ชุดติดตั้งระบบรักษาความปลอดภัยโดยใช้ RASBERRY PI

SECURITY SYSTEM SET BY RASPBERRY PI

โดย

นายธีระพัฒน์ สืบเมืองซ้าย

นายเธียรดนัย แสนอ้วน

นายปรีดิ์ จรูญวัฒนเลาหะ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2557

ชุดติดตั้งระบบรักษาความปลอดภัยโดยใช้ RASPBERRY PI
SECURITY SYSTEM SET BY RASPBERRY PI

โดย

นายธีระพัฒน์	สืบเมืองซ้าย	54010640
นายเธียรดนัย	แสนอ้วน	54010645
นายปรีดิ์	จรูญวัฒนเลาหะ	54010798

อาจารย์ที่ปรึกษา

ดร.สมปอง วิเศษพานิชกิจ

ผศ.ดร.นภัทร สระเอี่ยม

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2557

ผ่านการตรวจรูปเล่มแล้ว

(.....)

อาจารย์ที่ปรึกษา

14/5/58

KMITL วิศวกรรมโทรคมนาคม Telecommunications Engineering

ผ่านการตรวจชิ้นงานแล้ว

(.....)

กรรมการผู้ตรวจชิ้นงาน

14/5/58

KMITL วิศวกรรมโทรคมนาคม Telecommunications Engineering

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า การพาณิชย์ หรือในเชิงธุรกิจ รวมทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2557
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ชุดติดตั้งระบบรักษาความปลอดภัยโดยใช้ RASPBERRY PI
SECURITY SYSTEM SET BY RASPBERRY PI

ผู้จัดทำ

1. นายธีระพัฒน์ สืบเมืองซ้าย 54010640
2. นายเธียรดนัย แสนอ้วน 54010645
3. นายปรีดิ์ จรูญวัฒนเลาหะ 54010798

..... อาจารย์ที่ปรึกษา
(ดร.สมปอง วิเศษพานิชกิจ)

..... อาจารย์ที่ปรึกษา
(ผศ.ดร.นภัทร สระเอี่ยม)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานฉบับนี้สำเร็จลงได้ด้วยดี เนื่องจากได้รับความกรุณาอย่างสูงจาก ดร.สมปอง วิเศษพานิชกิจ และผู้ช่วยศาสตราจารย์ ดร.นภัทร สระเอี่ยม อาจารย์ที่ปรึกษาโครงงานที่กรุณาให้คำแนะนำปรึกษาในการทำงานตลอดจนปรับปรุงข้อบกพร่องต่างๆ พร้อมเอื้อเฟื้อสถานที่ทดลองและจัดทำโครงงานด้วยความเอาใจใส่อย่างดียิ่ง คณะผู้จัดทำโครงงานตระหนักถึงความตั้งใจจริงและความทุ่มเทของอาจารย์ คณะผู้จัดทำขอขอบคุณเป็นอย่างสูงไว้ ณ ที่นี้

ขอกราบขอบพระคุณบิดา มารดา ผู้เป็นกำลังใจให้แก่คณะผู้จัดทำเสมอมา ตลอดจนผู้มีพระคุณทุกท่านที่มีส่วนร่วมในการช่วยเหลือ

สุดท้ายนี้หากมีข้อบกพร่องประการใดที่อาจจะเกิดขึ้นนั้น คณะผู้จัดทำขอน้อมรับ และยินดีที่จะรับฟังคำแนะนำจากทุกท่านที่ได้เข้ามาศึกษา เพื่อเป็นประโยชน์ในการพัฒนางานโครงงานต่อไป

นายธีระพัฒน์ สืบเมืองซ้าย
นายเธียรดนัย แสนอ้วน
นายปรีดิ์ จรูญวัฒนเลาหะ
ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชุดติดตั้งระบบรักษาความปลอดภัยโดยใช้ RASPBERRY PI
SECURITY SYSTEM SET BY RASPBERRY PI

โดย นายธีระพัฒน์ สืบเมืองซ้าย 54010640
นายเธียรดนัน แสนอ้วน 54010645
นายปรีดิ์ จรูญวัฒนเลาหะ 54010798

อาจารย์ที่ปรึกษา ดร.สมปอง วิเศษพานิชกิจ
ผศ.ดร.นภัทร สระเอี่ยม

บทคัดย่อ

โครงงานนี้เป็นการพัฒนาระบบรักษาความปลอดภัยภายในบ้าน โดยนำ Raspberry Pi ใช้เป็นส่วนควบคุมซึ่งประกอบด้วยเซ็นเซอร์แม่เหล็กที่จะใช้ติดตั้งตามหน้าต่างหรือประตู เซ็นเซอร์ตรวจจับควันและเซ็นเซอร์ตรวจจับการเคลื่อนไหวจะคอยตรวจจับสิ่งที่มีการเคลื่อนไหวแล้วทำการอัดวีดีโอในระยะเวลาที่กำหนดแล้วจึงทำการบันทึกวีดีโอ ซึ่งในการแจ้งเตือนผู้ใช้งานนั้น ระบบจะแจ้งเตือนผู้ใช้งานผ่าน SMS และบนหน้าเว็บ และมีการแสดงผลจากการอัดวีดีโอของเซ็นเซอร์ตรวจจับการเคลื่อนไหว พร้อมทั้งมีการแสดงสถานะเปิด-ปิดของเซ็นเซอร์ให้ผู้ใช้งานได้รับรู้ โดยที่ผู้ใช้งานนั้นสามารถควบคุมเปิด-ปิดเซ็นเซอร์ผ่านหน้าเว็บได้

ABSTRACT

This project develops security system inside home. We use raspberry pi to control 3 type of sensors. Magnetic sensor is a sensor at windows for detect and warn when someone who are trying to break in house. Smoke sensor is a sensor for detect gas and smoke and camera motion sensor can detect and record to video files. This function can be use streaming mode for real time. Theses also warn user by send SMS. Furthermore, user can turn on and turn off devices via web browser and check status of devices.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

		หน้า
กิตติกรรมประกาศ		I
บทคัดย่อ		II
สารบัญ		III
สารบัญรูป		VI
สารบัญตาราง		X
บทที่ 1	**บทนำ**	**1**
	1.1 ความเป็นมาและความสำคัญของปัญหา	1
	1.2 วัตถุประสงค์	1
	1.3 ขอบเขตของโครงงาน	1
บทที่ 2	**ทฤษฎีและหลักการที่เกี่ยวข้อง**	**2**
	2.1 Raspberry Pi	2
	2.2 Raspberry Pi Camera Module	6
	2.3 ภาษาไพธอน (Python)	7
	2.4 หลักการทำงานของ World Wide Web	8
	2.5 ภาษา PHP	9
	2.6 MySQL	10
	2.7 Apache	11
	2.8 OpenCV	11
	2.9 อุปกรณ์ตรวจจับควัน	11
	2.10 ตัวตรวจจับก๊าซ	14
	2.11 Image Processing	16
	2.12 การประมวลผลดิจิตอล	17
	2.13 ระบบสี RGB	18
	2.14 Grayscale	18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

		หน้า
	2.15 วิธีการ Threshold	19
	2.16 สถาปัตยกรรมในการแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล	20
	2.17 การสื่อสารแบบ SPI	21
	2.18 ชิป MCP3002	22
	2.19 เทคโนโลยีสตรีมมิ่ง	25
บทที่ 3	**การออกแบบและการจัดทำปริญญานิพนธ์**	**27**
	3.1 การออกแบบ	27
	3.2 เครื่องมือที่ใช้ในการทดลอง	45
	3.3 การจัดเก็บผลการทดลอง	45
บทที่ 4	**ผลการทดลอง**	**47**
	4.1 ผลการทดลองวัดค่าแรงดันไฟฟ้า	47
	4.2 ผลการทดลองการทำงานของเซ็นเซอร์แม่เหล็กและแสดงผลบนค่าแรงดันไฟฟ้าบนขา GPIO	52
	4.3 ผลการทดลองการทำงานของเซ็นเซอร์ตรวจจับควัน	54
	4.4 ผลการทดลองการทำงานของเซ็นเซอร์ตรวจจับการเคลื่อนไหว	57
	4.5 ผลการทดลองในหน้า Database ที่สร้างขึ้นใน PHPMyAdmin เมื่อทำการเปิด-ปิดอุปกรณ์	59
	4.6 การแจ้งเตือนบนหน้าเว็บเพจเมื่อ Database มีการเปลี่ยนแปลงสถานะ อันเนื่องมาจากการเปิด-ปิดอุปกรณ์	65
	4.7 ผลการทำงานในโหมดสตรีม	73
	4.8 หน้าเว็บเพจที่เก็บไฟล์วิดีโอ	74
	4.9 การแจ้งเตือนโดนการส่ง SMS	76
	4.10 การแจ้งเตือนโดยการโทรเข้าเบอร์ผู้ใช้งาน	77

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

		หน้า
บทที่ 5	**สรุปผลและข้อเสนอแนะ**	**78**
	5.1 สรุปผล	78
	5.2 ข้อเสนอแนะ	79
บรรณานุกรม		**81**
ภาคผนวก		**82**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่		หน้า
2.1	Raspberry Pi	2
2.2	ส่วนประกอบของ Raspberry Pi	4
2.3	ขา GPIO ของ Raspberry Pi	6
2.4	Raspberry Pi Camera Module	7
2.5	การทำงานของ Client – Server	8
2.6	การทำงานของ Client – Server ในลักษณะของการให้บริการฐานข้อมูล	8
2.7	การทำงานของ World Wide Web	9
2.8	หลักการทำงานของระบบอุปกรณ์ตรวจจับควันชนิดไอโอไนเซชั่น (Smoke Detector Ionization)	12
2.9	อนุภาคควันติดที่แผ่น	12
2.10	ระบบการทำงานของอุปกรณ์ตรวจจับควันชนิดโฟโตอิเล็กตริกแบบกีดขวางแสง (Light Obscuration)	13
2.11	ระบบการทำงานของอุปกรณ์ตรวจจับควันแบบหักเหแสง (Light Scattering)	13
2.12	โครงสร้างภายใน MQ-2	14
2.13	การวัดหาค่าความต้านทานที่ขา A และ B โดยตรง	15
2.14	การวัดโดยอ้อมโดยใช้กฎแบ่งแรงดัน Voltage Divider	15
2.15	กฎการแบ่งแรงดัน	15
2.16	การออกแบบวงจรของ MQ-2	16
2.17	พิกัดของระบบภาพดิจิตอลแบบ Double Array	17
2.18	ระบบสี RGB	18
2.19	ระบบ Grayscale	18
2.20	กราฟความสัมพันธ์ Binary และ Grayscale	19
2.21	ภาพในระบบ Grayscale	20
2.22	ภาพที่แบ่ง Threshold	20
2.23	Basic Successive Approximation ADC	20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่		หน้า
2.24	SPI Bus : SPI master and SPI Slave	21
2.25	ชิป MCP3002	22
2.26	ขาของชิป MCP3002	23
2.27	Block Diagram ภายในตัวชิป MCP3002	24
2.28	การทำงานของเทคโนโลยีสตรีมมิ่ง	25
3.1	บล็อกไดอะแกรมของชุดติดตั้งระบบรักษาความปลอดภัยโดย Raspberry Pi	27
3.2	เซ็นเซอร์แม่เหล็ก (ขณะมีแท่งแม่เหล็กในบริเวณหน้าสัมผัส)	28
3.3	อุปกรณ์เซ็นเซอร์แม่เหล็ก (ขณะไม่มีแท่งแม่เหล็กในบริเวณหน้าสัมผัส)	29
3.4	โฟลว์ชาร์ตการทำงานโปรแกรมเซ็นเซอร์แม่เหล็ก	30
3.5	การติดตั้งรีดสวิตช์กำหนดให้ A : แม่เหล็ก B : รีดสวิตช์ C : Raspberry Pi D : หน้าต่างหรือประตู	31
3.6	อุปกรณ์เซ็นเซอร์ตรวจจับควัน	32
3.7	โฟลว์ชาร์ตการทำงานของโปรแกรมเซ็นเซอร์ตรวจจับควัน	33
3.8	การออกแบบเซ็นเซอร์ตรวจจับการเคลื่อนไหว	34
3.9	โฟลว์ชาร์ตการทำงานโปรแกรมเซ็นเซอร์ตรวจจับการเคลื่อนไหว	35
3.10	เชื่อมต่ออุปกรณ์รวม	36
3.11	โฟลว์ชาร์ตโปรแกรมรวม	37
3.12	หน้าเว็บเพ็จ ยืนยันการทำงานของเว็บเซิร์ฟเวอร์	39
3.13	ฐานข้อมูลของรหัสผ่านเข้าหน้าควมคุมเซ็นเซอร์และแสดงผล	40
3.14	แผนผังการทำงานเว็บ	41
3.15	แผนผังการทำงาน การเข้าใช้งานของผู้ใช้งานผ่านหน้าเว็บ	42
3.16	แผนผังการทำงาน การรับค่าการควบคุมการเลือกใช้อุปกรณ์ของผู้ใช้งาน	43
3.17	แผนผังการทำงานของกล้องในโหมดสตรีม	44
4.1	โปรแกรมการวัดค่าลอจิกโดยใช้ภาษาไพธอน	47
4.2	แรงดันไฟฟ้าจาก DC Power Supply ต่อเข้า GPIO Raspberry Pi	48
4.3	ผลการทดลองเมื่อค่าแรงดันไฟฟ้าของ DC Power Supply ในช่วง 0 - 1.2 V	48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่		หน้า
4.4	ผลการทดลองเมื่อค่าแรงดันไฟฟ้าของ DC Power Supply มากกว่า 1.2 V	48
4.5	วัดค่าแรงดันไฟฟ้า Input ก่อนปล่อยเข้าสู่สายไฟ	49
4.6	วัดค่าแรงดันไฟฟ้า Output เมื่อออกจากสายไฟ ที่มีความยาว 5 เมตร	50
4.7	วัดค่าแรงดันไฟฟ้า Output เมื่อออกจากสายไฟ ที่มีความยาว 10 เมตร	50
4.8	วัดค่าแรงดันไฟฟ้า Output เมื่อออกจากสายไฟ ที่มีความยาว 15 เมตร	51
4.9	วัดค่าแรงดันไฟฟ้า Output เมื่อออกจากสายไฟ ที่มีความยาว 20 เมตร	51
4.10	ทดลองการใช้งานรีดสวิตช์ ขณะมีสนามแม่เหล็กอยู่ในบริเวณ	52
4.11	ทดลองการใช้งานรีดสวิตช์ ขณะไม่มีสนามแม่เหล็กอยู่ในบริเวณ	52
4.12	วัดค่าแรงดันไฟฟ้า Input รีดสวิตช์	53
4.13	วัดค่าแรงดันไฟฟ้า Output รีดสวิตช์ ขณะ Close Circuit	53
4.14	วัดค่าแรงดันไฟฟ้า Output รีดสวิตช์ ขณะ Open Circuit	54
4.15	เชื่อมต่อเซ็นเซอร์ตรวจจับควันกับ Raspberry Pi	55
4.16	ทดลองนำควันหรือก๊าซไวไฟเข้ามาบริเวณตรวจจับของอุปกรณ์	55
4.17	ทดสอบการทำงานของเซ็นเซอร์โดยใช้สายไฟยาว 10 เมตร	56
4.18	ระดับควันในหน้าโปรแกรม	56
4.19	ระดับควันมีค่ามากเกินที่กำหนด ทางโปรแกรมจะแจ้งเตือน	57
4.20	เริ่มต้นโปรแกรมตรวจจับความเคลื่อนไหว	58
4.21	โปรแกรม Motion เริ่มทำการ Detect	58
4.22	หน้าเว็บ Login เข้าเว็บหลักในการควบคุมอุปกรณ์และแจ้งเตือน	59
4.23	หน้าเว็บเลือกโหมดการแจ้งเตือน	60
4.24	หน้าเว็บดูประวัติการเข้าใช้งาน	61
4.25	หน้าเว็บดูประวัติการตรวจจับของเซ็นเซอร์	62
4.26	หน้าเว็บหลัก ใช้ในการการควบคุมและแจ้งเตือน(วิธีปิดอุปกรณ์)	63
4.27	หน้าเว็บหลัก ที่ใช้ในการการควบคุมและแจ้งเตือน(วิธีเปิดอุปกรณ์)	63
4.28	ผลของ Database เมื่อปิดอุปกรณ์ทั้งหมด	64
4.29	ผลของ Database เมื่อเปิดเฉพาะเซ็นเซอร์ตรวจจับควัน	64
4.30	ผลของ Database เมื่อเปิดเฉพาะเซ็นเซอร์ตรวจจับการเคลื่อนไหว	64
4.31	ผลของ Database เมื่อเปิดเฉพาะเซ็นเซอร์แม่เหล็กตัวที่ 1	64
4.32	ผลของ Database เมื่อเปิดเฉพาะเซ็นเซอร์แม่เหล็กตัวที่ 2	65

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่		หน้า
4.33	ผลของ Database เมื่อปิดอุปกรณ์ทั้งหมด	65
4.34	หน้าเว็บเลือกโหมดการแจ้งเตือน	66
4.35	ผลของ Database เมื่อเปิดอุปกรณ์ทั้งหมดและอยู่ในสถานะปกติ	66
4.36	หน้าเว็บเพจ เมื่อเปิดอุปกรณ์ทั้งหมดและอยู่ในสถานะปกติ	67
4.37	ผลของ Database เมื่อเซ็นเซอร์ตรวจจับควันมีการตรวจจับ	67
4.38	หน้าเว็บเพจ เมื่อเซ็นเซอร์ตรวจจับควันเกิดการตรวจจับ	68
4.39	ผลของ Database เมื่อเซ็นเซอร์ตรวจจับการเคลื่อนไหวเกิดการตรวจจับ	68
4.40	หน้าเว็บเพจ เมื่อเซ็นเซอร์ตรวจจับการเคลื่อนไหวเกิดการตรวจจับ	69
4.41	ผลของ Database เมื่อเซ็นเซอร์แม่เหล็กตัวที่ 1 เกิดการตรวจจับ	69
4.42	หน้าเว็บเพจ เมื่อเซ็นเซอร์แม่เหล็กตัวที่ 1 เกิดการตรวจจับ	70
4.43	ผลของ Database เมื่อเซ็นเซอร์แม่เหล็กตัวที่ 2 เกิดการตรวจจับ	70
4.44	หน้าเว็บเพจ เมื่อเซ็นเซอร์แม่เหล็กตัวที่ 2 เกิดการตรวจจับ	71
4.45	ผลของ Database เมื่ออุปกรณ์ทุกตัวเกิดการตรวจจับ	71
4.46	หน้าเว็บเพจ เมื่ออุปกรณ์ทุกตัวเกิดการตรวจจับ	72
4.47	วิธีการเข้าใช้โหมดสตรีมผ่านหน้าเว็บเพจ	73
4.48	หน้าเว็บเพจเมื่อเปิดใช้งานโหมดสตรีม	73
4.49	หน้าเว็บเพจเมื่อไม่มีไฟล์วีดิโอ	74
4.50	ผลของ Database เมื่อมีไฟล์วีดิโอ โดยจะบันทึกวัน เวลาและเดือนด้วย	74
4.51	หน้าเว็บเพจแสดงวีดิโอบนหน้าเว็บพร้อมที่จะทำการดาวน์โหลด	75
4.52	ผลของ Database เมื่อมีไฟล์วีดิโอ โดยจะบันทึกวัน เวลาและเดือนด้วย	75
4.53	หน้าโทรศัพท์มือถือเมื่อมีการแจ้งเตือนโดย SMS	76
4.54	แจ้งเตือนผ่านการโทรเข้าเบอร์ผู้ใช้งาน	77

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่		หน้า
2.1	ตารางบอกชนิดและการทำงานของการสื่อสารแบบ SPI	22
2.2	ตารางการต่อใช้งานขาของชิป MCP3002 กับขาของ Raspberry pi	24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันการติดต่อสื่อสารผ่านเครือข่ายอินเทอร์เน็ตนั้นกว้างขวางและจำเป็นมากในปัจจุบันจนแทบจะเป็นปัจจัยในการดำรงชีวิตไปแล้ว และด้วยความกว้างขวางและเข้าถึงง่ายของเครือข่ายนี้เองจึงสามารถนำไปต่อยอดใช้ประโยชน์ได้ในหลายๆด้าน ในด้านการรักษาความปลอดภัยในที่อยู่อาศัยนั้นได้มีการพัฒนาขึ้นมาอย่างต่อเนื่องเพื่อตอบสนองกับการใช้ชีวิตของมนุษย์ในปัจจุบันซึ่งเวลาและความเร่งรีบเป็นสิ่งสำคัญจนทำให้เกิดความไม่ปลอดภัยต่อทรัพย์สินและที่อยู่อาศัย จึงได้นำเอาประโยชน์ของเครือข่ายอินเทอร์เน็ตมารวมเข้ากับงานด้านรักษาความปลอดภัยในที่อยู่อาศัยเพื่อให้สามารถควบคุมอุปกรณ์ที่ใช้ในการรักษาความปลอดภัยได้โดยผ่านเครือข่ายอินเทอร์เน็ต และเพื่อให้มีการแจ้งเตือนการทำงานของอุปกรณ์ไปยังผู้ใช้งานให้รับรู้ได้

1.2 วัตถุประสงค์

1) เพื่อสร้างระบบรักษาความปลอดภัยที่มีการแจ้งเตือนผู้ใช้งาน
2) เพื่อสร้างระบบความปลอดภัยที่สามารถเปิด-ปิดตัวอุปกรณ์เซ็นเซอร์และควบคุมผ่าน Web Browser โดยผู้ใช้งานได้

1.3 ขอบเขตของปริญญานิพนธ์

สามารถใช้งานระบบรักษาความปลอดภัยที่สามารถแจ้งเตือนผู้ใช้งานผ่าน SMS และ Web Browser ได้ โดยเซ็นเซอร์ที่ติดตั้งมีเซ็นเซอร์แม่เหล็ก เซ็นเซอร์ตรวจจับควัน และเซ็นเซอร์ตรวจจับการเคลื่อนไหว ซึ่งผู้ใช้งานสามารถควบคุมการเปิด-ปิด การทำงานของเซ็นเซอร์ได้ตามความต้องการของผู้ใช้งาน ผ่านทางหน้า Web Browser

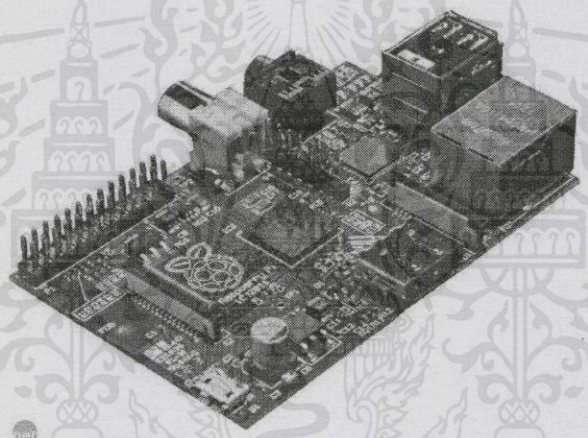
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 Raspberry Pi

Raspberry Pi คือ บอร์ดคอมพิวเตอร์ขนาด 32 บิตขนาดเล็ก บอร์ดไมโครคอมพิวเตอร์แบบแผ่นเดี่ยวที่บรรจุความสามารถไว้หลากหลายรองรับระบบปฏิบัติการ Linux บรรจุลงใน SD Card สำหรับการพัฒนาไปสู่บอร์ด Embedded Linux พร้อมจุดเชื่อมต่ออุปกรณ์อินพุตเอาท์พุตทั้งผ่านพอร์ต USB LAN HDMI ช่องสัญญาณภาพและ GPIO สำหรับต่อกับวงจรหรืออุปกรณ์อิเล็กทรอนิกส์ ลักษณะของบอร์ด Raspberry Pi แสดงได้ตามรูปที่ 2.1



รูปที่ 2.1 Raspberry Pi

2.1.1 คุณสมบัติทางเทคนิคของ Raspberry Pi

- ชิปควบคุมหลัก Broadcom BCM2835 หรือเทียบเท่า ซึ่งรวมซีพียู หน่วยประมวลกราฟิกหรือ GPU และหน่วยความจำ SDRAM ไว้ภายในตัวเดียวกัน

- หน่วยประมวลผลกลางหรือ CPU : ARM11 คอร์ ARM1176JZF-S ความเร็ว 700MHz

- หน่วยประมวลกราฟิกหรือ GPU : Broadcom Video Core IV หรือเทียบเท่า รองรับการแสดงผลผ่านจอภาพที่ใช้จุดต่อแบบ HDMI หน่วยความจำ SDRAM : 256MB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- จุดต่อ : USB 2.0 (2 พอร์ต) แจ๊ค RCA และ HDMI เอาต์พุตสัญญาณวีดีโอสำหรับต่อโทรทัศน์หรือจอแสดงผลที่มีจุดต่อแบบ RCA ตัวเมียหรือ HDMI จุดต่อเอาต์พุตเสียงเป็นแจ๊คหูฟัง 3.5 มิลลิเมตร จุดต่ออินเทอร์เน็ต คอนเน็กเตอร์หรือจุดต่อพอร์ตอินพุตเอาต์พุต (General Purpose Input/Output : GPIO) ที่มีขาต่อบัส SPI (Serial Peripheral Interface Bus) ขาสัญญาณรับส่งข้อมูลอนุกรมหรือ UART และซ็อกเก็ตของ SD Card สำหรับเสียบ SD Card ที่ติดตั้งระบบปฏิบัติการพร้อมแล้ว

- ความต้องการไฟเลี้ยง : +5 V 700 mA เป็นอย่างน้อย

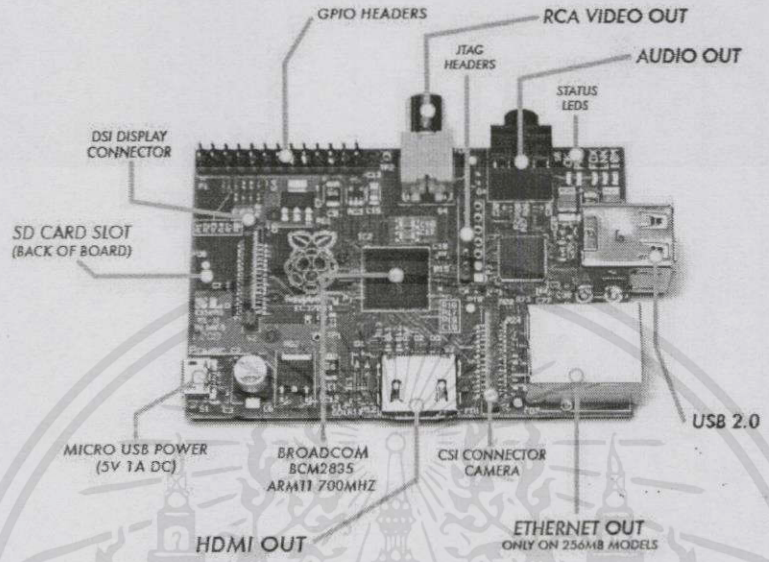
- ขนาด : 85.60 x 53.98 มิลลิเมตร หรือ 3.370 x 2.125 นิ้ว

2.1.2 ส่วนประกอบ Raspberry Pi

- HDMI Out (ใช้เชื่อมต่อพอร์ต HDMI เพื่อแสดงผลบนหน้าจอ)
- Ethernet Out (ใช้เชื่อมต่อพอร์ต RJ-45 เพื่อเชื่อมต่อกับอินเทอร์เน็ต)
- USB 2.0 (ใช้เชื่อมต่อสาย USB)
- Audio Out (ใช้เชื่อมต่ออุปกรณ์เสียง)
- RCA Video Out (ใช้เชื่อมต่อเอาต์พุตสัญญาณภาพ)
- CSI Connector Camera (ใช้เชื่อมต่อแบบ CSI Raspberry Pi Camera Module)
- Broadcom BCM2835 ARM11 700MHz (CPU และ SDRAM)
- SD Card Slot (ใช้เชื่อมต่อ SD Card)
- DSI Display Connector (ใช้เชื่อมต่อแบบ DSI สำหรับต่อกับจอแสดงผลเพิ่มเติม)
- GPIO Headers (ใช้เชื่อมต่อ GPIO)
- JTAG Header (ใช้เชื่อมต่อ JTAG)
- Status LED (สถานะไฟของ LED)

แสดงได้ตามรูปที่ 2.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 ส่วนประกอบของ Raspberry Pi

2.1.3 การเริ่มต้นใช้งานบอร์ด Raspberry pi

อุปกรณ์ที่ต้องใช้มีดังนี้

1) บอร์ด Raspberry Pi

2) SD Card สำหรับเก็บข้อมูลรวมไปถึงระบบปฏิบัติการควรมีความจุตั้งแต่ 4GB คลาส 4 ขึ้นไปซึ่งก็คือ SDHC การ์ดจะเป็นแบบ FAT32 หรือ NTFS ก็ได้

3) สาย microUSB ใช้สำหรับต่อกับแหล่งจ่ายไฟ +5V 700mA ที่มีจุดต่อแบบ USB ไม่ควรใช้แหล่งจ่ายไฟจากพอร์ต USB ของคอมพิวเตอร์ เนื่องจากมีความสามารถในการจ่ายกระแสไฟฟ้าไม่เพียงพอ

4) สาย HDMI หรือ RCA (อย่างใดอย่างหนึ่ง) สำหรับต่อกับจอแสดงผล

5) คีย์บอร์ดและเมาส์แบบ USB สำหรับควบคุมการทำงานของบอร์ด Raspberry Pi

6) สาย LAN (มีหรือไม่มีก็ได้) สำหรับเชื่อมต่อกับอินเทอร์เน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7) ลำโพงที่มีสายต่อเป็นปลั๊กหูฟัง 3.5 มิลลิเมตร เพื่อต่อกับขั้วเอาต์พุตสัญญาณเสียงของบอร์ด Raspberry Pi (มีหรือไม่มีก็ได้)

2.1.4 การติดตั้งข้อมูลลง Raspberry Pi

การติดตั้งระบบปฏิบัติการให้ Raspberry Pi จะต้องทำการติดตั้งลงใน SD Card

1) ดาวน์โหลด Win 32 Disk Imager

2) ดาวน์โหลด Raspbian "wheezy" เป็นไฟล์ระบบปฏิบัติการจาก http://www.raspberrypi.org/downloads

3) แตกไฟล์.rar ของ Raspbian "wheezy" ออกมาจะได้ไฟล์ .img

4) ต่อ SD Card เข้ากลับเครื่องคอมพิวเตอร์แล้วเปิดโปรแกรม Win32 Disk Imager ขึ้นมา โปรแกรมจะค้นหา Drive ของ SD Card ให้อัตโนมัติ

5) เลือกไฟล์ Raspbian "wheezy".img

6) คิดที่ปุ่ม Write เพื่อเขียนข้อมูลลงใน SD Card

7) ลงโปรแกรมจนเสร็จ ให้นำ SD Card ไปต่อกับ Raspberry Pi แล้วต่ออุปกรณ์ต่างๆ

8) เมื่อเริ่มใช้ Raspberry Pi เมื่อเริ่มทำงานระบบจะขึ้นหน้าล็อกอิน ให้ใส่ข้อมูลตามนี้ Username: pi Password: raspberry (หมายเหตุ: ใช้คำสั่ง startx เพื่อเข้าสู่หน้า Desktop)

2.1.5 การใช้งานกับระบบเครือข่ายอินเทอร์เน็ต

เมื่อต้องการใช้งานระบบที่เกี่ยวข้องกับการใช้อินเทอร์เน็ตให้ต่อสาย LAN RJ-45 เข้ากับตัว Raspberry Pi จะสามารถใช้งานได้ทันที โดยไม่ต้องทำการตั้งค่าใดๆ (ต้องมีการเปิดใช้งานจากผู้ให้บริการอินเทอร์เน็ตด้วย) เปิดหน้า LXTerminal แล้วใช้คำสั่ง sudo apt-get update เพื่ออัพเดทโปรแกรมล่าสุดจากนั้นใช้คำสั่ง sudo apt-get upgrade แล้วกดปุ่ม Y เพื่อยืนยันแล้วรอการดาวน์โหลดไฟล์ จากนั้นกด Y เพื่อลงทับไฟล์เดิม

2.1.6 การติดตั้งโปรแกรมอื่นลงใน SD Card

2.1.6.1 ในกรณีที่ต้องการลงโปรแกรมอื่นๆ เพิ่มเติมให้ใช้คำสั่ง sudo apt-get install (ชื่อโปรแกรม)

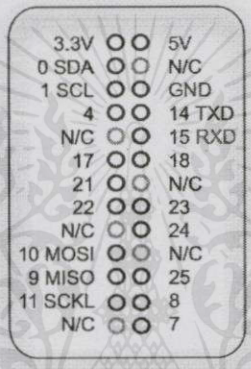
2.1.6.2 ตัวอย่างการลงโปรแกรม Chromium Web Browser sudo apt-get install chromium-browser

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.6.3 เมื่อติดตั้งเสร็จเรียกใช้งานโดย Start>Internet>Chromium Web Browser

2.1.7 การจัดการขา GPIO

GPIO ย่อมาจาก General Purpose Input/Output ขา Pin เหล่านี้ มีไว้ใช้ในงานทั่วๆไป โดยควบคุมผ่านซอฟต์แวร์ ขา GPIO นั้นจึงสามารถใช้ทำงานได้หลากหลายรูปแบบ ขึ้นอยู่กับผู้ใช้งานต้องการที่จะใช้งานอะไร รูปที่ 2.3 อธิบายขา GPIO ของ Raspberry pi



รูปที่ 2.3 ขา GPIO ของ Raspberry Pi

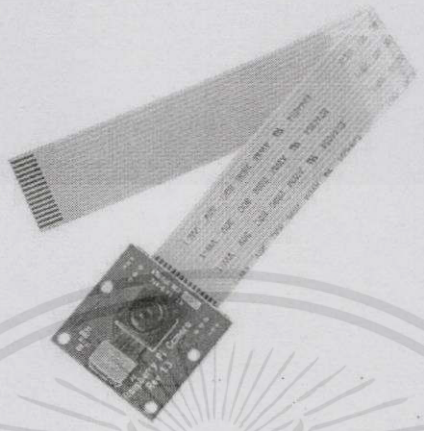
2.2 Raspberry Pi Camera Module

เป็นโมดูลกล้องที่ใช้งานร่วมกับ Raspberry Pi ผ่าน จุด Connector Camera มีความละเอียดสูงใช้งานได้หลากหลายรูปแบบ มีคุณสมบัติอธิบายเป็นข้อๆได้ดังนี้

- ความละเอียดสูง 5 ล้านพิกเซล
- ถ่ายวิดีโอที่ระดับความคมชัด 1080p, 720p และ 640x480 ด้วยอัตราแสดงผล 30 (1080p) 60 (720p และ 640x480) และ 90 (640x480) เฟรมต่อวินาที
- ขนาด 25 x 20 x 9 มิลลิเมตร
- น้ำหนัก 3 กรัม
- ต่อกับบอร์ด Raspberry Pi ด้วยบัส CSI (Common System Interface)

ลักษณะของโมดูลกล้องที่ใช้แสดงได้ตามรูปที่ 2.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 Raspberry Pi Camera Module

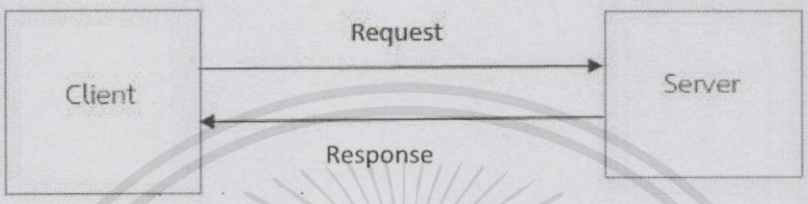
2.3 ภาษาไพธอน (Python)

ภาษาไพธอนพัฒนาโดยนาย Guido Van Rossum ออกแบบมาเพื่อให้ทำงานได้กับเว็บแอพพลิเคชั่นที่ลักษณะคล้ายกับภาษา Perl PHP JAVA และ ASP ภาษาไพธอนสามารถใช้ได้ทุกแพลตฟอร์ม กล่าวคือสามารถทำงานได้ทุกๆ CPU หลายๆระบบปฏิบัติการเพียงแต่ผู้เขียนโปรแกรมเขียนจากแพลตฟอร์มใดๆแล้วนำโปรแกรมที่ได้ไปให้ทำงานต่างแพลตฟอร์มกันได้ ไม่ต้องเสียค่าใช้จ่ายในการจัดซื้อโปรแกรมต้นฉบับ โดยปกติแล้วโปรแกรมภาษาทั่วๆไปจะต้องจัดซื้อโปรแกรมต้นฉบับเพื่อนำมาติดตั้งในราคาที่แพงมากแต่โปรแกรมภาษาไพธอนสามารถดาวน์โหลดจาก www.python.org ได้โดยตรงแล้วนำมาติดตั้งและศึกษาการใช้ด้วยตนเองเพราะเป็นโปรแกรมประเภท Open Source ภาษาไพธอนได้นำเอาข้อดีของโปรแกรมในอดีตมาไว้ด้วยกันเช่นภาษา C C++ Java และ Perl เป็นต้น จึงมีความปลอดภัยสูงเนื่องจากภาษาไพธอนทำงานอยู่ด้านเซิร์ฟเวอร์เป็นหลักเมื่อมีการร้องขอจากเครื่องไคลเอนต์ จะประมวลผลที่เครื่องเซิร์ฟเวอร์ ทำให้ผู้ใช้ทั่วไปไม่สามารถเข้าถึงเครื่องเซิร์ฟเวอร์ ได้โดยตรงจึงมีความปลอดภัยสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

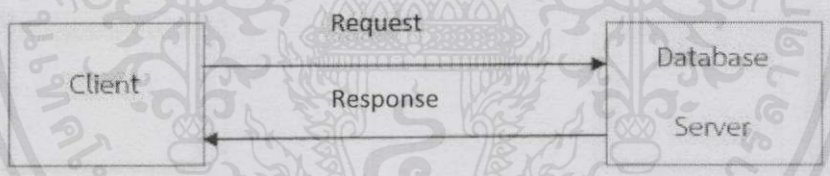
2.4 หลักการทำงานของ World Wide Web

การทำงานของ World Wide Web หรือ WWW จะมีลักษณะเดียวกันกับการทำงานในลักษณะไคลเอนต์-เซิร์ฟเวอร์ (Client-Server) คือ มีลักษณะของการเชื่อมต่อของเครื่องผู้ให้บริการ (Server) และเครื่องผู้ใช้บริการ (Client) พิจารณารูปต่อไปนี้



รูปที่ 2.5 การทำงานของ Client – Server

จากรูปจะเริ่มจากเครื่องผู้ใช้บริการทำการร้องขอ (Request) ใช้บริการจากเครื่องผู้ให้บริการ หลังจากเครื่องผู้ให้บริการจัดเตรียมข้อมูลหรือบริการตามที่ผู้ขอได้ร้องขอมาก็จะทำการตอบกลับ (Response) คืนไปยังเครื่องผู้ขอใช้บริการพิจารณาตัวอย่างของการทำงานแบบ Client-Server อีกตัวอย่างดังรูปต่อไปนี้

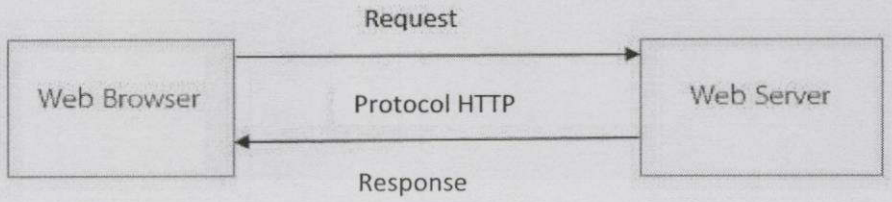


รูปที่ 2.6 การทำงานของ Client – Server ในลักษณะของการให้บริการฐานข้อมูล

การทำงานในลักษณะนี้จะมีเครื่องแม่ข่าย (Server) ทำหน้าที่จัดเก็บข้อมูลต่างๆ ทั้งหมดไว้ในฐานข้อมูล ดั้งนั้นเครื่องแม่ข่ายนี้เราจะเรียกว่า Database Server การทำงานก็จะเริ่มจากเครื่องลูกข่าย (Client) เช่นเดียวกัน คือ เครื่องลูกข่ายอาจจะทำการเพิ่ม ลบ แก้ไข หรือดึงข้อมูลซึ่งเก็บไว้ในฐานข้อมูล ข้อดีของการเชื่อมต่อแบบนี้คือ การจัดการฐานข้อมูลต่างๆ ซึ่งถูกจัดเก็บไว้เพียวที่เดียวทำให้ข้อมูลมีความเป็นหนึ่งเดียว คือข้อมูลจะไม่กระจัดกระจายและมีความถูกต้อง

ตัวอย่างการทำงานของ Client-Server ลำดับถัดไปจะกล่าวถึงลักษณะของการทำงานของ World Wide Web ซึ่งพิจารณาได้ดังรูปที่ 2.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 การทำงานของ World Wide Web

จากรูป เครื่องแม่ข่าย (Server) ในที่นี้จะให้บริการต่างๆที่เกี่ยวกับเว็บทั้งหมด ไฟล์เว็บเพจ รูปภาพ หรือโปรแกรมบนเว็บต่างๆ จะถูกจัดเก็บไว้ในเครื่องนี้ ซึ่งเราจะเรียกว่าเว็บเซิร์ฟเวอร์ (Web Server) เนื่องจากการแสดงผลจะถูกแสดงบนเว็บบราวเซอร์ (Web Browser) เช่น Internet Explorer หรือ Firefox เป็นต้น กระบวนการจะเริ่มจากเว็บบราวเซอร์จะทำการร้องขอหน้าเว็บโดยการพิมพ์ URL (Uniform Resource Locator) จากโปรแกรมบราวเซอร์ ซึ่งข้อมูลจะถูกกระทำผ่านโปรโตคอล HTTP (Hypertext Transfer Protocol) เครื่องที่ทำหน้าที่เว็บเซิร์ฟเวอร์ เมื่อได้รับคำร้องขอก็จะทำการจัดส่งหน้าเว็บเพจนั้นไปทางเว็บบราวเซอร์

2.5 ภาษา PHP

PHP ถูกสร้างขึ้นในปี ค.ศ.1994 โดย Rasmus Lerdorf โดยภาษา PHP เป็น Open Source ที่ผู้ใช้ทั่วไปสามารถดาวน์โหลด Source Code และโปรแกรมไปใช้ได้ฟรี

2.5.1 โครงสร้างของภาษา PHP

ภาษา PHP มีลักษณะเป็น Embedded Script ซึ่งหมายความว่าเราสามารถฝังคำสั่ง PHP ไว้ในเว็บเพจร่วมกับคำสั่ง (Tag) ของ HTML ได้ และสร้างไฟล์ที่มีนามสกุลเป็น .php ซึ่งไวยากรณ์ที่ใช้ใน PHP เป็นการนำรูปแบบของภาษาต่างๆ มารวมกันได้แก่ C Perl และ Java

2.5.2 ความสามารถของภาษา PHP

- เป็นภาษาที่มีลักษณะเป็นแบบ Open source ผู้ใช้สามารถ Download และนำ Source code ของ PHP ไปใช้ได้โดยไม่เสียค่าใช้จ่าย PHP เป็นสคริปต์แบบ Server Side Script ดังนั้นจึงทำงานบนเว็บเซิร์ฟเวอร์ ไม่ส่งผลกับการทำงานของเครื่อง Client โดย PHP จะอ่านโค้ด และทำงานที่เซิร์ฟเวอร์ จากนั้นจึงส่งผลลัพธ์ที่ได้จากการประมวลผลมาที่เครื่องของผู้ใช้ในรูปแบบของ HTML

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- PHP สามารถทำงานได้ในระบบปฏิบัติการที่ต่างชนิดกัน เช่น Unix Windows และ Mac OS ได้อย่างมีประสิทธิภาพ เนื่องจาก PHP เป็นสคริปต์ที่ต้องทำงานบนเซิร์ฟเวอร์ ดังนั้นคอมพิวเตอร์สำหรับเรียกใช้คำสั่ง PHP จึงจำเป็นต้องติดตั้งโปรแกรมเว็บเซิร์ฟเวอร์ไว้ด้วย เพื่อให้สามารถประมวลผล PHP ได้

- PHP สามารถทำงานได้ในเว็บเซิร์ฟเวอร์หลายชนิด เช่น Internet Information Service (IIS) และ Apache เป็นต้น

- PHP มีความสามารถในการทำงานร่วมกับระบบจัดการฐานข้อมูลที่หลากหลาย

- PHP อนุญาตให้ผู้ใช้สร้างเว็บไซต์ซึ่งทำงานผ่านโปรโตคอลชนิดต่างๆ ได้ เช่น LDAP IMAP SNMP POP3 และ HTTP เป็นต้น

2.6 MySQL

ฐานข้อมูลเป็นองค์ประกอบสำคัญในการพัฒนาแอพพลิเคชั่น (Application) เนื่องจากระบบงานต่างๆ จะมีการจัดเก็บข้อมูลไว้ในฐานข้อมูล เพื่อให้สามารถเข้าถึงข้อมูลที่ฐานข้อมูลได้ โดยเข้าถึงข้อมูลในที่นี้หมายความว่า การที่ระบบสามารถงานสามารถเพิ่ม ลบ แก้ไข และจัดทำรายงานได้ ซึ่งถือว่าเป็นฟังก์ชั่นพื้นฐานของระบบงานต่างๆ

MySQL เป็นโปรแกรมจัดการฐานข้อมูล ถูกพัฒนาให้ตัวโปรแกรมสามารถทำงานร่วมกับระบบปฏิบัติการ Windows หรือ Linux ได้ นอกจากนี้ตัวโปรแกรม MySQL ยังเป็นโปรแกรมประเภท Open Source ซึ่งทำให้นักพัฒนาทั้งหลายสามารถเข้าถึงได้ง่าย

2.6.1 การใช้งาน MySQL

MySQL เป็นที่นิยมใช้กันมากสำหรับฐานข้อมูลสำหรับเว็บไซต์ และนิยมใช้งานร่วมกับภาษาโปรแกรม PHP ซึ่งมักจะใช้งานควบคู่กัน นอกจากนี้ยังมีหลายภาษาโปรแกรมที่สามารถทำงานร่วมกับฐานข้อมูล MySQL อาทิเช่น C C++ Pascal C# Java Perl PHP และ Python เป็นต้น

2.6.2 ส่วนเชื่อมต่อกับภาษาการพัฒนาอื่น

มีส่วนติดต่อ (Interface) เพื่อเชื่อมต่อกับภาษาอื่นๆในการพัฒนาเพื่อให้เข้าถึงฟังก์ชั่นการทำงานกับฐานข้อมูล MySQL ได้

2.6.3 การจัดการฐานข้อมูล MySQL ด้วย PHPMyAdmin

MySQL มีข้อเสียในการใช้งานคือ ความยากลำบากในการจัดการฐานข้อมูล เช่น ต้องเข้าใจและเรียกใช้คำสั่งที่ถูกต้อง จึงมีการสร้างโปรแกรมสำเร็จรูปที่มีคุณสมบัติในการทำงานในลักษณะ Graphic User Interface (GUI) โดยที่ไม่ต้องรู้คำสั่งในการจัดการฐานข้อมูลเลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผู้ใช้มีหน้าที่เพียงคลิกเลือกคำสั่งต่างๆที่จัดเตรียมไว้ให้แล้ว โปรแกรมที่ใช้จัดการฐานข้อมูลมีอยู่มากมาย แต่ที่นิยมใช้ที่สุดก็คือโปรแกรม PHPMyAdmin ซึ่งเป็นโปรแกรมที่ออกแบบมาช่วยอำนวยความสะดวกในการจัดการฐานข้อมูล MySQL โดยสามารถจัดการการเข้าถึงฐานข้อมูลได้ทั้งหมด คือ สร้าง-ลบฐานข้อมูล สร้างตาราง แก้ไขรายละเอียดของตาราง เพิ่มเรคอร์ดใหม่ในตาราง ดูข้อมูลภายในตาราง แก้ไขข้อมูลของเรคอร์ด ลบตาราง ส่งออกฐานข้อมูล และนำเข้าฐานข้อมูล

2.7 Apache

Apache เป็นเว็บเซิร์ฟเวอร์ (Web Server) ที่มีผู้ใช้กันทั่วโลก ซึ่งปัจจุบันจัดได้ว่าเป็นเว็บเซิร์ฟเวอร์ที่น่าเชื่อถือเป็นอย่างมาก Apache เป็นซอฟต์แวร์ที่อยู่ในลักษณะของ Open Source ที่เปิดให้บุคคลทั่วไปได้สามารถเข้ามาร่วมพัฒนาให้เกิดประโยชน์ได้ และสามารถทำงานร่วมกับภาษาอื่นได้ แทนที่จะเป็นเพียงเซิร์ฟเวอร์ที่ให้บริการเพียงแค่ HTML อย่างเดียว

2.8 OpenCV (Open Source Computer Vision)

เป็นไลบรารีที่ช่วยในการพัฒนาซอฟต์แวร์ด้านคอมพิวเตอร์วิทัศน์ (Computer Vision) OpenCV เป็นซอฟต์แวร์ Open Source ภายในไลบรารีของ OpenCV ถูกพัฒนาขึ้นมาจาก ภาษา C และ C++ OpenCV สามารถรันซอฟต์แวร์ภายใต้ระบบปฏิบัติการ Linux Window หรือ MAC OS ได้ และสามารถพัฒนาแอพพลิเคชั่นจาก OpenCV โดยการใช้ภาษาอื่นที่ไม่ใช่ C หรือ C++ ได้ เช่น Python Ruby เป็นต้น

2.9 อุปกรณ์ตรวจจับควัน (Smoke Detector)

อุปกรณ์ตรวจจับควัน (Smoke Detector) เป็นอุปกรณ์ที่ทำหน้าที่ตรวจจับอนุภาคของควันได้ หลักการของอุปกรณ์ตรวจจับควันจะทำงานเมื่อมีอนุภาคควันลอยเข้าไปในอุปกรณ์ตรวจจับควัน อนุภาคควันจะเข้าไปกีดขวางวงจรไฟฟ้า หรือกีดขวางระบบแสงในวงจร

2.9.1 ชนิดของอุปกรณ์จับควัน

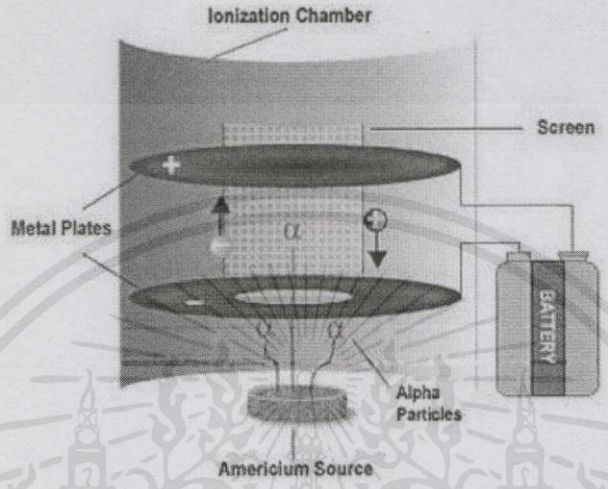
แบ่งได้เป็น 2 ชนิดคือ ชนิดไอโอไนเซชัน (Ionization) ชนิดโฟโตอิเล็กตริก (Photoelectric)

2.9.1.1 อุปกรณ์ตรวจจับควันชนิดไอโอไนเซชัน(Smoke Detector Ionization Type)

ภายในเป็นกล่อง (Chamber) มีแผ่นโลหะที่มีขั้วไฟฟ้าต่างกัน มีสารกัมมันตภาพรังสีซึ่งทำหน้าที่กระตุ้นอากาศภายในให้เกิดการแตกตัวเป็นไอออน โดยไอออนในกล่องจะทำหน้าที่เป็นตัวนำไฟฟ้าให้กระแสไฟฟ้าไหลผ่านทั้งสองขั้ว เมื่อเกิดควันเข้าไปในกล่องจะ

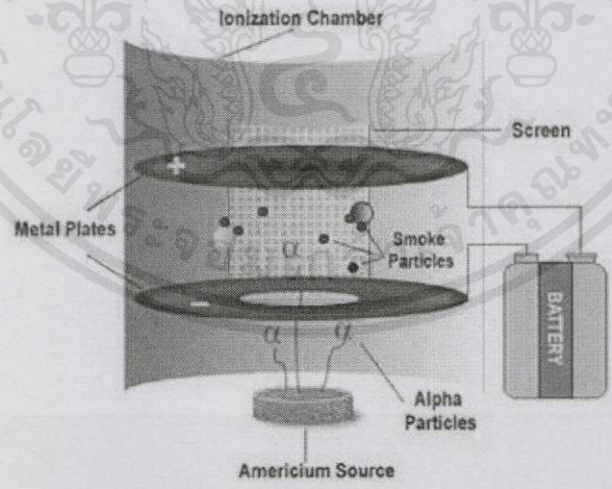
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำให้ค่าความนำไฟฟ้าของอากาศลดลง และลดลงเรื่อยๆตามปริมาณควันจนถึงค่าที่กำหนดไว้ ระบบก็จะทำงาน



รูปที่ 2.8 ระบบการทำงานของอุปกรณ์ตรวจจับควันชนิดไอโอไนเซชัน (Smoke Detector Ionization)

จากภาพที่ 2.8 อธิบายได้ว่า เริ่มแรกยังไม่มีอนุภาพของควัน ระบบจะยังไม่ทำงาน



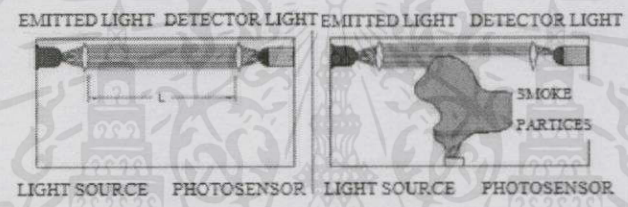
รูปที่ 2.9 อนุภาคควันติดที่แผ่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากภาพที่ 2.9 อธิบายได้ว่าอนุภาคควัน(จุดดำๆ)มาติดที่แผ่นจะเป็นตัวขัดขวางกระแสไฟฟ้าทำให้กระแสไฟฟ้าลดต่ำลงจนถึงจุดที่ระบบทำงาน

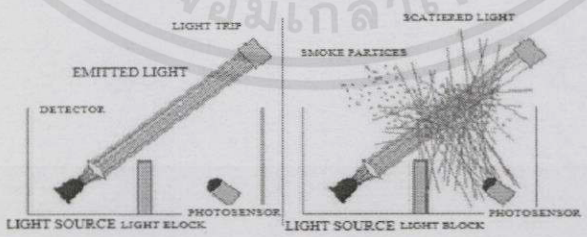
2.9.1.2 อุปกรณ์ตรวจจับควันชนิดโฟโตอิเล็กตริก (Smoke Detector Photoelectric Type)

มีหลักการทำงานสองแบบคือ แบบใช้ควันกีดขวางแสงและแบบหักเหของแสงแบบแรกอุปกรณ์ตรวจจับควัน ชนิดโฟโตอิเล็กตริก แบบควันกีดขวางแสง (Light Obscuration) ทำงานโดยใช้แหล่งกำเนิดแสง (Emitted Light) ยิงเข้าที่ตัวรับแสง (Detector Light) เมื่อไม่มีควันไฟ ปริมาณแสงจะคงที่ที่ค่าหนึ่งเสมอ เมื่ออนุภาคควันเข้ามาดังรูปขวามืออนุภาคควันจะเข้าไปกีดขวางลำแสงที่ส่องเข้าตัวรับ เมื่อค่าต่ำลงเรื่อยๆจนถึงค่าที่กำหนดไว้ ระบบก็จะทำงาน



รูปที่ 2.10 ระบบการทำงานของอุปกรณ์ตรวจจับควันชนิดโฟโตอิเล็กตริกแบบกีดขวางแสง (Light Obscuration)

ในส่วนของอุปกรณ์ตรวจจับควัน ชนิดโฟโตอิเล็กตริกแบบหักเหแสง (Light Scattering) ทำงานโดยมีแหล่งกำเนิดแสง แต่จะไม่ยิงไปที่ตัวรับแสงโดยตรง จะอาศัยหลักการที่ว่าเมื่อมีอนุภาคควันเข้ามาในอุปกรณ์ อนุภาคควันจะหักเหแสงบางส่วนไปที่ตัวรับแสง เมื่อมีควันมากขึ้นแสงก็จะหักเหเข้าตัวรับแสงมากขึ้นจนถึงจุดๆหนึ่งที่ระบบจะทำงาน

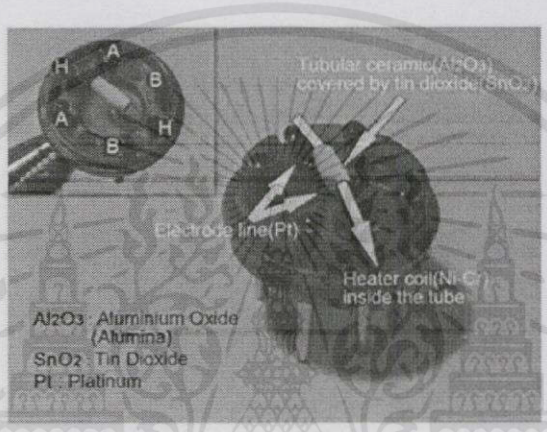


รูปที่ 2.11 ระบบการทำงานของอุปกรณ์ตรวจจับควันแบบหักเหแสง(Light Scattering)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10 ตัวตรวจจับก๊าซ (MQ-2 Flammable Gas & Smoke Sensor)

MQ-2 เป็นเซ็นเซอร์ตรวจสอบปริมาณก๊าซไวไฟและควันเช่น LPG I-Butane Propane Methane Alcohol Hydrogen ในอากาศที่มีความเข้มข้น 300-10,000 PPM (Part Per Million) โครงสร้างของ MQ-2 ประกอบด้วยหลอด Al_2O_3 และ SnO_2 ซึ่งเมื่อจ่ายพลังงานให้ MQ-2 ที่ขา H จะทำให้เกิดความร้อนขึ้นที่ขดลวด เมื่อก๊าซไวไฟต่างๆเข้ามาทำปฏิกิริยาจะทำให้ค่าความต้านทานที่เกิดขึ้นระหว่างขา A และ B (Rs = Sensing Resistance) ลดลง



รูปที่ 2.12 โครงสร้างภายใน MQ-2

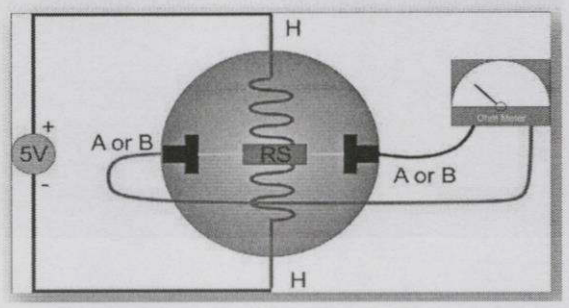
จากรูป 2.12 เมื่อจ่ายพลังงานให้กับขา H จะทำให้เกิดพลังงานความร้อนเพื่อให้สารเคมีภายในตัวเซ็นเซอร์สามารถทำปฏิกิริยากับก๊าซไวไฟต่างๆได้ เมื่อ MQ-2 ตรวจจับก๊าซไวไฟต่างๆได้ จะทำให้ค่าความต้านทานระหว่างขา A และขา B เปลี่ยนแปลง (ขา A และ B เป็นขาที่ไม่ตายตัว สามารถกำหนดเองได้โดยหากเลือกขาใดเป็นขา A ขาที่อยู่ตรงข้ามก็จะเป็นขา B) โดยเมื่อ MQ-2 ตรวจจับปริมาณก๊าซพิษต่างๆ ได้มากจะทำให้ค่าความต้านทาน Rs ลดลง หรือ ค่าความต้านทานแปรผกผันกับปริมาณของ ก๊าซไวไฟต่างๆ

2.10.1 วิธีการหาค่าความต้านทาน Rs (Sensing Ressistance)

การหาค่าความต้านทานของ Rs สามารถทำได้ 2 วิธีคือ

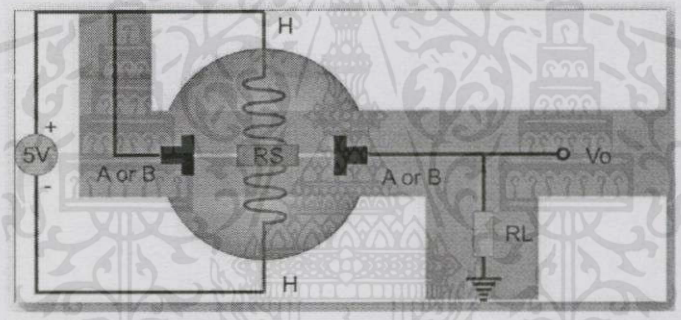
1) สามารถใช้ Ohm Meter วัดหาค่าความต้านทานที่ขา A และ B ได้โดยตรงดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



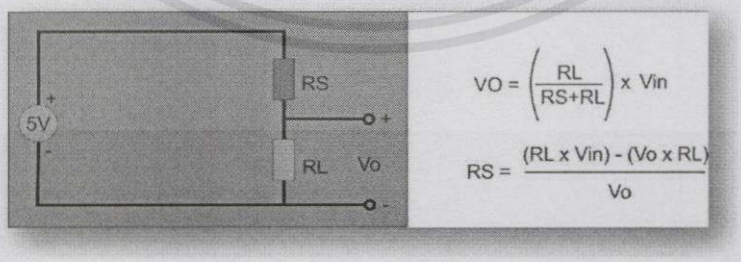
รูปที่ 2.13 การวัดหาค่าความต้านทานที่ขา A และ B โดยตรง

2) เป็นวิธีการวัดโดยอ้อม โดยใช้กฎแบ่งแรงดัน Voltage Divider



รูปที่ 2.14 การวัดโดยอ้อม โดยใช้กฎแบ่งแรงดัน Voltage Divider

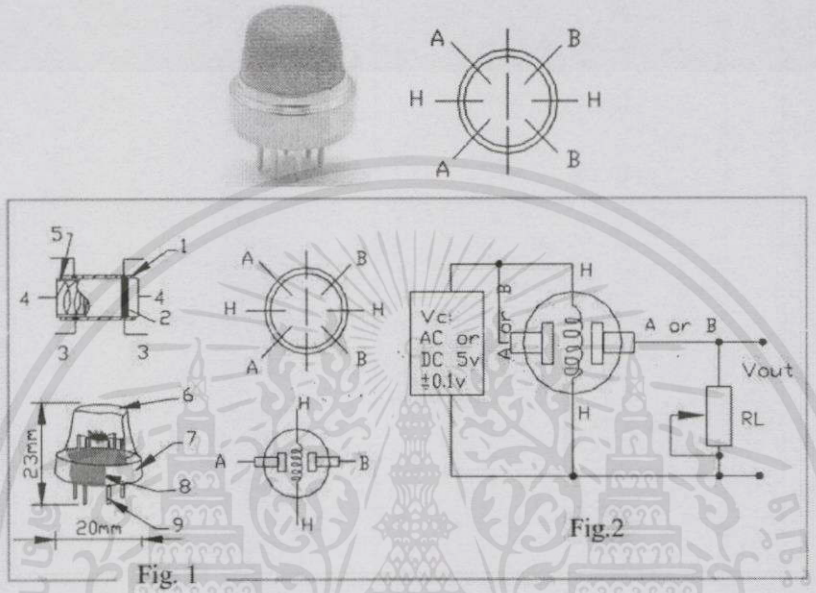
จากรูปที่ 2.14 จะเห็นได้ว่าเมื่อจ่ายกระแสไฟฟ้าเข้ามาที่ขา A หรือ B ทางด้านซ้าย จะเป็นไปตามกฎแบ่งแรงดัน ซึ่งกฎการแบ่งแรงดันอธิบายได้ดังรูปที่ 2.15



รูปที่ 2.15 กฎการแบ่งแรงดัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10.2 การออกแบบวงจรใช้งาน MQ-2



รูปที่ 2.16 การออกแบบวงจรของ MQ-2

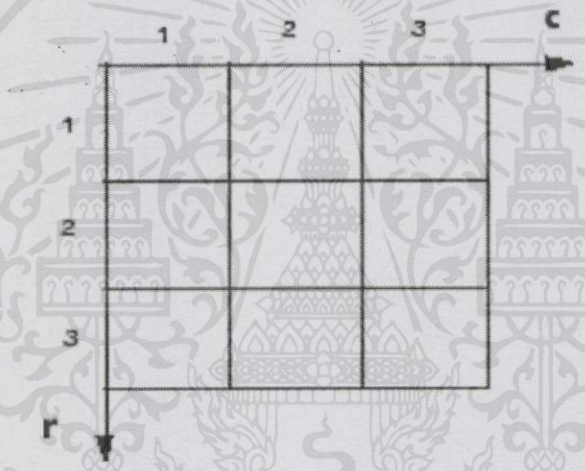
2.11 Image Processing

Image Processing คือการนำเสนอรายละเอียดของวัตถุ (Object) โดยการจับเอาหรือถ่ายเท (Maniltipulation) รายละเอียดของภาพที่ได้จากวัตถุเพื่อนำรายละเอียดของภาพเหล่านั้นไปทำประโยชน์ให้มากขึ้น วิธีการทาง Image Processing จะต่างกับวิธีการทาง Computer Graphic กล่าวคือ วิธีการทาง Computer Graphic ตัวคอมพิวเตอร์เองที่จะเป็นตัวที่สร้างภาพ แต่เทคนิคทาง Image Processing นั้นใช้คอมพิวเตอร์สำหรับจัดเก็บข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.12 การประมวลผลภาพดิจิตอล

การประมวลผลภาพดิจิตอล จะเกี่ยวข้องกับการแปลงข้อมูลภาพให้อยู่ในรูปแบบข้อมูลดิจิตอล เพื่อที่จะสามารถนำเอาข้อมูลนี้ไปผ่านกระบวนการต่างๆด้วยคอมพิวเตอร์ได้ ซึ่งการทำงานของคอมพิวเตอร์ระบบการรับข้อมูลเข้าหรือส่งข้อมูลออกจะอยู่ในรูปแบบดิจิตอลเท่านั้น ในการประมวลผลภาพดิจิตอล เมื่อระบบได้รับข้อมูลภาพเข้าไปแล้วจะทาการคำนวณและส่งออกมาเป็นข้อมูลที่ใช้แทนข้อมูลภาพดิจิตอลเหล่านั้น การเก็บข้อมูลภาพลงหน่วยความจำของคอมพิวเตอร์สามารถทำได้โดยการจองหน่วยความจำของเครื่องไว้ในรูปของตัวแปร Array ดังรูปที่ 2.17 โดยค่าในแต่ละช่องจะแสดงถึงคุณสมบัติของจุดภาพ (Pixel) และตำแหน่งของช่อง Array เป็นตัวกำหนดตำแหน่งของจุดภาพ



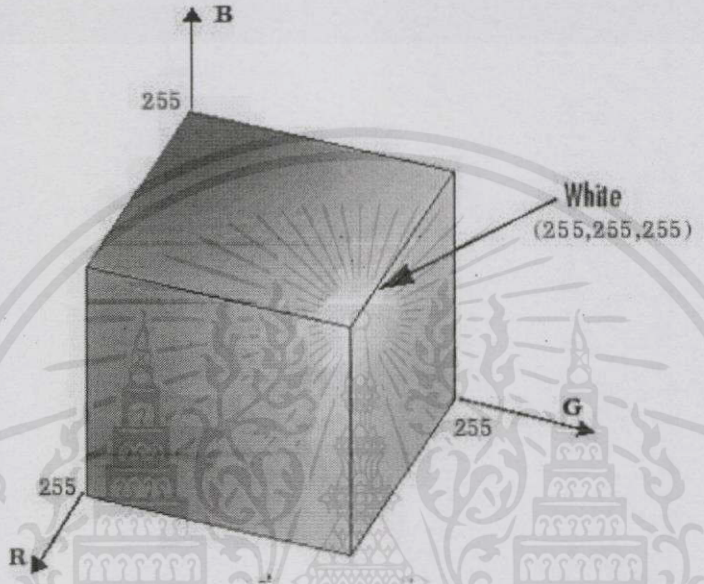
รูปที่ 2.17 พิกัดของระบบภาพดิจิตอลแบบ Double Array

การแปลงภาพเป็นภาพขาวดำมีขั้นตอนคือ แปลงภาพสีเป็น Grayscale แล้วแปลงภาพ Grayscale เป็นภาพขาวดำอีกที การแปลงภาพ Grayscale เป็นภาพขาวดำ จำเป็นต้องกำหนดค่า Threshold เพื่อให้ได้ภาพขาวดำที่เหมาะสม ค่า Threshold มีไว้สำหรับกำหนดว่าค่าความสว่างของรูปในแต่ละ Pixel เมื่อแปลงเป็นภาพขาวดำแล้ว Pixel ดังกล่าวควรจะเป็น สีดำ หรือสีขาว ตัวอย่างเช่น เรากำหนดค่า Threshold ไว้ที่ 128 แล้วอ่านค่าความสว่างใน Pixel แรกของรูปได้ 200 ดังนั้นเมื่อทำการแปลงเป็นภาพขาวดำแล้ว Pixel แรกนี้จะต้องเป็นสีขาว เป็นต้น หลายคนอาจจะเข้าใจผิดว่าภาพ Grayscale เป็นภาพขาวดำ แต่ความจริงแล้วไม่ใช่ ภาพ Grayscale เป็นภาพที่มีระดับของความสว่างอยู่ โดยทั่วไปจะอยู่ที่ 0-255 ภาพ Gray scale ก็คือภาพที่มีค่าสี RGB เท่ากันหมด ส่วนภาพขาวดำมีแค่ 0 กับ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.13 ระบบสี RGB

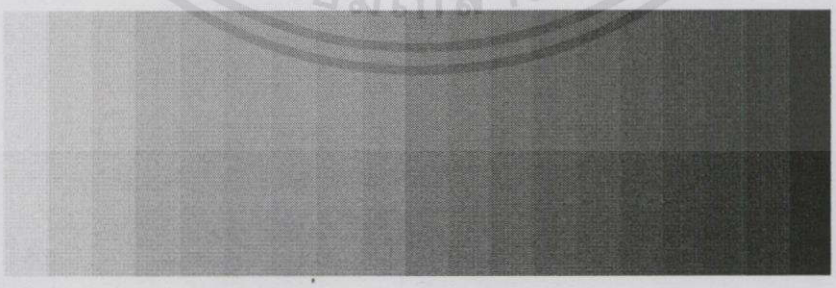
ระบบสี RGB คือระบบที่มีค่าของสีแดง สีเขียว และ สีน้ำเงิน ค่าใดค่าหนึ่งหรือหลายๆค่าร่วมกัน โดยแต่ละสีจะมีค่าตั้งแต่ 0-255 ดังรูป 2.18



รูปที่ 2.18 ระบบสี RGB

2.14 ระบบ Grayscale

ระบบ Grayscale คือระบบที่มีค่าของสีแดง สีเขียวและสีน้ำเงินเท่ากัน ภาพจึงออกมาในโทนสีขาวดำ



รูปที่ 2.19 ระบบ Grayscale

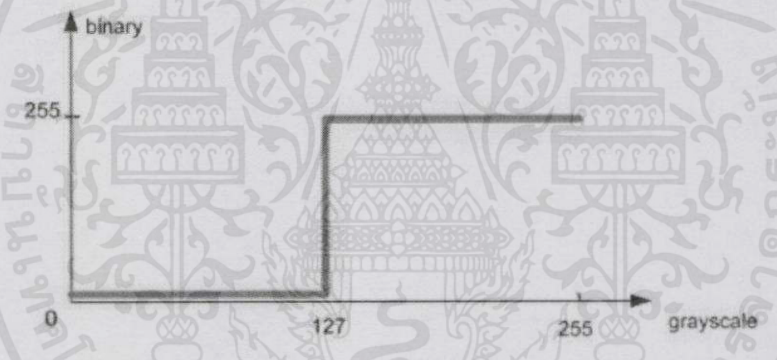
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.15 วิธีการ Threshold

Threshold เป็นการแปลงภาพระบบ Grayscale ซึ่งมีค่าอยู่ระหว่าง 0-255 ให้ เป็นภาพที่มีค่าเพียงสองระดับ (Binary Image) โดยมีเงื่อนไขว่า ถ้าความเข้มแสงของจุดภาพใดมีค่าต่ำกว่าหรือเท่ากับค่า Threshold ให้จุดภาพนั้นมีค่าเป็น 0 หรือเป็นสีดำ และจุดภาพ ใดที่มีค่าสูงกว่าค่า Threshold ให้จุดภาพนั้นมีค่าเป็น 1 หรือสีขาว Threshold เป็นการทำค่าของพิกเซลมีค่าเพียง 2 สถานะคือขาวและดำ มีสูตรดังนี้

$$\text{Grayscale} = \begin{cases} 255 & ; \text{ if grayscale } > 127 \\ 0 & ; \text{ else} \end{cases}$$

โดยแสดงความสัมพันธ์ได้ดังกราฟ



รูปที่ 2.20 กราฟความสัมพันธ์ Binary และ Grayscale

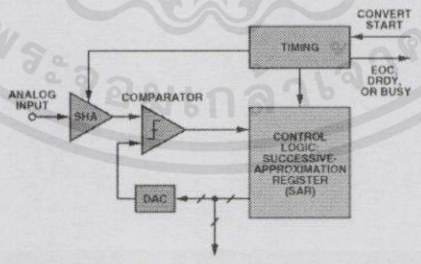
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.21 ภาพในระบบ Grayscale

รูปที่ 2.22 ภาพที่แบ่ง threshold

2.16 สถาปัตยกรรมในการแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล

ในการแปลงสัญญาณจากอนาล็อกเป็นดิจิตอลนั้นมีหลายแบบ แต่ในที่นี้ได้เลือกใช้การแปลงแบบสัญญาณแบบ Successive Approximation ซึ่งเป็นหลักการแปลงสัญญาณชนิดประมาณค่า เป็นวงจรที่อาศัยการทดสอบบิตของข้อมูลที่เป็นสัญญาณดิจิตอล โดยใช้วิธีการควบคุมลอจิก 1 หรือ 0 จากคอนโทรลรีจิสเตอร์และส่งไปยังวงจรแปลงสัญญาณ DAC (Digital to Analog Converter) แปลงให้เป็นสัญญาณอนาล็อกแล้วทำการเปรียบเทียบกับสัญญาณอนาล็อกที่เป็นอินพุท เมื่อผลลัพธ์ที่ได้มีค่าเป็น 1 แสดงว่าสัญญาณอินพุทที่เข้ามามีค่าสูงกว่าสัญญาณเปรียบเทียบกำหนดให้บิต ทดสอบเป็น 1 แล้วจึงตรวจสอบบิตต่อไป และมีดิสทริบิวชันรีจิสเตอร์ทำหน้าที่จดจำสภาวะการตรวจแต่ละบิต โดยเรียงจากบิตที่มีน้ำหนักมากสุด (MSB) ไปยังบิตที่มีน้ำหนักต่ำสุด (LSB) ตามลำดับ



รูปที่ 2.23 Basic Successive Approximation ADC

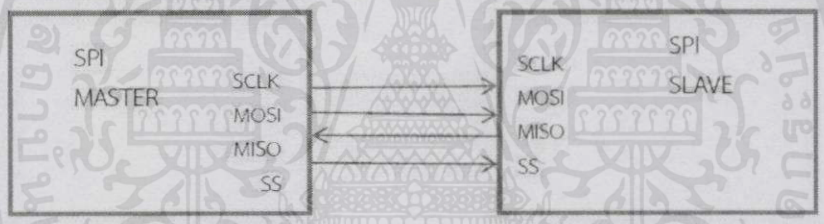
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.17 การสื่อสารแบบ SPI

การติดต่อสื่อสารกันระหว่าง Raspberry Pi กับ ชิป MCP3002 สามารถทำได้โดยการเลือกใช้วิธีการติดต่อแบบ SPI เพราะการติดต่อสื่อสารเป็นแบบ Synchronous Serial เหมาะสำหรับใช้ในระยะสั้นๆ

2.17.1 ภาพรวมของ SPI

SPI (Serial Peripheral Interface) คือ การสื่อสารอนุกรมแบบซิงโครนัส เป็นรูปแบบที่ใช้วิธีส่งข้อมูลโดยใช้สัญญาณ Clock มาเป็นตัวกำหนดจังหวะในการรับส่ง ข้อมูล SPI ทำงานในรูปแบบที่ให้อุปกรณ์ตัวหนึ่งทำหน้าที่เป็นมาสเตอร์ ในขณะที่อีกตัวหนึ่งทำหน้าที่เป็นสเลฟ และส่งข้อมูลในโหมด Full-duplex ซึ่งหมายความว่า สัญญาณสามารถส่งหากันระหว่างมาสเตอร์และสเลฟได้อย่างต่อเนื่อง ในการสื่อสารแบบ SPI นี้ไม่มีมาตรฐานกำหนดตายตัวว่าข้อมูลที่ส่งหากันต้องอยู่ในรูปแบบหรือฟอร์แมตแบบไหนเป็นการคิดโปรโตคอลการสื่อสารกันเอง ซึ่งรูปแบบการสื่อสารแสดงได้ดังรูปที่ 2.24



รูปที่ 2.24 SPI Bus : SPI master and SPI Slave

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.17.2 พื้นฐานการทำงานของการสื่อสารแบบ SPI

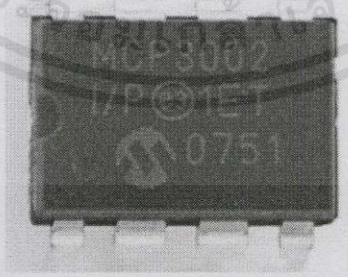
SPI ต้องการสายสัญญาณ 4 เส้นสัญญาณ บางครั้งเราเรียกว่าบัสอนุกรม Four Wire เส้นสัญญาณทั้ง 4 เส้นนี้อธิบายการทำงานได้ตามตารางที่ 2.1

ตารางที่ 2.1 ตารางบอกชนิดและการทำงานของการสื่อสารแบบ SPI

เส้น	ชื่อ	การทำงาน
SCLK	Serial Clock	สัญญาณนาฬิกาที่มาสเตอร์เป็นผู้ส่ง
MOSI	มาสเตอร์เอาท์พุท	เอาท์พุทจากมาสเตอร์ที่ส่งให้สเลฟ
MISO	มาสเตอร์อินพุท	เอาท์พุทจากสเลฟที่ส่งให้มาสเตอร์ (ชิป MCP3002)
SS	Slave Select	เป็นสัญญาณที่มาสเตอร์ใช้เป็นตัวเลือกว่าจะติดต่อกับสเลฟตัวใด

ตัวที่เป็นมาสเตอร์ ทำหน้าที่เป็นตัวควบคุมการสื่อสารทั้งหมด โดยควบคุมการสื่อสารตามสัญญาณนาฬิกา ตัวมาสเตอร์จะเป็นตัวที่ตัดสินใจรับหรือส่งข้อมูล ภายในการสื่อสารส่วนตัวที่เป็นสเลฟจะทำหน้าที่รับคำสั่งต่างๆจากมาสเตอร์ สายสัญญาณเส้น SS หรือ Slave select ใช้ในกรณีที่เรามีตัวสเลฟมากกว่า 1 ตัว โดยการทำให้เส้น SS มีระดับสัญญาณเป็น Low เมื่อต้องการติดต่อกับ สเลฟ ตัวใด

2.18 ชิป MCP3002



รูปที่ 2.25 ชิป MCP3002

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MCP3002 เป็นชิปที่ใช้ในเทคโนโลยีการแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล เป็นการใช้หลักการแปลงแบบ Successive Approximation หรือ SAR มีความละเอียด 10 bit มาพร้อมกับวงจร Sample และ Hold ภายในตัวชิป ตัวชิปมีคุณสมบัติดังนี้

- 10-bit resolution
- ±1 LSB maximum DNL
- ±1 LSB maximum INL
- Analog inputs programmable as single-ended or pseudo-differential pairs
- On-chip sample and hold
- SPI serial interface (modes 0,0 and 1,1)
- Single supply operation: 2.7V - 5.5V
- 200 ksps max sampling rate at VDD = 5V
- 75 ksps max sampling rate at VDD = 2.7V
- Low power CMOS technology
- 5 nA typical standby current, 2 μA maximum
- 550 μA maximum active current at 5V
- Industrial temp range: -40°C to +85°C
- 8-pin MSOP, PDIP, SOIC and TSSOP packages

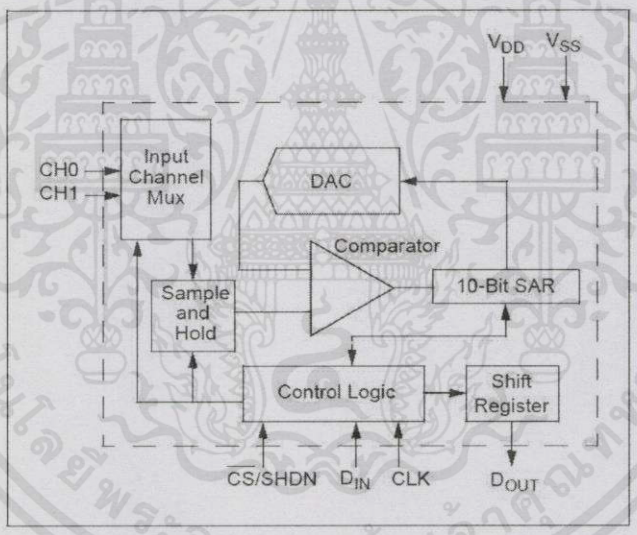
ชิป MCP3002 มีขาทั้งหมด 8 ขา ได้แก่ Chip Select (CS) CH0 CH1 VSS Vdd/Vref CLK Dout และ Din

รูปที่ 2.26 ขาของชิป MCP3002

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 ตารางการต่อใช้งานขาของชิป MCP3002 กับขาของ Raspberry Pi

MCP3002 pin connects to Raspberry Pi Pin		
MCP3002		Raspberry pi
CS	Pin1	CS0 หรือ CS1
CH0	Pin2	สัญญาณอนาล็อกขาเข้า
CH1	Pin3	สัญญาณอนาล็อกขาเข้า
Vss	Pin4	GND
Din	Pin5	MOSI
Dout	Pin6	MISO
CLK	Pin7	SCLK
Vdd/Vref	Pin8	3V3



รูปที่ 2.27 Block Diagram ภายในตัวชิป MCP3002

ไอซีชนิดนี้จะถูกนำไปใช้ร่วมกับอุปกรณ์ตรวจจับควัน เพราะควันเป็นสัญญาณอนาล็อกการที่จะนำสัญญาณอนาล็อกมาประมวลผลในสัญญาณดิจิตอลนั้นจำเป็นต้องมีวงจรที่ใช้สำหรับแปลงสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.19 เทคโนโลยีสตรีมมิ่ง

เทคโนโลยีสตรีมมิ่ง (Streaming Technology) คือ เทคโนโลยีที่ช่วยให้การถ่ายโอนข้อมูลมัลติมีเดียผ่านอินเทอร์เน็ต และมีการแสดงผลได้ในระยะเวลาอันสั้น โดยไม่ต้องรอให้ดาวน์โหลดข้อมูลไฟล์นั้นๆลงเครื่องคอมพิวเตอร์ให้ทั้งไฟล์ก่อน เนื่องจากการดาวน์โหลดไฟล์มัลติมีเดียทั้งไฟล์ จะใช้เวลาค่อนข้างมาก ดังนั้นการใช้เทคโนโลยีสตรีมมิ่งจึงช่วยให้ผู้ใช้งานสามารถรับข้อมูลไปพร้อมๆกับการรับชมข้อมูลได้ในเวลาเดียวกัน แสดงได้ดังนี้

รูปที่ 2.28 การทำงานของเทคโนโลยีสตรีมมิ่ง

ผู้ใช้บริการ (Client) เข้าถึงเว็บไซต์ที่มีสื่อที่ต้องการรับชม โดยมีการร้องขอไปยังเว็บเซิร์ฟเวอร์ (Web Server) ถัดมาเว็บเซิร์ฟเวอร์จะทำการร้องขอไปยังสตรีมมิ่งเซิร์ฟเวอร์ (Streaming Server) เพื่อหาไฟล์สื่อที่มีความเฉพาะเจาะจงกับคำร้องขอ จากนั้นสตรีมมิ่งเซิร์ฟเวอร์จะส่งไฟล์สื่อนั้นกับมาผ่านทางเว็บเซิร์ฟเวอร์ ฝั่งผู้ใช้บริการนั้นจะต้องมีซอฟต์แวร์เพื่อถอดรหัสไฟล์สื่อที่ได้รับมาเพื่อรับชมไฟล์

2.19.1 ลักษณะการส่งของเทคโนโลยีสตรีมมิ่ง

2.19.1.1 File on Demand

เป็นการบันทึกวีดีโอคลิปไว้บน ให้ผู้ใช้สามารถเลือกดูคลิปวีดีโอผ่านระบบอินเทอร์เน็ต ไฟล์เหล่านี้จะถูกเข้ารหัสในรูปแบบที่เหมาะสมต่อการแสดงผลแบบสตรีมมิ่งแล้วนำไฟล์จัดเก็บไว้ในเซิร์ฟเวอร์

2.19.1.2 File Live Broadcasting

เป็นการถ่ายทอดเหตุการณ์ที่เกิดขึ้น ณ เวลานั้น โดยผู้ชมได้รับชมเหตุการณ์ต่างๆ ในปัจจุบันและทันท่วงที ด้วยวิธีการแปลงสัญญาณจากกล้องวีดีโอเป็นข้อมูลดิจิตอลและเครื่องเซิร์ฟเวอร์จะทำการถ่ายทอดไปยังเครื่องผู้ชมปลายทาง นอกจากนี้ยังสามารถเรียกใช้งานพร้อมกันเป็นจำนวนมากได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.19.2 ประเภทการให้บริการผ่านอินเทอร์เน็ต

เมื่อนำวีดีโอที่แปลงให้เหมาะกับการให้บริการส่งขึ้นไปเก็บไว้บนเซิร์ฟเวอร์ของระบบอินเทอร์เน็ตแล้ว ผู้ใช้สามารถดูวีดีโอได้ด้วยการคลิกที่ไฟล์วีดีโอที่อยู่ในหน้าเว็บโดยตรง โดยเว็บเซิร์ฟเวอร์จะให้บริการวีดีโอสองแบบ

2.19.2.1 เว็บเซิร์ฟเวอร์แบบไม่มีสตรีมมิ่งเซิร์ฟเวอร์

แบบดาวน์โหลดในกรณีที่เว็บเซิร์ฟเวอร์ไม่มี สตรีมมิ่งเซิร์ฟเวอร์ให้บริการ เมื่อผู้ใช้คลิกหน้าเว็บเพื่อดูวีดีโอ เซิร์ฟเวอร์จะส่งไฟล์วีดีโอทั้งไฟล์ไปให้กับผู้ใช้ ผู้ใช้ต้องรอจนการดาวน์โหลดข้อมูลสมบูรณ์จึงจะดูได้ วิธีนี้จะมีการเก็บแฟ้มวีดีโอไว้ในเครื่องผู้ใช้ และไม่สามารถดูการถ่ายทอดสดได้

2.19.2.2 เว็บเซิร์ฟเวอร์มีสตรีมมิ่งเซิร์ฟเวอร์

การให้บริการแบบนี้ เมื่อผู้ใช้หน้าเว็บแสดงความจำนงที่จะดูวีดีโอ สตรีมมิ่งเซิร์ฟเวอร์ก็จะส่งข้อมูลที่แบ่งออกเป็นชิ้นเล็กๆ พอเหมาะกับความเร็วของอินเทอร์เน็ตของผู้ใช้ไปให้กับผู้ใช้ และเมื่อส่งไปจนเต็ม Buffer ของผู้ใช้แล้วก็จะแสดงผลวีดีโอในหน้าจอผู้ใช้ ปกติแล้ว Buffer เป็นการใช้หน่วยความจำขนาดเล็กในการเก็บชิ้นส่วนข้อมูลวีดีโอเพื่อการแสดงผลโดยในขณะที่ผู้ใช้ดูวิดีโออยู่ เซิร์ฟเวอร์ก็จะทยอยส่งข้อมูลให้ไปเรื่อยๆจนจบ เนื่องจาก Buffer มีขนาดเล็ก ดังนั้นวีดีโอที่เซิร์ฟเวอร์ส่งไปแทบจะแสดงผลทันทีที่ผู้ใช้ขอดู และวิธีนี้จะไม่มีการบันทึกข้อมูลในเครื่องผู้ใช้จึงมีความปลอดภัยมากกว่า

2.19.3 โปรโตคอลสำหรับเทคโนโลยีสตรีมมิ่ง

2.19.3.1 User Datagram Protocol (UDP)

UDP protocol จะเป็นการส่งข้อมูลทิศทางเดียว คือ ข้อมูลจะส่งจากเซิร์ฟเวอร์ไปยังผู้ใช้บริการโดยไม่มีการตรวจสอบความถูกต้องของข้อมูล ดังนั้นจึงสามารถส่งข้อมูลได้เร็วกว่า TCP Protocol จึงได้ถูกนำมาใช้ในการส่งข้อมูลมัลติมีเดีย

2.19.3.2 Real Time Streaming Protocol (RTSP)

คือโปรโตคอลที่ใช้ในการควบคุมสตรีมมิ่งข้อมูล RTSP มีสองโครงสร้าง คือ หน่วยควบคุม และการเชื่อมต่อข้อมูล RTSP เป็นโปรโตคอลที่ใช้รูปแบบ Client/Server ที่ถูกออกแบบเพื่อใช้ในการแสดงสื่อมัลติมีเดีย

2.19.4 ข้อดีและข้อเสียของเทคโนโลยีสตรีมมิ่ง

ข้อดีคือ ผู้ชมจะสามารถรับชมได้ทันทีและสามารถเลื่อนไปยังตำแหน่งที่ต้องการชมได้ โดยไม่ต้องรอให้ดาวน์โหลดจนเสร็จก่อน นอกจากนี้รายการที่เลือกชม ไม่ได้ถูกเก็บไว้ในเครื่อง ทำให้ประหยัด พื้นที่ใน Hard Disk เหมาะสำหรับผู้ที่มีพื้นที่ใน Hard Disk จำกัด ในส่วนของทางด้านข้อเสียนั้น สำหรับผู้ที่มีช่องสัญญาณ (Bandwidth) ที่มีความเร็วไม่คงที่จะพบปัญหาการ Buffering บ่อยครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

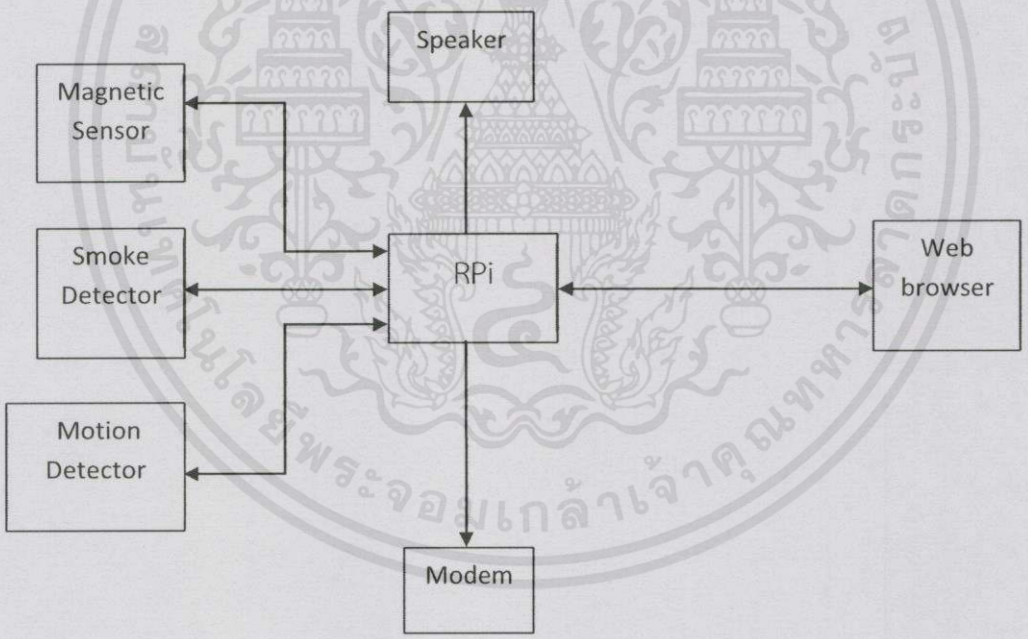
บทที่ 3

การออกแบบและจัดทำโครงงาน

3.1 การออกแบบ

ชุดติดตั้งระบบรักษาความปลอดภัยโดย Raspberry Pi ถูกออกแบบให้ติดตั้งได้ภายในที่อยู่อาศัยของมนุษย์ มีจุดประสงค์เพื่อให้ระบบรักษาความปลอดภัยที่ติดตั้งนี้สามารถแจ้งเตือนไปยังผู้ใช้งานได้ และให้ผู้ใช้งานสามารถควบคุมเปิดปิดตัวอุปกรณ์ในชุดติดตั้งได้ผ่านหน้าเว็บ โดยภายในระบบจะมีเซ็นเซอร์หลักๆอยู่สามส่วน คือ เซ็นเซอร์แม่เหล็ก (Magnetic Sensor) โดยมีลักษณะเป็นหน้าสัมผัสเหมาะสำหรับติดตั้งไว้ที่บานประตูหรือหน้าต่าง

เซ็นเซอร์ตรวจจับการเคลื่อนไหว (Motion Detector) โดยใช้กล้องและการเขียนโปรแกรมตรวจจับการเคลื่อนไหว เซ็นเซอร์ตรวจจับควัน (Smoke Detector) ซึ่งใช้ตรวจจับควันและก๊าซที่ติดไฟได้ในอากาศ



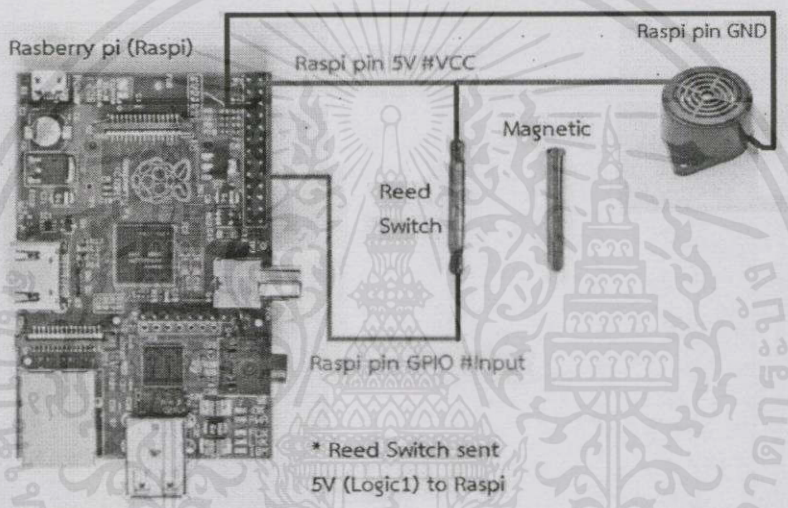
รูปที่ 3.1 บล็อกไดอะแกรมของชุดติดตั้งระบบรักษาความปลอดภัยโดย Raspberry Pi

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1 การออกแบบติดตั้งเซ็นเซอร์แม่เหล็ก

3.1.1.1 ออกแบบวงจรอุปกรณ์เซ็นเซอร์แม่เหล็ก

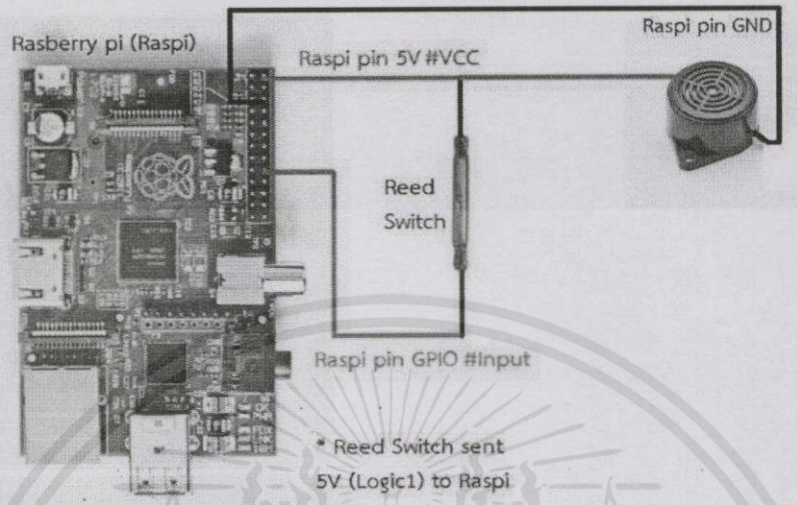
เซ็นเซอร์แม่เหล็ก (Magnetic Sensor) คือ การนำเซ็นเซอร์แม่เหล็กมาทำงานร่วมกับแท่งแม่เหล็ก โดยจะใช้ Raspberry Pi เพื่อใช้เป็นส่วนควบคุมการทำงาน โดยในการออกแบบจะให้เซ็นเซอร์แม่เหล็กมีลักษณะเป็นหน้าสัมผัสระหว่างเซ็นเซอร์แม่เหล็กกับแท่งแม่เหล็ก



รูปที่ 3.2 เซ็นเซอร์แม่เหล็ก (ขณะมีแท่งแม่เหล็กในบริเวณหน้าสัมผัส)

ในรูปที่ 3.2 จะแสดงถึงการทำงานของอุปกรณ์เซ็นเซอร์แม่เหล็ก เมื่อนำแท่งแม่เหล็กเข้ามาใกล้ๆ ในบริเวณหน้าสัมผัสของเซ็นเซอร์ประมาณ 1 เซนติเมตร วงจรเซ็นเซอร์แม่เหล็กจะทำงานเป็นลักษณะ Close Circuit จะส่งผลให้ เซ็นเซอร์แม่เหล็กส่ง Logic High ไปให้อุปกรณ์ควบคุม Raspberry Pi

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



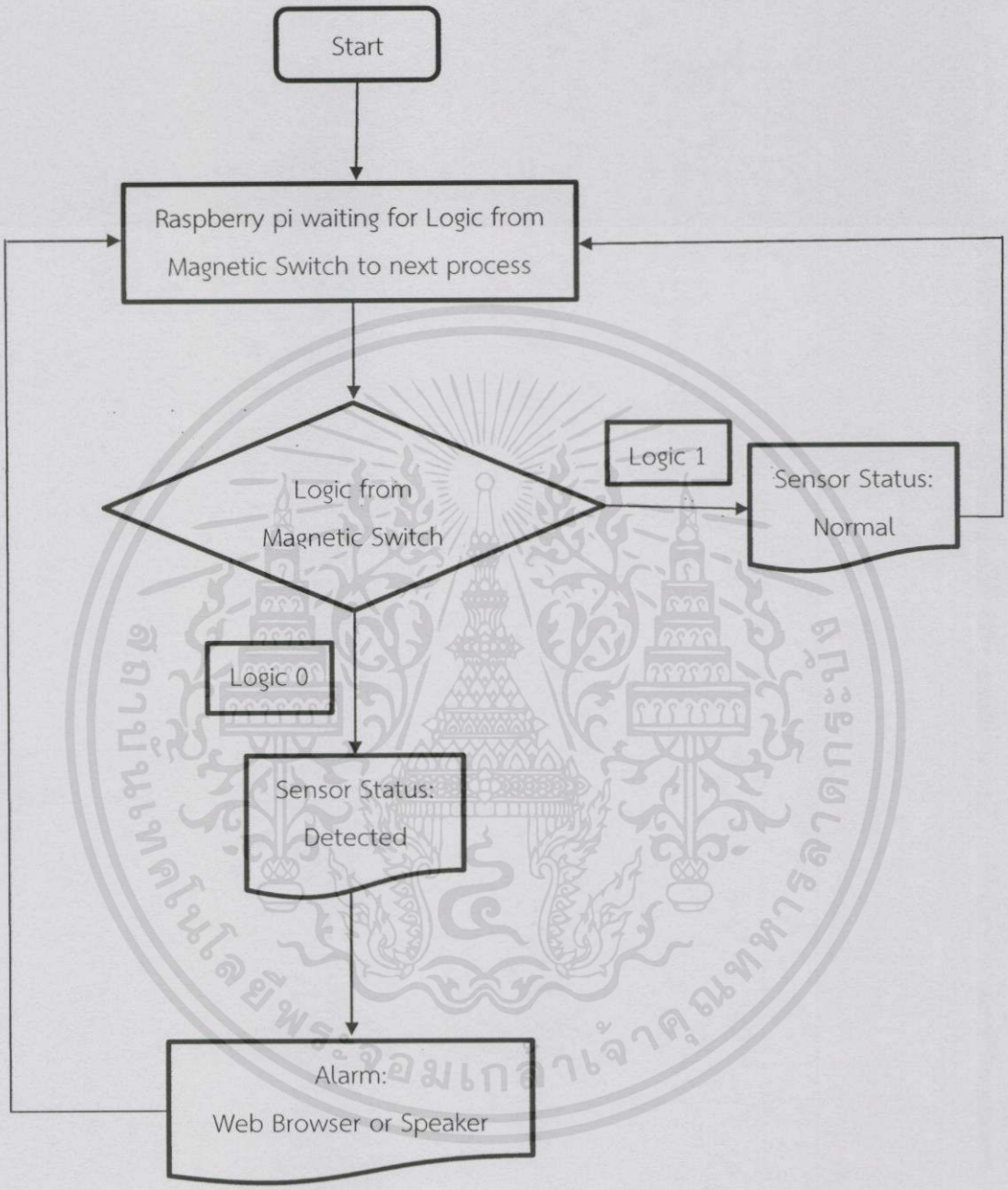
รูปที่ 3.3 อุปกรณ์เซ็นเซอร์แม่เหล็ก (ขณะไม่มีแท่งแม่เหล็กในบริเวณหน้าสัมผัส)

ในรูปที่ 3.3 จะแสดงถึงการทำงานของอุปกรณ์เซ็นเซอร์แม่เหล็ก เมื่อไม่มีแท่งแม่เหล็กในบริเวณหน้าสัมผัสของรีดสวิตช์ วงจรเซ็นเซอร์แม่เหล็กจะทำงานเป็นลักษณะ Open Circuit จะส่งผลให้ เซ็นเซอร์แม่เหล็กส่ง Logic Low ไปให้อุปกรณ์ควบคุม Raspberry Pi

3.1.1.2 ออกแบบโปรแกรมเซ็นเซอร์แม่เหล็ก

โปรแกรมของเซ็นเซอร์แม่เหล็กมีเงื่อนไขคือ เมื่อมีแท่งแม่เหล็กอยู่ในบริเวณหน้าสัมผัสของรีดสวิตช์ Raspberry Pi ได้รับ Logic High อยู่ตลอด เมื่อหน้าสัมผัสเคลื่อนออกทำให้ Raspberry Pi ได้รับ Logic Low ซึ่งเราสามารถใช้เงื่อนไขของ Logic Low นี้ให้เป็นจุดเริ่มต้นของโปรแกรมการแจ้งเตือนในรูปแบบต่างๆ ไม่ว่าจะเป็นการแจ้งเตือนผ่านหน้าเว็บหรือส่งเสียงแจ้งเตือนผ่านลำโพง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

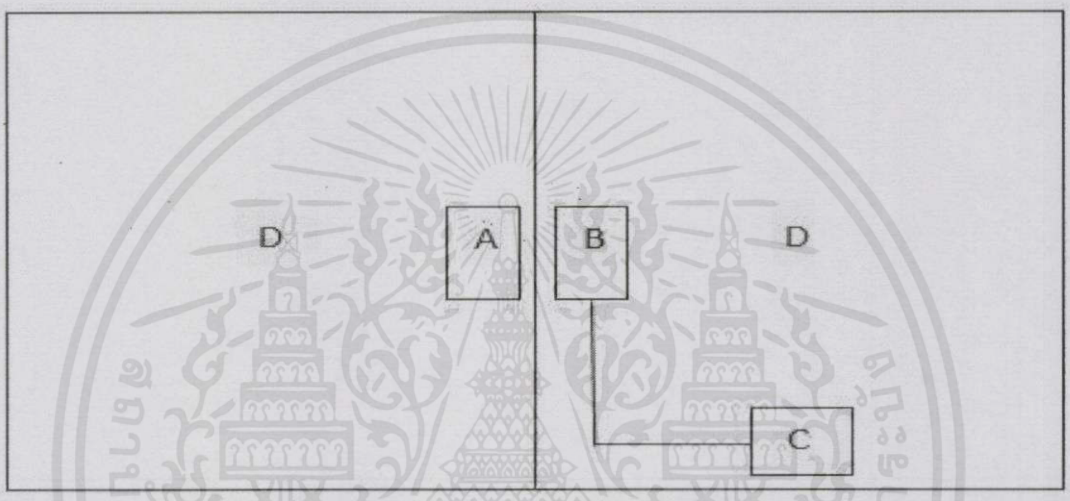


รูปที่ 3.4 โฟลว์ชาร์ตการทำงานโปรแกรมเซ็นเซอร์แม่เหล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1.3 ออกแบบที่ติดตั้งอุปกรณ์เซ็นเซอร์แม่เหล็ก

ในการใช้งานจริงเราจะนำรีดสวิชต์ไปติดกับบานหน้าต่างหรือประตูในลักษณะที่ทำให้เกิดบริเวณหน้าสัมผัส เพื่อเป็นสวิตช์ให้โปรแกรมทำงาน การที่เราติดอุปกรณ์ไว้เราจำเป็นต้องมีอุปกรณ์สามอย่างคือ รีดสวิตช์ แท่งแม่เหล็ก และ Raspberry Pi เพื่อใช้ในการทำงาน



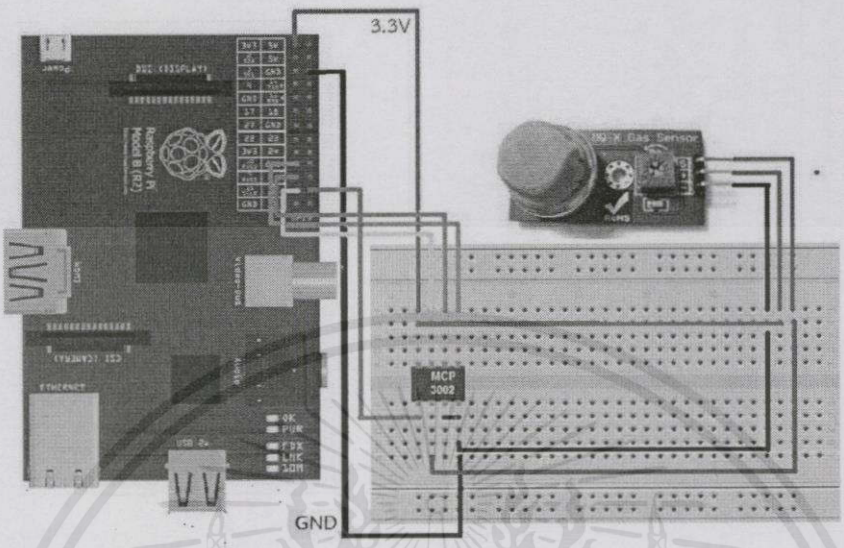
รูปที่ 3.5 การติดตั้งรีดสวิตช์กำหนดให้ A : แม่เหล็ก, B : รีดสวิตช์, C : Raspberry Pi, D : หน้าต่างหรือประตู

3.1.2 การออกแบบเซ็นเซอร์ตรวจจับควัน

3.1.2.1 การออกแบบอุปกรณ์เซ็นเซอร์ตรวจจับควัน

ตัวอุปกรณ์เซ็นเซอร์ตรวจจับควันที่ใช้คือ MQ–2 (Gas and Smoke Detector) เป็นตัวตรวจจับอนุภาคควัน MCP3002 เป็นตัวแปลงสัญญาณที่ได้จาก MQ-2 ซึ่งเป็นสัญญาณอนาล็อกให้เป็นสัญญาณดิจิตอลไว้นำไปประมวลผล ส่วน Raspberry Pi เป็นตัวประมวลผลปริมาณควันที่ตรวจพบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



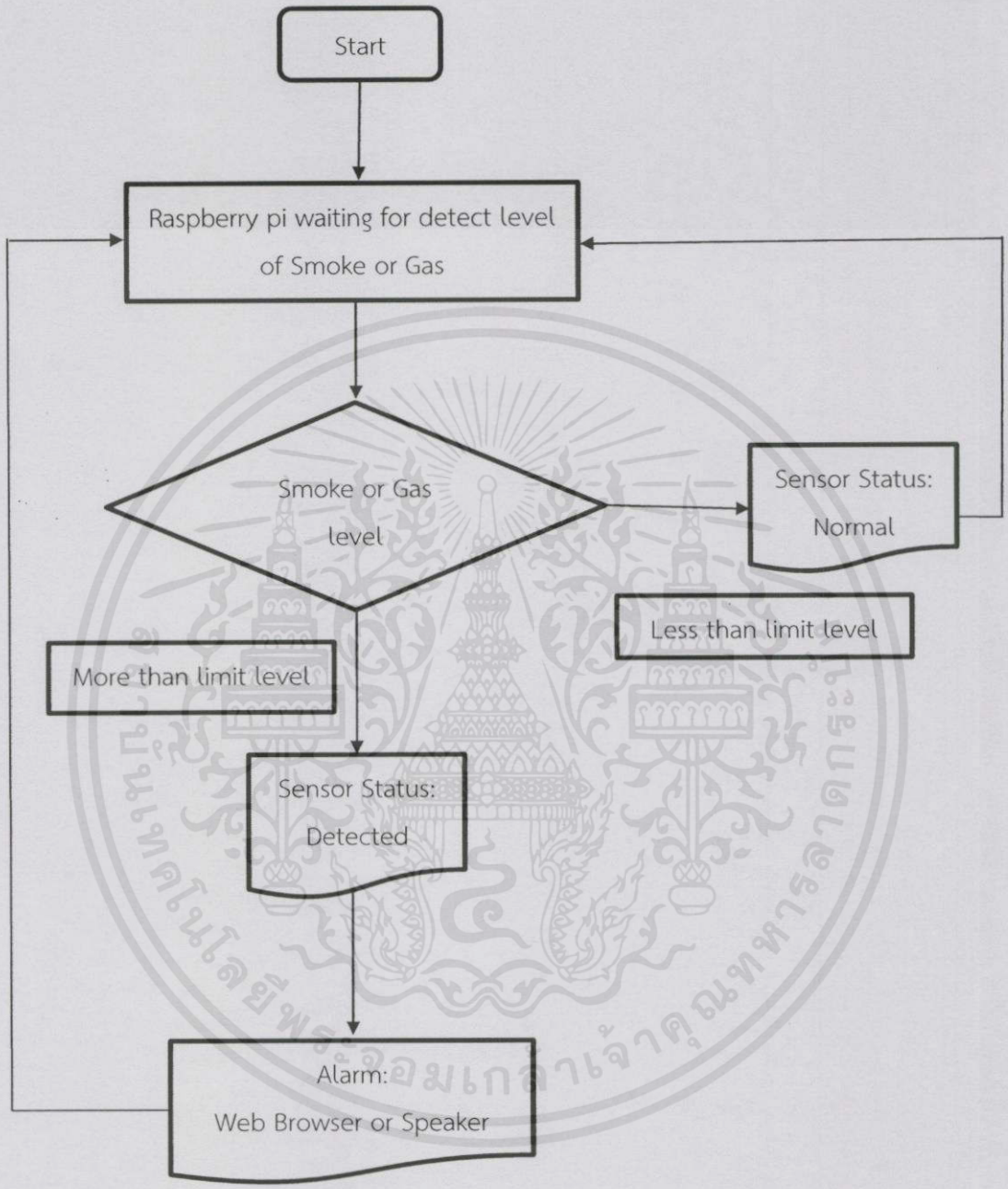
รูปที่ 3.6 อุปกรณ์เซ็นเซอร์ตรวจจับควัน

ในรูปที่ 3.6 จะแสดงถึงการเชื่อมต่อกันของอุปกรณ์เซ็นเซอร์ตรวจจับควัน เมื่อมีการตรวจพบควันหรือก๊าซไวไฟ เซ็นเซอร์ตรวจจับควัน จะส่งค่าที่ได้ให้ไปประมวลผลใน Raspberry Pi หากมีปริมาณควันหรือก๊าซไวไฟเกิดขึ้นมากเกินระดับที่กำหนดไว้ในส่วนของโปรแกรม Raspberry Pi จะทำการแจ้งเตือนในรูปแบบต่างๆ ไม่ว่าจะเป็นการแจ้งเตือนผ่านหน้าเว็บหรือส่งเสียงแจ้งเตือนผ่านลำโพง

3.1.2.2 ออกแบบโปรแกรมเซ็นเซอร์ตรวจจับควัน

โปรแกรมเซ็นเซอร์ควันมีเงื่อนไขอยู่กับการตั้งค่าระดับของควันหรือก๊าซที่จะสามารถตรวจพบโดยโมดูล MQ-2 ซึ่งในโปรแกรมระดับของควันหรือก๊าซอ้างอิงอยู่ที่ระดับตัวเลขค่าหนึ่ง ซึ่งหากระดับของควันอยู่ที่มากกว่าค่าระดับตัวเลขที่ตั้งไว้ เซ็นเซอร์จะทำงานเพื่อใช้เป็นเงื่อนไขในการส่ง SMS และการแจ้งเตือนด้วยเสียง ในส่วนของโฟลว์ชาร์ตการทำงานของโปรแกรมแสดงตามรูปที่ 3.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



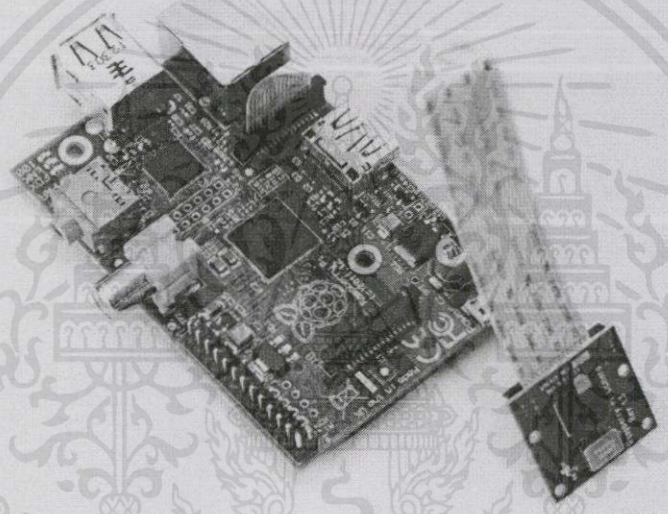
รูปที่ 3.7 โฟลว์ชาร์ตการทำงานของโปรแกรมเซ็นเซอร์ตรวจจับควัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 การออกแบบเซ็นเซอร์ตรวจจับการเคลื่อนไหว

สำหรับส่วนของเซ็นเซอร์ตรวจจับการเคลื่อนไหวนั้น จะทำการเขียนโปรแกรมให้กล้องเป็นเซ็นเซอร์ไปในตัวโดยอาศัยไลบรารี่เสริมในการช่วยเขียนโปรแกรม นั้นก็คือไลบรารี่ของ OpenCV จะออกแบบให้กล้องนั้นทำการ Capture ภาพไปเรื่อยๆและทำการเปรียบเทียบความต่างของรูปก่อนและหลังว่ามีความต่างเกินระดับที่กำหนดหรือไม่ หากมีค่าเกินระดับที่กำหนดไว้จะทำการใช้เป็นเงื่อนไขในการถ่ายวีดีโอต่อไป

3.1.3.1 การติดตั้งอุปกรณ์เซ็นเซอร์ตรวจจับการเคลื่อนไหว



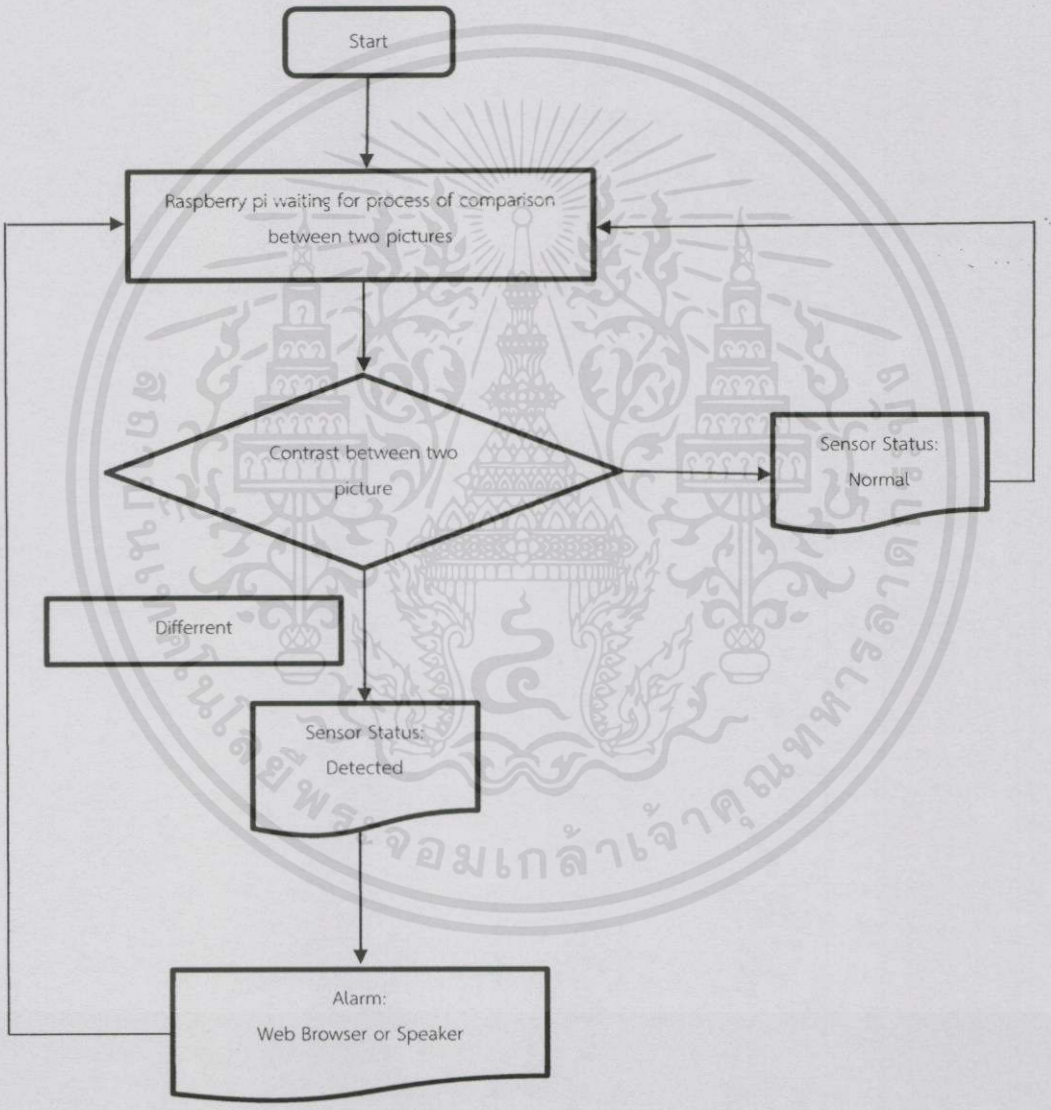
รูปที่ 3.8 การติดตั้งเซ็นเซอร์ตรวจจับการเคลื่อนไหว

อาศัยไลบรารี่ของ OpenCV กล้องจะทำการแปลงรูปที่ได้จากการ Capture ให้อยู่ในลักษณะภาพขาวดำ โดยอ้างอิงระดับของสีขาวและสีดำในลักษณะตัวเลขจาก Grayscale โดยมีระดับตั้งแต่ระดับ 0 – 255 และทำการเปรียบเทียบภาพว่ามีค่าต่างจากระดับอ้างอิง Motion Level ที่ได้กำหนดไว้หรือไม่หากมีความต่างมากกว่าระดับที่ได้อ้างอิงไว้ก็จะเป็นเงื่อนไขให้ทำการอัดวีดีโอต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3.2 ออกแบบโปรแกรมเซ็นเซอร์ตรวจจับการเคลื่อนไหว

อย่างที่ได้กล่าวไว้ข้างต้นโปรแกรมมีการใช้ไลบรารี่จาก OpenCV ในการช่วยเขียนโปรแกรมซึ่งมีโฟลว์ชาร์ตการทำงานเป็นไปตามรูปที่ 3.9 จะเห็นได้ว่าเมื่อรูปมีความต่างจะแสดงค่าขึ้นมาและเมื่อค่านั้นเกินระดับที่อ้างอิงเอาไว้จะทำการอัดวีดีโอ



รูปที่ 3.9 โฟลว์ชาร์ตการทำงานโปรแกรมเซ็นเซอร์ตรวจจับการเคลื่อนไหว

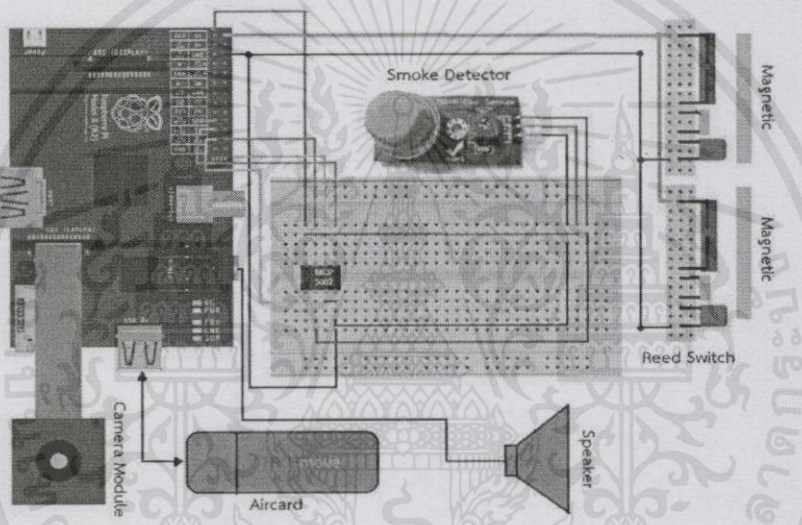
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4 การรวมระบบเซ็นเซอร์

จะทำการรวมระบบเซ็นเซอร์ให้อยู่ในโปรแกรมเดียวกัน มีรายละเอียดในการเชื่อมต่อและการโปรแกรมดังนี้

3.1.4.1 การเชื่อมต่ออุปกรณ์เซ็นเซอร์

การเชื่อมต่ออุปกรณ์แสดงได้ตามรูปที่ 3.10 ซึ่งมีเซ็นเซอร์แม่เหล็กสองชุด เซ็นเซอร์ตรวจจับควัน เซ็นเซอร์ตรวจจับการเคลื่อนไหวซึ่งเป็นตัวกล้อง มี Air Card และก็มีลำโพงสำหรับแจ้งเตือนผ่าน SMS และเสียง

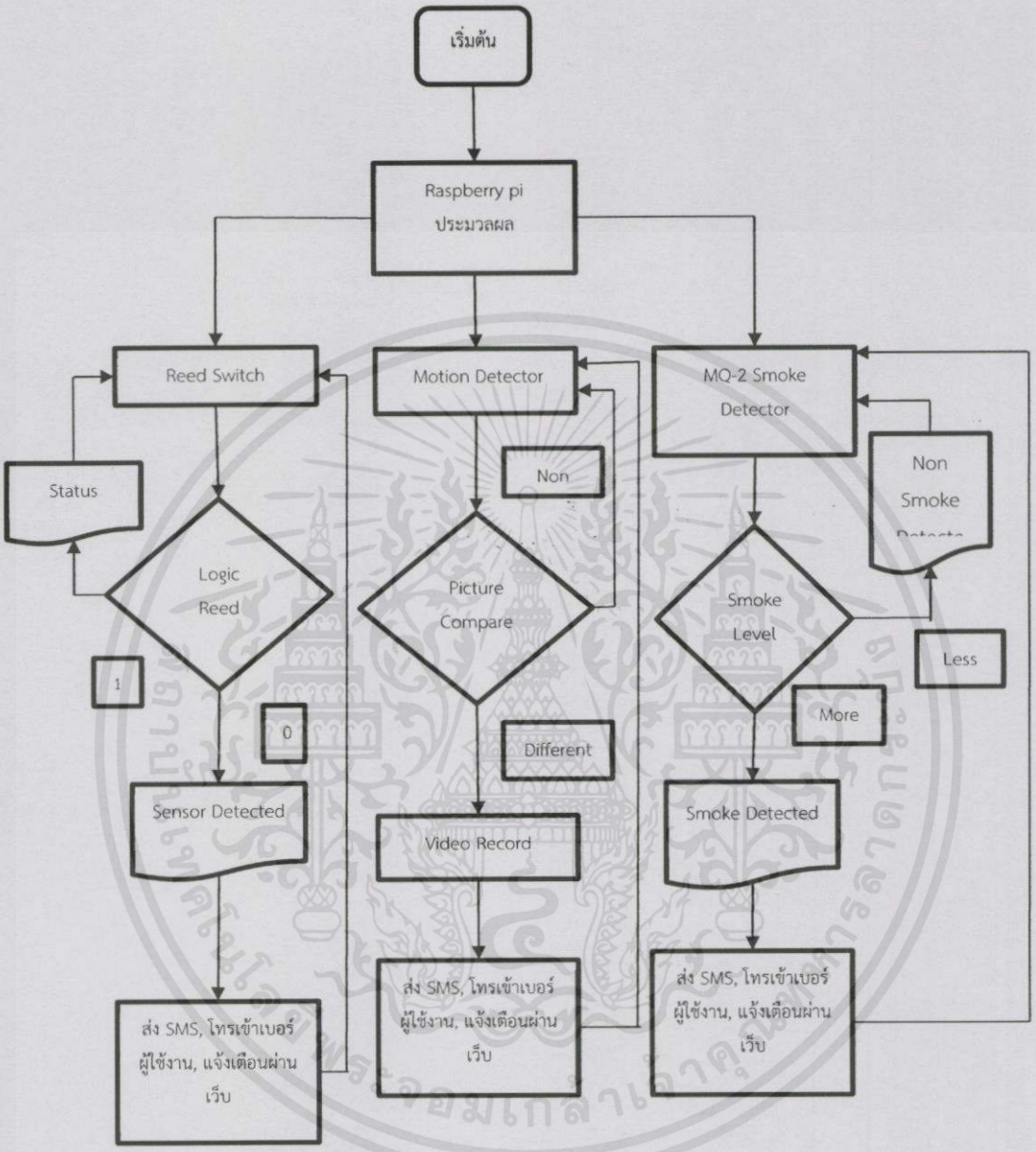


รูปที่ 3.10 เชื่อมต่ออุปกรณ์รวม

3.1.4.2 เขียนโปรแกรมรวมเซ็นเซอร์

โปรแกรมรวมจะมีการทำงานดังนี้ แสดงได้จากรูปที่ 3.11 ซึ่งอธิบายได้จาก เมื่อเซ็นเซอร์แต่ละตัวทำงานตามเงื่อนไข จะทำการส่ง SMS และโทรเข้าเบอร์ผู้ใช้งาน ในส่วนของเซ็นเซอร์ตรวจจับควันจะมีการแจ้งเตือนด้วยเสียงผ่านลำโพงด้วยในขณะเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 โฟลว์ชาร์ตโปรแกรมรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.5 การออกแบบการแจ้งเตือนผู้ใช้งานและการควบคุมอุปกรณ์ผ่านหน้าเว็บ

3.1.5.1 การแจ้งเตือนผ่าน SMS และแจ้งเตือนโดยการโทรเข้าเบอร์ผู้ใช้งาน

การแจ้งเตือนด้วย SMS นั้นจะใช้ตัวโมเดมที่ Raspberry Pi นั้นรองรับซึ่งตัวที่ใช้นั้นคือ Air Card ของ True Move โดยตัวโมเดมข้างในตัวของ Air Card นั้นใช้ของ Huawei รุ่น E303F ซึ่ง Raspberry Pi รองรับการทำงาน ส่วนเรื่องไลบรารี่นั้นจะใช้ของ Gammu ในการเขียนโปรแกรมให้ส่ง SMSโดยมีการเซ็ตตั้งค่า Air Card ตามขั้นตอนดังต่อไปนี้

- ติดตั้ง Gammu ซึ่งเป็นโปรแกรมที่ใช้ในการจัดการระบบโทรศัพท์ ด้วยคำสั่ง "sudo apt-get install gammu usb-modeswitch python-gammu" ซึ่งในขั้นต่อนี้ต้องทำการเชื่อมต่ออินเทอร์เน็ตไว้ด้วย

- ทำการเสียบ Air Card ที่มีซิมการ์ดที่ใช้งานได้ เช็คว่า Raspberry Pi อ่าน Air Card ได้ ด้วยคำสั่ง lsusb จะแสดงการเชื่อมต่อที่พอร์ต USB ของ Raspberry pi ถ้าใช้งานได้จะขึ้นคำว่า "Bus 001 Device 010: ID 12d1:1506 Huawei Technologies Co., Ltd. E398 LTE/UMTS/GSM Modem/Networkcard" แสดงว่า Raspberry Pi อ่าน Air Card ได้แล้ว

- ตรวจสอบชื่ออุปกรณ์ Air Card โดยคำสั่ง "dmesg | grep ttyUSB" ดูว่าอุปกรณ์ชื่ออะไรเพื่อนำไปใช้ในการปรับแต่ง เช่น ttyUSB0, ttyUSB1 เป็นต้น

- ตั้งค่าโปรแกรม Gammu โปรแกรม Gammu จะทำหน้าที่รับส่ง SMS ซึ่งจะต้องตั้งค่าให้โปรแกรมรู้จัก Air Card ที่ใช้ก่อน โดยพิมพ์คำสั่งต่อไปนี้ "sudo gammu-config" จะแสดงหน้าต่างการตั้งค่าขึ้นมา ให้เราตั้งค่าดังนี้

Port: /dev/ttyUSB0 << แล้วแต่ว่าเราใช้ port ไหน
CONNECTION: at19200
Model:
Synchronize time: yes
Log file:
Log format: nothing
Use locking:
Gammu localisation:

- เสร็จแล้วก็ทำการบันทึกแล้วออกจากโปรแกรมเมื่อเราทำการ Reboot เครื่องใหม่อุปกรณ์ก็พร้อมที่จะใช้งาน พร้อมสำหรับการแจ้งเตือนทั้งผ่าน SMS และการโทรออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.5.2 การแจ้งเตือนผ่านเสียง

ในการแจ้งเตือนเสียงนั้นจะเกิดขึ้นเมื่อเซ็นเซอร์ควันทำงานจากการที่ตรวจจับระดับของควันได้เกินกว่าระดับมาตรฐานที่ได้กำหนดไว้ในตัวโปรแกรม โดยจะทำการเล่นไฟล์เสียงแจ้งเตือนผ่านลำโพง และในส่วนของเซนเซอร์แม่เหล็กจะแจ้งเตือนผ่านเสียง Siren

3.1.5.3 การแสดงผลและควบคุมอุปกรณ์ผ่านหน้าเว็บ

เบื้องต้นในการที่เราจะสามารถแสดงผลและควบคุมอุปกรณ์ผ่านหน้าเว็บได้นั้นต้องทำการจัดการด้านระบบฐานข้อมูล ซึ่งทางผู้จัดทำได้เลือกใช้งาน MySQL ซึ่งใช้งานควบคู่กับภาษา PHP และ HTML ในการเขียนหน้าเว็บ โดยใช้ Apache ในการจัดทำเว็บเซิร์ฟเวอร์ ซึ่งในการจัดทำเว็บเซิร์ฟเวอร์นั้นซอฟต์แวร์เหล่านี้เป็น Open Source สำหรับนักพัฒนาหรือนักศึกษาทั่วไปสามารถเข้าถึงและมีตัวอย่างการสอนอยู่ทั่วไป และยังรองรับระบบปฏิบัติการอย่าง Raspbian ของ Raspberry Pi ซึ่งมีพื้นฐานมาจาก Linux ซึ่งการติดตั้งซอฟต์แวร์ดังกล่าวมีชื่อเรียกรวมๆว่า LAMP(Linux Apache MySQL PHP) มีการติดตั้งโดยใช้คำสั่งดังนี้

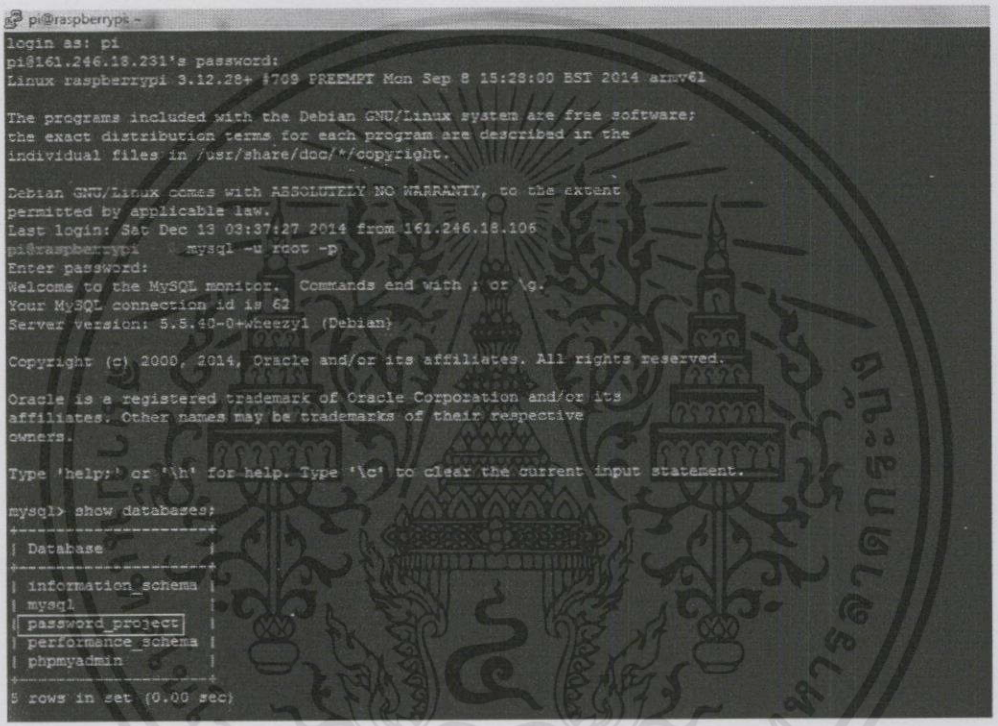
- ติดตั้ง Apache โดยใช้คำสั่ง sudo apt-get install apache2

เมื่อติดตั้ง Apache เว็บเซิร์ฟเวอร์แล้ว สามารถเช็คได้ว่าเซิร์ฟเวอร์นั้นทำงานแล้วโดยการเข้าไป IP เครื่องของ Raspberry Pi จะขึ้นหน้าเว็บเพ็จตามรูปที่ 3.12

รูปที่ 3.12 หน้าเว็บเพ็จ ยืนยันการทำงานของเว็บเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ติดตั้ง MySQL โดยการใช้คำสั่ง sudo apt-get install mysql-server เมื่อติดตั้ง MySQl เสร็จแล้วสามารถเข้าไปจัดทำระบบฐานข้อมูลได้โดยการใช้คำสั่ง mysql -u root –p แล้วทำการใส่รหัสผ่านของ root ที่ได้กำหนดไว้ในตอนที่เราติดตั้ง MySQL เมื่อเราเข้าไปแล้วเราสามารถทำฐานข้อมูลได้ เบื้องต้นทางผู้จัดทำได้ทำการสร้างฐานข้อมูลของรหัสเพื่อที่จะใช้ล็อคอินเข้าสู่หน้าเว็บควบคุมอุปกรณ์เซ็นเซอร์และแสดงผล แสดงการเข้า MySQL และ ฐานข้อมูลของรหัสที่ต้องการให้เข้าถึงหน้าควบคุมและแสดงผลของเซนเซอร์ดังรูปที่ 3.13



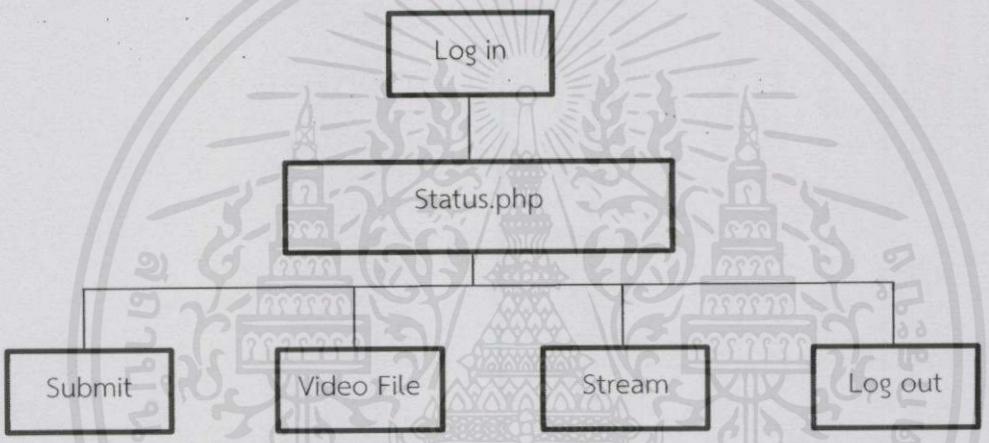
รูปที่ 3.13 ฐานข้อมูลของรหัสผ่านเข้าหน้าควมคุมเซ็นเซอร์และแสดงผล

- ติดตั้ง PHP โดยใช้คำสั่ง sudo apt-get install php5 libapache2-mod-php5 เมื่อทำการติดตั้ง PHP ไปแล้วเราสามารถเขียนหน้าเว็บที่มีการเชื่อมต่อกับส่วนของฐานข้อมูลโดยจะแสดงผลมาในรูปแบบของ HTML

ในส่วนของการควบคุมผ่านหน้าเว็บนั้น ผู้ใช้งานสามารถทำการควบคุมการเปิด-ปิดการทำงานของเซ็นเซอร์ในแต่ละส่วนได้ มีการแสดงสถานการณ์ทำงานของเซ็นเซอร์ และสามารถดูวีดีโอที่ได้จากการตรวจจับของเซ็นเซอร์ตรวจจับการเคลื่อนไหว อีกทั้งยังเปิดโหมดสตรีมได้

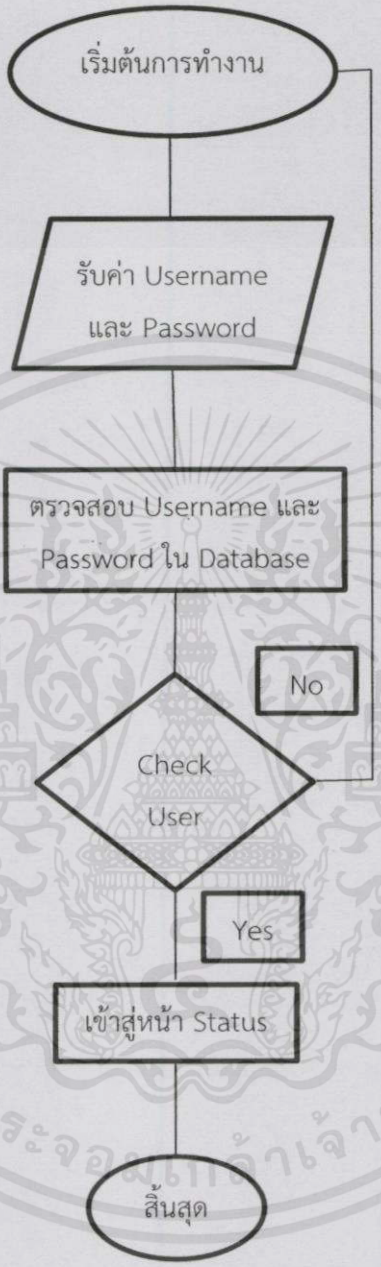
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมและแสดงผลผ่านหน้าเว็บถูกออกแบบมาให้ผู้ใช้งานเข้าถึงด้วย Username และ Password ผ่านหน้าเว็บซึ่งเก็บค่าไว้ใน Database เมื่อผู้ใช้งาน Log in เข้าระบบแล้ว ผู้ใช้จะสามารถตรวจสอบสถานการณ์ทำงานของเซ็นเซอร์ผ่านหน้าเว็บได้ ซึ่งจะแสดงทั้งสถานะของการเปิด – ปิด ตัวเซ็นเซอร์ และสถานะ การตรวจจับของเซ็นเซอร์ นอกจากนั้นผู้ใช้งานยังสามารถควบคุมได้ว่าจะให้เซ็นเซอร์ตัวไหนทำงานได้ตามต้องการ และมีโหมดสตรีมที่สามารถเข้าดูการสตรีมผ่านหน้าเว็บได้ และสามารถเข้าไปดูวิดีโอที่ได้จากการตรวจจับของเซ็นเซอร์ตรวจจับการเคลื่อนไหวได้



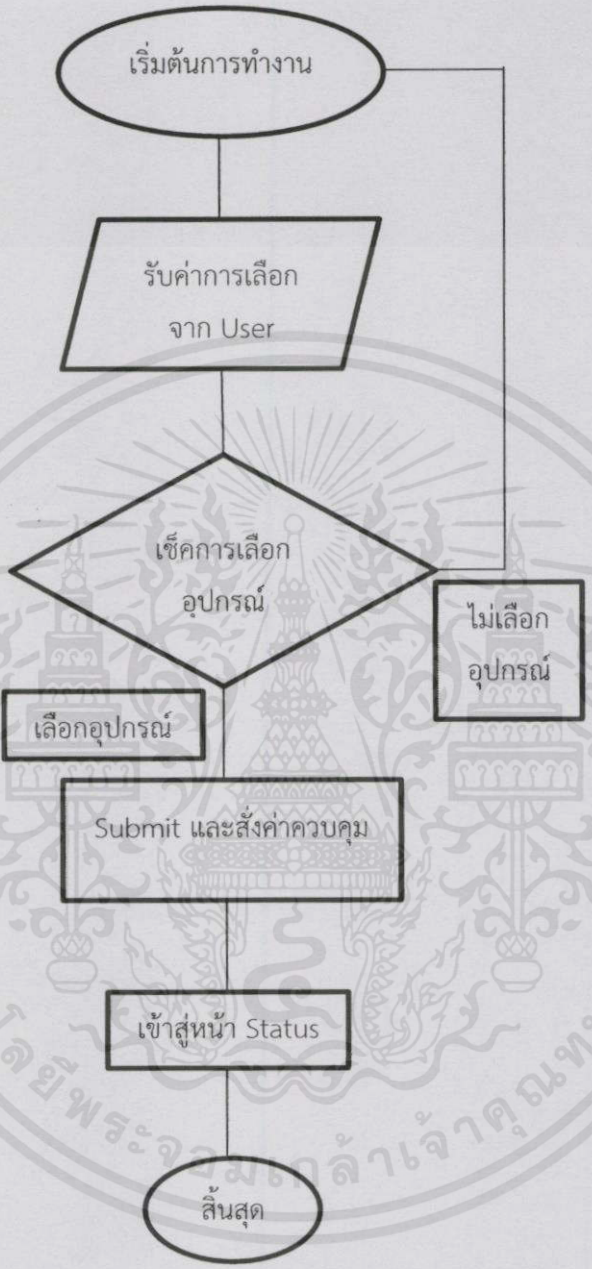
รูปที่ 3.14 แผนผังการทำงานเว็บ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.15 แผนผังการทำงาน การเข้าใช้งานของผู้ใช้งานผ่านหน้าเว็บ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

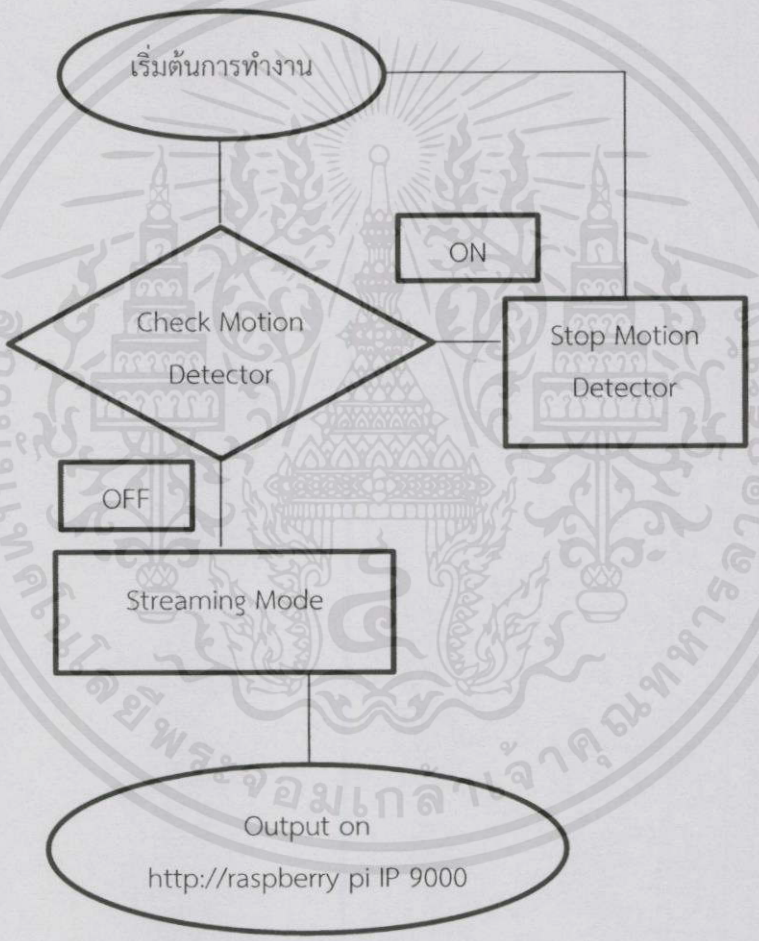


รูปที่ 3.16 แผนผังการทำงาน การรับค่าการควบคุมการเลือกใช้อุปกรณ์ของผู้ใช้งานผ่านหน้าเว็บ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.6 การทำงานของกล้องในโหมดสตรีม

สำหรับโหมดสตรีมนั้นเป็นอีกทางเลือกสำหรับผู้ใช้งาน นอกจากการทำงานของกล้องในโหมด ตรวจจับการเคลื่อนไหวแล้ว การใช้งานในโหมดสตรีมนั้นถูกออกแบบมาให้ใช้งานแยกจากโหมดตรวจจับการเคลื่อนไหว การที่จะใช้งานโหมดสตรีมจึงจำเป็นต้องปิดโหมดตรวจจับการเคลื่อนไหวก่อน เพราะอุปกรณ์ที่ใช้เป็นกล้องตัวเดียวกันจึงใช้พร้อมกันไม่ได้ เพราะฉะนั้นจึงขึ้นอยู่กับความต้องการของผู้ใช้งานเอง สำหรับผังการทำงานของโหมดสตรีมเป็นไปตามรูปที่ 3.17



รูปที่ 3.17 แผนผังการทำงานของกล้องในโหมดสตรีม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 เครื่องมือที่ใช้ในการทดลอง

1) Raspberry Pi

2) รีดสวิตซ์ 2 ตัว

3) แม่เหล็ก

4) MQ – 2 (เซ็นเซอร์ตรวจจับควัน)

5) MCP3002 (Analog to Digital Convertor)

6) ตัวต้านทาน 1K โอห์ม 2 ตัว

7) LED 2 ตัว

8) ลำโพง

9) คีย์บอร์ด

10) เมาส์

11) Oscilloscope

12) Raspberry Pi Camera Module

3.3 การจัดเก็บผลการทดลอง

1) เก็บผลการทดลองแรงดันไฟฟ้าที่ Raspberry Pi สามารถ Detect ได้ว่าเป็นลอจิค HIGH

2) เก็บผลการทดลองแรงดันไฟฟ้าเมื่อใช้สายไฟที่มีความยาว 5 เมตร 10 เมตร 15 เมตร และ 20 เมตร

3) เก็บผลการทดลองการทำงานของเซ็นเซอร์แม่เหล็ก และแสดงผลค่าแรงดันไฟฟ้าบนขา GPIO ที่เชื่อมต่อกับ Raspberry Pi เพื่อนำไปเป็นค่า Logic ที่ใช้ในโปรแกรม

4) เก็บผลการทดลองการทำงานของเซ็นเซอร์ตรวจจับควันโดยใช้สายไฟที่มีความยาว 10 เมตร และแสดงผลค่าตัวเลขปริมาณของควันหรือก๊าซไวไฟ บนหน้า Python Shell

5) เก็บผลการทดลองการทำงานของเซ็นเซอร์ตรวจจับความเคลื่อนไหว แสดงผลการบันทึกวีดีโอหากมีสิ่งที่เคลื่อนไหวผ่านกล้องและแสดงไฟล์วีดีโอที่บันทึก

6) เก็บผลการควบคุมการรับ-ส่งข้อมูล Database โดยการเลือกเปิด-ปิดอุปกรณ์ แสดงผล Database ที่เขียนบน PHPMyAdmin และแสดงผลการเปิด-ปิดการใช้งานอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7) เก็บผลการตอบสนองแจ้งเตือน Status Database บนหน้าเว็บ แสดงผลหน้าเว็บที่แสดง Status และหน้า database ที่เขียนบน PHPMyAdmin

8) เก็บผลการทำงานโปรแกรมในโหมดสตรีม แสดงผลหน้าที่มีสตรีม เมื่อเริ่มใช้งานโหมดนี้

9) เก็บผลการแจ้งเตือน Video File (Download)

10) เก็บผลการแจ้งเตือน SMS

11) เก็บผลการแจ้งเตือนโดยการโทรเข้าเบอร์ผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

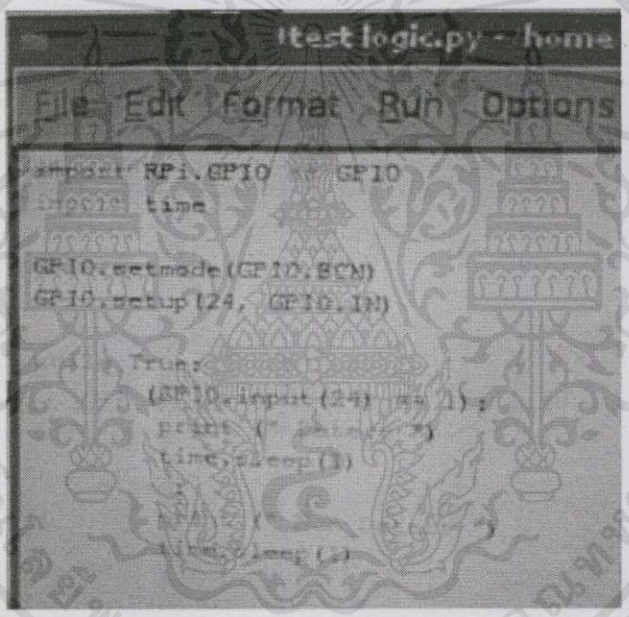
บทที่ 4

ผลการทดลอง

4.1 ผลการทดลองวัดค่าแรงดันไฟฟ้า

4.1.1 ค่าแรงดันไฟฟ้าที่ Raspberry Pi สามารถ Detect ได้ว่าเป็นลอจิก HIGH

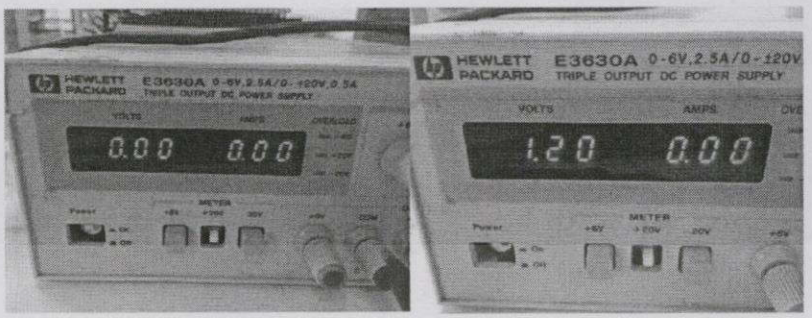
ไมโครคอมพิวเตอร์ Raspberry pi เป็นอุปกรณ์ในการควบคุมการทำงานของเซ็นเซอร์ทั้งหมด การเชื่อมต่อกับอุปกรณ์เซ็นเซอร์จะเชื่อมต่อผ่านขา GPIO ของ Raspberry pi การทดลองนี้สามารถบอกได้ว่าการเชื่อมต่อขา GPIO ของ Raspberry pi จะ Detect ลอจิก HIGH ได้ที่แรงดันไฟฟ้าเท่าไหร่ เพื่อที่จะได้นำข้อมูลนี้ไปใช้ในการเขียนโปรแกรมที่มีการใช้งาน Logic



รูปที่ 4.1 โปรแกรมการวัดค่าลอจิกโดยใช้ภาษาไพธอน

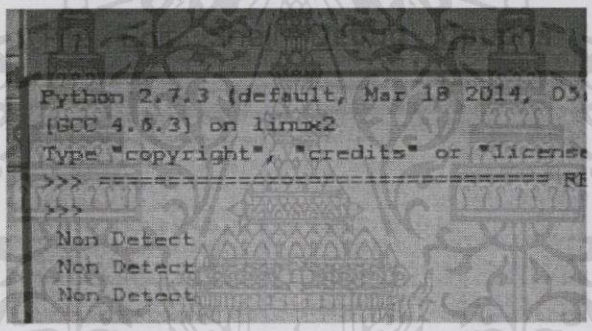
รูปที่ 4.1 การทำงานของโปรแกรมการวัดค่าลอจิกมีการทำงานดังนี้ ขา GPIO ที่ใช้มี 2 ขาคือ GND และ GPIO 24 จากนั้นจะเขียนโปรแกรมทำการติดตั้งขา GPIO 24 ตามการเรียงขา GPIO แบบ BCM และทำการกำหนดเงื่อนไขว่า หากขา GPIO 24 มีลอจิก HIGH จะทำให้โปรแกรมแสดงข้อความว่า “Detect” หากยังเป็นลอจิก LOW โปรแกรมจะแสดงคำว่า “Non Detect”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

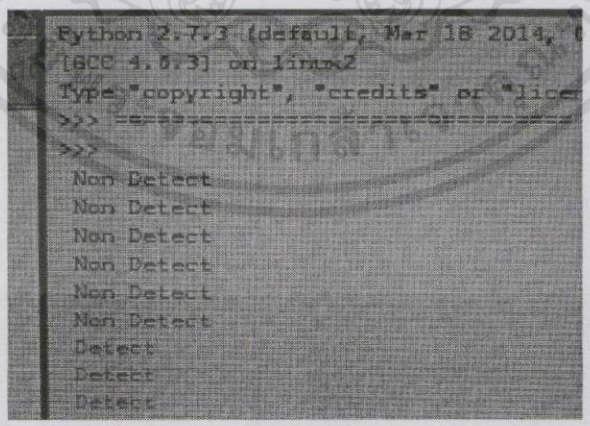


รูปที่ 4.2 แรงดันไฟฟ้าจาก DC Power Supply ต่อเข้า GPIO Raspberry Pi

รูปที่ 4.2 เมื่อทำการเขียนโปรแกรมแล้วทำการป้อนแรงดันไฟฟ้าเข้าสู่ขา GPIO 24 ของ Raspberry pi จากแรงดันไฟฟ้า 0 V แล้วค่อยๆเพิ่มขึ้นไปจนถึง 1.2 V



รูปที่ 4.3 ผลการทดลองเมื่อค่าแรงดันไฟฟ้าของ DC Power supply ในช่วง 0 - 1.2 V



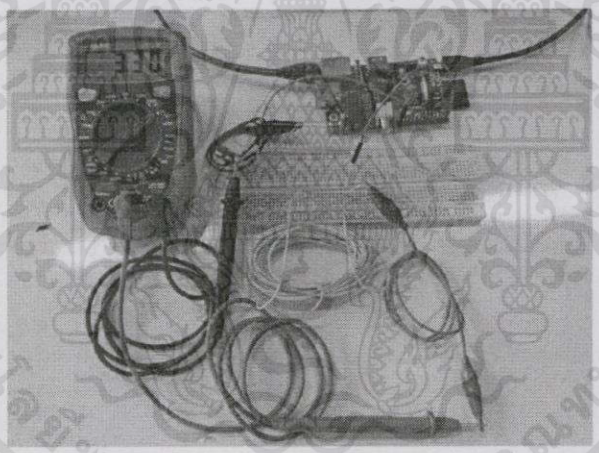
รูปที่ 4.4 ผลการทดลองเมื่อค่าแรงดันไฟฟ้าของ DC Power supply มากกว่า 1.2 V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.3 และ 4.4 ผลการทดลองเมื่อทำการป้อนแรงดันไฟฟ้าในช่วง 0 – 1.2 V Raspberry Pi ยังไม่สามารถ Detect หาค่าลอจิก HIGH ได้ แต่เมื่อเพิ่มขึ้นเป็น 1.3 V Raspberry pi สามารถ Detect ได้ แสดงว่า Raspberry pi จะสามารถ Detect แรงดันไฟฟ้าที่เข้ามาที่ขา GPIO ว่าเป็นลอจิก HIGH หากค่าแรงดันไฟฟ้าที่เข้ามามีไม่ถึง 1.3 V Raspberry Pi จะ Detect ว่าเป็นลอจิก LOW

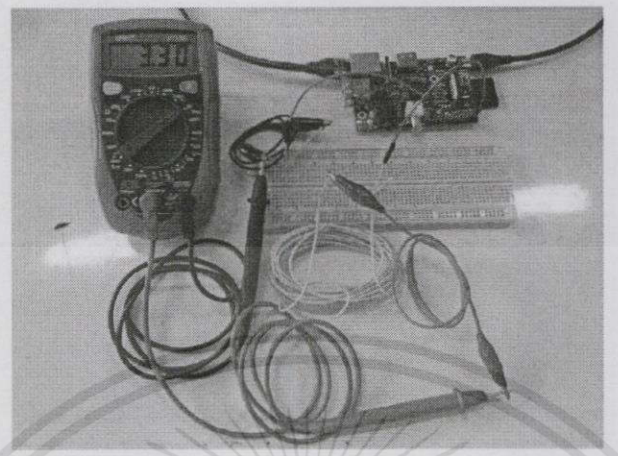
4.1.2 วัดค่าแรงดันไฟฟ้าระหว่าง Input และ Output เมื่อปล่อยแรงดันไฟฟ้าผ่านสายไฟที่มีความยาว 5 เมตร 10 เมตร 15 เมตร และ 20 เมตร

การเชื่อมต่อกันระหว่างอุปกรณ์เซ็นเซอร์และ Raspberry Pi จะใช้สายไฟขนาด 22AWG ความยาว ตามสถานที่ติดตั้งอุปกรณ์ แต่ในการทดลองเบื้องต้น ได้ทดลองวัดค่าแรงดันไฟฟ้าระหว่าง Input และ Output ในความยาว 5 เมตร 10 เมตร 15 เมตร และ 20 เมตร ตามลำดับ การทดลองนี้ได้จัดทำเพื่อจะหาว่าในสายไฟที่ได้นำมาใช้จะมีการสูญเสียค่าแรงดันไฟฟ้าเท่าใด เมื่อมีความยาวมากขึ้นเรื่อยๆ จะได้ออกแบบการวางตำแหน่งตัวอุปกรณ์เซ็นเซอร์และ Raspberry Pi ได้อย่างเหมาะสม

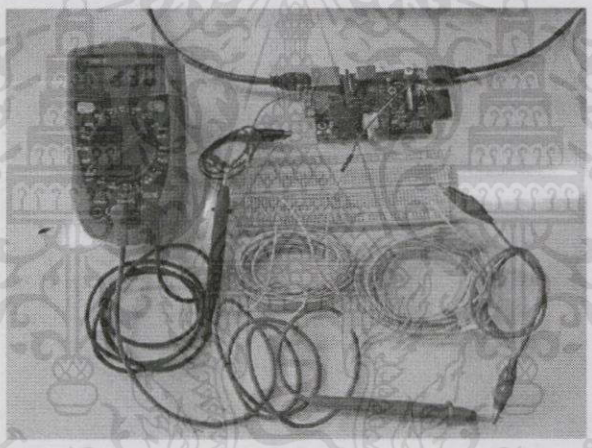


รูปที่ 4.5 วัดค่าแรงดันไฟฟ้า Input ก่อนปล่อยเข้าสู่สายไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

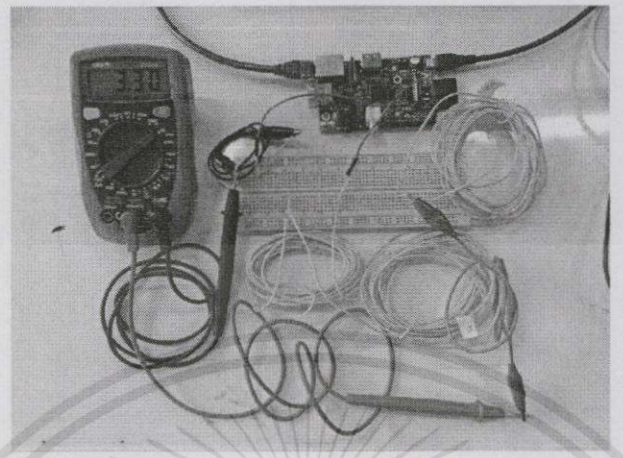


รูปที่ 4.6 วัดค่าแรงดันไฟฟ้า Output เมื่อออกจากสายไฟ ที่มีความยาว 5 เมตร

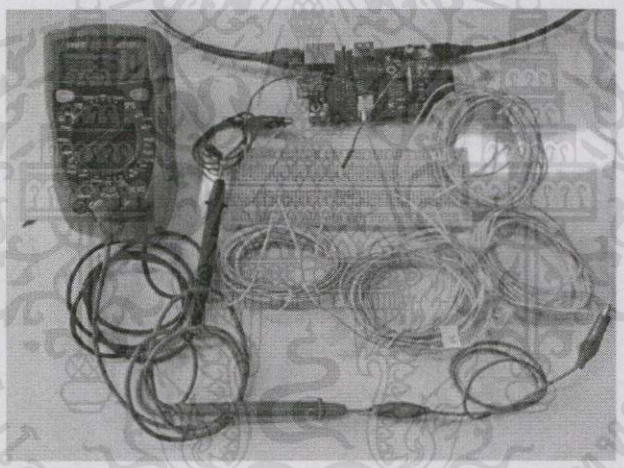


รูปที่ 4.7 วัดค่าแรงดันไฟฟ้า Output เมื่อออกจากสายไฟ ที่มีความยาว 10 เมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 วัดค่าแรงดันไฟฟ้า Output เมื่อออกจากสายไฟ ที่มีความยาว 15 เมตร



รูปที่ 4.9 วัดค่าแรงดันไฟฟ้า Output เมื่อออกจากสายไฟ ที่มีความยาว 20 เมตร

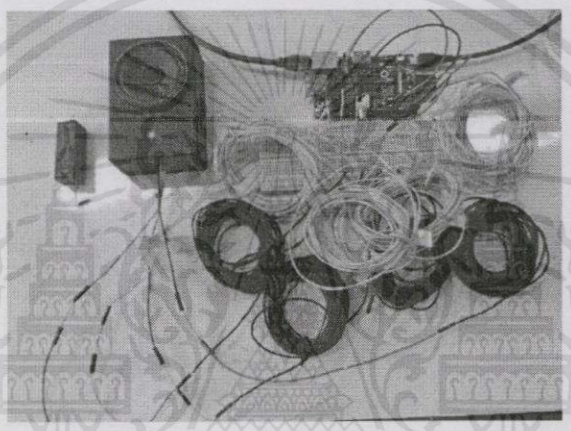
รูปที่ 4.5 เป็นการวัดค่า Input ที่เป็นแรงดันไฟฟ้าที่ต่อเข้ากับสายไฟ วัดค่าแรงดันไฟฟ้าได้ 3.3 V และรูปที่ 4.6 4.7 4.8 และ 4.9 เป็นการวัดค่า Output ที่เป็นแรงดันไฟฟ้าที่ผ่านสายไฟมา วัดค่าแรงดันไฟฟ้าได้ 3.3 V เช่นเดียวกัน ใช้แหล่งจ่ายแรงดันไฟฟ้าจาดขา GPIO 3.3 V ของ Raspberry Pi จากการทดลองนี้แสดงให้เห็นว่า ในสายไฟที่ได้นำมาใช้ไม่มีการสูญเสียค่าแรงดันไฟฟ้า เนื่องจากแรงดันไฟฟ้าที่ได้จาก Output มีค่าเท่ากับ Input ถือว่าสายไม่มีความต้านทานไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

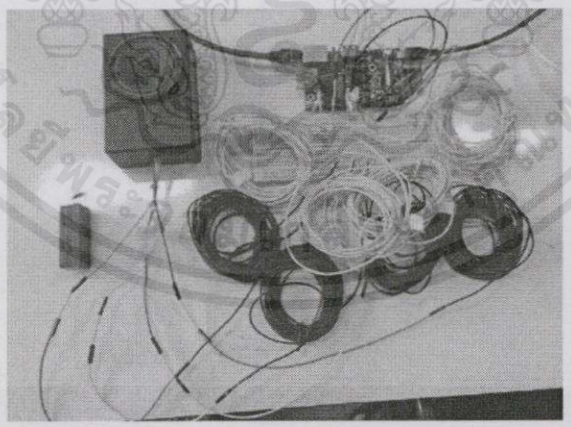
4.2 ผลการทดลองการทำงานของเซ็นเซอร์แม่เหล็ก และแสดงผลค่าแรงดันไฟฟ้าบนขา GPIO

4.2.1 การทดลองใช้งานเซ็นเซอร์แม่เหล็ก

ภายในตัวเซ็นเซอร์แม่เหล็กจะประกอบด้วย รีดสวิตซ์, Buzzer 5V, วงจร LED และมีตัวแม่เหล็กอยู่ภายนอก ทำการสร้างสนามแม่เหล็กบริเวณหน้าสัมผัสของรีดสวิตซ์ เพื่อทำให้รีดสวิตซ์ ทำหน้าที่เป็นสวิตซ์ เปิด-ปิดแรงดันไฟฟ้าเพื่อประมวลผลเป็นค่าลอจิกใน Raspberry Pi



รูปที่ 4.10 ทดลองการใช้งานรีดสวิตซ์ ขณะมีสนามแม่เหล็กอยู่ในบริเวณ



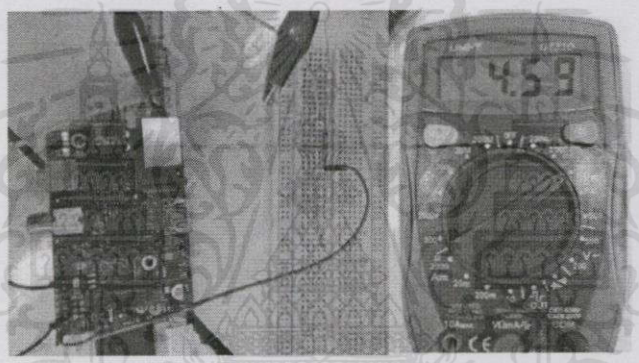
รูปที่ 4.11 ทดลองการใช้งานรีดสวิตซ์ ขณะไม่มีสนามแม่เหล็กอยู่ในบริเวณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

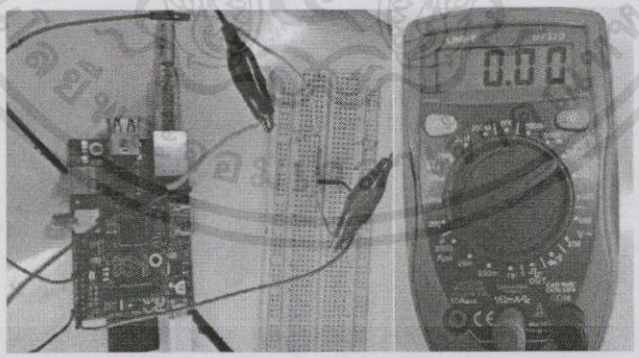
รูปที่ 4.10 และ 4.11 เป็นการทดลองเซ็นเซอร์แม่เหล็ก โดยการเคลื่อนแท่งแม่เหล็กเข้าและออกบริเวณหน้าสัมผัสของหรีดสวิตช์ ผลการทดลองจากรูปที่ 4.10 เป็นขั้นตอนที่รีดสวิตช์มีแท่งแม่เหล็กอยู่บริเวณหน้าสัมผัสเซ็นเซอร์จะมีแสงไฟที่หลอด LED แต่ Buzzer ยังไม่ทำงาน ผลการทดลองจากรูปที่ 4.11 เป็นขั้นที่รีดสวิตช์มีมีแท่งแม่เหล็กอยู่บริเวณหน้าสัมผัส แสงไฟ LED ที่เซ็นเซอร์แม่เหล็กจะดับ และ Buzzer จะทำงานโดยส่งเสียงดังออกมา การทดลองทั้งหมดนี้ทดลองโดยใช้สายที่มีความยาว 10 เมตร

4.2.2 การทดลองวัดค่าแรงดันของเซ็นเซอร์แม่เหล็ก

ภายในตัวเซ็นเซอร์แม่เหล็กจะประกอบด้วย รีดสวิตช์ Buzzer 5 V วงจร LED การทดลองนี้จะเป็นการวัดค่าแรงดันไฟฟ้า Input และ Output ที่ผ่านตัวรัดสวิตช์ เพราะรีดสวิตช์เป็นอุปกรณ์ที่เป็นทางผ่านของแรงดันไฟฟ้าที่ Raspberry Pi นับว่าเป็นการส่งค่าลอจิก

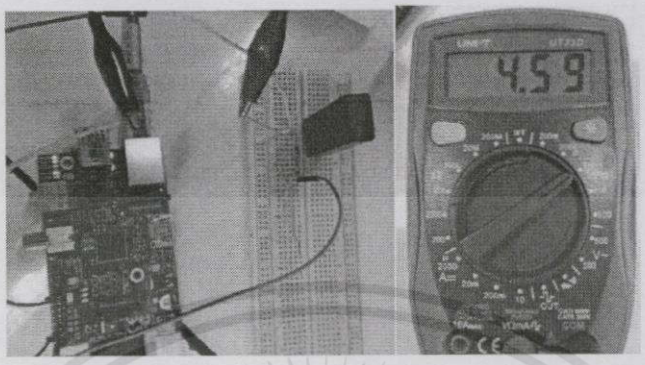


รูปที่ 4.12 วัดค่าแรงดันไฟฟ้า Input รีดสวิตช์



รูปที่ 4.13 วัดค่าแรงดันไฟฟ้า Output รีดสวิตช์ ขณะ Close Circuit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.14 วัดค่าแรงดันไฟฟ้า Output รีดสวิตซ์ ขณะ Open Circuit

จากรูปที่ 4.12 4.13 และ 4.14 เป็นการวัดค่าแรงดันที่ขา GPIO เพื่อที่จะนำไปใช้กับการเขียนโปรแกรมโดย Raspberry Pi ขา GPIO ของ Raspberry Pi เป็นขาที่เราจะนำค่าลอจิกไปใช้ ประเภทของลอจิกมีสองประเภทคือลอจิก High (สัญญาณไฟมากกว่า 1.3 V) และลอจิก Low (สัญญาณไฟน้อยกว่า 1.3 V)

ผลการทำงานของรีดสวิชต์มี 2 กรณี ดังนี้

กรณีที่ 1 รีดสวิชต์ เมื่อไม่มีแท่งแม่เหล็กอยู่บริเวณหน้าสัมผัส วัดแรงดันไฟฟ้าที่ขา Output ของ รีดสวิชต์ ได้ 0 V แสดงว่า เราจะได้ค่า Logic Low จากรีดสวิชต์

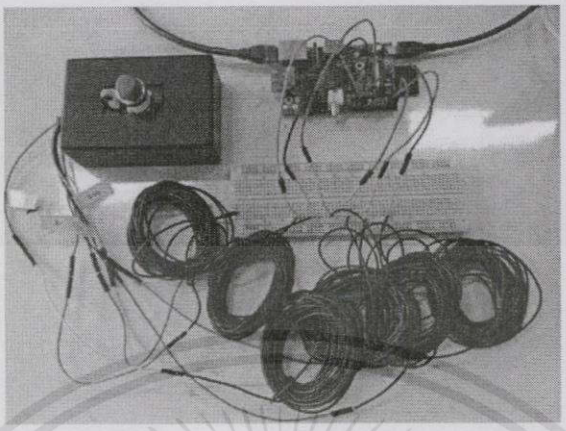
กรณีที่ 2 รีดสวิชต์เมื่อมีแท่งแม่เหล็กอยู่บริเวณหน้าสัมผัส วัดแรงดันไฟฟ้าที่ขา Output ของรีดสวิชต์ได้ 4.59 V ตามแรงดันไฟฟ้าที่จ่ายมา แสดงว่าเราจะได้ค่า Logic High จากรีดสวิชต์

ค่า Logic ที่ได้ สามารถนำไปต่อเข้ากับขา GPIO ของ Raspberry pi เพื่อนำเข้าไปใช้ในโปรแกรม

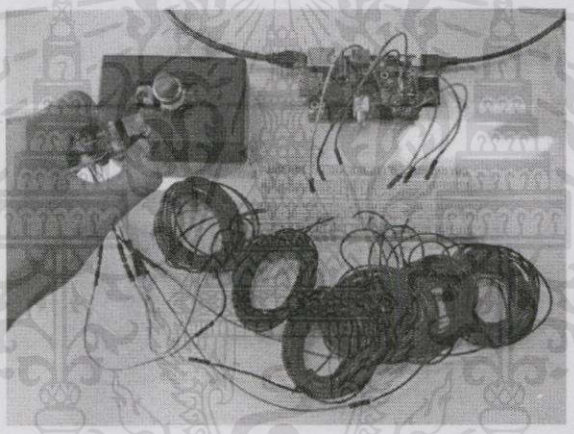
4.3 ผลการทดลองการทำงานของเซ็นเซอร์ตรวจจับควัน

เซ็นเซอร์ตรวจจับควัน สามารถตรวจจับควันและก๊าซไวไฟได้ ทดลองทำให้เซ็นเซอร์ทำงานโดยให้มีควันและก๊าซไวไฟอยู่ในบริเวณตัวตรวจจับ และรอดูว่าสามารถนำไปเขียนโปรแกรมผ่านการประมวลผลใน Raspberry pi ได้หรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

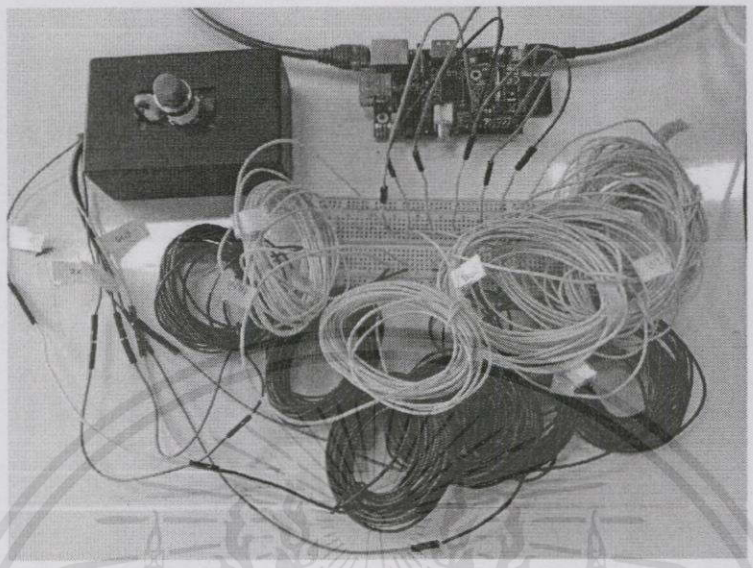


รูปที่ 4.15 เชื่อมต่อเซ็นเซอร์ตรวจจับควันกับ Raspberry Pi

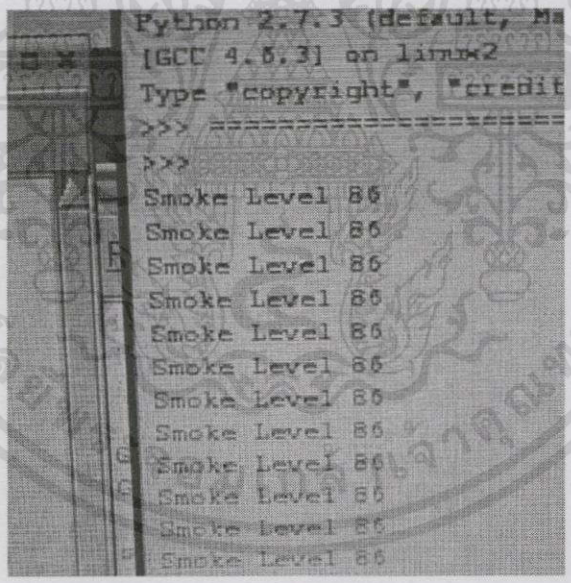


รูปที่ 4.16 ทดลองนำควันหรือก๊าซไวไฟเข้ามาบริเวณตรวจจับของอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

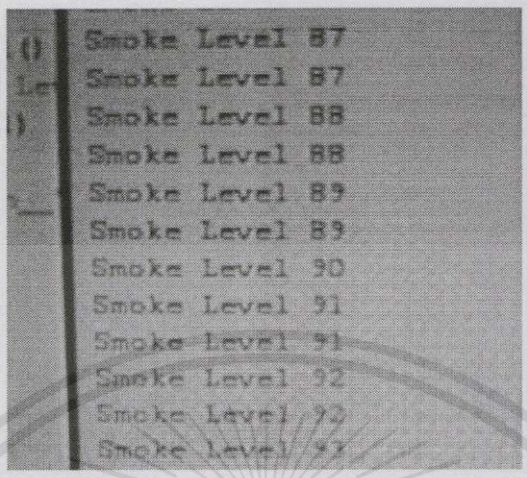


รูปที่ 4.17 ทดสอบการทำงานของเซ็นเซอร์โดยใช้สายไฟยาว 10 เมตร



รูปที่ 4.18 ระดับควันในหน้าโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



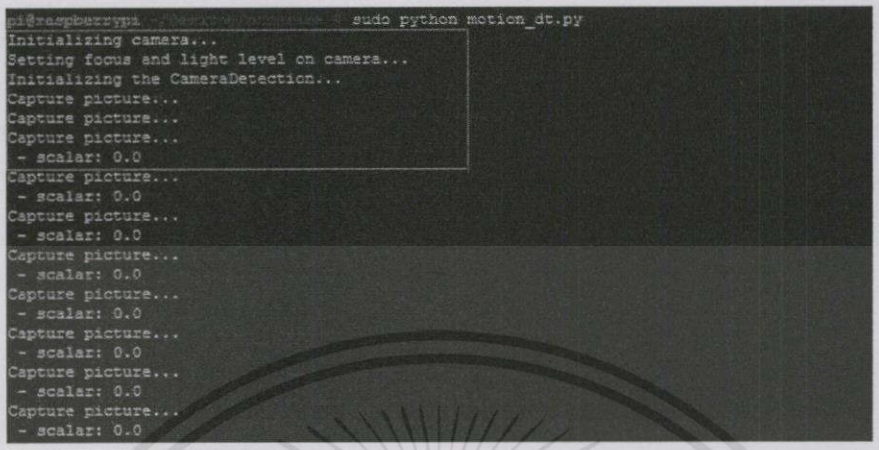
รูปที่ 4.19 ระดับควันมีค่ามากเกินที่กำหนด ทางโปรแกรมจะแจ้งเตือน

จากรูปที่ 4.15 4.16 4.17 4.18 และ 4.19 เป็นการทดสอบการใช้งานของเซ็นเซอร์ตรวจจับควันกับ Raspberry Pi ว่าจะสามารถอ่านค่าปริมาณของควันหรือก๊าซไวไฟออกมาเป็นรูปแบบใดโดยทดสอบร่วมกับการใช้สายไฟความยาว 10 เมตร หลังจากทดลองเสร็จการอ่านค่าที่ได้จะเป็นค่าปริมาณตัวเลขหลักสิบไปจนถึงหลักร้อย และสามารถปรับค่าได้โดยการปรับค่าความต้านทานที่ติดอยู่กับตัวเซ็นเซอร์ตรวจจับควัน ส่วนในด้านของโปรแกรมเราก็สามารถปรับค่าได้ว่าจะให้ตรวจจับตอนค่าปริมาณมากกว่าเท่าไหร่ เช่น ปรับความต้านทานที่ตัวเซ็นเซอร์ตรวจจับควันแล้วดูตัวเลขในโปรแกรมให้อยู่ประมาณ 87 แล้วเขียนโปรแกรมว่าให้ตรวจจับเมื่อมีค่าปริมาณควันมากกว่า 120 จะได้ผลการทำงานตามภาพที่ 4.18 และ 4.19

4.4 ผลการทดลองการทำงานของเซ็นเซอร์ตรวจจับความเคลื่อนไหว

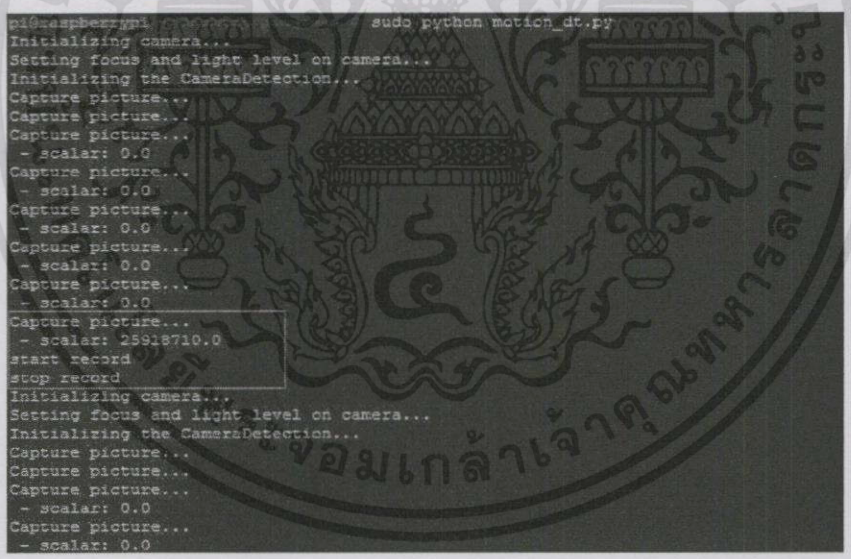
การทำงานของเซ็นเซอร์ตรวจจับความเคลื่อนไหวจะใช้กล้อง Raspberry Pi Camera Module เป็นอุปกรณ์หลักในการตรวจจับ โดยเขียนโปรแกรมเข้าไปเพื่อให้สามารถตรวจจับความเคลื่อนไหวได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.20 เริ่มต้นโปรแกรมตรวจจับความเคลื่อนไหว

จากรูปที่ 4.20 โปรแกรมจะทำการ Capture ภาพไว้ 2 ภาพ หลังจากนั้นเมื่อเริ่มต้น Capture ภาพที่ 3 ตัวโปรแกรมจะเริ่มทำการเปรียบเทียบความแตกต่างของพิกเซลภาพ



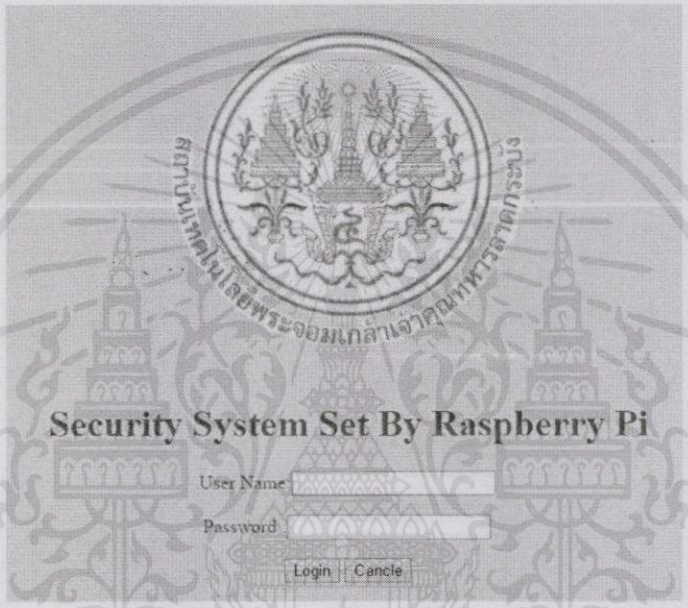
รูปที่ 4.21 โปรแกรม Motion เริ่มทำการ Detect

จากรูปที่ 4.21 โปรแกรมจะทำการเปรียบเทียบความต่างของภาพแล้วมีค่าความต่างเกินระดับพิกเซลที่กำหนดไว้ โปรแกรมจะทำการตรวจจับ เพื่อเป็นเงื่อนไขในการบันทึกวิดีโอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 ผลการทดลองในหน้า Database ที่สร้างขึ้นใน PhpMyAdmin เมื่อทำการเปิด-ปิดอุปกรณ์

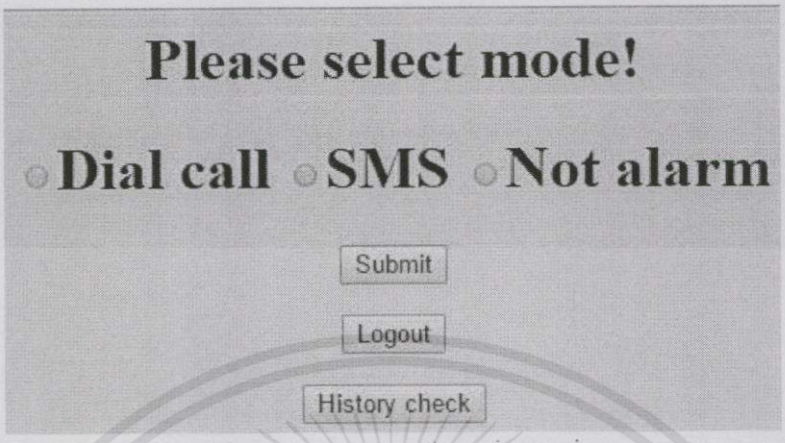
ผู้ใช้งานจะสามารถควบคุมการเปิด-ปิดอุปกรณ์เซ็นเซอร์ เช็คค่าสถานะของเซ็นเซอร์ ดาวน์โหลดไฟล์วีดีโอที่ได้จากเซ็นเซอร์ตรวจจับความเคลื่อนไหว และใช้งานในโหมด Streaming ได้ในหน้าเว็บไซต์



รูปที่ 4.22 หน้าเว็บ Login เข้าเว็บหลักในการควบคุมอุปกรณ์และแจ้งเตือน

จากรูปที่ 4.22 เป็นหน้าเว็บเพจที่ใช้ Login เพื่อเข้าใช้งานระบบ ส่วน User Name และ Password จะถูกลงทะเบียนไว้ใน Database อยู่แล้ว

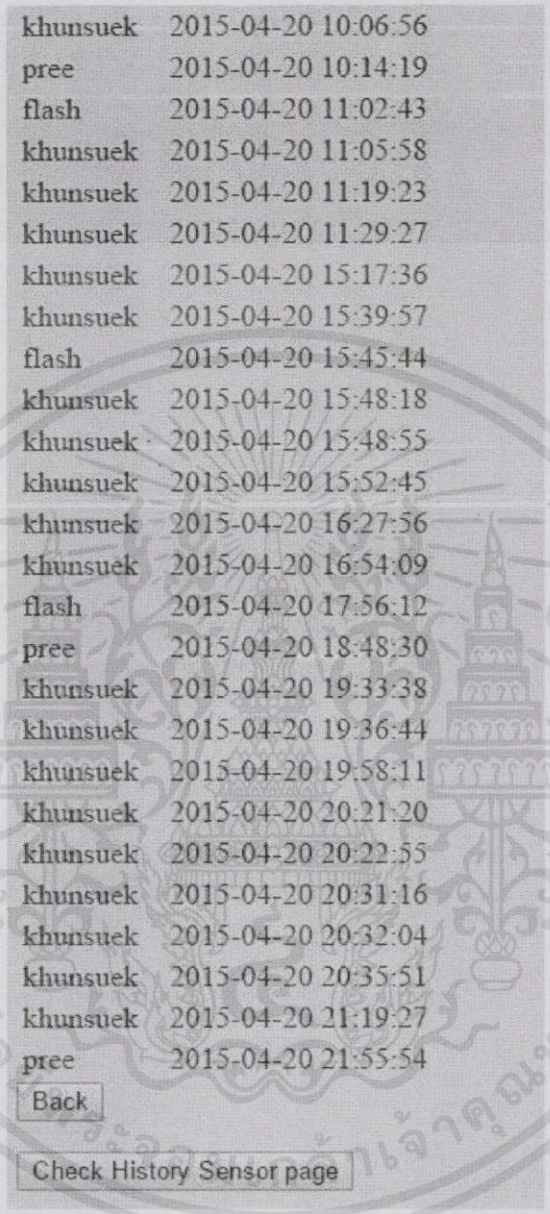
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.23 หน้าเว็บเลือกโหมดการแจ้งเตือน

จากรูปที่ 4.23 ภายในหน้านี้ ผู้ใช้สามารถเลือกโหมดการแจ้งเตือนได้ 3 แบบ คือ แจ้งเตือนโดยการโทรไปยังผู้ใช้ ส่งข้อความไปยังผู้ใช้ และไม่แจ้งเตือนเลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.24 หน้าเว็บดูประวัติการเข้าใช้งาน

จากรูปที่ 4.24 ภายในหน้านี้ ผู้ใช้สามารถดูประวัติการเข้าใช้งานของผู้ใช้ต่างๆได้ โดยจะระบุวัน เดือน ปี และเวลาที่ผู้ใช้ทำการล็อกอินเข้า-ออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

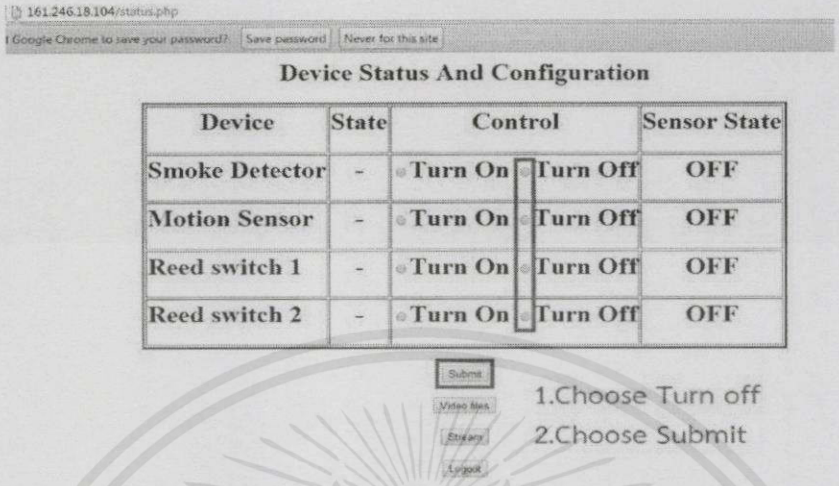
History Sensor Detected

Smoke Detector	Motion Detector	Reed switch 1	Reed switch 2
		2015-04-20 21:42:18	2015-04-20 21:42:18
		2015-04-20 21:41:36	2015-04-20 21:41:36
		2015-04-20 21:22:42	2015-04-20 21:22:42
		2015-04-20 21:22:41	2015-04-20 21:22:41
		2015-04-20 21:20:48	2015-04-20 21:20:48
		2015-04-20 21:17:15	2015-04-20 21:17:15
		2015-04-20 21:06:23	2015-04-20 21:06:23
2015-04-20 20:16:02	2015-04-20 21:47:57	2015-04-20 21:05:19	2015-04-20 21:05:19
2015-04-20 20:16:21	2015-04-20 21:44:05	2015-04-20 21:00:55	2015-04-20 21:00:55
2015-04-20 20:16:31	2015-04-20 20:39:05	2015-04-20 21:00:29	2015-04-20 21:00:29
2015-04-20 20:16:41	2015-04-20 20:38:29	2015-04-20 21:00:07	2015-04-20 21:00:07
2015-04-20 20:16:51	2015-04-20 20:21:56	2015-04-20 20:32:27	2015-04-20 20:32:27
2015-04-20 20:18:06	2015-04-20 19:35:18	2015-04-20 20:22:00	2015-04-20 20:22:00
2015-04-20 20:18:47	2015-04-20 16:27:21	2015-04-20 20:03:05	2015-04-20 20:03:05
2015-04-20 20:21:53	2015-04-20 15:26:56	2015-04-20 20:02:20	2015-04-20 20:02:20
		2015-04-20 19:34:26	2015-04-20 19:34:26
		2015-04-20 19:34:09	2015-04-20 19:34:09
		2015-04-20 18:48:00	2015-04-20 18:48:00
		2015-04-20 18:48:00	2015-04-20 18:48:00
		2015-04-20 18:47:59	2015-04-20 18:47:59
		2015-04-20 18:47:45	2015-04-20 18:47:45
		2015-04-20 18:47:45	2015-04-20 18:47:45
		2015-04-20 18:47:44	2015-04-20 18:47:44

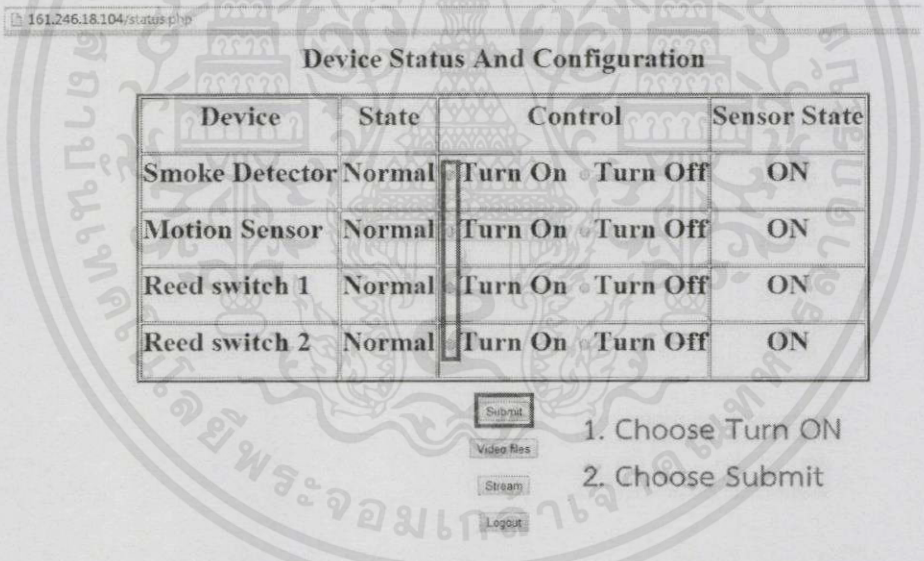
รูปที่ 4.25 หน้าเว็บดูประวัติการตรวจจับของเซ็นเซอร์

จากรูปที่ 4.25 ผู้ใช้สามารถดูประวัติการตรวจจับของเซ็นเซอร์ทุกชนิดย้อนหลังได้ โดยภายในหน้านี้จะบอกวัน เดือน ปี และเวลาที่เซ็นเซอร์ตรวจจับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.26 หน้าเว็บหลัก ใช้ในการการควบคุมและแจ้งเตือน(วิธีปิดอุปกรณ์)



รูปที่ 4.27 หน้าเว็บหลัก ที่ใช้ในการการควบคุมและแจ้งเตือน(วิธีเปิดอุปกรณ์)

จากรูปที่ 4.26 และ 4.27 เป็นการแสดงวิธีเปิด-ปิดการใช้งานของอุปกรณ์เซ็นเซอร์ เซ็นเซอร์ตรวจจับควัน เซ็นเซอร์ตรวจจับการเคลื่อนไหวและเซนเซอร์แม่เหล็กอีก 2 ตัว ซึ่งผู้ใช้งานสามารถเลือกที่จะเปิด-ปิดอุปกรณ์ได้ตามใจชอบ และในรูปภาพทั้งสองยังมี ปุ่มใช้เลือกการใช้งานของกล้องแบบโหมดสตรีมอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

←T→					device	status
	Edit	Inline Edit	Copy	Delete	smoke_detector	OFF
	Edit	Inline Edit	Copy	Delete	motion_detector	OFF
	Edit	Inline Edit	Copy	Delete	reed1	OFF
	Edit	Inline Edit	Copy	Delete	reed2	OFF

รูปที่ 4.28 ผลของ Database เมื่อปิดอุปกรณ์ทั้งหมด

←T→					device	status
	Edit	Inline Edit	Copy	Delete	smoke_detector	ON
	Edit	Inline Edit	Copy	Delete	motion_detector	OFF
	Edit	Inline Edit	Copy	Delete	reed1	OFF
	Edit	Inline Edit	Copy	Delete	reed2	OFF

รูปที่ 4.29 ผลของ Database เมื่อเปิดเฉพาะเซ็นเซอร์ตรวจจับควัน

←T→					device	status
	Edit	Inline Edit	Copy	Delete	smoke_detector	OFF
	Edit	Inline Edit	Copy	Delete	motion_detector	ON
	Edit	Inline Edit	Copy	Delete	reed1	OFF
	Edit	Inline Edit	Copy	Delete	reed2	OFF

รูปที่ 4.30 ผลของ Database เมื่อเปิดเฉพาะเซ็นเซอร์ตรวจจับการเคลื่อนไหว

←T→					device	status
	Edit	Inline Edit	Copy	Delete	smoke_detector	OFF
	Edit	Inline Edit	Copy	Delete	motion_detector	OFF
	Edit	Inline Edit	Copy	Delete	reed1	ON
	Edit	Inline Edit	Copy	Delete	reed2	OFF

รูปที่ 4.31 ผลของ Database เมื่อเปิดเฉพาะเซ็นเซอร์แม่เหล็กตัวที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

←T→					device	status
	Edit	Inline Edit	Copy	Delete	smoke_detector	OFF
	Edit	Inline Edit	Copy	Delete	motion_detector	OFF
	Edit	Inline Edit	Copy	Delete	reed1	OFF
	Edit	Inline Edit	Copy	Delete	reed2	ON

รูปที่ 4.32 ผลของ Database เมื่อเปิดเฉพาะเซ็นเซอร์แม่เหล็กตัวที่ 2

จากรูปที่ 4.28 4.29 4.30 4.31 และ 4.32 เป็นการแสดง Database บน PHPMyAdmin การเปิด-ปิดอุปกรณ์มาจากการสั่งงานบนหน้าเว็บเพจของโปรแกรม เมื่อบนหน้าเว็บเพจถูกใช้งานก็จะทำงานโดยส่งคำสั่งต่างๆ มาให้ Database เพื่อที่จะติดต่อกัน แล้วส่งค่าสถานะต่างๆ กลับไปแสดงผลให้ผู้ใช้งาน บนหน้าเว็บเพจ

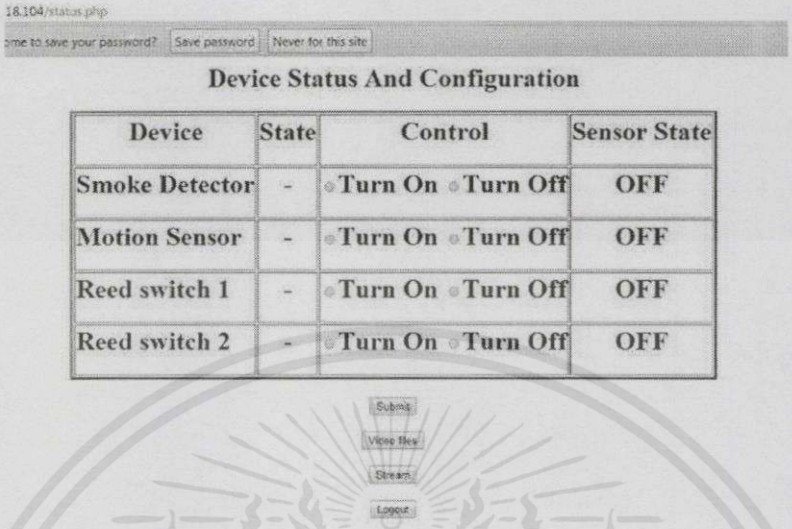
4.6 การแจ้งเตือนบนหน้าเว็บเพจ เมื่อ Database มีการเปลี่ยนแปลงสถานะ อันเนื่องมาจากการเปิด-ปิดอุปกรณ์ และการตรวจจับของอุปกรณ์

การทดลองนี้เป็นการทำงานร่วมกันระหว่างหน้าเว็บเพจกับ Database จะแสดงผลการทำงานเป็นหน้าการทำงานของโปรแกรมทั้งสองควบคู่กัน

←T→					sensor	status
	Edit	Inline Edit	Copy	Delete	smoke_detector	-
	Edit	Inline Edit	Copy	Delete	motion_detector	-
	Edit	Inline Edit	Copy	Delete	sensor1	-
	Edit	Inline Edit	Copy	Delete	sensor2	-

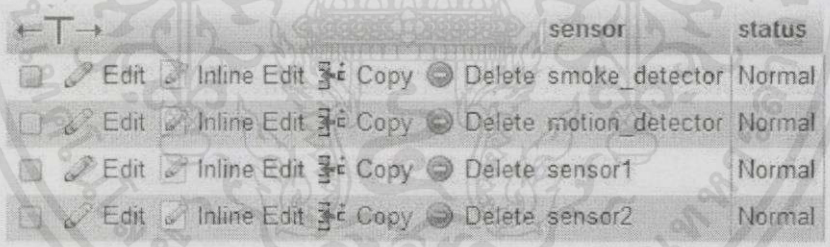
รูปที่ 4.33 ผลของ Database เมื่อปิดอุปกรณ์ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.34 หน้าเว็บเลือกโหมดการแจ้งเตือน

จากรูปที่ 4.33 และรูปที่ 4.34 จะเห็นได้ว่าเมื่อทดลองปิดอุปกรณ์ทั้งหมด หน้าเว็บเพจจะแสดงสถานะของอุปกรณ์เป็น OFF และสถานะของการตรวจจับเป็นเครื่องหมายขีดค่า



รูปที่ 4.35 ผลของ Database เมื่อเปิดอุปกรณ์ทั้งหมดและอยู่ในสถานะปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

18.104/status.php

Device Status And Configuration

Device	State	Control	Sensor State
Smoke Detector	Normal	Turn On Turn Off	ON
Motion Sensor	Normal	Turn On Turn Off	ON
Reed switch 1	Normal	Turn On Turn Off	ON
Reed switch 2	Normal	Turn On Turn Off	ON

Submit

Video files

Stream

Logout

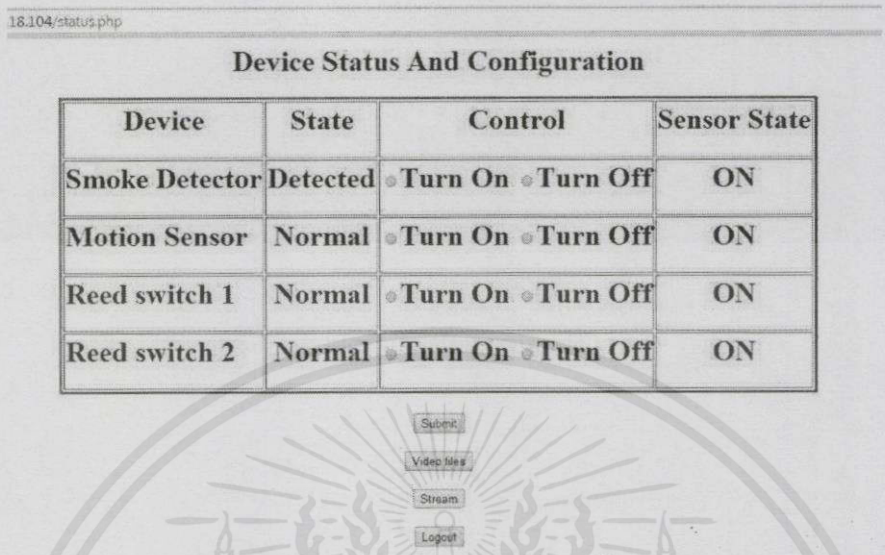
รูปที่ 4.36 หน้าเว็บเพจ เมื่อเปิดอุปกรณ์ทั้งหมดและอยู่ในสถานะปกติ

จากรูปที่ 4.35 และรูปที่ 4.36 จะเห็นได้ว่าเมื่อทดลองเปิดอุปกรณ์ทั้งหมด หน้าเว็บเพจจะแสดงสถานะของอุปกรณ์ ON และสถานะของการตรวจจับเป็น Normal

	sensor	status
Edit Inline Edit Copy Delete	smoke_detector	Detected
Edit Inline Edit Copy Delete	motion_detector	Normal
Edit Inline Edit Copy Delete	sensor1	Normal
Edit Inline Edit Copy Delete	sensor2	Normal

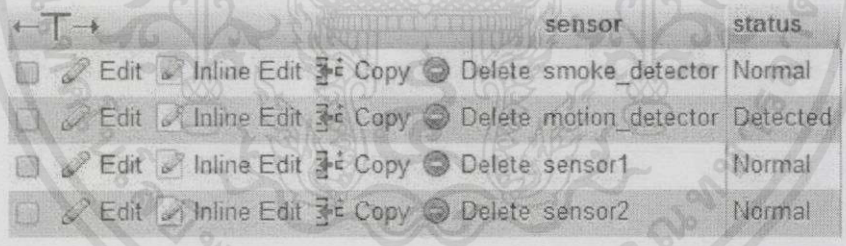
รูปที่ 4.37 ผลของ Database เมื่อเซ็นเซอร์ตรวจจับควันมีการตรวจจับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.38 หน้าเว็บเพจ เมื่อเซ็นเซอร์ตรวจจับควันเกิดการตรวจจับ

จากรูปที่ 4.37 และรูปที่ 4.38 จะเห็นว่าเมื่อเซ็นเซอร์ตรวจจับควันมีการตรวจจับ หน้าเว็บเพจจะแสดงสถานะของเซ็นเซอร์ตรวจจับควันว่า Detected



รูปที่ 4.39 ผลของ Database เมื่อเซ็นเซอร์ตรวจจับการเคลื่อนไหวเกิดการตรวจจับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.104/status.php

Device Status And Configuration

Device	State	Control	Sensor State
Smoke Detector	Normal	Turn On Turn Off	ON
Motion Sensor	Detected	Turn On Turn Off	ON
Reed switch 1	Normal	Turn On Turn Off	ON
Reed switch 2	Normal	Turn On Turn Off	ON

Submit

Video Mod

Stream

Logout

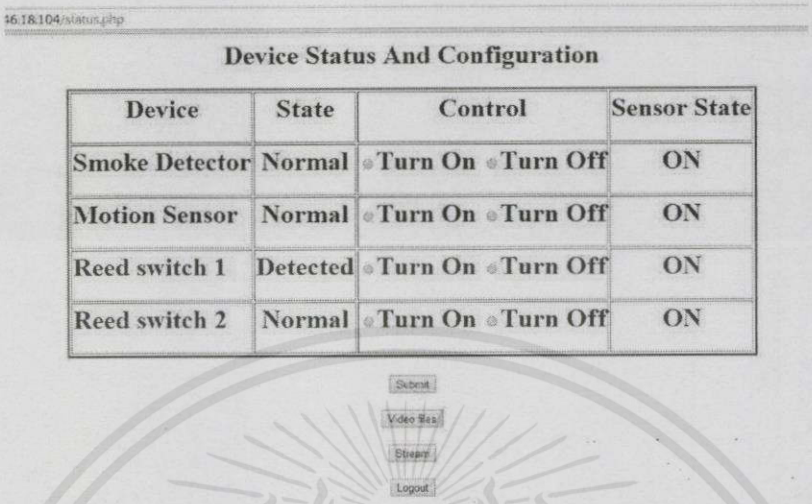
รูปที่ 4.40 หน้าเว็บเพจ เมื่อเซ็นเซอร์ตรวจจับการเคลื่อนไหวเกิดการตรวจจับ

จากรูปที่ 4.39 และรูปที่ 4.40 จะเห็นว่าเมื่อเซ็นเซอร์ตรวจจับการเคลื่อนไหวมีการตรวจจับ หน้าเว็บเพจจะแสดงสถานะของเซ็นเซอร์ตรวจจับการเคลื่อนไหวว่า Detected

←T→					sensor	status
Edit	Inline Edit	Copy	Delete		smoke_detector	Normal
Edit	Inline Edit	Copy	Delete		motion_detector	Normal
Edit	Inline Edit	Copy	Delete		sensor1	Detected
Edit	Inline Edit	Copy	Delete		sensor2	Normal

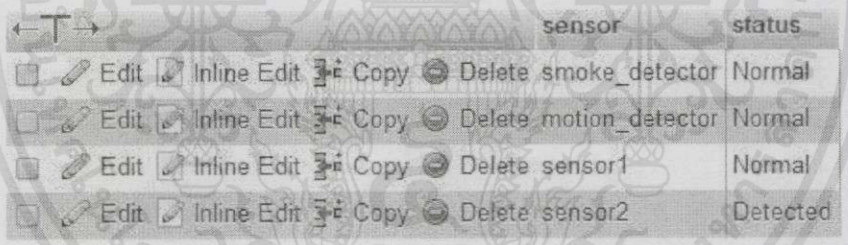
รูปที่ 4.41 ผลของ Database เมื่อเซ็นเซอร์แม่เหล็กตัวที่ 1 เกิดการตรวจจับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



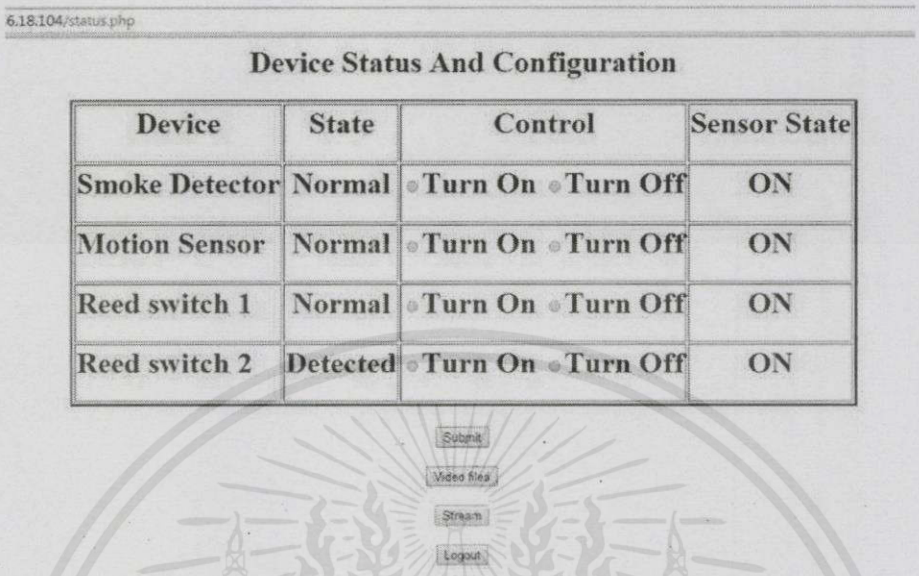
รูปที่ 4.42 หน้าเว็บเพจ เมื่อเซ็นเซอร์แม่เหล็กตัวที่ 1 เกิดการตรวจจับ

จากรูปที่ 4.41 และรูปที่ 4.42 จะเห็นว่าเมื่อเซ็นเซอร์แม่เหล็กตัวที่ 1 มีการตรวจจับ หน้าเว็บเพจจะแสดงสถานะของเซ็นเซอร์แม่เหล็กตัวที่ 1 ว่า Detected



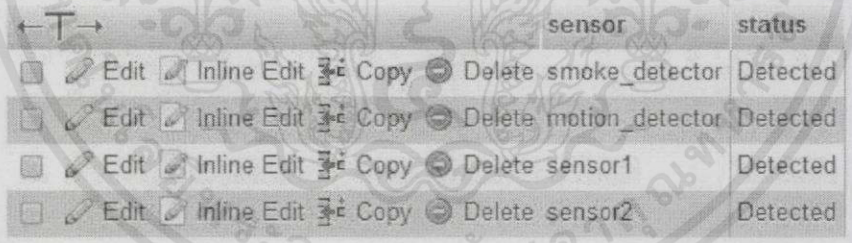
รูปที่ 4.43 ผลของ Database เมื่อเซ็นเซอร์แม่เหล็กตัวที่ 2 เกิดการตรวจจับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



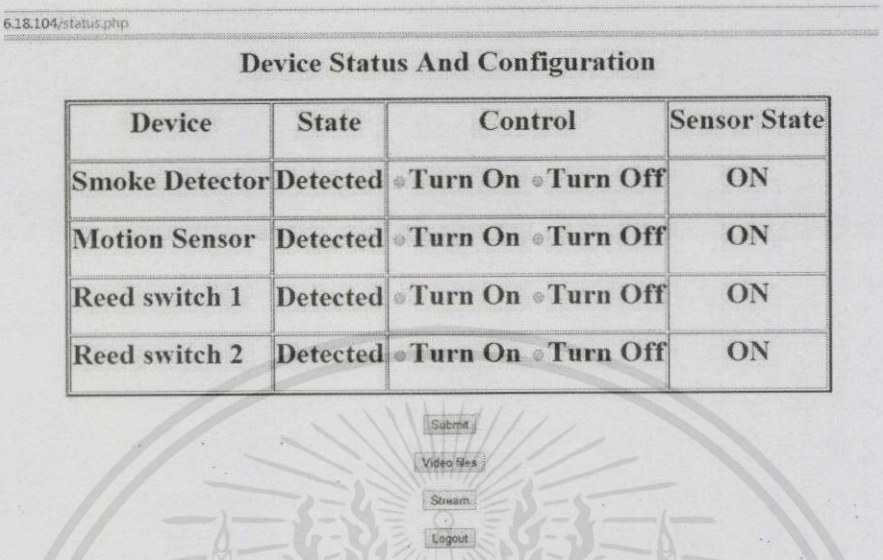
รูปที่ 4.44 หน้าเว็บเพจ เมื่อเซ็นเซอร์แม่เหล็กตัวที่ 2 เกิดการตรวจจับ

จากรูปที่ 4.43 และรูปที่ 4.44 จะเห็นว่าเมื่อเซ็นเซอร์แม่เหล็กตัวที่ 2 มีการตรวจจับ หน้าเว็บเพจจะแสดงสถานะของเซ็นเซอร์แม่เหล็กตัวที่ 2 ว่า Detected



รูปที่ 4.45 ผลของ Database เมื่ออุปกรณ์ทุกตัวเกิดการตรวจจับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



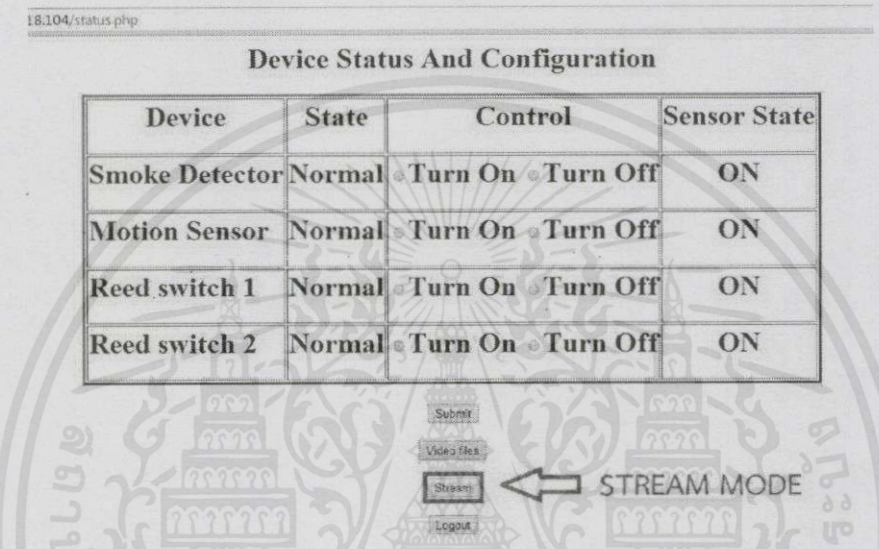
รูปที่ 4.46 หน้าเว็บเพจ เมื่ออุปกรณ์ทุกตัวเกิดการตรวจจับ

จากรูปที่ 4.45 และรูปที่ 4.46 จะเห็นว่าเมื่ออุปกรณ์ทุกตัวมีการตรวจจับ หน้าเว็บเพจจะแสดงสถานะของอุปกรณ์ทุกตัวว่า Detected

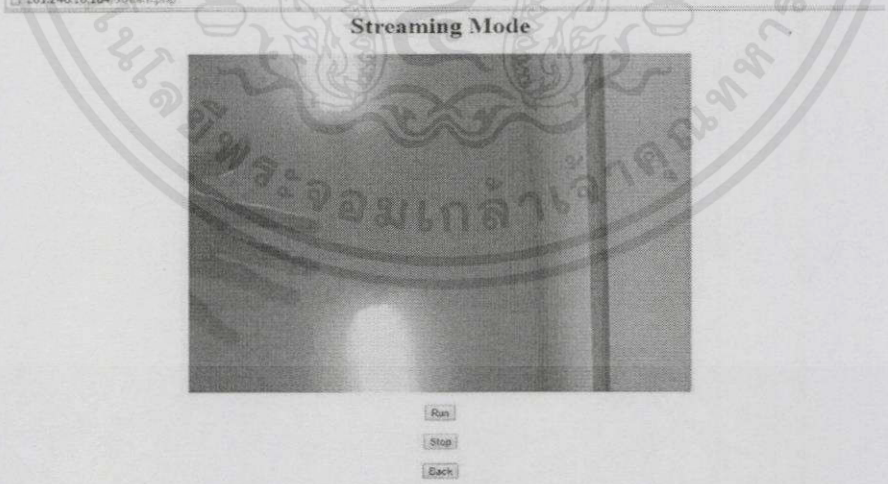
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7 ผลการทำงานในโหมดสตรีม

การทดลองการทำงานของโหมดสตรีม ทดลองดูสตรีมบนหน้าเว็บเพจ และแสดงผลบนหน้าเว็บเพจอีกหน้าหนึ่ง Delay ในการแสดงผลบนหน้าเว็บมีเวลาประมาณ 1 วินาที เกิดมาจากการที่ Raspberry Pi มีหน่วยการประมวลผลที่ค่อนข้างช้า คุณภาพของวีดีโอที่ถ่ายมีความคมชัด



รูปที่ 4.47 วิธีการเข้าใช้โหมดสตรีมผ่านหน้าเว็บเพจ



รูปที่ 4.48 หน้าเว็บเพจเมื่อเปิดใช้งานโหมดสตรีม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเปิดใช้งานหน้าโหมดสตรีมบนหน้าเว็บเพจจะทำการสั่งงานโปรแกรมสตรีมเพื่อทำการถ่ายทอดสดกล้องแล้วนำมาแสดงบนหน้าเว็บเพจทันที

4.8 หน้าเว็บเพจที่เก็บไฟล์วีดิโอ

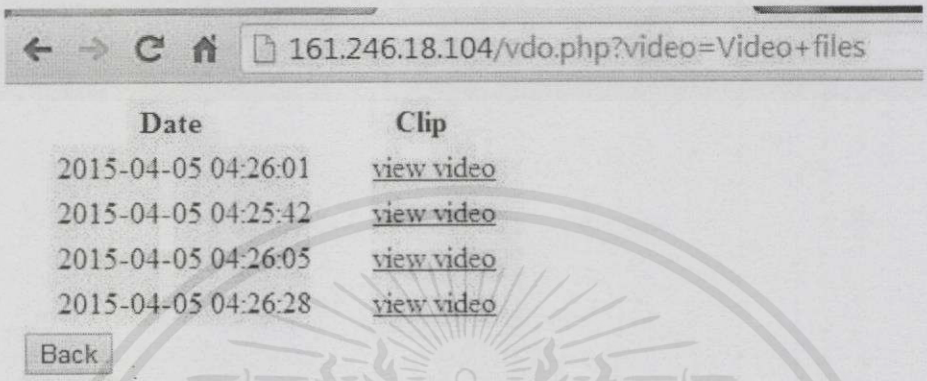
หน้าเว็บเพจที่เก็บไฟล์วีดีโอนี้จะทำหน้าที่เก็บไฟล์วีดิโอทั้งหมดที่เซ็นเซอร์ตรวจจับการเคลื่อนไหวได้ทำการบันทึกไว้ การบันทึกจะบันทึกโดย ใช้ข้อมูลของ วัน เดือน ปี และเวลา ในการสร้างไฟล์เพื่อบันทึกวีดิโอ จะทำให้ผู้ใช้งานรู้ว่า เวลาใดที่เซ็นเซอร์ตรวจจับการเคลื่อนไหวทำงาน

รูปที่ 4.49 หน้าเว็บเพจเมื่อไม่มีไฟล์วีดิโอ

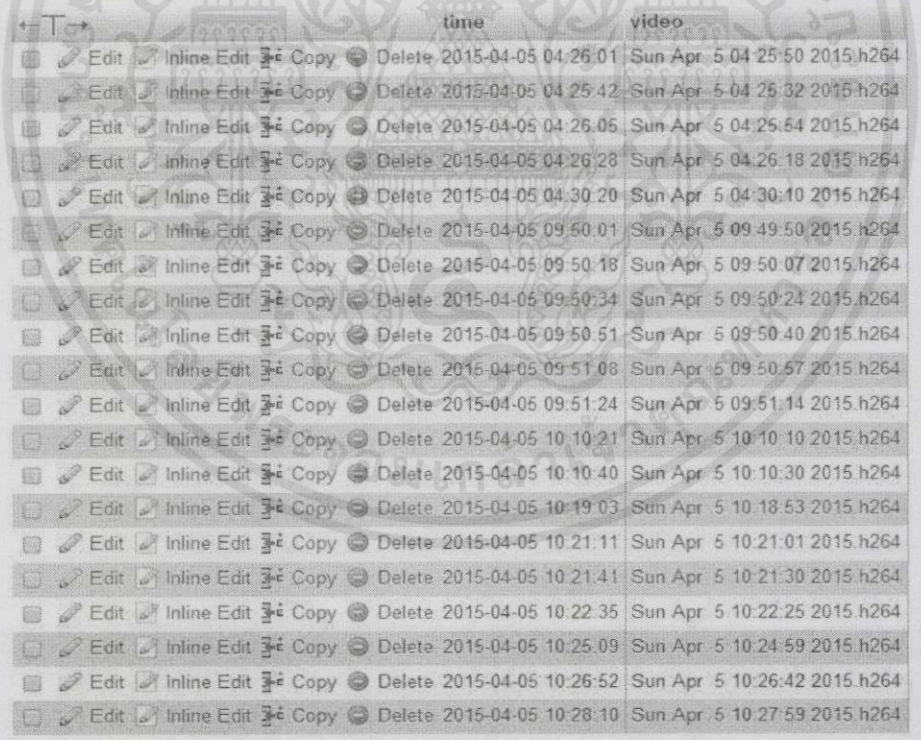
รูปที่ 4.50 ผลของ Database เมื่อมีไฟล์วีดิโอ โดยจะบันทึกวัน เวลาและเดือนด้วย

เอกสารนี้เป็นเอกสาร ... ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเปิดใช้งานหน้าโหมดสตรีมบนหน้าเว็บเพจจะทำการสั่งงานโปรแกรมสตรีมเพื่อทำการถ่ายทอดสดกล้องแล้วนำมาแสดงบนหน้าเว็บเพจทันที



รูปที่ 4.51 หน้าเว็บเพจแสดงวีดีโอบนหน้าเว็บพร้อมที่จะทำการดาวน์โหลด



รูปที่ 4.52 ผลของ Database เมื่อมีไฟล์วีดีโอ โดยจะบันทึกวัน เวลาและเดือนด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเปิดใช้งานหน้าโหมดสตรีมบนหน้าเว็บเพจจะทำการสั่งงานโปรแกรมสตรีมเพื่อทำการถ่ายทอดสดกล้องแล้วนำมาแสดงบนหน้าเว็บเพจทันที

4.9 การแจ้งเตือนโดยการส่ง SMS

การแจ้งเตือนนั้นถูกออกแบบมาเพื่อให้แจ้งเตือนผ่าน SMS โดยแต่ละข้อความจะบอกว่าการแจ้งเตือนเกิดจากการทำงานของเซ็นเซอร์ตัวใด ซึ่งการแจ้งเตือนผ่าน SMS นั้นแสดงได้ตามรูปที่ 4.53

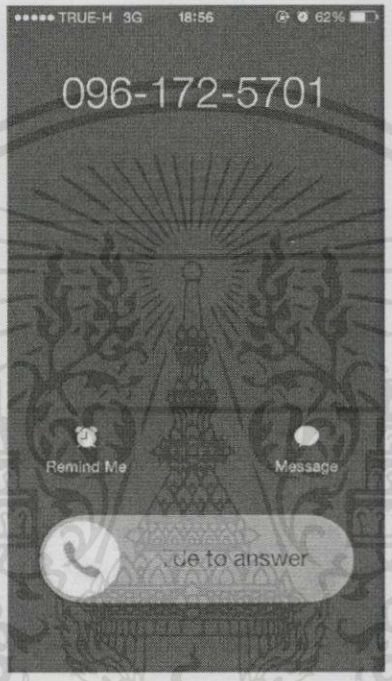
รูปที่ 4.53 หน้าโทรศัพท์มือถือเมื่อมีการแจ้งเตือนโดย SMS

จากรูปที่ 4.53 เป็นการแจ้งเตือนโดยส่ง SMS ไปที่มือถือของผู้ใช้งาน ในรูปเป็น SMS ที่ได้รับจากการทำงานของเซ็นเซอร์ตรวจจับการเคลื่อนไหว เซ็นเซอร์ตรวจจับควันและเซ็นเซอร์แม่เหล็กตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.10 การแจ้งเตือนโดยการโทรเข้าเบอร์ผู้ใช้งาน

การแจ้งเตือนในลักษณะนี้ถูกออกแบบมาเพื่อให้แจ้งเตือนโดยการโทรเข้าไปยังเบอร์ของผู้ใช้งานเมื่อมีการตรวจจับของเซ็นเซอร์ในทุกกรณีแสดงผลได้ตามรูปที่ 4.54



รูปที่ 4.54 แจ้งเตือนผ่านการโทรเข้าเบอร์ผู้ใช้งาน

จากรูปที่ 4.54 เป็นการแจ้งเตือนโดยส่ง SMS ไปที่มือถือของผู้ใช้งาน ในรูปเป็น SMS ที่ได้รับจากการทำงานของเซ็นเซอร์ตรวจจับการเคลื่อนไหว เซ็นเซอร์ตรวจจับควันและเซ็นเซอร์แม่เหล็กตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

5.1.1 การทำงานทางด้านอุปกรณ์เซ็นเซอร์

เซ็นเซอร์แม่เหล็กเป็นอุปกรณ์ที่สามารถใช้เป็นสวิชต์ไฟฟ้าได้ ในเวลาปกติจะมีแท่งแม่เหล็กติดตั้งอยู่ใกล้กับตัวอุปกรณ์รีดสวิชต์เสมอเพื่อให้วงจรของอุปกรณ์ทำงาน (ส่ง Logic 1 ให้ Raspberry Pi) แต่เมื่อมีการนำเอาแท่งเหล็กออกไปตำแหน่งอื่นที่ไม่ได้กำหนดไว้ตั้งแต่แรก จะทำให้วงจรอุปกรณ์รีดสวิชต์ไม่ทำงาน (ส่ง Logic 0 ให้ Raspberry Pi) แล้วจึงนำผลของลอจิคที่ได้ไปใช้งานในส่วนทางด้านของโปรแกรมแจ้งเตือน

โดยที่เซ็นเซอร์ตรวจจับควันเป็นอุปกรณ์ที่สามารถตรวจจับการรั่วไหลของแก๊สได้ โดยค่าที่ได้จะเป็นสัญญาณอนาล็อก มีลักษณะเป็นค่าแรงดันไฟฟ้าค่าหนึ่ง หลังจากนั้นจะนำเอาสัญญาณอนาล็อกที่ได้ไปแปลงเป็นสัญญาณสัญญาณดิจิตอลโดย IC CHIP MCP3002 ADC และทำการเขียนโปรแกรม ผลจากโปรแกรมจะทำให้สามารถอ่านค่าสัญญาณเป็นระดับตัวเลขได้ ระดับตัวเลขเป็นสิ่งสำคัญ สามารถนำไปใช้ในการกำหนดเงื่อนไขการทำงานของโปรแกรมการแจ้งเตือนได้เช่นเดียวกันกับ Logic

เซ็นเซอร์ตรวจจับการเคลื่อนไหวเป็นอุปกรณ์ตรวจพบสิ่งเคลื่อนไหวผ่านทางกล้อง Raspberry Pi Camera Module โดยการเขียนโปรแกรม การทำงานของโปรแกรมคือเมื่อมีการตรวจพบสิ่งเคลื่อนไหว กล้องจะทำการบันทึกวิดีโอเก็บไว้ แล้วไปสั่งงานในส่วนโปรแกรมแจ้งเตือน

5.1.2 การทำงานทางด้านโปรแกรมการแจ้งเตือน

การแจ้งเตือนจะใช้การส่ง SMS และการโทรออกไปยังมือถือผู้ใช้งาน การส่ง SMS และการโทรออก จะทำงานผ่านอุปกรณ์ Air Card ที่ติดตั้งไว้บน Raspberry Pi โดยจะนำเงื่อนไขจากเซ็นเซอร์ทั้งสามตัว มาเขียนลงในโปรแกรมสามารถสรุปการทำงานได้คือ

1) เซ็นเซอร์แม่เหล็กมีการแจ้งเตือนเมื่อแท่งแม่เหล็กอยู่ห่างจากรีดสวิชต์เกินกว่าระยะที่กำหนด

2) เซ็นเซอร์ตรวจจับควันจะมีการแจ้งเตือน เมื่อมีระดับควันเกิดขึ้นเกินโปรแกรมที่กำหนด

3) เซ็นเซอร์ตรวจจับการเคลื่อนไหวจะมีการแจ้งเตือนเมื่อมีสิ่งเคลื่อนไหวผ่านกล้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.3 การเปิด-ปิดการทำงานของอุปกรณ์เซ็นเซอร์และการตรวจสอบสถานะของอุปกรณ์ผ่านหน้าเว็บเพ็จ

ผู้ใช้สามารถเปิด-ปิดอุปกรณ์เซ็นเซอร์ได้ผ่านทางหน้าเว็บเพจ โดยออกแบบมาให้ง่ายต่อการใช้งาน นอกจากนี้ผู้ใช้ยังสามารถตรวจสอบสถานะการเปิด-ปิดของอุปกรณ์และสถานะการตรวจจับของอุปกรณ์ผ่านหน้าเว็บได้

5.1.4 การเปิดการใช้งานโหมดสตรีมในหน้าเว็บเพจ

ผู้ใช้สามารถเปิดใช้งานโหมดสตรีมเพื่อรับชมเหตุการณ์ในขณะนั้นได้ผ่านทางหน้าเว็บเพ็จ

5.1.5 การรับชมไฟล์วิดีโอที่บันทึกโดยเซ็นเซอร์ตรวจจับการเคลื่อนไหว

ผู้ใช้สามารถรับชมไฟล์วิดีโอที่บันทึกโดยเซ็นเซอร์ตรวจจับการเคลื่อนไหวในช่วงเวลาที่ผ่านมาได้ โดยสามารถรับชมได้โดยการเข้าไปเลือกในหน้าเว็บเพจที่จัดเก็บไฟล์วิดีโอไว้ ซึ่งภายในหน้าเว็บเพจนั้นจะมีการระบุวัน เดือน ปีและเวลาเอาไว้เพื่อสะดวกต้องผู้ใช้ในการค้นหาเพื่อรับชม

5.2 ข้อเสนอแนะ

5.2.1 ข้อเสนอแนะทางด้านอุปกรณ์เซ็นเซอร์

เซ็นเซอร์ตรวจจับควันเป็นอุปกรณ์ที่มีวงจรประกอบด้วย MQ-2 เป็นตัวทำปฏิกิริยากับควันหรือแก๊ส Variable Resistant เป็นตัวทำให้วงจรสามารถส่งสัญญาณ Output ออกมาเป็นแรงดันได้ IC CHIP MCP3002 ADC เป็นตัวที่แปลงสัญญาณ Analog ของ Output ที่เป็นสัญญาณไฟฟ้าให้เป็นสัญญาณ Digital ที่เป็นระดับตัวเลขได้ ในการใช้งานอุปกรณ์ยังมีความไม่เสถียรให้เห็นอยู่ คือค่า Variable Resistant จะมีการเปลี่ยนแปลงตลอดทำให้การวัดค่าระดับควันมีการคาดเคลื่อน

5.2.2 ข้อเสนอแนะทางด้านโปรแกรมแจ้งเตือน

จากข้อบกพร่องของการส่ง SMS และการโทรออกคือข้อจำกัดในด้านของเวลา และการส่ง SMS และโทรออกนั้นก็ขึ้นอยู่กับสัญญาณของผู้ให้บริการด้วย อาจจะเพิ่มทางเลือกในการแจ้งเตือนเข้าไปเช่น การแจ้งเตือนทางโปรแกรมไลน์(Line)หรือการแจ้งเตือนผ่านอีเมลล์เป็นต้น

5.2.3 ข้อเสนอแนะทางด้านเว็บเพจ

ในส่วนของทางด้านเว็บเพจนั้น ผู้ใช้สามารถเช็คสถานะของอุปกรณ์และสถานะของการตรวจจับได้ นอกจากนี้ผู้ใช้ยังสามารถควบคุมการเปิด-ปิดของอุปกรณ์ได้ ซึ่งตรงตาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขอบเขตที่ได้กล่าวไว้แล้ว แต่ในกลุ่มของผู้จัดทำคิดว่าควรจะมีคุณสมบัติเพิ่มเติมขึ้นมาจากขอบเขตซึ่งระบุได้เป็นหัวข้อดังนี้

- ควรเพิ่มโหมดการทำงาน อย่างเช่น โหมดตั้งเวลาเปิด-ปิด ซึ่งสามารถที่จะตั้งค่าได้ตามต้องการผ่านหน้าเว็บ หรือจะเป็นโหมดการทำงานที่มีรายละเอียดของการแจ้งเตือนหลายระดับ เช่นในส่วนของเซ็นเซอร์ตรวจจับควันมีการเตือนไปยังผู้ใช้งานเมื่อระดับควันอยู่ในระดับที่อันตราย และเมื่อในกรณีที่มีระดับควันนั้นอยู่ในระดับที่ปลอดภัยแล้วก็ให้แจ้งเตือนผ่านหน้าเว็บด้วย

- ควรมีการเพิ่มระบบสำรองพลังงานในกรณีที่ขาดแหล่งพลังงาน ไม่ว่าจะเกิดจากการที่ไฟดับหรืออุบัติเหตุใดๆก็ตาม

- เพิ่มระบบที่สามารถเซ็ตตั้งค่าการทำงานของเซ็นเซอร์ เช่นในเซ็นเซอร์ตรวจจับการเคลื่อนไหวสามารถปรับระดับอ้างอิงผ่านหน้าเว็บได้ ในส่วนของเซ็นเซอร์ตรวจจับควันสามารถปรับระดับของควันหรือก๊าซที่ต้องการให้ตรวจจับได้ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] ชาญชัย ศุภอรรถกร, และชิษณุพงศ์ ธัญญลักษณ์. "สร้างเว็บแอพพลิเคชั่น PHP+MySQL Ajax jQuery ฉบับสมบูรณ์." กรุงเทพฯ. : สำนักพิมพ์ Simplify, 2556.

[2] Magnus Lie Hetland. "Beginning Python from Novice to Professiona. New York",Apress Media LLC, 2004.

[3] http://picamera.readthedocs.org/

[4] http://www.raspberrypi.org/

[5] http://docs.opencv.org/

[6] http://wammu.eu/

[7] http://www.w3schools.com/

[8] http://www.raspberry-projects.com/pi/

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

โค้ดโปรแกรมการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Index.html (หน้า login)

```
<html>
<head>
      <title>Login</title>
</head>
<body bgcolor="#CCFFE5">
      <br><center><img src = "head.jpg"></center><br>
      <center><font color = #CD0000><h1>Security System Set By Raspberry
Pi</h1></font></center>
      <form method=post action="check_user.php">
            <center><font color = #CD0000>
            User Name:<input type="text" name="user"><p>
            Password :<input type="password" name="pass"><p>
            <input type="submit" name="Login" value="Login"> <input
type="reset" value="Cancle"></font></center>
      </form>
</body>
```

check_user.php (เช็ค username password)

```
<?php
      $user = $_POST['user'];
      $pass = $_POST['pass'];
      $conn = mysql_connect('','root','khunsuek') or die("Error".
mysql_error());
      $select_db = mysql_select_db('password_project',$conn) or
die("Error".mysql_error());

      $sql = "SELECT * FROM password WHERE username ='".$user."' AND
password ='".$pass."'";
      $result = mysql_query($sql);
      $rs = mysql_fetch_array($result);

      if(empty($rs))
      {

            echo "Username or Password  incorrect!";


      }
      else
      {
            $conn1 = mysql_connect('','root','khunsuek') or die("Error".
mysql_error());
            $select_db = mysql_select_db('password_project',$conn1) or
die("Error".mysql_error());
            $sql1="INSERT INTO history values('$rs[username]',NOW())";
            mysql_db_query('password_project',$sql1);

            header("Location: ./select.php");
      }
```

select.php (หน้าเลือกโหมดในการแจ้งเตือน)

```
<html>
      <title>select page</title>
      <form action="modealarm.php" method="get">
      <head><center><h1>Please select mode!</h1></center></head>
      <body bgcolor="#CCFFE5">
            <center><TD><h1><input type="radio" name="select"
value="dialcall">Dial call
                  <input type="radio" name="select" value="sms">SMS
                  <input type="radio" name="select"
value="not_alarm">Not alarm</TD></h1></center>
```

เอกสารนี้มีลิขสิทธิ์ สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      <center><input type="submit" value="Submit">
      </center></form>
      </body>
      <form action="index.html" method="get">
      <center><input name="Back" type="submit" value="Logout">
      </center></form>
      <form action="history.php" method="get">
      <center><input name="show" type="submit" value="History check">
      </center></form>
</html>
```

modealarm.php (ส่วนเชื่อมต่อโปรแกรมแจ้งเตือน)

```
<?php
      $mode = $_GET['select'];

      switch($mode)
            {
                  case "dialcall":
                              exec('bash -c "exec nohup setsid sudo python
/var/www/programe/alarm.py > /dev/null 2>&1 &"');
                              break;
                  case "sms":
                              exec('bash -c "exec nohup setsid sudo python
/var/www/programe/sms.py > /dev/null 2>&1 &"');
                              break;
                  case "not_alarm":
                              break;
            }
      header("Location: ./status.php")
?>
```

history_sensor.php (หน้าเก็บประวัติการตรวจจับของเซ็นเซอร์)

```
<?php
      $host = "localhost";
      $user = "root";
      $pass = "khunsuek";
      $db = "sensor status";
?>
<html>
      <title>History Sensor</title>
      <body bgcolor="#CCFFE5">
      <head><center><h1>History Sensor Detected</h1></center></head>
            <center><table border=3>
                  <TR>
                        <TD><h1><center>Smoke Detector</center></h1></TD>
                        <TD><h1><center>Motion Detector</center></h1></TD>
                        <TD><h1><center>Reed switch 1</center></h1></TD>
                        <TD><h1><center>Reed switch 2</center></h1></TD>
                  </TR>
                  <TR>
                        <TD><center>
                              <?php
                                    $con = new mysqli($host, $user, $pass,
$db);
                                    if($con->connect_error)
                                          {
                                                die("Connection
failed: " . $con->connect_error);
                                          }
                                    $sql1 = "SELECT * FROM smoke_detector";
                                    $result1 = $con->query($sql1);
                                    if ($result1->num_rows > 0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        {
                                            echo "<table>";
                                                while($row1 =
$result1->fetch_assoc())
                                                    {
                                                        echo
"<tr><td>".$row1["time"]."</td></tr>";
                                                    }
                                            echo "</table>";
                                        }
                                    else
                                        {
                                            echo "0 results";
                                        }
                                ?>
                                </center></TD>
                            <TD><center>
                                <?php
                                    $con = new mysqli($host, $user, $pass,
$db);
                                            if($con->connect_error)
                                                {
die("Connection failed: " . $con->connect_error);
                                                }
                                    $sql2 = "SELECT * FROM
motion_detector";
                                    $result2 = $con->query($sql2);
                                    if ($result2->num_rows > 0)
                                                {
                                            echo "<table>";
      while($row2 = $result2->fetch_assoc())
       {
            echo "<tr><td>".$row2["time"]."</td></tr>";
       }
                                            echo "</table>";
                                                }
                                    else
                                        {
                                            echo "0 result";
                                        }
                                ?>
                            </center></TD>
                            <TD><center>
                                <?php
                                    $con = new mysqli($host, $user, $pass,
$db);
                                            if($con->connect_error)
                                                {
die("Connection failed: " . $con->connect_error);
                                                }
                                            $sql3 = "SELECT * FROM
reed_sw1";
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        $result3 = $con-
>query($sql3);
                                        if ($result3->num_rows > 0)
                                                {
                                                        echo
"<table>";
while($row3 = $result3->fetch_assoc())
{
echo "<tr><td>".$row3["time"]."</td></tr>";
}
                                                        echo
"</table>";
                                                }
                                        else
                                                {
                                                        echo "0
result";
                                                }
                        ?></center></TD>
                    <TD><center>
                        <?php
                                $con = new mysqli($host, $user, $pass,
$db);
                                        if($con->connect_error)
                                                {
die("Connection failed: " . $con->connect_error);
                                                }
                                        $sql4 = "SELECT * FROM
reed_sw1";
                                        $result4 = $con-
>query($sql4);
                                        if ($result3->num_rows > 0)
                                                {
                                                        echo
"<table>";
while($row4 = $result4->fetch_assoc())
{
echo "<tr><td>".$row4["time"]."</td></tr>";
}
                                                        echo
"</table>";
                                                }
                                        else
                                                {
                                                        echo "0
result";
                                                }
                        ?></center></TD>
                </TR>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            </center></table><br>
      <form action="history.php" method="get"><center>
            <input name="Back" type="submit" class="confirm" value="Back">
      </center></form>
      </body>
</html>
```

history.php (หน้าเก็บประวัติการเข้าใช้งาน)

```
<?php
      $servername = "localhost";
      $username = "root";
      $password = "khunsuek";
      $dbname = "password_project";

      $conn = new mysqli($servername, $username, $password, $dbname);

      if ($conn->connect_error)
            {
                  die("Connection failed: " . $conn->connect_error);
            }

      $sql = "SELECT * FROM history";
      $result = $conn->query($sql);
      if ($result->num_rows > 0)
            {
                  echo "<table><tr><th>Username </th><th>Login
Time</th></tr>";

                        while($row = $result->fetch_assoc())
                        {
                              echo
"<tr><td>".$row["user"]."</td><td>".$row["logintime"]."</td></tr>";
                        }
                  echo "</table>";
            }
      else
            {
                  echo "0 results";
            }
      $conn->close();
?>
<html>
      <body bgcolor="#CCFFE5">
      <form action="select.php" method="get">
            <input name="Back" type="submit" class="confirm" value="Back">
      </form>
      <form action="history_sensor.php" method="get">
            <input name="Check" type="submit" class="confirm" value="Check
History Sensor page">
      </form>
      </body>
</html>
```

status.php (หน้าเว็บควบคุมและแสดงสถานะของเซ็นเซอร์)

```
<?php
      $host = "localhost";
      $user = "root";
      $pass = "khunsuek";
      $db = "sensor_status";

      $connect = mysql_connect($host, $user, $pass) or die(mysql_error());
      mysql_select_db($db, $connect) or die(mysql_error());
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      $getdata1 = mysql_query("SELECT * FROM sensor_status WHERE sensor =
'smoke_detector'") or die(mysql_error());
      $slot1 = mysql_fetch_array($getdata1, MYSQL_ASSOC);
      $status1 = $slot1["status"];

      $getdata2 = mysql_query("SELECT * FROM sensor_status WHERE sensor =
'motion_detector'") or die(mysql_error());
      $slot2 = mysql_fetch_array($getdata2, MYSQL_ASSOC);
      $status2 = $slot2["status"];

      $getdata3 = mysql_query("SELECT * FROM sensor_status WHERE sensor =
'sensor1'") or die(mysql_error());
      $slot3 = mysql_fetch_array($getdata3, MYSQL_ASSOC);
      $status3 = $slot3["status"];

      $getdata4 = mysql_query("SELECT * FROM sensor_status WHERE sensor =
'sensor2'") or die(mysql_error());
      $slot4 = mysql_fetch_array($getdata4, MYSQL_ASSOC);
      $status4 = $slot4["status"];

      $getdata5 = mysql_query("SELECT * FROM device WHERE device =
'smoke_detector'") or die(mysql_error());
        $slot5 = mysql_fetch_array($getdata5, MYSQL_ASSOC);
        $status5 = $slot5["status"];

        $getdata6 = mysql_query("SELECT * FROM device WHERE device =
'motion_detector'") or die(mysql_error());
        $slot6 = mysql_fetch_array($getdata6, MYSQL_ASSOC);
        $status6 = $slot6["status"];

        $getdata7 = mysql_query("SELECT * FROM device WHERE device =
'reed1'") or die(mysql_error());
        $slot7 = mysql_fetch_array($getdata7, MYSQL_ASSOC);
        $status7 = $slot7["status"];

        $getdata8 = mysql_query("SELECT * FROM device WHERE device =
'reed2'") or die(mysql_error());
        $slot8 = mysql_fetch_array($getdata8, MYSQL_ASSOC);
        $status8 = $slot8["status"];

?>

<html>
      <META HTTP-EQUIV='Refresh' CONTENT = '5'>

      <title>Configure Page</title>
      <form action="exec.php" method="get">
            <head><center><h1>Device Status And
Configuration</h1></center></head>
            <body bgcolor="#CCFFE5">
                  <center><table border=3>
                        <TR>
                              <TD><h1><center> Device </center></h1></TD>
                              <TD><h1><center> State </center></h1></TD>
                              <TD><h1><center> Control </center></h1></TD>
                              <TD><h1><center> Sensor State
</center></h1></TD>
                        </TR>
                        <TR>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                <TD><h1><center> Smoke Detector
</center></h1></TD>
                                <TD><h1><center> <?php echo "$status1"; ?>
</center></h1></TD>
                                <TD><h1><input type="radio" name="smoke"
value="on">Turn On
                                        <input type="radio" name="smoke"
value="off">Turn Off</h1></TD>
                                <TD><h1><center> <?php echo "$status5"; ?>
</center></h1></TD>
                        </TR>
                        <TR>
                                <TD><h1><center> Motion Sensor
</center></h1></TD>
                                <TD><h1><center> <?php echo "$status2"; ?>
</center></h1></TD>
                                <TD><h1><input type="radio" name="motion"
value="on">Turn On
                                        <input type="radio" name="motion"
value="off">Turn Off</h1></TD>
                                <TD><h1><center> <?php echo "$status6"; ?>
</center></h1></TD>
                        </TR>
                        <TR>
                                <TD><h1><center> Reed switch 1
</center></h1></TD>
                                <TD><h1><center> <?php echo "$status3"; ?>
</center></h1></TD>
                                <TD><h1><input type="radio" name="reed1"
value="on">Turn On
                                        <input type="radio" name="reed1"
value="off">Turn Off</h1></TD>
                                <TD><h1><center> <?php echo "$status7"; ?>
</center></h1></TD>
                        </TR>
                        <TR>
                                <TD><h1><center> Reed switch 2
</center></h1></TD>
                                <TD><h1><center> <?php echo "$status4"; ?>
</center></h1></TD>
                                <TD><h1><input type="radio" name="reed2"
value="on">Turn On
                                        <input type="radio" name="reed2"
value="off">Turn Off</h1></TD>
                                <TD><h1><center> <?php echo "$status8"; ?>
</center></h1></TD>
                        </TR>
                </center></table><br>
        <center><input type="submit" value="Submit">
        </form></center><br>

        <center><form action="stream.php" method="get">
        <input name="stream" type="submit" value="Stream page">
        </form></center>

        <center><form action="vdo.php" method="get">
        <input name="video" type="submit" value="Video files">
        </form></center>

        <center><form action="logout_stopprograme.php" method="get">
        <input name="Logout" type="submit" value="Logout">
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            </form></center>

            </body>
</html>
```

exec.php (ส่วนเชื่อมต่อโปรแกรมควบคุมอุปกรณ์ผ่านหน้าเว็บ)

```
<?php
            $smoke = $_GET['smoke'];
            $motion = $_GET['motion'];
            $reed1 = $_GET['reed1'];
            $reed2 = $_GET['reed2'];

            $host = "localhost";
            $user = "root";
            $pass = "khunsuek";
            $db = "sensor status";

            $connect = mysql_connect($host, $user, $pass) or
die(mysql_error());
            mysql_select_db($db, $connect) or die(mysql_error());

            switch($smoke)
                {
                        case "on":
                                    exec('bash -c "exec nohup setsid sudo
python /var/www/programe/smoke_dt.py > /dev/null 2>&1 &"');
                                    $sql = mysql_query("UPDATE device SET
status='ON' WHERE device='smoke_detector'") or die(mysql_error());
                                    break;

                        case "off":
                                    exec('sudo pkill -f
/var/www/programe/smoke_dt.py');
                                    $sql = mysql_query("UPDATE device SET
status='OFF' WHERE device='smoke_detector'") or die(mysql_error());
                                    $sql = mysql_query("UPDATE
sensor_status SET status='-' WHERE sensor='smoke_detector'") or
die(mysql_error());
                                    break;
                }
            switch($motion)
                {
                        case "on":
                                    exec('bash -c "exec nohup setsid sudo
python /var/www/programe/motion_dt.py > /dev/null 2>&1 &"');
                                    $sql = mysql_query("UPDATE device SET
status='ON' WHERE device='motion_detector'") or die(mysql_error());
                                    break;
                        case "off":
                                    exec('sudo pkill -f
/var/www/programe/motion_dt.py');
                                    $sql = mysql_query("UPDATE device SET
status='OFF' WHERE device='motion_detector'") or die(mysql_error());
                                    $sql = mysql_query("UPDATE
sensor_status SET status='-' WHERE sensor='motion_detector'") or
die(mysql_error());
                                    break;
                }
            switch($reed1)
                {
                        case "on":
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                    exec('bash -c "exec nohup setsid sudo
python /var/www/programe/reed_sw1.py > /dev/null 2>&1 &"');
                                    $sql = mysql_query("UPDATE device SET
status='ON' WHERE device='reed1'") or die(mysql_error());
                                    break;
                        case "off":
                                    exec('sudo pkill -f
/var/www/programe/reed_sw1.py');
                                    $sql = mysql_query("UPDATE device SET
status='OFF' WHERE device='reed1'") or die(mysql_error());
                                    $sql = mysql_query("UPDATE
sensor_status SET status='-' WHERE sensor='sensor1'") or
die(mysql_error());
                                    break;
                  }
            switch($reed2)
                  {
                        case "on":
                                    exec('bash -c "exec nohup setsid sudo
python /var/www/programe/reed_sw2.py > /dev/null 2>&1 &"');
                                    $sql = mysql_query("UPDATE device SET
status='ON' WHERE device='reed2'") or die(mysql_error());
                                    break;
                        case "off":
                                    exec('sudo pkill -f
/var/www/programe/reed_sw2.py');
                                    $sql = mysql_query("UPDATE device SET
status='OFF' WHERE device='reed2'") or die(mysql_error());
                                    $sql = mysql_query("UPDATE
sensor_status SET status='-' WHERE sensor='sensor2'") or
die(mysql_error());
                                    break;
                  }
            header("Location: ./status.php")

?>
```

stream.php (หน้าโหมด stream)

```
<html>
      <title>Streaming mode page</title>
      <head><center><h1>Streaming mode <meta http-equiv="refresh"
content="5"> </h1></center></head>
      <body bgcolor="#CCFFE5">
            <center><table border=3>
                  <TR>
                        <TD><h1><center> Streaming mode </center></h1></TD>
                  </TR>
                  <TR>
                        <TD><h1><center><img src="http://raspberrpy pi
IP:9000/?action=stream"/img></center></h1></TD>
                  </TR>
                  <TR>
                        <TD><h1><center><form action="run_stream.php"
method="get"><input name="Run" type="submit" value="Run"> <input
name="Stop" type="submit" value="Stop"></form></center></h1></TD>
                  </TR>
            </table><br>
      <center><form action="status.php" method="get"><input type="submit"
value="Back"></form></center>
      </body>
</html>
```

เอกสารนี้มีลิขสิทธิ์ ห้ามนำไปใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

run_stream.php (ส่วนเชื่อมต่อควบคุมในโหมด stream)

```
<?php
      $host = "localhost";
      $user = "root";
      $pass = "khunsuek";
      $db = "sensor status";
      $connect = mysql_connect($host, $user, $pass) or die(mysql_error());
      mysql_select_db($db, $connect) or die(mysql_error());
      $run = $_GET['Run'];
      $stop = $_GET['Stop'];
      switch($run)
            {
                  case "Run":
                              exec('sudo pkill -f
/var/www/programe/motion_dt.py');
                              $sql = mysql_query("UPDATE device SET
status='OFF' WHERE device='motion_detector'") or die(mysql_error());
                              $sql = mysql_query("UPDATE sensor_status SET
status='-' WHERE sensor='motion_detector'") or die(mysql_error());
                              exec('bash -c "exec nohup setsid sudo python
/var/www/programe/start_stream.py > /dev/null 2>&1 &" ');
                              break;
            }
      switch($stop)
            {
                  case "Stop":
                              exec('bash -c "exec nohup setsid sudo python
/var/www/programe/stop_stream.py > /dev/null 2>&1 &" ');
                              break;
            }
      header("Location: ./stream.php")
?>
```

vdo.php (หน้าเก็บไฟล์วีดีโอ)

```
<?php
      $host = "localhost";
      $user = "root";
      $pass = "khunsuek";
      $db = "sensor status";

      $connect = new mysqli($host, $user, $pass, $db);

      if ($connect->connect_error)
      {
    die("Connection failed: " . $connect->connect_error);
      }

      $sql = "SELECT * FROM video";
      $result = $connect->query($sql);

      if ($result->num_rows > 0)
      {
    echo
"<table><tr><th>          
;Date          </th><th>&n
bsp;         Clip  &n
bsp;       </th></tr>";

    while($row = $result->fetch_assoc())
      {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        echo
"<tr><td>    ".$row["time"]."</td><td>   
;    <a href=\"video/".$row['video']."\">view
video</a></td></tr>";
    }
    echo "</table>";
      }
      else
      {
    echo "0 results";
      }
      $connect->close()
?>
<html>
      <body bgcolor="#CCFFE5">
 <form action="status.php" method="get">
            <input name="Back" type="submit" class="confirm" value="Back">
            </form>
      </body>
</html>
```

logout_stopprograme.php (หน้าเชื่อมต่อปุ่ม logout กับโปรแกรมหยุดการแจ้งเตือน)

```
<?php
            $logout = $_GET['Logout'];


            switch($logout)
                  {
                        case "Logout":
                                    exec('sudo pkill -f
/var/www/programe/alarm.py');
                                    exec('sudo pkill -f
/var/www/programe/sms.py');
                                      break;
                  }
            header("Location: ./index.html")
?>
```

smoke_dt.py (โปรแกรม Smoke Detector)

```
import botbook_mcp3002 as mcp
import time
import commands
import threading
import MySQLdb
import RPi.GPIO as GPIO

GPIO.setmode(GPIO.BCM)
GPIO.setup(22, GPIO.OUT)

def readSmokeLevel():
     global smokeLevel
     smokeLevel = mcp.readAnalog()

def smoke():
      def process():
            while True:
                        readSmokeLevel()
                        if smokeLevel < 120:
                        GPIO.output(22, False)
                        print("Current smoke level is %i" %smokeLevel)
                        db = MySQLdb.connect("localhost", "root",
"khunsuek", "sensor status")
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        print("login db")
                        cur = db.cursor()
                        cur.execute("UPDATE sensor_status SET
status='Normal' WHERE sensor='smoke_detector'")
                        db.commit()
                        cur.close()
                        db.close()
                        print("close db")
                        time.sleep(1)
                        elif smokeLevel > 120 and smokeLevel < 125:
                        GPIO.output(22, True)
                        print("Current smoke level is %i" %smokeLevel)
                        print("Smoke detected")
                        db = MySQLdb.connect("localhost", "root",
"khunsuek", "sensor status")
                        print("login db")
                        cur = db.cursor()
                        cur.execute("UPDATE sensor_status SET
status='Detected' WHERE sensor='smoke_detector'")
                        db.commit()
                        cur.close()
                        db.close()
                        print("close db")
                        time.sleep(0.1)
                    elif smokeLevel > 125:
                        print("Current smoke level is %i" %smokeLevel)
                        print("Smoke detected")
                            db = MySQLdb.connect("localhost", "root",
"khunsuek", "sensor status")
                            print("login db")
                            cur = db.cursor()
                            cur.execute("UPDATE sensor_status SET
status='Detected' WHERE sensor='smoke_detector'")
                            db.commit()
                            cur.close()
                            db.close()
                        GPIO.output(22, True)
                        time.sleep(0.1)

    if __name__ == "__main__":
        process()

threading.Thread(target=smoke).start()

def check():
    def process():
        while True:
            if smokeLevel > 120:
                            db = MySQLdb.connect("localhost", "root",
"khunsuek", "sensor status")
                            print("login db")
                            cur = db.cursor()
                            cur.execute("INSERT INTO smoke_detector
values(NOW())")
                            db.commit()
                            cur.close()
                            db.close()
                            print("close db")
                            time.sleep(10)
    if __name__ == "__main__":
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            process()

threading.Thread(target=check).start()
```

motion_dt.py (โปรแกรม Motion Detector)

```
import time
import picamera
import cv2
import numpy as np
import commands
import io
import threading
import MySQLdb
import botbook_mcp3002 as mcp

class MotionDetection:
    def __init__(self):
        self.__image0 = None
        self.__image1 = None
        self.__image2 = None

        self._MOTION_LEVEL = 500000
        self._THRESHOLD = 35

    def _updateImage(self, image):
        self.__image2 = self.__image1
        self.__image1 = self.__image0
        self.__image0 = image

    def _ready(self):
        return self.__image0 != None and self.__image1 != None and
self.__image2 != None

    def _getMotion(self):
        if not self._ready():
            return None

        d1 = cv2.absdiff(self.__image1, self.__image0)
        d2 = cv2.absdiff(self.__image2, self.__image0)
        result = cv2.bitwise_and(d1, d2)

        (value, result) = cv2.threshold(result, self._THRESHOLD, 255,
cv2.THRESH_BINARY)

        scalar = cv2.sumElems(result)

        print " - scalar:", scalar[0]
        return scalar

    def detectMotion(self, image):
        self._updateImage(image)

        motion = self._getMotion()
        if motion and motion[0] > self._MOTION_LEVEL:
            return True
        return False

def motion():
    def process():
        print "Initializing camera..."
        with picamera.PiCamera() as camera:
```

เอกสารนี้มีลิขสิทธิ์ที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            print "Setting focus and light level on camera..."
            time.sleep(1)

            print "Initializing the CameraDetection..."
            detection = MotionDetection()

            count = 0

            while count <= 0:
                print "Capture picture..."
                db = MySQLdb.connect("localhost", "root", "khunsuek",
"sensor status")
                print("login db")
                cur = db.cursor()
                cur.execute("UPDATE sensor_status SET status='Normal' WHERE
sensor='motion_detector'")
                db.commit()
                cur.close()
                db.close()
                print("close db")
                stream = io.BytesIO()
                camera.capture(stream, format='jpeg')

                data = np.fromstring(stream.getvalue(), dtype=np.uint8)
                image = cv2.imdecode(data, 1)



                if detection.detectMotion(image):
                    db = MySQLdb.connect("localhost", "root", "khunsuek",
"sensor status")
                    print("login db")
                    cur = db.cursor()
                    cur.execute("UPDATE sensor_status SET status='Detected'
WHERE sensor='motion_detector'")
                cur.execute("INSERT INTO motion_detector values(NOW())")
                db.commit()
                    cur.close()
                    db.close()
                    print("close db")
                    tstmp = time.ctime(time.time())
                    print "start record"
                    camera.resolution = (640, 480)
                    camera.start_recording('/var/www/video/%s.h264' %
tstmp)
                    camera.wait_recording(10)
                    camera.stop_recording()
                    x = "%s.h264" % tstmp
                    print "record stop"
                    db = MySQLdb.connect("localhost", "root", "khunsuek",
"sensor status")
                    print("login db")
                    cur = db.cursor()
                    cur.execute("""INSERT INTO video
values(NOW(),%s)""",(x))
                    db.commit()
                    cur.close()
                    db.close()
                    print("close db")
                    count += 1
                    time.sleep(1)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if __name__== "__main__":
        process()

    def process2():

        print "Initializing camera..."
        with picamera.PiCamera() as camera:
            print "Setting focus and light level on camera..."
            time.sleep(1)

            print "Initializing the CameraDetection..."
            detection = MotionDetection()

            count = 0

            while count <= 0:
                print "Capture picture..."
                db = MySQLdb.connect("localhost", "root", "khunsuek",
"sensor status")
                print("login db")
                cur = db.cursor()
                cur.execute("UPDATE sensor_status SET status='Normal' WHERE
sensor='motion_detector'")
                db.commit()
                cur.close()
                db.close()
                print("close db")

                stream = io.BytesIO()
                camera.capture(stream, format='jpeg')

                data = np.fromstring(stream.getvalue(), dtype=np.uint8)
                image = cv2.imdecode(data, 1)


                if detection.detectMotion(image):
                    db = MySQLdb.connect("localhost", "root", "khunsuek",
"sensor status")
                    print("login db")
                    cur = db.cursor()
                    cur.execute("UPDATE sensor_status SET status='Detected'
WHERE sensor='motion_detector'")
                cur.execute("INSERT INTO motion_detector values(NOW())")
                    db.commit()
                    cur.close()
                    db.close()
                    print("close db")
                    tstmp = time.ctime(time.time())
                    print "start record"
                    camera.resolution = (640, 480)
                    camera.start_recording('/var/www/video/%s.h264' %
tstmp)
                    camera.wait_recording(10)
                    camera.stop_recording()
                    print "record stop"
                    x = "%s.h264" % tstmp
                    db = MySQLdb.connect("localhost", "root", "khunsuek",
"sensor status")
                    print("login db")
                    cur = db.cursor()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    cur.execute ("""INSERT INTO video
values(NOW(),%s)""",(x))
                    db.commit()
                    cur.close()
                    db.close()
                    print("close db")
                    count += 1
                    time.sleep(1)

    if __name__== "__main__":

        global ch
        ch = 0
        while True:
            if ch == 0:
                process()
                ch += 1
            elif ch == 1:
                process2()
                ch -= 1

threading.Thread(target=motion).start()
```

reed_sw1.py (โปรแกรม Reed Switch ตัวที่ 1)

```
import RPi.GPIO as GPIO
import time
import MySQLdb
import threading

GPIO.setmode(GPIO.BCM)

GPIO.setup(24, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(4, GPIO.OUT)


def reed_sw1():
      def process():
            print "sensor1 Ready"
            while(True):
                  if (GPIO.input(24) == 1):
                              db = MySQLdb.connect("localhost", "root",
"khunsuek", "sensor status")
                              print("login db")
                              cur = db.cursor()
                              cur.execute("UPDATE sensor_status SET
status='Normal' WHERE sensor='sensor1'")
                              db.commit()
                              cur.close()
                              db.close()
                              print("close db")
                              time.sleep(1)
                  elif (GPIO.input(24) == 0):
                              print "Sensor1 Start Connect"
                              db = MySQLdb.connect("localhost", "root",
"khunsuek", "sensor status")
                              print("login db")
                              cur = db.cursor()
                              cur.execute("UPDATE sensor_status SET
status='Detected' WHERE sensor='sensor1'")
                              db.commit()
                              cur.close()
                              db.close()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        print("close db")
                    time.sleep(1)

    if __name__ == "__main__":
        process()
threading.Thread(target=reed_sw1).start()

def alarm():
    def process():
        while(True):
            if(GPIO.input(24) == 0):
                GPIO.output(4,True)
                time.sleep(1)
            else:
                GPIO.output(4,False)

    if __name__ == "__main__":
        process()

threading.Thread(target=alarm).start()

def check():
    def process():
        while (True):
            if(GPIO.input(24) == 0):
                db = MySQLdb.connect("localhost", "root",
"khunsuek", "sensor status")
                print("login db")
                cur = db.cursor()
                cur.execute("INSERT INTO reed_sw1
values(NOW())")
                db.commit()
                cur.close()
                db.close()
                print("close db")
                time.sleep(15)
            else:
                print("condition invalid")

    if __name__ == "__main__":
        process()

threading.Thread(target=check).start()
```

reed_sw2.py (โปรแกรม Reed Switch ตัวที่ 2)

```
import RPi.GPIO as GPIO
import time
import threading
import MySQLdb

GPIO.setmode(GPIO.BCM)

GPIO.setup(23, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(27, GPIO.OUT)

def reed_sw2():
    def process():
        print "sensor2 Ready"
        while(True):
            if (GPIO.input(23) == 1):
                db = MySQLdb.connect("localhost", "root",
"khunsuek", "sensor status")
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                print("login db")
                                cur = db.cursor()
                                cur.execute("UPDATE sensor_status SET
status='Normal' WHERE sensor='sensor2'")
                                db.commit()
                                cur.close()
                                db.close()
                          print("close db")
                                time.sleep(1)
                    elif (GPIO.input(23) == 0):
                                print "Sensor2 Start Connect"
                                db = MySQLdb.connect("localhost", "root",
"khunsuek", "sensor status")
                                print("login db")
                                cur = db.cursor()
                                cur.execute("UPDATE sensor_status SET
status='Detected' WHERE sensor='sensor2'")
                                db.commit()
                                cur.close()
                                db.close()
                                print("close db")
                          time.sleep(1)

      if __name__ == "__main__":
            process()

threading.Thread(target=reed_sw2).start()

def alarm():
      def process():
            while(True):
                  if(GPIO.input(23) == 0):
                        GPIO.output(27, True)
                        time.sleep(1)
                  else :
                        GPIO.output(27, False)
      if __name__ == "__main__" :
            process()

threading.Thread(target=alarm).start()

def check():
      def process():
            while(True):
                  if(GPIO.input(23) == 0):
                        db = MySQLdb.connect("localhost", "root",
"khunsuek", "sensor status")
                        print("login db")
                              cur = db.cursor()
                              cur.execute("INSERT INTO reed_sw2
values(NOW())")
                              db.commit()
                              cur.close()
                              db.close()
                        print("close db")
                        time.sleep(15)
                  else:
                        print("condition invalid")

      if __name__ == "__main__":
            process()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
threading.Thread(target=check).start()
```

sms.py (โปรแกรมแจ้งเตือนผ่าน sms)

```
import time
import MySQLdb
import RPi.GPIO as GPIO
import botbook_mcp3002 as mcp
import commands

GPIO.setmode(GPIO.BCM)
GPIO.setup(23, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(24, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

db = MySQLdb.connect("localhost", "root", "khunsuek", "sensor status")
cur = db.cursor()
cur.execute("SELECT status FROM sensor_status WHERE sensor =
'motion_detector'")
result = cur.fetchall()
for i in result:
      a = i[0]

def readSmokeLevel():
      global smokeLevel
      smokeLevel = mcp.readAnalog()

def process():
      print "process start"
      while True:
            readSmokeLevel()
            print "in loop"
            time.sleep(0.5)
            if smokeLevel < 120 and (a == 'Normal' or a == '-') and
(GPIO.input(24) == 1) and (GPIO.input(23) == 0):
                  print "nnnd"
                  commandString = 'sudo gammu sendsms TEXT +66817097685 -
textutf8 "Reed switch1 Detected!"'
                  print commands.getoutput(commandString)
                  print "hang off"
                  time.sleep(0.5)
            elif smokeLevel < 120 and (a == 'Normal' or a == '-') and
(GPIO.input(24) == 0) and (GPIO.input(23) == 1):
                  print "nndn"
                  commandString = 'sudo gammu sendsms TEXT +66817097685 -
textutf8 "Reed switch2 Detected!"'
                        print commands.getoutput(commandString)
                  print "hang off"
                        time.sleep(0.5)
            elif smokeLevel < 120 and (a == 'Normal' or a == '-') and
(GPIO.input(24) == 0) and (GPIO.input(23) == 0):
                        print "nndd"
                        commandString = 'sudo gammu sendsms TEXT
+66817097685 -textutf8 "Reed switch1 and Reed switch2 Detected!"'
                  print commands.getoutput(commandString)
                  print "hang off"
                  time.sleep(0.5)
            elif smokeLevel < 120 and a == 'Detected' and (GPIO.input(24)
== 1) and (GPIO.input(23) == 1):
                        print "ndnn"
                        commandString = 'sudo gammu sendsms TEXT
+66817097685 -textutf8 "Motion Detector Detected!"'
                        print commands.getoutput(commandString)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                print "hang off"
                        time.sleep(0.5)
            elif smokeLevel < 120 and a == 'Detected' and (GPIO.input(24)
== 1) and (GPIO.input(23) == 0):
                print "ndnd"
                        commandString = 'sudo gammu sendsms TEXT
+66817097685 -textutf8 "Motion Detector and Reed switch1 Detected!"'
                        print commands.getoutput(commandString)
                        print "hang off"
                time.sleep(0.5)
            elif smokeLevel < 120 and a == 'Detected' and (GPIO.input(24)
== 0) and (GPIO.input(23) == 1):
                        print "nddn"
                        commandString = 'sudo gammu sendsms TEXT
+66817097685 -textutf8 "Motion Detector and Reed switch2 Detected!"'
                        print commands.getoutput(commandString)
                        print "hang off"
                        time.sleep(0.5)
            elif smokeLevel < 120 and a == 'Detected' and (GPIO.input(24)
== 0) and (GPIO.input(23) == 0):
                        print "nddd"
                        commandString = 'sudo gammu sendsms TEXT
+66817097685 -textutf8 "Motion Detector,Reed switch1 and Reed switch2
Detected!"'
                        print commands.getoutput(commandString)
                        print "hang off"
                        time.sleep(0.5)
            elif smokeLevel > 120 and smokeLevel < 125 and (a == 'Normal'
or a == '-') and (GPIO.input(24) == 1) and (GPIO.input(23) == 1):
                        print "dnnn"
                        commandString = 'sudo gammu sendsms TEXT
+66817097685 -textutf8 "Smoke Detector Detected!"'
                        print commands.getoutput(commandString)
                        print "hang off"
                        time.sleep(0.5)
            elif smokeLevel > 120 and smokeLevel < 125 and (a == 'Normal'
or a == '-') and (GPIO.input(24) == 1) and (GPIO.input(23) == 0):
                        print "dnnd"
                        commandString = 'sudo gammu sendsms TEXT
+66817097685 -textutf8 "Smoke Detector and Reed switch2 Detected!"'
                        print commands.getoutput(commandString)
                        print "hang off"
                        time.sleep(0.5)
            elif smokeLevel > 120 and smokeLevel < 125 and (a == 'Normal'
or a == '-') and (GPIO.input(24) == 0) and (GPIO.input(23) == 1):
                        print "dndn"
                        commandString = 'sudo gammu sendsms TEXT
+66817097685 -textutf8 "Smoke Detector and Reed switch1 Detected!"'
                        print commands.getoutput(commandString)
                        print "hang off"
                        time.sleep(0.5)
            elif smokeLevel > 120 and smokeLevel < 125 and (a == 'Normal'
or a == '-') and (GPIO.input(24) == 0) and (GPIO.input(23) == 0):
                        print "dndd"
                        commandString = 'sudo gammu sendsms TEXT
+66817097685 -textutf8 "Smoke Detector,Reed switch1 and Reed switch2
Detected!"'
                        print commands.getoutput(commandString)
                        print "hang off"
                        time.sleep(0.5)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            elif smokeLevel > 120 and smokeLevel < 125 and a == 'Detected'
and (GPIO.input(24) == 1) and (GPIO.input(23) == 1):
                        print "ddnn"
                        commandString = 'sudo gammu sendsms TEXT
+66817097685 -textutf8 "Smoke Detector and Motion Detector Detected!"'
                        print commands.getoutput(commandString)
                        print "hang off"
                        time.sleep(0.5)
            elif smokeLevel > 120 and smokeLevel < 125 and a == 'Detected'
and (GPIO.input(24) == 1) and (GPIO.input(23) == 0):
                        print "ddnd"
                        commandString = 'sudo gammu sendsms TEXT
+66817097685 -textutf8 "Smoke Detector,Motion Detector and Reed switch1
Detected!"'
                        print commands.getoutput(commandString)
                        print "hang off"
                        time.sleep(0.5)
            elif smokeLevel > 120 and smokeLevel < 125 and a == 'Detected'
and (GPIO.input(24) == 0) and (GPIO.input(23) == 1):
                        print "dddn"
                        commandString = 'sudo gammu sendsms TEXT
+66817097685 -textutf8 "Smoke Detector,Motion Detector and Reed switch2
Detected!"'
                        print commands.getoutput(commandString)
                        print "hang off"
                        time.sleep(0.5)
            elif smokeLevel > 120 and smokeLevel < 125 and a == 'Detected'
and (GPIO.input(24) == 0) and (GPIO.input(23) == 0):
                        print "dddd"
                        commandString = 'sudo gammu sendsms TEXT
+66817097685 -textutf8 "Smoke Detector,Motion Detector,Reed switch1 and
Reed switch2 Detected!"'
                        print commands.getoutput(commandString)
                        print "hang off"
                        time.sleep(0.5)
            else:
                    print "invalid condition"
            print "out loop"
            time.sleep(0.5)
if __name__ == "__main__":
      process()
```

alarm.py (โปรแกรมแจ้งเตือนโดยการโทรเข้าเบอร์ผู้ใช้บริการ)

```
import time
import MySQLdb
import RPi.GPIO as GPIO
import botbook_mcp3002 as mcp
import commands

GPIO.setmode(GPIO.BCM)
GPIO.setup(23, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(24, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

db = MySQLdb.connect("localhost", "root", "khunsuek", "sensor status")
cur = db.cursor()
cur.execute("SELECT status FROM sensor_status WHERE sensor =
'motion_detector'")
result = cur.fetchall()
for i in result:
      a = i[0]

def readSmokeLevel():
```

เอกสารนี้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      global smokeLevel
      smokeLevel = mcp.readAnalog()

def process():
      print "process start"
      while True:
            readSmokeLevel()
            print "in loop"
            time.sleep(0.5)
            if smokeLevel < 120 and (a == 'Normal' or a == '-')  and
(GPIO.input(24) == 1) and (GPIO.input(23) == 0):
                  print "nnnd"
                  commandString = 'sudo gammu maketerminatedcall
+66817097685 10'
                  print commands.getoutput(commandString)
                  print "hang off"
                  time.sleep(0.5)
            elif smokeLevel < 120 and (a == 'Normal' or a == '-') and
(GPIO.input(24) == 0) and (GPIO.input(23) == 1):
                  print "nndn"
                  commandString = 'sudo gammu maketerminatedcall
+66817097685 10'
                        print commands.getoutput(commandString)
                  print "hang off"
                        time.sleep(0.5)
            elif smokeLevel < 120 and (a == 'Normal' or a == '-') and
(GPIO.input(24) == 0) and (GPIO.input(23) == 0):
                        print "nndd"
                        commandString = 'sudo gammu maketerminatedcall
+66817097685 10'
                  print commands.getoutput(commandString)
                  print "hang off"
                  time.sleep(0.5)
            elif smokeLevel < 120 and a == 'Detected' and (GPIO.input(24)
== 1) and (GPIO.input(23) == 1):
                        print "ndnn"
                        commandString = 'sudo gammu maketerminatedcall
+66817097685 10'
                        print commands.getoutput(commandString)
                  print "hang off"
                        time.sleep(0.5)
            elif smokeLevel < 120 and a == 'Detected' and (GPIO.input(24)
== 1) and (GPIO.input(23) == 0):
                  print "ndnd"
                        commandString = 'sudo gammu maketerminatedcall
+66817097685 10'
                        print commands.getoutput(commandString)
                        print "hang off"
                  time.sleep(0.5)
            elif smokeLevel < 120 and a == 'Detected' and (GPIO.input(24)
== 0) and (GPIO.input(23) == 1):
                        print "nddn"
                        commandString = 'sudo gammu maketerminatedcall
+66817097685 10'
                        print commands.getoutput(commandString)
                        print "hang off"
                        time.sleep(0.5)
            elif smokeLevel < 120 and a == 'Detected' and (GPIO.input(24)
== 0) and (GPIO.input(23) == 0):
                        print "nddd"
```

เอกสารนี้เป็นเอกสาร... ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้... และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        commandString = 'sudo gammu maketerminatedcall
+66817097685 10'
                        print commands.getoutput(commandString)
                        print "hang off"
                        time.sleep(0.5)
            elif smokeLevel > 120 and smokeLevel < 125 and (a == 'Normal'
or a == '-') and (GPIO.input(24) == 1) and (GPIO.input(23) == 1):
                        print "dnnn"
                        commandString = 'sudo gammu maketerminatedcall
+66817097685 10'
                        print commands.getoutput(commandString)
                        print "hang off"
                        time.sleep(0.5)
            elif smokeLevel > 120 and smokeLevel < 125 and (a == 'Normal'
or a == '-') and (GPIO.input(24) == 1) and (GPIO.input(23) == 0):
                        print "dnnd"
                        commandString = 'sudo gammu maketerminatedcall
+66817097685 10'
                        print commands.getoutput(commandString)
                        print "hang off"
                        time.sleep(0.5)
            elif smokeLevel > 120 and smokeLevel < 125 and (a == 'Normal'
or a == '-') and (GPIO.input(24) == 0) and (GPIO.input(23) == 1):
                        print "dndn"
                        commandString = 'sudo gammu maketerminatedcall
+66817097685 10'
                        print commands.getoutput(commandString)
                        print "hang off"
                        time.sleep(0.5)
            elif smokeLevel > 120 and smokeLevel < 125 and(a == 'Normal' or
a == '-') and (GPIO.input(24) == 0) and (GPIO.input(23) == 0):
                        print "dndd"
                        commandString = 'sudo gammu maketerminatedcall
+66817097685 10'
                        print commands.getoutput(commandString)
                        print "hang off"
                        time.sleep(0.5)
            elif smokeLevel > 120 and smokeLevel < 125  and a == 'Detected'
and (GPIO.input(24) == 1) and (GPIO.input(23) == 1):
                        print "ddnn"
                        commandString = 'sudo gammu maketerminatedcall
+66817097685 10'
                        print commands.getoutput(commandString)
                        print "hang off"
                        time.sleep(0.5)
            elif smokeLevel > 120 and smokeLevel < 125 and a == 'Detected'
and (GPIO.input(24) == 1) and (GPIO.input(23) == 0):
                        print "ddnd"
                        commandString = 'sudo gammu maketerminatedcall
+66817097685 10'
                        print commands.getoutput(commandString)
                        print "hang off"
                        time.sleep(0.5)
            elif smokeLevel > 120 and smokeLevel < 125 and a == 'Detected'
and (GPIO.input(24) == 0) and (GPIO.input(23) == 1):
                        print "dddn"
                        commandString = 'sudo gammu maketerminatedcall
+66817097685 10'
                        print commands.getoutput(commandString)
                        print "hang off"
                        time.sleep(0.5)
```

เอกสารนี้มีลิขสิทธิ์ สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            elif smokeLevel > 120 and smokeLevel < 125 and a == 'Detected'
and (GPIO.input(24) == 0) and (GPIO.input(23) == 0):
                        print "dddd"
                        commandString = 'sudo gammu maketerminatedcall
+66817097685 10'
                        print commands.getoutput(commandString)
                        print "hang off"
                        time.sleep(0.5)
            else:
                  print "invalid condition"
            print "out loop"
            time.sleep(0.5)
if __name__ == "__main__":
      process()
```

start_stream.py (โปรแกรมเริ่มการ stream)

```
import subprocess
subprocess.call(['/var/www/programe/start_stream.sh'])
```

stop_stream.py (โปรแกรมหยุดการ stream)

```
import subprocess
subprocess.call(['/var/www/programe/stop_streamer.sh'])
```

start_stream.sh

```
#!/bin/bash
      mkdir /tmp/stream/
      chmod a+rw /tmp/stream/
if [pgrep raspistill > /dev/null]
then
      echo "raspistill already running"
else
      raspistill -w 640 -h 480 -q 5 -o /tmp/stream/pic.jpg -tl 100 -t 0:0:0
-n > /dev/null 2>&1&
      echo "raspistill started"
fi
if [pgrep mjpg_streamer > /dev/null]
then
      echo "mjpg_streamer already running"
else
      LD_LIBRARY_PATH=/opt/mjpg-streamer/mjpg-streamer/ /opt/mjpg-
streamer/mjpg-streamer/mjpg_streamer -i "input_file.so -f /tmp/stream -n
pic.jpg" -o "output_http.so -p 9000 -w /opt/mjpg-streamer/mjpg-
streamer/www"> /dev/null 2>&1&
      echo "mjpg_streamer started"
fi
```

stop_streamer.sh

```
#!/bin/bash

if pgrep raspistill
then
      kill $(pgrep raspistill) > /dev/null 2>&1
      echo "raspistill stopped"
else
      echo "raspistill not running"
fi

if pgrep mjpg_streamer
then
      kill $(pgrep mjpg_streamer) > /dev/null 2>&1
      echo "mjpg_streamer stopped"
else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        echo "mjpg_streamer not running"
fi
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

Datasheet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TECHNICAL DATA MQ-2 GAS SENSOR

FEATURES

Wide detecting scope
Fast response and High sensitivity
Stable and long life
Simple drive circuit

APPLICATION

They are used in gas leakage detecting equipments in family and industry, are suitable for detecting of LPG, i-butane, propane, methane ,alcohol, Hydrogen, smoke.

SPECIFICATIONS

A. Standard work condition

Symbol	Parameter name	Technical condition	Remarks
Vc	Circuit voltage	5V±0.1	AC OR DC
V_H	Heating voltage	5V±0.1	ACOR DC
R_L	Load resistance	can adjust	
R_H	Heater resistance	33 Ω ±5%	Room Tem
P_H	Heating consumption	less than 800mw	

B. Environment condition

Symbol	Parameter name	Technical condition	Remarks
Tao	Using Tem	-20℃-50℃	
Tas	Storage Tem	-20℃-70℃	
R_H	Related humidity	less than 95%Rh	
O_2	Oxygen concentration	21%(standard condition)Oxygen concentration can affect sensitivity	minimum value is over 2%

C. Sensitivity characteristic

Symbol	Parameter name	Technical parameter	Remarks
Rs	Sensing Resistance	3KΩ-30KΩ (1000ppm iso-butane)	Detecting concentration scope: 200ppm-5000ppm LPG and propane 300ppm-5000ppm butane 5000ppm-20000ppm methane 300ppm-5000ppm H_2 100ppm-2000ppm Alcohol
α (3000/1000) isobutane	Concentration Slope rate	≤0.6	
Standard Detecting Condition	Temp: 20℃±2℃ Humidity: 65%±5%	Vc:5V±0.1 Vh: 5V±0.1	
Preheat time	Over 24 hour		

D. Structure and configuration, basic measuring circuit

	Parts	Materials
1	Gas sensing layer	SnO_2
2	Electrode	Au
3	Electrode line	Pt
4	Heater coil	Ni-Cr alloy
5	Tubular ceramic	Al_2O_3
6	Anti-explosion network	Stainless steel gauze (SUS316 100-mesh)
7	Clamp ring	Copper plating Ni
8	Resin base	Bakelite
9	Tube Pin	Copper plating Ni

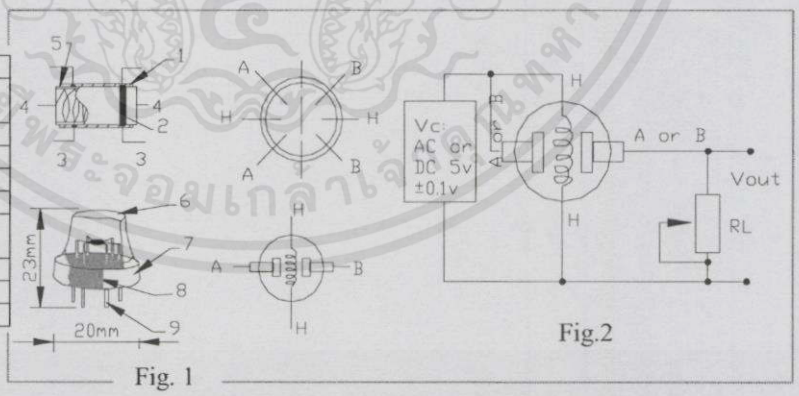
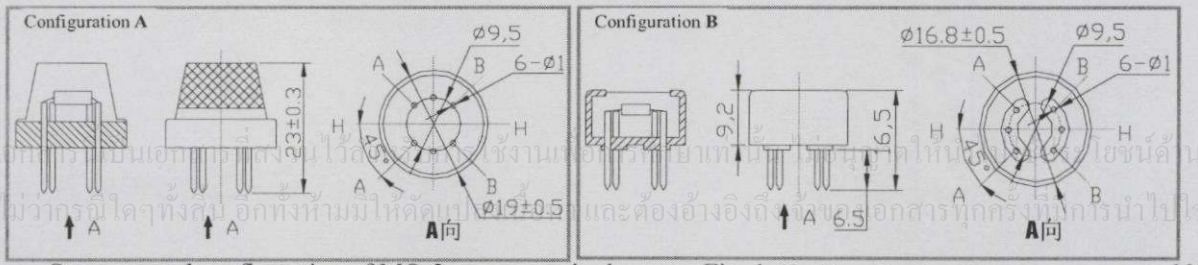


Fig. 1

Fig.2



Structure and configuration of MQ-2 gas sensor is shown as Fig. 1 (Configuration **A** or **B**), sensor composed by micro AL_2O_3 ceramic tube, Tin Dioxide (SnO_2) sensitive layer, measuring electrode and heater are fixed into a

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

crust made by plastic and stainless steel net. The heater provides necessary work conditions for work of sensitive components. The enveloped MQ-2 have 6 pin ,4 of them are used to fetch signals, and other 2 are used for providing heating current.

Electric parameter measurement circuit is shown as Fig.2

E. Sensitivity characteristic curve

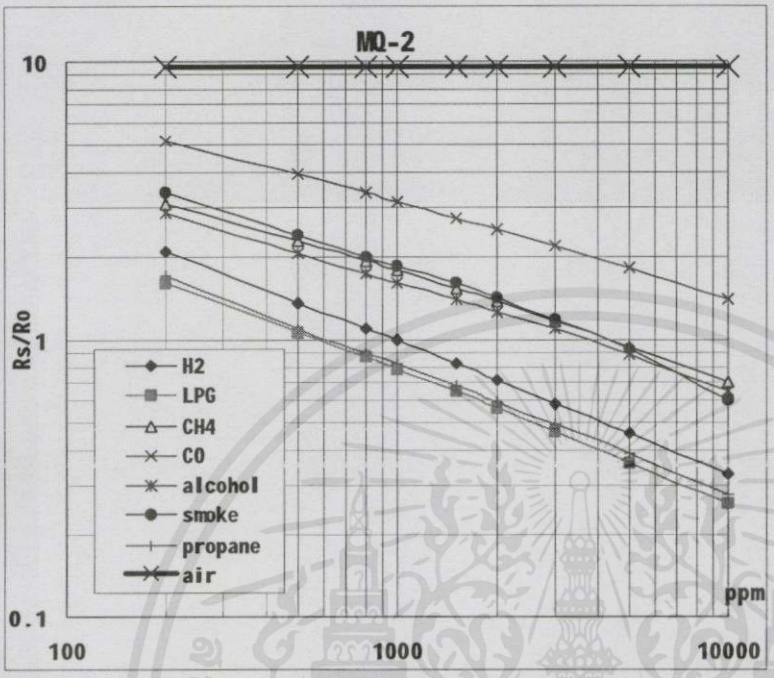


Fig.2 sensitivity characteristics of the MQ-2

Fig.3 is shows the typical sensitivity characteristics of the MQ-2 for several gases.
in their: Temp: 20℃、
Humidity: 65%、
O_2 concentration 21%
RL=5kΩ
Ro: sensor resistance at 1000ppm of H_2 in the clean air.
Rs:sensor resistance at various concentrations of gases.

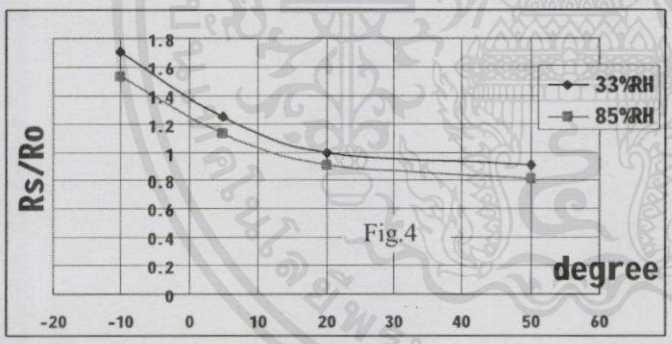


Fig.4

Fig.4 is shows the typical dependence of the MQ-2 on temperature and humidity.
Ro: sensor resistance at 1000ppm of H_2 in air at 33%RH and 20 degree.
Rs: sensor resistance at 1000ppm of H_2 at different temperatures and humidities.

SENSITVITY ADJUSTMENT

Resistance value of MQ-2 is difference to various kinds and various concentration gases. So,When using this components, sensitivity adjustment is very necessary. we recommend that you calibrate the detector for 1000ppm liquified petroleum gas<LPG>,or 1000ppm iso-butane<i-C_4H_{10}>concentration in air and use value of Load resistance that(R_L) about 20 KΩ (5KΩ to 47 KΩ).

When accurately measuring, the proper alarm point for the gas detector should be determined after considering the temperature and humidity influence.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

REED SWITCH

ORD213

Super Ultraminiature

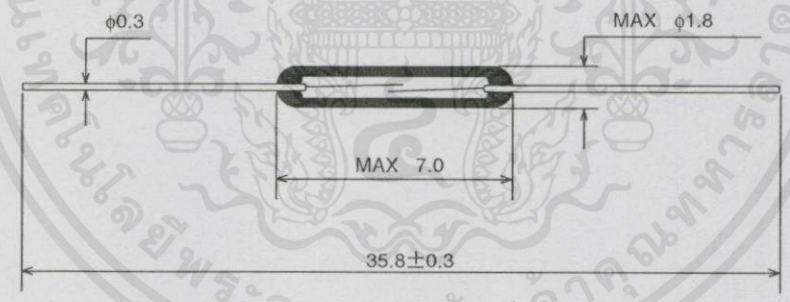
■ GENERAL DESCRIPTION

The ORD213 is a small single-contact reed switch designed for general control of low-level loads less than 24 V. The reed contacts are sealed within the glass tube within inert gas to maintain contact reliability.

■ FEATURES

(1) Reed contacts are hermetically sealed within a glass tube with inert gas and do not receive any influence from the external atmospheric environment.
(2) Quick response
(3) The structure comprises the operating parts and electrical circuits arranged coaxially. Reed switches are suited to applications in radio frequency operation.
(4) Reed switches are compact and light weight.
(5) Superior corrosion resistance and wear resistance of the contacts assures stable switching operation and long life.
(6) With a permanent magnet installed, reed switches economically and easily become proximity switches.

■ EXTERNAL DIMENSIONS (Unit: mm)



■ APPLICATIONS

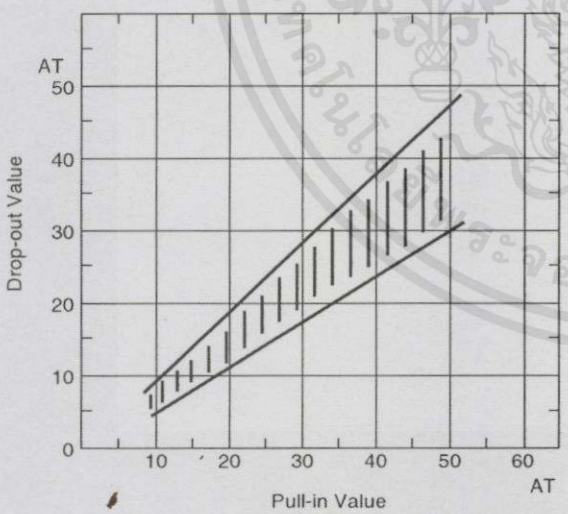
- Automotive electronic devices
- Control equipment
- Communication equipment
- Measurement equipment
- Household appliances

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

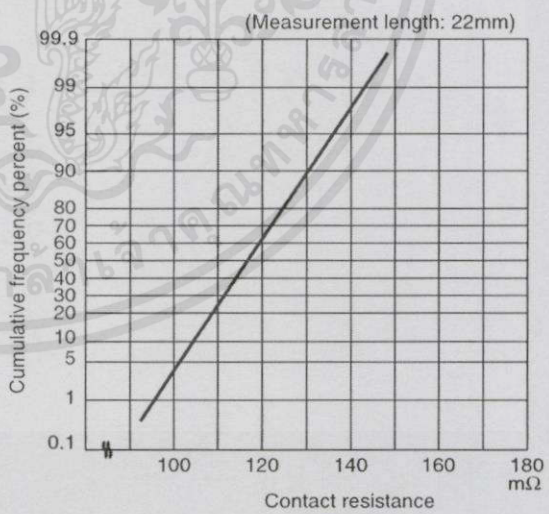
■ ELECTRICAL CHARACTERISTICS

Parameter	Rated value	Unit
Pull-in Value (PI)	10~40	AT
Drop-out Value (DO)	5min	AT
Contact resistance (CR)	200max	mΩ
Breakdown voltage	150min	VDC
Insulation resistance	10^9min	Ω
Electrostatic capacitance	0.4max	pF
Contact rating	1.0	VA
Maximum switching voltage	24 ($\frac{DC}{AC}$)	V
Maximum switching current	0.1	A
Maximum carry current	0.3	A

(1) Drop-out Value vs. Pull-in Value

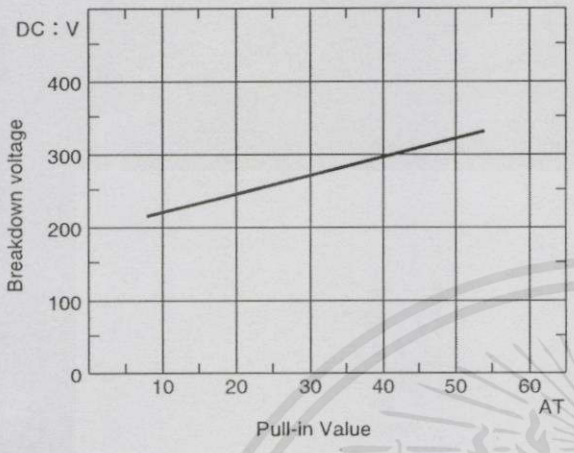


(2) Contact resistance

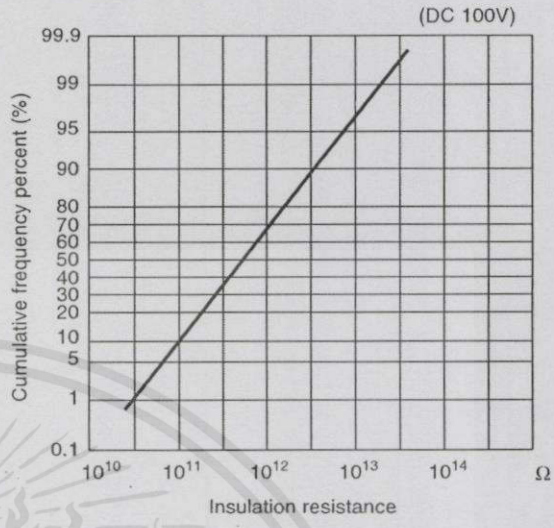


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

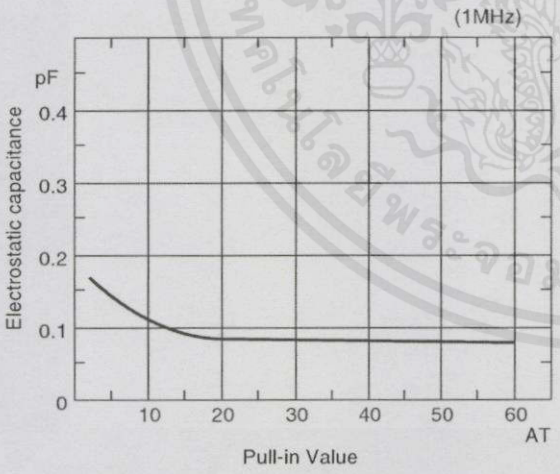
(3) Breakdown voltage



(4) Insulation resistance



(5) Electrostatic capacitance

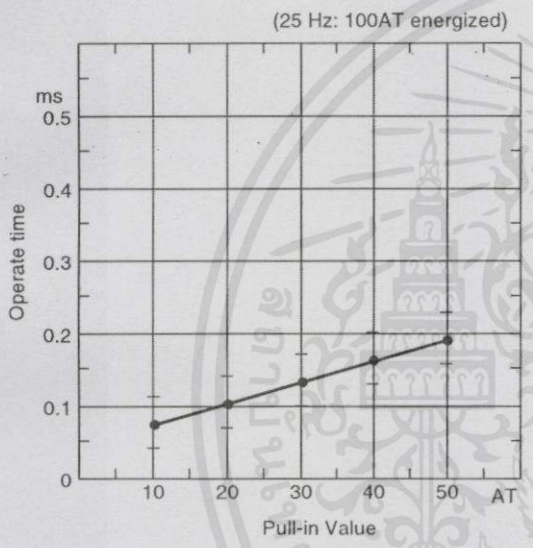


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

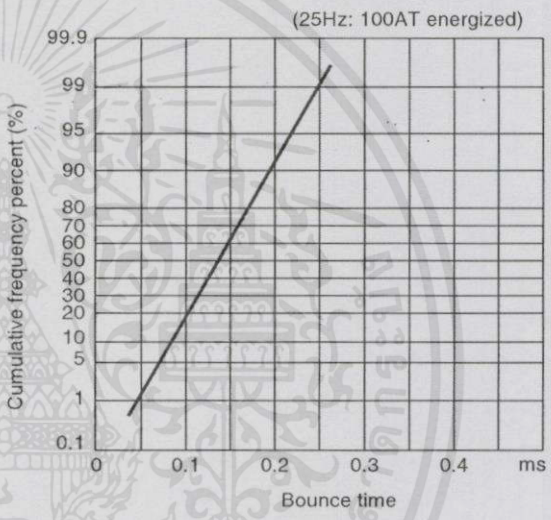
■ OPERATING CHARACTERISTICS

Parameter	Rated value	Unit
Operate time	0.3max	ms
Bounce time	0.3max	ms
Release time	0.05max	ms
Resonant frequency	11000±2000	Hz
Maximum operating frequency	500	Hz

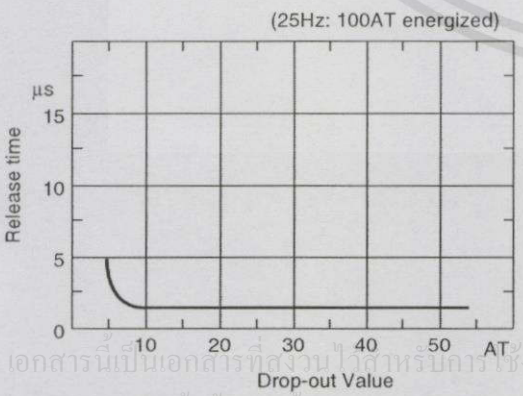
(1) Operate time



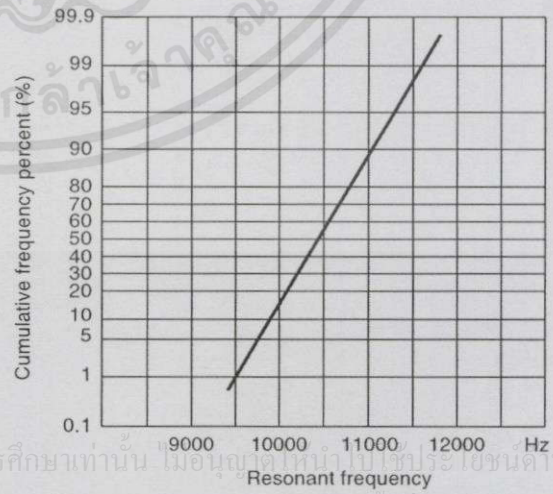
(2) Bounce time



(3) Release time



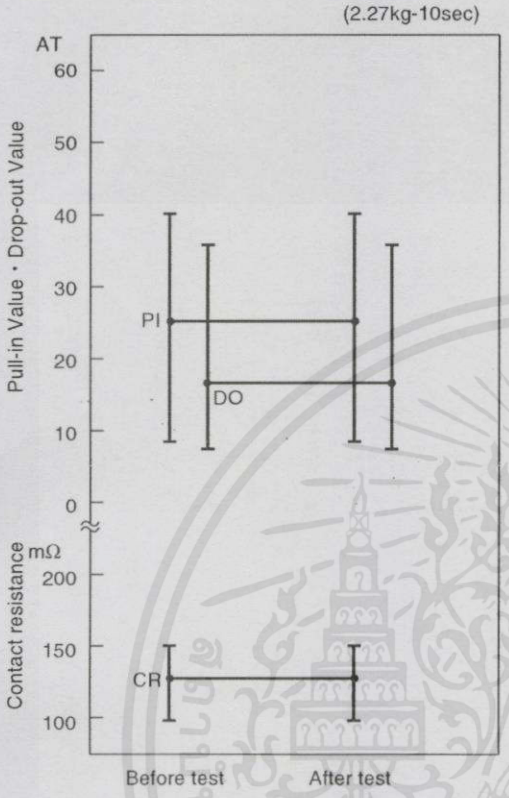
(4) Resonant frequency



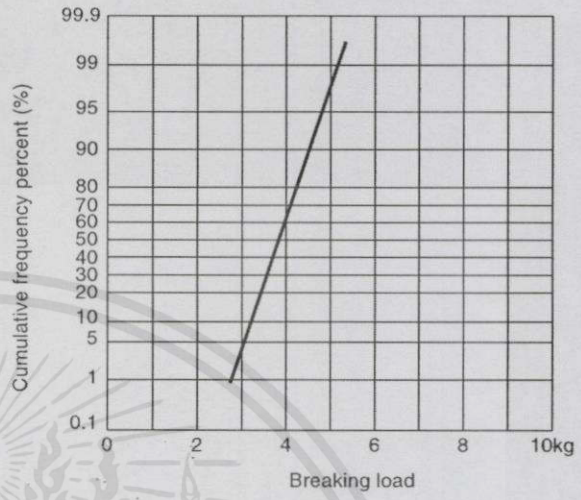
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

■ MECHANICAL CHARACTERISTICS

(1) Lead tensile test (static load)

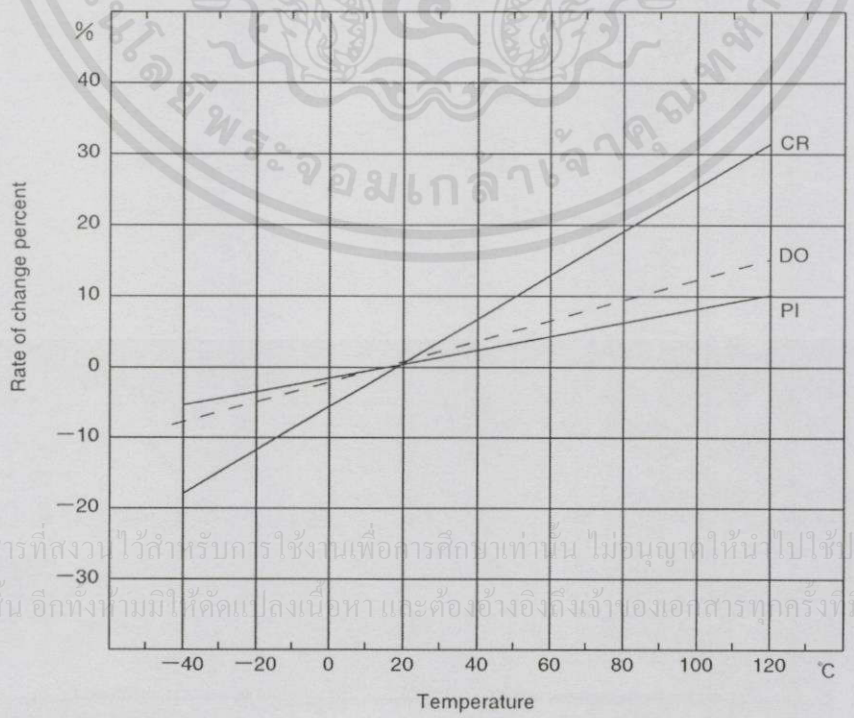


(2) Lead tensile strength



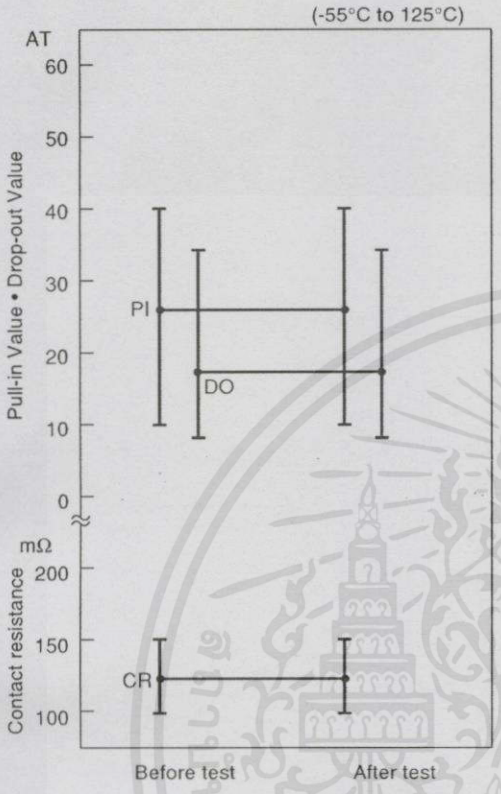
■ ENVIRONMENTAL CHARACTERISTICS

(1) Temperature characteristics

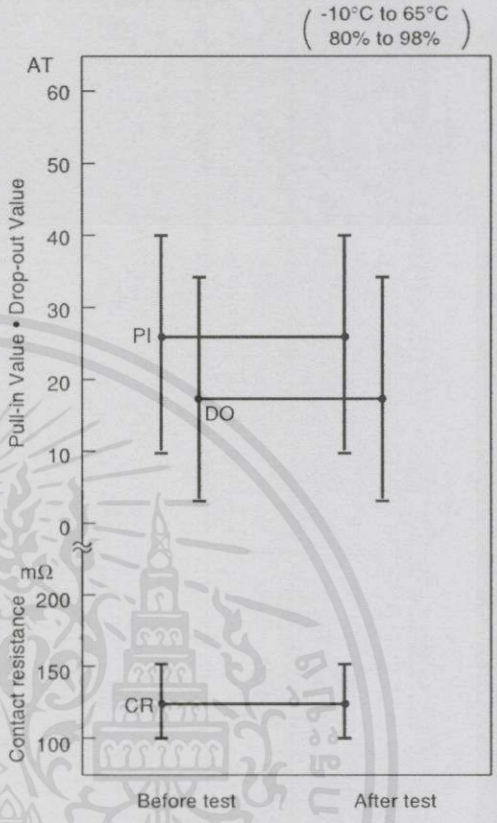


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

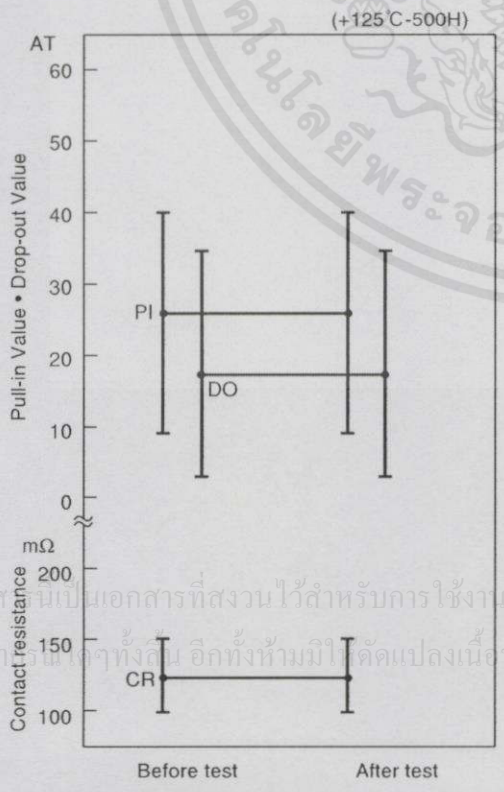
(2) Temperature cycle



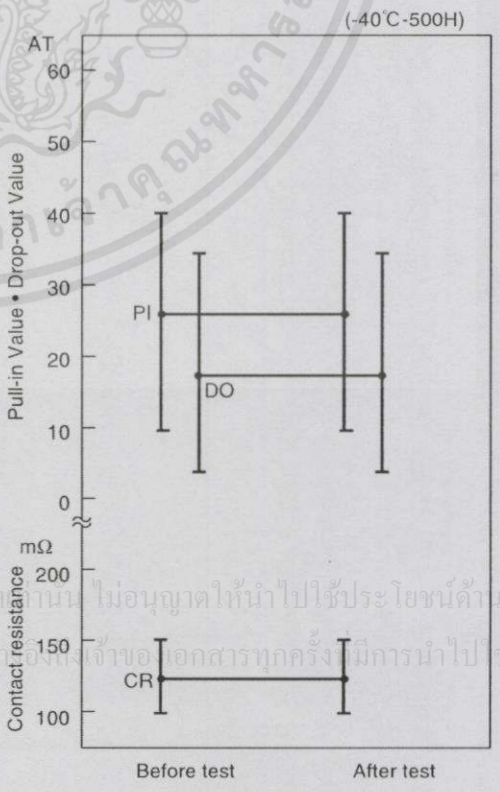
(3) Temperature and humidity cycle



(4) High temperature storage test



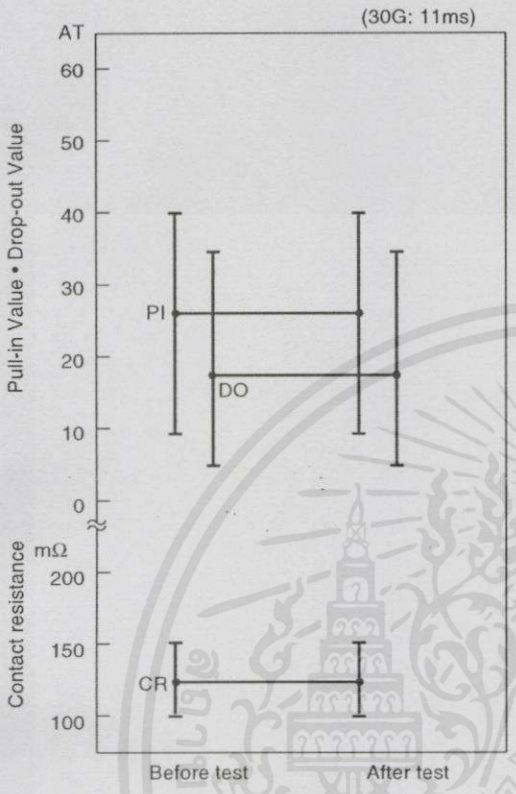
(5) Low temperature storage test



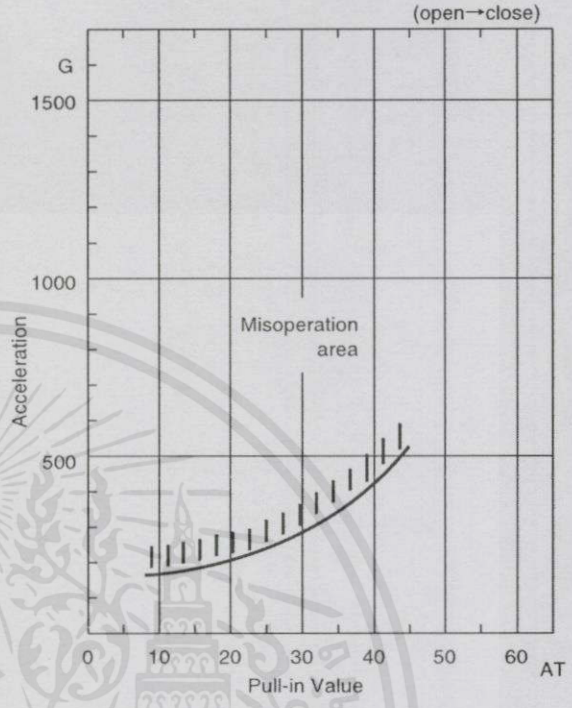
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(6) Shock test

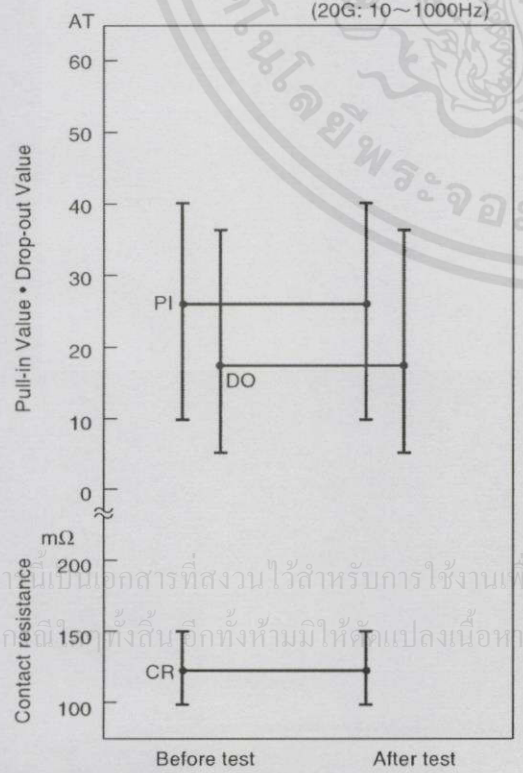
1) Electrical characteristics



2) Misoperation area



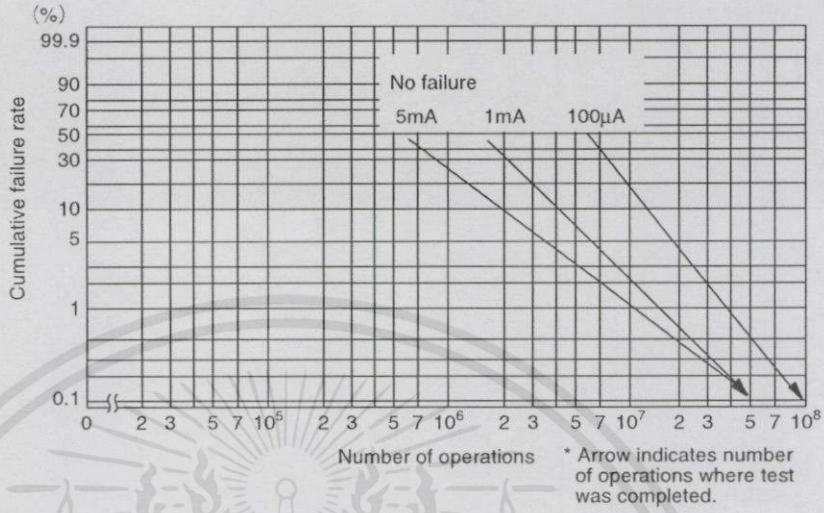
(7) Vibration test



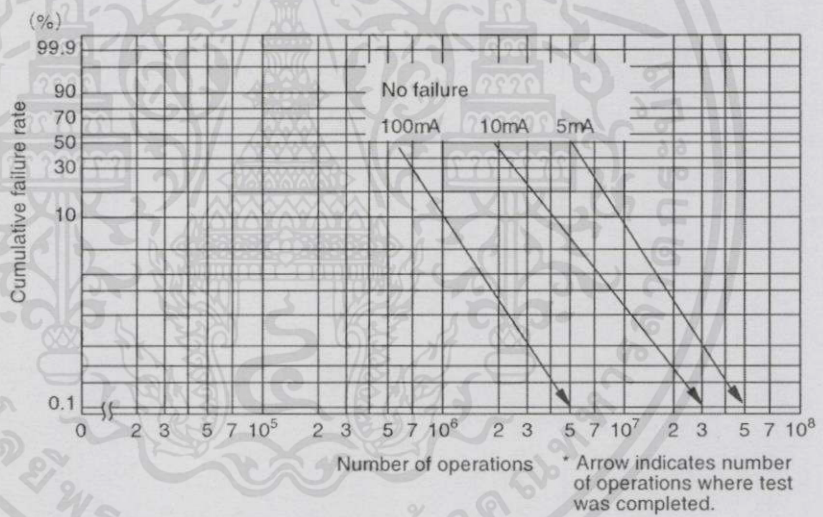
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

■ LIFE EXPECTANCY DATA: ORD213

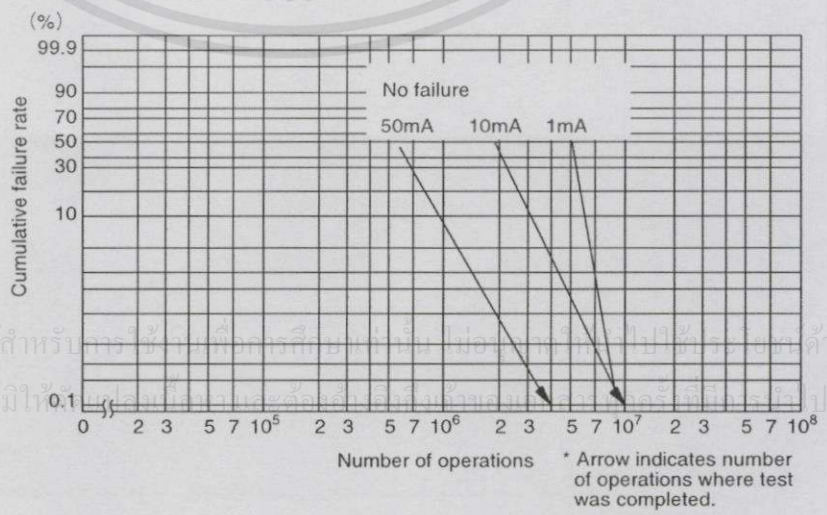
Load conditions
Voltage: 5VDC
Current: 100μA , 1mA , 5mA
Load: Resistive load



Load conditions
Voltage: 12VDC
Current: 5mA , 10mA , 100mA
Load: Resistive load



Load conditions
Voltage: 24VDC
Current: 1mA , 10mA , 50mA
Load: Resistive load



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของลิขสิทธิ์ทุกครั้งที่มีการนำไปใช้

รายละเอียด TruemoveH 3G+ Aircard E303

- รองรับเครือข่าย 3G : UMTS/HSDPA/HSUPA 2100/850 MHz
  2G : GSM 850/900/1800/1900 MHz
- ความเร็วในการดาวน์โหลดสูงสุด 7.2 Mbps, ความเร็วในการอัพโหลดสูงสุด 5.76 Mbps
- รองรับการ์ดหน่วยความจำ MicroSD สูงสุด 32 GB
- รองรับระบบปฏิบัติการ
  - Windows XP SP2-3, Windows Vista, Windows 7,
  - Mac OS X 10.5, 10.6 และ 10.7
- รองรับการใช้งานรับ-ส่งข้อความ SMS
- มีเมนูเสริมสำหรับการเช็คยอดเงิน, เติมเงิน จากโปรแกรมได้
- คู่มือการใช้งานโปรแกรมรองรับภาษาไทย
- รองรับการเชื่อมต่อ Wi-Fi, และสลับใช้งาน 3G และ Wi-Fi ผ่านโปรแกรม
- สนับสนุนฟังก์ชั่น SIM Lock (ใช้ได้เฉพาะซิมการ์ด TrueMove และ TrueMoveH)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้