข้อมูลและเช็คเงื่อนไข

ถ้า $mode มีค่าเท่ากับ true ให้ส่งข้อมูล 0 ไปยังคอลัมน์ status ที่แถวที่มีข้อมูลภายในคอลัมน์ address เท่ากับ $address

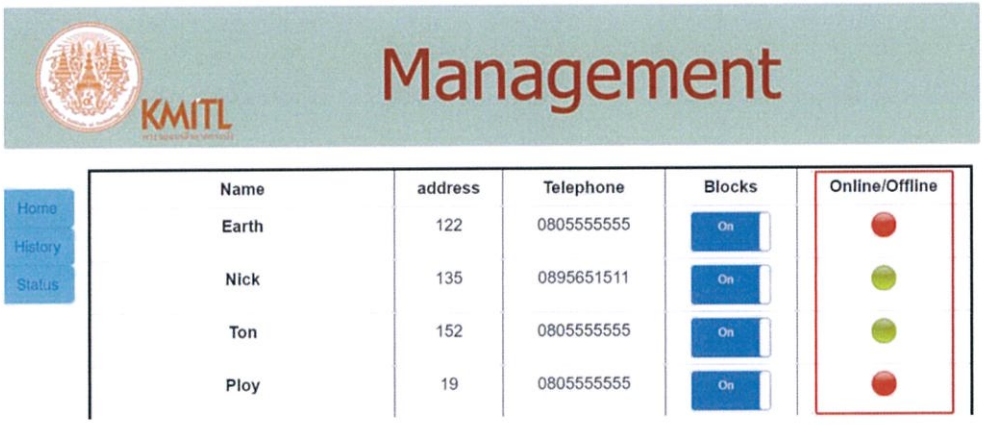
ถ้า $mode มีค่าเท่ากับ false ให้ส่งข้อมูล 1 ไปยังคอลัมน์ status ที่แถวที่มีข้อมูลภายในคอลัมน์ address เท่ากับ $address

ส่วนที่ 3 ในกรณีที่ไม่มีการไปเปลี่ยนแปลงสถานะของสวิตช์ระบบจะนำข้อมูลภายในฐานข้อมูลคอลัมน์ที่ชื่อว่า status มาเก็บไว้ในตัวแปล status[$j] ข้อมูลจะมีค่าได้แค่ 0 หรือ 1 นำตัวแปล status[$j] มาเขียนเงื่อนไขในการเช็ค

ถ้า status[$j] มีค่าเท่ากับ 1 ให้ไปเปลี่ยนรูปแบบสถานะของสวิตช์เป็น false

ถ้า status[$j] มีค่าเท่ากับ 0 ให้ไปเปลี่ยนรูปแบบของสวิตช์เป็น true

3.2 ส่วนนี้จะใช้สำหรับเช็คการเชื่อมต่อกันระหว่างไมโครคอลโทลเลอร์กับฐานข้อมูล




```php
<?php
    $host = "localhost";
    $username = "root";
    $password = "";
    $dbName     = "data";
    $objconncet= mysqli_connect($host,$username,$password,$dbName) or die("Error Connect to Database");
    $dateSQL = "SELECT * FROM security WHERE address ='$ad[$i]';";
    $dateQuery = mysqli_query ($objconncet,$dateSQL) or die ("Error Query [".$dateSQL."]");
    $dateResult = mysqli_fetch_array($dateQuery);
    $uptime = $dateResult["update_time"];
                date_default_timezone_set("Asia/Bangkok");
    $todaytime = date("Y-m-d G:i:s");
    $diff = abs(strtotime($uptime) - strtotime($todaytime));
    if($diff<120){
        echo "<img src='greencircle.png'  style='float:center;width:30px;height:30px '>";

    }else{
        echo "<img src='redcircle.png'  style='float:center;width:30px;height:30px '>";
    }
?>
```

ในส่วนนี้จะใช้ความต่างของเวลาในการตรวจสอบโดยไมโครคอลโทลเลอร์จะต้องส่งข้อมูลของเวลาในรูปแบบของ Unix Timestamp เข้าไปในฐานข้อมูลที่คอลัมน์ที่มีชื่อว่า update_time

เริ่มจากการเชื่อมต่อฐานข้อมูลชื่อ security ที่แต่ละบ้านเลขที่นั้นจะดึงข้อมูลเวลาจากคอลัมน์ update_time มาเก็บไว้ในตัวแปล $uptime และเครื่องเซิร์ฟเวอร์ก็จะดึงข้อมูลเวลาของเครื่องมาในรูปแบบ Unix Timestamp มาเก็บไว้ในตัวแปล $todaytime หาผลต่างของเวลาออกมา

จะได้เวลาในหน่วยวินาทีต่อจากนั้นเขียนโปรแกรมกำหนดเงื่อนไขว่าถ้าผลต่างของเวลาต่ำกว่า 2 นาที (120วินาที) ให้แสดงรูปภาพวงกลมเป็นสีเขียวแต่ถ้าผลต่างของเวลาเกิน 2 นาที (120วินาที) ให้แสดงรูปภาพวงกลมสีแดงซึ่งเงื่อนไขนี้หมายความว่าไมโครคอลโทลเลอร์ไม่ได้ส่งข้อมูลเวลามายังคอลัมน์ update_time เกิน 2 นาทีอาจจะเดาได้ว่าไม่สามารถเชื่อมต่อกับฐานข้อมูลได้