



เครื่องควบคุมสัญญาณไฟจราจรด้วยไมโครโปรเซสเซอร์

TRAFFIC LIGHT CONTROLLER BASED MICROPROCESSOR



นายฐระพงษ์ ปุกหุด
นางสาววรรณ กนิษฐานนท์
นายวรวุฒิ ภักดีสุสุข

29 ก.พ. 2541
วัน เดือน ปี.....
เลขทะเบียน..... 038153
เลขเรียกหนังสือ..... T.99173 2181๑

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการปีการศึกษา 2539 เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

เครื่องควบคุมสัญญาณไฟจราจรด้วยไมโครโปรเซสเซอร์
TRAFFIC LIGHT CONTROLLER BASED MICROPROCESSOR

โดย

นายวสุระพงษ์ ปุกหุด 36014367

นางสาววรวรรณ กนิษฐานนท์ 36014376

นายวรุดิ ภัคดีสุสุข 36014379

อาจารย์ที่ปรึกษา

รศ.ดร. ถวิล พึ่งมา

ปริญญาานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2539

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2539

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องควบคุมสัญญาณไฟจราจรด้วยไมโครโปรเซสเซอร์

TRAFFIC LIGHT CONTROLLER BASED MICROPROCESSOR

ผู้จัดทำ

1. นายวรุระพงษ์ ปุกหุด 36014367
2. นางสาววรวรรณ กนิษฐานนท์ 36014376
3. นายวรุฒติ ภักดีสุสุข 36014379



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องควบคุมสัญญาณไฟจราจรด้วยไมโครโปรเซสเซอร์
TRAFFIC LIGHT CONTROLLER BASED MICROPROCESSOR

โดย	นายวสุระพงษ์	ปุ๊กนุด	36014367
	นางสาววรวรรณ	กนิษฐานนท์	36014376
	นายวรุฒิ	ภักดีสุสุข	36014379

อาจารย์ที่ปรึกษา รศ.ดร. ถวิล พึ่งมา

บทคัดย่อ

เครื่องควบคุมสัญญาณไฟจราจร เป็นโครงการที่มีหลักการการทำงานโดยใช้ไมโครโปรเซสเซอร์เป็นตัวควบคุมสัญญาณไฟจราจรบริเวณสี่แยก ซึ่งมีขนาดถนน 4 ช่องทางจราจรไปกลับ โดยมีสัญญาณไฟจราจรหลายรูปแบบ เช่น ไฟเขียวทางตรงสวนกัน ทางตรงกับทางเลี้ยวขวา หรือทางเลี้ยวขวาสวนกัน เป็นต้น และสามารถปรับเปลี่ยนช่วงเวลาของสัญญาณไฟจราจรได้ตามสภาพการจราจร หรือตามเวลาที่ตั้งไว้ในโปรแกรมซึ่งอ้างอิงเวลาจริงกับนาฬิกาที่มีอยู่ โดยผู้ใช้สามารถเลือกใช้รูปแบบของสัญญาณไฟได้เองตามความเหมาะสม นอกจากนี้ยังออกแบบสำหรับใช้เชื่อมต่อสัญญาณกับแยกถัดไปได้ โดยใช้สายทองแดงเกลียวคู่เป็นตัวกลางในการส่งผ่านสัญญาณ โดยจะมีส่วนอินเทอร์เฟสระหว่างตัวควบคุมช่วยในการติดต่อระหว่างกัน เพื่อให้ได้สัญญาณไฟที่มีความสอดคล้องกัน

ABSTRACT

Traffic light controller is operated by using the microprocessor to control traffic light at 4 lanes cross section. It has many format of traffic light, it can be changed, for example, green light for straight and straight, straight and turn right, turn right and turn right and etc... . The mode of traffic light controller have manual mode and automatic mode. The user can select the format of traffic light to relisse the traffic ploblems. It will be worked by timer that is recorded. It is designed to link the other cross section for synchronized traffic light via the twist pair cable.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	เรื่อง	หน้า
บทที่ 1	บทนำ	1
บทที่ 2	ทฤษฎีและหลักการ	3
	2.1 ทฤษฎีและหลักการทำงานของสัญญาณไฟจราจร	3
	2.2 ทฤษฎีและหลักการทำงานโดยทั่วไปของอุปกรณ์ควบคุมสัญญาณไฟจราจร	4
บทที่ 3	ขั้นตอนการออกแบบบล็อคดีอะแกรมและวงจรการทำงานของเครื่องควบคุมสัญญาณไฟจราจร	12
	3.1 ลักษณะการทำงานของชุดควบคุมสัญญาณไฟจราจร	12
	3.2 การออกแบบในส่วนของหน่วยควบคุมและประมวลผลของชุดควบคุมสัญญาณไฟจราจร	13
	3.3 การออกแบบในส่วนคีย์รับข้อมูล	17
	3.4 การออกแบบโปรแกรมที่ใช้ควบคุมอุปกรณ์สัญญาณไฟจราจร	18
บทที่ 4	การทดลองและผลการทดลอง	22
	4.1 การทดลองทำการทดสอบการควบคุมการทำงานของส่วนประมวลผล	22
	4.2 การทดลองทำการทดสอบการทำงานของพอร์ทของ 8255 กับ ชุดแสดงสัญญาณไฟจราจรจำลอง	24
	4.3 การทดลองการทดสอบการทำงานของอุปกรณ์แสดงผล LCD	28
	4.4 การทดลองการทำงานของอุปกรณ์หน่วยความจำประเภท RAM	33
	4.5 การทดลองการทำงานของอุปกรณ์ให้สัญญาณนาฬิกา	36
	4.6 การทดลองการทำงานของอุปกรณ์คีย์รับข้อมูล	39
	4.7 การทดลองการทำงานของโปรแกรมควบคุมการทำงานของไฟจราจร	46
	4.8 การทดลองการทำงานเชื่อมต่อสัญญาณระหว่างอุปกรณ์ควบคุม 2 ชุด	47
	4.9 การทดลองการทำงานของชุดควบคุมไฟจราจรที่สามารถเชื่อมต่อสัญญาณระหว่างชุดควบคุมได้	52
บทที่ 5	บทสรุปและวิจารณ์	93
	กิตติกรรมประกาศ	
	หนังสืออ้างอิง	

สารบัญรูปภาพ

รูปที่		หน้า
2.1	แสดงรูปแบบของสัญญาณไฟจราจรทั้งหมดที่เป็นไปได้ใน 3 แยก	3
2.2	แสดงรูปแบบของสัญญาณไฟจราจรทั้งหมดที่เป็นไปได้ใน 4 แยก	4
2.3	แสดงภาพอุปกรณ์ในส่วนควบคุมและประมวลผล	5
2.4	แสดงภาพบอร์ดจำลองแสดงสัญญาณไฟจราจร	6
2.5	แสดงจอภาพแสดงข้อมูลการทำงานของอุปกรณ์ควบคุมสัญญาณไฟจราจร	7
2.6	แสดงรูปแบบสัญญาณไฟจราจรแบบที่ 1	7
2.7	แสดงรูปแบบสัญญาณไฟจราจรแบบที่ 2	8
2.8	แสดงรูปแบบสัญญาณไฟจราจรแบบที่ 3	8
2.9	แสดงรูปแบบสัญญาณไฟจราจรแบบที่ 4	9
2.10	แสดงรูปแบบสัญญาณไฟจราจรแบบที่ 5	9
2.11	แสดงรูปแบบสัญญาณไฟจราจรแบบที่ 6	9
2.12	แสดงรูปแบบสัญญาณไฟจราจรแบบที่ 7	10
2.13	แสดงรูปแบบสัญญาณไฟจราจรแบบที่ 8	10
2.14	แสดงภาพอุปกรณ์รับข้อมูล	11
3.1	แสดงภาพไฟจราจรบริเวณ 4 แยกซึ่งประกอบด้วยไฟตรงและไฟเลี้ยวขวา	12
3.2	แสดงการต่อเชื่อมสัญญาณจากตัวควบคุมหลักไปยังตัวควบคุมสัญญาณในแยกต่าง ๆ	13
3.3	แสดงบล็อกไดอะแกรมของส่วนควบคุมและประมวลผลของเครื่องควบคุมสัญญาณไฟจราจร	15
3.4	แสดงวงจรของเครื่องควบคุมสัญญาณไฟจราจร	16
3.5	แสดงรูปแบบการต่อพอร์ตเข้ากับสวิทช์ของคีย์รับข้อมูล	17
3.6	แสดงแผนภาพลำดับการทำงานของโปรแกรมหลักของเครื่องควบคุมไฟจราจร	18
3.7	แสดงแผนภาพลำดับการทำงานของโปรแกรมตรวจสอบคีย์	19
3.8	แสดงแผนภาพการทำงานของโปรแกรมเชื่อมต่อข้อมูล	20
3.9	แสดงแผนภาพการทำงานของโปรแกรมเชื่อมต่อเวลา	21
4.1	แสดงบล็อกของส่วนที่ใช้ในการทดลองส่วนควบคุมและประมวลผล	23
4.2	แสดงบล็อกส่วนของวงจรในการทดลองทดสอบการทำงานของ 8255	24
4.3	แสดงการเชื่อมต่อระหว่าง 8255 กับอุปกรณ์แสดงผลสัญญาณไฟจราจรจำลอง	25
4.4	แสดงการเชื่อมต่ออุปกรณ์แสดงผล LCD เข้ากับพอร์ตของ 8255	29
4.5	แสดงภาพผลการทดลองอุปกรณ์แสดงผล LCD	32

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ในชื่อของอาจารย์ ดร.สุวิทย์ งามชื่นงาม ไม่อนุญาตให้นำไปใช้ประโยชน์ในการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6	แสดงภาพผลการทดลองอุปกรณ์แสดงผล LCD	33
4.7	แสดงภาพผลการทดลองอุปกรณ์แสดงผล LCD	33
4.8	แสดงจอภาพ LCD เมื่อทำการทดลองโปรแกรมทดสอบ RAM	36
4.9	แสดงภาพจอ LCD หลังจากทำการทดลองโปรแกรมทดสอบนาฬิกา	39
4.10	แสดงลักษณะของสัญญาณที่ส่งออกจากตัวควบคุมของเครื่องส่ง	50
4.11	แสดงลักษณะของสัญญาณที่รับได้ในตัวควบคุมของเครื่องรับ	51
4.12	แสดงลักษณะสัญญาณที่ส่งผ่านอุปกรณ์อินเทอร์เฟซ RS-485	51



บทที่ 1

บทนำ

เนื่องจากในปัจจุบันนี้ ปัญหาการจราจรนับว่าเป็นปัญหาที่ต้องแก้ไขอย่างเร่งด่วนที่สุดเนื่องจากว่าจะทวีความรุนแรงมากขึ้นทุกวัน ระบบการควบคุมสัญญาณจราจรที่มีประสิทธิภาพจึงเป็นสิ่งจำเป็นอย่างยิ่งที่จะช่วยบรรเทาปัญหาการจราจรที่เป็นอยู่ในทุกวันนี้ให้บรรเทาลง ซึ่งสัญญาณไฟจราจรที่ใช้กันอยู่ในปัจจุบันนี้เป็นระบบขนาดเล็กที่ทำงานควบคุมได้แค่เพียงแยก ๆ เดียว โดยใช้ตำรวจจราจรคอยควบคุมหรือทำงานตามโปรแกรมที่ตั้งไว้แล้วโดยอัตโนมัติ ซึ่งปัญหาที่เกิดขึ้นนั้นก็คือจะสามารถควบคุมการจราจรได้เฉพาะแยกใดแยกหนึ่งเท่านั้น เมื่อรถผ่านจากแยกหนึ่งไปยังอีกแยกหนึ่งแล้ว ก็เกิดการติดขัดเนื่องจากสัญญาณไฟในแยกถัดไปเนื่องจากว่าสัญญาณไฟทำงานโดยไม่สอดคล้องกัน ซึ่งในการที่จะแก้ปัญหาเหล่านี้ นั้น เราสามารถนำระบบการควบคุมสัญญาณจราจรขนาดใหญ่ที่ใช้คอมพิวเตอร์เป็นตัวควบคุมการทำงานในอาณาบริเวณพื้นที่หนึ่ง ๆ มาใช้ โดยระบบแบบนี้ที่มีใช้ในปัจจุบันนั้น พบว่าสามารถแก้ไขการจราจรได้ในจุดหนึ่ง แต่ยังมีอุปสรรคปัญหาในด้านที่อุปกรณ์ของระบบขนาดใหญ่ยังไม่สามารถผลิตใช้งานขึ้นได้เองภายในประเทศ ดังนั้นจึงจำเป็นที่จะต้องส่งอุปกรณ์ที่จำเป็นที่จะต้องใช้ภายในระบบเข้ามาจากต่างประเทศซึ่งอุปกรณ์เหล่านี้จะมีมูลค่าสูงมาก รวมถึงปัญหาในการดูแลบำรุงรักษาซ่อมแซม และในการควบคุมระบบเหล่านี้ก็ยังจำเป็นต้องใช้บุคลากรที่มีความรู้และเชี่ยวชาญในด้านนี้โดยเฉพาะ เช่น ผู้ที่มีความรู้หรือเชี่ยวชาญในด้านวิศวกรรมจราจร (Traffic Engineer) ซึ่งในการที่จะสร้างบุคลากรที่มีความรู้ในด้านนี้ขึ้นมาอย่างเร่งด่วนนั้นไม่สามารถเป็นไปได้ รวมถึงว่าในระบบขนาดใหญ่ที่มีอยู่นี้จะมีส่วนของระบบซึ่งสามารถตรวจสอบปริมาณรถที่เข้ามาในแต่ละสี่แยกได้ โดยจะใช้ระบบเซ็นเซอร์ที่ฝังอยู่บนพื้นผิวถนน ซึ่งมีความยุ่งยากมากในการติดตั้งระบบ

ระบบควบคุมสัญญาณไฟจราจรที่ประดิษฐ์ขึ้นมาใหม่ มีวัตถุประสงค์ในการใช้งานหลักเพื่อให้ทำงานได้มีประสิทธิภาพกว่าระบบขนาดเล็กที่ใช้อยู่ตามแต่ละแยกอย่างในปัจจุบัน ซึ่งถึงแม้ว่าเครื่องควบคุมสัญญาณไฟจราจรที่ประดิษฐ์ขึ้นมาใหม่จะมีประสิทธิภาพจะไม่ดีเทียบเท่ากับระบบขนาดใหญ่ที่ใช้งานอยู่แล้ว แต่ก็ช่วยลดปัญหาที่เกิดจากระบบขนาดเล็กที่มีใช้อยู่มากมายในปัจจุบันได้บ้าง เนื่องจากว่าอุปกรณ์ที่ประดิษฐ์ขึ้นมาใหม่นี้จะมีระบบการทำงานที่เพิ่มเติมขึ้นมาจากระบบขนาดเล็ก โดยสามารถทำการควบคุมปริมาณการจราจร จากที่สามารถควบคุมได้แค่เพียงแยกใดแยกหนึ่งให้สามารถควบคุมการจราจรได้ต่อเนื่องกันไปยังแยกถัดไปข้างหน้า เพื่อช่วยลดปัญหาการจราจรแออัดที่จะเกิดขึ้นในการที่สามารถควบคุมได้แค่เพียงแยกเดียว ช่วยบรรเทาปัญหาในกรณีทีอาจจะต้องนำระบบควบคุมขนาดใหญ่เข้ามาช่วยแก้ไข ปัญหา เนื่องจากอุปกรณ์ที่ผลิตขึ้นมาใหม่นี้ เราสามารถผลิตได้โดยใช้อุปกรณ์อิเล็กทรอนิกส์ง่าย ๆ เช่นใช้ไมโครโปรเซสเซอร์ชิปเดียวเป็นตัวควบคุมการทำงานแทนที่จะใช้อุปกรณ์คอมพิวเตอร์ ซึ่งอุปกรณ์ไมโครโปรเซสเซอร์ที่นำมาใช้งานนี้มีราคาถูกและสามารถหาได้โดยทั่วไป รวมถึงวิธีการใช้งานของอุปกรณ์ที่ทำขึ้นมาใหม่ยังสามารถใช้งานได้ง่ายโดยไม่จำเป็นต้องเป็นผู้เชี่ยวชาญในระบบหรืออุปกรณ์นั้น ๆ มาเป็นผู้ควบคุม ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์ โดยในอุปกรณ์ควบคุมสัญญาณไฟจราจรที่ประดิษฐ์ขึ้นมาใหม่นี้จะมีคุณสมบัติคือมีระบบการทำงานเหมือนกับระบบควบคุมไฟจราจรที่มีใช้อยู่ทั่วไป แต่จะสามารถเชื่อมต่อไปยังแยกต่าง ๆ โดยใช้การอินเทอร์เฟซสัญญาณจากอุปกรณ์ควบคุมหลักไปยังเครื่องควบคุมที่ตั้งอยู่ยังแยกถัดไปข้างหน้า ซึ่งอุปกรณ์ที่นำเข้ามาใช้คืออุปกรณ์อินเทอร์เฟซ RS-485 ซึ่งมีข้อดีคือสามารถส่งสัญญาณโดยใช้การส่งข้อมูลแบบอนุกรมทำการส่งโดยใช้กระแส (current) รวมทั้งใช้สัญญาณนาฬิกาเข้ามาช่วยควบคุมระบบเวลาให้อุปกรณ์สามารถทำงานได้ตามเวลาที่เราสามารถตั้งโปรแกรมเอาไว้ ในกรณีที่ไม่มีคนคอยควบคุมอุปกรณ์ได้ในหลายรูปแบบ เช่น สามารถตั้งโปรแกรมการทำงานหนึ่งในช่วงเวลาหนึ่ง และสามารถเปลี่ยนไปใช้โปรแกรมอื่นได้ในเวลาที่เปลี่ยนไป เนื่องจากสภาพการจราจรที่เกิดขึ้นในแต่ละช่วงเวลามีรูปแบบไม่เหมือนกัน โดยค่าเวลาสามารถแสดงผลได้ทางจอภาพแสดงผลแบบ LCD



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

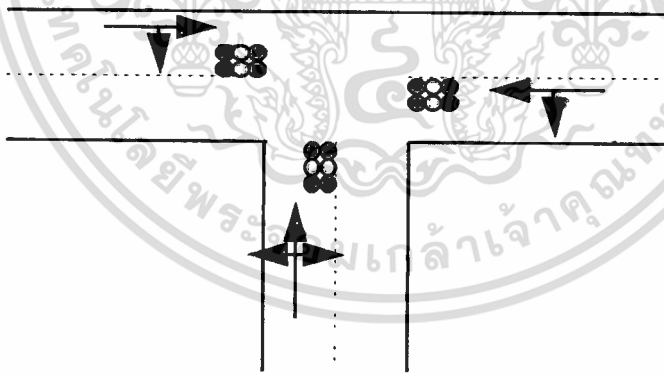
บทที่ 2 ทฤษฎีและหลักการ

2.1 ทฤษฎีและหลักการทํางานของสัญญาณไฟจราจร

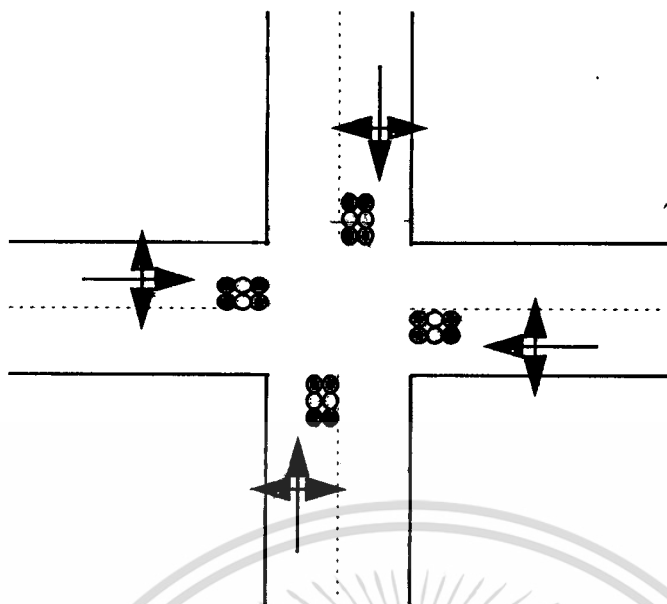
สำหรับไฟจราจรโดยทั่วไปนั้น จะมีสัญญาณไฟอยู่ทั้งหมด 3 ชนิด ได้แก่ สัญญาณไฟแดง ไฟเหลือง และไฟเขียว ซึ่งสัญญาณไฟทั้ง 3 สีนั้น จะสื่อความหมายภายในตัวของมันคือ

- สัญญาณไฟแดง หมายถึง ให้หยุดรถ
- สัญญาณไฟเหลือง หมายถึง ให้ชะลอความเร็ว
- สัญญาณไฟเขียว หมายถึง ให้รถไปได้

ซึ่งในลักษณะของสัญญาณไฟที่ทํางานในแต่ละแยกจะมีรูปแบบสัญญาณไฟในแต่ละแยกไม่เหมือนกัน ซึ่งในบางครั้ง จุดตัดของถนนจะเป็น 3 แยกหรือในบางครั้งอาจจะเป็น 4 แยกหรือ 5 แยกก็ได้ ดังนั้น รูปแบบในการเปิดสัญญาณไฟนั้น จะต้องเปลี่ยนแปลงไปตามสภาพของจุดตัดของถนน



รูปที่ 2.1 แสดงรูปแบบของสัญญาณไฟจราจรทั้งหมดที่เป็นไปได้ใน 3 แยก



รูปที่ 2.2 แสดงรูปแบบสัญญาณไฟจราจรทั้งหมดที่เป็นไปได้ใน 4 แยก

จากลักษณะของแยกที่แตกต่างกัน จึงมีผลให้ลักษณะรูปแบบการเปิดสัญญาณไฟในแต่ละแยกแตกต่างกันไปด้วย

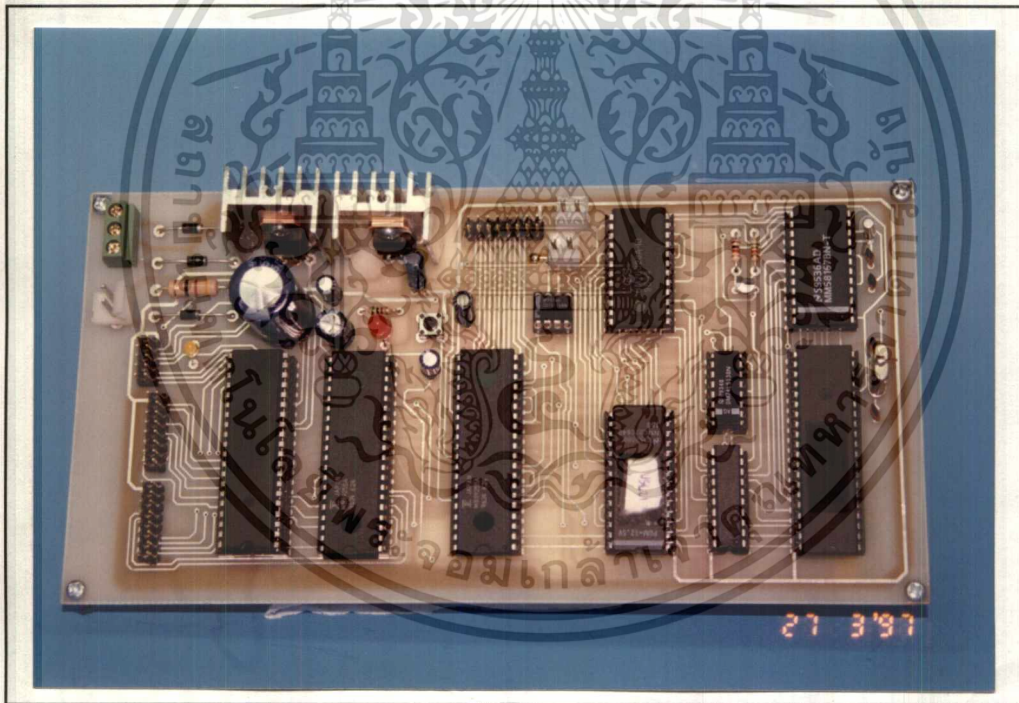
2.2 ทฤษฎีและหลักการทำงานโดยทั่วไปของอุปกรณ์ควบคุมสัญญาณไฟจราจร

อุปกรณ์ควบคุมสัญญาณจราจรส่วนใหญ่ที่ใช้อยู่ในปัจจุบันโดยทั่วไปนั้น จะเป็นระบบควบคุมขนาดเล็กซึ่งจะสามารถควบคุมสัญญาณไฟจราจรได้เฉพาะแค่แยกใดแยกหนึ่งเท่านั้น โดยอุปกรณ์ควบคุมจะทำหน้าที่ควบคุมจังหวะการเปิดปิดไฟสัญญาณจราจรให้เปิดปิดตามที่ตั้งไว้ตามโปรแกรมโดยอัตโนมัติหรือเปิดปิดตามที่บุคคลผู้ควบคุมต้องการ แต่อุปกรณ์ควบคุมสัญญาณไฟจราจรชุดที่ได้ทำการประดิษฐ์ขึ้นนี้นั้น จะพัฒนาจากอุปกรณ์ควบคุมระบบแบบเดิมให้มีประสิทธิภาพสูงขึ้นคือ จะเพิ่มส่วนซึ่งสามารถทำการควบคุมเชื่อมต่อไปยังแยกถัดไปข้างหน้าได้ โดยมีอุปกรณ์ควบคุมหลัก 1 ตัว (Master Controller) เป็นตัวคอยควบคุมโดยส่งสัญญาณควบคุมจากชุดควบคุมผ่านอุปกรณ์อินเทอร์เฟซ RS-485 ไปยังอุปกรณ์ควบคุมซึ่งอยู่ยังแยกถัดไปข้างหน้า (Slave) ในขณะที่ตัวอุปกรณ์เองนั้นยังสามารถทำงานได้เหมือนที่ในระบบเดิมซึ่งอุปกรณ์ควบคุมสัญญาณจราจรที่ใช้อยู่ในปัจจุบันทำได้อยู่ นอกจากนี้อุปกรณ์นี้ยังสามารถแสดงผลให้ผู้ควบคุมรู้ว่า ขณะนี้สัญญาณทำงานในโหมดโดยอยู่ แสดงถึงวัน-เวลาได้ และผู้ควบคุมยังสามารถตั้งช่วงเวลาของการเปิด-ปิดสัญญาณไฟ เพื่อให้ทำงานสอดคล้องกับสภาพการจราจรได้ในขณะที่ทำงานอยู่ในโหมดอัตโนมัติ ซึ่งภายในอุปกรณ์ควบคุมสัญญาณไฟจราจรนั้น เราได้มีการแบ่งส่วนประกอบต่าง ๆ ที่ใช้ควบคุมการทำงานออกเป็นดังต่อไปนี้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ส่วนควบคุมและประมวลผล (Controlling and Processing Unit)
2. ส่วนแสดงผล (Display Unit)
3. ส่วนคีย์รับข้อมูล (Input Unit)

สำหรับโปรแกรมที่ใช้ในการควบคุมการทำงานของอุปกรณ์นั้น ได้ทำการกำหนดรูปแบบเฉพาะไว้แล้ว โดยผู้ใช้สามารถเลือกใช้ได้อย่างง่ายตามรูปแบบโดยการควบคุมผ่านสวิตช์ควบคุมเท่านั้น เช่น เมื่อต้องการให้เปิดไฟเขียวที่แยกหนึ่ง เราก็ไปกดสวิตช์ที่ควบคุมไฟเขียวที่แยกนั้น หรือต้องการที่จะโปรแกรมการทำงานให้สัญญาณไฟทำงานอย่างต่อเนื่องโดยอัตโนมัติ ก็สามารถที่จะทำการบันทึกไว้ได้ตามลำดับ เช่น ให้เปิดไฟเขียวที่แยกนี้ก่อน แล้วหลังจากนั้นเป็นระยะเวลาหนึ่งก็เปลี่ยนไฟเขียวไปอีกด้านหนึ่ง แล้วให้ไฟแดงแก่ด้านนี้แทน เป็นต้น ซึ่งในส่วนอุปกรณ์ที่ใช้ควบคุมและประมวลผลนั้น ได้นำอุปกรณ์ไมโครโปรเซสเซอร์เบอร์ 8031 มาเป็นตัวควบคุม และในส่วนของการแสดงผลก็นำอุปกรณ์แสดงผลคืออุปกรณ์ Display LCD มาใช้



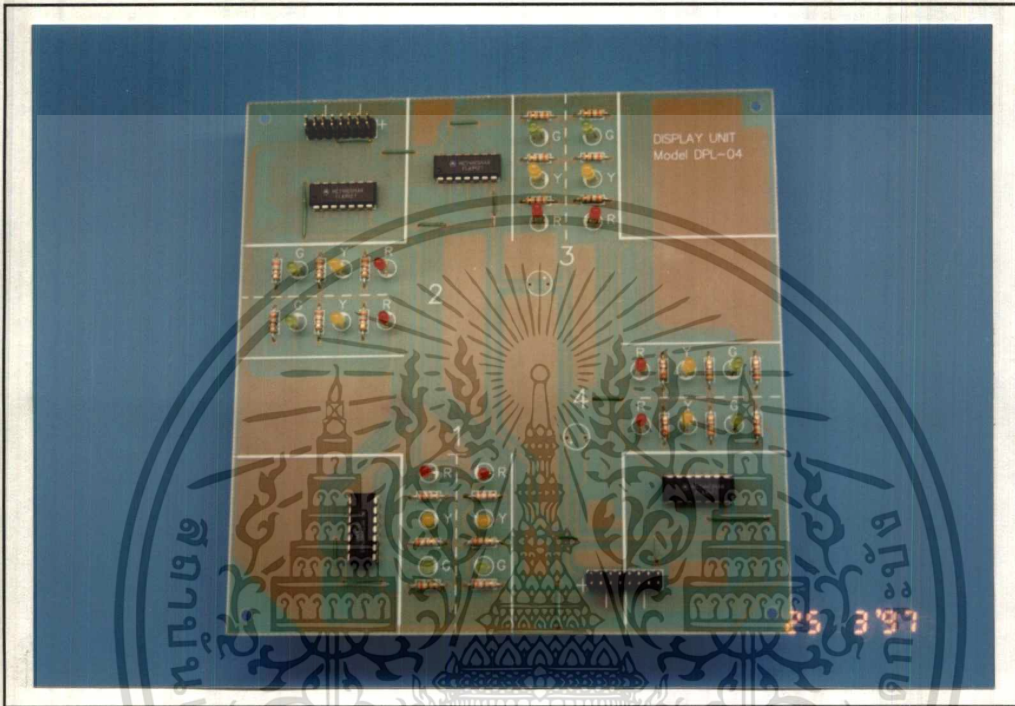
รูปที่ 2.3 แสดงภาพอุปกรณ์ในส่วนควบคุมและประมวลผล

สำหรับในส่วนแสดงผลของอุปกรณ์ควบคุมสัญญาณไฟจราจรนั้น สามารถจำแนกอุปกรณ์ที่ใช้ในการแสดงผลได้ออกเป็น 2 ชิ้น คือ

1. บอร์ดจำลองแสดงสัญญาณไฟจราจร (Traffic Light Display Board)
2. จอแสดงข้อมูลการทำงานแบบแอลซีดี (LCD Data Monitor)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับบอร์ดจำลองแสดงสัญญาณไฟจราจรนั้น ประกอบขึ้นโดยจำลองแบบจากลักษณะของสี่แยกไฟเขียว-ไฟแดง ซึ่งแต่ละแยกจะประกอบด้วยสัญญาณไฟจราจร 2 ชุด ได้แก่ไฟตรงและไฟเลี้ยวขวา รวมทั้งหมด 4 แยก จะประกอบด้วยสัญญาณไฟทั้งหมด 8 ชุด จำนวนไฟทั้งหมด 24 ดวง ซึ่งภาพแสดงบอร์ดจำลองแสดงสัญญาณไฟจราจร จะเป็นไปดังภาพ



รูปที่ 2.4 แสดงภาพบอร์ดจำลองแสดงสัญญาณไฟจราจร

และสำหรับส่วนจอแสดงข้อมูลการทำงานแบบแอลซีดีนั้น เราได้เลือกใช้อุปกรณ์ LCD Module ซึ่งเป็นอุปกรณ์สำเร็จรูปเนื่องจากว่ามีความสวยงามในการแสดงผลและยังสามารถเขียนโปรแกรมควบคุมให้ทำงานเป็นจอแสดงข้อมูลต่าง ๆ ได้ เทียบกับการใช้อุปกรณ์ Seven Segment ซึ่งจะสามารถแสดงได้แต่ค่าเวลาของนาฬิกาซึ่งอาจจะมีการเขียนโปรแกรมควบคุมได้ง่ายกว่าแต่ไม่สามารถแสดงข้อมูลที่เป็นตัวอักษรเพื่ออธิบายโหมดการทำงานปัจจุบันของอุปกรณ์ควบคุมได้ ซึ่งเนื่องจากอุปกรณ์ LCD Module เป็นอุปกรณ์สำเร็จรูป จึงสามารถหาได้ง่ายทั่วไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

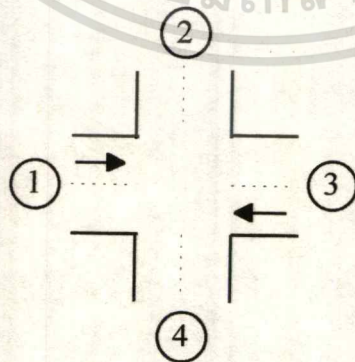


รูปที่ 2.5 แสดงจอภาพแสดงข้อมูลการทำงานของอุปกรณ์ควบคุมสัญญาณไฟจราจร

สำหรับในส่วนนี้รับข้อมูลของอุปกรณ์ควบคุมสัญญาณไฟจราจรนั้น จะประกอบด้วยสวิทช์ต่าง ๆ ทั้งหมด 12 ปุ่ม ซึ่งจะมีหน้าที่การทำงานแตกต่างกันไปตามการควบคุม โดยในรายละเอียดของสวิทช์ทั้ง 12 ปุ่มนั้นจะประกอบด้วยดังต่อไปนี้

1. สวิทช์ให้สัญญาณไฟจราจรแบบที่ 1 (Manual 1 Switch)

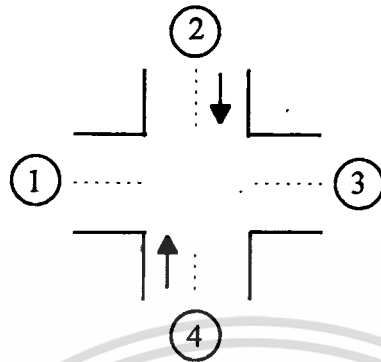
เมื่อกดสวิทช์นี้ สัญญาณไฟจราจรที่ได้จะเป็นไปดังภาพ



รูปที่ 2.6 แสดงรูปแบบสัญญาณไฟจราจรแบบที่ 1

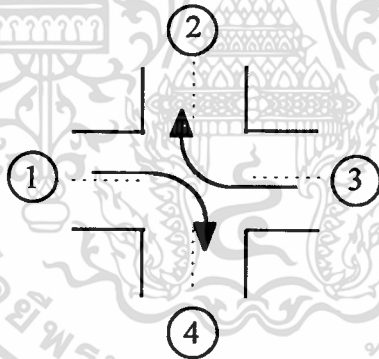
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. สวิตช์ให้สัญญาณไฟจราจรแบบที่ 2 (Manual 2 Switch)
เมื่อกดสวิตช์นี้ สัญญาณไฟจราจรที่ได้จะเป็นไปดังภาพ



รูปที่ 2.7 แสดงรูปแบบสัญญาณไฟจราจรแบบที่ 2

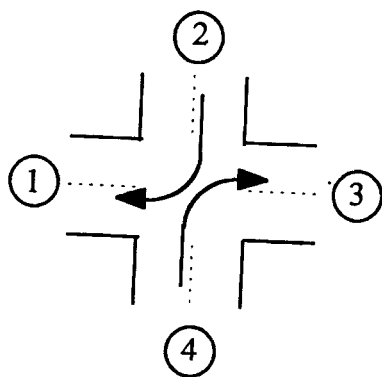
3. สวิตช์ให้สัญญาณไฟจราจรแบบที่ 3 (Manual 3 Switch)
เมื่อกดสวิตช์นี้ สัญญาณไฟจราจรที่ได้จะเป็นไปดังภาพ



รูปที่ 2.8 แสดงรูปแบบสัญญาณไฟจราจรแบบที่ 3

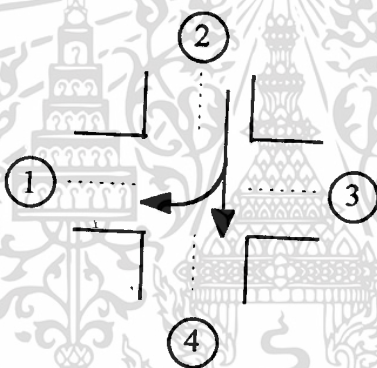
4. สวิตช์ให้สัญญาณไฟจราจรแบบที่ 4 (Manual 4 Switch)
เมื่อกดสวิตช์นี้ สัญญาณไฟจราจรที่ได้จะเป็นไปดังภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



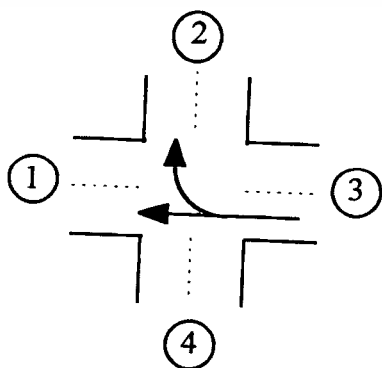
รูปที่ 2.9 แสดงรูปแบบสัญญาณไฟจราจรแบบที่ 4

- 5. สวิตช์ให้สัญญาณไฟจราจรแบบที่ 5 (Manual 5 Switch)
เมื่อกดสวิตช์นี้ สัญญาณไฟจราจรที่ได้จะเป็นไปดังภาพ



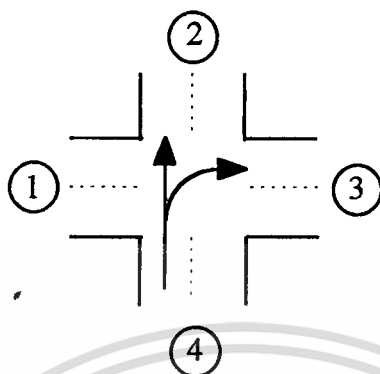
รูปที่ 2.10 แสดงรูปแบบสัญญาณไฟจราจรแบบที่ 5

- 6. สวิตช์ให้สัญญาณไฟจราจรแบบที่ 6 (Manual 6 Switch)
เมื่อกดสวิตช์นี้ สัญญาณไฟจราจรที่ได้จะเป็นไปดังภาพ



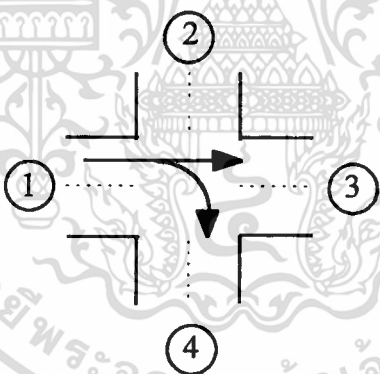
เอกสารนี้เป็นเอกสารที่มอบไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 รูปที่ 2.11 แสดงรูปแบบสัญญาณไฟจราจรแบบที่ 6
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องขออนุญาตไปยังเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. สวิตช์ให้สัญญาณไฟจราจรแบบที่ 7 (Manual 7 Switch)
เมื่อกดสวิตช์นี้ สัญญาณไฟจราจรที่ได้จะเป็นไปดังภาพ



รูปที่ 2.12 แสดงรูปแบบสัญญาณไฟจราจรแบบที่ 7.

8. สวิตช์ให้สัญญาณไฟจราจรแบบที่ 8 (Manual 8 Switch)
เมื่อกดสวิตช์นี้ สัญญาณไฟจราจรที่ได้จะเป็นไปดังภาพ



รูปที่ 2.13 แสดงรูปแบบสัญญาณไฟจราจรแบบที่ 8

สำหรับสวิตช์ให้สัญญาณไฟจราจรทั้ง 8 แบบบนนี้นั้น จะสามารถทำงานได้ เมื่อโหมดการทำงาน
ของอุปกรณ์ควบคุมสัญญาณอัตโนมัติทำงานอยู่ในโหมดควบคุมด้วยมือ (Manual Mode) เท่านั้น ถ้าอยู่ใน
โหมดการทำงานโดยอัตโนมัติ สวิตช์ทั้ง 8 นี้จะถูกล็อกโดยโปรแกรม คือ เมื่อกดก็จะมีอะไรเกิดขึ้น

9. สวิตช์ + (+ Switch)

ใช้สำหรับเพิ่มค่าข้อมูลที่ทำกรตั้ง โดยขึ้นอยู่กับการทำงานของโปรแกรมว่าขณะนั้นกำลัง
ทำงานอยู่ในโหมดใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. สวิทช์ - (- Switch)

ใช้สำหรับลดค่าข้อมูลที่ทำการตั้ง โดยขึ้นอยู่กับการทำงานของโปรแกรมว่าขณะนั้นกำลังทำงานอยู่ในโหมดใด

11. สวิทช์โหมด (Mode Switch)

ใช้สำหรับเลือกโหมดที่จะทำการติดตั้งค่า โดยโหมดที่สามารถเลือกได้นั้นมีอยู่ทั้งหมด 6 โหมดการทำงาน ได้แก่

- 1) โหมดตั้งค่าวันและเวลาของนาฬิกา (Set Time Mode)
- 2) โหมดตั้งรูปแบบการทำงานต่าง ๆ ของโปรแกรมอัตโนมัติ (Set Auto Mode)
- 3) โหมดตั้งระยะเวลาการทำงานของสัญญาณไฟแต่ละรูปแบบ (Set Time of each Manual)
- 4) โหมดตั้งการเปิด-ปิดการทำงานของโหมดอัตโนมัติและเลือกโหมดอัตโนมัติ (Set Auto ON/OFF)
- 5) โหมดตั้งการเปิด - ปิดการทำงานต่อเนื่องกันระหว่างชุดควบคุม (Set Link ON/OFF)
- 6) โหมดตั้งรูปแบบโปรแกรมอัตโนมัติของชุดควบคุมตัวถัด ๆ ไป (Set Slave)

12. สวิทช์คอบตกลาง (Enter Switch)

ใช้สำหรับเก็บค่าข้อมูลที่ได้ทำการตั้งค่าเอาไว้แล้ว เข้าสู่หน่วยความจำ เพื่อที่จะได้นำข้อมูลที่ทำการตั้งค่านั้นไปทำงานต่อไป

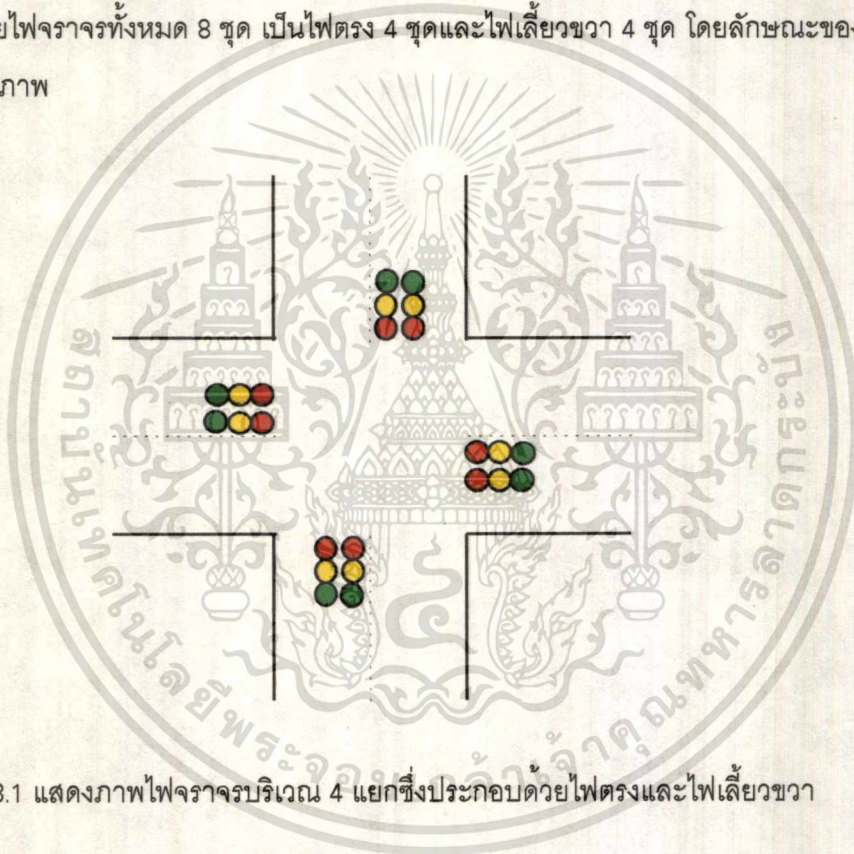


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 2.14 แสดงภาพอุปกรณ์รับข้อมูล ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
ขั้นตอนการออกแบบบลิคโคอะแกรม
และวงจรการทำงานของเครื่องควบคุมสัญญาณไฟจราจร

3.1 ลักษณะการทำงานของชุดควบคุมสัญญาณไฟจราจร

ในการออกแบบเครื่องควบคุมสัญญาณไฟจราจรนั้น เราจะทำการออกแบบเครื่องควบคุมสำหรับใช้ควบคุมสัญญาณไฟจราจร ณ. สี่แยกซึ่งมีขนาด 4 เลน 4 ช่องการจราจรไป-กลับ สัญญาณไฟจราจรที่ใช้นั้นแต่ละแยกนั้นจะแบ่งเป็น 2 ชุด โดยมีชุดของไฟตรง 1 ชุด และชุดของไฟเลี้ยว 1 ชุด รวมทั้ง 4 แยกทั้งหมดจะประกอบด้วยไฟจราจรทั้งหมด 8 ชุด เป็นไฟตรง 4 ชุดและไฟเลี้ยวขวา 4 ชุด โดยลักษณะของไฟจราจรที่สี่แยกจะเป็นไปดังภาพ



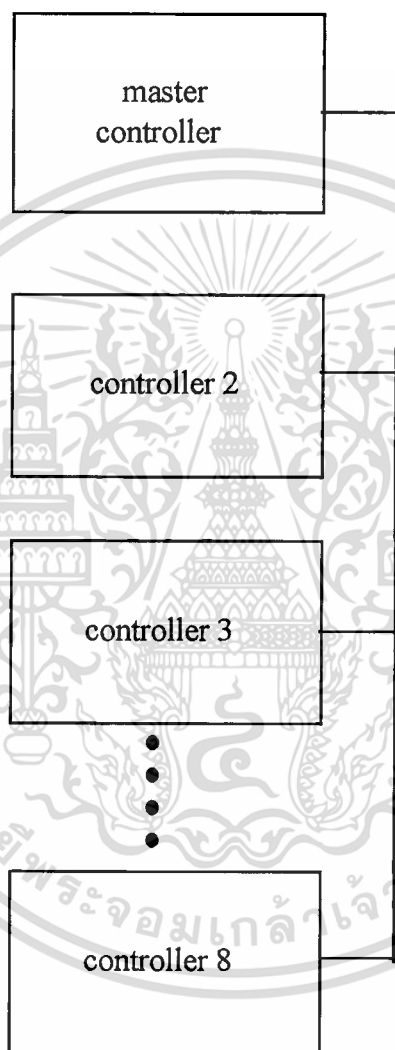
รูปที่ 3.1 แสดงภาพไฟจราจรบริเวณ 4 แยกซึ่งประกอบด้วยไฟตรงและไฟเลี้ยวขวา

สำหรับการทำงานของชุดควบคุมนั้น จะแบ่งการทำงานออกเป็นได้ทั้งหมด 2 แบบ คือ

1. การทำงานควบคุมสัญญาณไฟตามลำพัง
2. การทำงานควบคุมสัญญาณไฟอย่างต่อเนื่อง โดยสามารถส่งสัญญาณควบคุมไปยังชุดควบคุมชุดถัดไป เพื่อให้เกิดความต่อเนื่องของสัญญาณ

สำหรับการทำงานควบคุมสัญญาณไฟตามลำพังนั้น อุปกรณ์ควบคุมจะทำการควบคุมสัญญาณไฟจราจรโดยทำงานตามบุคคลผู้ควบคุมเครื่องหรือตามโปรแกรมที่ได้ทำการติดตั้งเอาไว้ก่อนแล้ว โดยการเปิดปิดสัญญาณไฟนั้น จะทำงานอย่างอิสระ ไม่มีความสัมพันธ์ใด ๆ กับชุดควบคุมชุดอื่น แต่สำหรับการทำงานควบคุมสัญญาณไฟอย่างต่อเนื่องนั้น ชุดควบคุมตัวหลักจะทำหน้าที่ส่งสัญญาณเพื่อไปควบคุมชุดควบคุมชุดไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ควบคุมสัญญาณไฟอย่างต่อเนื่องนั้น ชุดควบคุมตัวหลักจะทำหน้าที่ส่งสัญญาณเพื่อไปควบคุมชุดควบคุมชุดอื่นให้เปิดสัญญาณที่ทำงานต่อเนื่องจากชุดควบคุมตัวหลักโดยจะทำการเปิดสัญญาณตามโปรแกรมที่สามารถตั้งได้ในชุดควบคุมหลัก โดยในการทำงานนั้นเครื่องควบคุมจะส่งสัญญาณเชื่อมไปยังแยกข้างหน้าถัดไปให้เปิดสัญญาณที่ตั้งเอาไว้เพื่อให้เกิดสัญญาณไฟที่มีความต่อเนื่องระหว่างแยก โดยมีลักษณะรูปแบบของการต่อเชื่อมของชุดควบคุมที่อยู่บริเวณสี่แยกเป็นไปตามรูป



รูปที่ 3.2 แสดงการต่อเชื่อมสัญญาณจากตัวควบคุมหลักไปยังตัวควบคุมสัญญาณในแยกต่าง ๆ

3.2 การออกแบบในส่วนของหน่วยควบคุมและประมวลผลของชุดควบคุมสัญญาณไฟจราจร

สำหรับในส่วนของอุปกรณ์ควบคุมสัญญาณไฟจราจรนั้น สามารถทำงานได้ภายในตัวของมันเองอย่างอิสระ ดังนั้นในส่วนของหน่วยควบคุมและประมวลผลที่ทำการออกแบบนั้น ตามที่ได้กล่าวมาแล้วจะประกอบด้วยส่วนต่าง ๆ ซึ่งจะทำงานในลักษณะเกี่ยวข้องกันทั้งหมด โดยแบ่งเป็นส่วนต่าง ๆ ดังต่อไปนี้ ได้แก่

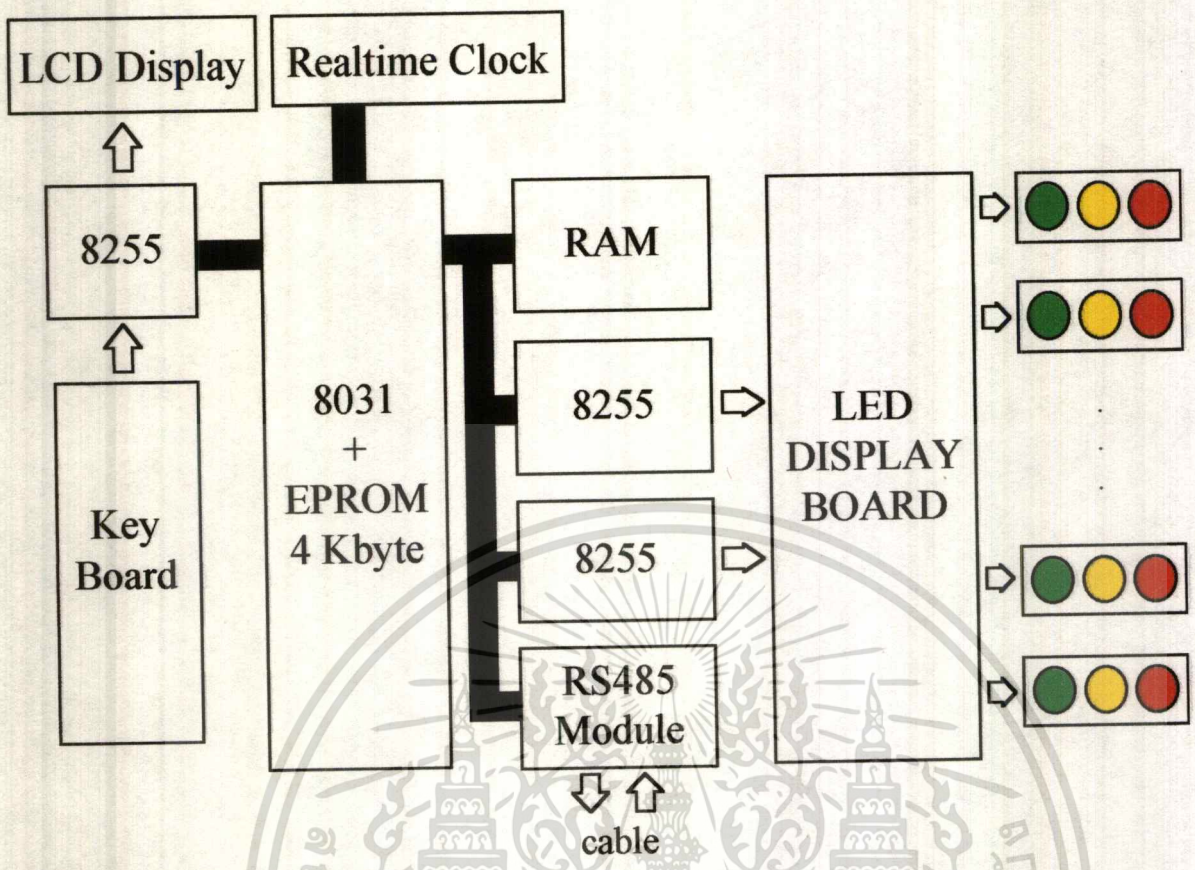
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ส่วนที่ทำการควบคุมสัญญาณไฟจราจร ซึ่งประกอบด้วย ส่วนของ CPU ซึ่งได้แก่ 8031+ EPROM, 8255 จำนวน 2 ตัวเป็นพอร์ทที่เข้าออกของข้อมูลของไฟจราจร, Key and Control Switch เป็นส่วนที่ใช้ควบคุมสัญญาณไฟจราจร โดยทั้งหมดจะเป็นชุดที่ทำหน้าที่ควบคุมการเปิดและปิดของสัญญาณไฟจราจร แต่ละดวงในแต่ละด้านของ 4 แยกว่าควรจะมีไฟใดขึ้นที่แยกด้านใด โดยจะมีตัว EPROM เป็นตัวเก็บโปรแกรมที่เขียนไว้เพื่อสั่งงานให้ CPU ทำงานตามขั้นตอนของโปรแกรมที่ตั้งไว้ให้ทำงานตามที่เราต้องการ

2. ส่วนที่เป็นนาฬิกา ประกอบด้วย Realtime IC, LCD Display, CPU ซึ่งใช้ 8031 ตัวเดียวกับที่ใช้ควบคุมสัญญาณไฟจราจรนั่นเอง โดยสัญญาณที่ได้จากส่วนนาฬิกานั้นเราจะนำมาเป็นตัวอ้างอิงในการเปิด-ปิดสัญญาณไฟจราจร เช่น ต้องการให้เปิดไฟแดงในแต่ละแยกนานเท่าใด ไฟเหลืองนานเท่าใด ไฟเขียวนานเท่าใดในช่วงเวลาใด หรือไม่ก็ใช้ควบคุมลักษณะการทำงานของไฟจราจรตามโปรแกรมอัตโนมัติที่ตั้งไว้ตามวันและเวลาต่าง ๆ ซึ่งอาจจะต่างกันไป เช่น ในวันอาทิตย์อาจจะมีการจราจรไม่หนาแน่นมากจึงไม่จำเป็นต้องเปิดไฟเขียวในแต่ละแยกนาน แต่ถ้าเป็นวันอื่น ๆ ในช่วงเวลาเร่งรีบซึ่งอาจจะมีปริมาณรถหนาแน่นมากในแต่ละแยก ก็จำเป็นต้องเปิดสัญญาณไฟเขียวให้นานมากขึ้นเพื่อระบายรถไปให้ได้มากที่สุด

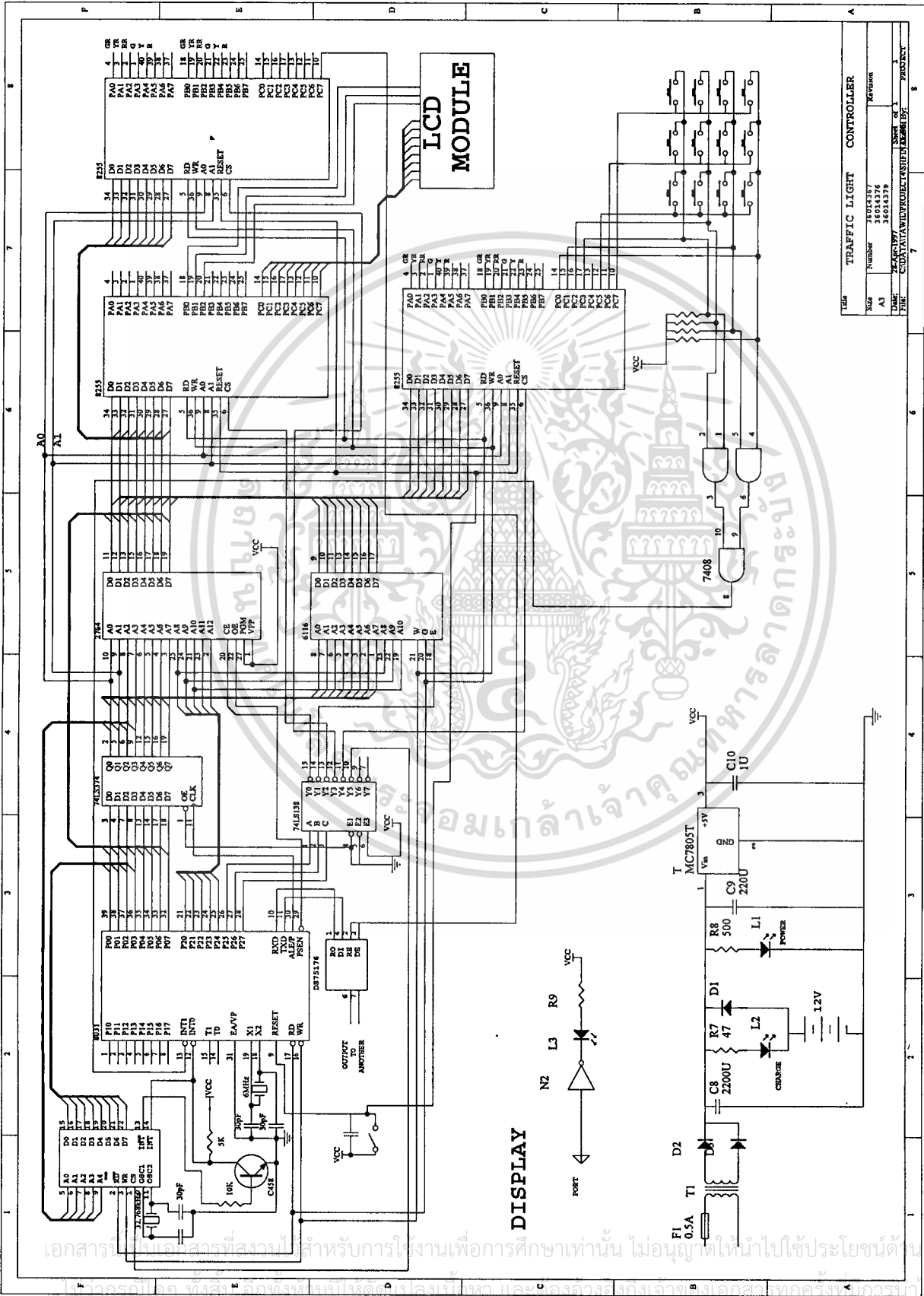
3. ส่วนที่เป็นตัวเชื่อมสัญญาณไฟจราจรจากชุดควบคุมชุดหนึ่ง ไปยังอีกชุดควบคุมอีกชุดหนึ่ง ประกอบด้วยส่วน CPU, RS485 Module โดยอาจจะทำหน้าที่เป็นตัวส่งสัญญาณไปยังตัวควบคุมชุดต่าง ๆ (ในส่วนของตัวควบคุมหลัก) หรือทำหน้าที่เป็นตัวรับสัญญาณที่ส่งมาจากตัวควบคุมหลักเพื่อทำงานตามที่ตัวควบคุมหลักส่งสัญญาณมาเพื่อให้ปล่อยรถต่อเนื่องกันไปเรื่อย ๆ

จากส่วนประกอบต่าง ๆ ของชุดควบคุมที่กล่าวมาแล้วนั้น เราสามารถนำมาสร้าง Block Diagram ซึ่งจะแสดงความสัมพันธ์ระหว่างส่วนประกอบของตัวควบคุมใน 1 ชุดได้ ซึ่งจะได้ Block Diagram เป็นไปดังที่แสดงไว้ในรูป



รูปที่ 3.3 แสดง Block Diagram ของส่วนควบคุมและประมวลผลของเครื่องควบคุมสัญญาณไฟจราจร

จากบล็อกไดอะแกรมข้างต้น สามารถนำแต่ละส่วนของบล็อกมาออกแบบเป็นวงจร และนำมาทำการเชื่อมต่อกันทั้งหมด ได้รูปวงจรรวมเป็นดังต่อไปนี้

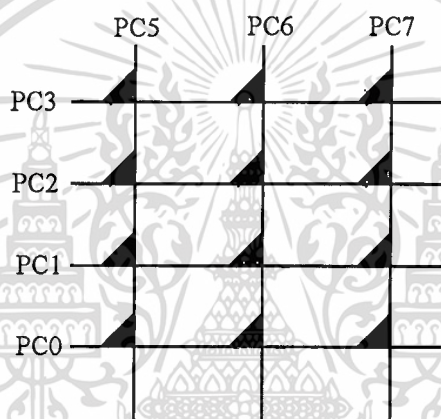


Title		TRAFFIC LIGHT CONTROLLER	
Drawn	Checked	Number	Revision
DATE	DATE	A3	3
FILE	PROJECT	NAME	PROJECT

รูปที่ 3.4 แสดงวงจรของเครื่องควบคุมสัญญาณไฟจราจร

3.3 การออกแบบในส่วนคีย์รับข้อมูล

สำหรับในส่วนของคีย์รับข้อมูลนั้น เราจะรับสัญญาณข้อมูลโดยผ่านทางพอร์ทของ 8255 ซึ่งในการทำงานของส่วนคีย์รับข้อมูลนั้น เมื่อทำการกดคีย์แล้วคีย์จะส่งสัญญาณอินเทอร์รัปต์เข้าสู่ส่วนควบคุมเพื่อบอกให้ตัวควบคุมทราบว่ามีการป้อนข้อมูลเข้ามาแล้ว ทางส่วนควบคุมก็จะหยุดการทำงานในส่วนที่ทำอยู่มาตรวจสอบคีย์รับข้อมูลซึ่งต่ออยู่กับ 8255 ว่าทางผู้ควบคุมนั้นกดคีย์อะไร แล้วก็จะไปประมวลผลทำงานตามโปรแกรมของคีย์ที่ผู้ควบคุมทำการกด สำหรับพอร์ทของ 8255 ที่ใช้ต่อเข้ากับอุปกรณ์คีย์รับข้อมูลนั้นจะต่อเข้ากับพอร์ท C โดยจะมีการกำหนดรูปแบบของพอร์ท C โดยให้ 4 บิตบน (PC5-PC7) เป็นเอาต์พุตและให้ 4 บิตล่าง (PC0-PC3) เป็นอินพุต



รูปที่ 3.5 แสดงรูปแบบการต่อพอร์ทเข้ากับสวิตช์ของคีย์รับข้อมูล

สำหรับขั้นตอนในการทำงานของคีย์รับข้อมูลนั้น จะมีลักษณะการทำงานโดย บิตที่ 4-7 ของพอร์ท C ของ 8255 จะให้ค่าสัญญาณ 0 ออกมาก่อน เมื่อสวิตช์ถูกกด ก็จะมีสัญญาณอินเทอร์รัปต์เข้าสู่หน่วยควบคุมและหน่วยควบคุมก็จะส่งให้บิตที่ 0-3 ของพอร์ท C ของ 8255 อ่านค่าสัญญาณลอจิกเข้ามา และทำการตรวจสอบดูว่าที่บิตไหนมีค่าลอจิกเป็น 0 เมื่อตรวจพบว่าที่บิตไหนมีค่าเป็น 0 แล้วก็จะสามารถทราบได้ว่าสวิตช์ในแถวไหนถูกกด ขั้นตอนต่อจากนั้น จะให้บิตที่ 5 ให้ค่า 0 ออกมาและให้บิตที่ 6 และ 7 ให้ค่า 1 ออกมา แล้วให้บิตในแถวของสวิตช์ที่พบแล้วทำการอ่านค่าเข้ามาตรวจสอบ ถ้าตรวจสอบแล้วค่าที่ได้เป็น 0 ก็แสดงว่าสวิตช์ในคอลัมน์แรกถูกกด ซึ่งก็จะนำไปประมวลผลว่าสวิตช์ในแถวและคอลัมน์ที่ตรวจพบนั้น คือสวิตช์ใด แต่ถ้าตรวจสอบแล้วค่าที่ได้เป็น 1 ก็จะต้องทำการตรวจสอบต่อไป โดยการให้บิตที่ 6 ให้ค่า 0 ออกมาและให้บิตที่ 5 และ 7 ให้ค่า 1 ออกมา แล้วทำการอ่านค่าโดยแถวที่รู้ค่าแล้ว ตรวจสอบดูว่าค่าที่ได้เป็น 0 หรือ 1 ถ้าเป็น 0 ก็แสดงว่าสวิตช์ในคอลัมน์ของบิต 6 ถูกกด และถ้าเป็น 1 ก็จะต้องทำการเช็คค่าในคอลัมน์ต่อไป

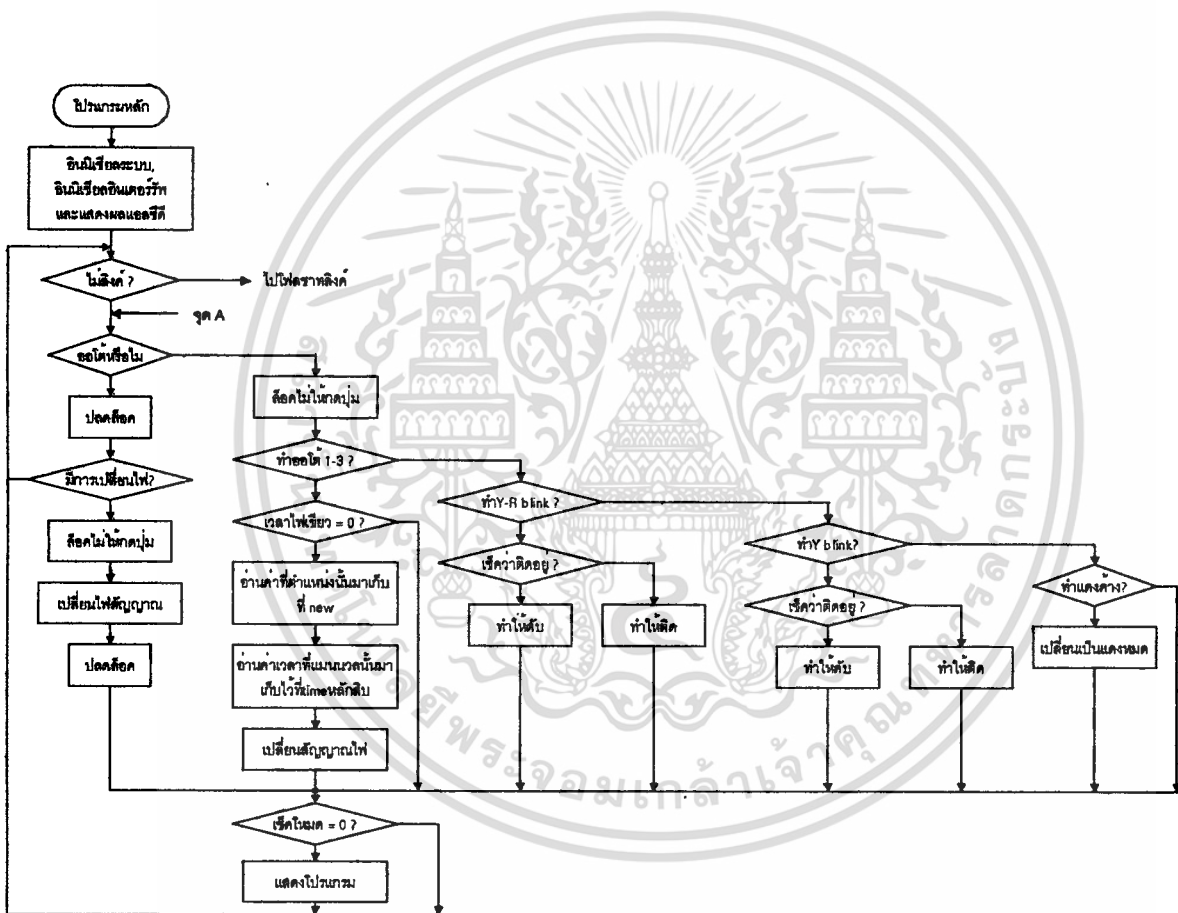
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การออกแบบโปรแกรมที่ใช้ควบคุมอุปกรณ์ควบคุมสัญญาณไฟจราจร

สำหรับในส่วนของโปรแกรมที่ใช้ควบคุมอุปกรณ์ควบคุมสัญญาณไฟจราจรนั้น สามารถแบ่งออกได้เป็นหลายส่วนด้วยกัน ซึ่งขั้นตอนของการออกแบบโปรแกรมนี้ เริ่มต้นจากการเขียนแผนภาพแสดงลำดับการทำงานของโปรแกรม (Flow Chart) แล้วค่อยนำแผนภาพนั้นมาเขียนเป็นโปรแกรมต่อไป

สำหรับแผนภาพแสดงลำดับการทำงานในส่วนของโปรแกรมหลัก (Main Program) นั้น มีดังต่อไปนี้

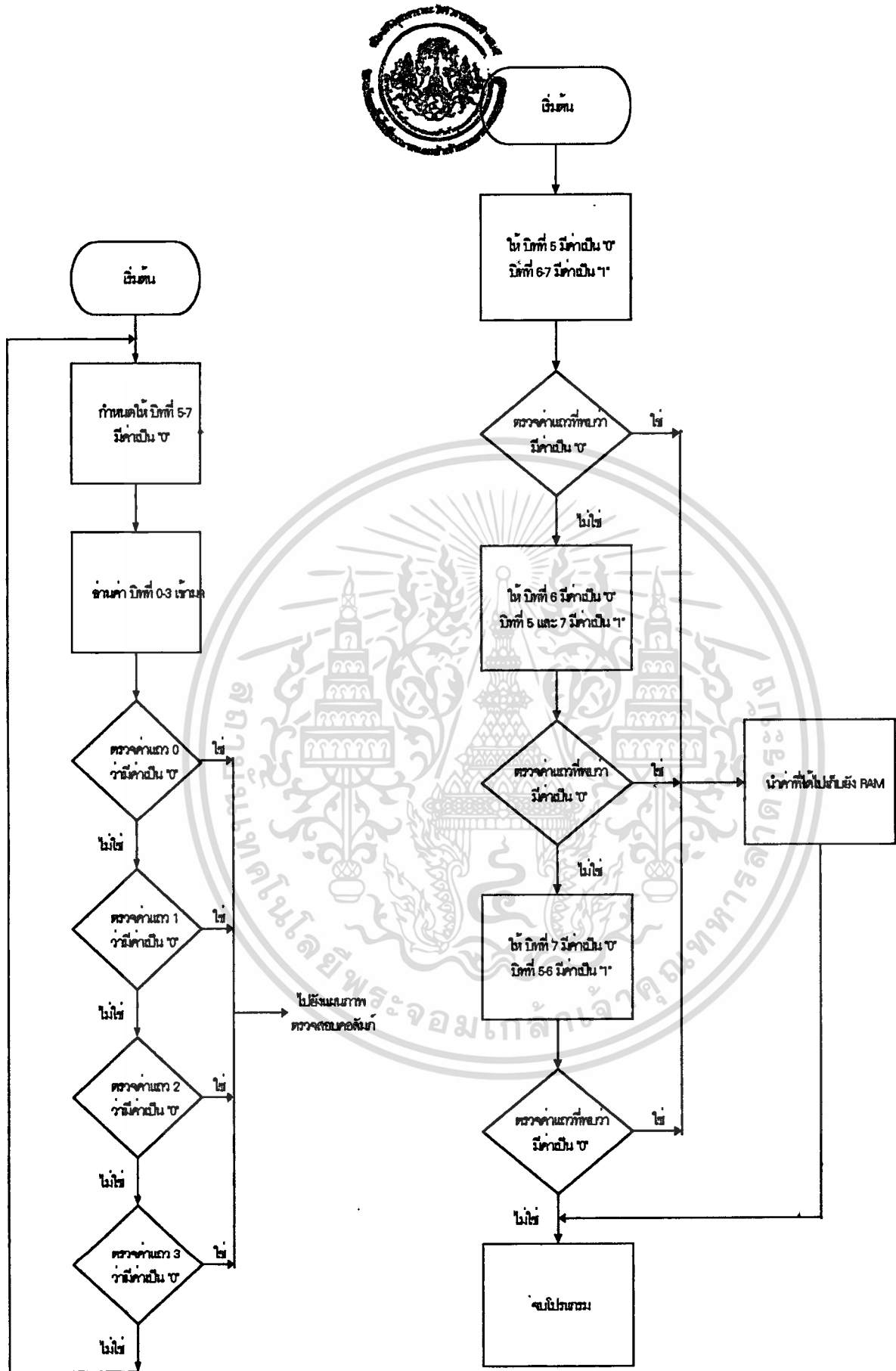
รูปที่ 3.6



รูปที่ 3.6 แสดงแผนภาพลำดับการทำงานของโปรแกรมหลักของเครื่องควบคุมสัญญาณไฟจราจร

สำหรับในส่วนของโปรแกรมที่ทำหน้าที่ตรวจสอบคีย์นั้น จะทำงานตามลำดับขั้นตอนตามแผนภาพต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 แสดงแผนภาพลำดับการทำงานของโปรแกรมตรวจสอบคีย์

เอกสารนี้เป็นเอกสารลับ... ไม่ว่าการมีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสาร... 038153

บทที่ 4

การทดลองและผลการทดลอง

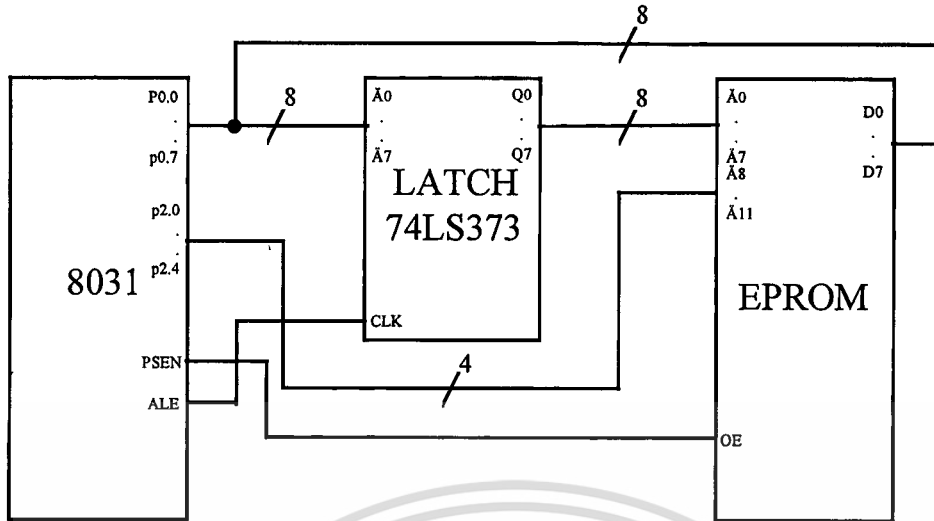
สำหรับในอุปกรณ์เครื่องควบคุมสัญญาณไฟจราจรนี้นั้น ได้แบ่งการทดลองออกเป็น ส่วนต่าง ๆ ที่ละส่วน โดยในการทดลองทีละส่วนนี้มีวัตถุประสงค์เพื่อที่จะได้ง่ายในการตรวจสอบและสังเกตผลต่าง ๆ ว่า ในการทำงานของส่วนต่าง ๆ จะสามารถทำงานได้ตามที่เราต้องการ โดยการทดลองทั้งหมดที่ได้ทำการทดลองในชุดควบคุมสัญญาณไฟจราจรนั้น แบ่งเป็นหัวข้อต่าง ๆ ได้ดังนี้ คือ

1. ทำการทดสอบการทำงานของส่วนควบคุมและประมวลผล
2. ทำการทดสอบการทำงานของพอร์ทของ 8255 กับ ชุดจำลองแสดงสัญญาณไฟจราจร
3. ทำการทดสอบการทำงานของอุปกรณ์แสดงผล LCD
4. ทำการทดสอบการทำงานของอุปกรณ์หน่วยความจำประเภท RAM
5. ทำการทดสอบการทำงานของอุปกรณ์ให้สัญญาณนาฬิกา
6. ทำการทดสอบการทำงานของอุปกรณ์ชุดคีร์รับข้อมูล
7. ทำการทดสอบการทำงานของโปรแกรมควบคุมการทำงานของไฟจราจร
8. ทำการทดสอบการทำงานเชื่อมต่อสัญญาณระหว่างอุปกรณ์ควบคุม 2 ชุด
9. ทำการทดสอบการทำงานของชุดควบคุมไฟจราจรที่สามารถเชื่อมต่อสัญญาณระหว่างชุดควบคุมได้

4.1 การทดลองทำการทดสอบการควบคุมการทำงานของส่วนประมวลผล

ในการทดลองนี้ มีวัตถุประสงค์ เพื่อทดสอบการทำงานของส่วนประมวลผลให้สามารถควบคุมการทำงานเพื่อให้ได้ผลตามโปรแกรมที่สั่งงานเข้าไปได้อย่างถูกต้อง เพื่อที่จะได้แน่ใจว่าส่วนประมวลผลนี้จะสามารถนำไปใช้ควบคุมไฟจราจรได้ตามที่ต้องการโดยไม่ผิดพลาด โดยส่วนของวงจรที่จะใช้ในการทดลองในส่วนนี้ จะใช้งานเฉพาะส่วนต่าง ๆ ดังต่อไปนี้ ได้แก่ ส่วนของ 8031, ส่วนของ IC 74LS373, โปรแกรมทดสอบ ซึ่งจะส่งผ่านเข้าไปเพื่อให้ 8031 ทำงานตามต้องการทางอินพุตอิมูเลเตอร์ ซึ่งจะแสดงส่วนของวงจรที่ใช้ในการทดลองตามบล็อกในภาพต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 แสดงบล็อกของส่วนที่ใช้ในการทดลองส่วนควบคุมและประมวลผล

วิธีการทดลอง สามารถทำได้โดยการใช้โปรแกรมทดสอบเพื่อสั่งให้ 8031 ทำงาน โดยในโปรแกรมจะสั่งงานให้ตัว 8031 ส่งข้อมูลออกทางพอร์ต 1 ของ 8031 โดยจะส่งข้อมูลออกเป็น " 0 " และ " 1 " สลับกันไป

สำหรับโปรแกรมที่ใช้ในการทดลองนี้ มีดังต่อไปนี้

```

MAIN:   MOV    A, #00H
        MOV    P1, A
        LCALL DELAY
        MOV    A, #0FFH
        MOV    P1, A
        LCALL DELAY
        JMP    MAIN

DELAY:  MOV    R1, #0FFH
D1:     MOV    R2, #0FFH
D2:     DJNZ   R2, D2
        DJNZ   R1, D1

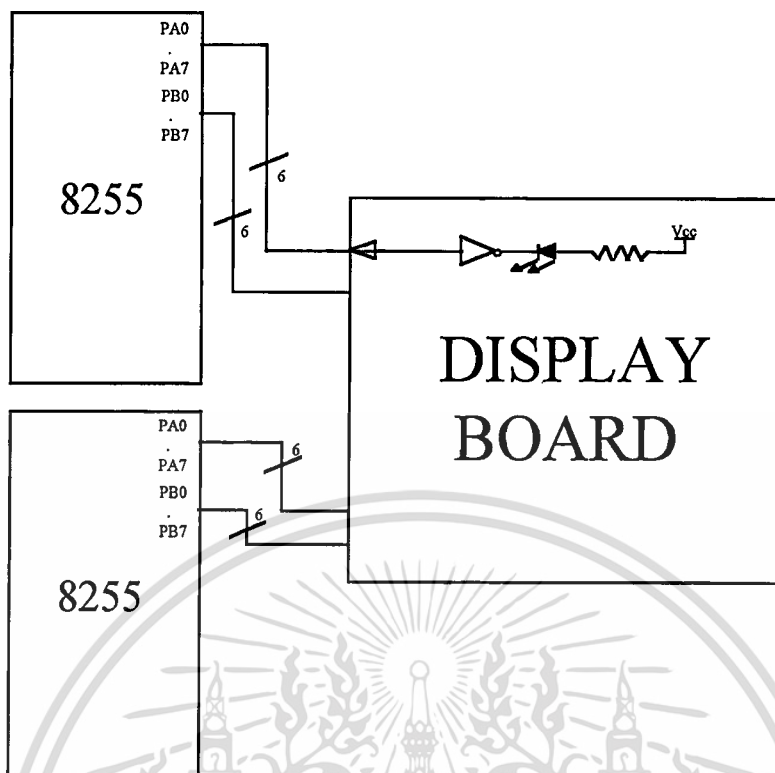
        RET

```

ผลการทดลอง

หลังจากที่ได้ทดลองตามขั้นตอนข้างต้นแล้ว ทำการวัดผลการทดลองโดยใช้ออสซิลโลสโคปวัดค่าลอจิกที่ขาของ 8031 ตำแหน่งพอร์ต 1 แล้ว ปรากฏว่าข้อมูลที่ส่งออกมาทางพอร์ต 1 แสดงค่าเป็น " 0 " และ " 1 " สลับกัน ซึ่งแสดงว่า 8031 สามารถทำงานได้ตามวัตถุประสงค์ของการทดลอง จึงทำการทดลองในหัวข้อต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 แสดงการเชื่อมต่อระหว่าง 8255 กับอุปกรณ์แสดงผลสัญญาณไฟจากรจ้ง

สำหรับการทดลองในหัวข้อนี้ เราจะทำการทดลองออกเป็น 2 ส่วนคือ ทดสอบการทำงานของพอร์ท 8255 กับชุดแสดงผลสัญญาณไฟจากรจ้ง โดยใช้โปรแกรมซึ่งจะให้อุปกรณ์แสดงผลสัญญาณไฟจรร้ง ทำการแสดงผลสัญญาณไฟในแต่ละด้านติดและดับสลับกันไป เพื่อที่จะทดสอบว่าอุปกรณ์ที่ใช้ทดลองนี้สามารถใช้งานได้ครบถ้วน เพื่อที่จะนำไปใช้ในการทดสอบโปรแกรมจรร้งรูปแบบไฟจรร้งใน ส่วนต่อไป โดยจะมีโปรแกรมที่ใช้ทดลองในส่วนแรกดังต่อไปนี้

```

MAIN:  MOV     A, #89H
        MOV     DPTR, #0F803H
        MOVX   @DPTR, A           ; INITIAL 8255
        MOV     DPTR, #0F903H
        MOVX   @DPTR, A           ; INITIAL 8255

```

```

YEL:   MOV     A, #00H
        MOV     DPTR, #0F800H
        MOVX   @DPTR, A
        MOV     DPTR, #0F900H
        MOVX   @DPTR, A
        MOV     A, #0FFH
        MOV     DPTR, #0F801H
        MOVX   @DPTR, A
        MOV     DPTR, #0F901H
        MOVX   @DPTR, A
        LCALL  DELAY
        MOV     A, #0FFH
        MOV     DPTR, #0F800H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องยกย่องถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOVX    @DPTR, A
MOV     DPTR, #0F900H
MOVX    @DPTR, A
MOV     A, #00H
MOV     DPTR, #0F801H
MOVX    @DPTR, A
MOV     DPTR, #0F901H
MOVX    @DPTR, A
LCALL   DELAY
JMP     YEL

DELAY:   MOV     R0, #0FFH
DELAY2:  MOV     R1, #0FFH
DELAY1:  DJNZ   R1, DELAY1
         DJNZ   R0, DELAY2
         RET

END

```

ผลการทดลอง

จากการทดลองโดยใช้โปรแกรมข้างบน ได้ผลการทดลองโดยชุดแสดงสัญญาณไฟจราจรจำลอง นั้นได้ให้สัญญาณไฟแต่ละด้านติดและดับสลับกันไปเรื่อย ๆ ตามที่ต้องการ ซึ่งแสดงว่าตัว 8255 สามารถแสดงผลโดยผ่านชุดแสดงสัญญาณไฟได้ครบถ้วนตามต้องการ ซึ่งจะนำไปทดลองในส่วนต่อไป

สำหรับโปรแกรมที่จะให้ทดลองในส่วนต่อไปนี้นั้น จะเป็นการทดลองการทำงานของไฟจราจรโดยใช้โปรแกรมที่เขียนขึ้น ให้เริ่มต้นโปรแกรมด้วยให้ไฟเหลืองทุกดวงกะพริบ จากนั้นให้ไฟแดงทุกดวงติด แล้วจึงให้ไฟเขียวตรงและไฟเขียวเลี้ยวขวาติดขึ้นที่แยกที่ 1 สักครู่จึงให้ไฟเหลืองติดและให้ไฟแดงติด โดยจะให้ไฟเขียว, เหลืองและแดง ติดเรียงไปที่ละแยกจนครบ แล้วจึงมาทำที่แยกที่ 1 ต่อ ทำดังนี้ไปเรื่อย ๆ

สำหรับโปรแกรมที่ใช้ในการทดลองในส่วนนี้นั้น มีดังต่อไปนี้

```

P1A     EQU     06F00H
P1B     EQU     06F01H
P1C     EQU     06F02H
P1P     EQU     06F03H
P2A     EQU     08F00H
P2B     EQU     08F01H
P2C     EQU     08F02H
P2P     EQU     08F03H

LCALL   DELAY

MOV     A, #89H                                ; INITIAL 8255
MOV     DPTR, #P1P
MOVX    @DPTR, A
MOV     DPTR, #P2P
MOVX    @DPTR, A                                ; INITIAL 8255

MOV     R4, #05H

MAIN:   LCALL   OPEN                            ; YELLOW BLINK
        LCALL   DELAY
        LCALL   CLOSE

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ในการใช้เพื่อการศึกษาเท่านั้น ไม่สามารถเผยแพร่หรือใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังให้คำปรึกษาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                LCALL    DELAY
                DJNZ     R4,MAIN
                LCALL    RED
VIC:           LCALL    TRAFF
                JMP      VIC

OPEN:          MOV      A,#12H
                LCALL    AOR
                RET

AOR:           MOV      DPTR,#P1A
                MOVX     @DPTR,A
                MOV      DPTR,#P1B
                MOVX     @DPTR,A
                MOV      DPTR,#P2A
                MOVX     @DPTR,A
                MOV      DPTR,#P2B
                MOVX     @DPTR,A
                RET

CLOSE:         MOV      A,#00H
                LCALL    AOR
                RET

SECTION RED:    MOV      A,#024H                ;RED          EVERY
                LCALL    AOR
                LCALL    DELAY
                LCALL    DELAY
                LCALL    DELAY
                RET

TRAFF:         MOV      A,#09H                ;GREEN 1ST SECTION
                MOV      DPTR,#P1A
                MOVX     @DPTR,A
                LCALL    DELAYG
                MOV      A,#012H            ;YELLOW 1ST SECTION
                MOV      DPTR,#P1A
                MOVX     @DPTR,A
                LCALL    DELAY
                MOV      A,#024H            ;RED 1ST SECTION
                MOV      DPTR,#P1A
                MOVX     @DPTR,A
                LCALL    DELAY
                MOV      A,#09H                ;GREEN 2ND SECTION
                MOV      DPTR,#P1B
                MOVX     @DPTR,A
                LCALL    DELAYG
                MOV      A,#012H            ;YELLOW 2ND SECTION
                MOV      DPTR,#P1B
                MOVX     @DPTR,A
                LCALL    DELAY
                MOV      A,#024H            ;RED 2ND SECTION
                MOV      DPTR,#P1B
                MOVX     @DPTR,A
                LCALL    DELAY
                MOV      A,#09H                ;GREEN 3RD SECTION
                MOV      DPTR,#P2A
                MOVX     @DPTR,A

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับภาควิชาการศึกษานานาชาติ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่ควรเผยแพร่หรือทำซ้ำโดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกมิให้ดัดแปลงหรือทำซ้ำ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        LCALL    DELAYG
        MOV      A,#012H                ;YELLOW 3RD
        SECTION
        MOV      DPTR,#P2A
        MOVX     @DPTR,A
        LCALL    DELAY
        MOV      A,#024H                ;RED 3RD SECTION
        MOV      DPTR,#P2A
        MOVX     @DPTR,A
        LCALL    DELAY
        MOV      A,#09H                 ;GREEN 4TH SECTION
        MOV      DPTR,#P2B
        MOVX     @DPTR,A
        LCALL    DELAYG
        MOV      A,#012H                ;YELLOW 4TH
        SECTION
        MOV      DPTR,#P2B
        MOVX     @DPTR,A
        LCALL    DELAY
        MOV      A,#024H                ;RED 4TH SECTION
        MOV      DPTR,#P2B
        MOVX     @DPTR,A
        LCALL    DELAY
        RET
DELAYG:  MOV      R3,#0AH
G1:      LCALL    DELAY
        DJNZ     R3,G1
        RET
DELAY:   MOV      R1,#0FFH
D1:      MOV      R2,#0FFH
DELAY1:  DJNZ     R2,DELAY1
        DJNZ     R1,D1
        RET
        END

```

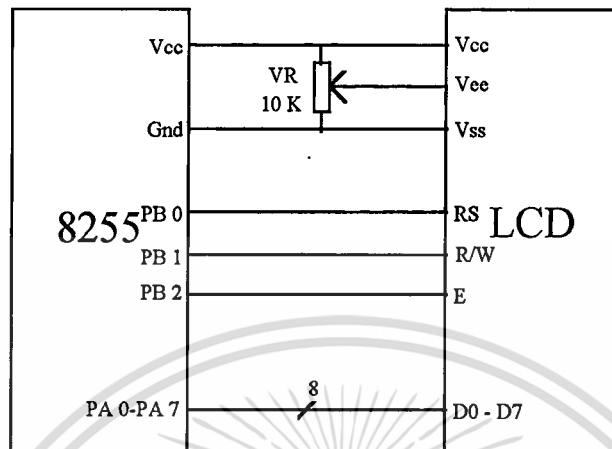
ผลการทดลอง

จากการทดลองในส่วนที่ใช้โปรแกรมข้างต้น ผลที่ได้คืออุปกรณ์แสดงผลสัญญาณไฟจราจรจำลองจะแสดงสัญญาณไฟตามที่โปรแกรมกำหนดไว้ ซึ่งจากสัญญาณไฟที่ได้จากโปรแกรมนี้อาจจะเป็นลักษณะจำลองของสัญญาณไฟจราจรจริง ๆ ซึ่งจะเป็นแนวทางต่อไป ในการเขียนโปรแกรมควบคุมสัญญาณไฟจราจรต่อไป

4.3 การทดลองการทดสอบการทำงานของอุปกรณ์แสดงผล LCD

สำหรับการทดลองนี้ มีวัตถุประสงค์เพื่อทดสอบและควบคุมการทำงานของอุปกรณ์แสดงผล LCD เพื่อให้สามารถแสดงผลตามที่ต้องการ โดยส่วนของวงจรที่ใช้ในการทดลองนั้น จะประกอบด้วยส่วนของอุปกรณ์ต่าง ๆ ได้แก่ 8031, 8255, อุปกรณ์แสดงผล LCD, ส่วนของ IC 74LS373, ส่วนของ IC 74LS138 และส่วนของโปรแกรมที่ใช้ในการทดลองใช้งานเพื่อการศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับในส่วนของวงจรที่ใช้ในการทดลองนี้นั้น จะใช้วงจรเหมือนในการทดลองที่แล้ว แต่จะเพิ่มเติมในส่วนของการเชื่อมต่อระหว่างตัว 8255 กับอุปกรณ์แสดงผล LCD โดยจะมีการเชื่อมต่ออุปกรณ์ดังรูป



รูปที่ 4.4 แสดงการเชื่อมต่ออุปกรณ์แสดงผล LCD เข้ากับพอร์ทของ 8255

สำหรับโปรแกรมที่ใช้ทดสอบการทำงานของอุปกรณ์แสดงผล LCD นั้น จะให้อุปกรณ์แสดงผล LCD ทำการแสดงผลเป็นข้อความต่อไปนี้

```

\*****\
\ THIS PROGRAM \
\ IS LCD TESTING \
\*****\
\ TRAFFIC \
\ LIGHT \
\ CONTROLLER \
\ \
\ BY \
\ VATARAPONG \
\ VORAWAN \
\ WORAWUT \

```

โดยโปรแกรมที่ใช้ในการทดสอบการทำงานของอุปกรณ์แสดงผล LCD นี้ มีดังต่อไปนี้

```

PORTC EQU 0F802H ;LCD DATA (PA0-
PA7)
PORTB EQU 0F801H ;LCD CONTROL
(PB0-PB2)
PORTP EQU 0F803H ;8255 CONTROL
PORT
MAIN: MOV DPTR,#PORTP ;SET 8255 CONTROL
PORT MOV A,#80H
MOVX @DPTR,A
MOV R2,#4 และต้องอ้างอิงถึงเจ้าา;FIRST DELAY
การนำไปใช้

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่โดยไม่ได้รับอนุญาต

```

        LCALL  DELAY
        MOV    A, #38H                ;FUNCTION SET
        LCALL  LCDWI
        MOV    A, #10H                ;CURSOR/DISPLAY
        LCALL  LCDWI
        MOV    A, #0EH                ;DISPLAY CONTROL
        LCALL  LCDWI
        MOV    A, #06H                ;MODE SET
        LCALL  LCDWI
        MOV    A, #01H                ;CLEAR
        LCALL  LCDWI

MAIN2:   MOV    DPTR, #MAINT1
        LCALL  LCDLD
        MOV    R2, #20H
        LCALL  DELAY
        LCALL  DELAY1
        MOV    DPTR, #MAINT2
        LCALL  LCDLD
        MOV    R2, #20H
        LCALL  DELAY
        LCALL  DELAY1
        MOV    DPTR, #MAINT3
        LCALL  LCDLD
        MOV    R2, #20H
        LCALL  DELAY
        LCALL  DELAY1
        SJMP   MAIN2

MAINT1:  DB    "*****"
        DB    "  THIS PROGRAM  "
        DB    " IS LCD TESTING  "
        DB    "*****"

MAINT2:  DB    "  TRAFFIC  "
        DB    "    LIGHT    "
        DB    " CONTROLLER  "
        DB    "              "

MAINT3:  DB    "    BY    "
        DB    " VATARAPONG  "
        DB    "  VORAWAN  "
        DB    "  WORAWUT   "

DELAY:   MOV    R3, #50
DELAY1:  MOV    R4, #0
        DJNZ   R4, $
        DJNZ   R3, DELAY1
        DJNZ   R2, DELAY
        RET

LCDLD:   MOV    A, #80H                ;SET ADDRESS LINE
1
        LCALL  LCDLDS
        MOV    A, #0C0H                ;SET ADDRESS LINE
2
        LCALL  LCDLDS
        MOV    A, #90H                ;SET ADDRESS LINE
3
        LCALL  LCDLDS

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่หรือใช้เพื่อการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามนำเอกสารนี้ไปเผยแพร่และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4

```

MOV      A, #0D0H                ;SET ADDRESS LINE

LCALL   LCDLDS
RET

LCDLDS:  PUSH   DPH                ;LOAD SUB.
        PUSH   DPL
        LCALL  LCDWI                ;WRITE ADDRESS
        POP    DPL
        POP    DPH
        MOV    R2, #16              ;16 CHAR.
LCDLDS1: CLR    A
        MOVC   A, @A+DPTR          ;MOVC FOR CODE/
        PUSH   DPH                ;MOVX FOR DATA
        PUSH   DPL
        LCALL  LCDWD                ;WRITE DATA
        POP    DPL
        POP    DPH
        INC    DPTR
        DJNZ   R2, LCDLDS1
        RET

LCDWI:   MOV    DPTR, #PORTC        ;DATA TO PORTC
        MOVX   @DPTR, A
        MOV    DPTR, #PORTB        ;PORTB READ
MODIFY WRITE
        MOVX   A, @DPTR
        CLR    ACC.0                ;RS=0
        CLR    ACC.1                ;R/W=0
        CLR    ACC.2                ;ENABLE=0
        MOVX   @DPTR, A
        SETB   ACC.2                ;ENABLE=1
        MOVX   @DPTR, A
        CLR    ACC.2                ;ENABLE=0
        MOVX   @DPTR, A
        MOV    A, #0                ;DELAY
LCDWI1:  DEC    A
        JNZ   LCDWI1
        RET

LCDWD:   MOV    DPTR, #PORTC        ;DATA TO PORTA
        MOVX   @DPTR, A
        MOV    DPTR, #PORTB        ;PORTB READ
MODIFY WRITE
        MOVX   A, @DPTR
        SETB   ACC.0                ;RS=1
        CLR    ACC.1                ;R/W=0
        CLR    ACC.2                ;ENABLE=0
        MOVX   @DPTR, A
        SETB   ACC.2                ;ENABLE=1
        MOVX   @DPTR, A
        CLR    ACC.2                ;ENABLE=0
        MOVX   @DPTR, A
        MOV    A, #0                ;DELAY
LCDWD1:  DEC    A
        JNZ   LCDWD1
        RET

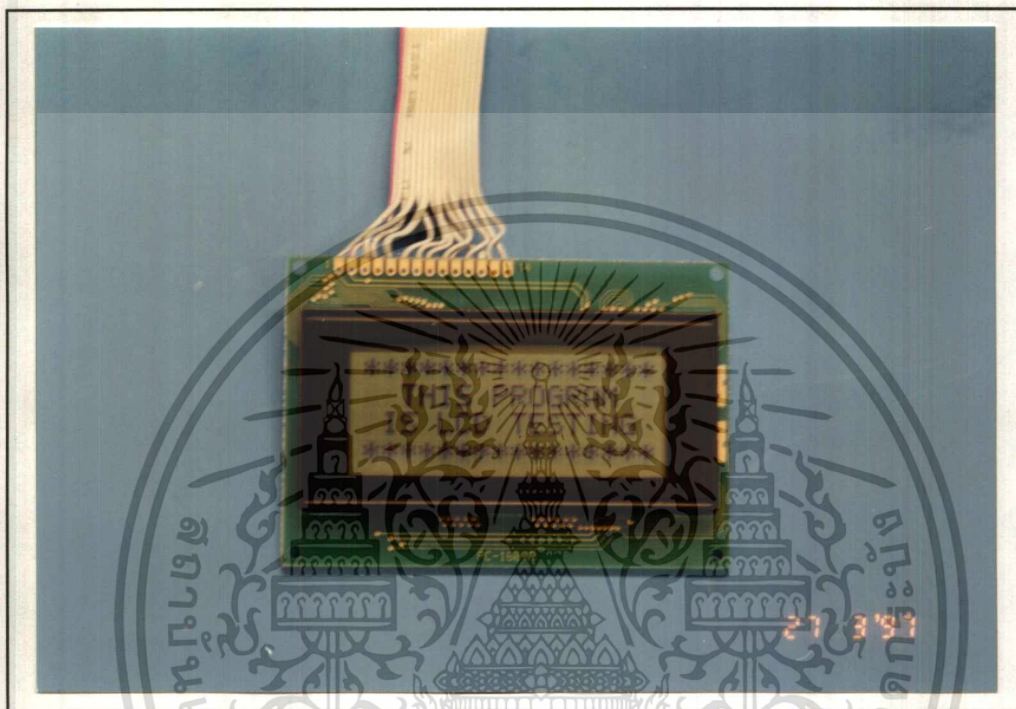
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานในห้องเรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

END

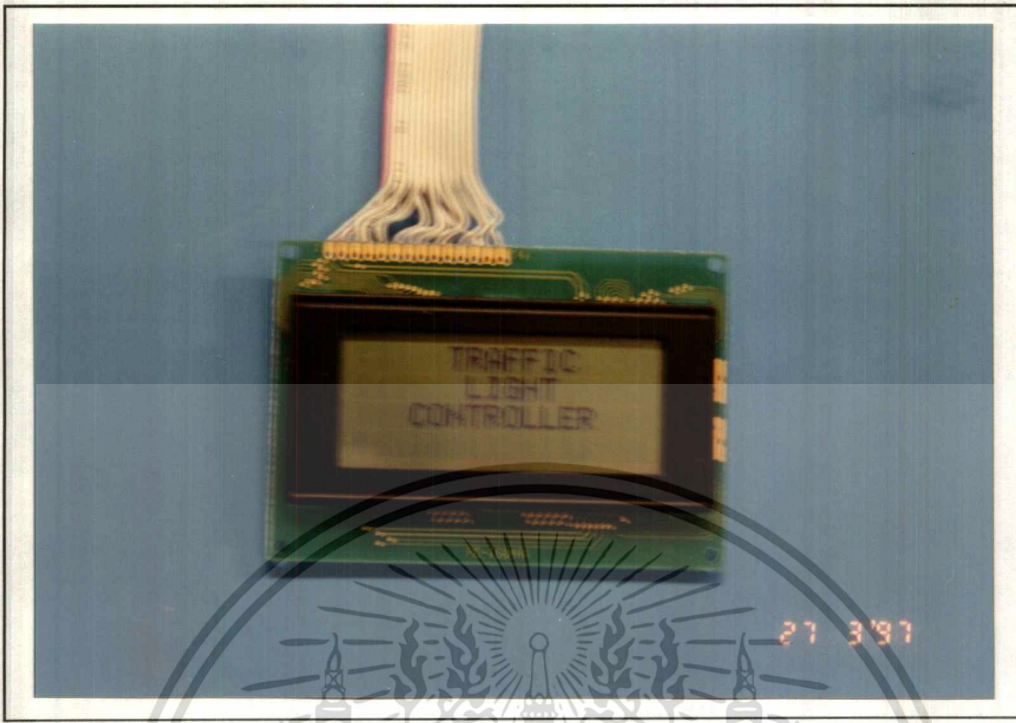
ผลการทดลอง

สำหรับผลที่ได้หลังจากทำการทดลองคืออุปกรณ์แสดงผล LCD แสดงข้อความออกมาตามที่เราต้องการเป็นไปดังรูป

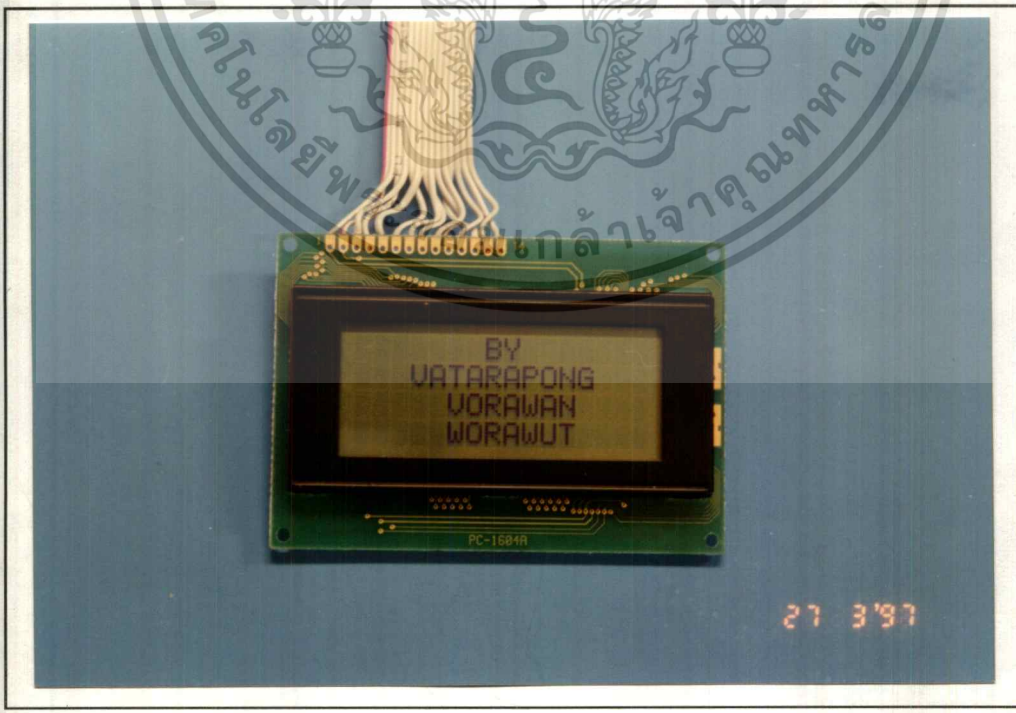


รูปที่ 4.5 แสดงภาพผลการทดลองอุปกรณ์แสดงผล LCD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 แสดงภาพผลการทดลองอุปกรณ์แสดงผล LCD



รูปที่ 4.7 แสดงภาพผลการทดลองอุปกรณ์แสดงผล LCD หน้าไปใช้ประโยชน์ด้านการค้า

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้ ห้ามมิให้ทำซ้ำโดยไม่ขออนุญาต หากฝ่าฝืนจะดำเนินการตามกฎหมาย
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การทดลองการทำงานของอุปกรณ์หน่วยความจำประเภท RAM

สำหรับในส่วนของการทดลองนี้ มีวัตถุประสงค์ในการทดลองเพื่อทดสอบการทำงานของหน่วยความจำประเภท RAM ว่าสามารถที่จะใช้งานได้หรือไม่ โดยในการทดลองนี้ จะใช้ส่วนประกอบต่าง ๆ ในวงจร ได้แก่ ส่วนของ 8031, ส่วนของตัว 8255, ส่วนของอุปกรณ์แสดงผลจำลองสัญญาณไฟจราจร, ส่วนของ IC 74LS138, ส่วนของ IC 74LS373, ส่วนของอุปกรณ์แสดงผล LCD, RAM 6116 และส่วนของโปรแกรมทดสอบ

สำหรับในส่วนของโปรแกรมที่ใช้ในการทดสอบนี้ จะมีการกำหนดค่าเอาไว้ในโปรแกรม แล้วให้โปรแกรมทำการเก็บค่าที่กำหนดเอาไว้นี้ เข้าสู่ RAM หลังจากนั้น ก็ให้ 8031 ทำการอ่านค่าใน RAM ตามตำแหน่งที่อยู่ที่เราเก็บค่าเอาไว้ แล้วให้แสดงผลออกทางอุปกรณ์แสดงผล LCD

โปรแกรมที่ใช้ในการทดลองนี้ มีดังต่อไปนี้

```

PORTC EQU 04F02H ;LCD DATA (PA0-PA7)
PORTB EQU 04F01H ;LCD CONTROL (PB0-
PB2)
PORTP EQU 04F03H ;8255 CONTROL PORT

LCALL DELAY

MAIN: MOV DPTR,#PORTP ;SET 8255 CONTROL
PORT
MOV A,#80H
MOVX @DPTR,A
MOV R2,#4 ;FIRST DELAY
LCALL DELAY
MOV A,#01H ;CLEAR
LCALL LCDWI
MOV A,#38H ;FUNCTION SET
LCALL LCDWI
MOV A,#00010000B ;DISPLAY SHIFT
LCALL LCDWI
MOV A,#00001100B ;DISPLAY CONTROL
LCALL LCDWI
MOV A,#00000100B ;MODE SET
LCALL LCDWI

RAM: MOV A,#8EH
MOV R5,#30H
MOV R6,#0AH
MOV DPTR,#02F10H

LOOP: MOV A,R5
MOVX @DPTR,A
CLR A
MOVX A,@DPTR
PUSH DPH
PUSH DPL
LCALL LCDWD
POP DPL
POP DPH
LCALL D1
INC R5

```

```

        INC     DPL
        DJNZ   R6, LOOP
        JMP    RAM

D1:     MOV     R1, #0A0H
D2:     MOV     R2, #0FFH
        DJNZ   R2, $
        DJNZ   R1, D2
        RET

DELAY:  MOV     R3, #50
DELAY1: MOV     R4, #0
        DJNZ   R4, $
        DJNZ   R3, DELAY1
        DJNZ   R2, DELAY
        RET

LCDWI:  MOV     DPTR, #PORTC           ;DATA TO PORTC
        MOVX   @DPTR, A
        MOV    DPTR, #PORTB         ;PORTB READ MODIFY
        WRITE
        CLR    ACC.0                 ;RS=0
        CLR    ACC.1                 ;R/W=0
        CLR    ACC.2                 ;ENABLE=0
        MOVX   @DPTR, A
        SETB   ACC.2                 ;ENABLE=1
        MOVX   @DPTR, A
        CLR    ACC.2                 ;ENABLE=0
        MOVX   @DPTR, A
        MOV    A, #0                 ;DELAY
LCDWI1: DEC    A
        JNZ    LCDWI1
        RET

LCDWD:  MOV     DPTR, #PORTC           ;DATA TO PORTA
        MOVX   @DPTR, A
        MOV    DPTR, #PORTB         ;PORTB READ MODIFY
        WRITE
        SETB   ACC.0                 ;RS=1
        CLR    ACC.1                 ;R/W=0
        CLR    ACC.2                 ;ENABLE=0
        MOVX   @DPTR, A
        SETB   ACC.2                 ;ENABLE=1
        MOVX   @DPTR, A
        CLR    ACC.2                 ;ENABLE=0
        MOVX   @DPTR, A
        MOV    A, #0                 ;DELAY
LCDWD1: DEC    A
        JNZ    LCDWD1
        RET

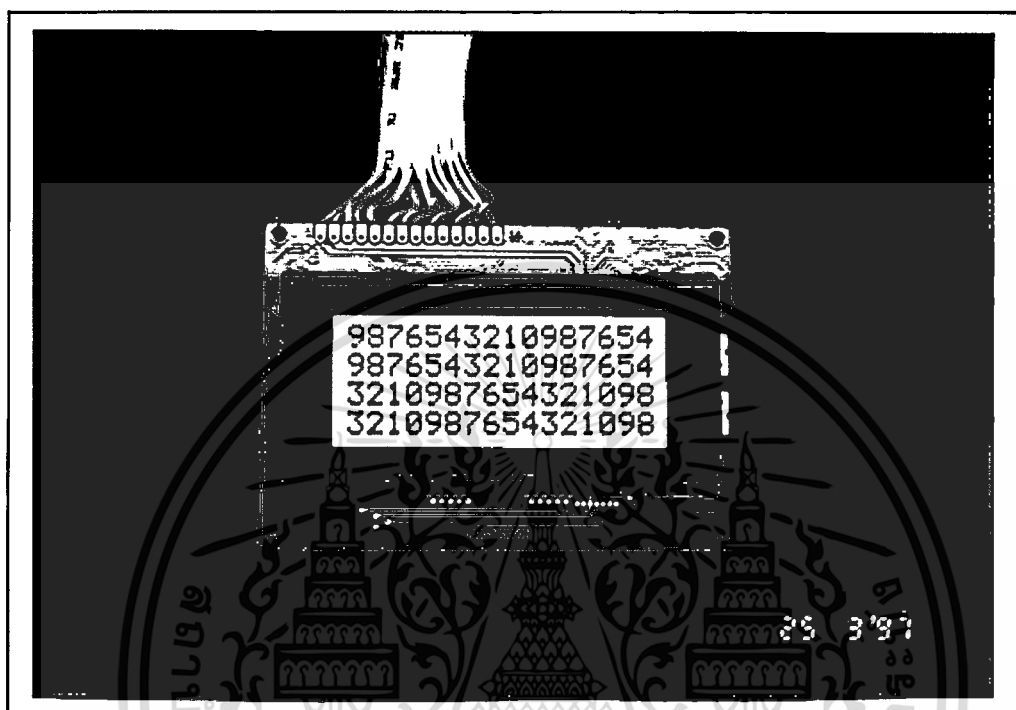
        END

```

ผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อได้ทำการทดลองโดยใช้โปรแกรมข้างต้นนี้แล้ว พบว่าอุปกรณ์แสดงผล LCD ได้แสดงค่าตามค่าที่ตั้งเอาไว้ในโปรแกรม ซึ่งสามารถสรุปได้ว่า RAM สามารถทำงานได้อย่างถูกต้องตามวัตถุประสงค์ของการทดลอง ซึ่งสามารถนำไปใช้งานในส่วนประกอบของอุปกรณ์ควบคุมสัญญาณไฟจราจรได้อย่างถูกต้อง



รูปที่ 4.8 แสดงจอภาพ LCD เมื่อทำการทดลองโปรแกรมทดสอบ RAM

4.5 การทดลองการทำงานของอุปกรณ์ให้สัญญาณนาฬิกา

สำหรับการทดลองในหัวข้อนี้แล้ว มีวัตถุประสงค์เพื่อทดสอบโปรแกรมที่จะใช้งานในการควบคุมอุปกรณ์ให้สัญญาณนาฬิกา และสามารถแสดงผลออกมาเป็นนาฬิกาบนส่วนแสดงผล LCD ได้อย่างถูกต้อง ซึ่งจะนำไปใช้งานเป็นส่วนหนึ่งของโปรแกรมหลักของอุปกรณ์ควบคุมสัญญาณไฟจราจร ซึ่งในส่วนของวงจรที่จะใช้ในการทดลอง จะเพิ่มส่วนที่ใช้ในการทดลองคือ IC เบอร์ MM58167 เข้าไปในส่วนที่ใช้ในการทดลองด้วย

สำหรับโปรแกรมที่ใช้ในการทดลองนี้แล้ว จะให้ตัว 8031 อ่านค่าเวลานาฬิกาจาก IC ที่ให้สัญญาณนาฬิกา แล้วส่งค่าที่อ่านได้เพื่อไปแสดงผลยังส่วนแสดงผลแบบ LCD เพื่อแสดงค่าเวลาออกมา ซึ่งโปรแกรมที่ใช้ในการทดลองนี้ มีดังต่อไปนี้

```

PORTC EQU 04F02H ;LCD DATA (PA0-PA7)
PORTB EQU 04F01H ;LCD CONTROL (PB0-
PB2)
PORTP EQU 04F03H ;8255 CONTROL PORT

```

```

                LCALL  DELAY

MAIN:          MOV     DPTR,#PORTP                ;SET 8255
CONTROL PORT

MOV     A,#80H
MOVX   @DPTR,A
MOV     R2,#4                                ;FIRST DELAY
LCALL  DELAY
MOV     A,#01H                                ;CLEAR
LCALL  LCDWI
MOV     A,#38H                                ;FUNCTION SET
LCALL  LCDWI
MOV     A,#00010000B                          ;DISPLAY SHIFT
LCALL  LCDWI
MOV     A,#00001100B                          ;DISPLAY CONTROL
LCALL  LCDWI
MOV     A,#00000100B                          ;MODE SET
LCALL  LCDWI
MOV     A,#0FFH
MOV     DPTR,#0AF12H                          ;RESET COUNTER
MOVX   @DPTR,A
MOV     DPTR,#0AF13H                          ;RESET RAM
MOVX   @DPTR,A
MOV     DPTR,#0AF15H                          ;START CLOCK
MOVX   @DPTR,A
MOV     A,#22H
MOV     DPTR,#0AF04H                          ;SET HOUR=22
MOVX   @DPTR,A
MOV     DPTR,#0AF03H                          ;SET MINUTE=58
MOV     A,#58H
MOVX   @DPTR,A
MOV     DPTR,#0AF11H
MOV     A,#00000100B                          ;SET INTERRUPT EVERY 1
SEC

MOVX   @DPTR,A
MOV     B,#0
CLK:    MOV     A,#0CCH
        LCALL  LCDWI
        MOV     DPTR,#0AF01H
        MOV     R7,#03H
CLK1:   INC     DPL
        MOVX   A,@DPTR
        MOV     B,A
        ANL    A,#00001111B
        ORL    A,#00110000B
        LCALL  LCDWD
        MOV     A,B
        SWAP  A
        ANL    A,#00000111B
        ORL    A,#00110000B
        LCALL  LCDWD
        MOV     A,#3AH
                LCALL  LCDWD
        MOV     A,P1
        JB     ACC.0,CHANGE
L:      DJNZ   R7,CLK1
        SJMP  CLK

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังได้แจ้งเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CHANGE:      MOV      R3,DPH
             MOV      R4,DPL
             MOV      A,#85H
             LCALL   LCDWI
             MOV      DPTR,#0AF10H
             MOVX    A,@DPTR
             MOV      A,B
             JB       ACC.0,SP0
             MOV      A,#3AH
             LCALL   LCDWD
             JMP      SP1

SP0:         MOV      A,#0A0H
             LCALL   LCDWD

SP1:         MOV      DPL,R4
             MOV      DPH,R3
             INC      B
             JMP      L

D1:          MOV      R1,#050H
D2:          MOV      R2,#0FFH
             DJNZ    R2,$
             DJNZ    R1,D2
             RET

DELAY:       MOV      R3,#50
DELAY1:      MOV      R4,#0
             DJNZ    R4,$
             DJNZ    R3,DELAY1
             DJNZ    R2,DELAY
             RET

LCDWI:       PUSH     DPH
             PUSH     DPL
             MOV      DPTR,#PORTC           ;DATA TO PORTC
             MOVX    @DPTR,A
             MOV      DPTR,#PORTB         ;PORTB READ MODIFY

WRITE

             CLR     ACC.0                 ;RS=0
             CLR     ACC.1                 ;R/W=0
             CLR     ACC.2                 ;ENABLE=0
             MOVX    @DPTR,A
             SETB    ACC.2                 ;ENABLE=1
             MOVX    @DPTR,A
             CLR     ACC.2                 ;ENABLE=0
             MOVX    @DPTR,A
             MOV     A,#0                   ;DELAY

LCDWI1:      DEC     A
             JNZ     LCDWI1
             POP     DPL
             POP     DPH
             RET

LCDWD:       PUSH     DPH
             PUSH     DPL
             MOV      DPTR,#PORTC         ;DATA TO PORTA
             MOVX    @DPTR,A
             MOV      DPTR,#PORTB         ;PORTB READ MODIFY

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาติให้นำไปใช้ประโยชน์ด้วยเอวราคา
 WRITE อนุมัติโดย ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SETB   ACC.0           ;RS=1
CLR    ACC.1           ;R/W=0
CLR    ACC.2           ;ENABLE=0
MOVX   @DPTR,A
SETB   ACC.2           ;ENABLE=1
MOVX   @DPTR,A
CLR    ACC.2           ;ENABLE=0
MOVX   @DPTR,A
MOV    A,#0            ;DELAY
LCDWD1: DEC    A
        JNZ    LCDWD1
        POP    DPL
        POP    DPH
        RET
END

```

ผลการทดลอง

เมื่อได้ทำการทดลองโดยใช้โปรแกรมข้างต้นนั้น พบว่าที่อุปกรณ์แสดงผล LCD แสดงค่าเวลาออกมา โดยเวลาในหน่วยวินาทีนั้น เปลี่ยนแปลงค่าตามเวลาจริง จึงสามารถสรุปได้ว่าอุปกรณ์ให้สัญญาณนาฬิกาสามารถทำงานได้ ซึ่งสามารถจะนำไปใช้งานได้โปรแกรมหลักของอุปกรณ์ควบคุมสัญญาณไฟจราจรต่อไป



รูปที่ 4.9 แสดงภาพจอ LCD หลังจากทำการทดลองโปรแกรมทดสอบนาฬิกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 การทดลองการทำงานของอุปกรณ์ชุดคีย์รับข้อมูล

สำหรับการทดลองนี้นั้น มีวัตถุประสงค์เพื่อทดสอบโปรแกรมที่ใช้ควบคุมการทำงานของคีย์รับข้อมูล ว่าจะสามารถใช้งานได้หรือไม่ โดยในส่วนของคีย์รับข้อมูลนั้น จะต่อเข้ากับวงจรส่วนควบคุมและประมวลผลทางพอร์ทของ 8255 ในพอร์ท C ซึ่งจะมีภาพแสดงการต่อคีย์และวงจรของคีย์รับข้อมูล ซึ่งภาพวงจรของส่วนคีย์รับข้อมูลนั้น ได้แสดงอยู่ในภาพวงจรรวมทั้งหมดที่อยู่ในบทที่ 3 แล้ว

สำหรับโปรแกรมที่ใช้ในการทดลองนี้ จะมีการทำงานคือ จะรับข้อมูลทางคีย์เข้ามาและทำการประมวลผลว่าสวิตช์ตัวไหนถูกกด แล้วจะแสดงค่าออกมาทางส่วนแสดงผล LCD เช่น ถ้ากดปุ่ม 1 ที่คีย์รับข้อมูล ที่จอ LCD ก็จะมีขึ้นเลข 1 เป็นต้น

โปรแกรมที่ใช้ในการทดลองมีดังต่อไปนี้

```
; ***** ADDRESS RAM *****
AUTSTA EQU 02C01H
MADD1 EQU 02C02H
MADD2 EQU 02C03H
MADD3 EQU 02C04H
MODE EQU 02C05H
ENT EQU 02C06H
NEW EQU 02C07H
OLD EQU 02C08H
LOCK EQU 02C09H
TM1 EQU 02C0AH
TM2 EQU 02C0BH
TM3 EQU 02C0CH
TM4 EQU 02C0DH
TM5 EQU 02C0EH
TM6 EQU 02C0FH
TM7 EQU 02C10H
TM8 EQU 02C11H
OCSTA EQU 02C12H
KEY EQU 02C13H
SMA EQU 02C14H
TEMP EQU 02C15H
ADDA1 EQU 02C16H
ADDA2 EQU 02C17H
ADDA3 EQU 02C18H
LINK EQU 02C19H
```

```
; ***** ADDRESS PORT *****
```

```
P1A EQU 08F00H
P1B EQU 08F01H
P1C EQU 08F02H
P1P EQU 08F03H
P2A EQU 06F00H
P2B EQU 06F01H
P2C EQU 06F02H
P2P EQU 06F03H
P3A EQU 04F00H
P3B EQU 04F01H
```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

P3C EQU 04F02H
P3P EQU 04F03H

JMP INT8255

ORG 0013H
JMP MAIN1

```

```
;***** INITIAL 8255 *****
```

```

INT8255:  ORG 0100H
          MOV R2,#01H
          LCALL DELAY
          MOV A,#81H
          MOV DPTR,#P1P
          MOVX @DPTR,A
          MOV DPTR,#P2P
          MOVX @DPTR,A
          MOV IE,#11000100B

```

```
;*****MAIN*****
```

```

          MOV DPTR,#P3P ;SET 8255 CONTROL PORT
          MOV A,#80H
          MOVX @DPTR,A
          MOV R2,#4 ;FIRST DELAY
          LCALL DELAY
          MOV A,#38H ;FUNCTION SET
          LCALL LCDWI
          MOV A,#10H ;CURSOR/DISPLAY
          LCALL LCDWI
          MOV A,#0EH ;DISPLAY CONTROL
          LCALL LCDWI
          MOV A,#06H ;MODE SET
          LCALL LCDWI
          MOV A,#01H ;CLEAR
          LCALL LCDWI

MAINS:   MOV A,#00101101B
          LCALL LCDWD
          MOV A,#0DH
          LCALL LCDWI

```

```

MAIN:    JMP MAIN
          JMP MAINS

```

```

MAIN1:   LCALL CHECKKEY
          MOV DPTR,#KEY
          MOVX A,@DPTR
          ANL A,#0FH ;WRITE DATA 0-12
          ORL A,#30H
          LCALL LCDWD
          MOV DPTR,#P1C
          MOV A,#0FH
          MOVX @DPTR,A
          RETI

```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DELAY:      MOV      R3,#50H
DELAY1:     MOV      R4,#0
            DJNZ    R4,$
            DJNZ    R3,DELAY1
            DJNZ    R2,DELAY
            RET

```

;***** CHECKKEY ROUTINE *****

```

CHECKKEY:   MOV      R2,#01H
            LCALL   DELAY
            MOV     DPTR,#P1C      ;READ KEYBOARD
            MOVX   A,@DPTR
            CJNE   A,#0FH,CKEY
            RET

CKEY:       MOV     A,#0FH
            MOVX   @DPTR,A
            MOVX   A,@DPTR
            JNB   ACC.3,RO3      ;JMP IF LINE 3 NOT EQUAL 0
            JNB   ACC.2,RO2      ;JMP IF LINE 2 NOT EQUAL 0
            JNB   ACC.1,RO1      ;JMP IF LINE 1 NOT EQUAL 0
            JNB   ACC.0,RO0      ;JMP IF LINE 0 NOT EQUAL 0
            RET

RO0:        JMP     ROW0
RO1:        JMP     ROW1
RO2:        JMP     ROW2
RO3:        JMP     ROW3

ROW0:       MOV     DPTR,#P1C      ;READ KEYBOARD
            MOVX   A,@DPTR
            CLR    ACC.5          ;CLEAR BIT 5
            SETB   ACC.6          ;SET BIT 6
            SETB   ACC.7          ;SET BIT 7
            MOVX   @DPTR,A        ;MOVE TO PORT C
            MOVX   A,@DPTR        ;READ KEYBOARD
            JB     ACC.0,K11       ;JMP IF LINE 0 NOT EQUAL 0
            MOV    DPTR,#KEY
            MOV    A,#0AH
            MOVX   @DPTR,A        ;MOVE #0A TO KEY
            RET

K11:        MOVX   A,@DPTR        ;READ KEYBOARD
            CLR    ACC.6          ;CLEAR BIT 6
            SETB   ACC.5          ;SET BIT 5
            MOVX   @DPTR,A        ;MOVE TO PORT C
            MOVX   A,@DPTR        ;READ KEYBOARD
            JB     ACC.0,K12       ;JMP IF LINE 0 NOT EQUAL 0
            MOV    DPTR,#KEY
            MOV    A,#0BH
            MOVX   @DPTR,A        ;MOVE #0B TO KEY
            RET

K12:        MOVX   A,@DPTR        ;READ KEYBOARD
            CLR    ACC.7          ;CLEAR BIT 7
            SETB   ACC.6          ;SET BIT 6
            MOVX   @DPTR,A        ;MOVE TO PORT C
            MOVX   A,@DPTR        ;READ KEYBOARD

```

```

JB      ACC.0,K10'      ;JMP IF LINE 0 NOT EQUAL 0
MOV     DPTR,#KEY
MOV     A,#0CH
MOVX    @DPTR,A        ;MOVE #0C TO KEY
RET

K10:    JMP      MAINS

ROW1:   MOVX     A,@DPTR      ;READ KEYBOARD
SETB    ACC.6              ;SET BIT 6
SETB    ACC.7              ;SET BIT 7
MOVX    @DPTR,A           ;MOVE TO PORT C
MOVX    A,@DPTR           ;READ KEYBOARD
JB      ACC.1,K8          ;JMP IF LINE 1 NOT EQUAL 0
MOV     DPTR,#KEY
MOV     A,#07H
MOVX    @DPTR,A          ;MOVE #07 TO KEY
RET

K8:     MOVX     A,@DPTR      ;READ KEYBOARD
CLR     ACC.6              ;CLEAR BIT 6
SETB    ACC.5              ;SET BIT 5
MOVX    @DPTR,A           ;MOVE TO PORT C
MOVX    A,@DPTR           ;READ KEYBOARD
JB      ACC.1,K9          ;JMP IF LINE 1 NOT EQUAL 0
MOV     DPTR,#KEY
MOV     A,#08H
MOVX    @DPTR,A          ;MOVE #08 TO KEY
RET

K9:     MOVX     A,@DPTR      ;READ KEYBOARD
CLR     ACC.7              ;CLEAR BIT 7
SETB    ACC.6              ;SET BIT 6
MOVX    @DPTR,A           ;MOVE TO PORT C
MOVX    A,@DPTR           ;READ KEYBOARD
JB      ACC.1,K7          ;JMP IF LINE 1 NOT EQUAL 0
MOV     DPTR,#KEY
MOV     A,#09H
MOVX    @DPTR,A          ;MOVE #09 TO KEY
RET

K7:     JMP      MAINS

ROW2:   MOVX     A,@DPTR      ;READ KEYBOARD
SETB    ACC.6              ;SET BIT 6
SETB    ACC.7              ;SET BIT 7
MOVX    @DPTR,A           ;MOVE TO PORT C
MOVX    A,@DPTR           ;READ KEYBOARD
JB      ACC.2,K5          ;JMP IF LINE 2 NOT EQUAL 0
MOV     DPTR,#KEY
MOV     A,#04H
MOVX    @DPTR,A          ;MOVE #04 TO KEY
RET

K5:     MOVX     A,@DPTR      ;READ KEYBOARD
CLR     ACC.6              ;CLEAR BIT 6
SETB    ACC.5              ;SET BIT 5

```

```

MOVX    @DPTR,A                ;MOVE TO PORT C
MOVX    A,@DPTR                ;READ KEYBOARD
JB      ACC.2,K6                ;JMP IF LINE 2 NOT EQUAL 0
MOV     DPTR,#KEY
MOV     A,#05H
MOVX    @DPTR,A                ;MOVE #05 TO KEY
RET

K6:     MOVX    A,@DPTR        ;READ KEYBOARD
CLR     ACC.7                  ;CLEAR BIT 7
SETB    ACC.6                  ;SET BIT 6
MOVX    @DPTR,A                ;MOVE TO PORT C
MOVX    A,@DPTR                ;READ KEYBOARD
JB      ACC.2,K4                ;JMP IF LINE 2 NOT EQUAL 0
MOV     A,#06H
MOV     DPTR,#KEY
MOVX    @DPTR,A                ;MOVE #06 TO KEY
RET

K4:     JMP     MAINS

ROW3:   MOVX    A,@DPTR        ;READ KEYBOARD
SETB    ACC.6                  ;SET BIT 6
SETB    ACC.7                  ;SET BIT 7
MOVX    @DPTR,A                ;MOVE TO PORT C
MOVX    A,@DPTR                ;READ KEYBOARD
JB      ACC.3,K2                ;JMP IF LINE 3 NOT EQUAL 0
MOV     DPTR,#KEY
MOV     A,#01H
MOVX    @DPTR,A                ;MOVE #01 TO KEY
RET

K2:     MOVX    A,@DPTR        ;READ KEYBOARD
CLR     ACC.6                  ;CLEAR BIT 6
SETB    ACC.5                  ;SET BIT 5
MOVX    @DPTR,A                ;MOVE TO PORT C
MOVX    A,@DPTR                ;READ KEYBOARD
JB      ACC.3,K3                ;JMP IF LINE 3 NOT EQUAL 0
MOV     DPTR,#KEY
MOV     A,#02H
MOVX    @DPTR,A                ;MOVE #02 TO KEY
RET

K3:     MOVX    A,@DPTR        ;READ KEYBOARD
CLR     ACC.7                  ;CLEAR BIT 7
SETB    ACC.6                  ;SET BIT 6
MOVX    @DPTR,A                ;MOVE TO PORT C
MOVX    A,@DPTR                ;READ KEYBOARD
JB      ACC.3,K1                ;JMP IF LINE 3 NOT EQUAL 0
MOV     A,#03H
MOV     DPTR,#KEY
MOVX    @DPTR,A                ;MOVE #03 TO KEY
RET

K1:     JMP     MAINS

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

;***** LCD ROUTINE *****

```

LCDLDS:    PUSH    DPH                ;LOAD SUB.
           PUSH    DPL
           LCALL   LCDWI              ;WRITE ADDRESS
           POP     DPL
           POP     DPH
           MOV     R2,#16             ;16 CHAR.
LCDLDS1:   CLR     A
           MOVC   A,@A+DPTR          ;MOVC FOR CODE / MOVX FOR
DATA
           PUSH    DPH
           PUSH    DPL
           LCALL   LCDWD              ;WRITE DATA
           POP     DPL
           POP     DPH
           INC     DPTR
           DJNZ   R2,LCDLDS1
           RET

```

```

LCDWI:     PUSH    DPH
           PUSH    DPL
           MOV     DPTR,#P3C          ;DATA TO PORTC
           MOVX   @DPTR,A
           MOV     DPTR,#P3B          ;PORTB READ MODIFY WRITE
           ;MOVX  A,@DPTR
           CLR    ACC.0               ;RS=0
           CLR    ACC.1               ;R/W=0
           CLR    ACC.2               ;ENABLE=0
           MOVX   @DPTR,A
           SETB   ACC.2               ;ENABLE=1
           MOVX   @DPTR,A
           CLR    ACC.2               ;ENABLE=0
           MOVX   @DPTR,A
           POP     DPL
           POP     DPH
           MOV     A,#0                ;DELAY
LCDWI1:    DEC    A
           JNZ   LCDWI1
           RET

```

```

LCDWD:     PUSH    DPH
           PUSH    DPL
           MOV     DPTR,#P3C          ;DATA TO PORTA
           MOVX   @DPTR,A
           MOV     DPTR,#P3B          ;PORTB READ MODIFY WRITE
           ;MOVX  A,@DPTR
           SETB   ACC.0               ;RS=1
           CLR    ACC.1               ;R/W=0
           CLR    ACC.2               ;ENABLE=0
           MOVX   @DPTR,A
           SETB   ACC.2               ;ENABLE=1
           MOVX   @DPTR,A
           CLR    ACC.2               ;ENABLE=0
           MOVX   @DPTR,A
           POP     DPL
           POP     DPH

```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามนำมาดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LCDWD1 :   MOV     A, #0           ; DELAY
           DEC     A
           JNZ     LCDWD1
           RET
           END

```

ผลการทดลอง

เมื่อทำการทดลองโดยใช้โปรแกรมข้างต้น พบว่าผลที่ได้จากการทดลองคือ เมื่อทำการกดสวิทช์ 1 ที่คีย์รับข้อมูลแล้ว จอ LCD จะขึ้นเลข 1 เมื่อกดสวิทช์ 2 จอ LCD จะขึ้นเลข 2 ฯลฯ ซึ่งจะเห็นได้ว่าคีย์รับข้อมูลนั้นสามารถทำงานได้อย่างถูกต้อง ซึ่งโปรแกรมนี้อาจสามารถนำไปเป็นต้นแบบในการสร้างโปรแกรมของอุปกรณ์รับข้อมูลของเครื่องควบคุมไฟจราจรได้ ต่อไป

4.7 การทดลองการทำงานของโปรแกรมควบคุมการทำงานของไฟจราจร

จากผลการทดลองทั้งหมดของการทดลองที่ผ่านมาสามารถที่จะนำวงจรทั้งหมดมาใช้งาน และทำการเขียนโปรแกรมสำหรับควบคุมสัญญาณไฟจราจรได้ ดังนั้นในการทดลองนี้ จะมีวัตถุประสงค์คือ ทดสอบโปรแกรมของเครื่องควบคุมสัญญาณไฟจราจร โดยการทำงานของเครื่องควบคุมไฟจราจรนั้น จะสามารถทำงานได้ทั้งหมด 2 โหมด คือโหมดควบคุมโดยมนุษย์ (Manual Mode) และโหมดควบคุมโดยอัตโนมัติ (Automatic Mode) วงจรที่ใช้ในการทดลองนี้ก็คือวงจรทั้งหมดซึ่งได้กล่าวไว้แล้วในบทที่ 3

สำหรับโปรแกรมที่ใช้ในการทดลองนี้ ก็คือโปรแกรมของเครื่องควบคุมสัญญาณไฟจราจรจริง ๆ เพียงแต่จะยังไม่สามารถทำงานเชื่อมต่อกับเครื่องควบคุมสัญญาณไฟจราจรเครื่องอื่น ๆ ได้เท่านั้น ซึ่งตัวโปรแกรมนั้นจะแสดงอยู่ในการทดลองสุดท้าย คือ การทดลองโปรแกรมทั้งหมดของเครื่องควบคุมสัญญาณไฟจราจร (เพียงแต่ตัดส่วนที่เป็นโปรแกรมเชื่อมต่อกับอุปกรณ์ควบคุมตัวอื่นออกเท่านั้น) สำหรับการทำงานของโปรแกรมนั้น เริ่มต้นจะทำงานตามค่าเริ่มต้นที่ตั้งไว้ และจะสามารถตั้งค่าใหม่ได้ โดยผ่านคีย์รับข้อมูล ซึ่งจะทำให้การส่งสัญญาณอินเทอร์รัปต์เมื่อมีการกดสวิทช์ แล้วเครื่องจะไปประมวลผลในโปรแกรมน้อยของการอินเทอร์รัปต์ เมื่อเสร็จสิ้นก็จะกลับมาทำงานยังโปรแกรมหลักเหมือนเดิม สำหรับตัวโปรแกรมที่ใช้ในการทดลองสามารถดูได้ใน การทดลอง 4.9

ผลการทดลอง

สำหรับในระหว่างการทดลองนี้นั้น พบว่าโปรแกรมที่ทำการเขียนขึ้นในครั้งแรก ๆ จะมีข้อผิดพลาดต่าง ๆ มากมาย ยกตัวอย่างเช่น สัญญาณไฟที่แสดงนั้นไม่ถูกต้อง คีย์รับข้อมูลไม่สามารถรับข้อมูลที่ป้อนเข้าไปได้ จอแสดงผล LCD แสดงผลไม่ถูกต้อง รวมถึงอุปกรณ์ภายนอกที่ช่วยในการทดลอง เช่น เพาเวอร์ซัพพลายจ่ายไฟไม่คงที่ ทำให้โปรแกรมเกิดการหยุดการทำงานในช่วงไฟตก ฯลฯ เป็นต้น แต่หลังจากที่ได้ทำการแก้ปัญหาต่าง ๆ ที่พบแล้วทั้งหมด ก็ได้โปรแกรมซึ่งสามารถใช้งานได้ถูกต้อง ซึ่งสำเร็จตามวัตถุประสงค์ของการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์อื่นใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.8 การทดลองการทำงานเชื่อมต่อสัญญาณระหว่างอุปกรณ์ควบคุม 2 ชุด

สำหรับการทดลองในหัวข้อนี้ นั้น มีวัตถุประสงค์เพื่อทดสอบการทำงานของวงจรในส่วนที่ใช้เชื่อมต่อสัญญาณระหว่างชุดควบคุม ได้แก่ ส่วนอินเทอร์เฟซ RS-485 โดยใช้โปรแกรมทดสอบส่งค่าจากชุดควบคุมหนึ่งไปยังอีกชุดควบคุมหนึ่ง แล้วให้ชุดควบคุมที่รับสัญญาณนั้น แสดงค่าที่รับออกมาว่าถูกต้องหรือไม่ สำหรับวงจรในส่วนที่ใช้เชื่อมต่อระหว่างชุดควบคุมสัญญาณไฟนั้น แสดงรวมอยู่ภายในวงจรรวมทั้งหมดที่อยู่ในบทที่ 3 โดยจะเชื่อมต่อ RS-485 เข้ากับ 8031 ที่ขา 10 และ 11

สำหรับโปรแกรมที่ใช้ในการทดสอบนั้น จะมีการทำงานของโปรแกรมโดยในชุดส่งจะทำการส่งค่าตัวเลข 0-9 ติดต่อกันไปเรื่อย ๆ แล้วให้ชุดรับคอยรับสัญญาณ เมื่อได้รับสัญญาณแล้วก็นำสัญญาณที่ได้มาประมวลผลและแสดงค่าออกมาทางจอ LCD ว่าค่าที่แสดงนั้นตรงกับค่าที่ส่งหรือไม่ สำหรับโปรแกรมที่ใช้ในการทดลองนี้ มีดังต่อไปนี้

โปรแกรมในส่วนของตัวส่ง

```

INI          LCALL  DELAY
            MOV    IE, #01000000B
            MOV    SCON, #10111100B ; SENDING
            MOV    DPTR, #06F03H
            MOV    A, #80H
            MOVX   @DPTR, A
            CLR    RI

START:      MOV    R0, #30H
            MOV    R1, #0AH
MAIN:      LCALL  SEND
            MOV    DPTR, #06F02H
            MOV    A, #00H
            MOVX   @DPTR, A
            JNB   RI, $
            CLR    RI
            MOV    A, #0FFH
            MOVX   @DPTR, A
            LCALL  DELAY
            INC    R0
            DJNZ  R1, MAIN
            JMP   START

SEND:      MOV    DPTR, #06F02H
            MOV    A, #0FFH
            MOVX   @DPTR, A
            MOV    SBUF, R0
HERE:      JNB   TI, HERE
            CLR    TI
            RET

```

เอกสารนี้เป็นทรัพย์สินส่วนตัวของสำนักงานที่สงวนไว้สำหรับใช้ในงานที่ R3, #00H เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 DELAY1: MOV R4, #00H
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DJNZ R4,$
DJNZ R3,DELAY1
RET

```

```
END
```

โปรแกรมในส่วนของตัวรับ

```

(PA0-PA7)   PORTC   EQU    04F02H           ;LCD DATA
(PB0-PB2)   PORTB   EQU    04F01H           ;LCD CONTROL
PORT        PORTP   EQU    04F03H           ;8255 CONTROL
           P2C     EQU    06F02H
           P2P     EQU    06F03H
           SLAM    EQU    02F02H

CONTROL PORT
           LCALL   DELAY
           MOV     DPTR,#PORTP           ;SET 8255
           MOV     A,#80H
           MOVX   @DPTR,A
           MOV     DPTR,#P2P
           MOVX   @DPTR,A
           MOV     R2,#4                 ;FIRST DELAY
           MOV     A,#38H               ;FUNCTION SET
           LCALL  LCDWI
           MOV     A,#10H
CURSOR/DISPLAY
           LCALL  LCDWI
CONTROL
           MOV     A,#0EH               ;DISPLAY
           LCALL  LCDWI
           MOV     A,#06H               ;MODE SET
           LCALL  LCDWI
           MOV     A,#01H               ;CLEAR
           LCALL  LCDWI
INI
           MOV     IE,#01000000B
           MOV     SCON,#10111100B      ;SENDING

           MOV     A,#0C4H
           LCALL  LCDWI
           MOV     A,#34H
           LCALL  LCDWD
           MOV     A,#35H
           LCALL  LCDWD
           MOV     DPTR,#SLAM
           MOV     A,#01H
           MOVX   @DPTR,A
           MOV     R1,#01H

```

MAIN:

```

CLR RI
LCALL RECIEVE
LCALL LCDWD
LCALL DELAY
LCALL SENDACK

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

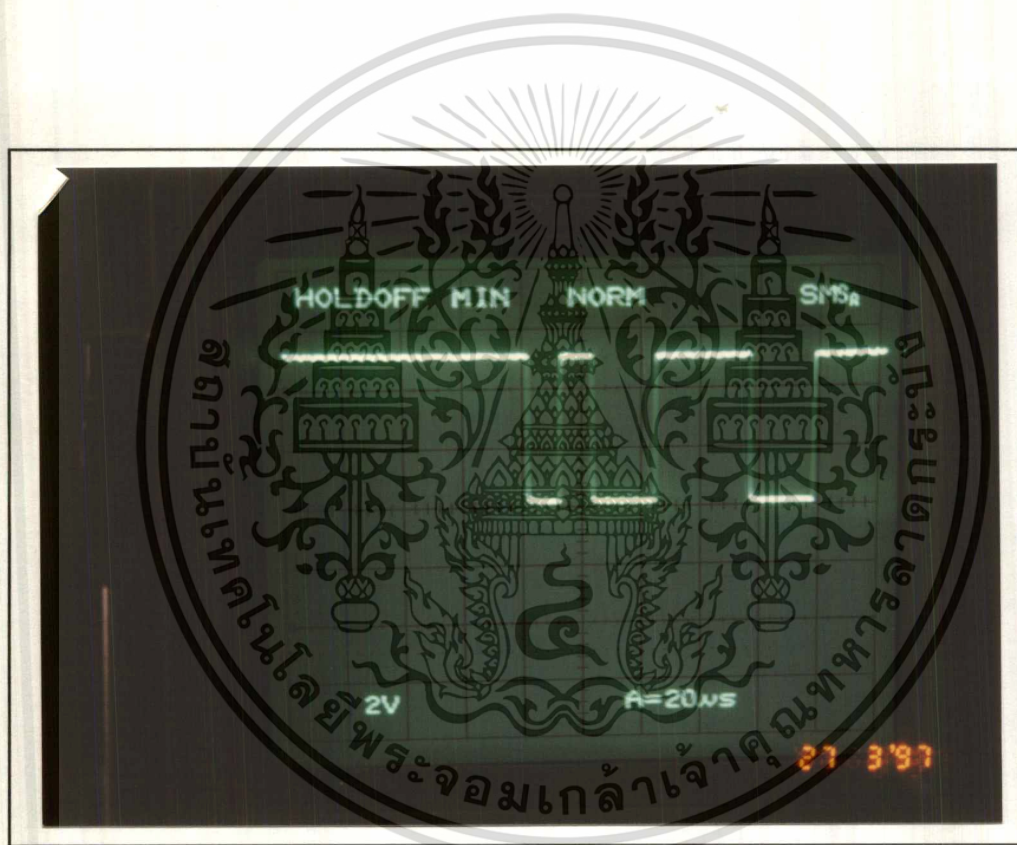
LCDWD1:      MOV     A, #03                ;DELAY
              DEC     A
              JNZ     LCDWD1
              RET

              END

```

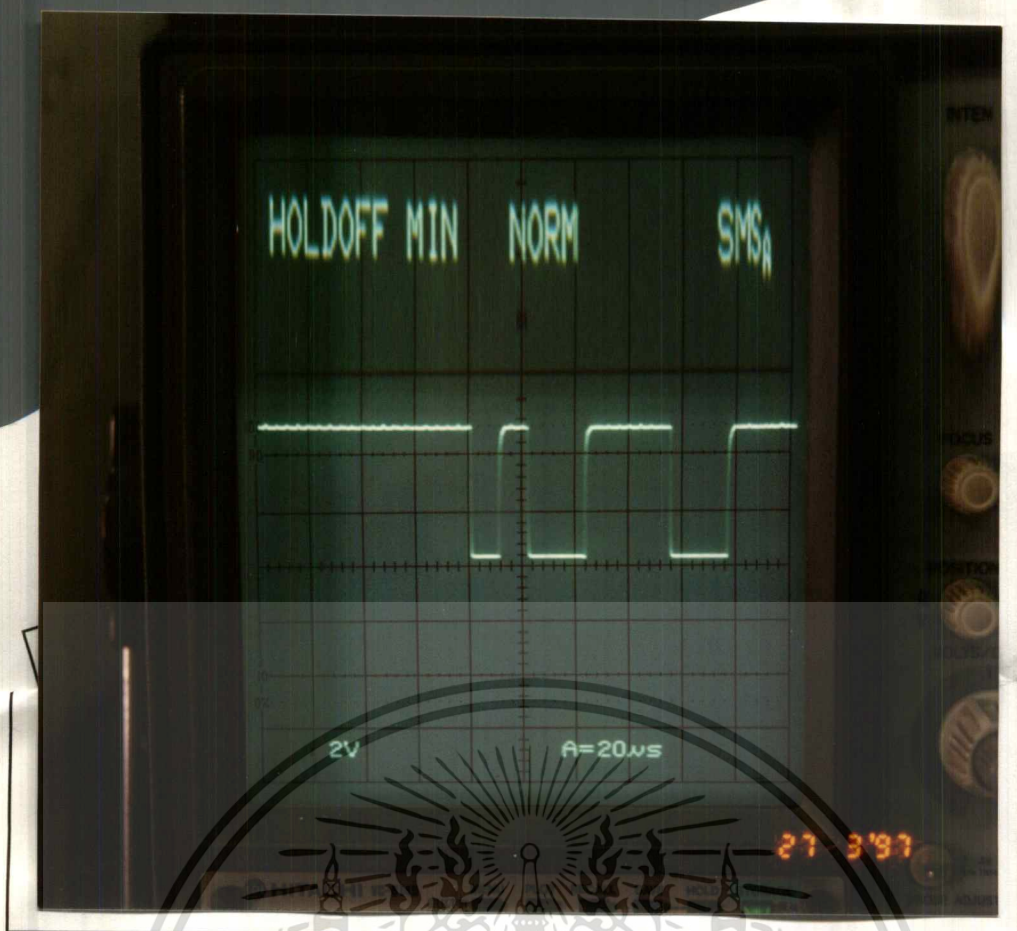
ผลการทดลอง

ภายหลังจากทำการทดลองด้วยโปรแกรมข้างต้นนั้น พบว่าจอภาพ LCD ของตัวรับจะแสดงตัวเลขตั้งแต่ 0-9 วนไปเรื่อย ๆ ซึ่งจะเห็นว่า โดยลักษณะของสัญญาณส่งที่ทำการวัดได้ในส่วนของเครื่องส่งและเครื่องรับเป็นไปดังภาพ

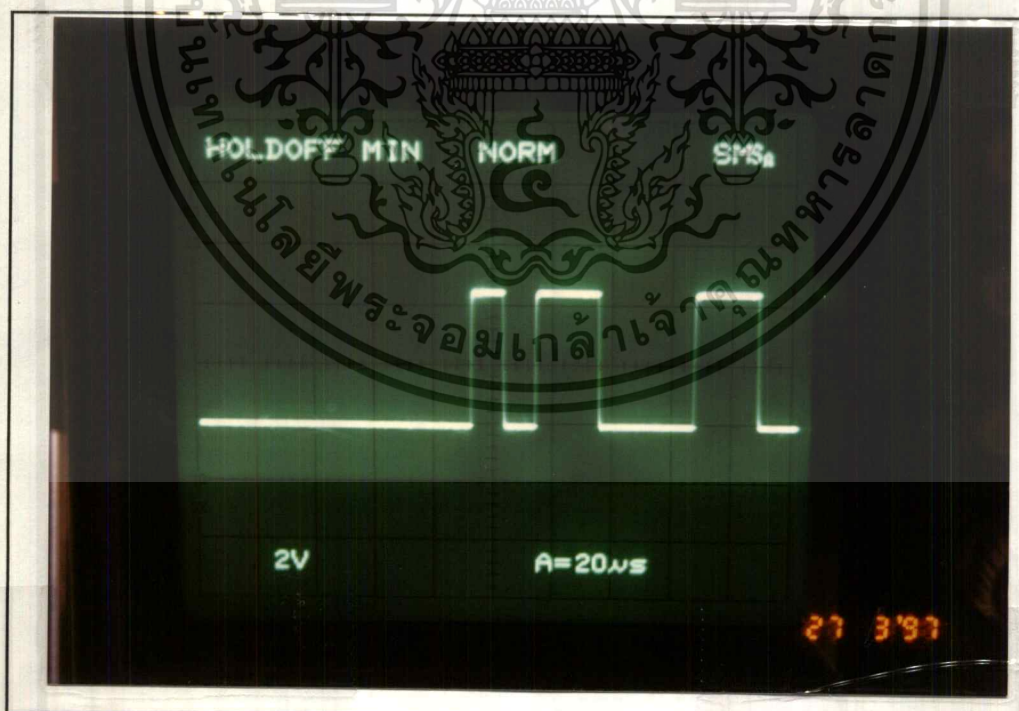


รูปที่ 4.10 แสดงลักษณะของสัญญาณที่ส่งออกจากตัวควบคุมของเครื่องส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

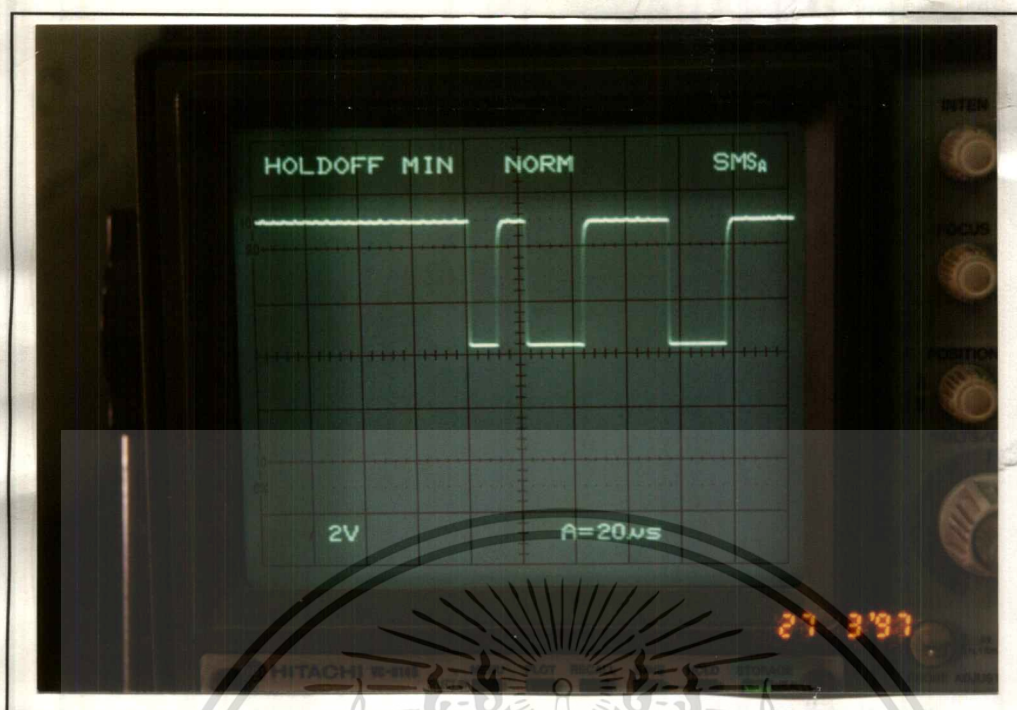


รูปที่ 4.11 แสดงลักษณะของสัญญาณที่รับได้ในตัวควบคุมของเครื่องรับ

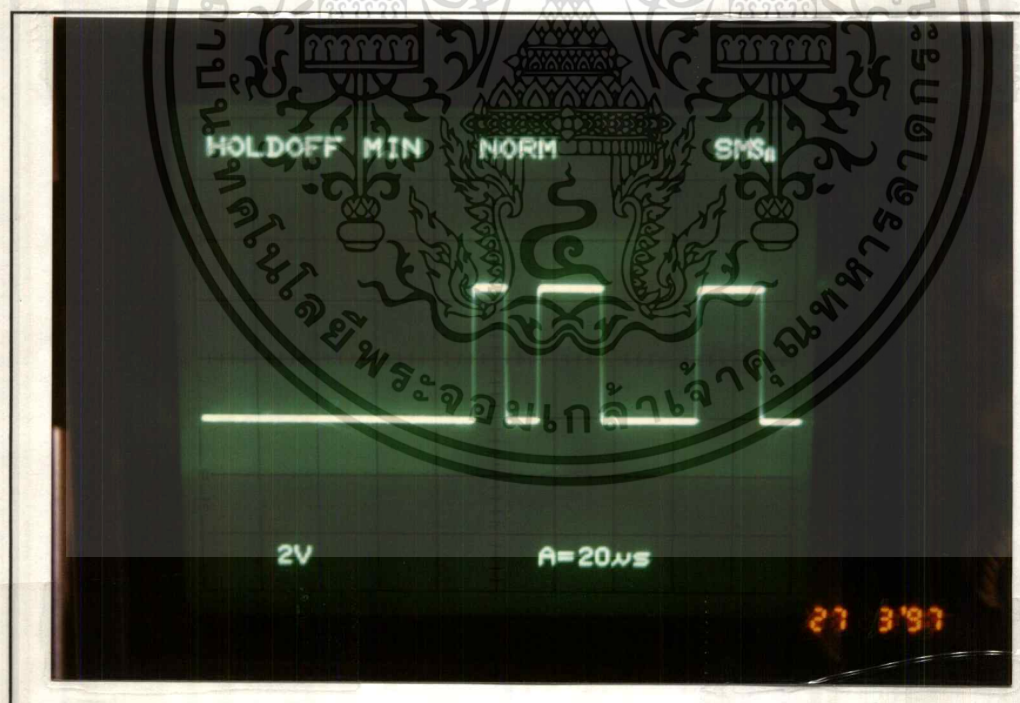


รูปที่ 4.12 แสดงลักษณะสัญญาณที่ส่งผ่านอุปกรณ์อินเทอร์เฟส RS-485

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 แสดงลักษณะของสัญญาณที่รับได้ในตัวควบคุมของเครื่องรับ



รูปที่ 4.12 แสดงลักษณะสัญญาณที่ส่งผ่านอุปกรณ์อินเทอร์เฟซ RS-485

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.9 การทดลองการทำงานของชุดควบคุมไฟจราจรที่สามารถเชื่อมต่อสัญญาณระหว่างชุดควบคุมได้

สำหรับในส่วนของการทดลองนี้ มีวัตถุประสงค์เพื่อทดลองโปรแกรมการทำงานของเครื่องควบคุมสัญญาณไฟจราจรทั้งหมด ซึ่งโปรแกรมที่ใช้ในการทดลองนี้นั้นจะเพิ่มส่วนของโปรแกรมที่ใช้สำหรับควบคุมการเชื่อมต่อสัญญาณระหว่างชุดควบคุม เพื่อให้ชุดควบคุมสามารถทำงานได้ในส่วนของการควบคุมสัญญาณไฟภายในชุดของตัวเอง และยังสามารถส่งสัญญาณเพื่อให้ชุดควบคุมชุดถัดไปทำงานให้สอดคล้องกับชุดแรกได้อีกด้วย

ขั้นตอนในการทำงานของโปรแกรมนี้นั้น ในส่วนของการควบคุมสัญญาณไฟภายในชุดของตัวเองจะทำงานเหมือนกับที่ผ่านมาในการทดลองที่ 4.7 แต่จะสามารถส่งสัญญาณไปควบคุมการทำงานของชุดควบคุมชุดถัดไปได้ โดยทำการตรวจสอบสถานะการทำงานของชุดควบคุมตัวส่งว่า อยู่ในสถานะที่ทำงานแบบเชื่อมต่อกันหรือไม่ ซึ่งในตัวรับจะต้องอยู่ในสถานะเชื่อมต่อไว้อยู่แล้ว ถ้าตัวส่งอยู่ในสถานะเชื่อมต่อกันจะเริ่มดำเนินการส่งสัญญาณ โดยการส่งข้อมูลที่เป็นแอดเดรสของชุดตัวรับเพื่อระบุให้ตัวรับทราบว่าเป็นตัวใด จากนั้นตัวส่งจะรอรับสัญญาณตอบกลับจากตัวรับว่าเป็นตัวรับที่ถูกตั้งตามที่ตั้งส่งไปหรือไม่ ถ้าถูกต้องตัวส่งก็จะทำการส่งข้อมูลกลับไปให้ตัวรับ ซึ่งข้อมูลที่ส่งจะประกอบด้วยข้อมูลซึ่งบอกว่าจะให้ตัวรับทำงานในโหมดอัตโนมัติใด และข้อมูลของเวลาของตัวส่ง ได้แก่ เวลาในหน่วยชั่วโมง นาที และวินาที จากนั้นก็จะรอสัญญาณตอบรับจากชุดตัวรับ โดยข้อมูลที่ส่งจะส่งทีละอย่างและรอสัญญาณตอบรับกลับก่อนจึงทำการส่งข้อมูลถัดไป

สำหรับโปรแกรมที่ใช้ในการทดลองมีดังต่อไปนี้

```

;***** ADDRES
RAM*****
AUTSTA EQU 02C01H
MADD1 EQU 02C02H
MADD2 EQU 02C03H
MADD3 EQU 02C04H

MODE EQU 02C05H
ENT EQU 02C06H
NEW EQU 02C07H
OLD EQU 02C08H
LOCK EQU 02C09H
OCSTA EQU 02C12H
KEY EQU 02C13H
SMA EQU 02C14H
TEMP EQU 02C15H
ADDA1 EQU 02C16H
ADDA2 EQU 02C17H
ADDA3 EQU 02C18H
LINK EQU 02C19H
MANUAL EQU 02C1AH
DAY EQU 02C1BH
TIMMIN EQU 02C1CH
TIMHOUR EQU 02C1DH
TIMMODE EQU 02C1EH
GTIME1 EQU 02C1FH
GTIME2 EQU 02C20H
GTIME3 EQU 02C21H
GTIME4 EQU 02C22H
GTIME5 EQU 02C23H
GTIME6 EQU 02C24H
GTIME7 EQU 02C25H
GTIME8 EQU 02C26H
MINUTES EQU 02C27H
AUTONF EQU 02C28H
ACCTMP EQU 02C29H
HOUR EQU 02C2AH
TENTH EQU 02C2BH
UNIT EQU 02C0AH
OLDSTA EQU 02C2CH
R3TMP EQU 02C2DH
R6TMP EQU 02C2EH
R7TMP EQU 02C2FH
MA EQU 02C30H
TIMNF EQU 02C31H
TONF EQU 02C0BH
R0TMP EQU 02C0CH
SLAVE EQU 02C32H
SLAVE1 EQU 02C33H
SLAVE2 EQU 02C34H
SLAVE3 EQU 02C35H
SLAVE4 EQU 02C36H
SLAVE5 EQU 02C37H
SLAVE6 EQU 02C38H
SLAVE7 EQU 02C39H
SR EQU 02C3AH

DAHO EQU 02C3BH
STARTL EQU 02C3CH
STEP EQU 02C3DH
SLAM EQU 02C3EH
OLDLINK EQU 02C3FH

;***** ADDRESS
PORT *****
P1A EQU 08F00H
P1B EQU 08F01H
P1C EQU 08F02H
P1P EQU 08F03H
P2A EQU 06F00H
P2B EQU 06F01H
P2C EQU 06F02H
P2P EQU 06F03H
P3A EQU 04F00H
P3B EQU 04F01H
P3C EQU 04F02H
P3P EQU 04F03H

;***** TIMER
ADDRESS *****
HOURCLK EQU 0
AF04H
MINUTECLK EQU 0
AF03H
SECOND EQU 0
AF02H
DAYCLOCK EQU 0
AF05H
INTSR EQU 0
AF10H
INTCR EQU 0
AF11H
GO EQU 0
AF15H

JMP START
ORG 0003H
JMP ISRST

ORG 0013H
JMP ISRFK

ORG 003AH
START:MOV R2,#03H
LCALL DELAY
;***** INITIAL 8255 *****

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV      A, #81H      DPTR, #MANUAL
MOV      DPTR, #P1P   MOVX      @DPTR, A
MOVX     @DPTR, A     MOV
MOV      DPTR, #P2P   DPTR, #MADD1
MOVX     @DPTR, A     MOVX     @DPTR, A
MOV      A, #80H     MOV
MOV      DPTR, #P3P   DPTR, #MADD2
MOVX     @DPTR, A     MOVX     @DPTR, A
;*** Initail program status
***
INI:     MOV      DPTR, #UNIT
MOV      A, #0AH     MOV      DPTR, #LOCK
MOVX     @DPTR, A     ;SETUP BEGIN=0FF
MOV      A, #04H     MOV      A, #0FFH
MOV      DPTR, #AUTSTA
MOVX     @DPTR, A     MOVX     @DPTR, A
MOV      DPTR, #MODE
MOVX     @DPTR, A     MOV      DPTR, #LINK
MOV      A, #00H     MOVX     @DPTR, A
;SETUP BEGIN=00     MOV      DPTR, #OLD
MOVX     @DPTR, A     MOVX     @DPTR, A
MOV      DPTR, #ENT   DPTR, #OLDSTA
MOVX     @DPTR, A     MOVX     @DPTR, A
MOV      DPTR, #DAY   MOV      DPTR, #TONF
MOVX     @DPTR, A     MOVX     @DPTR, A
MOV      DPTR, #HOUR
MOVX     @DPTR, A
MOV      DPTR, #MINUTES ;***** INITIAL GREEN
MOVX     @DPTR, A     TIME TO 30 SEC
MOV      DPTR, #TIMMIN
MOVX     @DPTR, A
MOV      DPTR, #TIMHOUR
MOVX     @DPTR, A
MOV      DPTR, #GTIME1
MOVX     @DPTR, A
MOV      DPTR, #TENTH
MOVX     @DPTR, A
MOV      DPTR, #OCSTA
MOVX     @DPTR, A
MOV      DPTR, #P2C
MOVX     @DPTR, A
MOV      A, #01H
MOVX     @DPTR, A
;SETUP BEGIN=01
MOV      DPTR, #TEMP
MOVX     @DPTR, A
MOV      DPTR, #SMA
MOVX     @DPTR, A
MOV      DPTR, #ADDA1
MOVX     @DPTR, A
MOV      DPTR, #ADDA2
MOVX     @DPTR, A
MOVX     @DPTR, A
MOV      DPTR, #ADDA3
MOVX     @DPTR, A
MOV      DPTR, #TIMMODE
MOVX     @DPTR, A
MOVX     @DPTR, A

```

เอกสารนี้เป็นเอกสารหลวงไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ;***** INITIAL CLOCK *****
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV     DPTR,#INTCR
MOV     A,#00000100B
MOVX    @DPTR,A
MOV     DPTR,#INTSR
MOVX    A,@DPTR

;***** INITIAL LCD MODULE
*****
; ** DISPLAY TIME AND STATUS
**
MOV     A,#38H
;FUNCTION SET
LCALL  LCDWI
MOV     A,#10H
;CURSOR/DISPLAY
LCALL  LCDWI
MOV     A,#0EH
;DISPLAY CONTROL
LCALL  LCDWI
MOV     A,#06H
;MODE SET
LCALL  LCDWI
MOV     A,#01H
;CLEAR
LCALL  LCDWI
LCALL  LDTD
LCALL  LMSD

;***** SET INTERRUPT INTO
FOR 1 SEC INT *****
;***** AND DISABLE
INTERRUPT FROM KEYBOARD INT1
*****

MOV     A,#11000101B
MOV     IE,A
CLR     TCON.0
CLR     TCON.2
MOV     SCON,#10111100B
;SENDING INI

;***** MAIN PROGRAM *****
MAIN: MOV     DPTR,#LINK
MOVX    A,@DPTR
CJNE   A,#00H,ASTATUS
;CHECK LINK STATUS
JMP     LMAIN
ASTATUS:MOV DPTR,#OLDLINK
MOV     A,#0FFH
MOVX    @DPTR,A
ASTA1:MOV DPTR,#AUTSTA
MOVX    A,@DPTR
CJNE   A,#0FFH,AMAIN
;CHECK AUTO STATUS

MOVX    A,@DPTR
CJNE   A,#00H,MNUAL11
MOV     A,#00H
MOV     DPTR,#LOCK
;OPEN LOCK
MOVX    @DPTR,A
;CHECK OLDSTATUS ADN CURRENT
STATUS

MNUAL11:MOV DPTR,#AUTSTA
MOVX    A,@DPTR
MOV     R0,A
MOV     DPTR,#OLDSTA
MOVX    A,@DPTR
XRL    A,R0
CJNE   A,#00H,MNEQ
JMP     MNUAL1

MNEQ: MOV     DPTR,#OLDSTA
MOV     A,R0
MOVX    @DPTR,A
MOV     A,#24H
MOV     DPTR,#P1A
MOVX    @DPTR,A
MOV     DPTR,#P1B
MOVX    @DPTR,A
MOV     DPTR,#P2A
MOVX    @DPTR,A
MOV     DPTR,#P2B
MOVX    @DPTR,A
MOV     A,#00
MOV     DPTR,#OLD
MOVX    @DPTR,A

MNUAL1: MOV     DPTR,#NEW
;CHECK
CHANGE TRAF
MOVX    A,@DPTR
MOV     R3,A
MOV     DPTR,#OLD
MOVX    A,@DPTR
XRL    A,R3
CJNE   A,#00H,CHANGE ;IF CHNGE
GOTO  CHANGE
JMP     ENDMAIN
CHANGE: MOV     A,#0FFH
;IF TRAF
CHANGE HERE
MOV     DPTR,#LOCK
MOVX    @DPTR,A
;LOCK KEY
LCALL  CHTRAF
MOV     A,#00H
MOV     DPTR,#LOCK

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ในเชิงพาณิชย์
 ไม่สามารถใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีเราไปใช้

```

MOVX    @DPTR,A
JMP     ENDMAIN

;***** END MAIN PROGRAM

ENDMAIN:      JMP     MAIN

;***** START AUTO MAIN
PROGRAM ****

AMAIN:      MOV     DPTR,#LOCK

MOV     A,#0FFH
MOVX    @DPTR,A
; LOCK KEY

MOV     DPTR,#OLDSTA ; CHECK OLD
STATUS

MOVX    A,@DPTR
; AND CURRENT STATUS

MOV     R0,A
; IF EQUAL CONTINUE
MOV     DPTR,#AUTSTA
; IF NOT, START NEW AUTO
MOVX    A,@DPTR
XRL    A,R0
; BY INITIAL TENTH
CJNE   A,#00H,AUTCHN
; AND COPY CURRENT STATUS
JMP     AUTNCH
; TO OLD STATUS

AUTCHN:MOV     A,#00H
MOV     DPTR,#TENTH
MOVX    @DPTR,A
MOV     A,#01H
MOV     DPTR,#MADD1
MOVX    @DPTR,A
MOV     DPTR,#MADD2
MOVX    @DPTR,A
MOV     DPTR,#MADD3
MOVX    @DPTR,A
MOV     DPTR,#AUTSTA
MOVX    A,@DPTR
MOV     DPTR,#OLDSTA
MOVX    @DPTR,A
MOV     A,#24H
MOV     DPTR,#P1A
MOVX    @DPTR,A
MOV     DPTR,#P1B
MOVX    @DPTR,A
MOV     DPTR,#P2A
MOVX    @DPTR,A
MOV     DPTR,#P2B
MOVX    @DPTR,A
MOV     A,#00
MOV     DPTR,#OLD
MOVX    @DPTR,A

AUTNCH:MOV     DPTR,#AUTSTA
MOVX    A,@DPTR
;**** YELLOW BLINK ****

AUTO4:CJNE   A,#04H,AUTO5
MOV     DPTR,#OCSTA
MOVX    A,@DPTR
CJNE   A,#00H,YCLOSE

YOPEN:MOV     A,#012H
;Y OPEN

MOV     DPTR,#P1A
MOVX    @DPTR,A
MOV     DPTR,#P1B
MOVX    @DPTR,A
MOV     DPTR,#P2A
MOVX    @DPTR,A
MOV     DPTR,#P2B
MOVX    @DPTR,A

MOV     DPTR,#P2B
MOVX    @DPTR,A
MOV     DPTR,#OCSTA
MOVX    A,@DPTR
CPL    A
MOVX    @DPTR,A
LCALL  DELAY
JMP     ENDMAIN

YCLOSE:      JMP     LGTCLOSE
;***** END YELLOW
BLINK

;***** R-Y BLINK
*****
AUTO5:      CJNE   A,#06H,AUTO6
MOV     DPTR,#OCSTA
MOVX    A,@DPTR
CJNE   A,#00,RYCLOSE

RYOPEN:      MOV     DPTR,#P1B
MOVX    @DPTR,A
MOV     DPTR,#P2B
MOVX    @DPTR,A
MOV     A,#12H
MOV     DPTR,#P1A
MOVX    @DPTR,A
MOV     DPTR,#P2A
MOVX    @DPTR,A
MOV     A,#0FFH
MOV     DPTR,#OLD
MOVX    @DPTR,A

```

```

MOV DPTR, #OCSTA
MOVX @DPTR, A
LCALL DELAY
JMP ENDMAIN

;***** Y-R BLINK *****
AUTO6: CJNE A, #07H, AUTO7
MOV DPTR, #OCSTA
MOVX A, @DPTR
CJNE A, #00, YRCLOSE

YROPEN: MOV A, #12H
MOV DPTR, #P1B
MOVX @DPTR, A
MOV DPTR, #P2B
MOVX @DPTR, A
MOV A, #24H
MOV DPTR, #P1A
MOVX @DPTR, A
MOV DPTR, #P2A
MOVX @DPTR, A
MOV A, #0FFH
MOV DPTR, #OCSTA
MOVX @DPTR, A
LCALL DELAY
JMP ENDMAIN

YRCLOSE: JMP LGTCLOSE

LGTCLOSE: MOV A, #00
MOV DPTR, #P1A
MOVX @DPTR, A
MOV DPTR, #P2A
MOVX @DPTR, A
MOV DPTR, #P1B
MOVX @DPTR, A
MOV DPTR, #P2B
MOVX @DPTR, A
MOV A, #00H
MOV DPTR, #OCSTA
MOVX @DPTR, A
LCALL DELAY
JMP ENDMAIN

;***** ALL RED *****
AUTO7: CJNE A, #05H, MAUTO1
MOV A, #00H
MOV DPTR, #OLD
MOVX A, #24H
MOV DPTR, #P1A
MOVX @DPTR, A
MOVX @DPTR, A
MOV DPTR, #P1B
MOVX @DPTR, A
MOV DPTR, #P2B
MOVX @DPTR, A
MOVX @DPTR, A
MOV DPTR, #P2A
MOVX @DPTR, A
MOVX @DPTR, A
MOV DPTR, #P2B
MOVX @DPTR, A
MOVX @DPTR, A
;*****
MAUTO1: MOV DPTR, #TENTH
MOVX A, @DPTR
CJNE A, #00, MAUTO111
;CHECK TIME IF <> 0
ENDPROGRAM
MOV DPTR, #AUTSTA
MOVX A, @DPTR
CJNE A, #01H, MAUTO2
MOV DPTR, #MADD1
MOVX A, @DPTR
MOV DPL, A
MOV DPH, #2DH
MOVX A, @DPTR
CJNE A, #0FFH, MAUTO11
MOV A, #01H
MOV DPTR, #MADD1
MOVX @DPTR, A
MAUTO111: JMP ENDMAIN
MAUTO11: LCALL GETNEW
LCALL CHTRAF
MOV A, #0AH
MOV DPTR, #UNIT
MOVX @DPTR, A
MOV DPTR, #TONF
MOV A, #00H
MOVX @DPTR, A
JMP ENDMAIN
MAUTO2: CJNE A, #02H, MAUTO3
MOV DPTR, #MADD2
MOVX A, @DPTR
MOV DPL, A
MOV DPH, #2EH
MOVX A, @DPTR
CJNE A, #0FFH, MAUTO21
MOV A, #01H
MOV DPTR, #MADD2
MOVX @DPTR, A
JMP ENDMAIN
MAUTO21: LCALL GETNEW
LCALL CHTRAF
MOV A, #0AH

```

```

MOV DPTR, #UNIT
MOVX @DPTR, A
MOV DPTR, #TONF
MOV A, #00H
MOVX @DPTR, A
JMP ENDMAIN

MAUTO3: MOV
DPTR, #MADD3
MOVX A, @DPTR
MOV DPL, A
MOV DPH, #2FH
MOVX A, @DPTR
CJNE
A, #0FFH, MAUTO31
MOV A, #01H
MOV
DPTR, #MADD3
MOVX @DPTR, A
JMP ENDMAIN

MAUTO31: LCALL
GETNEW
LCALL CHTRAF
MOV A, #0AH
MOV DPTR, #UNIT
MOVX @DPTR, A
MOV DPTR, #TONF
MOV A, #00H
MOVX @DPTR, A
JMP ENDMAIN

;***** END MAIN
PROGRAM *****

;***** PROGRAM GETNEW
*****
;** FOR MOV NEW AND GREEN
TIME **
GETNEW: MOV DPTR, #MANUAL
MOVX @DPTR, A
MOV DPTR, #NEW
CJNE A, #01H, GNEW2
MOV A, #88H
MOVX @DPTR, A
MOV DPTR, #GTIME1
MOVX A, @DPTR,
MOV DPTR, #TENTH
MOVX @DPTR, A
RET
GNEW2: CJNE A, #02H, GNEW3
MOV A, #22H
MOVX @DPTR, A
MOV DPTR, #GTIME2
MOVX A, @DPTR
MOV DPTR, #TENTH
MOVX @DPTR, A

MOV DPTR, #UNIT
MOVX @DPTR, A
MOV DPTR, #TONF
MOV A, #00H
MOVX @DPTR, A
JMP ENDMAIN

GNEW3: CJNE A, #03H, GNEW4
MOV A, #44H
MOVX @DPTR, A
MOV DPTR, #GTIME3
MOVX A, @DPTR
MOV DPTR, #TENTH
MOVX @DPTR, A
RET
GNEW4: CJNE A, #04H, GNEW5
MOV A, #11H
MOVX @DPTR, A
MOV DPTR, #GTIME4
MOVX A, @DPTR
MOV DPTR, #TENTH
MOVX @DPTR, A
RET
GNEW5: CJNE A, #05H, GNEW6
MOV A, #0CH
MOVX @DPTR, A
MOV DPTR, #GTIME5
MOVX A, @DPTR
MOV DPTR, #TENTH
MOVX @DPTR, A
RET
GNEW6: CJNE A, #06H, GNEW7
MOV A, #30H
MOVX @DPTR, A
MOV DPTR, #GTIME6
MOVX A, @DPTR
MOV DPTR, #TENTH
MOVX @DPTR, A
RET
GNEW7: CJNE A, #08, GNEW8
MOV A, #0C0H
MOVX @DPTR, A
MOV DPTR, #GTIME7
MOVX A, @DPTR
MOV DPTR, #TENTH
MOVX @DPTR, A
RET
GNEW8: MOV
A, #03H
MOVX @DPTR, A
MOV DPTR, #GTIME8
MOVX A, @DPTR
MOV DPTR, #TENTH
MOVX @DPTR, A
RET

;***** END GETNEW
PROGRAM *****
LMAIN: MOV
DPTR, #OLDLINK
CHKSTART STATUS

```

```

        clr          p1.1                MOVX      A,@DPTR
        MOVX        A,@DPTR            CJNE     A,#00H,S1RECV
        CJNE       A,#0FFH,LMAIN1     LCALL    SENDDATA
                                           JMP      ENDLINK

        ;**** INI STATUS
*****
        MOV        DPTR,#STARTL
        MOV        A,#00H
        MOVX       @DPTR,A
        MOV        DPTR,#DAHO
        MOVX       @DPTR,A
        MOV        DPTR,#SR
        MOVX       @DPTR,A
        MOV        DPTR,#STEP
        MOVX       @DPTR,A
        MOV        DPTR,#SLAM
        MOV        A,#01H
        MOVX       @DPTR,A

LMAIN1:
        MOV        DPTR,#STARTL
        MOVX       A,@DPTR
        CJNE      A,#0FFH,LEND1
        JMP       ENDLINK

LEND1:
        MOV        DPTR,#DAHO
        ;CHK
        DATA OR HOUR
        MOVX       A,@DPTR
        CJNE      A,#0FFH,DATALINK
        JMP       HOURLINK

DATALINK:
        MOV        DPTR,#STEP
        MOVX       A,@DPTR
        CJNE      A,#01H,DSTEP0
        STEP
        JMP       DSTEP1

DSTEP0:
        MOV        DPTR,#SR
        STEP0
        MOVX       A,@DPTR
        CJNE      A,#00H,SORECV
        LCALL     SENDADD
        JMP       ENDLINK

SORECV:JB      RI,SORECV1
        JMP      ENDLINK
SORECV1:LCALL  CHKACK
        JMP      ENDLINK

DSTEP1:MOV     DPTR,#SR
;START STEP1
        MOVX      A,@DPTR
        S1RECV:JB      RI,S1RECV1
        JMP      ENDLINK
S1RECV1:LCALL  CHKACK
        JB      P1.2,S1RECV2
        ;CHK ACK
        JMP      ENDLINK
S1RECV2:MOV   DPTR,#SLAM
        ;IF ACK GOTO NEXT SLAVE
        MOVX     A,@DPTR
        INC     A
        MOVX     @DPTR,A
        CJNE    A,#08H,S1RECV3
        ;IF COMOPLETE GOTO
        MOV     A,#01
        ;SEND HOUR
        MOVX     @DPTR,A
        MOV     DPTR,#DAHO
        MOV     A,#0FFH
        MOVX     @DPTR,A
S1RECV3:MOV   DPTR,#STEP
        MOV     A,#00H
        MOVX     @DPTR,A
        JMP     ENDLINK

HOURLINK:MOV   DPTR,#STEP
        MOVX     A,@DPTR
        CJNE    A,#00H,HSTEP1
        MOV     DPTR,#SR
        MOVX     A,@DPTR
        CJNE    A,#00H,HSOR
        LCALL   SENDADH
        JMP     ENDLINK
HSOR:JB      RI,HSOR1
        JMP     ENDLINK
HSOR1:LCALL  CHKACK
        JMP     ENDLINK

HSTEP1:CJNE   A,#01H,HSTEP2
        MOV     DPTR,#SR
        MOVX     A,@DPTR
        CJNE    A,#00H,HS1R
        LCALL   SENDHOUR
        JMP     ENDLINK
HS1R:      JB      RI,HS1R1
        JMP     ENDLINK
HS1R1:     LCALL   CHKACK
        JMP     ENDLINK

HSTEP2:      MOV     DPTR,#SR
        MOVX     A,@DPTR

```

```

      CJNE          MOVX      A,@DPTR
A,#00H,HS2R      JMP        SEND
      LCALL       SENDMIN   SDATA3:CJNE      A,#03H,SDATA4
      JMP        ENDLINK    MOV        DPTR,#SLAVE3
HS2R:           JB          MOVX      A,@DPTR
RI,HS2R1        JMP        SEND
      JMP        ENDLINK    SDATA4:CJNE      A,#04H,SDATA5
HS2R1:          LCALL      MOV        DPTR,#SLAVE4
CHKACK          MOVX      A,@DPTR
      JB         P1.2,HS2R2 JMP        SEND
HS2R2:          JMP        ENDLINK SDATA5:CJNE      A,#05H,SDATA6
DPTR,#SLAM      MOV        DPTR,#SLAVE5
      MOVX      A,@DPTR    MOVX      A,@DPTR
      INC       A          JMP        SEND
      MOVX     @DPTR,A    SDATA6:CJNE      A,#06H,SDATA7
      CJNE     A,#08H,HS2R3 MOV        DPTR,#SLAVE6
A,#08H,HS2R3   MOV        A,#01H    MOVX      A,@DPTR
      MOVX     @DPTR,A    JMP        SEND
      MOV        DPTR,#STARTL ;***** END SENDDATA ****
      MOV        A,#0FFH
      MOVX     @DPTR,A    SENDHOUR:MOV
HS2R3:          MOV        DPTR,#HOURLCK
DPTR,#STEP      MOVX      A,@DPTR
      MOV        A,#00H    JMP        SEND
      MOVX     @DPTR,A    SENDMIN:MOV
      JMP      ENDLINK    DPTR,#MINUTECLK
ENDLINK:        MOVX      A,@DPTR
ASTA1          JMP        SEND
      SEND:   MOV        R1,A
SENDADD:        MOV        A,#0FFH
DPTR,#SLAM     MOV        DPTR,#P2C
      MOVX     A,@DPTR    MOVX      @DPTR,A
      SETB    ACC.7      MOV        SBUF,R1
      JMP     SEND        JNB     TI,$
SENDADH:        MOV        CLR     TI
DPTR,#SLAM     MOV        DPTR,#SR
      MOVX     A,@DPTR    MOV        A,#0FFH
      SETB    ACC.6      MOVX     @DPTR,A
      JMP     SEND        MOV        DPTR,#P2C
SENDATA:        MOV        MOV        A,#00H
DPTR,#SLAM     MOVX     @DPTR,A
      MOVX     A,@DPTR    RET
      CJNE     A,@DPTR    CHKACK:MOV
A,#01H,SDATA2  MOV        R1,SBUF
      MOV        DPTR,#SLAVE
DPTR,#SLAVE1   MOVX     A,@DPTR
      MOVX     A,@DPTR    XRL     A,R1
      JMP     SEND        CJNE     A,#00H,NACK
SDATA2:        INC     A    MOV        DPTR,#STEP
A,#02H,SDATA3 MOVX     @DPTR,A
      MOV        DPTR,#SLAVE2 SETB    P1.2

```



```

MOV DPTR, #P1B
MOVX A, @DPTR
;MAKE R
CLR ACC.1
SETB ACC.2
MOVX @DPTR, A
MOV A, R5
JNB ACC.3, B2R
MOV DPTR, #P2A
MOVX A, @DPTR
CLR ACC.4
SETB ACC.5
;MAKE R
MOVX @DPTR, A
MOV A, R5
JNB ACC.2, B1R
MOV DPTR, #P2A
MOVX A, @DPTR
CLR ACC.1
SETB ACC.2
;MAKE R
MOVX @DPTR, A
MOV A, R5
JNB ACC.1, B0R
MOV DPTR, #P2B
MOVX A, @DPTR
CLR ACC.4
SETB ACC.5
;MAKE R
MOVX @DPTR, A
MOV A, R5
JNB ACC.0, DR
MOV DPTR, #P2B
MOVX A, @DPTR
CLR ACC.1
SETB ACC.2
;MAKE R
MOVX @DPTR, A
LCALL DELAYY
MOV DPTR, #NEW
MOVX A, @DPTR
ANL A, R0
;COMPARE XOR NEW
MOV R5, A
JNB ACC.7, B6G
MOV DPTR, #P1A
MOVX A, @DPTR
CLR ACC.5
;MAKE GREEN
SETB ACC.3
MOVX @DPTR, A
;CHANGE LIGHT
B6G: MOV A, R5
JNB ACC.6, B5G
;MAKE G
MOV DPTR, #P1A
MOVX A, @DPTR
CLR ACC.2
SETB ACC.0
MOVX @DPTR, A
;CHG ALREADY
B5G: MOV A, R5
JNB ACC.5, B4G
MOV DPTR, #P1B
MOVX A, @DPTR
CLR ACC.5
SETB ACC.3
MOVX @DPTR, A
;CHG G
B4G: MOV A, R5
JNB ACC.4, B3G
MOV DPTR, #P1B
MOVX A, @DPTR
CLR ACC.2
SETB ACC.0
;MAKE G
MOVX @DPTR, A
MOV A, R5
JNB ACC.3, B2G
MOV DPTR, #P2A
MOVX A, @DPTR
CLR ACC.5
SETB ACC.3
;MAKE G
MOVX @DPTR, A
MOV A, R5
JNB ACC.2, B1G
MOV DPTR, #P2A
MOVX A, @DPTR
CLR ACC.2
SETB ACC.0
;MAKE G
MOVX @DPTR, A
MOV A, R5
JNB ACC.1, B0G
MOV DPTR, #P2B
;MAKE G
MOVX A, @DPTR
CLR ACC.5
SETB ACC.3
MOVX @DPTR, A
B0G: MOV A, R5
JNB ACC.0, EN
MOV DPTR, #P2B
MOVX A, @DPTR
CLR ACC.2
SETB ACC.0
;MAKE G
MOVX @DPTR, A
EN: MOV DPTR, #NEW
MOVX A, @DPTR

```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการศึกษานี้เท่านั้น เพื่ออนุญาตให้ใช้โดยประชาชนด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV     DPTR, #OLD
MOVX   @DPTR, A
MOV     DPTR, #MODE
MOVX   A, @DPTR
        CJNE   A, #00, CHT1
        LCALL  LMSP
CHT1:   RET
        MOV     40
        MOV     41
        MOV     42
        DJNZ   42H, $
        DJNZ   41H, LYY1
        DJNZ   40H, LYY2
        RET
;***** INTERRUPT SERVICE
ROUTINE *****
;***** FOR KEYBOARD
INTERRUPT *****
ISRFBK:
        PUSH   DPH
        PUSH   DPL
        PUSH   ACC
        MOV    A, R6
        MOVX  @DPTR, A
        MOV    A, R7
        MOVX  @DPTR, A
        MOV    A, R3
        MOVX  @DPTR, A
        MOV    A, R0
        MOVX  @DPTR, A
        LCALL  CHECKKEY
        MOV    A, #0FH
        MOV    DPTR, #P1C
        MOVX  @DPTR, A
CHECKLOCK:
        MOV    A, @DPTR
        CJNE  A, #0FFH, CHK1T8
        JMP    CHKENT
;IF LOCK RETURN INT
CHK1T8:
        MOV    A, @DPTR
        MOV    DPTR, #NEW
        CJNE  A, #01H, CHKK2
        MOV    A, #88H
        MOVX  @DPTR, A
        MOV    DPTR, #MANUAL
        MOV    A, #01H
        MOVX  @DPTR, A
        JMP   ENDISRFBK
CHKK2:
        CJNE  A, #02, CHKK3
        MOV    A, #22H
        MOVX  @DPTR, A
        MOV    DPTR, #MANUAL
        MOV    A, #02H
        MOVX  @DPTR, A
        JMP   ENDISRFBK
CHKK3:
        CJNE  A, #03H, CHKK4
        MOV    A, #44H
        MOVX  @DPTR, A
        MOV    DPTR, #MANUAL
        MOV    A, #03H
        MOVX  @DPTR, A
        JMP   ENDISRFBK
CHKK4:
        CJNE  A, #04H, CHKK5
        MOV    A, #11H
        MOVX  @DPTR, A
        MOV    DPTR, #MANUAL
        MOV    A, #04H
        MOVX  @DPTR, A
        JMP   ENDISRFBK
CHKK5:
        CJNE  A, #05H, CHKK6
        MOV    A, #0CH
        MOVX  @DPTR, A
        MOV    DPTR, #MANUAL
        MOV    A, #05H
        MOVX  @DPTR, A
        JMP   ENDISRFBK
CHKK6:
        CJNE  A, #06H, CHKK7
        MOV    A, #30H
        MOVX  @DPTR, A
        MOV    DPTR, #MANUAL
        MOV    A, #06H
        MOVX  @DPTR, A
        JMP   ENDISRFBK
CHKK7:
        CJNE  A, #07H, CHKK8
        MOV    A, #0C0H
        MOVX  @DPTR, A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV     DPTR, #MANUAL
MOV     A, #07H
MOVX    @DPTR, A
JMP     ENDISRFBK

CHKK8 : CJNE    A, #08H, CHKENT
MOV     A, #03H
MOVX    @DPTR, A
MOV     DPTR, #MANUAL
MOV     A, #08H
MOVX    @DPTR, A
JMP     ENDISRFBK

CHKENT : MOV     DPTR, #KEY
MOVX    A, @DPTR
CJNE    A, #0CH, CHKMODE
JMP     ENTSW

CHKMODE : MOV     A, @DPTR
CJNE    A, #0BH, PMSW
JMP     MODESW

ENDPMSW : JMP     ENTSW
ENDISRFBK

PMSW : JNB     ACC.3, ENDPMSW
CLR     ACC.3
CJNE    A, #00H, PMSW1
JMP     ENDISRFBK

PMSW1 : MOV     A, @DPTR
MOV     DPTR, #MODE
MOVX    A, @DPTR
CJNE    A, #01H, CHECKPM2
LCALL   PMMODE1
JMP     ENDISRFBK

CHECKPM2 : CJNE    A, #02H, CHECKPM3
LCALL   PMMODE2
JMP     ENDISRFBK

CHECKPM3 : CJNE    A, #03H, CHECKPM4
LCALL   PMMODE3
JMP     ENDISRFBK

CHECKPM4 : CJNE    A, #04H, CHECKPM5
LCALL   PMMODE4
JMP     ENDISRFBK

CHECKPM5 : CJNE    A, #05H, CHECKPM6
LCALL   PMMODE5
JMP     ENDISRFBK

CHECKPM6 : CJNE    A, #06H, LOCKPMSW
LCALL   PMMODE6
JMP     ENDISRFBK

ENTSW : MOV     DPTR, #MODE
MOVX    A, @DPTR
CJNE    A, #01H, CHECKENTM2
LCALL   ENTM1
JMP     ENDISRFBK

CHECKENTM2 : CJNE    A, #02H, CHECKENTM3
LCALL   ENTM2
JMP     ENDISRFBK

CHECKENTM3 : CJNE    A, #03H, CHECKENTM4
LCALL   ENTM3
JMP     ENDISRFBK

CHECKENTM4 : CJNE    A, #04H, CHECKENTM5
LCALL   ENTM4
JMP     ENDISRFBK

CHECKENTM5 : CJNE    A, #05H, CHECKENTM6
LCALL   ENTM5
JMP     ENDISRFBK

CHECKENTM6 : CJNE    A, #06H, NORMALMODEENT
LCALL   ENTM6
NORMALMODEENT : JMP     ENDISRFBK

*****
;          +, - KEY ROUTINE
*****
;-----+, - MODE SET TIME-----
-----

PMMODE1 : MOV     DPTR, #ENT
MOVX    A, @DPTR
CJNE    A, #00H, A4
;CHECK ENT=MIN, JMP A4 IF NOT
MOV     DPTR, #KEY
MOVX    A, @DPTR
CJNE    A, #09H, A21
;CHECK +, JMP A21 IF NOT
MOV     DPTR, #MINUTES
MOVX    A, @DPTR

```

```

INC      A                                LCALL   CHECKHEX
DA       A                                CJNE
CJNE    A, #60H, A22                      A, #0F9H, A42    ;CHECK
;CHECK MIN=60, JMP A22 IF NOT             HOUR=99
MOV     A, #00H                            MOV      A, #23H
A22:    MOV     DPTR, #MINUTES              JMP      A42
MOVX    @DPTR, A
MOV     A, #0C8H                            A5:      MOV
LCALL   LCDWI                              DPTR, #KEY
MOV     DPTR, #MINUTES                      MOVX    A, @DPTR
MOVX    A, @DPTR                            CJNE    A, #09H, A51
LCALL   DISNUM                              ;CHECK +, JMP A51 IF NOT
MOV     A, #0C9H                            MOV     DPTR, #DAY
LCALL   LCDWI                              ;+1 DAY
MOV     A, #0DH                            MOVX    A, @DPTR
LCALL   LCDWI                              INC     A
A3:     RET                                CJNE    A, #07H, A52
A21:    MOV     DPTR, #MINUTES              ;CHECK DAY=7, JMP A52 IF NOT
MOVX    A, @DPTR                            MOV     A, #00H
DEC     A                                    JMP     A52
LCALL   CHECKHEX
CJNE    A, #0F9H, A22                      ;CHECK
MIN=99, JMP A23 IF NOT                     MOVX    A, @DPTR
MOV     A, #59H                            CJNE    A, #00H, A60
JMP     A22                                MOV     A, #07H
A4:     CJNE    A, #01H, A5                A60:    DEC     A
;CHECK                                     JMP     A52
ENT=HOUR, JMP A5 IF NOT                    A51:    JMP     A59
MOV     DPTR, #KEY                          A52:    MOV
MOVX    A, @DPTR                            DPTR, #DAY
CJNE    A, #09H, A41                        MOVX    @DPTR, A
;CHECK +, JMP A41 IF NOT                   MOV     A, #0C0H
MOV     DPTR, #HOUR                          LCALL   LCDWI
MOVX    A, @DPTR                            LCALL   WRITEDAY
INC     A                                    RET
DA     A
CJNE    A, #24H, A42
;CHECK HOUR=24
MOV     A, #00H
A42:    MOV     DPTR, #HOUR
MOVX    @DPTR, A
MOV     A, #0C5H
LCALL   LCDWI
MOV     DPTR, #HOUR
MOVX    A, @DPTR
LCALL   DISNUM
MOV     A, #0C6H
LCALL   LCDWI
MOV     A, #0DH
LCALL   LCDWI
RET
A41:    MOV     DPTR, #HOUR
MOVX    A, @DPTR
DEC     A
JMP     A41
***** +, - MODE
SET AUTO *****
PMMODE2: MOV
DPTR, #ENT
MOVX    A, @DPTR
CJNE
A, #00H, SETAUTO1 ;CHECK
ENT=0, JMP SETAUTO1 IF NOT
JMP
SELECTAUTOSET
SETAUTO1: MOV
DPTR, #SMA
MOVX    A, @DPTR
CJNE
A, #01H, SETAUTO2 ;CHECK
SMA=1, JMP SETAUTO2 IF NOT
JMP     AUTO1

```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ในการค้า

ไม่ว่ากรณีใดๆ หงสน อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SETAUTO2:          CJNE          MOV
A, #02H, SETAUTO3 ;CHECK      DPTR, #ADDA1      ;MODE 2
SMA=2, JMP SETAUTO3 IF NOT    LOWER ADDRESS FROM RAM
          JMP          AUTO2      MOVX          A, @DPTR
SETAUTO3:          CJNE          MOV          DPL, A
A, #03H, ERRORSETAUTO        MOV          DPH, #2DH
CHECK SMA=3, JMP ERRORSETAUTO ;
IF NOT                    MOV          A, R2
          JMP          AUTO3      MOVX         @DPTR, A
ERRORSETAUTO:      RET           MOV          A, DPL
                                INC          A
                                DA           A
                                CJNE         A, #00H, B3
SELECTAUTOSET:    MOV           ;CHECK PROGRAM=100, JMP KEEP
DPTR, #KEY          ;SELECT AUTO ;CHECK PROGRAM=100, JMP KEEP
SETUP              ;CHECK KEY=+, JMP - MODE AUTO
                                MOV          DPTR, #TEMP
                                MOVX         A, @DPTR
                                INC          A
                                CJNE         A, #04H, KEEP1M2
                                MOV          A, #01H
                                JMP          KEEP1M2
B1:                MOV          DPTR, #TEMP
                                MOVX         A, @DPTR
                                DEC          A
                                CJNE         A, #00H, KEEP1M2
                                MOV          A, #03H
                                ;CHECK OVERPROGRAM(=9), JMP
                                B4 IF NOT
KEEP1M2:          MOV           MOV          A, #01H
DPTR, #TEMP        ;WRITE DATA MODE AUTO TO LCD
                                MOVX         @DPTR, A
                                MOV          A, #0C9H
                                ;WRITE DATA MODE AUTO TO LCD
                                LCALL        LCDWI
                                MOV          DPTR, #TEMP
                                MOVX         A, @DPTR
                                ANL         A, #0FH
                                ORL         A, #30H
                                LCALL        LCDWD
                                MOV          A, #0C9H
                                LCALL        LCDWI
                                MOV          A, #0DH
                                LCALL        LCDWI
B11:              RET
AUTO1:            MOV           ;SET DATA
DPTR, #KEY          ;SET DATA
AUTO 1            MOVX         A, @DPTR
                                CJNE         A, #09H, B2
                                MOV          DPTR, #TEMP
                                MOVX         A, @DPTR
                                MOV          R2, A
                                MOV          DPTR, #ADDA1
                                ;MODE 2
                                LOWER ADDRESS FROM RAM
                                MOVX         A, @DPTR
                                MOV          DPL, A
                                MOV          DPH, #2DH
                                MOV          A, R2
                                MOVX         @DPTR, A
                                MOV          A, DPL
                                INC          A
                                DA           A
                                CJNE         A, #00H, B3
                                ;CHECK PROGRAM=100, JMP KEEP
                                ADDRES TO RAM
                                MOV          A, #01H
                                MOV
                                B3:          MOV
                                DPTR, #ADDA1
                                MOVX         @DPTR, A
                                JMP          B4
                                ;JMP TO WRITE LCD
                                B2:          CJNE
                                A, #0AH, ERRORSETAUTO ;CHECK
                                KEY=-, JMP ERRORSETAUTO IF
                                NOT
                                MOV          DPTR, #TEMP
                                ;READ DATAPROGRAM FROM RAM
                                MOVX         A, @DPTR
                                INC          A
                                CJNE         A, #09H, B4
                                ;CHECK OVERPROGRAM(=9), JMP
                                B4 IF NOT
                                MOV          A, #01H
                                MOVX         @DPTR, A
                                B4:          MOVX
                                @DPTR, A
                                MOV
                                DPTR, #DATA22
                                MOV          A, #90H
                                LCALL        LCDLDS
                                MOV          A, #98H
                                LCALL        LCDWI
                                MOV
                                DPTR, #ADDA1
                                MOVX         A, @DPTR
                                LCALL        DISNUM
                                MOV          A, #9BH
                                LCALL        LCDWI
                                MOV          DPTR, #TEMP
                                MOVX         A, @DPTR
                                ANL         A, #0FH
                                ORL         A, #30H
                                LCALL        LCDWD
                                MOV          A, #9BH
                                LCALL        LCDWI
                                MOV          A, #0DH
                                LCALL        LCDWI
                                RET

```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

AUTO2:          MOV          LCALL   LCDWD
DPTR,#KEY      ;SET DATA   MOV     A,#9BH
AUTO 2        MOVX     A,@DPTR  LCALL   LCDWI
              CJNE     A,#09H,B5  MOV     A,#0DH
;CHECK KEY=+,JMP KEY -  ERRORAUTO2:  RET
              MOV     DPTR,#TEMP
;MOVE DATA PROGRAM FROM RAM
              MOVX     A,@DPTR
              MOV     R2,A
              MOV
DPTR,#ADDA2
              MOVX     A,@DPTR
              MOV     DPL,A
              MOV     DPH,#2EH
              MOV     A,R2
              MOVX     @DPTR,A
              MOV     A,DPL
              INC     A
              DA     A
              CJNE     A,#00H,B6
;CHECK PROGRAM=100,JMP KEEP
DATA TO RAM
              MOV     A,#01H
B6:           MOV     MOVX     @DPTR,A
DPTR,#ADDA2   MOV     MOVX     @DPTR,A
              INC     A
              JMP     B7
;JMP TO WRITE LCD
              DA     A
              CJNE     A,#00H,B9
;CHECK PROGRAM=100,JMP KEEP
DATA TO RAM
B5:           CJNE     A,#0AH,ERRORAUTO2 ;CHECK
              KEY--,JMP ERRORAUTO2 IF NOT
              MOV     DPTR,#TEMP
              MOVX     A,@DPTR
              INC     A
              CJNE     A,#09H,B7
;CHECK OVERPROGRAM(=9);JMP
WRITE LCD IF NOT
              MOV     A,#01H
B7:           MOVX     @DPTR,A
              MOV
DPTR,#DATA22
              MOV     A,#90H
              LCALL   LCDLDS
              MOV     A,#98H
              LCALL   LCDWI
              MOV
DPTR,#ADDA2
              MOVX     A,@DPTR
              LCALL   DISNUM
              MOV     A,#9BH
              LCALL   LCDWI
              MOV     DPTR,#TEMP
              MOVX     A,@DPTR
              ANL     A,#0FH
              ORL     A,#30H
              LCALL   LCDWD
              MOV     A,#9BH
              LCALL   LCDWI
              MOV     A,#0DH
              LCALL   LCDWI
              RET
AUTO3:        MOV
DPTR,#KEY    ;SETUP DATA
AUTO 3
              MOVX     A,@DPTR
              CJNE     A,#09H,B8
;CHECK KEY=+,JMP KEY - IF
NOT
              MOV     DPTR,#TEMP
;READ DATA PROGRAM FROM RAM
              MOVX     A,@DPTR
              MOV     R2,A
              MOV
DPTR,#ADDA3
              MOVX     A,@DPTR
              MOV     DPL,A
              MOV     DPH,#2FH
              MOV     A,R2
              MOVX     @DPTR,A
              MOV     A,DPL
              INC     A
              DA     A
              CJNE     A,#00H,B9
;CHECK PROGRAM=100,JMP KEEP
DATA TO RAM
              MOV     A,#01H
B9:           MOV
DPTR,#ADDA3
              MOVX     @DPTR,A
              JMP     B10
;JMP TO WRITE LCD
B8:           CJNE     A,#0AH,ERRORAUTO3 ;CHECK
              KEY--,JMP ERRORAUTO3 IF NOT
              MOV     DPTR,#TEMP
              MOVX     A,@DPTR
              INC     A
              CJNE     A,#09H,B10
;CHECK ERRORPROGRAM,JMP
WRITE LCD IF NOT
              MOV     A,#01H
B10:        MOVX     @DPTR,A
              MOV
DPTR,#DATA22
              MOV     A,#90H
              LCALL   LCDLDS
              MOV     A,#98H
              LCALL   LCDWI
              MOV
DPTR,#ADDA3

```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOVX      A,@DPTR
LCALL    DISNUM
MOV      A,#9BH
LCALL    LCDWI
MOV      DPTR,#TEMP
MOVX     A,@DPTR
ANL      A,#0FH
ORL      A,#30H
LCALL    LCDWD
MOV      A,#9BH
LCALL    LCDWI
MOV      A,#0DH
LCALL    LCDWI
ERRORAUTO3:  RET
DATA22:  DB      "PROGRAM
"
;***** +,- MODE
SET GREEN TIMER
*****
PMMODE3:   MOV      DPTR,#ENT
MOVX     A,@DPTR
CJNE    A,#00H,D4
;CHECK ENT=0,JMP D4 IF NOT
JMP     SGT1
D4:      CJNE    ;CHECK
A,#01H,D5
ENT=1,JMP D5 IF NOT
JMP     SGT2
D5:      CJNE    ;CHECK
A,#02H,D6
ENT=2,JMP D6 IF NOT
JMP     SGT3
D6:      CJNE    ;CHECK
A,#03H,D7
ENT=3,JMP D7 IF NOT
JMP     SGT4
D7:      CJNE    ;CHECK
A,#04H,D8
ENT=4,JMP D8 IF NOT
JMP     SGT5
D8:      CJNE    ;CHECK
A,#05H,D9
ENT=5,JMP D9 IF NOT
JMP     SGT6
D9:      CJNE    ;CHECK
A,#06H,D10
ENT=6,JMP D10 IF NOT
JMP     SGT7
D10:     JMP      SGT8
D3:      RET
SGT1:    MOV      DPTR,#KEY
MOVX     A,@DPTR
CJNE    A,#09H,SGT1_1
MOV      DPTR,#TEMP
MOVX     A,@DPTR
INCRTIME
SGT1_1:  MOV
DPTR,#TEMP
MOVX     A,@DPTR
JMP     DECGTIME
SGT2:    MOV      DPTR,#KEY
MOVX     A,@DPTR
CJNE    A,#09H,SGT2_1
MOV      DPTR,#TEMP
MOVX     A,@DPTR
INCRTIME
SGT2_1:  MOV
DPTR,#TEMP
MOVX     A,@DPTR
JMP     DECGTIME
SGT3:    MOV      DPTR,#KEY
MOVX     A,@DPTR
CJNE    A,#09H,SGT3_1
MOV      DPTR,#TEMP
MOVX     A,@DPTR
INCRTIME
SGT3_1:  MOV
DPTR,#TEMP
MOVX     A,@DPTR
JMP     DECGTIME
SGT4:    MOV      DPTR,#KEY
MOVX     A,@DPTR
CJNE    A,#09H,SGT4_1
MOV      DPTR,#TEMP
MOVX     A,@DPTR
INCRTIME
SGT4_1:  MOV
DPTR,#TEMP
MOVX     A,@DPTR
JMP     DECGTIME
SGT5:    MOV      DPTR,#KEY
MOVX     A,@DPTR
CJNE    A,#09H,SGT5_1
MOV      DPTR,#TEMP
MOVX     A,@DPTR
INCRTIME
SGT5_1:  MOV
DPTR,#TEMP
MOVX     A,@DPTR
JMP     INCRTIME
SGT6:    MOV      DPTR,#KEY
MOVX     A,@DPTR
CJNE    A,#09H,SGT6_1
MOV      DPTR,#TEMP
MOVX     A,@DPTR
INCRTIME
SGT6_1:  MOV
DPTR,#TEMP
MOVX     A,@DPTR
JMP     INCRTIME
SGT7:    MOV      DPTR,#KEY
MOVX     A,@DPTR
CJNE    A,#09H,SGT7_1
MOV      DPTR,#TEMP
MOVX     A,@DPTR
INCRTIME
SGT7_1:  MOV
DPTR,#TEMP
MOVX     A,@DPTR
JMP     INCRTIME
SGT8:    MOV      DPTR,#KEY
MOVX     A,@DPTR
CJNE    A,#09H,SGT8_1
MOV      DPTR,#TEMP
MOVX     A,@DPTR
INCRTIME
SGT8_1:  MOV
DPTR,#TEMP
MOVX     A,@DPTR
JMP     INCRTIME

```

นี่เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ในการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SGT5_1:      MOV          CJNE
DPTR,#TEMP  A,#00,DECGTIME1
              MOVX      A,@DPTR          MOV      A,#99H
              JMP       DECGTIME        DECGTIME1:  MOV
                                              DPTR,#TEMP

SGT6:        MOV          MOVX      @DPTR,A
DPTR,#KEY   LCALL      OUTGTIME
              MOVX      A,@DPTR          RET
              CJNE
A,#09H,SGT6_1
              MOV       DPTR,#TEMP
              MOVX      A,@DPTR          JB
              JMP       INCGTIME        ACC.1,CONVHEX
                                              JB
SGT6_1:      MOV          ACC.2,CONVHEX
DPTR,#TEMP  CONVHEX:      CLR
              MOVX      A,@DPTR          ACC.1
              JMP       DECGTIME        CLR      ACC.2
                                              RET

SGT7:        MOV          ENDHEX:
DPTR,#KEY   OUTGTIME:      MOV
              MOVX      A,@DPTR          A,#0D0H
              CJNE
A,#09H,SGT7_1
              LCALL     LCDWI
              MOV       DPTR,#TEMP
              MOVX      A,@DPTR
              JMP       INCGTIME        LCALL     DISNUM
                                              MOV      A,#0DH
                                              LCALL     LCDWI
                                              RET

SGT7_1:      MOV          MOV
DPTR,#TEMP  A,@DPTR          DECGTIME
              MOVX      A,@DPTR          DECGTIME
              JMP       DECGTIME

SGT8:        MOV          ;***** +, - MODE
DPTR,#KEY   AUTO ON/OFF
              MOVX      A,@DPTR          *****
              CJNE
A,#09H,SGT8_1
              PPMODE4:      MOV
              MOV       DPTR,#TEMP
              MOVX      A,@DPTR          MOVX      A,@DPTR
              JMP       INCGTIME        CJNE      A,#00H,E3
                                              ;CHECK ENT=0, JMP CHECK
                                              MODEAUTO IF NOT
                                              JMP       SETAUTONF

SGT8_1:      MOV          E3:      MOV
DPTR,#TEMP  DPTR,#TEMP          ;CHECK
              MOVX      A,@DPTR          MODEAUTO
              JMP       DECGTIME        MOVX      A,@DPTR
                                              CJNE
INCGTIME:    INC          A          A,#01H,E4_1
              DA          A          JMP
              CJNE
A,#00H,INCGTIME1
              MOV       A,#01H
              MOV
INCGTIME1:  MOV          SETMODEAUTO1
DPTR,#TEMP  @DPTR,A          E4_1:      CJNE
              LCALL     OUTGTIME        A,#02H,E5
              RET          JMP
              SETMODEAUTO2
              E5:      CJNE
DECGTIME:   DEC          A          A,#03H,E6
              LCALL     CHECKHEX

```

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        JMP                MOV        A, #02H
SETMODEAUTO3          MOV        DPTR, #TEMP
E6:                   CJNE        MOVX       @DPTR, A
A, #04H, E7
        JMP                DPTR, #DATA53 ;WRITE MODE
SETMODEAUTO4          AUTO 2
E7:                   MOV        A, #90H
A, #05H, E8          LCALL       LCDLDS
        JMP                MOV        A, #9AH
SETMODEAUTO5          LCALL       LCDWI
E8:                   CJNE        MOV        DPTR, #TEMP
A, #06H, E9          MOVX       A, @DPTR
        JMP                ANL        A, #0FH
SETMODEAUTO6          ORL        A, #30H
E9:                   LCALL       LCDWD
SETMODEAUTO7          MOV        A, #9AH
        SETAUTONF:        MOV        LCALL       LCDWI
DPTR, #AUTONF        ;SET      MOV        A, #0DH
ON/OFF              LCALL       LCDWI
        MOVX       A, @DPTR      RET
CJNE        A, #0FFH, E4      E31:        MOV
MOV        A, #00H          MOV        DPTR, #TEMP
MOVX       @DPTR, A        MOVX       @DPTR, A
MOV        DPTR, #DATA51 ;WRITE AUTO ON      DPTR, #DATA54 ;WRITE MODE
ON              R/Y BLINK
        MOV        A, #0C0H      MOV        A, #90H
LCALL       LCDLDS          LCALL       LCDLDS
MOV        A, #0C7H          MOV        A, #0DH
LCALL       LCDWI          LCALL       LCDWI
MOV        A, #0CH          RET
LCALL       LCDWI
E20:              RET
E4:              MOV        SETMODEAUTO2: MOV
A, #0FFH          DPTR, #KEY
        MOVX       @DPTR, A      MOVX       A, @DPTR
MOV        ;CHECK +          CJNE        A, #09H, E41
DPTR, #DATA52 ;WRITE AUTO OFF      MOV        A, #03H
        MOV        A, #0C0H      MOV        DPTR, #TEMP
LCALL       LCDLDS          MOVX       @DPTR, A
MOV        A, #0C8H          MOV
LCALL       LCDWI          DPTR, #DATA53 ;WRITE MODE
MOV        A, #0CH          AUTO 3
LCALL       LCDWI          MOV        A, #90H
RET              LCALL       LCDLDS
        DATA51:        DB        "          MOV        DPTR, #TEMP
AUTO ON          "          MOVX       A, @DPTR
        DATA52:        DB        "          ANL        A, #0FH
AUTO OFF        "          ORL        A, #30H
        SETMODEAUTO1:  MOV        LCALL       LCDWD
DPTR, #KEY      MOV        A, #9AH
        MOVX       A, @DPTR      LCALL       LCDWI
CJNE        A, #09H, E31      MOV        A, #0DH
;CHECK +          LCALL       LCDWI
        RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่สามารถนำไปใช้ในการค้า
 ;CHECK + ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

E41:          MOV          RET
A, #01H

      MOV      DPTR, #TEMP      SETMODEAUTO4:  MOV
      MOVX    @DPTR, A          DPTR, #KEY
      MOV
DPTR, #DATA53 ;WRITE MODE          MOVX    A, @DPTR
AUTO 1          CJNE    A, #09H, E61

      MOV      A, #90H          MOV      A, #05H
      LCALL   LCDLDS          MOV      DPTR, #TEMP
      MOV      A, #9AH          MOVX    @DPTR, A
      LCALL   LCDWI          MOV
      MOV      DPTR, #TEMP      DPTR, #DATA56 ;WRITE MODE
      MOVX    A, @DPTR          RED BLINK
      ANL     A, #0FH          MOV      A, #90H
      ORL     A, #30H          LCALL   LCDLDS
      LCALL   LCDWD          MOV      A, #0DH
      MOV      A, #9AH          LCALL   LCDWI
      LCALL   LCDWI          RET
      MOV      A, #0DH      E61:          MOV      DPTR, #TEMP
      LCALL   LCDWI          A, #03H      MOVX    @DPTR, A
      RET

SETMODEAUTO3: MOV          MOV
DPTR, #KEY          DPTR, #DATA53 ;WRITE MODE
          MOVX    A, @DPTR      AUTO 3
          CJNE    A, #09H, E51  MOV      A, #90H
;CHECK +          LCALL   LCDLDS
          MOV      A, #04H      MOV      A, #09AH
          MOV      DPTR, #TEMP  LCALL   LCDWI
          MOVX    @DPTR, A      MOV      DPTR, #TEMP
          MOV          MOVX    A, @DPTR
DPTR, #DATA55 ;WRITE MODE      ANL     A, #0FH
YEL BLINK          ORL     A, #30H
          MOV      A, #90H      LCALL   LCDWD
          LCALL   LCDLDS      MOV      A, #9AH
          MOV      A, #0DH      LCALL   LCDWI
          LCALL   LCDWI      MOV      A, #0DH
          RET          LCALL   LCDWI

E51:          MOV
A, #02H

      MOV      DPTR, #TEMP      SETMODEAUTO5:  MOV
      MOVX    @DPTR, A          DPTR, #KEY
      MOV
DPTR, #DATA53 ;WRITE MODE          MOVX    A, @DPTR
AUTO 2          CJNE    A, #09H, E71
          MOV      A, #90H      MOV      A, #06H
          LCALL   LCDLDS      MOV      DPTR, #TEMP
          MOV      A, #9AH      MOVX    @DPTR, A
          LCALL   LCDWI      MOV
          MOV      DPTR, #TEMP      DPTR, #DATA57 ;WRITE MODE
          MOVX    A, @DPTR      Y/R BLINK
          ANL     A, #0FH      MOV      A, #90H
          ORL     A, #30H      LCALL   LCDLDS
          LCALL   LCDWD      MOV      A, #0DH
          MOV      A, #9AH      LCALL   LCDWI
          LCALL   LCDWI      RET
          MOV      A, #0DH      E71:          MOV
          LCALL   LCDWI          A, #04H
          MOV      DPTR, #TEMP

```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับใช้งานเพื่อการศึกษามาก่อน ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOVX @DPTR,A
MOV DPTR,#DATA55 ;WRITE MODE
YEL BLINK
MOV A,#90H
LCALL LCDLDS
MOV A,#0DH
LCALL LCDWI
RET

E91:
MOV DPTR,#TEMP
MOVX @DPTR,A
MOV DPTR,#DATA57 ;WRITE MODE
Y/R BLINK
MOV A,#90H
LCALL LCDLDS
MOV A,#0DH
LCALL LCDWI
RET

SETMODEAUTO6:
MOV DPTR,#KEY
MOVX A,@DPTR
CJNE A,#09H,E81
;CHECK +
MOV A,#07H
MOV DPTR,#TEMP
MOVX @DPTR,A
MOV DPTR,#DATA54 ;WRITE MODE
R/Y BLINK
MOV A,#90H
LCALL LCDLDS
MOV A,#0DH
LCALL LCDWI
RET

E81:
MOV DPTR,#TEMP
MOVX @DPTR,A
MOV DPTR,#DATA56 ;WRITE MODE
RED BLINK
MOV A,#90H
LCALL LCDLDS
MOV A,#0DH
LCALL LCDWI
RET

SETMODEAUTO7:
MOV DPTR,#KEY
MOVX A,@DPTR
CJNE A,#09H,E91
;CHECK +
MOV A,#01H
MOV DPTR,#TEMP
MOVX @DPTR,A
MOV DPTR,#DATA53 ;WRITE MODE
AUTO 1
MOV A,#90H
LCALL LCDLDS
MOV A,#9AH
LCALL LCDWI
MOV DPTR,#TEMP
MOVX A,@DPTR
ANL A,#0FH
ORL A,#30H

LCALL LCDWD
MOV A,#9AH
LCALL LCDWI
MOV A,#0DH
LCALL LCDWI
RET

E91:
MOV DPTR,#TEMP
MOVX @DPTR,A
MOV DPTR,#DATA57 ;WRITE MODE
Y/R BLINK
MOV A,#90H
LCALL LCDLDS
MOV A,#0DH
LCALL LCDWI
RET

DATA53:
DB "
MODE AUTO
DATA54:
DB "
MODE R/Y BLINK
DATA55:
DB "
MODE YEL BLINK
DATA56:
DB "
MODE 4 WAY RED
DATA57:
DB "
MODE Y/R BLINK

;***** +, - MODE
SET LINK *****

PMMODE5:
MOV DPTR,#TEMP
LINK
MOVX A,@DPTR
CJNE A,#0FFH,F5
;CHECK LINK ON
F6:
MOV @DPTR,A
MOV A,#0C0H
MOV DPTR,#DATA61
LCALL LCDLDS
MOV A,#0CH
LCALL LCDWI
RET
F3:
F5:
MOV @DPTR,A
MOV A,#0C0H
MOV DPTR,#DATA62
LCALL LCDLDS
MOV A,#0CH

```

```

                LCALL    LCDWI
                RET

DATA61:        DB      "SET
LINK           "
DATA62:        DB      "SET
NONLINK        "

;*****+, - MODE
SELECT
SLAVE*****

PMMODE6:      MOV
DPTR, #ENT

                MOVX    A, @DPTR
                CJNE
A, #00H, GOSETAUTOSLAVE
                JMP     SETSLAVE
GOSETAUTOSLAVE: CJNE
A, #01H, ERRORSLAVE
                JMP
SETAUTOSLAVE
ERRORSLAVE:   RET

SETSLAVE:     MOV
DPTR, #KEY
SELECT SLAVE
                MOVX    A, @DPTR
                CJNE
A, #09H, DEC SLAVE
                JMP     INCSLAVE
DEC SLAVE:    MOV
DPTR, #TEMP
                MOVX    A, @DPTR
                DEC
                A
                CJNE
A, #00H, KEEPTEMPSLAVE
                MOV
                A, #07H
                JMP
KEEPTEMPSLAVE
INCSLAVE:     MOV

                MOVX    A, @DPTR
                INC
                A
                CJNE
A, #08H, KEEPTEMPSLAVE
                MOV
                A, #01H
                JMP
KEEPTEMPSLAVE

KEEPTEMPSLAVE: MOV
DPTR, #TEMP
                MOVX    @DPTR, A
                MOV
                A, #99H
                LCALL   LCDWI
                MOV
                DPTR, #TEMP
                MOVX    A, @DPTR
                ANL
                A, #0FH
                ORL
                A, #30H
                LCALL   LCDWD
                RET

                MOVX    A, #30H
                LCALL   LCDWD
                RET

SETAUTOSLAVE: MOV
DPTR, #KEY
SELECT AUTO OF SLAVE
                MOVX    A, @DPTR
                CJNE
A, #09H, DECAUTOSLAVE
                JMP
INCAUTOSLAVE
DECAUTOSLAVE: MOV
DPTR, #TEMP
                MOVX    A, @DPTR
                DEC
                A
                CJNE
A, #00H, KEEPTEMPAUTOSLAVE
                MOV
                A, #03H
                JMP
KEEPTEMPAUTOSLAVE
INCAUTOSLAVE: MOV
DPTR, #TEMP
                MOVX    A, @DPTR
                INC
                A
                CJNE
A, #04H, KEEPTEMPAUTOSLAVE
                MOV
                A, #01H
                JMP
KEEPTEMPAUTOSLAVE
KEEPTEMPAUTOSLAVE: MOV
DPTR, #TEMP
                MOVX    @DPTR, A
                MOV
                A, #99H
                LCALL   LCDWI
                MOV
                DPTR, #TEMP
                MOVX    A, @DPTR
                ANL
                A, #0FH
                ORL
                A, #30H
                LCALL   LCDWD
                RET

;*****
;*****
;
ROUTINE
;*****
;*****
;-----ENTER
MODE SET TIME-----
---
```

เอกสารนี้เป็นเอกสารงานวิจัย งานเพื่อการ ENTM1: ัน ไม่อนุญาตให้ MOV ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOVX    A,@DPTR                RET
INC     A
CJNE   A,#03H,A7                ;-----ENTER
;CHECK ENT=3;JMP A7 IF NOT      MODE SET AUTO-----
JMP    A9                        -----
A7:    MOV
DPTR,#ENT
MOVX    @DPTR,A
CJNE   A,#01H,A8
;CHECK ENT=1;JMP A8 IF NOT
MOV    A,#0C6H
LCALL  LCDWI
MOV    A,#0DH
LCALL  LCDWI
A6:    RET
A8:    MOV
A,#0C2H
LCALL  LCDWI
MOV    A,#0DH
LCALL  LCDWI
RET
A9:    MOV
A,#0FFH                ;KEEP DATA
TO REALTIME IC
MOV
DPTR,#0AF12H
MOVX   @DPTR,A
MOV
DPTR,#MINUTES
MOVX   A,@DPTR
DA     A
MOV
DPTR,#0AF03H
MOVX   @DPTR,A
MOV    DPTR,#HOUR
MOVX   A,@DPTR
DA     A
MOV
DPTR,#0AF04H
MOVX   @DPTR,A
MOV    DPTR,#DAY
MOVX   A,@DPTR
INC    A
MOV
DPTR,#0AF05H
MOVX   @DPTR,A
MOV    A,#04H
MOV
DPTR,#0AF11H
MOVX   @DPTR,A
MOV    A,#00H
MOV    DPTR,#ENT
MOVX   @DPTR,A
LCALL  LDTD
MOV    A,#0C9H
LCALL  LCDWI
MOV    A,#0DH
LCALL  LCDWI
ENTM2: MOV
DPTR,#ENT
MOVX   A,@DPTR
CJNE   A,#00H,CHECKKEEPA1 ;CHECK
ENT=0,JMP CHECKKEEPA1 IF NOT
JMP
KEEPSETAUTO
CHECKKEEPA1: MOV
DPTR,#SMA
MOVX   A,@DPTR
CJNE   A,#01H,CHECKKEEPA2 ;CHECK
SMA=1,JMP CHECKKEEPA2 IF NOT
JMP    KEEPAUTO1
CHECKKEEPA2: CJNE
A,#02H,CHECKKEEPA3 ;CHECK
SMA=2,JMP CHECKKEEPA3 IF NOT
JMP    KEEPAUTO2
CHECKKEEPA3: CJNE
A,#03H,ERRORCHECKKEEP ;CHECK
SMA=3,JMP ERRORCHECKKEEP IF
NOT
JMP    KEEPAUTO3
ERRORCHECKKEEP: RET
KEEPSETAUTO: MOV
DPTR,#TEMP
MOVX   A,@DPTR
MOV    DPTR,#SMA
MOVX   @DPTR,A
MOV    A,#01H
MOV    DPTR,#ENT
MOVX   @DPTR,A
MOV    DPTR,#TEMP
MOVX   @DPTR,A
MOV
DPTR,#DATA22
MOV    A,#90H
LCALL  LCDLDS
MOV    A,#98H
LCALL  LCDWI
MOV
DPTR,#ADDA1
MOVX   A,@DPTR
MOV    R2,A
SWAP   A
ANL    A,#0FH
ORL    A,#30H
LCALL  LCDWD
MOV    A,R2
ANL    A,#0FH

```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของสำนักงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่โดยไม่ได้รับอนุญาต
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุเปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

MOV      A,#0C9H
LCALL   LCDWI
MOVX    A,@DPTR
ANL     A,#0FH
ORL     A,#30H
LCALL   LCDWD
MOV     DPTR,#DATABLANK
MOV     A,#90H
LCALL   LCDLDS
MOV     A,#0C9H
LCALL   LCDWI
MOV     A,#0DH
LCALL   LCDWI
RET

;***** ENTER
MODE SET GREEN TIMER
*****

ENTM3:   MOV     DPTR,#ENT
        MOVX    A,@DPTR
        CJNE   A,#00H,D1_2 ;CHECK
        ENT=0, JMP D1_2 IF NOT
        JMP    SETGREEN1
D1_2:   CJNE   A,#01H,D1_3 ;CHECK
        ENT=1, JMP D1_3 IF NOT
        JMP    SETGREEN2
D1_3:   CJNE   A,#02H,D1_4 ;CHECK
        ENT=2, JMP D1_4 IF NOT
        JMP    SETGREEN3
D1_4:   CJNE   A,#03H,D1_5 ;CHECK
        ENT=3, JMP D1_5 IF NOT
        JMP    SETGREEN4
D1_5:   CJNE   A,#04H,D1_6 ;CHECK
        ENT=4, JMP D1_6 IF NOT
        JMP    SETGREEN5
D1_6:   CJNE   A,#05H,D1_7 ;CHECK
        ENT=5, JMP D1_7 IF NOT
        JMP    SETGREEN6
D1_7:   CJNE   A,#06H,D1_8 ;CHECK
        ENT=6, JMP D1_8 IF NOT
        JMP    SETGREEN7
D1_8:   CJNE   A,#07H,D1_1 ;CHECK
        ENT=7, JMP RET IF NOT
        JMP    SETGREEN8
D1_1:   RET

SETGREEN1: MOV     DPTR,#TEMP
        MOVX    A,@DPTR
        MOV     DPTR,#GTIME1
        MOVX    @DPTR,A
        MOV     DPTR,#GTIME2
        JMP     KEEPGTIME
SETGREEN2: MOV     DPTR,#TEMP
        MOVX    A,@DPTR
        MOV     DPTR,#GTIME2
        MOVX    @DPTR,A
        MOV     DPTR,#GTIME3
        JMP     KEEPGTIME
SETGREEN3: MOV     DPTR,#TEMP
        MOVX    A,@DPTR
        MOV     DPTR,#GTIME3
        MOVX    @DPTR,A
        MOV     DPTR,#GTIME4
        JMP     KEEPGTIME
SETGREEN4: MOV     DPTR,#TEMP
        MOVX    A,@DPTR
        MOV     DPTR,#GTIME4
        MOVX    @DPTR,A
        MOV     DPTR,#GTIME5
        JMP     KEEPGTIME
SETGREEN5: MOV     DPTR,#TEMP
        MOVX    A,@DPTR
        MOV     DPTR,#GTIME5
        MOVX    @DPTR,A
        MOV     DPTR,#GTIME6
        JMP     KEEPGTIME
SETGREEN6: MOV     DPTR,#TEMP
        MOVX    A,@DPTR
        MOV     DPTR,#GTIME6
        MOVX    @DPTR,A
        MOV     DPTR,#GTIME7
        JMP     KEEPGTIME
SETGREEN7: MOV     DPTR,#TEMP
        MOVX    A,@DPTR
        MOV     DPTR,#GTIME7
        JMP     KEEPGTIME
SETGREEN8: MOV     DPTR,#TEMP
        MOVX    A,@DPTR

```

ไม่ว่าการณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV
DPTR, #GTIME7 ;-----ENTER
MOVX @DPTR, A MODE SET AUTO ON/OFF -----
MOV
DPTR, #GTIME8
JMP KEEPGTIME ENTM4: MOV
SETGREEN8: MOV DPTR, #ENT
DPTR, #TEMP MOVX A, @DPTR CJNE A, #00H, E11
MOVX A, @DPTR MOV
DPTR, #GTIME8 DPTR, #AUTONF MOVX A, @DPTR
MOVX @DPTR, A CJNE
DPTR, #GTIME1 A, #0FFH, E12 ; IF
AUTO OFF THEN RETURN
KEEPGTIME: MOVX DPTR, #AUTSTA ; IF
A, @DPTR MOV
MOV DPTR, #TEMP AUTO ON THEN CONTINUE
MOVX @DPTR, A MOVX @DPTR, A
MOV DPTR, #ENT RET
MOVX A, @DPTR E12: MOV
INC A A, #01H ;
MOVX @DPTR, A AUTO ON
CJNE MOV DPTR, #ENT
A, #08H, D2_1 ;CHECK MOVX @DPTR, A
ENT=8, JMP D2_1 IF NOT ;SET RAM ENT=01
MOV A, #00H MOV DPTR, #TEMP
MOV DPTR, #ENT MOVX @DPTR, A
MOVX @DPTR, A ;SET RAM TEMP=01
LCALL D2_1 MOV DPTR, #MA
MOV A, #00H MOVX A, @DPTR
MOV DPTR, #ENT INC A
MOVX @DPTR, A CJNE
MOV A, #01H A, #01H, E12_1
MOV DPTR, #TEMP JMP
MOVX @DPTR, A SETMODEAUTO1
RET E12_1: CJNE
D2_1: MOV A, #02H, E13
A, #0D0H JMP
LCALL LCDWI SETMODEAUTO2
;MOVE CURSOR TO LINE 4. E13: CJNE
MOV DPTR, #TEMP A, #03H, E14
MOVX A, @DPTR JMP
;READ TEMP SETMODEAUTO3
LCALL DISNUM E14: CJNE
MOV A, #97H A, #04H, E15
LCALL LCDWI JMP
MOV DPTR, #ENT SETMODEAUTO4
MOVX A, @DPTR E15: CJNE
INC A A, #05H, E16
ANL A, #0FH JMP
ORL A, #30H SETMODEAUTO5
LCALL LCDWD E16: CJNE
MOV A, #0D2H A, #06H, E17
LCALL LCDWI JMP
MOV A, #0DH SETMODEAUTO6
LCALL LCDWI E17: A, #07H, E10
RET CJNE

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้เผยแพร่โดยไม่ได้รับอนุญาต
 ไม่ว่าการมีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                JMP
SETMODEAUTO7
E10:            JMP
SETMODEAUTO1

E11:            MOV
DPTR, #TEMP

                MOVX    A, @DPTR
                MOV
DPTR, #AUTSTA
                MOVX    @DPTR, A
                MOV     DPTR, #MA
                MOVX    @DPTR, A
                MOV     A, #01H
                MOV     DPTR, #TEMP
;RESET RAM TEMPORARY
                MOVX    @DPTR, A
                MOV     A, #00H
                MOV     DPTR, #ENT
;RESET RAM ENTER
                MOVX    @DPTR, A
                MOV
DPTR, #DATABLANK
                MOV     A, #90H
                LCALL   LCDLDS
                MOV     A, #0C7H
                LCALL   LCDWI
                MOV     A, #0DH
                LCALL   LCDWI
                RET

;***** ENTER MODE
SET LINK/NONLINK
*****

ENTM5:         MOV
DPTR, #TEMP

                MOVX    A, @DPTR
                MOV     DPTR, #LINK
                MOVX    @DPTR, A
                MOV     DPTR, #ENT
                MOV     A, #00H
                MOVX    @DPTR, A
;RESET ENT=0
                MOV     A, #0CH
                LCALL   LCDWI
                RET

;*****ENTER
MODE SELECT
SLAVE*****

ENTM6:         MOV
DPTR, #ENT

                MOVX    A, @DPTR
                CJNE
A, #00H, KEEPAUTOSLAVE
                JMP
GOKEEPSLAVE
KEEPAUTOSLAVE: CJNE
A, #01H, FENT
                JMP
GOKEEPAUTOSLAVE
FENT:          RET

                MOV     DPTR, #ENT
                MOVX    @DPTR, A
                ;SET ENT=01
                MOV     DPTR, #TEMP
                MOVX    A, @DPTR
                MOV
DPTR, #SLAVE
                MOVX    @DPTR, A
                ;KEEP SLAVE
                MOV
DPTR, #DATABLANK
                MOV     A, #0C0H
                LCALL   LCDLDS
                ;CLEAR SCREEN
                MOV     A, #0C0H
                LCALL   LCDWI
                MOV
DPTR, #DATASLAVE1
                LCALL   TDISPLAY
                ;WRITE "SLAVE "
                MOV
DPTR, #SLAVE
                MOVX    A, @DPTR
                ANL     A, #0FH
                ORL     A, #30H
                LCALL   LCDWD
                MOV     A, #90H
                LCALL   LCDWI
                MOV
DPTR, #SLAVE
                MOVX    A, @DPTR
                ;READ SLAVE
                CJNE
A, #01H, CHECKSLAVE2
                MOV
DPTR, #SLAVE1
                MOVX    A, @DPTR
                JMP
WRITEAUTOSLAVE
CHECKSLAVE2:   CJNE
A, #02H, CHECKSLAVE3
                MOV
DPTR, #SLAVE2
                MOVX    A, @DPTR

```

เอกสารนี้เป็นเอกสารใช้งานเพื่อการศึกษาเท่านั้น เมื่อใช้กรุณาไปใช้ประโยชน์ภายใต้การดูแล
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        JMP                MOVX        A, @DPTR
WRITEAUTOSLAVE                MOV
CHECKSLAVE3 : CJNE          DPTR, #SLAVE2
A, #03H, CHECKSLAVE4        MOVX        @DPTR, A
        MOV                JMP
DPTR, #SLAVE3                WRITEAUTOSLAVE1
        MOVX        A, @DPTR    CHECKAUTOSLAVE3 : CJNE
        JMP                A, #03H, CHECKAUTOSLAVE4
WRITEAUTOSLAVE                MOV        DPTR, #TEMP
CHECKSLAVE4 : CJNE          MOVX        A, @DPTR
A, #04H, CHECKSLAVE5        MOV
DPTR, #SLAVE4                DPTR, #SLAVE3
        MOVX        @DPTR, A
        JMP                MOVX        @DPTR, A
WRITEAUTOSLAVE1                JMP
CHECKAUTOSLAVE4 : CJNE          WRITEAUTOSLAVE1
A, #04H, CHECKAUTOSLAVE5    CHECKAUTOSLAVE4 : CJNE
        MOV        DPTR, #TEMP    A, #04H, CHECKAUTOSLAVE5
        MOVX        A, @DPTR        MOV
        MOV                DPTR, #SLAVE4
        MOVX        @DPTR, A        MOVX
        JMP                @DPTR, A
WRITEAUTOSLAVE1                JMP
CHECKAUTOSLAVE5 : CJNE          WRITEAUTOSLAVE1
A, #05H, CHECKAUTOSLAVE6    CHECKAUTOSLAVE5 : CJNE
        MOV        DPTR, #TEMP    A, #05H, CHECKAUTOSLAVE6
        MOVX        A, @DPTR        MOV
        MOV                DPTR, #TEMP
        MOVX        A, @DPTR        MOVX
        MOV                A, @DPTR
        MOV                DPTR, #SLAVE5
        MOVX        @DPTR, A        MOVX
        JMP                @DPTR, A
WRITEAUTOSLAVE1                WRITEAUTOSLAVE1
CHECKSLAVE7 : MOV            CHECKAUTOSLAVE6 : CJNE
DPTR, #SLAVE7                A, #06H, CHECKAUTOSLAVE7
        MOVX        A, @DPTR        MOV
        JMP                DPTR, #TEMP
WRITEAUTOSLAVE                MOVX        A, @DPTR
CHECKSLAVE7 : MOV            MOVX
DPTR, #SLAVE7                A, @DPTR
        MOVX        A, @DPTR        MOV
        JMP                DPTR, #SLAVE6
WRITEAUTOSLAVE                WRITEAUTOSLAVE1
WRITEAUTOSLAVE : MOV        CHECKAUTOSLAVE6 : CJNE
DPTR, #TEMP                A, #06H, CHECKAUTOSLAVE7
        MOVX        @DPTR, A        MOV
        ANL        A, #0FH        MOVX
        ORL        A, #30H        A, @DPTR
        LCALL     LCDWD        MOV
        RET                DPTR, #SLAVE6
        MOVX        @DPTR, A        MOVX
        ANL        A, #0FH        @DPTR, A
        ORL        A, #30H        JMP
        LCALL     LCDWD        WRITEAUTOSLAVE1
        RET                CHECKAUTOSLAVE7 : MOV
        GOKEEPAUTOSLAVE : MOV    DPTR, #TEMP
        DPTR, #SLAVE                MOVX
        MOVX        A, @DPTR        A, @DPTR
        CJNE          MOV
A, #01H, CHECKAUTOSLAVE2    DPTR, #SLAVE7
        MOV        DPTR, #TEMP        MOVX
        MOVX        A, @DPTR        @DPTR, A
        MOV                JMP
DPTR, #SLAVE1                WRITEAUTOSLAVE1
        MOVX        @DPTR, A        WRITEAUTOSLAVE1 : MOV
        JMP                DPTR, #DATABLANK
WRITEAUTOSLAVE1                MOV
CHECKAUTOSLAVE2 : CJNE          A, #0C0H
A, #02H, CHECKAUTOSLAVE3    LCALL     LCDLDS
        MOV                ; CLEAR LINE 2
        DPTR, #DATABLANK        MOV
        MOV                DPTR, #DATABLANK
        MOV                MOV
        A, #90H                A, #90H

```

ไม่ว่ากรณีใดๆ หวังสน อีกทั้งนี้ ผมมีเทตแบบสงเนื้อหา และต้องอ้างอิงถึงเจ้า ของเอกสารทุกที่ที่มีมีการนำไปใช้

```

                LCALL    LCDLDS
;CLEAR LINE 3
                MOV     A,#0C0H
                LCALL   LCDWI
                MOV     A,#00H
DPTR,#DATA72
DATA "SET SLAVE "
                LCALL   TDISPLAY
                MOV     DPTR,#SLAVE
                MOVX    A,@DPTR
                MOV     DPTR,#TEMP
                MOVX    @DPTR,A
;KEEP SLAVE TO TEMP
                ANL    A,#0FH
                ORL    A,#30H
                LCALL   LCDWD
                MOV     A,#00H
                MOV     DPTR,#ENT
                MOVX    @DPTR,A
;RESET ENT=0
                RET

DATASLAVE1:    DB      "
SLAVE ."
DATASLAVE2:    DB      "SET
AUTO ."

;*****
;*****
;
ROUTINE
;*****
;*****
;*****MODE
SWITCH*****
MODESW:        MOV     DPTR,#MODE
                MOVX    A,@DPTR
                INC     A
                MOVX    @DPTR,A
                CJNE   A,#07H,CHECKMODE
                ;CHECK
SCREEN=FREE STATUS;JMP
CHECKMODE1 IF NOT
                MOV     A,#00H
                MOVX    @DPTR,A
                MOV     A,#01H
                LCALL   LCDWI
;CLEAR DISPLAY
                MOV     R2,#01H
                LCALL   DELAY
                LCALL   LDTD
                LCALL   LMSD
                JMP     ENDISRFK
CHECKMODE:
                MOV     A,#00H
                MOV     DPTR,#ENT
                MOVX    @DPTR,A
                MOV     DPTR,#MODE
                MOVX    A,@DPTR
                LCALL   MODEW
                JMP     ENDISRFK
MODEW:
                CJNE   A,#01H,CHECKMODE2
                ;
CHECK SCREEN=MODE1;JMP
CHECKMODE2 IF NOT
                JMP     MODE1
CHECKMODE2:
                CJNE   A,#02H,CHECKMODE3
                JMP     MODE2
CHECKMODE3:
                CJNE   A,#03H,CHECKMODE4
                JMP     MODE3
CHECKMODE4:
                CJNE   A,#04H,CHECKMODE5
                JMP     MODE4
CHECKMODE5:
                CJNE   A,#05H,CHECKMODE6
                JMP     MODE5
CHECKMODE6:
                CJNE   A,#06H,ERRORCHECKMODE
                MOV     DPTR,#LINK
                MOVX    A,@DPTR
                CJNE   A,#00H,JMPSTATUSMODE
                JMP     MODE6
JMPSTATUSMODE:
                JMP     MODESW
ERRORCHECKMODE:
                RET
;*****NORMAL
STATUS MODE
1*****
MODE1:
                MOV     DPTR,#HOURCLK
                MOVX    A,@DPTR
                MOV     DPTR,#HOUR
                MOVX    @DPTR,A
                MOV     DPTR,#MINUTECLK
                MOVX    A,@DPTR
                MOV     DPTR,#MINUTES
                MOVX    @DPTR,A
                MOV     A,#0C0H

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ในการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV
DPTR, #DATM11          MOV          A, #01001111B ;O
LCALL LCDLDS           LCALL LCDWD
MOV A, #90H
MOV
DPTR, #DATM12          MOV          A, #01001110B ;N
LCALL LCDLDS           LCALL LCDWD
MOV A, #0D0H           RET
MOV
DPTR, #DATBLANK        X54:          CJNE
MOV A, #0C0H           A, #02H, X55 ;CHECK TUE
LCALL LCDLDS           MOV
MOV R2, #06H           A, #01010100B ;T
LCALL DELAY            LCALL LCDWD
MOV A, #0C0H           MOV
MOV A, #01010101B ;U
DPTR, #DATBLANK        LCALL LCDWD
LCALL LCDLDS           MOV
MOV A, #90H           A, #01000101B ;E
MOV A, #90H           LCALL LCDWD
MOV A, #90H           RET
DPTR, #DATBLANK        X55:          CJNE
LCALL LCDLDS           A, #03H, X56 ;CHECK WED
MOV A, #90H           MOV
DPTR, #DAYCLOCK        ; A, #01010111B ;W
READ DAY FROM IC      LCALL LCDWD
MOVX A, @DPTR          MOV
DEC A                  A, #01000101B ;E
ANL A, #00000111B     LCALL LCDWD
MOV DPTR, #DAY         MOV
MOVX @DPTR, A          A, #01000100B ;D
MOV A, #0C0H           LCALL LCDWD
LCALL LCDWI           RET
LCALL WRITEDAY        X56:          CJNE
JMP X59                A, #04H, X57 ;CHECK THU
;JMP TO WRITE CLOCK   MOV
MOV A, #01010100B ;T
LCALL LCDWD
WRITEDAY:             MOV
DPTR, #DAY             MOV
MOVX A, @DPTR          A, #01001000B ;H
CJNE A, #00H, X53     LCALL LCDWD
;CHECK SUN            MOV
MOV A, #01010101B ;U
LCALL LCDWD           A, #01010101B ;U
MOV A, #01010011B ;S   LCALL LCDWD
LCALL LCDWD           RET
MOV A, #01010101B ;U   X57:          CJNE
LCALL LCDWD           A, #05H, X58 ;CHECK FRI
MOV A, #01001110B ;N   MOV
LCALL LCDWD           A, #01000110B ;F
RET                   LCALL LCDWD
MOV A, #01010010B ;R   MOV
LCALL LCDWD           A, #01010010B ;R
MOV A, #01001001B ;I   LCALL LCDWD
LCALL LCDWD           MOV
MOV A, #01001001B ;I   A, #01001001B ;I
LCALL LCDWD           LCALL LCDWD
RET                   RET
X53:                  CJNE
A, #01H, X54           ;CHECK MON
MOV A, #01001101B ;M   X58:          MOV
LCALL LCDWD           A, #01010011B ;S
LCALL LCDWD           LCALL LCDWD

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่สามารถนำออกจำหน่ายหรือทำซ้ำโดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV          MOV
A, #0100001B ;A          DPTR, #DATM21
LCALL       LCDWD          LCALL       LCDLDS
MOV          MOV          A, #90H
A, #01010100B ;T          MOV          MOV
LCALL       LCDWD          DPTR, #DATM22
RET          LCALL       LCDLDS
MOV          MOV          A, #0D0H
X59:        MOV          DPTR, #DATABLANK
A, #0C6H          LCALL       LCDLDS
LCALL       LCDWI          MOV          R2, #06H
MOV          MOV          LCALL       DELAY
DPTR, #0AF04H ;READ HOUR LCALL       MOV
FROM REALTIME MOV          A, #0C0H
MOVX        A, @DPTR          MOV          MOV
ANL         A, #0FH          DPTR, #DATABLANK
ORL         A, #30H          LCALL       LCDLDS
LCALL       LCDWD          MOV          A, #90H
MOV         A, #0C5H          MOV          MOV
LCALL       LCDWI          DPTR, #DATABLANK
MOVX        A, @DPTR          LCALL       LCDLDS
DA          A                MOV          A, #0C0H
SWAP       A                LCALL       LCDWI
ANL         A, #0FH          MOV          MOV
ORL         A, #30H          DPTR, #DATM23
LCALL       LCDWD          LCALL       TDISPLAY
MOV         A, #0C7H          MOV          A, #0C9H
LCALL       LCDWI          LCALL       LCDWI
MOV         ;READ MODE AUTO'S VALUE MOV          DPTR, #SMA
A, #00111010B MOVX        A, @DPTR
LCALL       LCDWD          MOV          DPTR, #TEMP
MOV         A, #0C8H          MOVX        @DPTR, A
LCALL       LCDWI          ANL         A, #0FH
MOV         ORL         A, #30H
DPTR, #0AF03H ;READ          LCALL       LCDWD
MINUTES FROM REALTIME MOV          A, #0C9H
MOVX        A, @DPTR          LCALL       LCDWI
DA          A                MOV          A, #0DH
LCALL       DISNUM          LCALL       LCDWI
MOV         A, #0DH          RET
LCALL       LCDWI          DATM21:      DB          "
RET          MODE          "
DATM11:     DB          "          DATM22:      DB          "
MODE        "          SET AUTO    "
DATM12:     DB          "          DATM23:      DB          "SET
SET TIME    "          AUTO. "
DATABLANK:  DB          "
"          ;*****NORMAL
;*****NORMAL STATUS          STATUS MODE 3*****
MODE 2*****          MODE3:      MOV
MODE2:      MOV          A, #0C0H
A, #0C0H          DPTR, #DATM41
MOV          MOV          DPTR, #DATM41
WRITE "MODE"          WRITE "MODE"

```

เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่ให้นำไปใช้ประโยชน์ทางการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

      LCALL LCDLDS
      MOV A,#90H
      MOV
DPTR,#DATM42
      LCALL LCDLDS
;WRITE "SET GREEN TIMER"
      MOV A,#0D0H
      MOV
DPTR,#DATABLANK
      LCALL LCDLDS
;CLEAR LINE 4
      MOV R2,#06H
      LCALL DELAY
      MOV A,#0C0H
      MOV
DPTR,#DATABLANK
      CLEAR LINE 2
      LCALL LCDLDS
      MOV A,#90H
      MOV
DPTR,#DATABLANK
      CLEAR LINE 3
      LCALL LCDLDS
      MOV A,#0C0H
      MOV
DPTR,#DATM42
      LCALL LCDLDS
;WRITE "SET GREEN TIMER"
      MOV A,#90H
      LCALL LCDWI
;MOVE CURSOR TO LINE 3
      MOV
DPTR,#DATM43
      LCALL TDISPLAY
;WRITE "MANUAL "
      MOV A,#97H
      LCALL LCDWI
;MOVE CURSOR TO LINE 3 COL.
8
      MOV A,#01H
      ANL A,#0FH
      ORL A,#30H
      LCALL LCDWD
;WRITE "1" AFTER MANUAL
      MOV A,#0D0H
      LCALL LCDWI
;MOVE CURSOR TO LINE 4
WRITEGTIME1:
      MOV
DPTR,#GTIME1
      MOVX A,@DPTR
      LCALL DISNUM
WRITEUNIT:
      MOV
DPTR,#DATM44
      LCALL TDISPLAY
;WRITE "0 SEC."
      MOV A,#2EH
      LCALL LCDWD
      MOV A,#0D2H
      LCALL LCDWI
      MOV A,#0DH
      LCALL LCDWI
      MOV A,#00H
      DPTR,#ENT
      MOVX @DPTR,A
;RESET RAM ENT=00
      MOV
DPTR,#GTIME1
      MOVX A,@DPTR
      MOV DPTR,#TEMP
      MOVX @DPTR,A
;MOVE DATA GTIME1 TO TEMP
      RET
DATM41: DB "
MODE "
DATM42: DB "
SET GREEN TIMER"
DATM43: DB "
MANUAL."
DATM44: DB "0
SEC."
;*****NORMAL
STATUS MODE
4*****
MODE4: MOV
A,#01H
MOV DPTR,#TEMP
MOVX @DPTR,A
MOV A,#0C0H
MOV
DPTR,#DATM51
LCALL LCDLDS
MOV A,#90H
MOV
DPTR,#DATM54
LCALL LCDLDS
MOV A,#0D0H
MOV
DPTR,#DATABLANK
LCALL LCDLDS
MOV R2,#06H
LCALL DELAY
MOV A,#0C0H
MOV
DPTR,#DATABLANK
LCALL LCDLDS
MOV A,#90H
MOV
DPTR,#DATABLANK
LCALL LCDLDS

```

```

MOV DPTR, #AUTSTA ;
READ AUTO ON/OFF
MOVX A, @DPTR
MOV DPTR, #AUTONF
MOVX @DPTR, A
CJNE A, #0FFH, ATON
JMP ATF

ATON:
MOV A, #0C0H
MOV DPTR, #DATM52 ;
WRITE AUTO ON
LCALL LCDLDS
MOV A, #0C7H
LCALL LCDWI
MOV A, #0DH
LCALL LCDWI
MOV DPTR, #AUTSTA
MOVX A, @DPTR
MOV DPTR, #MA
MOVX @DPTR, A
RET

ATF:
MOV A, #0C0H
MOV DPTR, #DATM53 ;
WRITE AUTO OFF
LCALL LCDLDS
MOV A, #0C8H
LCALL LCDWI
MOV A, #0DH
LCALL LCDWI
RET

DATM51: DB "
MODE "
DATM54: DB "
AUTO ON/OFF "
DATM52: DB "
AUTO ON "
DATM53: DB "
AUTO OFF "

;*****NORMAL
STATUS MODE
5*****

MODE5:
MOV A, #0C0H
MOV DPTR, #DATM61
LCALL LCDLDS

MOV A, #90H
MOV DPTR, #DATM62
LCALL LCDLDS
MOV A, #0D0H
MOV DPTR, #DATABLANK
LCALL LCDLDS
MOV R2, #06H
LCALL DELAY
MOV A, #0C0H
MOV DPTR, #DATABLANK
LCALL LCDLDS
MOV A, #90H
MOV DPTR, #DATABLANK
LCALL LCDLDS
MOV DPTR, #LINK
MOVX A, @DPTR
MOV DPTR, #TEMP
MOVX @DPTR, A
CJNE A, #00H, NONLINK
MOV A, #0C0H
MOV DPTR, #DATM63
LCALL LCDLDS
MOV A, #0CH
LCALL LCDWI
RET

NONLINK:
MOV A, #0C0H
MOV DPTR, #DATM64
LCALL LCDLDS
MOV A, #0CH
LCALL LCDWI
RET

DATM61: DB "
MODE "
DATM62: DB "SET
LINK/NONLINK"
DATM63: DB "SET
LINK "
DATM64: DB "SET
NONLINK "

;*****NORMAL STATUS
MODE 6*****

MODE6:
MOV DPTR, #DATM61
MOV A, #0C0H
LCALL LCDLDS
;WRITE "MODE" ให้นำไปใช้ประโยชน์ด้านการค้า
LCALL LCDLDS

```

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV DPTR, #DATA71
MOV A, #90H
LCALL LCDLDS
;WRITE "SELECT SLAVE"
MOV DPTR, #DATABLANK
MOV A, #0D0H
LCALL LCDLDS
;CLEAR LINE 4
MOV R2, #06H
LCALL DELAY
MOV DPTR, #DATABLANK
MOV A, #0C0H
LCALL LCDLDS
;CLEAR LINE 2
MOV DPTR, #DATABLANK
MOV A, #90H
LCALL LCDLDS
;CLEAR LINE 3
MOV A, #0C0H
LCALL LCDWI
MOV DPTR, #DATA72
LCALL TDISPLAY
;WRITE "SET SLAVE "
MOV DPTR, #DATA73
LCALL TDISPLAY
;WRITE "1"
MOV A, #00H
MOV DPTR, #ENT
MOVX @DPTR, A
;RESET RAM ENT=00
MOV A, #01H
MOV DPTR, #TEMP
MOVX @DPTR, A
;RESET RAM TEMP=01
RET

MOV DPTR, #R6TMP
MOVX A, @DPTR
MOV R6, A
MOV DPTR, #R7TMP
MOVX A, @DPTR
MOV R7, A
MOV DPTR, #R0TMP
MOVX A, @DPTR
MOV R0, A
POP ACC
POP DPL
POP DPH
RETI

;***** CHECKKEY ROUTINE *****
CHECKKEY: MOV R2, #01H
LCALL DELAY
MOV DPTR, #P1C
;READ KEYBOARD
MOVX A, @DPTR
CJNE A, #0FH, CKEY
RET
CKEY: MOV A, #0FH
MOVX @DPTR, A
MOVX A, @DPTR
JNB ACC.3, RO3
;JMP IF LINE 3 NOT EQUAL 0
JNB ACC.2, RO2
;JMP IF LINE 2 NOT EQUAL 0
JNB ACC.1, RO1
;JMP IF LINE 1 NOT EQUAL 0
JNB ACC.0, RO0
;JMP IF LINE 0 NOT EQUAL 0
RET

```

```

DATA71: DB "
SELECT SLAVE "
DATA72: DB "SET
SLAVE ."
DATA73: DB "1."
RO0: JMP ROW0
RO1: JMP ROW1
RO2: JMP ROW2
RO3: JMP ROW3

```

```

;***** ENDING INTERRUPT SERVICE FUNC KEY *****
ENDISRFK: MOV DPTR, #R3TMP
MOVX A, @DPTR
MOV R3, A
;CLEAR BIT 5
SETB ACC.6
;SET BIT 6
SETB ACC.7
;SET BIT 7
MOV DPTR, #P1C
;READ KEYBOARD
MOVX A, @DPTR
CLR ACC.5

```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้เพื่อประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOVX @DPTR,A
;MOVE TO PORT C
MOVX A,@DPTR
;READ KEYBOARD
JB ACC.0,K11
;JMP IF LINE 0 NOT EQUAL 0
MOV DPTR,#KEY
MOV A,#0AH
MOVX @DPTR,A
;MOVE #0A TO KEY
RET

K11: MOVX A,@DPTR
KEYBOARD
CLR ACC.6
;CLEAR BIT 6
SETB ACC.5
;SET BIT 5
MOVX @DPTR,A
;MOVE TO PORT C
MOVX A,@DPTR
;READ KEYBOARD
JB ACC.0,K12
;JMP IF LINE 0 NOT EQUAL 0
MOV DPTR,#KEY
MOV A,#0BH
MOVX @DPTR,A
;MOVE #0B TO KEY
RET

K12: MOVX A,@DPTR
KEYBOARD
CLR ACC.7
;CLEAR BIT 7
SETB ACC.6
;SET BIT 6
MOVX @DPTR,A
;MOVE TO PORT C
MOVX A,@DPTR
;READ KEYBOARD
JB ACC.0,K10
;JMP IF LINE 0 NOT EQUAL 0
MOV DPTR,#KEY
MOV A,#0CH
MOVX @DPTR,A
;MOVE #0C TO KEY
RET

K10: RET

ROW1: MOVX A,@DPTR
KEYBOARD
SETB ACC.6
;SET BIT 6

```

```

SETB ACC.7
;SET BIT 7
MOVX @DPTR,A
;MOVE TO PORT C
MOVX A,@DPTR
;READ KEYBOARD
JB ACC.1,K8
;JMP IF LINE 1 NOT EQUAL 0
MOV DPTR,#KEY
MOV A,#07H
MOVX @DPTR,A
;MOVE #07 TO KEY
RET

K8: MOVX A,@DPTR
KEYBOARD
CLR ACC.6
;CLEAR BIT 6
SETB ACC.5
;SET BIT 5
MOVX @DPTR,A
;MOVE TO PORT C
MOVX A,@DPTR
;READ KEYBOARD
JB ACC.1,K9
;JMP IF LINE 1 NOT EQUAL 0
MOV DPTR,#KEY
MOV A,#08H
MOVX @DPTR,A
;MOVE #08 TO KEY
RET

K9: MOVX A,@DPTR
KEYBOARD
CLR ACC.7
;CLEAR BIT 7
SETB ACC.6
;SET BIT 6
MOVX @DPTR,A
;MOVE TO PORT C
MOVX A,@DPTR
;READ KEYBOARD
JB ACC.1,K7
;JMP IF LINE 1 NOT EQUAL 0
MOV DPTR,#KEY
MOV A,#09H
MOVX @DPTR,A
;MOVE #09 TO KEY
RET

K7: RET

ROW2: MOVX A,@DPTR
KEYBOARD

```

```

K11: MOVX A,@DPTR
KEYBOARD
CLR ACC.6
;CLEAR BIT 6
SETB ACC.5
;SET BIT 5
MOVX @DPTR,A
;MOVE TO PORT C
MOVX A,@DPTR
;READ KEYBOARD
JB ACC.0,K12
;JMP IF LINE 0 NOT EQUAL 0
MOV DPTR,#KEY
MOV A,#0BH
MOVX @DPTR,A
;MOVE #0B TO KEY
RET

K12: MOVX A,@DPTR
KEYBOARD
CLR ACC.7
;CLEAR BIT 7
SETB ACC.6
;SET BIT 6
MOVX @DPTR,A
;MOVE TO PORT C
MOVX A,@DPTR
;READ KEYBOARD
JB ACC.0,K10
;JMP IF LINE 0 NOT EQUAL 0
MOV DPTR,#KEY
MOV A,#0CH
MOVX @DPTR,A
;MOVE #0C TO KEY
RET

K10: RET

ROW1: MOVX A,@DPTR
KEYBOARD
SETB ACC.6
;SET BIT 6

```

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;SET BIT 6      SETB      ACC.6      ROW3:      MOVX
;SET BIT 7      SETB      ACC.7      A,@DPTR    ;READ
;MOVE TO PORT C MOVX      @DPTR,A      ;SET BIT 6  SETB      ACC.6
;READ KEYBOARD MOVX      A,@DPTR      ;SET BIT 7  SETB      ACC.7
;JMP IF LINE 2 NOT EQUAL 0      ;MOVE TO PORT C MOVX      @DPTR,A
MOV      DPTR,#KEY      ;READ KEYBOARD MOVX      A,@DPTR
MOV      A,#04H      ;JMP IF LINE 3 NOT EQUAL 0
MOVX     @DPTR,A      MOV      DPTR,#KEY
;MOVE #04 TO KEY      MOV      A,#01H
RET      MOVX     @DPTR,A

K5:      MOVX      ;MOVE #01 TO KEY
A,@DPTR  ;READ      RET
KEYBOARD

;CLEAR BIT 6   CLR      ACC.6      K2:      MOVX
;SET BIT 5     SETB      ACC.5      A,@DPTR  ;READ
;MOVE TO PORT C MOVX     @DPTR,A      KEYBOARD
;READ KEYBOARD MOVX     A,@DPTR      ;CLEAR BIT 6  CLR      ACC.6
;JMP IF LINE 2 NOT EQUAL 0      ;SET BIT 5   SETB      ACC.5
MOV      DPTR,#KEY      ;MOVE TO PORT C MOVX     @DPTR,A
MOV      A,#05H      ;READ KEYBOARD MOVX     A,@DPTR
MOVX     @DPTR,A      ;JMP IF LINE 3 NOT EQUAL 0
;MOVE #05 TO KEY      MOV      DPTR,#KEY
RET      MOV      A,#02H
;MOVE #02 TO KEY      MOVX     @DPTR,A
RET

K6:      MOVX      ;MOVE #02 TO KEY
A,@DPTR  ;READ      RET
KEYBOARD

;CLEAR BIT 7   CLR      ACC.7      K3:      MOVX
;SET BIT 6     SETB      ACC.6      A,@DPTR  ;READ
;MOVE TO PORT C MOVX     @DPTR,A      KEYBOARD
;READ KEYBOARD MOVX     A,@DPTR      ;CLEAR BIT 7  CLR      ACC.7
;JMP IF LINE 2 NOT EQUAL 0      ;SET BIT 6  SETB      ACC.6
MOV      A,#06H      ;MOVE TO PORT C MOVX     @DPTR,A
MOV      DPTR,#KEY      ;READ KEYBOARD MOVX     A,@DPTR
MOVX     @DPTR,A      ;JMP IF LINE 3 NOT EQUAL 0
;MOVE #06 TO KEY      MOV      DPTR,#KEY
RET      MOV      A,#03H
;MOVE #03 TO KEY      MOVX     @DPTR,A
RET

K4:      RET      ;MOVE #03 TO KEY
RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

K1: RET

```

;***** LCD DATE/TIME
DISPLAY *****
;***** SHOW
DATE/TIME/PROGRAM *****

LDTD:          MOV
A, #80H

          LCALL LCDWI
;GIVE CORSOR AT 83
          MOV
DPTR, #DAYCLOCK ;READ DATE
FROM IC

          MOVX A, @DPTR
          ANL
A, #00000111B
          CJNE A, #01, M
;COMPARE FOR FIND DATE
          MOV DPTR, #SUN
          LCALL TDISPLAY
          JMP LDTD1
M:
A, #02, TU
          MOV DPTR, #MON
          LCALL TDISPLAY
          JMP LDTD1
TU:
A, #03H, WE
          MOV DPTR, #TUE
          LCALL TDISPLAY
          JMP LDTD1
WE:
A, #04H, TH
          MOV DPTR, #WED
          LCALL TDISPLAY
          JMP LDTD1
TH:
A, #05H, FR
          MOV DPTR, #THU
          LCALL TDISPLAY
          JMP LDTD1
FR:
A, #06H, SA
          MOV DPTR, #FRI
          LCALL TDISPLAY
          JMP LDTD1
SA:
DPTR, #SAT
          MOV
          ;ENDING COMP
          LCALL TDISPLAY
LDTD1:
DPTR, #HOURCLK ;READ HOUR
          MOVX A, @DPTR
          LCALL DISNUM
;CALL DISPLAY NUMERICAL
          MOV A, #3AH
          LCALL LCDWD

          MOV
          DPTR, #MINUTECLK
          MOVX A, @DPTR
          LCALL DISNUM
;CALL DISPLAY NUMERICAL
          MOV A, #3AH
          LCALL LCDWD
          MOV
          DPTR, #SECOND
          MOVX A, @DPTR
          LCALL DISNUM
          MOV
          DPTR, #SPACE
          LCALL TDISPLAY
          RET

SUN:          DB "
MON:          DB "
TUE:          DB "
WED:          DB "
THU:          DB "
FRI:          DB "
SAT:          DB "
SPACE:        DB "

;***** END DATE/TIME
DISPLAY PROGRAM *****

;***** LCD MAIN
STATUS DISPLAY *****

LMSD:          MOV
A, #0C0H
          LCALL LCDWI
          MOV DPTR, #LINK
          MOVX A, @DPTR
          CJNE A, #0FFH, L
;CHECK LINK STATUS
          MOV DPTR, #NL
          LCALL TDISPLAY
          JMP L1
L:
DPTR, #LN
          MOV
          LCALL TDISPLAY
L1:
          MOV
A, #0CAH
          LCALL LCDWI
          MOV DPTR, #AUTSTA

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOVX    A,@DPTR
CJNE   A,#0FFH,AU      TDISPLAY:      CLR    A
MOV     A,#0CAH        MOVC   A,@A+DPTR
LCALL  LCDWI           CJNE   A,#2EH,TD1
MOV     DPTR,#MN      RET
LCALL  TDISPLAY       TD1:          PUSH   DPH
JMP     AUSTA2        PUSH   DPL
AU:      MOV           LCALL  LCDWD
DPTR,#AUT          POP    DPL
                POP    DPH
                INC    DPTR
                JMP    TDISPLAY
AUTSTA1: MOV           DISNUM:      MOV    R2,A
DPTR,#AUTSTA      MOV           SWAP   A
                ORL    A,#00110000B
                LCALL LCDWD
AUSTA2: MOV           A,#00001111B
A,#92H           ORL    A,#00110000B
                LCALL LCDWI
                MOV   DPTR,#PROG
                LCALL TDISPLAY
                MOV   A,R2
DPTR,#MANUAL     A,#00001111B
                MOVX  A,@DPTR
                ORL    A,#00110000B
A,#00110000B     LCALL  LCDWD
                LCALL LCDWD
                MOV   A,#01H
                MOV   DPTR,#TEMP
                MOVX  @DPTR,A
                MOV   A,#0CH
                LCALL LCDWI
                RET
                LCDLDS:      PUSH   DPH
                ;LOAD SUB.
LMSP:      MOV           PUSH   DPL
A,#9AH        LCALL  LCDWI
                ;WRITE ADDRESS
                MOV   DPTR,#MANUAL
                POP    DPL
                MOVX  A,@DPTR
                POP    DPH
                ORL    A,#00110000B
                ;16 CHAR.
                MOV   R2,#16
A,#00110000B     LCDLDS1:      CLR    A
                LCALL LCDWD
                MOVC  A,@A+DPTR
                RET           ;MOVC FOR CODE / MOVX FOR
                DATA
NL:         DB    "
NONLINK    "
LN:         DB    "
LINK       "
MN:         DB    "
MANUAL     "
AUT:       DB    "
AUTO      "
PROG:      DB    "
PROGRAM    "
                PUSH   DPH
                PUSH   DPL
                LCALL  LCDWD
                ;WRITE DATA
                POP    DPL
                POP    DPH
                INC    DPTR
                DJNZ  R2,LCDLDS1
                RET

```

;***** TDISPLAY สำหรับการ LCDWI *****
ROUTINE *****
ไม่ว่ากรณีใดๆ ที่สิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV DPTR,#P3C ;***** INTERRUPT
;DATA TO PORTC SERVICE ROUTINE *****
MOVX @DPTR,A ;***** FOR 10 SEC
MOV DPTR,#P3B TIME *****
;PORTB READ MODIFY WRITE ISRST: PUSH DPH
MOVX A,@DPTR ;INITIAL INTERRUPT
CLR ACC.0 PUSH DPL
;RS=0 PUSH ACC
CLR ACC.1 MOV DPTR,#R6TMP
;R/W=0 CLR ACC.2 MOV A,R6
;ENABLE=0 MOVX @DPTR,A MOVX @DPTR,A
MOVX @DPTR,A DPTR,#R7TMP
SETB ACC.2 MOV A,R7
;ENABLE=1 MOVX @DPTR,A MOVX @DPTR,A
CLR ACC.2 MOV DPTR,#R3TMP
MOVX @DPTR,A MOV A,R3
POP DPL MOVX @DPTR,A
POP DPH MOV
MOV A,#90H DPTR,#R0TMP
;DELAY MOV A,R0
LCDWI1: DEC A MOVX @DPTR,A
JNZ LCDWI1 MOV DPTR,#AUTSTA
RET MOVX A,@DPTR
LCDWD: PUSH DPH CJNE A,#0FFH,ISRAUT
PUSH DPL A,#0FFH,ISRAUT ;
MOV DPTR,#P3C CHECK AUTO STATUS
;DATA TO PORTA JMP ISRLINK
MOVX @DPTR,A ;IF NOT AUTO GOTO CHKLINK
MOV DPTR,#P3B
;PORTB READ MODIFY WRITE ISRAUT: MOV
MOVX A,@DPTR DPTR,#tonf
SETB ACC.0 MOVX A,@DPTR
;RS=1 CJNE
CLR ACC.1 A,#0ffh,ISRAUT1 ;
;R/W=0 CLR ACC.2 CHECK TENTH
;ENABLE=0 JMP ENDISRST
MOVX @DPTR,A ISRAUT1: MOV
SETB ACC.2 DPTR,#UNIT
;ENABLE=1 MOVX A,@DPTR
MOVX @DPTR,A CJNE
CLR ACC.2 A,#00H,ISRAUT2 ;
;ENABLE=0 CHECK UNIT
MOVX @DPTR,A mov a,#0ah
POP DPL movx @dptr,a
POP DPH JMP endisrst
MOV A,#90H
;DELAY ISRAUT2: DEC A
LCDWD1: DEC A MOVX @DPTR,A
JNZ LCDWD1 CJNE
RET A,#00H,ENDISRST

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอญญาตเหนาไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ISRAUT3:      MOV          ISRAUT4:      JMP
A, #0AH
                MOVX      @DPTR, A
                MOV
DPTR, #TENTH   MOV          ISRLINK:      JMP
                MOVX      A, @DPTR
                DEC        A
                LCALL    CHECKHEX
                MOVX      @DPTR, A
                CJNE
A, #00H, ISRAUT4
                MOV          ENDISRST:      LCALL    LDTD
DPTR, #MADD1   MOV          DPTR, #INTSR
                MOVX      A, @DPTR
                INC        A
                MOVX      @DPTR, A
                MOV          ;*****DELAY
                MOV          ROUTINE*****
DPTR, #MADD2   MOVX      A, @DPTR
                INC        A
                MOVX      @DPTR, A
                MOV          DELAY:      MOV
                MOV          R7, #50H
                MOV          DELAY1:      MOV
                MOV          R6, #0
DPTR, #MADD3   DJNZ      R6, $
                MOVX      A, @DPTR
                INC        A
                MOVX      @DPTR, A
                mov       dptr, #tonf
                mov       a, #0ffh
                movx     @dptr, a
                DJNZ      R7, DELAY1
                DJNZ      R2, DELAY
                RET
                END

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ที่สิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลอง

จากการทดลองซึ่งรวมโปรแกรมควบคุมสัญญาณไฟและสัญญาณเชื่อมต่อระหว่างชุดควบคุมเข้าด้วยกันนั้น ผลที่ได้หลังจากที่ได้ทดลองและทำการแก้ไขจุดผิดพลาดจนสามารถใช้งานได้แล้ว ก็คือเครื่องควบคุมสัญญาณไฟจากรถนั้นจะทำงานโดยโปรแกรมทั้งสองนั้นจะทำงานกันได้อย่างสัมพันธ์กัน ซึ่งจากผลที่ได้นี้ก็สรุปได้ว่า เครื่องควบคุมสัญญาณไฟสามารถทำงานได้ตรงตามที่ได้ออกแบบไว้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5 บทสรุปและวิจารณ์

จะเห็นว่า ในการสร้างชุดควบคุมสัญญาณไฟจราจรขึ้นมา นั้น เราได้ทำการทดลองส่วนต่าง ๆ ที่ละส่วน เพื่อทดสอบการทำงานของแต่ละส่วนนั้นดูว่า สามารถทำงานได้อย่างถูกต้องหรือไม่ ซึ่งหลังจากทดลองได้ส่วนหนึ่งแล้ว ก็จะนำส่วนที่ทดลองแล้วมาเพิ่มเติมทดลองส่วนต่อไป ซึ่งในการทดลองแรก ๆ นั้นวงจรที่นำมาใช้ทดลองยังไม่สามารถใช้งานได้ถูกต้องทั้งหมด จึงต้องทดสอบประกอบกันทั้งส่วนของโปรแกรมและวงจร ซึ่งถ้าตรวจสอบโปรแกรมทดลองจนแน่ใจแล้วว่าไม่มีจุดผิดพลาดแน่นอน ดังนั้น ก็ต้องตรวจสอบวงจรที่ใช้ทดลองใหม่หมด จนพบจุดผิดพลาด ซึ่งหลังจากได้ทดลองทุกส่วนของวงจรจนครบและมั่นใจว่าสามารถทำงานได้แน่นอนแล้ว ก็ที่เหลือในส่วนของโปรแกรมสัญญาณไฟ ซึ่งจะทดลองโปรแกรมของสัญญาณไฟที่ละโหมดการทำงานไปเรื่อย ๆ จนถึงในการทดลองสุดท้ายก็คือการนำการทดลองทุกส่วนนั้นมารวมกัน แล้วทำการทดสอบรวมครั้งสุดท้าย ซึ่งจะเห็นได้ว่าความผิดพลาดซึ่งเกิดขึ้นนั้น จะถูกแก้ไขได้ที่จุดในแต่ละการทดลองนั้น ๆ ซึ่งในการแก้ไขนั้น ก็คือทำการตรวจเช็คโปรแกรมใหม่และแก้ไขส่วนที่คิดว่าผิดพลาดให้ถูกต้อง

ปัญหาและแนวทางแก้ไข

ในการทดลองที่ผ่านมาทั้งหมดนี้ ได้ประสบปัญหาต่าง ๆ ดังต่อไปนี้คือ

1. ในตอนแรกที่ทำกรทดลอง อุปกรณ์อินพุท / เอาท์พุท 8255 ไม่ทำการส่งข้อมูลออกมาทางพอร์ทที่เราสั่งงานจากโปรแกรม ทำให้ผลการทดลองไม่ออกมาตามต้องการ หลังจากทดลองแก้ไขโดยการเปลี่ยนโปรแกรมใหม่หลาย ๆ แบบ พบว่า ปัญหาที่เกิดขึ้นเกิดจากการที่ 8255 ใช้เวลาในการรีเซ็ตตัวเองนาน ซึ่งแก้ไขด้วยการเขียนโปรแกรมให้มีการหน่วงเวลาออกไป 8255 ก็ทำงานได้ตามที่ต้องการ
2. ในการทดลองต่าง ๆ ปรากฏว่ามีหลายครั้งที่ทำการทดลองแล้วผลไม่ออกมาตามต้องการ เนื่องจากว่าในการทดลองนั้น สายเชื่อมต่อระหว่างอุปกรณ์มีการหลุดหรือขาด ซึ่งจะต้องใช้เวลาในการตรวจหาที่เสียเป็นเวลานานๆ ทำให้เกิดการล่าช้าในการที่จะทำการทดลองในส่วนอื่นต่อไป
3. ในการทดลองต่าง ๆ นั้น โปรแกรมที่ใช้ในการทดลองแต่ละการทดลองจะมีจุดผิดพลาดอยู่เสมอ ซึ่งจากขั้นตอนในการออกแบบโปรแกรมตามแผนภาพที่ออกแบบก่อนล่วงหน้า นั้น ช่วยให้สามารถเขียนโปรแกรมได้อย่างสะดวกขึ้นและยังลดข้อผิดพลาดในการเขียนโปรแกรมให้น้อยลงด้วย
4. ในการต่อวงจรนั้น ตัว 8031 ไม่ยอมทำงาน โดยไม่สามารถหาสาเหตุได้ ซึ่งจากการตรวจทานดูวงจรก็ไม่สามารถหาได้ว่าผิดพลาดอย่างไร จึงได้ทำการแก้ไขโดยการต่อวงจรขึ้นใหม่อีกหลาย ๆ ครั้ง ซึ่งในการต่อแต่ละครั้งก็มีวงจรเหมือนกัน จนกระทั่งสามารถทำงานได้จึงนำไปใช้งานได้

5. ในขณะที่ออกแบบลายวงจรบนแผ่นวงจรมานั้น เกิดจุดผิดพลาดขึ้นภายใน เนื่องจากลายวงจรเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับอาจารย์ผู้สอนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ซ้ำซ้อนมาก และจำเป็นที่จะต้องทำให้มีขนาดเล็ก จึงเกิดจุดผิดพลาดขึ้น ซึ่งต้องทำการแก้ไขในภายหลัง ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ในการทดลองจอแสดงผลข้อมูล LCD นั้น เมื่อแสดงผลในครั้งแรกพบว่า การแสดงผลมีความเข้มสูงมาก ซึ่งจะลำบากมากในการที่จะอ่าน จึงต้องแก้ไขโดยการเพิ่มตัวต้านทานเข้าไปบริเวณขา Vee เพื่อลดความเข้มของผลที่แสดงลง

แนวทางการพัฒนา

จากชิ้นงานที่ได้ทำมานั้น ยังสามารถแก้ไขให้ดีขึ้นได้ โดยการแก้ไขในส่วนโปรแกรมใหม่ เช่น ให้สามารถเชื่อมต่อกับตัวควบคุมหลาย ๆ ชุดมากขึ้น (เป็นจำนวนสูง ๆ) หรือประยุกต์เพิ่มส่วนต่าง ๆ ขึ้นมาได้ เนื่องจากในตัว 8031 ยังมีพอร์ทที่ยังไม่ได้ใช้งานอีก 1 พอร์ท และ 8255 ยังมีพอร์ทที่ยังไม่ได้ใช้งานอีก 1 พอร์ท ซึ่งสามารถนำไปใช้งานตามแต่ที่จะออกแบบต่อไปได้อีก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิติกรรมประกาศ

ขอกราบขอบพระคุณอาจารย์ที่ปรึกษาคือ รศ. ดร. ถวิล พึ่งมา เป็นอย่างยิ่งที่ได้กรุณาเสียสละเวลาให้คำปรึกษา ดูแลเอาใจใส่ และให้คำแนะนำตลอดจนช่วยแก้ปัญหาต่าง ๆ จนสำเร็จไปได้ด้วยดี ขอกราบขอบพระคุณอาจารย์ภาควิชาวิศวกรรมโทรคมนาคมทุกท่านสำหรับความรู้ที่ถ่ายทอดให้มาและข้อแก้ไขปัญหาต่าง ๆ และท้ายที่สุดขอขอบพระคุณที่ปรีญญาโททุก ๆ คนที่อยู่ภายในห้อง ซึ่งช่วยเหลือและให้กำลังใจตลอดมา ทำให้รายงานฉบับนี้เสร็จสมบูรณ์ลุล่วงไปได้ด้วยดี



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

- [1] Charles E. Sporck, "Interface Databook", National Semiconductor Corporation, California USA., 1986
- [2] Matra-Harris Semiconducteurs., "Architectural Overview of the MHS C51 Family", Le Chesnay, France
- [3] Texas Instruments, "The TTL Databook for Design Engineers", Second Edition, 1981
- [4] Sila Research Co., Ltd., "DMC XXX", พระโขง กทม.
- [5] ผศ. สมยศ จุณณะปิยะ, "การใช้งานไมโครคอนโทรลเลอร์ ตระกูล MCS-51", คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง, กทม., พิมพ์ครั้งที่ 1, 2537
- [6] บริษัท ซีเอ็ดดูเคชั่น จำกัด, "คู่มือไอซี ไมโครโปรเซสเซอร์", พิมพ์ครั้งที่ 1, 2521



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้