

ระบบเตือนภัย
ผ่านทาง การส่งข้อความสั้น และโครงข่ายอินเทอร์เน็ต
Security Systems via SMS and Internet



มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าลาดกระบัง
[Signature]



โดย
นายรัชชัย คตดี
นายราชกิจ สกุดดี
นายวรรณลพ ดุลนีย์

เลขหมู่.....
เลขทะเบียน 61961
วัน,เดือน,ปี 25 ก.ค. 2549

b.....
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2547

[Signature]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดสิทธิ์การพิมพ์หรือการนำเอกสารทุกครั้งที่มีการนำไปใช้



ระบบเตือนภัย
ผ่านทาง การส่งข้อความสั้น และโครงข่ายอินเทอร์เน็ต

Security Systems via SMS and Internet



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2547

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบเตือนภัยผ่านทาง การส่งข้อความสั้น และโครงข่ายอินเทอร์เน็ต

Security Systems via SMS and Internet

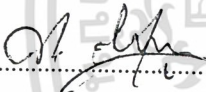
ผู้จัดทำ

1. นาย ธวัชชัย กตติ 45015011
2. นาย ราชกิจ สกุสดี 45015028
3. นาย วรรณลพ คุณนีย์ 45015030



..... อาจารย์ที่ปรึกษา

(รศ.ดร. กอบชัย เดชหาญ)



..... อาจารย์ที่ปรึกษา

(อ. สรวัดน์ ชิวปรีชา)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบเตือนภัยผ่านทาง การส่งข้อความสั้น และโครงข่ายอินเทอร์เน็ต
Security Systems via SMS and Internet

โดย นายรัชชัย กตดี 45015051
นายราชกิจ สกุลดี 45015052
นายวรรณพล คุณนีย์ 45015070

อาจารย์ที่ปรึกษา รศ.ดร. กอบชัย เดชหาญ
อ. ศรวัฒน์ ชิวปรีชา

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้เป็นการนำเสนอระบบเตือนภัย เช่น มีผู้บุกรุกเข้ามาในบ้าน หรือเกิดไฟไหม้ ก็จะทำการส่งข้อความสั้นเตือนภัยไปยัง โทรศัพท์เคลื่อนที่ของเจ้าของบ้าน เมื่อเจ้าของบ้านรับทราบ จะสามารถตรวจสอบสถานะ และควบคุมการทำงาน ผ่านทางเว็บเบราว์เซอร์ โดยการใช้ไมโครคอนโทรลเลอร์ Rabbit RCM2200 มาประยุกต์ใช้งานผ่านระบบเครือข่าย โดยใช้โปรโตคอล TCP/IP ในการติดต่อสื่อสาร

ABSTRACT

This project presents the security systems via SMS and internet, such as when some invader enters the house area and when the house go up in smoke. Then it automatically sends the SMS to you own mobile phone. After that you answer this sends SMS already, could to check status and control the system via web browser with the microcontroller Rabbit RCM2200 for applied to work that through network system which uses TCP/IP communication standart.

สารบัญ

เรื่อง	หน้า
บทคัดย่อ	
สารบัญ	
สารบัญรูป	
สารบัญตาราง	
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีหรือหลักการ	2
2.1 ลักษณะที่สำคัญและความสามารถของ RCM2200	2
2.2 ส่วนประกอบและอุปกรณ์ย่อยของระบบ RCM2200	3
2.3 รายละเอียดของไมโครโปรเซสเซอร์ Rabbit2000	6
2.4 หน้าที่ของขาอินพุต/เอาต์พุตบนไมโครโปรเซสเซอร์ Rabbit2000	7
2.5 พอร์ตในการเชื่อมต่อ RabbitCore RCM2200	8
2.6 Parallel Ports	9
2.7 โปรแกรม Dynamic C	12
2.8 HTTP Server	16
2.9 Configuration Macros	17
2.10 ทฤษฎีเกี่ยวกับเครือข่ายเน็ตเวิร์ค (Network)	19
2.11 คำสั่ง AT commands	25
2.12 ระบบการทำงานการรับส่งข้อความในโทรศัพท์เคลื่อนที่	27
บทที่ 3 การคำนวณและการสร้าง	30
3.1 การออกแบบวงจรหลัก	30
3.2 อุปกรณ์ตรวจจบการเปิด / ปิด	31
3.3 อุปกรณ์ตรวจจบควีน	32
3.4 ไอซี เบอร์ TA7279P	33
3.5 ทฤษฎีของไดรแอกเพื่อใช้ในการตัดต่อกับอุปกรณ์ไฟฟ้าภายนอก	35
3.6 วงจรตรวจสอบสถานะ	38
3.7 การนำไอซี เบอร์ 74LS139 ไปใช้งาน	40
3.8 แหล่งจ่ายไฟหลักและไฟสำรอง	41
3.9 สาย UTP (Unshieled Twisted Pair)	43
3.10 วิธีหา IP Address : Windows 2000/XP	45
3.11 หลักการอินเตอร์เฟสมือถือ	46
3.12 ขั้นตอนการออกแบบ Web Server	47
3.13 โฟลว์ชาร์ตการทำงานแต่ละส่วน	48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
บทที่ 4 การทดลองและผลการทดลอง	60
4.1 อุปกรณ์ที่ใช้ในการทดลอง	60
4.2 การทดลองการรับค่าจากชุด Manual Switch	62
4.3 การทดลองการส่งค่า	62
4.4 การทดลองการทำงานของ Web Browser	62
4.5 การวัดสัญญาณ	69
บทที่ 5 บทวิจารณ์และบทสรุป	74
5.1 สรุปปัญหาที่พบใน โครงการงาน	74
5.2 สรุปผลของการดำเนิน โครงการงาน	74
5.3 แนวทางการพัฒนาโครงการ	74
5.4 ข้อดีของการใช้งาน Short Message	75
ภาคผนวก ก. โปรแกรม (วงจร)	76
ภาคผนวก ข. โปรแกรม (Software)	81
บรรณานุกรม	97



สารบัญรูป

	หน้า
รูปที่ 2.1 บอร์ดวงจร RCM2200	2
รูปที่ 2.2 ระบบภายใน RCM2200	2
รูปที่ 2.3 หัวข้อต่อสัญญาณที่ต่อออกใช้งานภายนอกของ RabbitCore RCM2200	3
รูปที่ 2.4 การติดต่อกับหน่วยความจำ(RAM)	6
รูปที่ 2.5 โครงสร้างภายในของ Rabbit2000	6
รูปที่ 2.6 บล็อกไดอะแกรม ของพอร์ตอนุกรม	8
รูปที่ 2.7 พอร์ตการเขียน โปรแกรม	9
รูปที่ 2.8 ชั้นสื่อสาร,กฎการสื่อสารข้อมูลและการเชื่อมต่อ	19
รูปที่ 2.9 โครงสร้าง TCP/IP	21
รูปที่ 2.10 กฎการสื่อสารในรูปแบบ TCP/IP	23
รูปที่ 2.11 โครงสร้างที่อยู่แบบ IP	24
รูปที่ 2.12 การรับส่งข้อความแบบพีดียูโหมด	27
รูปที่ 2.13 การกำหนดแอดเดรสสำหรับคลาสต่างๆ	35
รูปที่ 3.1 ไดอะแกรมของระบบ	30
รูปที่ 3.2 รูปร่างของสวิทช์แม่เหล็ก	31
รูปที่ 3.3 โครงสร้างของสวิทช์แม่เหล็ก	32
รูปที่ 3.4 วงจรเซนเซอร์	32
รูปที่ 3.5 วงจรตรวจสอบควีน	33
รูปที่ 3.6 ขาของไอซี TA7279P และ โครงสร้างภายใน	34
รูปที่ 3.7 การต่อไอซี TA7279P เพื่อใช้งานในวงจรรวม	34
รูปที่ 3.8 โครงสร้าง, สัญลักษณ์และสัญลักษณ์เสมือนของไตรแอก	35
รูปที่ 3.9 การใช้งานไตรแอกเป็นสวิทช์ในวงจรไฟฟ้า	36
รูปที่ 3.10 วงจรส่วนที่ใช้ในการตัดต่อกับอุปกรณ์ไฟฟ้าภายนอก	37
รูปที่ 3.11 วงจรตรวจสอบสถานะ	39
รูปที่ 3.12 ตำแหน่งขาของไอซี 74LS139 และ โครงสร้างภายใน	40
รูปที่ 3.13 การต่อไอซี 74LS139 เพื่อใช้งาน	41
รูปที่ 3.14 หลักการของระบบไฟหลักและไฟสำรอง	41
รูปที่ 3.15 วงจรแหล่งจ่ายไฟหลักและไฟสำรอง	42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

	หน้า
รูปที่ 3.16 แสดงลักษณะของสาย UTP	43
รูปที่ 3.17 แสดงการหาค่า IP	45
รูปที่ 3.18 แสดงค่า IP	45
รูปที่ 3.19 โฟลว์ชาร์ตการทำงานของโปรแกรมหลัก	48
รูปที่ 3.20 โฟลว์ชาร์ตระบบแสดงผลของ Sensor ผ่าน Web Browser	49
รูปที่ 3.21 โฟลว์ชาร์ตการทำงานของ Manual Switch	51
รูปที่ 3.22 โฟลว์ชาร์ตการทำงานของระบบ System ในสถานะ ON / OFF	52
รูปที่ 3.23 โฟลว์ชาร์ตการทำงานของระบบ Alarm ในสถานะ ON / OFF	52
รูปที่ 3.24 โฟลว์ชาร์ตการทำงานของระบบ AC ในสถานะ ON / OFF	53
รูปที่ 3.25 โฟลว์ชาร์ตการทำงานของระบบ Lock ในสถานะ ON / OFF	54
รูปที่ 3.26 โฟลว์ชาร์ตตรวจสอบระบบ System Scan	55
รูปที่ 3.27 โฟลว์ชาร์ตเปิด และปิดระบบเตือนภัย	57
รูปที่ 3.28 โฟลว์ชาร์ตเปิด และปิดระบบเตือนภัยอัคคีภัย	58
รูปที่ 3.29 โฟลว์ชาร์ตส่งข้อความสั้น เข้าโทรศัพท์เคลื่อนที่ เมื่อมีเหตุผิดปกติ	59
รูปที่ 4.1 วงจรภายในของระบบเตือนภัย	60
รูปที่ 4.2 การติดตั้งระบบเซนเซอร์ และมอเตอร์	61
รูปที่ 4.3 ชุดระบบเตือนภัย	61
รูปที่ 4.4 แสดงหน้า Web Browser เพื่อการเข้าใช้ระบบ	63
รูปที่ 4.5 ผลบนหน้า Web Browser ในกรณีปกติ	63
รูปที่ 4.6 ผลบนหน้า Web Browser ในกรณีที่ทำให้การเปิดประตูที่ 1	64
รูปที่ 4.7 ผลบนหน้า Web Browser ในกรณีที่ทำให้การกดปุ่มล๊อค ประตูที่ 2	64
รูปที่ 4.8 แสดงผลการสั่งงานการล๊อคประตูที่ 2	65
รูปที่ 4.9 ผลบน Web Browser เมื่อทำการเปิดระบบเตือนภัย	66
รูปที่ 4.10 ผลบน Web Browser แจ้งเตือนเมื่อมีผู้บุกรุก เข้ามาที่ประตูที่ 1	67
รูปที่ 4.11 ข้อความ “Door1 Open!” เตือนภัยที่โทรศัพท์ปลายทาง	67
รูปที่ 4.12 ผลบน Web Browser แจ้งเตือนเมื่อมีควัน	68
รูปที่ 4.13 ข้อความ “Warning Fire!” เตือนภัยที่โทรศัพท์ปลายทาง	68
รูปที่ 4.14 แสดงสถานะของหน้าเว็บเพจ	69

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

	หน้า
รูปที่ 4.15 แสดงสถานะหน้าเว็บเพจเมื่อมีการกดปุ่ม	70
รูปที่ 4.16 แสดงสถานะแรงดันเมื่อทำการปิดประตู	70
รูปที่ 4.17 แสดงสถานะแรงดันเมื่อทำการเปิดประตู	71
รูปที่ 4.18 แสดงสถานะแรงดันกรณีที่ไม่มีควัน	71
รูปที่ 4.19 แสดงสถานะแรงดันกรณีที่มีควัน	72
รูปที่ 4.20 สัญญาณการตอบกลับของการส่งข้อความ	73



สารบัญตาราง

	หน้า
ตารางที่ 2.1 ความสามารถในการนำไปใช้งานของคอนเน็คเตอร์ J4	4
ตารางที่ 2.2 ความสามารถในการนำไปใช้งานของหัวต่อสัญญาณ J5	5
ตารางที่ 2.3 รายละเอียดของข้อมูลในการส่ง SMS	26
ตารางที่ 2.4 การแปลงเลขฐานสิบหกเป็นเลขฐานสอง	27
ตารางที่ 2.5 การแปลงเลขฐานสองเป็นเลขพีดิวโคด	28
ตารางที่ 2.6 การแปลงเลขพีดิวโคดเป็นเลขฐานสอง	28
ตารางที่ 2.7 การแปลงเลขฐานสองเป็นรหัสแอสกีโคด	28
ตารางที่ 3.1 แสดงสถานะ และการทำงานของมอเตอร์	33
ตารางที่ 3.2 แสดงหน้าที่ของสาย UTP	43
ตารางที่ 3.3 แสดงการต่อเชื่อมระหว่างคอมพิวเตอร์ กับ RCM 2200	44
ตารางที่ 3.4 แสดงการต่อเชื่อมระหว่างฮับ กับ RCM 2200	44
ตารางที่ 3.5 พอร์ตของมือถือ C35	46
ตารางที่ 4.1 แสดงค่าที่รับเข้ามาทาง PB ของ IC 8255	62

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ในปัจจุบันนี้ทุกคนต้องมีภารกิจที่ต้องทำในแต่ละวัน ดังนั้นจำเป็นที่ต้องออกจากบ้านเพื่อทำภารกิจ จึงมีความจำเป็นต้องทิ้งบ้าน ทำให้ไม่ทราบว่าจะมีภัยที่ไม่พึงปรารถนาเกิดขึ้นที่บ้านหรือสำนักงานเมื่อไร โดยเฉพาะภัยที่สร้างความเสียหายแก่ชีวิตและทรัพย์สิน ด้วยเหตุผลที่กล่าวมานี้จึงได้มีแนวคิดที่จะทำโครงการที่ใช้ในการเตือนภัยหรือแจ้งเหตุขึ้น เพื่อความไว้วางใจได้ระดับหนึ่ง ซึ่งระบบการเตือนภัยที่ดีจำเป็นต้องมีความรวดเร็วในการแจ้งเหตุทันทีทันใด

สำหรับรูปแบบการทำงาน เมื่อเกิดเหตุการณ์ที่ไม่พึงปรารถนา ระบบก็จะตรวจจับความผิดปกติแล้วส่งสัญญาณเข้า RCM2200 เพื่อแจ้งสถานะของเซนเซอร์ไปที่หน้าจอเว็บเพจ แล้วทำการส่งสัญญาณเพื่อให้ MCS 51 ทำการควบคุมโทรศัพท์เคลื่อนที่ต้นทาง ให้ทำการแจ้งเหตุไปยังโทรศัพท์เคลื่อนที่ปลายทาง ในขณะเดียวกันที่ปลายทางก็จะสามารถตรวจสอบสถานะภาพการเปลี่ยนแปลงที่เกิดขึ้นด้วยว่าเกิดขึ้นที่ใดบ้างโดยตรวจสอบได้ที่ เว็บเบราว์เซอร์ ในเครือข่ายอินเทอร์เน็ต ในขณะเดียวกันก็ยังสามารถควบคุมกลับทางเว็บเพจได้อีกด้วย

1.1 วัตถุประสงค์

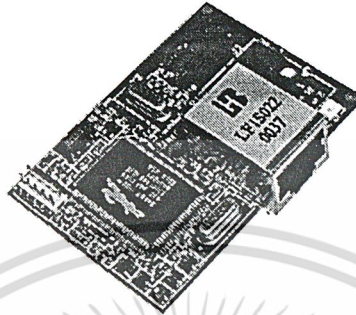
- เพื่อสร้างระบบเตือนภัยผู้บุกรุกผ่านทางโทรศัพท์มือถือ
- เพื่อศึกษาการส่งข้อความสั้น (SMS)
- เพื่อศึกษาการทำงานของ Rabbit Core Module RCM2200
- เพื่อสร้างระบบเตือนภัยในหน้าเว็บเพจ
- เพื่อสร้างระบบควบคุมผ่านทางเครือข่ายอินเทอร์เน็ต

1.2 ขอบเขตของโครงการ

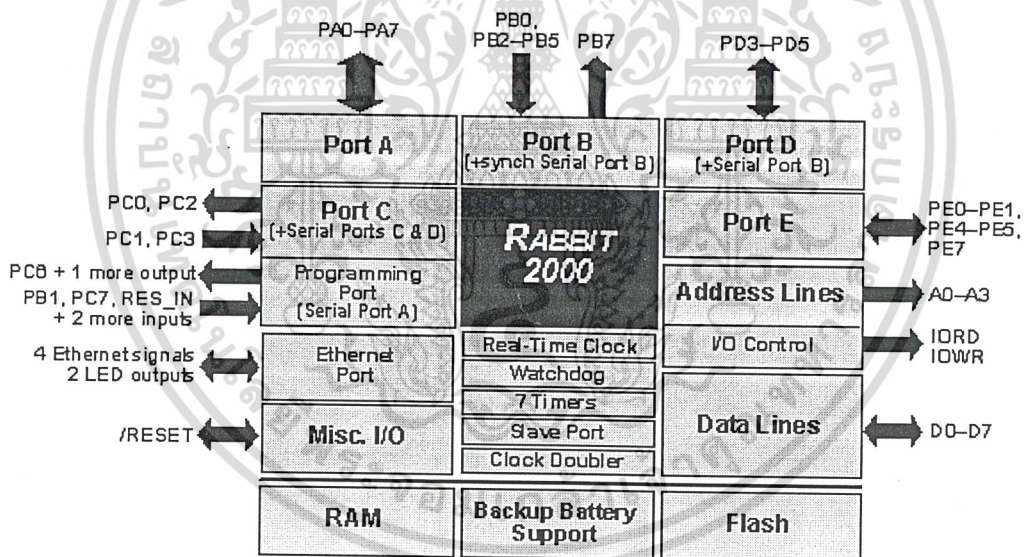
เพื่อนำไมโครคอนโทรลเลอร์ RCM2200 เป็นตัวเชื่อมต่อและควบคุมการทำงานของระบบรวมทั้งตรวจสอบสถานะการทำงานของเซนเซอร์ ผ่านทางเว็บเบราว์เซอร์ แล้วทำการแจ้งเตือนภัยด้วยการส่ง SMS ไปยังโทรศัพท์มือถือปลายทาง

บทที่ 2 ทฤษฎีและหลักการ

2.1 ลักษณะที่สำคัญและความสามารถของ RCM2200



รูปที่ 2.1 บอร์ดวงจร RCM2200



รูปที่ 2.2 ระบบภายใน RCM2200

- ขนาดเล็ก 1.60" x 2.30" x 0.86" หรือ (41 mm x 58 mm x 22 mm)
- ไมโครโปรเซสเซอร์ Rabbit 2000 ใช้ความถี่ในการทำงานที่ 22.1 MHz
- มีขาที่เป็น Input/Output ทั้งหมด 26 ขา เป็นแบบ Parallel มีโหมดการทำงาน หลากหลาย
- มี Data Bus 8 เส้น (D0 - D7)
- มี Address Bus 4 เส้น (A0 - A3)
- มี Flash Memory ขนาด 256K และ Static Ram ขนาด 128K
- มีสัญญาณนาฬิกาให้จังหวะการทำงาน

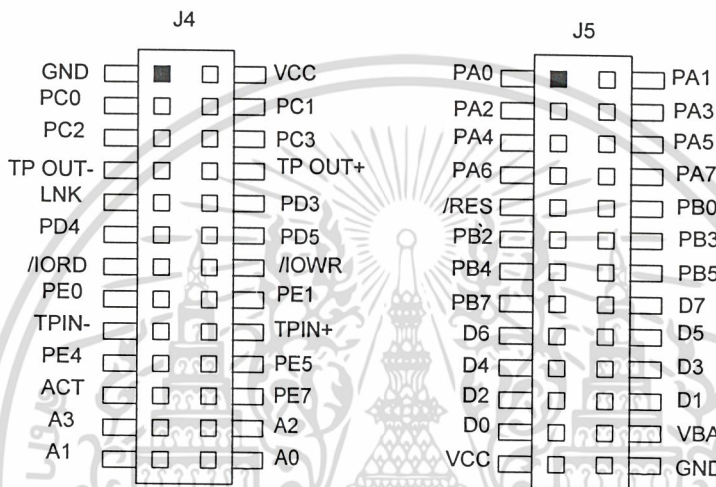
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ใช้หัวต่อสัญญาณแบบ RJ-45 เป็น Ethernet Port
- Port Serial เป็นชนิด CMOS ซึ่งเป็นอัตราบอร์ด์แบบ Asynchronous ได้ความถี่สูงสุดถึง 345,600 bps อัตราบอร์ด์ที่ใช้แบบ Synchronous ได้ความถี่สูงสุดถึง 138,240 bps

2.2 ส่วนประกอบและอุปกรณ์ย่อยของระบบ RCM2200

2.2.1 อินพุต/เอาต์พุต ของ RCM2200

RCM2200 มี Input/Output ที่แบ่งเป็น Port 5 Port แต่ละ Port มี 8 บิต อยู่ที่หัวข้อต่อสัญญาณ J4 และ J5 โดยที่เส้น Input /Output จะอยู่ที่ PA0-PA7, PD3-PD5, PE0-PE1, PE4, PE5 และ PE7 สำหรับหัวข้อต่อสัญญาณ J4 และ J5 ถูกแสดงในรูปที่ 2.3



รูปที่ 2.3 หัวข้อต่อสัญญาณที่ต่อออกใช้งานภายนอกของ RCM2200

2.2.2 ขาสัญญาณอินพุต

ในส่วนองขาที่ทำหน้าที่เป็นขาอินพุตทั่วไป คือขา PB0 -PB5 เมื่อ RCM2200 ไม่มีการใช้พอร์ต B หรือการใช้พอร์ต B ในโหมด Asynchronous มีสี่ขาที่ใช้งานเป็นอินพุต คือขา PB2 -PB5 ขาเหล่านี้สามารถใช้สำหรับพอร์ตควบคุม

ขา PB2 และ PB3 คือขาที่ควบคุมการเขียนและควบคุมการอ่าน strobes ขณะที่ PB4 และ PB5 ใช้สำหรับควบคุมบัสตำแหน่ง (address bus) คือขา A0 และ A1 และใช้เพื่อเข้าถึงข้อมูลในรีจิสเตอร์ (registers)

ขา PC1 และ PC3 ทั้งสองขาสัญญาณนี้สามารถใช้แทนอินพุตและเอาต์พุตของข้อมูลสำหรับพอร์ต D และ C

2.2.3 ขาสัญญาณเอาต์พุต

ขาที่ทำหน้าที่เป็นขาอินพุตทั่วไป คือขา PB7 สามารถใช้ควบคุมพอร์ตเอาต์พุต ของ SLAVE ATTN ขา PC0 และ PC2 เป็นขาสัญญาณเอาต์พุตเพียงอย่างเดียวสามารถแทนในการส่งออกข้อมูลสำหรับพอร์ต D และ C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขาสัญญาณ	ชื่อขาสัญญาณ	Default Use	Alternate Use	Notes	
Header J4	1	GND			
	2	VCC			
	3	PC0	Output	TXD	
	4	PC1	Input	RXD	
	5	PC2	Output	TXC	
	6	PC3	Input	RXC	
	7	TPOUT-			Etehernat transmit port
	8	TPOUT+			
	9	LNK			Etehernat LNK LED Indicator
	10	PD3	Bitwise or Parallel Programmable I/O		
	11	PD4		ATXB output	
	12	PD5		ARXB input	
	13	/IORD	Input (I/O read strobe)		
	14	/IOWR	Output (I/O write strobe)		
	15	PE0	Bitwise or parallel programmable I/O	I0 control or INT0A input	
	16	PE1		I1 control or INT1A input	
	17	TPIN-			Ethernet Receiver Port
	18	TPIN+			
	19	PE4	Bitwise or parallel programmable I/O	I4 control or INT0B input	
	20	PE5		I5 control or INT1B input	
	21	ACT			Ethernet active (ACT) LED indicator
	22	PE7	Bitwise or parallel programmable I/O	I7 control or slave port chip select / SCS	
23-26	A[3:0]			Rabbit 2000 address bus	

ตารางที่ 2.1 ความสามารถในการนำไปใช้งานของคอนเน็คเตอร์ J4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

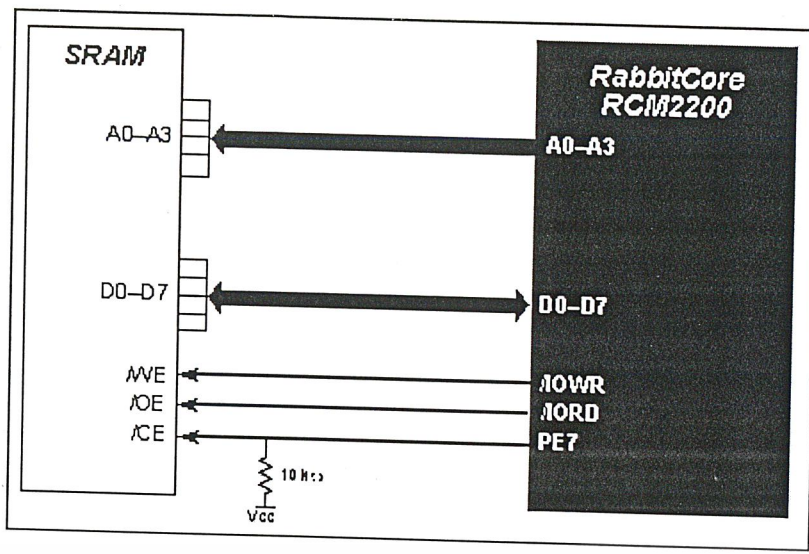
ขาสัญญาณ	ชื่อขาสัญญาณ	Default Use	Alternate Use	Notes
Header J5	1-8	PA[0:7]	Byte-wide programmable parallel I/O	Slave port data bus SD0-SD7
	9	/RESET	Reset output	Reset input This weak output can be driven externally
	10	PB0	Input	Serial port clock CLKB input or output
	11	PB2	Input	Slave port write /SWR
	12	PB3	Input	Slave port read /SRD
	13	PB4	Input	SA0 Slave port address lines
	14	PB5	Input	
	15	PB7	Output	Slave port attention line /SLAVEATTN
	16-23	D[7:0]	Input/Output	Rabbit 2000 data bus
	24	VBAT	3 V battery input	
	25	Vcc		
	26	Gnd		

ตารางที่ 2.2 ความสามารถในการนำไปใช้งานของหัวต่อสัญญาณ J5

2.2.4 อินพุต / เอาต์พุตที่ติดต่อหน่วยความจำ

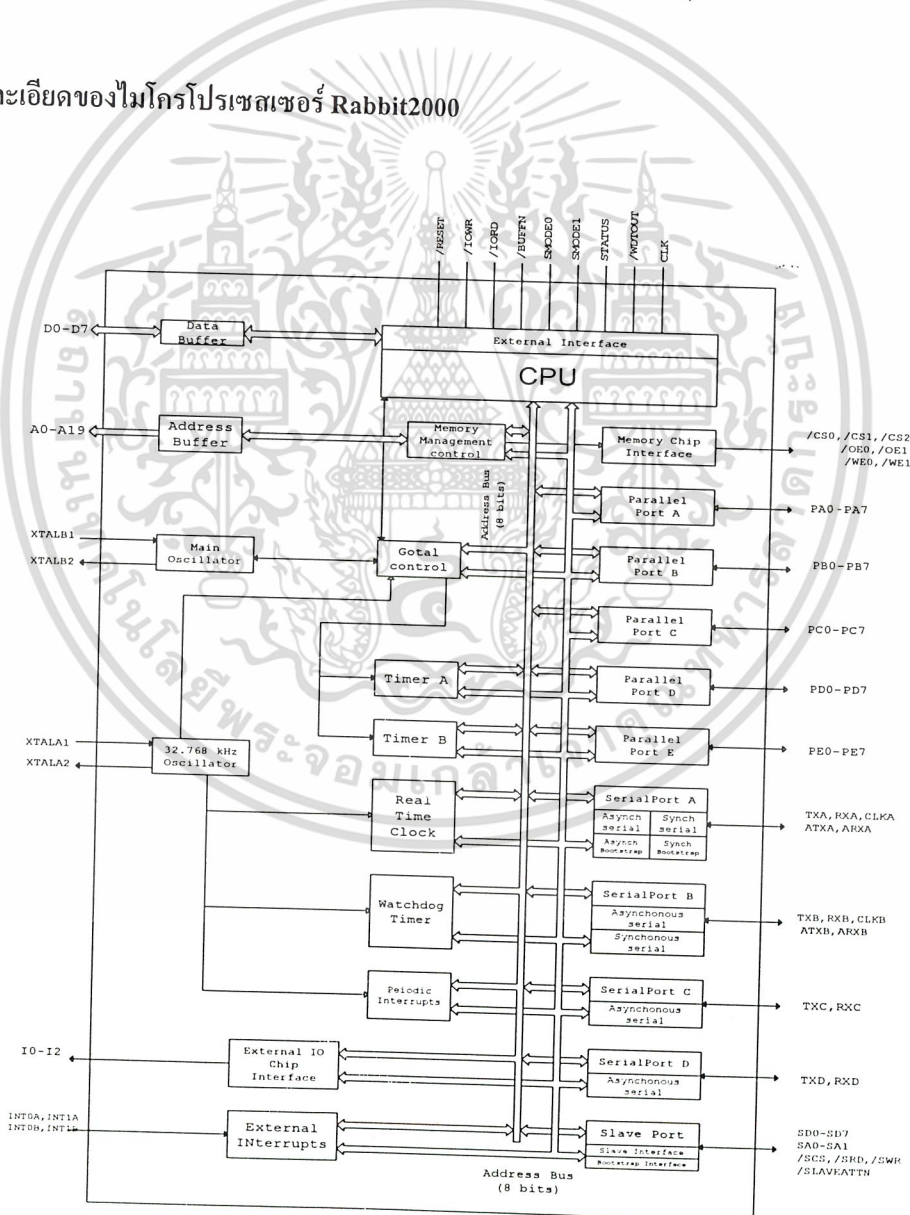
จะประกอบด้วยบัสตำแหน่ง (Address Bus) จาก A0 - A3 และบัสข้อมูล (Data Bus) จาก D0 -D7 สำหรับการติดต่อกับอุปกรณ์ภายนอก จะใช้ตำแหน่งขาสัญญาณ IOWR และ IOWD เป็นอินพุต / เอาท์พุต ในการอ่านและเขียนข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 การติดต่อกับหน่วยความจำ (RAM)

2.3 รายละเอียดของไมโครโปรเซสเซอร์ Rabbit2000



รูปที่ 2.5 โครงสร้างภายในของ Rabbit2000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไมโครโปรเซสเซอร์ Rabbit 2000 เป็นไมโครโปรเซสเซอร์ที่ผลิตโดยบริษัท Rabbit Semiconductor Corporation เป็นไมโครโปรเซสเซอร์ขนาด 8 บิต โดยมีความเร็วของสัญญาณนาฬิกาเป็น 22.1 MHz ถูกออกแบบมาเพื่อให้ใช้กับงานที่เกี่ยวข้องกับระบบ Ethernet โดยเฉพาะ และไมโครโปรเซสเซอร์ Rabbit 2000 ก็เป็นเหมือนหัวใจหลักของ Rabbit Core Module RCM2200 ซึ่งคอยควบคุมการทำงานต่างๆไม่ว่าจะเป็นอุปกรณ์ภายในและภายนอกของ Rabbit Core Module RCM2200

2.4 หน้าที่ของขาอินพุต/เอาต์พุตบนไมโครโปรเซสเซอร์ Rabbit2000

- CLK เป็นขาชนิดเอาต์พุตที่ป้อนสัญญาณนาฬิกาที่ใช้ในการประมวลผลออกมา
- RESET เป็นขาชนิดอินพุตที่ใช้ในการกำหนดค่าเริ่มต้นหรือการทำงานใหม่
- XTALB1, XTALB2 เป็นขาชนิดอินพุต/เอาต์พุต ที่ใช้สำหรับการเชื่อมต่อกับ Oscillator quartz crystal 11.0592 MHz
- XTALA1, XTALA2 เป็นขาชนิดอินพุต/เอาต์พุต ที่ใช้สำหรับการเชื่อมต่อกับ Oscillator Quartz Crystal 32.678 KHz
- A0-A19 เป็นขาชนิดเอาต์พุต ทำหน้าที่กำหนดการเลือกใช้และควบคุมหน่วยความจำภายนอก
- D0-D7 เป็นขาชนิดอินพุต/เอาต์พุต ที่ใช้เป็นทางผ่านของข้อมูลระหว่างพอร์ตต่างๆ ของไมโครโปรเซสเซอร์
- WDTOUT เป็นขาชนิดเอาต์พุต ที่ใช้ตรวจเช็คสัญญาณลูกคลื่นมีค่าต่ำกว่า 30us หรือไม่ภายในระยะเวลาที่กำหนด
- STATUS เป็นขาชนิดเอาต์พุต ที่ใช้ตรวจสอบสถานะการทำงานของ RCM2200
- SMODE0, SMODE1 เป็นขาชนิดอินพุต ที่ใช้กำหนดเริ่มการทำงาน
- CS0, OE0, WE0 เป็นขาชนิดเอาต์พุต ตามปกติใช้เป็นการ Enable เพื่อให้ทำการเขียนข้อมูลลง Flash Memory ใน Program Code
- CS1, OE1, WE1 เป็นขาชนิดเอาต์พุต ตามปกติใช้เป็นการ Enable เพื่อให้ทำการเขียนข้อมูลลง Static RAM และทำให้ความเร็วและการสูญเสียพลังงานเหมาะสมกัน
- CS2 เป็นขาชนิดเอาต์พุต โดยทั่วไปใช้ในการ Memory Decode
- BUFFEN เป็นขาชนิดเอาต์พุต ปกติตั้งไว้เป็น Enable ทำงานกับ I/O ภายนอกจะทำงานเมื่อไมโครโปรเซสเซอร์ Rabbit 2000 ขอมมา
- IORD, IOWR เป็นขาชนิดเอาต์พุต ใช้ให้ระบุนการอ่าน และเขียนข้อมูลกับส่วน I/O
- PA0-PA7 เป็นขาชนิดอินพุต/เอาต์พุต ของพอร์ต A (Slave Port) ทำหน้าที่ Reset พอร์ต
- PB0-PB7 เป็นขาชนิดอินพุต/เอาต์พุต ของพอร์ต B (Slave Port) ทำหน้าที่ควบคุม Slave Port และ Serial Port Clocks
- PC0-PC7 เป็นขาชนิดอินพุต/เอาต์พุต ของพอร์ต C (Slave Port) โดยทั่วไปทำหน้าที่เป็น Serial Data ของ Slave Port

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- PD0-PD7 เป็นขาชนิดอินพุท/เอาต์พุท ของพอร์ต D (Slave Port)
- PE0-PE7 เป็นขาชนิดอินพุท/เอาต์พุท ของพอร์ต E (Slave Port) โดยทั่วไปทำหน้าที่เป็น Interrupts จากภายนอก
- VBAT ใช้จ่ายพลังงานชนิด Real-Time Clock (25uA at 2.3V)
- VDD ใช้จ่ายพลังงานให้กับไมโครโปรเซสเซอร์ Rabbit 2000 ที่แรงดัน 2.7 ถึง 5.0 Volts
- VSS คือ Ground

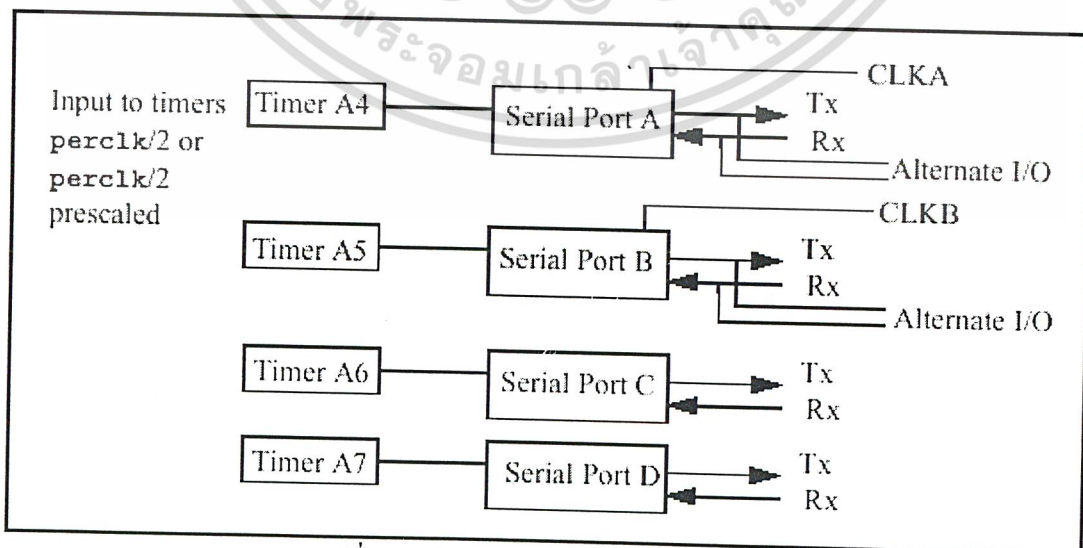
2.5 พอร์ตในการเชื่อมต่อ Rabbit Core Module RCM2200

พอร์ตในการเชื่อมต่อ RCM2200 บอร์ดไม่มีสาย RS - 232 หรือสาย RS - 485 โดยตรงบนบอร์ด การจะติดต่อด้วยสาย RS - 232 หรือ RS - 485 สามารถทำได้โดยการติดต่อผ่านบอร์ด Prototyping ซึ่งมีการสนับสนุนมาตรฐาน RS - 232 ไว้ในบอร์ดเดียวกัน

2.5.1 พอร์ตอนุกรม (Serial Ports)

พอร์ตอนุกรม (Serial Ports) มีอยู่ 4 พอร์ตคือ พอร์ต A, B, C และ D ทั้งหมด 4 พอร์ตถูกใช้สำหรับการพัฒนา Software สามารถกระทำใน โหมด Asynchronous ที่มีอัตราบอร์ด์เลตสูงๆ ในโหมด Asynchronous สามารถจัดการ 7 หรือ 8 บิตข้อมูล เมื่อมีการส่งข้อมูลข่าวสารไปทีแรกของข้อมูลจะถูกกำหนดเพื่อให้ทราบถึงการรับส่งข้อมูล

พอร์ต A และ B สามารถทำงานในโหมด Asynchronous ทำงานไปพร้อมกับ พอร์ต C และ D ได้ อย่างไรก็ตามพอร์ต A และ B ยังสามารถที่จะกำหนดการทำงานให้เป็นอย่างใดอย่างหนึ่ง ระหว่างการรับหรือส่ง อุปกรณ์ในการคิดต้องจะสัมพันธ์กับสัญญาณนาฬิกา เมื่อ Rabbit 2000 สร้างสัญญาณนาฬิกา อัตรา (Baud Rate) สูงโดยคิดจากความถี่ของสัญญาณนาฬิกาหารด้วยค่า 128 หรือ 138,240 bps สำหรับ 22.1 MHz



รูปที่ 2.6 บล็อกไดอะแกรม ของพอร์ตอนุกรม

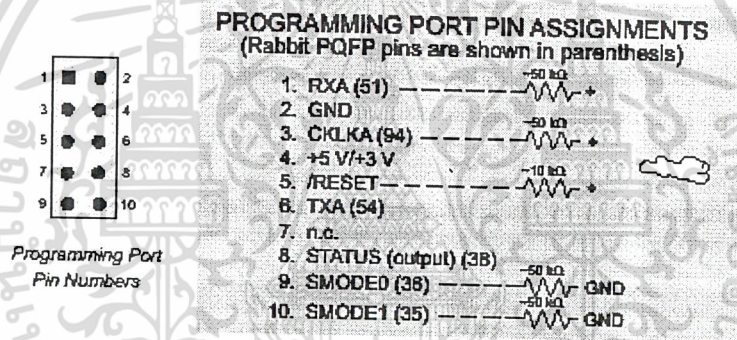
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2 อีเทอร์เน็ต พอร์ต (Ethernet Port)

อีเทอร์เน็ต พอร์ต (Ethernet Port) ที่ใช้บนบอร์ด RCM2200 จะใช้หัวต่อแบบ RJ-45 ซึ่งจะมี LED อยู่สองดวงมีหน้าที่ บอกระยะการทำงานของบอร์ด RCM2200 โดยขาสัญญาณ (LNK) มีหน้าที่ บอกระยะการเชื่อมต่อ ส่วนขาสัญญาณ (ACT) จะบอกระยะการถ่ายโอนข้อมูลโดย อีเทอร์เน็ต พอร์ต (Ethernet Port) จะอยู่ที่คอนเน็คเตอร์ J2

2.5.3 พอร์ตการเขียนโปรแกรม (Programming Port)

พอร์ตการเขียนโปรแกรม สื่อสารกับ RCM2200 โดยทาง Chip Serial Port พอร์ต A มีความสามารถพิเศษที่ใช้สำหรับการพัฒนาซอฟต์แวร์ภายใต้ โปรแกรมไคนามิก C ขาสัญญาณ SMODE0 กับ SMODE1 แสดงการเขียนโปรแกรม เชื่อมต่อกับอุปกรณ์ภายนอกตำแหน่งที่ใช้ในการเขียนโปรแกรม สำหรับ RCM2200 ในโหมดที่จะดาวน์โหลดโปรแกรมจากคอมพิวเตอร์ให้แก่ RCM2200 จนจบโปรแกรม พร้อมทั้งทำการตรวจสอบตรวจสอบจุดบกพร่องในโปรแกรม



รูปที่ 2.7 พอร์ตการเขียนโปรแกรม

2.6 Parallel Ports

RCM2200 มีพอร์ตขนานทั้งหมด 5 พอร์ต โดยแต่ละพอร์ตจะมีพอร์ตละ 8 บิตซึ่งพอร์ตขนานจะแบ่งออกเป็น A, B, C, D และ E โดยขาของสัญญาณที่ใช้สำหรับพอร์ตขนานจะทำการแชร์กับฟังก์ชันอื่นๆมากมาย คุณสมบัติสำคัญของพอร์ตถูกสรุปดังนี้

พอร์ต A - จะทำการแชร์กับส่วนที่ติดต่อข้อมูลของ Slave Port

พอร์ต B - จะทำการแชร์กับเส้นทางของหน่วยควบคุมของ Slave Port

พอร์ต C - จะทำการแชร์กับพอร์ตที่เป็นชุดข้อมูลของชุดอินพุท และเอาต์พุท

พอร์ต D - จะมีบิตอยู่ 4 บิตที่จะทำการแชร์และสลับกันของขาสัญญาณอินพุท และเอาต์พุท สำหรับพอร์ตอนุกรมของ A และ B ส่วนอีก 4 บิตจะไม่มีกรแชร์ โดยพอร์ต D มีความสามารถที่จะแก้ไขเอาต์พุท ของมันตามเอาต์พุททั้งหมด

พอร์ต E - บิตทั้งหมดของพอร์ต E สามารถที่จะแก้ไขเช่นเดียวกับ I/O Strobes โดยบิต 4 บิตของพอร์ตสามารถนำไปใช้ในการอินเทอร์รัปต์ภายในของอินพุท ส่วนอีก 1 บิตจะทำการแชร์ในการ Slave Port

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.1 Parallel Port A

Parallel Port A จะมีหน้าที่ในการอ่าน และเขียนรีจิสเตอร์ โดยจะมีข้อมูล PADR อยู่ที่ตำแหน่ง 030h (adr = 030h)

รีจิสเตอร์นี้ไม่ควรจะใช้ถ้า Slave Port มีการใช้งานรีจิสเตอร์ของหน่วยควบคุมเพื่อกำหนดให้เป็นอินพุท หรือเอาต์พุท ถ้าต้องการให้พอร์ต A เป็นอินพุทต้องให้เก็บค่า 080h ใน SPCR (Slave Port Control Register) ส่วนการทำให้พอร์ต A เป็นเอาต์พุทต้องให้เก็บค่า 084h ใน SPCR พอร์ต A จะทำการเช็ทค่าอินพุทโดยการ รีเช็ท

2.6.2 Parallel Port B

Parallel Port B จะมีอินพุทอยู่ 6 ตัว และเอาต์พุทอยู่ 2 ตัว ซึ่งจะมีขาของสัญญาณ PB2 - PB7 เป็นตัวกำหนดฟังก์ชันของ Slave Port อย่างไรก็ตามก็ยังสามารถที่จะอ่านข้อมูลได้ โดยอาศัยขาสัญญาณ PB0 - PB5 ถึงแม้ว่าขาสัญญาณ PB2 - PB7 จะมีการนำไปใช้งานแล้วก็ตาม และยังสามารถที่จะอ่านโดยอาศัยขาสัญญาณ PB6 และ PB7 ได้อีกด้วย โดยไม่คำนึงถึงว่า Slave Port จะถูกใช้หรือไม่ก็ตาม เพราะพอร์ตอนุกรม B มีสัญญาณนาฬิกาภายใน ซึ่งก็เป็นสาเหตุที่ทำให้ขาของสัญญาณนี้ถูกใช้งานโดยสัญญาณนาฬิกาของพอร์ตอนุกรม ส่วนกรณีของขาสัญญาณ PB1 จะมีการคืนค่าหากมีการนำข้อมูลเข้ามา

2.6.3 Parallel Port C

Parallel Port C จะมีอินพุทอยู่ 4 บิต และเอาต์พุทอยู่ 4 บิต โดยจะแบ่งออกเป็นบิตที่เป็นเลขคู่จะประกอบไปด้วย PC0, PC2, PC4 และ PC6 จะเป็นเอาต์พุท และบิตที่เป็นเลขคี่จะประกอบไปด้วย PC1, PC3, PC5 และ PC7 จะเป็นอินพุท เมื่อรีจิสเตอร์ของข้อมูลทำการอ่าน บิตที่ 1, 3, 5 และ 7 จะทำการคืนค่าของแรงดันไฟฟ้าบนขาสัญญาณ ส่วนบิตที่ 0, 2, 4 และ 6 จะทำการคืนค่าโดยการส่งสัญญาณบัฟเฟอร์ไปที่เอาต์พุท โดยในการส่งสัญญาณบัฟเฟอร์ ผลลัพธ์ที่ได้ และค่าของขาสัญญาณจะเหมือนกัน

Parallel Port C จะมีการแชร์กันของขาสัญญาณของพอร์ตอนุกรมทั้งหมด 4 บิต โดยขาสัญญาณของพอร์ตขนาน C สามารถเป็นอินพุท และยังเป็นขาสัญญาณของพอร์ตอนุกรมที่สามารถเป็นอินพุทได้ด้วย (พอร์ตอนุกรม A และ B สามารถใช้สลับบิต 7 และ 5 ตามลำดับในพอร์ต C ที่เป็นอินพุท และ Source ของอินพุทที่พอร์ตอนุกรมจะขึ้นอยู่กับหน่วยควบคุมพอร์ตโดยตรงของรีจิสเตอร์ควบคุมของพอร์ตอนุกรม) เส้นทางของข้อมูลสามารถอ่านได้จากพอร์ตขนาน C โดยรีจิสเตอร์ของข้อมูลเอาต์พุทของพอร์ตขนานสามารถถูกนำไปเลือกใช้พอร์ตอนุกรม โดยจะทำการเก็บในตำแหน่งที่ตรงกันของพอร์ต C ในรีจิสเตอร์ฟังก์ชัน (PCFR) เมื่อขาสัญญาณของพอร์ตขนานถูกเลือกใช้โดยผลลัพธ์ของพอร์ตอนุกรมค่าที่เก็บในรีจิสเตอร์ของข้อมูลจะไม่ถูกสนใจ

2.6.4 Parallel Port D

พอร์ตขนาน D มีขาสัญญาณทั้งหมด 8 บิต ซึ่งสามารถที่จะโปรแกรมเฉพาะไม่ว่าจะเป็น อินพุท หรือเอาต์พุท เมื่อโปรแกรมเป็นเอาต์พุทขาสัญญาณก็สามารถที่จะเลือกเอาต์พุท หรืออินพุทที่เป็นมาตรฐาน ขาสัญญาณของพอร์ต D สามารถเป็นบิตที่เป็นตำแหน่ง Address รีจิสเตอร์ของเอาต์พุทจะมีรูปแบบและมี Timers ควบคุมตามลำดับขั้นตอน การทำงานจะมีการสร้างสัญญาณพัลส์ในการจับเวลานอกจากนี้เอาต์พุทของพอร์ต D มีความสามารถในการส่งสัญญาณที่สูง พอร์ต D จะมีบิต 4 และ 5 ที่สามารถที่จะทำการสลับขาสัญญาณกันของพอร์ตอนุกรม B และบิต 6 และ 7 ยังสามารถถูกใช้ในการสลับขาสัญญาณกันของพอร์ตอนุกรม A การสลับกันของบิตพอร์ตอนุกรม จะกระทำในพอร์ตอนุกรมโดยจะมีการเชื่อมต่อกับขาสัญญาณในการติดต่อกันที่แตกต่างกันในเวลาเดียวกัน ในกรณีการรีเซ็ต ข้อมูลในรีจิสเตอร์โดยตรงจะมีค่าเท่ากับ “0” การทำงานของขาสัญญาณทั้งหมดจะเป็นอินพุท นอกจากนี้ในการส่งค่าให้กับรีจิสเตอร์หน่วยควบคุมมีค่าเท่ากับ “0” (บิต 0, 1, 4 และ 5) ก็เพื่อที่จะให้แน่ใจว่าข้อมูลได้ถูกใช้สัญญาณนาฬิกาเข้าไปในรีจิสเตอร์เมื่อมีการบันทึก รีจิสเตอร์อื่นๆ ทั้งหมดที่เกี่ยวข้องกับพอร์ต D จะไม่ถูกทำงานตอนเริ่มต้นเมื่อเราทำการรีเซ็ต

PDDR เป็นรีจิสเตอร์ข้อมูลของพอร์ตขนาน D จะทำการอ่าน และเขียน

PDDDR เป็นรีจิสเตอร์โดยตรงข้อมูลของพอร์ตขนาน D มีค่าเท่ากับ “1” ทำให้ขาสัญญาณเป็นเอาต์พุทใช้ในการเขียนเท่านั้น

PDDCR เป็นรีจิสเตอร์ของหน่วยควบคุม ของพอร์ตขนาน D มีค่าเท่ากับ “1” ทำให้ขาสัญญาณเป็นเอาต์พุท ที่ใช้ในการเขียนเท่านั้น

PDFR เป็นรีจิสเตอร์หน่วยควบคุมฟังก์ชัน ของพอร์ตขนาน D พอร์ตนี้อาจจะถูกใช้เพื่อหาตำแหน่งของพอร์ต 4 และ 6 ของเอาต์พุทอนุกรม ที่ใช้ในการเขียนเท่านั้น

PDBxR เป็นจะมีรีจิสเตอร์ทั้งหมด 8 ตัวโดยจะถูกใช้เพื่อเซตตำแหน่งของพอร์ตที่จะใช้งาน

PDCR เป็นรีจิสเตอร์หน่วยควบคุม ของพอร์ตขนาน D รีจิสเตอร์นี้จะถูกนำไปใช้ในการควบคุมสัญญาณนาฬิกาของด้าน High และ Low ของเอาต์พุทตัวสุดท้ายของ รีจิสเตอร์ของพอร์ต ส่วนการรีเซ็ต บิต 0, 1, 4, และ 5 จะมีค่าเป็น “0”

2.6.5 Parallel Port E

มีขาสัญญาณอินพุท/เอาต์พุททั้งหมด 8 บิตที่สามารถถูก โปรแกรมเฉพาะอย่างได้ ไม่ว่าจะเป็น อินพุทและเอาต์พุท พอร์ต E มีความสามารถในการส่งสัญญาณที่สูงกว่าพอร์ตอื่นๆ โดย PE7 จะถูกใช้เมื่อ Slave Port ถูกใช้งานเป็นเอาต์พุทของพอร์ต E สามารถที่จะแก้ไข I/O Strobe นอกจากนี้ยังมีขาสัญญาณจำนวน 4 บิตของพอร์ต E สามารถเกิดการอินเตอร์รัพท์เมื่อมีการร้องขอการอินเตอร์รัพท์ที่อินพุท รีจิสเตอร์ของเอาต์พุทมีรูปแบบและ Timers อย่างมีแบบแผนการทำงานที่เป็นไปตามการสร้างสัญญาณเวลา

PEDR เป็นรีจิสเตอร์ข้อมูลของพอร์ต E มีหน้าที่อ่านค่าจากขาสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PEDDR เป็นรีจิสเตอร์โดยตรงข้อมูลของพอร์ต E หากเซ็ทเป็น “1” จะทำให้ขาสัญญาณเป็นเอาต์พุต รีจิสเตอร์นี้จะรีเซ็ทก็ต่อเมื่อมีค่าเท่ากับ “0”

PEFR เป็นรีจิสเตอร์ฟังก์ชันของพอร์ต E หากเซ็ทเป็น “1” จะทำให้เอาต์พุตตรงกับ I/O strobe โดยปรกติ I/O strobe จะถูกควบคุมโดย รีจิสเตอร์ I/O ของหน่วยควบคุม (IBxCR) ต้องมีการเซ็ทเป็นเอาต์พุตสำหรับ I/O strobe ถึง จะทำงานได้

PEBxR เป็นรีจิสเตอร์นี้จะเป็นส่วนของรีจิสเตอร์ที่ใช้ในการเซ็ทให้เป็นเอาต์พุตของแต่ละบิต

PECR เป็นรีจิสเตอร์หน่วยควบคุม ของพอร์ตขนาน E เป็นรีจิสเตอร์ที่ถูกใช้เพื่อควบคุมสัญญาณนาฬิกาของด้าน High และ Low ของเอาต์พุตตัวสุดท้ายของรีจิสเตอร์ของพอร์ต ในการรีเซ็ท บิต 0, 1, 4 และ 5 จะมีค่าเท่ากับ “0”

2.7 โปรแกรม Dynamic C

Dynamic C คือ ระบบการพัฒนารวมของการเขียนซอฟต์แวร์ สร้างมาจากระบบคอมพิวเตอร์ของ IBM ซึ่งออกแบบให้เข้ากับคอมพิวเตอร์ทั่วไป ให้ใช้งานได้

Dynamic C มีการใช้ในเครื่องขายตั้งแต่ปี ค.ศ. 1989 ถูกออกแบบสำหรับการเขียนโปรแกรมฝังระบบ และมีความสามารถตรวจสอบในตัวเอง สำหรับไมโครโปรเซสเซอร์ Rabbit 2000 ที่ใช้งานกันทั่วไป สามารถติดต่อกันโดย สายแพ 10 สายที่ พอร์ต B โปรแกรมพื้นฐานของระบบ มีข้อมูลประมาณ 1,000 ไบท์ ที่ใช้ในการจัดเตรียม Debugging และการติดต่อข้อมูลต่างๆ Dynamic C ต้องการ BIOS เพื่อใช้ในการตรวจสอบโปรแกรม เพื่อที่จะใช้งานได้สะดวก ถ้าผู้ใช้หยุดการ Run โปรแกรมและใช้โปรแกรมใหม่ BIOS ก็จะเซ็ทการทำงานใหม่ตลอด Dynamic C ออกแบบให้เข้ากับภาษา Assembly และโปรแกรมภาษาซี

2.7.1 Using Dynamic C

ผู้ใช้โปรแกรม มีตัวเลือกในการที่จะพัฒนาซอฟต์แวร์ ภาษาเขียนทำการบันทึกใน Flash Memory ขนาด 256 Kbyte หรือ ใน Static Ram ขนาด 128 Kbyte ผลการทำงานในหน่วยความจำ คือการบันทึกข้อมูลสามารถบันทึกได้ถึง 100,000 ครั้งของการเขียน

ข้อเสียของการใช้ Flash Memory คือเมื่อมีการดีบั๊กโปรแกรมเพื่อขจัดจ้งหะการทำงาน จะทำให้ Interrupt เกิดข้อผิดพลาด การทำงานของโปรแกรมก็จะหยุดตามไปด้วย

ไดนามิก C เป็นภาษาที่ใช้ในการสนับสนุน TCP/IP โดยไดนามิก C จะประกอบไปด้วย Libraries ต่างๆ โดยจะมี Libraries หลัก คือ DCRTCP.LIB อีกทั้งไดนามิก C ยังมี Libraries ส่วนของ DNS (Domain Name Server), IP, TCP และ UDP (User Datagram Protocol)

UDP จะมี DNS.LIB, IP.LIB, NET.LIB, TCP.LIB และ UDP.LIB ส่วนในการติดต่อหรือใน ส่วนของชั้นเครือข่ายของโปรโตคอล TCP/IP จะมี Libraries ที่ชื่อ ARP.LIB และ ICMP.LIB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของ Libraries หลัก DCRTCP.LIB จะประกอบไปด้วย มาโครที่ทำการตั้งค่าไว้ส่วน โครงสร้างข้อมูลและฟังก์ชันที่ใช้กับ IP เวอร์ชัน 4 ได้มีการสนับสนุนโดย DCRTCP.LIB ในการคอมไพล์ TCP/IP จะต้องให้ส่วนของบอร์คควบคุมรู้ค่า IP Address, Netmask และ Default Gateway

2.7.2 การเซตค่า IP Addresses

ค่า IP Address มีความจำเป็นที่จะต้องทำการเซตค่าในช่วงของการคอมไพล์ โดยจะกำหนดการตั้งค่าไว้ที่ MY_IP_ADDRESS, MY_NETMASK, MY_GATEWAY และ MY_NAMESERVER ตามลำดับ ในช่วงของการคอมไพล์ในส่วนของฟังก์ชัน tcp_config sethostid, sethostname โดยสามารถที่จะควบคุม โดย Macros

2.7.3 IP Addresses Set Dynamically

BOOTP.LIB เป็นไลบรารีที่จะยอมให้บอร์คในส่วนของ BOOTP หรือ DHCP ของ Client ให้เป็นส่วนที่จะนำไปใช้ โดยโปรโตคอลจะยึดหลักของขนาดของเครื่องเซิร์ฟเวอร์ที่ทำการติดตั้งบนเครือข่าย ภายใน เครื่องเซิร์ฟเวอร์ BOOTP และ DHCP จะติดตั้งในส่วนกลางของระบบเครือข่ายภายในและจะคอยจัดการในการวางแผนงานของระบบเครือข่าย

โปรโตคอลทั้ง 2 นี้ จะมีค่าพารามิเตอร์ที่ใช้ในการส่งไปยัง Client รวมถึง

- IP Address ของ Client
- Netmask
- รายชื่อ Gateway
- Host และ รายชื่อโดเมนพื้นฐาน
- รายชื่อของเครื่องเซิร์ฟเวอร์

ในการนำไปใช้งาน ต้องโปรแกรมดังนี้

```
#define USE_DHCP
#use DCRTCP.LIB
```

2.7.4 BOOTP/DHCP Control Macros

มาโครต่างๆ สามารถที่จะทำการควบคุมการ DHCP ของ Client ได้ หากมีการเซตค่าก่อนบรรทัด #use "dcrtcp.lib" ในส่วนโปรแกรมแอปพลิเคชัน USE_DHCP ถ้ามาโครนี้ถูกกำหนดจุดมุ่งหมายในการใช้ BOOTP หรือ DHCP เพื่อแก้ไขตัวแปรที่ต้องการ ถ้า USE_DHCP ไม่ถูกกำหนดจะทำให้ MY_IP_ADDRESS, MY_NETMASK, MY_GATEWAY และ MY_NAMESERVER อาจจะถูกกำหนด โดยในส่วนของโปรแกรมแอปพลิเคชัน

2.7.5 Sizes for TCP/IP

ในการเริ่มการทำงานของไดนามิก C เวอร์ชัน 8.01 บัฟเฟอร์ TCP และ UDP ได้มีการแบ่งแยกออกมาเป็น

TCP_BUF_SIZE ได้มีการกำหนดขนาดบัฟเฟอร์ของ TCP ไว้ที่ 4096 ไบท์

UDP_BUF_SIZE ได้มีการกำหนดขนาดบัฟเฟอร์ของ UDP ไว้ที่ 4096 ไบท์

ถ้า SOCK_BUF_SIZE มีการกำหนดจะทำให้ค่า TCP_BUF_SIZE และ UDP_BUF_SIZE จะตรงกับ SOCK_BUF_SIZE แต่ถ้า SOCK_BUF_SIZE ไม่มีการกำหนด จะทำให้ค่า TCP_BUF_SIZE และ UDP_BUF_SIZE จะเท่ากับ tcp_MaxBufSize คูณกับ 2

2.7.6 Number of Sockets

การเริ่มต้นการทำงานของ Dynamic C เวอร์ชัน 8.01 จะมีการกำหนดมาโคร 2 ตัวให้กับหมายเลขซ็อกเกตได้ดังนี้

MAX_TCP_SOCKET_BUFFERS จะมีการกำหนดหมายเลขสูงสุดให้กับซ็อกเกตของ TCP โดยจะมีการเรียก tcp_open(); หรือ tcp_listen();

MAX_UDP_SOCKET_BUFFERS จะมีการกำหนดหมายเลขสูงสุดให้กับซ็อกเกตของ UDP udp_open();

2.7.7 Passive Open

จะมีอยู่ 2 เส้นทางในการเปิดซ็อกเกตของ TCP คือ Passive และ Active ซึ่งการเปิดซ็อกเกตนี้จะต้องทำการเรียก tcp_listen(); โดยจะทำการรอการติดต่อกับอุปกรณ์และชนิดของการเปิดจะใช้กับเซิร์ฟเวอร์ของอินเตอร์เน็ตหรืออาจจะให้ tcp_listen(); เป็นตัวชี้ตำแหน่งของข้อมูล ถ้าต้องการที่จะติดต่อโดยได้ยอมรับข้อตกลงของหมายเลขพอร์ต และ IP Address โดยทำการเซทค่าให้เป็นศูนย์หรือหนึ่งทั้งคู่

2.7.8 Active Open

เมื่อมีการเรียกหน้าเว็บเบราว์เซอร์ก็จะทำการเชื่อมต่อกับเซิร์ฟเวอร์เพื่อทำการเปิด โดยจะทำการเรียกใช้ tcp_open(); และยังใช้พารามิเตอร์คล้ายๆกับการใช้ tcp_listen(); โดยหลักพารามิเตอร์นี้จะเกี่ยวกับ IP Address และหมายเลขพอร์ต สามารถทำการติดต่อ โดยทำการเซทค่าพารามิเตอร์ lport ให้เป็นศูนย์ ส่วน DCRTCP.LIB เป็นการแสดงถึงการเลือกพอร์ตระหว่าง 1024 และ 65535 หาก tcp_open(); กลับค่ามาเป็นศูนย์อีกครั้งจะไม่สามารถทำการติดต่อได้

2.7.9 Delay a Connection

การยอมรับการร้องขอการติดต่อเมื่อกระบวนการวิธีการร้องขอไม่เหมาะสมก็จะทำการเรียกฟังก์ชัน tcp_reserveport(); เมื่อมีการตอบรับการติดต่อพารามิเตอร์ในส่วนหัวของ TCP จะทำการเซทค่าเป็นศูนย์ในช่วงเวลานี้จะทำการรอค่าจากพารามิเตอร์ tcp_clearreserve (Port Number) เพื่อทำการดูแลในเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การติดต่อก่อนจากนี้ยังมีมาโคร USE_RESERVEDPORTS จะทำการกำหนดเพื่อให้เกิดการทำงานขึ้นของสองฟังก์ชันคือเมื่อมีการใช้ tcp_reserveport, 2MSL (Maximum Segment Lifetime)

2.7.10 Skeleton Program

โปรแกรมข้างล่างนี้เป็นโครงสร้างพื้นฐานของ Dynamic C ที่ใช้ติดต่อกับ TCP/IP โดยส่วนแรกจะทำการกำหนดการเชื่อมต่อ IP ของข้อมูลในบรรทัด "memmap" จะเป็นการให้โปรแกรมทำการคอมไพล์และตรวจสอบค่าส่วนในบรรทัด "use" จะเป็นตัวสั่งให้ตัวคอมไพล์ทำการคอมไพล์ของข้อมูลที่มีการตั้งค่าไว้ในส่วน TCP/IP ของ Dynamic C

```
#define MY_IP_ADDRESS "10.10.6.101"
#define MY_NETMASK "255.255.255.0"
#define MY_GATEWAY "10.10.6.19"
#memmap xmem
#use dcrtcp.lib
main() {
    sock_init();
    for (;;) {
        tcp_tick(NULL);
    }
}
```

2.7.11 TCP/IP Stack Initialization

ใน TCP/IP จะมีฟังก์ชัน Main () เป็นส่วนเริ่มแรกและจะมีการเรียก sock_init() เพื่อใช้กับโครงสร้างข้อมูลภายในและชิป Ethernet หรือชิป RealTek ส่วน DCRTCP.LIB จะคอยจัดการกับแพ็คเก็ตที่เข้ามา

2.7.12 Packet Processing

เมื่อไรก็ตามเมื่อแพ็คเก็ตมีการนำเข้ามาจะมีการเรียก tcp_tick(), tcp_open, udp_open, sock_read, sock_write, sock_close และ sock_abort ซึ่งเป็นวิธีที่ดีที่ tcp_tick() จะรู้ระยะเวลาในการประมวลผลโปรแกรมที่ส่งมาในรูปแบบของแพ็คเก็ต

2.7.13 Function Reference

ในส่วนนี้จะกล่าวถึงฟังก์ชันที่ใช้ใน DCRTCP.LIB ในเริ่มต้นนั้น DCRTCP.LIB ของ Dynamic C 8.01 จะมีฟังก์ชันที่มีการใช้ภายนอก DNS.LIB, IP.LIB, NET.LIB, TCP.LIB และ UDP.LIB และยังมีฟังก์ชันอีกส่วนหนึ่งที่สามารถเลือกใช้ได้ เช่น ARP.LIB, ICMP.LIB, BSDNAME.LIB และ XMEM.LIB

2.7.14 Macros

- MAX_SOCKETS มาโครนี้จะเป็นตัวกำหนดหมายเลขซ็อกเกตเพื่อทำการจัดสรรพื้นที่
- MY_GATEWAY มาโครนี้จะทำการส่งค่านี้เพื่อไปควบคุม Gateway ในช่วงเวลา Runtime
- MY_NETMASK มาโครนี้เป็นพื้นฐานของ NETMASK ที่ใช้ในการควบคุม
- MY_NAMESERVER มาโครนี้เป็นการแสดงชื่อของเซิร์ฟเวอร์ในช่วง Run โปรแกรม
- MY_DOMAIN มาโครนี้จะทำการควบคุมตำแหน่งเริ่มต้นของโดเมนในช่วงเวลา Runtime
- MY_IP_ADDRESS มาโครนี้จะทำการระบุตำแหน่ง IP Address เพื่อใช้ในการควบคุมในช่วงเวลา Runtime

2.8 HTTP Server

HTTP (Hypertext Transfer Protocol) เซิร์ฟเวอร์จะทำการสร้าง HTML (Hypertext Markup Language) เป็นส่วนที่ทำให้ Client ได้มีการเรียกใช้งานของเอกสารได้ง่ายขึ้น

ServerSpec	Type	Field
SSPEC_FILE		ข้อมูลของไฟล์ที่ใช้
SSPEC_VARIABLE		ข้อมูลชนิด variable ที่ใช้ใน HTTP
SSPEC_FUNCTION		ข้อมูลของฟังก์ชันที่เรียกใช้งาน

โครงสร้างข้อมูลเครื่องเซิร์ฟเวอร์ HTTP

โครงสร้างข้อมูลใน HTTP.LIB ที่น่าสนใจมีอยู่ 4 ส่วนเพื่อผู้พัฒนาเครื่องเซิร์ฟเวอร์ HTTP นำไปพัฒนาต่อคือ

- HttpSpec

โครงสร้างข้อมูล HttpSpec จะบรรจุเพิ่มของทั้งหมด, ตัวแปร, และฟังก์ชันการทำงานของเครื่องเว็บเซิร์ฟเวอร์ที่ใช้ในการเข้าถึงข้อมูลโครงสร้าง ServerSpec มาจาก ZSERVER.LIB

- HttpSpec fields

type เป็นส่วนพื้นที่นี้จะเป็นตัวบอกเครื่องเซิร์ฟเวอร์ว่ามีไฟล์ข้อมูล, ตัวแปรหรือฟังก์ชันเข้ามา (HTTPSPEC_FILE, HTTPSPEC_VARIABLE หรือ HTTPSPEC_FUNCTION ตามลำดับ)

name เป็นพื้นที่นี้จะทำการระบุชื่อเฉพาะที่ใช้ในการอ้างอิงส่วนที่มีการนำข้อมูลเข้ามา

data เป็นส่วนของตำแหน่งทางกายภาพของข้อมูล

addr ส่วนนี้จะทำหน้าที่เป็นตัวชี้ข้อมูลไฟล์ของข้อมูลทั้งหมดต้องตำแหน่งทางกายภาพ ซึ่งตัวแปรและฟังก์ชันจะต้องใช้ตัวชี้เป็นตัวชี้ตำแหน่งของข้อมูล

vartype ส่วนนี้จะเป็นตัวบอกชนิดของตัวแปร เช่น INT8, INT16, PTR16, INT32, FLOAT

format ส่วนนี้จะเป็นการอ้างอิงถึงรูปแบบ printf เพื่อประกาศตัวแปร

realm เป็นส่วนของชื่อและรหัสผ่านที่ใช้ในการเข้าถึงข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- HttpType

โครงสร้าง HttpType จะเกี่ยวข้องการขยายไฟล์ข้อมูลร่วมกับ MIME (Multipurpose Internet Mail Extension) และฟังก์ชันที่เกี่ยวข้อง MIME ผู้ใช้ยังสามารถควบคุม HTTP_MAXNAME ได้

- HttpRealm

โครงสร้าง HttpRealm จะมีการใช้งานที่ควบคู่กันคือ ID ของผู้ใช้และรหัสผ่านสำหรับพื้นที่ที่การเรียก Realm โดย Realm จะมีการป้องกันข้อมูลในเครื่องเซิร์ฟเวอร์อีกทั้งยังมีการแบ่งพื้นที่เพื่อทำการเซ็ทส่วนของพื้นที่ที่ใช้ในการป้องกันข้อมูล

- ใน โครงสร้าง HttpSpec จะมีตัวชี้โดยตัวชี้จะมีโครงสร้างชนิด HttpRealm เพื่อเป็นรหัสผ่าน ถ้าหากไม่ต้องการรหัสผ่านหรือการป้องกันก็ไม่ต้องใส่

- HttpState

การใช้สำหรับฟังก์ชัน CGI

2.9 Configuration Macros

มาโครใน HTTP.LIB มีดังต่อไปนี้

HTTP_MAXNAME

เป็นขนาดความยาวที่สุดของชื่อใน โครงสร้างของ HttpSpec ซึ่งจะมีตัวอักษรได้ 20 ตัวอักษร โดยความยาวสูงสุดของชื่อใดๆ นั้นจะมีได้ 19 ตัวอักษรเพราะว่าหนึ่งตัวอักษรได้มีการถูกใช้สำหรับเป็นส่วนท้าย

HTTP_MAXRAMSPEC

เป็นหมายเลขสูงสุดของส่วน HttpSpec ซึ่งสามารถทำการเพิ่มการ Runtime มาโครนี้ควบคุมโดย SSPEC_MAXSPEC

HTTP_MAXSERVERS

เป็นจำนวนสูงสุดของ HTTP server บนพอร์ตหมายเลข 80 แต่ไม่เกิน 4 Server

HTTP_PORT

มาโครนี้ยอมให้ผู้ใช้สามารถควบคุมพอร์ตโดยการกำหนดจะต้องกำหนดก่อนบรรทัด #use

http.lib

Functions

- cgi_redirectto

รูปแบบ

```
void cgi_redirectto(HttpState* state, char* url);
```

คำอธิบาย

เป็นส่วนที่ทำการเรียกฟังก์ชัน CGI

พารามิเตอร์

State จะทำเป็นตัวเครื่องเซิร์ฟเวอร์หรือเป็นตัวรับฟังค์ชั้น CGI
 url จะเป็นตัวแสดงคุณสมบัติของ url

Library

HTTP.LIB

- http_handler

รูปแบบ

```
int http_handler();
```

คำอธิบาย

เป็นเป็นฟังก์ชันพื้นฐานที่ใช้จ่ายความของฟังก์ชันต่างๆ

Library

HTTP.LIB

- http_init

รูปแบบ

```
int http_init(void);
```

คำอธิบาย

เริ่มต้นการทำงานของ HTTP

Library

HTTP.LIB

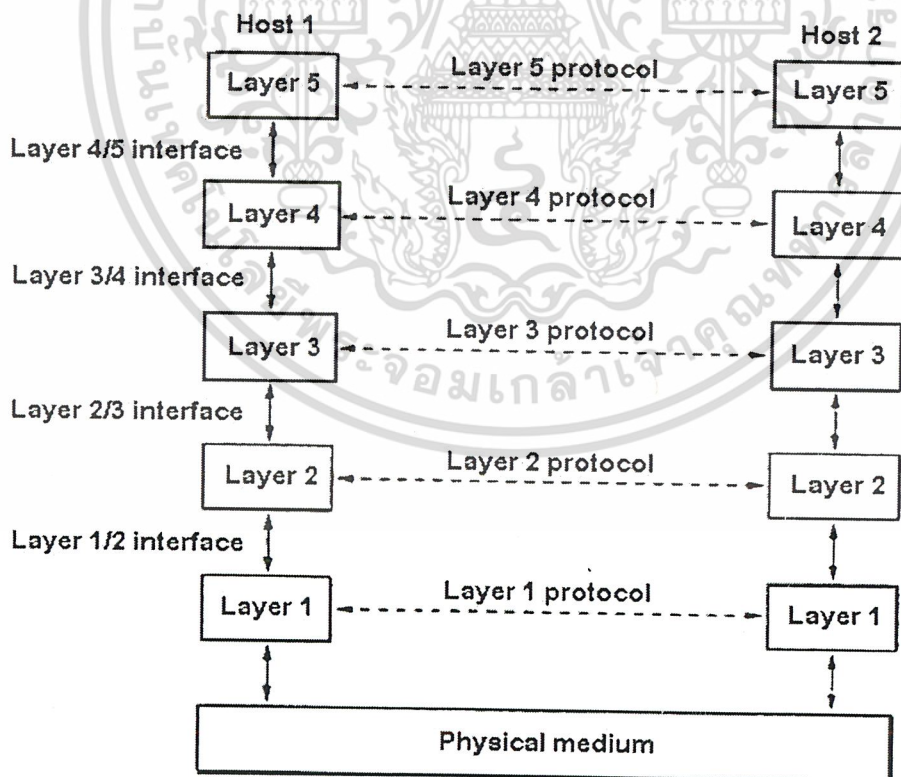
2.10 ทฤษฎีเกี่ยวกับเครือข่ายเน็ตเวิร์ก (Network)

TCP/IP เป็นข้อกำหนดเกี่ยวกับรูปแบบการเชื่อมโยงในเครือข่าย (Network Protocol) จัดทำขึ้นเพื่อใช้เป็นเครื่องคอมพิวเตอร์ใช้งานร่วมกัน ในลักษณะของระบบเปิด (Open System) คือไม่ว่าจะเป็นคอมพิวเตอร์ชนิดใดหรือระบบใดก็ตามจะสามารถติดต่อสื่อสารและแลกเปลี่ยนข้อมูลกันได้

2.10.1 ลำดับชั้นของกฎการสื่อสารข้อมูล

ระบบโปรแกรมเครือข่ายส่วนมากจะแบ่งแยกการทำงานออกเป็นหลายระดับ (Levels) หรือหลายชั้น (Layers) แต่ละชั้นจะสร้างฟังก์ชันการทำงานขึ้นโดยอาศัยการทำงานของเครือข่ายต่างๆ แตกต่างกันไปอย่างไรก็ตาม ทุกระบบจะมีแนวคิดอย่างเดียวกันคือการใช้บริการจากชั้นล่างและการให้บริการแก่ชั้นบนโดยซ่อนรายละเอียดและความซับซ้อนของฟังก์ชันในแต่ละชั้นไว้ภายใน

การสื่อสารที่เกิดขึ้นระหว่างผู้ส่งข้อมูลกับผู้รับข้อมูลนั้นจะเป็นการติดต่อของโปรแกรมแบบชั้นต่อชั้น หมายความว่าโปรแกรมในชั้นที่หนึ่งของโปรแกรมผู้ส่งจะติดต่อกับโปรแกรมในชั้นที่หนึ่งของโปรแกรมผู้รับ โปรแกรมชั้นสองจะติดต่อกับโปรแกรมชั้นสอง ฯลฯ กฎระเบียบในการติดต่อนี้เรียกว่ากฎการสื่อสารข้อมูล (Protocol) ซึ่งในแต่ละชั้นจะใช้กฎแตกต่างกัน กฎการสื่อสารข้อมูลนี้ช่วยให้ผู้ส่งข้อมูลและผู้รับข้อมูลสามารถติดต่อกันได้ หากผู้ใดผู้หนึ่งไม่ปฏิบัติตามกฎ หรือไม่รู้จักกฎเหล่านี้ การสื่อสารข้อมูลก็ไม่อาจกระทำได้



รูปที่ 2.8 ชั้นสื่อสาร, กฎการสื่อสารข้อมูล และการเชื่อมต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 2.8 แสดงให้เห็นโปรแกรมสื่อสารแบบห้าชั้น ในแต่ละชั้นที่แสดงในรูปจะแทนด้วยรูปสี่เหลี่ยมเล็กนั้นๆ ก็คือโปรแกรมที่ทำหน้าที่ สำหรับแต่ละชั้นที่กล่าวถึง โดยแต่ละชั้นจะมีกฎการสื่อสารข้อมูลเป็นของตนเอง ดังนั้นการสื่อสารที่เกิดขึ้นจึงหมายถึงการติดต่อระหว่างโปรแกรมเหล่านั้นนั่นเอง

ในความเป็นจริงแล้วการสื่อสารจะเกิดขึ้นจริงโดยผ่านสายสื่อสารที่อยู่ใต้ชั้นล่างสุดเท่านั้น ข้อมูลที่สื่อสารในระหว่างชั้นต่างๆ จะถูกส่งต่อกันเป็นลำดับดังนี้ ชั้นบนสุด (ชั้นห้า) ของผู้ส่งจะส่งข้อมูลลงมายังชั้นสี่ ซึ่งโปรแกรมชั้นที่สี่ก็จะส่งข้อมูลนั้นรวมทั้งข้อมูลของตนเองลงมายังชั้นสาม และต่อมารี้อย่าง จนถึงชั้นล่างสุด ซึ่งข้อมูลทั้งหมดก็จะส่งผ่านสายสื่อสารจากผู้ส่งไปยังผู้ใช้ ทางฝ่ายผู้ใช้ก็จะรับข้อมูลในระดับชั้นล่างสุดก่อน ข้อมูลที่เป็นของชั้นนี้จะถูกแยกออกไปข้อมูลที่เหลือจะถูกส่งขึ้นไปชั้นสอง ในทำนองเดียวกัน ข้อมูลที่เป็นของชั้นสองจะถูกแยกออกแล้วส่งส่วนที่เหลือต่อไปชั้นสาม ลักษณะเช่นนี้จะดำเนินต่อไปจนในที่สุดข้อมูลที่เป็นของชั้นห้าเท่านั้นที่จะถูกส่งขึ้นไปให้โปรแกรมชั้นที่ห้าเพื่อประมวลผลต่อไปในระหว่างชั้นที่อยู่ติดกันของโปรแกรมสื่อสารจะมีส่วนที่ติดต่อกัน (Interface) ซึ่งเราจะต้องมีการกำหนดมาตรฐานให้ชัดเจน มาตรฐานนี้คือการกำหนดรูปแบบการติดต่อในทางโปรแกรมคอมพิวเตอร์ อันได้แก่

1. กำหนดจำนวนฟังก์ชันการทำงานพื้นฐานที่จำเป็นทั้งหมดที่ใช้ในการรับ-ส่งข้อมูลระหว่างชั้น

2. กำหนดหน้าที่ของแต่ละฟังก์ชันให้ชัดเจน

3. กำหนดจำนวนและประเภทตัวแปรต่างๆ ของแต่ละฟังก์ชัน

ทั้งนี้ในแต่ละชั้นยังมีฟังก์ชันการทำงานอื่นๆ อีกจำนวนหนึ่งซึ่งไม่เกี่ยวข้องกับส่วนติดต่อและการทำงานของชั้นอื่น โปรแกรมเหล่านี้เรียกว่ามีความเป็นอิสระ (Independence) ต่อโปรแกรมส่วนอื่นๆ ดังนั้นการแก้ไขหรือเปลี่ยนแปลงโปรแกรมในส่วนนี้จึงสามารถทำได้โดยไม่มีข้อจำกัด ตัวอย่างที่ชัดเจนได้แก่การเปลี่ยนอุปกรณ์สื่อสารข้อมูลในชั้นล่างสุดเป็นอุปกรณ์ชนิดใหม่จะบังคับให้แก้ไขโปรแกรมชั้นล่างสุดเพียงชั้นเดียวเท่านั้น โปรแกรมในชั้นอื่นจะไม่มีผลกระทบแต่อย่างใด หรือในกรณีที่ต้องการปรับปรุงการทำงานของโปรแกรมในชั้นใดชั้นหนึ่ง สิ่งที่ผู้เขียนโปรแกรมจะต้องระลึกถึงคือ การรักษามาตรฐานในส่วนของการติดต่อไว้เท่านั้น เมื่อปรับปรุงเสร็จแล้วก็สามารถนำโปรแกรมใหม่มาใช้แทนโปรแกรมเก่าได้ทันที

2.10.2 รูปแบบระบบเครือข่าย TCP/IP

ระบบเครือข่ายระดับโลกที่มีใช้อยู่ในปัจจุบันคือระบบ Internet นั้นมีกำเนิดมาจากระบบเครือข่ายชื่อ ARPANET ซึ่งได้รับการสนับสนุนให้ดำเนินการวิจัยโดยมีกระทรวงกลาโหมประเทศสหรัฐอเมริกาเป็นผู้ออกค่าใช้จ่าย ในยุคแรกๆ นั้นเป็นระบบเครือข่ายที่เชื่อมการติดต่อระหว่างเครื่องคอมพิวเตอร์ในมหาวิทยาลัยและสถานที่ราชการหลายร้อยแห่งในสหรัฐอเมริกาเข้าด้วยกัน โดยใช้สายโทรศัพท์เช่า (Leased Lines) เป็นสายสื่อสารหลัก ต่อมาเมื่อระบบการสื่อสารแบบคลื่นวิทยุความถี่สูงและการสื่อสารดาวเทียมเริ่มเข้ามามีบทบาทและนำมาใช้ในระบบมากขึ้นทำให้กฎการสื่อสาร (Protocol) ที่เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เคยใช้ได้ผลดีเกิดปัญหาไม่สามารถใช้งานได้อีกต่อไป กฎการสื่อสารรุ่นต่อมาจึงได้รับการออกแบบเพื่อนำมาใช้ทดแทนแบบเก่าโดยมีวัตถุประสงค์ในการเชื่อมการติดต่อระหว่างระบบที่มีความแตกต่างกันเป็นเรื่องหลัก ผลที่ได้รับคือกฎการสื่อสารที่เรียกว่า กฎสื่อสารมาตรฐานแบบ TCP/IP ซึ่งได้รับการปรับปรุงจนนำมาใช้งานจริงได้ในปี ค.ศ.1974 การปรับปรุงรุ่นต่อมา สำเร็จในปี ค.ศ.1988

กฎสื่อสารมาตรฐานแบบ TCP/IP ยังมีวัตถุประสงค์หลักอีกสองข้อสำคัญคือ ความสามารถในการแก้ไขปัญหาที่เกิดขึ้นในระบบเครือข่าย เช่นในกรณีที่ผู้ส่งและผู้รับยังคงมีการติดต่อกันอยู่ แต่โหนดตัวกลางที่ใช้เป็นผู้ช่วยรับ-ส่งข้อมูลเกิดเสียหายใช้การไม่ได้ หรือสายสื่อสารบางช่วงถูกตัดขาด กฎการสื่อสารนี้จะต้องสามารถจัดการหาทางเลือกอื่นเพื่อให้การสื่อสารดำเนินต่อไปได้โดยอัตโนมัติ ข้อที่สองคือจะต้องมีความอ่อนตัวต่อการสื่อสารข้อมูลได้หลายชนิดทั้งแบบที่ไม่มีความเร่งด่วน เช่น การจัดส่งแฟ้มข้อมูล และแบบที่ต้องการรับประกันความเร่งด่วนของข้อมูล เช่น การสื่อสารแบบ Real - Time หรือการสื่อสารแบบโทรศัพท์ (Voice)

2.10.3 ชั้นการสื่อสารระบบ TCP/IP

Application layer
Transport layer
Network layer
Data Link layer
Physical layer

รูปที่ 2.9 โครงสร้าง TCP/IP

ชั้นที่1. ฟิสิกส์กัลเลเยอร์ (Physical Layer)

ภายใน Physical Layer จะเป็นการเกี่ยวพันกับการส่งข้อมูล เป็นบิตผ่านช่องสื่อสารข้อมูล วิธีการออกแบบจะต้องทำให้แน่ใจได้ว่าข้อมูล “1” ที่ส่งออกไปและที่ปลายทางสามารถรับ “1” ได้ถูกต้อง คำถามที่เกี่ยวข้องก็คือจะต้องใช้แรงดันเท่าใดสำหรับแทนเลข “1” และเท่าใดสำหรับแทนเลข “0” และแต่ละบิตจะต้องห่างกันเท่าใด การส่งข้อมูลสามารถทำแบบสองทิศทางได้หรือไม่ ก่อนจะเริ่มต้นส่งข้อมูลจะต้องทำให้เกิดการเชื่อมต่ออย่างไร และเมื่อส่งเสร็จแล้วจะยกเลิกการเชื่อมต่ออย่างไร ลึกลงไปอีกก็คือคอนเน็คเตอร์ (Connector หรือเป็นอุปกรณ์ที่ทำหน้าที่เป็นขั้วต่อของสายเข้ากับเครื่องคอมพิวเตอร์หรืออุปกรณ์สื่อสาร) ของ Network จะใช้กี่ขา และแต่ละขาทำหน้าที่อะไร จะเห็นว่าประเด็นที่ต้องสนใจจะเกี่ยวกับทั้งทางไฟฟ้า ทางรูปร่างภายนอกและทั้งวิธีการเชื่อมต่อ รวมทั้งตัวกลางที่อยู่ได้ Physical Layer ซึ่งจะทำหน้าที่ส่งข้อมูลจริงๆ

ชั้นที่ 2. คาต้าลิงก์เลเยอร์ (Data Link Layer).

จุดประสงค์หลักของ Data Link Layer ก็คือ พยายามทำให้การส่งข้อมูลเหมือนกับไม่มี ความผิดพลาดเกิดขึ้น ทำให้ Layer ที่สูงขึ้นไป (ซึ่งก็คือ Network Layer) นำไปใช้งานได้อย่างถูกต้อง วิธีการ ก็คือทางฝ่ายส่งจะทำการแตกข้อมูลออกเป็นก้อนๆ เรียกว่า เฟรมข้อมูล (Data – Frame) ปกติประมาณไม่กี่ ร้อยไบต์ ทำการส่ง Frame ข้อมูลออกไปทีละชุดและรอรับการตอบรับ (Acknowledge Frame) ซึ่งตอบ กลับมาโดยข้อมูล โดยปกติแล้ว Physical Layer จะไม่สนใจข้อมูลว่ามีความหมายใด จึงเป็นหน้าที่ของ Data Link Layer ที่จะต้องทำการสร้างและตรวจรับขอบเขตของเฟรม (Frame Boundary) ซึ่งสามารถทำ ไปได้โดยการเติมบิตเข้าไปในจุดเริ่มต้นและจุดสุดท้ายของ Frame แต่จุดที่ต้องระวังก็คือ ถ้าหากข้อมูลที่เพิ่ม เข้าไปซ้ำกับข้อมูลที่ต้องการส่งจริงๆ

สัญญาณรบกวนจากภายนอกก็เป็นปัญหาหนึ่งที่จะทำให้ Frame ขาดหายไปได้ ในกรณี นี้โปรแกรมที่ควบคุม Data Link Layer ที่เครื่องต้นทางจะส่งข้อมูลเข้ามาใหม่ อย่างไรก็ตามการส่ง Frame เดียวกันออกมาหลายๆ ครั้งก็อาจจะทำให้เกิด Frame ซ้ำกันได้ วิธีการป้องกันก็คือ Frame ที่ซ้ำกันจะ ส่งออกไปก็ต่อเมื่อมีการตอบรับ (Acknowledgment Frame) ส่งมาบอกว่าข้อมูลหายหรือถูกทำลายไปก่อน ภายใน Data Link Layer จะมีการเชื่อมต่อ (Interface) หลายๆ แบบเพื่อให้ Network Layer สามารถเลือกใช้ ไปได้

ชั้นที่ 3. เน็ตเวิร์กเลเยอร์ (Network Layer)

ภายใน Network Layer จะเกี่ยวข้องกับการควบคุมการทำงานของเครือข่ายย่อย ประเด็นที่ สำคัญก็คือการพิจารณาว่า Packet จะถูกส่งจากต้นทางไปยังปลายทางได้อย่างไร การหาเส้นทาง (Route) อาจจะวางอยู่บนตารางที่คงที่อยู่ และเชื่อมโดยตรงเข้ากับ Network และมีการเปลี่ยนตารางน้อยมาก การ กำหนดเส้นทางจะกำหนดตอนเริ่มต้นติดต่อก็ได้หรืออาจจะใช้วิธีที่สามารถเปลี่ยนแปลงได้ตลอดเวลา (Dynamic) ซึ่งเส้นทางที่ Packet เดินทางไปจะถูกกำหนดแบบ Packet ต่อ Packet

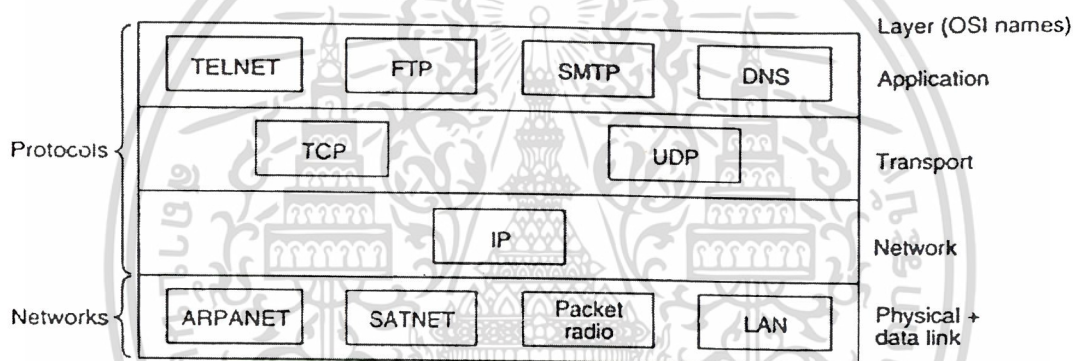
ถ้ามีจำนวน Packet มากเกินไปภายในเครือข่ายย่อย จะทำให้เกิดปัญหาคอขวด คือจำนวน Packet เข้ามามากแต่ทางออกน้อย ภายใน Network Layer จะต้องจัดการกับปัญหาเหล่านี้ด้วย เมื่อ Packet จะต้องเดินทางจาก Network หนึ่งไปยังอีก Network หนึ่งจะทำให้เกิดปัญหาต่างๆ ขึ้นเช่น Address ของ Network ที่ 2 อาจจะแตกต่างจาก Network ที่ 1 ทำให้ Network ที่ 2 ไม่สามารถรับ Packet นี้ได้เลย หรือ โพรโตคอลที่ใช้อาจจะแตกต่างกัน Network Layer จะต้องจัดการกับปัญหาเหล่านี้เพื่อให้ Network ทั้งหลายติดต่อกันได้เสมือน Network เดียวกัน ในระบบ Network แบบกระจาย (Broadcast Network) วิธีการหาเส้นทางของ Packet จะง่ายมาก ทำให้ Network Layer มีขนาดเล็กหรืออาจจะไม่มีเลยก็ได้

ชั้นที่ 4. ชั้นสื่อสารนำส่งข้อมูล (Transport Layer)

ชั้นสื่อสารนำส่งข้อมูล (Transport Layer) ซึ่งมีหน้าที่การทำงานเหมือนกันกับชั้นจัดการ นำส่งข้อมูลของมาตรฐาน OSI แบ่งออกเป็นโพรโตคอลสองประเภท ประเภทแรกเรียกว่า TCP (Transmission Control Protocol) เป็นแบบที่มีการกำหนดช่วงการสื่อสารตลอดระยะเวลาการสื่อสาร เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Connection-Oriented) ซึ่งจะยอมให้มีการส่งข้อมูลเป็นไบต์ที่ไว้วางใจได้โดยไม่มีข้อผิดพลาด ข้อมูลที่มีปริมาณมากจะถูกแบ่งออกเป็นส่วนเล็กๆ เรียกว่าข้อมูลซึ่งจะถูกส่งไปยังผู้รับผ่านทางชั้นสื่อสารของอินเทอร์เน็ตทางฝ่ายผู้รับจะนำข้อมูลมาเรียงต่อกันตามลำดับเป็นข้อมูลตัวเดิม TCP ยังมีความสามารถในการควบคุมการไหลของข้อมูลเพื่อป้องกันไม่ให้ผู้ส่งส่งข้อมูลเร็วเกินกว่าที่ผู้รับจะทำงานได้ทันอีกด้วย

กฎการส่งข้อมูลที่สองเรียกว่า UDP (User Datagram Protocol) เป็นการติดต่อแบบไม่ต่อเนื่อง (Connection Less) ไม่มีการตรวจสอบความถูกต้องของข้อมูลเหมือนกับแบบ TCP อย่างไรก็ตามวิธีการนี้มีข้อดีในด้านความเร็วในการส่งข้อมูลจึงนิยมใช้ในระดับผู้ให้และผู้ให้บริการ (Client/Server System) ซึ่งมีการสื่อสารแบบถาม และตอบ (Request/Reply) นอกจากนี้ยังใช้ในการส่งข้อมูลประเภทภาพเคลื่อนไหวหรือการส่งเสียง (Voice) ทางอินเทอร์เน็ต รูปที่ 2.10 แสดงให้เห็นความสัมพันธ์ของ IP, TCP และ UDP กฎการสื่อสารแบบ IP นี้เป็นระบบที่ได้รับความนิยม และได้รับการนำไปใช้ในระบบสื่อสารหลายระบบ



รูปที่ 2.10 กฎการสื่อสารในรูปแบบ TCP/IP

ชั้นที่ 5. ชั้นสื่อสารการประยุกต์ (Application Layer)

ตามมาตรฐาน TCP/IP ไม่มีการกำหนดชั้น Transport Layer และชั้นควบคุมหน้าต่างสื่อสารตามที่ปรากฏ ในรูปแบบมาตรฐาน OSI ทั้งนี้จากการกำหนดมาตรฐานของ OSI พบว่าโปรแกรมสำหรับชั้นควบคุมการสื่อสารสองชั้นนี้จะมีประโยชน์ในการใช้งานจริงน้อยมาก ระบบเครือข่าย TCP/IP จึงตัดทั้งสองชั้นนี้ออกไป ดังนั้นชั้นการ Application Layer จึงกลายเป็นชั้นที่อยู่เหนือชั้น Transport Layer

โปรแกรมในชั้น Application Layer จะรวมเอาหน้าที่การทำงานที่จำเป็นของสองชั้นที่หายไปไว้ที่นี่ อันได้แก่ โพรโตคอลสำหรับสร้างจอเทอร์มินัลเสมือน เรียกว่า TELNET โพรโตคอลสำหรับการจัดการเพิ่มข้อมูล เรียกว่า FTP และโพรโตคอลสำหรับการให้บริการให้บริการจดหมายอิเล็กทรอนิกส์ เรียกว่า SMTP ดังรูปที่ 2.18 โพรโตคอลสำหรับจอเทอร์มินัลเสมือนช่วยให้ผู้ใช้สามารถติดต่อกับเครื่อง Host ที่อยู่ไกลออกไปโดยผ่านทางอินเทอร์เน็ตและสามารถทำงานได้เสมือนกับว่ากำลังนั่งทำงานอยู่ที่ Host นั้น โพรโตคอลสำหรับการจัดการเพิ่มข้อมูลช่วยในการคัดลอกเพิ่มข้อมูลมาจากเครื่องอื่นที่อยู่ในระบบเครือข่ายหรือส่งสำเนาเพิ่มข้อมูลไปยังเครื่องใดๆ ก็ได้ โพรโตคอลสำหรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้บริการจดหมายอิเล็กทรอนิกส์ช่วยในการจัดส่งข้อความไปยังผู้ใช้ในระบบ หรือรับข้อความที่มีผู้ส่งเข้ามา

2.10.4 การกำหนด IP Address

ที่อยู่บนอินเทอร์เน็ตกำหนดโดยใช้เลขฐานสองความยาว 4 ไบต์ หรือ 32 บิตซึ่งจะไม่มีที่อยู่ที่มีหมายเลขซ้ำกันเลย ประกอบด้วยตัวเลข 3 ประเภทคือ เลขบอก Class, เลขบอกเครือข่าย และ เลขบอก Host การกำหนดที่อยู่แบ่งออกเป็น 4 Class คือ A, B, C, D และ E ดังแสดงในรูปที่ 2.20 มีรายละเอียดดังนี้

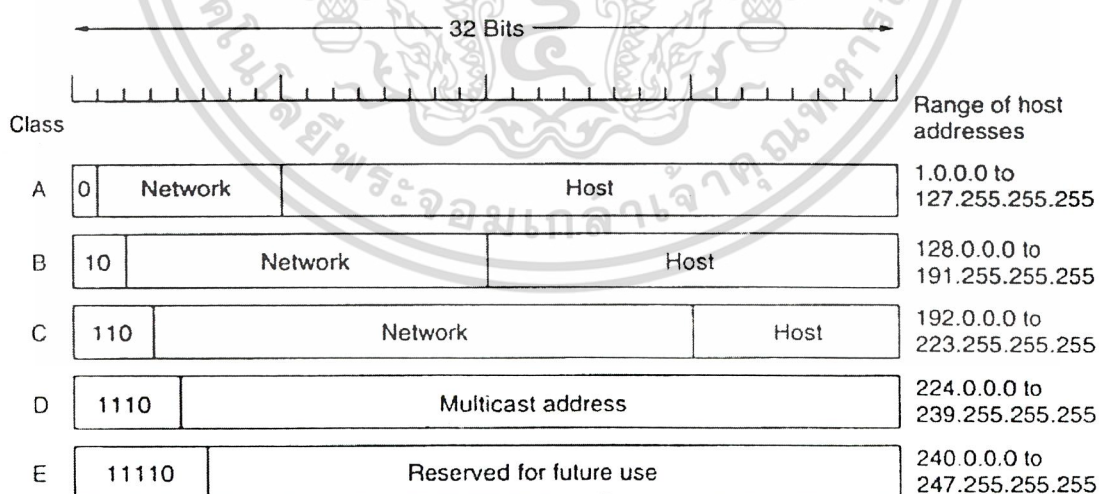
Class A : บิตแรกเป็น “0” เลขบอกเครือข่ายยาว 7 บิตประกอบด้วย 126 เครือข่ายแต่ละเครือข่ายมี 16 ล้าน Host บอกด้วยตัวเลข 24 บิต ขอบเขตหมายเลขที่อยู่คือ 1.0.0.0 ถึง 127.255.255.255

Class B : สองบิตแรกเป็น “10” เลขบอกเครือข่ายยาว 14 บิตประกอบด้วย 16,382 เครือข่าย แต่ละเครือข่ายมี 65,536 Host บอกด้วยตัวเลข 16บิต ขอบเขตหมายเลขที่อยู่คือ 128.0.0.0 ถึง 191.255.255.255

Class C : สามบิตแรกเป็น “110” เลขบอกเครือข่ายยาว 21 บิตประกอบด้วย 2,097,152 เครือข่ายแต่ละเครือข่ายมี 254 Host บอกด้วยตัวเลข 8 บิต ขอบเขตหมายเลขที่อยู่คือ 192.0.0.0 ถึง 223.255.255.255

Class D : สี่บิตแรกเป็น “1110” ใช้สำหรับการกระจายข้อมูลข่าวสารแบบ หลายจุด ขอบเขตหมายเลขที่อยู่คือ 224.0.0.0 ถึง 239.255.255.255

Class E : ห้าบิตแรกเป็น “11110” สำรองไว้ใช้ในอนาคต ขอบเขตหมายเลขที่อยู่คือ 240.0.0.0 ถึง 247.255.255.255



รูปที่ 2.11 โครงสร้างที่อยู่แบบ IP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11 คำสั่ง AT commands

คือ ชุดคำสั่งที่ใช้ในการติดต่อกับเครื่องโทรศัพท์หรืออุปกรณ์ ติดต่อโดยส่งรหัสตัวอักษรที่ขึ้นต้นด้วย AT+ คำสั่ง ตามด้วย รหัส CR + LF (Return) + (Line Feed)

ในรูปแบบคำสั่ง AT commands (Attention Commands) ใช้ในการตรวจสอบการติดต่อสื่อสารระหว่างโทรศัพท์และอุปกรณ์ คำสั่ง AT COMMAND ที่ใช้ในโครงการนี้มีดังนี้

2.11.1 AT คือ คำสั่งเช็คความพร้อมของมือถือ

รูปแบบคำสั่ง

AT <CR>

2.11.2 ATD คือ คำสั่งที่ใช้สำหรับการโทรออก

รูปแบบคำสั่ง

ATDyyxxxxxxx; <CR>

- yy หมายถึง รหัสขึ้นต้นเบอร์โทรศัพท์ เช่น 01,06,09
- xxxxxxxx หมายถึง เบอร์โทรศัพท์

2.11.3 ATH คือ คำสั่งที่ใช้ในการวางสายโทรศัพท์ ยกเลิกการโทรออก

รูปแบบคำสั่ง

ATH <CR>

2.11.4 AT+CMGS คือ คำสั่งที่ใช้สำหรับส่งข้อความ

รูปแบบคำสั่ง

AT+CMGS = ความยาวของข้อมูลที่จะส่งตามด้วยรหัสแอสกี <CR>

โดยความยาวของข้อมูลที่จะส่ง คือ

ความยาวของส่วน (Header + ความยาวของข้อมูลที่เป็น PDU Code -3)/2

หลังจากนั้นโทรศัพท์จะตอบกลับ ">" ให้เราส่งข้อมูลส่วน Header + PDU Code ตามด้วยรหัสแอสกี 26

2.11.5 AT+CMNI คือ คำสั่งที่ทำให้รู้ว่าได้รับข้อความเข้ามาที่โทรศัพท์ เอาไว้เก็บลำดับข้อความที่เข้ามาว่าเป็นข้อความที่เท่าไร

รูปแบบคำสั่ง

AT+CNMI = <mode>, <mt>, <bm>, <ds>, <bfr> <CR>

กำหนดให้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AT+CNMI = 1, 1, 2, 2, 1 <CR>

2.11.6 AT+CMGR คือ คำสั่งที่ใช้สำหรับอ่านข้อความ

รูปแบบคำสั่ง

AT+CMGR = <index> <CR>

- index ลำดับของข้อความ

เมื่อใช้คำสั่งอ่านข้อความจะได้รายละเอียดของข้อมูลดังนี้

-	ตัวอย่างข้อมูล
	การส่ง SMS เป็นข้อความว่า “Fire!” ไปที่หมายเลข 040734957
	06916681118088040A916686091635000030302002477182054669726521
06	ความยาวของ SMSC Information 6 Octets (bytes)
91	รูปแบบเลขหมาย SMSC 91 หมายถึงเลขหมายแบบสากล
66 81 11 80 88	เลขหมาย SMSC (แบบ decimal semi-octets) ซึ่งจะเป็นเลขฐานสิบสลับ nibble
04	เป็น Octet แรกที่ใช้ในการส่งข้อความ
0A	ความยาวของหมายเลขผู้ส่ง (10 ตัว)
91	ประเภทของ Address
66 86 09 16 35	เลขหมายผู้ส่ง (แบบ decimal semi-octets) ซึ่งจะเป็นเลขฐานสิบสลับ nibble
00	ระบุดตัวโปรโตคอล
00	Data Coding Scheme
30 30 20 02 47 71 82	TP-SCTS. ข้อมูล Time stamp
05	จำนวนตัวอักษร
46 69 72 65 21	เป็น PDU Code ซึ่งมีข้อความ “Fire!”

ตารางที่ 2.3 รายละเอียดของข้อมูลในการส่ง SMS

2.11.7 AT+CMGL คือ คำสั่งเรียกดูข้อความ

รูปแบบคำสั่ง

AT+CMGL = <Stat> <013>

- <Stat> = (0, 1, 2, 3, 4)

0 = ข้อความที่ยังไม่อ่าน

1 = ข้อความที่อ่านแล้ว

2 = ข้อความที่ไม่ได้ส่ง

3 = ข้อความที่ส่งแล้ว

4 = ดูข้อความทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11.8 AT+CMGD คือ คำสั่งที่ใช้สำหรับลบข้อความ

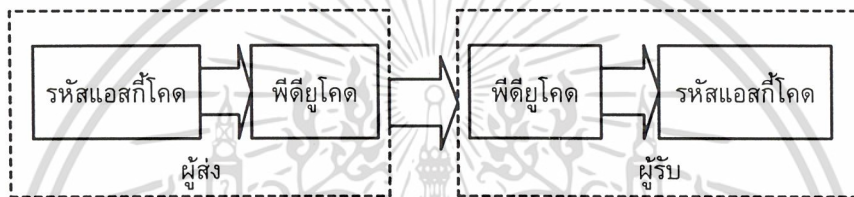
รูปแบบคำสั่ง

AT+CMGD = <index> <CR>

- index ลำดับของข้อความ

2.12 ระบบการทำงานการรับส่งข้อความในโทรศัพท์เคลื่อนที่

การรับส่งข้อความแบบพีดียูโหมด (PDU Mode) คือ ในการเขียนตัวอักษรหรือข้อความจะเหมือนกับเท็กซ์โหมด แต่ในการส่งข้อความจะมีการแปลงจากรหัสแอสกีโคดให้เป็นรหัสพีดียูโคดและนำรหัสพีดียูโคดที่ได้ส่งให้กับเลขหมายปลายทาง และเลขหมายปลายทางจะต้องแปลงจากรหัสพีดียูโคดให้เป็นรหัสแอสกีโคดโดยแสดงดังรูป



รูปที่ 2.12 การรับส่งข้อความแบบพีดียูโหมด

การแปลงข้อมูลแอสกีโคดเป็นพีดียูโคด (ASCII Code To PDU Code) เนื่องจากเป็นขบวนการย้อนกลับ กล่าวคือให้เปลี่ยนรหัสแอสกีโคดทั้งหมดแปลงให้อยู่ในเลขฐานสอง หลังจากนั้นให้ตัดบิตบนสุดของแต่ละไบท์ทิ้งให้เหลือ 7 บิต ดังนี้

เลขฐานสิบหก	เลขฐานสอง
46	100 0110
69	110 1001
72	111 0010
65	110 0101
21	010 0001

ตารางที่ 3.4 การแปลงเลขฐานสิบหกเป็นเลขฐานสอง

หลังจากนั้นให้ทำการชิพบิตไปทางขวา โดยการชิพบิตมีหลักเกณฑ์ดังนี้ คือให้เริ่มชิพจากบรรทัดที่สองเป็นต้นไป โดยจำนวนการชิพจะเท่ากับจำนวนบรรทัดลบด้วย 1 เสมอ เมื่อชิพบิตแล้วข้อมูลในบรรทัดนั้นจะเหลือเท่ากับ 7 เช่น ชิบบรรทัดที่สองจำนวนชิพเท่ากับ 2-1 ข้อมูลบรรทัดจะเท่ากับ 7-1 คือเท่ากับ 6 ตัว ในบรรทัดที่สามจะเหลือเท่ากับ 5 ตัวเป็นต้น ข้อมูลที่ถูกชิพออกมาให้นำไปต่อที่หัวบรรทัดก่อนหน้าเสมอ แต่กรณีที่ชิพจนถึง 7 บิตแล้วจะเป็นการเริ่มขบวนการใหม่ จะได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลขฐานสอง(แอสกีโค้ด)	เลขฐานสอง(พีดียู)	พีดียูโค้ด
100 0110	1100 0110	C6
110 100 <u>1</u>	1011 0100	B4
111 00 <u>10</u>	1011 1100	BC
110 0 <u>101</u>	0001 1100	1C
010 000 <u>1</u>	0000 0010	02
000 000 <u>0</u>		

ตารางที่ 3.5 การแปลงเลขฐานสองเป็นเลขพีดียูโค้ด

การแปลงข้อมูลพีดียูโค้ดเป็นแอสกีโค้ด (PDU Code To ASCII Code) เริ่มจากนำพีดียูโค้ดทั้งหมดแปลงให้อยู่ในเลขฐานสองเพื่อง่ายต่อความเข้าใจดังนี้

พีดียูโค้ด	เลขฐานสอง
C6	1100 0110
B4	1011 0100
BC	1011 1100
1C	0001 1100
02	0000 0010

ตารางที่ 3.6 การแปลงเลขพีดียูเป็นเลขฐานสอง

หลังจากนั้นให้ทำการชิพบิตบนสุดไปทางซ้าย โดยการชิพบิตมีหลักเกณฑ์ดังนี้ คือ ให้เริ่มชิพจากบรรทัดแรกโดยจำนวนการชิพจะเท่ากับจำนวนบรรทัด เมื่อชิพบิตแล้วให้นำเลข "0" มาใส่ไว้ข้างหน้าหนึ่งตัวเสมอ ข้อมูลที่ถูกชิพออกมาในแต่ละบรรทัดให้นำไปต่อที่บิตสุดท้ายของบรรทัดถัดไป เช่น ข้อมูลที่ถูกชิพมาจากบรรทัดที่หนึ่งจะถูกชิพมาต่อท้ายบรรทัดที่สองดังนี้

เลขฐานสอง	เลขฐานสิบหก	รหัสแอสกีโค้ด
1100 0110	46	F
1011 0100,1	69	i
1011 1100,10	72	r
0001 1100,101	65	e
0000 0010,0001	21	!

ตารางที่ 3.7 การแปลงเลขฐานสองเป็นรหัสแอสกีโค้ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

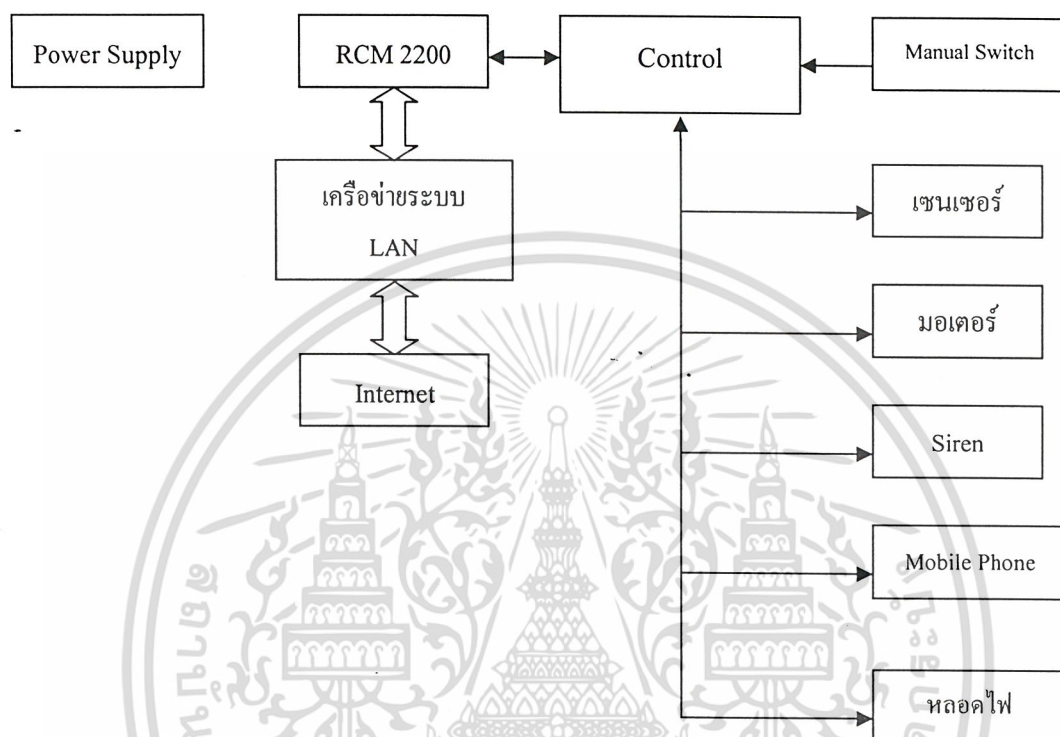
แต่กรณีที่ซิพจนถึง 7 บิตแล้วจะเป็นการเริ่มขบวนการใหม่นั้นเอง ในกรณีที่เมื่อทำงานถึงตัวสุดท้าย แต่ว่าข้อมูลตัวสุดท้ายไม่ครบ 8 บิต ที่จะอ่านเป็นแอสกีโคดได้ให้เติม"0"จนครบแล้วอ่านข้อมูลที่ เป็นเลขฐานสองเปลี่ยนให้เป็นเลขฐานสิบหกและแทนในรหัสแอสกีโคด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3 การคำนวณและการสร้าง

3.1 การออกแบบวงจรหลัก



รูปที่ 3.1 โค้ดแกรมของระบบ

จากรูปที่ 3.1 วงจร RCM 2200 จะเป็นวงจรที่สามารถประมวลผลเพื่อควบคุมการทำงานของระบบ และสามารถรับค่าสถานะของเซนเซอร์นำมาประมวลผลเพื่อแสดงค่าผ่านทางหน้า webpage แล้วยังสามารถควบคุมกลับได้ โดยใช้เครือข่ายระบบ LAN ในการติดต่อสื่อสารทางอินเทอร์เน็ต

ส่วนวงจร Control จะใช้ IC 8255 เพื่อทำการขยายพอร์ทของ RCM2200 โดยในส่วนของเซนเซอร์จะใช้ Magnetic Switch ติดตั้งไว้ที่ประตูและหน้าต่างเพื่อแสดงสถานะ และจะมีการตรวจสอบทุกวัน ซึ่งจะเป็นสาเหตุของการเกิดอัคคีภัย

ชุดวงจร Manual Switch จะมีไว้ติดตั้งอยู่ในบ้านเพื่อทำการ เปิด/ปิดระบบ, ล็อก/ปลดล็อกประตู, เปิด/ปิด Siren และทำการ เปิด/ปิด ระบบไฟฟ้า ซึ่งสวิตซ์ทั้ง 4 ตัว ยังสามารถควบคุมผ่านทางหน้าจอ webpage ได้ด้วย การที่ระบบจะสามารถเปิดได้ ต้องทำการปิดประตูและหน้าต่างก่อน และระบบเตือนภัยต้องอยู่ในสถานะปกติด้วย ถ้าประตูหรือหน้าต่างเปิดอยู่ ระบบก็จะไม่สามารถเปิดได้ แล้วจะมีการแจ้งเตือนทาง หน้า webpage

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของการควบคุมมอเตอร์จะใช้เพื่อทำการ ล็อกและปลดล็อกประตูของห้อง ซึ่งเมื่อทำการ ล็อกประตูอัตโนมัติเมื่อทำการเปิดระบบเตือนภัย ถ้าประตูยังเปิดอยู่ระบบจะไม่สามารถล็อกประตูได้ และ จะแจ้ง ผ่านที่ หน้า webpage

ในส่วนของชุด Siren ในวงจรนี้จะสามารถเลือก แรงดันไฟฟ้าของ Siren ได้ 2 สถานะ คือ 5V หรือ 12V โดยจะใช้รีเลย์ในการทำให้ Siren ทำงาน โดยในโครงการนี้จะสามารถเปิด/ปิด Siren ได้ทั้ง ทาง Manual Switch และปุ่มที่หน้า webpage

ในส่วนของหลอดไฟฟ้า จะใช้แสดง สถานการณ์เตือนภัย ถ้ามีผู้บุกรุกหลอดไฟจะสว่าง โดย สามารถควบคุมได้ทั้งทาง Manual Switch และปุ่มที่หน้า webpage

วงจรส่วนแจ้งเตือนภัยด้วยข้อความสั้น ในโครงการนี้ จะใช้ MCS 51 ในการทำการติดต่อกับ โทรศัพท์เคลื่อนที่ ซีเมนส์รุ่น C35 โดยจะส่งข้อความเมื่อระบบต้องมีการเปิด และเกิดเหตุผิดปกติขึ้น ภายในบ้าน เช่น มีการเปิดประตู หน้าต่าง หรือ เกิดควันขึ้นภายในบ้านก็จะให้ RCM 2200 ทำการสั่งงาน ให้ MCS 51 ทำการส่ง PDU Code ไปที่พอร์ทอนุกรม เพื่อติดต่อให้โทรศัพท์เคลื่อนที่ทำการส่งข้อความ สั้นไปที่โทรศัพท์มือถือปลายทาง

3.2 อุปกรณ์ตรวจจับการเปิด/ปิด

ในส่วนของอุปกรณ์ตรวจจับนี้จะใช้สวิทช์แม่เหล็ก (Magnetic Switch) จากรูปที่ 3.2 รูปร่างของ สวิทช์แม่เหล็ก ที่นำมาใช้เป็นตัวตรวจจับเซ็นเซอร์ รูปที่ 3.2 (ก) รูปร่างของสวิทช์แม่เหล็กแบบปกติปิด รูปที่ 3.2 (ข) รูปร่างของสวิทช์แม่เหล็กแบบปกติเปิด



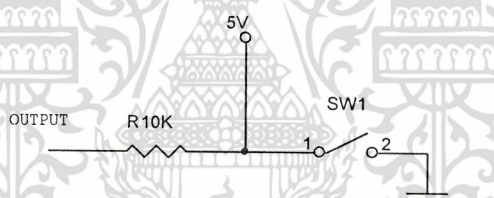
รูปที่ 3.2 รูปร่างของสวิทช์แม่เหล็ก

จาก รูปที่ 3.3 เป็นโครงสร้างของสวิทช์แม่เหล็กที่นำมาใช้งาน ส่วนรูปที่ 3.3 (ก) เป็นโครงสร้าง ของสวิทช์แบบปกติปิด หน้าสัมผัสของสวิทช์ต่อกันอยู่ (NC) และ รูปที่ 3.3 (ข) เป็นโครงสร้างของ สวิทช์แบบปกติเปิด หน้าสัมผัสของสวิทช์จะจากกัน (NO)



รูปที่ 3.3 โครงสร้างของสวิตช์แม่เหล็ก

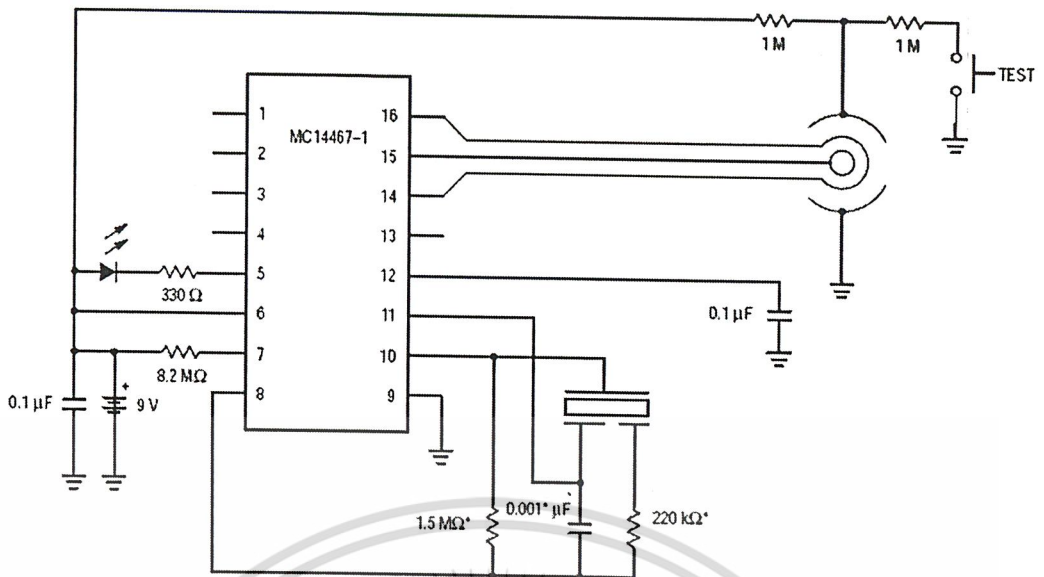
การออกแบบวงจรเซ็นเซอร์ จะใช้สวิตช์แม่เหล็กเป็นตัวตรวจจับสิ่งผิดปกติที่เกิดขึ้นโดยสวิตช์จะยึดติดอยู่กับขอบประตู



รูป 3.4 วงจรเซ็นเซอร์

3.3 อุปกรณ์ตรวจจับควัน

อุปกรณ์ตรวจจับควันโดยอาศัยหลักการเกิดไอออน (Ionization Smoke Detection) ส่วนใหญ่จะเป็นอุปกรณ์ชนิดจุด ประกอบด้วยสารกัมมันตภาพรังสี บรรจุอยู่ในกล่องตรวจจับ เป็นผลให้อากาศภายในกล่องเกิดไอออนมีสภาพเป็นตัวนำ และทำให้กระแสไฟฟ้าไหลผ่านได้ระหว่างขั้วอิเล็กโทรด เมื่ออนุภาคควันผ่านเข้ากล่อง อนุภาคควันจะจับตัวรวมกับไอออนทำให้การเคลื่อนที่ช้าลง และค่าความนำไฟฟ้าลดลง อุปกรณ์ตรวจจับจะเริ่มสัญญาณเมื่อค่าความนำลดลงถึงจุดพิคัดที่กำหนดไว้



รูป 3.5 วงจรตรวจสอบควัน

3.4 ไอซี เบอร์ TA7279P

ภายใน IC จะประกอบด้วยวงจร H-Bridge SW จำนวน 2 ชุด (สามารถควบคุมมอเตอร์ได้ 2 ตัว) แต่ละชุดมีคุณสมบัติดังนี้

- รับแรงดันได้ตั้งแต่ 0 ถึง 18 Volts
- เอาท์พุทจ่ายกระแสได้ถึง 1.0A (AVE), 3.0A (Peak)

นอกจากนั้นภายใน ยังมีฟังก์ชันหยุดการทำงานเมื่ออุณหภูมิสูงเกินไป และยังมีวงจรจำกัดกระแสอีกด้วย การนำไปใช้งาน

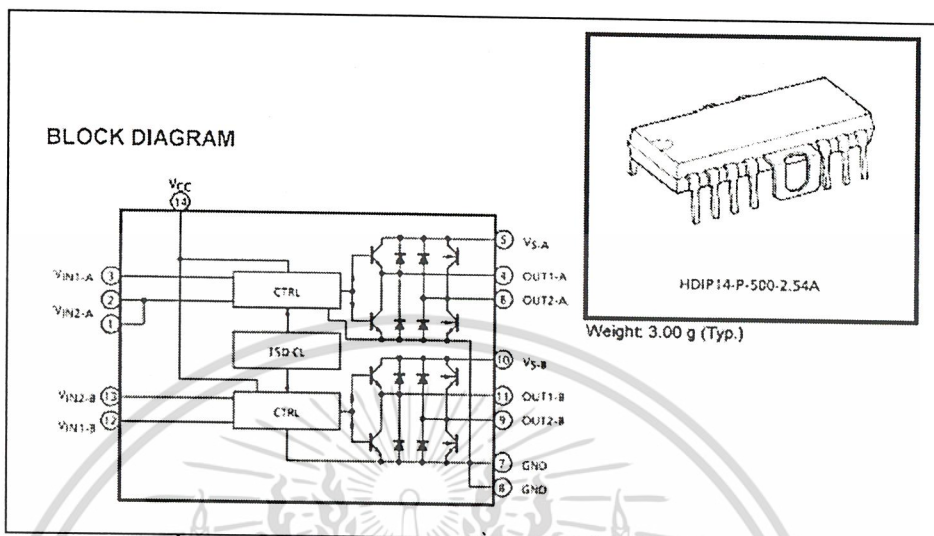
การนำไอซี TA7279P ไปใช้งานนั้นไม่จำเป็นต้องต่ออุปกรณ์มากมาย คือเราจะต่อไฟเลี้ยงให้กับตัวไอซีที่ 12V แล้วเราก็จะทำการป้อนอินพุท PA0-PA3 ที่มาจากไอซี 8255 ให้กับตัวไอซี TA7279P เพื่อควบคุมการหมุนดังนี้

อินพุท	เอาท์พุท
00	มอเตอร์ไม่หมุน
01	มอเตอร์หมุนซ้าย
10	มอเตอร์หมุนขวา
11	มอเตอร์ไม่หมุน

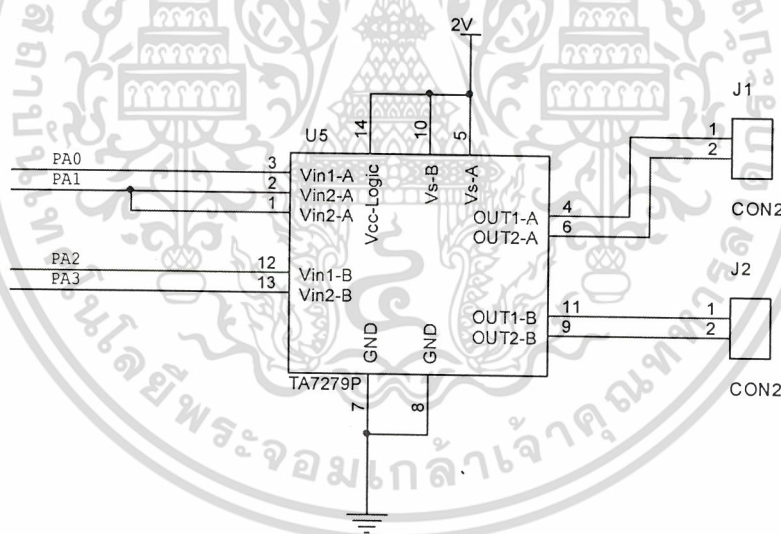
ตารางที่ 3.1 แสดงสถานะ และการทำงานของมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากวงจรรวมนั้นเราจะใช้ควบคุมมอเตอร์ตัวเดียวซึ่งไอซี TA7279P นั้นจะสามารถควบคุมมอเตอร์ได้ 2 ตัวโดยอินพุตจะมี 2 ชุด คือ ขา 2 กับ 3 และขา 12 กับ 13 ส่วนเอาต์พุตก็จะมี 2 ชุดเช่นกันคือขา 4 กับ 6 และขา 11 กับ 9



รูปที่ 3.6 ขาของไอซี TA7279P และ โครงสร้างภายใน



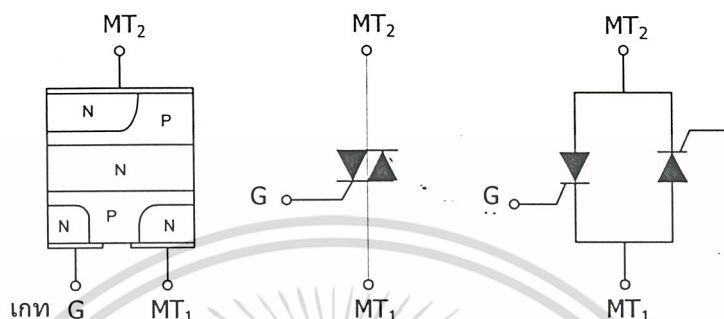
รูปที่ 3.7 การต่อไอซี TA7279P เพื่อใช้งานในวงจรรวม

เนื่องจากไอซี TA7279P นี้ใช้ง่ายแต่เราควรระวังเรื่องของเอาต์พุตของตัวไอซีเพราะไอซีชนิดนี้เป็นไอซีที่ใช้ขับกระแสของมอเตอร์ดังนั้นห้ามต่อเอาต์พุตที่ติดกันเด็ดขาดเพราะอาจจะทำให้ไอซีพังได้ ส่วนในวงจรขนาดใหญ่ขึ้นนั้นควรมีตัวคาปาซิเตอร์ต่อบายพาส โดยมีค่าประมาณ 10 μ F เพื่อขับกระแสออกเอาต์พุตเพราะในวงจรใหญ่จะมีโหลดอยู่มากมายทำให้กระแสที่ได้ไม่นิ่งไม่เพียงพอสำหรับขับมอเตอร์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 ทฤษฎีของไทรแอกเพื่อใช้ในการตัดต่อกับอุปกรณ์ไฟฟ้าภายนอก

ไทรแอกเป็นอุปกรณ์จำพวกสารกึ่งตัวนำที่จัดอยู่ในสารกึ่งตัวนำประเภททรินสเตอร์ สามารถนำไปใช้กับแรงดันไฟสลับได้ และจะนำได้สองทิศทาง ทั้งช่วงบวกและช่วงลบของแรงดันไฟสลับ

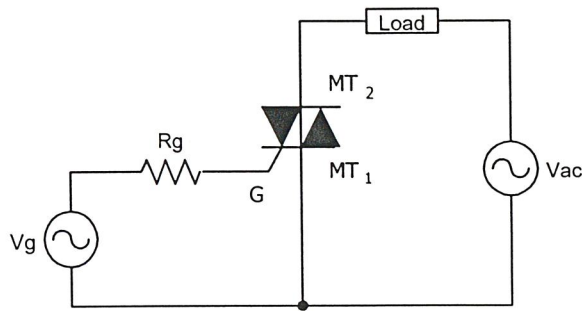


รูปที่ 3.8 โครงสร้าง, สัญลักษณ์และสัญลักษณ์เสมือนของไทรแอก

การนำกระแสของไทรแอกจะขึ้นอยู่กับแรงดันที่ป้อนกระตุ้นขา G (Gate) และแรงดันที่จ่ายให้ MT_2 และ MT_1 จะเป็นช่วงบวกหรือช่วงลบ ไทรแอกก็ยังสามารถทำงานได้ ดังนั้นสามารถที่จะไบอัสเพื่อให้ตัวไทรแอกทำงานได้ทั้งหมด 4 สถานะ คือ

- สถานะที่ 1 หรือควอดแดนต์ 1 จ่ายแรงดันบวกให้ขา MT_2 จ่ายแรงดันลบให้ขา MT_1 และจ่ายแรงดันบวกกระตุ้นขา G (กระแสไหลเข้าขา G)
- สถานะที่ 2 หรือควอดแดนต์ 2 ยังคงจ่ายแรงดันบวกให้ขา MT_2 จ่ายแรงดันลบให้ขา MT_1 แต่จ่ายแรงดันลบกระตุ้นขา G (กระแสไหลออกขา G)
- สถานะที่ 3 หรือควอดแดนต์ 3 กลับขั้วแรงดันที่จ่ายให้ MT_2 เป็นลบ จ่ายแรงดันให้ MT_1 เป็นบวก และส่วนขา G ยังคงจ่ายเป็นลบเหมือนเดิม (กระแสไหลออกขา G)
- สถานะที่ 4 หรือควอดแดนต์ 4 ยังคงจ่ายแรงดันลบให้ขา MT_2 จ่ายแรงดันบวกให้ขา MT_1 ส่วนขา G กลับขั้วแรงดันเป็นจ่ายบวกให้ขา G เป็นแรงดันกระตุ้นเทียบขา MT_1 (กระแสไหลเข้าขา G)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 การใช้งานไครแอคเป็นสวิตช์ ในวงจรไฟฟ้า

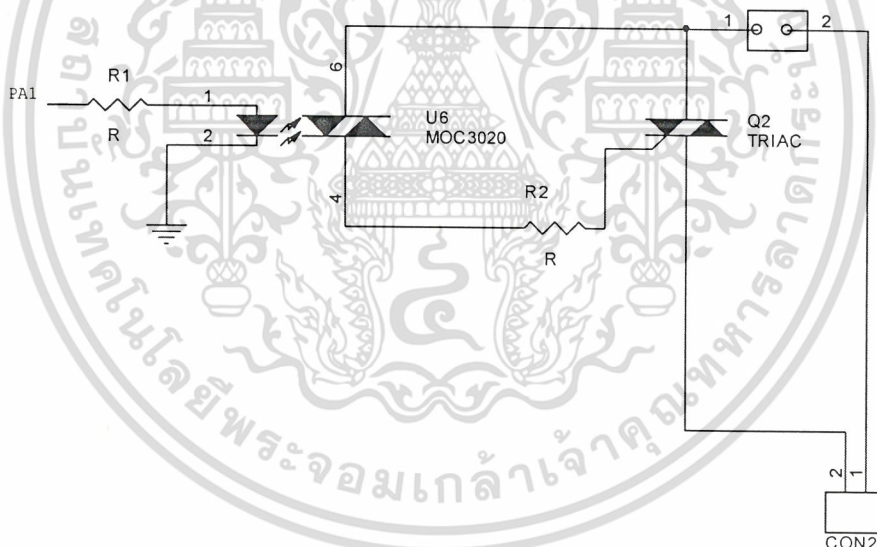
โดยการใช้งานไครแอคควรเลือกใช้ควอดแดนท์ที่ 1 และควอดแดนท์ที่ 3 ซึ่งเป็น ควอดแดนท์ที่มีการจุดชนวนง่ายที่สุด จากรูปที่ 3.9 เป็นการใช้งานไครแอคในวงจรไฟฟ้าซึ่งการจุดชนวนเกิดจะต้องมีทั้งในวัฏจักรบวกและลบของแรงดันกระแสสลับ เพราะเมื่อแรงดันกระแสสลับทางด้านโหลดมีค่าลดลงใกล้ศูนย์โวลต์กระแสที่ไหลผ่านไครแอคจะมีค่าต่ำกว่ากระแสยึด (Holding Current) ไครแอคจะหยุดนำกระแสทันที

ไครแอคใช้เป็นสวิตช์ให้กระแสไฟสลับที่มี 3 ขั้ว ขั้วหนึ่งเป็นเกดซึ่งใช้ในการพิจารณาเบรคโอเวอร์โวลต์เดจ (Breakover Voltage) การทำงานของไครแอคเหมือน SCR อย่างไรก็ตามไครแอคสามารถที่จะทำงานได้ 2 ทิศทางเนื่องจากกระแสสะสมในตัวไครแอคทำให้ไม่สามารถส่งสัญญาณจากขั้วหนึ่งไปขั้วหนึ่งอย่างรวดเร็วได้ ดังนั้นไครแอคจึงเป็นอุปกรณ์ที่มีความถี่ต่ำมากโดยทั่วไปใช้กับ ความถี่ประมาณ 50 ถึง 60 เฮริทซ์

การออกแบบวงจรนั้น ได้คำนึงว่าสัญญาณเอาต์พุตที่ได้จากไมโครคอนโทรลเลอร์ RCM2200 นั้นเป็นสัญญาณไฟตรงขนาด 5 โวลต์ แต่เนื่องจากต้องการนำสัญญาณที่ได้ไปควบคุมอุปกรณ์ไฟฟ้า เมื่อคำนึงถึงจุดนี้แล้ว จึงใช้ไครแอคในการควบคุมทางด้านไฟสลับ เพราะว่าไครแอคสามารถทำงานทางไฟสลับได้ โดยที่การทำงานของไครแอคนั้นจะสามารถนำกระแสได้ทั้งช่วงบวกและช่วงลบ ซึ่งขึ้นอยู่กับแรงดันไฟสลับที่ไปกระตุ้นที่ขาเกด โดยได้เลือกใช้ไครแอคเบอร์ BTA12 600C ซึ่งสามารถรับแรงดันได้ 600 โวลท์ และทนกระแสได้ 12 แอมป์ ส่วนการใช้อุปกรณ์ที่ส่งสัญญาณจากไมโครคอนโทรลเลอร์ RCM2200 กับไครแอคนั้นใช้ เป็นออปโต้ไอโซเลเตอร์แบบไดแอค เหตุผลที่ใช้ชนิดนี้เพราะว่าเป็นออปโต้ไอโซเลเตอร์แบบไดแอคนั้นเป็นอุปกรณ์ที่แยกกราวด์ภายในตัวเองดังนั้นจึงแยกส่วนไฟตรง กับไฟสลับออกจากกันทำให้ป้องกันการเสียหายได้ และเหตุผลที่ใช้ชนิดที่เป็นไดแอคเพราะว่าไดแอคเป็นอุปกรณ์ที่สร้างขึ้นมาเพื่อทำงานกับไครแอคโดยใช้เป็นอุปกรณ์ทริกเพื่อเป็นการป้องกันการเสียหายที่ขาเกดของไครแอคเนื่องจากการกระชอกของแรงดันไฟ

วงจรนี้เป็นวงจรที่ใช้ในการติดต่อระหว่าง ไมโครคอนโทรลเลอร์ RCM2200 โดยวงจรได้ทำเป็นโซลิดสเตตรีเลย์ โดยมีเป็นออปโต้ไอโซเลเตอร์แบบไดแอค เพื่อนำไปทริกให้ขาเกดของไครแอคเพื่อไปควบคุมหลอดไฟฟ้าหรืออุปกรณ์ไฟฟ้าภายนอกที่ต้องการต่อไป โดยต่อเข้ากับโหลด โดยที่การเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำงานนั้น วงจรจะรับสัญญาณมาจาก ไมโครคอนโทรลเลอร์ RCM2200 ที่ Port PA1 เมื่อมีสัญญาณเอาต์พุต ออกมา จะส่งสัญญาณขนาด 5 โวลต์เข้ามา โดยสัญญาณจะไปตกคร่อมที่ตัวต้านทาน R_1 จากนั้นจะเข้าไปที่ขา 1 ของออปโตไอโซเลเตอร์ เมื่อออปโตไอโซเลเตอร์ ได้รับสัญญาณก็จะเริ่มทำงาน โดยการทำงานนั้นต้องพิจารณาภายในตัวออปโตไอโซเลเตอร์ ซึ่งจะเห็นว่าขาที่ทำการรับสัญญาณ คือขา 1 และ 2 เป็นขาที่ต่อกับไดโอด เมื่อมีการรับสัญญาณที่ขา 1 เทียบกราวด์ ที่ขา 2 จะทำให้ไดโอดนี้ทำงาน เมื่อไดโอดนี้ทำงานก็จะส่งสัญญาณ ผ่านไปให้ไดแอกซึ่งต่ออยู่ที่ขา 6 และ 4 โดยที่สัญญาณเอาต์พุตที่ได้ จะเป็นสัญญาณที่ใช้ในการทรริก หรือกระตุ้นขาเกตของไครแอก เพื่อเป็นการป้องกันแรงดันไฟกระชอกจำนวนมาก ที่อาจป้อนเข้ามาเกตของไครแอก จนเป็นเหตุให้ขาเกตของ ไครแอกเกิดการชำรุดเสียหายทันที โดยที่สัญญาณที่ออกจาก ออปโตไอโซเลเตอร์นั้นออกที่ขา 4 ผ่านตัวต้านทาน R_2 เป็นสัญญาณที่ใช้กระตุ้นให้ไครแอกทำงาน ส่วนการทำงานของไครแอกนั้น จะเห็นว่าไครแอกสามารถนำกระแสได้ทั้งช่วงบวกและช่วงลบของแรงดัน ซึ่งจะขึ้นอยู่กับแรงดันไฟสลับที่ไปกระตุ้นขาเกต จากวงจรจะเห็นว่าแรงดันที่ตกคร่อมตัวไครแอกและตกคร่อมโหลดจะสลับกันคือในจังหวะที่ไครแอกไม่นำกระแส ไครแอกจะมีความต้านทานสูง แรงดันที่ตกคร่อม จะตกคร่อมตัวไครแอกทั้งหมด ไม่มีแรงดันตกคร่อมโหลด และในจังหวะที่ไครแอกนำกระแส ตัวไครแอกจะมีความต้านทานต่ำ ทำให้ไม่มีแรงดันตกคร่อมตัวไครแอก แรงดันทั้งหมดจะตกคร่อมโหลด



รูปที่ 3.10 วงจรส่วนที่ใช้ในการตัดต่อกับอุปกรณ์ไฟฟ้าภายนอก

วิธีการคำนวณหาค่าอุปกรณ์ที่ใช้ในวงจรตัดต่อกับอุปกรณ์ไฟฟ้าภายนอก โดยข้อมูลต่างๆ อ้างอิงจากข้อมูลทางเทคนิคของอุปกรณ์แต่ละตัว

หา R_1 โดย $I_f = 15 \text{ mA}$, $V_d = 1.3 \text{ V}$, $V_s = 5 \text{ V}$

$$R_1 = \frac{V_s - V_d}{I_f}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แทนค่าลงในสมการหา R_1

$$R_1 = \frac{5V-1.3V}{15mA}$$

$$= 246.66\Omega$$

R_1 จะมีค่าประมาณ 300Ω

หา R_2 โดย $V_{in} = 155 V$, $I_{tsm} = 1 A$

$$R_2 = \frac{V_{in}}{I_{tsm}}$$

แทนค่าในสมการ

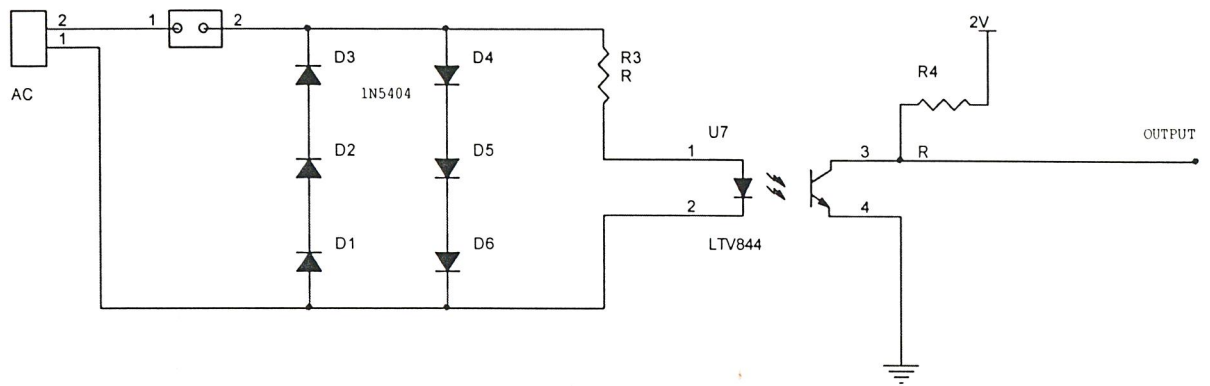
$$R_2 = \frac{155V}{1A}$$

R_2 จะมีค่าประมาณ 150Ω

3.6 วงจรตรวจสอบสถานะ

วงจรตรวจสอบสถานะ ดังรูปที่ 3.11 จะทำหน้าที่ตรวจสอบการทำงานของอุปกรณ์ไฟฟ้าแต่ละตัว โดยใช้สัญญาณที่ได้จากวงจรตรวจสอบสถานะเป็นสัญญาณอินพุตให้กับไมโครคอนโทรลเลอร์ โดยถ้าสัญญาณเป็นลอจิก “0” แสดงว่าอุปกรณ์ทำงาน แต่ถ้าสัญญาณเป็นลอจิก “1” แสดงว่าอุปกรณ์ไม่ทำงาน โดยวงจรตรวจสอบสถานะในโครงการจะกำหนดให้อุปกรณ์หรือโหลดใช้กระแสไม่เกิน 1 แอมป์ โดยใช้ ออปโตทรานซิสเตอร์ (Optotransistor) เบอร์ LVT 844 ซึ่งเป็นอุปกรณ์สารกึ่งตัวนำของอิเล็กทรอนิกส์ตัวหนึ่ง โดยการใช้เป็นสวิทช์เพื่อแยกทางไฟฟ้า ระหว่างอินพุตกับเอาต์พุตซึ่งเชื่อมโยงด้วยแสง (Optocoupling) โดยใช้ไฟกระแสสลับเป็นอินพุตสำหรับป้อนสัญญาณควบคุมเพื่อบังคับให้สวิทช์ทางด้านเอาต์พุตเปิดหรือปิดซึ่งนำกระแสได้เหมือนไดรแอก และใช้ไดโอดเบอร์ 1N5404 ซึ่งสามารถรับแรงดันได้ 400 โวลต์ และทนกระแสได้ 3 แอมป์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 วงจรตรวจสอบสถานะ

จากโหลด จะมีกระแสไฟฟ้าไหลผ่านมายังไดโอด และมีแรงดันตกคร่อมที่แอโนดและแคโทดประมาณ 2.1 โวลต์ ซึ่งแรงดันไฟฟ้าที่ได้จะเป็นเสมือนแหล่งจ่ายแรงดันให้เป็นอินพุทของออปโตทรานซิสเตอร์ ส่วน R_3 จะทำหน้าที่จำกัดกระแสที่ไหลเข้าไปยังออปโตทรานซิสเตอร์ส่วนทางด้านเอาต์พุทจะต่อ R_4 เพื่อให้กระแสไหลเพียงเล็กน้อยเมื่อออปโตทรานซิสเตอร์ทำงาน โดยนำแรงดันตกคร่อมระหว่างขาคอลเล็กเตอร์กับกราวด์ป้อนเข้าพอร์ตที่ใช้เช็คสถานะของ RCM2200

วิธีการคำนวณหาค่าอุปกรณ์ที่ใช้ในวงจรตรวจสอบสถานะ โดยข้อมูลต่างๆ อ้างอิงจากข้อมูลทางเทคนิคของอุปกรณ์แต่ละตัว

หาค่า R_3 โดย $I_f = 20 \text{ mA}$, $V_d = 1.4 \text{ V}$, $V_D = 2.1 \text{ V}$

$$R_3 = \frac{V_D - V_d}{I_f}$$

แทนค่าลงในสมการหา R_3

$$R_3 = \frac{2.1\text{V} - 1.4\text{V}}{20\text{mA}}$$

ดังนั้นจะได้ R_3 ประมาณ 39Ω

หาค่า R_4 โดย $I_c = 1 \text{ mA}$, $V_s = 5 \text{ V}$, $V_c = 3 \text{ V}$

$$R_4 = \frac{V_s - V_c}{I_c}$$

แทนค่าลงในสมการ R_4

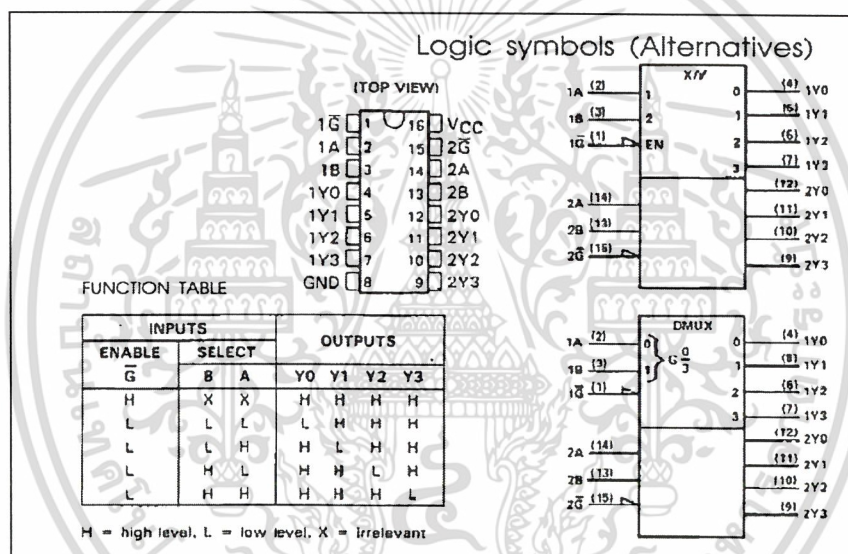
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$R_4 = \frac{5V-3V}{1mA}$$

ดังนั้นจะได้ R_4 เท่ากับ 2 k Ω

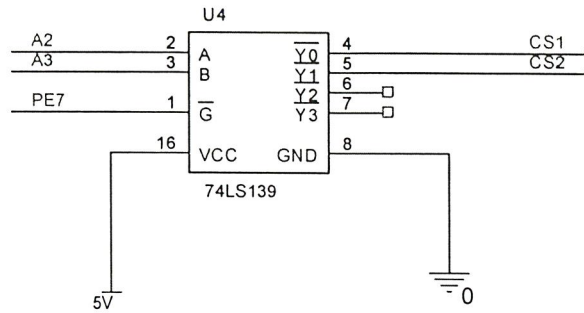
3.7 การนำไอซี เบอร์ 74LS139 ไปใช้งาน

การออกแบบวงจรถอดรหัสโดยใช้ไอซี 74LS139 เริ่มต้นโดยการต่อสัญญาณ A2 กับ A3 จาก Rabbit Core Module RCM2200 ไปยังขา 2 และ 3 ไอซี 74LS139 ซึ่ง A2 กับ A3 เป็นขา Address ที่ส่งมาจาก Rabbit Core Module RCM2200 ต่อขา PE เข้ายังขา 1 ของ 74LS139 ซึ่งขา PE7 ทำหน้าที่เป็นตัว Enable ให้กับไอซี 8255 ทำงานโดยกำหนดวง Address ไว้ที่ E000-E007



รูปที่ 3.12 ตำแหน่งขาของไอซี 74LS139 และ โครงสร้างภายใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

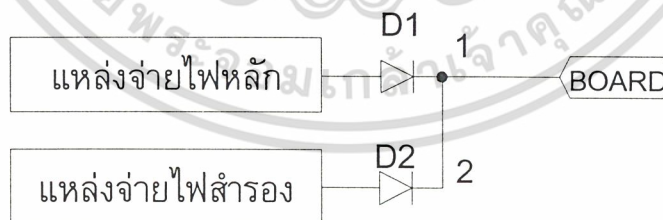


รูปที่ 3.13 การต่อไอซี 74LS139 เพื่อใช้งาน

3.8 แหล่งจ่ายไฟหลักและไฟสำรอง

ในส่วนของระบบไฟสำรองนี้ ทำหน้าที่ในการตัดต่อระบบไฟหลัก ในกรณีไฟฟ้าดับจะทำการตัดต่อระบบไฟให้มาใช้ในส่วนของระบบไฟสำรอง ซึ่งในการตัดต่อวงจรในโครงการนี้ใช้ไดโอดในการตัดต่อระบบซึ่งแสดงหลักการของระบบไฟสำรองได้ดังรูปที่ 3.14

จากรูปใช้คุณสมบัติการนำกระแสของไดโอด คือเมื่อแรงดันไฟหลักทำงาน ทำให้ไดโอด D_1 นำกระแส ซึ่งมีแรงดันตกคร่อมตัวมัน 0.7 โวลต์ หรือตามชนิดของไดโอด ซึ่งการที่ทำให้ไดโอด D_1 นำกระแสนั้นขึ้นอยู่กับศักดาไฟฟ้าที่จุด 1 ต้องมากกว่าศักดาไฟฟ้าที่จุด 2 เมื่อศักดาไฟฟ้าที่จุด 1 มากกว่า ทำให้ไดโอด D_2 ไปอัสถกลับ (Reverse Bias) จึงทำให้กระแสจากแหล่งจ่ายไฟสำรองไม่ไหลผ่านออกมา แต่เมื่อเกิดกรณีไฟฟ้าดับทำให้แหล่งจ่ายไฟหลักไม่สามารถจ่ายกระแสออกมาได้เป็นผลให้ศักดาที่จุด 1 ไม่มีจึงทำให้ไดโอด D_2 สามารถนำกระแสได้ (Forward Bias) ซึ่งทำให้ระบบยังสามารถทำงานได้แม้จะเกิดไฟฟ้าดับก็ตาม



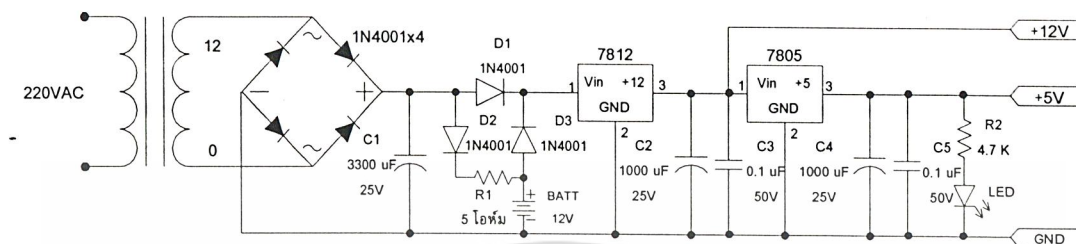
รูปที่ 3.14 หลักการของระบบไฟหลักและไฟสำรอง

ในการออกแบบวงจรแหล่งจ่ายไฟหลักและไฟสำรอง เลือกใช้ไดโอด IN4001 ชนิดซิลิกอน โดยมีจุดเบรคความถี่ 50 โวลต์ 1 แอมป์ สามารถอธิบายหลักการทำงานดังนี้

เมื่อเริ่มเปิดสวิตช์วงจรเริ่มทำงาน ไฟฟ้ากระแสสลับ 220 โวลต์เอซี ไหลผ่านวงจรบริดจ์เรกติฟายเออร์ทำการแปลงแรงดันไฟฟ้ากระแสสลับไปเป็นไฟฟ้ากระแสตรง และผ่าน C_1 ทำการกรอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แรงดันไฟฟ้ากระแสตรงให้เรียบยิ่งขึ้น ส่วนไดโอด D_1 ทำหน้าที่ให้กระแสไฟฟ้าไหลผ่านทางเดียวไม่ให้ไหลย้อนกลับโดยมีไดโอด D_2 กับตัวต้านทาน R_1 ทำหน้าที่เติมประจุให้กับแบตเตอรี่ที่ละน้อย ๆ ไดโอด D_1 นำกระแสเมื่อไฟฟ้ามาปกติ ทำให้ D_2 ไม่นำกระแส แบตเตอรี่จึงไม่สามารถจ่ายกระแสได้



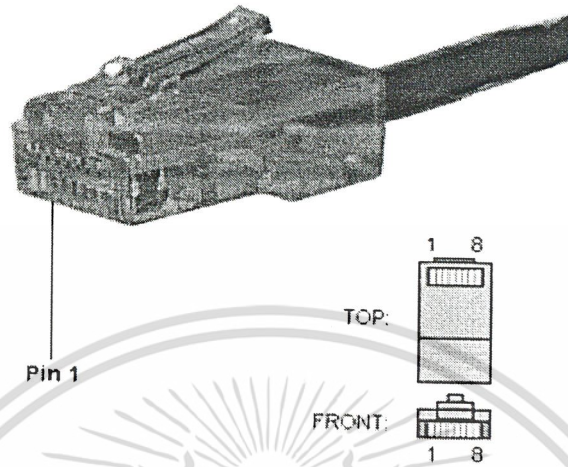
รูปที่ 3.15 วงจรแหล่งจ่ายไฟหลักและไฟสำรอง

โดยเอาต์พุตจะถูกกำหนดให้มีแรงดันคงที่ด้วยไอซีรีกยาระดับแรงดัน (Voltage Regulator) แรงดันไฟ 12 โวลต์ เลือกใช้ไอซีเบอร์ LM7812 และมี C_2 เป็นตัวกรองแรงดันให้เรียบยิ่งขึ้น ส่วน C_3 เป็นตัวบายพาส (By Pass) สัญญาณรบกวนความถี่สูงลงกราวด์ และในส่วนของไฟ 5V ใช้ไอซีเบอร์ LM7805 และมี C_4 เป็นตัวกรองแรงดันให้เรียบยิ่งขึ้น ส่วน C_5 เป็นตัว By Pass สัญญาณรบกวนความถี่สูงลงกราวด์ ส่วน R_2 ทำหน้าที่จำกัดกระแสที่จะ ไบอัสให้แก่หลอดแอลอีดี เพื่อเป็นตัวบอกสถานะว่าขณะนี้ไฟเลี้ยงวงจร ดังนั้นจะเห็นว่าเมื่อไฟฟ้าดับระบบรักษาความปลอดภัยสามารถทำงานได้ตามปกติ สามารถแสดงวงจรแหล่งจ่ายไฟหลักและไฟสำรองได้ดังรูปที่ 3.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.9 สาย UTP (Unshielded Twisted Pair)

ลักษณะของสายที่ใช้กับ RJ-45 เป็นแบบ 8 เส้น โดยตีเกลียวกัน 4 คู่ โดยไม่มีการชีลด์เรียกว่า UTP เป็น สายมาตรฐานที่ใช้กันในปัจจุบันคือสายแบบ CAT-5



รูปที่ 3.16 แสดงลักษณะของสาย UTP

การกำหนดหน้าที่ของสายย่อยภายในสายแบบ UTP	
หมายเลขสาย	หน้าที่ของสายสัญญาณ
1	Output Transmit Data +
2	Output Transmit Data -
3	Input Receive Data +
6	Input Receive Data -
4, 5, 7, 8	Reserved for other use

ตารางที่ 3.2 แสดงหน้าที่ของสาย UTP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเข้าหัวสายกับหัวต่อ มี 2 แบบ คือ

1. การต่อเชื่อมระหว่างคอมพิวเตอร์ กับ RCM 2200 เป็นการต่อเชื่อมแบบ Peer to Peer 2 เครื่อง ไม่ต้องใช้ HUB การต่อสายต้องมีการสลับสายนำสัญญาณทั้งสองด้านดังตาราง

Computer	สายที่	RCM2200
ขาวส้ม	1	ขาวเขียว
ส้ม	2	เขียว
ขาวเขียว	3	ขาวส้ม
น้ำเงิน	4	น้ำเงิน
ขาวน้ำเงิน	5	ขาวน้ำเงิน
เขียว	6	ส้ม
ขาวน้ำตาล	7	ขาวน้ำตาล
น้ำตาล	8	น้ำตาล

ตารางที่ 3.3 แสดงการต่อเชื่อมระหว่างคอมพิวเตอร์ กับ RCM 2200

2. การต่อเชื่อมระหว่างฮับ กับ RCM 2200 จะมีการเข้าสายกับหัว RJ-45 จะเหมือนกันทั้งสองข้าง

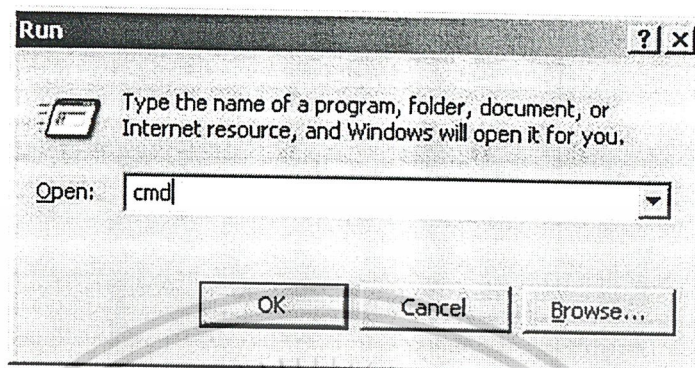
Hub	สายที่	RCM2200
ขาวส้ม	1	ขาวส้ม
ส้ม	2	ส้ม
ขาวเขียว	3	ขาวเขียว
น้ำเงิน	4	น้ำเงิน
ขาวน้ำเงิน	5	ขาวน้ำเงิน
เขียว	6	เขียว
ขาวน้ำตาล	7	ขาวน้ำตาล
น้ำตาล	8	น้ำตาล

ตารางที่ 3.4 แสดงการต่อเชื่อมระหว่างฮับ กับ RCM 2200

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

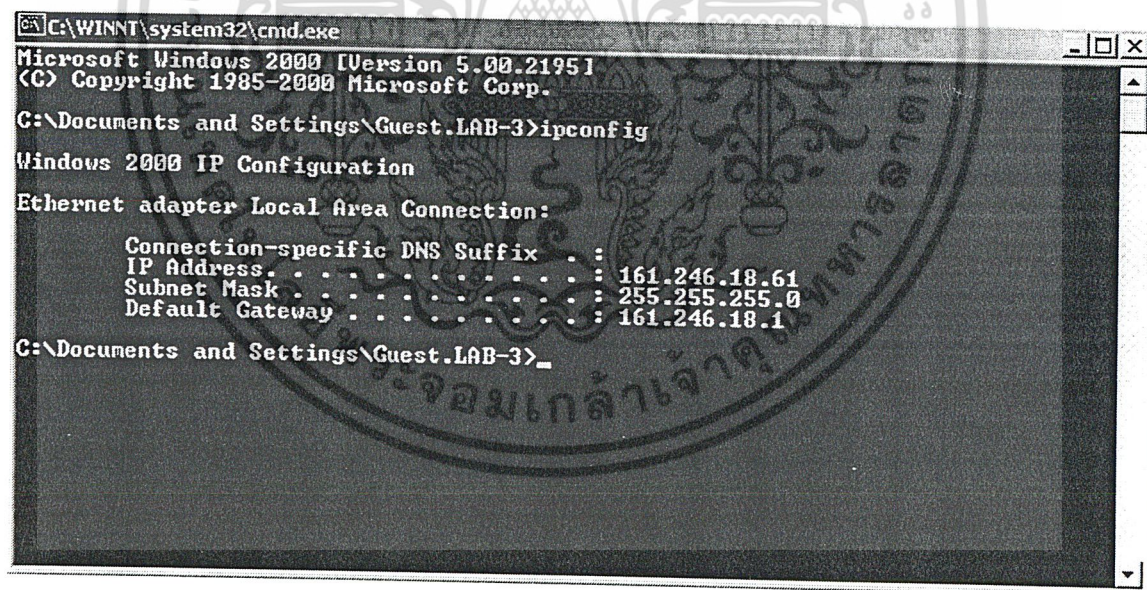
3.10 วิธีหา IP Address : Windows 2000/XP

1. ให้พิมพ์คำว่า "cmd" แล้วคลิกที่ปุ่ม [OK]



รูปที่ 3.17 แสดงการหาค่า IP

2. จากนั้นให้พิมพ์คำว่า "ipconfig" ณ ตำแหน่งที่ cursor กระทบอยู่แล้วกด Enter จะปรากฏเลข IP Address ของเครื่องดังกล่าว

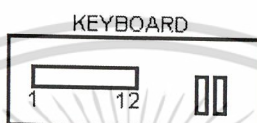


รูปที่ 3.18 แสดงค่า IP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.11 หลักการอินเตอร์เฟซมือถือ

ในที่นี้จะกล่าวถึงแต่เฉพาะโทรศัพท์มือถือยี่ห้อซีเมนส์รุ่น C35 ในส่วนของการติดต่อกันระหว่างโทรศัพท์มือถือกับไมโครคอนโทรลเลอร์นั้น เราใช้สายส่งข้อมูล (DATA LINK) ในการส่งข้อมูลระหว่างกัน โดยสัญญาณของโทรศัพท์มือถือและไมโครคอนโทรลเลอร์นั้น เป็นสัญญาณระดับ TTL ทั้งคู่ ดังนั้นจึงไม่ต้องทำการแปลงสัญญาณใดๆทั้งสิ้น ในการใช้งานจริงเราจะใช้แค่ 3 ขา คือ ขา 1, 5 และ 6 ของพอร์ตข้อมูลของโทรศัพท์มือถือซีเมนส์รุ่น C35 ดังตาราง รูปแบบข้อมูลของโทรศัพท์มือถือซีเมนส์รุ่น C35 ส่งข้อมูลในลักษณะ 19200-N-8-1 คือมีอัตราการส่งข้อมูล 19200 บิตต่อวินาที ไม่มีพาริตีบิต, ข้อมูลมีขนาด 8 บิต และมีบิตหยุด 1 บิต ตัวอย่างรหัสคำสั่งที่ใช้มีดังนี้



Pin	Name	Function	in/out
1	GND	Ground	
2	SELF-SERVICE	Recognition/control battery charger	in/out
3	LOAD	Charging voltage	in
4	BATTERY	Battery	out
5	DATA OUT	Data sent	out
6	DATA IN	Data received	in
7	Z_CLK	Recognition/control accessories	
8	Z_DATA	Recognition/control accessories	
9	MICG	Ground for microphone	in
10	MIC	Microphone input	
11	AUD	Loudspeaker	out
12	AUDG	Ground for external speaker	

ตารางที่ 3.5 พอร์ตของมือถือ C35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.12 ขั้นตอนการออกแบบ Web Server

ขั้นตอนการออกแบบ Web Server บน RCM2200 Core module เพื่อใช้ในโครงการ จะต้องมีการกำหนดค่าต่างในโปรแกรมดังนี้

3.11.1 ไฟล์ HTML โดยมีการประกาศตัวแปรในไฟล์ HTML เพื่อรับค่าตัวแปรจากโปรแกรมหลัก โดยมีรูปแบบการประกาศตัวแปรในไฟล์ HTML ดังนี้

```
<html>
  ">
-
  </html>
```

จากการกำหนดค่าข้างต้น จะทำให้ไฟล์ HTML สามารถปรับเปลี่ยนรูปภาพ เพื่อใช้ในการแสดงผลของ เซ็นเซอร์ โดยใช้ไฟล์ภาพในการแสดงสถานะแทน โดยรูปแบบในการแสดงผลจะมีดังนี้

3.11.2 โยงค่าตัวแปรโปรแกรมภายในโปรแกรมหลัก ให้บราวเซอร์สามารถใส่ค่าตัวแปรเดียวกับในโปรแกรมหลักได้ โดยมีรูปแบบการโยงค่าตัวแปรโดยใช้ฟังก์ชัน HTTPSPEC_VARIABLE โดยใช้รูปแบบคำสั่งดังนี้

```
{ HTTPSPEC_VARIABLE, "door1", 0, door1, PTR16, "%s", NULL},
```

หลังจากมีการเรียกใช้งานฟังก์ชัน HTTPSPEC_VARIABLE โปรแกรมหลักจะสามารถทำการส่งค่าตัวแปร door1 ไปยัง HTTP SERVER ได้

3.11.3 ทำการสร้าง link ในไฟล์ HTML เพื่อให้ไฟล์ HTML สามารถทำการเรียกฟังก์ชัน CGI ที่เราเขียนรอเอาไว้ในโปรแกรมหลักได้ โดยมีรูปแบบการเขียน Code HTML ดังนี้

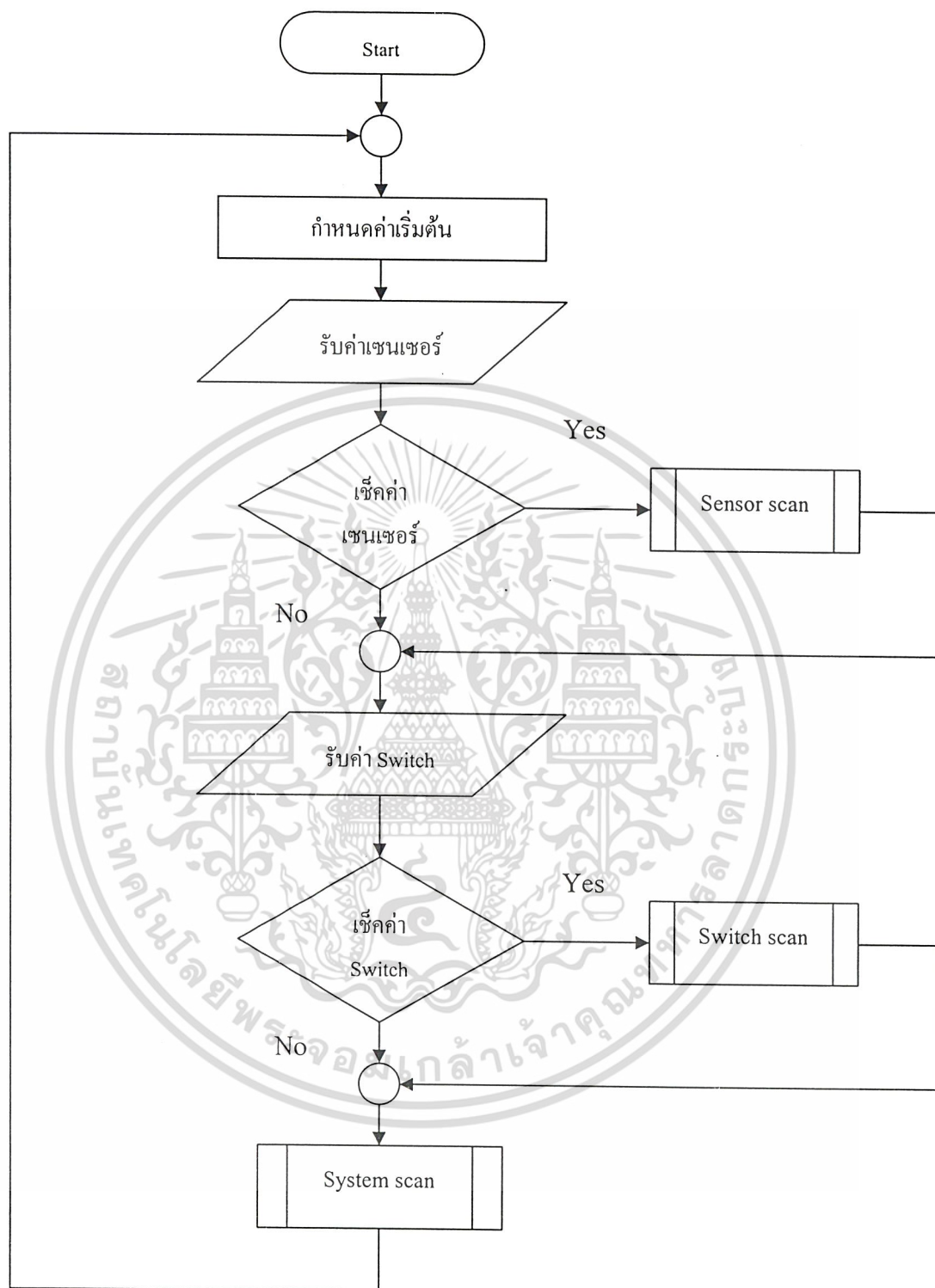
```
<html>
<input name="button1" type="button" onClick="parent.location='on1.cgi'" value="Open/Close ">
</html>
```

3.11.4 ทำการเรียกฟังก์ชัน SSPEC_FILE เพื่อระบุไฟล์ต่างๆที่ต้องการใช้บน HTTP SERVER โดยมีรูปแบบการใช้งานดังนี้

```
{ HTTPSPEC_FILE, "/index.shtml", index_shtml, NULL, 0, NULL, &admin},
{ HTTPSPEC_FILE, "/ledon.gif", ledon_gif, NULL, 0, NULL, NULL},
{ HTTPSPEC_FILE, "/ledoff.gif", ledoff_gif, NULL, 0, NULL, NULL},
```

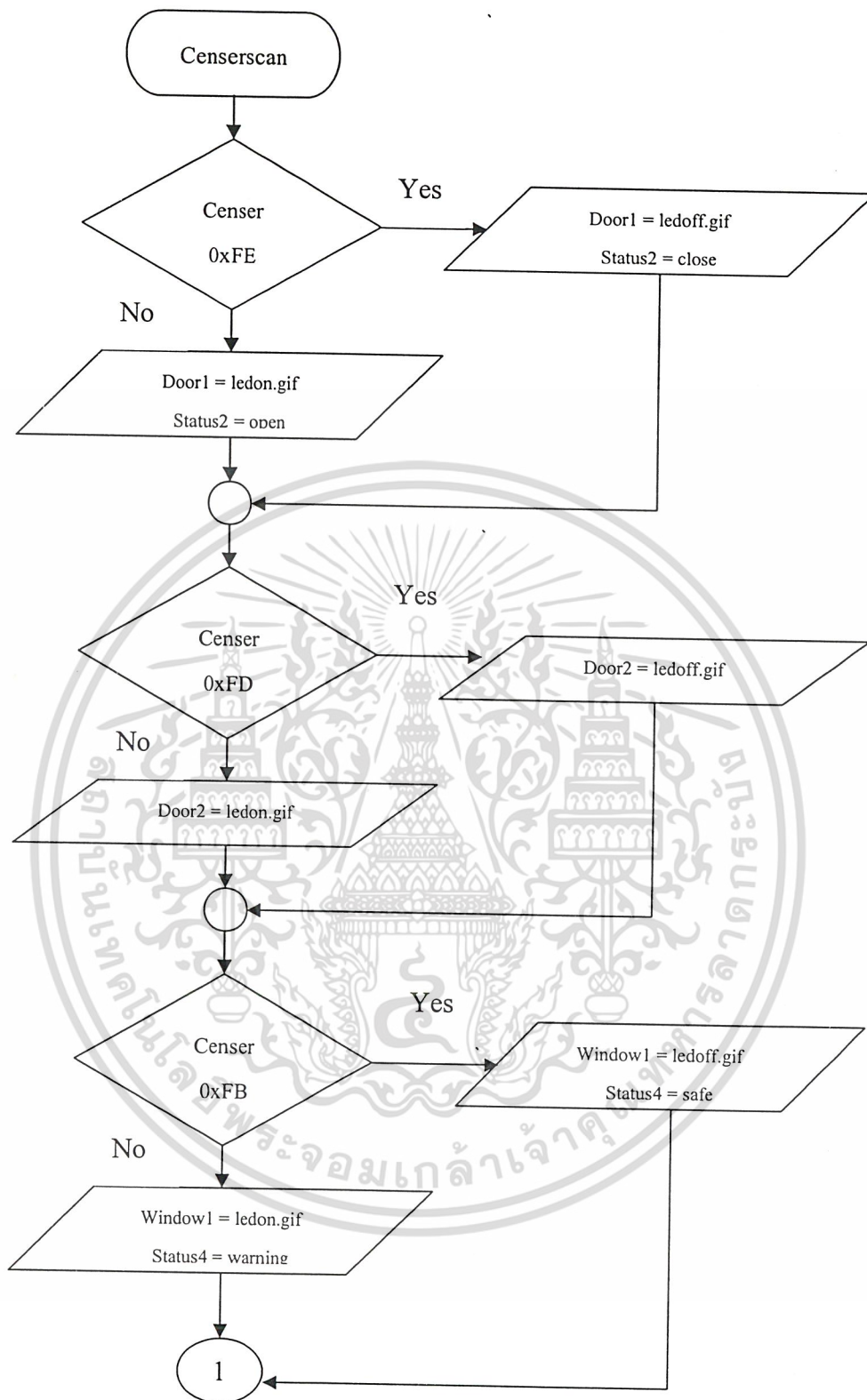
หลังจากมีการเรียกใช้ฟังก์ชัน จะทำให้ HTTP SERVER สามารถระบุไฟล์ที่มีอยู่ในระบบได้อย่างถูกต้อง

3.13 โฟลว์ชาร์ตการทำงานแต่ละส่วน



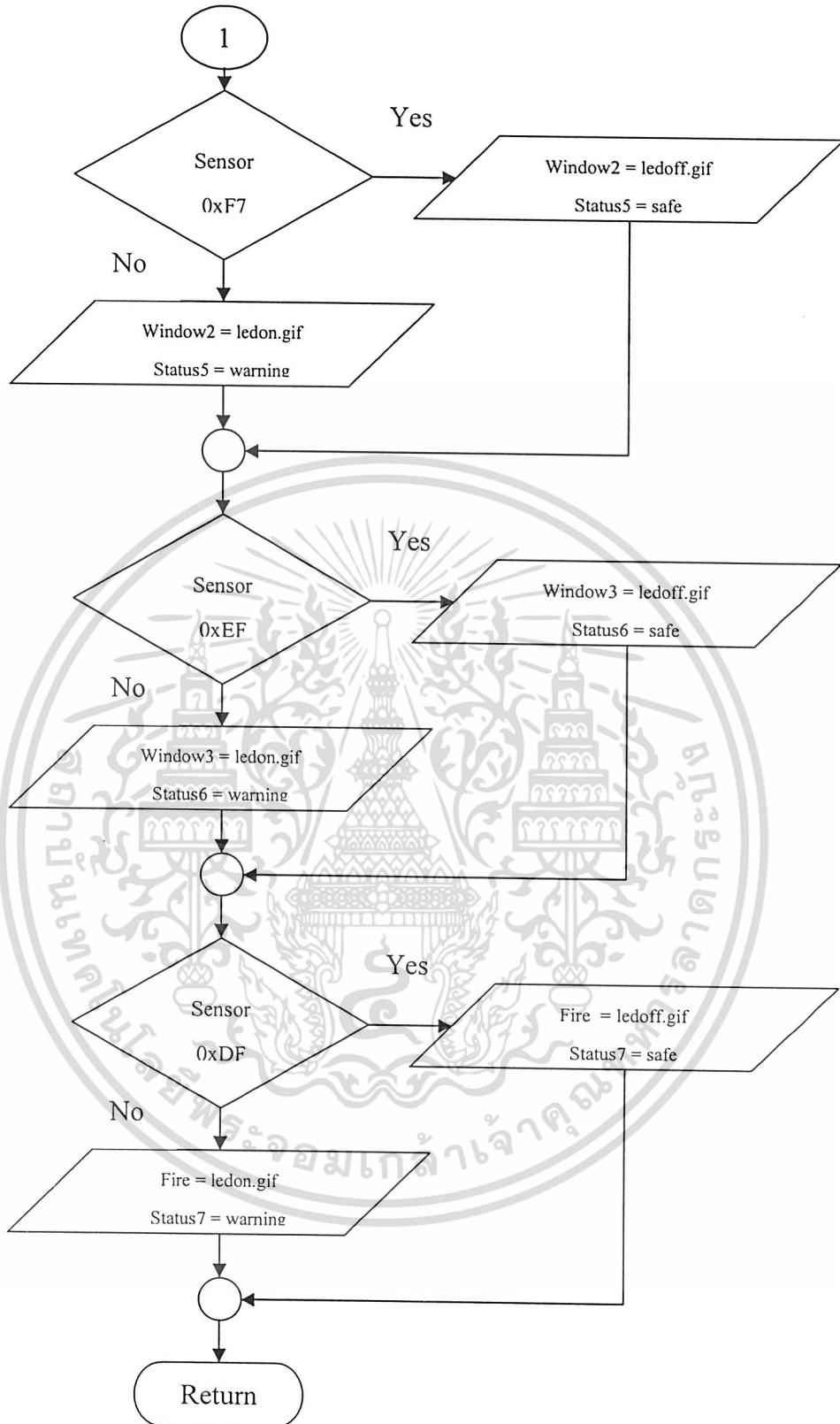
รูปที่ 3.19 โฟลว์ชาร์ตการทำงานของโปรแกรมหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



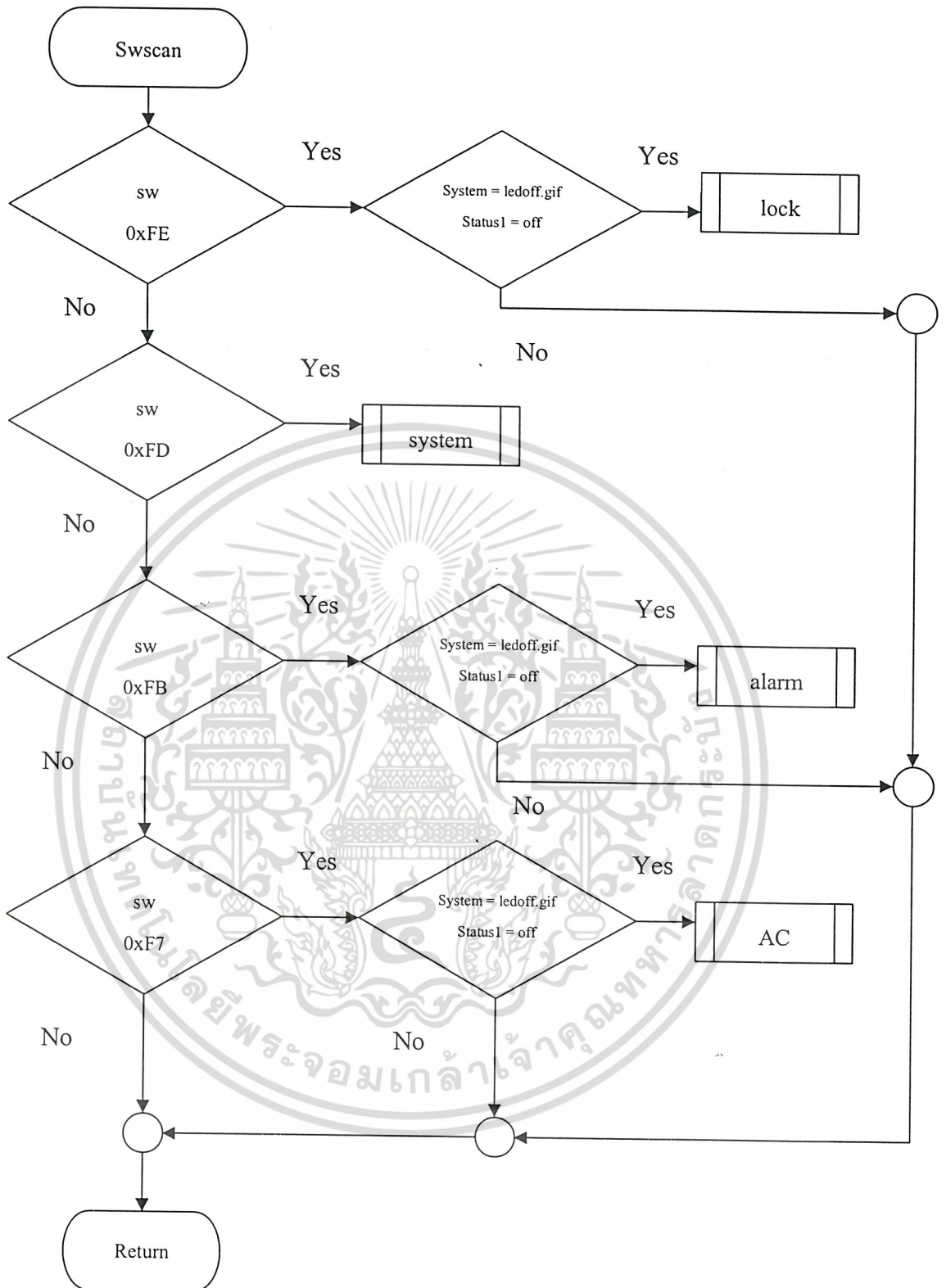
รูปที่ 3.20 โฟลว์ชาร์ตระบบแสดงผลของ Senser ผ่าน Web Browser

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



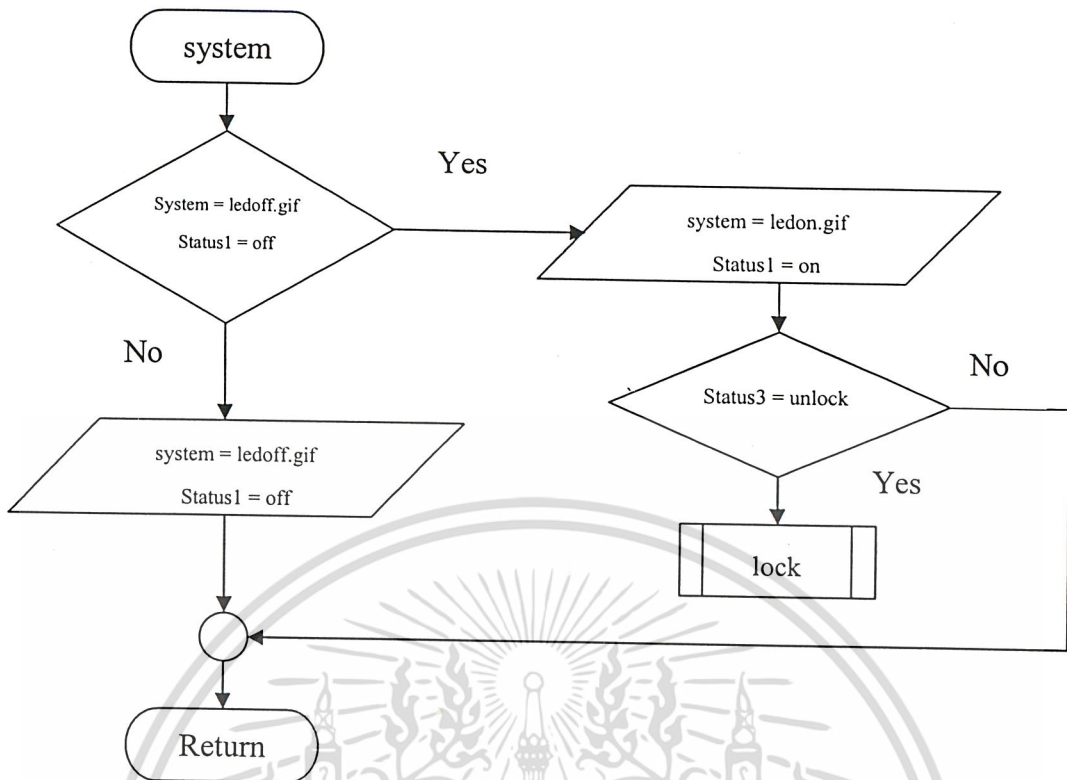
รูปที่ 3.20 (ต่อ) โพล์ชาร์ตระบบแสดงผลของ Sensor ผ่าน Web Browser

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

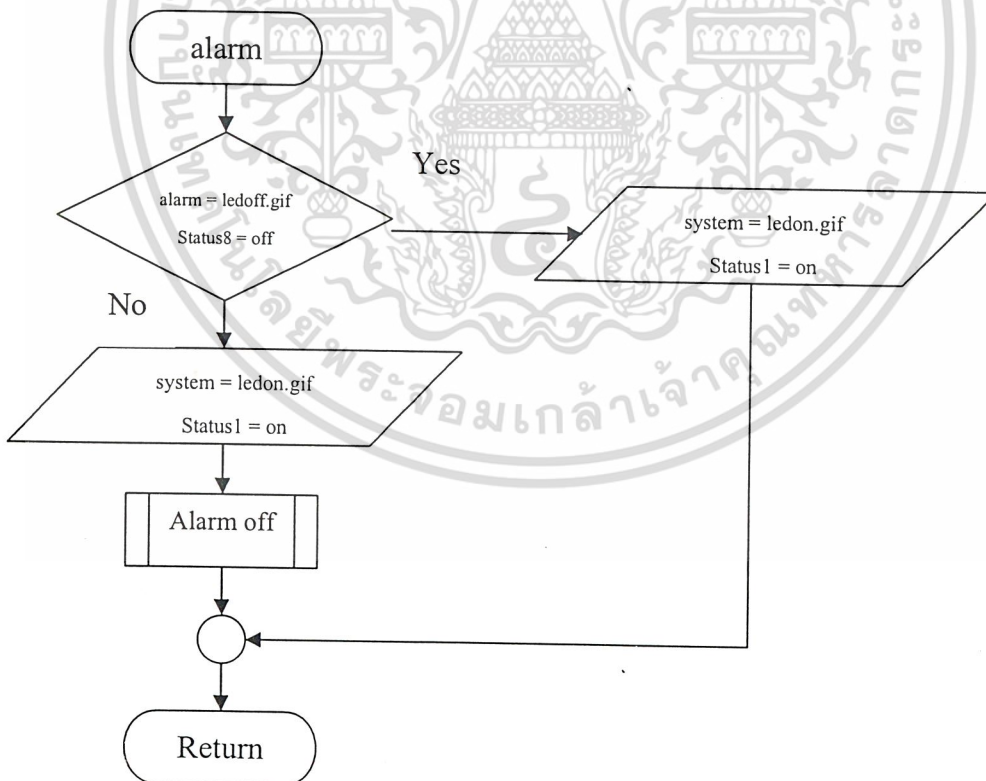


รูปที่ 3.21 โฟลว์ชาร์ตการทำงานของ Manual Switch

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

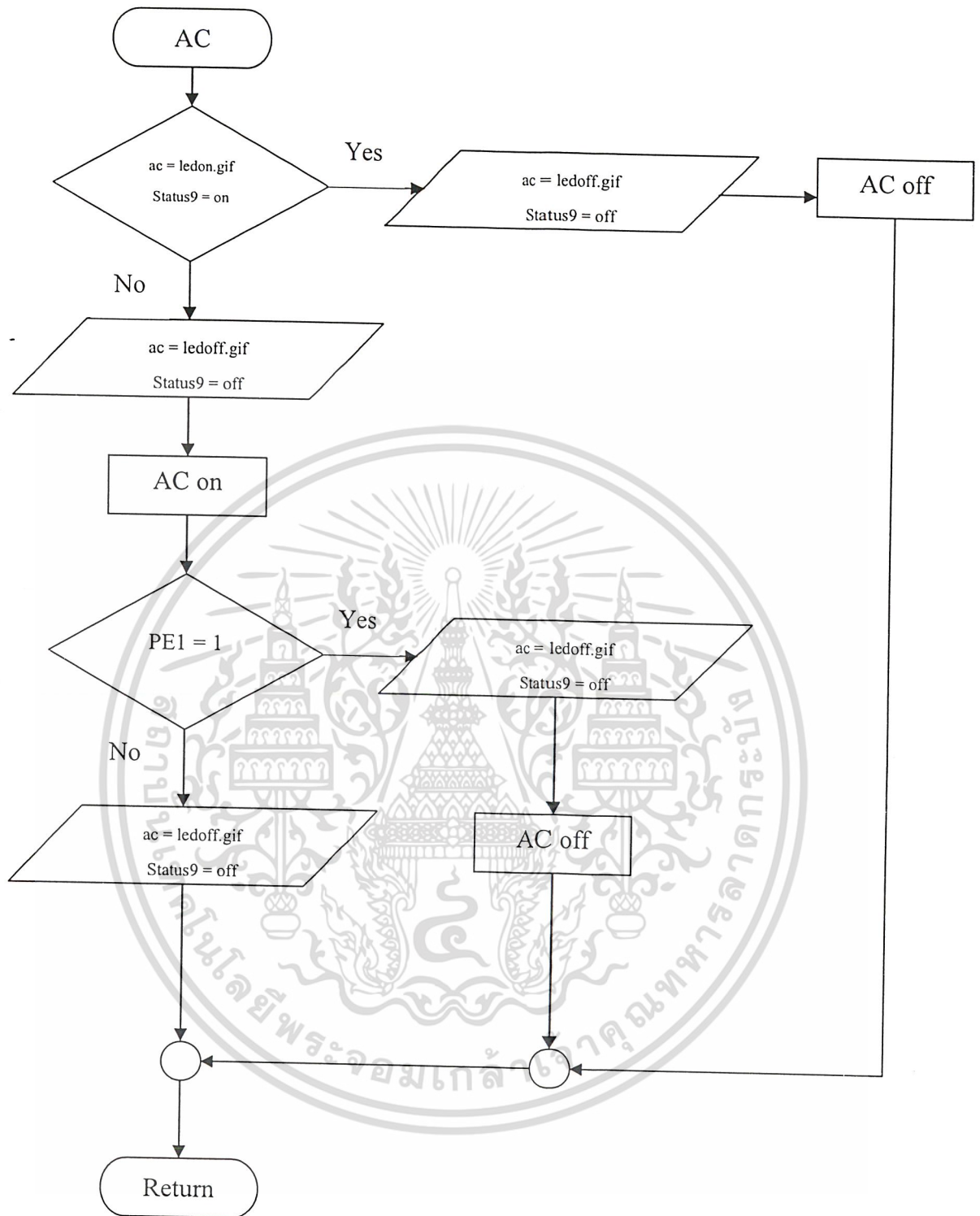


รูปที่ 3.22 โฟลว์ชาร์ตการทำงานของระบบ System ในสถานะ ON/OFF



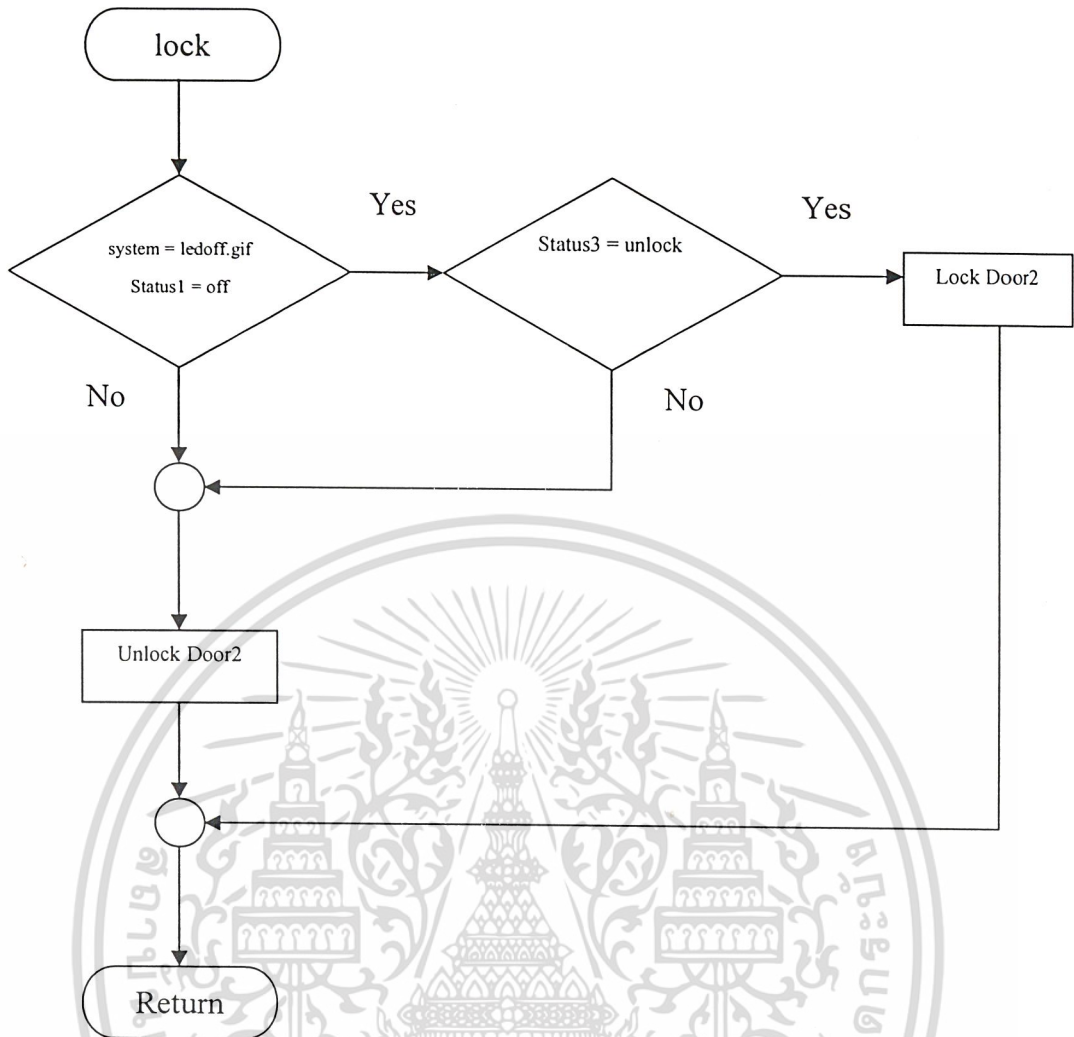
รูปที่ 3.23 โฟลว์ชาร์ตการทำงานของระบบ Alarm ในสถานะ ON/OFF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



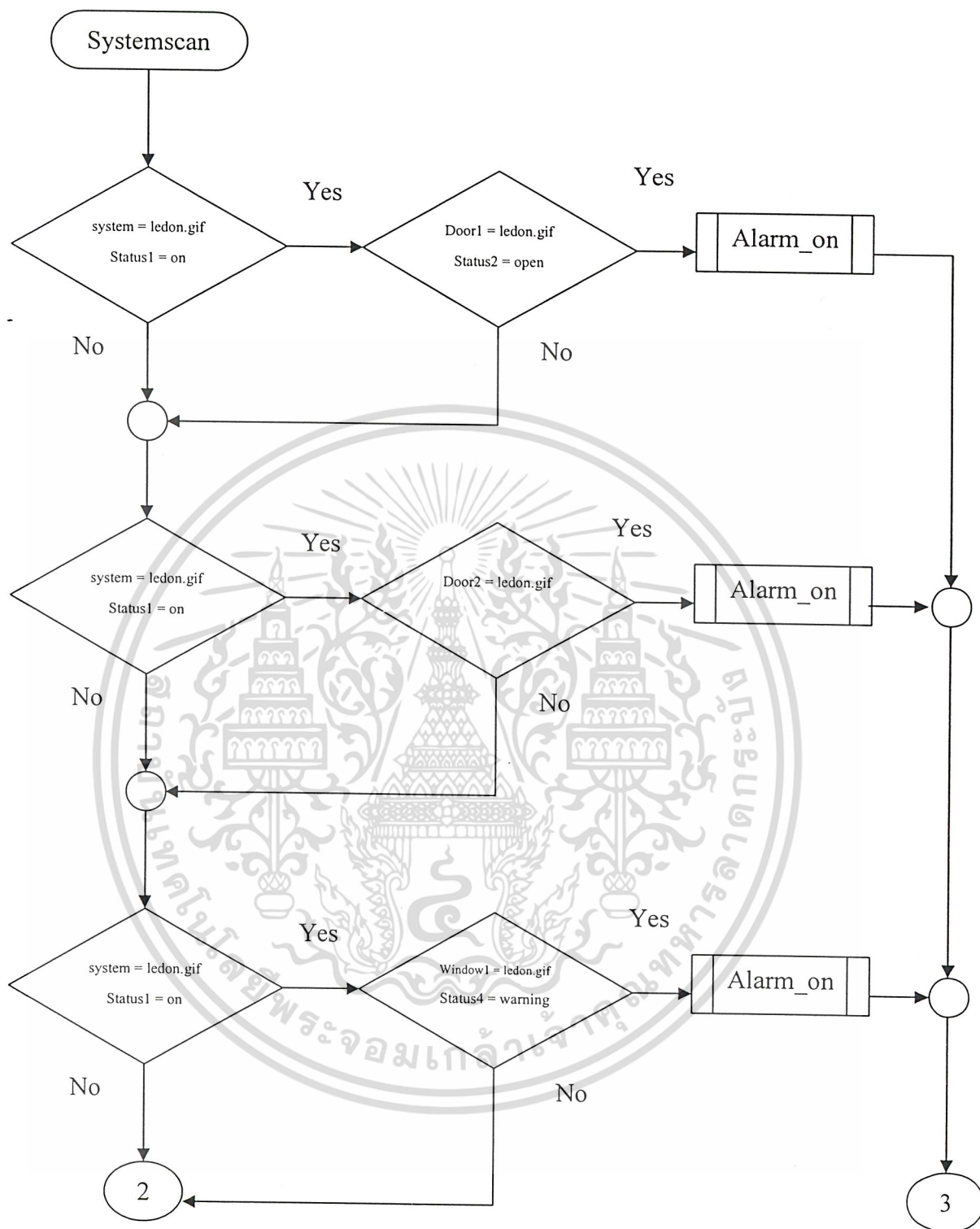
รูปที่ 3.24 โฟลว์ชาร์ตการทำงานของระบบ AC ในสถานะ ON/OFF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



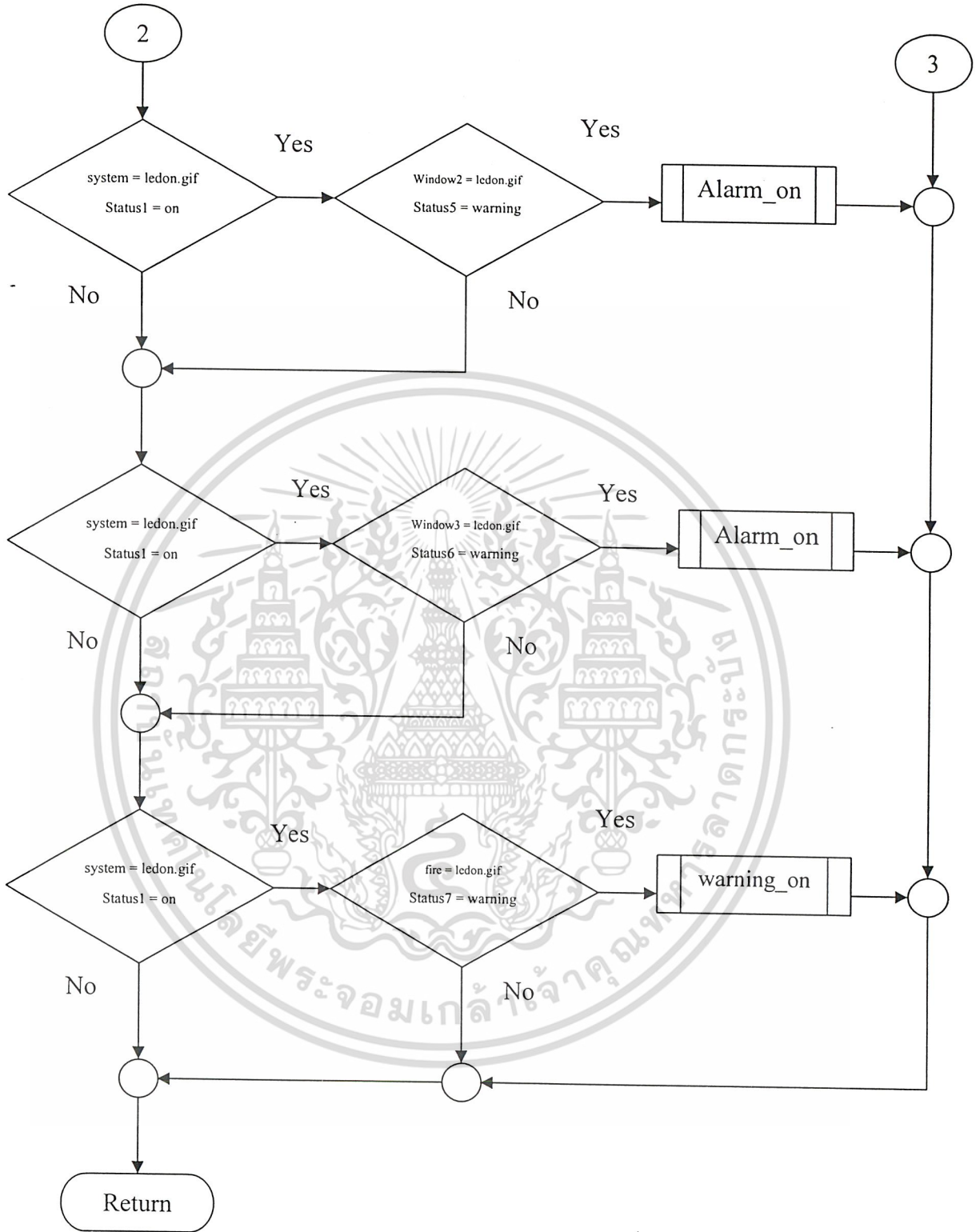
รูปที่ 3.25 โฟลว์ชาร์ตการทำงานของระบบ Lock ในสถานะ ON / OFF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



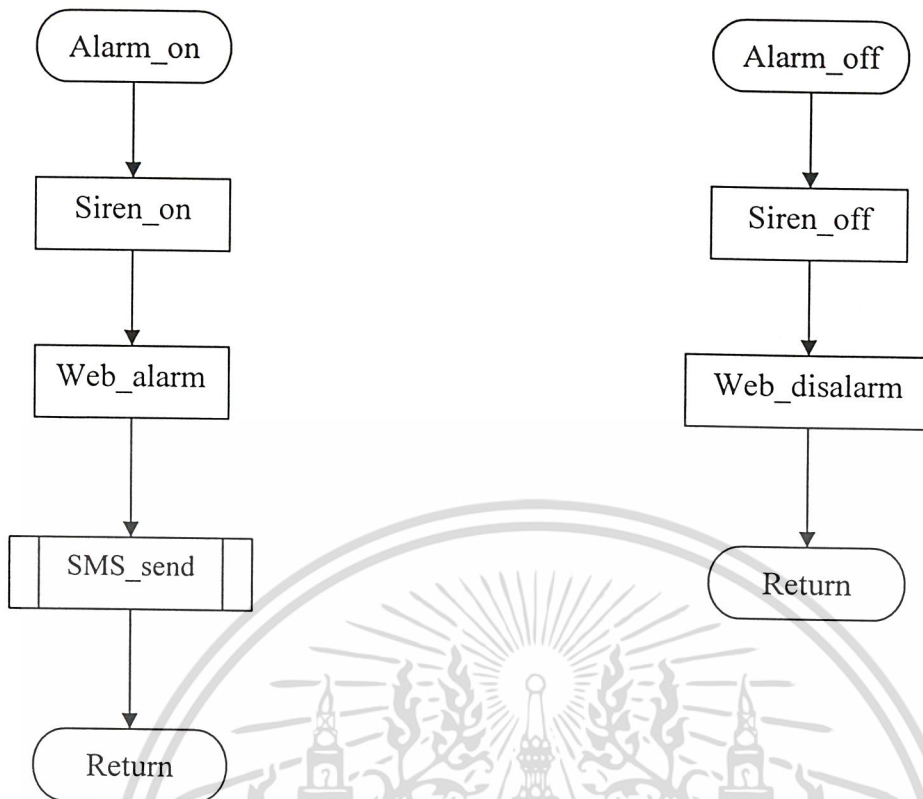
รูปที่ 3.26 โพล์ซาร์ตตรวจสอบระบบ System Scan

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



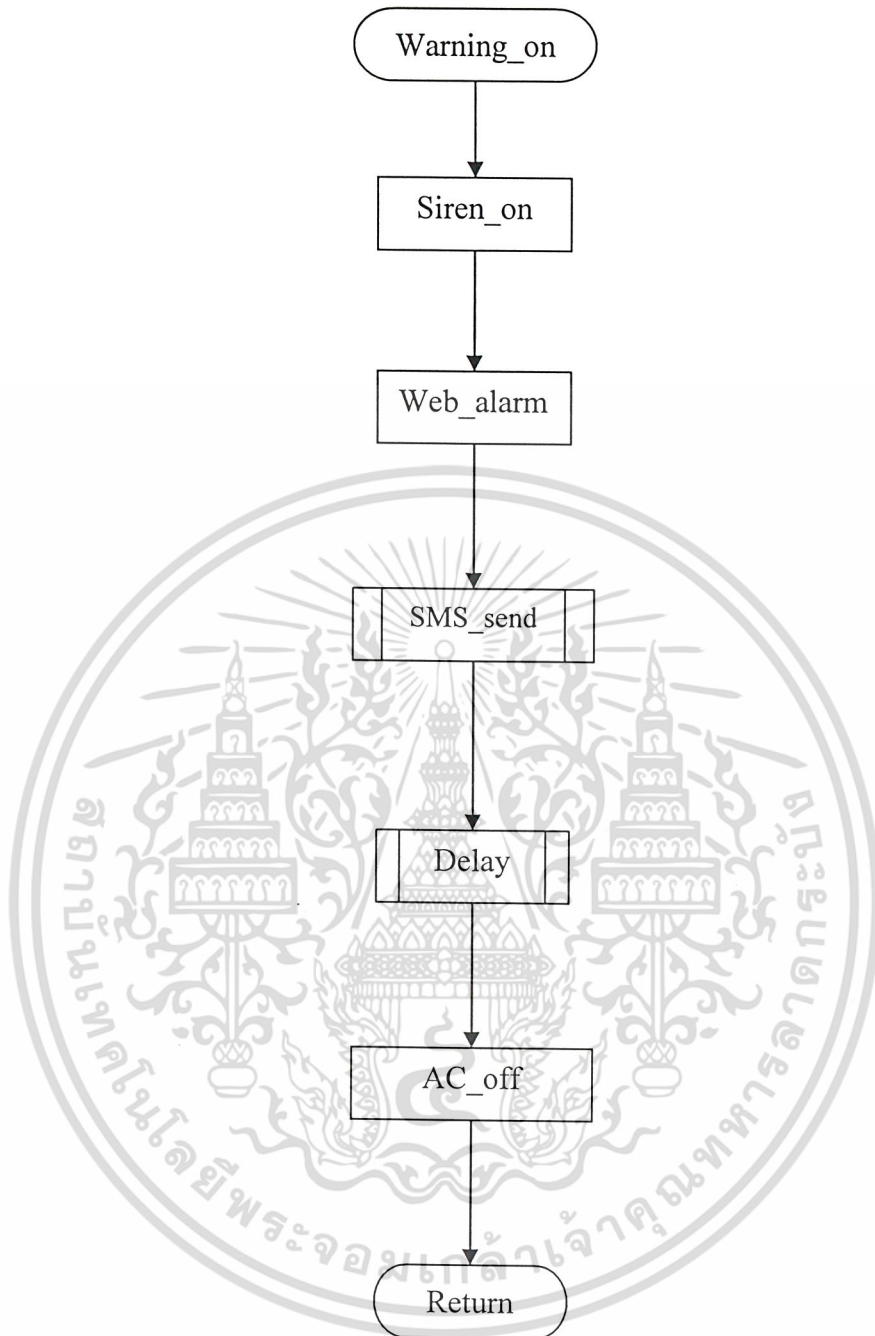
รูปที่ 3.26 (ต่อ) โฟลว์ชาร์ตตรวจสอบระบบ System Scan

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



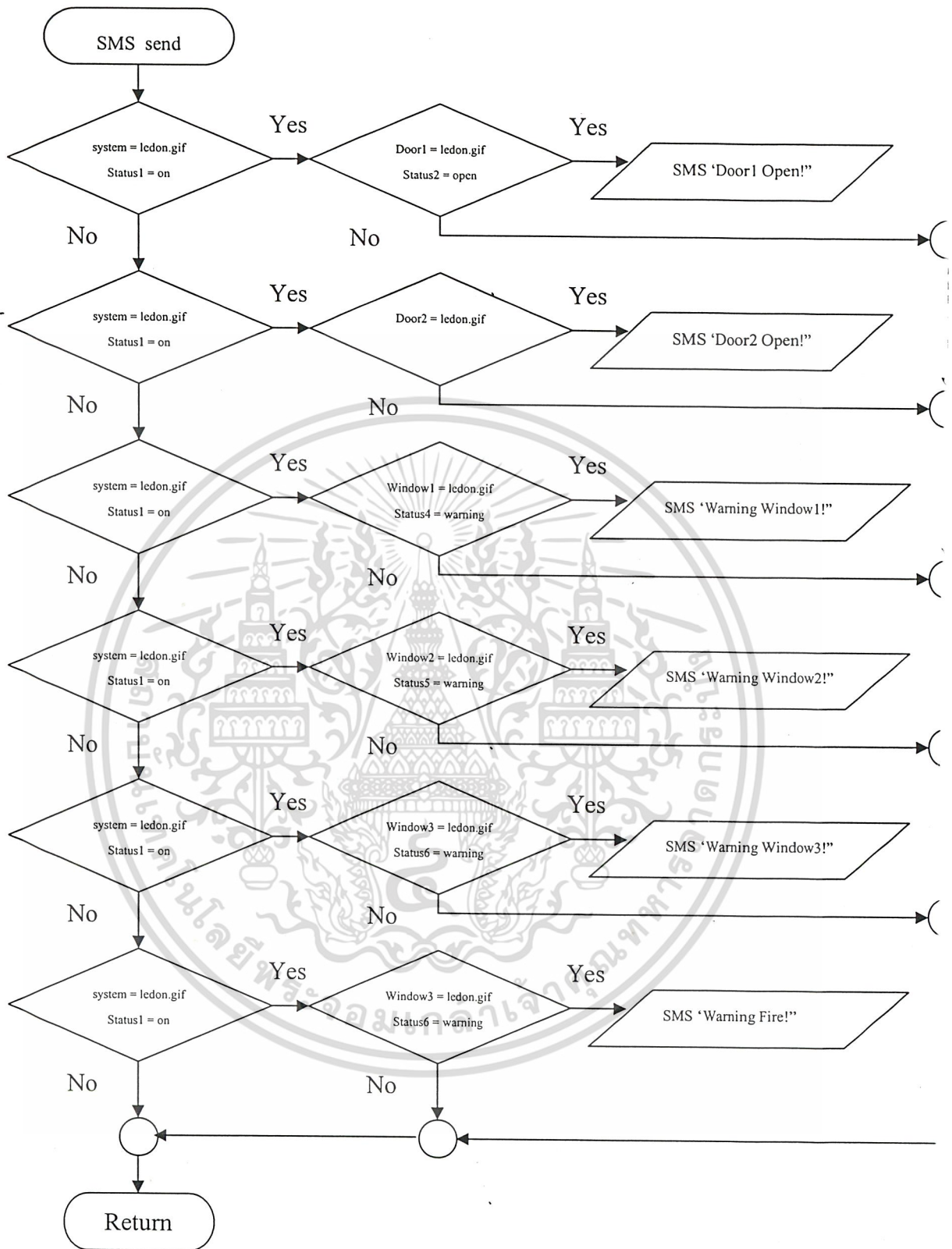
รูปที่ 3.27 โฟลว์ชาร์ตเปิด และปิดระบบเตือนภัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.28 โฟลว์ชาร์ตเปิด และปิดระบบเตือนภัยอัคคีภัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.29 โฟลว์ชาร์ตส่งข้อความสั้น เข้าโทรศัพท์เคลื่อนที่ เมื่อมีเหตุผิดปกติ

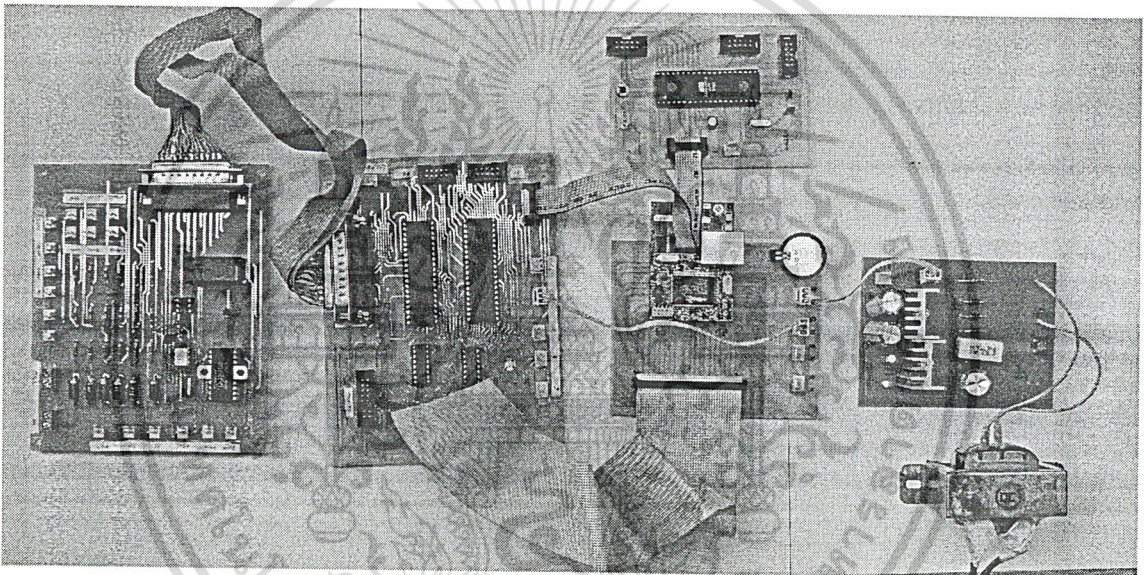
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลอง และผลการทดลอง

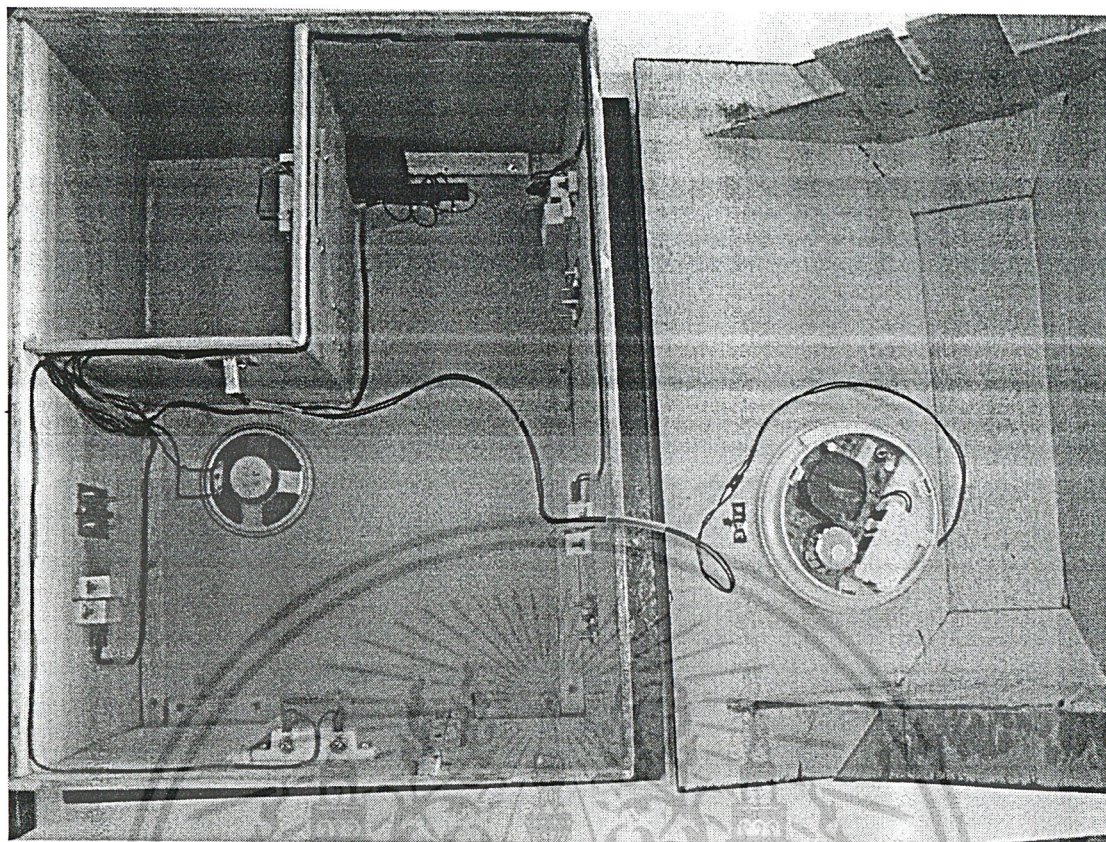
4.1 อุปกรณ์ที่ใช้ในการทดลอง

- เครื่องคอมพิวเตอร์ ทำหน้าที่เป็น Client เพื่อใช้ควบคุมและแสดงผลการทำงาน
- วงจร RCM 2200 ทำหน้าที่รับข้อมูลการสั่งงานจากเครื่องคอมพิวเตอร์ และส่งข้อมูล
- วงจรควบคุม ทำหน้าที่ขยายพอร์ท, รับค่าเซนเซอร์ และสั่งงานควบคุมมอเตอร์ และ Siren
- ชุดวงจรเซนเซอร์
- วงจรสื่อสารกับโทรศัพท์เคลื่อนที่ พร้อมกับโทรศัพท์เคลื่อนที่รุ่น ซีอีเอ็มเอสรุ่น C35

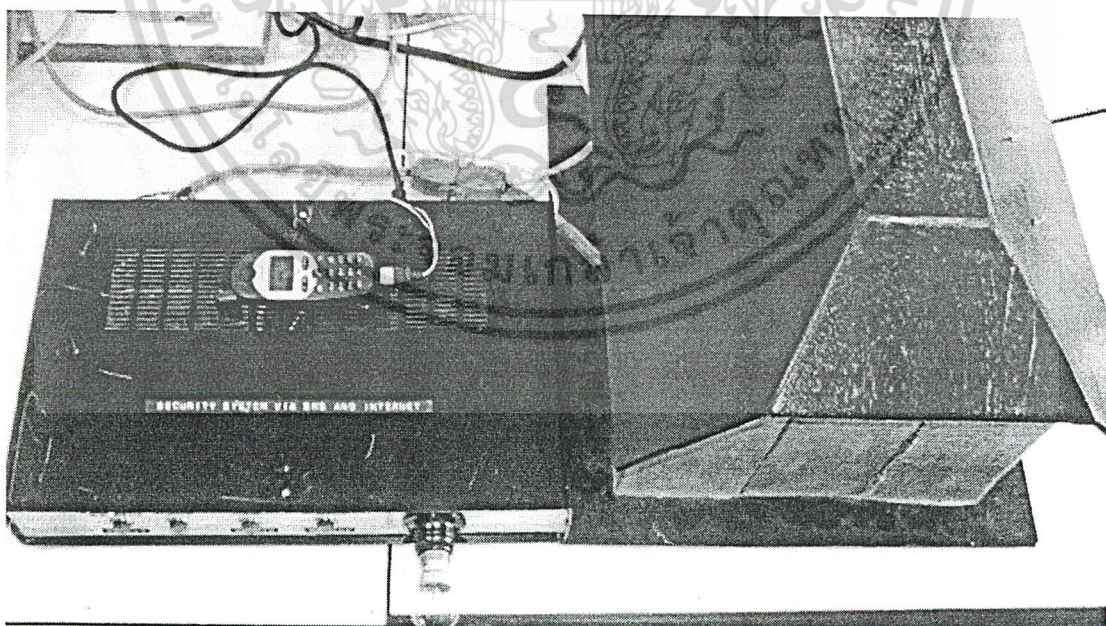


รูปที่ 4.1 วงจรภายในของระบบเตือนภัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 การติดตั้งระบบเซนเซอร์ และมอเตอร์



รูปที่ 4.3 ชุดระบบเตือนภัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทดลองการรับค่าจากชุด Manual Switch

ในโครงการนี้ จะทำการกำหนดการกดปุ่มสวิตช์เพื่อส่งค่าไปที่ IC 8255 ซึ่งจะกำหนดสถานะเป็นลอจิก “0” เมื่อมีการกดปุ่ม และมีสถานะเป็นลอจิก “1” เมื่อไม่มีการกดปุ่ม โดยจะมีการกำหนดค่ารับค่าที่ PB ของ IC 8255 ดังนี้

สวิตช์	ค่าที่ได้	ทำหน้าที่
SW 1	FEH	ล๊อค / ปลดล๊อค ประตูที่ 2
SW 2	FDH	เปิด / ปิด ระบบ
SW 3	FBH	เปิด / ปิด ไชเรน
SW 4	F7H	เปิด / ปิด ระบบไฟฟ้า

ตารางที่ 4.1 แสดงค่าที่รับเข้ามาทาง PB ของ IC 8255

เมื่อทำการกดปุ่มแล้วทำการวัดค่า แรงดันที่ ขา PB7 – PB4 เมื่อกดปุ่มจะวัดแรงดันได้ 0 V และเมื่อไม่กดปุ่มจะวัดค่าได้ 5V

ค่าที่กำหนดจากตาราง จะนำค่าโค้ดการทำงานไปใช้ในการเขียนโปรแกรม

4.3 การทดลองการส่งค่า

ในการส่งค่าของ RCM 2200 จะทำการส่งค่าไปที่ PA0 และ PA1 ของ RCM 2200 เพื่อทำการเปิดปิด ไชเรน และระบบไฟฟ้าตามลำดับ

ในการส่งค่าเพื่อควบคุมมอเตอร์ และส่งงานการส่งข้อความสั้นไปที่ MCS-51 จะใช้ PA1 และ PA2 ของ IC 8255 ตัวที่ 1 และ 2 ตามลำดับ

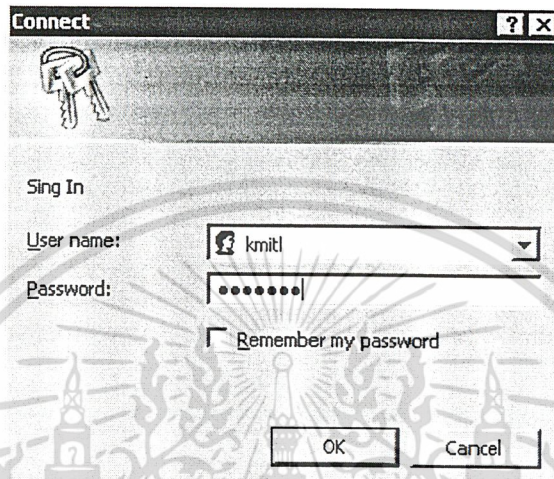
ในการทดลองการส่งค่าจะใช้ LED เพื่อรับค่าและแสดงผลค่าที่รับเข้ามา โดย LED จะติดสว่างเมื่อมีแหล่งจ่ายเข้ามา เช่น ป้อนค่า 0xFF มาจาก IC 8255 จะทำให้ LED ติดสว่างทั้ง 8 ดวง แสดงว่า RCM 2200 จะทำการส่งค่าให้ IC 8255 ได้

4.4 การทดลองการทำงานของ Web Browser

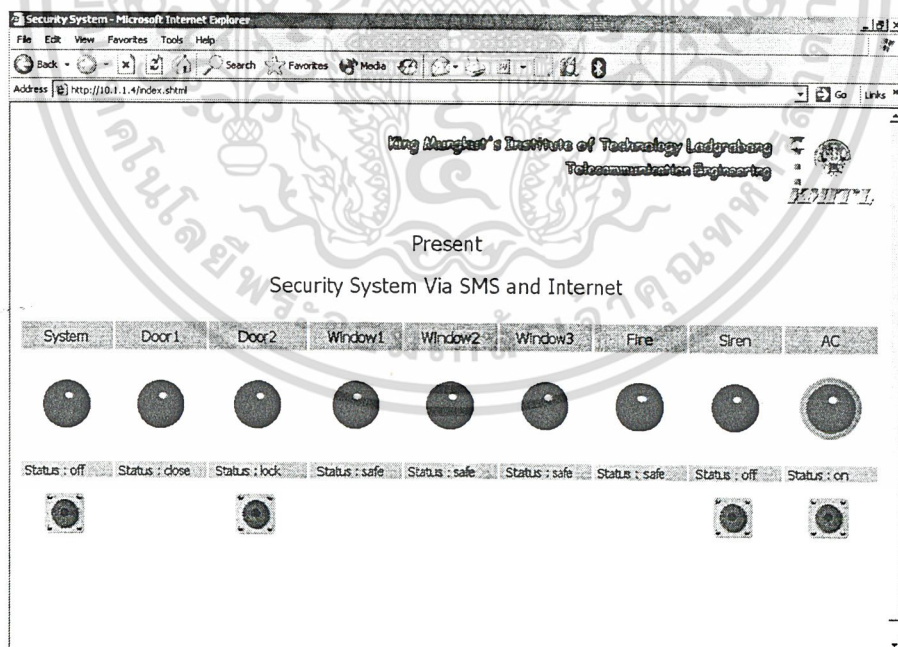
การทดลองนี้จะทำการทดสอบการสื่อสารการทำงานระหว่าง คอมพิวเตอร์, RCM 2200, วงจรควบคุม และ Manual Switch ซึ่งการทดสอบการทำงานนี้จะแสดงผ่าน Web Browser ดังรูปที่ 4.5 โดยที่เครื่อง คอมพิวเตอร์จะใช้หน้า Web Browser ในการส่งข้อมูล และรับข้อมูล เพื่อควบคุมการทำงานของ RCM 2200 และส่งข้อมูลไปควบคุมที่วงจรควบคุม หรือทำการตรวจสอบสถานะเซนเซอร์ที่จะแสดงผลบน Web Browser เพื่อแสดงผลการเปิดหรือปิดของประตู หรือ หน้าต่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการป้อนค่า IP ที่เป็นของ RCM 2200 เพื่อเข้าสู่หน้าจอ Web Page โดยจะต้องใส่ User Name และ Password ให้ถูกต้องจึงจะเข้าไปสู่หน้าจอที่ใช้ในการสั่งงาน เพื่อป้องกันผู้ที่ไม่ได้รับอนุญาตเข้าไปใช้งาน ซึ่ง User Name ที่ตั้งไว้คือ “kmitl” และ Password คือ “telecom” เมื่อใส่ถูกต้องโปรแกรมจึงจะยอมให้ผ่านเข้าไปหน้า Web Page แต่ถ้าใส่ไม่ถูกต้องก็ไม่สามารถใช้งานได้ โปรแกรมจะนำค่าที่ใส่ส่งไปยัง RCM 2200 ซึ่งเป็น Server แล้วนำค่าไปเช็กกับค่าที่ตั้งไว้ว่าตรงกันหรือไม่



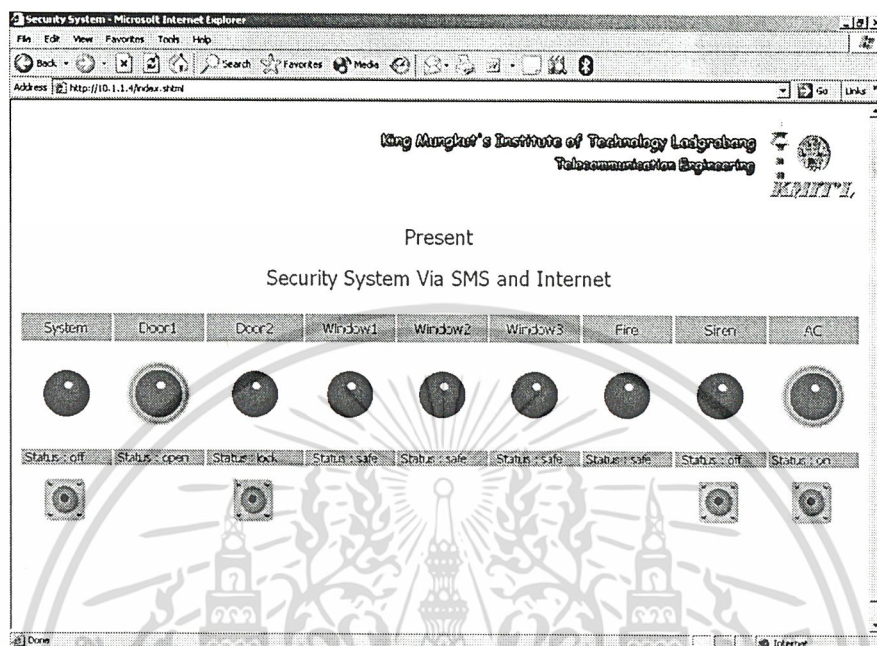
รูปที่ 4.4 แสดงหน้า Web Browser เพื่อการเข้าใช้ระบบ



รูปที่ 4.5 ผลบนหน้า Web Browser ในกรณีปกติ

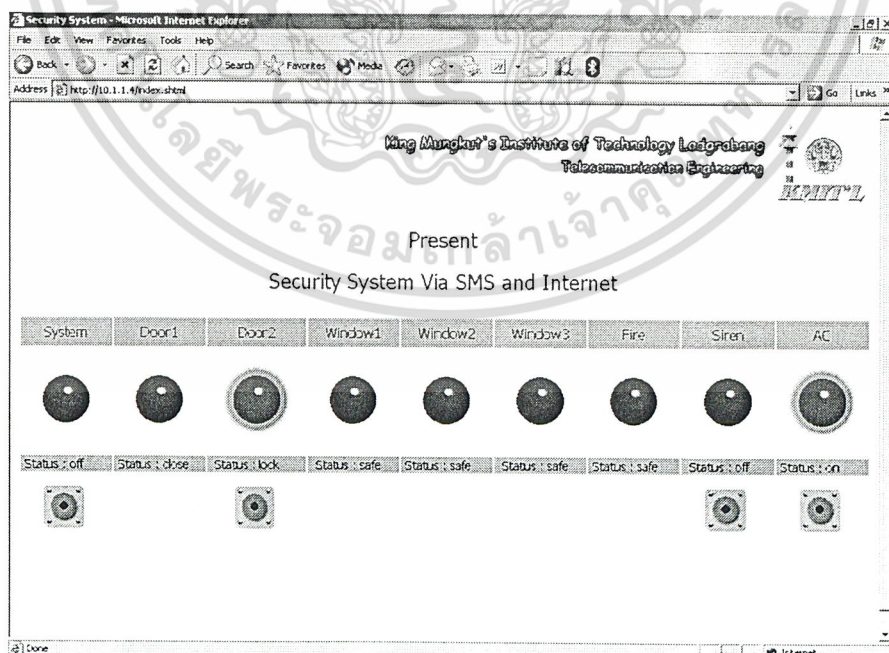
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดสอบการเปิด หรือปิดประตู จะมีระบบเช็คการเปิดปิดของประตู โดยใช้ Magnetic Switch เพื่อตรวจสอบว่าประตูเปิด หรือปิดจริงหรือไม่ โดยในโปรแกรมจะให้สถานะของประตูปิดเป็นลอจิก “0” และให้สถานะของประตูเปิดเป็นลอจิก “1”



รูปที่ 4.6 ผลบนหน้า Web Browser ในกรณีที่ทำการเปิดประตูที่ 1

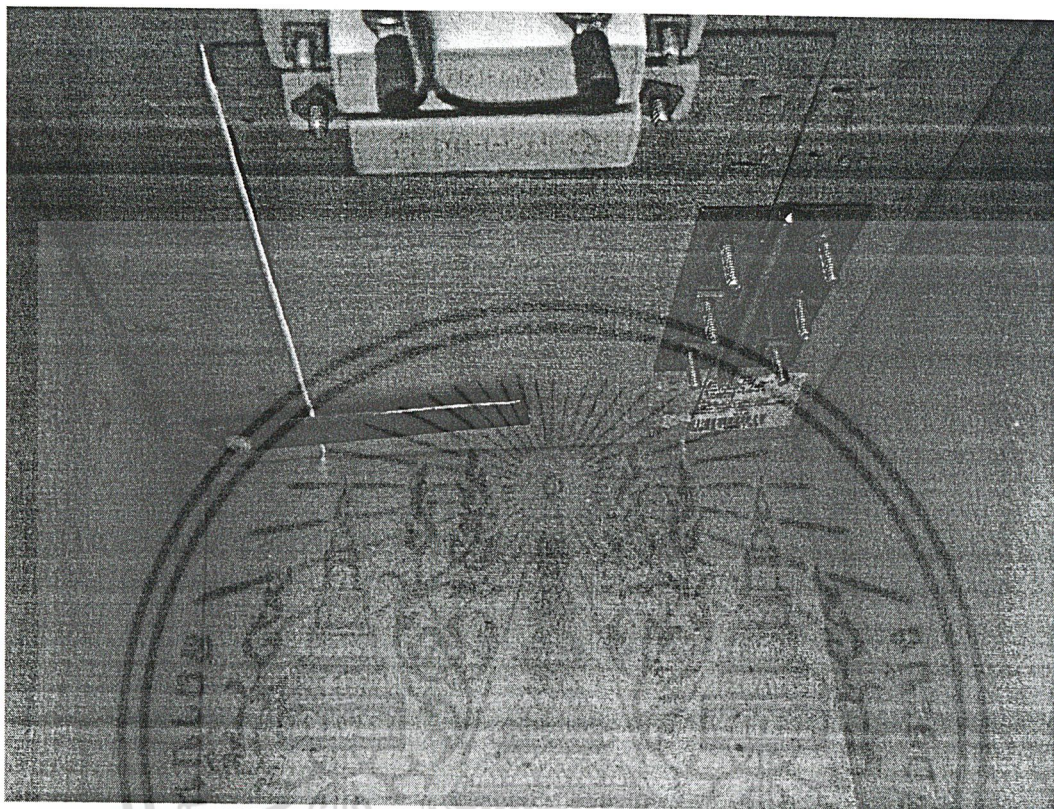
จากรูปที่ 4.8 จะเป็นการรับค่ามาจาก IC 8255 เพื่อทำการแสดงสถานะ ผ่านหน้า Web Page



รูปที่ 4.7 ผลบนหน้า Web Browser ในกรณีที่ทำการกดปุ่มล๊อค ประตูที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

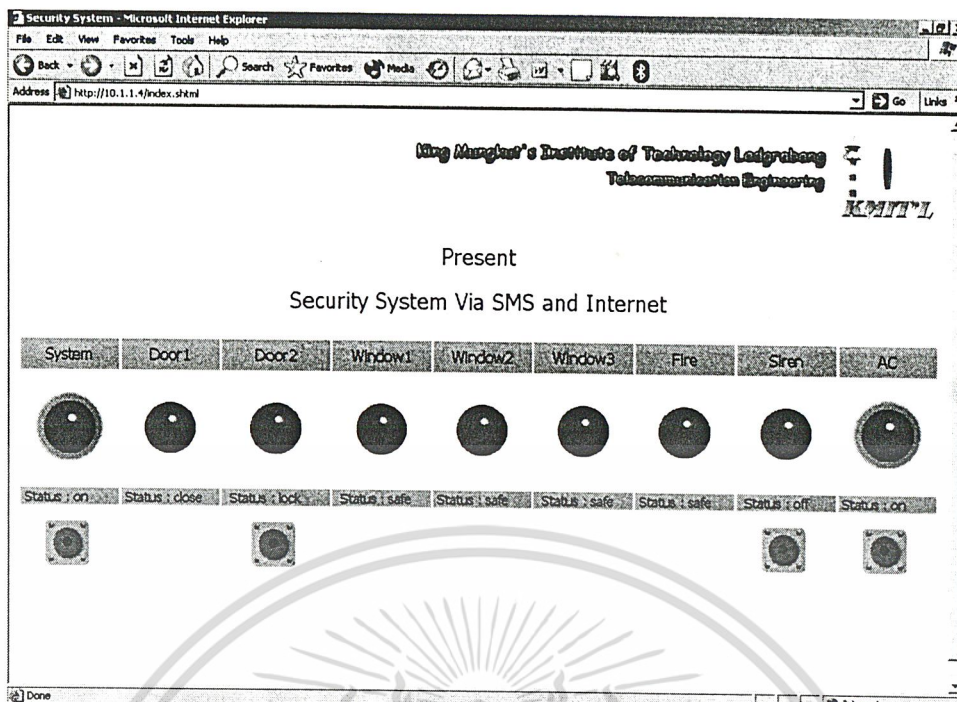
จากรูปที่ 4.9 สั้งงานโดยการกดปุ่ม Lock /Unlock ประะตูที่ 2 จะทำการลือกคั้ทที่ ถ้าประะตูที่ 2 ปิดอยู่ ถ้าประะตูที่ 2 เป็คอยู่ก็จะไม่สามารถสั้งงานได้ ในการแสดงผลผ่าน Web Browser จะ Refresh ตัวเองหลังจากที่ทำการกดปุ่มไปแล้ว



รูปที่ 4.8 แสดงผลการสั้งงานการลือกประะตูที่ 2

ในกรณีที่ต้องการเป็คระบบการทำงานของระบบเต็อนกั้ย สามารถทำได้โดยการกดปุ่ม On /Off ที่ System แล้วสถานะของระบบจะแสดงผ่านทางรูปภาพ และช่อง Status คือ On ดังรูปที่ 4.9 จะสั้งเกตได้ว่าเมื่อกั้กดปุ่มเป็คระบบเต็อนกั้ย จะทำการ ลือกประะตูที่ 2 จากด้านใน โดยอัตโนมัติ แล้วแสดงสถานะของประะตูที่ 2 ทางช่อง Status เป็นค่า Lock นั่นก็คือระบบพร้อมที่จะทำการเต็อนกั้ย

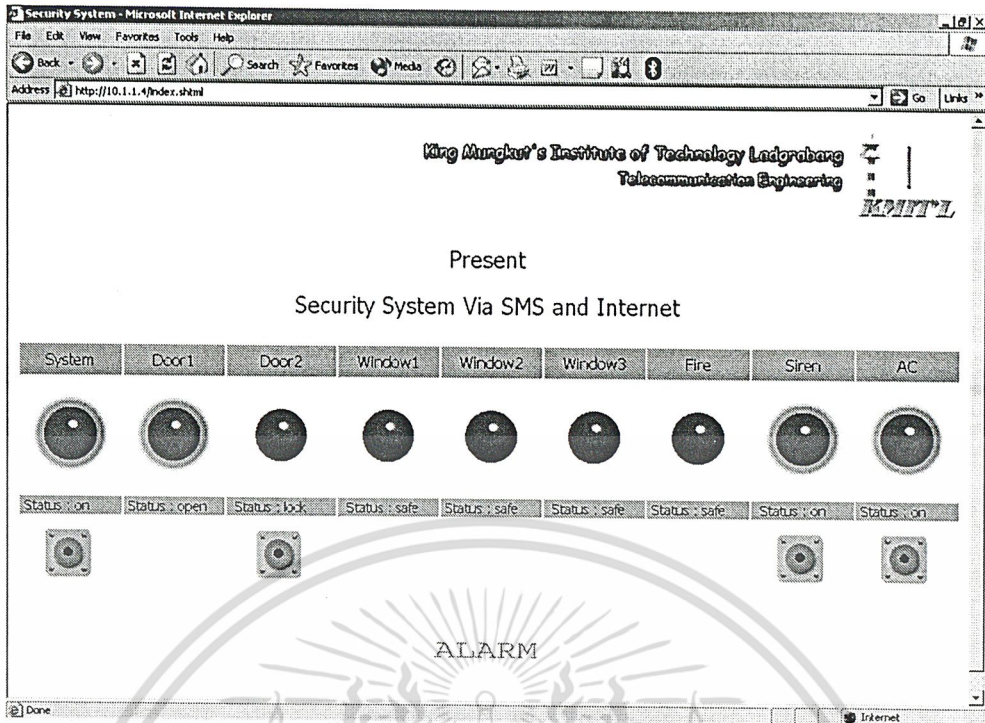
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 ผลบน Web Browser เมื่อทำการเปิดระบบเตือนภัย

เมื่อเกิดเหตุผิดปกติขึ้น เช่น มีผู้บุกรุก หรือ เกิดควันขึ้น โดยทำให้สถานะของเซนเซอร์รับค่าได้มาเปลี่ยนไป ระบบก็จะทำการตรวจสอบเช็คสถานะต่างๆ จากนั้นก็จะแสดงข้อความ ALARM ออกทาง Web Browser แล้วไซเรนที่เราติดตั้งไว้ก็จะส่งเสียงเตือนออกมา พร้อมกับทำการส่งข้อความ SMS เช่น “Door1 Open!” และไม่ว่าจะทำการปิดประตูแล้ว หรือ กดปุ่มปิดเสียงไซเรน แต่เสียงจากไซเรนก็ยังคงทำงานตลอด จนกว่าจะทำการปิดระบบเตือนภัย โดยการกดปุ่ม On / Off ที่ System ให้ Status แสดงค่า “Off” แล้วก่อนทำการกดปุ่ม ปิดระบบ ไซเรน ก็จะหยุดทำงานทันที ดังรูปที่ 4.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



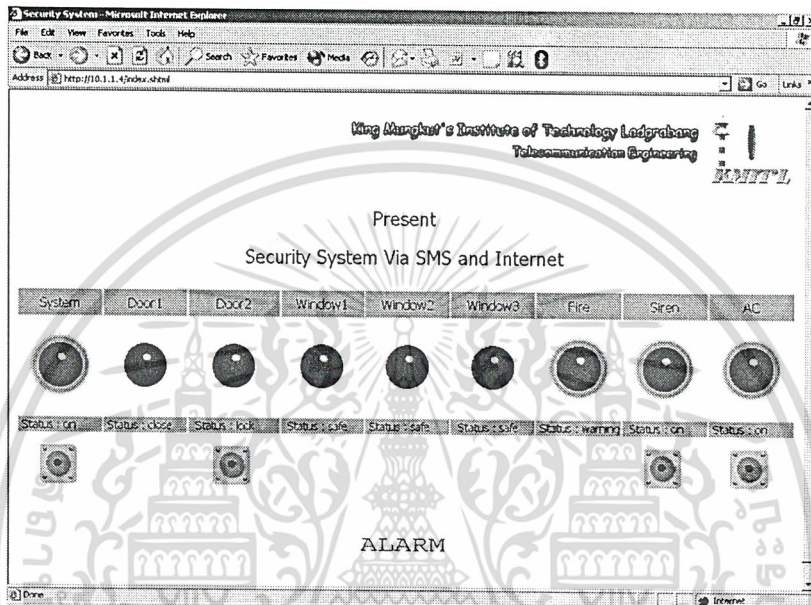
รูปที่ 4.10 ผลบน Web Browser แจ้งเตือนเมื่อมีผู้บุกรุก เข้ามาที่ประตูที่ 1



รูปที่ 4.11 ข้อความ“Door1 Open!” เตือนภัยที่โทรศัพท์ปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเกิดควันขึ้น จะทำให้สภาวะของเซนเซอร์รับค่าได้มาเปลี่ยนไป ระบบก็จะทำการตรวจสอบ เซ็คสถานะต่างๆ จากนั้นก็จะแสดงข้อความ ALARM ออกทาง Web Browser แล้วไซเรนที่เราติดตั้งไว้ก็จะส่งเสียงเตือนออกมา พร้อมกับทำการส่งข้อความ SMS เช่น “Warning Fire!” แล้วระบบจะทำการ หน่วงเวลาเพื่อแจ้งเตือนภัยด้วยเสียงจากไซเรน และจะทำการตัดไฟฟ้าภายในบ้าน เพื่อเข้าสู่ระบบ ดับเพลิง แต่ระบบเตือนภัย ยังคงทำงานได้จากแหล่งจ่ายไฟฟ้าสำรอง ดังรูปที่ 4.14



รูปที่ 4.12 ผลบน Web Browser แจ้งเตือนเมื่อมีควัน

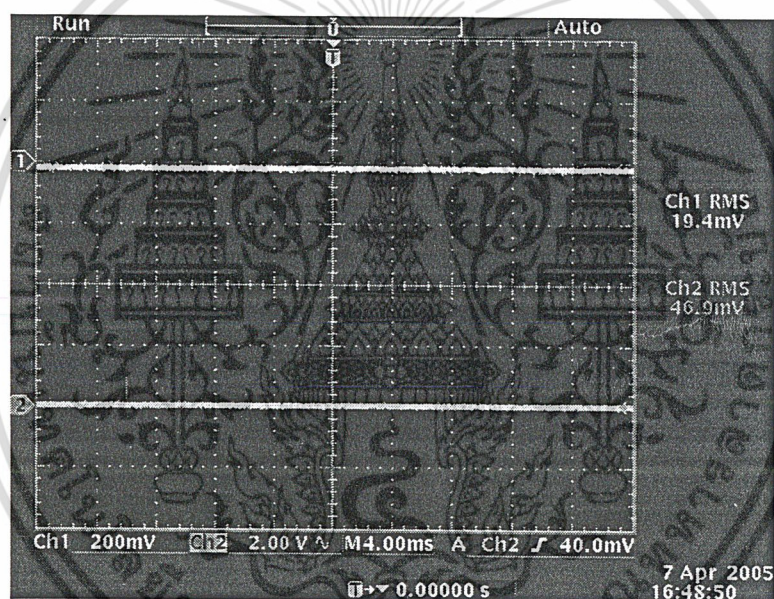


รูปที่ 4.13 ข้อความ “Warning Fire!” เตือนภัยที่โทรศัพท์ปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

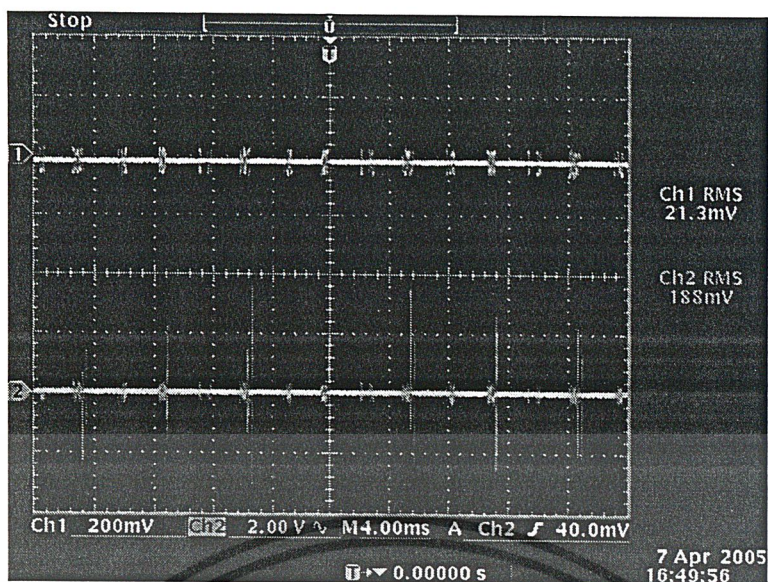
จากการทดลองจะเห็นได้ว่าการใช้งาน RCM2200 มีความสามารถในการรับค่าผ่านเข้ามาใน Port C โดยการรับค่าจะรับเข้ามาเป็น Byte โดยเราสามารถนำค่าที่ได้มาแล้วนำมาใช้ในการตรวจสอบสถานะของ Magnetic Switch ได้ในการนำ RCM2200 มาใช้งานจริงกับอุปกรณ์ต่าง ๆ โดยใช้ในการรับค่าเมื่อได้สัญญาณเข้ามา จะนำมาประมวลผลว่ามี Magnetic Switch ตัวใดเกิดการเปลี่ยนแปลงสถานะเป็น ON หรือ OFF จากนั้น RCM2200 จะส่งค่าสถานะของ Magnetic Switch ไปยัง Web Browser โดยมีความสามารถในการแสดงผลเป็นรูปภาพได้ โดยที่ Web Browser จะทำการ Refresh ตัวเองภายในเวลาที่กำหนดเพื่อทำการ Update ข้อมูลสถานะของ Magnetic Switch ให้เป็นข้อมูลปัจจุบัน แต่การแสดงผลของ Web Browser จะมีข้อเสียคือ ไม่สามารถแสดงข้อมูลให้เป็นแบบ Real Time ได้ จึงต้องมีการ Refresh ตัวเองอยู่เสมอถึงจะมีการ Update ข้อมูลได้

4.5 วัดสัญญาณ



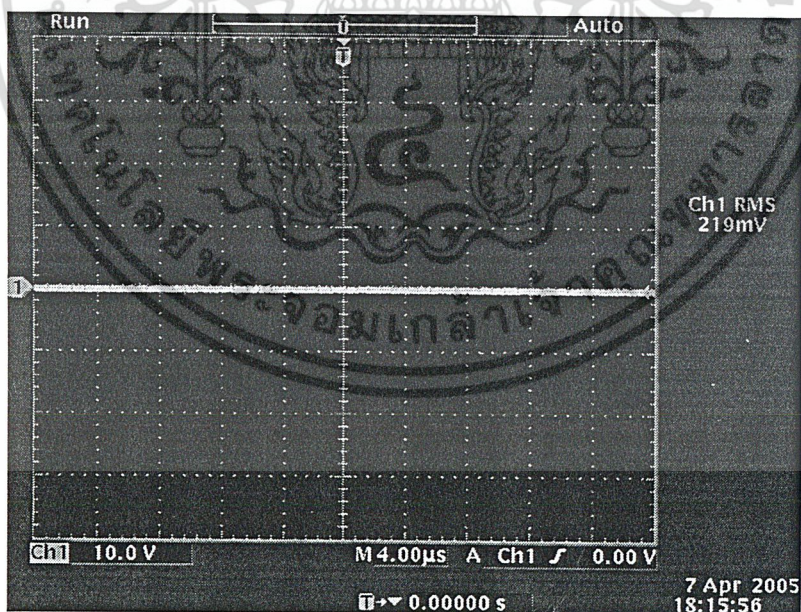
รูปที่ 4.15 แสดงสถานะของหน้าเว็บเพจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



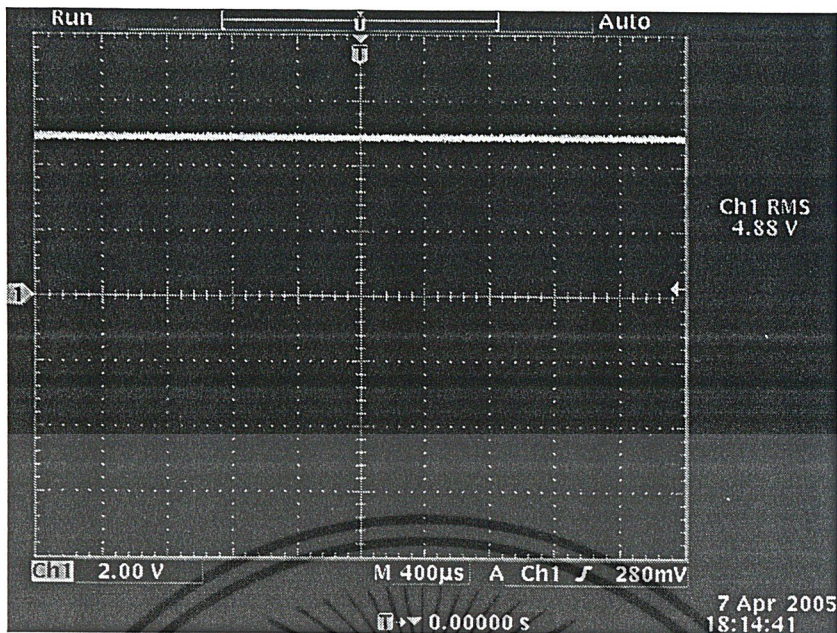
รูปที่ 4.16 แสดงสถานะหน้าเว็บเพจเมื่อมีการกดปุ่ม

จากรูปที่ 4.15 และ 4.16 จะเป็นการจับสัญญาณจากสาย UTP โดยที่ Ch 1 จะทำการจับสัญญาณที่ สายส่งที่มีขั้วเป็นบวกและลบ (Output Transmit Data) ส่วนที่ Ch 2 จะทำการจับสัญญาณที่สายรับที่มีขั้วเป็นบวกและลบ (Input Receive Data) ซึ่งจะเป็นสัญญาณการเชื่อมต่อระหว่างฮับ กับ RCM 2200 จากรูป จะสังเกตเห็นว่าถ้ามีการติดต่อกันระหว่างคอมพิวเตอร์ กับ RCM2200 จะทำให้มีค่าระดับแรงดันที่สูงขึ้น เมื่อทำการกดปุ่มบนหน้าเว็บเพจ



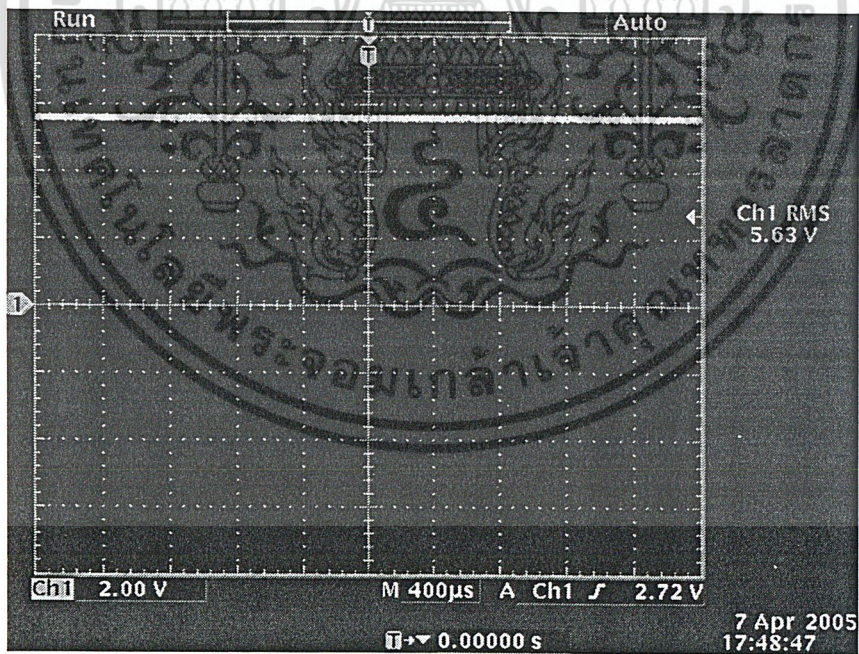
รูปที่ 4.17 แสดงสถานะแรงดันเมื่อทำการปิดประตู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



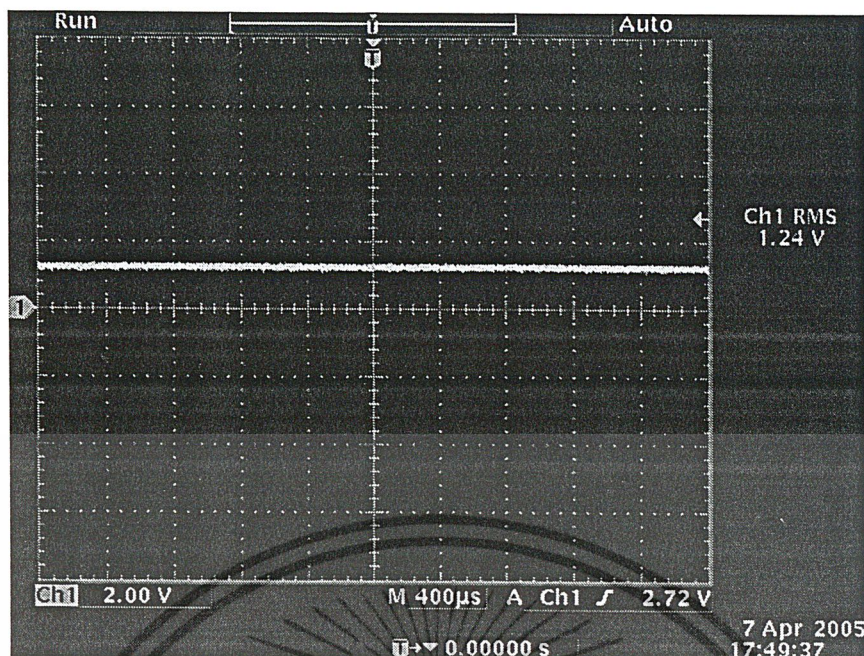
รูปที่ 4.18 แสดงสถานะแรงดันเมื่อทำการเปิดประตู

จากรูปที่ 4.17 และ 4.18 จะเป็นการจับสัญญาณที่เข้ามาที่ Port C ของ 8255 ตัวที่ 1 ซึ่งเป็นพอร์ตที่ทำกรรับค่า การเปิดปิดของประตูและหน้าต่าง เพื่อนำค่าไปให้ RCM 2200 ทำการประมวลผล โดยค่าที่ได้จะได้อาจจากการต่อวงจร ดังรูป 3.4



รูปที่ 4.19 แสดงสถานะแรงดันกรณีที่ไม่มีการเดิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.20 แสดงสถานะแรงดันกรณีที่มีควัน

จากรูปที่ 4.19 และ 4.20 จะเป็นการจับสัญญาณที่เข้ามาทาง Port C ของ 8255 ตัวที่ 1 ซึ่งเป็นพอร์ตที่ทำการรับค่า ของ วงจรตรวจสอบควัน โดยอาศัยหลักการของไอออน คือเมื่อมีควันเข้าไปในกล่องที่บรรจุ Ionization จะไปรบกวนกระแสไฟฟ้า โดยอนุภาคของควันจะไปจับกับไอออน ซึ่งจะทำให้สมดุลถูกรบกวน การปรับการไหลของกระแสไฟระหว่างแผ่นโลหะ 2 แผ่น ในกล่องที่บรรจุสาร Ionization และเมื่อระดับแรงดันลดลงเรื่อยๆ จนถึงศูนย์ โดยคอนโทรลเลอร์จะรับสถานะการเปลี่ยนแปลงแล้วแจ้งเตือนภัยเมื่อ มีค่าแรงดันประมาณ 1.3 V

```

e:8888 - HyperTerminal
File Edit View Call Transfer Help
[Icons]
AT+CHGS=17
> 001100098140704359F700F2AA05C4F75B1E02
+CHGS: 18
OK
Connected 0:01:02 Auto detect 19200 8+H-1 [SCPI] [CAPS] [MM] [Capture] [Print]

```

รูปที่ 4.21 สัญญาณการตอบกลับของการส่งข้อความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.21 แสดงสัญญาณข้อมูลการส่งข้อความสั้นไปที่โทรศัพท์เคลื่อนที่ จากรูปจะใช้ให้คอมพิวเตอร์ทำหน้าที่แทนโทรศัพท์เคลื่อนที่โดยจะทำการติดต่อกับ MCS 51 ที่ขา Tx และ Rx โดยทางพอร์ต com 1 โดยใช้ค่าบรอดเลขเป็น 9600



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุปและวิจารณ์

5.1 สรุปปัญหาที่พบในโครงการ

โดยปัญหานั้นจะเป็นส่วนของวงจรที่ต่อใช้งาน เช่น วงจรควบคุมมอเตอร์ที่ต่อมาใช้งานจากตัวไอซี 8255 คือตอนที่ต่อวงจรนี้ทางชุดทดลองวงจรนี้จะจับมอเตอร์ได้จริงแต่พอเราต่อวงจรนี้กับวงจรรวมจะพบว่าไม่สามารถจับมอเตอร์ได้ เหตุเพราะว่าวงจรรวมนั้นจะมีโหลดมากจึงทำให้เราได้กระแสที่โหลดของวงจรมอเตอรืมีค่าน้อยมากๆ ไม่สามารถที่จะจับมอเตอร์ได้ ปัญหานี้เราแก้ได้โดยใส่ตัวคาปาซิเตอร์สักหนึ่งตัวที่วงจรเพื่อขั้กระแสให้วงจรนี้

ส่วนวงจรอื่นนั้นจะไม่ค่อยมีปัญหาเพราะไม่ได้ใช้กระแสมากเหมือนมอเตอร์ปัญหาที่จะอยู่ที่การต่อวงจรผิดพลาดแต่ก็ไม่ส่งผลกระทบมากเท่าใดนัก

เนื่องจากการใช้งาน Web Browser ในการควบคุมการทำงานเพื่อแสดงสถานะของระบบ ซึ่งการทำงานของ Web Browser นั้นจะมีปัญหาในการทำงานคือ จะไม่สามารถทำงานในรูปแบบ Real Time ได้ จึงจำเป็นต้องมีการตั้งค่าเพื่อให้ Web Browser ทำการ Refresh ตัวเอง เพื่อการสถานการณ์ทำงาน

5.2 สรุปผลของการดำเนินโครงการ

โครงการนี้ได้นำ RCM2200 มาใช้ในการสื่อสารผ่านระบบอินเทอร์เน็ต ซึ่งใช้โปรโตคอล TCP/IP มาใช้ในการทำงานโดยการเชื่อมต่อ ข้อดีของการใช้งานผ่านระบบอินเทอร์เน็ต ไม่ว่าจะอยู่ในระยะการใช้งานใกล้หรือไกล ก็สามารถตรวจสอบหรือสั่งงานผ่านการเชื่อมต่อของระบบ Network ได้ ในการใช้งานการควบคุมจะทำผ่าน Web Browser แล้วทำการพิมพ์ค่า IP Address ของ RCM2200 ที่ได้ทำการตั้งไว้ ซึ่งจะทำงานเป็น Server

ข้อดีของการนำ Web Browser ของโปรแกรม Internet Explorer มาใช้งานซึ่งทำให้ไม่ต้องเขียนโปรแกรมที่เครื่อง Client เพราะสามารถเปิดใช้งานได้เลย จึงสะดวกที่จะนำมาใช้งาน

จากการทดลองของโครงการนี้ ซึ่งจะใช้ RCM2200 ทำงานเป็น Server อยู่ภายในบ้าน แล้วใช้ควบคุมอุปกรณ์ต่อพ่วงต่างๆ เมื่อเกิดเหตุการณ์ผิดปกติขึ้นภายในบ้าน ระบบก็จะทำการตรวจสอบการทำงาน และแจ้งเตือนภัยผ่านทาง Web Browser และส่งสัญญาณให้ MCS 51 ทำการส่ง SMS ไปยังโทรศัพท์เคลื่อนที่ปลายทาง ว่าเกิดเหตุผิดปกติที่ไหน เช่น มีการบุกรุกที่ประตูที่ 1 ก็จะทำการส่งข้อความ "Door1 Open!" เมื่อปลายทางได้รับข้อความเรียบร้อยแล้ว สามารถตรวจสอบทาง Web Browser ได้

5.3 แนวทางการพัฒนาโครงการ

1. สามารถสั่งงานควบคุมกลับทาง SMS ได้
2. สามารถเปลี่ยนหมายเลขโทรศัพท์มือถือปลายทางได้
3. สามารถเปลี่ยนแปลง Username และ Password ได้
4. สามารถเปลี่ยนแปลงหมายเลข IP Address ได้

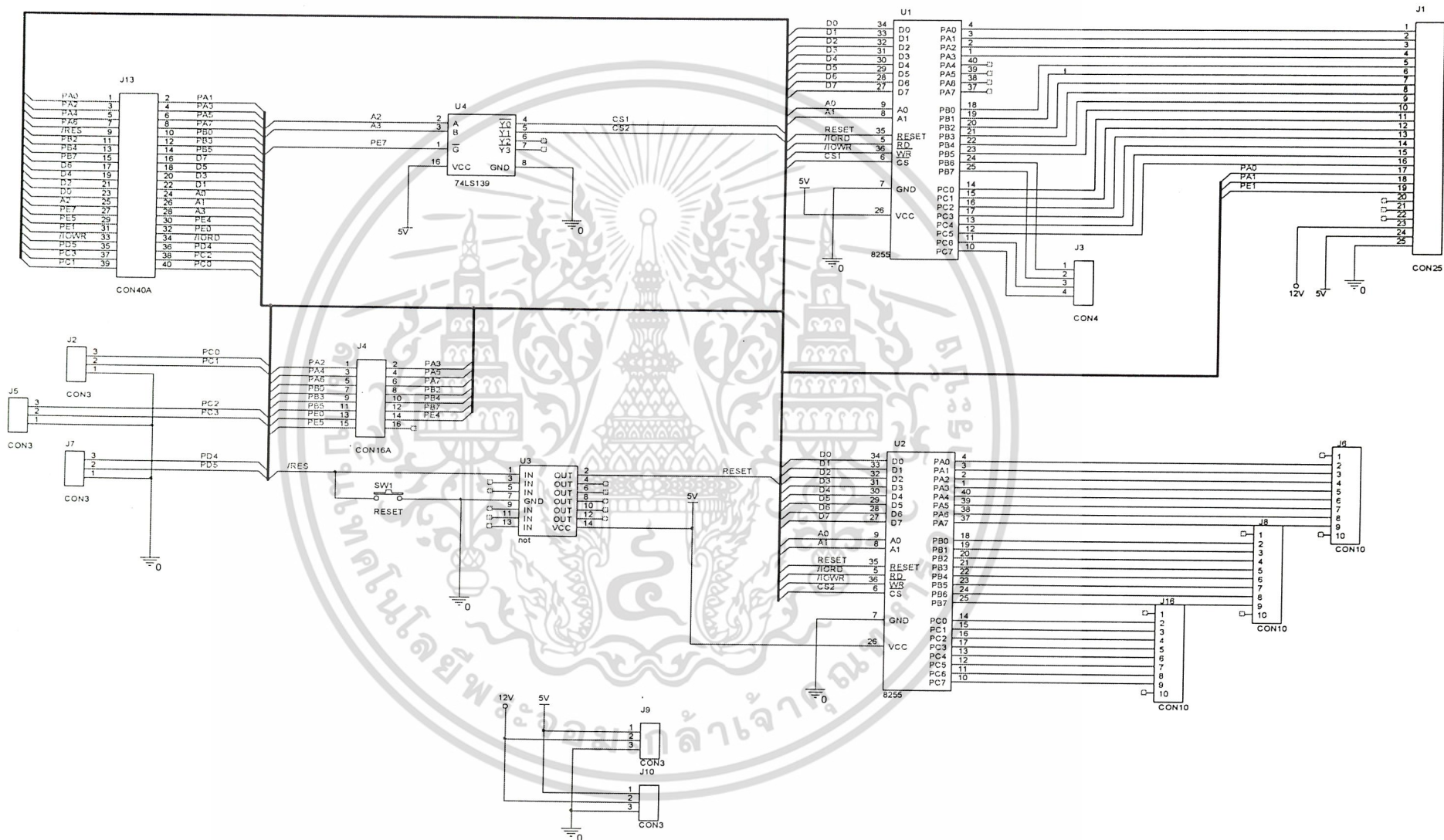
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 ข้อดีของการใช้งาน Short Message

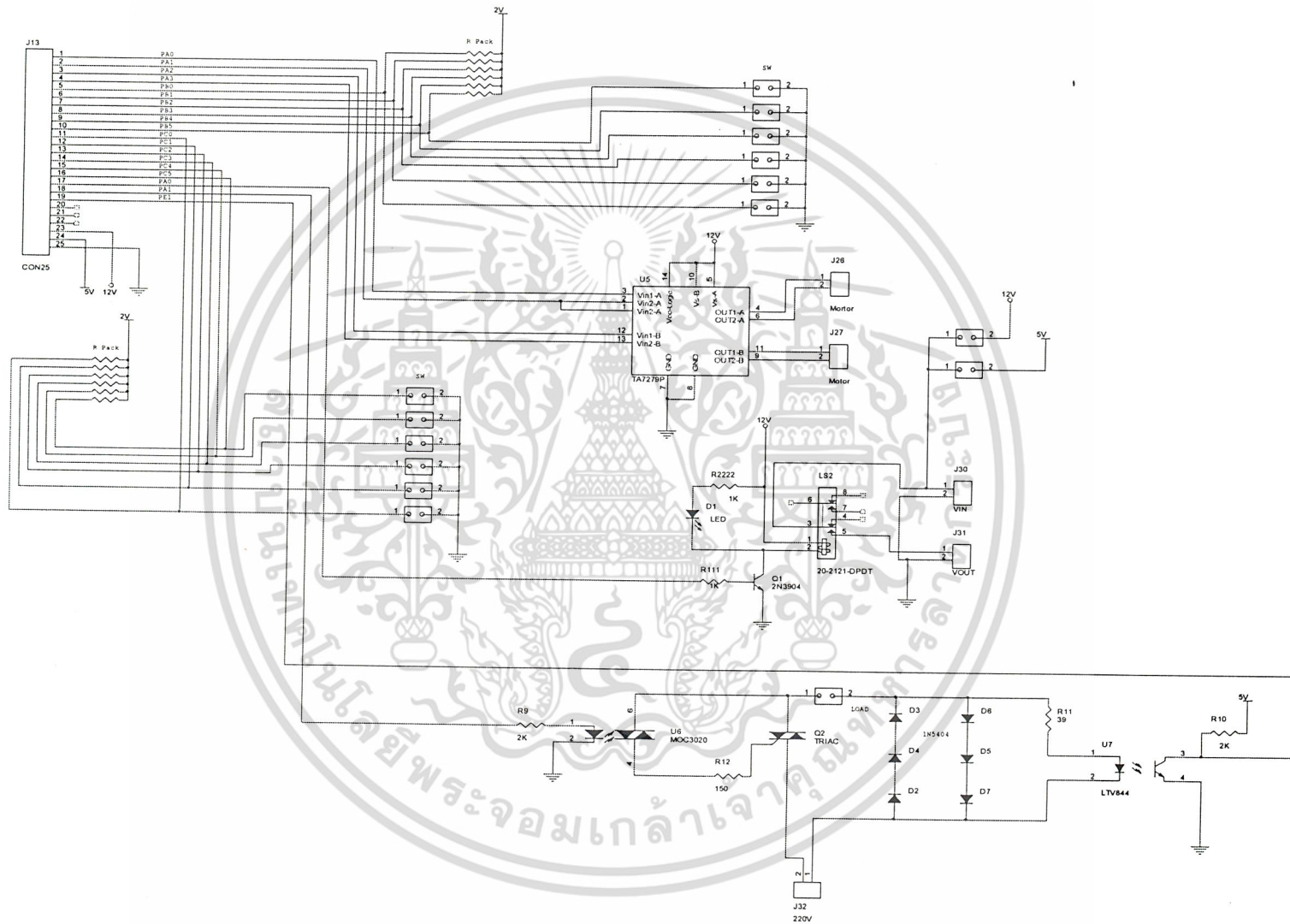
1. มีความเชื่อถือได้สูง เนื่องจากข้อมูลที่ถูกส่งไปนั้นจะไม่สูญหายไปจนกว่าจะถึงปลายทางในช่วงระยะเวลาหนึ่ง
2. ครอบคลุมพื้นที่การใช้งานได้เป็นบริเวณกว้าง เนื่องจากสามารถทำงานได้ในทุกพื้นที่ที่มีโครงข่ายของระบบโทรศัพท์เคลื่อนที่อยู่



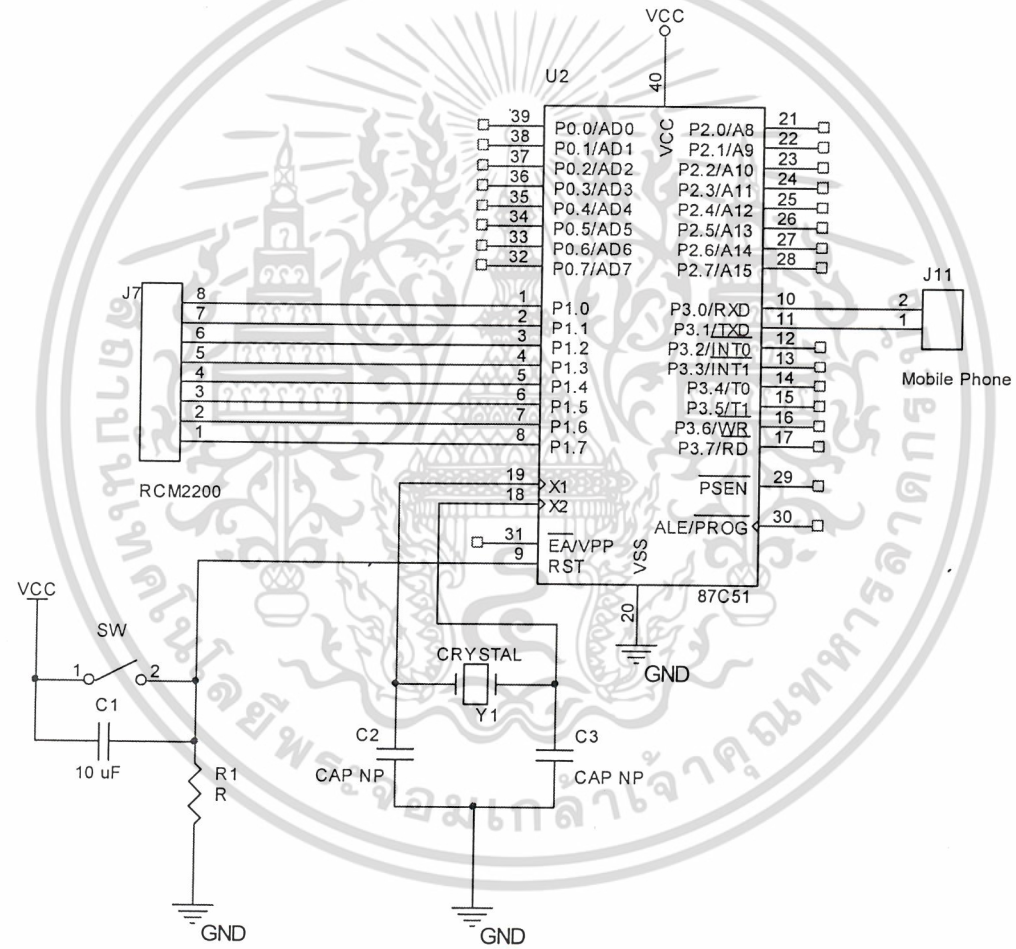
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



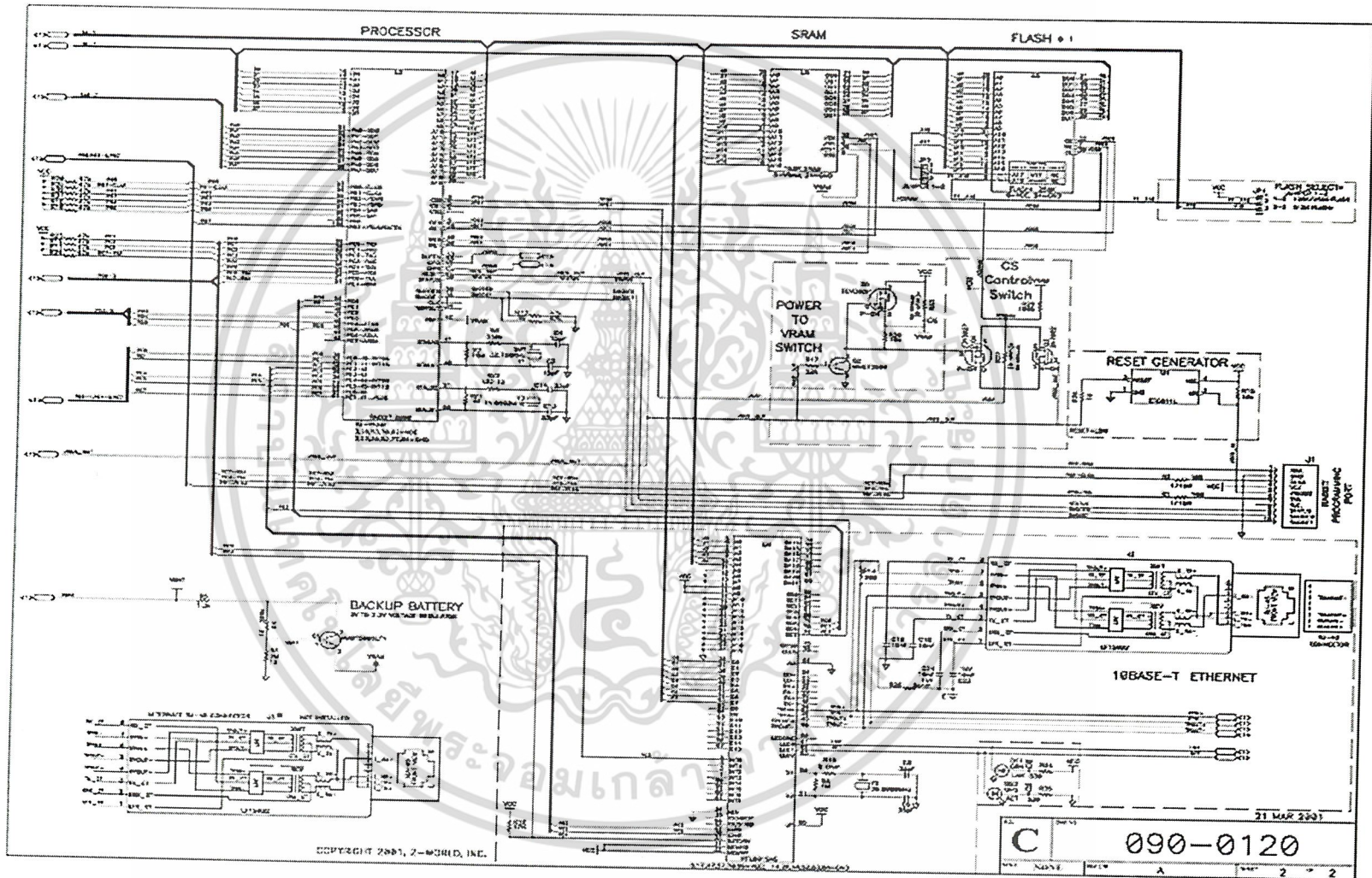
รูปที่ ก-1 วงจรขยายพอร์ต RCM2200



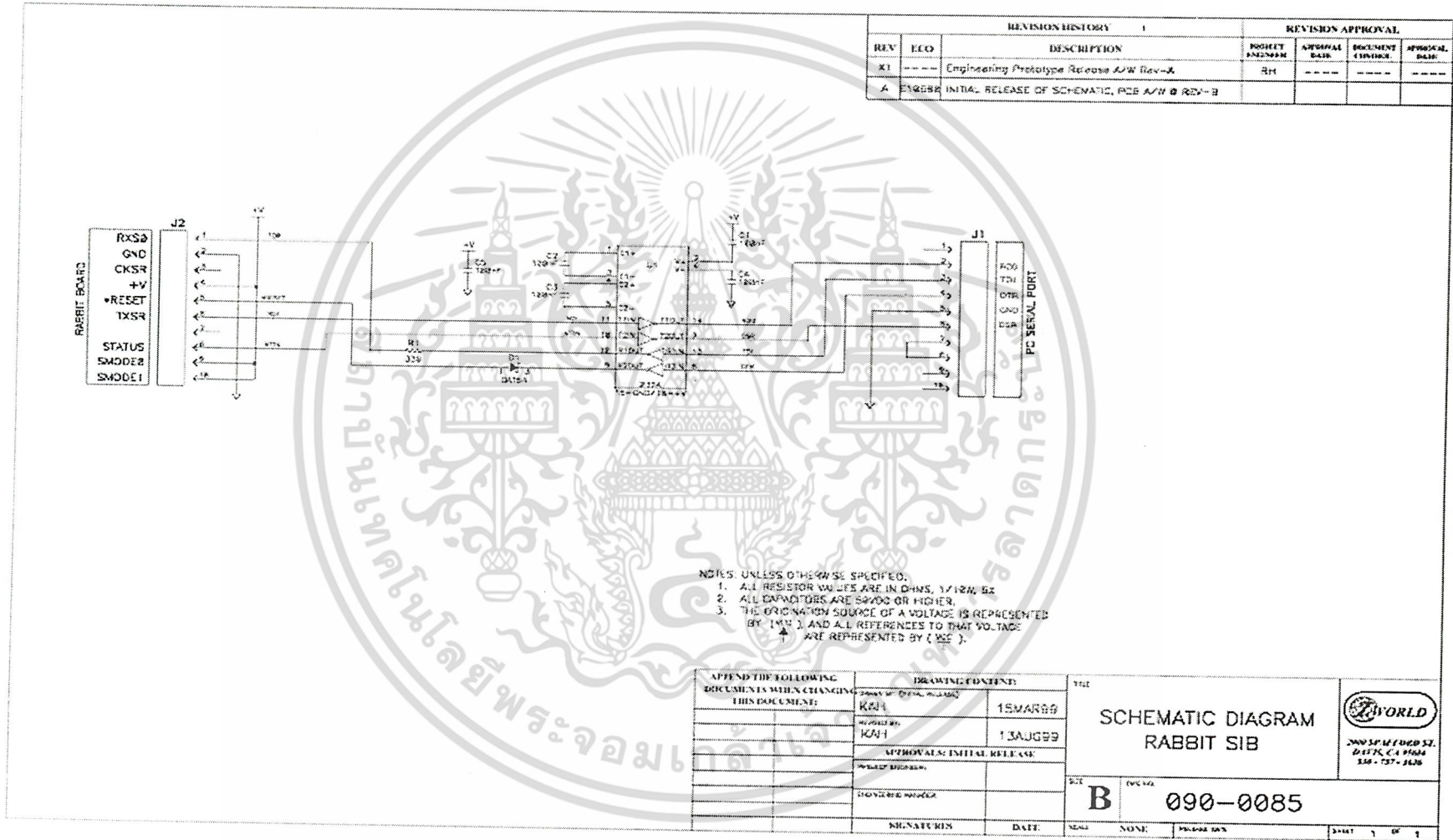
รูปที่ ก-2 วงจรควบคุม



รูปที่ ก-3 วงจรส่ง SMS



รูปที่ ก-4 วงจร RCM 2200



รูปที่ ก-5 วงจร Programming cable

ภาคผนวก ข

โปรแกรมควบคุมระบบ

```

#define My_IP_ADDRESS "161.246.18.61"
#define My_NETMASK    "255.255.255.0"
#define REDIRECTHOST    MY_IP_ADDRESS

#define HTTP_MAXSERVERS 1
#define MAX_TCP_SOCKET_BUFFERS 1

#define TCP_BUF_SIZE 2048
#define DINBUFSIZE 15
#define DOUTBUFSIZE 15
#define SSPEC_MAXSPEC 24

#define memmap xmem
#define use "dcrtcp.lib"
#define use "http.lib"

#define ximport "C:\DCRABBIT_8.01\SMS3t\index.shtml" index_shtml
#define ximport "C:\DCRABBIT_8.01\SMS3t\ledon.gif" ledon_gif
#define ximport "C:\DCRABBIT_8.01\SMS3t\ledoff.gif" ledoff_gif
#define ximport "C:\DCRABBIT_8.01\SMS3t\on.gif" on_gif
#define ximport "C:\DCRABBIT_8.01\SMS3t\off.gif" off_gif
#define ximport "C:\DCRABBIT_8.01\SMS3t\f1.gif" f1_gif
#define ximport "C:\DCRABBIT_8.01\SMS3t\all.gif" all_gif
#define ximport "C:\DCRABBIT_8.01\SMS3t\button.gif" button_gif
#define ximport "C:\DCRABBIT_8.01\SMS3t\security11.c" security11_c

#define PORT_A1 0xe000
#define PORT_B1 0xe001
#define PORT_C1 0xe002
#define PORT_CON1 0xe003
#define PORT_A2 0xe004
#define PORT_B2 0xe005
#define PORT_C2 0xe006
#define PORT_CON2 0xe007
#define REDIRECTTO "http://" REDIRECTHOST "/index.shtml"

const HttpType http_types[] =
{
    { ".shtml", "text/html", shtml_handler}, // ssi
    { ".html", "text/html", NULL}, // html
    { ".cgi", "", NULL}, // cgi
    { ".gif", "image/gif", NULL}
};

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

const HttpRealm admin =
    {
        "kmitl","telecom","Sing In"
    };

char system[15];
char door1[15];
char door2[15];
char window1[15];
char window2[15];
char window3[15];
char fire[15];
char alarm[15];
char ac[15];
char status1[7];
char status2[9];
char status3[10];
char status4[11];
char status5[11];
char status6[11];
char status7[11];
char status8[7];
char status9[7];
char pic[15];
unsigned char censer,sw;

void i8255_int1(void)
{
    WrPortI ( SPCR, &SPCRShadow, 0x84 );
    // Disable slave port function
    WrPortI ( PEFr, &PEFRShadow, (PEFRShadow | 0x80));
    // Set PE7 to be external I/O Chip Select
    WrPortI ( PEDDR, &PEDDRShadow, (PEDDRShadow | 0x80));
    // Set PE7 as an output
    WrPortI ( IB7CR, &IB0CRShadow, 0x08 );
    WrPortE(PORT_CON1, NULL, 0x8B);
    // Set port B1 and C1 to be input
    WrPortE(PORT_CON2, NULL, 0x80);
    // Set 8255_2 to All output
}

void delay()
{
    int i;
    i=0;

    while (i!=1) {
        costate {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        waitFor(DelayMs(200));
waits until 200 ms have passed since //
        //printf("i = %d\n", i);
        i++;
    }
}

void delay2()
{
    int j;
    j=0;

    while (j!=1) {
        costate {
            waitFor(DelaySec(300));
waits until 5 min have passed since //
            //printf("j = %d\n", j);
            j++;
        }
    }
}

void censorscan(void)
{
    if((censer|0xfe)==0xfe) // Door 1
        {strcpy(door1,"ledoff.gif");
        strcpy(status2,"close");}
    else
        {strcpy(door1,"ledon.gif");
        strcpy(status2,"open");}

    if((censer|0xfd)==0xfd) // Door 2
        {strcpy(door2,"ledoff.gif");}
    else
        {strcpy(door2,"ledon.gif");}

    if((censer|0xfb)==0xfb) // Window 1
        {strcpy(window1,"ledoff.gif");
        strcpy(status4,"safe");}
    else
        {strcpy(window1,"ledon.gif");
        strcpy(status4,"warning");}

    if((censer|0xf7)==0xf7) // Window 2
        {strcpy(window2,"ledoff.gif");
        strcpy(status5,"safe");}
    else
        {strcpy(window2,"ledon.gif");
        strcpy(status5,"warning");}

    if((censer|0xef)==0xef) // Window 3

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        {strcpy(window3, "ledoff.gif");
        strcpy(status6, "safe");}
    else
    {strcpy(window3, "ledon.gif");
    strcpy(status6, "warning");}

    if((censer|0xdf)==0xdf) // Fire
        {strcpy(fire, "ledoff.gif");
        strcpy(status7, "safe");}
    else
        {strcpy(fire, "ledon.gif");
        strcpy(status7, "warning");}
}

void swscan(void)
{
    switch(sw)
    {
        case 0xfe:
        {
            if((strcmp(status1, "off")==0)&(strcmp(system, "ledoff.gif")=
=0))
            {
                locktog();}
            break;
        }
        case 0xfd:
        {
            onltog();}
            break;
        }
        case 0xfb:
        {
            if((strcmp(status1, "off")==0)&(strcmp(system, "ledoff.gif")=
=0))
            {
                on2tog();}
            break;
        }
        case 0xf7:
        {
            if((strcmp(status1, "off")==0)&(strcmp(system, "ledoff.gif")=
=0))
            {
                on3tog();}
            break;
        }
    }
}

int onltog(HttpState* state) //System
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    if((strcmp(status1,"off")==0)&(strcmp(system,"ledoff.gif")=
=0))
    {
        //Open System

        if((strcmp(door1,"ledoff.gif")==0)&(strcmp(door2,"ledoff.gi
f")==0)&(strcmp(window1,"ledoff.gif")==0)&(strcmp(window2,"ledoff
.gif")==0)&(strcmp(window3,"ledoff.gif")==0)&(strcmp(fire,"ledoff
.gif")==0))
        {
            strcpy(system,"ledon.gif");
            strcpy(status1,"on");
            WrPortE(PORT_A2,NULL,0x40);

            if((strcmp(status3,"unlock")==0)&(strcmp(door2,"ledoff.gif"
)==0)) //Lock Door2
            {
                locktog();
                strcpy(status3,"lock");
                strcpy(status1,"on");
                strcpy(system,"ledon.gif");
            }
        }
        else //Close System
        {
            strcpy(system,"ledoff.gif");
            strcpy(status1,"off");
            WrPortE(PORT_A2,NULL,0x00);
            if(strcmp(status3,"lock")==0)
            {
                locktog();
                strcpy(status3,"unlock");
                strcpy(status1,"off");
                strcpy(system,"ledoff.gif");
            }
        }
        cgi_redirectto(state,REDIRECTTO);
        return 0;
    }

int locktog(HttpState* state)
{
    if((strcmp(status1,"off")==0)&(strcmp(system,"ledoff.gif")=
=0))
    {
        if(strcmp(status3,"unlock")==0)
        {
            strcpy(status3,"lock");
            WrPortE(PORT_A1,NULL,0x01);
            delay();
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        WrPortE(PORT_A1, NULL, 0x00);
        delay();
    }
    else
    {
        strcpy(status3, "unlock");
        WrPortE(PORT_A1, NULL, 0x02);
        delay();
        WrPortE(PORT_A1, NULL, 0x00);
        delay();
    }
}
cgi_redirectto(state, REDIRECTTO);
return 0;
}

int on3tog(HttpState* state) //AC
{
    if((strcmp(status9, "on")==0)&(strcmp(ac, "ledon.gif")==0))
    {
        strcpy(ac, "ledoff.gif");
        strcpy(status9, "off");
        WrPortI(SPCR, &SPCRShadow, 0x84);
        BitWrPortI(PADR, &PADRShadow, 0, 1);
//off
    }
    else
    {
        strcpy(ac, "ledon.gif");
        strcpy(status9, "on");
        WrPortI(SPCR, &SPCRShadow, 0x84);
        BitWrPortI(PADR, &PADRShadow, 1, 1);
//on
    }
    cgi_redirectto(state, REDIRECTTO);
    return 0;
}

void send(void)
{
    if((strcmp(status1, "on")==0)&(strcmp(door1, "ledon.gif")==0))
    { WrPortE(PORT_A2, NULL, 0xFE); }
    if((strcmp(status1, "on")==0)&(strcmp(door2, "ledon.gif")==0))
    { WrPortE(PORT_A2, NULL, 0xFD); }
    if((strcmp(status1, "on")==0)&(strcmp(window1, "ledon.gif")==0))
    { WrPortE(PORT_A2, NULL, 0xFB); }
    if((strcmp(status1, "on")==0)&(strcmp(window2, "ledon.gif")==0))
    { WrPortE(PORT_A2, NULL, 0xF7); }
    if((strcmp(status1, "on")==0)&(strcmp(window3, "ledon.gif")==0))
    { WrPortE(PORT_A2, NULL, 0xEF); }
    if((strcmp(status1, "on")==0)&(strcmp(fire, "ledon.gif")==0))

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    {   WrPortE(PORT_A2, NULL, 0xDF); }
}

void alarm_on(void)
{
    if((strcmp(pic, "off.gif")==0)&(strcmp(status1, "on")==0)&(strcmp(system, "ledon.gif")==0))
    {
        WrPortI ( SPCR, &SPCRShadow, 0x84 );
        // Disable slave port function
        WrPortI ( PBDR, &PBDRShadow, 0x80 );
        WrPortI(PADR, NULL, 0x01); //open
        WrPortE(PORT_A2, NULL, 0x80);
        strcpy(pic, "on.gif");
        strcpy(alarm, "ledon.gif");
        strcpy(status8, "on");
        send();
    }
}

void warning_on(void)
{
    if((strcmp(pic, "off.gif")==0)&(strcmp(status1, "on")==0)&(strcmp(system, "ledon.gif")==0))
    {
        WrPortI ( SPCR, &SPCRShadow, 0x84 );
        // Disable slave port function
        WrPortI ( PBDR, &PBDRShadow, 0x80 );
        WrPortI(PADR, NULL, 0x01); //open
        WrPortE(PORT_A2, NULL, 0x80);
        strcpy(pic, "on.gif");
        strcpy(alarm, "ledon.gif");
        strcpy(status8, "on");
        send();
        delay2();
        strcpy(ac, "ledoff.gif");
        strcpy(status9, "off");
        BitWrPortI(PADR, &PADRShadow, 0, 1);
    }
}

void alarm_off(void)
{
    WrPortI ( SPCR, &SPCRShadow, 0x84 );
    // Disable slave port function
    WrPortI ( PBDR, &PBDRShadow, 0x80 );
    WrPortI(PADR, NULL, 0x00); //open
    WrPortE(PORT_A2, NULL, 0x00);
    strcpy(pic, "off.gif");
    strcpy(alarm, "ledoff.gif");
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        strcpy(status8,"off");
    }

int testsiren(HttpState* state)
{
    if((strcmp(pic,"off.gif")==0)&(strcmp(status1,"off")==0)&(s
trcmp(system,"ledoff.gif")==0))
    {
        WrPortI ( SPCR, &SPCRShadow, 0x84 );
        // Disable slave port function
        WrPortI ( PBDR, &PBDRShadow, 0x80 );
        WrPortI(PADR,NULL,0x01); //open
        WrPortE(PORT_A2,NULL,0x80);
        strcpy(alarm,"ledon.gif");
        strcpy(status8,"on");
    }
}

int on2tog(HttpState* state) //siren
{
    if((strcmp(status8,"off")==0)&(strcmp(alarm,"ledoff.gif")==
0))
    {
        testsiren();
    }
    else
    {
        alarm_off();
    }
    cgi_redirectto(state,REDIRECTTO);
    return 0;
}

void systemscan(void)
{
    if((strcmp(system,"ledon.gif")==0)&(strcmp(status1,"on")==0)&(str
cmp(pic,"off.gif")==0)&(strcmp(status2,"open")==0)&(strcmp(door1,
"ledon.gif")==0))
    {
        alarm_on();
    }

    if((strcmp(system,"ledon.gif")==0)&(strcmp(status1,"on")==0)&(str
cmp(pic,"off.gif")==0)&(strcmp(status3,"open")==0)&(strcmp(door2,
"ledon.gif")==0))
    {
        alarm_on();
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if((strcmp(system,"ledon.gif")==0)&(strcmp(status1,"on")==0)&(str
cmp(pic,"off.gif")==0)&(strcmp(status4,"warning")==0)&(strcmp(win
dow1,"ledon.gif")==0))
{
    alarm_on();
}

if((strcmp(system,"ledon.gif")==0)&(strcmp(status1,"on")==0)&(str
cmp(pic,"off.gif")==0)&(strcmp(status5,"warning")==0)&(strcmp(win
dow2,"ledon.gif")==0))
{
    alarm_on();
}

if((strcmp(system,"ledon.gif")==0)&(strcmp(status1,"on")==0)&(str
cmp(pic,"off.gif")==0)&(strcmp(status6,"warning")==0)&(strcmp(win
dow3,"ledon.gif")==0))
{
    alarm_on();
}

if((strcmp(system,"ledon.gif")==0)&(strcmp(status1,"on")==0)&(str
cmp(pic,"off.gif")==0)&(strcmp(status7,"warning")==0)&(strcmp(fir
e,"ledon.gif")==0))
{
    warning_on();
}

}

const HttpSpec http_flashspec[] =
{
    { HTTPSPEC_FILE, "/", index_shtml, NULL, 0,
    NULL, &admin},
    { HTTPSPEC_FILE, "/index.shtml", index_shtml, NULL, 0,
    NULL, &admin},
    { HTTPSPEC_FILE, "/ledon.gif", ledon_gif, NULL, 0, NULL,
    NULL},
    { HTTPSPEC_FILE, "/ledoff.gif", ledoff_gif, NULL, 0, NULL,
    NULL},
    { HTTPSPEC_FILE, "/on.gif", on_gif, NULL, 0, NULL, NULL},
    { HTTPSPEC_FILE, "/off.gif", off_gif, NULL, 0, NULL, NULL},
    { HTTPSPEC_FILE, "/f1.gif", f1_gif, NULL, 0, NULL, NULL},
    { HTTPSPEC_FILE, "/all.gif", all_gif, NULL, 0, NULL, NULL},
    { HTTPSPEC_FILE, "/button.gif", button_gif, NULL, 0, NULL,
    NULL},
    { HTTPSPEC_FILE, "/security11.c", security11_c, NULL, 0,
    NULL, NULL},
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    { HTTPSPEC_VARIABLE, "system", 0, system, PTR16, "%s",
  NULL},
    { HTTPSPEC_VARIABLE, "door1", 0, door1, PTR16, "%s",
  NULL},
    { HTTPSPEC_VARIABLE, "door2", 0, door2, PTR16, "%s",
  NULL},
    { HTTPSPEC_VARIABLE, "window1", 0, window1, PTR16, "%s",
  NULL},
    { HTTPSPEC_VARIABLE, "window2", 0, window2, PTR16, "%s",
  NULL},
    { HTTPSPEC_VARIABLE, "window3", 0, window3, PTR16, "%s",
  NULL},
    { HTTPSPEC_VARIABLE, "fire", 0, fire, PTR16, "%s",
  NULL},
    { HTTPSPEC_VARIABLE, "alarm", 0, alarm, PTR16, "%s",
  NULL},
    { HTTPSPEC_VARIABLE, "ac", 0, ac, PTR16, "%s", NULL},
    { HTTPSPEC_VARIABLE, "pic", 0, pic, PTR16, "%s", NULL},

    { HTTPSPEC_VARIABLE, "status1", 0, status1, PTR16,
"%s", NULL},
    { HTTPSPEC_VARIABLE, "status2", 0, status2, PTR16, "%s",
  NULL},
    { HTTPSPEC_VARIABLE, "status3", 0, status3, PTR16, "%s",
  NULL},
    { HTTPSPEC_VARIABLE, "status4", 0, status4, PTR16, "%s",
  NULL},
    { HTTPSPEC_VARIABLE, "status5", 0, status5, PTR16, "%s",
  NULL},
    { HTTPSPEC_VARIABLE, "status6", 0, status6, PTR16, "%s",
  NULL},
    { HTTPSPEC_VARIABLE, "status7", 0, status7, PTR16, "%s",
  NULL},
    { HTTPSPEC_VARIABLE, "status8", 0, status8, PTR16, "%s",
  NULL},
    { HTTPSPEC_VARIABLE, "status9", 0, status9, PTR16, "%s",
  NULL},

    { HTTPSPEC_FUNCTION, "/on1tog.cgi", 0, on1tog, 0, NULL,
  NULL},
    { HTTPSPEC_FUNCTION, "/locktog.cgi", 0, locktog, 0, NULL,
  NULL},
    { HTTPSPEC_FUNCTION, "/on2tog.cgi", 0, on2tog, 0, NULL,
  NULL},
    { HTTPSPEC_FUNCTION, "/on3tog.cgi", 0, on3tog, 0, NULL,
  NULL},

};

void main()
{
  unsigned char tmp1,tmp2;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

strcpy(status1,"off");
strcpy(status2,"close");
strcpy(status3,"unlock");
strcpy(status4,"safe");
strcpy(status5,"safe");
strcpy(status6,"safe");
strcpy(status7,"safe");
strcpy(status8,"off");
strcpy(status9,"off");

strcpy(system,"ledoff.gif");
strcpy(door1,"ledoff.gif");
strcpy(door2,"ledoff.gif");
strcpy(window1,"ledoff.gif");
strcpy(window2,"ledoff.gif");
strcpy(window3,"ledoff.gif");
strcpy(fire,"ledoff.gif");
strcpy(alarm,"ledoff.gif");
strcpy(ac,"ledoff.gif");
strcpy(pic,"off.gif");

censer=0xC0;
sw=0xFF;

i8255_int1();

sock_init();
http_init();
tcp_reserveport(80);

while(1)
{
    http_handler();

    if(strcmp(ac,"ledon.gif")==0) // check main sw AC
    {
        WrPortI ( SPCR, &SPCRShadow, 0x84 );
        BitWrPortI(PADR,&PADRShadow,1,1);
    }
    if(strcmp(ac,"ledoff.gif")==0)
    {
        WrPortI ( SPCR, &SPCRShadow, 0x84 );
        BitWrPortI(PADR,&PADRShadow,0,1);
    }

    if(strcmp(alarm,"ledon.gif")==0)
    {
        WrPortI ( SPCR, &SPCRShadow, 0x84 );
        // Disable slave port function
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        WrPortI ( PBDR, &PBDRShadow, 0x80 );
        WrPortI(PADR,NULL,0x01);           //open
        WrPortE(PORT_A2,NULL,0x80);
    }
    if(strcmp(alarm,"ledoff.gif")==0)
    {
        WrPortI ( SPCR, &SPCRShadow, 0x84 );
        // Disable slave port function
        WrPortI ( PBDR, &PBDRShadow, 0x80 );
        WrPortI(PADR,NULL,0x00);           //close
        WrPortE(PORT_A2,NULL,0x00);
    }

    tmp1=RdPortE(PORT_C1); // READ DATA FROM 8255_PORT C1
    tmp2=RdPortE(PORT_B1); // READ DATA FROM 8255_PORT B1

    if (censer != tmp1)
    {
        censer=tmp1;
        censorscan();
    }

    if (sw != tmp2)
    {
        sw=tmp2;
        swscan();
    }

    systemscan();
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมการส่ง SMS

```

        DLAY EQU 3BH
        DLAY1 EQU 3CH
        DLAY2 EQU 3DH

        ORG 0000H
MAIN: MOV IE, #00H
      MOV TMOD, #20H
      MOV TL1, #0FDH
      MOV TH1, #0FDH
      MOV PCON, #80H
      MOV SCON, #50H
      MOV P1, #0FFH
      SETB TR1

NEW:  JNB P1.0, DOOR1
      JNB P1.1, DOOR2
      JNB P1.2, WINDOW1
      JNB P1.3, WINDOW2
      JNB P1.4, WINDOW3
      JNB P1.5, FIRE
      SJMP NEW

DOOR1: MOV DPTR, #CMGS1
      LCALL CHECK_SMS1
      MOV DPTR, #HEADER1
      LCALL SEND_SMS1
      CLR RI
      CLR TI
      SJMP MAIN

DOOR2: MOV DPTR, #CMGS2
      LCALL CHECK_SMS1
      MOV DPTR, #HEADER2
      LCALL SEND_SMS1
      CLR RI
      CLR TI
      SJMP MAIN

WINDOW1: MOV DPTR, #CMGS3
      LCALL CHECK_SMS1
      MOV DPTR, #HEADER3
      LCALL SEND_SMS1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CLR RI
CLR TI
SJMP MAIN

```

```

WINDOW2: MOV DPTR,#CMGS4
          LCALL CHECK_SMS1
          MOV DPTR,#HEADER4
          LCALL SEND_SMS1
          CLR RI
          CLR TI
          SJMP MAIN

```

```

WINDOW3: MOV DPTR,#CMGS5
          LCALL CHECK_SMS1
          MOV DPTR,#HEADER5
          LCALL SEND_SMS1
          CLR RI
          CLR TI
          SJMP MAIN

```

```

FIRE: MOV DPTR,#CMGS6
        LCALL CHECK_SMS1
        MOV DPTR,#HEADER6
        LCALL SEND_SMS1
        CLR RI
        CLR TI
        SJMP MAIN

```

```

;*****
CMGS1:  DB "AT+CMGS=23",0DH,0EDH
HEADER1: DB
        "001100098140704359F700F2AA08C4F75B1E033DE1657708",0EDH
;Door1 Open!
CMGS2:  DB "AT+CMGS=23",0DH,0EDH
HEADER2: DB
        "001100098140704359F700F2AA08C4F75B2E033DE1657708",0EDH
;Door2 Open2!
CMGS3:  DB "AT+CMGS=27",0DH,0EDH
HEADER3: DB
        "001100098140704359F700F2AA08D7B0DC9D769F41D7B49BFCBEC742",
0EDH ;Warning Window1!
CMGS4:  DB "AT+CMGS=27",0DH,0EDH
HEADER4: DB
        "001100098140704359F700F2AA08D7B0DC9D769F41D7B49BFCBECB42",
0EDH ;Warning Window2!
CMGS5:  DB "AT+CMGS=27",0DH,0EDH
HEADER5: DB
        "001100098140704359F700F2AA08D7B0DC9D769F41D7B49BFCBECF42",
0EDH ;Warning Window1!
CMGS6:  DB "AT+CMGS=25",0DH,0EDH

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

HEADER6:  DB
           "001100098140704359F700F2AA08D7B0DC9D769F41C6B4BC1C02",0EDH
;Warning Fire!

```

```
;*****
```

```

CHECK_SMS1:  MOV  A,#00H
              MOV  A,@A+DPTR
              INC  DPTR
              CJNE A,#0EDH,CHECK_SMS3
CHECK_SMS2:  LCALL DELAY
              CLR  RI
              RET

```

```

CHECK_SMS3:  LCALL SEND
              SJMP CHECK_SMS1

```

```
;*****
SEND_SMS1:  MOV  A,#00H
              MOV  A,@A+DPTR
              INC  DPTR
              CJNE A,#0EDH,SEND_SMS3
SEND_SMS2:  MOV  A,#1AH
              LCALL SEND
              LCALL DELAY_8S
              CLR  RI
              RET

```

```

SEND_SMS3:  LCALL SEND
              SJMP SEND_SMS1

```

```
;*****
SEND: MOV     SBUF,A
        JNB   TI,$
        CLR  TI
        RET

```

```
;*****
DELAY:  MOV  DLAY,#00H
        MOV  DLAY1,#00H
        MOV  DLAY2,#8
DELAY1:  DJNZ DLAY,DELAY1
        DJNZ DLAY1,DELAY1
        DJNZ DLAY2,DELAY1
        RET

```

```
;*****
DELAY_8S:  MOV  DLAY,#00H
           MOV  DLAY1,#00H
           MOV  DLAY2,#64
DELAY_8SN:  DJNZ DLAY,DELAY_8SN
           DJNZ DLAY1,DELAY_8SN

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DJNZ DLAY2,DELAY_8SN
RET

;*****

END



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

โสรัถย์ อุณหะวารากร, “เซมิคอนดักเตอร์อิเล็กทรอนิกส์.” ฉบับที่ 235 กรุงเทพฯ
ปรมาภรณ์ เนตรวิกรม, “เซมิคอนดักเตอร์อิเล็กทรอนิกส์.” ฉบับที่ 264 กรุงเทพฯ
สิริลักษณ์ จันทรางศุ, “มาตรฐานการติดตั้งสัญญาณเตือนอัคคีภัย”

Z-World, “Dynamic C [CD-ROM]”

<http://www.rabbitsemiconductor.com/products/rcm2200/index.shtml>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้