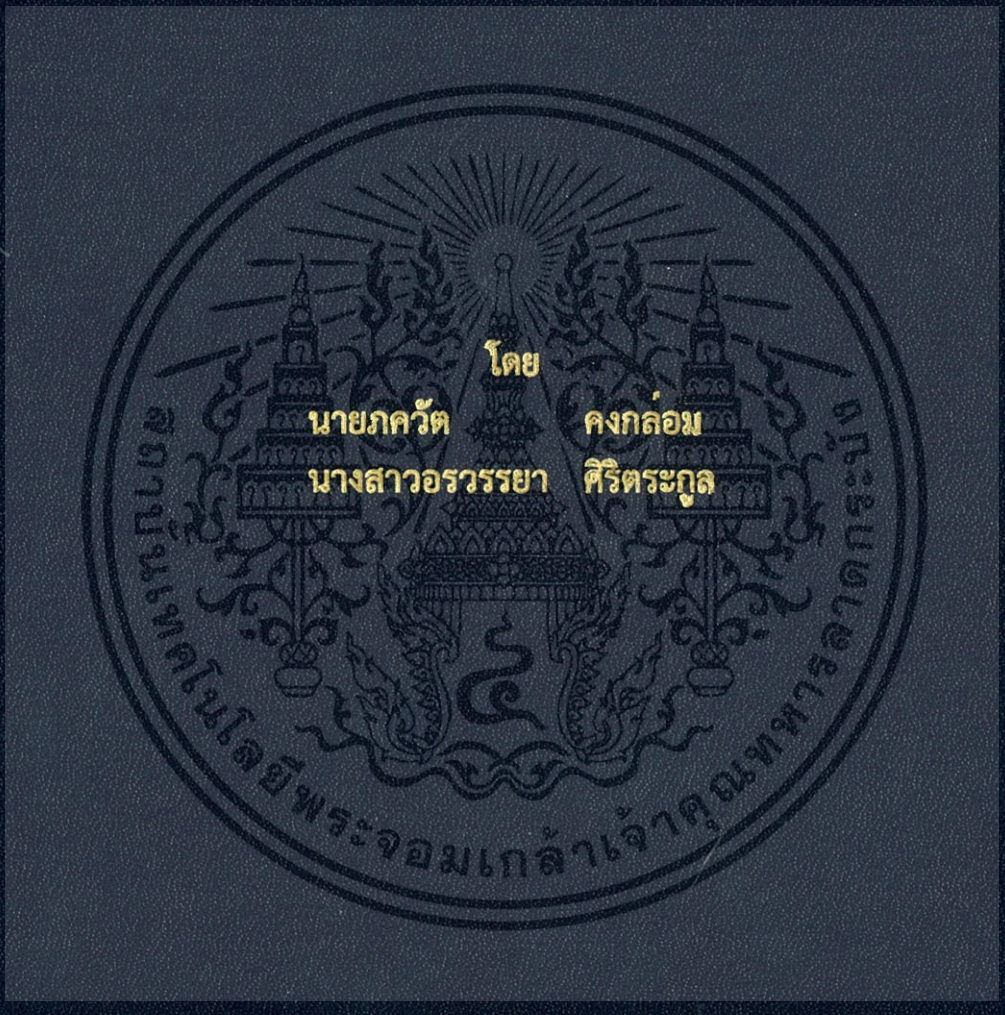
ระบบแจ้งเตือนการเกิดเพลิงไหม้สำหรับอาคารขนาดใหญ่
FIRE ALARM MONITORING SYSTEM FOR LARGE BUILDING



โดย

นายภควัต คงกล่อม

นางสาวอรวรรยา ศิริตระกูล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2559

ระบบแจ้งเตือนการเกิดเพลิงไหม้สำหรับอาคารขนาดใหญ่

FIRE ALARM MONITORING SYSTEM FOR LARGE BUILDING

โดย

นายภควัต คงกล่อม

นางสาวอรวรรยา ศิริตระกูล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2559

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบแจ้งเตือนการเกิดเพลิงไหม้สำหรับอาคารขนาดใหญ่
FIRE ALARM MONITORING SYSTEM FOR LARGE BUILDING

โดย

นายภควัต	คงกล่อม	56010900
นางสาวอรวรรยา	ศิริตระกูล	56011428

อาจารย์ที่ปรึกษา

ผศ.ดร. พิเชฐ ม่วงนวล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2559

ผ่านการตรวจรูปเล่มแล้ว

(.....)

อาจารย์ที่ปรึกษา

19/05/60

ผ่านการตรวจชิ้นงานแล้ว

(.....)

กรรมการผู้ตรวจชิ้นงาน

19/05/60

ลิขสิทธิ์เป็นของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ขอสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2559
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เรื่อง ระบบแจ้งเตือนการเกิดเพลิงไหม้สำหรับอาคารขนาดใหญ่
FIRE ALARM MONITORING SYSTEM FOR LARGE BUILDING

ผู้จัดทำ
1. นายภควัต คงกล่อม 56010900
2. นางสาวอรวรรยา ศิริตระกูล 56011428

..... อาจารย์ที่ปรึกษา
(ผศ.ดร. พิเชฐ ม่วงนวล)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานเล่มนี้คงไม่อาจสำเร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และความร่วมมือจากหลายๆฝ่ายด้วยกัน บุคคลแรกที่กลุ่มของข้าพเจ้าต้องขอกล่าวถึง เพราะเป็นบุคคลสำคัญที่ทำให้โครงงานนี้สำเร็จได้ คือ ผศ.ดร. พิเชฐ ม่วงนวล อาจารย์ที่ปรึกษาโครงงาน ที่คอยให้คำปรึกษา เอาใจใส่ และให้คำแนะนำ คอยย้ำเตือนในช่วงระยะเวลาและการวางแผนการดำเนินงานมาโดยตลอด และให้ความช่วยเหลือเป็นอย่างดีโดยตลอดมา

ขอขอบพระคุณคณาจารย์ ภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้ประสิทธิ์ประสาทวิชาให้กับข้าพเจ้า

ขอขอบคุณเพื่อนๆพี่ๆน้องๆทุกคน ที่คอยให้ความช่วยเหลือ และเป็นกำลังใจให้กันโดยตลอดมา จนทำให้โครงงานฉบับนี้ลุล่วงไปได้ด้วยดี

และอันดับสุดท้าย ต้องกราบขอบพระคุณบุคคลที่สุดอันหาที่ทดแทนไม่ได้ ที่ทำให้ข้าพเจ้ามีวันนี้ได้ ก็คือ บิดา มารดา อันเป็นที่เคาพรักที่สุดของข้าพเจ้า ที่ได้กำเนิดและเลี้ยงดูข้าพเจ้าเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และให้กำลังใจ เอาใจใส่และให้ความช่วยเหลือในทุกๆด้านเป็นอย่างดี อันหาที่เปรียบไม่ได้

นายภควัต คงกล่อม
นางสาวอรวรรยา ศิริตระกูล
ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบแจ้งเตือนการเกิดเพลิงไหม้สำหรับอาคารขนาดใหญ่
FIRE ALARM MONITORING SYSTEM FOR LARGE BUILDING

โดย นายภควัต คงกล่อม 56010900
นางสาวอรวรรยา ศิริตระกูล 56011428

อาจารย์ที่ปรึกษา ผศ.ดร. พิเชฐ ม่วงนวล

บทคัดย่อ

โครงงานนี้เป็นการนำเสนอ ระบบการแจ้งเตือนการเกิดเหตุเพลิงไหม้สำหรับอาคารขนาดใหญ่ ภายในอาคารจะทำการติดตั้งเซนเซอร์ในทุกห้องของอาคาร หากเกิดเหตุเพลิงไหม้ เซนเซอร์จะทำการส่งสัญญาณไปยังไมโครคอนโทรลเลอร์ ไมโครคอนโทรลเลอร์ทำการหาตำแหน่งเกิดเหตุเพื่อส่งไปยังห้องควบคุมและทำการแจ้งเตือน ห้องควบคุมจะมีจอมอนิเตอร์คอยแจ้งเตือนสถานะความปลอดภัยในแต่ละห้องเอาไว้ หากเกิดเหตุเพลิงไหม้ Buzzer จะทำการแจ้งเตือนเป็นสัญญาณเสียง และมอนิเตอร์จะแสดงตำแหน่งเกิดเหตุบนจอมอนิเตอร์ในห้องควบคุม นอกจากนี้ผู้ใช้งานสามารถตรวจสอบการเกิดเหตุเพลิงไหม้ผ่านแอปพลิเคชันจากโทรศัพท์มือถือได้เช่นกัน

ABSTRACT

This project presents fire alarm monitoring system for large building. The system works by use sensors to monitors fire. Inside, the building will install sensors in every room. When the sensor can be detected fire, sensor will send a signal to the microcontroller. Microcontroller can be find any area where the fire occurred. Then it sends an alarm via buzzer and shows any area where the fire occurred on the monitor screen in the control room. In addition, users can check about fire through the mobile application as well.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

		หน้า
กิตติกรรมประกาศ		I
บทคัดย่อ		II
สารบัญ		III
สารบัญรูป		V
สารบัญตาราง		VIII
บทที่ 1	**บทนำ**	**1**
	1.1 ความเป็นมาและความสำคัญของปัญหา	1
	1.2 วัตถุประสงค์	1
	1.3 ขอบเขตของโครงงาน	1
บทที่ 2	**ทฤษฎีและหลักการที่เกี่ยวข้อง**	**3**
	2.1 ไมโครคอนโทรลเลอร์	3
	2.2 การตรวจจับอัคคีภัย	8
	2.3 MICROSOFT VISUAL BASIC	12
	2.4 INFRARED IR FLAME DETECTOR SENSOR MODULE	17
	2.5 ACTIVE BUZZER	20
	2.6 NRF24L01 MODULE	20
	2.7 MCP3008 (16-BIT I/O EXPANDER WITH SERIAL INTERFACE)	22
	2.8 NODE MCU	24
บทที่ 3	**การออกแบบและการจัดทำปริญญานิพนธ์**	**26**
	3.1 การออกแบบ	26
	3.2 เครื่องมือที่ใช้ในการทดลอง	39
	3.3 การจัดเก็บผลการทดลอง	41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

		หน้า
บทที่ 4	**ผลการทดลอง**	**42**
	4.1 ผลการทดสอบเซนเซอร์	42
	4.2 ผลการทดสอบเซนเซอร์เมื่อผ่าน MCP23017	46
	4.3 ผลการทดสอบ NRF24L01	48
	4.4 ผลการทดสอบในส่วนของแอปพลิเคชัน	48
	4.5 ผลการทดสอบในส่วนของจอมอนิเตอร์	50
	4.6 แบบชิ้นงาน	53
บทที่ 5	**สรุปผลและข้อเสนอแนะ**	**55**
	5.1 สรุปผล	55
	5.2 ข้อเสนอแนะ	55
บรรณานุกรม		**56**
ภาคผนวก		**57**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่		หน้า
1.1	บล็อกไดอะแกรมของโครงงาน	2
2.1	บอร์ดอาร์ดูอิโน (ARDUINO BOARD)	3
2.2	การเขียนโปรแกรมบน ARDUINO	6
2.3	เลือกรุ่นบอร์ด ARDUINO ที่ต้องการ UPLOAD	6
2.4	เลือกหมายเลข COMPORT ของบอร์ด	7
2.5	กดปุ่ม VERIFY เพื่อตรวจสอบความถูกต้องและ COMPILE โค้ดโปรแกรม	7
2.6	UPLOAD โค้ด โปรแกรม	8
2.7	แสดงขั้นตอนการเกิดการลุกไหม้ของอัคคีภัย (FIRE DEVELOPMENT STAGES)	9
2.8	หน้าต่าง START PAGE	13
2.9	หน้าต่าง SOLUTION EXPLORER	14
2.10	หน้าต่าง PROPERTIES	14
2.11	หน้าต่าง COMMON CONTROL	15
2.12	หน้าต่าง CONTAINERS	16
2.13	หน้าต่าง COMPONENTS	16
2.14	หน้าต่าง STANDARD TOOLBAR	17
2.15	หน้าต่าง MENU BAR	17
2.16	INFRARED IR FLAME DETECTOR SENSOR MODULE	18
2.17	หน้าที่ของขาเซนเซอร์	19
2.18	การเชื่อมต่อ INFRARED IR FLAME DETECTOR SENSOR MODULE กับ ARDUINO	19
2.19	ACTIVE BUZZER	20
2.20	NRF24L01 MODULE	20
2.21	การต่อขาโมดูล NRF24L01 กับ ARDUINO	21
2.22	ไอซี MCP23017 (16-BIT I/O EXPANDER WITH SERIAL INTERFACE)	22
2.23	ขาของไอซี MCP23017 (16-BIT I/O EXPANDER WITH SERIAL INTERFACE)	23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่		หน้า
2.24	NODE MCU	24
2.25	ขาของ NODE MCU	25
3.1	บล็อกไดอะแกรมของโครงงาน	26
3.2	วงจรรวมของระบบฝั่งเครื่องส่ง	27
3.3	วงจรรวมของระบบฝั่งเครื่องรับ	28
3.4	หน้าล็อกอิน	29
3.5	ข้อความแจ้งเตือนเมื่อกรอกรหัสผิด	29
3.6	หน้าต่างการกรอกรายละเอียดของอาคาร	29
3.7	หน้าต่างการยืนยันลายละเอียดของอาคาร	30
3.8	หน้าจอหลัก	30
3.9	หน้าจอหลักแสดงการแจ้งเตือนเมื่อเกิดเหตุเพลิงไหม้	31
3.10	แอปพลิเคชันแจ้งเตือนการเกิดเหตุเพลิงไหม้	31
3.11	ผังการทำงานของหน้าล็อกอิน	32
3.12	ผังการทำงานของหน้าจอหลัก	33
3.13	ผังการทำงานของเครื่องส่งในระบบ	34
3.14	ผังการทำงานของเครื่องรับในระบบ	35
3.15	ผังการทำงานของส่วนแสดงผลจอมอนิเตอร์	36
3.16	ผังการทำงานของส่วนแอปพลิเคชัน	37
3.17	ผังการทำงานรวมของระบบ	38
3.18	ไมโครคอนโทรลเลอร์ ARDUINO UNO3	39
3.19	INFRARED IR FLAME DETECTOR SENSOR MODULE	39
3.20	ACTIVE BUZZER	40
3.21	NRF24L01 MODULE	40
3.22	ไอซี MCP23017 (16-BIT I/O EXPANDER WITH SERIAL INTERFACE)	41
4.1	สัญญาณเอาต์พุตจากจากเครื่องออสซิลโลสโคป	42
4.2	ค่าเอาต์พุตจาการ์ดูอิโน	43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่		หน้า
4.3	ผลการแจ้งเตือนจาการ์ดูอิโน	43
4.4	สัญญาณเอาต์พุตจากเครื่องออสซิลโลสโคป	44
4.5	ค่าเอาต์พุตจาการ์ดูอิโน	45
4.6	ผลการแจ้งเตือนจาการ์ดูอิโนเมื่อเกิดเพลิงไหม้เพียงห้องเดียว	46
4.7	ผลการแจ้งเตือนจาการ์ดูอิโนเมื่อเกิดเพลิงไหม้หลายห้อง	46
4.8	การต่อวงจรเพื่อทดสอบระบบกับเซนเซอร์ 1 ตัว	47
4.9	การแจ้งเตือนผ่านจอมอนิเตอร์	47
4.10	สเปกตรัมของสัญญาณจากอุปกรณ์ NRF24L01	48
4.11	การแจ้งเตือนผ่านแอปพลิเคชันสำหรับผู้ใช้งานเมื่อเกิดเหตุเพลิงไหม้	49
4.12	การแจ้งเตือนผ่านแอปพลิเคชันสำหรับผู้ใช้งานเมื่อเหตุเพลิงไหม้หยุดลง	49
4.13	หน้าต่างการเข้าสู่โปรแกรม	50
4.14	หน้าต่างเข้าสู่โปรแกรมเมื่อใส่ USER NAME หรือ PASSWORD ผิด	50
4.15	หน้าต่างการกรอกรายละเอียดของอาคาร	51
4.16	หน้าต่างการยืนยันรายละเอียดของอาคาร	51
4.17	หน้าต่างจอมอนิเตอร์แสดงข้อมูล	52
4.18	หน้าต่างจอมอนิเตอร์แสดงข้อมูลเมื่อไม่มีเหตุเพลิงไหม้	52
4.19	หน้าต่างจอมอนิเตอร์แสดงข้อมูลเมื่อมีเหตุเพลิงไหม้	53
4.20	ภาพชิ้นงานทางด้านฝั่งเครื่องส่ง	53
4.21	ภาพชิ้นงานทางด้านฝั่งเครื่องรับ	54
4.22	ภาพชิ้นงานของตัวเซนเซอร์ที่ติดอาคารภาพ	54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่

2.1 ชนิดของตัวแปรในภาษาซี 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เดิมระบบแจ้งเตือนการเกิดเหตุเพลิงไหม้นั้น สามารถบอกได้เพียงว่าภายในอาคารนั้นเกิดเหตุเพลิงไหม้หรือไม่ โดยหากมีเปลวไฟความร้อนหรือควันไฟ ระบบจะทำได้เพียงทำการแจ้งเตือนผ่านสัญญาณเตือนภัยว่าเกิดเหตุเพลิงไหม้ แต่ไม่สามารถระบุตำแหน่งต้นเพลิงได้ว่าเกิดที่ตำแหน่งใด จึงหาวิธีการระบุตำแหน่งต้นเพลิงนั้น เพื่อจะได้แก้ไขได้อย่างรวดเร็วและสะดวกขึ้น ที่ผ่านมามีการศึก Automated Fire Detection and Controlling System [5]. และ An Intelligent Fire Detection and Mitigation System Safe from Fire [4].

ระบบแจ้งเตือนการเกิดเพลิงไหม้สำหรับอาคารขนาดใหญ่ ทำขึ้นเพื่อแก้ปัญหาเรื่องการเกิดเพลิงไหม้ ให้สามารถหาตำแหน่งต้นเพลิงได้ง่ายและรวดเร็ว เมื่อเกิดเพลิงไหม้ระบบจะเตือนภัยผ่านสัญญาณเสียง และแสดงผลไปยังส่วนควบคุมกลาง โดยระบบจะทำงานโดยใช้ Infrared IR Flame Detector Sensor Module ซึ่งเป็นเซนเซอร์ตรวจจับเปลวไฟโดยใช้คลื่นอินฟราเรด เมื่อเซนเซอร์สามารถตรวจจับเพลิงไหม้ได้ ไมโครคอนโทรลเลอร์ที่คอยควบคุมจะทำการค้นหาตำแหน่งต้นเพลิงไหม้ จากนั้นจะส่งสัญญาณเสียงเตือนภัยผ่านตัว Buzzer และแสดงตำแหน่งต้นเพลิงภายในอาคาร ไปยังจอมอนิเตอร์ของส่วนควบคุมกลาง นอกจากนี้ผู้ใช้งานหรือผู้ที่อยู่ภายในอาคารสามารถตรวจสอบสถานและ ตำแหน่งของการเกิดเหตุเพลิงไหม้ผ่านแอปพลิเคชัน

1.2 วัตถุประสงค์

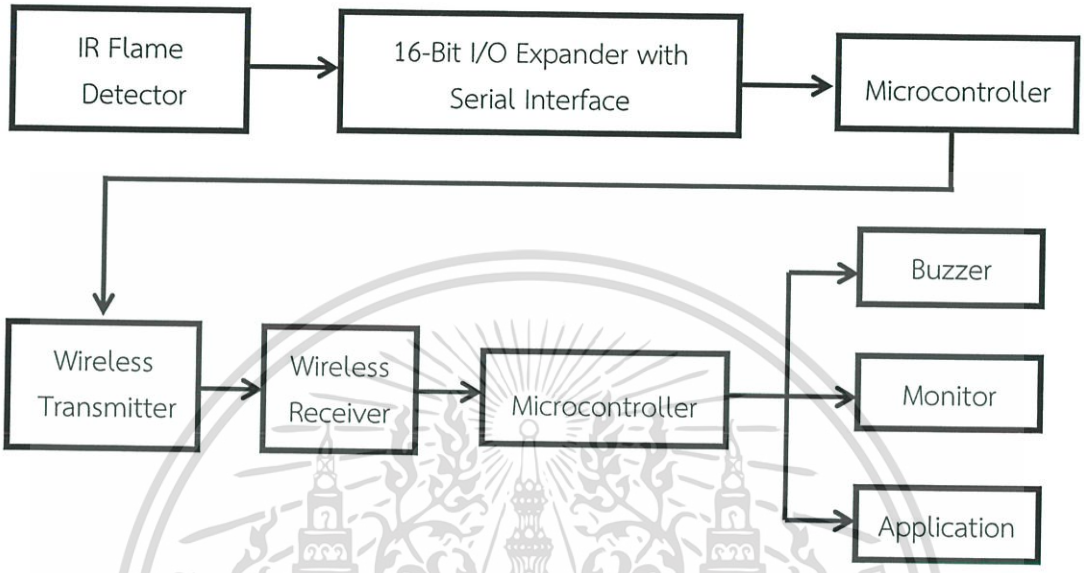
1) เพื่อศึกษามาตราฐานระบบแจ้งเตือนการเกิดเพลิงไหม้
2) เพื่อศึกษาการสร้างและพัฒนาระบบแจ้งเตือนการเกิดเพลิงไหม้

1.3 ขอบเขตของปริญญานิพนธ์

1) ออกแบบและสร้างระบบแจ้งเตือนการเกิดเพลิงไหม้ภายในอาคาร
2) สร้างระบบเตือนภัย โดยส่งสัญญาณเสียงแจ้งเตือนผ่านตัว Buzzer
3) สร้างระบบเตือนภัย โดยบอกตำแหน่งต้นเพลิงภายในอาคารไปยังส่วนควบคุมกลาง
4) สร้างระบบเตือนภัยโดยผู้ใช้งานหรือผู้ที่อยู่ภายในอาคารสามารถตรวจสอบสถานะและตำแหน่งของการเกิดเหตุเพลิงไหม้ผ่านแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บล็อกไดอะแกรมของโครงงานที่นำเสนอ



รูปที่ 1.1 บล็อกไดอะแกรมของโครงงาน

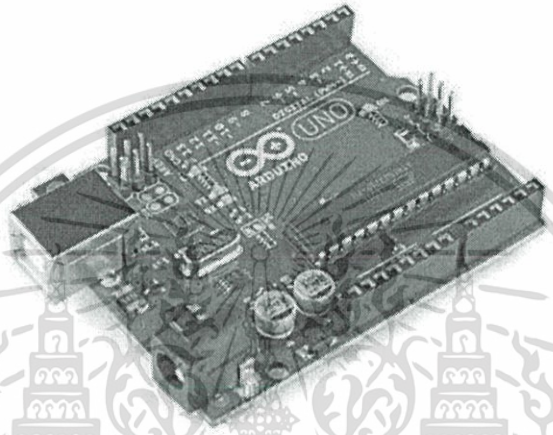
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 ไมโครคอนโทรลเลอร์

2.1.1 บอร์ดอาร์ดูอิโน (Arduino Board)



รูปที่ 2.1 บอร์ดอาร์ดูอิโน (Arduino Board)

บอร์ดอาร์ดูอิโน เป็นบอร์ดไมโครคอนโทรลเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open Source คือ มีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software ตัวบอร์ดอาร์ดูอิโนถูกออกแบบมาให้ใช้งานได้ง่าย และ สามารถใช้ต่ออุปกรณ์เสริมต่างๆ ดังรูปที่ 2.1 โดยผู้ใช้สามารถต่อวงจรอิเล็กทรอนิกส์จากภายนอกแล้วเชื่อม ต่อมาที่ขา I/O ของบอร์ด หรือเลือดต่อกับบอร์ดเสริมประเภทต่างๆได้ จุดเด่นของบอร์ดอาร์ดูอิโน (Arduino Board)

1. ง่ายต่อการพัฒนา มีรูปแบบค าสั่งพื้นฐานไม่ซับซ้อน
2. Arduino Community กลุ่มคนที่ร่วมกันตั้งใจพัฒนา
3. Open Hardware ไปต่อยอดใช้งานได้หลายด้าน
4. ราคาไม่แพง
5. Cross Platform สามารถพัฒนาโปรแกรมบน OS ใดก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 คำสั่งโปรแกรมของอาร์ดูอิโน

จะใช้ Library ของภาษา C/C++ ในการเขียนโปรแกรม ซึ่งภาษา C++ ดังนี้

1. คำสั่ง void setup หน้าที่ของฟังก์ชัน setup() ในอาร์ดูอิโนคือ ใช้ทำหน้าที่เป็นส่วนของโปรแกรมย่อย สำหรับใช้บรรจุ คำสั่งต่างๆ ที่ใช้สำหรับการกำหนดการทำงานของระบบหรือกำหนดคุณสมบัติของการทำงานให้กับอุปกรณ์ ต่างๆซึ่งคำสั่งทั้งหมดที่บรรจุไว้ภายใต้ฟังก์ชันของ setup() นี้ จะถูกเรียกขึ้นมาทำงานเพียงรอบเดียวคือตอน เริ่มต้นการทำงานของโปรแกรม (หลังการรีเซ็ตให้ไมโครคอนโทรลเลอร์เริ่มต้นทำงาน) เท่านั้น โดยคำสั่งที่นิยม บรรจุไว้ในฟังก์ชันส่วนนี้ได้แก่ คำสั่งกำหนดโหมดการทำงานของ Digital Pin หรือคำสั่งสำหรับกำหนดคุณสมบัติของพอร์ตสื่อสารอนุกรม เป็นต้น

2. คำสั่ง void loop หน้าที่ของฟังก์ชัน loop() ในอาร์ดูอิโนคือ ใช้ทำหน้าที่เป็นส่วนของโปรแกรมหลัก สำหรับใช้บรรจุ คำสั่งควบคุมการทำงานต่างๆ ของโปรแกรมที่ต้องการใช้โปรแกรมทำงาน โดยคำสั่งที่บรรจุไว้ในฟังก์ชันนี้จะถูก เรียกขึ้นมาทำงานซ้ำๆ กันตามลำดับและเงื่อนไขที่กำหนดไว้

3. คำสั่ง if...else คำสั่ง if...else ใช้สำหรับการตรวจสอบเงื่อนไขที่มีหลายทางเลือกเพิ่มขึ้นอีก 1 ทางเลือก คำสั่ง if...else แบบ 2 ทางเลือก มีความหมายว่า ถ้าเงื่อนไขเป็นจริงให้ทำตามเงื่อนไขของ if ถ้าเงื่อนไขเป็นเท็จให้ทำตามเงื่อนไขของ else จะเห็นได้ว่าโปรแกรมจะมีทางเลือกในการทำงานที่เพิ่มขึ้นมากกว่า 1 ทาง รวมเป็น 2 ทาง โดยทางเลือกแรกเป็นทางเลือกที่โปรแกรมจะทำงานเมื่อเงื่อนไขเป็นจริง ส่วนทางเลือกที่ 2 เป็นทางเลือก ที่จะให้โปรแกรมทำงานเมื่อเงื่อนไขเป็นเท็จ

4. คำสั่ง Serial.print (data) และ Serial.println (data) คำสั่งนี้ใช้ทำหน้าที่สำหรับสั่งให้ข้อมูลออกไปยังพอร์ตสื่อสารอนุกรมอย่างต่อเนื่องในลักษณะของการ แสดงข้อมูล ตัวแปรหรือข้อความต่างๆ ที่กำหนดไว้ในพารามิเตอร์ภายในวงเล็บ () ออกทางพอร์ตสื่อสารอนุกรม (พร้อมแสดงผลบนหน้าจอ หากบอร์ดอาร์ดูอิโนต่ออยู่กับคอมพิวเตอร์) กลุ่มคำสั่ง Serial.print และ Serial.println โดยทั้ง 2 กลุ่มคำสั่งจะมีรูปแบบการใช้งานและให้ผลการ ทำงานที่เหมือนกัน แต่ต่างกันที่ Serial.println จะมีการเติมค่ารหัสขึ้นบรรทัดใหม่ (0x0D, 0x0A) ปิดท้ายการ ทำงานของคำสั่งไปด้วย แต่ส่วนอื่นๆจะเหมือนกันหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 ชนิดและประเภทของตัวแปร

ชนิดและประเภทของตัวแปรในภาษาซีแบ่งเป็น 5 ชนิด แสดงดังตารางที่

1. คำสั่ง Char ใช้เก็บข้อมูลที่เป็นตัวอักษร (Character) เก็บข้อมูลที่เป็นเลข จำนวนเต็มได้ 256 ค่า

2. คำสั่ง int ใช้เก็บข้อมูลที่เป็นเลขจำนวนเต็ม (Integer) เก็บข้อมูลที่เป็นเลขจำนวนเต็มได้ 6,553 ค่า

3. คำสั่ง Float ใช้เก็บข้อมูลที่เป็นเลขทศนิยมแบบ Single Precision

4. คำสั่ง Double ใช้เก็บข้อมูลที่เป็นเลขทศนิยมแบบ Double Precision ซึ่งสามารถเก็บค่าตัวเลข ทศนิยมที่มีความละเอียดและถูกต้องมากกว่าแบบ float ถึง 2 เท่า

5. คำสั่ง Void ใช้เก็บตัวแปรที่ไม่มีค่า

ตารางที่ 2.1 ชนิดของตัวแปรในภาษาซี

ชนิดตัวแปร	จำนวนบิต	ค่าข้อมูลที่เก็บได้
char	8	-128 ถึง +127
int	16	-32,768 ถึง +32,767
float	32	3.4E-38 ถึง 3.4E+38
double	64	1.7E-308 ถึง +1.7E+308
void	0	0

2.1.4 รูปแบบการเขียนโปรแกรมบน Arduino

ขั้นตอนการใช้โปรแกรม เป็นดังนี้

1. เขียนโปรแกรมบนคอมพิวเตอร์ โดยผ่านทางโปรแกรม Arduino IDE ดังรูปที่ 2.2

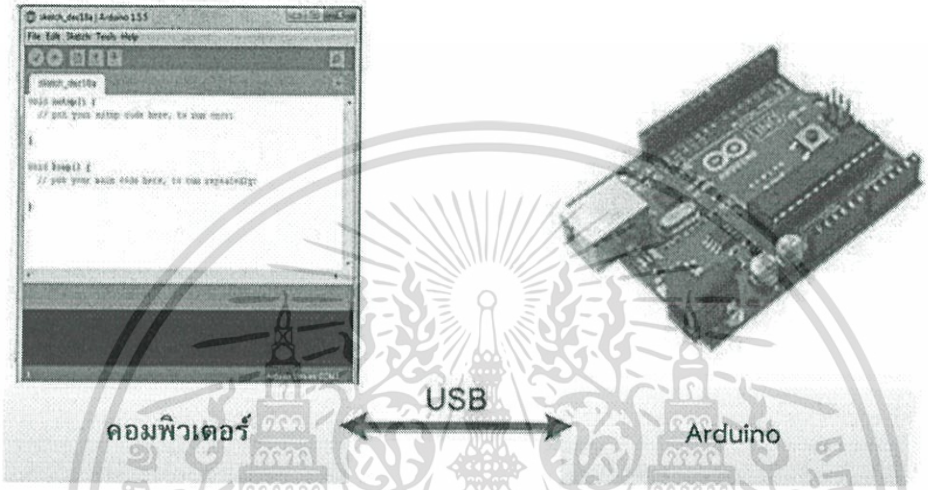
2. จากนั้นเขียนโค้ดโปรแกรมให้เรียบร้อย แล้วโหลดโปรแกรมลงบอร์ด Arduino ที่ใช้ ดังรูปที่ 2.3

3. หมายเลข Com port ดังรูปที่ 2.4

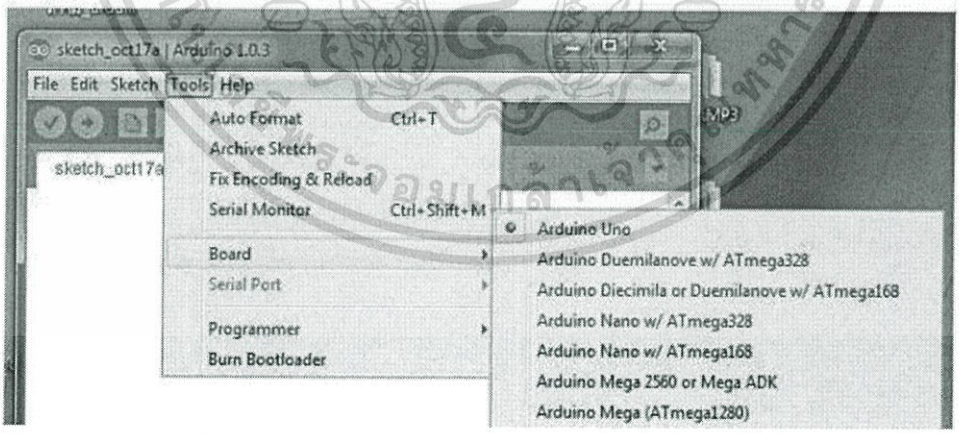
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. กดปุ่ม Verify เพื่อตรวจสอบความถูกต้องและ Compile โค้ดโปรแกรม จากนั้นกดปุ่ม Upload โค้ด โปรแกรมไปยังบอร์ดอาร์ดูอิโน ผ่านทางสาย USB ดังรูปที่ 2.5

5. เมื่อ Upload เรียบร้อยแล้ว จะแสดงข้อความแถบข้างล่าง “Done uploading”และบอร์ดจะเริ่ม ทำงานตามที่เขียนโปรแกรมไว้ได้ทันทีดังรูปที่ 2.6

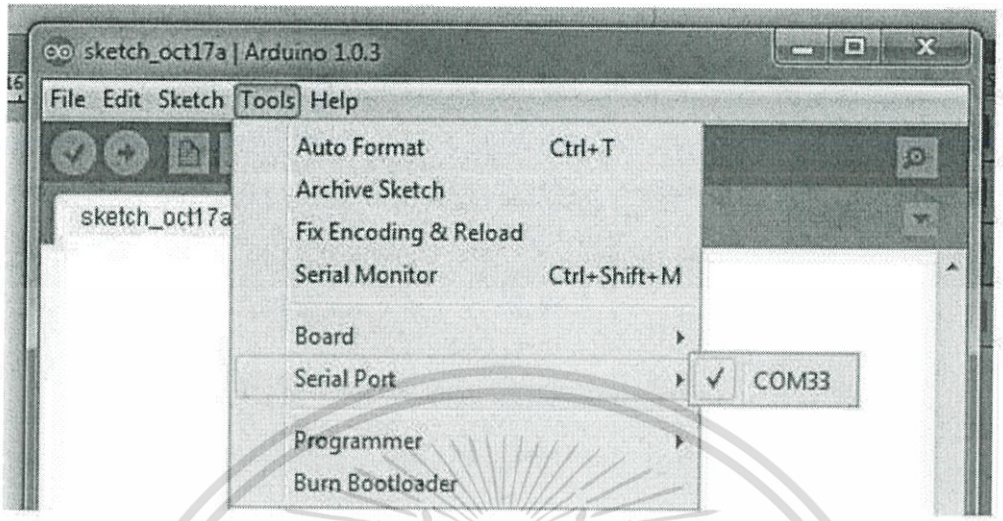


รูปที่ 2.2 การเขียนโปรแกรมบน Arduino

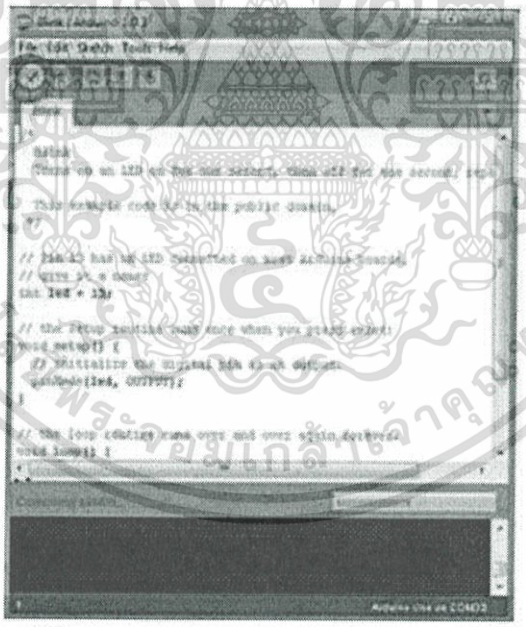


รูปที่ 2.3 เลือกรุ่นบอร์ด Arduino ที่ต้องการ upload

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

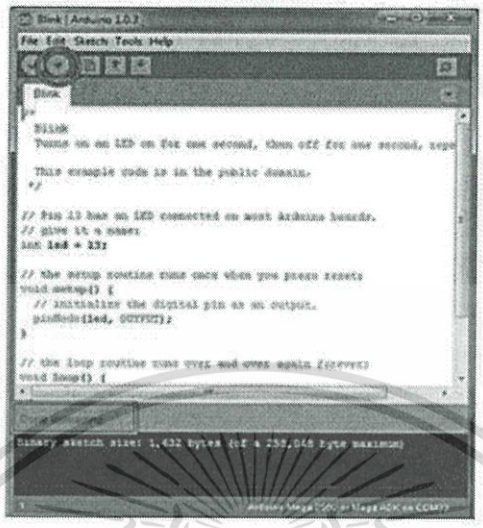


รูปที่ 2.4 เลือกหมายเลข Comport ของบอร์ด



รูปที่ 2.5 กดปุ่ม Verify เพื่อตรวจสอบความถูกต้องและ Compile โค้ดโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 Upload โค้ด โปรแกรม

2.2 การตรวจจับอัคคีภัย

2.2.1 หลักการตรวจจับอัคคีภัย

การตรวจจับอัคคีภัยเป็นมาตรการเฝ้าระวังอัคคีภัยโดยใช้เทคนิคทางกลตรวจจับความเปลี่ยนแปลงที่เกิดขึ้นระหว่างมีการลุกไหม้เป็นไฟ (Fire Development Stages) แล้วส่งสัญญาณออกมาเพื่อเป็นการเตือนภัย หรือสั่งการอย่างใดอย่างหนึ่งเพื่อยุติเหตุก่อนจะเกิดอัคคีภัยหรือการลุกลาม จุดประสงค์หลักของการตรวจจับอัคคีภัยคือกระตุ้นให้มีการตอบโต้สัญญาณเตือนที่เกิดขึ้น เช่น ดับเพลิงทั้งแบบปกติทั่วไปและ/หรือด้วยอุปกรณ์อัตโนมัติ อพยพคนออกจากพื้นที่นั้น ขนย้ายทรัพย์สิน หยุดการผลิต ฯลฯ และเพื่อให้เกิดความเข้าใจเรื่องของการ ตรวจจับอัคคีภัยดียิ่งขึ้นขออธิบายขั้นตอนการเกิดไฟซึ่งมีด้วยกัน 4 ขั้นตอนแสดงดังรูปที่ 2.7 ดังต่อไปนี้

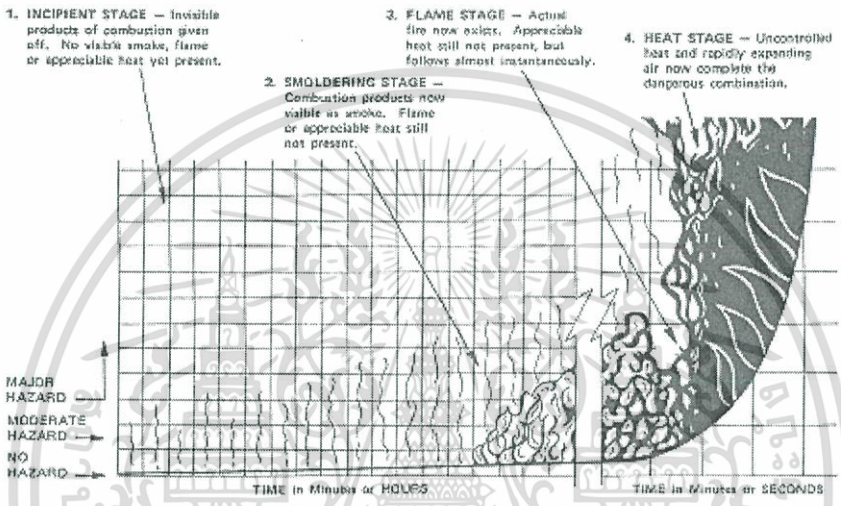
1. ขั้นเริ่มต้น (Incipient Stage) เริ่มมีการเผาไหม้ในขั้นแรกสุดแต่ไม่สามารถสังเกตผลผลิตของไฟ (Products of fire) ได้ไม่ว่าจะเป็นควัน เปลวไฟ หรือปริมาณความร้อนที่วัดค่าได้(Appreciable heat) ค่าอันตรายโดยเฉลี่ยจะอยู่ในระดับ “ไม่มีอันตราย” (No hazard)

2. ขั้นมีควัน (Smoldering Stage) เริ่มมีควัน แต่ยังไม่มีเปลวไฟหรือปริมาณความร้อนที่วัดค่าได้ ค่าอันตรายโดยเฉลี่ยจะอยู่ในระดับ“อันตรายปานกลาง” (Moderate hazard)

3. ขั้นมีเปลวไฟ (Flame Stage) เริ่มมีเปลวไฟทำให้มองเห็นว่าเป็นไฟแต่ยังไม่สามารถวัดค่าความร้อนได้ทว่าอุณหภูมิเริ่มสูงขึ้นเรื่อยๆค่าอันตรายโดยเฉลี่ยจะอยู่ในระดับ “อันตรายปานกลาง” (Moderate hazard) จนถึง “อันตรายมาก” (Major hazard)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ขั้นมีความร้อน (Heat Stage) มีความร้อนที่สามารถวัดค่าได้เกิดขึ้นแล้วและเริ่มมีการลุกลามจนไม่สามารถควบคุมได้ ค่าอันตรายโดยเฉลี่ยอยู่ในระดับ “อันตรายมาก” (Major hazard) ระยะห่างระหว่าง Incipient Stage และ Smoldering Stage จะกินเวลานับเป็นนาทีหรือนับเป็นชั่วโมงระยะห่างระหว่าง Flame Stage และHeat Stage จะกินเวลานับเป็นนาทีหรือวินาที



รูปที่ 2.7 แสดงขั้นตอนการเกิดการลุกไหม้ของอัคคีภัย (Fire Development Stages)

สำหรับไฟประเภท B บางชนิด Heat Stage จะเกิดขึ้นหลังจาก Incipient Stage อย่างกระชั้นชิด นั่นคือ นับตั้งแต่เริ่มมีการเผาไหม้ขั้นแรกสุดจนถึงการเกิดไฟลุกลามจะกินเวลาไม่ถึงหนึ่งนาทีตามแนวทางอุดมคติ การตรวจจับอัคคีภัยจะต้องทำให้ได้ในช่วงที่ไฟกำลังอยู่ในขั้นของ Smoldering Stage เพราะหลังจากนั้นแล้วแทบจะไม่มีผลอะไรเนื่องจากมีเวลาน้อยมากที่จะกระทำการตอบโต้อย่างได้ผล ไม่ว่าจะเป็นการดับไฟหรือการอพยพ พูดง่ายๆ ถ้าทางเลือกอยู่ที่การดับเพลิง เราจะต้องตัดวงจรของการเกิดไฟไว้ที่ขั้นของ Smoldering Stage อย่างน้อยที่วินาทีสุดท้ายของขั้นตอนนี้ก็ยังดีกว่าจะปล่อยให้ไฟไหม้ไปถึง Flame Stage และหากทางเลือกอยู่ที่การหนีไฟ เราก็จะต้องมีมาตรการตรวจจับและส่งสัญญาณเตือนให้มีการหนีก่อนที่จะมีการลุกลามไปถึงขั้น Flame Stage และ Heat Stage

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 อุปกรณ์ตรวจจับอัคคีภัย (Fire Detection Device)

อุปกรณ์ตรวจจับอัคคีภัยจัดเป็นอุปกรณ์เริ่มต้น (Initiating Device) ของระบบแจ้งเหตุเพลิงไหม้ทำงานร่วมกับอุปกรณ์หลักในระบบอีก 3 ส่วน ได้แก่ อุปกรณ์แจ้งเตือน (Notification Device) แหล่งจ่ายกำลังไฟฟ้า (Power Supply) ตู้ควบคุม(Control Panel) โดยทั่วไปแล้วอุปกรณ์เริ่มต้นจะประกอบด้วยอุปกรณ์ตรวจจับความร้อน ควันไฟ เปลวไฟ และอุปกรณ์ที่เป็นตัวกำเนิดสัญญาณเตือนภัยที่ติดตั้งอยู่กับอุปกรณ์อื่นๆ เช่น วาล์วประกอบสำหรับเครื่องสูบน้ำดับเพลิง วาล์วในระบบหัวกระจายน้ำดับเพลิงแบบอัตโนมัติ ฯลฯ

2.2.2.1 อุปกรณ์ตรวจจับความร้อน (Heat Detector)

อุปกรณ์ตรวจจับความร้อนเป็นอุปกรณ์เริ่มต้นในระบบการเตือนอัคคีภัยโดยทั่วไปแล้วมีอยู่ด้วยกัน 3 ชนิด คือ

1. อุปกรณ์ตรวจจับความร้อนแบบคงที่ (Fixed Temperature) จะทำงานตรวจจับอัคคีภัยเมื่ออุณหภูมิภายในพื้นที่ที่ติดตั้งสูงขึ้นถึงจุดที่ตั้งไว้ล่วงหน้าโดยที่ตัวอุปกรณ์อยู่ในตำแหน่งที่สัมผัสความร้อนนั้นโดยตรง

2. อุปกรณ์ตรวจจับความร้อนที่เพิ่มขึ้นจนถึงจุดที่ตั้งไว้ (Rate Compensate) จะทำงานตรวจจับอัคคีภัยเมื่ออุณหภูมิภายในพื้นที่ที่ติดตั้งสูงขึ้นถึงจุดที่ตั้งไว้ล่วงหน้าซึ่งอุปกรณ์สามารถตรวจจับความร้อนบริเวณรอบๆ จุดที่ติดตั้งห่างออกไปได้ในระยะที่กำหนดไว้ได้

3. อุปกรณ์ตรวจจับความร้อนที่เพิ่มขึ้นตามสัดส่วนที่ตั้งไว้ (Rate of Rise) จะทำงานตรวจจับอัคคีภัยเมื่ออุณหภูมิภายในพื้นที่ที่ติดตั้งสูงขึ้นแต่ไม่มีการตั้งไว้ล่วงหน้าว่าจะตรวจจับ ณ ที่อุณหภูมิใดอุณหภูมิหนึ่ง เป็นการตรวจจับเมื่อมีแนวโน้มว่าอุณหภูมิภายในพื้นที่นั้นขยับสูงขึ้นเรื่อยๆ ตามสัดส่วน (องศาต่อนาที) ปัจจุบันผู้ผลิตหลายรายได้เอาอุปกรณ์สองชนิดมารวมกันเป็นชุดเดียวเรียกว่าอุปกรณ์ตรวจจับความร้อนแบบรวม (Combination Heat Detector) ซึ่งมีการตรวจจับความร้อนแบบคงที่และแบบการเพิ่มของอุณหภูมิรวมอยู่ภายในอุปกรณ์เดียวกัน ทั้งนี้ ในการทำงานของอุปกรณ์แบบความร้อนคงที่นั้น อุปกรณ์จะทำงานเมื่อความร้อนถึงจุดที่กำหนดไว้ โลหะที่จับยึดที่จุด F จะเกิดการหลอมละลายซึ่งสปริงที่จุด G จะทำงานแล้วแกนจะเลื่อนตัวไปกระแทกกับจุด D ทำให้หน้าสัมผัสของจุด D และจุด E เชื่อมต่อถึงกันส่งผลให้ระบบเกิดการทำงาน สำหรับการทำงานแบบการเพิ่มอุณหภูมิความร้อนนั้น อุปกรณ์จะทำงานเมื่ออุณหภูมิภายในพื้นที่เพิ่มขึ้น 8 องศาเซลเซียสภายในเวลา 1นาที ซึ่งอากาศที่อยู่ภายในห้อง A จะเกิดการขยายตัวมีผลทำให้แผ่นไดอะแฟรมที่จุด C มีการเคลื่อนตัวขึ้นแล้วหน้าสัมผัสของจุด D กับจุด E จะสัมผัสกันและจะส่งสัญญาณกลับไปที่ระบบควบคุมต่อไปอุปกรณ์ตรวจจับความร้อนจะมีการเลือกค่าอุณหภูมิการทำงานเพื่อใช้ในการออกแบบติดตั้งในแต่ละพื้นที่ป้องกันที่แตกต่างกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2.2 อุปกรณ์ตรวจจับควันไฟ (Smoke Detector)

อุปกรณ์ตรวจจับควันไฟเป็นอุปกรณ์เริ่มต้นที่นิยมใช้กันมากที่สุด ทำการตรวจจับควันไฟที่เกิดขึ้นจากการเผาไหม้ซึ่งสามารถแยกได้เป็น 3 ประเภทตามวิธีการตรวจจับควันไฟ ต่อไปนี้

1. แบบประจุไฟฟ้า (Ionization Type) ทำงานโดยอาศัยการตรวจวัดการนำกระแสไฟฟ้าของประจุไฟฟ้า (Ion) ที่ปล่อยออกมาจากแหล่งกำเนิดภายในตัวอุปกรณ์นั้นอยู่ตลอดเวลา เมื่อมีควันไฟผ่านเข้ามาในห้องตรวจจับ (Detection Chamber) ของอุปกรณ์ ประจุไฟฟ้าจะไปเกาะติดกับอนุภาคของควันไฟ ทำให้ปริมาณกระแสไฟฟ้าที่ไหลผ่านภายในวงจรมีค่าลดลง รูปแบบการทำงานแสดงไว้ในรูป อุปกรณ์ชนิดนี้ ปกติจะใช้ติดตั้งภายในบริเวณที่มีไฟลุกไหม้อย่างรวดเร็วและมีควันแบบเจือจาง ไม่แนะนำให้ใช้นอกสถานที่หรือบริเวณที่ความชื้นสูง มีสภาพลมแรงมีฝุ่นหรือในครัวอาคาร

2. แบบพลังแสง (Photoelectric Type) อุปกรณ์ตรวจจับแบบพลังแสง แบ่งตามลักษณะการตรวจจับเป็นแบบจุด (Spot Type) และแบบต่อเนื่อง (Linear Type) รูปข้างบนแสดงลักษณะการทำงานโดยอาศัยการหักเหของแสงที่ปล่อยออกมาแหล่งกำเนิด (Photo cell) ไปกระทบกับอนุภาคของควัน แล้วตกไปที่อุปกรณ์รับแสง เมื่อมีควันลอยเข้ามาในช่องรับควันของอุปกรณ์ ไม่ว่าจะเป็นในลักษณะบังหรือเป็นเงาสะท้อนก็จะเกิดการเปลี่ยนแปลงของลำแสงดังกล่าวซึ่งจะกระตุ้นให้อุปกรณ์เกิดการทำงานขึ้น อุปกรณ์ชนิดนี้เหมาะสำหรับติดตั้งภายในอาคารเพื่อตรวจจับไฟที่มีควันเจือจางหรือมองไม่เห็นในขั้น Smoldering Stage แต่ไม่เหมาะที่จะติดตั้งนอกสถานที่หรือบริเวณที่มีฝุ่นละอองหนาแน่น

3. แบบตรวจสอบตัวอย่างอากาศ (Air Sampling) มีความละเอียดอ่อนและให้ผลในเชิงป้องกันสูง โดยจะมีการดูดอากาศในบริเวณติดตั้งอุปกรณ์ชนิดนี้เข้าห้องตรวจสอบอย่างต่อเนื่อง หากตัวอย่างอากาศที่ทำการทดสอบนั้นมีส่วนผสมของควันก็จะมีความหนาแน่นเกินค่ากำหนดที่ตั้งไว้ทำให้อุปกรณ์ตรวจจับก็จะเริ่มต้นทำงานทันที

2.2.2.3 อุปกรณ์ตรวจจับเปลวไฟ (Flame Detector)

อุปกรณ์ตรวจจับเปลวไฟ เป็นอุปกรณ์เริ่มต้นที่ทำการตรวจจับรังสีอินฟราเรด และรังสีอุลตราไวโอเลตที่เกิดจากเปลวไฟของเพลิงไหม้การเลือกอุปกรณ์ตรวจจับเปลวไฟเพื่อใช้ติดตั้งในพื้นที่ป้องกัน ควรปรึกษาผู้ผลิตในการเลือกประเภทของการตรวจจับ เพื่อให้เหมาะสมกับสภาพงานจริงๆ ที่จะทำการติดตั้ง การเลือกผิดประเภทจะทำให้การตรวจจับเปลวไฟมีความผิดพลาดและเกิดการแจ้งเตือนที่ผิดพลาดเช่นกันเท่าที่มีใช้งานอยู่ในปัจจุบัน อุปกรณ์ตรวจจับเปลวไฟมีอยู่ด้วยกัน 3 ชนิด คือ

1. Infrared Flame Detector สำหรับตรวจจับรังสีอินฟราเรด (IR) และแสงที่เกิดจากเปลวไฟในช่วงเวลา 3-5 วินาที นิยมใช้กันในบริเวณที่มีความเสี่ยงสูงต่อการเกิดเพลิงไหม้จากเชื้อเพลิงประเภทไฮโดรคาร์บอน เช่น น้ำมันเชื้อเพลิง ก๊าซแอลพีจี ฯลฯ แต่ไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เหมาะสมในการใช้ตรวจจับไฟจากเชื้อเพลิงประเภทโพลาร์โซลเว้นท์หรือประเภทก๊าซความดันสูง รวมทั้งไฟที่ลุกในขั้น Smoldering Stage

2. Ultraviolet Flame Detector สำหรับตรวจจับความยาวคลื่นของรังสีอุลตร้าไวโอเลต (UV) ที่เกิดจากเปลวไฟในช่วงเวลา 0.1 วินาทีเหมาะสำหรับการตรวจจับอัคคีภัยที่ลุกไหม้อย่างรวดเร็ว ติดตั้งได้ทั้งในและนอกสถานที่เทียบกับเครื่องตรวจจับรังสีอินฟราเรดแล้ว เครื่องตรวจจับรังสีอุลตร้าไวโอเลตสามารถตรวจจับได้เร็วกว่า แต่มีข้อจำกัดหลายประการเช่น ไม่สามารถทำงานได้อย่างมีประสิทธิภาพในบริเวณที่มีแสงวาบรบกวน บริเวณที่มีฝุ่นหรือสิ่งสกปรกต่างๆ ในอากาศ ในขณะที่เครื่องตรวจจับรังสีอินฟราเรดไม่มีปัญหาในเรื่องนี้

3. UV/IR Flame Detector เป็นอุปกรณ์แบบผสมเพื่อตรวจจับรังสีอินฟราเรดและอุลตร้าไวโอเวตพร้อมกัน (แยกคนละหัวจับรังสี) โดยที่ผลการตรวจจับจะต้องออกมาว่า กรณีนั้นมีทั้งรังสี UV และรังสี IR จึงจะตอบรับและแสดงผลเป็นอัคคีภัย หากตรวจจับได้เพียง UV หรือ IR อย่างใดอย่างหนึ่ง อุปกรณ์จะไม่ตอบรับและจะไม่ส่งสัญญาณเตือนออกไป นิยมใช้กันมากตามโรงกลั่นน้ำมัน แท่นขุดเจาะ สนามบิน โรงเก็บอากาศยาน ฯลฯ

2.3 Microsoft Visual Basic

2.3.1 ภาษา Visual Basic

Visual Basic หรือ VB เป็นภาษาโปรแกรมแบบ GUI สร้างโดยบริษัทไมโครซอฟท์ ภาษานี้เป็นหนึ่งในภาษาโปรแกรมยอดนิยมสำหรับโปรแกรมที่ใช้ในด้านธุรกิจ ภาษานี้พัฒนามาจากภาษาเบสิก และยังได้พัฒนาต่อเป็นภาษา VB.NET อีกด้วย วิชวลเบสิกสนับสนุน Rapid Application Development (RAD) ทั้งด้านการพัฒนาโปรแกรมประยุกต์แบบ graphical user interface (GUI), การเข้าถึงฐานข้อมูลโดยใช้การเชื่อมต่อแบบ DAO, RDO, หรือ ADO, และการสร้าง ActiveX control จุดเด่นอีกอย่างหนึ่งของวิชวลเบสิกคือนักเขียนโปรแกรมสามารถนำโปรแกรมประยุกต์หลาย ๆ โปรแกรมมารวมกันในโปรแกรมเดียว และยังสามารถประยุกต์ใช้คอมโพเนนต์ของวิชวลเบสิกที่มีเตรียมไว้ให้แล้วได้อีกด้วย

2.3.2 จุดเด่นของภาษา Visual Basic

เป็นเครื่องมือที่ช่วยในการพัฒนา Application บนระบบปฏิบัติการวินโดวส์ ทำได้โดยง่ายแม้มิใช่โปรแกรมเมอร์ก็สามารถสร้างโปรแกรมได้ในเวลาอันรวดเร็ว

1. มีโครงสร้างใกล้เคียงภาษามนุษย์ทำให้เรียนรู้ได้ง่าย
2. มีเครื่องมือในการพัฒนา Application จำนวนมาก
3. สามารถสร้างไฟล์ .EXE ที่สามารถทำงานได้ด้วยตัวเอง
4. ออกแบบการติดต่อกับผู้ใช้ (ฟอร์ม) ได้ทันที
5. พัฒนา Application ได้หลายแบบ เช่น โปรแกรมด้านธุรกิจ ด้านอินเตอร์เน็ต และ Web Application

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ง่ายต่อการเรียนรู้เหมาะสำหรับผู้เริ่มต้น ทั้งในเรื่องไวยากรณ์ของภาษาเองและเครื่องมือการใช้งาน

7. ความนิยมของตัวภาษา โดยอาจกล่าวได้ว่าภาษา Basic นั้นเป็นภาษาที่คนเรียนรู้และใช้งานมากที่สุดในประวัติศาสตร์ของคอมพิวเตอร์

8. การพัฒนาอย่างต่อเนื่อง การปรับปรุงประสิทธิภาพในด้านของตัวภาษาและความเร็วของการประมวลผล และในเรื่องของความสามารถใหม่ๆ เช่น การติดต่อกับระบบฐานข้อมูล การเชื่อมต่อกับเครือขายอินเตอร์เน็ต

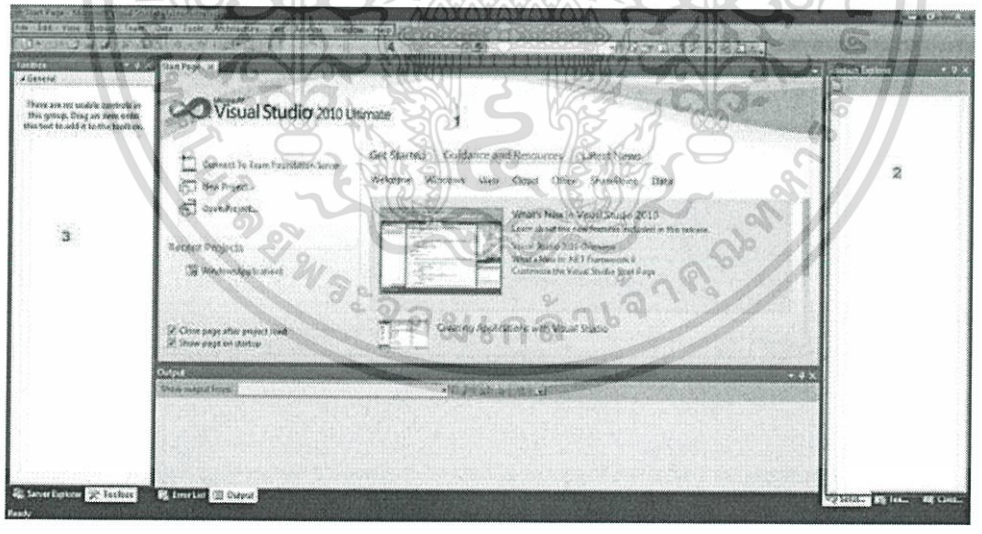
9. ผู้พัฒนาสำคัญของ Visual Basic คือบริษัทไมโครซอฟท์ซึ่งจัดว่าเป็นยักษ์ใหญ่ของวงการคอมพิวเตอร์ในปัจจุบัน เราจึงสามารถมั่นใจได้ว่า Visual Basic จะยังมีการพัฒนา ปรับปรุงและคงอยู่ไปอีกนาน

2.3.3 ส่วนประกอบ Visual Basic 2010

2.3.3.1 Start Page

เป็นหน้าต่างที่แสดงขึ้นมาตลอด เมื่อเปิดโปรแกรมประกอบด้วย

1. New Project ไว้สร้างโปเจคใหม่ในการเขียนโปแกรม
2. Open Project เปิดโปรเจค ที่เราบันทึกเอาไว้ กลับมาแก้ไขใหม่ได้
3. Recent Project จะแสดงรายชื่อโปรเจคที่เคยเปิดมาล่าสุด จำนวนหนึ่งสามารถทำให้เราเปิดโปรเจคได้รวดเร็ว ดังรูปที่ 2.8

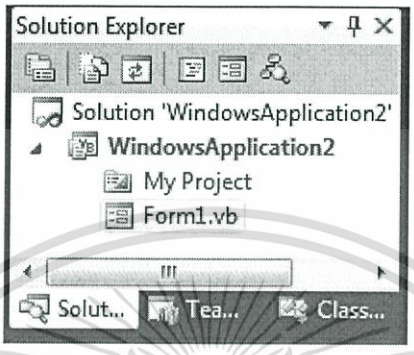


รูปที่ 2.8 หน้าต่าง Start Page

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

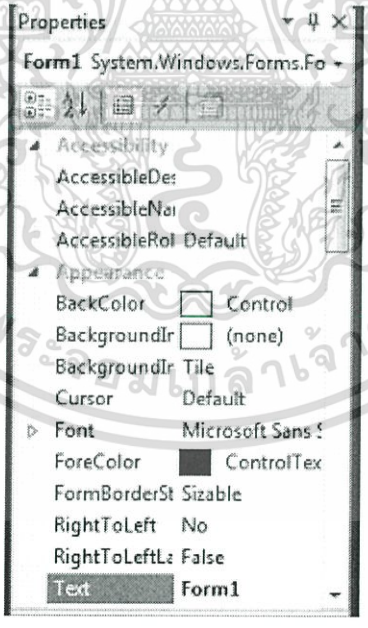
2.3.3.2 Solution Explorer และ Properties

1. Solution Explorer เป็นหน้าต่างแสดงโปรเจคประกอบด้วยอะไร หรือหมายถึง My Family ประกอบด้วยสิ่งที่สร้างขึ้นมา ดังรูปที่ 2.9



รูปที่ 2.9 หน้าต่าง Solution Explorer

2. Properties เป็นหน้าต่างแสดงคุณสมบัติของ Object ที่เราเลือกไว้แก้คุณสมบัติต่างๆ ดังรูปที่ 2.10



รูปที่ 2.10 หน้าต่าง Properties

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3.3 Toolbox

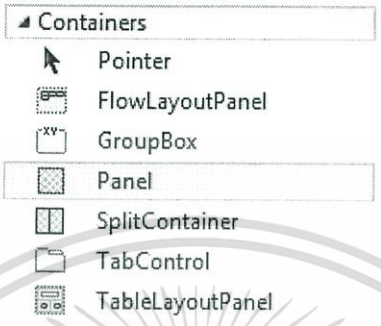
เป็นหน้าต่างที่รวมเครื่องมือต่างๆไว้สร้าง Application ไว้ออกแบบหน้าจอ

1. Common Control เป็นคอนโทรลพื้นฐานต่างๆ เช่น Checkbox, Label, List Box ดังรูปที่ 2.11

รูปที่ 2.11 หน้าต่าง Common Control

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Containers เป็นคอนโทรลที่ไว้จัดกลุ่มให้กลับคอนโทรลอื่นๆ โดยบรรจุคอนโทรลนั้นไว้ภายในตัวเดียวกัน เช่น Group Box, Panel ดังรูปที่ 2.12



รูปที่ 2.12 หน้าต่าง Containers

3. Components เป็นคอนโทรลที่ไม่แสดงรูปร่างออกมาทางฟอร์ม แต่จัดเตรียมทำงานให้กับโปรแกรม เช่น Timer, Process ดังรูปที่ 2.13

รูปที่ 2.13 หน้าต่าง Components

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3.4 Standard Toolbar

เป็นแถบเครื่องมือที่รวบรวมปุ่มต่างๆเอาไว้ ซึ่งปุ่มเหล่านี้จะเรียกใช้คำสั่งที่ใช้บ่อย เพื่อความสะดวกในการทำงาน เช่น ปุ่มแรกจะเทียบเท่ากับการสร้างโปรเจ็คใหม่ (New Project) ในเมนูไฟล์ เป็นต้น ดังรูปที่ 2.14

รูปที่ 2.14 หน้าต่าง Standard Toolbar

2.3.3.5 Menu Bar

เป็นส่วนที่รวบรวมคำสั่งทุกอย่างในการทำงานของ Visual Studio โดยแบ่งเป็นเมนูคำสั่งตามไอเท็มต่างๆ ดังรูปที่ 2.14

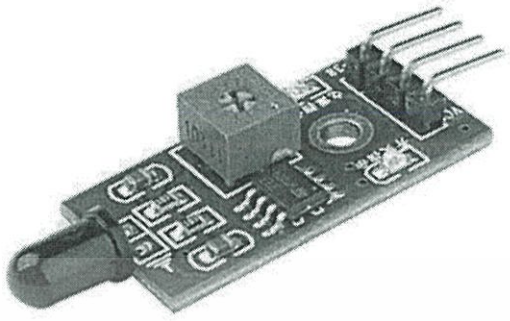
1. File คำสั่งที่ใช้สร้างโปรเจ็คใหม่ เปิดโปรเจ็ค ปิดโปรเจ็ค
2. Edit คำสั่งที่ใช้แก้ไข ตัด คัดลอก วาง ย้อนกลับ เป็นต้น
3. View คำสั่งที่ใช้แสดงเครื่องมือต่างๆของ Visual Studio
4. Properties ใช้ แก้ไขคุณสมบัติต่างๆ
5. Build คำสั่งที่ใช้คอมไพล์โปรเจค เป็นไฟล์ *.EXE
6. Debug ช่วยในการรันและตรวจสอบหาข้อผิดพลาด
7. Data คำสั่งที่ใช้ติดต่อกับฐานข้อมูล
8. Format คำสั่งที่ใช้จัดตำแหน่งของออบเจ็ค
9. Tool คำสั่งที่ใช้เรียกเครื่องมือส่วนเสริม

รูปที่ 2.15 หน้าต่าง Menu Bar

2.4 Infrared IR Flame Detector Sensor Module

Infrared IR Flame Detector Sensor Module หรือ เซนเซอร์ตรวจจับเปลวไฟด้วยอินฟราเรด ทำหน้าที่ในการตรวจจับเปลวไฟโดยใช้คลื่นอินฟราเรด แสดงผลการทำงานได้ทั้งในรูปดิจิตอล คือ ให้ค่า 0 หรือ 1 เพื่อแสดงค่าที่ตรวจจับได้ และในรูปอนาล็อก คือ ให้ค่าสัญญาณเพื่อใช้ประมวลผลต่อ การแสดงค่าเป็นดิจิตอลสามารถปรับค่า Threshold ได้โดยใช้ Potentiometer ที่มีให้บนบอร์ด ดังรูปที่ 2.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

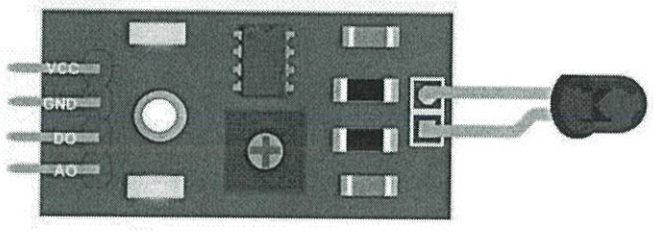


รูปที่ 2.16 Infrared IR Flame Detector Sensor Module

2.4.1 คุณสมบัติของเซนเซอร์

1. สามารถตรวจจับเปลวไฟหรือความยาวคลื่นในช่วง 760 นาโนเมตร ถึง 1,100 นาโนเมตร สามารถตรวจจับเปลวไฟในระยะการทดสอบประมาณ 80 เซนติเมตร
2. สามารถตรวจจับเปลวไฟในมุมกว้างประมาณ 60 องศา
3. มีความไวต่อเปลวไฟ
4. การเปรียบเทียบสัญญาณค่อนข้างดี สัญญาณรบกวนต่ำ
5. มีความแม่นยำ และมีความไว
6. การทำงานใช้แรงดัน 3.3-5 โวลต์
7. การแสดงผลมีทั้งรูปแบบดิจิตอล คือ แสดงค่า 0 1 และแบบอะนาล็อก คือ แสดงค่าเป็นแรงดันไฟฟ้า
8. การเดินสายไฟและติดตั้งง่าย
9. มีขนาดเล็กประมาณ 3.2 เซ็นติเมตร x 1.4 เซ็นติเมตร
10. การใช้งานมีช่วงกว้างกว่า เมื่อเปรียบเทียบกับ LM393
11. หน้าที่ของขาเซนเซอร์ แสดงดังรูปที่ 2.16
    1) VCC คือ 3.3-5 โวลต์
    2) GND คือ กราวด์
    3) DO คือ แสดงผลในรูปแบบดิจิตอลเป็นค่า 0 และ 1
    4) AO คือ แสดงผลในรูปแบบอนาล็อก

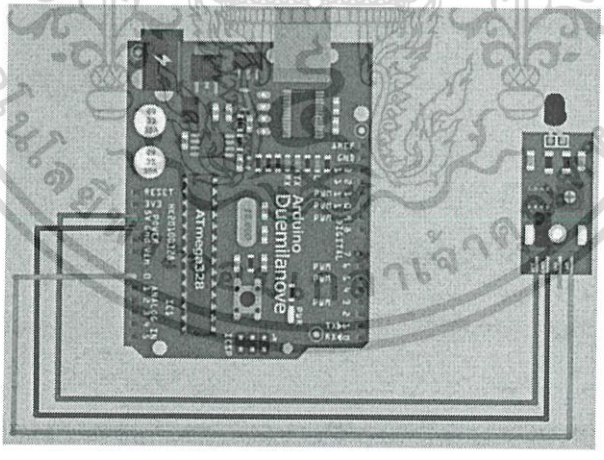
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.17 หน้าที่ของขาเซนเซอร์

2.4.2 การใช้งานของเซนเซอร์

1. เซนเซอร์มีความไวสูงต่อการตรวจจับเปลวไฟ สามารถตรวจจับเปลวไฟได้ดี เหมาะต่อการใช้เป็นเซนเซอร์ตรวจจับไฟไหม้ เพื่อแจ้งเตือน
2. สามารถเชื่อมต่อโดยตรงไปยังพอร์ตไมโครคอนโทรลเลอร์
3. การติดตั้งเซนเซอร์ควรมีระยะห่างจากความร้อน หรือเปลวไฟ ประมาณ 80 เซนติเมตร เพื่อที่จะไม่ให้เกิดความเสียหายต่อเซนเซอร์
4. การแสดงผลในรูปแบบอนาล็อก สามารถผ่านขั้นตอนการแปลงสัญญาณจากอนาล็อกเป็นดิจิตอล (AD) ได้โดยไม่มีการสูญเสีย มีความถูกต้องและแม่นยำสูง
5. มีการติดตั้งและใช้งานง่าย ดังรูปที่ 2.17

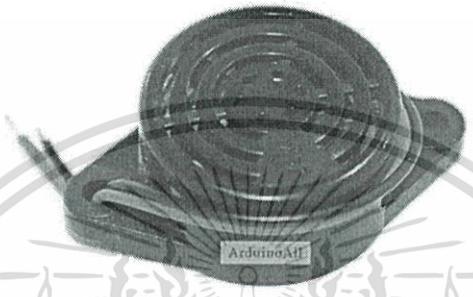


รูปที่ 2.18 การเชื่อมต่อ Infrared IR Flame Detector Sensor Module กับ Arduino

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 Active Buzzer

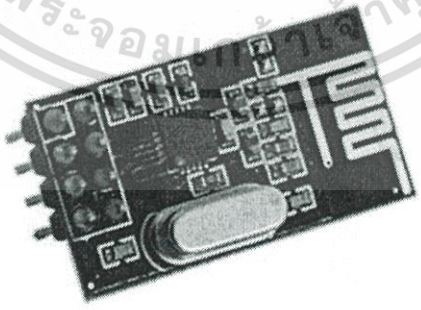
Active Buzzer เป็นลำโพงแบบแม่เหล็กไฟฟ้าหรือ เรียกอีกอย่างว่า แบบเปียโซเป็นลำโพงที่มีวงจรกำเนิดความถี่ (oscillator) อยู่ภายในตัวสามารถใช้แรงดันไฟฟ้ากระแสตรงได้ในช่วงกว้าง 3 - 24 โวลต์ เมื่อป้อนแรงดันสามารถกำเนิดเสียงได้ด้วยตัวเอง แต่ไม่สามารถเปลี่ยนความถี่ของเสียงได้ ดังรูปที่ 2.18



รูปที่ 2.19 Active Buzzer

2.6 NRF24L01 Module

โมดูล NRF24L01 เป็นโมดูลสื่อสารไร้สาย ที่สามารถเขียนโปรแกรมให้เป็นได้ทั้งตัวรับและตัวส่ง สามารถใช้กับ Arduino ได้หลาย ๆ ตัวพร้อมกัน มีความเร็ว 2.4GHz จึงสื่อสารได้รวดเร็วและไม่ต้องการเสาอากาศที่ยาว มีขนาดเล็กสะดวกในการต่อใช้งาน สามารถประยุกต์ใช้งานได้หลายอย่าง เช่น ใช้เป็นอุปกรณ์ส่งข้อมูลของเซนเซอร์อัตโนมัติสำหรับควบคุม อุณหภูมิ ความชื้น การแจ้งเตือนต่าง ๆ ควบคุมและติดตามหุ่นยนต์ Robot Control and Monitoring ได้ในระยะ 15-500 เมตร โมดูลนี้ใช้ชิฟ nRF24L01+ m ทำงานด้วยความเร็วสูง High-speed SPI interface ใช้พลังงานต่ำ รองรับการทำงานร่วมกับ Arduino ดังรูปที่ 2.19



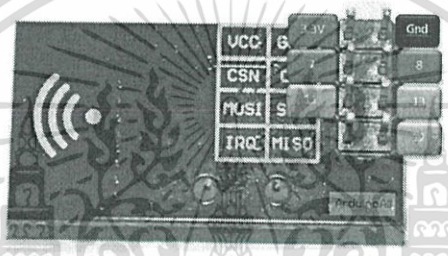
รูปที่ 2.20 NRF24L01 Module

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.1 คุณสมบัติของเซนเซอร์

การต่อขาโมดูล NRF24L01 กับ Arduino เป็นดังรูปที่ 2.20

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



รูปที่ 2.21 การต่อขาโมดูล NRF24L01 กับ Arduino

2.6.2 การสื่อสารข้อมูลด้วยระบบ SPI Bus

SPI Bus (Serial Peripheral Interface Bus) เป็นรูปแบบหนึ่งของการสื่อสารข้อมูลระหว่างอุปกรณ์แบบดิจิทัลที่พบเห็นได้บ่อย และใช้กับอุปกรณ์ได้มากกว่าสองขึ้นไป และนำมาต่อกันเป็นบัส (Bus) บัส SPI ส่งและรับข้อมูลทีละบิต (Bit Serial) และใช้สัญญาณ Clock เป็นตัวกำหนดจังหวะการทำงาน (ดังนั้นจึงเรียกว่า Synchronous, Bit-Serial Data Communication) มีการกำหนดบทบาทในการทำงานของอุปกรณ์ในระบบบัส แบ่งเป็น SPI Master และ SPI Slave โดยที่ SPI Master เป็นฝ่ายเริ่มการสื่อสารข้อมูล และสร้างสัญญาณ Clock (มักใช้ชื่อสัญญาณว่า SCK) มากำหนดจังหวะการส่งและรับข้อมูล และด้าน SPI Slave จะเป็นฝ่ายคอยตอบสนอง และในระบบบัส SPI อาจมีอุปกรณ์ที่เป็น SPI Slaveได้มากกว่าหนึ่ง (Single-Master, Multi-Slave)

SPI ใช้สัญญาณ 4 เส้น (ใช้งานในแบบที่เรียกว่า 4-Wire SPI) ได้แก่

1. SCK (Serial Clock) เป็นสัญญาณ CLK ที่ถูกสร้างเป็น SPI Master
2. MOSI (Master-Out Slave-In) เป็นสัญญาณสำหรับส่งข้อมูลบิตออกจาก SPI Master ไปยัง SPI Slave
3. MISO (Master-In Slave-Out) เป็นสัญญาณสำหรับส่งข้อมูลบิตออกจาก SPI Slave ไปยัง SPI Master

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

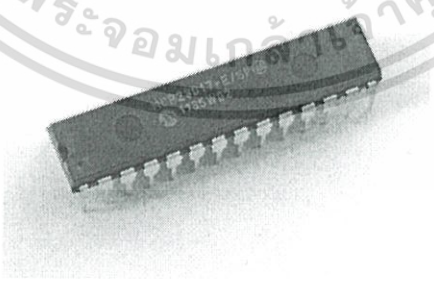
4. /SS (Slave Select, Active-Low) เป็นสัญญาณที่สร้างโดย SPI Master เพื่อใช้ระบุว่า ต้องการสื่อสารกับ SPI Slave หรือไม่ในกรณีที่มีอุปกรณ์ SPI Slave มากกว่าหนึ่งชุด จะต้องมีสัญญาณ Slave Select มากกว่าหนึ่งเส้น และแยกสำหรับแต่ละอุปกรณ์

เมื่อจะส่ง-รับข้อมูลผ่านบัส SPI (เรียกว่า SPI Data Transfer) สัญญาณ Slave Select (/SS) จะต้องเปลี่ยนจากHIGH เป็น LOW จากนั้นข้อมูลหนึ่งไบต์จะถูกเลื่อนบิตและส่งออกไปทีละบิตจาก SPI Master ตามจังหวะของ SCK และเลือกได้ว่าจะให้บิต MSB (Most-Significant Bit) หรือ LSB (Least-Significant Bit) ถูกส่งออกไปก่อน และในขณะเดียวกันก็จะรับข้อมูลทีละบิตจาก SPI Slave จนได้ครบหนึ่งไบต์ (หรือกล่าวได้ว่า Data Frame = 8 บิต) ดังนั้นเมื่อ SPI Master ส่งข้อมูลจำนวนหนึ่งไบต์ไปยัง SPI Slave ก็จะได้ข้อมูลหนึ่งไบต์จาก SPI Slave เช่นกัน ในช่วงเวลาที่สัญญาณ /SS เป็น LOW อาจมีการส่ง-รับข้อมูลได้มากกว่าหนึ่งไบต์ (Multi-byte SPI transfer)

การทำงานของ SPI มี 4 โหมด จำแนกตามพารามิเตอร์สองตัวที่เรียกว่า CPOL (Clock Polarity) และ CPHA (Clock Phase) ซึ่งจะเป็นตัวกำหนดลักษณะการทำงานอย่างเช่น จะส่ง-รับบิตที่ขอบขาขึ้นหรือลงของสัญญาณ CLK และสัญญาณ CLK จะอยู่ที่ลอจิก HIGH หรือ LOW เมื่อไม่อยู่ในช่วงของการส่งข้อมูลใดใดในบัส SPI (ช่วงที่เรียกว่า Idle) แต่โดยทั่วไปนิยมใช้ SPI Mode 0

2.7 MCP23017 (16-Bit I/O Expander with Serial Interface)

ไอซี MCP23017 ทำให้ขาที่มีอย่างจำกัดของ Arduino เพิ่มขึ้น 16 ขา โดยใช้ขาจาก Arduino เพียง 2 ขา เป็นรูปแบบการติดต่อแบบ I2C สามารถกำหนดแต่ละขาว่าจะให้เป็นอินพุตหรือเอาต์พุตก็ได้ ใช้ไฟช่วง 2.7-5.5 โวลต์ สามารถขับกระแสได้สูงสุด 20 มิลลิแอมป์ สามารถกำหนดอ้างอิง Address ได้ 8 ตำแหน่ง แปลว่าต่อเซ็นเซอร์รวมกันได้ 8 ตัวหรือ ใช้ขาจาก Arduino 2 ขา สามารถขยายขาได้สูงสุดถึง 128 I/O ดังรูปที่ 2.22



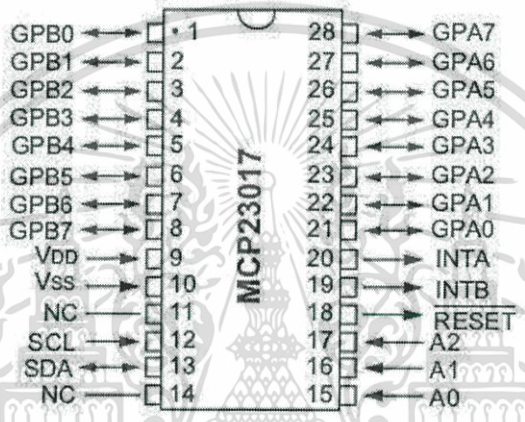
รูปที่ 2.22 ไอซี MCP23017 (16-Bit I/O Expander with Serial Interface)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.1 คุณสมบัติของไอซี MCP23017

การต่อขาไอซี MCP23017 เป็นดังรูปที่ 2.23

1. VDD คือ 2.7-5.5 โวลต์
2. VSS คือ กราวด์
3. ขา 1-8 คือ INPUT
4. ขา 21-28 คือ INPUT
5. ขา 15-17 คือ OUTPUT I



รูปที่ 2.23 ขาของไอซี MCP23017 (16-Bit I/O Expander with Serial Interface)

2.7.2 การสื่อสารข้อมูลด้วยระบบ I2C Bus

I2C Bus เป็นการสื่อสารแบบ Synchronous& Serial (หมายถึง การส่งข้อมูลทีละบิต และใช้สัญญาณ Clock ในการกำหนดจังหวะการส่งข้อมูล) ข้อดีของการสื่อสารข้อมูลแบบบัส I2C คือ ใช้สายสัญญาณเพียง 2 เส้น คือ SCL (สายสัญญาณ Serial Clock) และ SDA (สายสัญญาณข้อมูล Serial Data) และเป็นสัญญาณแบบ 2 ทิศทาง (Bidirectional) มีวงจรภายในสำหรับ I/O แบบ Open-Drain/Open-Collector (เวลาใช้งานต้องมีตัวต้านทานแบบ Pull-up Resistors ต่ออยู่ด้วย)

บัส I2C สามารถพ่วงอุปกรณ์ได้หลายอุปกรณ์ แต่ละอุปกรณ์จะมีหมายเลขที่อยู่ (Device Address) ที่ต้องไม่ซ้ำกัน โดยทั่วไปจะใช้หมายเลขที่อยู่ขนาด 7 บิต (7-bit Device Address) ซึ่งระบุได้ถึง 128 อุปกรณ์ หรือถ้ามีมากกว่านั้น จะเป็น 10 บิต (10-bit Device Address)

อุปกรณ์ที่ทำหน้าที่เป็น I2C Master จะเป็นฝ่ายเริ่มการสื่อสารข้อมูล และสร้างสัญญาณ SCL มาควบคุมจังหวะ มีอัตราการส่งข้อมูลอยู่ที่ 100kHz และ 400kHz (บางกรณี

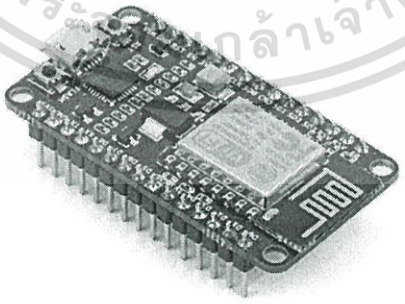
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้สูงกว่า 1MHz) เมื่อไม่มีการสื่อสารใดๆ สถานะลอจิกของ SCL และ SDA จะเป็น 1 หรือ HIGH เมื่อบัส I2C เริ่มต้นสื่อสาร อุปกรณ์ I2C Master จะส่งบิต Start (หรือเรียกว่า Start Condition) ตามด้วยการส่งไบต์ควบคุม (Control Byte) ออกไปก่อน ซึ่งจะเป็นการระบุหมายเลขของอุปกรณ์ Slave ที่อุปกรณ์ Master ต้องการจะสื่อสารด้วย และในไบต์ดังกล่าวจะมีบิตที่เรียกว่า Read/Write (R/W) Bit สำหรับระบุว่า จะเป็นการเขียนหรืออ่านข้อมูลต่อจากนั้น ถ้าเป็นบิตเขียน (R/W Bit = 0) อุปกรณ์ Master จะส่งข้อมูลไบต์ไปยังอุปกรณ์ Slave เท่านั้น แต่ถ้าเป็นบิตอ่าน (R/W Bit = 1) ต่อไปจะเป็นการรับข้อมูลไบต์จากอุปกรณ์ Slave เพียงอย่างเดียว นอกจากนั้นในการรับส่งข้อมูลแต่ละไบต์ ฝ่ายรับจะต้องทำการส่งบิตที่เรียกว่า ACK (Acknowledge) Bit ซึ่งจะต้องเป็นลอจิก 0 (ดึงสัญญาณ SDA ลง GND) เมื่อ SCL เป็น 1 เพื่อแจ้งให้ฝ่ายส่งทราบว่า ได้รับข้อมูลไบต์แล้วและพร้อมจะทำงานต่อไป ถ้าจบการสื่อสาร ก็จะต้องส่งบิต Stop (หรือเรียกว่า Stop Condition)

การเขียนโปรแกรมสำหรับ Arduino เพื่อใช้สื่อสารข้อมูลผ่านบัส I2C ก็ทำได้ไม่ยาก เพราะสามารถเรียกใช้คำสั่งจากไลบรารี่ของ Arduino (ใช้ซอฟต์แวร์ตามเวอร์ชัน 1.0.x) ที่ชื่อว่า Wire และสามารถใช้งานได้ทั้งกรณี I2C Master หรือ I2C Slave ความเร็วในการรับส่งข้อมูลจะอยู่ที่ 100kHz (default) และมีการใช้งานตัวต้านทาน pull-up ที่อยู่ภายในชิปไมโครคอนโทรลเลอร์ของ Arduino อีกด้วย

2.8 Node MCU

ESP8266 เป็นโมดูลเชื่อมต่อ WiFi ที่ผลิตจากจีน มีขนาดเล็กมากๆ และราคาถูก เป็นบอร์ดที่รวมเอา ESP8266 (ESP-12) + USB to Serial + NodeMCU firmware เข้าไว้ด้วยกัน ทำให้การใช้งานง่ายมากขึ้น ไม่ต้องมีอุปกรณ์ต่อพ่วงเยอะ มี GPIO เพิ่มเป็น 10 พอร์ท เพียงพอต่อการใช้งาน สำหรับ NodeMCU devkit ที่ได้มาเป็นบอร์ดเปล่าๆ ใช้ AT Command ในการสั่งงานชีวิตไม่อิสระเท่าไร แต่ทางผู้พัฒนา Board ก็มี NodeMCU Firmware ให้คุณสามารถเขียน Lua ใส่ลงไปได้ ทำให้การใช้งานง่ายมากขึ้น ดังรูปที่ 2.24

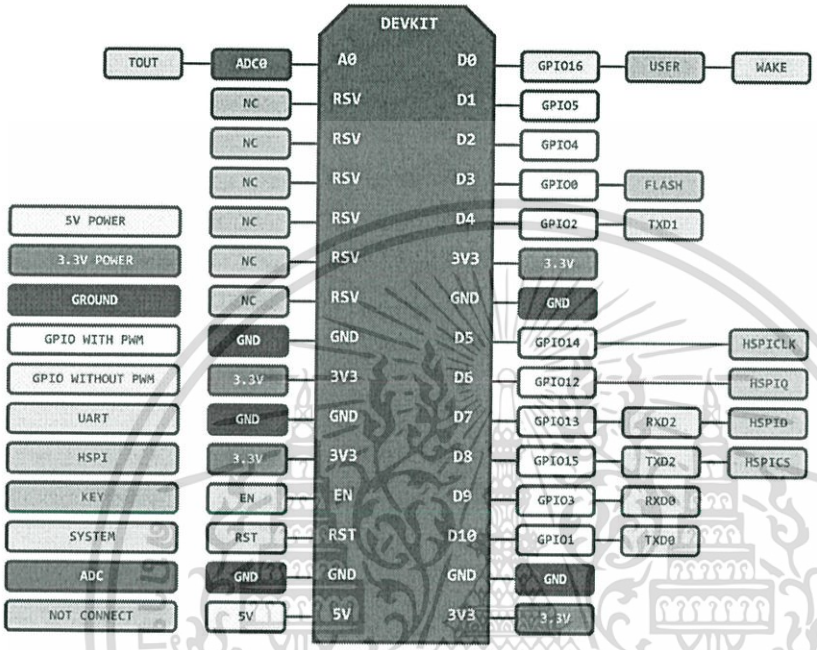


รูปที่ 2.24 Node MCU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.1 คุณสมบัติของ Node MCU

การต่อขา Node MCU เป็นดังรูปที่ 2.25



รูปที่ 2.25 ขาของ Node MCU

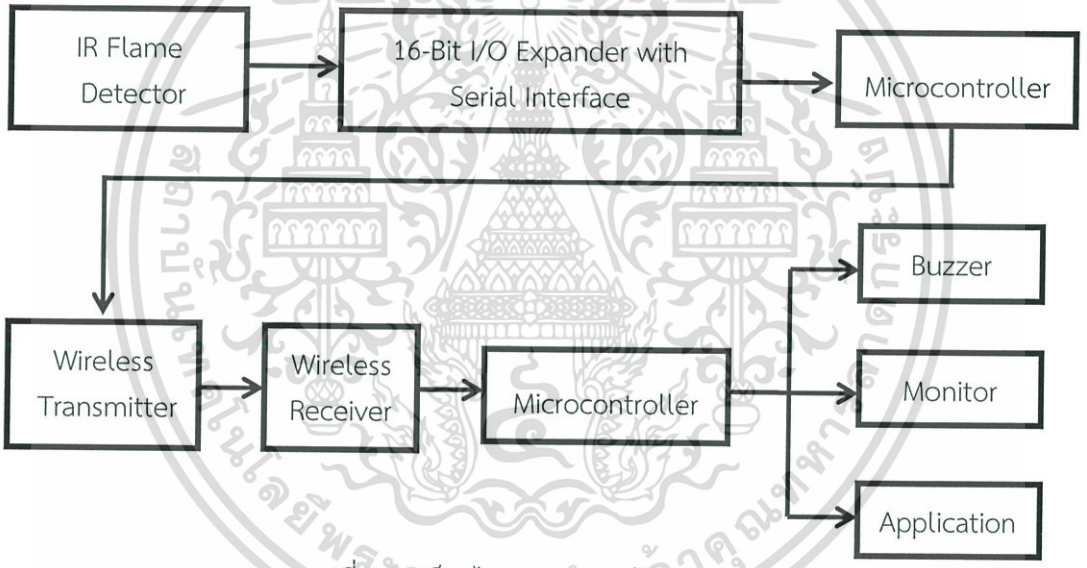
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบและการจัดทำปริญญานิพนธ์

3.1 การออกแบบ

3.1.1. การออกแบบบล็อกไดอะแกรมของระบบ

ระบบแจ้งเตือนการเกิดเพลิงไหม้สำหรับอาคารขนาดใหญ่ ทำงานโดยใช้ Infrared IR Flame Detector Sensor Module เชื่อมต่อกับ 16- Bit I/O Expander เมื่อเซนเซอร์ตรวจจับเพลิงไหม้ได้ ไมโครคอนโทรลเลอร์ที่คอยควบคุมจะทำการค้นหาตำแหน่งต้นเพลิงไหม้ จากนั้นจะส่งสัญญาณข้อมูลแบบไร้สาย จากนั้นไมโครคอนโทรลเลอร์ฝั่งรับจะทำการประมวลผลเพื่อทำการแจ้งเตือนผ่านตัว Buzzer และแสดงตำแหน่งต้นเพลิงภายในอาคารไปยังจอมอนิเตอร์ในห้องควบคุมกลาง รวมถึงแอปพลิเคชันจากโทรศัพท์มือถือของผู้ใช้งาน ดังรูปที่ 3.1



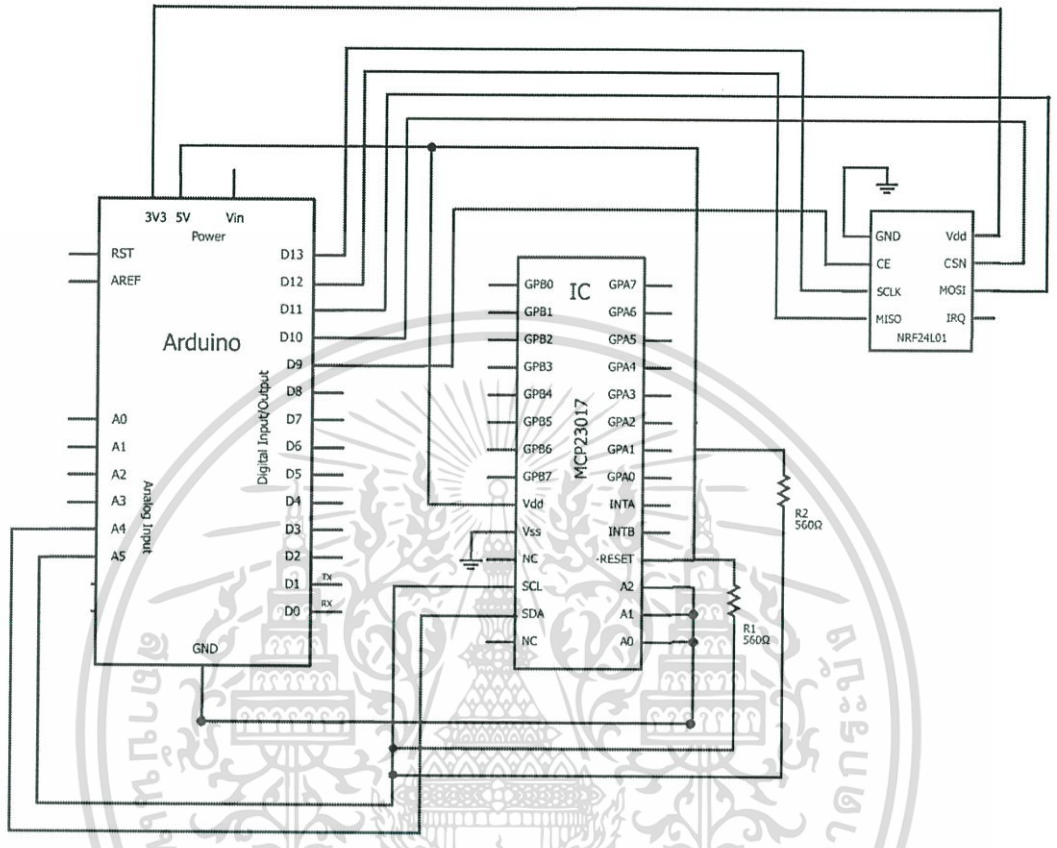
รูปที่ 3.1 บล็อกไดอะแกรมของโครงงาน

3.1.2. การออกแบบส่วนของระบบเซนเซอร์

3.1.2.1 วงจรรวมของระบบฝั่งเครื่องส่ง

หลักการทำงานของฝั่งเครื่องส่ง มีการทำการประมวลผลผ่าน Arduino โดยใช้ MCP23017 เป็นตัวขยายพอร์ตดิจิตอล เพื่อเชื่อมต่อกับเซนเซอร์ Infrared IR Flame Detector ที่ใช้ตรวจจับเพลิงไหม้ จากนั้นจะรับส่งข้อมูลผ่าน NRF24L01 ซึ่งเป็นอุปกรณ์ที่ใช้ส่งข้อมูลผ่านสัญญาณความถี่วิทยุ 2.4 GHz ไปยังฝั่งเครื่องรับ ดังรูปที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

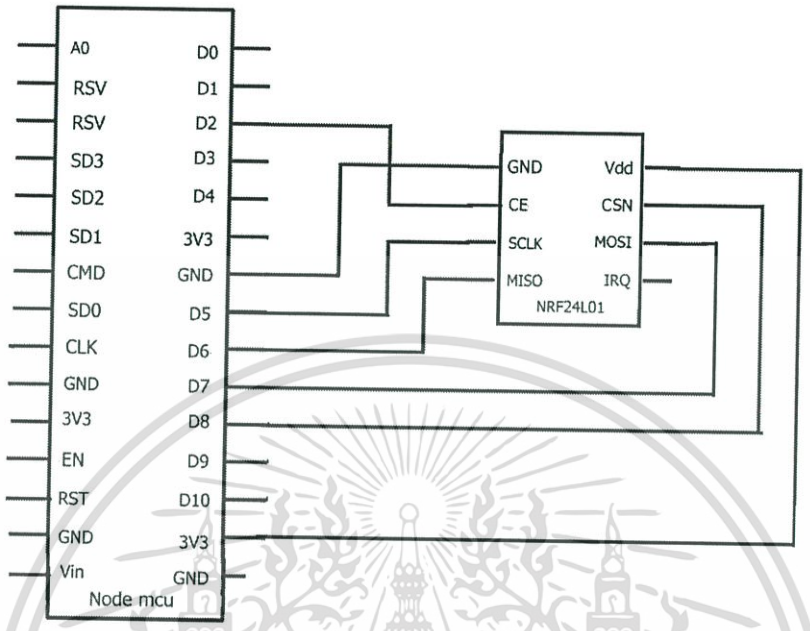


รูปที่ 3.2 วงจรรวมของระบบฝั่งเครื่องส่ง

3.1.2.2 วงจรรวมของระบบฝั่งเครื่องรับ

หลักการทำงานของฝั่งเครื่องรับ มีการทำการประมวลผลผ่าน NodeMCU เพื่อเชื่อมต่อ WI-FI และ รับข้อมูลผ่าน NRF24L01 ซึ่งเป็นอุปกรณ์ที่ใช้รับส่งข้อมูลผ่านสัญญาณความถี่วิทยุ 2.4 GHz จากนั้นทำการแจ้งเตือนแบบสัญญาณเสียงผ่านตัว Buzzer และแสดงตำแหน่งต้นเพลิงภายในอาคารผ่าน Serial port ไปยังโปรแกรมเพื่อแสดงขึ้นจอมอนิเตอร์ในห้องควบคุมกลาง และส่งข้อมูลผ่าน WI-FI ไปยังแอปพลิเคชันจากโทรศัพท์มือถือของผู้ใช้งาน แสดงดังรูปที่ 3.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 วงจรรวมของระบบฝั่งเครื่องรับ

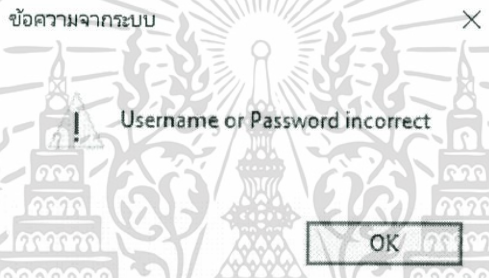
3.1.3 การออกแบบหน้าจอมอนิเตอร์สำหรับผู้ใช้งานระบบ

3.1.3.1 หน้าล็อกอิน

เริ่มแรกเมื่อทำการเปิดมอนิเตอร์ขึ้นมา ระบบจะแสดงหน้าล็อกอิน เพื่อทำการเข้าสู่ระบบ ดังรูปที่ 3.4 เมื่อทำการเข้าสู่ระบบแล้ว จอมอนิเตอร์จะแสดงหน้าหลัก เพื่อรอรับค่าต่างๆ หากทำการกรอกรหัสผิดจะไม่สามารถเข้าสู่ระบบได้ โดยมีข้อความแจ้งเตือนเพื่อให้กรอกรหัสใหม่อีกครั้ง ดังรูปที่ 3.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

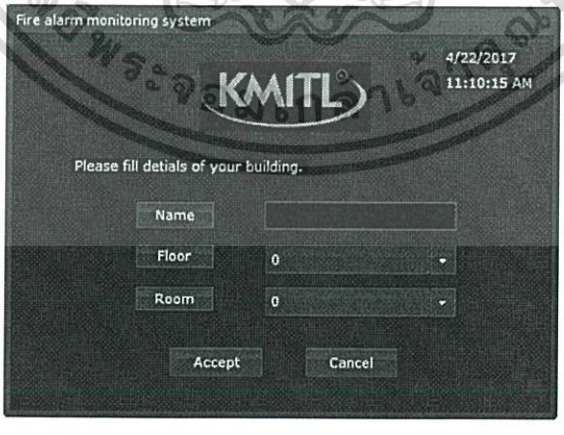
รูปที่ 3.4 หน้าล็อกอิน



รูปที่ 3.5 ข้อความแจ้งเตือนเมื่อกรอกรหัสผิด

3.1.3.2 การกรอกรายละเอียดของอาคาร

เมื่อทำการเข้าสู่ระบบแล้ว จะมีช่องให้กรอกชื่อของอาคาร รายละเอียดของอาคาร เช่น จำนวนชั้น จำนวนห้องในแต่ละชั้น ดังรูปที่ 3.6

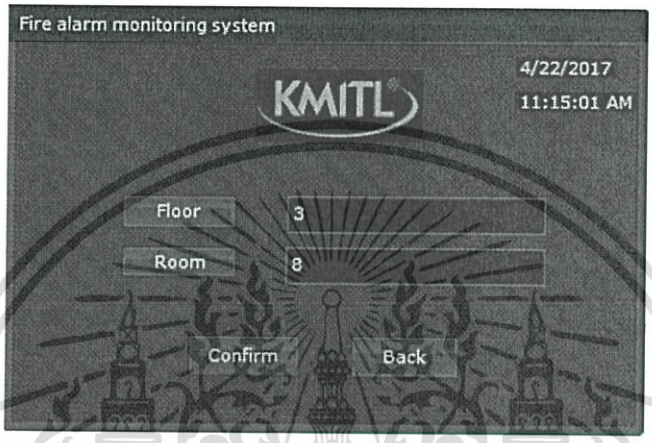


รูปที่ 3.6 หน้าต่างการกรอกรายละเอียดของอาคาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3.3 การทดสอบการยืนยันรายละเอียดของอาคาร

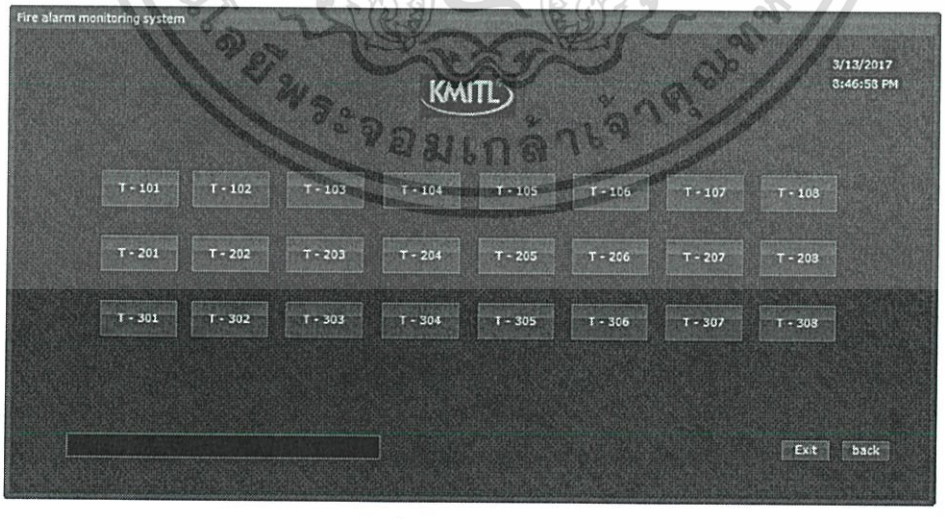
เมื่อทำการระบุจำนวนชั้นและจำนวนห้องเรียบร้อยแล้ว ระบบจะแสดงหน้าจอการยืนยันรายละเอียด เพื่อให้ผู้ใช้งานสามารถทำการตรวจสอบความถูกต้อง หากกรอกรายละเอียดผิดสามารถกลับไปแก้ไขข้อมูลได้ จากนั้นจึงกดยืนยันเพื่อไปยังหน้าจอมอนิเตอร์หลัก ดังรูปที่ 3.7



รูปที่ 3.7 หน้าต่างการยืนยันลายละเอียดของอาคาร

3.1.3.4 หน้าหลักของระบบ

เมื่อทำการเข้าสู่ระบบแล้ว จอมอนิเตอร์จะแสดงหน้าจอหลัก ดังรูปที่ 3.8 ในหน้าจอหลักจะประกอบไปด้วยแผงผังภายในอาคาร โดยแบ่งเป็นชั้นและห้องตามโครงสร้างของแต่ละตึก ซึ่งไม่เหมือนกัน

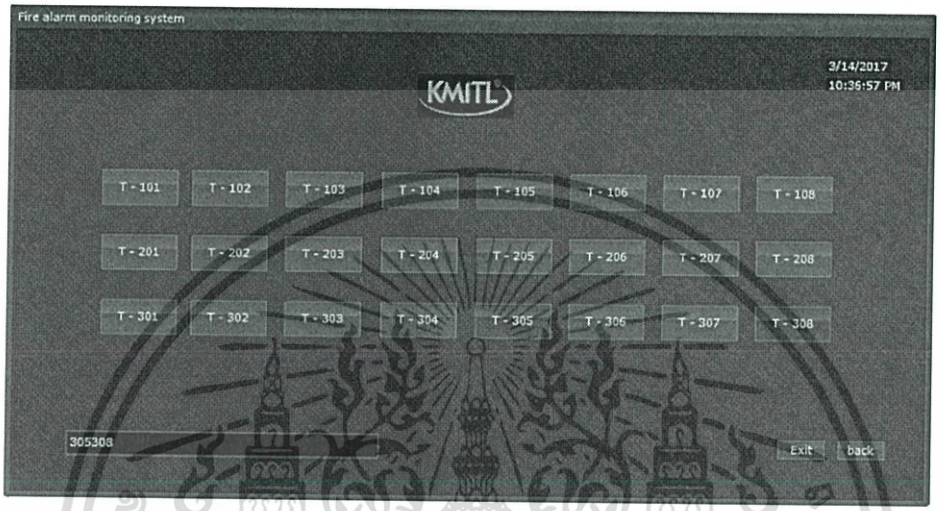


รูปที่ 3.8 หน้าจอหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3.5 การแจ้งเตือนเมื่อเกิดเหตุเพลิงไหม้

หากเกิดเหตุการณ์เพลิงไหม้ภายในอาคาร ตัวเซนเซอร์จะทำการส่งสัญญาณไปยังไมโครคอนโทรลเลอร์ โดยแต่ละชั้นจะใช้ Arduino เป็นไมโครคอนโทรลเลอร์เพื่อหาตำแหน่งเกิดเหตุว่าต้นเพลิงเกิดขึ้นที่ใด เพื่อส่งไปยังห้องควบคุมและแจ้งเตือน ดังรูปที่ 3.9



รูปที่ 3.9 หน้าจอหลักแสดงการแจ้งเตือนเมื่อเกิดเหตุเพลิงไหม้

3.1.4 การออกแบบแอปพลิเคชันสำหรับผู้ใช้งานระบบ

นอกจากระบบจะแจ้งเตือนเป็นสัญญาณเสียงผ่านตัว Buzzer และแสดงตำแหน่งการเกิดเหตุเพลิงไหม้บนจอมอนิเตอร์ในส่วนควบคุมกลางแล้ว ผู้ใช้งานหรือผู้ที่อยู่ภายในอาคารยังสามารถตรวจสอบสถานะและตำแหน่งของเหตุเพลิงไหม้ผ่านแอปพลิเคชันได้ดังรูปที่ 3.10

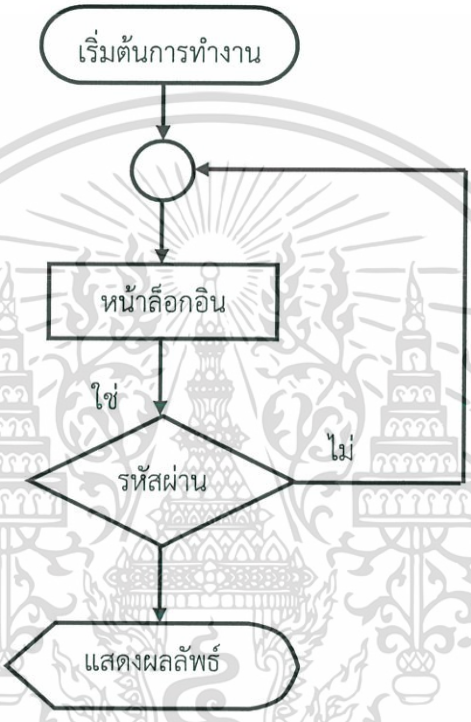
รูปที่ 3.10 แอปพลิเคชันแจ้งเตือนการเกิดเหตุเพลิงไหม้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.5 ผังการทำงานของระบบแสดงผลและการควบคุม

3.1.5.1 ผังการทำงานของหน้าล็อกอิน

เริ่มแรกเมื่อทำการเปิดโปรแกรมขึ้นมา จะต้องทำการเข้าสู่ระบบ เมื่อทำการเข้าสู่ระบบแล้ว จะแสดงหน้าจอหลักและรอรับค่าการกระทำต่างๆ และจะทำงานตามคำสั่งที่เลือก ดังแสดงในรูปที่ 3.11

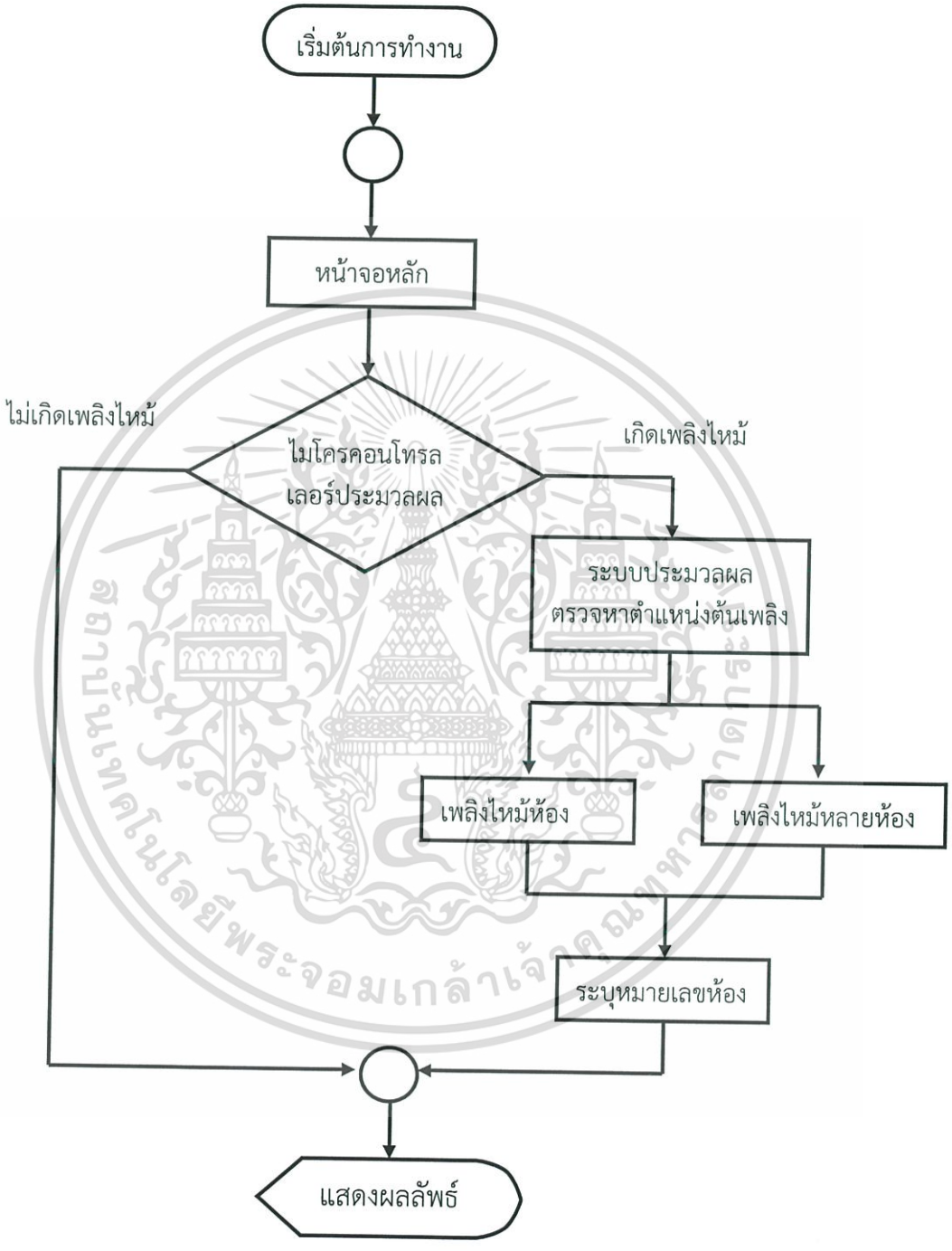


รูปที่ 3.11 ผังการทำงานของหน้าล็อกอิน

3.1.5.2 ผังการทำงานของหน้าจอหลัก

เมื่อทำการเข้าสู่ระบบแล้ว จอมอนิเตอร์จะแสดงหน้าจอหลัก ในหน้าจอหลักจะประกอบไปด้วยแผงผังภายในอาคาร โดยแบ่งเป็นชั้นและห้องตามโครงสร้างของแต่ละตึก ซึ่งไม่เหมือนกัน หากเกิดเหตุการณ์เพลิงไหม้ภายในอาคาร ตัวเซนเซอร์จะทำการส่งสัญญาณไปยังไมโครคอนโทรลเลอร์ โดยแต่ละชั้นจะใช้ Arduino เป็นไมโครคอนโทรลเลอร์คอยควบคุมเพื่อหาตำแหน่งเกิดเหตุว่าต้นเพลิงเกิดขึ้นที่ใด กล่าวคือเพลิงไหม้เกิดห้องและชั้นใดเพื่อส่งไปยังห้องควบคุมและแจ้งเตือน ดังรูปที่ 3.12

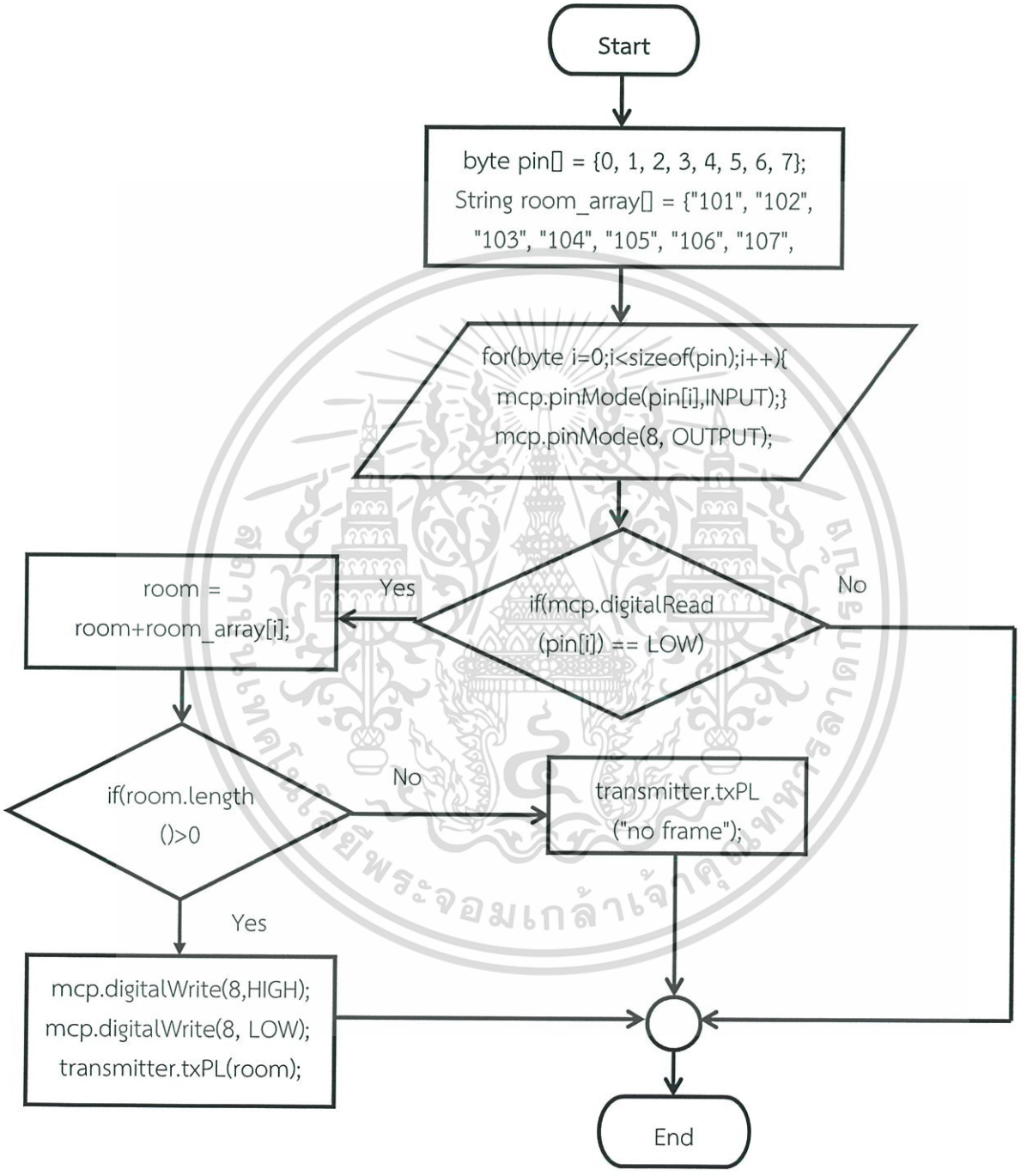
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 ผังการทำงานของหน้าจอหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

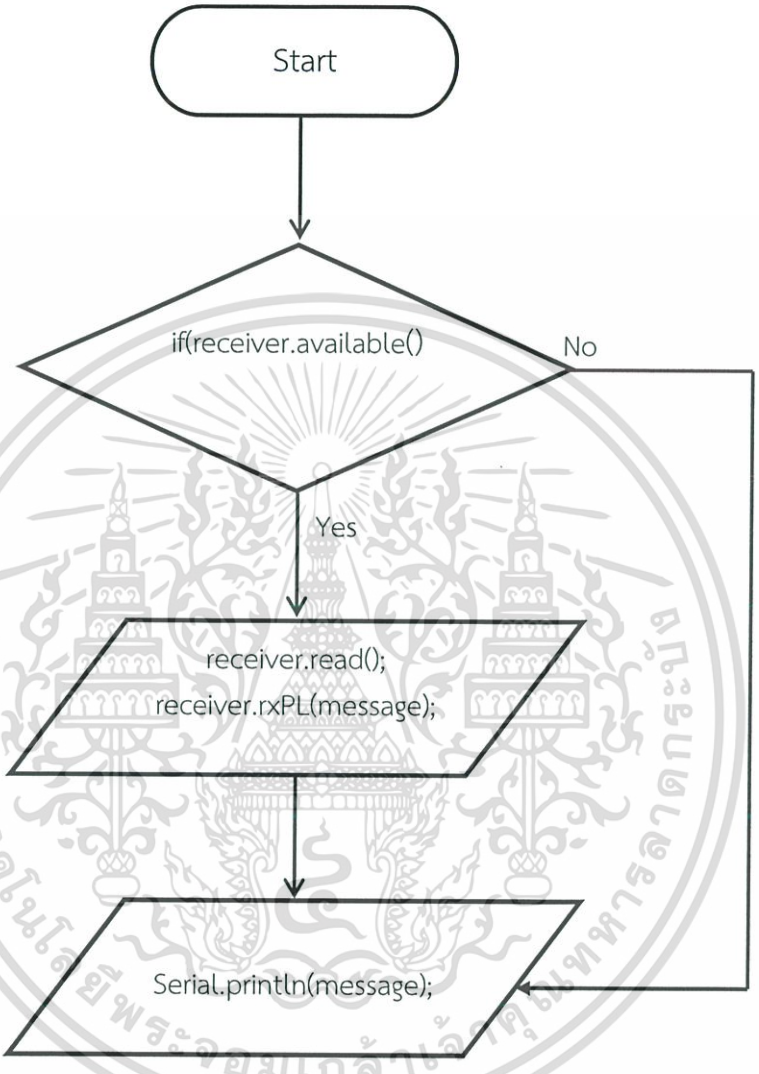
3.1.5.3 ผังการทำงานของเครื่องส่งในระบบ



รูปที่ 3.13 ผังการทำงานของเครื่องส่งในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

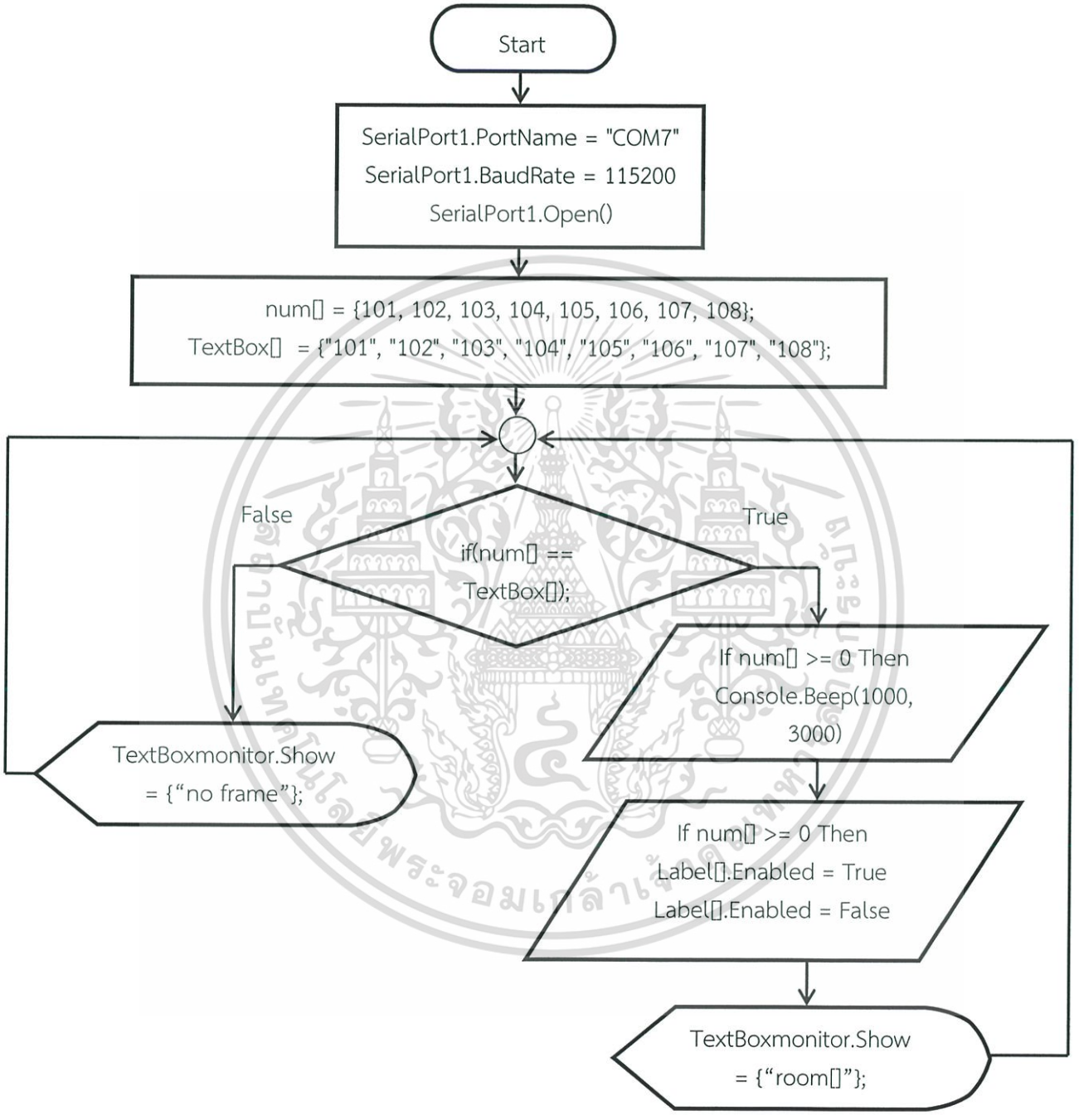
3.1.5.4 ผังการทำงานของเครื่องรับในระบบ



รูปที่ 3.14 ผังการทำงานของเครื่องรับในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

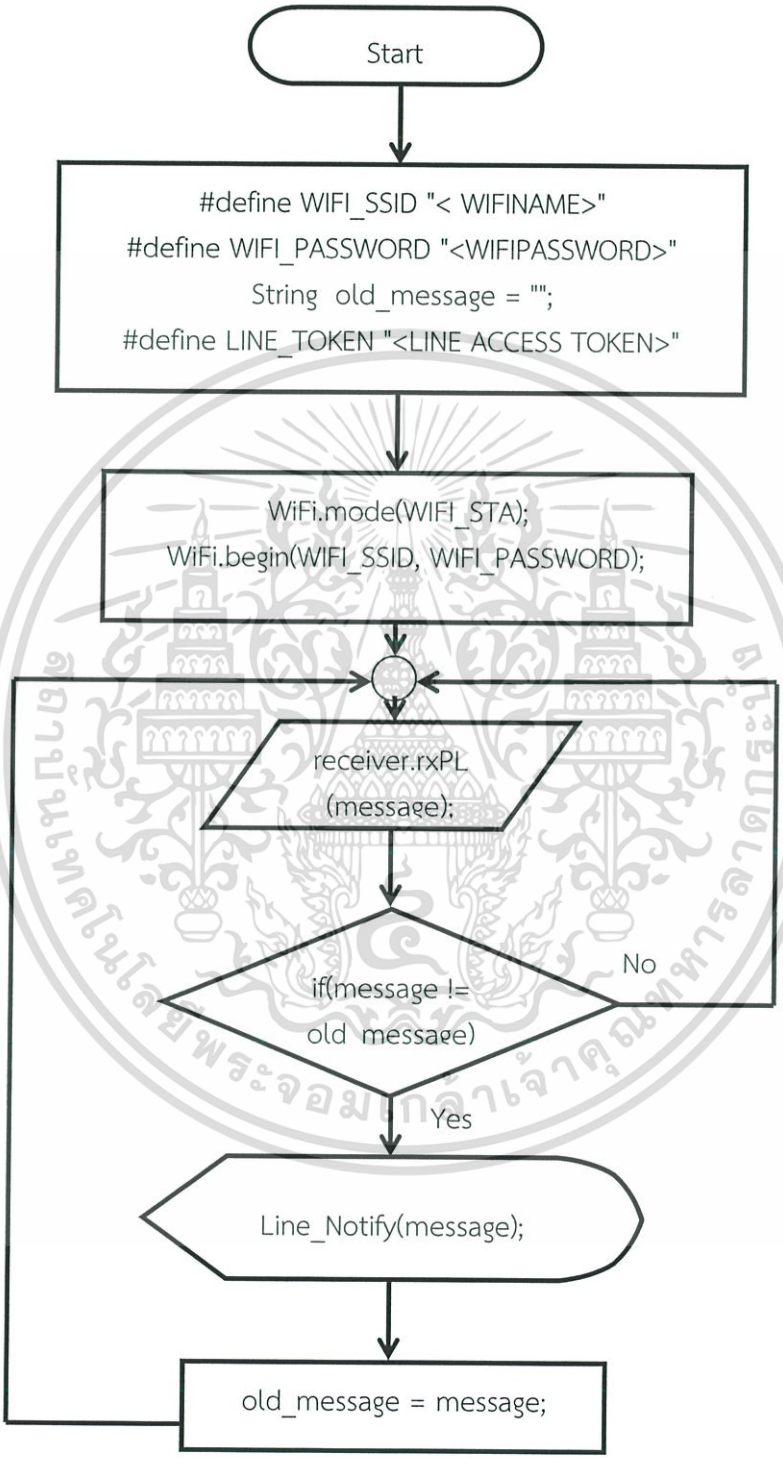
3.1.5.5 ผังการทำงานของส่วนแสดงผลจอมอนิเตอร์



รูปที่ 3.15 ผังการทำงานของส่วนแสดงผลจอมอนิเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

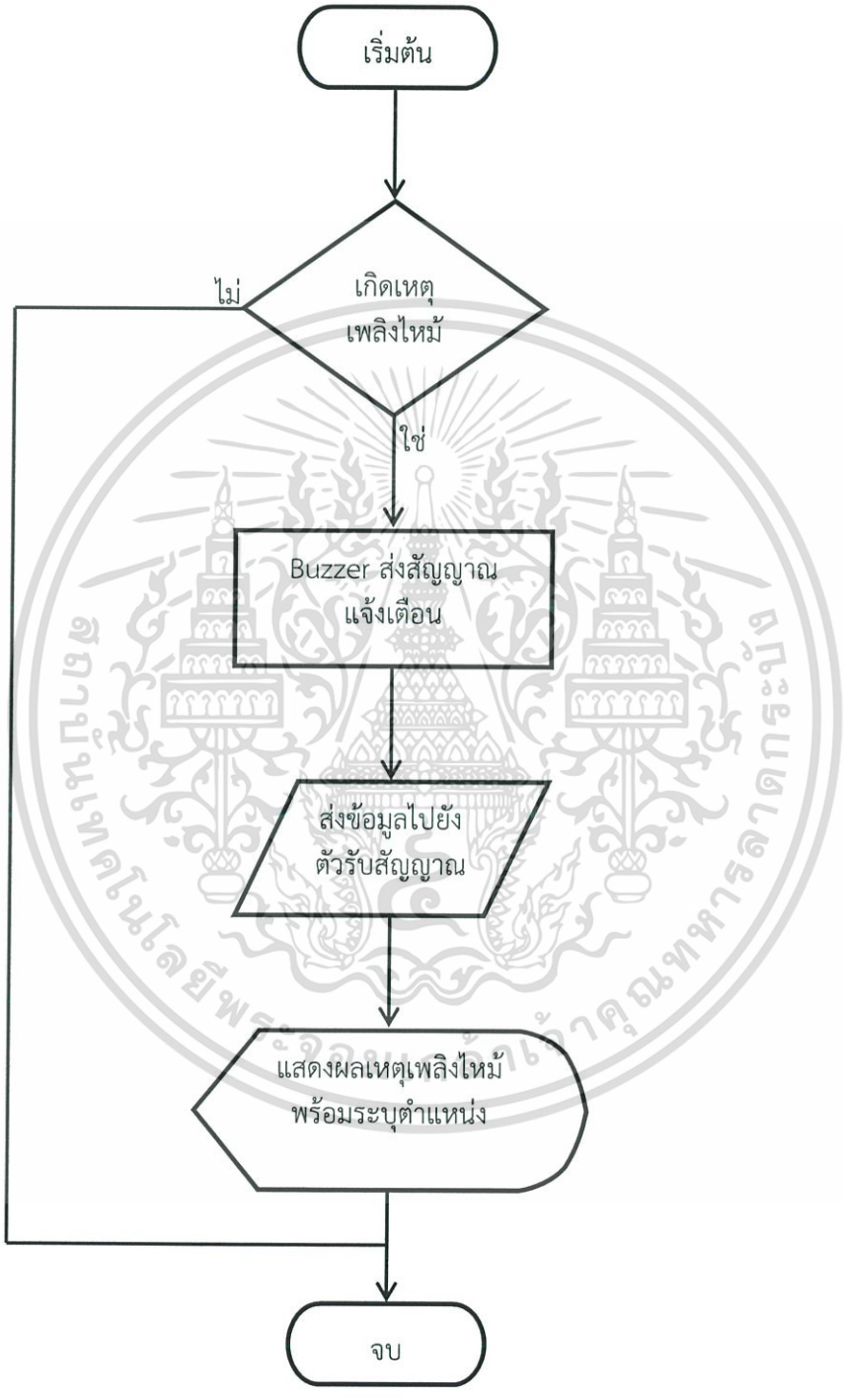
3.1.5.6 ผังการทำงานของส่วนแอปพลิเคชัน



รูปที่ 3.16 ผังการทำงานของส่วนแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.5.7 ผังการทำงานรวมของระบบ



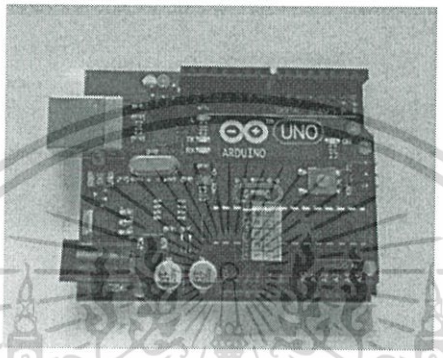
รูปที่ 3.17 ผังการทำงานรวมของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 เครื่องมือที่ใช้ในการทดลอง

3.2.1 ไมโครคอนโทรลเลอร์ Arduino Uno3

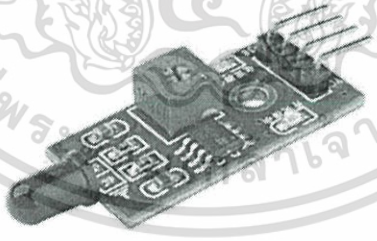
เป็นบอร์ดไมโครคอนโทรเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open Source คือ มีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software ตัวบอร์ด Arduino ถูกออกแบบมาให้ใช้งานได้ง่าย ดังนั้นจึงใช้เป็นไมโครคอนโทรเลอร์เพื่อทำหน้าที่ควบคุมและประมวลผล ดังรูปที่ 3.18



รูปที่ 3.18 ไมโครคอนโทรลเลอร์ Arduino Uno3

3.2.2 Infrared IR Flame Detector Sensor Module

Infrared IR Flame Detector Sensor Module หรือ เซนเซอร์ตรวจจับเปลวไฟด้วยอินฟราเรด ทำหน้าที่ในการตรวจจับเปลวไฟโดยใช้คลื่นอินฟราเรด แสดงผลการทำงานได้ทั้งในรูปดิจิตอล และอนาล็อก ดังรูปที่ 3.19



รูปที่ 3.19 Infrared IR Flame Detector Sensor Module

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

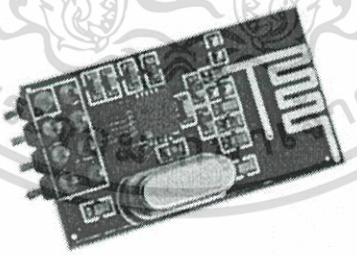
3.2.3 Active Buzzer

Active Buzzer เป็นลำโพงแบบแม่เหล็กไฟฟ้าหรือ เรียกอีกอย่างว่าแบบเปียโซ เป็นลำโพงที่มีวงจรกำเนิดความถี่ (oscillator) อยู่ภายในตัว เมื่อป้อนแรงดันสามารถกำเนิดเสียงได้ด้วยตัวเอง แต่ไม่สามารถเปลี่ยนความถี่ของเสียงได้ ดังรูปที่ 3.20

รูปที่ 3.20 Active Buzzer

3.2.4 NRF24L01 Module

โมดูล NRF24L01 เป็นโมดูลสื่อสารไร้สาย ที่สามารถเขียนโปรแกรมให้เป็นได้ทั้งตัวรับและตัวส่ง สามารถใช้กับ Arduino ได้หลาย ๆ ตัวพร้อมกัน มีความเร็ว 2.4GHz จึงสื่อสารได้รวดเร็วและไม่ต้องการเสาอากาศที่ยาว มีขนาดเล็กสะดวกในการต่อใช้งาน สามารถประยุกต์ใช้งานได้หลายอย่าง เช่น ใช้เป็นอุปกรณ์ส่งข้อมูลของเซนเซอร์อัตโนมัติสำหรับควบคุม อุณหภูมิ ความชื้น การแจ้งเตือนต่าง ๆ ควบคุมและติดตามหุ่นยนต์ Robot Control and Monitoring ได้ในระยะ 15-500 เมตร โมดูลนี้ใช้ชิฟ nRF24L01+ m ทำงานด้วยความเร็วสูง High-speed SPI interface ใช้พลังงานต่ำ รองรับการทำงานร่วมกับ Arduino ดังรูปที่ 3.21

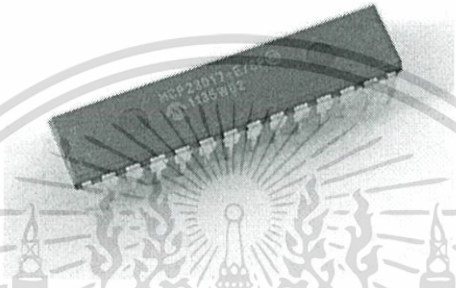


รูปที่ 3.21 NRF24L01 Module

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.5 MCP23017 (16-Bit I/O Expander with Serial Interface)

ไอซี MCP23017 ทำให้ขาที่มีอย่างจำกัดของ Arduino เพิ่มขึ้น 16 ขา โดยใช้ขาจาก Arduino เพียง 2 ขา เป็นรูปแบบการติดต่อแบบ I2C สามารถกำหนดแต่ละขาว่าจะให้เป็นอินพุตหรือเอาต์พุตก็ได้ ใช้ไฟช่วง 2.7-5.5 โวลต์ สามารถขับกระแสได้สูงสุด 20 มิลลิแอมป์ สามารถกำหนดอ้างอิง Address ได้ 8 ตำแหน่ง แปลว่าต่อเซ็นเซอร์รวมกันได้ 8 ตัวหรือ ใช้ขาจาก Arduino 2 ขา สามารถขยายขาได้สูงสุดถึง 128 I/O ดังรูปที่ 3.22



รูปที่ 3.22 ไอซี MCP23017 (16-Bit I/O Expander with Serial Interface)

3.3 การจัดเก็บผลการทดลอง

3.3.1 ผลการทดสอบเซนเซอร์

1) ผลการทดสอบระบบเมื่อไม่มีเหตุเพลิงไหม้

2) ผลการทดสอบระบบเมื่อมีเหตุเพลิงไหม้

3.3.2 ผลการทดสอบเซนเซอร์เมื่อผ่าน MCP23017

3.3.3 ผลการทดสอบ NRF24L01

3.3.4 ผลการทดสอบในส่วนของแอปพลิเคชัน

1) การแจ้งเตือนเมื่อเกิดเหตุเพลิงไหม้

2) การแจ้งเตือนเมื่อเหตุเพลิงไหม้หยุดลง

3.3.5 ผลการทดสอบในส่วนของจอมอนิเตอร์

1) ผลการทดสอบการเข้าสู่โปรแกรม

2) ผลการทดสอบจอมอนิเตอร์แสดงข้อมูล

3) แสดงผลการทดสอบเมื่อไม่มีเหตุเพลิงไหม้

4) แสดงผลการทดสอบเมื่อเหตุเพลิงไหม้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

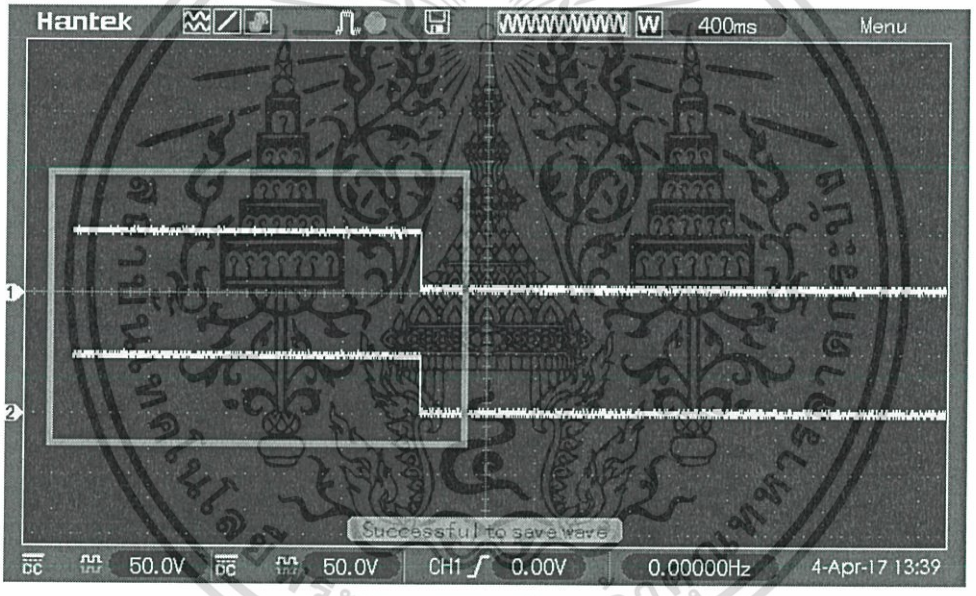
ผลการทดลอง

4.1 ผลการทดสอบเซนเซอร์

4.1.1 ผลการทดสอบเซนเซอร์เมื่อไม่มีเหตุเพลิงไหม้

4.1.1.1 ผลการทดสอบเซนเซอร์เมื่อไม่มีเหตุเพลิงไหม้จากออสซิลโลสโคป

เมื่อนำเซนเซอร์มาทดสอบ เพื่อวัดสัญญาณเอาต์พุตจากเครื่องออสซิลโลสโคป ปรากฏว่าสัญญาณที่ได้เมื่อไม่มีเหตุเพลิงไหม้ จะมีค่าสัญญาณเป็นบิต 1 หรือเท่ากับค่าแรงดันที่ใช้ในการป้อนเพื่อเลี้ยงตัวเซนเซอร์คือ 5 โวลต์ โดยช่องสัญญาณแรกเป็นสัญญาณอินพุต ช่องสัญญาณที่สองเป็นสัญญาณเอาต์พุต ดังรูปที่ 4.1

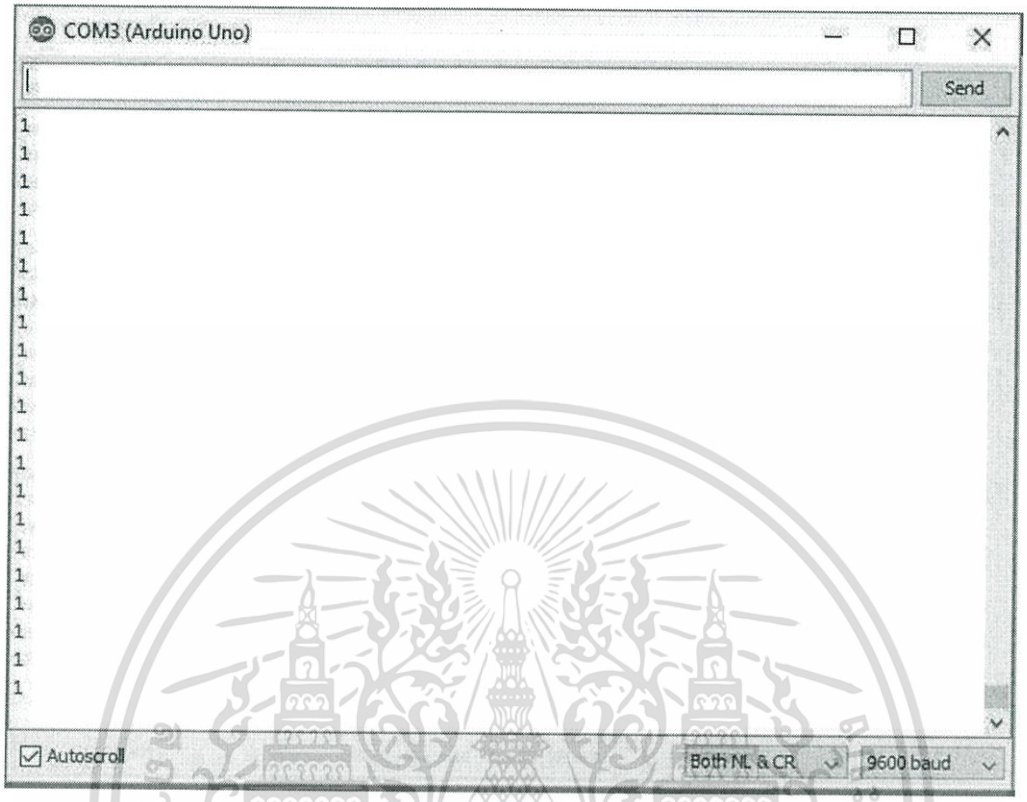


รูปที่ 4.1 สัญญาณเอาต์พุตจากเครื่องออสซิลโลสโคป

4.1.1.2 ผลการทดสอบเซนเซอร์เมื่อไม่มีเหตุเพลิงไหม้จากอาร์ดูอิโน

เมื่อนำเซนเซอร์มาทำการทดสอบ เพื่อวัดค่าเอาต์พุตจากการ์ดูอิโนปรากฏว่าค่าที่ได้เมื่อไม่มีเหตุเพลิงไหม้ จะมีค่าเป็นบิต 1 ดังรูปที่ 4.2

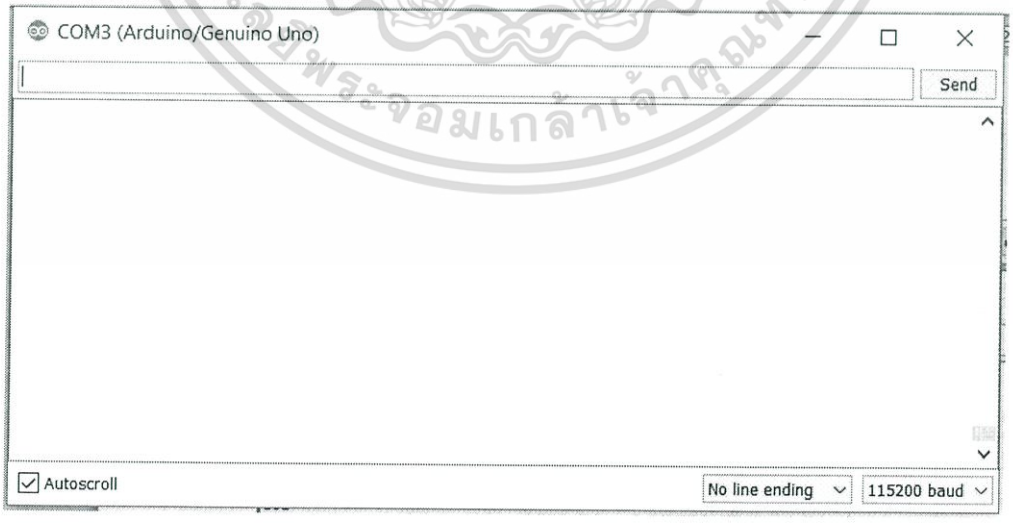
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 ค่าเอาต์พุตจาการ์ดูอิโน

4.1.1.3 ผลการแสดงผลเมื่อไม่มีเหตุเพลิงไหม้จากอาร์ดูอิโน

เมื่อนำเซนเซอร์มา ทดสอบเพื่อแสดงผลการแจ้งเตือนจาการ์ดูอิโน ปรากฏว่าเมื่อไม่มีเหตุเพลิงไหม้ หน้าจอจะไม่แสดงค่าใดใดออกมา เนื่องจากคำสั่งในโปรแกรม ดังรูปที่ 4.3



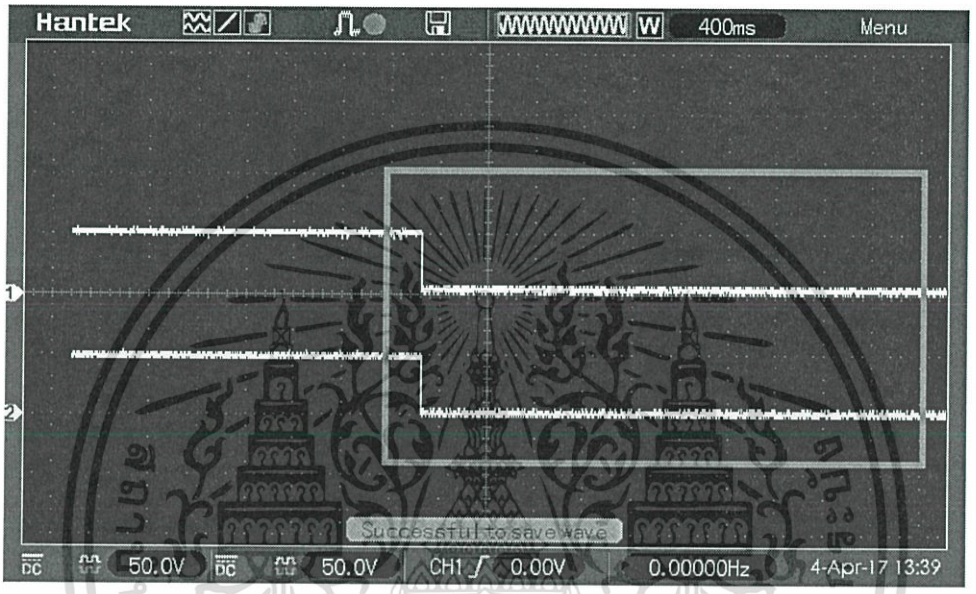
รูปที่ 4.3 ผลการแจ้งเตือนจาการ์ดูอิโน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 ผลการทดสอบเซนเซอร์เมื่อมีเหตุเพลิงไหม้

4.1.2.1 ผลการทดสอบเซนเซอร์เมื่อมีเหตุเพลิงไหม้จากออสซิลโลสโคป

เมื่อนำเซนเซอร์มาทดสอบ เพื่อวัดสัญญาณเอาต์พุตจากเครื่องออสซิลโลสโคปปรากฏว่าสัญญาณที่ได้เมื่อมีเหตุเพลิงไหม้ จะมีค่าสัญญาณเป็นบิต 0 โดยช่องสัญญาณแรกเป็นสัญญาณอินพุต ช่องสัญญาณที่สองเป็นสัญญาณเอาต์พุต ดังรูปที่ 4.4

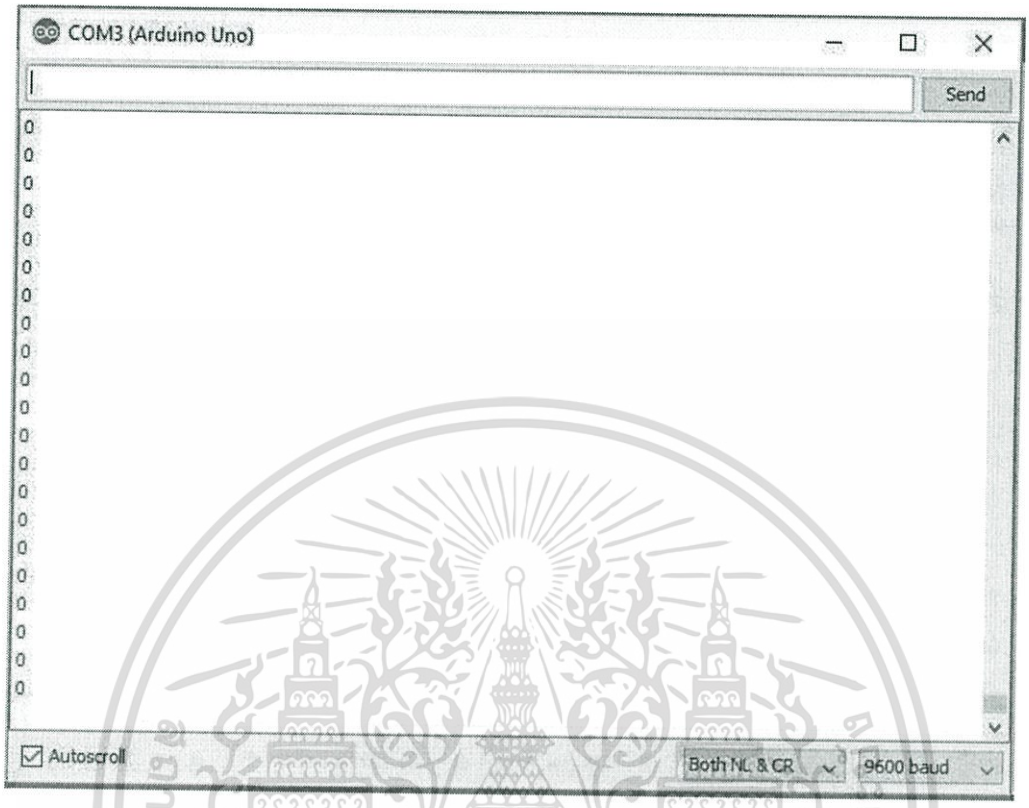


รูปที่ 4.4 สัญญาณเอาต์พุตจากเครื่องออสซิลโลสโคป

4.1.2.2 ผลการทดสอบเซนเซอร์เมื่อมีเหตุเพลิงไหม้จากอาร์ดูอิโน

เมื่อนำเซนเซอร์มาทำการทดสอบ เพื่อวัดค่าเอาต์พุตจาการ์ดูอิโนปรากฏว่าค่าที่ได้เมื่อมีเหตุเพลิงไหม้ จะมีค่าสัญญาณเป็นบิต 0 ดังรูปที่ 4.5

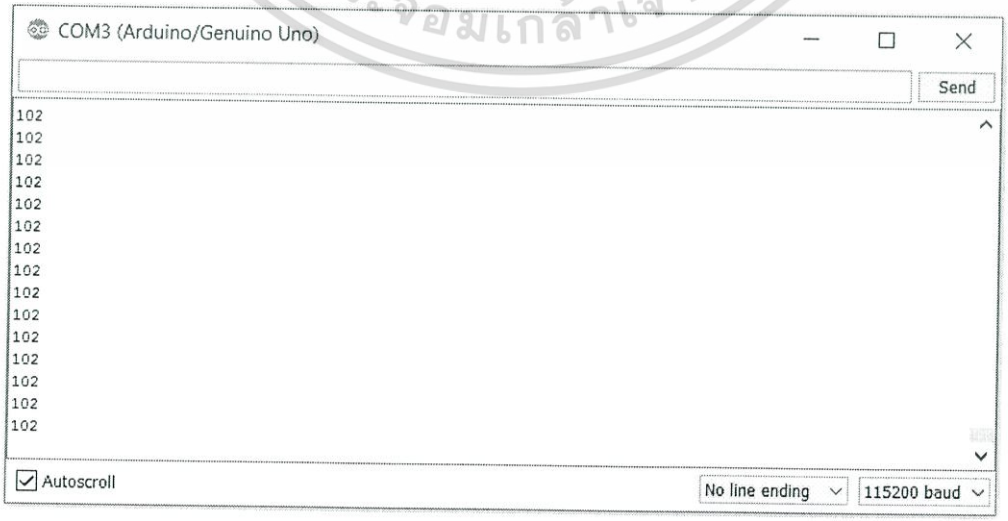
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



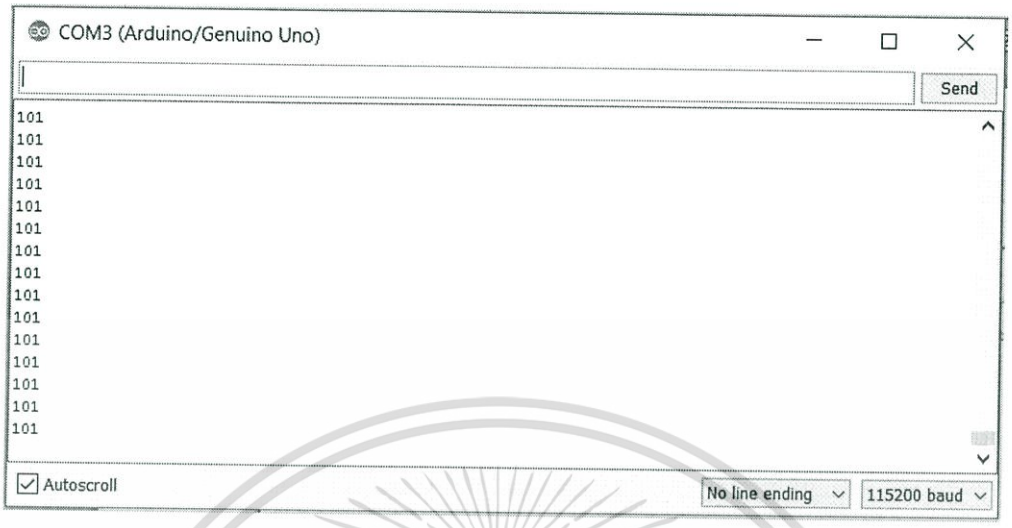
รูปที่ 4.5 ค่าเอาต์พุตจาการ์ดูอิโน

4.1.1.3 ผลการแสดงผลเมื่อมีเหตุเพลิงไหม้จากอาร์ดูอิโน

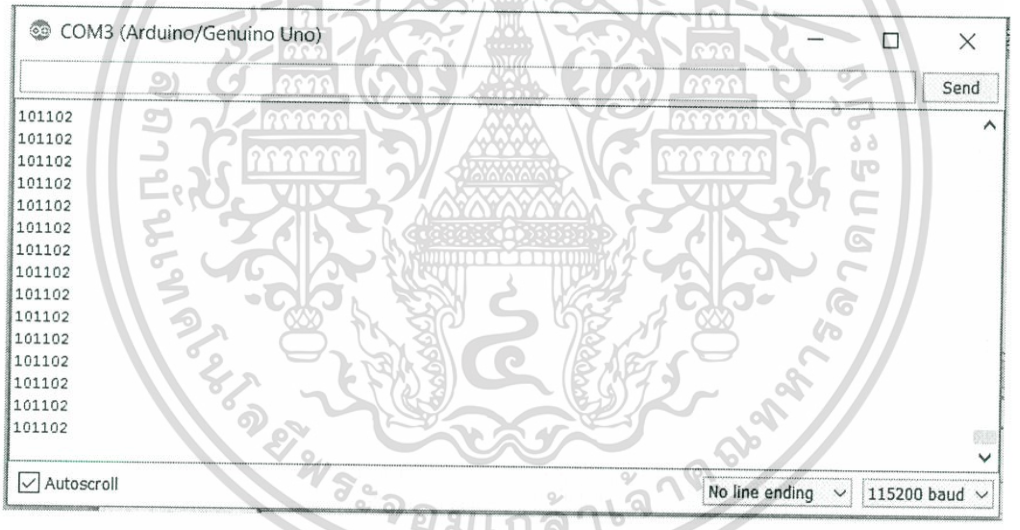
เมื่อนำเซนเซอร์มาทำการทดสอบ เพื่อแสดงผลการแจ้งเตือนจาการ์ดูอิโน ปรากฏว่าเมื่อมีเหตุเพลิงไหม้ หน้าจอจะแสดงค่าออกมา ซึ่งค่าที่แสดงออกมาคือหมายเลขห้องที่เป็นต้นเพลิง หากมีไฟไหม้เพียงห้องเดียวก็จะแสดงค่าเพียงค่าเดียว ดังรูปที่ 4.6 แต่หากมีต้นเพลิงหลายห้องระบบจะแสดงค่าเท่ากับจำนวนห้องที่เป็นต้นเพลิง ดังรูปที่ 4.7



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 ผลการแจ้งเตือนจาการ์ดอิโนเมื่อเกิดเพลิงไหม้เพียงห้องเดียว

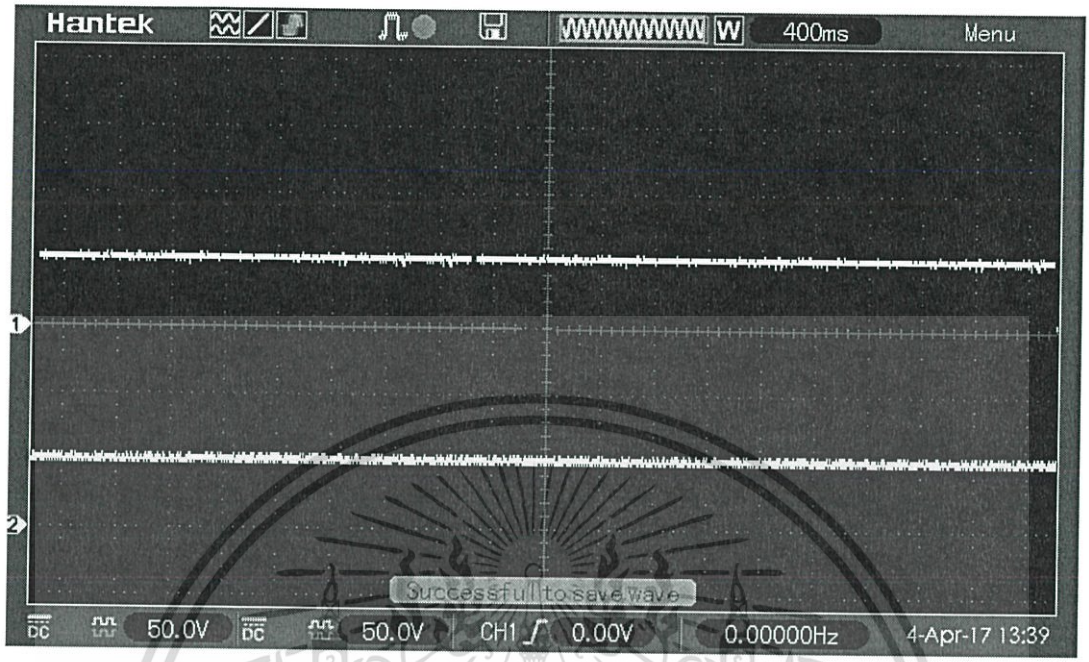


รูปที่ 4.7 ผลการแจ้งเตือนจาการ์ดอิโนเมื่อเกิดเพลิงไหม้หลายห้อง

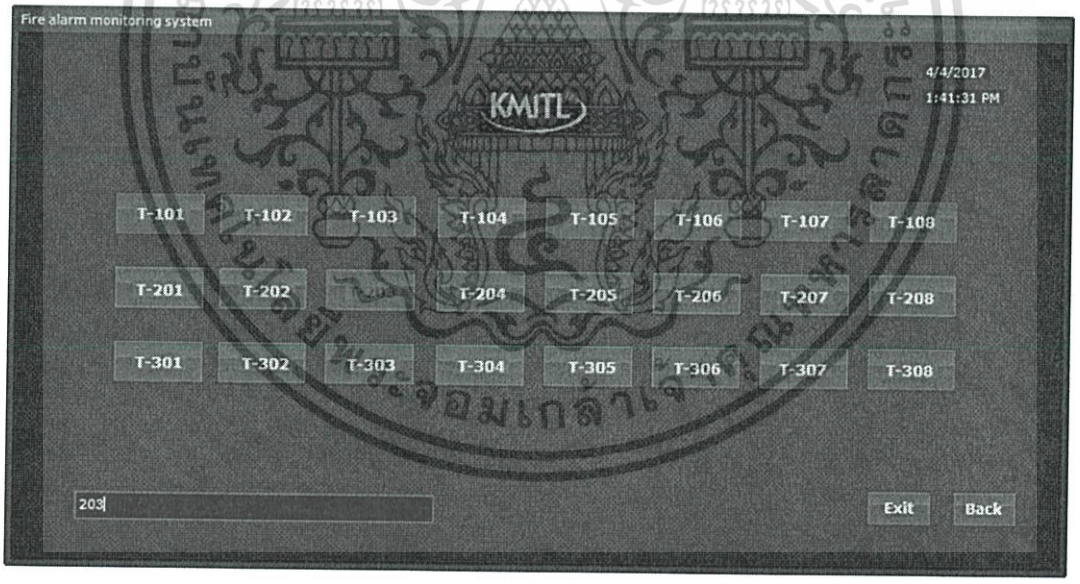
4.2 ผลการทดสอบเซนเซอร์เมื่อผ่าน MCP23017

ผลการทดสอบเซนเซอร์เมื่อผ่าน MCP23017 เมื่อทำการทดสอบระบบด้วยเซนเซอร์ปรากฏว่าเมื่อจุดไฟ เซนเซอร์อ่านค่าสัญญาณที่ผ่านสายไฟจากต้นทางถึงปลายทางได้โดยไม่มีการสูญเสียใด โดยช่องสัญญาณที่ 1 เป็นสัญญาณที่อ่านได้ ณ ตำแหน่งต้นทาง และช่องสัญญาณที่ 2 เป็นสัญญาณที่อ่านได้ ณ ตำแหน่งปลายทาง ดังรูปที่ 4.8 เมื่อทำการต่อวงจรกับระบบ เซนเซอร์สามารถตรวจจับเปลวไฟและส่งค่าให้ไมโครคอนโทรลเลอร์ทำการประมวลผลเพื่อหาตำแหน่งต้นเพลิง เพื่อแจ้งเตือนแบบเสียงผ่านตัว Buzzer และแสดงตำแหน่งต้นเพลิงบนจอมอนิเตอร์ได้ ดังรูปที่ 4.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 สัญญาณจากเซนเซอร์ที่ต้นทางและปลายทาง

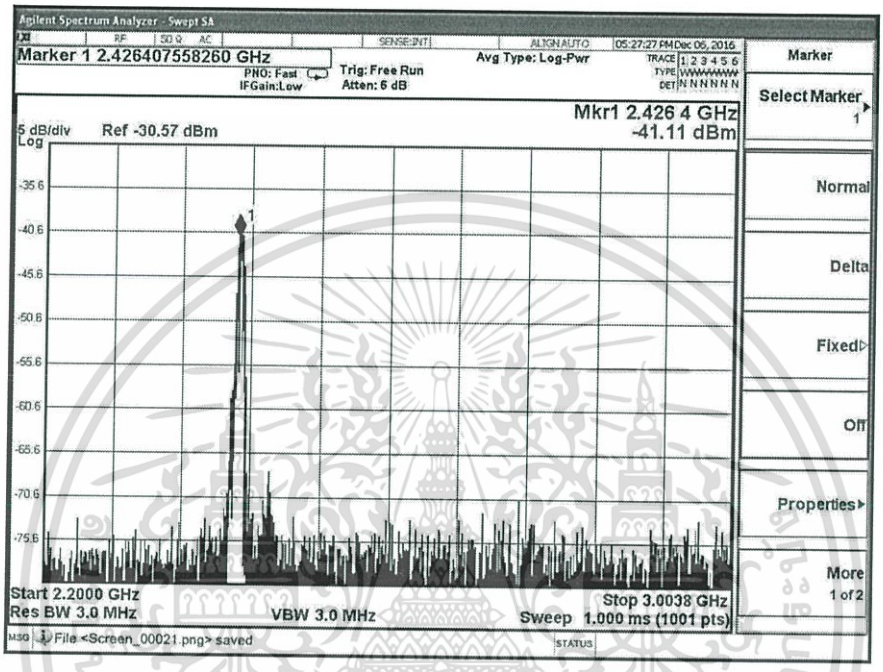


รูปที่ 4.9 การแจ้งเตือนผ่านจอมอนิเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ผลการทดสอบ NRF24L01

ในการทดลองวัดสเปกตรัมของสัญญาณส่งจะทำการวัดค่าความถี่ของสัญญาณที่ส่งออกไปจากอุปกรณ์ NRF24L01 เพื่อหาค่าความถี่ และกำลังที่ส่งออกไปของอุปกรณ์แสดงได้ ดังรูปที่ 4.10



รูปที่ 4.10 สเปกตรัมของสัญญาณจากอุปกรณ์ NRF24L01

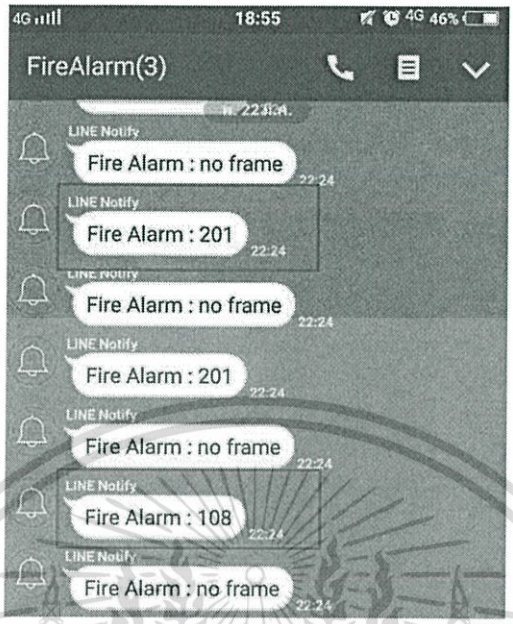
4.4 ผลการทดสอบในส่วนของแอปพลิเคชัน

ผู้ใช้งานหรือผู้ที่อยู่ภายในอาคาร สามารถตรวจสอบสถานะและตำแหน่งของเหตุเพลิไหม้ผ่านแอปพลิเคชันได้ หากเกิดเหตุเพลิงไหม้ระบบจะทำการแจ้งเตือนผู้ใช้งานหรือผู้ที่อยู่ภายในอาคารผ่านทางแอปพลิเคชันเป็นข้อความตัวอักษร ในข้อความตัวอักษรจะระบุตำแหน่งหมายเลขห้องที่เกิดเหตุเพลิงไหม้ไว้ โดยการแจ้งเตือนแบ่งได้เป็น 2 ส่วน

4.4.1 การแจ้งเตือนเมื่อเกิดเหตุเพลิงไหม้

เมื่อเกิดเหตุเพลิงไหม้ขึ้น ระบบจะทำการค้นหาตำแหน่งต้นเพลิง จากนั้นระบบจะส่งตำแหน่งของต้นเพลิงผ่านทางแอปพลิเคชันไลน์ โดยระบุเป็นหมายเลขห้อง ดังแสดงในรูปที่ 4.11

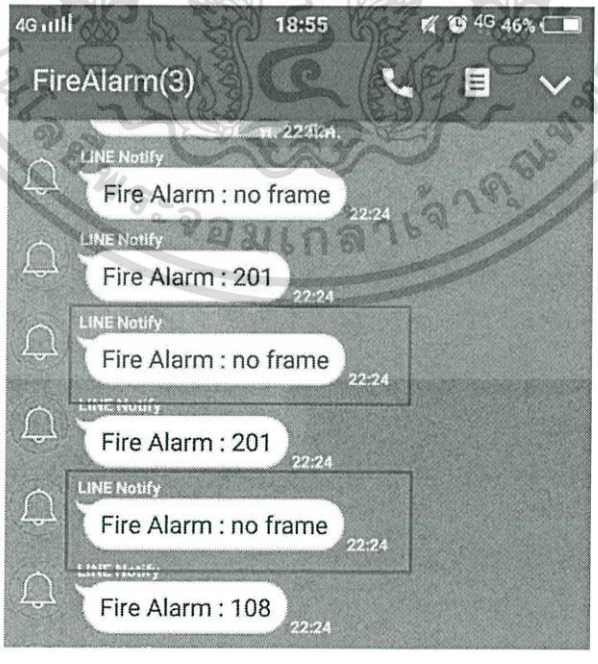
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 การแจ้งเตือนผ่านแอปพลิเคชันสำหรับผู้ใช้งานเมื่อเกิดเหตุเพลิงไหม้

4.4.2 การแจ้งเตือนเมื่อเหตุเพลิงไหม้หยุดลง

เมื่อเหตุเพลิงไหม้หยุดลง ระบบจะส่งข้อความ no flame ผ่านทางแอปพลิเคชันไลน์ เพื่อบอกผู้ใช้งานว่าเหตุเพลิงไหม้ได้หยุดลงแล้ว ดังแสดงในรูปที่ 4.12



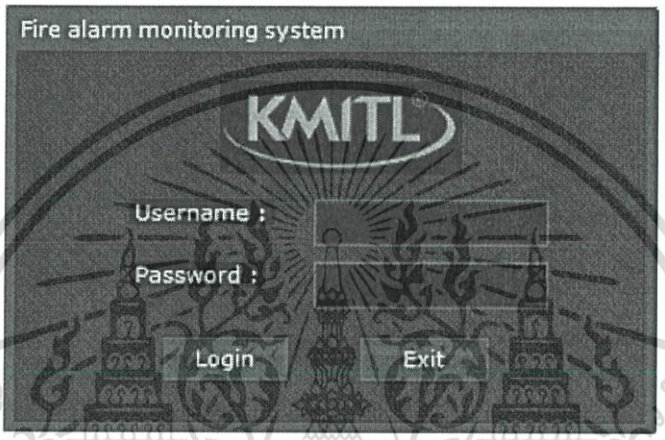
รูปที่ 4.12 การแจ้งเตือนผ่านแอปพลิเคชันสำหรับผู้ใช้งานเมื่อเหตุเพลิงไหม้หยุดลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 ผลการทดสอบในส่วนของจอมอนิเตอร์

4.5.1 ผลการทดสอบการเข้าสู่โปรแกรม

เมื่อเราทำการเข้าสู่ระบบนั้น จะมีช่องให้ผู้ใช้กรอก User name และ Password โดยที่ได้ตั้ง User name : admin และ Password : 1234 เมื่อใส่ข้อมูลอย่างถูกต้องแล้ว จะสามารถเข้าระบบได้ ดังรูปที่ 4.13



รูปที่ 4.13 หน้าต่างการเข้าสู่โปรแกรม

หากว่าใส่ User name และ Password ผิด หน้าต่างโปรแกรมจะแสดง ดังรูปที่ 4.14

รูปที่ 4.14 หน้าต่างเข้าสู่โปรแกรมเมื่อใส่ User name หรือ Password ผิด

4.5.2 ผลการทดสอบการกรอกรายละเอียดของอาคาร

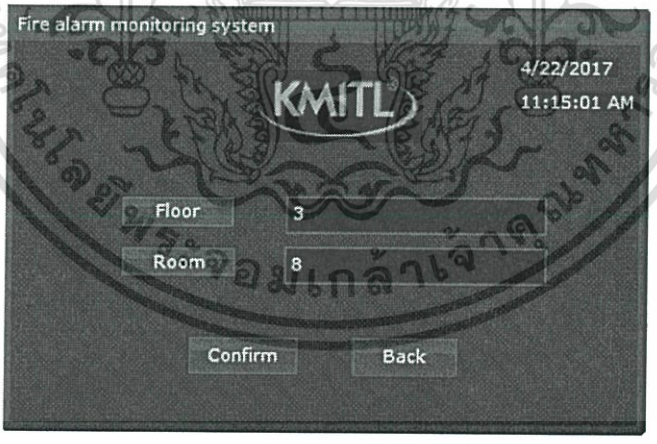
เมื่อทำการเข้าสู่ระบบแล้ว จะมีช่องให้กรอกชื่อของอาคาร รายละเอียดของอาคาร เช่น จำนวนชั้น จำนวนห้องในแต่ละชั้น ดังรูปที่ 4.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.15 หน้าต่างการกรอกรายละเอียดของอาคาร

4.5.3 ผลการทดสอบการยืนยันรายละเอียดของอาคาร

เมื่อทำการระบุจำนวนชั้นและจำนวนห้องเรียบร้อยแล้ว ระบบจะแสดงหน้าจอการยืนยันรายละเอียด เพื่อให้ผู้ใช้งานสามารถทำการตรวจสอบความถูกต้อง หากกรอกรายละเอียดผิดสามารถกลับไปแก้ไขข้อมูลได้ จากนั้นจึงกดยืนยันเพื่อไปยังหน้าจอมอนิเตอร์หลัก ดังรูปที่ 4.16

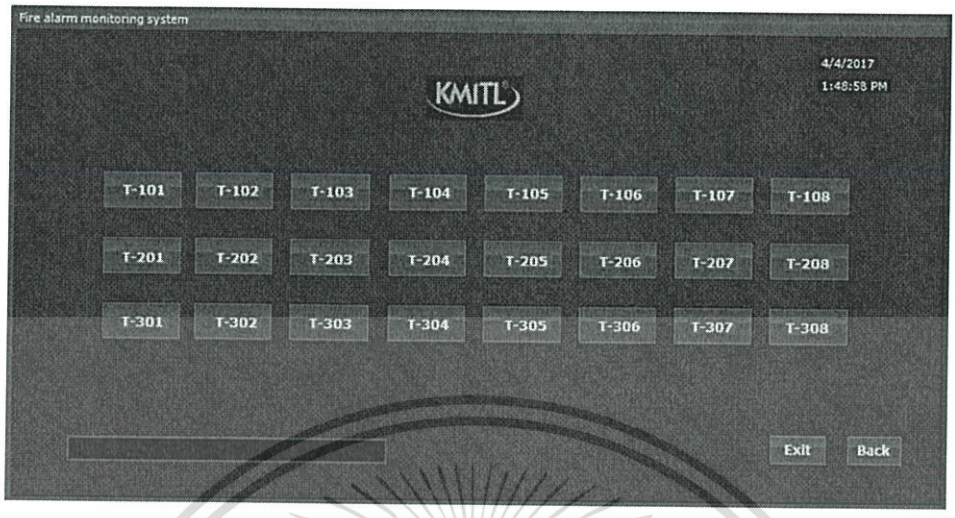


รูปที่ 4.16 หน้าต่างการยืนยันรายละเอียดของอาคาร

4.5.4 ผลการทดสอบจอมอนิเตอร์แสดงข้อมูล

เมื่อทำการระบุจำนวนชั้นและจำนวนห้องเรียบร้อยแล้ว ระบบจะแสดงหน้าจอมอนิเตอร์เพื่อแสดงข้อมูล หากมีเหตุการ์เพลิงไหม้ระบบจะทำการแจ้งเตือน บอกตำแหน่งต้นเพลิงบนหน้าจอและบอกหมายเลขห้องที่เกิดเหตุเพลิงไหม้ ดังรูปที่ 4.17

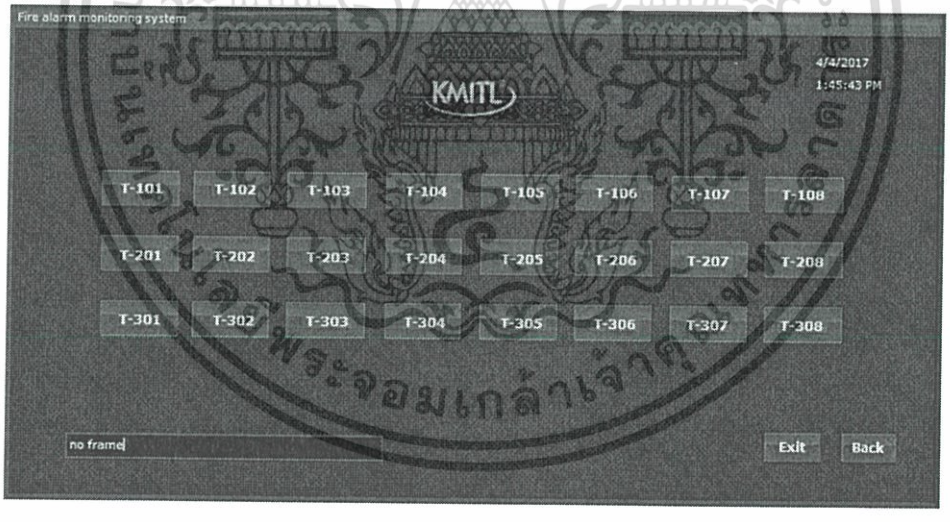
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.17 หน้าต่างจอมอนิเตอร์แสดงข้อมูล

4.5.5 แสดงผลการทดสอบเมื่อไม่มีเหตุเพลิงไหม้

หากไม่มีเหตุเพลิงไหม้ระบบจะไม่ทำการแจ้งเตือน บอกตำแหน่งต้นเพลิงบนหน้าจอ และบอกหมายเลขห้องที่เกิดเหตุเพลิงไหม้ ดังรูปที่ 4.18

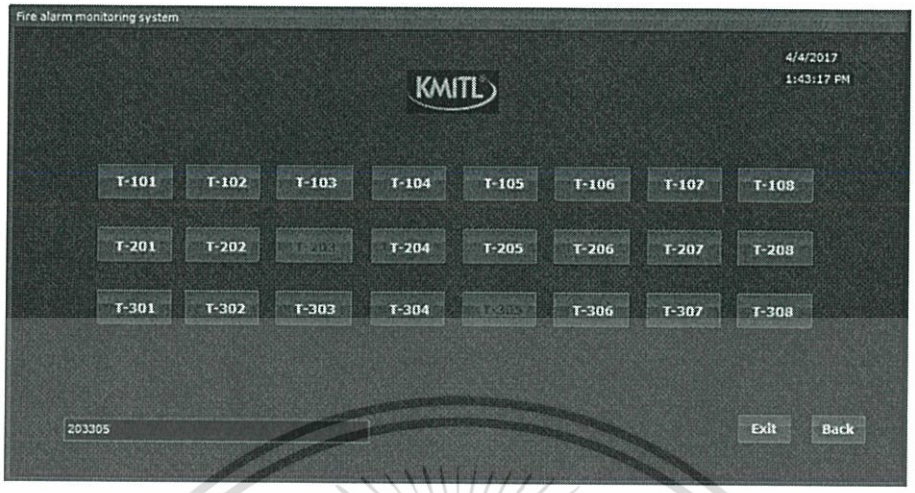


รูปที่ 4.18 หน้าต่างจอมอนิเตอร์แสดงข้อมูลเมื่อไม่มีเหตุเพลิงไหม้

4.5.6 แสดงผลการทดสอบเมื่อเหตุเพลิงไหม้

หากมีเหตุเพลิงไหม้ระบบจะทำการแจ้งเตือน บอกตำแหน่งต้นเพลิงบนหน้าจอ และบอกหมายเลขห้องที่เกิดเหตุเพลิงไหม้ ดังรูปที่ 4.19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.19 หน้าต่างจอมอนิเตอร์แสดงข้อมูลเมื่อมีเหตุเพลิงไหม้

4.6 แบบชิ้นงาน

4.6.1 ชิ้นงานทางด้านฝั่งเครื่องส่ง

ภาพชิ้นงานทางด้านฝั่งเครื่องส่ง ดังรูปที่ 4.20

รูปที่ 4.20 ภาพชิ้นงานทางด้านฝั่งเครื่องส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

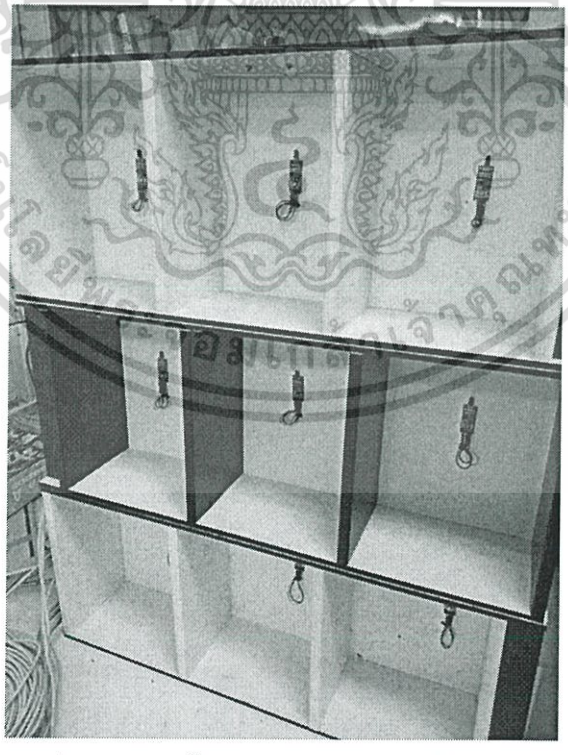
4.6.2 ชิ้นงานทางด้านฝั่งเครื่องรับ

ภาพชิ้นงานทางด้านฝั่งเครื่องส่ง ดังรูปที่ 4.21

รูปที่ 4.21 ภาพชิ้นงานทางด้านฝั่งเครื่องรับ

4.6.3 ชิ้นงานของตัวเซนเซอร์ที่ติดอาคาร

ภาพชิ้นงานชิ้นงานของตัวเซนเซอร์ที่ติดอาคาร ดังรูปที่ 4.22



รูปที่ 4.22 ภาพชิ้นงานของตัวเซนเซอร์ที่ติดอาคาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

การทดลองระบบแจ้งเตือนการเกิดเพลิงไหม้สำหรับอาคารขนาดใหญ่ มีการทำงานดังนี้ ภายในอาคารจะทำการติดตั้ง Infrared IR Flame Detector Sensor Module ในทุกๆ ห้องของอาคาร ซึ่งเป็นเซนเซอร์ที่ทำการตรวจจับเปลวไฟ หากเกิดเหตุการณ์เพลิงไหม้ เซนเซอร์จะทำการส่งสัญญาณไปยังไมโครคอนโทรลเลอร์ โดยจะใช้ Arduino เป็นไมโครคอนโทรลเลอร์คอยควบคุมเพื่อหาตำแหน่งเกิดเหตุว่าต้นเพลิงเกิดขึ้นที่ใด กล่าวคือ เพลิงไหม้เกิดห้องและชั้นใด เพื่อส่งไปยังห้องควบคุมและแจ้งเตือน

ห้องควบคุมจะมีจอมอนิเตอร์คอยแจ้งเตือนสถานะความปลอดภัยในแต่ละห้องเอาไว้ ถ้าเกิดเหตุเพลิงไหม้ขึ้นในห้องใดห้องหนึ่ง Buzzer จะแจ้งเตือนเป็นสัญญาณเสียงว่าเกิดเพลิงไหม้ และที่มอนิเตอร์จะแจ้งเตือนเหตุเพลิงไหม้ว่าต้นเพลิงเกิดขึ้นที่ตำแหน่งใด

นอกจากนี้ผู้ใช้งานหรือผู้ที่อยู่ภายในอาคาร สามารถตรวจสอบการเกิดเหตุเพลิงไหม้ภายในอาคารผ่านทางแอปพลิเคชันได้เช่นกัน

5.2 ข้อเสนอแนะ

เพื่อความแม่นยำในการตรวจจับเปลวไฟ เพื่อแจ้งเตือนการเกิดเพลิงไหม้ หากห้องมีขนาดกว้างหรือใหญ่มากๆ ควรเพิ่มจำนวนเซนเซอร์หรือใช้เซนเซอร์ที่มีระยะการตรวจจับครอบคลุมพื้นที่ภายในห้อง ในการติดตั้ง เนื่องจากเซนเซอร์แต่ละตัวมีการทำงานในช่วงที่แตกต่างกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

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

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

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

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

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

แหล่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แหล่งข้อมูล

1. ทดสอบเซ็นเซอร์ในรูปของสัญญาณดิจิตอล

```
const int analogPin = A0;
 // Flame Sensor (A0) to Arduino analog input pin A0

 void setup() {
    Serial.begin(9600);
 }

void loop() {
  // read the value of the Flame Sensor:
  int analogValue = analogRead(analogPin);
  Serial.println(analogValue);
  // serial print the Flame sensor Value

  delay(1);
}
```

2. ทดสอบการใช้งานตัวส่งสัญญาณ NRF24L01 Module

```
#include <SPI.h>
#include <nRF24L01p.h>

nRF24L01p transmitter(7,8);

void setup() {
  delay(150);
  Serial.begin(115200);
  SPI.begin();
  SPI.setBitOrder(MSBFIRST);
  transmitter.channel(90);
  transmitter.TXaddress("ALL");
  transmitter.init();
}

String message;

void loop () {
  transmitter.txPL("Welcome");
  transmitter.send(FAST);
  delay(1000);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ทดสอบการใช้งานตัวรับสัญญาณ NRF24L01 Module

```
#include <SPI.h>
#include <nRF24L01p.h>

nRF24L01p receiver(7,8);

void setup() {
  delay(150);
  Serial.begin(115200);
  SPI.begin();
  SPI.setBitOrder(MSBFIRST);
  receiver.channel(90);
  receiver.TXaddress("ALL");
  receiver.init();
}

String message;

void loop () {
  if(receiver.available()) {
    receiver.read();
  transmitter.rxPL(message);
  transmitter.println(message);
  message="";
}
}
```

4. ตัวส่งข้อมูล

```
#include <Wire.h>
#include "Adafruit_MCP23017.h"
#include <SPI.h>
#include <nRF24L01p.h>
nRF24L01p transmitter(7,8);
Adafruit_MCP23017 mcp;
byte pin[] = {0, 2, 3, 4};
String room_array[] = {"108", "201", "305","308"};
void setup() {
 mcp.begin(); //ใช้ตำแหน่งต่ำเริ่มต้นที่ 0

 for(byte i=0;i<sizeof(pin);i++){
  mcp.pinMode(pin[i], INPUT);
 }
Serial.begin(115200);
 mcp.pinMode(1, OUTPUT);
// mcp.pullUp(7, HIGH);

SPI.begin();
SPI.setBitOrder(MSBFIRST);
transmitter.channel(90);
transmitter.TXaddress("FIRE");
transmitter.init();
}
String message;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void loop() {
  String room = "";
  for(byte i=0;i<sizeof(pin);i++){
    if(mcp.digitalRead(pin[i]) == LOW){
      room = room+room_array[i];
    }
  }

  if(room.length()>0){
    Serial.println(room);
    mcp.digitalWrite(1,HIGH);
    delay(1);
    mcp.digitalWrite(1, LOW);
    transmitter.txPL(room);
  }else{
      Serial.println("no frame");
      transmitter.txPL("no frame");
      delay(1);
  }
  delay(1);

  transmitter.txPL(room);
  transmitter.send(FAST);
  delay(1);
}
```

5. ตัวรับข้อมูลและแอปพลิเคชันไลน์

```
#include <SPI.h>
#include <nRF24L01p.h>
#include <ESP8266WiFi.h>
nRF24L01p receiver(D8,D2);//CSN,CE
// Config connect WiFi
#define WIFI_SSID "filmmm"
#define WIFI_PASSWORD "kk123456"
String old_message = "";
// Line config
#define LINE_TOKEN "WIQO1srzDQOyaS5j6FgzRCGIVZKla1ZNRUvUB96vfGY"
void setup()
delay(150);
Serial.begin(115200);
SPI.begin();
SPI.setBitOrder(MSBFIRST);
receiver.channel(90);
receiver.RXaddress("FIRE");
receiver.init();
 WiFi.mode(WIFI_STA);
  // connect to wifi.
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("connecting");

  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
String message;

void loop()
{
if(receiver.available())
{
receiver.read();
receiver.rxPL(message);
Serial.println(message);
if(message != old_message){
  Line_Notify(message);
  old_message = message;
}
delay(50);
message="";
}
}

void Line_Notify(String message) {
  WiFiClientSecure client;

  if (!client.connect("notify-api.line.me", 443)) {
    Serial.println("connection failed");
    return;
  }

  String req = "";
  req += "POST /api/notify HTTP/1.1\r\n";
  req += "Host: notify-api.line.me\r\n";
  req += "Authorization: Bearer " + String(LINE_TOKEN) + "\r\n";
  req += "Cache-Control: no-cache\r\n";
  req += "User-Agent: ESP8266\r\n";
  req += "Content-Type: application/x-www-form-urlencoded\r\n";
  req += "Content-Length: " + String(String("message=" + message).length()) + "\r\n";
  req += "\r\n";
  req += "message=" + message;
  Serial.println(req);
  client.print(req);

  delay(20);
}
```

6. หน้าล็อกอิน

```
'
Public Class login
    Private Sub GhostButton1_Click(sender As System.Object, e As
System.EventArgs) Handles GhostButton1.Click

        If GhostTextBox1.Text = "" Or GhostTextBox2.Text = "" Then '
            MessageBox.Show("Please fill in complete data.", "ข้อความจากระบบ",
MessageBoxButtons.OK, MessageBoxIcon.Error)
        ElseIf GhostTextBox1.Text = "1" And GhostTextBox2.Text = "1" Then
            MessageBox.Show("Welcome to Fire alarm monitoring system", "ข้อความจาก
ระบบ", MessageBoxButtons.OK, MessageBoxIcon.Information)
            Me.Hide()
            type.Show()

        Else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MessageBox.Show("Username or Password incorrect", "ข้อความจากระบบ",
MessageBoxButtons.OK, MessageBoxIcon.Warning)
        End If

    End Sub


    Private Sub GhostButton2_Click(sender As System.Object, e As
System.EventArgs) Handles GhostButton2.Click
        Me.Close()
    End Sub
End Class
```

7. หน้ากรอกลายละเอียด

```
Public Class type

    Private Sub GhostButton2_Click(sender As System.Object, e As
System.EventArgs) Handles GhostButton2.Click
        Me.Close()
    End Sub

    Private Sub GhostButton1_Click(sender As System.Object, e As
System.EventArgs) Handles GhostButton1.Click
        Me.Close()
        confirm.Show()
        confirm.GhostTextBox2.Text = GhostComboBox2.SelectedIndex
        confirm.GhostTextBox3.Text = GhostComboBox1.SelectedIndex
    End Sub

    Private Sub GhostTheme1_Load(sender As System.Object, e As System.EventArgs)
Handles MyBase.Load
        Timer1.Start()
    End Sub

    Private Sub Timer1_Tick(sender As System.Object, e As System.EventArgs)
Handles Timer1.Tick
        Label1.Text = Now.ToShortDateString
        Label2.Text = TimeOfDay
    End Sub

    Private Sub GhostListboxLessPretty1_SelectedIndexChanged(sender As
System.Object, e As System.EventArgs)

    End Sub
End Class
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. หน้ายืนยันรายละเอียด

```
Private Sub GhostButton4_Click(sender As System.Object, e As System.EventArgs)
Handles GhostButton4.Click

        If GhostTextBox2.Text = "0" Or GhostTextBox3.Text = "0" Then
            MessageBox.Show("Your type is incorrect.", "ข้อความจากระบบ",
MessageBoxButtons.OK, MessageBoxIcon.Error)

        ElseIf GhostTextBox2.Text = "1" And GhostTextBox3.Text = "1" Then
            Me.Hide()
            display.Show()
        ElseIf GhostTextBox2.Text = "1" And GhostTextBox3.Text = "2" Then
            Me.Hide()
            display1.Show()
        ElseIf GhostTextBox2.Text = "1" And GhostTextBox3.Text = "3" Then
            Me.Hide()
            display2.Show()
        ElseIf GhostTextBox2.Text = "1" And GhostTextBox3.Text = "4" Then
            Me.Hide()
            display3.Show()
        ElseIf GhostTextBox2.Text = "1" And GhostTextBox3.Text = "5" Then
            Me.Hide()
            display4.Show()
        ElseIf GhostTextBox2.Text = "1" And GhostTextBox3.Text = "6" Then
            Me.Hide()
            display5.Show()
        ElseIf GhostTextBox2.Text = "1" And GhostTextBox3.Text = "7" Then
            Me.Hide()
            Form4.Show()
        ElseIf GhostTextBox2.Text = "1" And GhostTextBox3.Text = "8" Then
            Me.Hide()
            Form1.Show()
        ElseIf GhostTextBox2.Text = "2" And GhostTextBox3.Text = "1" Then
            Me.Hide()
            display6.Show()
        ElseIf GhostTextBox2.Text = "2" And GhostTextBox3.Text = "2" Then
            Me.Hide()
            display7.Show()
        ElseIf GhostTextBox2.Text = "2" And GhostTextBox3.Text = "3" Then
            Me.Hide()
            display8.Show()
        ElseIf GhostTextBox2.Text = "2" And GhostTextBox3.Text = "4" Then
            Me.Hide()
            display9.Show()
        ElseIf GhostTextBox2.Text = "2" And GhostTextBox3.Text = "5" Then
            Me.Hide()
            display10.Show()
        ElseIf GhostTextBox2.Text = "2" And GhostTextBox3.Text = "6" Then
            Me.Hide()
            display11.Show()
        ElseIf GhostTextBox2.Text = "2" And GhostTextBox3.Text = "7" Then
            Me.Hide()
            Form5.Show()
        ElseIf GhostTextBox2.Text = "2" And GhostTextBox3.Text = "8" Then
            Me.Hide()
            Form2.Show()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ElseIf GhostTextBox2.Text = "3" And GhostTextBox3.Text = "1" Then
            Me.Hide()
            display12.Show()
        ElseIf GhostTextBox2.Text = "3" And GhostTextBox3.Text = "2" Then
            Me.Hide()
            display13.Show()
        ElseIf GhostTextBox2.Text = "3" And GhostTextBox3.Text = "3" Then
            Me.Hide()
            display14.Show()
        ElseIf GhostTextBox2.Text = "3" And GhostTextBox3.Text = "4" Then
            Me.Hide()
            display15.Show()
        ElseIf GhostTextBox2.Text = "3" And GhostTextBox3.Text = "5" Then
            Me.Hide()
            display16.Show()
        ElseIf GhostTextBox2.Text = "3" And GhostTextBox3.Text = "6" Then
            Me.Hide()
            display17.Show()
        ElseIf GhostTextBox2.Text = "3" And GhostTextBox3.Text = "7" Then
            Me.Hide()
            Form6.Show()
        ElseIf GhostTextBox2.Text = "3" And GhostTextBox3.Text = "8" Then
            Me.Hide()
            Form3.Show()
        ElseIf GhostTextBox2.Text = "4" And GhostTextBox3.Text = "1" Then
            Me.Hide()
            display18.Show()
        ElseIf GhostTextBox2.Text = "4" And GhostTextBox3.Text = "2" Then
            Me.Hide()
            display19.Show()
        ElseIf GhostTextBox2.Text = "4" And GhostTextBox3.Text = "3" Then
            Me.Hide()
            display20.Show()
        ElseIf GhostTextBox2.Text = "4" And GhostTextBox3.Text = "4" Then
            Me.Hide()
            display21.Show()
        ElseIf GhostTextBox2.Text = "4" And GhostTextBox3.Text = "5" Then
            Me.Hide()
            display22.Show()
        ElseIf GhostTextBox2.Text = "4" And GhostTextBox3.Text = "6" Then
            Me.Hide()
            display23.Show()
        ElseIf GhostTextBox2.Text = "5" And GhostTextBox3.Text = "1" Then
            Me.Hide()
            display24.Show()
        ElseIf GhostTextBox2.Text = "5" And GhostTextBox3.Text = "2" Then
            Me.Hide()
            display25.Show()
        ElseIf GhostTextBox2.Text = "5" And GhostTextBox3.Text = "3" Then
            Me.Hide()
            display26.Show()
        ElseIf GhostTextBox2.Text = "5" And GhostTextBox3.Text = "4" Then
            Me.Hide()
            display27.Show()
        ElseIf GhostTextBox2.Text = "5" And GhostTextBox3.Text = "5" Then
            Me.Hide()
            display28.Show()
        End If
    End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. หน้าจอหลักแสดงผล

```
Public Class Form3
    Private Sub GhostTheme3_Load(sender As System.Object, e As System.EventArgs)
Handles MyBase.Load

        SerialPort1.PortName = "COM8"    'กำหนด Port ที่เชื่อต่อกับ Arduino

        SerialPort1.BaudRate = 115200 ' กำหนดอัตราการส่งข้อมูล

        Timer4.Enabled = True  'เปิดใช้งาน Timer

        Timer4.Interval = 1 ' เมื่อเวลาเดินครบ 1000 มิลลิวินาที ให้มาทำใน function ของ Timer

        SerialPort1.Open()  ' เปิด Port ที่จะส่งข้อมูล

    End Sub

    Private Sub Timer4_Tick(sender As System.Object, e As System.EventArgs)
Handles Timer4.Tick
        If SerialPort1.BytesToRead > 0 Then  ' ตรวจสอบว่าArduino มีการส่งข้อมูลเข้ามาทาง Port
ไหม

            Dim NewTemp() As String = Split(SerialPort1.ReadLine, ",")

            Dim TestLen As Integer = (NewTemp(0)).Length() - 1
            GhostTextBox1.Text = NewTemp(0) 'เอาค่าที่ Arduino ส่งมาแสดงใน Textbox
        End If
    End Sub

    Private Sub GhostTheme1_Load(sender As System.Object, e As System.EventArgs)
Handles MyBase.Load
        Timer1.Start()
    End Sub

    Private Sub Timer1_Tick(sender As System.Object, e As System.EventArgs)
Handles Timer1.Tick
        Label25.Text = Now.ToShortDateString
        Label26.Text = TimeOfDay
    End Sub

    Private Sub GhostButton96_Click(sender As System.Object, e As
System.EventArgs) Handles GhostButton96.Click
        Me.Close()
    End Sub

    Private Sub GhostButton95_Click(sender As System.Object, e As
System.EventArgs) Handles GhostButton95.Click
        Me.Hide()
        type.Show()
    End Sub

    Private Sub GhostTextBox1_TextChanged(sender As System.Object, e As
System.EventArgs) Handles GhostTextBox1.TextChanged
        Dim num1 As Integer = GhostTextBox1.Text.IndexOf("101")
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Dim num2 As Integer = GhostTextBox1.Text.IndexOf("102")
Dim num3 As Integer = GhostTextBox1.Text.IndexOf("103")
Dim num4 As Integer = GhostTextBox1.Text.IndexOf("104")
Dim num5 As Integer = GhostTextBox1.Text.IndexOf("105")
Dim num6 As Integer = GhostTextBox1.Text.IndexOf("106")
Dim num7 As Integer = GhostTextBox1.Text.IndexOf("107")
Dim num8 As Integer = GhostTextBox1.Text.IndexOf("108")
Dim num9 As Integer = GhostTextBox1.Text.IndexOf("201")
Dim num10 As Integer = GhostTextBox1.Text.IndexOf("202")
Dim num11 As Integer = GhostTextBox1.Text.IndexOf("203")
Dim num12 As Integer = GhostTextBox1.Text.IndexOf("204")
Dim num13 As Integer = GhostTextBox1.Text.IndexOf("205")
Dim num14 As Integer = GhostTextBox1.Text.IndexOf("206")
Dim num15 As Integer = GhostTextBox1.Text.IndexOf("207")
Dim num16 As Integer = GhostTextBox1.Text.IndexOf("208")
Dim num17 As Integer = GhostTextBox1.Text.IndexOf("301")
Dim num18 As Integer = GhostTextBox1.Text.IndexOf("302")
Dim num19 As Integer = GhostTextBox1.Text.IndexOf("303")
Dim num20 As Integer = GhostTextBox1.Text.IndexOf("304")
Dim num21 As Integer = GhostTextBox1.Text.IndexOf("305")
Dim num22 As Integer = GhostTextBox1.Text.IndexOf("306")
Dim num23 As Integer = GhostTextBox1.Text.IndexOf("307")
Dim num24 As Integer = GhostTextBox1.Text.IndexOf("308")
If num1 >= 0 Then
    Console.Beep(1000, 3000)
End If
If num2 >= 0 Then
    Console.Beep(1000, 3000)
End If
If num3 >= 0 Then
    Console.Beep(1000, 3000)
End If
If num4 >= 0 Then
    Console.Beep(1000, 3000)
End If
If num5 >= 0 Then
    Console.Beep(1000, 3000)
End If
If num6 >= 0 Then
    Console.Beep(1000, 3000)
End If
If num7 >= 0 Then
    Console.Beep(1000, 3000)
End If
If num8 >= 0 Then
    Console.Beep(1000, 3000)
End If
If num9 >= 0 Then
    Console.Beep(1000, 3000)
End If
If num10 >= 0 Then
    Console.Beep(1000, 3000)
End If
If num11 >= 0 Then
    Console.Beep(1000, 3000)
End If
If num12 >= 0 Then
    Console.Beep(1000, 3000)
End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        If num13 >= 0 Then
            Console.Beep(1000, 3000)
        End If
        If num14 >= 0 Then
            Console.Beep(1000, 3000)
        End If
        If num15 >= 0 Then
            Console.Beep(1000, 3000)
        End If
        If num16 >= 0 Then
            Console.Beep(1000, 3000)
        End If
        If num17 >= 0 Then
            Console.Beep(1000, 3000)
        End If
        If num18 >= 0 Then
            Console.Beep(1000, 3000)
        End If
        If num19 >= 0 Then
            Console.Beep(1000, 3000)
        End If
        If num20 >= 0 Then
            Console.Beep(1000, 3000)
        End If
        If num21 >= 0 Then
            Console.Beep(1000, 3000)
        End If
        If num22 >= 0 Then
            Console.Beep(1000, 3000)
        End If
        If num23 >= 0 Then
            Console.Beep(1000, 3000)
        End If
        If num24 >= 0 Then
            Console.Beep(1000, 3000)
        End If

    End Sub

    Private Sub Timer2_Tick(sender As System.Object, e As System.EventArgs)
Handles Timer2.Tick
        Dim num1 As Integer = GhostTextBox1.Text.IndexOf("101")
        Dim num2 As Integer = GhostTextBox1.Text.IndexOf("102")
        Dim num3 As Integer = GhostTextBox1.Text.IndexOf("103")
        Dim num4 As Integer = GhostTextBox1.Text.IndexOf("104")
        Dim num5 As Integer = GhostTextBox1.Text.IndexOf("105")
        Dim num6 As Integer = GhostTextBox1.Text.IndexOf("106")
        Dim num7 As Integer = GhostTextBox1.Text.IndexOf("107")
        Dim num8 As Integer = GhostTextBox1.Text.IndexOf("108")
        Dim num9 As Integer = GhostTextBox1.Text.IndexOf("201")
        Dim num10 As Integer = GhostTextBox1.Text.IndexOf("202")
        Dim num11 As Integer = GhostTextBox1.Text.IndexOf("203")
        Dim num12 As Integer = GhostTextBox1.Text.IndexOf("204")
        Dim num13 As Integer = GhostTextBox1.Text.IndexOf("205")
        Dim num14 As Integer = GhostTextBox1.Text.IndexOf("206")
        Dim num15 As Integer = GhostTextBox1.Text.IndexOf("207")
        Dim num16 As Integer = GhostTextBox1.Text.IndexOf("208")
        Dim num17 As Integer = GhostTextBox1.Text.IndexOf("301")
        Dim num18 As Integer = GhostTextBox1.Text.IndexOf("302")
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Dim num19 As Integer = GhostTextBox1.Text.IndexOf("303")
Dim num20 As Integer = GhostTextBox1.Text.IndexOf("304")
Dim num21 As Integer = GhostTextBox1.Text.IndexOf("305")
Dim num22 As Integer = GhostTextBox1.Text.IndexOf("306")
Dim num23 As Integer = GhostTextBox1.Text.IndexOf("307")
Dim num24 As Integer = GhostTextBox1.Text.IndexOf("308")
If num1 >= 0 Then
    Label1.Enabled = True
    Timer3.Enabled = True
    Timer2.Enabled = False
End If
If num2 >= 0 Then
    Label2.Enabled = True
    Timer3.Enabled = True
    Timer2.Enabled = False
End If
If num3 >= 0 Then
    Label3.Enabled = True
    Timer3.Enabled = True
    Timer2.Enabled = False
End If
If num4 >= 0 Then
    Label4.Enabled = True
    Timer3.Enabled = True
    Timer2.Enabled = False
End If
If num5 >= 0 Then
    Label5.Enabled = True
    Timer3.Enabled = True
    Timer2.Enabled = False
End If
If num6 >= 0 Then
    Label6.Enabled = True
    Timer3.Enabled = True
    Timer2.Enabled = False
End If
If num7 >= 0 Then
    Label7.Enabled = True
    Timer3.Enabled = True
    Timer2.Enabled = False
End If
If num8 >= 0 Then
    Label8.Enabled = True
    Timer3.Enabled = True
    Timer2.Enabled = False
End If
If num9 >= 0 Then
    Label9.Enabled = True
    Timer3.Enabled = True
    Timer2.Enabled = False
End If
If num10 >= 0 Then
    Label10.Enabled = True
    Timer3.Enabled = True
    Timer2.Enabled = False
End If
If num11 >= 0 Then
    Label11.Enabled = True
    Timer3.Enabled - True
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Timer2.Enabled = False
End If
If num12 >= 0 Then
    Label12.Enabled = True
    Timer3.Enabled = True
    Timer2.Enabled = False
End If
If num13 >= 0 Then
    Label13.Enabled = True
    Timer3.Enabled = True
    Timer2.Enabled = False
End If
If num14 >= 0 Then
    Label14.Enabled = True
    Timer3.Enabled = True
    Timer2.Enabled = False
End If
If num15 >= 0 Then
    Label15.Enabled = True
    Timer3.Enabled = True
    Timer2.Enabled = False
End If
If num16 >= 0 Then
    Label16.Enabled = True
    Timer3.Enabled = True
    Timer2.Enabled = False
End If
If num17 >= 0 Then
    Label17.Enabled = True
    Timer3.Enabled = True
    Timer2.Enabled = False
End If
If num18 >= 0 Then
    Label18.Enabled = True
    Timer3.Enabled = True
    Timer2.Enabled = False
End If
If num19 >= 0 Then
    Label19.Enabled = True
    Timer3.Enabled = True
    Timer2.Enabled = False
End If
If num20 >= 0 Then
    Label20.Enabled = True
    Timer3.Enabled = True
    Timer2.Enabled = False
End If
If num21 >= 0 Then
    Label21.Enabled = True
    Timer3.Enabled = True
    Timer2.Enabled = False
End If
If num22 >= 0 Then
    Label22.Enabled = True
    Timer3.Enabled = True
    Timer2.Enabled = False
End If
If num23 >= 0 Then
    Label23.Enabled = True
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Timer3.Enabled = True
            Timer2.Enabled = False
        End If
        If num24 >= 0 Then
            Label24.Enabled = True
            Timer3.Enabled = True
            Timer2.Enabled = False
        End If
    End Sub


    Private Sub Timer3_Tick(sender As System.Object, e As System.EventArgs)
Handles Timer3.Tick
        Dim num1 As Integer = GhostTextBox1.Text.IndexOf("101")
        Dim num2 As Integer = GhostTextBox1.Text.IndexOf("102")
        Dim num3 As Integer = GhostTextBox1.Text.IndexOf("103")
        Dim num4 As Integer = GhostTextBox1.Text.IndexOf("104")
        Dim num5 As Integer = GhostTextBox1.Text.IndexOf("105")
        Dim num6 As Integer = GhostTextBox1.Text.IndexOf("106")
        Dim num7 As Integer = GhostTextBox1.Text.IndexOf("107")
        Dim num8 As Integer = GhostTextBox1.Text.IndexOf("108")
        Dim num9 As Integer = GhostTextBox1.Text.IndexOf("201")
        Dim num10 As Integer = GhostTextBox1.Text.IndexOf("202")
        Dim num11 As Integer = GhostTextBox1.Text.IndexOf("203")
        Dim num12 As Integer = GhostTextBox1.Text.IndexOf("204")
        Dim num13 As Integer = GhostTextBox1.Text.IndexOf("205")
        Dim num14 As Integer = GhostTextBox1.Text.IndexOf("206")
        Dim num15 As Integer = GhostTextBox1.Text.IndexOf("207")
        Dim num16 As Integer = GhostTextBox1.Text.IndexOf("208")
        Dim num17 As Integer = GhostTextBox1.Text.IndexOf("301")
        Dim num18 As Integer = GhostTextBox1.Text.IndexOf("302")
        Dim num19 As Integer = GhostTextBox1.Text.IndexOf("303")
        Dim num20 As Integer = GhostTextBox1.Text.IndexOf("304")
        Dim num21 As Integer = GhostTextBox1.Text.IndexOf("305")
        Dim num22 As Integer = GhostTextBox1.Text.IndexOf("306")
        Dim num23 As Integer = GhostTextBox1.Text.IndexOf("307")
        Dim num24 As Integer = GhostTextBox1.Text.IndexOf("308")
        If num1 >= 0 Then
            Label1.Enabled = False
            Timer2.Enabled = True
            Timer3.Enabled = False
        End If
        If num2 >= 0 Then
            Label2.Enabled = False
            Timer2.Enabled = True
            Timer3.Enabled = False
        End If
        If num3 >= 0 Then
            Label3.Enabled = False
            Timer2.Enabled = True
            Timer3.Enabled = False
        End If
        If num4 >= 0 Then
            Label4.Enabled = False
            Timer2.Enabled = True
            Timer3.Enabled = False
        End If
        If num5 >= 0 Then
            Label5.Enabled = False
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Timer2.Enabled = True
        Timer3.Enabled = False
    End If
    If num6 >= 0 Then
        Label6.Enabled = False
        Timer2.Enabled = True
        Timer3.Enabled = False
    End If
    If num7 >= 0 Then
        Label7.Enabled = False
        Timer2.Enabled = True
        Timer3.Enabled = False
    End If
    If num8 >= 0 Then
        Label8.Enabled = False
        Timer2.Enabled = True
        Timer3.Enabled = False
    End If
    If num9 >= 0 Then
        Label9.Enabled = False
        Timer2.Enabled = True
        Timer3.Enabled = False
    End If
    If num10 >= 0 Then
        Label10.Enabled = False
        Timer2.Enabled = True
        Timer3.Enabled = False
    End If
    If num11 >= 0 Then
        Label11.Enabled = False
        Timer2.Enabled = True
        Timer3.Enabled = False
    End If
    If num12 >= 0 Then
        Label12.Enabled = False
        Timer2.Enabled = True
        Timer3.Enabled = False
    End If
    If num13 >= 0 Then
        Label13.Enabled = False
        Timer2.Enabled = True
        Timer3.Enabled = False
    End If
    If num14 >= 0 Then
        Label14.Enabled = False
        Timer2.Enabled = True
        Timer3.Enabled = False
    End If
    If num15 >= 0 Then
        Label15.Enabled = False
        Timer2.Enabled = True
        Timer3.Enabled = False
    End If
    If num16 >= 0 Then
        Label16.Enabled = False
        Timer2.Enabled = True
        Timer3.Enabled = False
    End If
    If num17 >= 0 Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Label17.Enabled = False
            Timer2.Enabled = True
            Timer3.Enabled = False
        End If
        If num18 >= 0 Then
            Label18.Enabled = False
            Timer2.Enabled = True
            Timer3.Enabled = False
        End If
        If num19 >= 0 Then
            Label19.Enabled = False
            Timer2.Enabled = True
            Timer3.Enabled = False
        End If
        If num20 >= 0 Then
            Label20.Enabled = False
            Timer2.Enabled = True
            Timer3.Enabled = False
        End If
        If num21 >= 0 Then
            Label21.Enabled = False
            Timer2.Enabled = True
            Timer3.Enabled = False
        End If
        If num22 >= 0 Then
            Label22.Enabled = False
            Timer2.Enabled = True
            Timer3.Enabled = False
        End If
        If num23 >= 0 Then
            Label23.Enabled = False
            Timer2.Enabled = True
            Timer3.Enabled = False
        End If
        If num24 >= 0 Then
            Label24.Enabled = False
            Timer2.Enabled = True
            Timer3.Enabled = False
        End If

    End Sub


End Class
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้