สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบควบคุมอุปกรณ์ไฟฟ้าอัตโนมัติ

Electric Appliance Control System

T055790

โดย

นายกิตติชัย ปรีน้อย

นายธนชัย ใจเที่ยง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลขหมู่ 55790

เลขทะเบียน

วัน,เดือน,ปี 25 พ.ค. 2548

b

i

ระบบควบคุมอุปกรณ์ไฟฟ้าอัตโนมัติ

Electric Appliance Control System

โดย

นายกิตติชัย ปรีน้อย รหัส 44015185

นายธนชัย ใจเที่ยง รหัส 44015193

อาจารย์ที่ปรึกษา

อ. ชินภัทร นันทจิวากรชัย

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2546

ภาควิชา อิเล็กทรอนิกส์

คณะ วิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบควบคุมอุปกรณ์ไฟฟ้าอัตโนมัติ

Electric Appliance Control System

ผู้จัดทำ

1. นายกิตติชัย ปรีน้อย
2. นายธนชัย ใจเที่ยง

.....อาจารย์ที่ปรึกษา

อาจารย์ชินภัทร นันทจิวากรชัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ในการจัดทำโครงงาน ระบบควบคุมไฟฟ้าอัตโนมัติ เป็นการนำเอาเทคโนโลยีของไมโครคอนโทรเลอร์ MCS-51 มาสร้างเป็นโครงงาน โดยให้เป็นไปตามแผนงานที่ได้วางไว้ และบรรลุตามวัตถุประสงค์ ครั้งนี้นั้น ต้องใช้ทั้งศาสตร์และศิลป์ ทำให้ผู้จัดทำประสบกับอุปสรรคต่างๆ มากมาย แต่ก็ได้รับคำแนะนำจากหลายๆท่าน เป็นต้นว่า อาจารย์ รุ่นพี่ เพื่อนๆ ทำให้ผู้จัดทำสามารถแก้ปัญหาไปได้เป็นอย่างมาก ถ้าหากว่าขาดซึ่งบุคคลเหล่านี้แล้วผู้จัดทำคงต้องใช้เวลานานกว่าที่จะสามารถแก้ปัญหาเองได้ นอกจากนั้นยังทำให้มีแนวคิด และแนวทางทางในการทำงานต่อไปอีกด้วย

สุดท้ายนี้ ผู้จัดทำขอขอบคุณทุกท่านที่ให้การสนับสนุน รวมทั้งให้คำปรึกษา แนะนำ จนกระทั่งโครงงาน ระบบควบคุมไฟฟ้าอัตโนมัติ สำเร็จลุล่วงไปด้วยดี

.....

(นายกิตติชัย ปรีน้อย)

.....

(นายธนชัย ใจเที่ยง)

ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมอุปกรณ์ไฟฟ้าอัตโนมัติ

Electric Appliance Control System

กิตติชัย ปรีน้อย
ธนชัย ใจเที่ยง
อ. ชินภัทร นันทจิวากรชัย อาจารย์ที่ปรึกษา
ปีการศึกษา 2546

บทคัดย่อ

ระบบควบคุมอุปกรณ์ไฟฟ้าอัตโนมัติ เป็นการนำเอาไมโครคอนโทรลเลอร์ MCS-51 มาประยุกต์ใช้งานในโหมดมัลติโปรเซสเซอร์ผ่านทางพอร์ตอนุกรม ซึ่งระบบประกอบด้วยโมดูลควบคุมอย่างน้อย 2 ชุด เชื่อมต่อกันด้วยสายตีเกลียวคู่ (twisted pair) แบบบัส และใช้มาตรฐานการสื่อสารข้อมูลอนุกรม RS-485 จะทำให้สามารถต่อโมดูลควบคุมในระบบได้สูงสุดถึง 32 ชุด ซึ่งในแต่ละโมดูลประกอบด้วยวงจรขับโหลด 8 พอร์ต สวิตช์ควบคุมแบบเมทริกซ์ และจอแสดงผลแบบ LCD 16 ตัวอักษร 2 บรรทัด

ในการใช้งาน สามารถควบคุมเปิด - ปิดอุปกรณ์ไฟฟ้าที่ต่ออยู่บนระบบจากโมดูลใดก็ได้ โดยระบบจะทำการสั่งงานไปยังโมดูลอื่นด้วยระบบของ Token ซึ่งจะมีข้อดีตรงที่ชุดควบคุมแต่ละจุดจะสามารถติดต่อกันเองได้โดยตรง โดยไม่ต้องการโมดูลมาสเตอร์แม้ว่ามีโมดูลใดโมดูลหนึ่งหรือมากกว่าเสียอยู่ ระบบก็ยังคงสามารถทำงานได้ ระบบนี้ยังสามารถพัฒนาไปใช้กับระบบควบคุมอุปกรณ์ไฟฟ้าอื่นๆ ในโรงงานอุตสาหกรรมขนาดใหญ่ได้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Electric Appliance Control System

Mr.Kittichai Preenoi
Mr.Thanachai Jaitieng
Mr.Chinnapat Nantajiwakornchai
(Advisor)

1st Semester, Educational Year 2003

Abstract

The Electric Appliance Control System in this project, we use microcontroller MCS - 51 to apply in the multiprocessor mode which connect through the serial port. There are at least 2 module controllers which connect through the standard switch pair line. we will use the RS485 port to communicate so that we can connect the module controllers up to 32 sets. And each module controller contain 8 ports of the driving load circuit by using matrix switch. We use the LCD with 2 lines 16 letters monitor.

In practicing we can control any electrical switches in which connect to any module in the system. the system will transfer the data to the module by using Token system. The advantage of this system is that every control points or module controller can communicate directly to each other without master module. Even though there will be any module in the system does not work, the system still running. According to our experience, we think that we can apply this system to control the electrical instrument in the industrial.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
Abstract	III
สารบัญ	IV
สารบัญรูปภาพ	V
สารบัญตาราง	VI
บทที่ 1 บทนำ	1
บทที่ 2 ไมโครคอนโทรลเลอร์ MCS-51	4
2.1 โครงสร้างของ MCS-51	4
2.1.1 การจัดการขาต่างๆ ของ MCS-51	7
2.1.2 โครงสร้างของพอร์ทอินพุตและเอาท์พุต	9
2.1.3 โครงสร้างของความจำ	11
2.1.4 การรีเซต MCS-51	12
2.1.5 การเชื่อมต่อหลอด LED	15
2.1.6 การเชื่อต่อคีย์สวิตซ์กับ MCS-51	16
2.2 พอร์ทอนุกรมของไมโครคอนโทรลเลอร์ MCS-51	17
2.2.1 การสื่อสารข้อมูลแบบอะซิงโครนัส	17
2.2.2 รีจิสเตอร์ที่เกี่ยวข้องกับการทำงานของพอร์ทอนุกรมภายใน MCS-51	19
2.2.3 โหมดการทำงานของพอร์ทอนุกรม	20
2.2.4 อัตราบอดของพอร์ทอนุกรม	24
บทที่ 3 มาตรฐานการสื่อสารข้อมูล RS – 485	28
3.1 หลักการสื่อสารข้อมูลแบบ RS-485	29
3.2 จำนวนอุปกรณ์ต่อพ่วงในระบบ	31
3.3 การต่อความต้านทานเทอมิเนต	32
3.4 คุณสมบัติของสายส่งสัญญาณ	33
บทที่ 4 การสร้างและการใช้งาน	34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1 หลักการทำงานของระบบควบคุมไฟฟ้าอัตโนมัติ 34
4.2 การสร้าง 37
4.3 การใช้งาน 39
บทที่ 5 บทสรุป 48

สารบัญรูปภาพ

หน้า

รูปที่ 1.1 การเชื่อมระหว่าง ชุด control สองชุดโดย RS-485 2
รูปที่ 1.2 การเชื่อมชุด control หลายๆ ชุดโดยใช้ RS-485 (ระบบ Token) 3
รูปที่ 2.1 แสดงโครงสร้างภายในของ MCS – 51 5
รูปที่ 2.2 แสดงขาต่างๆ ของ 8051 7
รูปที่ 2.3โครงสร้างพอร์ททั้ง 4 ของ MCS – 51 9
รูปที่ 2.4 การต่อพอร์ทเข้ากับระบบบัสภายในของ MCS – 51 10
รูปที่ 2.5 การจัดหน่วยความจำของ MCS – 51 11
รูปที่ 2.6 ตำแหน่งของหน่วยความจำทั้งแบบไบต์และแบบบิต 12
รูปที่ 2.7 การสร้างสัญญาณรีเซต 13
รูปที่ 2.8 การรีเซต MCS – 51 13
รูปที่ 2.9 แสดง LED แต่ละแบบ และการต่อเข้ากับพอร์ท 15
รูปที่ 2.10 วงจรเชื่อมต่อคีย์แพดเข้ากับไมโครคอนโทรลเลอร์ MCS – 51 16
รูปที่ 2.11 รูบแบบข้อมูลอนุกรมแบบอะซิงโครนัส 18
รูปที่ 2.12 ไดอะแกรมการทำงานในโหมด 0 ของพอร์ทอนุกรม ภายในไมโครคอนโทรลเลอร์ MCS – 51 23
รูปที่ 2.13 ไดอะแกรมการทำงานในโหมด 1 ของพอร์ทอนุกรม ภายในไมโครคอนโทรลเลอร์ MCS – 51 25
รูปที่ 2.14 ไดอะแกรมการทำงานในโหมด 2 ของพอร์ทอนุกรม ภายในไมโครคอนโทรลเลอร์ MCS – 51 26
รูปที่ 2.15 ไดอะแกรมการทำงานในโหมด 3 ของพอร์ทอนุกรม ภายในไมโครคอนโทรลเลอร์ MCS – 51 27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.1 ความสัมพันธ์ระหว่างอัตราการส่งข้อมูลกับระยะทาง 29
รูปที่ 3.2 สัญญาณเอาท์พุตของไดรฟ์เวอร์ 30
รูปที่ 3.3 การรับสัญญาณอินพุตแบบดิฟเฟอเรนเชียล 31
รูปที่ 3.4 การต่อตัวต้านทานเทอมิเนตในระบบ 32
รูปที่ 3.5 การต่อความต้านทานเทอมิเนต 32
รูปที่ 3.6 อัตราการสูญเสียแรงดัน 6 dB ที่ระยะต่างๆ 33
รูปที่ 4.1 แบบจำลองระบบการทำงานของระบบ Token 34
รูปที่ 4.2 ระบบการทำงานติดต่อกันระหว่าง 2 เครื่อง 35
รูปที่ 4.3 การสื่อสารในระบบของ TOKEN 36
รูปที่ 4.4 วงจรการทำงานของระบบ 36
รูปที่ 4.5 วงจรและอุปกรณ์ของระบบ 41
รูปที่ 4.6 ภาพแสดงการต่อโหลด 8 เอาท์พุท 41
รูปที่ 4.7 ภาพแสดงการเชื่อมต่อโมดูลด้วยสาย twisted pair 41

สารบัญตาราง

หน้า

ตารางที่ 2.1 แสดงไมโครโปรเซสเซอร์ตระกูล MCS - 51 เบอร์ต่างๆ 5
ตารางที่ 2.2 แสดงบิตและหน้าที่ต่างๆ ของพอร์ท 3 8
ตารางที่ 2.3 แสดงค่าต่างๆ ที่เกิดหลังการรีเซต 14
ตารางที่ 3.1 ข้อกำหนดของ RS-485 และ RS-422 27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

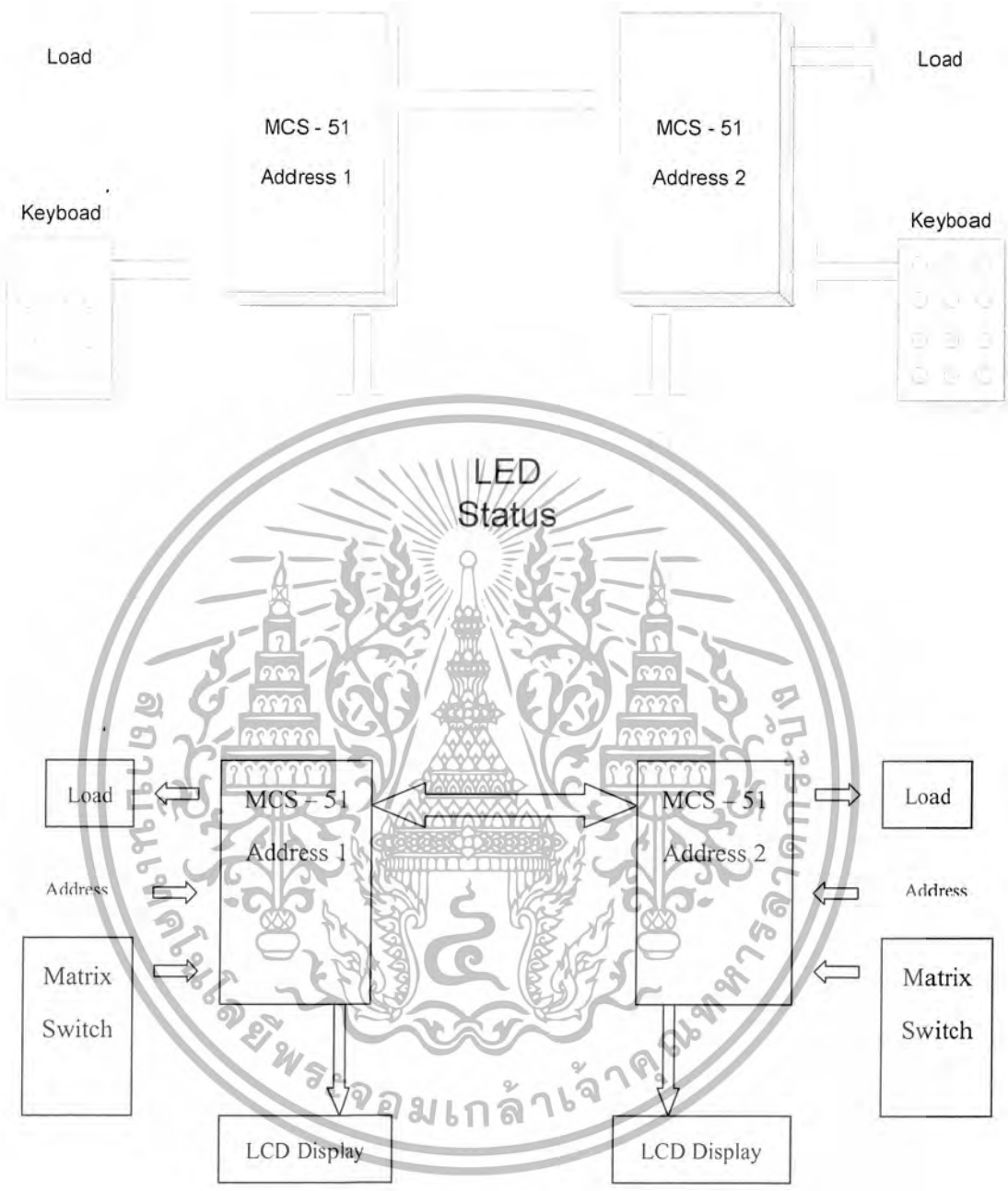
บทที่ 1
บทนำ

ระบบควบคุมอุปกรณ์ไฟฟ้าอัตโนมัติได้นำเอาเทคโนโลยีของไมโครคอนโทรลเลอร์ MCS – 51 มาประยุกต์ใช้งาน โดยมีวัตถุประสงค์ที่จะนำเอาความรู้ ความเข้าใจ ในเรื่องไมโครคอนโทรลเลอร์มาสร้างเป็นชิ้นงานที่สามารถใช้งานได้จริงในโรงงานอุตสาหกรรม หรือ ในอาคารที่มีการใช้แสงสว่างจำนวนมากทำให้ยุ่งยากในการเปิด-ปิดสวิตซ์ควบคุมในแต่ละครั้ง ซึ่งบางทีอาจมีสิ่งของวางกีดขวางอยู่บ้าง

การทำงานของระบบควบคุมอุปกรณ์ไฟฟ้าอัตโนมัติคือ แบ่งพื้นที่ควบคุมเป็นส่วนย่อยๆ ตามความจำเป็นแล้วทำการติดตั้งชุด control ไว้ในทุกๆ ส่วน โดยเชื่อมต่อชุด control แต่ละชุดเข้าด้วยกันด้วยสาย control ชุด control แต่ละชุดจะทำการติดต่อสื่อสารกัน ผ่านสาย control โดยพอร์ตอนุกรม ก็คือ ขา TxD และขา RxD ของไมโครคอนโทรลเลอร์ MCS-51 ซึ่งเป็นแบบ full-duplex จากนั้นแปลงเป็น half-duplex โดยใช้ RS-485 เราสามารถควบคุมหลอดไฟฟ้าทุกดวงที่ขึ้นอยู่กับระบบ โดยใช้ชุด control ชุดใดก็ได้ผ่านคีย์แพด และยังสามารถตรวจสอบสถานะของหลอดไฟฟ้า ต่างๆ จากจอแสดงผล เราสามารถที่จะเลือกควบคุมการทำงานของอุปกรณ์ไฟฟ้าได้จากคีย์สวิตซ์ตัวนี้

การส่งผ่านข้อมูลจะใช้การส่งข้อมูลที่พอร์ตอนุกรมแบบอะซิงโครนัส ให้ข้อมูลรับเข้าและส่งออกผ่านรีจิสเตอร์ที่ชื่อ SBUF ใช้ระบบการสั่งงานของชุดควบคุมแต่ละจุดด้วยระบบ TOKEN ซึ่งมีข้อดีตรงที่ชุดควบคุมแต่ละจุดจะสามารถติดต่อกันเองได้โดยตรง และอย่างที่กล่าวมาแล้วข้างต้นว่าเครื่องควบคุมระบบอุปกรณ์ไฟฟ้าอัตโนมัตินี้สามารถควบคุมหลอดไฟฟ้าได้ทุกดวง จากชุดcontrol ใดๆ ก็ได้ จึงทำให้การควบคุมได้ง่าย การควบคุม เปิด-ปิด หลอดไฟเท่าที่จำเป็นต่อการใช้งาน เพิ่มความสะดวก รวดเร็ว ประหยัดเวลา และกำลังงาน รวมทั้งบุคลากรที่ดูแลในส่วนนี้ด้วย ในการแก้ไขปรับปรุงก็สามารถทำได้ง่ายโดยแก้ไขที่โปรแกรมควบคุม และยังสามารถเขียนโปรแกรม option อื่นๆ ได้ตามต้องการ เช่น ตั้งเวลา เปิด-ปิด อัตโนมัติ เป็นต้น

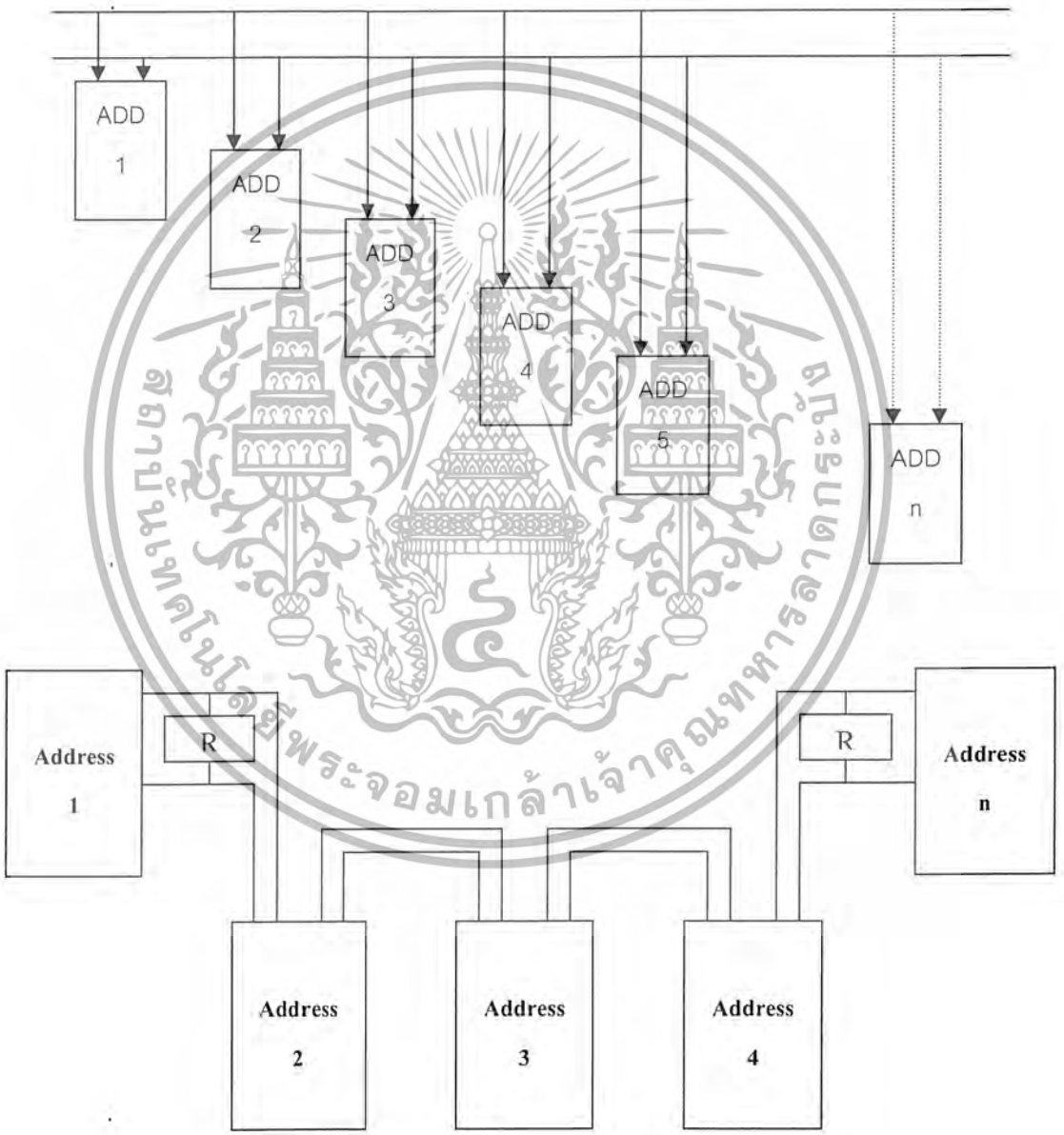
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.1 การเชื่อมระหว่าง ชุด control สองชุดโดย RS-485

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ฉบับนี้ ผู้ออกแบบได้ออกแบบให้ผู้ใช้สามารถควบคุมอุปกรณ์ไฟฟ้าในส่วนของที่อยู่อื่นๆได้ Address ละ 8 ตัว โดยสร้างเป็นต้นแบบขึ้นมาด้วยกันทั้งสิ้น 3 วงจร หรือ 3 Address นั่นเอง ซึ่งสามารถควบคุมได้ทุก Address รวมถึง Address ตัวเองด้วย และได้สร้างวงจรชุดขับเอาท์พุท 220 V เป็นต้นแบบไว้ 1 ชิ้นด้วย



รูปที่ 1.2 การเชื่อมชุด control หลายๆ ชุดโดยใช้ RS-485 (ระบบ Token)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ไมโครคอนโทรลเลอร์ MCS – 51

2.1 โครงสร้างของ MCS – 51

ไมโครคอนโทรลเลอร์ตระกูล MCS – 51 มีด้วยกันหลายเบอร์ขึ้นกับโครงสร้างภายในของมัน บางเบอร์จะมีหน่วยความจำภายในแบบ ROM บางเบอร์เป็นแบบ EPROM บางเบอร์มี RAM ภายใน 128 ไบต์ บางเบอร์มี 256 ไบต์ เป็นต้น ซึ่งรายละเอียดจะศึกษาได้จากคู่มือของมันโดยตรง และลักษณะของขาต่างๆ จะเหมือนกัน คุณสมบัติที่สำคัญของ MCS – 51 มีดังนี้

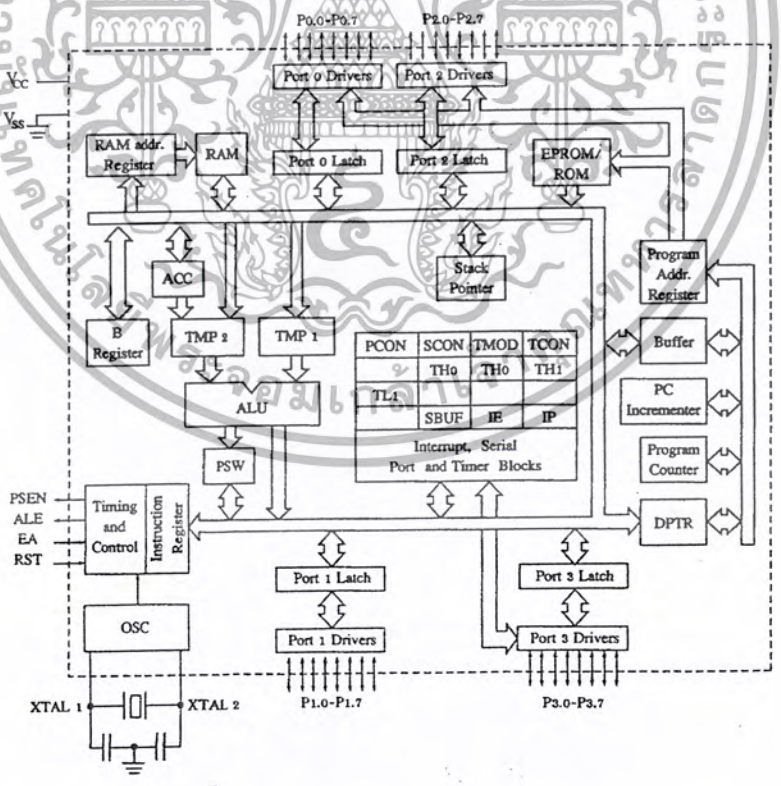
- มีหน่วยความจำ ROM 4K bytes
- มีหน่วยความจำ RAM 128 bytes
- มีพอร์ท I/O ขนาด 8 บิต 4 พอร์ท
- มี Timer 16 บิต 2 ตัว
- สามารถอินเทอร์รัพท์ได้ 5 แหล่ง
- มีวงจรออสซิลเลเตอร์และวงจรนาฬิกาชิพ
- มีพอร์ทอนุกรมที่สามารถรับส่งข้อมูลแบบ Full Duplex
- อ้างหน่วยความจำโปรแกรมภายนอกได้ 64 K
- อ้างหน่วยความจำข้อมูลภายนอกได้ 64 K
- สามารถประมวลผลทีละบิตได้
- สามารถอ้างหน่วยความจำแบบบิตได้ 210 ตำแหน่ง
- หนึ่งรอบคำสั่งกินเวลาประมาณ 1 ไมโครวินาที เมื่อทำงานด้วย Clock 12 MHz

ตัวอย่างไมโครคอนโทรลเลอร์ตระกูล MCS – 51 และลักษณะต่างๆ สามารถแสดงได้ในตารางที่ 2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เบอร์	หน่วยความจำโปรแกรมบนชิพ	หน่วยความจำข้อมูลบนชิพ	TIMERS
8051	4K ROM	128 bytes	2
8031	-	128 bytes	2
8751	4K EPROM	128 bytes	2
8052	8K ROM	256 bytes	3
8032	-	256 bytes	3
8752	8K EPROM	256 bytes	3
AT89C51	4K FLASH	128 bytes	2
AT89C52	8K FLASH	256 bytes	3
AT89C2051	2K FLASH	128 bytes	2

ตารางที่ 2.1 แสดงไมโครโปรเซสเซอร์ตระกูล MCS - 51 เบอร์ต่างๆ



รูปที่ 2.1 แสดงโครงสร้างภายในของ MCS – 51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งาน MCS – 51 จะต้องมีหน่วยความจำ ROM สำหรับเก็บโปรแกรมควบคุมการทำงานโดยหน่วยความจำ ROM นี้แบ่งได้เป็น ROM,PROM,EPROM และ EEPROM/Flash โดย Mask นำมาโปรแกรมเองได้ แต่จะโปรแกรมได้ครั้งเดียว สำหรับ EPROM/Flash ผู้ใช้สามารถโปรแกรมเองได้และลบได้หลายครั้งโดยใช้ไฟฟ้า ตัว MCS – 51 บางรุ่นตัวหน่วยความจำ ROM ที่เก็บโปรแกรมนี้จะต้องต่อเพิ่มขึ้นภายนอกชิพ แต่บางรุ่นจะมีหน่วยความจำ ROM อยู่ภายในตัวมันเอง เช่นเบอร์ 8051,8052 จะมีหน่วยความจำประเภท Mask ROM โดยโปรแกรมที่อยู่ภายในชิพจะต้องถูกโปรแกรมมาจากโรงงานที่ผลิต MCS – 51 ถ้าหากผู้ใช้งานต้องการโปรแกรมเองอาจใช้เบอร์ 8751 ซึ่งมี ROM แบบ EPROM อยู่ในชิพ ทำให้สามารถโปรแกรมและลบแก้ไขโปรแกรมได้ง่าย แต่ถ้าเป็นเบอร์ 8951 จะมี ROM แบบ EEPROM/Flash ซึ่งเป็นที่นิยมใช้กันมาก โดยการโปรแกรมและแก้ไขโปรแกรมจะทำได้โดยไม่ต้องถอดชิพออกจากแผงวงจร การใช้งาน MCS – 51 ที่มีหน่วยความจำโปรแกรมภายในตัวชิพนี้ทำให้สามารถป้องกันการคัดลอกโปรแกรมได้เป็นอย่างดี ปัจจุบันมีไมโครคอนโทรลเลอร์ตระกูล MCS – 51 ของบริษัท Atmel ที่สามารถโปรแกรมข้อมูลลงไปในตัวชิพได้โดยไม่ต้องถอดชิพออกจากแผงวงจร เรียกว่าเป็นการโปรแกรมในระบบ (In – system Programming) เช่นเบอร์ AT89XX ทำให้สามารถแก้ไขโปรแกรมได้ง่ายขึ้น

จากตารางที่ 2.1 เราอาจมองง่ายๆ ว่า ถ้ามีไมโครคอนโทรลเลอร์ตระกูล MCS – 51 เบอร์ที่ลงท้ายด้วย 2 จะมีหน่วยความจำบนข้อมูลชิพ 256 ไบท์ เช่นเบอร์ 8052 8032 แต่ถ้าลงท้ายด้วย 1 จะมีหน่วยความจำข้อมูลจำนวน 128 ไบต์ หรือถ้ามองเป็น 8 * 51 เป็น 9 หมายความว่ามีหน่วยความจำข้อมูลแบบ EEPROM/Flash แต่ถ้ามีตัว C อยู่ด้วยหมายความว่าเทคโนโลยีในการผลิตจะเป็นแบบ CMOS เช่นเบอร์ 87C51, 89C51 เป็นต้น

เมื่อไมโครคอนโทรลเลอร์ทำงานจะต้องมีหน่วยความจำสำหรับเก็บข้อมูลต่างๆ ที่สามารถเปลี่ยนแปลงได้ ซึ่งอาจเป็นหน่วยความจำของตัวแปรที่ใช้ในการประมวลผล หรือหน่วยความจำเก็บข้อมูลชั่วคราวที่ได้จากการประมวลผล โดยจะใช้หน่วยความจำประเภท RAM (Random – Access Memory) ซึ่งไมโครคอนโทรลเลอร์จะมีหน่วยความจำ RAM ภายในตัวมันเอง และสามารถต่อ RAM ภายนอกเพิ่มเติมได้อีกด้วย สำหรับงานเล็กๆ อาจใช้เพียงหน่วยความจำ RAM ภายในก็เพียงพอ ใน MCS – 51 จะมีหน่วยความจำ RAM สำหรับเก็บข้อมูลภายในอยู่ 4 ประเภทคือ เก็บแบบบิต เก็บเป็นรีจิสเตอร์ เก็บเป็นข้อมูลใช้งานทั่วไป และหน่วยความจำสแตก (Stack Memory)

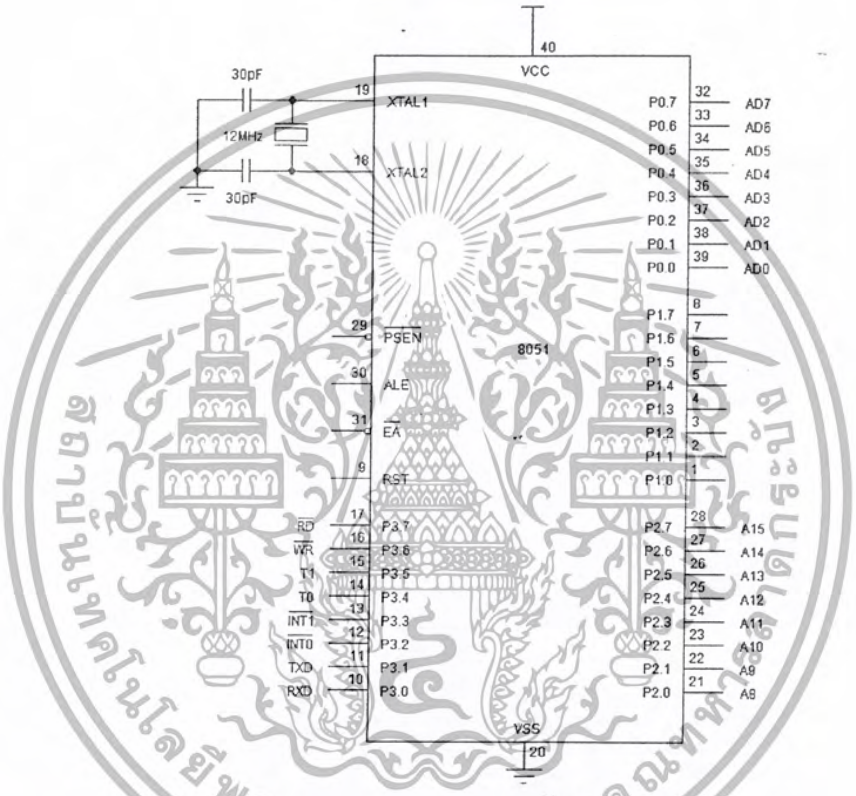
หน่วยความจำสแตกสามารถใช้เก็บข้อมูลทั่วไปได้ แต่การเขียนและอ่านข้อมูลจะเป็นการกระทำกับข้อมูลซ้อนๆ กัน โดยข้อมูลที่ถูกเขียนลงไปที่หลังจะต้องถูกอ่านออกมาก่อน บางครั้งจะเรียกหน่วยความจำนี้ว่า Last In/First Out (LIFO) Memory

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1 การจัดการขาต่างๆ ของ MCS – 51

ไอซีไมโครคอนโทรลเลอร์ 8051 โครงสร้าง IC เป็นแบบ DIP มีขาทั้งหมด 40 ขา โดยขาต่างๆ จะใช้เป็นขาพอร์ทอินพุต เอาต์พุต ขาสัญญาณควบคุม ขาตำแหน่งหน่วยความจำ และขาข้อมูลดังรูปที่ 2.2

ความหมายของขาต่างๆ มีดังนี้



รูปที่ 2.2 แสดงขาต่างๆ ของ 8051

1. Port 0

พอร์ท 0 ได้แก่ขาที่ 32 – 39 ของ MCS – 51 สามารถใช้เป็นอินพุตได้นอกจากนี้ในการติดต่อกับหน่วยความจำภายนอกยังใช้เป็นขา Address Bus และ Data Buss อีกด้วย

2. Port 1

พอร์ท 1 ได้แก่ขาที่ 1 – 8 เป็นพอร์ท 8 บิต สามารถอ้างทีละบิตได้ คือ P1.0, P1.1,...ete

3. Port 2

พอร์ท 2 ได้แก่ขาที่ 21 – 28 จะใช้งาน 2 หน้าที่คือใช้เป็นพอร์ท 8 บิตกับใช้เป็นขาแอดแดรส 8 บิตในการอ้างหน่วยความจำภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. Port 3

พอร์ท 3 ได้แก่ขาที่ 10 – 17 จะใช้งานสองหน้าที่คือ เป็นพอร์ทอินพุตและเอาต์พุต และใช้เป็นขาควบคุมต่างๆ ดังตารางที่ 2.2

บิต	ชื่อ	หน้าที่พิเศษ
P3.0	RXD	ใช้รับข้อมูลทางพอร์ทอนุกรม
P3.1	TXD	ใช้ส่งข้อมูลทางพอร์ทอนุกรม
P3.2	INT0	อินเตอร์รัพท์ภายนอกหมายเลข 0
P3.3	INT1	อินเตอร์รัพท์ภายนอกหมายเลข 1
P3.4	T0	ตัวจับเวลา / ตัวนับตัวที่ 0
P3.5	T1	ตัวจับเวลา / ตัวนับตัวที่ 1
P3.6	WR	สัญญาณเขียนข้อมูลหน่วยความจำภายนอก
P3.7	RD	สัญญาณอ่านข้อมูลหน่วยความจำภายนอก

ตารางที่ 2.2 แสดงบิตและหน้าที่ต่างๆ ของพอร์ท 3

5. PSEN (Program Store Enable)

ขานี้เป็นขาที่ส่งสัญญาณออกคือขา 29 ขานี้จะแอคทีฟเมื่อ MCS – 51 ต้องการอ่าน Code โปรแกรมภายนอก โดยปกติแล้วถ้าหน่วยความจำภายนอกเป็น EPROM ขานี้ จะต่อกับขา Output Enable ของ EPROM

6. ALE (Address Latch Enable)

เนื่องจากพอร์ท 0 สามารถใช้เป็นขาอ้างตำแหน่ง และขาข้อมูล MCS – 51 จะมีขา ALE ได้แก่ขา 30 ขานี้จะใช้ Multiplex สัญญาณ Address Bus ของ Port 0 ในการใช้งานระบบ MCS – 51 นั้นจะต้องมีอุปกรณ์มาต่อกับ Port 0 ที่ทำหน้าที่ Latch สัญญาณ Address Bus เมื่อ MCS – 51 ต้องการติดต่อกับหน่วยความจำภายนอก MCS – 51 จะส่งสัญญาณ Address Bus ออกมาก่อนทาง Port 0 จากนั้นจะส่งสัญญาณ ALE มา Latch อุปกรณ์ภายนอก ให้เก็บค่า Address Bus ของ Port 0 ไว้เพื่อใช้ Port 0 เป็น Data Bus ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. EA (External Access)

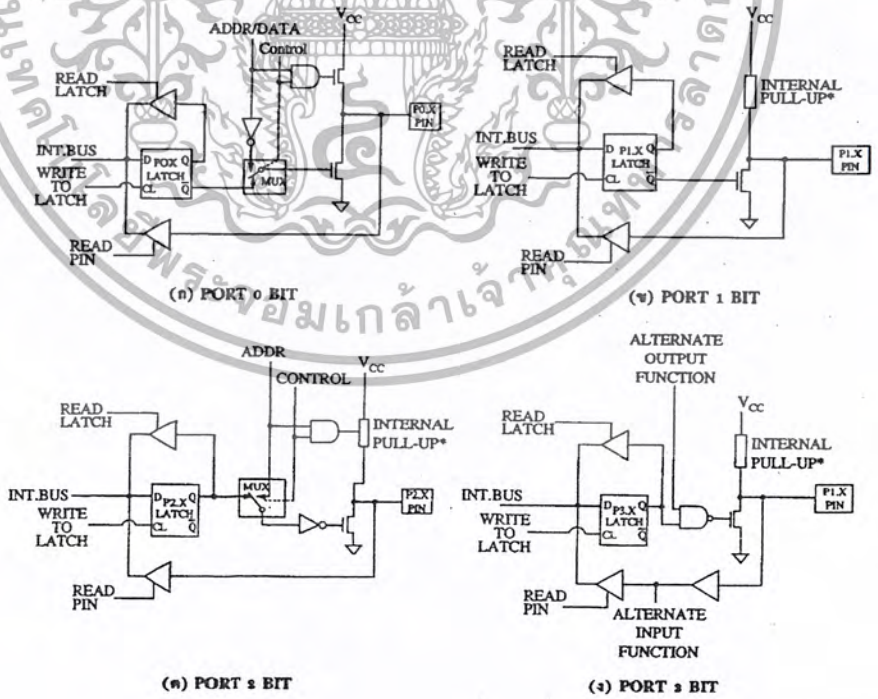
ขานี้ได้แก่ขาที่ 31 ถ้าขานี้เป็นลอจิก “1” จะใช้กับเบอร์ 8051/8052 เพื่อบอกว่าให้อ่านโปรแกรม จากหน่วยความจำโปรแกรมภายใน แต่ถ้าเป็นลอจิก “0” จะบอกว่าให้ MCS – 51 ทำโปรแกรมโดยอ่านจากหน่วยความจำโปรแกรมภายนอก (ถ้าขา EA เป็น “0” ขา PSEN จะแอคทีฟ) ถ้าหากเป็นเบอร์ 8031/8052 ซึ่งมีหน่วยความจำโปรแกรมภายในและให้ขานี้เป็น “0” ซึ่งจะ Disabled ROM ภายในและจะอ่านโปรแกรมจาก EPROM ภายนอกแทน

8. RST (Reset)

ขา RST ได้แก่ขา 9 จะใช้ในการรีเซต MCS – 51 โดยจะให้ขานี้เป็นลอจิก “1” อย่างน้อย 2 Machine Cycles จึงจะรีเซตระบบได้

2.1.2 โครงสร้างของพอร์ทอินพุตเอาต์พุต (I/O Port Structure)

ขาของพอร์ทจะแสดงโครงสร้างภายในได้ ดังรูปที่ 2.3 โดยจะมีโครงสร้างเป็นField – effect Transistor ต่ออยู่กับขาภายนอก และมีความต้านทานต่อ Pull – up อยู่สำหรับพอร์ท 1, 2, 3 แต่ถ้าเป็นพอร์ท 0 จะไม่มีตัวต้านทาน Pull – up ภายใน เพราะว่าต้องใช้เป็นขา Address Bus และ Data Bus

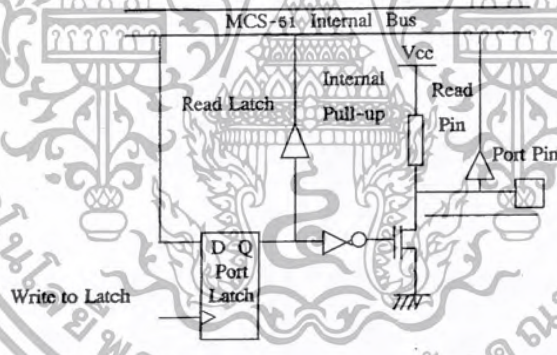


รูปที่ 2.3 โครงสร้างพอร์ททั้ง 4 ของ MCS – 51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.3 จะเห็นว่าพอร์ท 0 จะไม่มีตัวต้านทานต่อพูลอัพอยู่ภายในชิพ ดังนั้นถ้าหากต้องการใช้พอร์ทนี้เป็นพอร์ทอินพุตจะต้องต่อตัวต้านทานพูลอัพภายนอกให้กับพอร์ทนี้ทุกบิตด้วย และจากโครงสร้างของพอร์ท 0 จะเห็นว่ามีวงจรมัลติเพล็กซ์อยู่ภายในเพื่อใช้ในการเลือกว่าจะใช้เป็นพอร์ทหรือใช้ในการติดต่อสื่อสารกับหน่วยความจำภายนอก เช่นเดียวกับพอร์ท 2 แต่พอร์ท 2 จะมีตัวต้านทานพูลอัพอยู่ภายในชิพ ถ้าหากจะใช้งานเป็นพอร์ทอินพุตสามารถใช้ได้ทันที สำหรับพอร์ท 3 จะมีโครงสร้างคล้ายกับพอร์ท 1 แต่จะมีสัญญาณอินพุตเอาต์พุตพิเศษเพิ่มเข้ามา เนื่องจากพอร์ทนี้สามารถใช้งานเป็นหน้าที่พิเศษแต่ละบิตได้

พอร์ทนี้สามารถใช้เป็นอินพุตหรือเอาต์พุตกับอุปกรณ์ภายนอกได้ ในการอ่านข้อมูลจากพอร์ทจะอ่านได้สองแบบคือ Read Latch และ Read Pin โดย Read Latch หมายถึงการอ่านข้อมูลที่ถูก Latch เอาไว้ เข้าสู่บัสภายในของ MCS – 51 เช่นการทำคำสั่ง CPL P1.5 แต่ถ้าเป็นการ Read pin จะเป็นการใช้พอร์ทอินพุต โดยจะอ่านค่าจากขาของไอซีเข้าสู่บัสภายในโดยการอ่านแบบ Read Latch และ Read Pin จะมีสัญญาณมาควบคุมที่บัพเฟอร์ดังรูปที่ 2.4



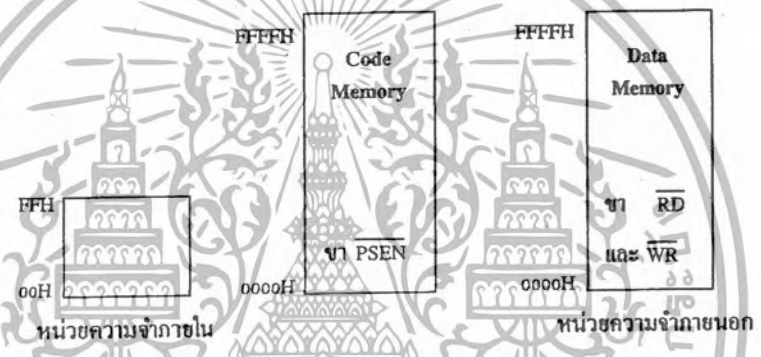
รูปที่ 2.4 การต่อพอร์ทเข้ากับระบบบัสภายในของ MCS – 51

จากรูปที่ 2.4 จะเห็นว่าถ้าหากจะใช้งานพอร์ทเป็นพอร์ทเอาต์พุตสามารถส่งข้อมูลออกไปได้ทันที เนื่องจากพอร์ทถูกออกแบบให้เป็นเอาต์พุตอยู่แล้ว แต่การนำไปต่อกับอุปกรณ์ภายนอกจะจ่ายแรงดัน (Source current) ได้ประมาณ 10 mA การใช้งานพอร์ทของ MCS – 51 เป็นพอร์ทเอาต์พุตนี้ควรจะมีการต่อบัฟเฟอร์ทางเอาต์พุตด้วย เพื่อให้ขับกระแสได้ดีขึ้นและป้องกันตัวไมโครคอนโทรลเลอร์เสียหาย ส่วนการใช้งานเป็นพอร์ทอินพุตจะต้องส่งค่าลอจิก “1” ไปยังบิตของพอร์ตก่อน เพื่อให้เฟสที่ต่ออยู่หยุดนำกระแส ซึ่งจะทำให้ขาที่อยู่กับอุปกรณ์ภายนอกมีลอจิก “1” โดยค่าลอจิก “1” นี้จะทำให้รับข้อมูลจากอุปกรณ์ภายนอกได้ง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 โครงสร้างหน่วยความจำ (I/O Port Structure)

หน่วยความจำสำหรับ MCS – 51 จะมี 2 ชนิดคือ หน่วยความจำที่ใช้เก็บโปรแกรม (ROM) กับหน่วยความจำที่ใช้เก็บข้อมูลในการประมวลผล (RAM) MCS – 51 บางเบอร์เช่น 8051, 8052 จะมีหน่วยความจำโปรแกรมภายในชิพ และ MCS – 51 ทุกเบอร์สามารถอ้างหน่วยความจำโปรแกรมภายนอกได้มากที่สุด 64 K และอ้างหน่วยความจำข้อมูลภายนอกได้มากที่สุด 64 K สำหรับหน่วยความจำ RAM ภายใน จะประกอบไปด้วยพื้นที่ใช้งานทั่วไป รีจิสเตอร์แบงค์ พื้นที่ใช้งานระดับบิต และรีจิสเตอร์ฟังก์ชันพิเศษ เราอาจเขียนไดอะแกรมของหน่วยความจำของ 8031 ได้ดังรูปที่ 2.5 โดยในรูปจะบอกด้วยว่าขาใดจะแอคทีฟ



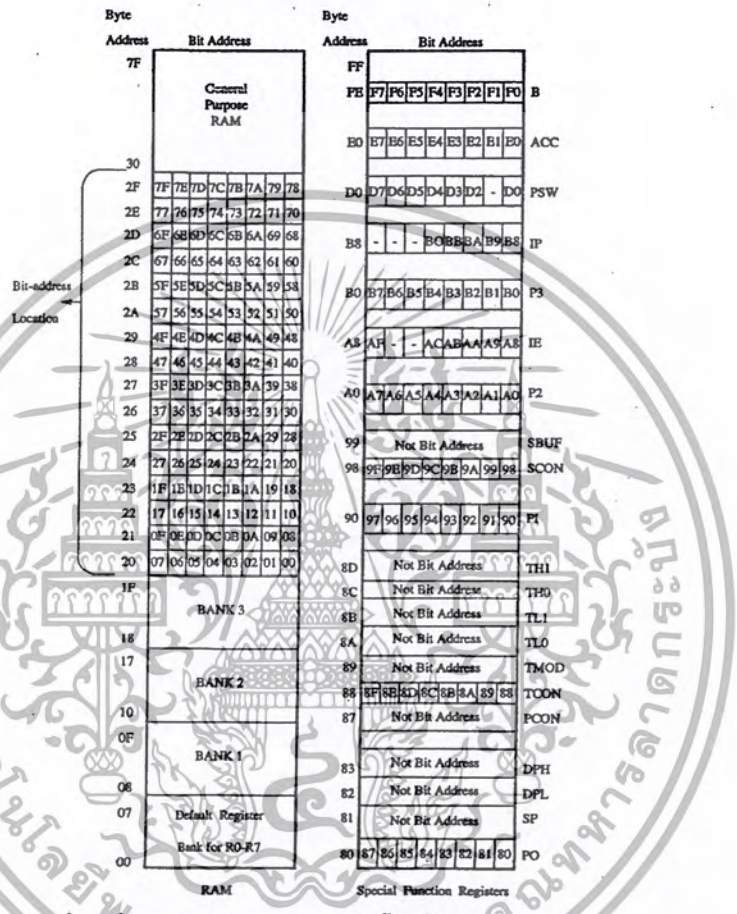
รูปที่ 2.5 การจัดหน่วยความจำของ MCS – 51

ใน 8031 จะมีหน่วยความจำภายในตั้งแต่ตำแหน่ง 00H จึง FFH และสามารถอ้างหน่วยความจำโปรแกรมภายนอกได้ 64K ตำแหน่ง ถ้าอ่านข้อมูลจากหน่วยความจำโปรแกรมขา PSEN จะแอคทีฟ นอกจากนี้ 8031 สามารถอ้างหน่วยความจำข้อมูลภายนอกได้ 64 K ตำแหน่ง โดยการติดต่อกับหน่วยความจำนี้ขา RD และ WR จะแอคทีฟ สำหรับหน่วยความจำข้อมูลภายในนั้นจะแบ่งออกได้ดังนี้

1. ชุดรีจิสเตอร์ 4 ชุดแต่ละชุดเรียกว่ารีจิสเตอร์แบงค์ ที่ตำแหน่ง 00H ถึง 1FH โดยแต่ละชุดประกอบด้วยรีจิสเตอร์ Ro ถึง R7
2. หน่วยความจำที่สามารถเข้าถึงข้อมูลระดับบิตได้ ตำแหน่ง 20H ถึง 2FH
3. หน่วยความจำใช้งานทั่วไป ตำแหน่ง 30H ถึง 7FH
4. รีจิสเตอร์ฟังก์ชั่นพิเศษ ตำแหน่ง 80H ถึง FFH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนผังการจัดหน่วยความจำข้อมูลภายในแสดงได้ดังรูปที่ 2.6 จากแผนผังจะเห็นว่าการอ้างตำแหน่งหน่วยความจำภายในจะอ้างได้สองแบบ คือ การอ้างไปที่ตำแหน่งของไบต์ (เขียนหมายเลขตำแหน่งด้านใน) โดยตำแหน่งของหน่วยความจำที่อ้างเป็นแบบบิตได้จะมีตำแหน่งบิตที่แน่นอน

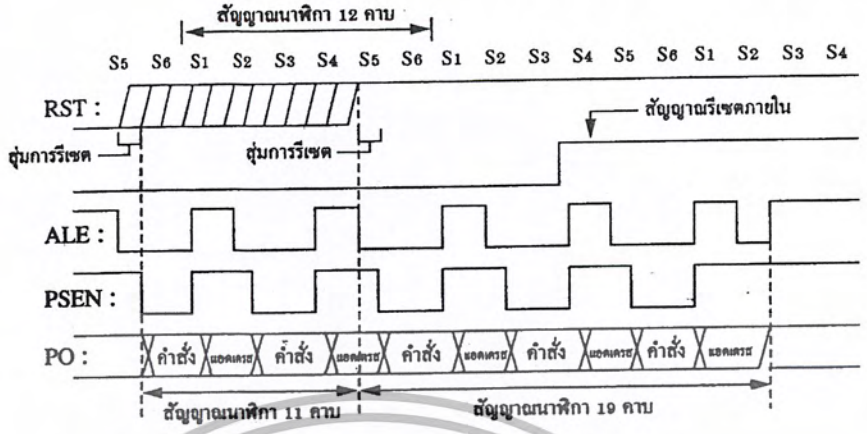


รูปที่ 2.6 ตำแหน่งของหน่วยความจำทั้งแบบไบต์และแบบบิต

2.1.4 การรีเซต MCS – 51

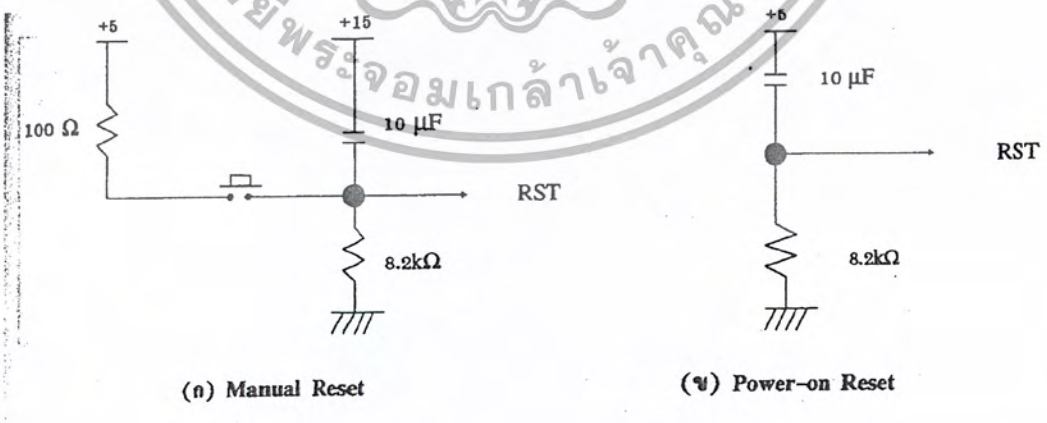
การรีเซต MCS – 51 เป็นการให้ MCS – 51 เริ่มต้นทำงานโปรแกรมใหม่ทั้งหมด เมื่อระบบมีการทำงานผิดพลาดการรีเซตตัวมันก็เป็นสิ่งจำเป็น รวมทั้งเมื่อเริ่มต้นจ่ายไฟให้กับระบบก็จะเป็นการรีเซตให้กับระบบใหม่ด้วย การรีเซต MCS – 51 จะเป็นแบบ Positive Active โดยต้องให้ลอจิก “1” ที่ขา RST เป็นเวลาอย่างน้อย 2 แมชชีนไซเคิล หรือสัญญาณนาฬิกา 24 ลูกตัว MCS – 51 จึงจะจับสัญญาณรีเซตได้ จากนั้น MCS – 51 จะมีการสร้างสัญญาณรีเซตขึ้นภายในตัวมันเอง ดังรูปที่ 2.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 การสร้างสัญญาณรีเซต

การสร้างสัญญาณรีเซตจะต้องทำให้เป็นลอจิก "1" เป็นเวลาอย่างน้อย 2 แมชชีนไซเคิล และให้สัญญาณกลับเป็นลอจิก "0" ตามเดิม เราอาจทำได้โดยสวิตช์กดดังรูปที่ 2.8 (ก) โดยถ้าไม่มีการกดสวิตช์สัญญาณ RST จะเป็นลอจิก "0" ถ้ามีการกดสวิตช์จะทำให้สัญญาณเป็นลอจิก "1" เมื่อปล่อยสวิตช์ตัวเก็บประจุจะคลายประจุออกมาทำให้สามารถรีเซตระบบได้



(ก) Manual Reset (ข) Power-on Reset

รูปที่ 2.8 การรีเซต MCS – 51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Register (S)	Counter
Program Counter	0000H
Accumulator	00H
B Register	00H
PSW	00H
SP	07H
DPTR	0000H
Port 0-3	FFH
IP (8031/8051)	XXX00000B
IP (8032/8052)	XX000000B
IE (8031/8051)	0XX00000B
IE (8032/8052)	0X000000B
Timer Registers	00H
SCON	00H
SBUF	00H
PCON (NMOS)	0XXXXXXXB
PCON (NMOS)	0XXX0000B

ตารางที่ 2.3 แสดงค่าต่างๆ ที่เกิดหลังการรีเซต

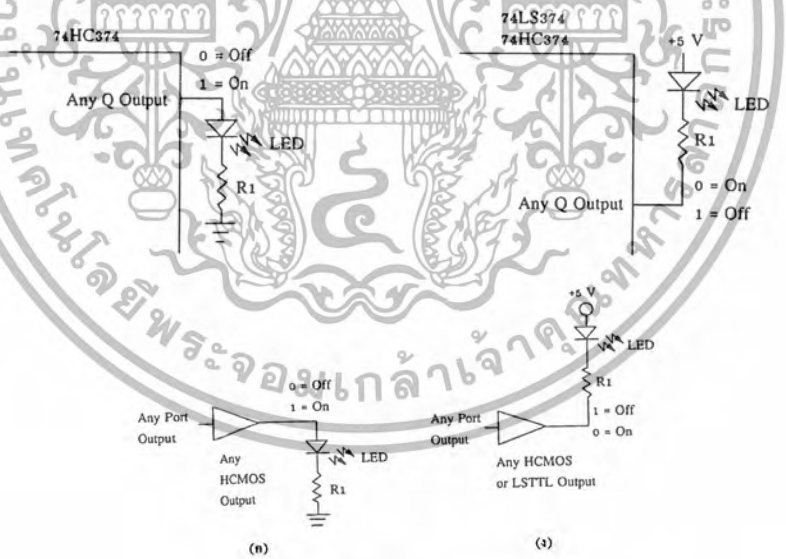
สำหรับการรีเซตเมื่อเริ่มต้นจ่ายพลังงานจะเรียกว่าเพาเวอร์ออนรีเซต(Power-on Reset) เราอาจต่อวงจรได้ดังรูปที่ 2.8 (ข) โดยเริ่มจ่ายพลังงานจะทำให้ขา RST มีค่าเป็นลอจิก "1" ช่วงเวลาหนึ่งโดยขึ้นกับค่าของตัวเก็บประจุ จากวงจรถ้าเริ่มต้นจ่ายไฟตัวเก็บประจุจะยังไม่มีประจุไฟฟ้าอยู่จะทำให้เริ่มมีกระแสไหลและมีการประจุไฟฟ้าในตัวเก็บประจุ แรงดันไฟฟ้าจะตกคร่อมตัวต้านทานเท่ากับ +5 โวลต์ เมื่อเวลาผ่านไปตัวเก็บประจุมีประจุไฟฟ้าเต็มจะทำให้ขา RST มีแรงดันเป็น 0 โวลต์ การกำหนดค่าของ R-C นี้จะต้องกำหนดให้เหมาะสมกับสัญญาณนาฬิกาที่ป้อนให้กับ MCS – 51 ด้วย โดยจะต้องให้เวลาของการประจุไฟฟ้านานกว่า 2 แมชชีนไซเคิล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ MCS – 51 ถูกรีเซต ค่าในรีจิสเตอร์ใช้งานเฉพาะต่างๆ จะมีค่าเป็น “0” ยกเว้นรีจิสเตอร์ P0 – P3, Stack Pointer และ SBUF ดังแสดงในตารางที่ 2.3 โดยค่า P0 – P3 จะเป็นลอจิก “1” ทุกบิต ค่า Stack Pointer จะเป็น 07H สำหรับค่า SBUF จะมีค่าไม่แน่นอนสำหรับหน่วยความจำข้อมูลภายในภายหลังการรีเซตจะมีค่าไม่เปลี่ยนแปลง แต่ถ้าเป็นการรีเซตแบบเพาเวอร์ออนรีเซต ค่าในหน่วยความจำภายในจะมีค่าไม่แน่นอน หลังจากที่ MCS – 51 ถูกรีเซตแล้วค่ารีจิสเตอร์ PC จะชี้ไปที่ตำแหน่ง 0000H ซึ่งเป็นการให้ MCS – 51 เริ่มต้นการทำงานในตำแหน่งแรกนั้นเอง

2.1.5 การเชื่อมต่อหลอด LED

ตัว MCS – 51 และไอซีพอร์ทขนานเบอร์ 8255 เอาต์พุตของมันไม่สามารถขับหลอด LED แต่ถ้าเราฝืนเอาไปต่อก็อาจใช้ได้แต่อาจทำให้ชิพพัง เพราะไอซีเหล่านั้นขับกระแสได้ไม่พอ แต่หลอด LED สามารถต่อกับไอซี TTL บางเบอร์ หรือไอซี 74LS373 ได้ โดยการต่อจะต้องมีตัวต้านทานต่ออนุกรมเพื่อให้ได้กระแสในย่านที่ต้องการ ในรูปที่ 2.9 แสดงการต่อพอร์ทกับหลอด LED โดยรูป (ก) จะสว่างเมื่อเอาต์พุตเป็นลอจิก “1” รูป (ข) จะสว่างเมื่อเอาต์พุตเป็นลอจิก “0” สำหรับรูป (ค) และ (ง) จะใช้บัฟเฟอร์มาช่วยเมื่อต่อกับพอร์ทของระบบไมโครคอนโทรลเลอร์ทั่วๆไป



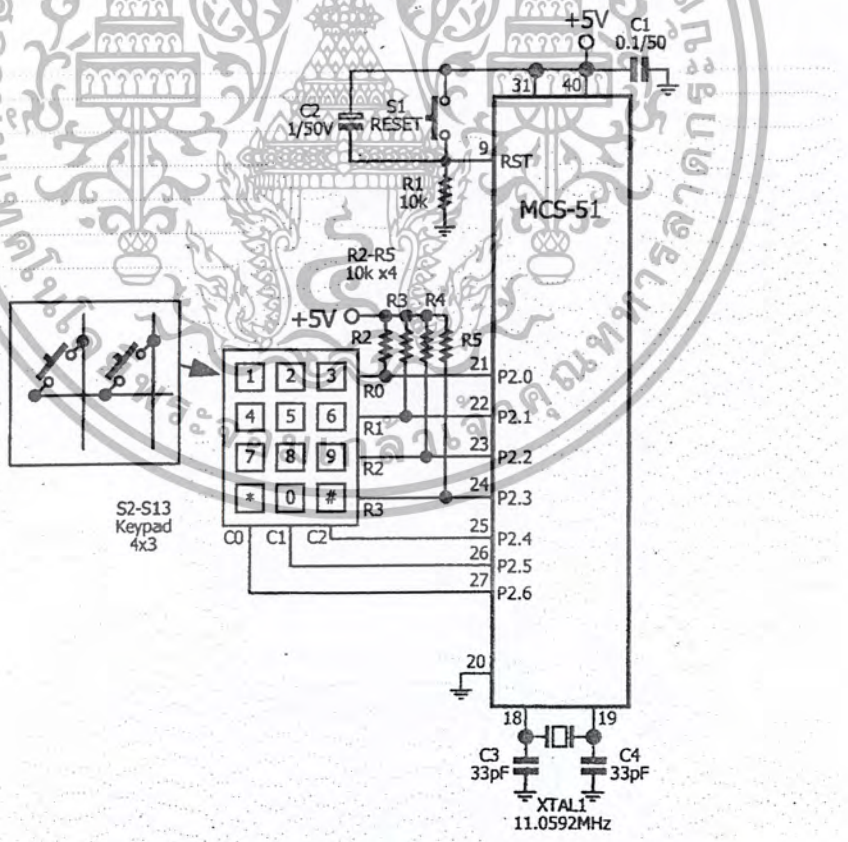
สี LED	R
Red	180
Green	150
Yellow	150
Blue	150

รูปที่ 2.9 แสดง LED แต่ละแบบ และการต่อเข้ากับพอร์ท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.6 การเชื่อมต่อคีย์แพดเข้ากับไมโครคอนโทรลเลอร์ MCS – 51

ในรูปที่ 2.10 จะใช้พอร์ท 2 ของไมโครคอนโทรลเลอร์เชื่อมต่อเข้ากับคีย์แพดทั้ง 7 เส้นคือ สายของคอลัมน์ 3 สาย C0-C2 และสายทางโรว์หรือแถวอีก 4 สายคือ R0-R3 โดยเฉพาะที่ขาพอร์ท P2.0-P2.3 จะต้องต่อตัวต้านทานพูลอัปไว้เพื่อกำหนดสภาวะเริ่มต้นที่ไม่มีการกดคีย์ โดยไมโครคอนโทรลเลอร์จะทำการส่งข้อมูล "0" ไปยัง P2.6, P2.5 และ P2.4 ตามลำดับ ในทุกครั้งที่มีการส่งข้อมูลไปยังสายคอลัมน์ของคีย์แพด MCS – 51 จะทำการอ่านค่าที่ P2.0-P2.3 เข้ามาด้วย หากไม่มีการกด ค่าของ P2.0-P2.3 ก็จะเป็น "1" ทั้งหมด ถ้าหากมีการกดคีย์ ค่าของ P2.0-P2.3 ก็จะไม่เป็น 1111 อีกต่อไป เป็นการแจ้งให้ทราบว่ามีการกดคีย์แพดขึ้นแล้ว จากนั้นไมโครคอนโทรลเลอร์ก็จะทำการค้นหาตำแหน่งต่อไป โดยการค้นหาตำแหน่งของคีย์นั้น จากนั้นก็จะนำค่าตำแหน่งนั้นไปเปิดตารางข้อมูล เพื่อที่จะได้หมายเลขของคีย์ที่กดอย่างแท้จริง ยกตัวอย่างจากวงจรในรูปที่ 2.10 หากคีย์ 0 ถูกกดไมโครคอนโทรลเลอร์จะได้ค่าตำแหน่งของคีย์ 0 เป็น 0BH จากนั้นนำค่า 0BH นี้ไปเปิดตารางก็จะได้ข้อมูลเป็น 0 หมายถึงคีย์ 0 ซึ่งกระบวนการหลังนี้จะเป็นหน้าที่ของซอฟต์แวร์



รูปที่ 2.10 วงจรเชื่อมต่อคีย์แพดเข้ากับไมโครคอนโทรลเลอร์ MCS – 51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.2 พอร์ทอนุกรมของไมโครคอนโทรลเลอร์ MCS – 51

ไมโครคอนโทรลเลอร์ MCS – 51 มีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์ 1 ชุด (วงจรสื่อสารแบบฟูลดูเพล็กซ์ หมายถึง วงจรสื่อสารที่สามารถทำการรับและส่งข้อมูลในลักษณะ 2 ทิศทางได้ในเวลาเดียวกัน) โดยใช้ขาสัญญาณของพอร์ท 3 คือ ขา P3.0 เป็นขารับข้อมูลเข้าหรือ RxD และขา P3.1 เป็นขาส่งข้อมูลออกหรือ TxD โดยวงจรสื่อสารข้อมูลแบบอนุกรมของไมโครคอนโทรลเลอร์ตระกูล MCS – 51 แบบแฟล็ชเป็นแบบอะซิงโครนัสปกติแล้วพอร์ทอนุกรมของไมโครคอนโทรลเลอร์ MCS – 51 จะใช้ในการติดต่อสื่อสารกับพอร์ทอนุกรมของคอมพิวเตอร์ โดยใช้มาตรฐาน RS-232 แต่ในปัจจุบันสามารถติดต่อกันในมาตรฐาน RS-422 หรือ RS-485 ได้แล้วโดยใช้ไอซีพิเศษที่ทำหน้าที่ในการแปลงสัญญาณการสื่อสารดังกล่าว

2.2.1 การสื่อสารข้อมูลแบบอะซิงโครนัส

การสื่อสารข้อมูลแบบอะซิงโครนัสคือการรับและส่งข้อมูล โดยไม่จำเป็นต้องมีสัญญาณนาฬิการ่วมด้วย แต่จะใช้การกำหนดค่าอัตราเร็วในการรับและส่งข้อมูลให้มีค่าเท่ากัน ซึ่งเรียกอัตราเร็วนี้ว่า อัตราบอด หรือบอดเรต (baud rate) มีหน่วยเป็น บิตต่อวินาที (bit per second : bps)

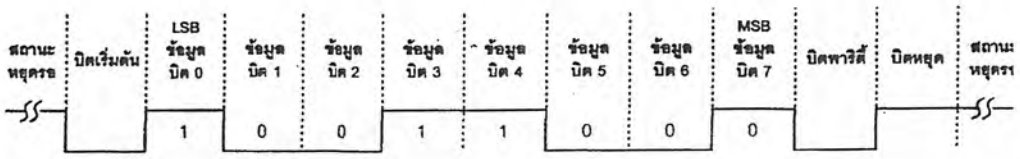
รูปแบบของข้อมูลที่ใช้ในการรับส่งข้อมูลที่ใช้ในการรับส่งแบบอะซิงโครนัสประกอบด้วย 4 ส่วนด้วยกันคือ

1. บิตเริ่มต้น (start bit) มีขนาด 1 บิต
2. บิตข้อมูลแบบอนุกรม มีขนาด 8 บิต
3. บิตตรวจสอบพาริตี้ (parity bit) มีขนาด 1 บิต หรือไม่มี
4. บิตปิดท้ายหรือบิตหยุด (stop bit) มีขนาด 1 บิต

รูปที่ 3.1 แสดงรูปแบบของข้อมูลอนุกรมแบบอซิงโครนัส เมื่อไม่มีการส่งข้อมูล ขา DATA จะมีสถานะลอจิก “1” เรียกสถานะนี้ว่า สถานะหยุดรอ (waiting stage) การเริ่มต้นส่งข้อมูลจะเริ่มจากการให้ขา DATA มีลอจิก “0” ด้วยช่วงเวลา 1 บิต เรียกบิตนี้ว่า บิตเริ่มต้น (start bit) จากนั้นบิตข้อมูลจะถูกส่งออกไป โดยเริ่มจากบิตที่มีนัยสำคัญต่ำสุดหรือบิต LSB ก่อน ซึ่งข้อมูลที่ต้องการส่งมีจำนวน 8 บิต จากนั้นตามด้วย บิตพาริตี้ ซึ่งใช้ในการตรวจสอบความผิดพลาดที่เกิดขึ้นจากการส่งข้อมูล บิตสุดท้ายที่จะส่งคือ บิตปิดท้าย หรือบิตหยุด โดยจะเป็นการทำให้ขา DATA มีสถานะลอจิก “1” อีกครั้งด้วยระยะเวลาอย่างน้อย 1 บิต,1.5 บิต หรือ 2 บิต เพื่อเป็นการแสดงว่าสิ้นสุดข้อมูลแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

55790



รูปที่ 2.11 รูบแบบข้อมูลอนุกรมแบบอะซิงโครนัส

อัตราความเร็วในการรับและส่งข้อมูลของการรับส่งข้อมูลแบบอะซิงโครนัสหรืออัตราบอดหรือบอดเรตที่ใช้สำหรับพอร์ทอนุกรม RS-232 มีด้วยกันหลายค่า ตั้งแต่ 110 ถึง 19,200 บิตต่อวินาที โดยมีค่าเพิ่มมากขึ้นตามเทคโนโลยีของคอมพิวเตอร์ เนื่องจากอัตราบอดคือค่าของจำนวนบิตที่สามารถส่งได้ใน 1 วินาทีสมมติว่า ข้อมูลอนุกรมมีขนาด 8 บิต ไม่มีการตรวจสอบพาริตี้ มีบิตเริ่มต้น 1 บิต และบิตปิดท้าย 1 บิต ความยาวของข้อมูล 1 ไบต์จะมีความยาวเท่ากับ 10 บิต ถ้าใช้บอดเรตในการส่งข้อมูลเท่ากับ 9,600 บิตต่อวินาที ก็จะสามารถรับส่งข้อมูลได้ด้วยความเร็ว 960 ไบต์ต่อวินาที

การตรวจสอบพาริตี้สามารถกำหนดให้เป็นแบบคี่ (odd),แบบคู่ (even) หรือไม่มีการตรวจสอบพาริตี้ก็ได้ พาริตี้คู่แสดงถึงจำนวนลอจิก “1” ทั้งหมดภายในข้อมูลที่ส่งไป 1 ไบต์รวมบิตพาริตี้ว่ามีจำนวนเป็นเลขคู่หรือคี่ ยกตัวอย่าง ข้อมูลที่จะทำการส่งมีขนาด 8 บิต มีค่าเท่ากับ 99H หรือ 10011001B จะเห็นว่าข้อมูลในไบต์นี้มีจำนวนลอจิก “1” จำนวน 4 ตัวซึ่งเป็นเลขคู่ ดังนั้นถ้ากำหนดค่าพาริตี้เป็นคู่ ค่าของบิตพาริตี้จะต้องมีลอจิกเป็น “0” แต่ถ้ากำหนดพาริตี้เป็นคี่ ค่าของบิตพาริตี้จะต้องเป็น “1” เพื่อให้ข้อมูล 1 ไบต์รวมทั้งบิตพาริตี้เป็นคี่

บิตพาริตี้ถูกสร้างขึ้นจากภาคส่งข้อมูลของ UART (Universal Asynchronous Reciver Transmitter:เป็นอุปกรณ์ที่ใช้ในการรับส่งข้อมูลอนุกรม) ซึ่งทางภาครับต้องกำหนดการตรวจสอบพาริตี้ที่ตรงกันเอาไว้ว่าจะตรวจสอบพาริตี้คี่หรือคู่ จากนั้นภาครับของ UART จะทำการตรวจสอบค่าพาริตี้ที่เกิดขึ้นว่าเป็นคู่หรือเป็นคี่ โดยการนับจำนวนลอจิก “1” ทั้งหมดรวมทั้งบิตพาริตี้ด้วย ถ้ากำหนดพาริตี้ไว้เป็นคู่แต่อ่านค่าตัวเลขในการนับออกมาได้ตัวเลขเป็นคี่ ทางภาครับจะแสดงข้อผิดพลาดออกมาให้ผู้ใช้ทราบกระบวนการดังกล่าวเป็นวิธีการตรวจสอบความผิดพลาดที่เกิดขึ้นในการรับส่งข้อมูลที่ง่ายที่สุด แต่มันสามารถตรวจสอบได้เมื่อมีบิตข้อมูลที่ทำการรับส่งผิดพลาดเพียงบิตเดียวเท่านั้น ถ้าข้อมูลที่ทำการส่งมีบิตที่ผิดพลาดมากกว่า 1 บิตการตรวจสอบวิธีนี้จะไม่ได้ผล สำหรับการตั้งพาริตี้เป็น NONE นั้นทั้งภาครับและส่งจะไม่มีการตรวจสอบพาริตี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 รีจิสเตอร์ที่เกี่ยวข้องกับการทำงานของพอร์ทอนุกรมใน MCS-51

ในการทำงานของพอร์ทอนุกรมในไมโครคอนโทรลเลอร์ MCS-51 มีรีจิสเตอร์ที่เกี่ยวข้องอยู่ 2 ตัว ดังนี้

2.2.2.1 รีจิสเตอร์บัฟเฟอร์พอร์ทอนุกรมหรือ SBUF (Serial data buffer register)

มีแอดเดรสอยู่ที่ 99H ในพื้นที่ของรีจิสเตอร์ฟังก์ชั่นพิเศษหรือ SFR มีขนาด 8 บิต แบ่งเป็น 2 ส่วนคือรีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูล (transmit buffer register) และรับข้อมูล (receive buffer register) เมื่อมีการเขียนข้อมูลมายังรีจิสเตอร์ SBUF ข้อมูลนั้นจะถูกส่งต่อไปยังบัฟเฟอร์สำหรับส่งข้อมูล เพื่อส่งออกจากไมโครคอนโทรลเลอร์ผ่านทางขา TxD หรือขา P3.1 ในกรณีที่มีการอ่านข้อมูลจากรีจิสเตอร์ SBUF ข้อมูลจะถูกส่งผ่านไปยังรีจิสเตอร์บัฟเฟอร์สำหรับรับข้อมูลเพื่อส่งต่อไปยังไมโครคอนโทรลเลอร์ต่อไป สำหรับการรับข้อมูลอนุกรมจากภายนอกนั้นจะผ่านมาทางขา RxD หรือ P3.0 ของไมโครคอนโทรลเลอร์ MCS-51 แบบเฟลช

2.2.2.2 รีจิสเตอร์ควบคุมการทำงานของพอร์ทอนุกรมหรือSCON (Serial port Control Register)

เป็นรีจิสเตอร์ขนาด 8 บิต มีแอดเดรสอยู่ที่ 98H ในพื้นที่ของรีจิสเตอร์SFR สามารถเข้าถึงได้ในระดับบิตมีรายละเอียดการทำงานดังนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SMO-SM1 (Serial port mode bit 0-1) : ใช้ในการเลือกโหมดการทำงานของพอร์ทอนุกรมภายในไมโครคอนโทรลเลอร์ MCS-51 ดังมีรายละเอียดต่อไปนี้

SM2 : ใช้ในการเอ็นเอเบิลการสื่อสารในแบบมัลติโปรเซสเซอร์ (multiprocessor) ในการทำงานของโหมด 2 และโหมด 3 ของพอร์ทอนุกรมในไมโครคอนโทรลเลอร์ MCS-51 ถ้าบิตนี้เป็น " 1" บิต RI จะไม่แอกตีฟถ้าบิตที่ 9 ที่รับเข้ามาเป็น "0" (ข้อมูลบิตที่ 9 เก็บไว้ที่บิต RB8) ในการทำงานโหมด 1 ถ้าบิตนี้เซต บิต RI จะไม่แอกตีฟถ้ายังไม่ได้รับบิตหยุด ส่วนในโหมด 0 บิตนี้ไม่มีการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

REN (Enable serial reception) : ใช้ในการเอ็นเอเบิลการรับข้อมูลของพอร์ทอนุกรม ทำการเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์ ถ้าต้องการให้มีการรับข้อมูลต้องเซตบิตนี้ให้เป็น "1"

TB8 : ใช้สำหรับเก็บข้อมูลบิตที่ 9 ที่ต้องการส่งออกไปในการทำงานโหมด 2 และ 3 ของพอร์ทอนุกรมในไมโครคอนโทรลเลอร์ MCS-51 เซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

RB8 : ใช้สำหรับรับข้อมูลบิตที่ 9 ที่เข้ามาในการทำงานโหมด 2 และ 3 ของพอร์ทอนุกรมในไมโครคอนโทรลเลอร์ MCS-51 แต่ถ้าหากพอร์ทอนุกรมทำงานอยู่ในโหมด 1 และบิต SM2 เป็น "0" ข้อมูลที่บิต RB8 คือข้อมูลของบิตหยุด (STOP bit) สำหรับในการโหมด 0 บิตนี้จะไม่ใช้งาน บิต RB8 นี้สามารถเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

TI (Transmit Interrupt flag) : ใช้ในการแสดงการเกิดอินเตอร์รัปต์เมื่อมีการส่งข้อมูลออกจากพอร์ทของอนุกรมไมโครคอนโทรลเลอร์ MCS-51 สามารถเซตได้ด้วยกระบวนการทางฮาร์ดแวร์ เมื่อทำการส่งข้อมูลบิตที่ 8 ไปเรียบร้อยแล้วในการทำงานโหมด 0 ส่วนในการทำงานโหมดอื่น บิตนี้จะเซตเมื่อมีการเริ่มต้นส่งบิตหยุดออกไป การเคลียร์บิตนี้ต้องใช้กระบวนการทางซอฟต์แวร์เท่านั้น

RI (Receive Interrupt flag) : ใช้แสดงการเกิดอินเตอร์รัปต์เมื่อมีการรับข้อมูลเข้าสู่พอร์ทอนุกรม สามารถเซตได้ด้วยกระบวนการทางฮาร์ดแวร์ เมื่อทำการรับข้อมูลบิตที่ 8 เรียบร้อยแล้วในการทำงานโหมดอื่น บิตนี้จะเซตเมื่อมีสามารถรับบิตหยุดของข้อมูลอนุกรมไปได้ครึ่งทางแล้ว ยกเว้นในกรณีที่บิต SM2 มีการเซต บิตนี้จะเซตได้ก็ต่อเมื่อการรับบิตหยุดหรือบิตที่ 9 เกิดขึ้นอย่างสมบูรณ์แล้ว การเคลียร์บิตนี้ต้องใช้กระบวนการทางซอฟต์แวร์เท่านั้น

2.2.3 โหมดการทำงานของพอร์ทอนุกรมใน MCS-51

พอร์ทอนุกรมในไมโครคอนโทรลเลอร์ MCS-51 สามารถเลือกทำงานได้ถึง 4 โหมดคือ

1. โหมด 0 เป็นการกำหนดให้พอร์ทอนุกรมทำงานในลักษณะชิฟต์รีจิสเตอร์
2. โหมด 1 เป็นการกำหนดให้เป็น UART ขนาด 8 บิต สามารถเลือกอัตราบอดได้
3. โหมด 2 เป็นการกำหนดให้เป็น UART ขนาด 9 บิต โดยมีอัตราบอดคงที่
4. โหมด 3 เป็นการกำหนดให้เป็น UART ขนาด 9 บิต สามารถเลือกอัตราบอดได้

การเลือกโหมดทำได้ด้วยการกำหนดข้อมูลให้แก่บิต SM0 และ SM1 ในรีจิสเตอร์ SCON

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3.1 การทำงานในโหมด 0 ของวงจรพอร์ทอนุกรม

มีไดอะแกรมการทำงานและไดอะแกรมเวลาแสดงในรูปที่3.2 ข้อมูลอนุกรมจะผ่านเข้าและออกทางขา RxD ส่วนขา TxD ทำหน้าที่เป็นสัญญาณนาฬิกาของการเคลื่อนข้อมูล (Shift clock) ในโหมดนี้มีจำนวนข้อมูล 8 บิต โดยทำการรับและส่งข้อมูลในบิต LSB ก่อน อัตราในการรับส่งข้อมูลหรืออัตราบอดถูกกำหนดไว้คงที่ 1/12 ของความถี่สัญญาณนาฬิกา

เริ่มต้นการส่งข้อมูลด้วยการเขียนข้อมูลที่ต้องการส่งมายังรีจิสเตอร์ SBUF สัญญาณเขียนข้อมูล SBUF แอกตีฟเป็น "1" ที่สเตต 6 เฟส 2 (S6P2) ของแมชีนไซเกิล ส่งมายังวงจรควบคุมการส่ง (TX control) ทำให้วงจรควบคุมเริ่มต้นส่งข้อมูล สัญญาณ SEND จะแอกตีฟเป็น "1" ตลอดการส่งข้อมูล

ข้อมูลจากรีจิสเตอร์ SBUF จะถูกเลื่อนออกที่ขา P3.0 หรือขา RxD ครั้งละบิต ตามจังหวะของสัญญาณนาฬิกาที่ส่งออกมาทางขา P3.1 หรือ TxD โดยสัญญาณนาฬิกาของการเลื่อนข้อมูลจะมีขอบขาลงของสัญญาณที่สเตต 3 เฟส และมีขอบขาขึ้นของสัญญาณที่สเตต 6 เฟส 1 ของแต่ละแมชีนไซเกิลในกระบวนการส่งข้อมูลจนกระทั่งเมื่อส่งข้อมูลครบ 8 บิตแล้ว บิต IT ในรีจิสเตอร์ SCON จะเกิดการเซต เป็นการแจ้งให้ทราบว่าส่งข้อมูลครบแล้ว หากการอินเตอร์รัปต์จากพอร์ทอนุกรมได้รับการเอ็นเอเบิลไว้ ก็จะเกิดการอินเตอร์รัปต์ขึ้นในระบบ เมื่อเสร็จสิ้นกระบวนการรับข้อมูล สัญญาณ SEND จะกลายเป็น "0" จนกว่าจะเริ่มต้นกระบวนการรับข้อมูลใหม่

ในกระบวนการรับข้อมูล เริ่มต้นด้วยการเซต REN ให้เป็น "1" และเคลียร์ RI ในรีจิสเตอร์ SCON ก่อนที่สเตต 6 เฟส 2 ของแมชีนไซเกิลถัดไป วงจรควบคุมการรับ (RX control) จะทำการเขียนข้อมูล 11111110 ไปยังชิฟรีจิสเตอร์สำหรับรับข้อมูลและทำการแอกตีฟสัญญาณ RECEIVE ให้เป็น "1" ในสัญญาณนาฬิกาลูกถัดไป

เมื่อสัญญาณ RECEIVE แอกตีฟ ก็จะเกิดการส่งสัญญาณนาฬิกาของการเลื่อนข้อมูล (Shift clock) ขึ้นผ่านทางขา P3.1 หรือ TxD เพื่อทำการกำหนดจังหวะการรับข้อมูลครั้งละบิต โดยสัญญาณนาฬิกานี้จะเกิดขึ้นในช่วงสเตต 3 เฟส 1 ถึงสเตต 6 เฟส 1 ของแต่ละแมชีนไซเกิล การรับข้อมูลเข้ามาทางขา P3.0 หรือ RxD จะเกิดขึ้นที่สเตต 5 เฟส 2 ในแมชีนไซเกิลเดียวกับสัญญาณนาฬิกาของการเลื่อนข้อมูล จนกระทั่งรับข้อมูลครบทั้ง 8 บิต บิต RI จะได้รับการเซตเพื่อแจ้งการเสร็จสิ้นกระบวนการรับข้อมูล หากการอินเตอร์รัปต์จากพอร์ทอนุกรมได้รับการเอ็นเอเบิลไว้ ก็จะเกิดการอินเตอร์รัปต์ขึ้นในระบบ เมื่อเสร็จสิ้นกระบวนการรับข้อมูล สัญญาณ RECEIVE จะกลายเป็น "0" จนกว่าจะเริ่มต้นกระบวนการรับข้อมูลใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

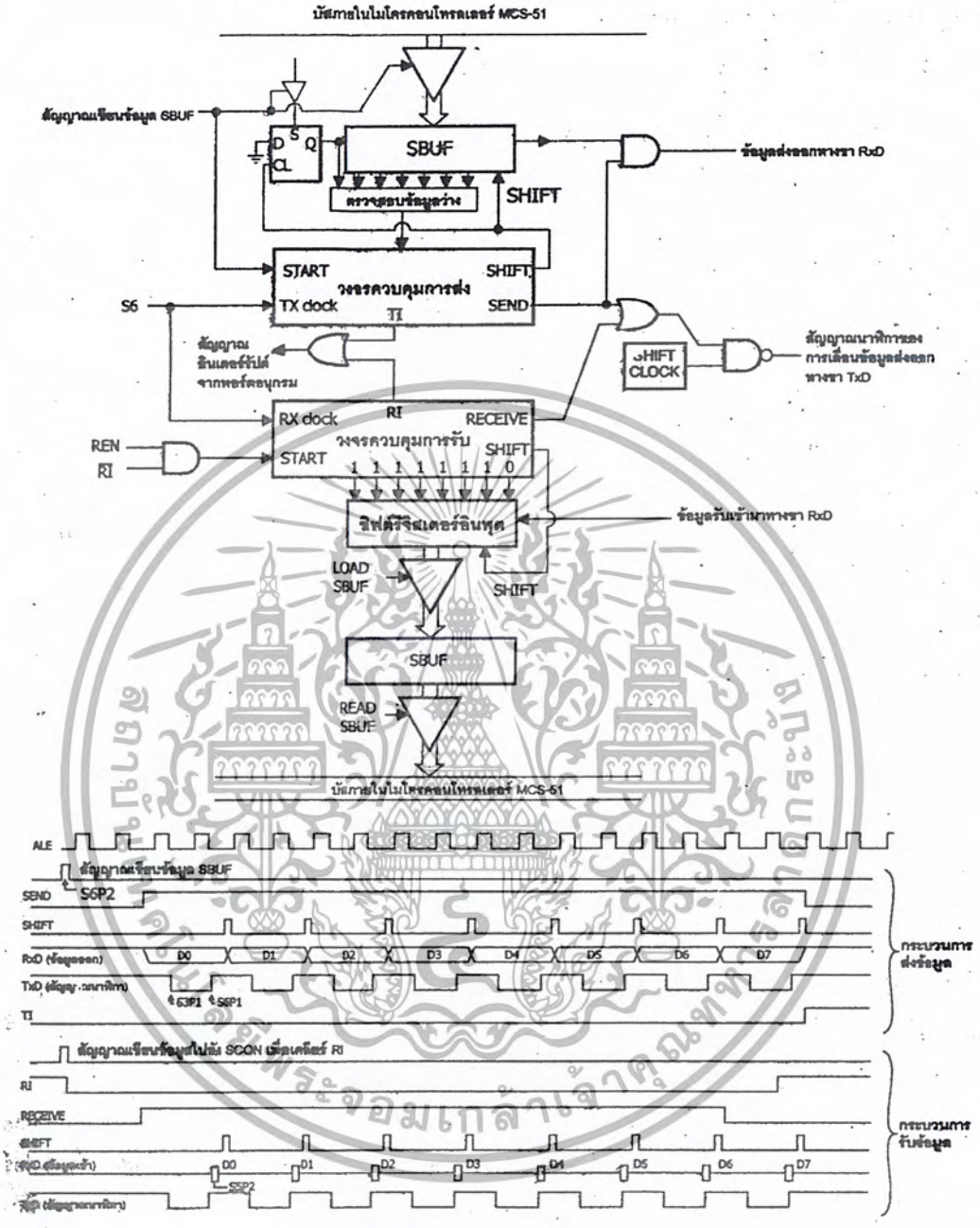
การทำงานในโหมดนี้ของพอร์ทอนุกรมในไมโครคอนโทรลเลอร์ MCS-51 จะใช้ในการเชื่อมต่อกับไอซีรีจิสเตอร์ภายนอกเพื่อทำการขยายจำนวนพอร์ทอินพุตหรือเอาต์พุต แต่ไม่เป็นที่นิยมใช้งานมากนัก เนื่องจากในไมโครคอนโทรลเลอร์ MCS-51 เองมีพอร์ทอยู่ค่อนข้างมาก และติดต่อกับพอร์ทเหล่านั้นได้ง่ายและเร็วกว่ามาก

2.2.3.2 การทำงานในโหมด 1 ของวงจรพอร์ทอนุกรม

มีไดอะแกรมแสดงในรูปที่ 3.3 ในโหมดนี้ใช้ในการรับส่งข้อมูลรวม 10 บิต โดยส่งข้อมูลออกทางขา P3.1 หรือ TxD และรับข้อมูลเข้าทางขา P3.0 หรือ RxD ข้อมูลทั้ง 10 บิตประกอบด้วยบิตเริ่มต้น(มีค่าเป็น "0") 1 บิต บิตข้อมูล 8 บิต โดยรับหรือส่งข้อมูลในบิต LSB ก่อน และบิตหยุดหรือบิตปิดท้าย(มีค่าเป็น "1") ในการรับข้อมูลบิตหยุดจะถูกเก็บไว้ในบิต RB8 ในรีจิสเตอร์ SCON อัตราบอดในโหมดนี้ได้รับการกำหนดโดยอัตราการเกิดโดเวอร์โฟลวของไทเมอร์ 1 ใน AT89C51 ส่วนในไมโครคอนโทรลเลอร์เบอร์ AT89C52 และในอนุกรม AT89Sxx สามารถเลือกใช้อัตราการเกิดโอเวอร์โฟลวของไทเมอร์ 1 หรือไทเมอร์ 2 ในการกำหนดอัตราบอดได้

กระบวนการส่งข้อมูลเริ่มต้นด้วยการแอกติฟสัญญาณเขียนข้อมูลมายังรีจิสเตอร์ SBUF ส่งมายังวงจรควบคุมการส่ง (TX control) จากนั้นวงจรควบคุมจะทำการแอกติฟสัญญาณ SEND ที่สเตต 1 เฟส 1 ของแมชีนไซเคิลต่อมา โดยสัญญาณ SEND จะเป็น "0" ตลอดการส่งข้อมูล เมื่อสัญญาณ SEND แอกติฟ จะทำการส่งบิตเริ่มต้นก่อนเป็นบิตแรก โดยมีคาบเวลาของบิตเริ่มต้นเท่ากับ 1 แมชีนไซเกิล จากนั้นตามด้วยการส่งบิตข้อมูล 8 บิต เรียงลำดับจากบิต LSB โดยข้อมูลที่ทำการส่งถูกเรียกออกมาจากรีจิสเตอร์บัฟเฟอร์สำหรับการส่งข้อมูล ในทุกๆ บิตข้อมูลที่ทำการส่งออกไป จะเกิดสัญญาณพัลส์ SHIFT ขึ้น เพื่อเรียกข้อมูลในแต่ละบิตจากรีจิสเตอร์บัฟเฟอร์ การกำหนดจังหวะการส่งข้อมูลใช้สัญญาณนาฬิกาการส่ง (TX clock) เป็นตัวกำหนด โดยสัญญาณนาฬิกานี้ได้มาจากการหารสัญญาณ TCLK จากไทเมอร์ 1 ด้วย 16 หลังจากการส่งบิตข้อมูลในแต่ละบิตจากรีจิสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.12 ไดอะแกรมการทำงานในโหมด 0 ของพอร์ทอนุกรม
ภายในไมโครคอนโทรลเลอร์ MCS - 51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3.3 การทำงานในโหมด 2 และ 3 ของวงจรพอร์ทอนุกรม

ในทั้งสองโหมดนี้จะใช้รูปแบบข้อมูลรวม 11 บิต ประกอบด้วยบิตเริ่มต้น มีค่าเป็น "0" จำนวน 1 บิต บิตข้อมูล 8 บิต โดยทำการรับและส่งบิต LSB ก่อน บิตข้อมูลที่ 9 และบิตปิดท้ายมีค่าเป็น "1" จำนวน 1 บิต ในการส่งข้อมูล ข้อมูลบิตที่ 9 จะได้รับการเก็บไว้ในบิต TB8 ในรีจิสเตอร์ SCON และในการรับข้อมูล ข้อมูลบิตที่ 9 จะนำไปเก็บไว้ในบิต RB8 ในรีจิสเตอร์ SCON สำหรับอัตราบอดในโหมด 2 จะคงที่โดยเลือกได้ 2 ค่าคือ 1/32 หรือ 1/64 ของความถี่สัญญาณนาฬิกา สำหรับในโหมด 3 อัตราบอดสามารถปรับได้เหมือนกับโหมด 1

ในรูปที่ 3.4 และ 3.5 เป็นไดอะแกรมการทำงานและไดอะแกรมของการทำงานในโหมด 2 และ 3 ของพอร์ทอนุกรมการทำงาน โดยรวมจะคล้ายกับการทำงานในโหมด 1 ส่วนที่แตกต่างกันคือจำนวนบิตของข้อมูลที่ในโหมด

2.2.4 อัตราบอดของพอร์ทอนุกรมในไมโครคอนโทรลเลอร์ MCS – 51

2.2.4.1 โหมด 0

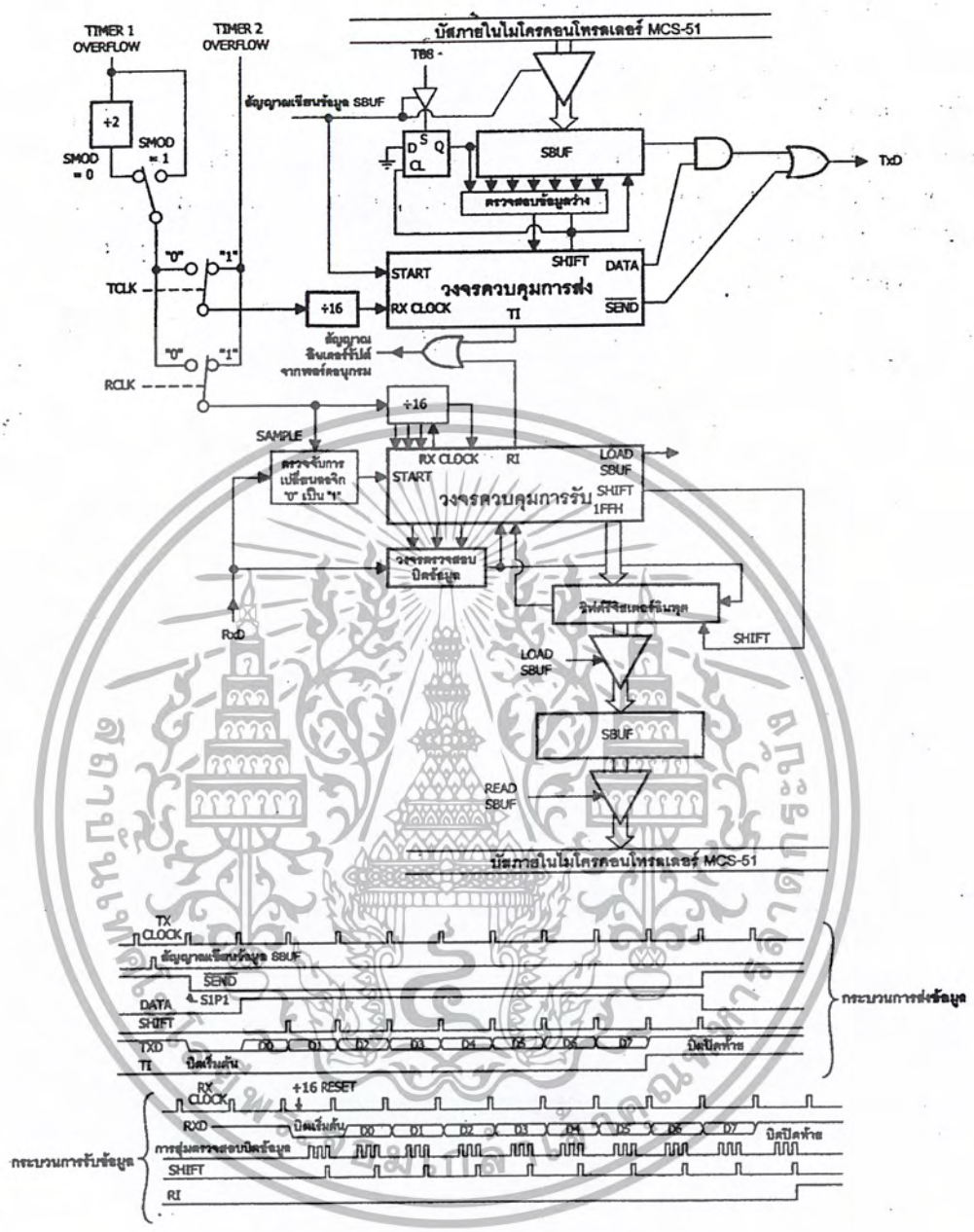
อัตราบอดของโหมดนี้มีค่าคงที่ โดยสามารถคำนวณได้จากสูตร

อัตราบอดในโหมด 0 = ความถี่ของสัญญาณนาฬิกา / 12 หน่วยเป็นบิตต่อวินาที

2.2.4.1 โหมด 1 และ 3

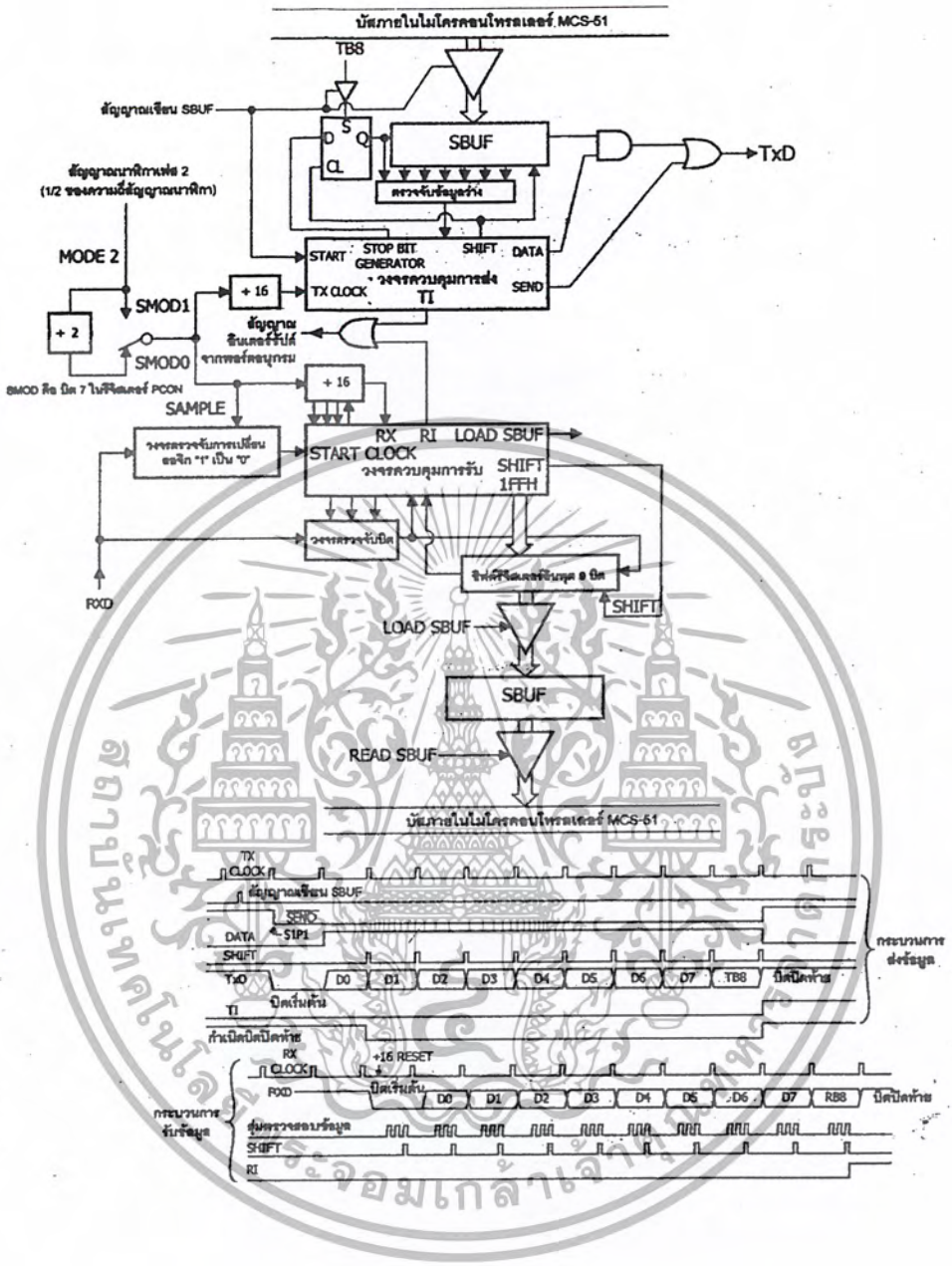
เนื่องจากทั้งสองโหมดนี้สามารถเลือกแหล่งกำเนิดอัตราบอดได้ 2 แหล่ง คือ จากอัตราโอเวอร์โฟลวของไทเมอร์ 1 และ 2 สำหรับอัตราบอดเมื่อใช้การโอเวอร์โฟลวของไทเมอร์ 1 จะต้องใช้ค่าของบิต SMOD ในรีจิสเตอร์ PCON มาพิจารณาด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



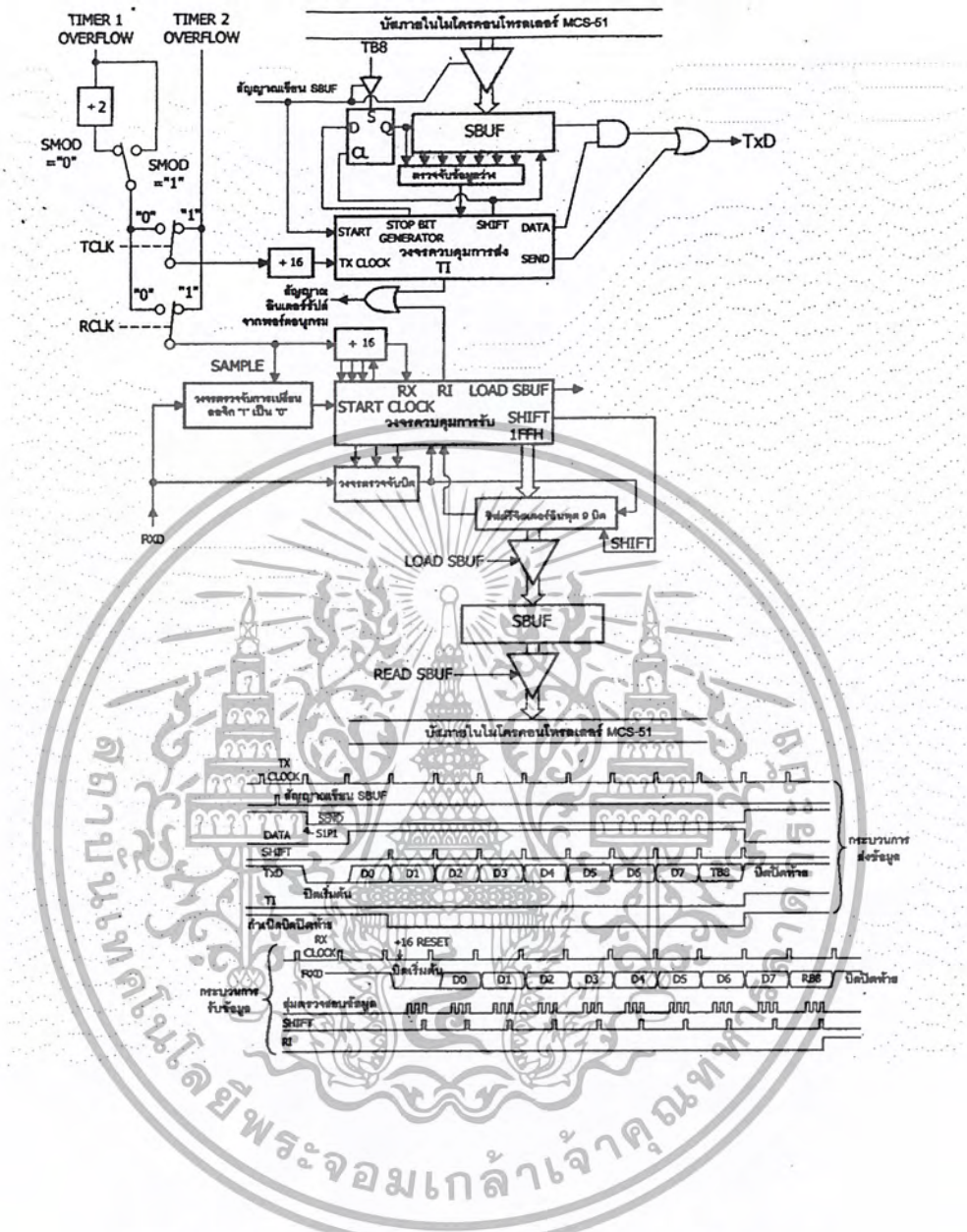
รูปที่ 2.13 ไดอะแกรมการทำงานในโหมด 1 ของพอร์ทอนุกรม ภายในไมโครคอนโทรลเลอร์ MCS - 51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 ไดอะแกรมการทำงานในโหมด 2 ของพอร์ทอนุกรม
ภายในไมโครคอนโทรลเลอร์ MCS – 51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 ไดอะแกรมการทำงานในโหมด 3 ของพอร์ทอนุกรม ภายในไมโครคอนโทรลเลอร์ MCS - 51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
หลักการสื่อสารข้อมูลแบบ RS-485

มาตรฐาน RS-485 ถูกพัฒนาขึ้นโดยอาศัยความร่วมมือระหว่างสองสมาคมคือ Electronics Industries Association (EIA) และ Telecommunications Industry Association (TIA) ซึ่ง **RS** มาจากคำว่า Recommended Standard (ข้อกำหนดมาตรฐาน) ส่วน **485** คือ Identify ของมาตรฐานชุดนี้ ซึ่ง RS-485 จะคล้ายกันกับ RS-422 โดยแตกต่างกันที่ RS-485 ใช้ระบบการสื่อสารแบบ half–duplex แต่ RS-422 ใช้แบบ full-duplex ซึ่งข้อกำหนดพื้นฐานของ RS-485 และ RS-422 แสดงดังตารางที่ 3.1

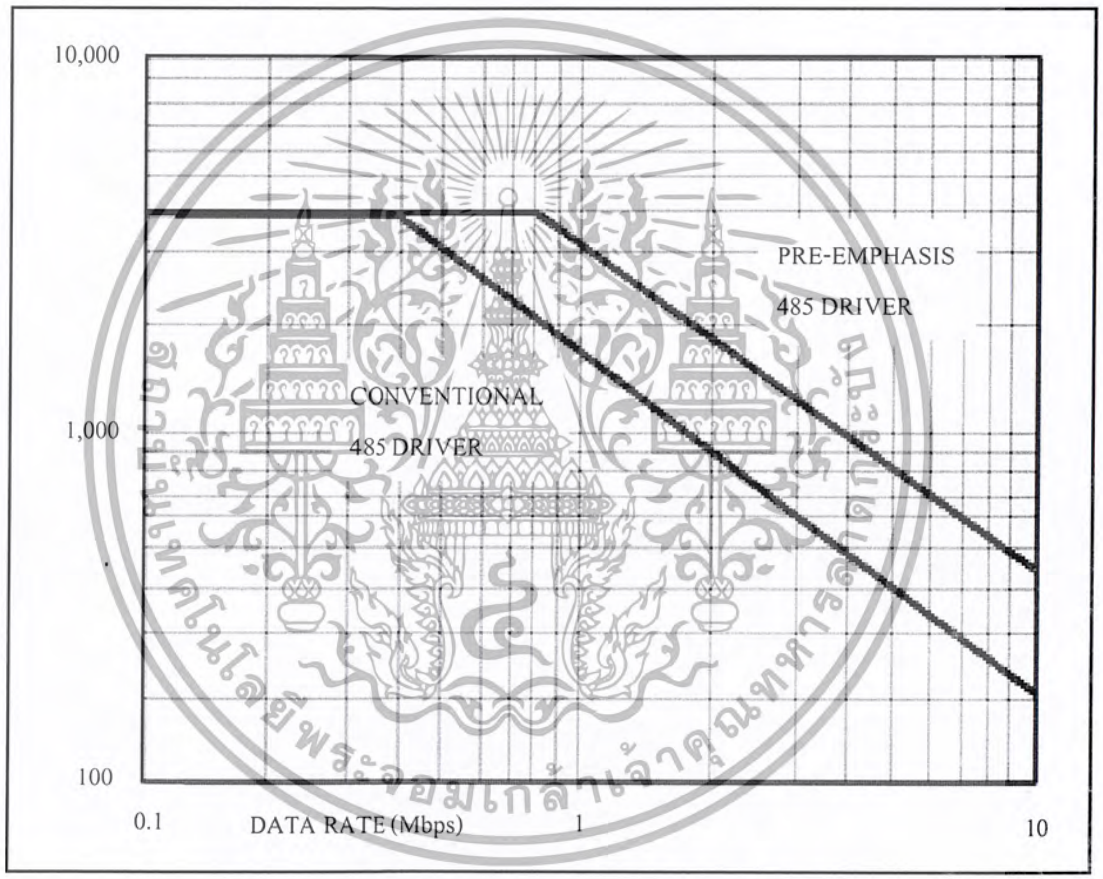
	RS-485	RS-422
Mode of operation	Differential	Differential
Allowed no. of Tx and Rx	1 Tx ,10 Rx	32 Tx , 32 Rx
Maximum cable length	4000 ft	4000 ft
Maximum data rate	10 Mbps	10 Mbps
Minimum driver output range	2V	1.5V
Maximum driver output range	5V	5V
Maximum driver short-circuit current	150 mA	250 mA
Tx load impedance	100 ohm	54 ohm
Rx input sensitivity	200 mV	200 mV
Maximum Rx input resistance	4 k ohm	12 k ohm
Rx input voltage range	7 V	-7 V to +12 V
Rx logic high	More than 200 mV	More than 200 mV
Rx logic low	Less than 200 mV	Less than 200 mV

ตาราง 3.1 ข้อกำหนดของ RS-485 และ RS-422

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1 หลักการสื่อสารข้อมูลของ RS-485

มาตรฐาน RS-485 เป็นการสื่อสารข้อมูลแบบ half-duplex หมายความว่าขณะที่อุปกรณ์ตัวใดตัวหนึ่งใน network กำลังส่งข้อมูล อุปกรณ์ตัวอื่นๆ จะส่งข้อมูลไม่ได้นั่นคือจะต้องเป็นตัวรับข้อมูล(receivers) ซึ่ง RS-485 กำหนดจำนวน receivers สูงสุดได้ 32 ตัว ระยะทางมากที่สุด 1200 เมตร(4,000ฟุต)ที่ 400 kbps โดยจะลดลงเรื่อยๆ เมื่ออัตราการส่งข้อมูลสูงขึ้น จนกระทั่งถึง 10 Mbps ซึ่งเป็นอัตราเร็วสูงสุดของ RS-485 ดังรูปที่ 3.1



รูปที่ 3.1 ความสัมพันธ์ระหว่างอัตราการส่งข้อมูลกับระยะทาง

การสื่อสารแบบบาลานซ์ไลน์

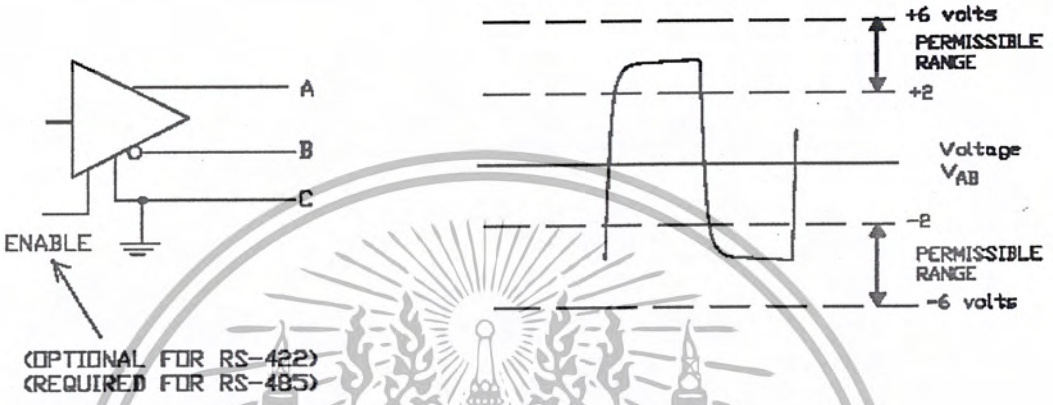
ในระบบบาลานซ์ไลน์ ดิฟเฟอเรนเชียลโวลท์เตจขึ้นอยู่กับไดรฟ์เวอร์ที่อยู่บนขั้วทั้งสองของคู่สายสัญญาณเท่านั้นดังรูปที่ 3.2 บาลานซ์ไลน์ไดรฟ์เวอร์จะกำเนิดโวลท์เตจช่วง 2 ถึง 6 โวลท์ ตรงข้ามกันระหว่าง A และ B และจำเป็นจะต้องมีสัญญาณกราวด์ร่วมด้วย

แม้ว่าการต่อที่ถูกต้องนั้นจำเป็นต้องมีกราวด์ แต่ในตัวรับ(reciever)นั้นไม่จำเป็นต้องใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคส่งสัญญาณ

ภาคส่งสัญญาณ จะต้องมีสัญญาณ “อินาเบิ้ล” ด้วยเพื่อทำการต่อภาคขยายกับเอาท์พุตถ้าไม่มีสัญญาณอีนาเบิ้ลสัญญาณก็จะถูกพิจารณาว่าถูกตัดขาดจากสายส่งสัญญาณ

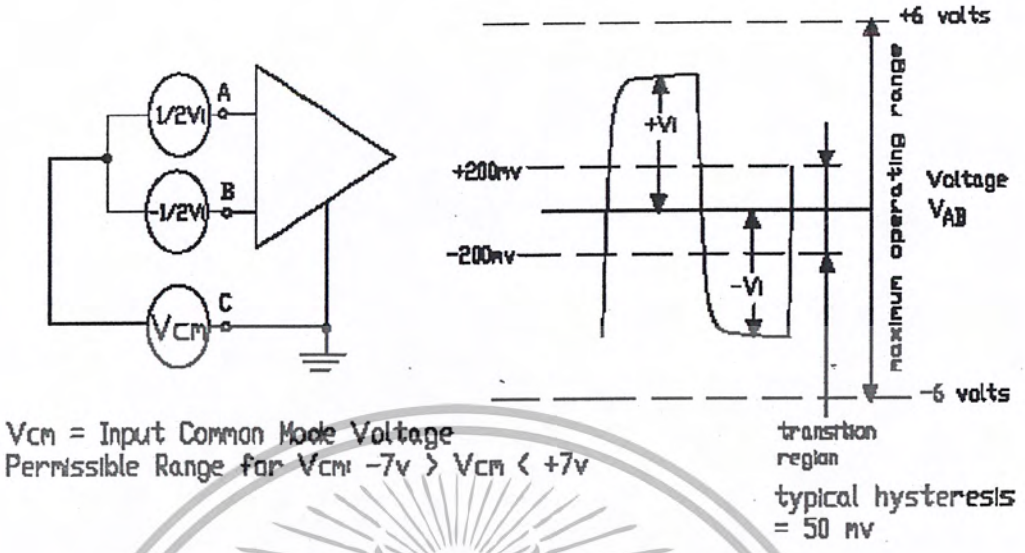


รูปที่ 3.2 สัญญาณเอาท์พุตของไดรฟ์เวอร์

ภาครับสัญญาณ

ภาครับจะเปรียบเทียบความแตกต่างระหว่างสัญญาณอินพุตที่ขา A และ B ซึ่งถ้าความแตกต่างระหว่างขาทั้งสองมากกว่า 200 มิลลิโวลท์จะเป็นสัญญาณบวก(logic 1) แต่ถ้าน้อยกว่า –200 มิลลิโวลท์ก็จะเป็นสัญญาณ ลบ (logic 0) ดังรูป 3.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

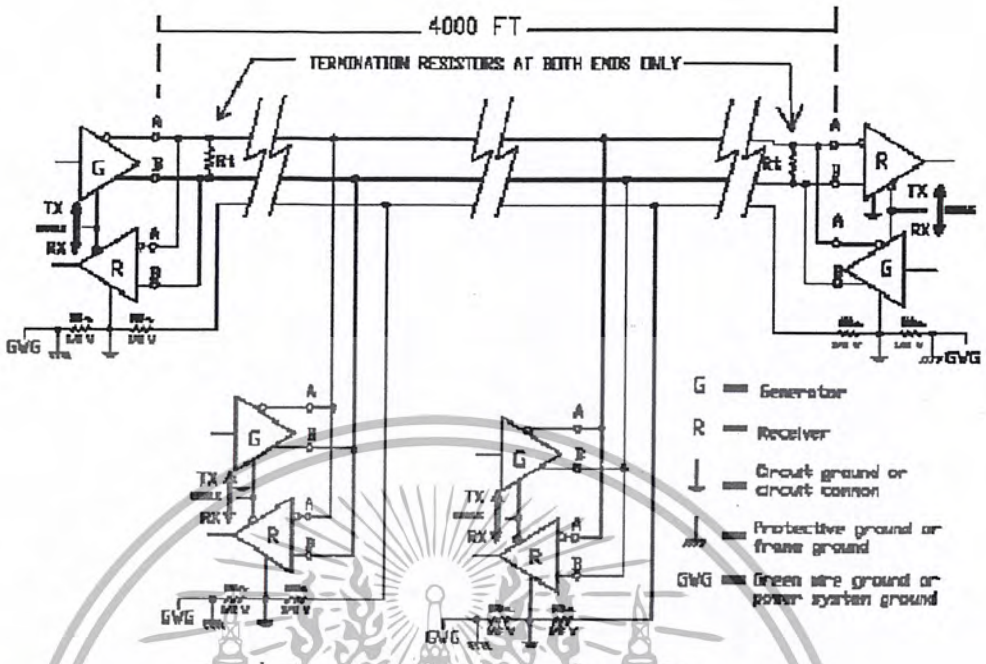


รูปที่ 3.3 การรับสัญญาณอินพุตแบบดิฟเฟอเรนเชียล

3.2 จำนวนอุปกรณ์ต่อพ่วงในระบบ

มาตรฐาน RS-485 แบบบาลานซ์ไลน์ยอมให้มีอุปกรณ์ต่อพ่วงภายในเครือข่ายได้ถึง 32 อันซึ่งมีคอมมอนโหมดโวลท์เตจอยู่ในช่วง +12 และ –7 โวลท์ โดยปกติแล้วการส่งสัญญาณแบบใช้สายสัญญาณสองเส้นมัลติดรอปจะต้องมีความต้านทานเทอมิเนตที่ปลายทั้งสองข้างของสายสัญญาณแต่อุปกรณ์ที่ต่อพ่วงตรงกลางไม่ต้องมีความต้านทานเทอมิเนตดังรูปที่ 3.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

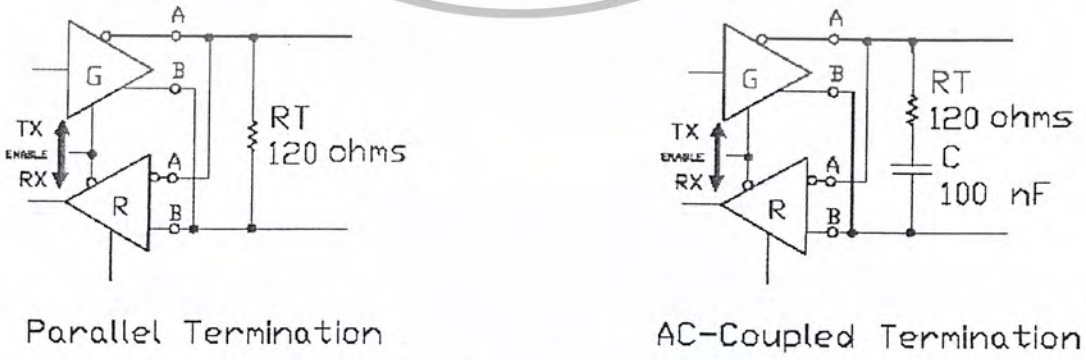


รูปที่ 3.4 การต่อตัวต้านทานเทอมิเนตในระบบ

3.3 การต่อความต้านทานเทอมิเนต

ความต้านทานเทอมิเนตควรจะใช้เมื่อมีอัตราการส่งข้อมูลสูงและการใช้สายที่ยาวๆ ในระบบควรจะสัญญาณกราวน์เพื่อรักษาคอมมอนโหมดโวลท์เตจแต่วงจรที่ต่อพ่วงอาจจะไม่ต่อก็ได้แต่ต้องยอมเสีย reliability และ noise immunity

ความต้านทานเทอมิเนตจะต้องแมทกับสายสัญญาณที่ใช้เพื่อให้เกิดการส่งถ่ายพลังงานสูงสุดโดยจะต่อขนานกับสายส่งสัญญาณ ซึ่งกำหนดโดยผู้ผลิตปกติแล้วจะมีอิมพิแดนซ์ที่ 120 โอห์ม และเพื่อป้องกันผลกระทบทางไฟตรงให้ต่อคาปาซิเตอร์อนุกรมไว้กับความต้านทาน ดังรูปที่ 3.5

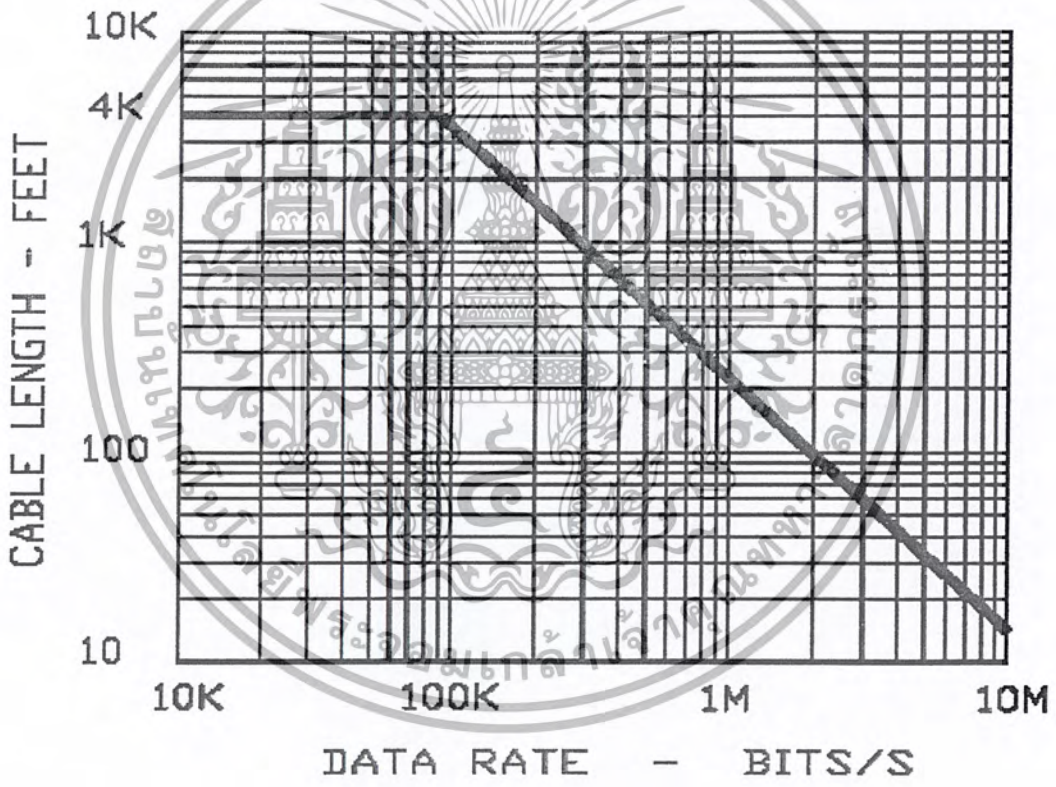


รูปที่ 3.5 การต่อความต้านทานเทอมิเนต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 คุณสมบัติของสายส่งสัญญาณ

การเลือกสายส่งสัญญาณควรคำนึงถึงระยะทางของเน็ตเวิร์ก , บิตเรต ซึ่งสายสายส่งสัญญาณ 24 AWG twist-pair มีคุณลักษณะของสายคือ 16pf/ft อิมพีแดนซ์ 100 โอห์ม และแรงดันจะลดลง 6 dB ตามตารางดังรูปที่ 3.6 ซึ่งการสูญเสียนี้ได้รวมเอาการสูญเสียบนลวดตัวนำ การสูญเสียทางกระแสสลับเนื่องจากวัสดุห่อหุ้ม และค่าไดอิเล็กตริกในสายแล้ว



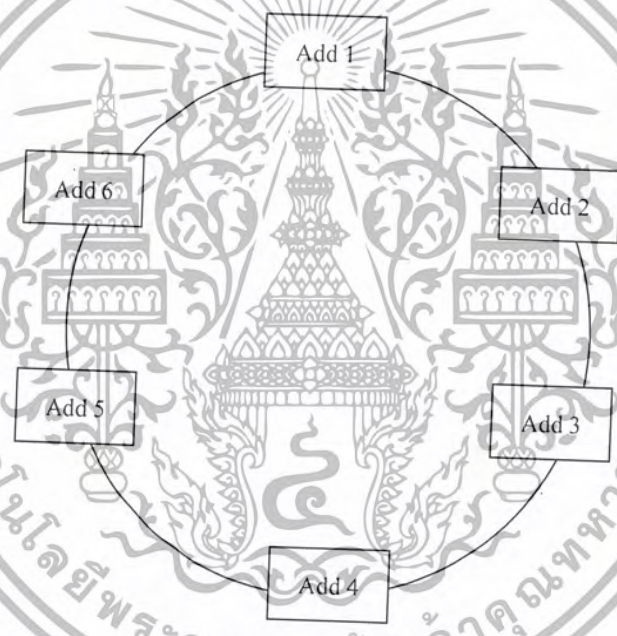
รูปที่ 3.6 อัตราการสูญเสียแรงดัน 6 dB ที่ระยะต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การสร้าง และการใช้งาน

4.1 หลักการทำงานของระบบควบคุมไฟฟ้าอัตโนมัติ

เครื่องควบคุมไฟฟ้าอัตโนมัติ หลักการทำงานของเครื่องควบคุมไฟอัตโนมัติเครื่องนี้ อย่างที่กล่าวมาข้างต้นแล้วว่าใช้หลักการของไมโครโปรเซสเซอร์ MCS – 51 โดยการเขียนโปรแกรมการทำงาน ให้เครื่องหลายๆ เครื่องสามารถติดต่อสั่งงาน หรือเพื่อดูสถานะการทำงานซึ่งกันและกันได้ ซึ่งมีหัวใจหลักอยู่การใช้ระบบ Token โดยไม่มีตัวใดตัวหนึ่งเป็นมาสเตอร์ของระบบ



รูปที่ 4.1 แบบจำลองระบบการทำงานของระบบ Token

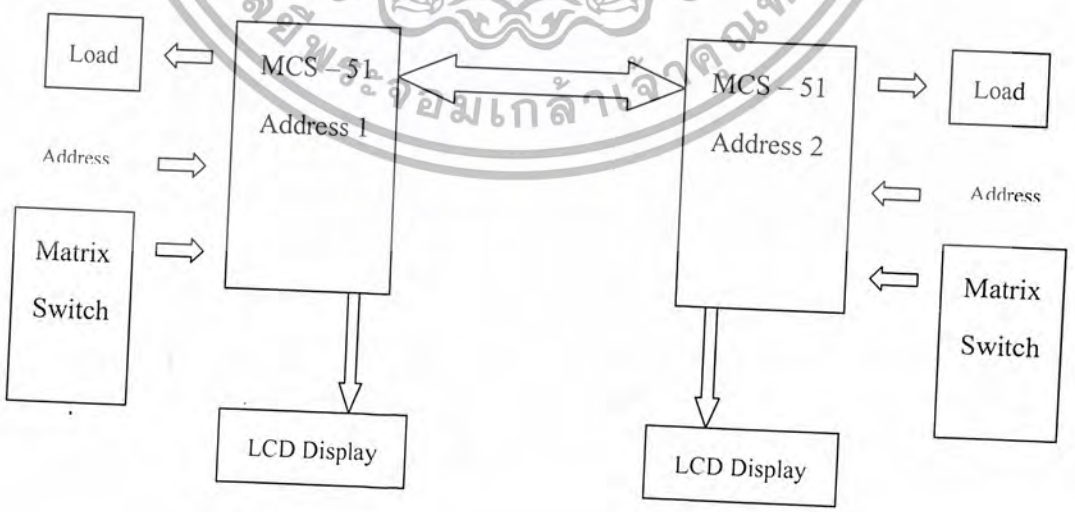
ในแต่ละชุดควบคุมจะถูกกำหนด Address เอาไว้โดยมีมาตรฐานอยู่ที่ไม่เกิน 32 เครื่องต่อระบบ การทำงานจะแบ่งเวลาทำงานกันเป็นช่วงๆ และเสมือนทำงานเป็นรูปวงกลม โดยจะแบ่งเวลาการทำงานกัน สมมติว่าเราใช้เครื่องในระบบทั้งสิ้น 32 เครื่อง โดยเวลาที่เรายอมรับได้ของระบบอย่างแย่ที่สุดกำหนดให้เป็น 1 วินาที ดังนั้น แต่ละเครื่องจะมีค่า Time Shering อยู่ประมาณ 31.25 ms แต่ถ้าเราไม่มีความจำเป็นที่จะต้องใช้เครื่องในระบบหลายตัว อัตราส่วนของค่า Time Shering ก็จะลดลงมา ซึ่งก็ต้องขึ้นอยู่กับการเขียนโปรแกรมของเราด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อดีข้อหนึ่งของการสื่อสารระบบ TOKEN นั้น ก็คือ การแบ่งเวลาให้กับโมดูลต่างๆ ที่อยู่ในระบบให้สามารถทำงานได้ ซึ่งจะทำให้การทำงานไม่ต้องอ้างอิงโมดูลใดโมดูลหนึ่งในการทำงาน ด้วยเหตุนี้เองจึงทำให้ทุกตัวสามารถสั่งงานหรือควบคุมอุปกรณ์ตัวอื่นๆ ได้โดยไม่ต้องรอตัวมาสเตอร์ตัวใดตัวหนึ่ง โดยการทำงานของวงจรก็ตัวคอนโทรลเลอร์ก็จะต้องรู้ว่าตัวเองอยู่ใน Address ที่เท่าไร เพื่อการกำหนดลำดับของการทำงานได้อย่างถูกต้องโดยเริ่มจากการอ่านค่าบิต SW-DIP ที่ผู้ใช้ได้กำหนดก่อนการรีเซตระบบ เมื่อระบบถูกรีเซตขึ้นแล้ว ระบบโดยโปรแกรมจะสั่งให้เกิดการเปลี่ยนการทำงานของแต่ละโมดูล โดยจะไล่ไปตั้งแต่ Address ที่ 1 จนถึง Address สุดท้าย ก่อนที่จะย้อนกลับมาตัวแรกใหม่เมื่อช่วงเวลาของ Address ตัวสุดท้ายสิ้นสุดลง

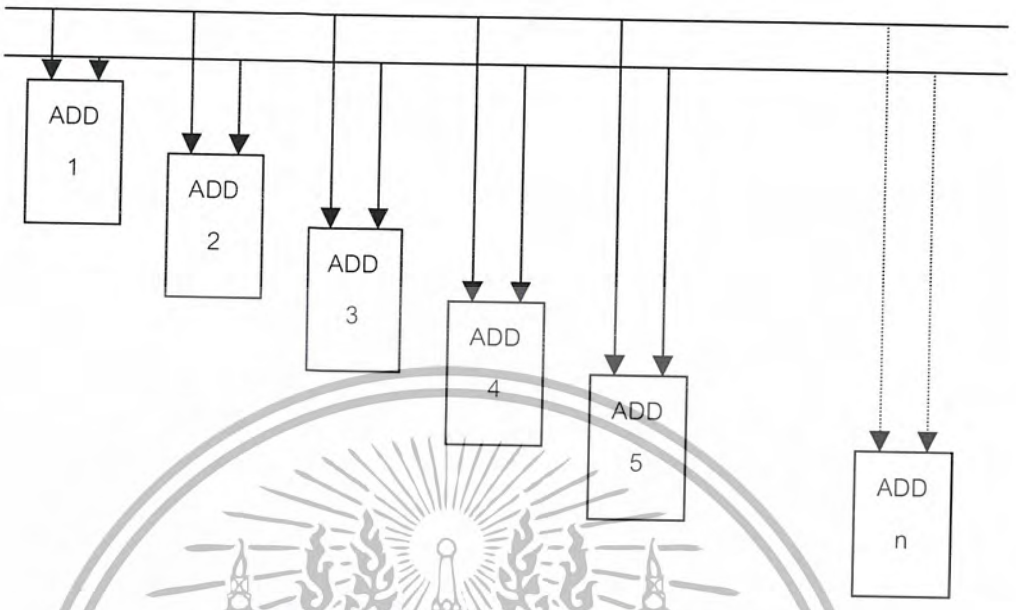
ข้อดีอีกอย่างหนึ่งก็คือการทำงานของระบบเมื่อการการเสียหายขึ้นที่โมดูลใดๆ ก็ตาม หรือแม้กระทั่งโมดูลใดๆ ถูกยกออกจากระบบ ระบบก็ไม่เกิดการรวนและยังสามารถทำงานได้เหมือนเดิม เพราะในระบบ TOKEN เมื่อมีโมดูลใดโมดูลหนึ่งเสียหรือขาดไปก็จะไม่เกิดผลอะไรกับระบบ แค่เพียงรอให้ช่วงเวลาของการทำงานตัวนั้นๆ หมดไป ก็จะเลื่อนไปเซ็ท Address ตัวถัดไปเป็นตัวทำงานใหม่ และวนไปเรื่อยๆ จนกระทั่งตัดแหล่งไฟให้กับระบบ

ในการสร้างครั้งนี้ เรื่องของแหล่งจ่ายไฟฟ้า ผู้สร้างได้สร้างให้ระบบสามารถ ใช้แหล่งจ่ายไฟฟ้าที่โมดูลใดโมดูลหนึ่งก็ได้ ซึ่งที่ไม่จำเป็นต้องใช้แหล่งจ่ายไฟฟ้าในทุกๆโมดูล โดยผู้สร้างออกแบบให้ระบบใช้แหล่งจ่ายไฟฟ้าร่วมกัน โดยผ่านสายเกลียวคู่ ซึ่งเป็นสายสัญญาณที่ใช้ในการสื่อสารของระบบอยู่แล้ว ดังนั้นจึงไม่มีความจำเป็นที่ผู้ใช้งานจะต้องต่อแหล่งจ่ายไฟฟ้ามากกว่า 1 แหล่งจ่าย

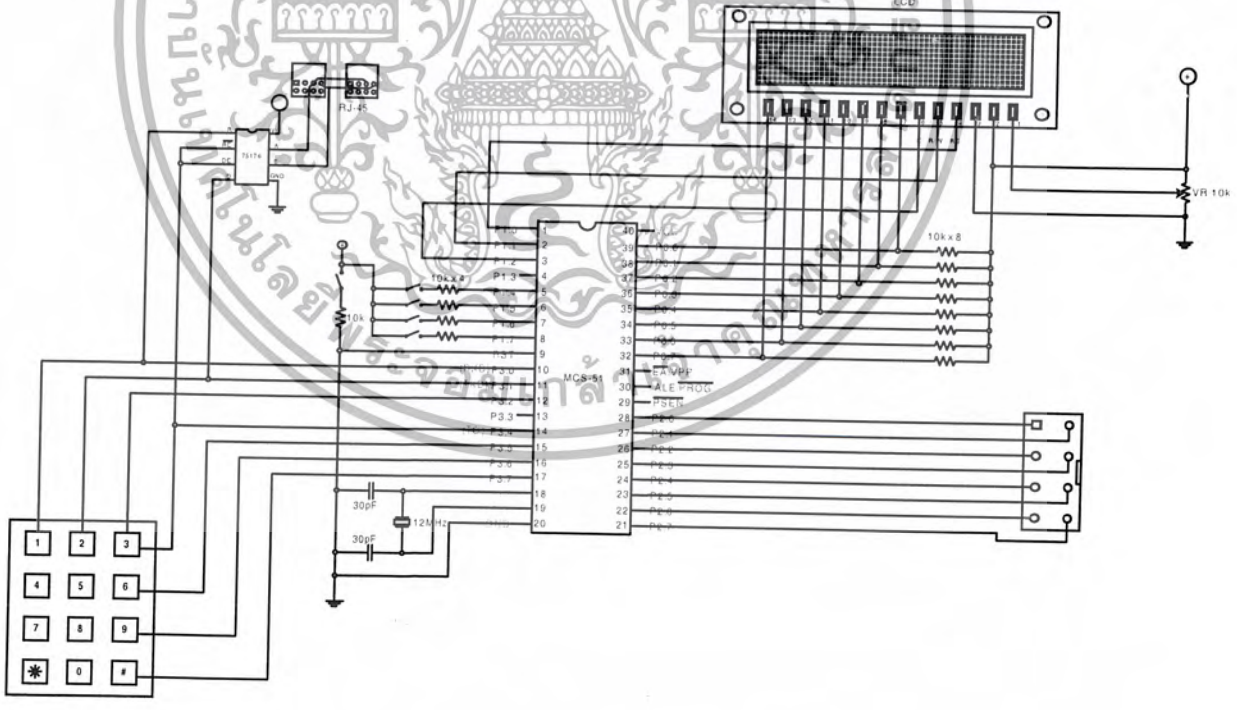


รูปที่ 4.2 ระบบการทำงานติดต่อกันระหว่าง 2 เครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 การสื่อสารในระบบของ TOKEN



รูปที่ 4.4 วงจรการทำงานของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

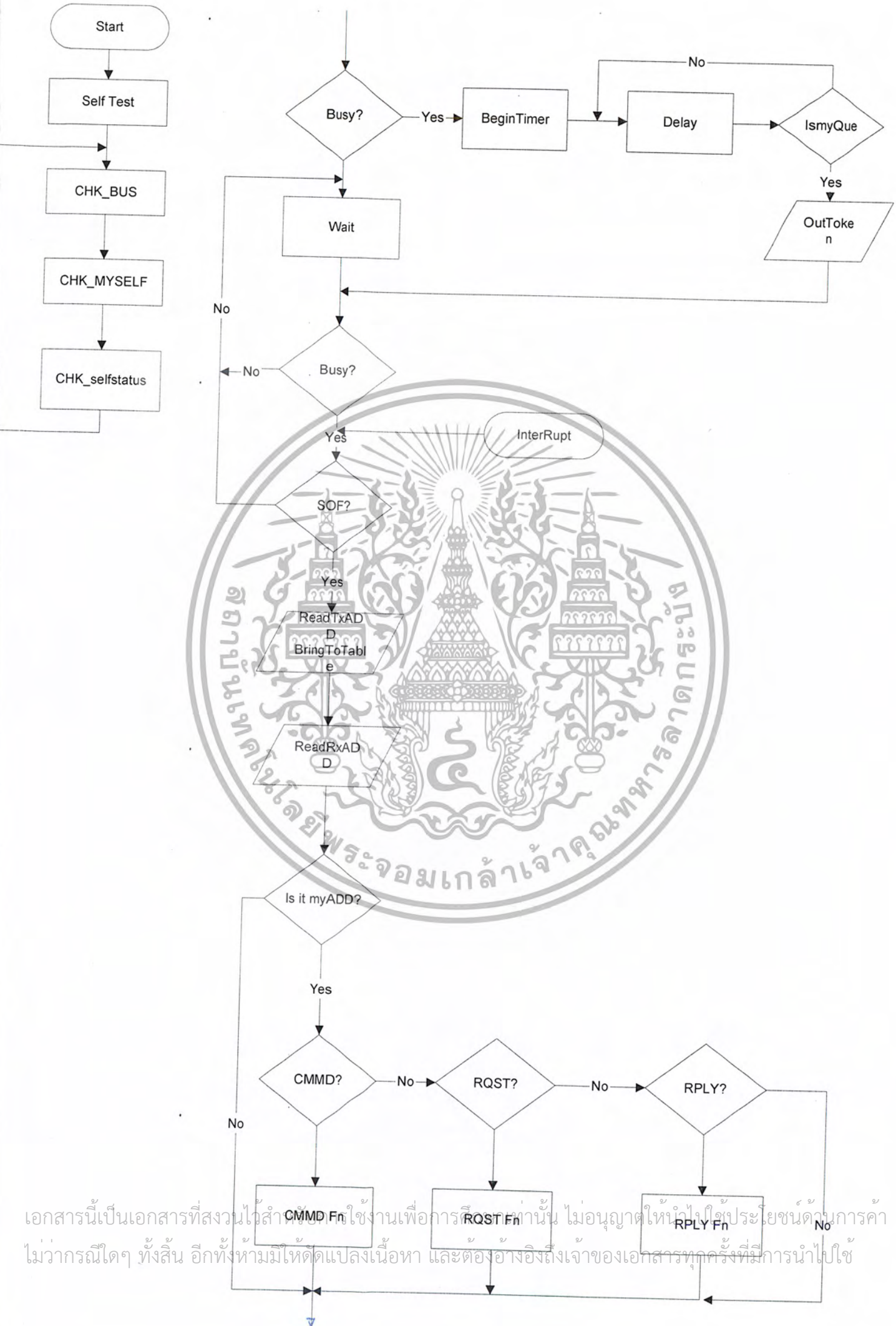
4.2 การสร้าง

การสร้างระบบควบคุมไฟฟ้าอัตโนมัติ 1 โมดูล จะประกอบไปด้วยส่วนประกอบต่างๆ 4 ส่วนด้วยกัน คือ ส่วนโมดูลควบคุม ส่วนแมทริกซ์สวิตซ์ ส่วนจอแสดงผล ส่วนวงจรขับโหลด

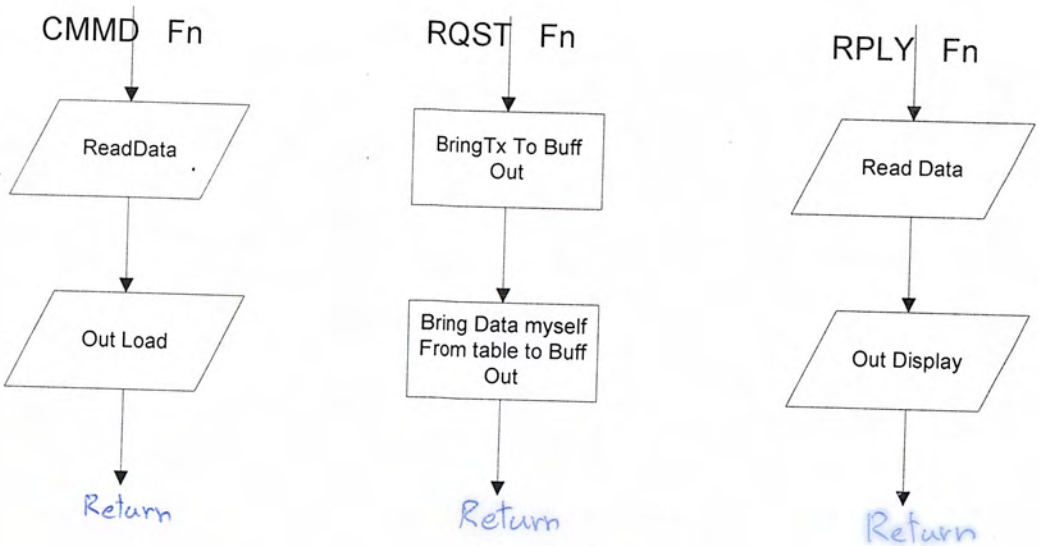
วัสดุและอุปกรณ์

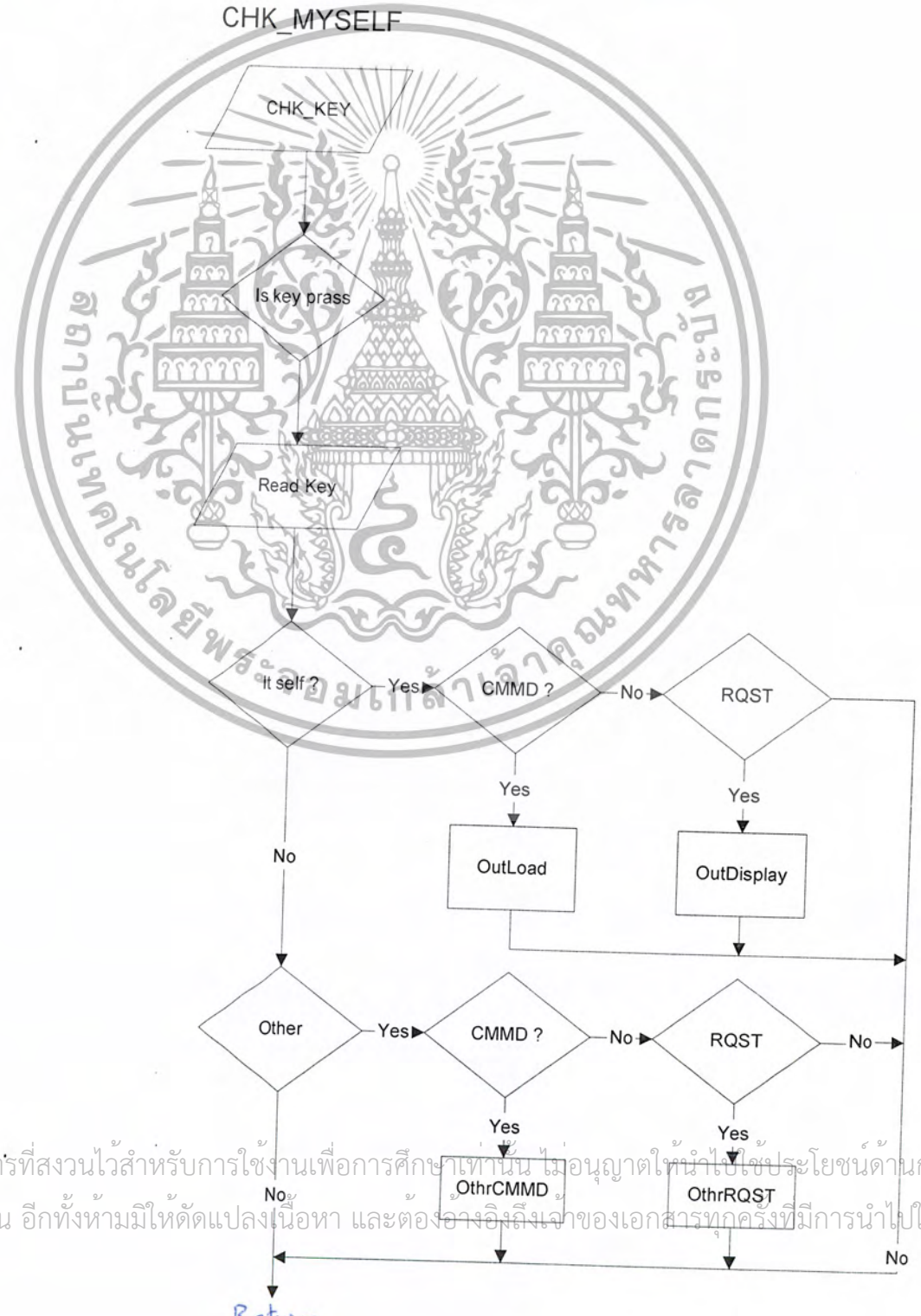
1. ตัวต้านทาน	R1	120 โอห์ม	จำนวน	2	ตัว
2. ตัวต้านทาน	R2	8.2 กิโลโอห์ม	จำนวน	1	ตัว
3. ตัวต้านทานปรับค่า	VR1	5 กิโลโอห์ม	จำนวน	1	ตัว
4. ตัวต้านทาน(R PAK)	CON9	4.7 กิโลโอห์ม	จำนวน	1	ตัว
5. ตัวเก็บประจุ	C1, C2	100 ไมโครฟารัด	จำนวน	2	ตัว
6. ตัวเก็บประจุ	C3	10 ไมโครฟารัด	จำนวน	1	ตัว
7. ตัวเก็บประจุ	C4, C5	33 พิกโกฟารัด	จำนวน	2	ตัว
8. ตัวเก็บประจุ	C6, C7	0.1 ไมโครฟารัด	จำนวน	2	ตัว
9. ไดโอด	D1	1N4148	จำนวน	1	ตัว
10. ไดโอด	D2-D5	1N4002	จำนวน	4	ตัว
11. ไดโอด	D6-D9	1N4148	จำนวน	4	ตัว
12. ไดโอดเปล่งแสง	LED1		จำนวน	1	ตัว
13. IC 7805	V1		จำนวน	1	ตัว
14. X-Tal	Y1	12 MHz	จำนวน	1	ตัว
15. ดิปสวิตซ์	S1	SW-DIP8	จำนวน	1	ตัว
16. สวิตซ์กดติดปล่อยดับ	S2		จำนวน	1	ตัว
17. แมทริกซ์สวิตซ์	12 คีย์		จำนวน	1	ตัว
18. LCD	16 ตัวอักษร 2 แถว		จำนวน	1	ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

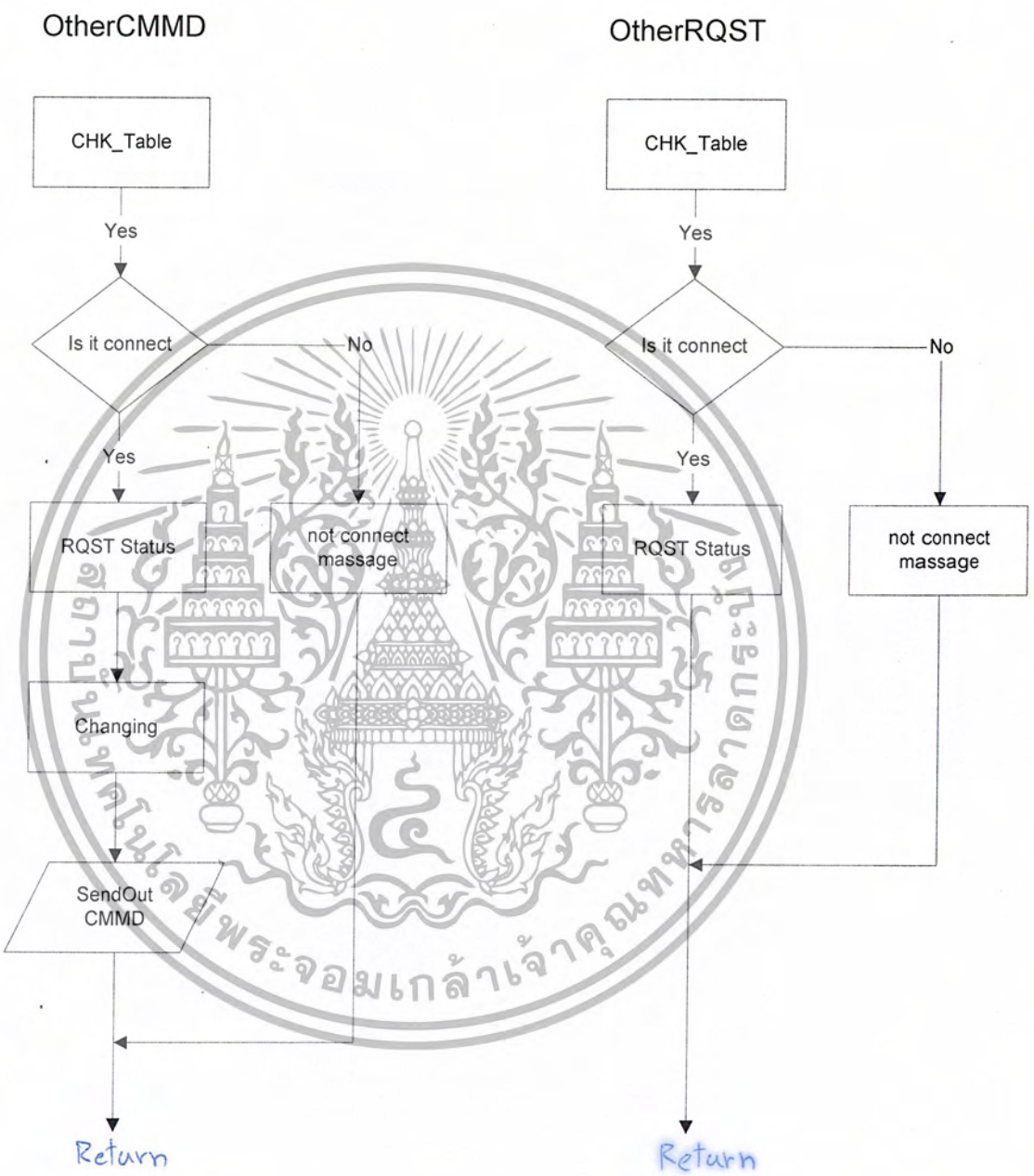


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

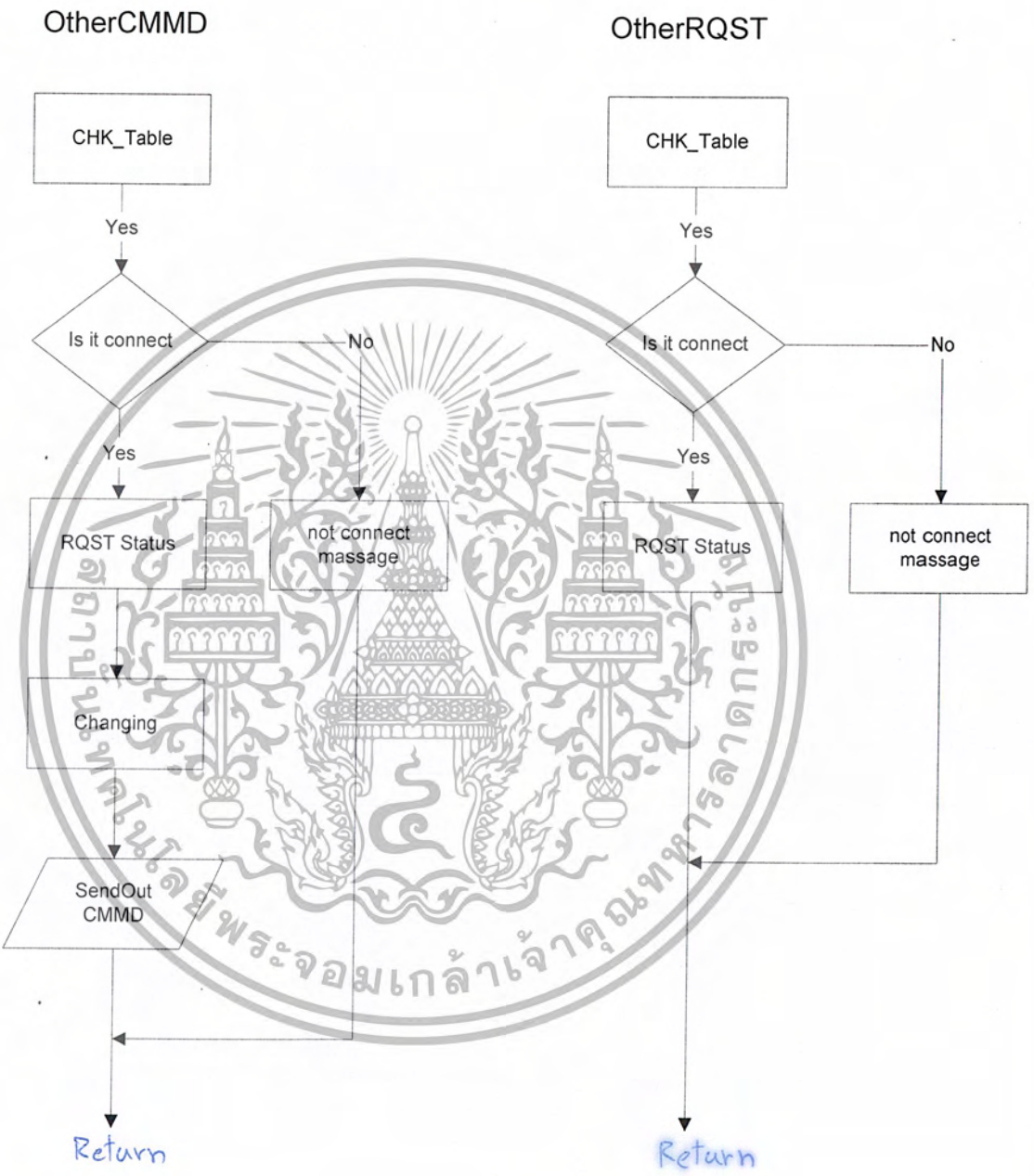




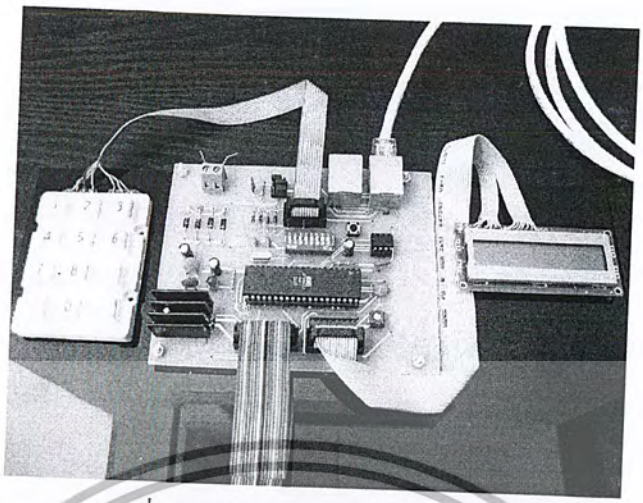
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



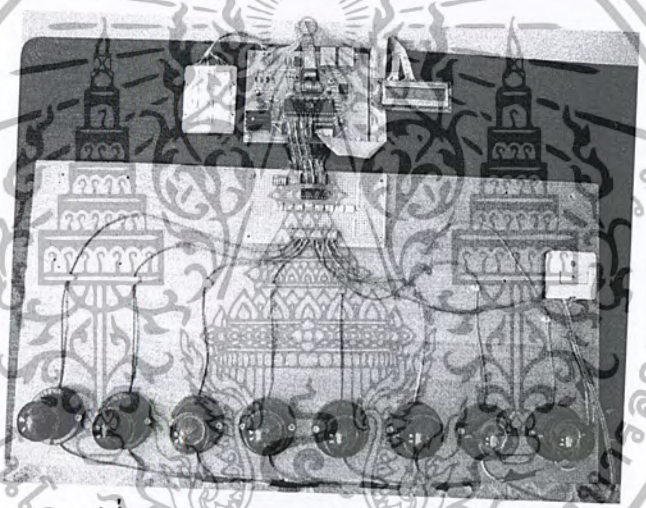
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



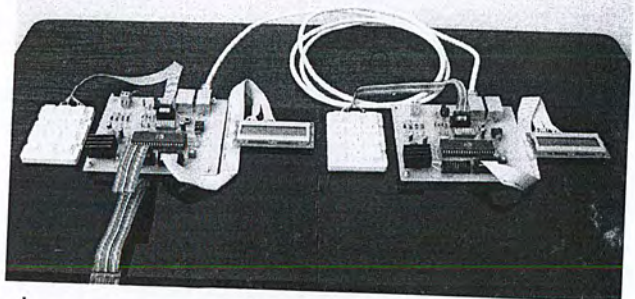
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 วงจรและอุปกรณ์ของระบบ



รูปที่ 4.6 ภาพแสดงการต่อโหลด 8 เอาท์พุท



รูปที่ 4.7 ภาพแสดงการเชื่อมต่อโมดูลด้วยสาย twisted pair

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การใช้งาน

ระบบควบคุมไฟฟ้าอัตโนมัติประกอบไปด้วย

1. ชุดโมดูลควบคุม
2. ชุดวงจรขับโหลด 8 พอร์ต
3. ชุดคีย์แมทริกซ์สวิตซ์
4. จอแสดงผลแบบ LCD 16 ตัวอักษร 2 บรรทัด

การกำหนดรูปแบบคำสั่งการทำงาน ผู้สร้างได้กำหนดให้ผู้ใช้งานสั่งงานครั้งเดียวเมื่อต้องการที่ติดต่อควบคุมอุปกรณ์ไฟฟ้าของ Address อื่นๆ โดยได้กำหนดรูปแบบของการสั่งงานดังนี้

1 2 3 ..... 19 20

- 2 ตัวแรกเป็นการกำหนด Address ที่ต้องการจะติดต่อหรือควบคุม
- ตัวที่ 3 เป็นการกำหนดสถานะของคำสั่งโดยจะใช้ 2 ปุ่ม บนคีย์แมทริกซ์สวิตซ์ คือ " * " (ON) และ " 0 " (OFF)
- ตัวที่ 4 บังคับกดเลือกหมายเลข " 1 " หมายถึง อุปกรณ์ตัวที่ 1
- ตัวที่ 5 เป็นการกำหนดการเลือกเปิดหรือปิดอุปกรณ์ โดยที่ " * " (ON) หมายถึงต้องการเปิดอุปกรณ์ตัวที่ 1 และ " 0 " (OFF) หมายถึงต้องการปิดอุปกรณ์ตัวที่ 1
- ตัวที่ 6 บังคับกดเลือกหมายเลข " 2 " หมายถึง อุปกรณ์ตัวที่ 2
- ตัวที่ 7 เป็นการกำหนดการเลือกเปิดหรือปิดอุปกรณ์ โดยที่ " * " (ON) หมายถึงต้องการเปิดอุปกรณ์ตัวที่ 2 และ " 0 " (OFF) หมายถึงต้องการปิดอุปกรณ์ตัวที่ 2
- ตัวที่ 8 บังคับกดเลือกหมายเลข " 3 " หมายถึง อุปกรณ์ตัวที่ 3
- ตัวที่ 9 เป็นการกำหนดการเลือกเปิดหรือปิดอุปกรณ์ โดยที่ " * " (ON) หมายถึงต้องการเปิดอุปกรณ์ตัวที่ 3 และ " 0 " (OFF) หมายถึงต้องการปิดอุปกรณ์ตัวที่ 3
- ตัวที่ 10 บังคับกดเลือกหมายเลข " 4 " หมายถึง อุปกรณ์ตัวที่ 4
- ตัวที่ 11 เป็นการกำหนดการเลือกเปิดหรือปิดอุปกรณ์ โดยที่ " * " (ON) หมายถึงต้องการเปิดอุปกรณ์ตัวที่ 4 และ " 0 " (OFF) หมายถึงต้องการปิดอุปกรณ์ตัวที่ 4
- ตัวที่ 12 บังคับกดเลือกหมายเลข " 5 " หมายถึง อุปกรณ์ตัวที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

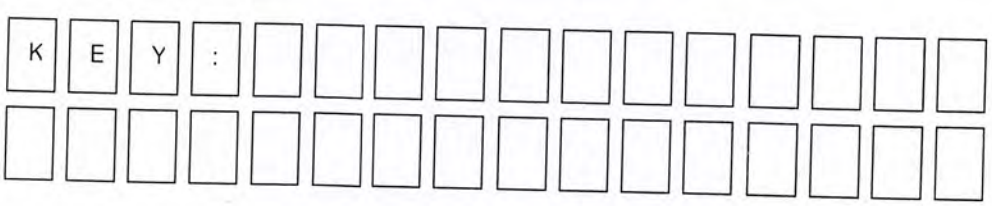
- ตัวที่ 13 เป็นการกำหนดการเลือกเปิดหรือปิดอุปกรณ์ โดยที่ " * " (ON) หมายถึง ต้องการเปิดอุปกรณ์ตัวที่ 5 และ " 0 " (OFF) หมายถึงต้องการปิดอุปกรณ์ตัวที่ 5
- ตัวที่ 14 บังคับกดเลือกหมายเลข " 6 " หมายถึง อุปกรณ์ตัวที่ 6
- ตัวที่ 15 เป็นการกำหนดการเลือกเปิดหรือปิดอุปกรณ์ โดยที่ " * " (ON) หมายถึง ต้องการเปิดอุปกรณ์ตัวที่ 6 และ " 0 " (OFF) หมายถึงต้องการปิดอุปกรณ์ตัวที่ 6
- ตัวที่ 16 บังคับกดเลือกหมายเลข " 7 " หมายถึง อุปกรณ์ตัวที่ 7
- ตัวที่ 17 เป็นการกำหนดการเลือกเปิดหรือปิดอุปกรณ์ โดยที่ " * " (ON) หมายถึง ต้องการเปิดอุปกรณ์ตัวที่ 7 และ " 0 " (OFF) หมายถึงต้องการปิดอุปกรณ์ตัวที่ 1
- ตัวที่ 18 บังคับกดเลือกหมายเลข " 8 " หมายถึง อุปกรณ์ตัวที่ 7
- ตัวที่ 19 เป็นการกำหนดการเลือกเปิดหรือปิดอุปกรณ์ โดยที่ " * " (ON) หมายถึง ต้องการเปิดอุปกรณ์ตัวที่ 8 และ " 0 " (OFF) หมายถึงต้องการปิดอุปกรณ์ตัวที่ 8
- ตัวที่ 20 เป็นการกำหนดการยืนยันคำสั่ง " # " (Ent)

ในการเริ่มต้นการใช้งาน ผู้ใช้จะต้องกำหนดค่า Address ของโมดูล แต่ละโมดูลให้มีค่าตามที่ผู้ใช้ต้องการ โดยไปสวิตซ์ที่ SW-DIP ซึ่งควรจะเริ่มต้นที่ Address 01 เป็นต้นไป

ในการสร้างระบบเครื่องควบคุมอุปกรณ์ไฟฟ้าอัตโนมัติในครั้งนี้ ผู้สร้างได้สร้างวงจต้นจรแบบเบื้องต้นเพียงแค่ 3 ตัว เพราะสามารถที่จะทดลองระบบทั้งหมดได้โดยไม่จำเป็นต้องมีจำนวนวงจรทั้งระบบ โดยอาศัยจุดเด่นของระบบการสื่อสารแบบ TOKEN ที่ทำให้แต่ละโมดูลสามารถทำงานได้เองโดยไม่ต้องอาศัยตัวแม่ตัวใดตัวหนึ่งเพราะระบบได้แบ่งเวลาให้ทุกตัวเป็นตัวมาสเตอร์หรือตัวแม่สลับกันทำงานอยู่แล้ว

กรณีตรวจสอบสถานะของอุปกรณ์ (Address อื่นๆ)

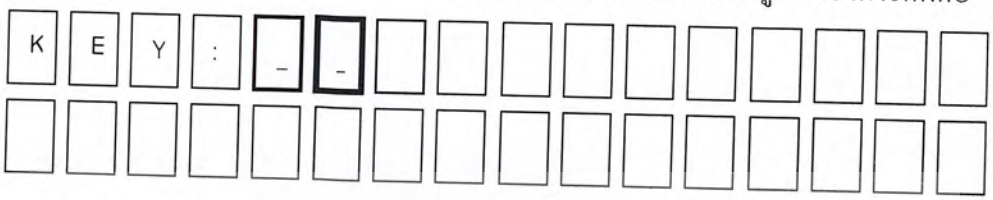
1. เมื่อต่อไฟเข้าวงจรระบบจะพร้อมรับคำสั่ง โดยจะปรากฏคำว่า " KEY : " ที่จอแสดงผล



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

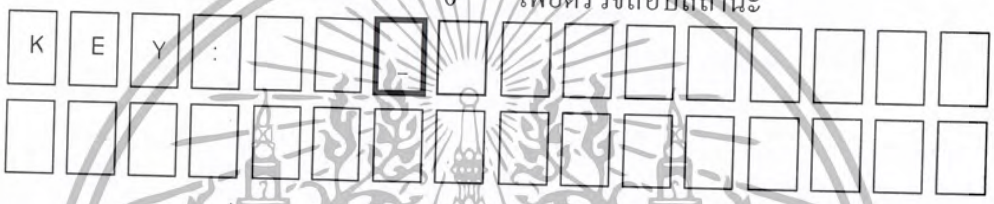
2. ให้ผู้ใช้กดคีย์ชุดคำสั่ง ดังนี้

2.1 " 01 - 32 " เพื่อกำหนด Address ของโมลดูลที่ต้องการติดต่อ



2.2 " 0 " เพื่อกำหนดการควบคุม (ตรวจสอบสถานะ)

" 0 " เพื่อตรวจสอบสถานะ



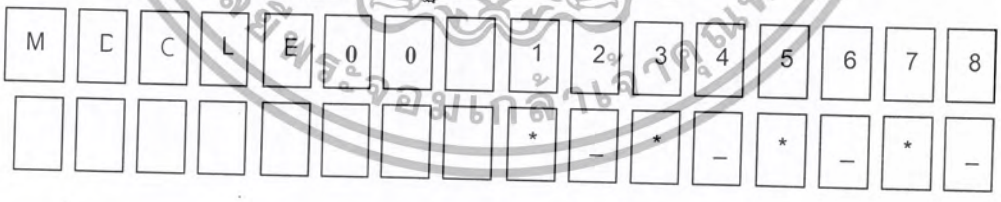
2.3 " # " (Ent) เพื่อยืนยันคำสั่ง

กรณีต้องการตรวจสอบสถานะของตัวเอง

กด " 000 " ตามด้วย " # " (Ent)

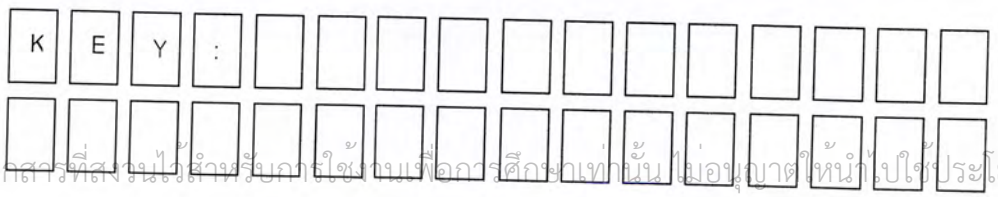
<u>ตัวอย่าง</u> ถ้าสมมติให้อุปกรณ์ตัวที่ 1, 3, 5, 7 เปิดอยู่ และ 2, 4, 6, 8 เป็นสถานะ ปิด

หน้าจอแสดงผลจะปรากฏดังนี้



กรณีต้องการสั่งงาน (เปิด-ปิดอุปกรณ์)

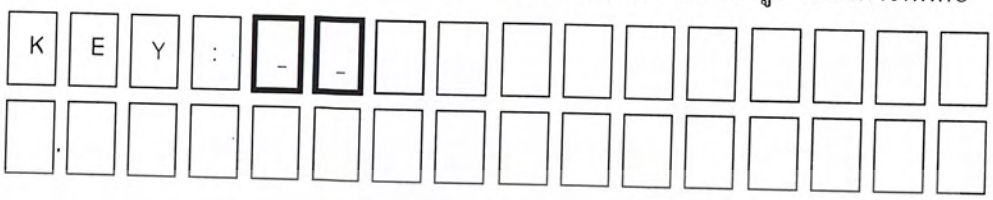
1. เมื่อต่อไฟเข้าวงจรระบบจะพร้อมรับคำสั่ง โดยจะปรากฏคำว่า " KEY : " ที่จอแสดงผล



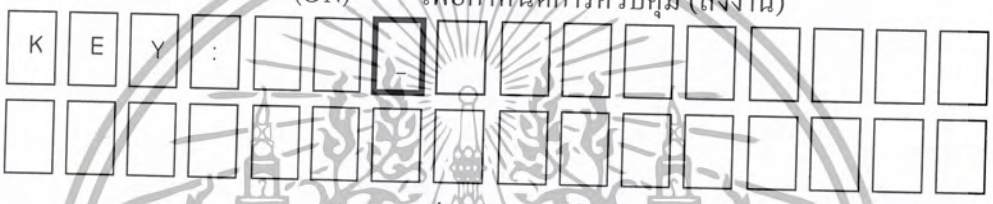
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ให้ผู้ใช้กดคีย์ชุดคำสั่ง ดังนี้

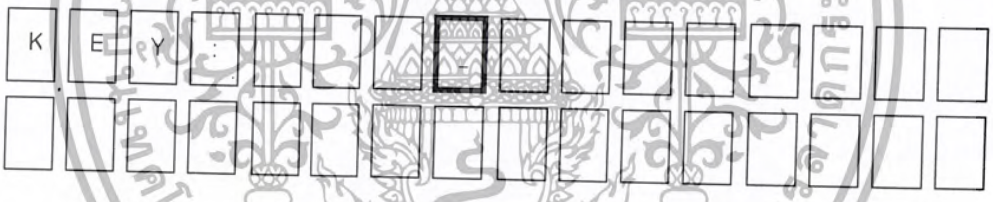
2.1 “01 - 32” เพื่อกำหนด Address ของโมดูลที่ต้องการติดต่อ



2.2 “ * ” (ON) เพื่อกำหนดการควบคุม (สั่งงาน)



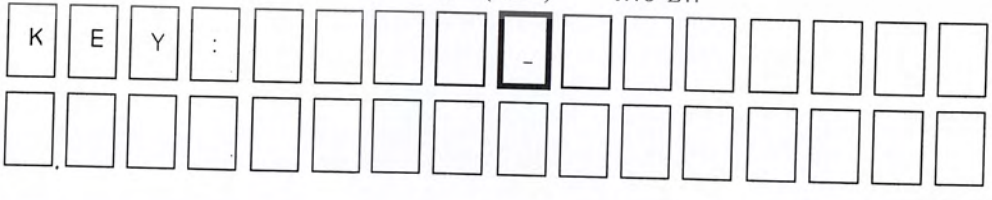
2.3 “1” เพื่อกำหนดโหลดตัวแรกที่ต้องการควบคุม
(เลือกได้ทีละตัว)



2.4 “ * ”, “ 0 ” เพื่อกำหนดคำสั่ง (เปิด หรือ ปิด)

“ * ” (ON) เพื่อ เปิด

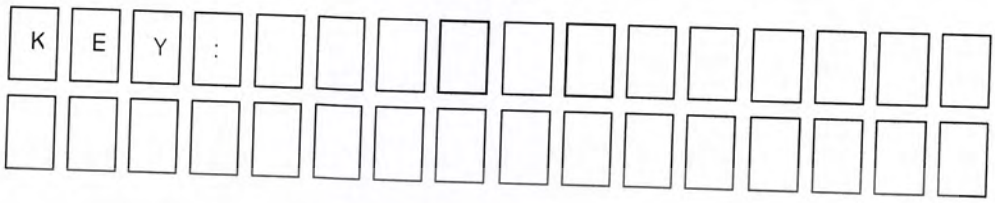
“ 0 ” (OFF) เพื่อ ปิด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

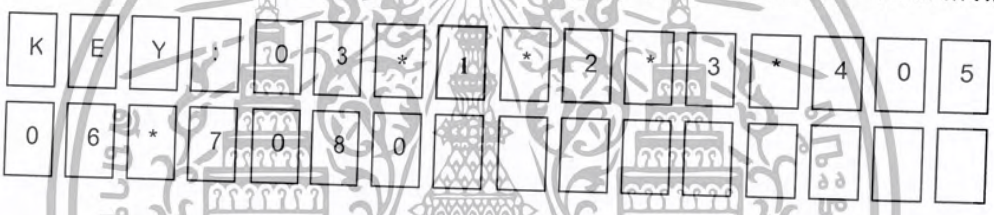
2.5 ย้อนกลับไปทำข้อ 2.4 และ 2.5 กด หมายเลขของเอาท์พุททีละตัวแล้วตามด้วย " * " (ON) หรือ " 0 " (ON) โดยกดไล่ไปจนถึงเอาท์พุทสุดท้าย (มี 8 เอาท์พุท)

2.6 " # " (Ent) เพื่อยืนยันคำสั่ง



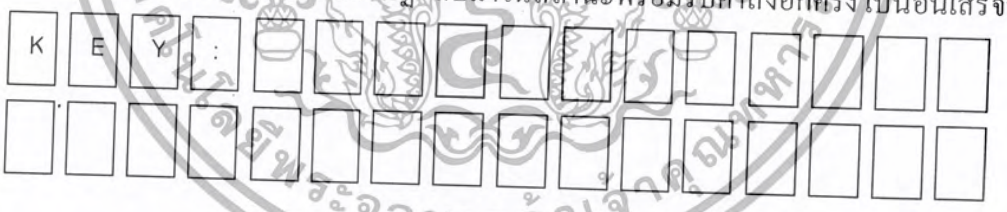
3. เมื่อผู้ใช้ คีย์คำสั่งผิด สามารถเคลียร์คำสั่งใหม่ได้โดยกดปุ่ม " 9 " (Cls)

<u>ตัวอย่าง</u> ต้องการสั่งให้ เอาท์พุทตัวที่ 1, 2, 3, 6 เปิด ที่Address 03 สามารถคีย์คำสั่งดังนี้

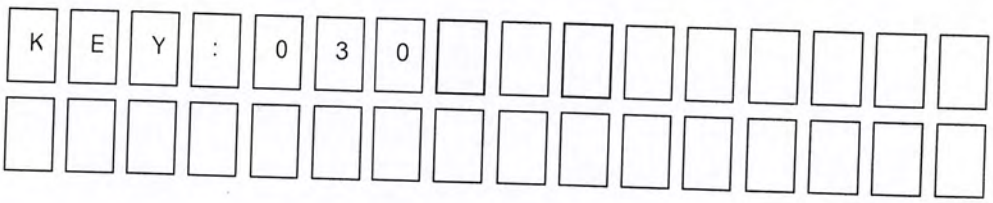


ตามด้วยคีย์ " # " (Ent)

หน้าจอแสดงผลก็ปรากฏกลับมาในสถานะพร้อมรับคำสั่งอีกครั้ง เป็นอันเสร็จสิ้น



<u>ตัวอย่าง</u> จากตัวอย่างที่แล้ว ต้องการเช็คสถานะของเอาท์พุท ที่ Address 03 สามารถคีย์คำสั่งได้ดังนี้



ตามด้วยคีย์ " # " (Ent)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าจอแสดงผลก็จะปรากฏดังนี้

M	D	C	L	E	0	3		1	2	3	4	5	6	7	8
								*	*	*	_	_	*	_	_

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุป

จากการทดลองสร้างชุดควบคุมระบบไฟฟ้าอัตโนมัติขึ้นมาจำนวน 3 ชุด ซึ่งแต่ละชุดประกอบด้วย

- ไมโครคอนโทรลเลอร์ MCS-51
- คีย์แพด
- LCD Display
- IC แปลงสัญญาณ RS-232 เป็น RS-485
- ชุดวงจรขับโหลด
- แหล่งกำเนิดไฟตรง 5 โวลท์

จากการสร้างวงจรระบบควบคุมไฟฟ้าอัตโนมัติในแบบจำลองทั้ง 3 ชุดนั้น ผลการสร้างได้ออกมาเป็นที่น่าพอใจ การทำงานของระบบออกมาได้ตรงตามต้องการ ทั้งการควบคุมสั่งการการเปิดปิดอุปกรณ์ การตรวจสอบสถานการทำงานของอุปกรณ์ทั้งที่อยู่ใน Address เดียวกัน และ Address อื่นๆ การแสดงผลของเอาท์พุท รวมถึงการแสดงผลของหน้าจอ LCD Display

การใช้แหล่งจ่ายไฟร่วมกันของระบบ โดยอาศัยวงจรเพียงตัวใดตัวหนึ่งของระบบต่อเข้ากับแหล่งจ่ายไฟฟ้า ก็ไม่ได้สร้างปัญหาให้กับระบบแต่อย่างใด ตรงนี้ก็จะยิ่งเพิ่มความสะดวกให้กับผู้ใช้งานได้อีกมาก เนื่องจากไม่ต้องกังวลและคอยยุ่งยากกับการต่อแหล่งจ่ายในทุกๆ จุดของระบบ แต่ถ้าระบบมีระยะทางในการเชื่อมต่อมากๆ ก็อาจทำให้เกิดปัญหาได้ ซึ่งก็คงต้องทดลองและพัฒนาต่อไป

ผู้สร้างได้ออกแบบวงจร และระบบ ทั้งทางด้านในส่วนที่เป็น ฮาร์ดแวร์หรือซอฟท์แวร์ ให้ง่ายต่อการใช้งานของผู้ใช้งานในระดับหนึ่งเท่านั้น จะเห็นได้ว่าการสั่งงานควบคุมโมดูลต่างๆ ยังคงใช้คำสั่งที่มีความยาวและค่อนข้างยุ่งยากอยู่สักเล็กน้อย เนื่องจากความจำกัดของคีย์ที่ใช้เป็นสวิตซ์ มีจำนวนคีย์ที่จำกัดอยู่ ทั้งด้านการออกแบบที่ยังขาดความซับซ้อนทางด้านโปรแกรมที่จะช่วยทำให้ผู้ใช้สามารถใช้งานได้ง่ายยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. ชัยวัฒน์ ลิ้มพรจิตวิไล, วรพจน์ กรแก้ววัฒนกุล , "โครงสร้างและสถาปัตยกรรมของไมโครคอนโทรลเลอร์MCS–51", เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์MCS-51,หน้า 7-27
2. ธีรวัฒน์ ประกอบผล "การประยุกต์ใช้งานไมโครคอนโทรลเลอร์"
3. สมยศ จุณณะปิยะ, " การใช้งานไมโครคอนโทรลเลอร์ ตระกูล MCS-51 " ,สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, พิมพ์ครั้งที่ 1, 2537,
4. ปรเมษฐ์ ประนยานันทน์ ปิยพงศ์ เผ่าวะณิช "คู่มือและการใช้งานไมโครคอนโทรลเลอร์ MCS-51"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

โปรแกรมการทำงานของเครื่องควบคุมไฟอัตโนมัติ

```
;****************************************
;     Define port&pin Name
;****************************************
KPAD_ROW0           EQU   P1.0
KPAD_ROW1           EQU   P1.1
KPAD_ROW2           EQU   P1.2
KPAD_ROW3           EQU   P1.3
KPAD_COL2     EQU   P1.4
KPAD_COL1     EQU   P1.5
KPAD_COL0     EQU   P1.6

;****************************************
;     Define LED STATUS
;****************************************
READY         EQU   P0.0
SELECT_ROOM         EQU   P0.1
SELECT_DEVICE       EQU   P0.2
STATUS_1      EQU   P0.3
STATUS_2      EQU   P0.4
STATUS_3      EQU   P0.5

;****************************************
;     Define DEVICE OUTPUT
;****************************************
DEVICE_1      EQU   P2.3
DEVICE_2      EQU   P2.4
DEVICE_3      EQU   P2.5

;****************************************
;     Define  User Register
;****************************************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
KPAD_DATA     EQU   032H

;**************************************
;     MAIN PROGRAM
;**************************************
              ORG   0000H
              AJMP  MAIN

              ORG   0023H
              AJMP  RECI

              ORG   0030H
MAIN:      MOV   TMOD,#020H
              MOV   TH1,#0FDH
              MOV   TL1,#0FDH
              SETB  TR1
              MOV   IE,#10010000B      ;HAVE INTERRAF
              SETB  REN                ;HAVE RECIVE DATA
              MOV   SCON,#01010000

              CLR   READY
              CLR   SELECT_ROOM

              ACALL GET_KPAD
              MOV   A,KPAD_DATA
              AJMP  FUNCTION_1
              AJMP  STOP

;******************************************
;     FUNCTION 1  FOR SELECT ADDRESS
;******************************************

FUNCTION_1:   CJNE  A,#01,CHK_ADD2
              AJMP  ADD_1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CHK_ADD2:    CJNE   A,#02,MAIN
             ACALL  ADD_2

ADD_1:       MOV    P0,P2
             CLR    SELECT_DEVICE
             ACALL  GET_KPAD
        MOV  A,KPAD_DATA
             AJMP   FUNCTION_21

ADD_2:       ANL    A,#00000000B
             ACALL  SEND_DATA
             ACALL  GET_KPAD
             MOV    A,KPAD_DATA
             AJMP   FUNCTION_22


;*****************************************
;    FUNCTION 2.1 FOR SELECT DEVICE IN
;*****************************************
FUNCTION_21:  CJNE   A,#01,CHK_DEVICE_212
              CPL    DEVICE_1
              CPL    STATUS_1
              AJMP   ADD_1

CHK_DEVICE_212:  CJNE  A,#02,CHK_DEVICE_213
              CPL    DEVICE_2
              CPL    STATUS_2
        AJMP   ADD_1

CHK_DEVICE_213:  CJNE  A,#03,CANCLE_21
              CPL    DEVICE_3
              CPL    STATUS_3
        AJMP   ADD_1

CANCLE_21:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                AJMP   ADD_1

;*****************************************
;     FUNCTION 2.2  FOR SELECT DEVICE 0UT
;*****************************************
FUNCTION_22:  CJNE   A,#01,CHK_DEVICE_222
              ORL    A,#00010000B
              CPL    STATUS_1
              ACALL  SEND_DATA
              AJMP   ADD_2

CHK_DEVICE_222:  CJNE   A,#02,CHK_DEVICE_223
              ORL    A,#00010000B
              CPL    STATUS_2
              ACALL  SEND_DATA
              AJMP   ADD_2

CHK_DEVICE_223:  CJNE   A,#03,CANCLE_22
              ORL    A,#00010000B
              CPL    STATUS_3
              ACALL  SEND_DATA
              AJMP   ADD_1

CANCLE_22:    AJMP   FUNCTI0N_1

;*****************************************
;     INTERRUP RECIEVE DATA
;*****************************************
RECI:         PUSH   A
        SETB  SM1        ;SCON MODE 1
              CLR    RI
              MOV    A,SBUF
              MOV    R7,A

              ANL    A,#00110000B
              CJNE   A,#00000000B,CHK_DEMAND
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV    A,P2
                ORL    A,#00100000B
                AJMP   SEND_STATUS

CHK_DEMAND:   CJNE   A,#00010000B,ANS_STATUS

                MOV    A,R7
                CJNE   A,#01,CHK_2
                CPL    DEVICE_1
                CPL    STATUS_1
                AJMP   END_INTERRUP

CHK_2:        CJNE   A,#02,CHK_3
                CPL    DEVICE_1
                CPL    STATUS_1
                AJMP   END_INTERRUP
CHK_3:        CJNE   A,#03,END_INTERRUP
                CPL    DEVICE_1
                CPL    STATUS_1
                AJMP   END_INTERRUP

SEND_STATUS:  ACALL  SEND_DATA
                AJMP   END_INTERRUP

ANS_STATUS:   MOV    P0,R7

END_INTERRUP: POP    A
                RETI

;****************************************
;Keypad Scan key Subroutine
;****************************************
GET_KPAD:     MOV    P1,#0FFH
                MOV    KPAD_DATA,#0
CHK_COL0:     CLR    KPAD_COL0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV    KPAD_DATA,A
                RET

CHK_ROW2:    JB     KPAD_ROW2,CHK_ROW3
                MOV    A,KPAD_DATA
                ADD    A,#6
                MOV    KPAD_DATA,A
                RET

CHK_ROW3:    MOV    A,KPAD_DATA
                ADD    A,#9
                MOV    KPAD_DATA,A
                RET

;*****************************************
;            SEND DATA
;*****************************************

SEND_DATA:    CLR    TI
                MOV    SBUF,A
                JNB    TI,$
                RET


STOP:         END.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV   A,P1
                ANL   A,#00FH
                CJNE  A,#00FH,COL0_DETECT
                AJMP  CHK_COL1

COL0_DETECT:    MOV   KPAD_DATA,#01
                AJMP  GET_ROW

CHK_COL1:       SETB  KPAD_COL0
                CLR   KPAD_COL1
                MOV   A,P1
                ANL   A,#00FH
                CJNE  A,#00FH,COL1_DETECT
                AJMP  CHK_COL2

COL1_DETECT:    MOV   KPAD_DATA,#02
                AJMP  GET_ROW

CHK_COL2:       SETB  KPAD_COL1
                CLR   KPAD_COL2
                MOV   A,P1
                ANL   A,#00FH
                CJNE  A,#00FH,COL2_DETECT
                RET

COL2_DETECT:    MOV   KPAD_DATA,#03

GET_ROW:        CLR   KPAD_COL0
                CLR   KPAD_COL1
                CLR   KPAD_COL2
                JB    KPAD_ROW0,CHK_ROW1
                RET

CHK_ROW1:       JB    KPAD_ROW1,CHK_ROW2
                MOV   A,KPAD_DATA
                ADD   A,#3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้