สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ชุดควบคุมตำแหน่งเครื่องกัดซีเอ็นซี ด้วยไมโครคอนโทรลเลอร์

นาย ธีรวัฒน์ ธานีรัตน์
นาย สุธี ปิงยอม
นาย สุมิตร วงษ์ไชย
นาย อนันตศักดิ์ อินทวงค์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

วน.
ส 647ช
2547

เลขหมู่.....
เลขทะเบียน 61812
วัน,เดือน,ปี 21 ก.ค. 2549

b 11603171
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CNC POSITION CONTROLLER

WITH MICROCONTROLLER

MR. THEERAWAT HANEERAT

MR. SUTEE PINGYOM

MR. SUMITH VONGCHAI

MR. ANUNTASAK INTAWONG

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN INDUSTRIAL ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2004

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

หัวข้อปริญญานิพนธ์ ชุดควบคุมตำแหน่งเครื่องกัดซีเอ็นซีด้วยไมโครคอนโทรลเลอร์

CNC POSITION CONTROLLER WITH MICROCONTROLLER

นักศึกษา

นาย ธีรวัฒน์	ธานีรัตน์	รหัสประจำตัว	45015898
นาย สุธี	ปิงยอม	รหัสประจำตัว	45015918
นาย สุมิตร	วงษ์ไชย	รหัสประจำตัว	45015919
นาย อนันตศักดิ์	อินทวงค์	รหัสประจำตัว	45015923

หลักสูตร วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมอุตสาหการ

อาจารย์ผู้ควบคุมปริญญานิพนธ์

(รศ.พรศักดิ์ อรรถวานิช)

(อาจารย์พลชัย โชติปรายนกุล)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	ชุดควบคุมตำแหน่งเครื่องกัดซีเอ็นซีด้วยไมโครคอนโทรลเลอร์
นักศึกษา	นาย ธีรวัฒน์ ธานีรัตน์
	นาย สุธี ปิงยอม
	นาย สุมิตร วงษ์ไชย
	นาย อนันตศักดิ์ อินทวงค์
หลักสูตร	วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมอุตสาหการ
	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา	2547
อาจารย์ผู้ควบคุมปริญญานิพนธ์	รศ.พรศักดิ์ อรรถวานิช
	อาจารย์พลชัย โชติปรายนกุล

บทคัดย่อ

วัตถุประสงค์ของปริญญานิพนธ์ฉบับนี้ เป็นการศึกษาการสร้างชุดควบคุมของเครื่องกัดซีเอ็นซี 3 แกน เพื่อให้สามารถเคลื่อนที่ได้ตามความต้องการด้วยคำสั่งจีโค้ด (G-Code) โดยการสั่งงานด้วยไมโครคอมพิวเตอร์ผ่านทางพอร์ตอนุกรม (RS-232C) ไปยังไมโครคอนโทรลเลอร์ MCS-51 เพื่อควบคุมการทำงานของมอเตอร์ ผ่านชุดควบคุมการขับมอเตอร์ และการนับพัลซ์สัญญาณจากเอ็นโคเดอร์ ส่งกลับไปยังไมโครคอนโทรลเลอร์ MCS-51 และไมโครคอมพิวเตอร์เพื่อประมวลผลและรับคำสั่งการทำงานใหม่อีกครั้งหนึ่ง โดยใช้โปรแกรมแอสเซมบลี ควบคุมการทำงานในไมโครคอนโทรลเลอร์ MCS-51 และโปรแกรมวิชวลเบสิกควบคุมการทำงานของไมโครคอมพิวเตอร์ ดังนั้นประโยชน์ของการศึกษาในครั้งนี้คือนำไมโครคอนโทรลเลอร์ MCS-51 ประยุกต์ใช้ในการควบคุมซึ่งทำให้ประหยัดค่าใช้จ่ายลงแต่มีประสิทธิภาพเมื่อนำไปใช้ปรับปรุงในเครื่องรุ่นเก่า หรือนำไปสร้างเครื่องใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	CNC Position Controller with Microcontroller
Student	Mr. Theerawat Thaneerat
	Mr. Sutee Pingyom
	Mr. Sumith Vongchai
	Mr. Anuntasak Intawong
Degree	Bachelor of Engineering in Industrial Engineering
	King Mongkut's Institute of Technology Ladkrabang
Academic Year	2004
Thesis Advisor	Assoc. Prof.Pornsak Attavanich
	Mr.Pholchai Chotipraynakul

ABSTRACT

The objective of thesis is collage control setting of CNC Milling Machine it is controlled by order of G-Code and work with microcomputer link serial port (RS-232C) to microcontroller MCS-51. It use to drive motor and count signal from encoder. Then it process the signal to position of CNC table and send to microcomputer. It receive and calculate position for next position transmitting. Apply in control it's that safe expend, but sufficient when lead improve with old machine or new machine.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดีคณะผู้จัดทำขอกราบขอบพระคุณ

อาจารย์พลชัย โชติปรายนกุล อาจารย์ที่ปรึกษาเป็นอย่างสูงสำหรับความรู้ (ทั้งในตำราและนอกตำรา) คำแนะนำ ความช่วยเหลือ ความเอาใจใส่ ความปรารถนาดี สำหรับเวลาที่อยู่จนดึกดื่น และทุกสิ่งทุกอย่างที่มีต่อนักศึกษา

รศ.พรศักดิ์ อรรถวานิช หัวหน้าภาควิชาวิศวกรรมอุตสาหการ ให้คำแนะนำ กำลังใจ คอยสอบถามปัญหาในการทำงานอยู่บ่อยๆ

ผศ.ดร.สรรพสิทธ์ ลิ่มนรรัตน์ เป็นผู้ดูแลเอาใจใส่ทุกๆ เรื่อง และคำปรึกษาที่ดีๆ ตลอด 24 ชั่วโมง

ดร.สิทธิพร พิมพ์สกุล เป็นผู้ดูแลเรื่องการสอบโปรเจ็ค ค่าใช้จ่ายในการทำโปรเจ็ค และความปรารถนาดี

อาจารย์เชาวลิต หามนตรี เป็นผู้ให้คำปรึกษาในทุกๆ เรื่อง และกำลังใจ

ขอขอบพระคุณอาจารย์ทุกๆ ท่านที่มอบวิชาความรู้

ขอขอบพระคุณ คุณพ่อ คุณแม่ และญาติๆ ทุกคน ที่ให้โอกาสและ คอยเป็นกำลังใจเสมอมา

ขอขอบพระคุณ ป้ามะลิ สำหรับภาควิชาและห้องเรียน สะอาดๆ

ขอขอบคุณ พี่ติ่งสำหรับเครื่องมืออุปกรณ์ และข้อมูลจากคอมพิวเตอร์ในห้องพี่

ขอขอบคุณ พี่แตนสำหรับดูแลเอกสารต่างๆ และจัดการให้ทุกๆ เรื่อง

ขอขอบคุณเพื่อนๆ น้องๆ ภาควิชาวิศวกรรมอุตสาหการทุกๆ คน

ขอขอบคุณบ้านซอยลีลาและทุกกิจกรรมดีๆ ที่มีอยู่ตลอด

นาย ธีรวัฒน์ ธานีรัตน์

นาย สุธี ปิงยอม

นาย สุมิตร วงษ์ไชย

นาย อนันตศักดิ์ อินทวงค์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก

บทคัดย่อภาษาอังกฤษ ..... ข

กิตติกรรมประกาศ ..... ค

สารบัญ ..... ง

สารบัญตาราง ..... ช

สารบัญรูป ..... ซ

บทที่ 1 บทนำ

1.1 ความสำคัญของโครงงานและที่มาของโครงงาน 1

1.2 จุดมุ่งหมายและวัตถุประสงค์ 1

1.3 ขอบเขตการศึกษา 1

1.4 ประโยชน์ที่คาดว่าจะได้รับ 1

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง

2.1 ไมโครคอนโทรลเลอร์ MCS-51 2

2.1.1 โครงสร้างของ MCS-51 2

2.1.2 คุณสมบัติของไมโครคอนโทรลเลอร์ ตระกูล MCS-51 อนุกรม AT89xx 3

2.1.3 การจัดขาของไมโครคอนโทรลเลอร์ MCS-51 4

2.1.4 โครงสร้างและการทำงานของพอร์ต 6

2.1.5 จังหวะการทำงานของไมโครคอนโทรลเลอร์ MCS-51 8

2.1.6 การจัดหน่วยความจำของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช 11

2.2 พอร์ตอนุกรม 19

2.1.7 การสื่อสารแบบอนุกรม 19

2.1.8 การสื่อสารข้อมูลแบบอะซิงโครนัส 19

2.1.9 มาตรฐานพอร์ตอนุกรมแบบ RS-232 21

2.1.10 คอนเน็คเตอร์สำหรับพอร์ต RS-232 และการเชื่อมต่อ 23

2.1.11 UART (Universal Asynchronou Receiver Transmitter) 24

2.1.12 วงจรภายในและรีจิสเตอร์ของพอร์ตอนุกรม RS-232 25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.2 การควบคุมมอเตอร์ 32
2.2.1 มอเตอร์ 32
2.2.2 การควบคุมการขับเซอร์โว 34
2.2.3 ชุดขับและภาคขยายกำลังของเซอร์โวมอเตอร์ 34
2.2.4 เอนโคเดอร์ชนิดแสง 36
2.2.5 ชุดควบคุมการทำงานของเครื่อง 37
2.2.6 ส่วนติดต่อกับผู้ใช้งาน 38
2.2.7 การโปรแกรมเครื่องซีเอ็นซี 39
2.3 การขับป้อน 41
2.4 อินเตอร์โพเลเตอร์ในระบบการผลิต 42
2.4.1 วิธีของ MIT 42
2.4.2 วิธี Digital Differential Analyzer: DDA 43
2.4.3 วิธีคำนวณทางพีชคณิต 47

บทที่ 3 การออกแบบและการดำเนินการ

3.1 การวางแผนการดำเนินงาน 48
3.1.1 ศึกษาระบบการทำงานของเครื่องซีเอ็นซี 48
3.1.2 ตรวจสอบส่วนประกอบทางกลและทางไฟฟ้า 48
3.1.3 วางแผนการแก้ไขและปรับปรุง 48
3.1.4 ทำการซ่อมหรือเปลี่ยนชิ้นส่วนอุปกรณ์ใหม่ 48
3.1.5 ทดสอบอุปกรณ์ขั้นสุดท้าย 49
3.2 หลักการทำงานและการออกแบบ 49
3.1.6 หลักการทำงานโดยรวม 49
3.1.7 วงจรไมโครคอนโทรลเลอร์ MCS-51 50

บทที่ 4 ผลการดำเนินงาน

4.1 ส่วนประกอบทางกล 51
4.2 ส่วนประกอบทางไฟฟ้า 51
4.3 ส่วนซอร์ฟแวร์ 54
4.4 ทดสอบกัดชิ้นงาน 54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 5 สรุปผลและวิเคราะห์ผลการดำเนินงาน

5.1 สรุปผลการดำเนินงาน..... 56

5.1.1 สรุปผลการดำเนินงานด้านฮาร์ดแวร์..... 56

5.1.2 สรุปผลการดำเนินงานด้านไฟฟ้าและอิเลคทรอนิคส์..... 56

5.1.3 สรุปผลการดำเนินงานด้านซอร์ฟแวร์.....56

5.2 ปัญหาที่พบในการทำงาน..... 56

5.3 แนวทางการพัฒนาและปรับปรุง..... 57

หนังสืออ้างอิง..... 58

ภาคผนวก ก.....ผก1

ภาคผนวก ข.....ผข1

ภาคผนวก ค.....ผค1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 2.1 แสดงบิตพาริตี้ของข้อมูล.....21
ตารางที่ 2.2 การจัดขาของคอนเน็กเตอร์พอร์ตอนุกรมตามมาตรฐาน RS-232.....22
ตารางที่ 2.3 ตัวอย่างคำสั่งรหัสจีพีนฐาน.....41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่ 2.1 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 ในอนุกรม AT89Cxx....3
รูปที่ 2.2 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ MCS-51 ในอนุกรม AT89Cxx....5
รูปที่ 2.3 วงจรภายในของพอร์ตทุกพอร์ตในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช....6
รูปที่ 2.4 ไซเคิลการทำงานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช....9
รูปที่ 2.5 ไดอะแกรมเวลาแสดงการติดต่อและการเข้าถึงหน่วยความจำโปรแกรมภายนอกของ MCS-51 ไมโครคอนโทรลเลอร์แบบแฟลช....10
รูปที่ 2.6 การจัดสรรหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช....11
รูปที่ 2.7 การเชื่อมต่อหน่วยความจำโปรแกรมภายนอกของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช....13
รูปที่ 2.8 การจัดสรรพื้นที่ของหน่วยความจำข้อมูลภายในของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช....13
รูปที่ 2.9 การจัดสรรพื้นที่ของหน่วยความจำข้อมูลภายในส่วนล่างของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช....14
รูปที่ 2.10 โครงสร้างของหน่วยความจำข้อมูลส่วนบนของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช....15
รูปที่ 2.11 โครงสร้างของหน่วยความจำข้อมูลส่วนบนของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช....16
รูปที่ 2.12 รายละเอียดของรีจิสเตอร์แสดงสถานะของโปรแกรมหรือ PSW....17
รูปที่ 2.13 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรม....19
รูปที่ 2.14 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบอะซิงโครนัส....20
รูปที่ 2.15 คอนเน็กเตอร์อนุกรม 9 ขาหรือแบบ DB-9 (มองจากด้านหลังคอมพิวเตอร์) 22
รูปที่ 2.16 คอนเน็กเตอร์อนุกรม 25 ขาหรือแบบ DB-25 (มองจากด้านหลังคอมพิวเตอร์)....22
รูปที่ 2.17 การอุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์แบบ Null modem....24
รูปที่ 2.18 การอุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์แบบ RS-232 โดยใช้สายสัญญาณเพียง 3 เส้น....24
รูปที่ 2.19 ไดอะแกรมแสดงโครงสร้างทางฮาร์ดแวร์ของพอร์ตอนุกรม....31
รูปที่ 2.20 หลักการทำงานของมอเตอร์ไฟฟ้ากระแสตรงแบบมีแปรงถ่าน....32
รูปที่ 2.21 หลักการทำงานของมอเตอร์ไฟฟ้ากระแสสลับไม่ต้องใช้แปรงถ่าน....33
รูปที่ 2.22 ชุดขับมอเตอร์แบบเชิงเส้น....35
รูปที่ 2.23 หลักการสร้างสัญญาณ PWM....35
รูปที่ 2.24 วงจรขับเซอร์โวมอเตอร์กระแสตรงแบบมีแปรงถ่าน....36
รูปที่ 2.25 พัลซ์ที่เกิดจากเอนโคเดอร์แบบส่วนเพิ่ม....37
รูปที่ 2.26 ตัวอย่างระบบการควบคุมการเคลื่อนที่....37
รูปที่ 2.27 วิธีของ MIT....43
รูปที่ 2.28 ทฤษฎีของวิธี DDA....43
รูปที่ 2.29 วงจรพื้นฐานของ DDA....44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

หน้า

รูปที่ 2.30 วงจรอินเตอร์โพเลชั่นแบบเส้นตรง..... 45
รูปที่ 2.31 การอินเตอร์โพเลชั่นแบบเส้นตรง..... 45
รูปที่ 2.32 วงจรอินเตอร์โพเลชั่นเส้นโค้ง.....46
รูปที่ 2.33 การอินเตอร์โพเลชั่นเส้นโค้ง.....47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญของโครงงานและที่มาของโครงงาน

ระบบการควบคุมเครื่องจักรด้วยคอมพิวเตอร์ (Computer Numerical Control: CNC) ที่ใช้งานในปัจจุบันระบบการควบคุมตำแหน่งของเครื่องที่ใช้เป็นชุดมีหน่วยประมวลผลเดี่ยวทำให้มีราคาสูง การใช้หน่วยประมวลผลที่มีราคาต่ำอย่างไมโครคอนโทรลเลอร์มาแทนที่ และสามารถตรวจสอบตำแหน่งในความเร็วสุ่มค่าที่เท่ากัน จะทำให้ชุดควบคุมมีราคาต่ำลง และสามารถใช้ทดแทนการซ่อมบำรุงเครื่องซีเอ็นซีรุ่นเก่าที่สามารถใช้งานได้

1.2 จุดมุ่งหมายและวัตถุประสงค์

1. เพื่อศึกษาระบบการทำงานของเครื่องซีเอ็นซี
2. ประยุกต์ใช้ไมโครคอนโทรลเลอร์ MCS-51 เพื่อควบคุมการเคลื่อนที่ของเซอร์โวมอเตอร์
3. ประยุกต์ใช้ไมโครคอมพิวเตอร์ควบคุมการทำงานของเครื่องซีเอ็นซี
4. สามารถนำเครื่องซีเอ็นซีรุ่นเก่ากลับมาใช้งานได้

1.3 ขอบเขตการศึกษา

1. สร้างวงจรไมโครคอนโทรลเลอร์ MCS-51 สามารถควบคุมการเคลื่อนที่ของเซอร์โวมอเตอร์
2. โปรแกรมสามารถรับจีโค้ด (G-Code) ไปประมวลผลสั่งให้เครื่องทำงานตามที่สั่งงานได้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. เครื่องซีเอ็นซีสามารถนำกลับมาใช้งานได้
2. รูปแบบของวงจรควบคุมด้วยไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 ไมโครคอนโทรลเลอร์ MCS-51

2.1.1 โครงสร้างของ MCS-51

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 (Microcontroller MCS-51) มีด้วยกันหลายเบอร์ขึ้นกับโครงสร้างภายในของมัน บางเบอร์จะมีหน่วยความจำภายในเป็นแบบ ROM (Read Only Memory) บางเบอร์เป็น EPROM (Erasable Programmable Read-Only Memory) บางเบอร์มี RAM (Random Access Memory) ภายใน 128ไบต์ บางเบอร์มี 256 ไบต์ เป็นต้น คุณสมบัติที่สำคัญของ MCS-51 มีดังนี้

- มีหน่วยความจำ ROM 4 K bytes
- มีหน่วยความจำ RAM 128 bytes
- มีพอร์ต I/O ขนาด 8 บิต 4 พอร์ต
- มีไทม์เมอร์ (Timer) 16 บิต 2ตัว
- สามารถอินเทอร์รัพท์ได้ 5 แหล่ง
- มีวงจรออสซิลเลเตอร์และวงจรนาฬิกาบนชิพ
- มีพอร์ตอนุกรมที่สามารถรับส่งข้อมูลแบบฟูลดูเพล็กซ์ (Full Duplex) ความเร็ว
- อ้างหน่วยความจำภายนอกได้ 64 K
- อ้างหน่วยความจำภายในได้ 64 K
- สามารถประมวลผลทีละบิตได้
- สามารถอ้างหน่วยความจำแบบบิตได้ 210 ตำแหน่ง
- หนึ่งวัฏจักรคำสั่งกินเวลาประมาณ 1 ไมโครวินาที ขณะทำงานด้วยสัญญาณนาฬิกา (Clock) 12 MHz

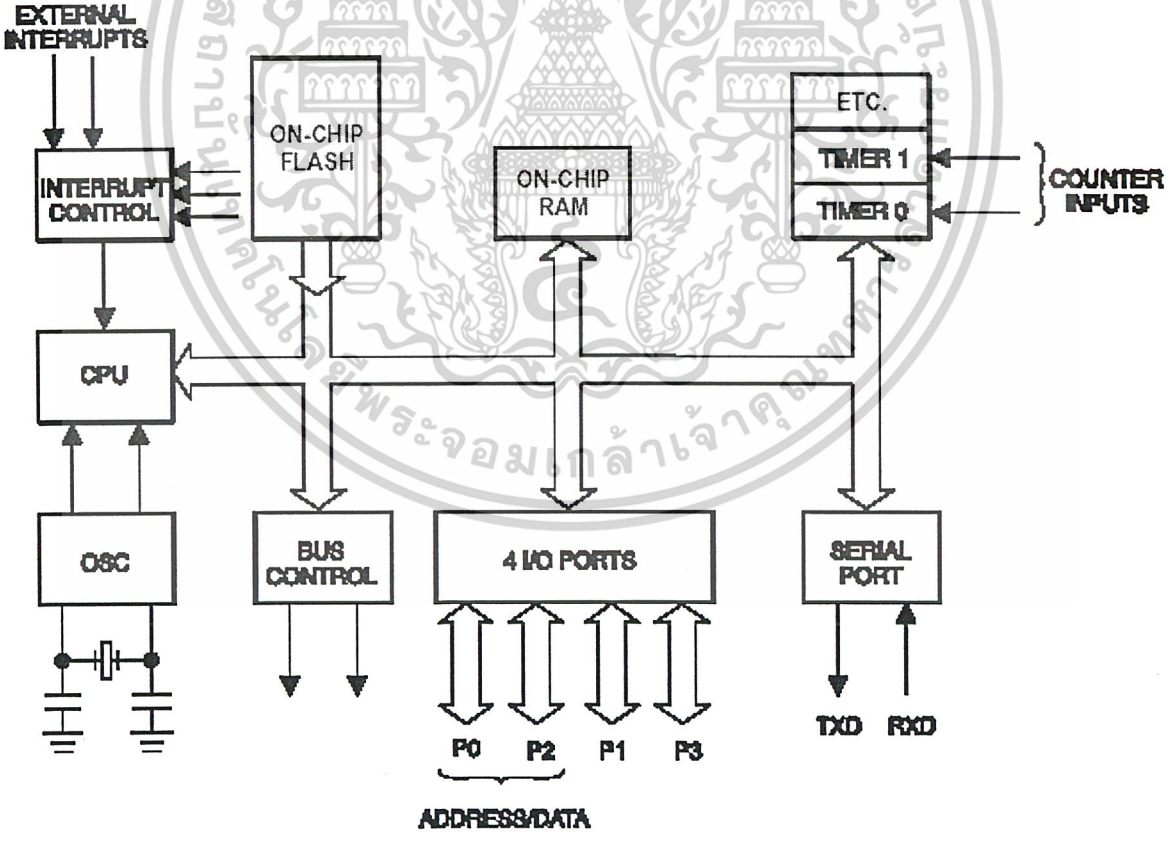
การใช้งาน MCS-51 จะต้องมีหน่วย ROM สำหรับเก็บโปรแกรมควบคุมการทำงาน โดยหน่วยความจำ ROM นี้แบ่งได้เป็น Mask ROM, PROM (Programmable Read-Only Memory), EPROM และ EEPROM (Electronically Erasable Programmable Read Only Memory) /Flash โดย Mask ROM นั้นโปรแกรมภายในจะต้องถูกโปรแกรมมาจากการผลิตในโรงงาน สำหรับ PROM นั้นผู้ใช้สามารถนำมาโปรแกรมได้ 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 อยู่ในชิพสามารถโปรแกรมและลบแก้ไขโปรแกรมได้ง่าย แต่ถ้าเป็นเบอร์ 8951 จะมี ROM แบบ EEPROM/Flash ซึ่งเป็นที่นิยมใช้กัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาก โดยการโปรแกรมและแก้ไขโปรแกรมจะทำได้โดยไม่ต้องถอดชิพออกจากแผงวงจรการใช้งาน MCS-51 ที่มีหน่วยความจำโปรแกรมภายในชิพนี้ทำให้สามารถป้องกันการคัดลอกโปรแกรมได้อย่างดี ปัจจุบันมีไมโครคอนโทรลเลอร์ตระกูล MCS-51 ที่สามารถโปรแกรมข้อมูลลงไปในชิพได้โดยไม่ต้องถอดชิพออกจากแผงวงจรเรียกว่าเป็นการโปรแกรมในระบบ (In-System Programming) เช่นเบอร์ AT89xx ทำให้สามารถใช้งานได้ง่ายขึ้น

2.1.2 คุณสมบัติของไมโครคอนโทรลเลอร์ ตระกูล MCS-51 อนุกรม AT89xx

- เป็นไมโครคอนโทรลเลอร์ที่มีซีพียูขนาด 8 บิต
- ภายในมีหน่วยความจำโปรแกรมเป็นแบบแฟลชสามารถลบและเขียนใหม่ได้พันครั้ง
- หน่วยความจำข้อมูลพื้นฐานเป็นหน่วยความจำแบบแรม (RAM) ในบางเบอร์จะมีหน่วยความจำอีพรอมเพิ่มเติม
- ขาพอร์ตเป็นแบบสองทิศทาง สามารถใช้งานเป็นได้ทั้งอินพุตและเอาต์พุต
- มีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์
- ไทเมอร์/เคาน์เตอร์ ขนาด 16 บิตอย่างน้อย 2 ตัว
- สามารถรองรับแหล่งกำเนิดอินเตอร์รัปต์ได้ 6 ประเภท



รูปที่ 2.1 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 ในอนุกรม AT89Cxx

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สามารถขยายหน่วยความจำภายนอกเพิ่มเติมได้สูงสุด 64 กิโลไบต์
- มีวงจรกำเนิดสัญญาณนาฬิกาอยู่ภายในชิพ
- มีวงจรอนุกรมแบบ SPI สำหรับในอนุกรม AT89Sxx
- มีวอตช์ด็อกไทเมอร์ในตัว สำหรับในอนุกรม AT89Sxx

รูปที่ 2.1 (คู่มือการใช้งานไมโครคอนโทรลเลอร์ Atmel,2540) เป็นโครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 ในอนุกรม AT89Cxx จะเห็นได้ว่าโครงสร้างของ AT89Cxx จะเหมือนกับไมโครคอนโทรลเลอร์ตระกูล MCS-51 พื้นฐาน หากแต่ต่างกันเฉพาะหน่วยความจำโปรแกรมแบบแฟลชที่เพิ่มเติมขึ้นมา หากเป็นไมโครคอนโทรลเลอร์ในอนุกรม 87xx หน่วยความจำโปรแกรมภายในจะเป็นแบบอีพรอม และบางเบอร์สามารถโปรแกรมได้เพียงครั้งเดียว

โครงสร้างพื้นฐานของอนุกรม AT89Sxx มีส่วนประกอบที่เพิ่มเติมแตกต่างจาก AT89Cxx อยู่หลายส่วน เช่น วงจรเชื่อมต่ออนุกรมแบบ SPI ซึ่งในไมโครคอนโทรลเลอร์อนุกรมนี้ใช้ในการเขียนข้อมูลลงในหน่วยความจำโปรแกรมโดยไม่ต้องถอดตัวชิพออกไปจากระบบหรือเรียกว่าการโปรแกรมในวงจร ไทเมอร์/เคาน์เตอร์ขนาด 16 บิตที่เพิ่มเติมเข้ามาอีกหนึ่งตัว เป็นไทเมอร์ 2 และวงจรวอตช์ด็อกที่ใช้ในการตรวจสอบการทำงานผิดพลาดของซีพียู

2.1.3 การจัดขาของไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ MCS-51 ทุกเบอร์จะมีขาใช้งานพื้นฐานเหมือนกัน ดังรูปที่ 2.2 (คู่มือการใช้งานไมโครคอนโทรลเลอร์ (Philips, 2540) โดยมีรายละเอียดดังนี้

ขา Vcc ใช้สำหรับต่อไฟเลี้ยง +5V

ขา GND เป็นขากราวด์ สำหรับต่อกับกราวด์ของระบบ

ขาพอร์ต 0 (P0.0-P0.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็น ได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (Float) จึงมีอินพุตอิมพีแดนซ์สูงสามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ตนี้ยังถูกใช้งานในการติดต่อกับขาแอดเดรสไบต์ต่ำของหน่วยความจำภายนอก (A0-A7) และขา (D0-D7) โดยใช้กระบวนการมัลติเพล็กซ์เข้าช่วย เพื่อสลับการทำงานเป็นได้ทั้งขาติดต่อแอดเดรสและขาข้อมูล

ขาพอร์ต 1 (P1.0-P1.7) มี 8 ขาแต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดขาพอร์ตใดเป็นอินพุต สามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย นอกจากนั้นในอนุกรม AT89Sxx จะใช้ขา P1.0 เป็นขาอินพุตสำหรับค่าของไทเมอร์ 2 และพอร์ต P1.1 เป็นขาอินพุตทริกเกอร์ของไทเมอร์ 2 ในขณะที่ขา P1.4-P1.7 เป็นขาสำหรับต่อเชื่อมต่อแบบ SPI เพื่อทำการโปรแกรมข้อมูลในระบบ

ขาพอร์ต 2 (P2.0-P2.7) มี 8 ขาแต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดขาพอร์ตใดเป็นอินพุต สามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขานั้นมีสถานะปล่อยลอย (Float) จึงมีอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนี้ยังถูกใช้งานในการติดต่อกับขาแอดเดรสไบต์สูงของหน่วยความจำภายนอก (A8-A15)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขาพอร์ต 3 (P3.0-P7.7) มี 8 ขาแต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดขาพอร์ตใดเป็นอินพุต สามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (Float) จึงมีอินพุตอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ต 3 ยังเป็นขาที่มีหน้าที่การใช้งานพิเศษ มีรายละเอียดดังนี้

P3.0 ใช้เป็นขาอินพุตสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรม หรือขา RxD

P3.1 ใช้เป็นขาอินพุตสำหรับส่งข้อมูลจากการสื่อสารแบบอนุกรม หรือขา TxD

P3.2 ใช้เป็นขาอินพุตรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 0 หรือขา INT0

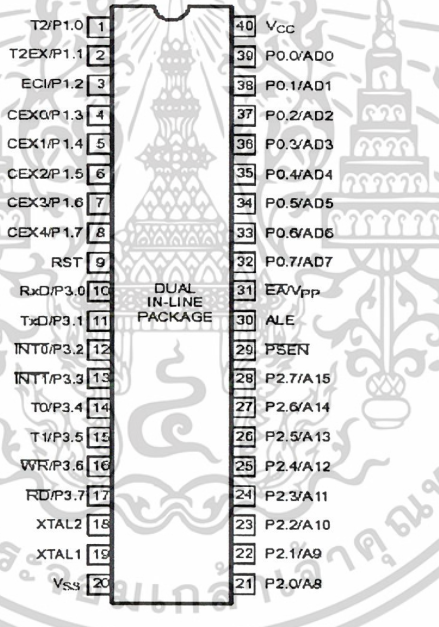
P3.3 ใช้เป็นขาอินพุตรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 1 หรือขา INT1

P3.4 ใช้เป็นขาอินพุตสำหรับรับสัญญาณไทเมอร์จากภายนอกช่อง 0 หรือขา T0

P3.5 ใช้เป็นขาอินพุตสำหรับรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 1 หรือขา T1

P3.6 ใช้เป็นขาสัญญาณ WR ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก

P3.7 ใช้เป็นขาสัญญาณ RD ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก



รูปที่ 2.2 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ MCS-51 ในอนุกรม AT89Cxx

ขารีเซต (Reset) ใช้ในการรีเซตการทำงานของไมโครคอนโทรลเลอร์โดยในการป้อนสัญญาณเพื่อรีเซตสถานะที่ขานี้ต้องอยู่ในระดับรีเซตอย่างน้อย 2 แมชีนไซเคิล โดยที่วงจรกำเนิดนาฬิกายังคงทำงานต่อเนื่องไปอย่างปกติ

ขา ALE/PROG (Address Enable/Program Pulse Input) เป็นขาที่ใช้ในการควบคุมการแลตช์ของขาพอร์ต 0 เมื่อมีการใช้งานหน่วยความจำภายนอก นอกจากนั้นขานี้ยังใช้เป็นขาสำหรับรับพัลส์ของการโปรแกรมสำหรับโปรแกรมข้อมูลลงในไมโครคอนโทรลเลอร์ MCS-51 ในรุ่นที่หน่วยความจำโปรแกรมเป็นแบบอีพรอม

ขา PSEN (Program Store Enable) ขานี้ใช้ในการส่งสัญญาณเพื่อร้องขอติดต่อกับหน่วยความจำโปรแกรมภายนอก เมื่อไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก ตัวไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

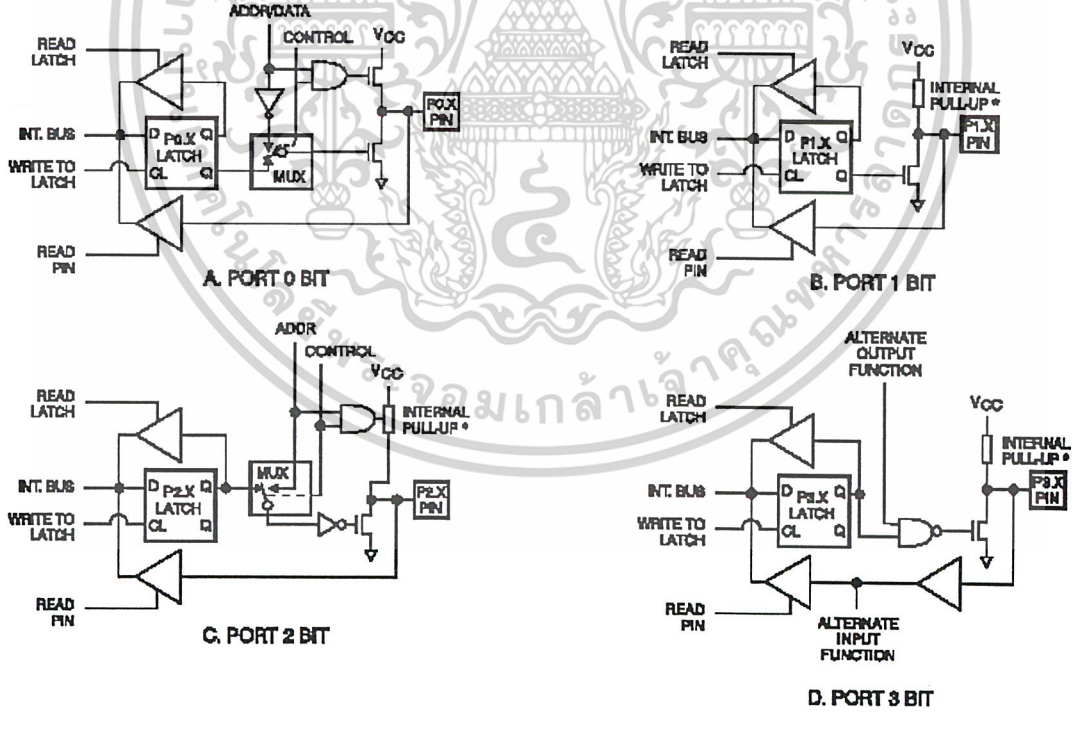
จะส่งสัญญาณออกมาที่ขานี้ 2 ครั้งในแต่ละแมชีนไซเคิล แต่ถ้าหากติดต่อกับหน่วยความจำข้อมูลภายนอก ขานี้จะไม่มีการส่งสัญญาณใดๆออกมา

ขา EA/Vpp (External Access Enable/Programming Voltage Input) ใช้สำหรับเลือกการติดต่อหน่วยความจำโปรแกรมจากภายนอกหรือภายในตัวไมโครคอนโทรลเลอร์ ถ้าหากขานี้เป็น "0" เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำโปรแกรมภายนอก แต่ถ้าขานี้เป็น "1" เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำภายในตัวไมโครคอนโทรลเลอร์ นอกจากนี้ ที่ขานี้ยังใช้เป็นขาอินพุตสำหรับรับแรงดันไฟฟ้าสูงสำหรับการโปรแกรมหน่วยความจำภายในไมโครคอนโทรลเลอร์ สำหรับไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชต้องการแรงดันสำหรับการโปรแกรม +12V

ขา **XTAL1** และ **XTAL2** เป็นขาสำหรับต่อคริสตอลเพื่อสร้างสัญญาณนาฬิกาในการกำหนดจังหวะการทำงานของไมโครคอนโทรลเลอร์

2.1.4 โครงสร้างและการทำงานของพอร์ต

ไมโครคอนโทรลเลอร์ MCS-51 และแฟลชมีพอร์ตให้ใช้งานทั้งสิ้น 4 พอร์ตคือ พอร์ต 0 ถึงพอร์ต 3 แต่ละพอร์ตมีขนาด 8 บิต เป็นพอร์ตแบบ 2 ทิศทาง กล่าวคือ สามารถเป็นได้ทั้งอินพุตสำหรับรับสัญญาณข้อมูลเข้าและเอาต์พุตสำหรับส่งสัญญาณข้อมูลออก ทุกพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชมีวงจรแลตช์และวงจรขับตลอดจนบัฟเฟอร์อินพุต



รูปที่ 2.3 วงจรภายในของพอร์ตทุกพอร์ตในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

ที่พอร์ต 0 และพอร์ต 2 จะใช้งานเป็นพอร์ตอินพุตและเอาต์พุตสำหรับงานทั่วไปและในการติดต่อกับหน่วยความจำภายนอก สำหรับพอร์ต 3 ทั้งพอร์ต และพอร์ต 1 บางขานอกจากใช้เป็นขาพอร์ตอินพุตเอาต์พุตตามปกติแล้ว ยัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถใช้งานในหน้าที่พิเศษได้อีก รูปที่ 2.3 (คู่มือการใช้งานไมโครคอนโทรลเลอร์ Atmel, 2540) วงจรภายในของแต่ละพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช โดยในรูปที่ 2.3 (A) เป็นวงจรของพอร์ต 0 วงจรแลตช์ของแต่ละบิตในแต่ละพอร์ตก็คือวงจรดีฟลิฟลอปนั่นเอง การอ่านค่าสถานะของพอร์ตและสถานะของแลตช์สามารถกระทำได้อย่างอิสระด้วยสัญญาณที่แยกจากกัน นั่นคือสัญญาณอ่านข้อมูลจากขาพอร์ต และสัญญาณอ่านข้อมูลจากวงจรแลตช์ ส่วนการเขียนข้อมูลมายังพอร์ตต้องส่งสัญญาณมายังขา CLK ของดีฟลิปฟล็อปในขณะที่ข้อมูลจะผ่านมาทางขาบัสข้อมูลภายในเข้าสู่ขา D ของดีฟลิฟลอป

ที่พอร์ตนี้มีวงจรมัลติเพล็กซ์สำหรับกำหนดลักษณะการทำงานของพอร์ตว่าต้องการใช้งานเป็นขาพอร์ตอินพุตปกติหรือใช้ในการติดต่อกับหน่วยความจำภายนอกไมโครคอนโทรลเลอร์

เนื่องจากที่ขาพอร์ต 0ไม่มีวงจรพูลอัปภายใน หากมีการนำพอร์ต 0 ไปใช้งานเป็นพอร์ตอินพุตจะต้องต่อตัวต้านทานพูลอัปภายนอกเข้าที่ขาพอร์ต 0 ทุกขาด้วย

ในรูปที่ 2.3 (B) เป็นวงจรของพอร์ต 1 ซึ่งมีลักษณะโดยทั่วไปคล้ายกับพอร์ต 0 หากแต่ไม่มีวงจรมัลติเพล็กซ์เนื่องจากพอร์ตนี้จะไม่ใช้ในการติดต่อกับหน่วยความจำภายนอก แต่จะมีวงจรพูลอัปภายในที่แต่ละบิตของพอร์ตนี้แทน

ในรูปที่ 2.3 (C) เป็นวงจรภายในของพอร์ต 2 จะคล้ายกับพอร์ต 0 มาก ต่างกันเพียงมีวงจรพูลอัปเพิ่มเติมเข้ามา ส่วนในรูปที่ 2.3 (D) เป็นวงจรภายในของพอร์ต 3 จะเห็นได้ว่าคล้ายกับพอร์ต 1มีการเพิ่มเติมวงจรบัฟเฟอร์และวงจรอินพุตเอาต์พุตเมื่อทำงานในฟังก์ชั่นพิเศษเข้ามา เนื่องจากพอร์ต 3 สามารถนำไปใช้งานในหน้าที่พิเศษได้ทุกขา

1. การใช้งานเป็นพอร์ตอินพุต

เนื่องจากพอร์ตทั้งหมดของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชสามารถเป็นได้ทั้งอินพุตและเอาต์พุต ดังนั้นจึงจำเป็นอย่างยิ่งต้องทำความเข้าใจถึงการกำหนดลักษณะการทำงานให้แก่พอร์ตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

ในการกำหนดให้เป็นพอร์ตอินพุต ต้องเริ่มต้นด้วยการเขียนข้อมูล “1” มาที่แต่ละบิตของพอร์ตที่ต้องการใช้งานเป็นอินพุต เพื่อหยุดการทำงานของเฟสที่ใช้ในการขับสัญญาณเอาต์พุตของบิตนั้นๆทำให้ขาของสัญญาณของพอร์ตเชื่อมต่อกับวงจรพูลอัปภายในโดยตรง ส่งผลให้ขาพอร์ตนั้นมีลอจิกเป็น “1” สามารถรับสัญญาณลอจิก “0” จากอุปกรณ์ภายนอกได้ง่าย สัญญาณข้อมูลจากอุปกรณ์ภายนอกจะถูกส่งเข้ามาแล้วเก็บไว้ในวงจรบัฟเฟอร์ภายในพอร์ตแล้วรอให้ซีพียูมาอ่านค่าเข้าไป เมื่อเป็นเช่นนี้ อุปกรณ์ภายนอกที่เชื่อมต่อกับพอร์ตอินพุตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชควรกำหนดให้ทำงานในสภาวะลอจิกเป็น “0” จะดีและสะดวกที่สุด

2. การใช้งานเป็นพอร์ตเอาต์พุต

โดยปกติแล้ว ขาพอร์ตจะกำหนดให้มีลักษณะเป็นเอาต์พุตอยู่แล้ว ดังนั้นจึงสามารถส่งข้อมูลออกไปได้อย่างง่ายดายและตรงไปตรงมา กล่าวคือ เมื่อต้องการส่งข้อมูล “0” ออกไปทางเอาต์พุตก็ให้เขียนข้อมูล “0” ไปยังวงจรแลตช์ ซึ่งก็จะส่งผลต่อไปขับเฟส ทำให้เฟสทำงาน ที่ขาพอร์ตที่กำหนดให้ทำงานก็จะเกิดลอจิก “0” ขึ้น ในทางตรงกันข้ามหากต้องการส่งข้อมูล “1” ออกไป ก็ให้เขียนข้อมูล “1” ไปยังวงจรแลตช์ วงจรขับก็จะหยุดทำงาน ทำให้ที่ขาพอร์ตเชื่อมต่อกับวงจรพูลอัปภายในเกิดเป็นลอจิก “1” ที่ขาพอร์ตนั้น ซึ่งจะคล้ายกับการกำหนดให้เป็นขาอินพุตมาก เพียงแต่แตกต่างกันที่กระบวนการในการเคลื่อนย้ายข้อมูล โดยถ้าเป็นอินพุตจะมีสัญญาณมาอ่านข้อมูลที่บัฟเฟอร์ แต่ถ้าเป็นเอาต์พุตจะไม่มีการอ่านข้อมูลที่บัฟเฟอร์แต่อย่างใด เว้นแต่ในกรณีที่ต้องการตรวจสอบข้อมูลที่ส่งออกมาทางเอาต์พุต

เมื่อใช้งานเป็นพอร์ตเอาต์พุต แต่ละขาของแต่ละพอร์ตมีความสามารถในการจ่ายกระแสหรือที่เรียกว่า กระแสซอร์ส (Source Current) ได้สูงสุด 10 mA และทุกขารวมกันในแต่ละพอร์ต (ทั้ง 8 บิต) สูงสุด 26 mA สำหรับพอร์ต 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ 15 mA สำหรับพอร์ต1-3 ในกรณีที่ใช้งานทุกพอร์ตเอาต์พุตจะสามารถจ่ายกระแสได้รวมกันสูงสุด 71 mA ดังนั้นในการใช้งานเป็นพอร์ตเอาต์พุตเพื่อไม่ให้เกิดปัญหาเกี่ยวกับความสามารถในการจ่ายกระแสจึงควรต่อวงจรบัฟเฟอร์ทางเอาต์พุตเพื่อช่วยในการขับกระแสอีกทางหนึ่ง

3. การอ่านค่าลอจิกจากพอร์ต

ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชสามารถอ่านค่าลอจิกจากพอร์ตได้ 2 ลักษณะคือ อ่านจากพอร์ตโดยตรง และอ่านจากวงจรแลตช์ของแต่ละพอร์ต

ในกรณีที่พอร์ตต่อกับทรานซิสเตอร์ชนิด NPN และขาอิมิตเตอร์ของทรานซิสเตอร์ตัวนั้นต่อลงกราวด์ หากมีการส่งข้อมูล "1" ไปยังทรานซิสเตอร์ จะทำให้ทรานซิสเตอร์ทำงานสถานะลอจิกที่มีขาพอร์ตจะเป็น "0" เนื่องจากเมื่อทรานซิสเตอร์ทำงาน จะเหมือนว่าขาพอร์ตนั้นถูกต่อลงกราวด์ ทำให้อ่านค่าลอจิกที่ขาพอร์ตจะให้ผลตรงข้ามกับที่ต่อออกมา แต่ถ้าหากทำการอ่านลอจิกที่วงจรแลตช์ จะได้ค่าที่ตรงกับค่าที่ต้องการส่งจริง ดังนั้นในการอ่านค่าลอจิกจากพอร์ตจึงต้องเลือกวิธีการให้เหมาะสมกับอุปกรณ์ที่นำมาต่อด้วย

2.1.5 จังหวะการทำงานของไมโครคอนโทรลเลอร์ MCS-51

ในการใช้งานไมโครคอนโทรลเลอร์ MCS-51 จะต้องทำความเข้าใจถึงจังหวะการทำงานของซีพียูและลำดับขั้นตอนการประมวลผลคำสั่ง ในการประมวลผลคำสั่งของซีพียูจะมีขั้นตอนหลักๆ 2 ขั้นตอนคือ กระบวนการเฟตช์ (Fetch) เป็นการเรียกคำสั่งออกจากหน่วยความจำโปรแกรมแล้วทำการแปลงรหัสคำสั่งนั้นเป็นภาษาเครื่องเพื่อเตรียมการประมวลผล ขั้นตอนต่อมาคือ กระบวนการเอ็กซิคิวต์ (Execute) เป็นการกระทำตามคำสั่งที่กำหนดหรือตามที่เฟตช์ขึ้นมา โดยกระบวนการก่อนหน้านี้ เมื่อทำการเอ็กซิคิวต์คำสั่งเรียบร้อยแล้วก็จะไปเริ่มกระบวนการเฟตช์คำสั่งใหม่ต่อไป

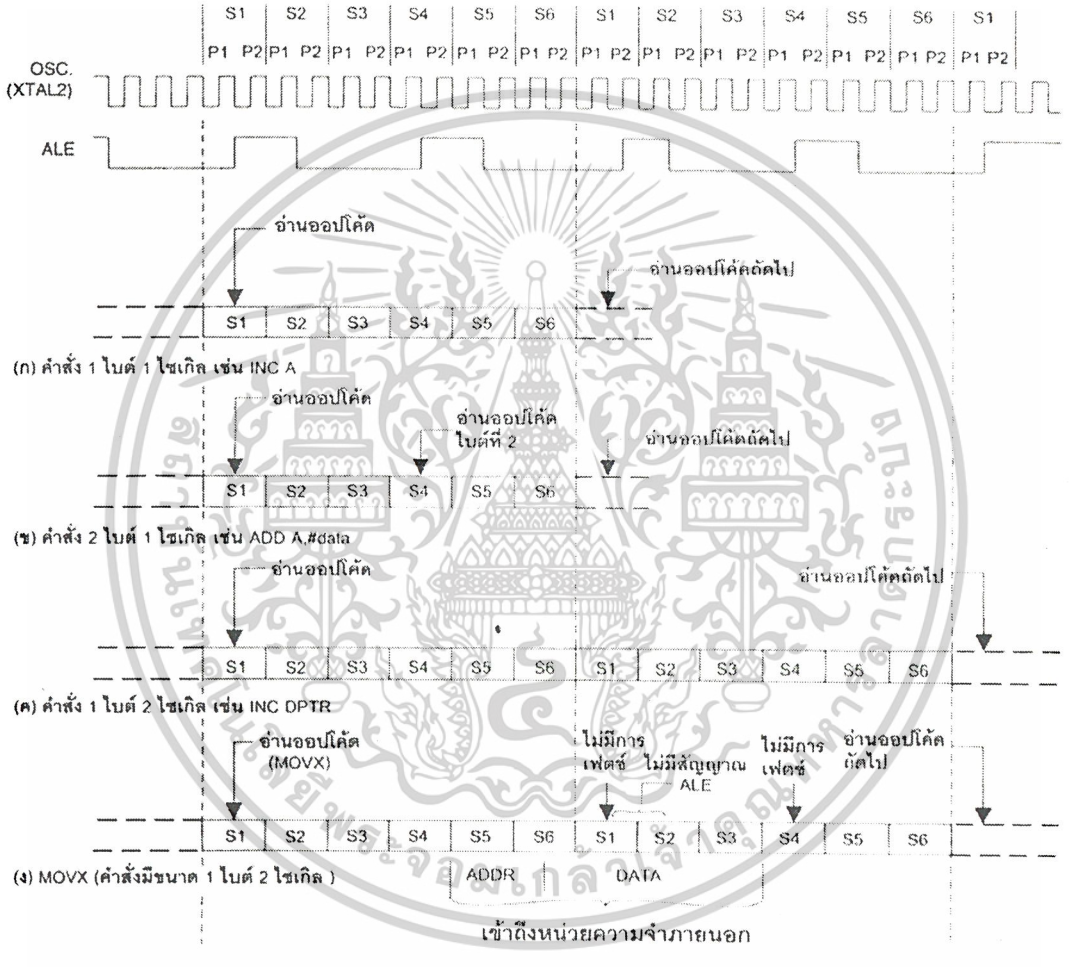
เมื่อเริ่มจ่ายไฟให้แก่ไมโครคอนโทรลเลอร์จะเกิดการรีเซตในลักษณะที่เรียกว่า เพาเวอร์ออนรีเซต (Power On Reset) ซีพียูเริ่มต้นการทำงานที่แอดเดรส 0000H ของหน่วยความจำโปรแกรม จังหวะการทำงานของซีพียูจะเป็นไปตามรูปแบบ โดยได้รับการกำหนดมาจากรอบการทำงานหรือแมชีนไซเคิล (Machine Cycle) ในรูปที่ 2.4 เป็นไดอะแกรมเวลาแสดงจังหวะการทำงานของไมโครคอนโทรลเลอร์ MCS-51 โดยใน 1 รอบการทำงานหรือแมชีนไซเคิลจะแบ่งย่อยออกเป็น 6 สเตต (State) กำหนดชื่อเป็น S1-S6 ในแต่ละสเตตมีค่าเวลาเท่ากับ 1 ms คาบเวลาทั้งสองภายในหนึ่งสเตตจะเรียกว่า เฟส 1 (Phase 1) และ เฟส 2 (Phase 2) ในรูปที่ 2.4 (วรพจน์ กรแก้ววัฒนกุล และ ชัยวัฒน์ ลิ้มพรจิตรวิไล, 2546) (ก) และ (ข) จะเป็นการเอ็กซิคิวต์คำสั่งที่ใช้เวลา 1 ไซเคิล เริ่มต้นที่สเตต 1 จะเป็นการอ่านค่าออปโค้ด อันเป็นกระบวนการแลตช์ค่าออปโค้ดส่งไปให้รีจีสเตอร์คำสั่ง (Instruction Register: IR) การเฟตช์ครั้งที่สองจะเกิดขึ้นที่สเตต 4 ภายในแมชีนไซเคิลเดียวกัน ในกรณีที่เป็นคำสั่งไบต์เดียว การแฟตช์ครั้งที่ 2 ภายในแมชีนไซเคิลเดียวกันจะถูกตัดทิ้งไป ในคำสั่งที่มีใช้เวลา 1 ไซเคิล จะสิ้นสุดการทำงานในสเตต 6 ของแมชีนไซเคิลเดียวกัน

ในกรณีที่คำสั่งใช้เวลา 2 ไซเคิลการทำงานของคำสั่งนั้นจะสิ้นสุดลงในสเตต 6 ของแมชีนไซเคิลที่สองดังในไดอะแกรมรูปที่ 2.4 (ค) สำหรับในการกระทำคำสั่ง MOVX ซึ่งเป็นคำสั่งขนาด 1 ไบต์ 2 ไซเคิล จะไม่มีการเฟตช์เกิดขึ้นในไซเคิลที่สองของคำสั่ง MOVX นี้เนื่องจากซีพียูจะไปทำการติดต่อกับหน่วยความจำภายนอกดังแสดงในไดอะแกรมรูปที่ 2.4 (ง) จะเห็นได้ว่าเวลาในการเอ็กซิคิวต์จะไม่ได้ขึ้นกับว่าทำการติดต่อหน่วยความจำโปรแกรมในหรือนอก

ในรูปที่ 2.5 (วรพจน์ กรแก้ววัฒนกุล และ ชัยวัฒน์ ลิ้มพรจิตรวิไล, 2546) แสดงสัญญาณและไดอะแกรมเวลาของการเข้าถึงโปรแกรมภายนอก โดยในรูปที่ 2.5 (ก) เป็นไดอะแกรมเวลาในขณะที่ยังไม่มีการกระทำคำสั่ง MOVX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณที่ขา ALE และ PSEN จะเกิดการแอกตีฟ 2 ครั้งภายในหนึ่งแมชีนไซเคิล ในทุกครั้งที่ ALE เกิดการแอกตีฟที่พอร์ต 0 (P0) จะมีค่าของรีจีสเตอร์ PC ในไบต์ต่ำออกมา ในขณะที่พอร์ต 2 (P2) ก็จะมีค่า PC ในไบต์สูงเพื่อชี้ไปยังแอดเดรสต่อไปที่ต้องการไปดำเนินการ สำหรับขา PSEN ก็จะเกิดการแอกตีฟเมื่อมีการติดต่อกับหน่วยความจำโปรแกรมภายนอก ในกรณีที่กระทำคำสั่ง MOVX เพื่อเข้าถึงหน่วยความจำข้อมูลภายนอก ที่ขา PSEN จะไม่เกิดการแอ็คทีฟ 2 ครั้งภายใน 1 แมชีนไซเคิลเนื่องจากบัสแอดเดรสและบัสข้อมูลจะถูกใช้ในการติดต่อหน่วยความจำข้อมูลภายนอกแทน แต่สำหรับสัญญาณ ALE ยังคงแอ็คทีฟตามจังหวะการทำงานเหมือนเดิม



รูปที่ 2.4 ไซเคิลการทำงานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

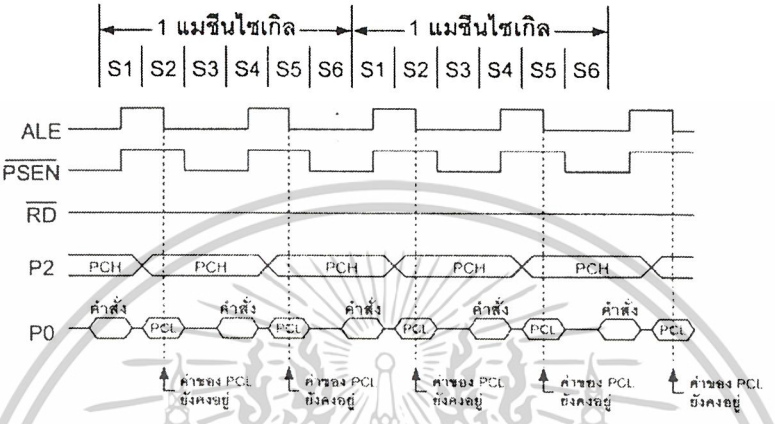
จากไดอะแกรมเวลาสามารถสรุปได้ว่าในการทำงาน 1 รอบ หรือ 1 แมชีนไซเคิล ซีพียูในไมโครคอนโทรลเลอร์ MCS-51 จะใช้เวลา 12 คาบเวลาของสัญญาณนาฬิกานั่นคือ เวลาในการทำงาน 1 ไซเคิลมีค่าเท่ากับ 1 ms หรือมีความเร็วในการทำงานภายใน 1 MHz ในกรณีที่ใช้ความถี่สัญญาณนาฬิกา 12 MHz ดังนั้นถ้าหากต้องการทราบความเร็วของการทำงานภายในของไมโครคอนโทรลเลอร์ MCS-51 สามารถหาได้จาก ค่าความถี่สัญญาณนาฬิกาหารด้วย 12 และถ้าต้องการหาเวลาของ 1 รอบการทำงานหรือ 1 ไซเคิล สามารถทำได้โดยการหาส่วนกลับของความเร็ว ในการทำงานภายในส่วนของไมโครคอนโทรลเลอร์ MCS-51 สามารถสรุปเป็นสูตรทางคณิตศาสตร์ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

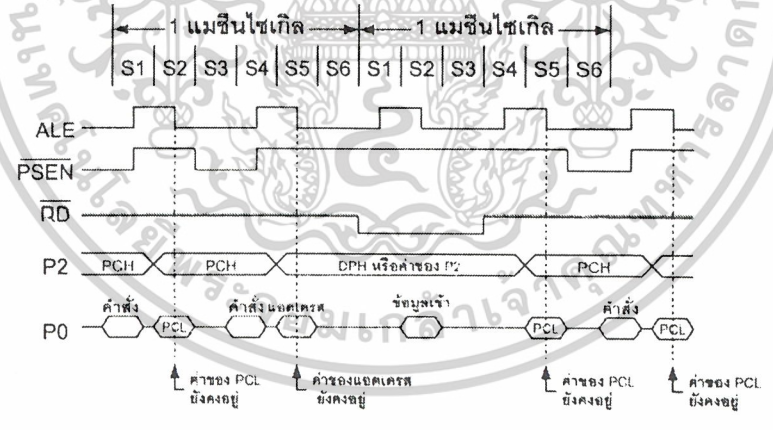
ความเร็วในการทำงานภายในของไมโครคอนโทรลเลอร์เท่ากับ

ความถี่ของสัญญาณนาฬิกา (ค่าของคริสตอลที่ต่ออยู่ที่ขา XTAL1 และ XTAL2)/12

เวลา 1 แมชีนไซเคิล = 1/ความเร็วในการทำงานภายในของไมโครคอนโทรลเลอร์



(ก) เมื่อไม่กระทำคำสั่ง MOVX



(ข) เมื่อกระทำคำสั่ง MOVX

รูปที่ 2.5 ไดอะแกรมเวลาแสดงการติดต่อและการเข้าถึงหน่วยความจำโปรแกรมภายนอกของไมโครคอนโทรลเลอร์MCS-51 แบบแฟลช

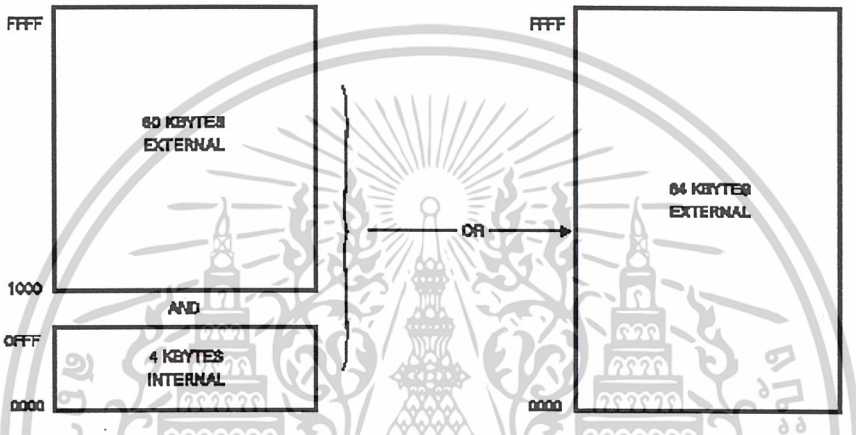
2.1.6 การจัดหน่วยความจำของไมโครคอนโทรลเลอร์ MCS-51 แบบเฟลช

2.1.6.1 <u>หน่วยความจำโปรแกรม</u>

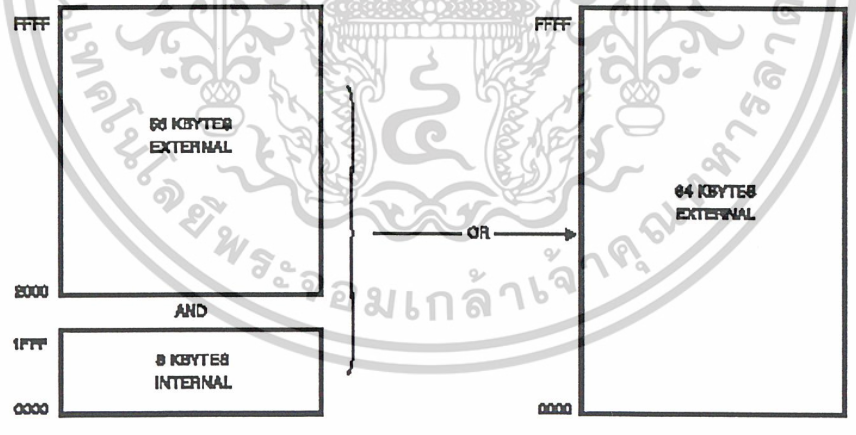
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 2.6 (คู่มือการใช้งานไมโครคอนโทรลเลอร์ Atmel, 2540) แสดงการจัดหน่วยข้อมูลความจำโปรแกรม (Program memory) ของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชในเบอร์ต่างๆที่นิยมใช้งาน อันประกอบด้วยเบอร์ AT89C51 และAT89C52 ทั้งสองเบอร์สามารถติดต่อหน่วยความจำโปรแกรมได้สูงสุด 64 กิโลไบต์ โดยสามารถเลือกใช้หน่วยความจำโปรแกรมภายในอย่างเดียวหรือรวมกันภายนอกหรือเลือกใช้หน่วยความจำภายนอกอย่างเดียวก็ได้ ดังรูปที่ 2.6 (ก) โดยภายใน AT89C51 จะมีหน่วยความจำภายใน 4 กิโลไบต์ ในขณะที่ AT89C52 จะมีขนาด 8 กิโลไบต์

ในกรณีใช้หน่วยความจำภายในและภายนอกรวมกัน หากใช้ AT89C51 ก็จะสามารถติดต่อกับหน่วยความจำภายนอกได้ 60 กิโลไบต์ และถ้าใช้เบอร์ AT89C52 จะสามารถติดต่อกับหน่วยความจำโปรแกรมภายนอกได้ 56 กิโลไบต์



(ก) การจัดสรรหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ MCS-51 เบอร์ AT89C51



(ข) การจัดสรรหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ MCS-51 เบอร์ AT89C52

รูปที่ 2.6 การจัดสรรหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

หน่วยความจำโปรแกรมใช้เก็บข้อมูลของโปรแกรมควบคุมการทำงานของไมโครคอนโทรลเลอร์หรือที่เรียกว่าโปรแกรมมอนิเตอร์ (Monitor program) หากใช้หน่วยความจำภายนอกมักจะบรรจุอยู่ในหน่วยความจำชนิดอีพรอม (EPROM: Erasable Programmable Read-Only memory) ซึ่งสามารถทำการอ่านได้เพียงอย่างเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยความจำโปรแกรมมีแอดเดรสเริ่มต้นที่ 0000H เมื่อซีพียูได้รับการรีเซตให้เริ่มต้นการทำงาน จะต้องมาเริ่มที่แอดเดรส 0000H นี้เสมอ อย่างไรก็ตาม ในพื้นที่ของหน่วยความจำโปรแกรมไม่ว่าจะใช้งานภายในหรือภายนอกก็ตาม ต้องมีการสงวนพื้นที่บางตำแหน่งเอาไว้สำหรับการบริการอินเตอร์รัปต์ 6 ประเภท ประเภทละ 8 ไบต์ ประกอบด้วย

พื้นที่สำหรับการอินเตอร์รัปต์ 0 จากภายนอก กำหนดไว้ที่แอดเดรส 0003H

พื้นที่สำหรับการอินเตอร์รัปต์จากไทเมอร์ 0 กำหนดไว้ที่แอดเดรส 000BH

พื้นที่สำหรับการอินเตอร์รัปต์ 1 จากภายนอก กำหนดไว้ที่แอดเดรส 0013H

พื้นที่สำหรับการอินเตอร์รัปต์จากไทเมอร์ 1 กำหนดไว้ที่แอดเดรส 001BH

พื้นที่สำหรับการอินเตอร์รัปต์ของการสื่อสารอนุกรม กำหนดไว้ที่แอดเดรส 00023H

พื้นที่สำหรับการอินเตอร์รัปต์จากไทเมอร์ 2 กำหนดไว้ที่แอดเดรส 002BH

กรณีที่ใช้ไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชที่มีหน่วยความจำโปรแกรมภายในแต่ต้องการติดต่อกับหน่วยความจำโปรแกรมภายนอกด้วย สามารถทำได้โดยต้องกำหนดแอดเดรสของหน่วยความจำโปรแกรมให้ต่อจากแอดเดรสสุดท้ายของหน่วยความจำโปรแกรมภายในของไมโครคอนโทรลเลอร์ ยกตัวอย่างไมโครคอนโทรลเลอร์ AT89C51 มีหน่วยความจำโปรแกรมขนาด 4 กิโลไบต์ มีแอดเดรสอยู่ระหว่าง 0000H-0FFFH เมื่อต่อหน่วยความจำโปรแกรมภายนอกต้องกำหนดแอดเดรสอยู่ในช่วง 1000H-FFFFH

การต่อหน่วยความจำภายนอกแสดง ดังรูปที่ 2.6 ขาพอร์ต P0.0-P0.7 ใช้เป็นขาข้อมูล D0-D7 และขาแอดเดรสไบต์ต่ำ โดยผ่านวงจรแลตช์ซึ่งปกติใช้เบอร์ 74HC573 และใช้สัญญาณ ALE และ PSEN ในการเลือกใช้งานขา P0.0-P0.7 ใช้ในการเชื่อมต่อกับขาแอดเดรสไบต์สูง A8-A15 ดังนั้นเมื่อการติดต่อกับหน่วยความจำโปรแกรมภายนอกไมโครคอนโทรลเลอร์จะเหลือขาพอร์ตเพียง 16 บิต คือ ขาพอร์ต P1.0-P1.7 และ P3.0-P3.7

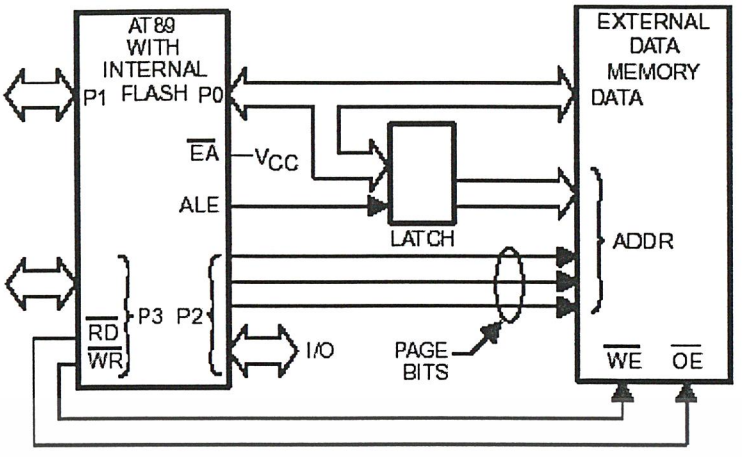
2.1.6.2 <u>หน่วยความจำข้อมูล</u>

มีด้วยกัน 2 แบบคือ หน่วยความจำข้อมูลภายนอกและภายใน (Data memory) โดยไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชในอนุกรม AT89 สามารถติดต่อหน่วยความจำภายนอกได้สูงสุด 64 กิโลไบต์ โดยการใช้คำสั่ง MOVX ในการติดต่อกับหน่วยความจำข้อมูลภายนอก การติดต่อกับหน่วยความจำภายนอกของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชแสดงดังรูปที่ 2.7 (คู่มือการใช้งานไมโครคอนโทรลเลอร์ Atmel, 2540) จะเห็นได้ว่า มีลักษณะคล้ายกับการติดต่อกับหน่วยความจำโปรแกรมภายนอก แตกต่างกันตรงที่มีสัญญาณที่ใช้สำหรับการอ่านและเขียนหน่วยความจำข้อมูลภายนอก นั่นคือ ขา RD และ WR

สำหรับไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชในอนุกรม AT89 ทุกเบอร์จะมีหน่วยความจำข้อมูลภายในเป็นแบบแรม (RAM: Random Access Memory) โดยแต่ละเบอร์จะมีขนาดแตกต่างกันไป ในเบอร์ AT89C51 มีหน่วยความจำข้อมูลภายในขนาด 128 ไบต์ ในขณะที่เบอร์ AT89C52 มีขนาด 256 ไบต์ สำหรับการจัดสรรหน่วยความจำข้อมูลภายในแบ่งเป็น 3 ส่วนคือ หน่วยความจำข้อมูลส่วนล่าง (Lower), ส่วนบน (Upper) และรีจีสเตอร์ฟังก์ชั่นพิเศษ (SFR: Special Function Register) แต่ละส่วนมีขนาด 128 ไบต์ ดังรูปที่ 2.8 (คู่มือการใช้งานไมโครคอนโทรลเลอร์ Atmel, 2540)

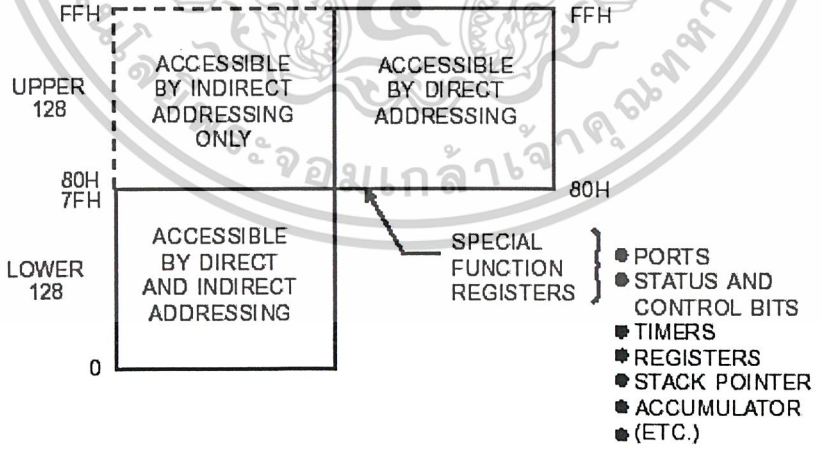
จะเห็นได้ว่าหน่วยความจำข้อมูลส่วนบนและรีจีสเตอร์ฟังก์ชั่นพิเศษมีตำแหน่งทับซ้อนกันแต่จะใช้การติดต่อที่แตกต่างกัน และไมโครคอนโทรลเลอร์ MCS-51 บางเบอร์จะไม่มีหน่วยความจำส่วนบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 การเชื่อมต่อหน่วยความจำโปรแกรมภายนอกของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

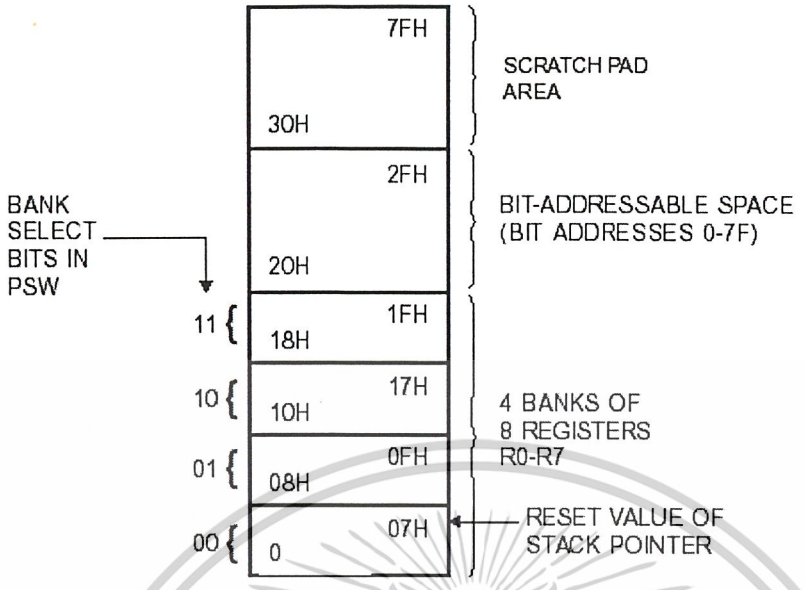
ขนาดของหน่วยความจำข้อมูลไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชโดยแท้จริงแล้วมีเพียง 256 ไบต์ แต่ด้วยการจัดการเข้าถึงที่แตกต่างกัน จึงดูเสมือนว่าไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชมีหน่วยความจำข้อมูลภายในสูงถึง 384 ไบต์ โดยในหน่วยความจำข้อมูลส่วนล่างขนาด 128 ไบต์ มีแอดเดรสอยู่ที่ 00H-7FH สามารถเข้าถึงได้โดยตรงและโดยอ้อม สำหรับหน่วยความจำข้อมูลส่วนบนมีขนาด 128 ไบต์ เช่นกัน มีแอดเดรสอยู่ที่ 80H-FFH สามารถเข้าถึงแบบโดยอ้อมเท่านั้น ในขณะที่รีจิสเตอร์ SFR มีแอดเดรสอยู่ที่ 80H-FFH เช่นเดียวกันหน่วยความจำข้อมูลส่วนบนแต่สำหรับรีจิสเตอร์ SFR ใช้การเข้าถึงแบบตรง ดังนั้นเพื่อความสะดวกและง่าย ตลอดจนป้องกันการสับสนในการเขียนโปรแกรมสำหรับผู้เริ่มต้นจึงควรใช้หน่วยความจำข้อมูลภายในเพียง 128 ไบต์ จากหน่วยความจำข้อมูลส่วนล่างร่วมกับรีจิสเตอร์ SFR



รูปที่ 2.8 การจัดสรรพื้นที่ของหน่วยความจำข้อมูลภายในของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

ใน รูปที่ 2.9 (คู่มือการใช้งานไมโครคอนโทรลเลอร์ Atmel, 2540) แสดงการจัดสรรหน่วยความจำข้อมูลส่วนล่างหน่วยความจำ 32 ไบต์ต่ำสุดที่แอดเดรส 00H-1FH แบ่งเป็น 4 กลุ่ม เรียกว่า 4 แบงก์ (Bank) แต่ละแบงก์มีรีจิสเตอร์ 8 ตัว คือ R0-R7 การติดต่อกับหน่วยความจำในแบงก์ใดให้กำหนดที่รีจิสเตอร์ PSW (Program Status Word register)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 การจัดสรรพื้นที่ของหน่วยความจำข้อมูลภายในส่วนล่างของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

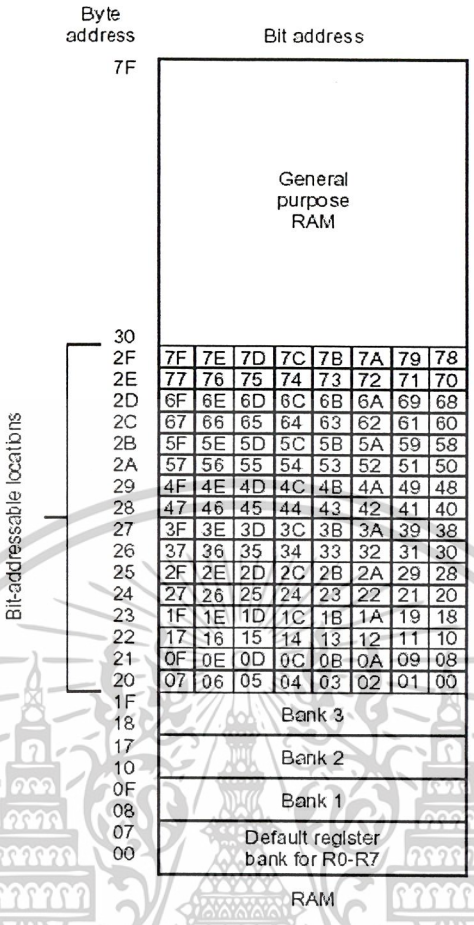
หน่วยความจำข้อมูล 16 ไบต์ถัดมาที่แอดเดรส 20H-2FH เป็นพื้นที่สำหรับใช้งานทั่วไป สามารถเข้าถึงได้ในระดับบิต (Bit Addressable) และหน่วยความจำที่เหลือ 80ไบต์ จะต้องแบ่งส่วนหนึ่งสำรองไว้เป็นพื้นที่ของสแต็ก (Stack: ที่พักข้อมูลชั่วคราวในกรณีที่ซีพียูมีการกระโดดไปทำงานในโปรแกรมย่อย) การเข้าถึงหน่วยความจำในส่วนนี้ต้องใช้การเข้าถึงในระดับไบต์ ใน รูปที่ 2.9 (คู่มือการใช้งานไมโครคอนโทรลเลอร์ Atmel, 2540) แสดงโครงสร้างของหน่วยความจำข้อมูลส่วนบน ซึ่งจะมีลักษณะที่คล้ายกับหน่วยความจำส่วนล่าง หากแต่ใน 80 ไบต์บนไม่จำเป็นต้องสำรองไว้สำหรับสแต็ก และต้องใช้การเข้าถึงหน่วยความจำในลักษณะโดยอ้อมเท่านั้น

2.1.6.3 <u>รีจีสเตอร์ฟังก์ชั่นพิเศษ</u>

เป็นรีจีสเตอร์ที่ใช้การควบคุมการทำงานของไมโครคอนโทรลเลอร์ MCS-51 มีด้วยกัน 22 ตัวสำหรับเบอร์ AT89C51 และ 28 ตัวในเบอร์ AT89C52 และอนุกรม AT89Sxx ทั้งนี้เนื่องจากใน AT89C52 และ AT89Sxx มีจำนวนไทเมอร์เคาน์เตอร์มากกว่า AT89C51

รีจีสเตอร์ SFR มีแอดเดรสอยู่ระหว่าง 80H-FFH ในพื้นที่ของหน่วยความจำข้อมูลส่วนบน สามารถเข้าถึงได้โดยตรง (Direct Addressing) ในรูปที่ 2.11 (คู่มือการใช้งานไมโครคอนโทรลเลอร์ Atmel, 2540) แสดงการจัดสรรพื้นที่ของรีจีสเตอร์ SFR แต่ละตัวในหน่วยความจำข้อมูลส่วนบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 โครงสร้างของหน่วยความจำข้อมูลส่วนบนของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

1. รีจีสเตอร์แสดงสถานะของโปรแกรม (Program Status Word: PSW)

เป็นรีจีสเตอร์ขนาด 8 บิต สามารถเข้าถึงได้ในระดับบิต จึงสามารถกำหนดค่าในแต่ละบิตของรีจีสเตอร์ตัวนี้ได้อย่างอิสระ มีแอดเดรสอยู่ที่ D0H ทำหน้าที่เก็บสถานะของการทำงานของโปรแกรมในขณะนั้นจะเรียกสถานะต่างๆ ของโปรแกรมว่า แฟลก (Flag) เมื่อซีพียูกระทำคำสั่งทางคณิตศาสตร์และลอจิกแล้วเกิดการเปลี่ยนแปลงสถานะขึ้น ผลการเปลี่ยนแปลงนั้นจะมาปรากฏที่บิตต่างๆ ของรีจีสเตอร์ PSW รายละเอียดแสดงดัง รูปที่ 2.12 (คู่มือการใช้งานไมโครคอนโทรลเลอร์ Atmel, 2540)

จะเห็นได้ว่า นอกจากรีจีสเตอร์ PSW ถูกใช้ในการเก็บสถานะของโปรแกรมแล้ว ที่บิต RS0 และ RS1 ยังใช้ในการเลือกแบงก์ของหน่วยความจำส่วนล่าง ซึ่งเป็นพื้นที่ของรีจีสเตอร์ R0-R7 ด้วย โดยปกติแล้วในการใช้รีจีสเตอร์ R0-R7 มักนิยมใช้แบงก์ 0 เป็นอันดับแรก หากไม่เพียงพอจึงเลือกในแบงก์อื่นๆมาใช้ แต่ต้องระมัดระวังในการกำหนดค่าและลำดับการติดต่อให้ดี มิเช่นนั้น อาจทำให้การเขียนโปรแกรมเกิดการสับสน ดังนั้น สำหรับผู้เริ่มใช้งานไมโครคอนโทรลเลอร์ MCS-51 จึงควรเลือกใช้รีจีสเตอร์ R0-R7 ในแบงก์ 0 เพียงแบงก์เดียวให้ชำนาญเสียก่อน

การกำหนดค่าของรีจีสเตอร์ PSW เพื่อเลือกใช้งานรีจีสเตอร์ R0-R7 ควรกำหนดไว้ที่ตอนต้นของโปรแกรมเสมอ เพื่อจะได้เขียนโปรแกรมติดต่อกับรีจีสเตอร์ R0-R7 ได้อย่างสะดวกและไม่เกิดความผิดพลาด

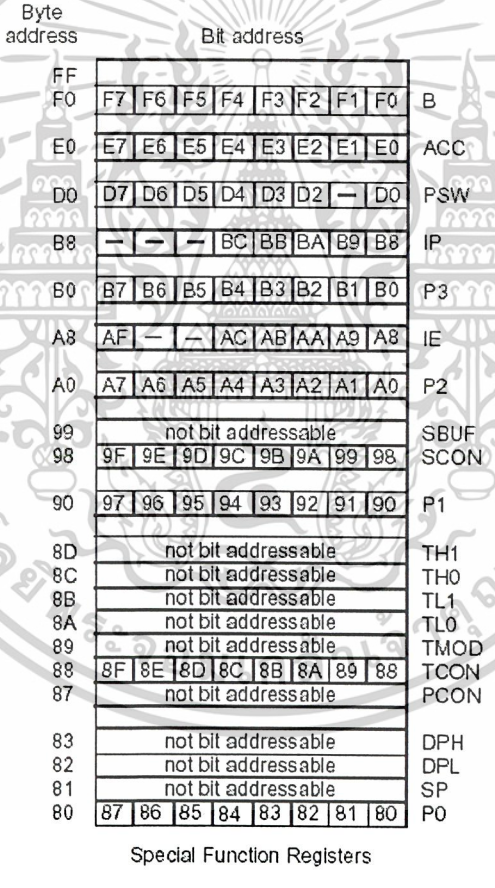
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. แอกคิวมูเลเตอร์ (Accumulator: ACC)

มีขนาด 8 บิต มีแอดเดรสอยู่ที่ตำแหน่ง E0H เป็นรีจีสเตอร์ที่ใช้สำหรับเก็บข้อมูลหรือผลลัพธ์ที่ได้จากการทำงานของไมโครคอนโทรลเลอร์ โดยเฉพาะอย่างยิ่งในการคำนวณทางคณิตศาสตร์และลอจิก ก่อนที่จะส่งข้อมูลหรือผลลัพธ์ที่ได้ให้แก่ซีพียูเพื่อทำการประมวลผลต่อไป อาจเรียกสั้นๆ ว่า รีจีสเตอร์ A หรือ ACC รีจีสเตอร์นี้สามารถเข้าถึงระดับบิตได้

3. รีจีสเตอร์ B (Register B)

มีขนาด 8 บิต มีแอดเดรสอยู่ที่ F0H มีหน้าที่พิเศษคือ หากต้องการคูณหรือหารทางคณิตศาสตร์ ต้องนำข้อมูลที่ต้องการหารหรือคูณมาเก็บไว้ในรีจีสเตอร์ B แล้วจึงกระทำคำสั่งการคูณหรือหารกับค่าในรีจีสเตอร์ A ต่อไป ในกรณีที่ไม่ได้มีความต้องการคูณหรือหารข้อมูล สามารถใช้รีจีสเตอร์ B นี้ในการเก็บข้อมูลทั่วไปได้เหมือนกับรีจีสเตอร์ปกติ และสามารถเข้าถึงในระดับบิตได้เช่นเดียวกับรีจีสเตอร์ A



รูปที่ 2.11 โครงสร้างของหน่วยความจำข้อมูลส่วนบนของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

4.โปรแกรมเคาน์เตอร์ (Program Counter: PC)

มีขนาด 16 บิต มีหน้าที่แจ้งแอดเดรสของหน่วยความจำโปรแกรมในตำแหน่งถัดไปที่ซีพียูต้องไปทำงาน รีจิสเตอร์ PC เป็นรีจิสเตอร์ตัวเดียวที่ไม่ได้จัดสรรไว้รวมกับรีจิสเตอร์ SFR ตัวอื่นๆ การเปลี่ยนแปลงค่าของรีจิสเตอร์ PC จะขึ้นอยู่กับผลของการกระทำคำสั่งภายในหน่วยความจำโปรแกรมที่ผู้เขียนโปรแกรมกำหนด

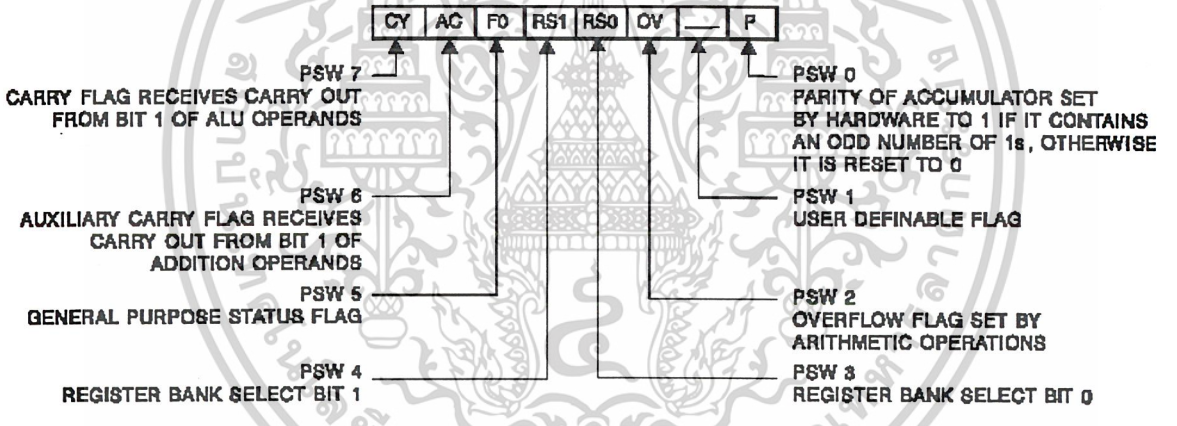
รีจิสเตอร์ PC มีความสำคัญมาก โดยเฉพาะอย่างยิ่งในการตรวจสอบการทำงานของโปรแกรมว่าดำเนินไปตามลำดับขั้นตอนที่กำหนดไว้หรือไม่

5. สแต็กพอยน์เตอร์ (Stack Pointer: SP)

หรือรีจิสเตอร์ตัวชี้สแต็ก มีขนาด 8 บิต มีแอดเดรสอยู่ที่ 81H ใช้ในการหาค่าตำแหน่งของตัวชี้สแต็ก ซึ่งสามารถเปลี่ยนแปลงได้เมื่อซีพียูมีการกระโดดไปทำงานที่โปรแกรมย่อยกลับมายังโปรแกรมหลักเมื่อมีการรีเซตขึ้น ค่าของรีจิสเตอร์ SP จะเท่ากับ 07H ดังนั้นแอดเดรสแรกของพื้นที่ที่สำรองไว้ทำหน้าที่เป็นสแต็กเท่ากับ 08H

6. รีจิสเตอร์ชี้ข้อมูลหรือดาต้าพอยน์เตอร์ (Data Pointer: DPTR)

มีขนาด 16 บิต โดยแบ่งเป็นรีจิสเตอร์ชี้ข้อมูลไบต์สูง (DPH) และรีจิสเตอร์ชี้ข้อมูลไบต์ต่ำ (DPL) แต่ละตัวมีขนาด 8 บิต มีแอดเดรสอยู่ที่ 82H สำหรับ DPL และ 83H สำหรับ DPH รีจิสเตอร์ DPTR นี้ใช้ในการเก็บค่าแอดเดรสของหน่วยความจำหรืออุปกรณ์ภายนอกที่ไมโครคอนโทรลเลอร์ต้องการติดต่อด้วย



รูปที่ 2.12 รายละเอียดของรีจิสเตอร์แสดงสถานะของโปรแกรมหรือ PSW

7. รีจิสเตอร์พอร์ต (Port Register)

เป็นรีจิสเตอร์ขนาด 8 บิต ที่ใช้เก็บข้อมูลของแต่ละพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 มี 4 ตัวคือรีจิสเตอร์พอร์ต 0 หรือ P0 มีแอดเดรสอยู่ที่ 80H, รีจิสเตอร์พอร์ต 1 หรือ P1 มีแอดเดรสอยู่ที่ 90H, รีจิสเตอร์พอร์ต 2 หรือ P2 มีแอดเดรสอยู่ที่ A0H และรีจิสเตอร์พอร์ต 3 หรือ P3 มีแอดเดรสอยู่ที่ B0H รีจิสเตอร์ทุกตัวสามารถเข้าถึงระดับบิต เมื่อต้องการอ่านหรือเขียนข้อมูลออกไปยังพอร์ตของไมโครคอนโทรลเลอร์ จะต้องกระทำผ่านรีจิสเตอร์นี้ทุกครั้ง

8. รีจิสเตอร์บัฟเฟอร์ข้อมูลอนุกรม (Serial Data Buffer: SBUF)

เป็นรีจิสเตอร์ขนาด 8 บิต มีแอดเดรสอยู่ที่ 99H ใช้ในการเก็บข้อมูลที่ส่งออกหรือรับเข้าวงจรสื่อสารอนุกรมที่มีอยู่ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช โดยภายในรีจิสเตอร์ SBUF นี้จะแบ่งออกเป็น 2 ส่วนคือ รีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูล (Transmit Buffer Register) และรีจิสเตอร์บัฟเฟอร์รับข้อมูล (Receive Buffer Register) เมื่อมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
61812

การเขียนข้อมูลมายังรีจีสเตอร์ SBUF ข้อมูลนั้นจะถูกส่งต่อไปยังบัฟเฟอร์สำหรับส่งข้อมูลเพื่อส่งออกจากไมโครคอนโทรลเลอร์ ผ่านทางขา TxD หรือขา P3.1 ในกรณีที่มีการอ่านข้อมูลจากรีจีสเตอร์ SBUF ข้อมูลจะถูกส่งผ่านไปยังรีจีสเตอร์บัฟเฟอร์สำหรับข้อมูลเพื่อส่งไปยังไมโครคอนโทรลเลอร์ ต่อไป สำหรับการรับข้อมูลอนุกรมจากภายนอกนั้นจะผ่านมาทางขา RxD หรือ P3.0 ไมโครคอนโทรลเลอร์ MCS-51

9. รีจีสเตอร์ไทเมอร์ (Timer Register)

เป็นรีจีสเตอร์ขนาด 16 บิต แบ่งเป็นไบต์สูงและไบต์ต่ำเช่นเดียวกับรีจีสเตอร์ DPTR รีจีสเตอร์ไทเมอร์ใช้ในการเก็บค่าของตัวนับหรือเคาน์เตอร์ (Counter) ภายในไมโครคอนโทรลเลอร์เพื่อใช้ในการสร้างฐานเวลา, จับเวลา หรือนับจำนวนพัลส์สัญญาณนาฬิกาภายใน บางทีเรียกรีจีสเตอร์นี้ว่า รีจีสเตอร์ไทเมอร์/เคาน์เตอร์

ในไมโครคอนโทรลเลอร์เบอร์ AT89C51 มีรีจีสเตอร์ไทเมอร์เคาน์เตอร์ 2 ตัว แบ่งเป็น T0 หรือ Timer 0 และ T1 หรือ Timer 1 ในรีจีสเตอร์แบ่งเป็นรีจีสเตอร์ไทเมอร์ไบต์ต่ำ (TL) และรีจีสเตอร์ไทเมอร์สูง (TH) เหมือนกัน โดยรีจีสเตอร์ TL0 มีแอดเดรสอยู่ที่ 8AH รีจีสเตอร์ TH0 มีแอดเดรสอยู่ที่ 8BH ในขณะที่ TL1 และ TH1 มีแอดเดรสอยู่ที่ 8CH และ 8DH สำหรับเบอร์ AT89C52 และในอนุกรม AT89Sxx จะมีรีจีสเตอร์ไทเมอร์/เคาน์เตอร์ถึง 3 ตัว โดยมีรีจีสเตอร์ TL2 และ TH2 ซึ่งมีแอดเดรสอยู่ที่ 0CCH และ 0CDH เพิ่มเติมเข้ามา

10. รีจีสเตอร์แคปเจอร์ (Capture register)

เป็นรีจีสเตอร์ขนาด 16 บิต มีเฉพาะในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชเบอร์ AT89C52 และ ในอนุกรม AT89Sxx เท่านั้น เนื่องจากต้องใช้ร่วมกับไทเมอร์/เคาน์เตอร์ 2 (Timer 2) โดยรีจีสเตอร์แคปเจอร์นี้มีชื่อเรียกย่อว่า รีจีสเตอร์ RCAP2 ซึ่งแบ่งออกเป็นไบต์ต่ำคือ RCAP2L มีแอดเดรสอยู่ที่ 0CAH และไบต์สูงคือ RCAP2H มีแอดเดรสอยู่ที่ 0CBH

รีจีสเตอร์แคปเจอร์จะถูกใช้งานเมื่อกำหนดให้ไทเมอร์ 2 ทำงาน ในโหมดแคปเจอร์ซึ่งเป็นโหมดที่กำหนดให้ไมโครคอนโทรลเลอร์ทำการตรวจสอบการเปลี่ยนแปลงสถานะทางลอจิกที่ขา T2EX ทั้งนี้เพื่อใช้ประโยชน์ในการวัดคาบเวลา ความถี่ และการเปลี่ยนแปลงของสัญญาณพัลส์ที่ขา T2EX

11. รีจีสเตอร์ควบคุม (Control Register)

รีจีสเตอร์ PCON เป็นรีจีสเตอร์ที่เกี่ยวข้องกับการกำหนดอัตรารับส่งข้อมูลของวงจรสื่อสารอนุกรมและกำหนดการทำงานในโหมดประหยัดพลังงานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

รีจีสเตอร์ SCON เป็นรีจีสเตอร์ที่ใช้ในการควบคุมการทำงานของของวงจรสื่อสารอนุกรมภายในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

รีจีสเตอร์ TCON และ T2CON เป็นรีจีสเตอร์ที่ใช้ในการควบคุมการทำงานของไทเมอร์/เคาน์เตอร์ภายในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช โดย T2CON ใช้สำหรับไทเมอร์/เคาน์เตอร์ 2 ของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชเบอร์ AT89C52 1และในอนุกรม AT89Sxx

รีจีสเตอร์ TMOD และ T2MOD เป็นรีจีสเตอร์ที่ใช้กำหนดโหมดหรือลักษณะในการทำงานของไทเมอร์/เคาน์เตอร์ภายในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชโดย T2MOD ใช้สำหรับไทเมอร์/เคาน์เตอร์ 2 ของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชเบอร์ AT89C52 และในอนุกรม AT89Sxx

รีจีสเตอร์ IE และ IP เป็นรีจีสเตอร์ที่เกี่ยวกับการตอบสนองการอินเตอร์รัปต์โดย IE เป็นรีจีสเตอร์สำหรับเอ็นเอเบิลหรือใช้ในการกำหนดลักษณะของการตอบสนองการอินเตอร์รัปต์ ในขณะที่ IP เป็นรีจีสเตอร์สำหรับกำหนดลำดับความสำคัญของการตอบสนองการอินเตอร์รัปต์ว่า จะให้ซีพียูตอบสนองการเกิดอินเตอร์รัปต์ในลักษณะใดก่อนหรือหลัง

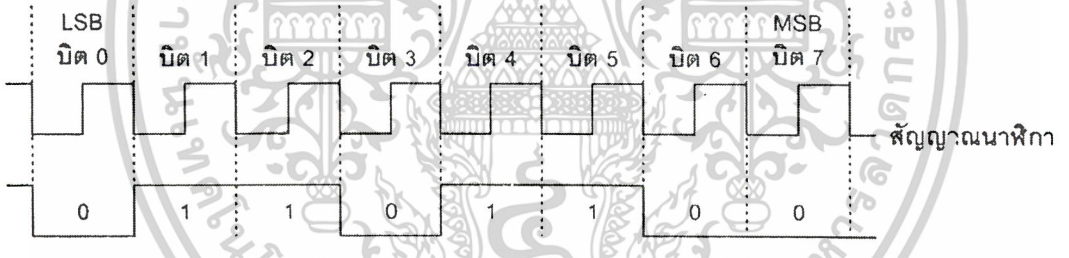
2.2 พอร์ตอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเคลื่อนย้ายข้อมูลจากคอมพิวเตอร์ไปยังอุปกรณ์ต่อพ่วงอื่น ๆ หรือคอมพิวเตอร์ด้วยกันแบบอนุกรม เป็นการรับส่งข้อมูลครั้งละ 1 บิต แต่ก็สามารถรับส่งข้อมูลได้คราวละหลาย ๆ บิตได้หากแต่จะต้องมีการตกลงกันระหว่างตัวส่งและตัวรับว่า จะรับฟังข้อมูลคราวละกี่บิต ตัวรับจะต้องรอข้อมูลมาให้ครบทุกบิตเสียก่อนจึงทำการประมวลผล ส่งผลให้การสื่อสารข้อมูลอนุกรมอาจมีความเร็วต่ำกว่าแบบขนานในด้านจำนวนสายสัญญาณการรับส่งข้อมูลแบบอนุกรมจะใช้จำนวนสายที่น้อยกว่ามาก อย่างน้อยที่สุดให้เพียง 2-3 เส้นเท่านั้น แต่อัตราเร็วในการรับส่งข้อมูลอาจต่ำกว่าแบบขนาน อย่างไรก็ตามการรับส่งข้อมูลแบบอนุกรมสามารถใช้สายสัญญาณที่มีความยาวมากกว่าแบบขนาน ทำให้ระยะทางในการสื่อสารข้อมูลแบบอนุกรมสามารถทำได้มากกว่า

2.2.1 การสื่อสารแบบอนุกรม

การสื่อสารแบบอนุกรมนั้นจะแบ่งออกได้เป็น 2 แบบคือการสื่อสารอนุกรมแบบซิงโครนัสและการสื่อสารอนุกรมแบบอะซิงโครนัส การสื่อสารแบบซิงโครนัสจะมีสัญญาณนาฬิการ่วมอยู่กับการรับและส่งสัญญาณด้วย ตัวอย่างการส่งข้อมูลแบบซิงโครนัสก็คือคีย์บอร์ดของคอมพิวเตอร์ ซึ่งสายเส้นหนึ่งจะเป็นสายของสัญญาณนาฬิกา ส่วนสายอีกเส้นจะเป็นสายของข้อมูล ดังนั้นการติดต่อกันแบบซิงโครนัสนี้จะต้องใช้สายในการเชื่อมต่ออย่างน้อยที่สุด 3 เส้นคือ สัญญาณนาฬิกา ข้อมูลและกราวด์ รูปที่ 2.13 (อรรถพล บุญยะโภคา, วรพจน์ กรแก้ววัฒนกุล และ ชัยวัฒน์ ลิ้มพรจิตรวิไล, 2546) แสดงให้เห็นถึงไทม์มิ่งไดอะแกรมของการส่งข้อมูลแบบซิงโครนัส



รูปที่ 2.13 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรม

2.2.2 การสื่อสารข้อมูลแบบอะซิงโครนัส

การสื่อสารข้อมูลแบบอะซิงโครนัสคือการรับและส่งข้อมูลไปในสายโดยไม่จำเป็นต้องมีสัญญาณนาฬิการ่วมด้วยเหมือนกับการส่งข้อมูลแบบซิงโครนัสแต่จะใช้การกำหนดค่าสัญญาณนาฬิกาทั้งภาครับและภาคส่งให้มีค่าเท่ากัน ซึ่งเรียกสัญญาณนาฬิกาที่ใช้ในการกำหนดค่าให้ภาครับและภาคส่งนี้ว่า อัตราการถ่ายทอดข้อมูล หรือ บอดเรต (Baudrate) มีหน่วยเป็น บิตต่อวินาที (Bit Per Secone : bps)

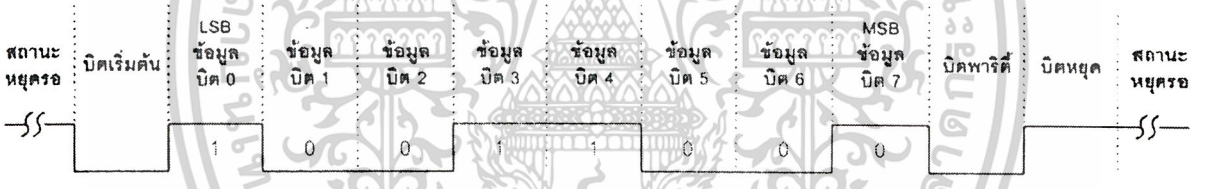
รูปแบบของข้อมูลที่ใช้ในการรับส่งแบบอะซิงโครนัสประกอบด้วย 4 ส่วนด้วยกันคือ

1. บิตเริ่มต้น (Stat Bit) ซึ่งจะมีขนาด 1 บิต
2. บิตข้อมูลแบบอนุกรมจะมีขนาด 5,6,7 หรือ 8 บิต
3. บิตตรวจสอบพาริตี้ (Parity Bit) จะมีขนาด 1 บิตหรือไม่มี
4. บิตปิดท้าย (Stop Bit) จะมีขนาด 1,1.5 หรือ 2 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.14 (อรรถพล บุญยะโภคา, วรพจน์ กรแก้ววัฒนกุล และ ชัยวัฒน์ ลิ้มพรจิตรวิไล, 2546) แสดงรูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัส เมื่อไม่มีข้อมูลที่จะส่ง ขา DATA จะมีสถานะลอจิก "1" ซึ่งจะเรียกสถานะนี้ว่า สถานะหยุดรอ (Waiting Stage) การเริ่มต้นส่งข้อมูลจะเริ่มจากการให้ขา DATA มีลอจิก " 0" ด้วยช่วงระยะเวลา 1 บิต เรียกบิตนี้ว่า บิตเริ่มต้น จากนั้นบิตข้อมูลจะถูกส่งออกไป โดยเริ่มจากบิตที่มีนัยสำคัญต่ำสุด (LSB) ก่อน ซึ่งข้อมูลในไบต์ที่จะส่งอาจจะมีจำนวนบิต 5,6,7 หรือ 8 บิตก็ได้ จากนั้นตามด้วย บิตพาริตี้ ซึ่งใช้เพื่อตรวจสอบความผิดพลาดที่เกิดขึ้นจากการส่งข้อมูล บิตสุดท้ายที่ส่งคือ บิตปิดท้าย ซึ่งจะให้ขาดาต้ามีสถานะลอจิก "1" อีกครั้งด้วยระยะเวลาอย่างน้อย 1 บิต,1.5 บิต หรือ 2 บิต เพื่อเป็นการแสดงว่าสิ้นสุดข้อมูลแล้ว

อุปกรณ์พิเศษที่ได้รับการออกแบบมาสำหรับการรับและส่งข้อมูลแบบอะซิงโครนัสเรียกว่า Universal Asynchronous Receiver/Transmitter หรือ UART อัตราความเร็วในการรับและส่งข้อมูล ของการรับส่งข้อมูลแบบอะซิงโครนัสคือ ค่าบอดเรต ซึ่งก็คือค่าจำนวนบิตต่อวินาทีที่ใช้ในการรับและส่งข้อมูล บอดเรตมาตรฐานที่ใช้สำหรับพอร์ตอนุกรม RS-232 ได้แก่ 110, 150, 300, 600, 1200,2400, 4800, 9600 และ 19200 บิตต่อวินาที และมีค่าเพิ่มมากขึ้นตามเทคโนโลยีของคอมพิวเตอร์ซึ่งการรับส่งแบบอนุกรมโดยไม่ผ่านโมเด็มอาจจะสามารถกำหนดค่าบอดเรตได้สูงถึง 115200 บิตต่อวินาที เนื่องจากบอดเรตคือจำนวนบิตของข้อมูลที่สามารถถ่ายทอดได้ภายใน 1 วินาที ยกตัวอย่าง ข้อมูลอนุกรมถูกส่งในลักษณะ 8 บิต ไม่มีการตรวจสอบพาริตี้ มีบิตเริ่มต้น 1 บิต และบิตปิดท้าย 1 บิต ความยาวของข้อมูลที่รับส่งนี้เท่ากับ 10 บิต ถ้าใช้บอดเรตในการส่งข้อมูลเท่ากับ 9600 บิตต่อวินาที ก็จะสามารถรับส่งข้อมูลได้ด้วยความเร็ว 960 ไบต์ต่อวินาที และถ้ามีการให้พาริตี้ความเร็วในการรับส่งข้อมูลจะเหลือเป็น 872 ไบต์ต่อวินาที



รูปที่ 2.14 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบอะซิงโครนัส

การตรวจสอบพาริตี้สามารถกำหนดให้เป็นแบบคี่ (Odd) แบบคู่ (Even) หรือไม่มีการตรวจสอบพาริตี้ก็ได้ การตรวจสอบพาริตี้เป็นการตรวจสอบจำนวนรวมของบิตที่เป็นลอจิก "1 " ภายในข้อมูลที่ส่งไป 1 ไบต์ว่ามีจำนวนรวมเป็นเลขคู่หรือเลขคี่โดยต้องรวมบิตพาริตี้เข้าไปด้วย ยกตัวอย่าง ข้อมูลที่จะทำการส่งมีขนาด 8 บิตและมีค่าเท่ากับ 99 ฐานสิบหก หรือ 10011001 ฐานสองจะเห็นว่าข้อมูลในไบต์นี้มีจำนวนลอจิก "1" จำนวน 4 ตัวซึ่งเป็นเลขคู่ ดังนั้นถ้ากำหนดค่าพาริตี้เป็นคู่ ค่าในบิตพาริตี้ จะต้องมีลอจิกเป็น "0" แต่ถ้าพาริตี้เป็นคี่ ค่าที่บิตพาริตี้จะต้องเป็น "1" เพื่อให้ข้อมูล 1 ไบต์รวมทั้งบิตพาริตี้มีจำนวนบิตที่เป็นลอจิก "1" รวมกันเป็นเลขคี่ในตารางที่ 2.1 แสดงตัวอย่างของบิตพาริตี้ในการรับส่งข้อมูลอนุกรม

บิตพาริตี้ถูกสร้างขึ้นจากภาคส่งข้อมูลของ UART โดยภาครับจะต้องกำหนดคุณสมบัติการตรวจสอบพาริตี้ให้ตรงกันว่า จะตรวจสอบพาริตี้คี่หรือคู่ จากนั้นภาครับของ UART จะตรวจสอบ ค่าพาริตี้ที่เกิดขึ้นว่าเป็นคู่หรือเป็นคี่ โดยการนับจำนวนลอจิก "1" ทั้งหมดรวมทั้งบิตพาริตี้ด้วย ถ้ากำหนดพาริตี้ไว้เป็นคู่แต่อ่านค่าตัวเลขในการนับออกมาได้ตัวเลขเป็นคี่ ทางภาครับจะแจ้งข้อผิดพลาดให้ผู้ใช้ทราบ นับเป็นการตรวจสอบความผิดพลาดของการถ่ายทอดข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ง่ายที่สุด แต่จะเชื่อถือได้เมื่อมีบิตข้อมูลที่ทำการส่งผิดพลาดเพียงบิตเดียวเท่านั้น ถ้าข้อมูลที่ส่งมีบิตที่ผิดพลาดมากกว่า 1 บิต การตรวจสอบด้วยวิธีนี้จะไม่ได้ผล สำหรับการตั้งพาริตี้บิตเป็น NONE นั้นทั้งภาครับ และภาคส่ง จะไม่มีการตรวจสอบพาริตี้

ตารางที่ 2.1 แสดงบิตพาริตี้ของข้อมูล

ข้อมูล	บิตพาริตี้คู่	บิตพาริตี้คี่
00000000	0	1
00000001	1	0
00000010	1	0
00000011	0	1
00000100	1	0
11111110	1	0
11111111	0	1

ที่มา: อรรถพล บุญยะโภคา, วรพจน์ กรแก้ววัฒนกุล และ ชัยวัฒน์ ลิ้มพรจิตรวิไล (2546)

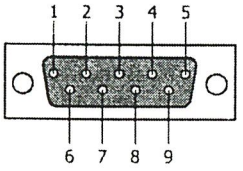
2.2.3 มาตรฐานพอร์ตอนุกรมแบบ RS-232

มาตรฐานพอร์ตอนุกรมแบบ RS-232 เป็นมาตรฐานที่ออกแบบมาเพื่อใช้ส่งข้อมูลแบบอะซิงโครนัส 2 ทิศทาง โดยมาตรฐาน RS-232 ในอดีตถูกออกแบบมาเพื่อการส่งข้อมูลจากคอมพิวเตอร์ไปยังโมเด็มเพียงอย่างเดียว เพื่อที่จะนำข้อมูลจากโมเด็มนี้สื่อสารผ่านสายโทรศัพท์ไปยังคอมพิวเตอร์อีกชุดหนึ่งที่อยู่ห่างไกลกัน โดยคณะกรรมการที่เรียกว่าสมาคมอุตสาหกรรมอิเล็คทรอนิคส์ (Electronic Industrial Association : EIA) ได้วางมาตรฐานที่มีชื่อเรียกว่า EIA RS-232 มาตรฐานนี้ช่วงแรกใช้คอนเน็คเตอร์แบบ DB-25 โดยกำหนดความยาวสูงสุดของสายสัญญาณไว้ที่ 50 ฟุต มีระดับสัญาณตั้งแต่ -3 ถึง -12 V แสดงว่ามีข้อมูล (Mark) และ +3 ถึง +12 V แสดงว่าเป็นช่องว่าง (Space)

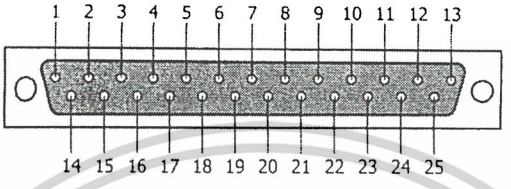
มาตรฐาน RS-232 ได้กำหนดรูปแบบของอุปกรณ์การเชื่อมต่อข้อมูล (Data Terminal Equipment: DTE) กับวงจรข้อมูลปลายทาง (Data Circuit Terminating: DCE) ได้ว่า อุปกรณ์ DTE จะต้องเป็นอุปกรณ์ประมวลผลในตัวเช่น ไมโครคอนโทรลเลอร์หรือไมโครคอมพิวเตอร์ซึ่งมีความสามารถสร้างข้อมูลแบบอนุกรมได้ ส่วนอุปกรณ์ DCE จะทำหน้าที่เป็นตัวรับข้อมูลที่ส่งมาจาก DTE เท่านั้น โดยการรับส่งข้อมูลระหว่างอุปกรณ์ทั้งสองจะกระทำผ่านมาตรฐาน RS-232

ข้อแตกต่างของอุปกรณ์ DTE และอุปกรณ์ DCE ที่เห็นได้ชัดคือ คอนเน็คเตอร์ของ DCE จะเป็นตัวเมีย ซึ่งพอร์ตอนุกรมของคอมพิวเตอร์ที่ใช้กันทั่วไปจะเป็นแบบ DTE ส่วนคอนเน็คเตอร์ที่อยู่ที่โมเด็มจะเป็นแบบ DCE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 คอนเน็กเตอร์อนุกรม 9 ขาหรือแบบ DB-9 (มองจากด้านหลังคอมพิวเตอร์)



รูปที่ 2.16 คอนเน็กเตอร์อนุกรม 25 ขาหรือแบบ DB-25 (มองจากด้านหลังคอมพิวเตอร์)

ตารางที่ 2.2 การจัดขาของคอนเน็กเตอร์พอร์ตอนุกรมตามมาตรฐาน RS-232 ทั้งแบบ DB-9 และแบบ DB-25

คอนเน็คเตอร์ DB-9	คอนเน็คเตอร์ DB-25	ชื่อของสายสัญญาณ	ชนิดของสายสัญญาณ
1	8	Data Carrier Detect	อินพุต
2	3	Receive Data :RxD	อินพุต
3	2	Transmitted Data : TxD	เอาต์พุต
4	20	Data Terminal Ready : DTR	เอาต์พุต
5	7	Signal Ground : GND	-
6	6	Data Set Ready : DSR	อินพุต
7	4	Request To Send : RTS	เอาต์พุต
8	5	Clear To Send : CTS	อินพุต
9	22	Ring Indicator : RI	อินพุต

ที่มา: อรรถพล บุญยะโภคา, วรพจน์ กรแก้ววัฒนกุล และ ชัยวัฒน์ ลิ้มพรจิตรวิไล (2546)

2.2.4 คอนเน็คเตอร์สำหรับพอร์ต RS-232 และการเชื่อมต่อ

มาตรฐานการเชื่อมต่อแบบ RS-232 จะใช้คอนเน็คเตอร์แบบ DB-25 ตัวผู้หรือ DB-9 ตัวผู้ซึ่งคอนเน็คเตอร์แบบ DB-25 จะมาต่อเพียง 9 เส้น เช่นเดียวกับคอนเน็คเตอร์แบบ DB-9 เนื่องจากขาอื่นๆที่เคยใช้งานในอดีต ปัจจุบันมีการใช้งานไม่มากนัก จึงถูกยกเลิกไป โดยแสดงรูปร่างและตำแหน่งขาใน รูปที่ 2.15 (อรรถพล บุญยะโภคา, วรพจน์ กร

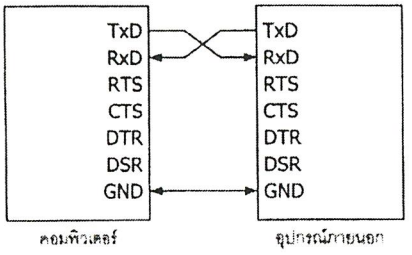
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แก้ววัฒนกุล และ ชัยวัฒน์ ลิ้มพรจิตรวิไล, 2546) และ รูปที่ 2.16 (อรรถพล บุญยะโภคา, วรพจน์ กรแก้ววัฒนกุล และ ชัยวัฒน์ ลิ้มพรจิตรวิไล, 2546)

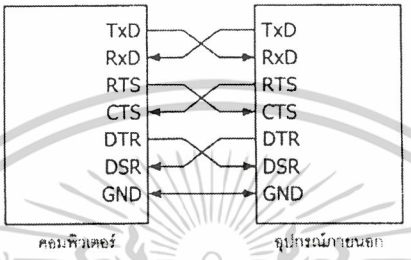
สำหรับการเชื่อมต่อกับอุปกรณ์ภายนอกแสดงดังในรูปที่ 2.17 (อรรถพล บุญยะโภคา, วรพจน์ กรแก้ววัฒนกุล และ ชัยวัฒน์ ลิ้มพรจิตรวิไล, 2546) เป็นการเชื่อมต่อแบบ Null Modem หรือการเชื่อมต่อโดยตรงโดยไม่ต้องผ่านโมเด็ม โดยมีการตรวจสอบหรือแฮนด์เช็กเต็มรูปแบบ ส่วนรูปที่ 2.18 (อรรถพล บุญยะโภคา, วรพจน์ กรแก้ววัฒนกุล และ ชัยวัฒน์ ลิ้มพรจิตรวิไล, 2546) เป็นการเชื่อมต่อแบบ Null Modem ในลักษณะที่มีการใช้สายสัญญาณเพียง 3 เส้น โดยเส้นหนึ่งสำหรับส่งข้อมูล อีกเส้นสำหรับรับข้อมูล และเส้นสุดท้ายเป็นกราวด์ สำหรับรายละเอียดการทำงานของแต่ละขาของพอร์ตอนุกรม มีดังนี้

- Data Carrier Detect : DCD หรือที่เรียกว่า Carrier Detect : CD ขานี้จะแอ็คทีฟเมื่อมีการส่งสัญญาณพัลส์จากอุปกรณ์สื่อสารข้อมูลเช่น โมเด็มสำหรับการใช้งานปกติขานี้จะไม่ได้ถูกใช้งานมากนัก
- Receive Data :RD หรือ RxD ขานี้ใช้เพื่อรับสัญญาณอนุกรมเข้ามายังคอมพิวเตอร์ โดยนำข้อมูลมาเก็บไว้ในรีจิสเตอร์บัฟเฟอร์
- Transmitted Data : TD หรือ TxD ใช้ส่งข้อมูลจากคอมพิวเตอร์ โดยนำข้อมูลออกจากคอมพิวเตอร์โดยนำข้อมูลที่เก็บไว้ในรีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูลออกไป
- Data Terminal Ready : DTR เป็นขาสัญญาณที่ส่งออกจากคอมพิวเตอร์ เพื่อให้อุปกรณ์ปลายทางรับรู้ว่า ต้องการติดต่อด้วยโดยขา DTR นี้ต้องเชื่อมต่อขา DSR ของอุปกรณ์ปลายทางต้องเชื่อมต่อกับขา DSR ของคอมพิวเตอร์ถ้าใช้การเชื่อมต่อแบบ Null Modem ซึ่งมีการใช้สายสัญญาณเพียง 3 เส้นจะต้องต่อขา DTR ของตัวมันเองเข้าด้วยกันและต้องต่อกับขา DCD ด้วยในกรณีที่โปรแกรมสื่อสารมีการใช้สัญญาณพัลส์
- Signal Ground : GND กราวด์ระบบ
- Data Set Ready : DSR ขานี้จะใช้คู่กับขา DTR เพื่อตรวจสอบการเชื่อมต่อกันระหว่างอุปกรณ์ปลายทาง ซึ่งขาสำหรับรับข้อมูลจากภายนอกซึ่งถูกส่งมาจากขา DTR
- Request To Send : RTS เป็นขาสำหรับส่งสัญญาณร้องขอให้กับทางอุปกรณ์ปลายทางส่งข้อมูลย้อนกลับมายังคอมพิวเตอร์ โดยขาที่รับสัญญาณ RTS ก็คือขา CTS ในกรณีที่เชื่อมต่อแบบ Null Modem 3 สาย จะต้องเชื่อมต่อขา RTS กับ CTS ของตัวมันเองเข้าด้วยกัน เพื่อจะให้การรับและการส่งข้อมูลสามารถเกิดขึ้นได้ตลอดเวลา
- Clear To Send : CTS ขานี้จะคอยรับสัญญาณจากขา RTS เมื่อรับสัญญาณได้ ข้อมูลที่ TxD จะถูกส่งออกไป ดังนั้นขานี้จึงถูกใช้เพื่อตรวจสอบอุปกรณ์ต่อพ่วงว่าพร้อมที่จะรับข้อมูลหรือไม่
- Ring Indicator : RI ใช้แสดงสถานะสัญญาณเรียกจากสายโทรศัพท์ ปกติในการสื่อสารโดยทั่วไปสายนี้จะไม่ถูกใช้งาน จะใช้งานก็ต่อเมื่อเชื่อมต่อกับโมเด็มและโปรแกรมการตรวจสอบสัญญาณเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.17 การอุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์แบบ Null modem



รูปที่ 2.18 การอุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์แบบ RS-232 โดยใช้สายสัญญาณเพียง 3 เส้น

2.2.5 UART

UART (Universal AsynchronouS Receiver Transmitter) หมายถึงอุปกรณ์ที่ทำหน้าที่รับและส่งข้อมูลแบบอะซิงโครนัสนั่นเอง สำหรับการสื่อสารอนุกรมบนคอมพิวเตอร์แล้ว UART ถือว่าเป็นหัวใจสำคัญของการสื่อสารอนุกรม

หน้าที่หลักของ UART คือทำหน้าที่แปลงข้อมูลที่อยู่ในรูปแบบขนานจากคอมพิวเตอร์ให้อยู่ในรูปแบบอนุกรมแบบอะซิงโครนัส แล้วส่งออกไป และทำหน้าที่แปลงสัญญาณอนุกรมแบบอะซิงโครนัสที่ป้อนเข้ามายัง UART ให้เป็นแบบขนานก่อนที่จะส่งเข้าสู่คอมพิวเตอร์ ซึ่งนอกจากUART จะส่งข้อมูลไปยังคอมพิวเตอร์แล้ว ยังแจ้งข้อมูลอื่น ๆ ให้คอมพิวเตอร์รับทราบด้วย เช่น อัตราเร็วในการรับส่งข้อมูล (บอดเรต) รูปแบบการส่งข้อมูล ความผิดพลาดที่เกิดขึ้นระหว่างการถ่ายทอดข้อมูล (ผิดพลาดจากพาริตี้เฟรมข้อมูลโอเวอร์รัน) เป็นต้น

ภายใน UART จะมีส่วนของวงจรสร้างบอดเรตแบบโปรแกรมได้ (Programmable Buadrate Generator) โดยการกำหนดค่าตัวหารให้กับสัญญาณนาฬิกาของ UART โดยตัวหารนี้มีขนาด 16 บิต ดังนั้นจึงสามารถกำหนดตัวหารอยู่ในช่วง 1- 65,535 UART สามารถรับส่งข้อมูลได้ทั้งแบบฮาล์ฟดูเพล็กซ์ (Half Duplex) และฟูลดูเพล็กซ์ (Full Duplex) โดยการส่งแบบฮาล์ฟดูเพล็กซ์เป็นการส่งแบบทิศทางเดียว ส่วนการส่งแบบฟูลดูเพล็กซ์นั้นสามารถรับและส่งข้อมูลได้ในคราวเดียวกัน

ชนิดของ UART

ในเครื่องคอมพิวเตอร์ทั่วไปมี UART ที่ใช้งานกันอยู่ 2 เบอร์คือ 8250 ซึ่งเป็น UART มาตรฐานที่มีใช้กันมายาวนาน UART เบอร์นี้จะมีบัฟเฟอร์สำหรับรับและส่งข้อมูลตำแหน่งเดียวกัน ทำให้การรับและส่งข้อมูลถูกจำกัดความเร็วอยู่ที่ 57.6 กิโลบิตต่อวินาทีเท่านั้น แต่ UART เบอร์นี้ก็ถือว่าเป็นต้นแบบของ UART ที่ใช้ในคอมพิวเตอร์ โดยคอมพิวเตอร์ทุกๆ รุ่นจะต้องสนับสนุนการทำงานตามรูปแบบของ UART เบอร์นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

UART อีกเบอร์หนึ่งคือ 16450 มีความสามารถรับส่งข้อมูลได้ที่ความเร็ว 115,200 บิตต่อวินาที และเพิ่มรีจิสเตอร์สำหรับพักข้อมูลสำหรับ UART นอกจากนั้นยังเพิ่มส่วนของชิฟต์รีจิสเตอร์แบบ FIFO (First In First Out) ขนาด 16 ไบต์เข้าไป ทำให้สามารถสนับสนุนความเร็วในการรับส่งข้อมูลที่ 256 กิโลบิตต่อวินาทีได้ โดยคอมพิวเตอร์ในปัจจุบันใช้ ARFI เบอร์นี้หรือใหม่กว่า เช่น เบอร์ TLI6C750 ซึ่งมีรีจิสเตอร์แบบ FIFO ขนาด 64 ไบต์ ทำงานได้ที่ระดับเร็วละ +5V และ +3V มีโหมดประหยัดพลังงาน สามารถรับส่งข้อมูลได้ที่ความเร็ว 1 เมกะบิตต่อวินาทีเมื่อใช้สัญญาณนาฬิกา 16 MHZ

อย่างไรก็ตาม ความเร็วในการส่งข้อมูลที่มากมายของ UART เบอร์ใหม่ ๆ ก็ไม่ได้ช่วยให้การรับส่งข้อมูลของคอมพิวเตอร์เร็วขึ้น เนื่องจากว่าคอมพิวเตอร์ยังให้ความถี่ของสัญญาณนาฬิกาในการแปลงข้อมูลเพียง 1.8432 MHZ เท่านั้น

2.2.6 วงจรภายในและรีจิสเตอร์ของพอร์ตอนุกรม RS-232

เครื่องคอมพิวเตอร์โดยทั่วไปสามารถต่อพอร์ตอนุกรมสูงสุดได้ 4 พอร์ต มีชื่อเรียกเป็น COM1,COM2,COM 3 และ COM4 ซึ่งพอร์ตอนุกรมแต่ละตัวต่างก็ใช้งาน UART ภายในคอมพิวเตอร์ในการติดต่อกับอุปกรณ์ภายนอกเช่นเดียวกัน ในรูปที่ 1-5 แสดงผังการทำงานภายในของพอร์ตอนุกรมประกอบด้วยรีจิสเตอร์ 8 บิต 8 ตัวที่ใช้งานร่วมกับ UART แอดเดรสของรีจิสเตอร์ภายในพอร์ตอนุกรมสามารถคำนวณได้จากค่ารีจิสเตอร์พื้นฐานของพอร์ตอนุกรม ยกตัวอย่าง พอร์ตอนุกรม COM1 มีแอดเดรสอยู่ที่ 3F8H ตำแหน่งของรีจิสเตอร์ต่าง ๆ จะเป็นตำแหน่งที่บวกเข้าไปกับค่า 3F8H โดยรีจิสเตอร์ที่ใช้งานกับพอร์ตอนุกรมมีดังนี้

00H เป็นรีจิสเตอร์บัฟเฟอร์สำหรับเก็บข้อมูลที่รับเข้ามาหรือเตรียมข้อมูลก่อนที่จะส่งออกไป

01H รีจิสเตอร์เอ็นเอเบิลการอินเตอร์รัปต์ ใช้เซตโหมดการอินเตอร์รัปต์ของพอร์ตอนุกรม

02H รีจิสเตอร์แสดงโหมดการอินเตอร์รัปต์ ใช้เพื่อตรวจสอบโหมดของการอินเตอร์รัปต์

03H รีจิสเตอร์กำหนดรูปแบบของข้อมูล

04H รีจิสเตอร์ควบคุมโมเด็ม ใช้ตรวจสอบบิตสำหรับติดต่อกับโมเด็ม เช่น RTS หรือ DTR

05H รีจิสเตอร์แสดงสถานะการรับและการส่งข้อมูลแบบอนุกรม

06H รีจิสเตอร์แสดงสถานะของโมเด็ม ซึ่งจะแสดงสถานะของขา DCD,RI,DSR และ CTS

07H รีจิสเตอร์สำหรับการเก็บข้อมูลชั่วคราว

2.2.6.1 <u>รีจิตเตอร์ตำแหน่ง 00H : รีจิสเตอร์บัฟเฟอร์</u>

เป็นรีจิสเตอร์เก็บข้อมูลที่รับเข้ามาและส่งออกโดยการติดต่อกับรีจิสเตอร์นี้เพื่อเก็บข้อมูลจะต้องกำหนดให้บิต DLAB ในรีจิสเตอร์กำหนดรูปแบบข้อมูล (03H) มีสถานะเป็น "0" ซึ่งการเขียนข้อมูลมายังแอดเดรสนี้ เป็นการส่งข้อมูลไปยังรีจิสเตอร์ส่งข้อมูลและข้อมูลจะถูกส่งออกไปแบบอนุกรม สำหรับการรับข้อมูล เมื่อรับเข้ามาแล้ว จะส่งต่อไปยังรีจิสเตอร์เก็บข้อมูล หลังจากอ่านค่าจากรีจิสเตอร์นี้ออกไป รีจิสเตอร์นี้จะถูกเคลียร์ และเตรียมพร้อมสำหรับรับข้อมูลในไบต์ต่อไป

2.2.6.2 <u>รีจิสเตอร์ตำแหน่ง 01H : รีจิสเตอร์เอ็นเอเบิลการอินเตอร์รัปต์</u>

เป็นรีจิสเตอร์สำหรับการเอ็นเอเบิลการอินเตอร์รัปต์ ซึ่งเป็นการกำหนดให้ UART สร้างสัญญาณอินเตอร์รัปต์ขึ้นมา ฟังก์ชันการทำงานในแต่ละบิตของรีจิสเตอร์นี้มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	0	0	0	SINP	ERBK	TBE	RxRD

บิต 4-7 บิตเหล่านี้ไม่ถูกใช้งาน กำหนดให้เท่ากับ “0”

SINP เอ็นเอเบิลการอินเตอร์รัปต์เนื่องจากเกิดการเปลี่ยนสถานะที่ขาอินพุต CTS,DSR,DCD หรือขา RI

“1” เอ็นเอเบิลการอินเตอร์รัปต์

“0” ดิสเอเบิล

ERBK เอ็นเอเบิลการอินเตอร์รัปต์เนื่องจากเกิดความผิดพลาดขึ้นด้วยสาเหตุจาก พาริตี้โอเวอร์รัน เฟรมข้อมูล หรือการเบรกข้อมูล

“1” เอ็นเอเบิลการอินเตอร์รัปต์

“0” ดิสเอเบิล

TBE เอ็นเอเบิลการอินเตอร์รัปต์เมื่อรีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูลว่าง

“1” เอ็นเอเบิลการอินเตอร์รัปต์

“0” ดิสเอเบิล

RxRD เอ็นเอเบิลการอินเตอร์รัปต์เนื่องจากรีจิสเตอร์บัฟเฟอร์ได้รับข้อมูลแล้ว

“1” เอ็นเอเบิลการอินเตอร์รัปต์

“0” ดิสเอเบิล

2.2.6.3 <u>รีจิสเตอร์ตำแหน่ง 02H : รีจิสเตอร์แสดงโหมดและสถานะการอินเตอร์รัปต์</u>

มีรายละเอียดของแต่ละบิตดังนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	0	0	0	0	ID1	ID0	PND

บิต 3-7 ไม่ได้ใช้งาน อ่านค่าได้เท่ากับ “0”

ID1,IDO ใช้งานร่วมกันเพื่อแจ้งสาเหตุของการเกิดอินเตอร์รัปต์

“00” เกิดการอินเตอร์รัปต์เนื่องจากการเปลี่ยนแปลงของขาอินพุตขึ้นการอินเตอร์รัปต์แบบนี้มีนัยสำคัญเป็นอันดับ 4

“01” เกิดการอินเตอร์รัปต์เนื่องจากรีจิสเตอร์บัฟเฟอร์ส่งข้อมูลว่างขึ้นการอินเตอร์รัปต์แบบนี้มีนัยสำคัญเป็นอันดับ 3

“10” เกิดการอินเตอร์รัปต์เนื่องจากข้อมูลถูกเก็บลงในรีจิสเตอร์บัฟเฟอร์สำหรับรับข้อมูลเรียบร้อยแล้ว การอินเตอร์รัปต์แบบนี้มีนัยสำคัญเป็นอันดับ 2

“11” เกิดการอินเตอร์รัปต์เนื่องจากความผิดพลาดในการถ่ายทอดข้อมูล หรือเกิดการเบรก (Break : เกิดการหยุดถ่ายทอดข้อมูลกระทันหัน) การอินเตอร์รัปต์แบบนี้มีนัยสำคัญเป็นอันดับ 1 หรือมีนัยสำคัญสูงสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PND ใช้แสดงสถานะของการเกิดอินเตอร์รัปต์

"1" แสดงว่าไม่มีการอินเตอร์รัปต์

"0" แสดงว่ามีการอินเตอร์รัปต์เกิดขึ้น

เมื่อมีการสร้างสัญญาณอินเตอร์รัปต์ขึ้น จะต้องมีการเคลียร์ค่าก่อนที่จะให้เกิดอินเตอร์รัปต์ครั้งต่อไป โดยสามารถทำได้ดังนี้คือ

- ถ้าเกิดอินเตอร์รัปต์เนื่องจากการเปลี่ยนแปลงของขาอินพุตจะต้องอ่านค่าจากรีจิสเตอร์แสดงสถานะของโมเด็ม (รีจิสเตอร์ตำแหน่ง 06H) เพื่อเคลียร์ค่าการอินเตอร์รัปต์
- ถ้าเกิดการอินเตอร์รัปต์เนื่องจากบัฟเฟอร์ส่งข้อมูลว่าง จะต้องเขียนข้อมูลไปยังรีจิสเตอร์บัฟเฟอร์ส่งข้อมูล (รีจิสเตอร์ตำแหน่ง 00H) หรืออ่านค่ารีจิสเตอร์แสดงสถานะอินเตอร์รัปต์(รีจิสเตอร์ตำแหน่ง 02H) เพื่อเคลียร์ค่าการอินเตอร์รัปต์
- ถ้าเกิดอินเตอร์รัปต์เนื่องจากการเก็บข้อมูลลงในรีจิสเตอร์บัฟเฟอร์สำหรับรับข้อมูลเรียบร้อย จะต้องเคลียร์ค่าอินเตอร์รัปต์โดยการอ่านข้อมูลจากรีจิสเตอร์บัฟเฟอร์
- ถ้าเกิดอินเตอร์รัปต์เนื่องจากความผิดพลาดในการรับส่งข้อมูลหรือเกิดการเบรก จะต้องเคลียร์ค่าอินเตอร์รัปต์โดยการอ่านค่ารีจิสเตอร์แสดงสถานะการรับและส่งข้อมูลแบบอนุกรม

2.2.6.4 <u>รีจิสเตอร์ตำแหน่ง 03H : รีจิสเตอร์กำหนดรูปแบบของข้อมูล</u>

มีรายละเอียดหน้าที่ของแต่ละบิตดังนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
DLAB	**BRK**	**BRK2**	**BRK1**	**BRK0**	**STOP**	**DAB1**	**DAB0**

DLAB ใช้ในการกำหนดหน้าที่การทำงานของรีจิสเตอร์บัฟเฟอร์ (00H)

"1" เป็นการเข้าสู่โหมดการหารค่าบอดเรต

"0" เป็นการเข้าถึงรีจิสเตอร์บัฟเฟอร์ (รีจิสเตอร์ตำแหน่ง 00H) และรีจิสเตอร์สำหรับเอ็นเอเบิลการอินเตอร์รัปต์ (รีจิสเตอร์ตำแหน่ง 01H) เมื่อ บิต DLAB เป็น"1" รีจิสเตอร์บัฟเฟอร์ (00H) และรีจิสเตอร์เอ็นเอเบิล การอินเตอร์รัปต์ (01H) จะใช้สำหรับโหลดค่าการหารความถี่สำหรับกำหนด ค่าบอดเรต โดยรีจิสเตอร์ 00H เก็บค่าตัวหารไบต์ต่ำ ส่วนรีจิสเตอร์ 01H ใช้เก็บค่าตัวหารไบต์สูง การหาค่าบอดเรตสามารถเขียนเป็นสมการได้ดังนี้

บอดเรต = 1 15200/ค่าตัวหาร 16 บิต

ค่าตัวเลข 115200 มาจากความถี่ของคริสตอลในวงจร UART ภายในเครื่องคอมพิวเตอร์ โดยคริสตอลที่ใช้มีความถี่ 1.8432 MHZ วงจรภายใน UART จะหารค่าความถี่นี้ด้วย 16 ทำให้ได้ค่าความถี่ 115200 Hz ออกมา

ค่าตัวหาร 16 บิต = ข้อมูลในรีจิสเตอร์ 00H + (256 x ข้อมูลในรีจิสเตอร 0IH)

ถ้าต้องการบอดเรตเท่ากับ 9600 ค่าตัวหารที่ใช้จะต้องมีค่าเท่ากับ 12 ซึ่ง ค่านี้จะถูกเขียนลงในรีจิสเตอร์ 00H และเขียนค่า 0 ลงไปในรีจิสเตอร์ 01H ค่าตัวหารที่ทำให้เกิดค่าบอดเรตสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สุดที่ 115200 บิตต่อวินาทีคือ ค่า 0001 นั่นคือรีจิสเตอร์ 00H มีค่าเท่ากับ 1 และรีจิสเตอร์ 01H มีค่าเท่ากับ 0

BRK ใช้ควบคุมการหยุดถ่ายทอดข้อมูล

"1" สามารถหยุดหรือเบรกได้

"0" ไม่มีการหยุดหรือเบรกได้

PAR2,PAR1,PAR0 ใช้เพื่อกำหนดบิตพาริตี้

"000" ไม่ใช้บิตพาริตี้

"001" กำหนดพาริตี้คี่

"011" กำหนดพาริตี้คู่

"101" มาร์ก (Mark)

"111" ช่องว่าง (Space)

STOP ให้กำหนดจำนวนบิตปิดท้าย

"1" มีบิตปิดท้าย 2 บิต

"0" มีบิตปิดท้าย 1 บิต

DAB1,DAB0 ใช้ร่วมกันในการกำหนดจำนวนบิตของข้อมูลที่ต้องการถ่ายทอด

"00" จำนวนบิตข้อมูลเท่ากับ 5 บิต

"01" จำนวนบิตข้อมูลเท่ากับ 6 บิต

"10" จำนวนบิตข้อมูลเท่ากับ 7 บิต

"11" จำนวนบิตข้อมูลเท่ากับ 8 บิต

2.2.6.5 <u>รีจิสเตอร์ตำแหน่ง 04H : รีจิสเตอร์ควบคุมโมเด็ม</u>

มีรายละเอียดหน้าที่ของแต่ละบิตดังนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	0	0	LOOP	OUT2	OUT1	RTS	DTR

บิต 5-7 ไม่มีการใช้งาน อ่านค่าได้เท่ากับ 0

LOOP "1" เอ็นเอเบิลการส่งค่ากลับ

"0" ดิสเอเบิล

OUT1,OUT2 "1" เอ็นเอเบิลการใช้งานภายใน

"0" ดิสเอเบิล

RTS ใช้ควบคุมการทำงานของขา RTS (Ready To Send)

"1" เอ็นเอเบิล

"0" ดิสเอเบิล

DTR ใช้ควบคุมการทำงานของขา DTR (Data Teminal Ready)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

"1" เอ็นเอเบิล

"0" ดิสเอเบิล

2.2.6.6 รีจิสเตอร์ตำแหน่ง 05H : รีจิสเตอร์แสดงสถานะการรับและส่งข้อมูลอนุกรมของ UART

ใช้งานร่วมกับรีจิสเตอร์แสดงโหมดและสถานะของการอินเตอร์รัปต์ (รีจิสเตอร์ตำแหน่ง02H) เพื่อแสดงสาเหตุของการเกิดอินเตอร์รัปต์ มีรายละเอียดหน้าที่ของแต่ละบิตดังนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	TXE	TBE	BREK	FRME	PARE	OVRE	RxRD

TXE (Transmitter Empty)

"1" แสดงว่ารีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูลว่าง

"0" แสดงว่ายังคงมีข้อมูล 1 ไบต์เก็บอยู่ในรีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูล

TBE (Transmitter Buffer Empty)

"1" รีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูลว่าง

"0" ยังคงมีข้อมูล 1 ไบต์เก็บอยู่ในรีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูล

BREK (Break)

"1" UART ตรวจพบการเบรก

"0" ไม่มีการเบรก

FRME (Frame Error)

"1" UART ตรวจพบความผิดพลาดด้านเฟรมข้อมูล

"0" ไม่พบความผิดพลาดด้านเฟรมข้อมูล

PARE (Parity Error)

"1" UART ตรวจพบความผิดพลาดทางพาริตี้

"0" ไม่พบความผิดพลาดทางพาริตี้

OVRE (Overrun Error)

"1" UART ตรวจพบความผิดพลาดแบบโอเวอร์รัน

"0" ไม่พบความผิดพลาดแบบโอเวอร์รัน

RxRD (Received Data Ready)

"1" มีการรับข้อมูลเข้ามาเก็บไว้ในบัฟเฟอร์

"0" ไม่มีข้อมูล

2.2.6.7 รีจิสเตอร์ตำแหน่ง 06H : รีจิสเตอร์แสดงสถานะของโมเด็ม

ใช้เพื่อกำหนดสถานะสัญญาณอินพุต ของพอร์ตอนุกรม RS-232 ซึ่งได้แก่ สัญญาณ DCD,DSR,CTS และ RI สำหรับการเชื่อมต่อใช้งานแบบอเนกประสงค์ ดังมีรายละเอียดหน้าที่ของแต่ละ บิต ต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
DCD	RI	DSR	CTS	DDCD	DRI	DDSR	DCTS

DCD ใช้แสดงสภาวะของขา DCD

"1" แสดงว่าที่ขา DCD เป็นลอจิก "1"

"0" แสดงว่าที่ขา DCD เป็นลอจิก "0"

RI ใช้แสดงสภาวะของขา RI

"1" แสดงว่าที่ขา RI เป็นลอจิก "1"

"0" แสดงว่าที่ขา RI เป็นลอจิก "0"

DSR ใช้แสดงสภาวะของขา DSR

"1" แสดงว่าที่ขา DSR เป็นลอจิก "1"

"0" แสดงว่าที่ขา DSR เป็นลอจิก "0"

DCTS (Delta clear To send) ให้แจ้งการเปลี่ยนแปลงที่เกิดขึ้นของบิต CTS

"1" แสดงว่าบิต CTS (Clear To Send) เกิดการเปลี่ยนแปลงเมื่อเทียบจากการ อ่านค่าครั้งที่แล้ว

"0" แสดงว่าไม่มีการเปลี่ยนแปลงเมื่อเทียบกับการอ่านค่าครั้งที่แล้ว

DDSR (Delta Data set Ready) ใช้แจ้งการเปลี่ยนแปลงที่เกิดขึ้นของบิต DSR

"1" แสดงว่าบิต DSR (Data Set Ready) เกิดการเปลี่ยนแปลงเมื่อเทียบจากการ อ่านค่าครั้งที่แล้ว

"0" แสดงว่าไม่มีการเปลี่ยนแปลงเมื่อเทียบกับการอ่านค่าครั้งที่แล้ว

DRI (Delta Ring Indicator) ให้แจ้งการเปลี่ยนแปลงที่เกิดขึ้นของบิต RI

"1" แสดงว่าบิต RI (Ringing Indicator) เกิดการเปลี่ยนแปลงเมื่อเทียบจากการ อ่านค่าครั้งที่แล้ว

"0" แสดงว่าไม่มีการเปลี่ยนแปลงเมื่อเทียบกับการอ่านค่าครั้งที่แล้ว

DDCD (Delta Data carrier Detect) ใช้แจ้งการเปลี่ยนแปลงที่เกิดขึ้นของบิต DDCD

"1" แสดงว่าบิต CTS (Clear To Send) เกิดการเปลี่ยนแปลงเมื่อเทียบจากการอ่านค่าครั้งที่แล้ว

"0" แสดงว่าไม่มีการเปลี่ยนแปลงเมื่อเทียบกับการอ่านค่าครั้งที่แล้ว

DCTS (Delta Clear To Send) ให้แสดงสภาวะของขา CTS

"1" แสดงว่าที่ขา CTS เป็นลอจิก "1"

"0" แสดงว่าที่ขา CTS เป็นลอจิก "0"

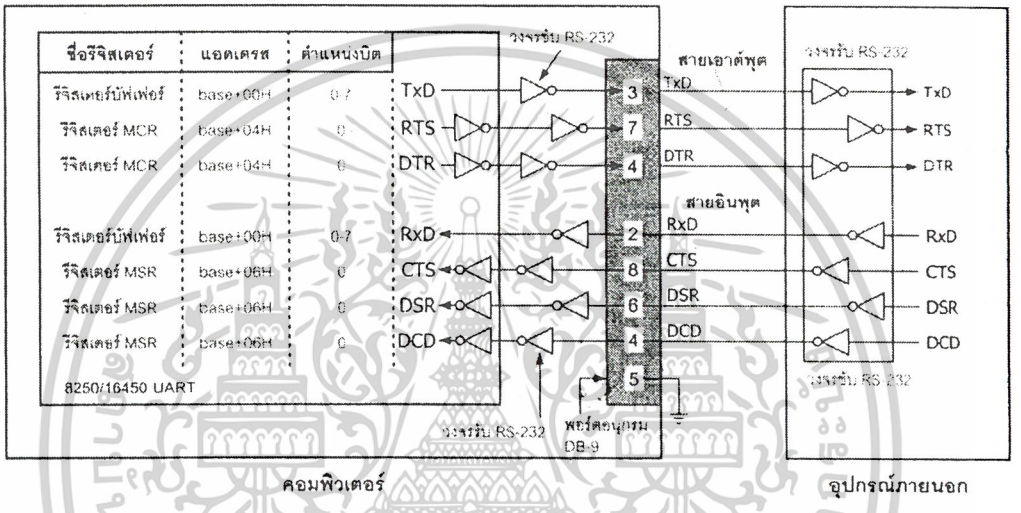
2.2.6.8 <u>รีจิสเตอร์ตำแหน่ง 07H : รีจิสเตอร์สำหรับเก็บข้อมูลชั่วคราว</u>

ทำหน้าที่เป็นหน่วยความจำแรมขนาด 1 ไบต์ การอ่านและเขียนข้อมูลที่รีจิสเตอร์ตัวนี้ไม่ส่งผลใดๆ ต่อการใช้งาน UART

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.6.9 ลักษณะสัญญาณอินพุตและเอาต์พุตของพอร์ต RS-232

สัญญาณเอาต์พุตที่ใช้ควบคุม (RTS และ DTR) และสัญญาณแสดงสถานะอินพุต(CTS,DSR และ DCD) ของพอร์ตอนุกรม RS-232 จะถูกกลับสถานะภายในตัว UART ส่วนสัญญาณข้อมูลทั้งภาคส่งและรับจะไม่ถูกกลับสถานะ UART จะให้ระดับสัญญาณเอาต์พุตออกมาเป็นแบบทีทีแอลเท่านั้น ดังนั้นเมื่อสัญญาณถูกส่งออกมาจาก UART จึงต้องส่งเข้าสู่วงจรขับเพื่อปรับระดับแรงดันให้ได้ระดับสัญญาณเป็นไปตามมาตรฐาน RS-232 ก่อนส่งออกไปจากคอมพิวเตอร์ สำหรับอุปกรณ์ต่อเชื่อมปลายทางก็จะต้องมีวงจรขับในลักษณะนี้เช่นเดียวกัน เพื่อให้ได้ระดับสัญญาณในระดับเดียวกัน แต่วงจรขับที่ใช้ทั้งภายในคอมพิวเตอร์และอุปกรณ์ต่อเชื่อมปลายทางนั้นจะถูกกลับสถานะ ดังแสดงเป็นบล็อกไดอะแกรมในรูปที่ 2.19 (อรรถพล บุญยะโภคา, วรพจน์ กรแก้ววัฒนกุล และ ชัยวัฒน์ ลิ้มพรจิตรวิไล, 2546)



รูปที่ 2.19 ไดอะแกรมแสดงโครงสร้างทางฮาร์ดแวร์ของพอร์ตอนุกรม

2.2.6.10 แอดเดรสของพอร์ตอนุกรม

แอดเดรสพื้นฐานของพอร์ตอนุกรมมี 4 ตำแหน่งดังนี้คือ

COM1 : 3F8H

COM2 : 2F8H

COM3 : 3E8H

COM4 : 2E8H

เมื่อเริ่มเปิดเครื่องเพื่อใช้งานคอมพิวเตอร์ ไบออสภายในคอมพิวเตอร์จะทำการตรวจสอบแอดเดรสของพอร์ตอนุกรมทั้งหมด ถ้าไบออสตรวจพบแอดเดรสของพอร์ตอนุกรมไบออสจะนำแอดเดรสที่ตรวจพบไปเก็บไว้ในหน่วยความจำขนาด 2 ไบต์ สำหรับพอร์ตอนุกรม COM1 จะเก็บไว้ที่

แอดเดรส 0000:0400H และ 0000:0401H ส่วนตำแหน่งอื่นๆ มีรายละเอียดดังนี้

COM2 = 0000:0402H - 0000:0403H

COM3 = 0000:0404H - 0000:0405H

COM4 = 0000:0406H - 0000:0407H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้ที่หน่วยความจำแอดเดส 0000:0411H ยังใช้สำหรับแสดงจำนวนของพอร์ตอนุกรมที่มีใช้อยู่ในคอมพิวเตอร์อีกด้วย โดยมีรายละเอียดดังแสดงในตารางที่ 1-2

2.3 การควบคุมมอเตอร์

2.3.1 มอเตอร์

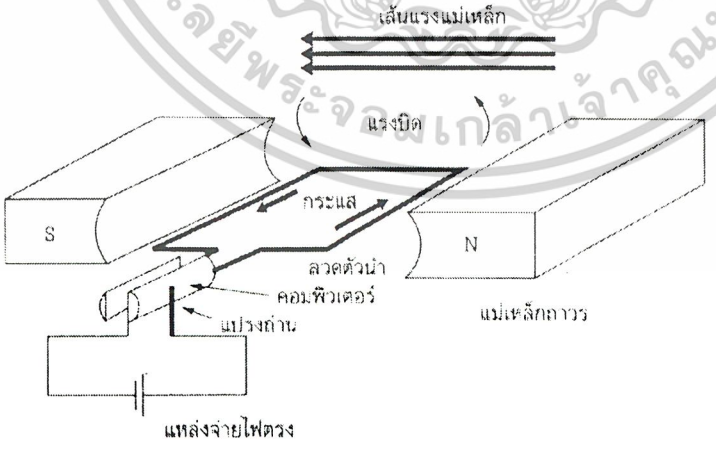
เครื่องจักรกลซีเอ็นซีสมัยใหม่ จะออกแบบระบบขับแบบเซอร์โว (Servo Driver) ทำให้สามารถปรับอัตราป้อนและความเร็วรอบได้โดยไม่มีขีดจำกัดของขั้นความเร็วรอบและอัตราป้อน มอเตอร์ที่ใช้ในระบบขับป้อนโดยทั่วไปจะมี อยู่ 3 ชนิดด้วยกันคือ

2.3.1.1 มอเตอร์แบบขั้น

เป็นมอเตอร์ที่ทำงานแบบต่อเนื่อง โดยการแปลงเคลื่อนสัญญาณที่ป้อนเข้าไปในระบบให้เคลื่อนที่เชิงมุม การหมุนในแต่ละมุมหรือขั้นที่เปลี่ยนไปหนึ่งขั้นจะเท่ากับหนึ่งคลื่นสัญญาณ ดังนั้น ตำแหน่งของเพลาจะถูกกำหนดโดยจำนวนคลื่นสัญญาณที่ป้อนเข้าไปในระบบ และความเร็วในการหมุนของเพลาวัดเป็นจำนวนขั้นต่อวินาที (Step Per Second) ซึ่งจะเท่ากับความถี่ของสัญญาณที่ป้อนเข้าในระบบที่วัดเป็นจำนวนคลื่นต่อวินาที (Pulses Per Second) ความเที่ยงตรงของระบบจะขึ้นอยู่กับความสามารถของมอเตอร์ในการแบ่งขั้นการหมุนตามจำนวนคลื่นสัญญาณที่ป้อนเข้าระบบ แรงบิดของมอเตอร์นี้จะลดความเร็วในการหมุนแบ่งเพิ่มขึ้น ดังนั้น จึงเหมาะสำหรับเครื่องจักรกลขนาดเล็กๆที่ไม่ต้องใช้กำลังขับมาก

2.3.1.2 มอเตอร์กระแสตรง

ลักษณะของมอเตอร์กระแสตรงจะใช้แม่เหล็กถาวรที่มี 4, 6 หรือ 8 ขั้ว ประกอบด้วย เบรก (Break) แกนมอเตอร์ (Rotor) อุปกรณ์วัดรอบ (Tachogenerator) และอุปกรณ์วัด (Measuring Box) การใช้มอเตอร์กระแสตรง ทำให้



รูปที่ 2.20 หลักการทำงานของมอเตอร์ไฟฟ้ากระแสตรงแบบมีแปรงถ่าน

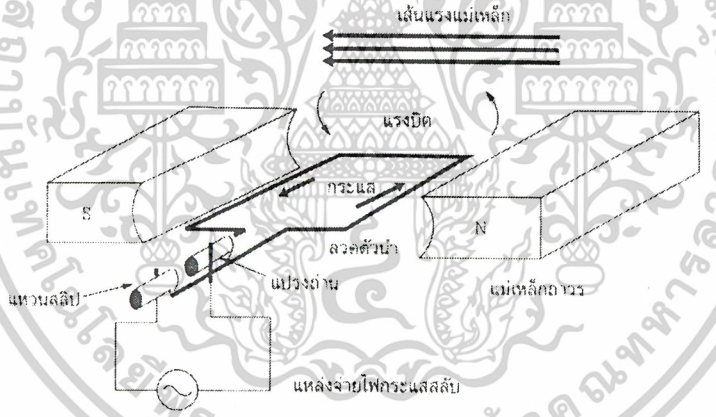
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถปรับอัตราการป้อนได้ละเอียดและมีวงจรควบคุมที่ไม่ซับซ้อน แต่มีข้อเสียตรงที่มอเตอร์ชนิดนี้ต้องใช้แปรงถ่าน ซึ่งต้องคอยทำความสะอาดและเปลี่ยนเมื่อแปรงถ่านหมด นอกจากนี้แปรงถ่านยังทำให้แกนมอเตอร์สึกหรออันเป็นผลทำให้กำลังมอเตอร์ลดลง ข้อเสียอีกอย่างหนึ่งก็คือ หากต้องการกำลังขับสูง มอเตอร์ก็จะมีขนาดใหญ่ด้วย และเมื่อใช้ความเร็วรอบสูงๆจะทำให้แรงบิดน้อยลง ดังนั้น จึงมักใช้เครื่องจักรกลซีเอ็นซีขนาดเล็กและขนาดกลาง

หลักการพื้นฐานของมอเตอร์ไฟฟ้ากระแสตรงแสดงได้ดังรูปที่ 2.20 กระแสไฟฟ้าจากแหล่งจ่ายไฟกระแสตรงจะไหลเข้าสู่ตัวนำในโรเตอร์ (Rotor) โดยผ่านแปรงถ่านและคอมมิวเตเตอร์ (Commutator) ทำให้เกิดอำนาจแม่เหล็กไฟฟ้าตัดกับสนามแม่เหล็กที่เกิดจากแม่เหล็กถาวร (Permanent Magnet) ขั้ว N และ S แรงบิดที่เกิดขึ้นจะมีทิศทางตามลูกศรในรูปที่ 2.20 (วโรดม ตู้จินดา, 2547) เมื่อโรเตอร์หมุนไปได้ 90 องศา คอมมิวเตเตอร์จะทำให้ทิศทางของกระแสจะไหลย้อนกลับเป็นตรงข้ามเป็นผลให้โรเตอร์ยังคงหมุนต่อไป

2.3.1.3 <u>มอเตอร์กระแสสลับ</u>

ส่วนใหญ่จะเป็นมอเตอร์แบบซิงโครนัส (Synchronous) ข้อดีของมอเตอร์ชนิดนี้คือไม่ต้องใช้แปรงถ่าน ทำให้สามารถลดการบำรุงรักษาได้มาก และมอเตอร์ขนาดเดียวกันเมื่อ เปรียบเทียบกับมอเตอร์จะสามารถให้แรงบิดได้ดีกว่าและมีขนาดเล็กกว่าด้วย ส่วนข้อเสียของมอเตอร์แบบนี้คือ วงจรควบคุมจะมีความซับซ้อนมากกว่าวงจรมอเตอร์กระแสตรง



รูปที่ 2.21 หลักการทำงานของมอเตอร์ไฟฟ้ากระแสสลับไม่ต้องใช้แปรงถ่าน

รูปที่ 2.21 (วโรดม ตู้จินดา, 2547) แสดงหลักการทำงานของมอเตอร์ไฟฟ้ากระแสสลับ โดยคอมมิวเตเตอร์จะถูกแทนด้วยแหวนสลิป (Slip Ring) ที่จะได้รับแรงดันไฟฟ้ากระแสสลับ สมมุติว่าเริ่มต้นด้วยการที่แปรงถ่าน A มีศักย์เป็นบวกเมื่อเทียบกับ B ก็จะทำให้เกิดแรงบิดขึ้นเช่นเดียวกับในกรณีของมอเตอร์ไฟฟ้ากระแสตรง แต่หลังจากช่วงหนึ่งมอเตอร์ก็เริ่มที่จะหยุดหมุนเนื่องจากไม่มีการคอมมิวเตชัน ดังนั้น จะต้องมีการกลับขั้วของแรงดันในช่วงเวลาที่เหมาะสม นั่นคือ การใช้ไฟฟ้ากระแสสลับเป็นตัวจ่ายกำลังจะทำให้มอเตอร์หมุนอย่างต่อเนื่องได้โดยอัตราความเร็วของการหมุนจะ ขึ้นอยู่กับความถี่ของแรงดันไฟฟ้ากระแสสลับ ในรูปที่ 2.21 เป็นมอเตอร์แบบมีแปรงถ่าน แต่ว่าโครงสร้างของมอเตอร์อาจออกแบบให้ไม่ต้องใช้แปรงถ่านได้ดังรูปที่ 2.20 โดยการสลับแม่เหล็กถาวรไปอยู่ที่ส่วนโรเตอร์แทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 การควบคุมการขับเซอร์โว

ในระบบ ซีเอ็นซี นี้จำเป็นต้องมีระบบเซอร์โว เนื่องจากคำสั่งที่อยู่ในรูปแบบของบล็อคจะถูกเก็บไว้ในอุปกรณ์เก็บข้อมูล และจะถูกส่งออกจากเครื่องไมโครคอมพิวเตอร์เพื่อขับเซอร์โวให้ควบคุมการเคลื่อนที่ของเครื่อง ซีเอ็นซี

1. ระบบเซอร์โวแบบวงรอบเปิด (Open-Loop System) ระบบการขับเซอร์โวด้วยวิธีนี้เครื่องจักร ซีเอ็นซี จะรับข้อมูลจากเครื่องอ่านเทป ส่งของมูลทาง RS-232-Cหรือวิธีอื่นๆ และเก็บข้อมูลไว้ในหน่วยความจำของคอมพิวเตอร์ เมื่อต้องการข้อมูลนั้นก็ส่งข้อมูลไปยังตัวขยายสัญญาณ (Amplifier) ก่อนจะส่งไปยังตัวขับเซอร์โวอีกครั้งหนึ่งของระบบการขับเซอร์โวแบบวงรอบเปิดนี้ ปกติจะใช้มอเตอร์แบบขั้น (Stepping Motor) เป็นตัวควบคุมการขับเคลื่อนส่วนการส่งสัญญาณนั้นจะใช้บอลสกรู (Ball Screw) ส่งกำลังขับ

ระบบการขับเซอร์โวมอเตอร์แบบวงรอบเปิดใช้กับการขึ้นรูปชิ้นงานที่มีตัดเฉือนชิ้นงานเบาๆ (Light Load) และมักใช้กับระบบที่ต้องการความแม่นยำสูง ค่าพิกัดมากกว่า 0.0001 นิ้ว

2. ระบบการขับเซอร์โวแบบวงปิด (Close-Loop System) ความแตกต่างของระบบการขับเซอร์โวแบบวงปิดและแบบวงเปิด คือในแบบปิดจะมีระบบการส่งสัญญาณย้อนกลับเพื่อตรวจสอบสัญญาณที่ส่งมาว่ามีความถูกต้องหรือมีความผิดพลาดอย่างไรจะมีการแสดงให้เห็นที่หน้าจอของคอมพิวเตอร์ซึ่งระบบการส่งสัญญาณย้อนกลับนี้จะมีอยู่ 2 ลักษณะ คือ แบบอนาล็อกและแบบดิจิตอล ระบบการขับเซอร์โวแบบวงรอบปิดมักใช้กับงานที่มีความเที่ยงตรงและต้องการความละเอียดสูง (ค่าความผิดพลาดไม่เกิน 0.0001 นิ้ว)

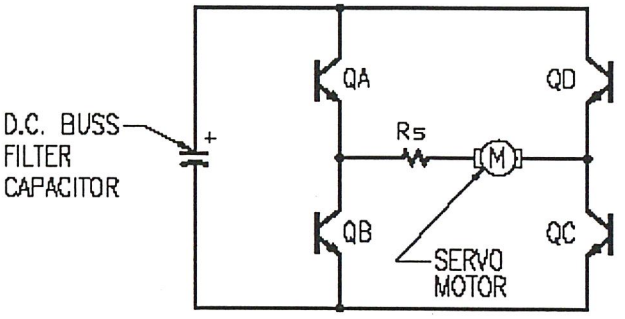
2.3.3 ชุดขับและภาคขยายกำลังของเซอร์โวมอเตอร์

ชุดขับและภาคขยายกำลัง ซึ่งจะเรียกโดยรวมว่า ชุดขับมอเตอร์ คือมีวงจรอิเล็กทรอนิกส์ที่จ่ายกระแสให้มอเตอร์ให้ทำงานตามสัญญาณคำสั่งที่ส่งมาจากชุดควบคุม โดยมากชุดขับมอเตอร์จะถูกติดตั้งไว้ข้างหลังของเครื่อง ซีเอ็นซี ชุดขับมอเตอร์ที่ดีมักมีการควบคุมมอเตอร์นอกเหนือจากการจ่ายกำลัง คือ อาจมีวงจรป้อนกระแสเพื่อควบคุมความเร็วเป็นการลดภาระงานของชุดควบคุม ซึ่งมีการแบ่งชุดขับมอเตอร์ออกเป็น 2 ประเภท คือ แบบเชิงเส้น (Linear) และแบบแปรตามความกว้างของพัลส์ (Pulse Width Modulation: PWM)

1. ชุดขับแบบเชิงเส้น (Linear)

รูปที่ 2.22 (คู่มือการใช้งานชุดขับมอเตอร์ Glentek, 2534) แสดงชุดขับแบบเชิงเส้นจะมีทรานซิสเตอร์กำลัง QA-QB ที่ถูกจัดเป็นวงจรบริดจ์เพื่อทำการควบคุมกระแสที่จ่ายให้กับขดลวดอาร์เมเจอร์ของมอเตอร์ การควบคุมทิศทางการหมุนและแรงบิดกระทำโดยการสั่งงานทรานซิสเตอร์เป็นคู่ คือ QA-QC และ QB-QD การที่ทรานซิสเตอร์ต้องทำงานในช่วงที่เป็นเชิงเส้นมีผลทำให้เกิดความร้อนสูงและต้องใช้ครีบระบายความร้อนขนาดใหญ่ช่วย ชุดขับแบบเชิงเส้นจึงมักมีขนาดใหญ่และใช้งานสำหรับพิกัดกำลังไม่สูงมาก

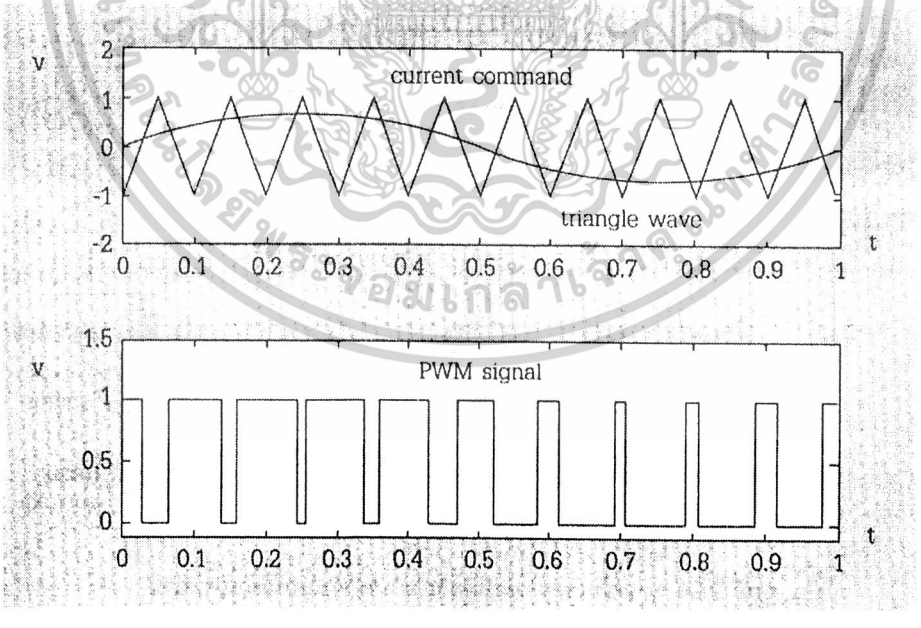
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.22 ชุดขับมอเตอร์แบบเชิงเส้น

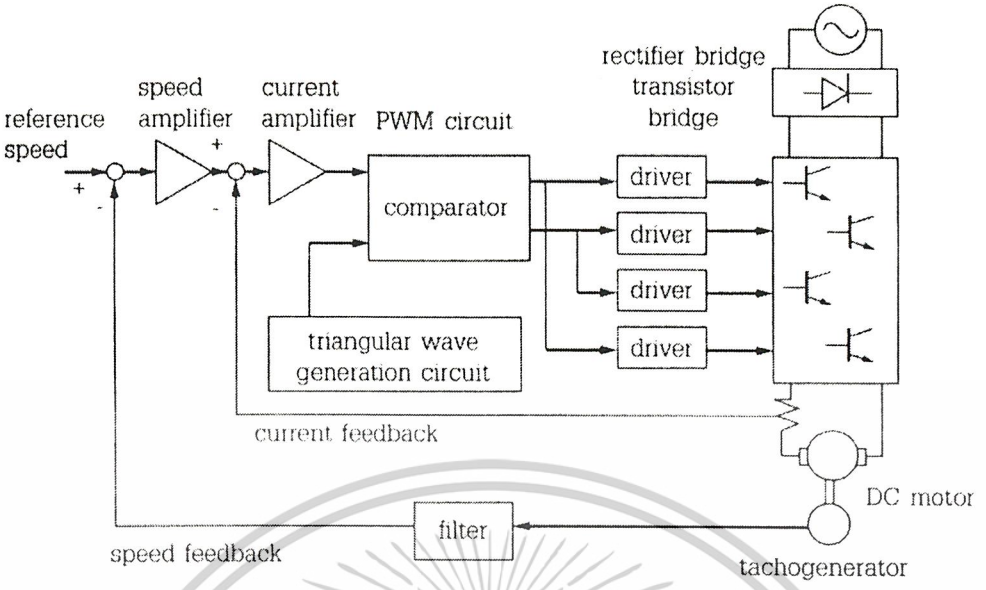
2. ชุดขับแบบแปรตามความกว้างของพัลส์ (Pulse Width Modulation: PWM)

รูปที่ 2.23 (วโรดม ตู้จินดา, 2547) แสดงหลักการของ PWM คือ เราจะแปลงรูปคลื่นสัญญาณต่อเนื่องให้เป็นสัญญาณต่อเนื่องเป็นพัลส์ที่มีความกว้างเป็นสัดส่วนกับระดับของสัญญาณ คือ จะทำการเปรียบเทียบสัญญาณคำสั่งควบคุมกระแสมอเตอร์กับรูปคลื่นสามเหลี่ยมที่สร้างขึ้น และให้แรงดันของสัญญาณ PEM เป็นบวกหรือศูนย์ตามผลการเปรียบเทียบ จะเห็นว่าเมื่อสัญญาณคำสั่งมีสัญญาณระดับสูง สัญญาณของ PWM จะมีความกว้างของพัลส์บวกมาก ซึ่งเมื่อส่งไปให้สวิตช์ซึ่งทรานซิสเตอร์ก็จะกระจายกระแสให้มอเตอร์มีค่าเฉลี่ยให้มอเตอร์มีค่าเฉลี่ยตามที่เราต้องการ การที่ทรานซิสเตอร์ทำงานในแบบสวิตช์ซึ่งทำให้กำลังสูญเสียในรูปความร้อนมีน้อยกว่าชุดขับเคลื่อนเชิงเส้น



รูปที่ 2.23 หลักการสร้างสัญญาณ PWM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.24 วงจรขับเซอร์โวมอเตอร์กระแสตรงแบบมีแปรงถ่าน

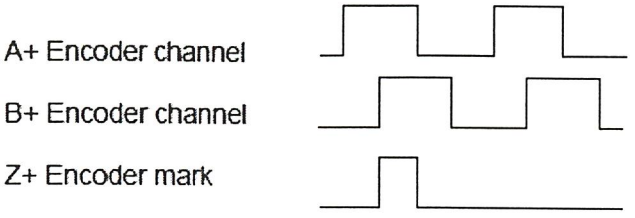
เซอร์โวมอเตอร์กระแสตรงแบบมีแปรงถ่านและเซอร์โวมอเตอร์กระแสตรงแบบไม่มีแปรงถ่าน ข้อแตกต่าง คือ มอเตอร์แบบมีแปรงถ่านต้องการการควบคุมแบบเฟสเดียวส่วนการไม่มีแปรงถ่านต้องการควบคุมสามเฟส มอเตอร์แบบไม่มีแปรงถ่านยังต้องการวงจรบางส่วนเพิ่มเติม เช่น การป้อนกลับตำแหน่ง ส่วนกำเนิดสัญญาณรูปซายน์ ส่วนแปลงสัญญาณไฟตรงเป็นสัญญาณรูปคลื่นซายน์ ส่วนจับความเร็ว

2.3.4 เอนโคเดอร์ชนิดแสง

เอนโคเดอร์ชนิดแสง (Optical Encoder) จำแนกได้เป็น 2 ประเภท คือ แบบส่วนเพิ่ม (Incremental) และแบบสมบูรณ์ (Absolute) หลักการเอนโคเดอร์ชนิดแสง คือ จะมีต้นกำเนิดแสงที่เป็นอุปกรณ์จำพวกไดโอดเปล่งแสง (LED) ที่จะให้แสงไปยังอุปกรณ์รับแสง โดยระหว่างอุปกรณ์ทั้งสองจะมีจานหมุนเจาะรูซึ่งเจาะช่องไว้เป็นระยะ เมื่อมอเตอร์หมุนจานก็จะหมุนตามและตัดแสงทำให้เกิดเป็นสัญญาณพัลซ์ที่มีความถี่แปรผันตามความเร็วของการหมุน จำนวนช่องที่อยู่บนจานจะเป็นตัวกำหนดความละเอียดของเอนโคเดอร์ ถ้าหากว่าเราป้อนสัญญาณพัลซ์นี้ให้กับวงจรนับก็จะได้ค่าตัวเลขที่สามารถแปลงเป็นค่าตำแหน่งเชิงเส้นของการเคลื่อนที่ได้

เอนโคเดอร์ชนิดแสงจะมีตัวส่งแสง 2 ชุดและตัวรับ 3 ชุด ผลที่ได้ คือ จะเกิดสัญญาณพัลซ์จำนวน 3 สัญญาณ คือ A, B และ Z โดยสัญญาณ A กับ B จะมีเฟสต่างกันอยู่ 90 องศา เมื่อมอเตอร์หมุนไปทางหนึ่งสัญญาณ A จะนำหน้า B เมื่อหมุนกลับ B จะนำหน้า A ดังแสดงในรูปที่ 2.25 (คู่มือการใช้งานชุดขับมอเตอร์ Glentek, 2534) ดังนั้นเราสามารถรู้ทิศทางการหมุนได้ ส่วนสัญญาณ Z จะเกิดขึ้นเพียงหนึ่งพัลซ์ในทุกๆ รอบการหมุน ซึ่งจะมีประโยชน์ในการหาตำแหน่งอ้างอิง (Home) ของเครื่องจักร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

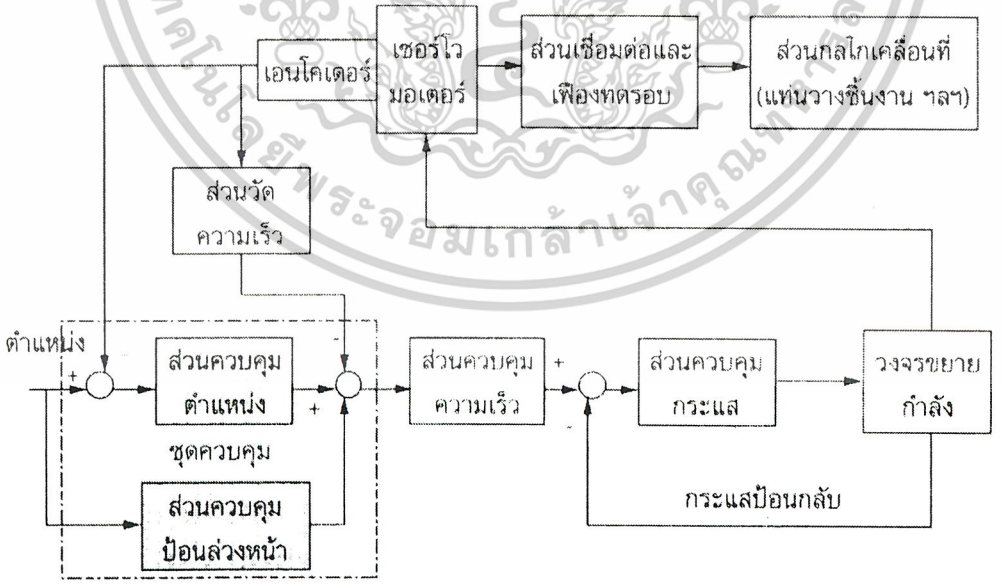


รูปที่ 2.25 พัลซ์ที่เกิดจากเอนโคเดอร์แบบส่วนเพิ่ม

เอนโคเดอร์ชนิดแสงแบบสมบูรณ์จะมีการให้ค่าเป็นเลขฐานสอง โดยความละเอียดจะขึ้นอยู่กับจำนวนบิตของเอนโคเดอร์ ซึ่งจะมีสายสัญญาณของเอนโคเดอร์เท่ากับจำนวนบิตในการเคลื่อนที่หนึ่งรอบก็จะมีตัวนับจำนวนรอบของการหมุนอยู่ด้วย

2.3.5 ชุดควบคุมการทำงานของเครื่อง

ส่วนประกอบหลักของชุดควบคุม คือ ระบบคอมพิวเตอร์ที่อยู่ภายในซึ่งประกอบไปด้วยตัวประมวลผล หน่วยความจำ หน่วยเก็บข้อมูลถาวร และส่วนอินเตอร์ เฟสกับเครื่องจักรและผู้ใช้งาน ชุดคอมพิวเตอร์ในลักษณะนี้สามารถจัดรวมอยู่ในระบบในระบบคอมพิวเตอร์ที่ทำงานเฉพาะอย่างที่เรียกกันว่า ระบบสมองกลฝังตัว (Embedded Systems) เราสามารถสรุปหน้าที่การทำงานของชุดควบคุมเครื่อง ซีเอ็นซี ดังแสดงในรูปที่ 2.26 (วโรดม ตู้จินดา, 2547) ตามลำดับความสำคัญได้ดังนี้



รูปที่ 2.26 ตัวอย่างระบบการควบคุมการเคลื่อนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ควบคุมการเคลื่อนที่แบบเวลาจริง
2. ทำฟังก์ชันพีแอลซี
3. แปลคำสั่งจากโปรแกรมและสร้างจุดคำสั่ง (Set Points)
4. ติดต่อสื่อสารกับผู้ใช้และ/หรือคอมพิวเตอร์หลัก

อุปกรณ์ในชุดควบคุมประกอบด้วย

1. ตัวประมวลผลที่ทำหน้าที่เป็นตัวควบคุม โดยอาจเป็นตัวประมวลสัญญาณดิจิตอล ไมโครคอนโทรลเลอร์ หรือตัวประมวลผลที่ใช้ในเครื่องพีซี เช่น ตระกูล x86
2. หน่วยความจำแบบชั่วคราว (RAM) ใช้สำหรับเก็บโปรแกรมสำหรับผู้ใช้ ข้อมูลจะสูญหายเมื่อปิดเครื่องยกเว้นหากมีแบตเตอรี่สำรอง
3. หน่วยความจำแบบคงข้อมูลได้ (Non–Volatile) เช่น Flash Memory หรือ EEPROM ใช้เก็บพารามิเตอร์ที่สำคัญของระบบ เช่น ค่าพีไอดี เป็นต้น ชุดข้อมูลบางรุ่นอาจมีหน่วยความจำแบบถาวร เช่น ฮาร์ดดิสก์ เครื่องอ่านซีดีหรือแผ่นดิสก์
4. หน่วยความจำแบบอ่านได้อย่างเดียว (ROM) ใช้เก็บข้อมูลที่เปลี่ยนแปลงไม่ได้ เช่นโปรแกรมตั้งค่าระบบช่วงเริ่มต้นการทำงาน ตารางค่าทางคณิตศาสตร์ เป็นต้น
5. ส่วนต่อประสานกับเอนโคเดอร์และชุดขับมอเตอร์ ต้องมีจำนวนแชนแนลอย่างน้อยเท่ากับจำนวนแกนของเครื่อง ซีเอ็นซี
6. ส่วนควบคุมฟังก์ชั่นพีแอลซี มีลักษณะเป็นอินพุท/เอาต์พุทแบบดิจิตอล ใช้ในการควบคุมปั๊มน้ำ ลิมิตสวิตซ์ ฯลฯ
7. ส่วนติดต่อกับผู้ใช้งาน เช่น จอแสดงผล แผงควบคุม หรืออาจเป็นอุปกรณ์ของเครื่องพีซี
8. ส่วนติดต่อสื่อสารกับคอมพิวเตอร์หลัก (PIC, RS-232, Parallel Port, USB, LAN)

2.3.6 ส่วนติดต่อกับผู้ใช้งาน

ความจริงแล้วส่วนติดต่อกับผู้ใช้งาน (Man-Machine Interface : MMI) จะเป็นฟังก์ชันที่สำคัญของชุดควบคุมเครื่องซีเอ็นซี เพราะถึงแม้ว่าชุดควบคุมจะมีสมรรถนะดีเพียงใด ถ้าหากใช้งานยากหรือมีการแสดงผลที่ไม่สื่อความหมายกับผู้ใช้ก็คงเป็นผลิตภัณฑ์ที่ขายได้ไม่ดี หรืออาจก่อให้เกิดอันตรายในขณะการทำงาน เราจึงแยกส่วนติดต่อผู้ใช้งานออกมาเพื่อศึกษาว่าควรจะประกอบด้วยฟังก์ชันพื้นฐานอะไรบ้างรูปที่ 2.32 แสดงส่วนติดต่อผู้ใช้งานแบบที่สร้างบนซอร์ฟแวร์เครื่องพีซี โดยมีส่วนแสดงผลและส่วนสั่งงานที่สำคัญที่จำเป็นต้องมี ส่วนแสดงผลประกอบด้วยค่าตำแหน่งของแต่ละแกนของเครื่องจักรที่จะต้องถูกเปลี่ยนค่าตามความเป็นจริงทุก ๆ ช่วงเวลาที่เหมาะสมค่าพารามิเตอร์ต่าง ๆ ที่ตั้งไว้ และยังแสดงโปรแกรมรหัสจีโค้ดปัจจุบัน โดยบล็อกที่กำลังทำงานจะมีสีแตกต่างออกไป สำหรับส่วนสั่งงานในรูปที่ 2.32 จะทำลักษณะเหมือนปุ่มกด ซึ่งสามารถสั่งงานได้โดยใช้เมาส์คลิกหรือคีย์บอร์ดในการติดต่อกับผู้ใช้งาน เราอาจแบ่งได้เป็น 3 ส่วน คือ ส่วนแสดงผลการทำงาน ส่วนติดต่อสั่งงานและส่วนเก็บข้อมูล โดยแต่ละหัวข้อสามารถแยกส่วนการทำงานย่อยได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ส่วนแสดงผลการทำงาน

- ส่วนแสดงตำแหน่งส่วนเคลื่อนที่เป็นตัวเลข
- ส่วนแสดงสถานะต่างๆ ของเครื่อง
- ส่วนแก้ไขโปรแกรม (Editor)
- ส่วนแสดงผลการทำงานแบบ 2 มิติ หรือ 3 มิติโดยอาจมีการจำลองการทำงานก่อนทำงานจริงมีประโยชน์ในการทดสอบโปรแกรม
- ส่วนช่วยเหลือหรือคำอธิบายเพิ่มเติม

2. ส่วนติดต่อสั่งงาน

- เมนูการทำงานต่าง ๆ สั่งงานผ่านเมาส์หรือคีย์บอร์ด
- สวนสั่งงานอ้างอิงกับจอแสดงผล (Soft Key)
- ส่วนสวิตซ์สั่งงานจากภายนอก เช่น Emergency Swich,Cycle Hold,Cycle Start ส่วนติดต่อกับพีแอลซี
- ส่วนติดค่อสื่อสารผ่านพอร์ตต่าง ๆ หรือระบบ LAN
- ส่วนตั้งค่าพารามิเตอร์ของระบบ

3. ส่วนเก็บข้อมูล

- โปรแกรมที่ทำงานในปัจจุบัน
- ส่วนเก็บข้อมูลทั้งค่าต่าง ๆ
- ส่วนเก็บโปรแกรมอื่น ๆ เช่น Subroutines หรือโปรแกรมพีแอลซี
- ส่วนเก็บข้อมูลเกี่ยวกับเครื่องมือและตำแหน่งบนตัวเปลี่ยนเครื่องมือ

ที่กล่าวมาเป็นส่วนประกอบพื้นฐานของส่วนติดต่อผู้ใช้งานที่ดี ส่วนรายละเอียดปลีกย่อยอาจแตกต่างกันไปตามลักษณะการทำงานของเครื่องมือซีเอ็นซีแต่ละแบบปกติแล้วเวลาเราสั่งซื้อชุดควบคุมสำหรับงานรีโทรฟิต บริษัทผู้ผลิตจะถามว่าต้องการใช้กับเครื่องจักรชนิดใด และจะให้หรือขายซอฟต์แวร์ส่วนติดค่อผู้ใช้ที่เหมาะสมกับเครื่องจักรนั้นให้กับเรา ซึ่งในบางครั้งอาจต้องมีการปรับ (Customize) ให้ใช้งานได้กับเครื่องจักรของเรา หรืออาจต้องการแสดงผลเป็นภาษาไทย แต่โดยส่วนใหญ่ผู้ผลิตจะไม่ให้ซอร์สโค้ดกับเราดังนั้น การแก้ไขจะต้องกระทำโดยผู้ผลิตโดยคิดค่าใช้จ่ายเพิ่ม นี่เป็นข้อจูงใจอีกประการหนึ่งในการพัฒนาชุดควบคุมขึ้นใช้เอง

2.3.7 การโปรแกรมเครื่องซีเอ็นซี

การเขียนโปรแกรมสำหรับเครื่องซีเอ็นซีแต่ละแบบอาจมีรายละเอียดแตกต่างกัน แต่สิ่งที่เหมือนกัน คือ จะต้องมีการเขียนชุดคำสั่งหรือที่เรียกว่า โปรแกรมผลิตชิ้นส่วน (Part Program) โปรแกรมนี้อาจจะป้อนเข้าสู่เครื่องซีเอนซีโดยตรงโดยผู้ใช้งานกดปุ่มบนแผงควบคุม แต่วิธีที่นิยมใช้กว่า คือ การโหลดโปรแกรมจากแหล่งเก็บข้อมูล เช่น แผ่นดิสก์ หรืออาจโหลดผ่านระบบเครือข่าย โปรแกรมผลิตชิ้นส่วนโดยมากจะถูกสร้างจากซอฟต์แวร์อื่น เช่น ซอฟต์แวร์แคด/แคม (CAD/CAM)

คำสั่งหรือภาษาที่ใช้สั่งงานเครื่องซีเอ็นซีแบบหนึ่งที่นิยมใช้มากและจะใช้อ้างอิงในหนังสือนี้ คือคำสั่งที่เรียกกันว่า รหัสจี (G-Code) โดยมาตรการจะกำหนดหมายเลขรหัสจีให้แทนลักษณะการทำงานในรูปแบบต่าง ๆ ในแต่ละบรรทัดคำสั่งจะถูกเรียกว่า บล็อก (Block) ซึ่งจะมีหมายเลขบรรทัดกำกับอยู่โดยขึ้นต้นด้วยตัวอักษร N ลองพิจารณาคำสั่ง 2 บล็อกดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

N0008 G00 X0 Y0 Z0

N0009 G01 X10 F100

ในบล็อกที่ 8 เป็นคำสั่งเคลื่อนที่เร็วไปยังตำแหน่ง X = 0 Y = 0 Z = 0 ซึ่งปกติจะเป็นจุดอ้างอิง (Home) ของแต่ละแกน การเคลื่อนที่เร็ว คือ การสั่งให้มอเตอร์หมุนด้วยความเร็วสูงสุด เพราะจะไม่มีการกัดเนื้อชิ้นงานในคำสั่งนี้และเส้นทางในการเคลื่อนที่ก็ไม่มีความสำคัญ ต่างจากบล็อกที่ 9 ซึ่งเป็นคำสั่งให้แกน x เคลื่อนที่แนวเส้นตรงไปยังตำแหน่ง x = 10 หน่วย ด้วยอัตราป้อน 100 หน่วย/นาทีขณะที่ระบบควบคุมจะพยายามคงค่าตำแหน่งของแกน Y และ Z ไว้ที่ 0

การที่โปรแกรมผลิตชิ้นงานแบบนี้มีชื่อเรียกว่า รหัสจี ไม่ได้หมายความว่าคำสั่งจะขึ้นต้นด้วยตัวอักษรจี (G) ทั้งหมด แต่จะรวมถึงคำสั่งที่ขึ้นต้นด้วยตัวอักษรเอ็ม (M) ซึ่งมักใช้เป็นตัวสั่งงานอุปกรณ์ที่ ควบคุมด้วยพีแอลซี ยกตัวอย่างเช่น MO7-MO9 จะเป็นการควบคุมการเปิด-ปิดสารหล่อเย็น หรือตัวอักษรที (T) ใช้ในการเรียกใช้เครื่องมือเป็นต้น ตารางที่ 2.3 (วโรดม ตู้จินดา, 2547) แสดงตัวอย่างรหัสจีที่มีการใช้บ่อย

ข้อสังเกตประการหนึ่ง คือ ซอร์ฟแวร์ทางด้านการออกแบบ (CAD) อาจให้เอาต์พุตเป็นไฟล์โปรแกรมที่ไม่ได้อยู่ในรูปของรหัสจีโดยตรง แต่อาจเป็นไฟล์ที่เรียกว่า CL-Files ดังนั้น จึงต้องมีซอร์ฟแวร์ที่แปลงจาก CL เป็นไฟล์รหัสจี ซอร์ฟแวร์ดังกล่าวเรียกว่า Post- Processor เราจะไม่กล่าวถึงการพัฒนาซอร์ฟแวร์ในส่วนนี้ โดยจะตั้งสมมุติฐานว่าชุดควบคุมซีเอ็นซีที่สร้างขึ้นจะถูกสั่งงานโดยรหัสจี

รายละเอียดของคำสั่งรหัสจีทั้งหมดไม่สามารถรวบรวมในหนังสือเล่มนี้ ขอให้ศึกษาจากหนังสือเกี่ยวกับการโปรแกรมเครื่องซีเอ็นซีทั่ว ๆ ไป สิ่งเดียวที่เราต้องคำนึงถึงสำหรับการออกแบบสร้างชุดควบคุมก็คือ คอมพิวเตอร์จะต้องทำการแปลงรหัสจีให้เป็นจุดย่อยของตำแหน่ง (Set Points) ก่อนทำการส่งให้กับระบบควบคุมป้อนกลับเพื่อควบคุมแต่ละแกนให้เคลื่อนที่ที่ไปยังตำแหน่งนั้น นอกจากนั้นในชุดควบคุมซีเอ็นซีขั้นสูงจะต้องสามารถแปลงคำสั่งรหัสจีล่วงหน้าหลายบล็อกเพื่อไม่ให้การเคลื่อนที่ของหัวกัดสะดุด

ตารางที่ 2.3 ตัวอย่างคำสั่งรหัสจีพื้นฐาน

คำสั่ง รหัส	ความหมาย/การทำงาน
G00	การเคลื่อนที่เร็ว
G01	การเคลื่อนที่แนวเส้นตรงโดยสามารถควบคุมอัตราการป้อน
G 02	การเคลื่อนที่แนวเส้นตรงโดยสามารถควบคุมอัตราป้อน
G03	การเคลื่อนที่แนวส่วนโค้งทวนเข็มนาฬิกา
G04	เวลาหยุด (Dwell)
G13-G16	การเลือกแกน
G17-G19	การเลือกระนาบ
G40	ยกเลิกการชดเชยขนาดรัศมีมีดกัด
G41	เรียกให้การชดเชยขนาดรัศมีมีดกัดด้านซ้ายของเส้นขอบรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 (ต่อ) ตัวอย่างคำสั่งรหัสจีพื้นฐาน

คำสั่ง รหัส	ความหมาย/การทำงาน
G42	เรียกให้การชดเชยขนาดรัศมีมีดกัดด้านขวาของเส้นขอบรูป
G43	เรียกให้การชดเชยขนาดรัศมีมีดกัดมุมใน
G44	เรียกให้การขดเชยขนาดรัศมีมีดกัดมุมนอก
G90	การกำหนดขนาดแบบสัมบูรณ์ (Absolute)
G91	การกำหนดขนาดเชิงส่วนเพิ่ม (Incremental)
G94	อัตราป้อนเป็น มม. ต่อนาที
G95	อัตราป้อนเป็น มม. ต่อรอบ
M00	หยุดโปรแกรม
M03	สปินเดิลหมุนตามเข็มนาฬิกา
M04	สปินเดิลหมุนทวนเข็มนาฬิกา
M05	หยุดสปินเดิล
M06	เปลี่ยนเครื่องมือ
M07	เปิดปั๊มสารหล่อเย็นหมายเลข 2
M08	เปิดปั๊มสารหล่อเย็นหมายเลข 1
M09	ปิดปั๊มสารหล่อเย็น

ที่มา: วโรดม ตู้จินดา (2547)

2.4 การขับป้อน

การขับป้อนจะทำให้เกิดการเคลื่อนที่ของแท่นเลื่อน ในขณะตัดเฉือนแท่นเลื่อนอาจพาให้ชิ้นงานเคลื่อนที่หรือคมตัดเคลื่อนที่ก็ได้ ระบบขับป้อนโดยทั่วไปจะใช้มอเตอร์กระแสตรงในการหมุนขับและควบคุมการทำงาน ด้วยวงจรอิเล็กทรอนิกส์จากภายนอก มอเตอร์ชนิดนี้จะสามารถหยุดหมุนและเบรกให้หยุดได้ทั้งสองทิศทางขณะตัดเฉือนชิ้นงาน การเคลื่อนที่ขับป้อนจึงต้องเป็นไปอย่างสม่ำเสมอและสามารถต้านแรงกระทำจากภายนอกได้ เช่น แรงตัดเฉือน ด้วยเหตุนี้ระบบขับป้อนจะต้องมีการออกแบบให้มีความแข็งแรงสูง การเคลื่อนที่สม่ำเสมอ สามารถตอบสนองต่อการเปลี่ยนอัตราป้อนได้อย่างรวดเร็วนอกจากนี้การเร่งอัตราป้อนให้เคลื่อนที่เร็วแล้วหยุดโดยทันทีทันใด ทำให้มอเตอร์รับภาวะโหลดมากเกินไปซึ่งอาจทำให้มอเตอร์เสียหายได้ ปัจจัยที่จะทำให้ระบบขับป้อนสามารถทำงานได้อย่างมีประสิทธิภาพก็คือ การเลือกอุปกรณ์ในระบบขับป้อนให้เหมาะสมกับการทำงานของเครื่องจักรและการออกแบบวงจรควบคุม

บอลสกรู
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบขับป้อนของเครื่องจักรซีเอ็นซี คือ การส่งกำลังขับด้วยบอลสกรู ซึ่งจะมีลูกบอลไหลหมุนเวียนอยู่ตลอดเวลา บอลสกรูจะประกอบด้วยสกรูกับนัตที่มีลักษณะเป็นเกลียวกลม ร่องเกลียวกลมบนบอลสกรูและในนัตจะชุบแข็งและเจียระไนผิวเรียบมันเพื่อลดความฝืด และเพิ่มความแม่นยำในการเคลื่อนที่เมื่อมอเตอร์หมุนขับสกรู นัตจะเคลื่อนที่ไปตลอดความยาวของสกรูพาให้แท่นเลื่อนและโต๊ะงานเคลื่อนที่ไปตามรางเลื่อน

ภายในตัวนัตจะประกอบด้วยชุดของลูกบอลจำนวนมาก ทำให้มั่นใจได้ว่าความเสียดทานในการส่งกำลังขับจากสกรูไปยังแท่นเลื่อนจะมีน้อยมาก นัตจะถูกแบ่งออกเป็นสองซีก และขันประกอบเข้าด้วยกัน โดยมีการเตรียมแรงอัดไว้ก่อน (Preload) ทำให้สามารถลดระยะคลอน (Backlash) ให้เหลือน้อยที่สุดหรือแทบจะไม่มีเลย ทำให้การเคลื่อนที่ของแท่นมีความเที่ยงตรงสูงสามารถทำงานซ้ำๆกันได้

การต่อกำลังระหว่างมอเตอร์กับบอลสกรู จะมีชุดคลัตช์ความฝืดเป็นตัวเชื่อม ซึ่งนอกจากจะมีหน้าที่ต่อกำลังขับแล้วยังมีหน้าที่ป้องกันอุบัติเหตุที่เกิดจากแท่นเลื่อนหรือโต๊ะงานชนหรือกระแทกกับสิ่งกีดขวางไม่ให้เครื่องจักรซีเอ็นซีเกิดความเสียหายมากเกินไปกล่าวคือเมื่อมีการชน หรือกระแทกกันขึ้นจนแรงมากถึงค่าหนึ่งชุดคลัตช์ก็จะตัดระบบการส่งกำลังขับระหว่างมอเตอร์กับบอลสกรูทันที

2.5 อินเตอร์โพเลเตอร์ในระบบการผลิต

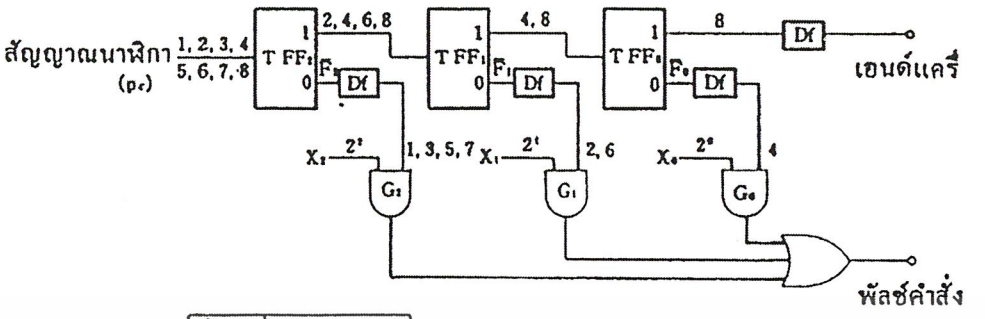
สิ่งที่จำเป็นในระบบการผลิต คือ การสร้างคู่ลำดับการเคลื่อนที่ของแกนแต่ละแกน เพื่อให้เกิดทางเดินที่ต้องการของเครื่องมือตัดให้สัมพันธ์กับชิ้นงาน ซึ่งรวมทั้งการสร้างสัญญาณที่กำหนดรูปร่างของชิ้นงานที่จะผลิต และส่งสัญญาณนั้นเป็นอินพุตอ้างอิงของลูปการควบคุมที่เกี่ยวข้อง ตัวอินเตอร์โพเลเตอร์จะทำหน้าที่สร้างสัญญาณอ้างอิงนี้ ในระบบเอ็นซีจะมีวงจรอินเตอร์โพเลเตอร์ซึ่งเป็นวงจรดิจิตอล ในระบบซีเอ็นซีตัวอินเตอร์โพเลเตอร์ถูกสร้างโดยซอร์ฟแวร์

2.5.1 วิธีของ MIT

วิธีของ MIT รูปที่ 2.27 (จงกล งามวิวิทย์, 2535) วิธีนี้ใช้เมื่อตอนพัฒนาเครื่องเอ็นซีและใช้เป็นอินเตอร์โพเลเตอร์แบบเส้นตรงแต่เพียงอย่างเดียว ถ้าจะตัดเส้นโค้งอย่างง่ายๆ ต้องตัดเป็นเส้นตรงสั้นๆ โดยประมาณ ซึ่งจะทำให้เกิดความยุ่งยากต่อการสร้างเทปเอ็นซี อย่างไรก็ตาม วิธีนี้ให้หลักการของวงจรแบบไบนารี่แบบง่ายแต่สามารถทำงานได้ตามประสงค์

จากรูปที่ 2.27 ซึ่งประกอบด้วยฟลิปฟล็อป 3 ตัว เอาท์พุตของฟลิปฟล็อปแต่ละตัวซึ่งเขียนได้เป็น F_I (I=0,1,2) นั้นจะถูกควบคุมโดยจำนวนของพัลซ์คำสั่งที่อ่านจากเทป และเก็บบันทึกไว้ในรีจิสเตอร์ X มีความจำเป็นที่จะต้องแปลงจำนวนพัลซ์เหล่านี้เป็นไบนารี่ก่อน ในกรณีนี้เนื่องจากมีจำนวนบิตอยู่ทั้งหมด 3 บิต ดังนั้นจำนวนพัลซ์นาฬิกาก็คือ $8(2^3)$ ซึ่งเท่ากับ 1 คาบ สมมติว่า ต้องการจำนวนพัลซ์คำสั่งทั้งหมดเป็น 6 จะได้เลขฐาน 2 เป็น 110 ซึ่งแต่ละตัวจะไปควบคุมเกท AND G_2, G_1, G_0 ให้เปิดและปิด (ถ้า 1 = เปิด , ถ้า 0 = ปิด) ดังนั้นจะเห็นได้ว่าที่เกท G_2 จะมีพัลซ์นาฬิกาเลขที่ 1, 3, 5, 7 ผ่านออกไปยังเกท OR และที่เกจ G_1 จะมีพัลซ์นาฬิกาเลขที่ 2, 6 ผ่านออกไปยังเกท OR ด้วยส่วนพัลซ์นาฬิกาที่ 4 จะไม่ผ่านเกท G_0 ทั้งนี้เพราะว่า X_0 เท่ากับ 0 เกท G_0 จึงปิด ดังนั้นเอาท์พุทของเกท OR จึงมีจำนวนพัลซ์ทั้งหมด 6 พัลซ์ออกมา ซึ่งจะเป็นพัลซ์ คำสั่งที่จะถูกนำไปใช้งานต่อไป การเพิ่มความถี่ของพัลซ์ นาฬิกาจะทำให้พัลซ์คำสั่งเปลี่ยนตามไปด้วย ซึ่งหมายความว่า ความเร็วของเครื่องจักรจะถูกควบคุมได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

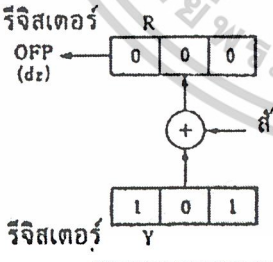


จำนวนพัลซ์คำสั่ง	ไบนารี่			1	2	3	4	5	6	7	8
	X_1	X_2	X_3								
7	1	1	1								
6	1	1	0								
5	1	0	1								
4	1	0	0								
3	0	1	1								
2	0	1	0								
1	0	0	1								

รูปที่ 2.27 วิธีของ MIT

2.5.2 วิธี Digital Differential Analyzer: DDA

วิธี DDA นี้เป็นวงจรที่ได้รับการคิดค้นเพื่อใช้กับ เอ็นซี ซึ่งมีชื่อว่า ตัววิเคราะห์อนุพันธ์ดิจิตอล (Digital Differential Analyzer) สามารถใช้เป็นอินเตอร์โพเลชั่นเส้นตรงในสามมิติ และเป็นอินเตอร์โพเลชั่นเส้นตรงและเส้นโค้งในสองมิติได้



จำนวนครั้ง	0	1	2	3	4	5	6	7	8
r	000	101	010	111	100	001	110	011	000
OFP	0	0	1	0	1	1	0	1	1

รูปที่ 2.28 ทฤษฎีของวิธี DDA

ในการอธิบายทฤษฎีของวิธี DDA จะสมมติว่ามีรีจิสเตอร์ 3 บิตอยู่ 2 ตัวดังแสดงในรูปที่ 2.28 (จงกล งามวิวิทย์, 2535) เมื่อมีพัลซ์คำสั่งบวกเข้ามา เนื้อหา y ของรีจิสเตอร์ Y จะบวกเข้ากับเนื้อหา r ของรีจิสเตอร์ R ทุกครั้งแล้วจึงเก็บเนื้อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หาที่ได้ไว้ในรีจีสเตอร์ R จากรูปที่ 2.28 จะเห็นได้ว่าเมื่อทำการบวกเป็นจำนวน $2^3 = 8$ ครั้งแล้ว รีจีสเตอร์ R จะกลับไปสู่สภาพเดิม แต่ในระหว่างการบวกนี้ จะมีพัลซ์ซึ่งเรียกว่า โอเวอรีโฟลพัลซ์ (OFP) จำนวน 5 พัลซ์ ถูกสร้างออกมา จำนวนพัลซ์นี้จะมีค่าเท่ากับเนื้อหา y = 101 (=5) ของรีจีสเตอร์ Y ถ้าให้พัลซ์คำสั่งบวก 1 พัลซ์เป็น dx และ OFP 1 พัลซ์เป็น dz จะเขียนความสัมพันธ์ซึ่งได้กล่าวมาก่อนนี้เป็น

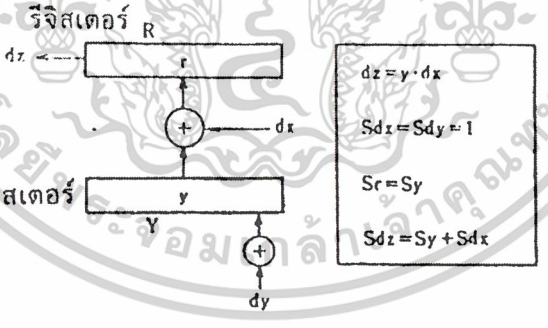
$$dz = \frac{5}{8}.dx$$

การที่จะเขียนสมการของพัลซ์ dx จำนวน 8 พัลซ์ ซึ่งทำให้เกิดพัลซ์ dz จำนวน 5 พัลซ์ เป็นสมการที่เท่ากันนั้นเป็นไปได้ไม่ง่าย จึงต้องคิดค่าประมาณ นั่นคือ ทุกครั้งของพัลซ์ dz จำนวน $\frac{5}{8}$ พัลซ์โดยเฉลี่ย

ดังนั้น ถ้าคิดเช่นนี้แล้ว เนื้อหา y ของรีจีสเตอร์ Y จะไม่คงที่ แต่ให้เปลี่ยนตามพัลซ์ dx ทุกครั้งได้ เมื่อทำให้อยู่ในรูปทั่วไปและมีโครงสร้างของวงจรที่สอดคล้องกับสมการใน รูปที่ 2.29 วงจรนี้เรียกว่า วงจรพื้นฐานของ DDA ดังแสดงไว้

$$dz = y.dx$$

ในรูปที่ 2.29 (จงกล งามวิวิทย์, 2535) อย่างไรก็ตาม สมการในรูปที่ 2.29 เป็นสมการโดยประมาณ การวิเคราะห์ความผิดพลาดทำได้ไม่ง่ายนัก แต่การเลือกใช้จำนวนบิตของรีจีสเตอร์ที่เหมาะสมจะทำให้เกิดประสิทธิภาพในการคำนวนนั้นละเอียดถูกต้องมากยิ่งขึ้น S ที่แสดงใน รูปที่ 2.29 เป็นสเกลแฟคเตอร์ที่กำหนดขึ้นเพื่อใช้ให้เหมาะสม ซึ่งเป็นตัวอย่างที่แสดงไว้เท่านั้น



รูปที่ 2.29 วงจรพื้นฐานของ DDA

2.5.2.1 <u>การอินเตอร์โพเลชันเส้นตรง</u>

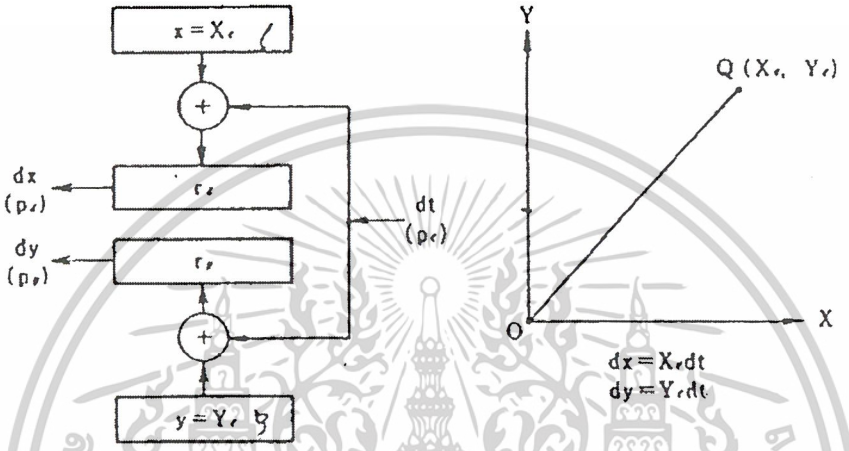
กำหนดให้จุดกำเนิด O เป็นจุดเริ่มต้น มีจุดปลายอยู่ที่ Q (X_e, Y_e) สมการอนุพันธ์เส้นตรง OQ จะเขียนได้เป็น

$$dx = X_e.dx$$

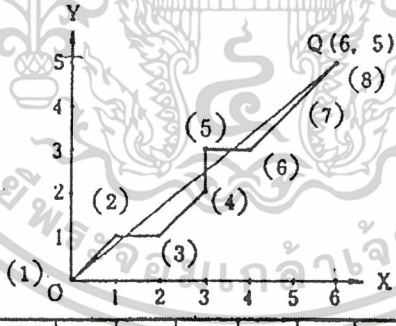
$$dy = Y_e.dx$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่มี t เป็นพารามิเตอร์ของเวลา เมื่อเปรียบเทียบสมการนี้กับสมการในรูปที่ 2.29 แล้วรวมวงจรพื้นฐานเข้าด้วยกัน จะได้วงจรอินเตอร์โพเลชั่นเส้นตรงดังรูปที่ 2.30 (จงกล งามวิวิทย์, 2535) ซึ่งกำหนดให้ dx เป็นพัลซ์คำสั่งของ P_x ของแกน X และ dy เป็นพัลซ์คำสั่ง P_y ของแกน Y สำหรับ dt นั้นเป็นพัลซ์นาฬิกา ถ้าความถี่ของพัลซ์นาฬิกาเปลี่ยน จะทำให้ความถี่ของพัลซ์เปลี่ยนไปด้วย ดังนั้น จึงสามารถควบคุมความถี่ของพัลซ์คำสั่งได้ ในกรณีของสามมิตินั้น เพียงเพิ่มวงจรพื้นฐานสำหรับแกน Z เข้าไปเท่านั้น



รูปที่ 2.30 วงจรอินเตอร์โพเลชั่นแบบเส้นตรง (DDA)



บวก	0	1	2	3	4	5	6	7	8
r_x	000	110	100	010	000	110	100	010	000
dx	0	0	1	1	1	0	1	1	1
r_y	000	101	010	111	100	001	110	011	000
dy	0	0	1	0	1	1	0	1	1

รูปที่ 2.31 การอินเตอร์โพเลชั่นแบบเส้นตรง (DDA)

รูปที่ 2.32 (ในการอธิบายทฤษฎีของวิธี DDA จะสมมติว่ามีรีจีสเตอร์ 3 บิตอยู่ 2 ตัวดังแสดงในรูปที่ 2.28 (จงกล งามวิวิทย์, 2535)) แสดงถึง การอินเตอร์โพเลชั่นเส้นตรงด้วยวิธี DDA ซึ่งได้กำหนด $X_e = 6$, $Y_e = 5$

2.5.2.2 <u>การเขียนอินเตอร์โพเลชั่นเส้นโค้ง</u>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมการที่มีอนุพันธ์ของวงกลมที่มีจุดศูนย์กลาง O นั้นสามารถเขียนได้เป็น

$$dx = y.dx$$

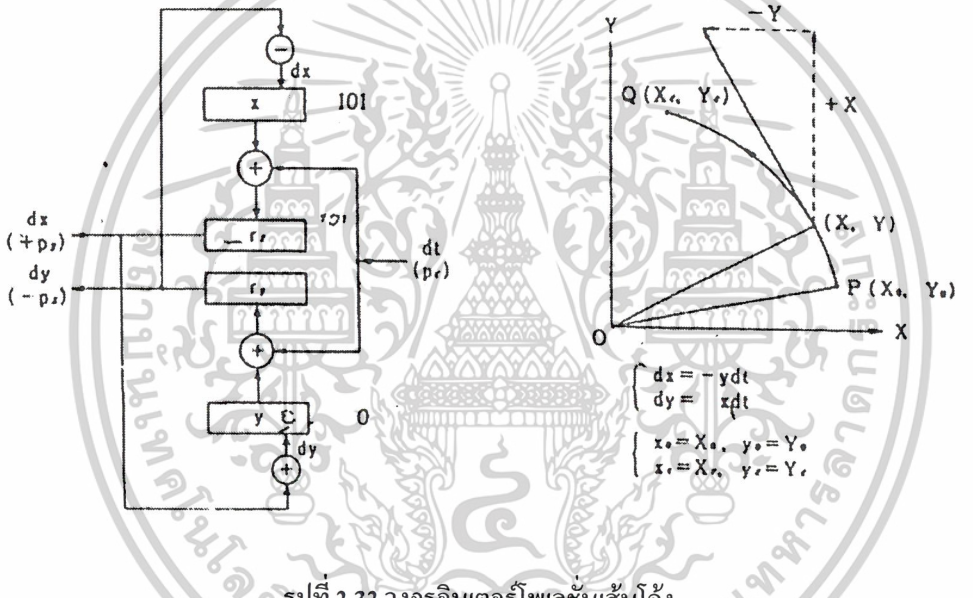
$$dy = x.dx$$

โดยกำหนดเงื่อนไขเริ่มต้นและเงื่อนไขตอนปลายไว้ตามลำดับ ดังนี้

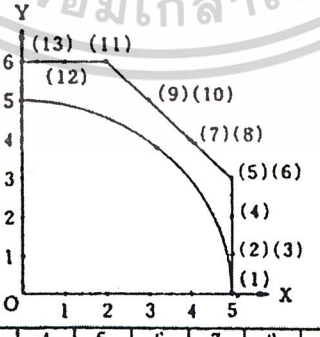
$$x_o = X_o, y_o = Y_o$$

$$x_e = X_e, y_e = Y_e$$

สำหรับค่าของ (X_o, Y_o) และ (X_e, Y_e) เป็นคู่ลำดับของจุดเริ่มต้นและจุดปลายของเส้นโค้งรูปวงกลม อันมีจุดศูนย์กลางอยู่ที่จุดกำเนิด รูปที่ 2.33 (จงกล, 2540)



รูปที่ 2.32 วงจรอินเตอร์โพเลชั่นเส้นโค้ง



บวก	0	1	2	3	4	5	6	7	8	9	10	11	12	13
x	101	101	101	101	101	101	101	101	100	100	011	011	010	001
r_x	000	101	010	111	100	001	110	011	111	011	110	001	011	100
$dx(+p_x)$	0	0	1	0	1	1	0	1	0	1	0	1	0	0
y	000	000	000	001	001	010	011	011	100	100	101	101	110	110
r_y	000	000	000	001	010	100	111	010	110	010	111	100	010	000
$dy(-p_y)$	0	0	0	0	0	0	0	1	0	1	0	1	1	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.33 การอินเตอร์โพเลชั่นเส้นโค้ง

จากสมการในรูปที่ 2.32 สามารถสร้างวงจรอินเตอร์โพเลชั่นเส้นโค้งได้โดยประกอบขึ้นจากวงจรพื้นฐาน ซึ่งวงจรอินเตอร์โพเลชั่นนี้ได้แสดงไว้ในรูปที่ 2.32 เนื่องจากองค์ประกอบของเวคเตอร์ที่สัมผัสจุดใดๆ บนเส้นโค้ง (จุด X,Y) นั้นมีค่าเป็น (-Y,X) ดังนั้น สัญญาณ OFP ของ dx จะต้องเป็นพัลซ์คำสั่ง Y ซึ่งกำหนดเป็น p_y และ dy จะเป็นพัลซ์คำสั่ง X ในทิศทางที่เป็นลบ ซึ่งกำหนดเป็น -Px

ในรูปที่ 2.33 (ในการอธิบายทฤษฎีของวิธี DDA จะสมมติว่ามีรีจีสเตอร์ 3 บิตอยู่ 2 ตัวดังแสดงในรูปที่ 2.28 (จงกล งามวิวิทย์, 2535)) เป็นรูปที่แสดงถึงการทำงานของอินเตอร์โพเลชั่นเส้นโค้ง ซึ่งต้องการอินเตอร์โพเลชั่นเส้นโค้งที่มีจุดเริ่มต้นเป็น (5, 0) และจุดปลายเป็น (0, 5) ส่วนค่าต่างๆ ในตารางเป็นการแสดงถึงการเปลี่ยนแปลงเนื้อหาของรีจีสเตอร์แต่ละตัว

2.5.3 วิธีคำนวณทางพีชคณิต

ในวิธี DDA นั้น ต้องแก้สมการอนุพันธ์แล้วจึงสร้างพัลซ์คำสั่ง แต่สำหรับวิธีการคำนวณทางพีชคณิตตรวจสอบเงื่อนไขแล้วจึงสร้างพัลซ์คำสั่ง ซึ่งวิธีนี้บริษัทฟูจิยีได้คิดค้นมา จากทฤษฎีพื้นฐานจะเห็นได้ว่าไม่สามารถทำอินเตอร์โพเลชั่นในสามมิติได้ แต่ถ้าใช้พิคฟีด (Pick Feed) ในทิศทาง Z แล้วทำการตัดใน 2 มิติซ้ำไปซ้ำมา จะสามารถตัดชิ้นงานที่มีรูปร่างแบบ 3 มิติได้

นอกจากนี้ยังเป็นวิธีที่ได้นำวงจรชดเชยระยะรัศมีของเครื่องมือตัดไว้ในเครื่องเอ็นซีนั่นคือเวลาเขียนโปรแกรมจะเขียนตามค่าต่างๆ ของชิ้นงานสำเร็จรูปโดยไม่ต้องให้สัมพันธ์กับค่าระยะรัศมีของมีด แต่ที่เอ็นซีเทปนั้นจะต้องให้ข้อมูลดังกล่าวพร้อมคำสั่งที่บ่งถึงการใช้โหมดชดเชยระยะรัศมีของเครื่องมือตัดและทิศทางของออฟเซ็ทเวลาที่จะกำหนดรัศมีของเครื่องมือตัดในการใช้งานจริง จะต้องกำหนดจากปุ่มหมุน (Dial) ที่แผงปฏิบัติงานของเครื่องเอ็นซีวิธีนี้จึงไม่จำเป็นต้องสร้างเทปเอ็นซีใหม่ถ้าต้องการเปลี่ยนรัศมีของเครื่องมือตัด หรือการใช้รัศมีต่างออกไปของเครื่องมือตัดเพื่อการทำชิ้นงานอื่น ซึ่งทำให้มีประสิทธิภาพสูงต่อการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบและวิธีการดำเนินงาน

3.1 การวางแผนการดำเนินงาน

3.1.1 ศึกษาระบบการทำงานของเครื่องซีเอ็นซี

- ส่วนประกอบทางกล
- ส่วนประกอบทางไฟฟ้า
- ระบบควบคุมของเครื่อง
- โปรแกรมควบคุมเครื่องจักร

3.1.2 ตรวจสอบส่วนประกอบทางกลและทางไฟฟ้า

เนื่องจากเครื่องได้มีการปล่อยทิ้งไว้นานจึงต้องมีการตรวจสอบส่วนประกอบต่างๆ ว่ามีอุปกรณ์ส่วนไหนบ้างที่มีการชำรุด เสีย หรือหายไป และหาแนวทางการแก้ไขเมื่อเจอปัญหา

- อุปกรณ์มีความชำรุดมากน้อยแค่ไหน
- ใช้อุปกรณ์อื่นแทนได้หรือไม่
- ทำขึ้นมาเองหรือซื้อชิ้นส่วนสำเร็จรูป
- ประสิทธิภาพของการทำงานของอุปกรณ์
- ความเข้ากันได้ของอุปกรณ์ใหม่ที่สร้างขึ้นมา หรือซื้อมาประกอบใหม่
- มีความยืดหยุ่นสูง สามารถรองรับการใช้งานที่หลากหลายในการใช้งานในปัจจุบัน
- งบประมาณของการแก้ไขปัญหา

3.1.3 วางแผนการแก้ไขและปรับปรุง

หลังจากทำการตรวจสอบอุปกรณ์ข้างต้นแล้ว จะมีการวางแผนการทำงานเพื่อให้มีการทำงานอย่างมีประสิทธิภาพ และลดขั้นตอนที่ไม่จำเป็น

3.1.4 ทำการซ่อมหรือเปลี่ยนชิ้นส่วนอุปกรณ์ใหม่

จากการตรวจสอบพร้อมทั้งคำนึงถึงเงื่อนไขข้างต้นก็จะทำการทำงานตามขั้นตอนตามแผนงานที่ได้วางไว้ โดยชิ้นส่วนที่สามารถซ่อมได้ก็จะทำการซ่อมให้ใช้งานได้ตามเดิม และชิ้นส่วนที่ไม่สามารถซ่อมได้หรือขาดหายไปหรือทำการซ่อมแล้วไม่สามารถใช้งานได้ก็ทำการเปลี่ยนชิ้นส่วนอุปกรณ์ใหม่

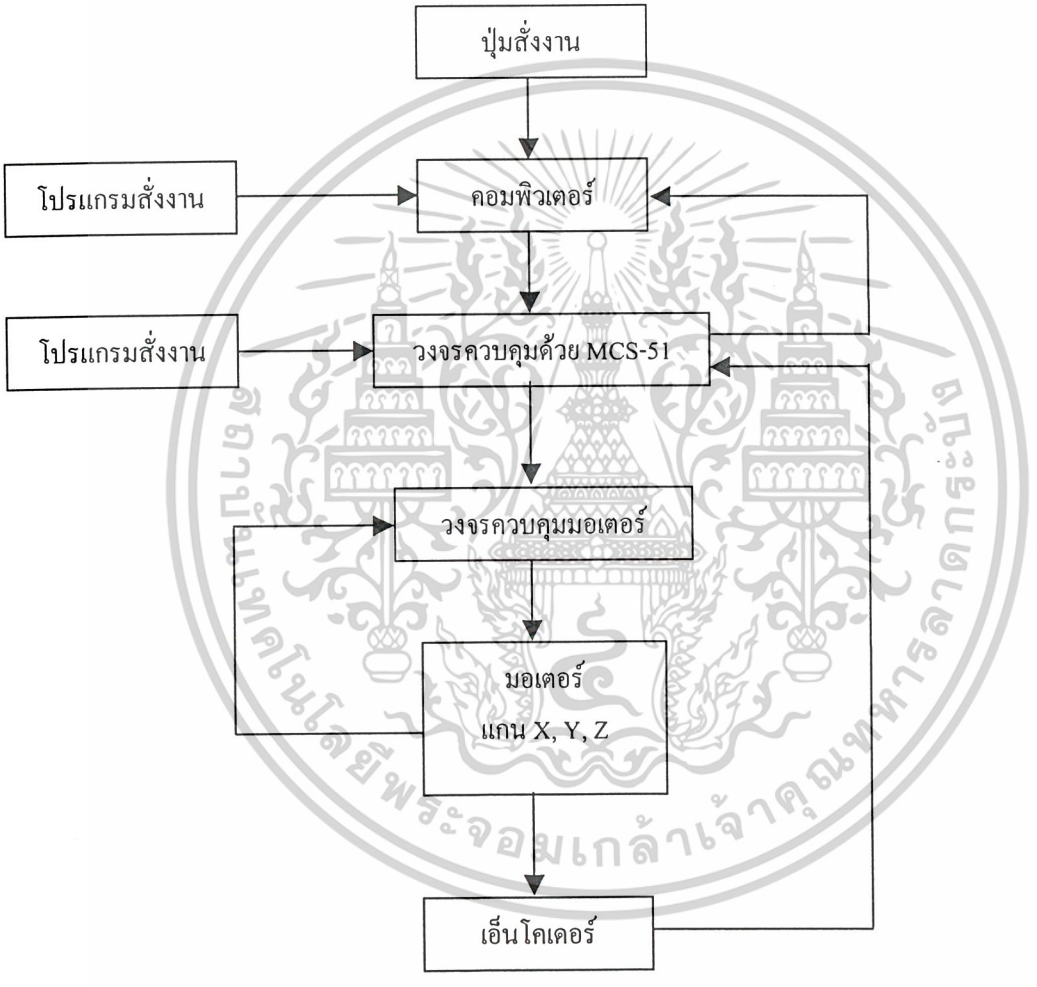
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.5 ทดสอบอุปกรณ์ขั้นสุดท้าย

หลังจากที่ประกอบชิ้นส่วนต่างๆแล้วจะมีการตรวจสอบความเข้ากันได้ของอุปกรณ์ต่างๆ เพราะอาจจะมีอุปกรณ์บางชิ้นมีการติดตั้งที่ไม่ถูกต้องหรือมีการทำงานที่ผิดพลาด

3.2 หลักการทำงานและการออกแบบ

3.2.1 หลักการทำงานโดยรวม



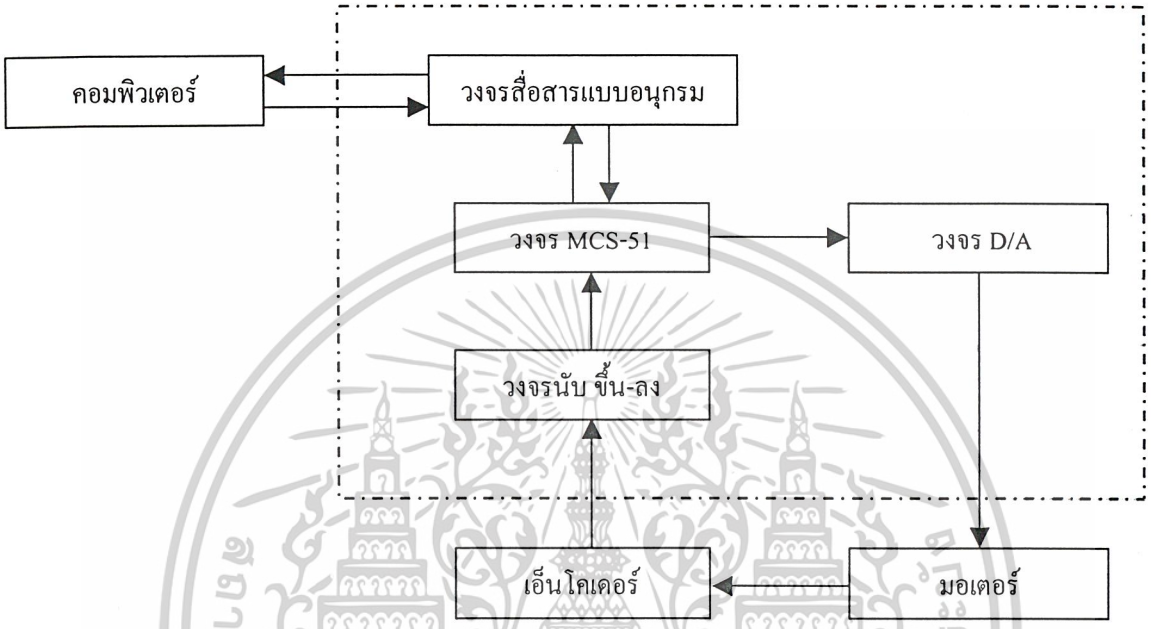
รูปที่ 3.1 หลักการทำงานโดยรวม

เมื่อมีการป้อนจีโค้ด (G-Code) และกดปุ่มเริ่มการทำงาน คอมพิวเตอร์จะประมวลผลจีโค้ด ที่รับมาแปลงเป็นตำแหน่งการเคลื่อนที่ในแนวแกน X, Y, Z และความเร็วการเคลื่อนที่ของมอเตอร์ จากนั้นส่งค่าที่ได้ไปยังวงจรที่ควบคุมด้วยไมโครคอนโทรลเลอร์ MCS-51 ทางพอร์ตอนุกรม (RS-232) ไมโครคอนโทรลเลอร์ MCS-51 จะแปลงค่าที่รับมาสั่งงานให้มอเตอร์เคลื่อนที่ เอ็นโคเดอร์ซึ่งติดอยู่กับมอเตอร์จะทำงานส่งพลัสการเคลื่อนที่กลับไปยังไมโคร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอนโทรลเลอร์ MCS-51 เพื่อตรวจสอบและควบคุมตำแหน่งให้ตรงตามตำแหน่งที่ต้องการ ค่าที่ได้จะถูกส่งต่อไปยังคอมพิวเตอร์เพื่อตรวจสอบและส่งค่ากลับไปอีกจนจบโปรแกรมการทำงาน

3.2.2 วงจรไมโครคอนโทรลเลอร์ MCS-51



รูปที่ 3.2 วงจร MCS-51 ควบคุมการทำงาน

วงจรที่ควบคุมด้วยไมโครคอนโทรลเลอร์ MCS-51 ประกอบไปด้วย

1. วงจรสื่อสารแบบอนุกรม
2. วงจรนับ ขึ้น-ลง
3. วงจรแปลงสัญญานจากสัญญาณดิจิตอลเป็นสัญญาณอนาลอก(D/A)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการดำเนินงาน

4.1 ส่วนประกอบทางกล

ส่วนประกอบทางกลมีการทิ้งไว้นานจึงมีสนิมและบางชิ้นส่วนก็หายไปจึงต้องมีการทำความสะอาด และซ่อมแซมให้สามารถใช้งานได้ ส่วนชิ้นส่วนที่หายไปต้องมีการทำใหม่มาทดแทน ซึ่งส่วนมากก็ยังใช้งานได้ดีอยู่

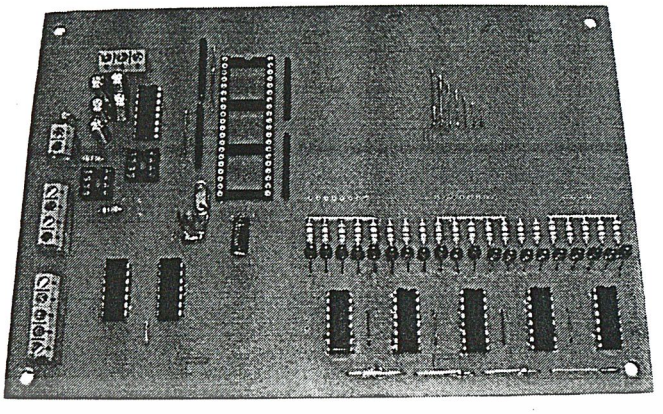
4.2 ส่วนประกอบทางไฟฟ้า

เนื่องจากเครื่องซีเอ็นซีเป็นของเก่าอุปกรณ์ที่มีมากับเครื่องเหลือเพียงชุดควบคุมมอเตอร์เพียงอย่างเดียวจึงต้องออกแบบและติดตั้งส่วนประกอบไฟฟ้าใหม่ทั้งหมด

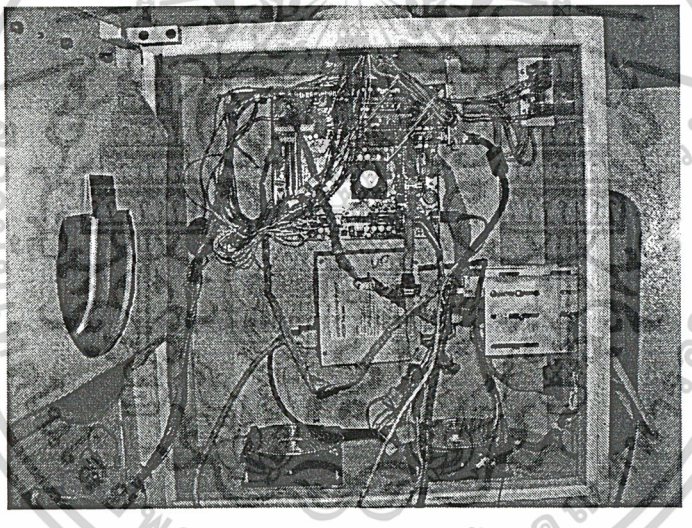
สิ่งที่ได้ติดตั้งใหม่

1. ชุดวงจรควบคุมการเคลื่อนที่ของมอเตอร์ซึ่งได้ประยุกต์ใช้ไมโครคอนโทรลเลอร์ MCS -51
   คุณสมบัติของชุดควบคุมการเคลื่อนที่ของมอเตอร์
   1. ใช้ไมโครคอนโทรลเลอร์ MCS -51 รับค่าตำแหน่งการเคลื่อนที่ของมอเตอร์จากเอ็นโคเดอร์
   2. ใช้โปรแกรมภายในไมโครคอนโทรลเลอร์ MCS -51 ควบคุมการทำงาน
   3. ส่ง-รับข้อมูลระหว่างคอมพิวเตอร์ด้วยพอร์ตอนุกรม
   4. แสดงการเคลื่อนที่ของมอเตอร์เป็นเลขฐานสอง
2. ไมโครคอมพิวเตอร์เพื่อควบคุมการทำงานของเครื่อง
3. ระบบไฟฟ้าที่จ่ายไฟให้วงจรควบคุมทำงานและอุปกรณ์อินพุตและเอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

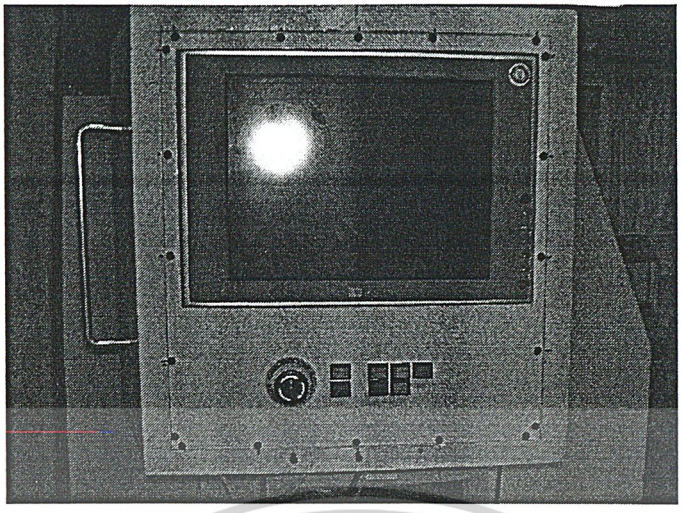


รูปที่ 4.1 ชุดควบคุมที่ใช้ไมโครคอนโทรลเลอร์ MCS -51 สั่งงาน

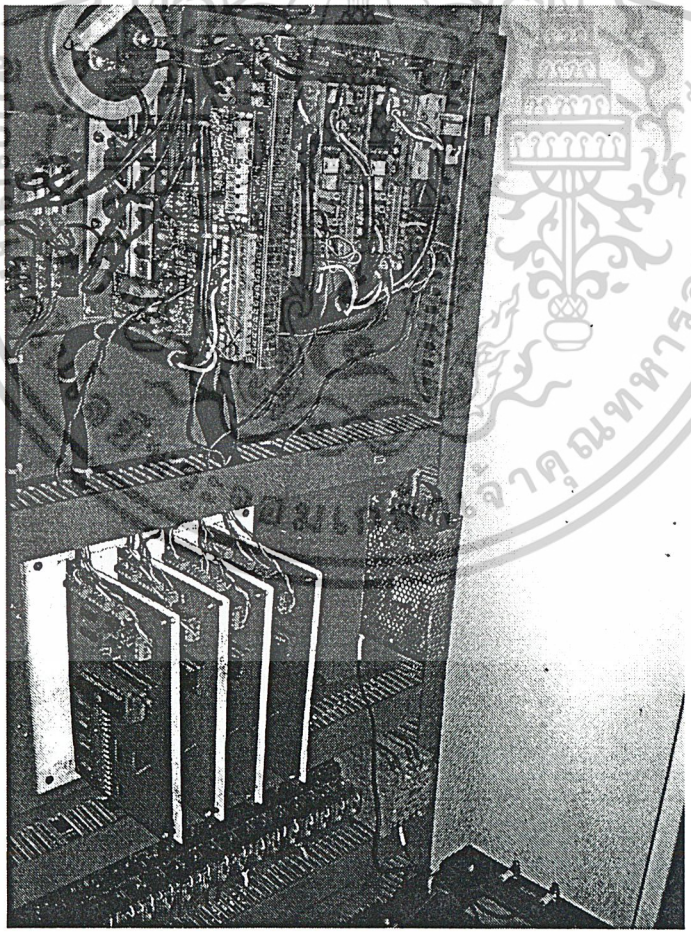


รูปที่ 4.2 หน่วยประมวลผลกลางที่ใช้ไมโครคอมพิวเตอร์ควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 จอแสดงผลและปุ่มสั่งงาน



รูปที่ 4.4 ติดตั้งชุดควบคุมที่ใช้ไมโครคอนโทรลเลอร์ MCS -51 สั่งงาน

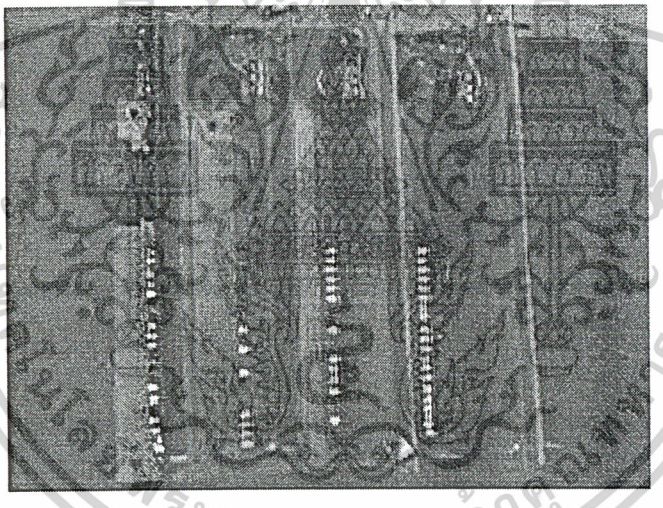
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ส่วนซอร์ฟแวร์

โปรแกรมที่นำมาเขียนเพื่อควบคุมเครื่องจักรเป็นโปรแกรมวิชวลเบสิก 6.0 (Visual Basic 6.0) ซึ่งมีคอนโทรลไว้สำหรับติดต่อกับพอร์ตอนุกรม

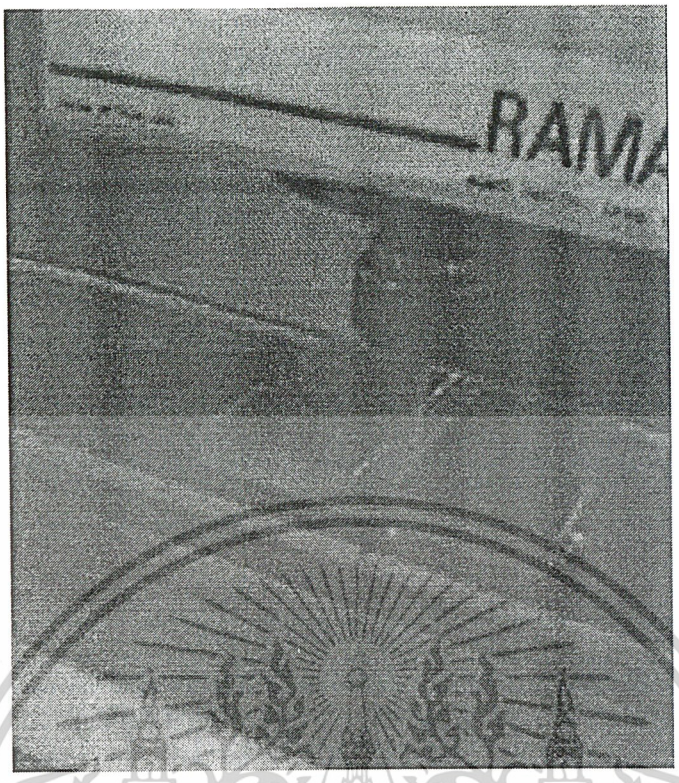
1. โปรแกรมสามารถรับค่าตำแหน่งการเคลื่อนที่โดยมีการกำหนดค่าเป็น แกน X แกน Y และแกน Z ในระบบแอบโซลูทและอินครีเมนทอลมีการเคลื่อนที่สัมพันธ์กันทั้ง 3 แกน และมีดกัดสามารถเคลื่อนที่ตรงตามตำแหน่งที่ต้องการ
2. หากเกิดการผิดพลาดมีดมีการเคลื่อนที่ออกนอกพื้นที่ของโต๊ะจับชิ้นงานก็จะมีการเตือน ผู้ใช้งานสามารถหยุดการทำงานได้ทันที

4.4 ทดสอบกัดชิ้นงาน



รูปที่ 4.5 การแสดงผลเป็นเลขฐานสองของชุดควบคุมที่ใช้ไมโครคอนโทรลเลอร์ MCS -51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 การทดสอบกัดชิ้นงานที่เป็นอะคิริค

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการดำเนินงาน

5.1 สรุปผลการดำเนินงาน

จากผลการทดลองเครื่องซีเอ็นซีที่ผ่านการปรับปรุงสามารถกัดชิ้นงานได้

5.1.1 สรุปผลการดำเนินงานด้านฮาร์ดแวร์

การดำเนินการด้าน ฮาร์ดแวร์ คือ การศึกษาการทำงานระบบควบคุมเครื่องจักร และทำการซ่อมหรือปรับปรุงเครื่องเพื่อให้เครื่องสามารถนำกลับมาใช้งานได้ ซึ่งเป็นเครื่องจักรที่ควบคุมด้วยคอมพิวเตอร์ ผ่านทางพอร์ตอนุกรมของคอมพิวเตอร์ ซึ่งสามารถเคลื่อนที่ในแนวแกน X แกน Y และแกน Z

ผลจากการดำเนินงานในการปรับปรุงเครื่องกัดซีเอ็นซี สามารถดำเนินการตามที่ได้วางแผนไว้ และควบคุมการทำงานทั้งหมดผ่านโปรแกรมควบคุมเครื่องกัดซีเอ็นซีที่ทำการสร้างขึ้น

5.1.2 สรุปผลการดำเนินงานด้านไฟฟ้าและอิเล็คโทรนิคส์

การดำเนินงานด้านไฟฟ้าและอิเล็คโทรนิคส์เป็นอุปกรณ์สำคัญที่ช่วยในการขับเคลื่อนระบบทางกลของเครื่องจักรให้ทำงานตามที่ต้องการ

ผลจากการดำเนินงานด้านไฟฟ้าและอิเล็คโทรนิคส์ คอมพิวเตอร์สามารถสั่งงานชุดควบคุมที่ใช้ไมโครคอนโทรลเลอร์ MCS -51 เพื่อไปสั่งชุดขับมอเตอร์ควบคุมมอเตอร์ให้เคลื่อนที่อย่างสัมพันธ์กันเพื่อกัดชิ้นงานได้ และมีระบบไฟแสดงสถานะการทำงาน เมื่อเกิดข้อผิดพลาดในขณะที่กัดชิ้นงานสามารถสั่งให้เครื่องหยุดการทำงานได้ทันที

5.1.3 สรุปผลการดำเนินงานด้านซอร์ฟแวร์

การดำเนินการด้าน ซอร์ฟแวร์ คือการสร้างโปรแกรมคอมพิวเตอร์ขึ้นมา เพื่อใช้ในการควบคุมงานกัดของเครื่องกัดซีเอ็นซี โดยโปรแกรมคอมพิวเตอร์สามารถรับจีโค้ด (G-Code) ที่ใช้ในการกัดชิ้นงานที่ต้องการ ควบคุมการเคลื่อนที่ของมอเตอร์แกน X แกน Y และแกน Z ให้เคลื่อนที่ตามที่สั่งได้ และเมื่อเกิดข้อผิดพลาดขึ้นขณะที่เครื่องกัดกำลังทำงาน โปรแกรมสามารถหยุดการทำงานได้ในทันที ซึ่งโปรแกรมสามารถทำการควบคุมงานกัดจนสามารถกัดชิ้นงานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 ปัญหาที่พบในการทำงาน

1. ระบบขับเคลื่อนของมอเตอร์ในแนวแกน Y เป็นระบบส่งถ่ายกำลังโดยตรงจากมอเตอร์ซึ่งต่างจากระบบขับเคลื่อนของมอเตอร์แกน X และแกน Z ซึ่งใช้สายพานเป็นตัวส่งถ่ายกำลัง ซึ่งจะทำให้มอเตอร์ในแนวแกน Y นั้นรับภาระมากเกินไปและโปรแกรมที่ใช้ในการสั่งงานมอเตอร์ให้เคลื่อนที่นั้นแตกต่างไป

2. ประตูปิด – เปิดของเครื่องเล็กเกินไปทำให้พื้นที่ที่จะทำงานมีน้อย

3. การ์ดอินพุต/เอาท์พุตเกิดการค้างเมื่อมีการสั่งงานมากเกินไป

4. มีสัญญาณรบกวนเมื่อมีการส่งสัญญาณของเอ็นโคเดอร์ไปยังชุดควบคุมไมโครคอนโทรลเลอร์ MCS -51 ทำให้การเคลื่อนที่ของมอเตอร์เกิดการคลาดเคลื่อน

5. เนื่องจากคอมพิวเตอร์ที่ใช้สั่งงานเป็นรุ่นเก่าทำให้มีการทำงานที่ผิดพลาด

5.3 แนวทางการพัฒนาและปรับปรุง

1. พัฒนาโปรแกรมให้สามารถรับจีโค๊ด (G-Code) ได้อย่างครอบคลุมและสามารถแสดงเส้นทางการเดินของมีดกัด และให้โปรแกรมมีการแจ้งเตือนเมื่อมีการทำงานที่ผิดพลาด

2. ประยุกต์ใช้ไมโครคอนโทรลเลอร์ตระกูลอื่นที่มีความสามารถและเสถียรภาพมากกว่านี้มาใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อ้างอิง

-วรพจน์ กรแก้ววัฒนกุล, ชัยวัฒน์ ลิ้มพรจิตรวิไล, **เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51**, กรุงเทพ: อินโนเวทีฟ เอ็กเพอริเมนต์ จำกัด,2546

-วโรดม ตู้จินดา, **การออกแบบสร้างและรีโทรฟิตชุดควบคุมเครื่องซีเอ็นซี**, กรุงเทพ: สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น), 2547

-รวี อุตตมธนินทร์, **การสร้างต้นแบบส่วนควบคุมเครื่องกัดแนวดิ่งซีเอ็นซี**, กรุงเทพ: สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2541

-จงกล งามวิวิทย์, **การควบคุมเชิงตัวเลข**, กรุงเทพ: สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2535

-อรรถพล บุญยะโภคา, วรพจน์ กรแก้ววัฒนกุล, ชัยวัฒน์ ลิ้มพรจิตรวิไล, **เรียนรู้และปฏิบัติการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกผ่าน พอร์ตอนุกรม**, กรุงเทพ: อินโนเวทีฟ เอ็กเพอริเมนต์ จำกัด, 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Forms

```
VERSION 5.00
Object = "{648A5603-2C6E-101B-82B6-000000000014}#1.1#0"; "MSCOMM32.OCX"
Begin VB.Form Form1
   Caption        =  "Form1"
   ClientHeight   =  6900
   ClientLeft     =  60
   ClientTop      =  345
   ClientWidth    =  7680
   LinkTopic      =  "Form1"
   ScaleHeight    =  6900
   ScaleWidth     =  7680
   StartUpPosition =  3  'Windows Default
   WindowState    =  2  'Maximized
   Begin VB.CommandButton Command5
      Caption        =  "Command5"
      Height         =  495
      Left           =  2880
      TabIndex       =  70
      Top            =  6120
      Width          =  1215
   End
   Begin VB.TextBox Text1
      Height         =  495
      Index          =  2
      Left           =  240
      TabIndex       =  69
      Text           =  "0"
      Top            =  5520
      Width          =  1215
   End
   Begin VB.CommandButton Command2
      Caption        =  "Command2"
      Height         =  495
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Index        = 2
  Left         = 1560
  TabIndex     = 68
  Top          = 5520
  Width        = 1215
End
Begin VB.TextBox Text2
  Height       = 495
  Index        = 2
  Left         = 2880
  TabIndex     = 67
  Text         = "0"
  Top          = 5520
  Width        = 1215
End
Begin VB.CommandButton Command4
  Caption      = "Command4"
  Height       = 495
  Index        = 2
  Left         = 4200
  TabIndex     = 66
  Top          = 5520
  Width        = 1215
End
Begin VB.TextBox Text1
  Height       = 495
  Index        = 1
  Left         = 240
  TabIndex     = 65
  Text         = "0"
  Top          = 4920
  Width        = 1215
End
Begin VB.CommandButton Command2
  Caption      = "Command2"
  Height       = 495
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Index          = 1
  Left           = 1560
  TabIndex       = 64
  Top            = 4920
  Width          = 1215
End
Begin VB.TextBox Text2
  Height         = 495
  Index          = 1
  Left           = 2880
  TabIndex       = 63
  Text           = "0"
  Top            = 4920
  Width          = 1215
End
Begin VB.CommandButton Command4
  Caption        = "Command4"
  Height         = 495
  Index          = 1
  Left           = 4200
  TabIndex       = 62
  Top            = 4920
  Width          = 1215
End
Begin VB.CommandButton Command4
  Caption        = "Command4"
  Height         = 495
  Index          = 0
  Left           = 4200
  TabIndex       = 61
  Top            = 4320
  Width          = 1215
End
Begin VB.TextBox Text2
  Height         = 495
  Index          = 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Left        = 2880
  TabIndex    = 60
  Text        = "0"
  Top         = 4320
  Width       = 1215
End
Begin VB.CommandButton Command3
  Caption     = "Command3"
  Height      = 495
  Left        = 240
  TabIndex    = 59
  Top         = 2760
  Width       = 1215
End
Begin VB.CommandButton Command2
  Caption     = "Command2"
  Height      = 495
  Index       = 0
  Left        = 1560
  TabIndex    = 58
  Top         = 4320
  Width       = 1215
End
Begin VB.TextBox Text1
  Height      = 495
  Index       = 0
  Left        = 240
  TabIndex    = 57
  Text        = "0"
  Top         = 4320
  Width       = 1215
End
Begin VB.TextBox speed
  Height      = 495
  Index       = 3
  Left        = 600
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  TabIndex      = 56
  Text          = "127"
  Top           = 1920
  Width         = 1215
End
Begin VB.CommandButton turn_on
  Caption       = "Z"
  Height        = 495
  Index         = 3
  Left          = 1920
  TabIndex      = 55
  Top           = 1920
  Width         = 1215
End
Begin VB.TextBox speed
  Height        = 495
  Index         = 2
  Left          = 600
  TabIndex      = 54
  Text          = "127"
  Top           = 1320
  Width         = 1215
End
Begin VB.CommandButton turn_on
  Caption       = "Y"
  Height        = 495
  Index         = 2
  Left          = 1920
  TabIndex      = 53
  Top           = 1320
  Width         = 1215
End
Begin VB.TextBox speed
  Height        = 495
  Index         = 1
  Left          = 600
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  TabIndex      = 52
  Text          = "127"
  Top           = 720
  Width         = 1215
End
Begin VB.CommandButton turn_on
  Caption       = "X"
  Height        = 495
  Index         = 1
  Left          = 1920
  TabIndex      = 51
  Top           = 720
  Width         = 1215
End
Begin VB.TextBox speed
  Height        = 495
  Index         = 0
  Left          = 600
  TabIndex      = 49
  Text          = "127"
  Top           = 120
  Width         = 1215
End
Begin MSCommLib.MSComm MSComm1
  Index         = 0
  Left          = 120
  Top           = 120
  _ExtentX      = 1005
  _ExtentY      = 1005
  _Version      = 393216
  CommPort      = 5
  DTREnable     = -1 'True
  BaudRate      = 38400
End
Begin VB.CommandButton turn_on
  Caption       = "Spindle"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Height        = 495
  Index         = 0
  Left          = 1920
  TabIndex      = 50
  Top           = 120
  Width         = 1215
End
Begin VB.CommandButton Command1
  Caption       = "Clear All"
  Height        = 495
  Left          = 6840
  TabIndex      = 48
  Top           = 5520
  Width         = 1215
End
Begin VB.CommandButton CYC_OFF_OFF
  Caption       = "CYC_OFF_OFF"
  Height        = 495
  Left          = 8280
  TabIndex      = 47
  Top           = 3840
  Width         = 1215
End
Begin VB.CommandButton CYC_OFF_ON
  Caption       = "CYC_OFF_ON"
  Height        = 495
  Left          = 8280
  TabIndex      = 46
  Top           = 3360
  Width         = 1215
End
Begin VB.CommandButton NC_OFF_OFF
  Caption       = "NC_OFF_OFF"
  Height        = 495
  Left          = 8280
  TabIndex      = 45
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Top        = 1680
  Width      = 1215
End
Begin VB.CommandButton NC_OFF_ON
  Caption    = "NC_OFF_ON"
  Height     = 495
  Left       = 8280
  TabIndex   = 44
  Top        = 1200
  Width      = 1215
End
Begin VB.CommandButton OTH_OFF
  Caption    = "OTH_OFF"
  Height     = 495
  Left       = 9720
  TabIndex   = 43
  Top        = 1680
  Width      = 1215
End
Begin VB.CommandButton OTH_ON
  Caption    = "OTH_ON"
  Height     = 495
  Left       = 9720
  TabIndex   = 42
  Top        = 1200
  Width      = 1215
End
Begin VB.CommandButton COL_OFF
  Caption    = "COL_OFF"
  Height     = 495
  Left       = 9720
  TabIndex   = 41
  Top        = 600
  Width      = 1215
End
Begin VB.CommandButton COL_ON
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Caption      = "COL_ON"
  Height       = 495
  Left         = 9720
  TabIndex     = 40
  Top          = 120
  Width        = 1215
End
Begin VB.CommandButton OUT_OFF
  Caption      = "OUT_OFF"
  Height       = 495
  Left         = 9720
  TabIndex     = 39
  Top          = 1680
  Width        = 1215
End
Begin VB.CommandButton OUT_ON
  Caption      = "OUT_ON"
  Height       = 495
  Left         = 9720
  TabIndex     = 38
  Top          = 1200
  Width        = 1215
End
Begin VB.CommandButton CLO_OFF
  Caption      = "CLO_OFF"
  Height       = 495
  Left         = 9720
  TabIndex     = 37
  Top          = 600
  Width        = 1215
End
Begin VB.CommandButton CLO_ON
  Caption      = "CLO_ON"
  Height       = 495
  Left         = 9720
  TabIndex     = 36
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Top             =  120
  Width           =  1215
End
Begin VB.CommandButton OPT_OFF
  Caption         =  "OPT_OFF"
  Height          =  495
  Left            =  8280
  TabIndex        =  35
  Top             =  4920
  Width           =  1215
End
Begin VB.CommandButton OPT_ON
  Caption         =  "OPT_ON"
  Height          =  495
  Left            =  8280
  TabIndex        =  34
  Top             =  4440
  Width           =  1215
End
Begin VB.CommandButton CYC_ON_OFF
  Caption         =  "CYC_ON_OFF"
  Height          =  495
  Left            =  8280
  TabIndex        =  33
  Top             =  2760
  Width           =  1215
End
Begin VB.CommandButton CYC_ON_ON
  Caption         =  "CYC_ON_ON"
  Height          =  495
  Left            =  8280
  TabIndex        =  32
  Top             =  2280
  Width           =  1215
End
Begin VB.CommandButton NC_ON_OFF
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Caption      = "NC_ON_OFF"
  Height       = 495
  Left         = 8280
  TabIndex     = 31
  Top          = 600
  Width        = 1215
End
Begin VB.CommandButton NC_ON_ON
  Caption      = "NC_ON_ON"
  Height       = 495
  Left         = 8280
  TabIndex     = 30
  Top          = 120
  Width        = 1215
End
Begin VB.CommandButton GRE_LAMP_OFF
  Caption      = "GRE_LAMP_OFF"
  Height       = 495
  Left         = 9720
  TabIndex     = 29
  Top          = 4920
  Width        = 1215
End
Begin VB.CommandButton GRE_LAMP_ON
  Caption      = "GRE_LAMP_ON"
  Height       = 495
  Left         = 9720
  TabIndex     = 28
  Top          = 4440
  Width        = 1215
End
Begin VB.CommandButton YEL_LAMP_OFF
  Caption      = "YEL_LAMP_OFF"
  Height       = 495
  Left         = 9720
  TabIndex     = 27
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Top        = 3840
  Width      = 1215
End
Begin VB.CommandButton YEL_LAMP_ON
  Caption    = "YEL_LAMP_ ON"
  Height     = 495
  Left       = 9720
  TabIndex   = 26
  Top        = 3360
  Width      = 1215
End
Begin VB.CommandButton RAD_LAMP_OFF
  Caption    = "RAD_LAMP_ OFF"
  Height     = 495
  Left       = 9720
  TabIndex   = 25
  Top        = 2760
  Width      = 1215
End
Begin VB.CommandButton RAD_LAMP_ON
  Caption    = "RAD_LAMP_ ON"
  Height     = 495
  Left       = 9720
  TabIndex   = 24
  Top        = 2280
  Width      = 1215
End
Begin VB.CommandButton INH_Z_OFF
  Caption    = "INH_Z_OFF"
  Height     = 495
  Left       = 6840
  TabIndex   = 23
  Top        = 4920
  Width      = 1215
End
Begin VB.CommandButton INH_Z_ON
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Caption      = "INH_Z_ON"
  Height       = 495
  Left         = 6840
  TabIndex     = 22
  Top          = 4440
  Width        = 1215
End
Begin VB.CommandButton INH_Y_OFF
  Caption      = "INH_Y_OFF"
  Height       = 495
  Left         = 6840
  TabIndex     = 21
  Top          = 3840
  Width        = 1215
End
Begin VB.CommandButton INH_Y_ON
  Caption      = "INH_Y_ON"
  Height       = 495
  Left         = 6840
  TabIndex     = 20
  Top          = 3360
  Width        = 1215
End
Begin VB.CommandButton INH_X_OFF
  Caption      = "INH_X_OFF"
  Height       = 495
  Left         = 6840
  TabIndex     = 19
  Top          = 2760
  Width        = 1215
End
Begin VB.CommandButton INH_X_ON
  Caption      = "INH_X_ON"
  Height       = 495
  Left         = 6840
  TabIndex     = 18
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Top        = 2280
  Width      = 1215
End
Begin VB.Timer Timer1
  Interval   = 1
  Left       = 600
  Top        = 3600
End
Begin VB.CommandButton Mag_OFF
  Caption    = "Mag OFF"
  Height     = 495
  Left       = 6840
  TabIndex   = 3
  Top        = 600
  Width      = 1215
End
Begin VB.CommandButton Mag_On
  Caption    = "Mag ON"
  Height     = 495
  Left       = 6840
  TabIndex   = 2
  Top        = 120
  Width      = 1215
End
Begin VB.CommandButton INH_S_OFF
  Caption    = "INH_S_OFF"
  Height     = 495
  Left       = 6840
  TabIndex   = 1
  Top        = 1680
  Width      = 1215
End
Begin VB.CommandButton INH_S_ON
  Caption    = "INH_S_ON"
  Height     = 495
  Left       = 6840
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  TabIndex      = 0
  Top           = 1200
  Width         = 1215
End
Begin MSCommLib.MSComm MSComm1
  Index         = 1
  Left          = 120
  Top           = 720
  _ExtentX      = 1005
  _ExtentY      = 1005
  _Version      = 393216
  CommPort      = 6
  DTREnable     = -1  'True
  BaudRate      = 38400
End
Begin MSCommLib.MSComm MSComm1
  Index         = 2
  Left          = 120
  Top           = 1320
  _ExtentX      = 1005
  _ExtentY      = 1005
  _Version      = 393216
  CommPort      = 7
  DTREnable     = -1  'True
  BaudRate      = 38400
End
Begin MSCommLib.MSComm MSComm1
  Index         = 3
  Left          = 120
  Top           = 1920
  _ExtentX      = 1005
  _ExtentY      = 1005
  _Version      = 393216
  CommPort      = 8
  DTREnable     = -1  'True
  BaudRate      = 38400
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End
Begin VB.Label Label1
  Caption     = "EMERGENCY"
  Height      = 375
  Index       = 13
  Left        = 5520
  TabIndex    = 17
  Top         = 2520
  Width       = 1095
End
Begin VB.Label Label1
  Caption     = "OUT_SW"
  Height      = 375
  Index       = 12
  Left        = 5520
  TabIndex    = 16
  Top         = 2040
  Width       = 735
End
Begin VB.Label Label1
  Caption     = "COL_ON/OFF"
  Height      = 375
  Index       = 11
  Left        = 5520
  TabIndex    = 15
  Top         = 1560
  Width       = 1095
End
Begin VB.Label Label1
  Caption     = "OPT_ON/OFF"
  Height      = 375
  Index       = 10
  Left        = 5520
  TabIndex    = 14
  Top         = 1080
  Width       = 1095
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End
Begin VB.Label Label1
   Caption         =  "CYC_OFF"
   Height          =  375
   Index           =  9
   Left            =  5520
   TabIndex        =  13
   Top             =  600
   Width           =  735
End
Begin VB.Label Label1
   Caption         =  "CYC_ON"
   Height          =  375
   Index           =  8
   Left            =  5520
   TabIndex        =  12
   Top             =  120
   Width           =  735
End
Begin VB.Label Label1
   Caption         =  "NC_OFF"
   Height          =  375
   Index           =  7
   Left            =  4080
   TabIndex        =  11
   Top             =  3480
   Width           =  735
End
Begin VB.Label Label1
   Caption         =  "NC_ON"
   Height          =  375
   Index           =  6
   Left            =  4080
   TabIndex        =  10
   Top             =  3000
   Width           =  735
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End
Begin VB.Label Label1
  Caption     = "IN_3"
  Height      = 375
  Index       = 5
  Left        = 4080
  TabIndex    = 9
  Top         = 2520
  Width       = 735
End
Begin VB.Label Label1
  Caption     = "IN_2"
  Height      = 375
  Index       = 4
  Left        = 4080
  TabIndex    = 8
  Top         = 2040
  Width       = 735
End
Begin VB.Label Label1
  Caption     = "DOOR"
  Height      = 375
  Index       = 3
  Left        = 4080
  TabIndex    = 7
  Top         = 1560
  Width       = 735
End
Begin VB.Label Label1
  Caption     = "LIT_Z"
  Height      = 375
  Index       = 2
  Left        = 4080
  TabIndex    = 6
  Top         = 1080
  Width       = 735
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End
Begin VB.Label Label1
  Caption     = "LIT_Y"
  Height      = 375
  Index       = 1
  Left        = 4080
  TabIndex    = 5
  Top         = 600
  Width       = 735
End
Begin VB.Label Label1
  Caption     = "LIT_X"
  Height      = 375
  Index       = 0
  Left        = 4080
  TabIndex    = 4
  Top         = 120
  Width       = 735
End
Begin VB.Shape Shape1
  FillStyle   = 0 'Solid
  Height      = 375
  Index       = 15
  Left        = 5040
  Shape       = 2 'Oval
  Top         = 3480
  Width       = 375
End
Begin VB.Shape Shape1
  FillStyle   = 0 'Solid
  Height      = 375
  Index       = 14
  Left        = 5040
  Shape       = 2 'Oval
  Top         = 3000
  Width       = 375
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End
Begin VB.Shape Shape1
  FillStyle     = 0 'Solid
  Height        = 375
  Index         = 13
  Left          = 5040
  Shape         = 2 'Oval
  Top           = 2520
  Width         = 375
End
Begin VB.Shape Shape1
  FillStyle     = 0 'Solid
  Height        = 375
  Index         = 12
  Left          = 5040
  Shape         = 2 'Oval
  Top           = 2040
  Width         = 375
End
Begin VB.Shape Shape1
  FillStyle     = 0 'Solid
  Height        = 375
  Index         = 11
  Left          = 5040
  Shape         = 2 'Oval
  Top           = 1560
  Width         = 375
End
Begin VB.Shape Shape1
  FillStyle     = 0 'Solid
  Height        = 375
  Index         = 10
  Left          = 5040
  Shape         = 2 'Oval
  Top           = 1080
  Width         = 375
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End
Begin VB.Shape Shape1
  FillStyle     = 0 'Solid
  Height        = 375
  Index         = 9
  Left          = 5040
  Shape         = 2 'Oval
  Top           = 600
  Width         = 375
End
Begin VB.Shape Shape1
  FillStyle     = 0 'Solid
  Height        = 375
  Index         = 8
  Left          = 5040
  Shape         = 2 'Oval
  Top           = 120
  Width         = 375
End
Begin VB.Shape Shape1
  FillStyle     = 0 'Solid
  Height        = 375
  Index         = 7
  Left          = 3600
  Shape         = 2 'Oval
  Top           = 3480
  Width         = 375
End
Begin VB.Shape Shape1
  FillStyle     = 0 'Solid
  Height        = 375
  Index         = 6
  Left          = 3600
  Shape         = 2 'Oval
  Top           = 3000
  Width         = 375
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End
Begin VB.Shape Shape1
  FillStyle     =  0  'Solid
  Height        =  375
  Index         =  5
  Left          =  3600
  Shape         =  2  'Oval
  Top           =  2520
  Width         =  375
End
Begin VB.Shape Shape1
  FillStyle     =  0  'Solid
  Height        =  375
  Index         =  4
  Left          =  3600
  Shape         =  2  'Oval
  Top           =  2040
  Width         =  375
End
Begin VB.Shape Shape1
  FillStyle     =  0  'Solid
  Height        =  375
  Index         =  3
  Left          =  3600
  Shape         =  2  'Oval
  Top           =  1560
  Width         =  375
End
Begin VB.Shape Shape1
  FillStyle     =  0  'Solid
  Height        =  375
  Index         =  2
  Left          =  3600
  Shape         =  2  'Oval
  Top           =  1080
  Width         =  375
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End
Begin VB.Shape Shape1
  FillStyle     =  0  'Solid
  Height        =  375
  Index         =  1
  Left          =  3600
  Shape         =  2  'Oval
  Top           =  600
  Width         =  375
End
Begin VB.Shape Shape1
  FillStyle     =  0  'Solid
  Height        =  375
  Index         =  0
  Left          =  3600
  Shape         =  2  'Oval
  Top           =  120
  Width         =  375
End
End
Attribute VB_Name = "Form1"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Dim cont As Boolean
Private Sub Command1_Click()
Out &H100, 0
Out &H101, 0
End Sub
Private Sub Command5_Click()
If cont = True Then cont = False Else cont = True
End Sub
Private Sub Timer1_Timer()
Dim i As Byte
For i = 0 To 15
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
If readin(i) = True Then Shape1(i).FillColor = vbRed Else Shape1(i).FillColor = vbBlack
Next i
Command2(0) = True
Command2(1) = True
Command2(2) = True
If cont = True Then
Command4(0) = True
Command4(1) = True
Command4(2) = True
End If
End Sub
Private Sub Command2_Click(Index As Integer)
a$ = "@J000"
MSComm1(Index + 1).Output = a$
'For i = 1 To 100
'Next i
a$ = ""
a$ = MSComm1(Index + 1).Input
If a$ <> "" And Len(a$) = 3 Then
aa = ((Asc(Mid(a$, 1, 1))) * (256 ^ 2)) + ((Asc(Mid(a$, 2, 1))) * (256 ^ 1)) + (Asc(Mid(a$, 3, 1)) * (256 ^ 0))
Text1(Index) = aa
DoEvents
End If
End Sub
Private Sub Command4_Click(Index As Integer)
kk! = Int((Val(Text1(Index)) - Val(Text2(Index))) / 500)
If kk! > 127 Then kk! = 127
If kk! < -128 Then kk! = -128
k! = 127 - 5 * kk!
If k! > 255 Then k! = 255
If k! < 0 Then k! = 0
a$ = "@T0" + Trim(Chr(k!)) + "0"
MSComm1(Index + 1).Output = a$
End Sub
Private Sub Command3_Click()
speed(1).Text = "120"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
turn_on(1).Value = True
inhx True
Do Until readin(0) = True
Command2(0) = True
Loop
inhx False
speed(1).Text = "127"
turn_on(1).Value = True
speed(2).Text = "120"
turn_on(2).Value = True
inhy True
Do Until readin(1) = True
Command2(1) = True
Loop
inhy False
speed(2).Text = "127"
turn_on(2).Value = True
speed(3).Text = "140"
turn_on(3).Value = True
inhz True
Do Until readin(2) = True
Command2(2) = True
Loop
inhz False
speed(3).Text = "127"
turn_on(3).Value = True
a$ = "@R000"
MSComm1(1).Output = a$
MSComm1(2).Output = a$
MSComm1(3).Output = a$
End Sub
Private Sub Form_Load()
MSComm1(0).PortOpen = True
MSComm1(1).PortOpen = True
MSComm1(2).PortOpen = True
MSComm1(3).PortOpen = True
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Sub
Private Sub Form_Unload(Cancel As Integer)
MSComm1(0).PortOpen = False
MSComm1(1).PortOpen = False
MSComm1(2).PortOpen = False
MSComm1(3).PortOpen = False
End Sub
Private Sub COL_OFF_Click()
col False
End Sub
Private Sub COL_ON_Click()
col True
End Sub
Private Sub CYC_OFF_OFF_Click()
cy_sp False
End Sub
Private Sub CYC_OFF_ON_Click()
cy_sp True
End Sub
Private Sub CYC_ON_OFF_Click()
cy_st False
End Sub
Private Sub CYC_ON_ON_Click()
cy_st True
End Sub
Private Sub GRE_LAMP_OFF_Click()
lgreen False
End Sub
Private Sub GRE_LAMP_ON_Click()
lgreen True
End Sub
Private Sub INH_S_OFF_Click()
inhs False
End Sub
Private Sub INH_S_ON_Click()
inhs True
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Sub
Private Sub INH_X_OFF_Click()
inhx False
End Sub
Private Sub INH_X_ON_Click()
inhx True
End Sub
Private Sub INH_Y_OFF_Click()
inhy False
End Sub
Private Sub INH_Y_ON_Click()
inhy True
End Sub
Private Sub INH_Z_OFF_Click()
inhz False
End Sub
Private Sub INH_Z_ON_Click()
inhz True
End Sub
Private Sub Mag_OFF_Click()
mag False
End Sub
Private Sub Mag_On_Click()
mag True
End Sub
Private Sub nc_on_Click()
nc_on
End Sub
Private Sub NC_OFF_OFF_Click()
nc_off False
End Sub
Private Sub NC_OFF_ON_Click()
nc_off True
End Sub
Private Sub NC_ON_OFF_Click()
nc_on False
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Sub
Private Sub NC_ON_ON_Click()
nc_on True
End Sub
Private Sub OPT_OFF_Click()
opt False
End Sub
Private Sub OPT_ON_Click()
opt True
End Sub
Private Sub OTH_OFF_Click()
other False
End Sub
Private Sub OTH_ON_Click()
other True
End Sub
Private Sub RAD_LAMP_OFF_Click()
lred False
End Sub
Private Sub RAD_LAMP_ON_Click()
lred True
End Sub
Private Sub YEL_LAMP_OFF_Click()
lyellow False
End Sub
Private Sub YEL_LAMP_ON_Click()
lyellow True
End Sub
Private Sub turn_on_Click(Index As Integer)
a$ = "@T0" + Trim(Chr(speed(Index))) + "0"
'a$ = "@R000"
MSComm1(Index).Output = a$
End Sub
Sub tmp()
Dim zzz As Long
Dim zx(20) As Byte
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
zzz = Val(Text2)
If zzz >= 2 ^ 19 Then zx(19) = 1: zzz = zzz - 2 ^ 19
If zzz >= 2 ^ 18 Then zx(18) = 1: zzz = zzz - 2 ^ 18
If zzz >= 2 ^ 17 Then zx(17) = 1: zzz = zzz - 2 ^ 17
If zzz >= 2 ^ 16 Then zx(16) = 1: zzz = zzz - 2 ^ 16
If zzz >= 2 ^ 15 Then zx(15) = 1: zzz = zzz - 2 ^ 15
If zzz >= 2 ^ 14 Then zx(14) = 1: zzz = zzz - 2 ^ 14
If zzz >= 2 ^ 13 Then zx(13) = 1: zzz = zzz - 2 ^ 13
If zzz >= 2 ^ 12 Then zx(12) = 1: zzz = zzz - 2 ^ 12
If zzz >= 2 ^ 11 Then zx(11) = 1: zzz = zzz - 2 ^ 11
If zzz >= 2 ^ 10 Then zx(10) = 1: zzz = zzz - 2 ^ 10
If zzz >= 2 ^ 9 Then zx(9) = 1: zzz = zzz - 2 ^ 9
If zzz >= 2 ^ 8 Then zx(8) = 1: zzz = zzz - 2 ^ 8
If zzz >= 2 ^ 7 Then zx(7) = 1: zzz = zzz - 2 ^ 7
If zzz >= 2 ^ 6 Then zx(6) = 1: zzz = zzz - 2 ^ 6
If zzz >= 2 ^ 5 Then zx(5) = 1: zzz = zzz - 2 ^ 5
If zzz >= 2 ^ 4 Then zx(4) = 1: zzz = zzz - 2 ^ 4
If zzz >= 2 ^ 3 Then zx(3) = 1: zzz = zzz - 2 ^ 3
If zzz >= 2 ^ 2 Then zx(2) = 1: zzz = zzz - 2 ^ 2
If zzz >= 2 ^ 1 Then zx(1) = 1: zzz = zzz - 2 ^ 1
If zzz >= 2 ^ 0 Then zx(0) = 1
'For i = 19 To 0 Step -1
'Text2 = Text2 + Trim(Str(zx(i)))
'Next i
For i = 0 To 7
k1% = k1% + (zx(i) * (2 ^ i))
Next i
For i = 0 To 7
k2% = k2% + (zx(i + 8) * (2 ^ i))
Next i
For i = 0 To 3
k3% = k3% + (zx(i + 16) * (2 ^ i))
Next i
t$ = "@P" + Chr(k3%) + Chr(k2%) + Chr(k1%)
'Text2 = t$
MSComm1(1).Output = t$
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End Sub

Modules

```
Attribute VB_Name = "iocontrol"
Public bito(8) As Byte
'Declare Inp and Out for port I/O
#If Win32 Then
   Public Declare Sub Out Lib "io.dll" Alias "PortOut" (ByVal Port As Integer, ByVal Data As Byte)
   Public Declare Function Inp Lib "io.dll" Alias "PortIn" (ByVal Port As Integer) As Byte
#Else
   Declare Function Inp Lib "InpOut.DLL" (ByVal Port As Integer) As Byte
   Declare Sub Out Lib "InpOut.DLL" (ByVal Port As Integer, ByVal Value As Byte)
#End If
Public op(15) As Boolean
Function readin(k As Byte) As Boolean
Dim jx As Byte
If k <= 7 Then
jx = Inp(&H100)
If (jx And (2 ^ k)) = 2 ^ k Then readin = True Else readin = False
End If
If k > 7 And k < 16 Then
kk = k - 8
jx = Inp(&H101)
If (jx And (2 ^ kk)) = (2 ^ kk) Then readin = True Else readin = False
End If
End Function
Sub inhs(k As Boolean)
op(0) = k
outx
End Sub
Sub inhx(k As Boolean)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
op(1) = k
outx
End Sub
Sub inhy(k As Boolean)
op(2) = k
outx
End Sub
Sub inhz(k As Boolean)
op(3) = k
outx
End Sub
Sub lred(k As Boolean)
op(4) = k
outx
End Sub
Sub lyellow(k As Boolean)
op(5) = k
outx
End Sub
Sub lgreen(k As Boolean)
op(6) = k
outx
End Sub
Sub mag(k As Boolean)
op(7) = k
outx
End Sub
Sub col(k As Boolean)
op(8) = k
outx
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Sub nc_on(k As Boolean)
op(9) = k
outx
End Sub
Sub nc_off(k As Boolean)
op(10) = k
outx
End Sub
Sub cy_st(k As Boolean)
op(11) = k
outx
End Sub
Sub cy_sp(k As Boolean)
op(12) = k
outx
End Sub
Sub opt(k As Boolean)
op(13) = k
outx
End Sub
Sub coolant(k As Boolean)
op(14) = k
outx
End Sub
Sub other(k As Boolean)
op(15) = k
outx
End Sub
Sub outx()
For i = 0 To 7
opx = opx + Abs(op(i) * 2 ^ i)
```

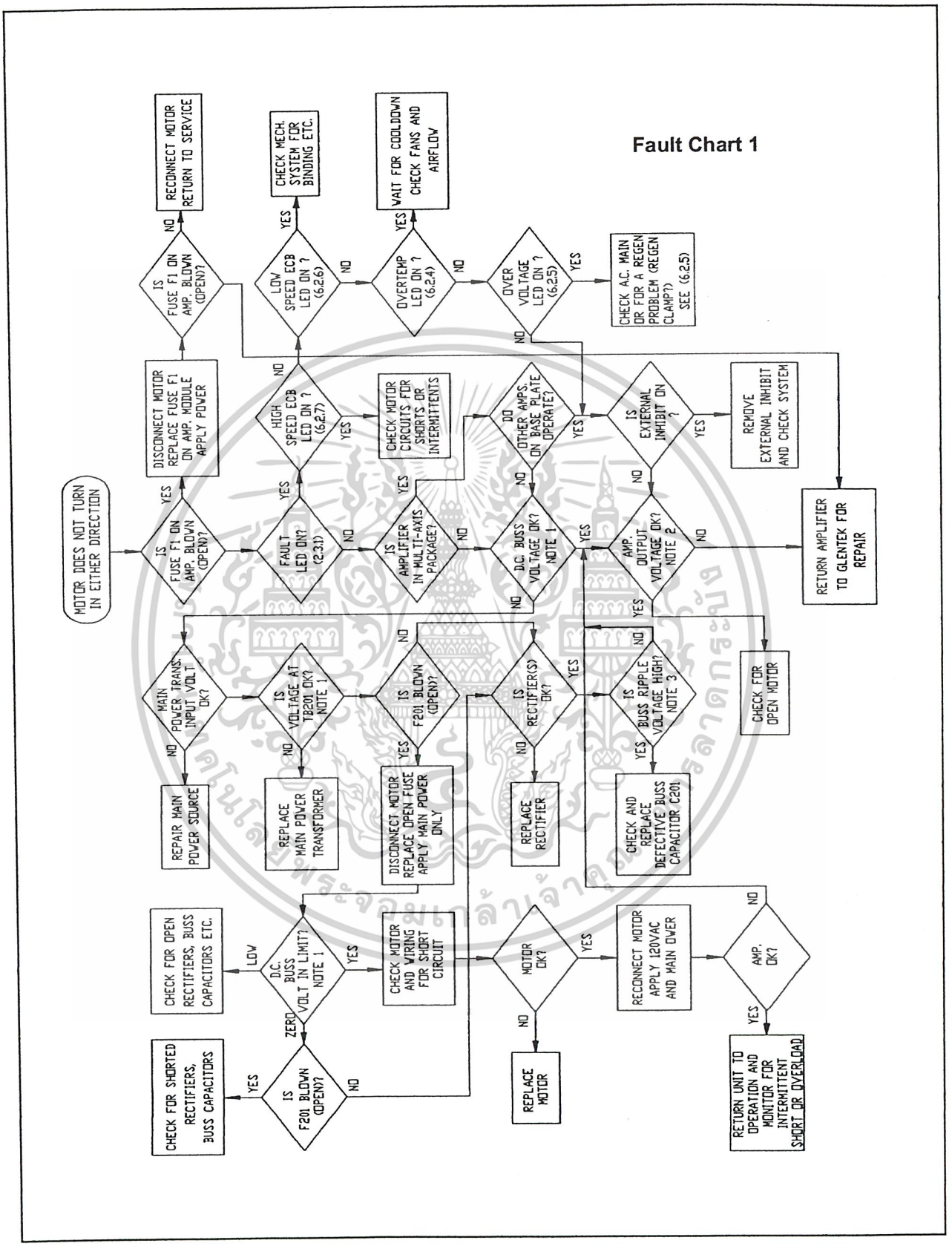
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Next i
Out &H100, opx
opx = 0
For i = 8 To 15
opx = opx + Abs(op(i) * 2 ^ (i - 8))
Next i
Out &H101, opx
End Sub
Sub delay()
For i = 1 To 15000
Next i
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

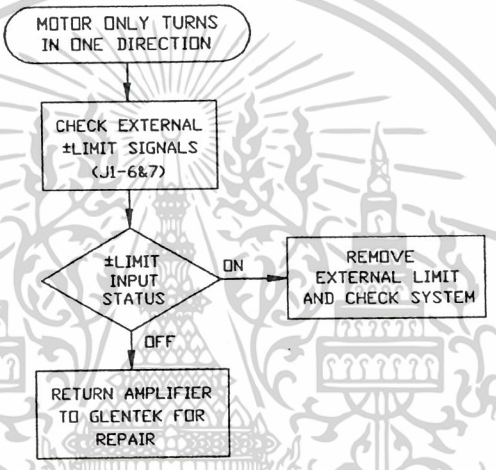
ภาคผนวก ข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



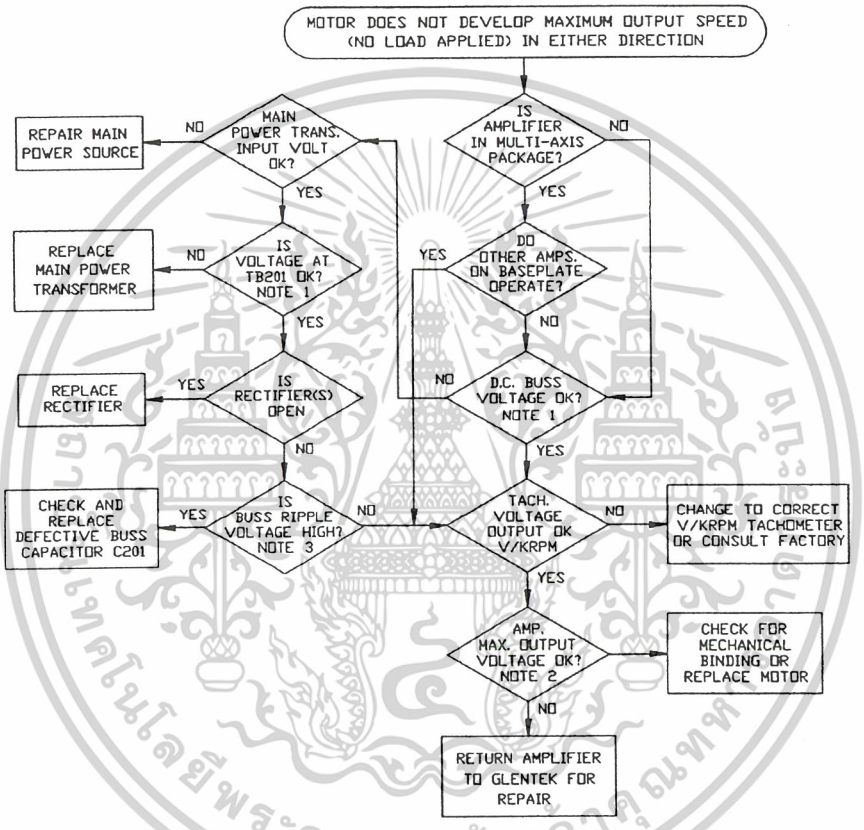
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Fault Chart 2



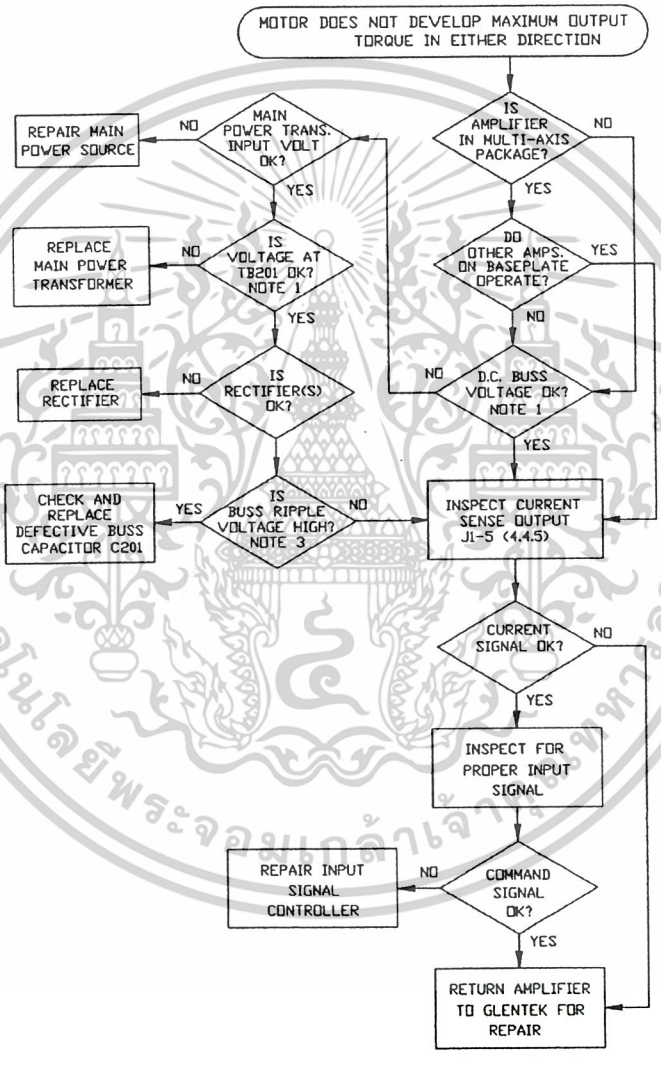
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Fault Chart 3



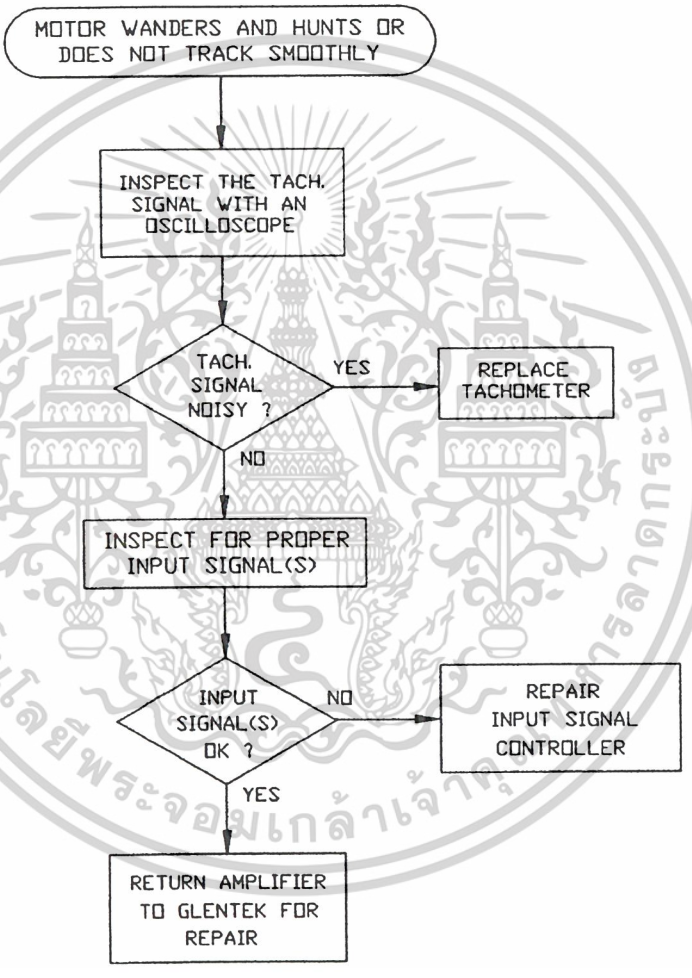
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Fault Chart 4

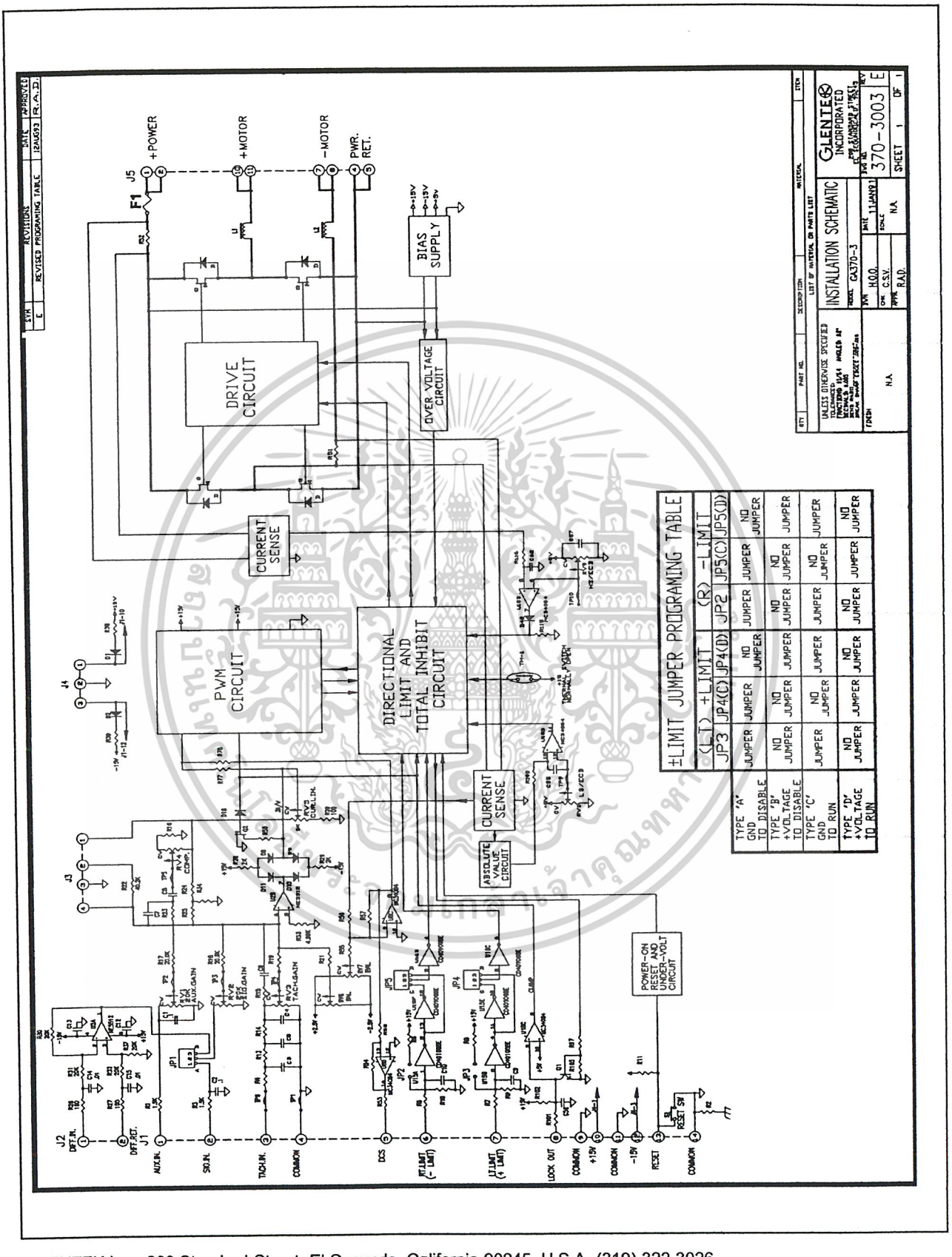


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

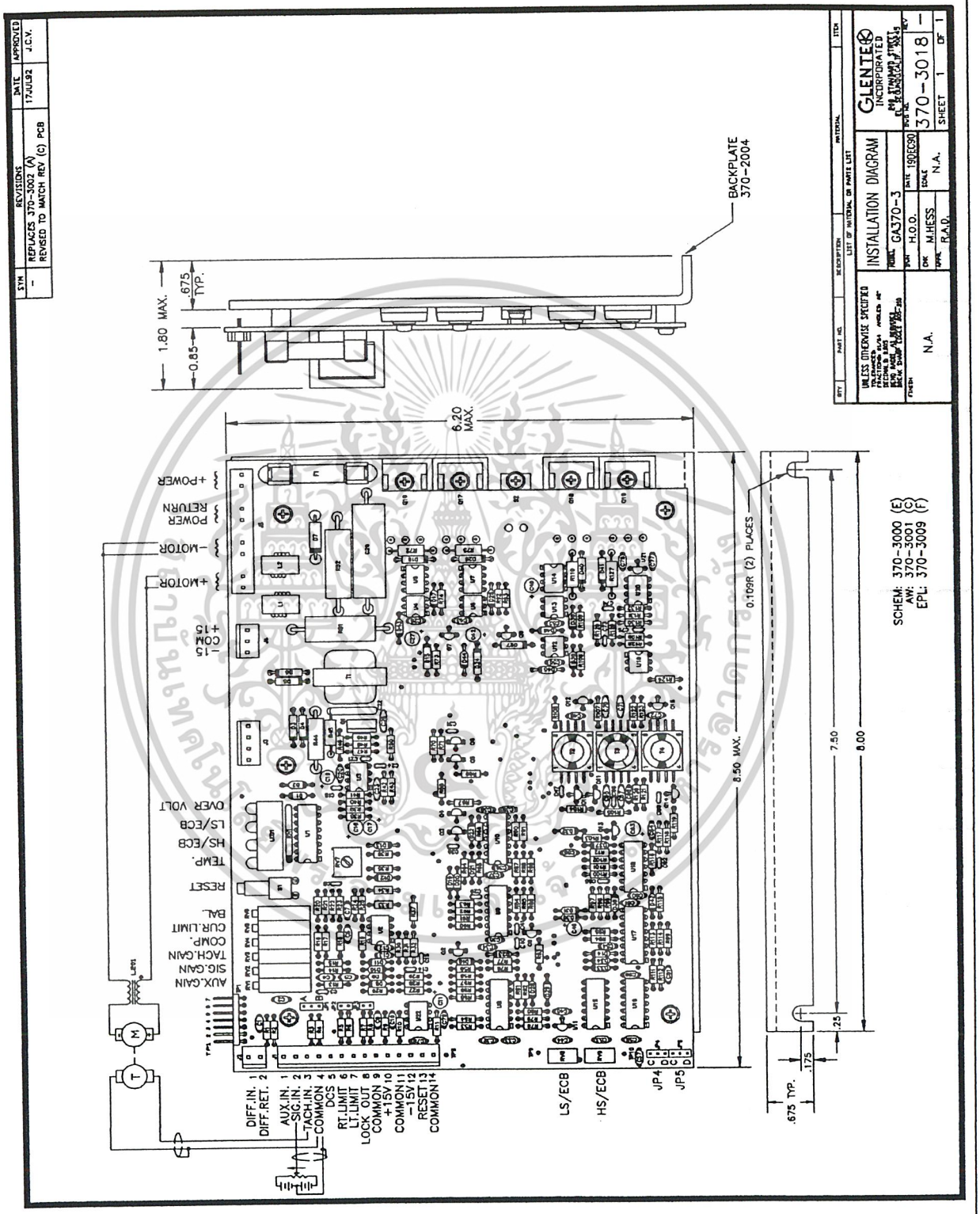
Fault Chart 5



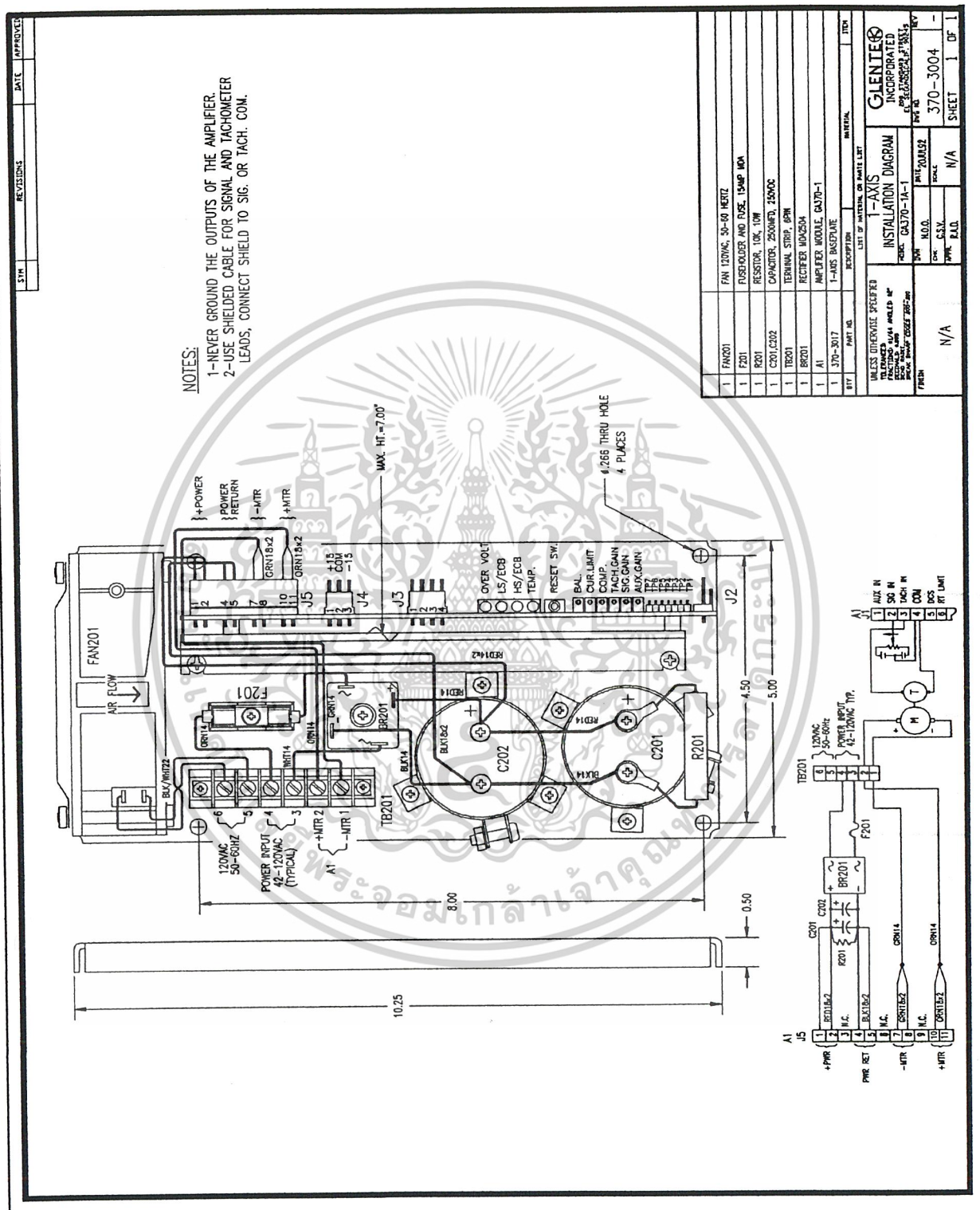
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



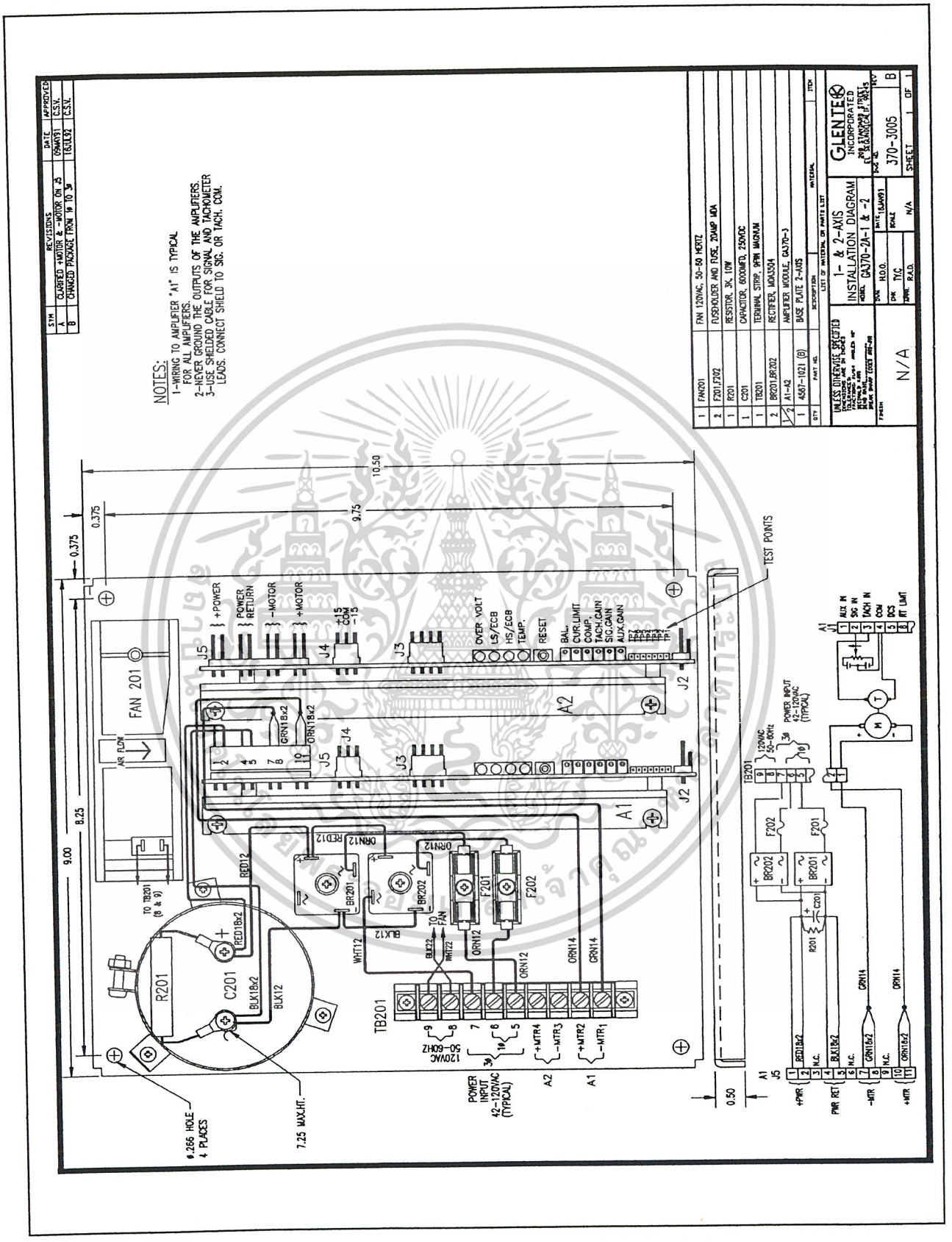
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

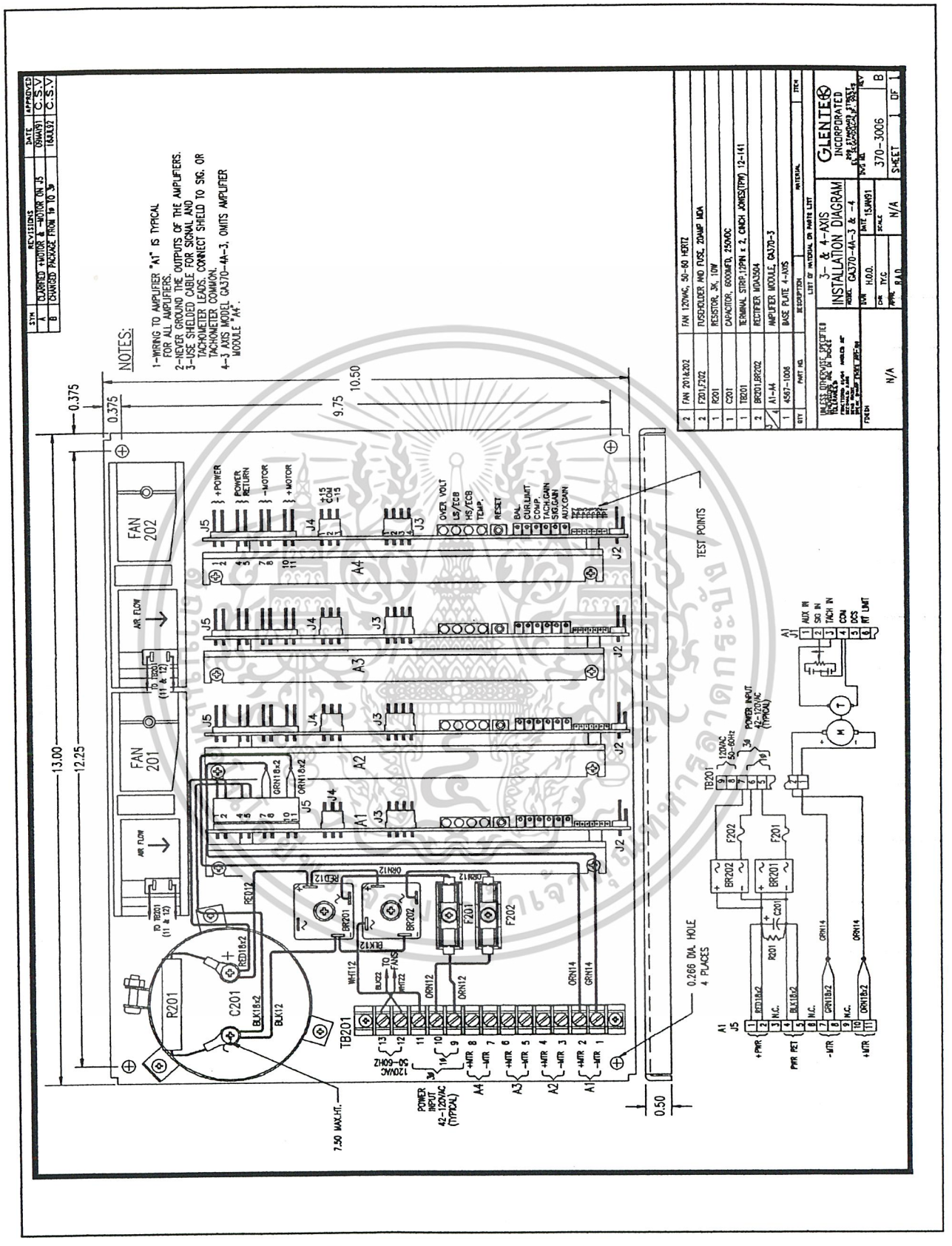


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

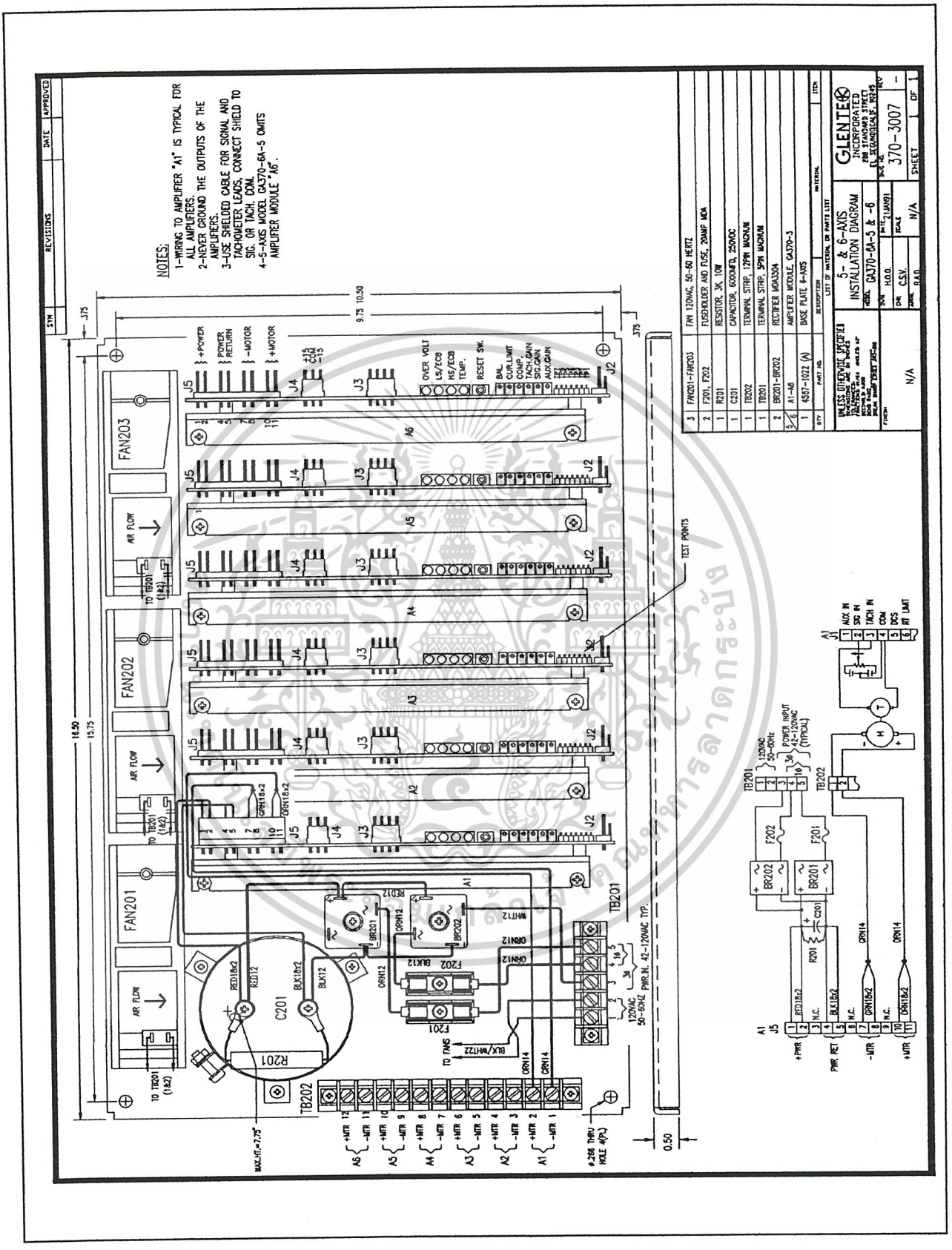


GLENTEK Inc.. 208 Standard Street. El Segundo. California 90245. U.S.A. (310) 322-3026

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



GLENTEK Inc.. 208 Standard Street. El Segundo. California 90245. U.S.A. (310) 322-3026

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Features

- Compatible with MCS-51™ Products
- 4K Bytes of In-System Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 24 MHz
- Three-level Program Memory Lock
- 128 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial Channel
- Low-power Idle and Power-down Modes

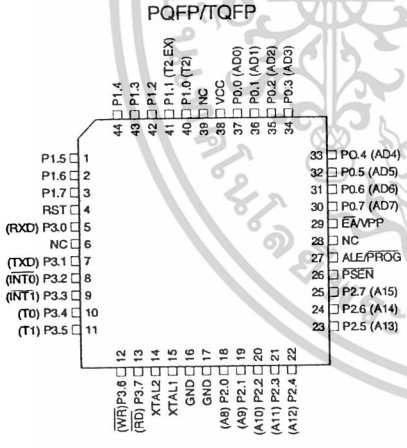
Description

The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard MCS-51 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

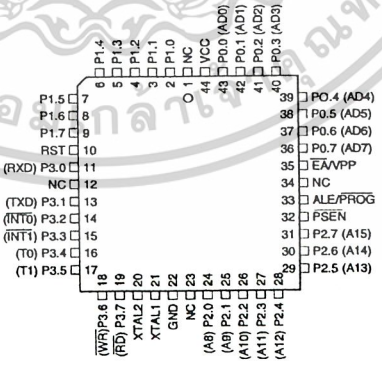
Pin Configurations

PDIP

PQFP/TQFP



PLCC



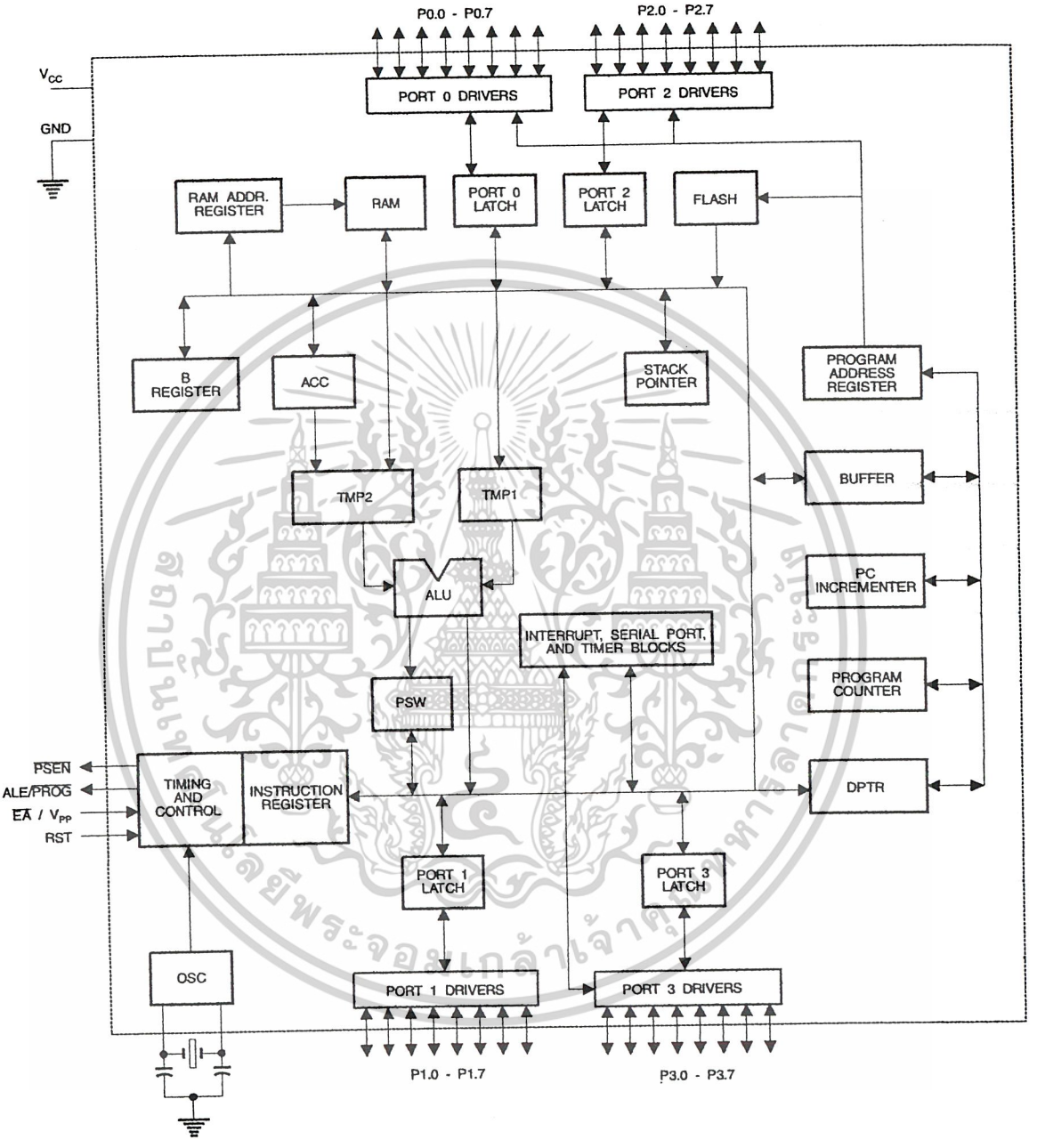
8-bit Microcontroller with 4K Bytes Flash

AT89C51

Rev. 0265G–02/00

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The AT89C51 provides the following standard features: 4K bytes of Flash, 128 bytes of RAM, 32 I/O lines, two 16-bit timer/counters, a five vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator and clock circuitry. In addition, the AT89C51 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. The Power-down Mode saves the RAM contents but freezes the oscillator disabling all other chip functions until the next hardware reset.

Pin Description

VCC

Supply voltage.

GND

Ground.

Port 0

Port 0 is an 8-bit open-drain bi-directional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.

Port 0 may also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode P0 has internal pullups.

Port 0 also receives the code bytes during Flash programming, and outputs the code bytes during program verification. External pullups are required during program verification.

Port 1

Port 1 is an 8-bit bi-directional I/O port with internal pullups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Port 1 also receives the low-order address bytes during Flash programming and verification.

Port 2

Port 2 is an 8-bit bi-directional I/O port with internal pullups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, it uses strong internal pullups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

Port 3

Port 3 is an 8-bit bi-directional I/O port with internal pullups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pullups.

Port 3 also serves the functions of various special features of the AT89C51 as listed below:

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{WR}$ (external data memory write strobe)
P3.7	$\overline{RD}$ (external data memory read strobe)

Port 3 also receives some control signals for Flash programming and verification.

RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

ALE/$\overline{PROG}$

Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input ($\overline{PROG}$) during Flash programming.

In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

pulse is skipped during each access to external Data Memory.

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

$\overline{\text{PSEN}}$

Program Store Enable is the read strobe to external program memory.

When the AT89C51 is executing code from external program memory, $\overline{\text{PSEN}}$ is activated twice each machine cycle, except that two $\overline{\text{PSEN}}$ activations are skipped during each access to external data memory.

$\overline{\text{EA}}$/VPP

External Access Enable. $\overline{\text{EA}}$ must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, $\overline{\text{EA}}$ will be internally latched on reset.

$\overline{\text{EA}}$ should be strapped to V_{CC} for internal program executions.

This pin also receives the 12-volt programming enable voltage (V_{PP}) during Flash programming, for parts that require 12-volt V_{PP}.

XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XTAL2

Output from the inverting oscillator amplifier.

Oscillator Characteristics

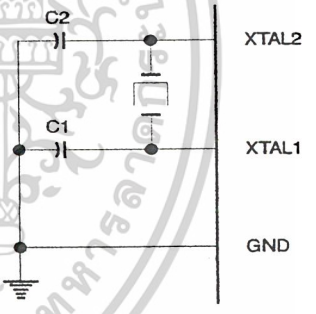
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 2. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Idle Mode

In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

It should be noted that when idle is terminated by a hard ware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

Figure 1. Oscillator Connections



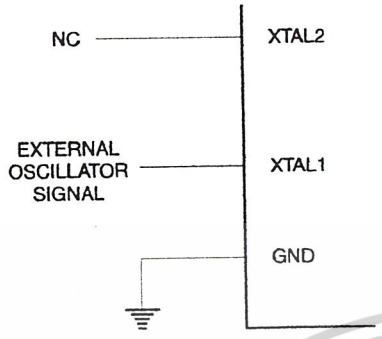
Note: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

Status of External Pins During Idle and Power-down Modes

Mode	Program Memory	ALE	$\overline{\text{PSEN}}$	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 2. External Clock Drive Configuration



Power-down Mode

In the power-down mode, the oscillator is stopped, and the instruction that invokes power-down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power-down mode is terminated. The only exit from power-down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

Program Memory Lock Bits

On the chip are three lock bits which can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the table below.

When lock bit 1 is programmed, the logic level at the $\overline{EA}$ pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random value, and holds that value until reset is activated. It is necessary that the latched value of $\overline{EA}$ be in agreement with the current logic level at that pin in order for the device to function properly.

Lock Bit Protection Modes

Program Lock Bits				
	LB1	LB2	LB3	Protection Type
1	U	U	U	No program lock features
2	P	U	U	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory, $\overline{EA}$ is sampled and latched on reset, and further programming of the Flash is disabled
3	P	P	U	Same as mode 2, also verify is disabled
4	P	P	P	Same as mode 3, also external execution is disabled

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Programming the Flash

The AT89C51 is normally shipped with the on-chip Flash memory array in the erased state (that is, contents = FFH) and ready to be programmed. The programming interface accepts either a high-voltage (12-volt) or a low-voltage (V_{CC}) program enable signal. The low-voltage programming mode provides a convenient way to program the AT89C51 inside the user's system, while the high-voltage programming mode is compatible with conventional third-party Flash or EPROM programmers.

The AT89C51 is shipped with either the high-voltage or low-voltage programming mode enabled. The respective top-side marking and device signature codes are listed in the following table.

	V_{PP} = 12V	V_{PP} = 5V
Top-Side Mark	AT89C51 xxxx yyww	AT89C51 xxxx-5 yyww
Signature	(030H) = 1EH (031H) = 51H (032H) =F FH	(030H) = 1EH (031H) = 51H (032H) = 05H

The AT89C51 code memory array is programmed byte-by-byte in either programming mode. *To program any non-blank byte in the on-chip Flash Memory, the entire memory must be erased using the Chip Erase Mode.*

Programming Algorithm: Before programming the AT89C51, the address, data and control signals should be set up according to the Flash programming mode table and Figure 3 and Figure 4. To program the AT89C51, take the following steps.

1. Input the desired memory location on the address lines.
2. Input the appropriate data byte on the data lines.
3. Activate the correct combination of control signals.
4. Raise $\overline{EA}/V_{PP}$ to 12V for the high-voltage programming mode.
5. Pulse ALE/$\overline{PROG}$ once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 1.5 ms. Repeat steps 1 through 5, changing the address and data for the entire array or until the end of the object file is reached.

$\overline{Data}$ Polling: The AT89C51 features $\overline{Data}$ Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written datum on PO.7. Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin. $\overline{Data}$ Polling may begin any time after a write cycle has been initiated.

Ready/$\overline{Busy}$: The progress of byte programming can also be monitored by the RDY/$\overline{BSY}$ output signal. P3.4 is pulled low after ALE goes high during programming to indicate BUSY. P3.4 is pulled high again when programming is done to indicate READY.

Program Verify: If lock bits LB1 and LB2 have not been programmed, the programmed code data can be read back via the address and data lines for verification. The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

Chip Erase: The entire Flash array is erased electrically by using the proper combination of control signals and by holding ALE/$\overline{PROG}$ low for 10 ms. The code array is written with all "1"s. The chip erase operation must be executed before the code memory can be re-programmed.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 030H, 031H, and 032H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows.

(030H) = 1EH indicates manufactured by Atmel
(031H) = 51H indicates 89C51
(032H) = FFH indicates 12V programming
(032H) = 05H indicates 5V programming

Programming Interface

Every code byte in the Flash array can be written and the entire array can be erased by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion.

All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your local programming vendor for the appropriate software revision.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flash Programming Modes

Mode		RST	PSEN	ALE/PROG	EA/V_{PP}	P2.6	P2.7	P3.6	P3.7
Write Code Data		H	L		H/12V	L	H	H	H
Read Code Data		H	L	H	H	L	L	H	H
Write Lock	Bit - 1	H	L		H/12V	H	H	H	H
	Bit - 2	H	L		H/12V	H	H	L	L
	Bit - 3	H	L		H/12V	H	L	H	L
Chip Erase		H	L	(1)	H/12V	H	L	L	L
Read Signature Byte		H	L	H	H	L	L	L	L

Note: 1. Chip Erase requires a 10 ms PROG pulse.

Figure 3. Programming the Flash

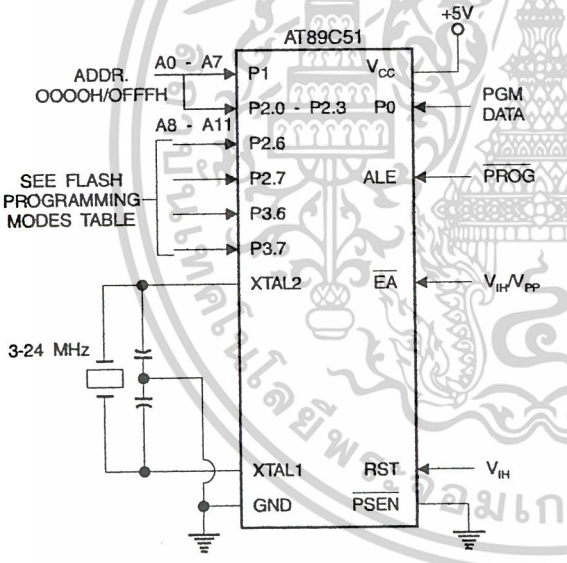
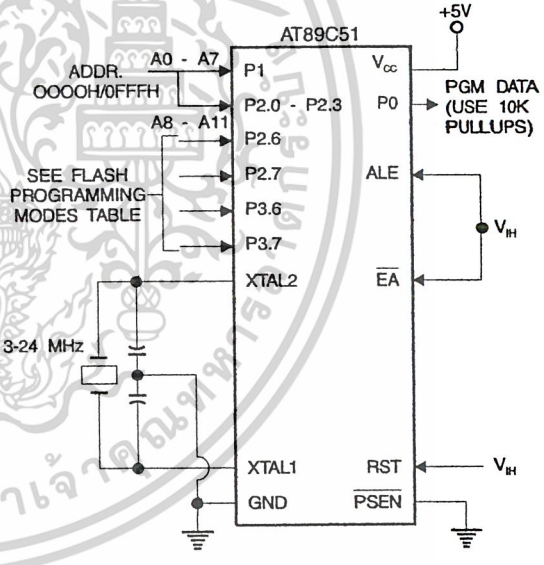
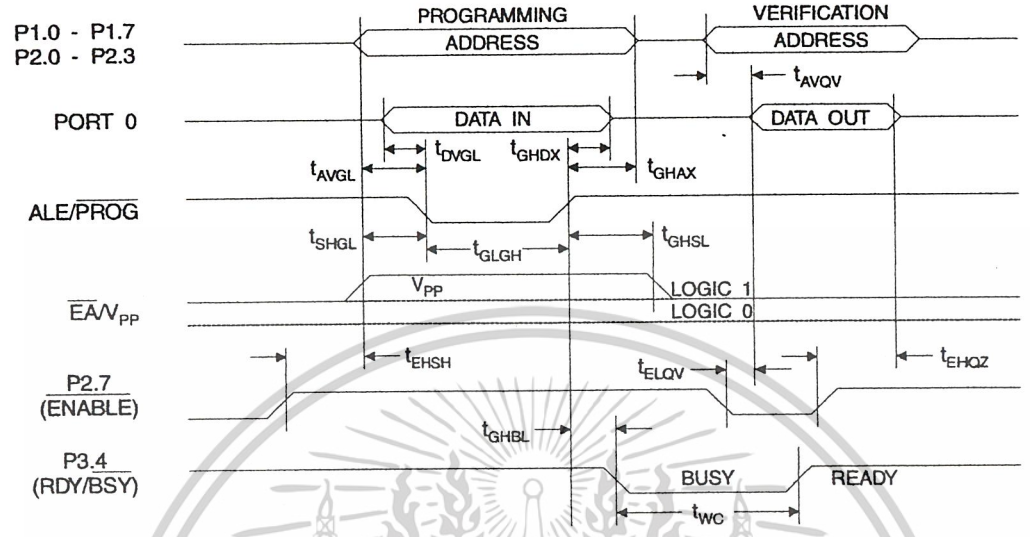


Figure 4. Verifying the Flash

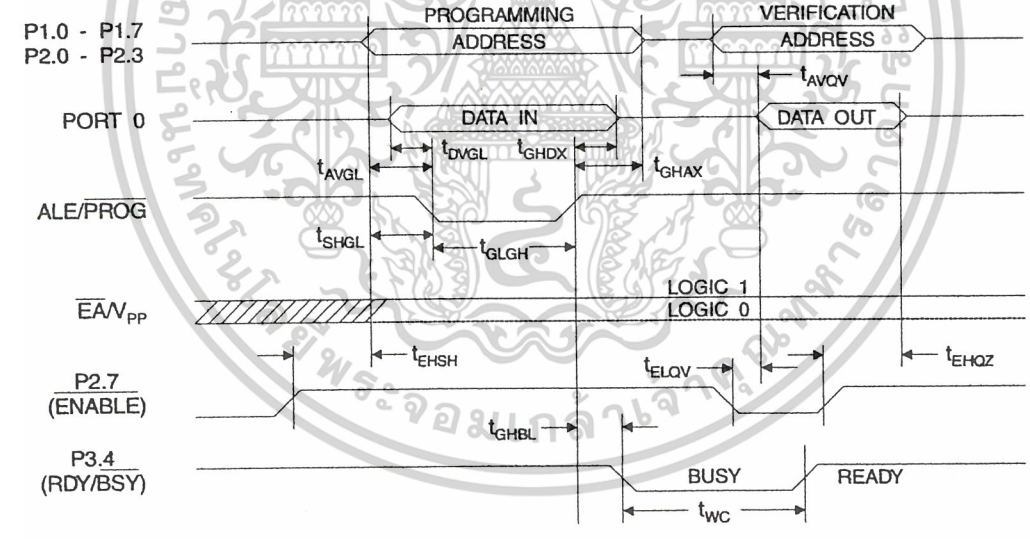


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flash Programming and Verification Waveforms - High-voltage Mode (V_{PP} = 12V)



Flash Programming and Verification Waveforms - Low-voltage Mode (V_{PP} = 5V)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on Any Pin with Respect to Ground	-1.0V to +7.0V
Maximum Operating Voltage	6.6V
DC Output Current	15.0 mA

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

T_A = -40°C to 85°C, V_{CC} = 5.0V ± 20% (unless otherwise noted)

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low-voltage	(Except $\overline{EA}$)	-0.5	0.2 V_{CC} - 0.1	V
V_{IL1}	Input Low-voltage ($\overline{EA}$)		-0.5	0.2 V_{CC} - 0.3	V
V_{IH}	Input High-voltage	(Except XTAL1, RST)	0.2 V_{CC} + 0.9	V_{CC} + 0.5	V
V_{IH1}	Input High-voltage	(XTAL1, RST)	0.7 V_{CC}	V_{CC} + 0.5	V
V_{OL}	Output Low-voltage(1) (Ports 1,2,3)	I_{OL} = 1.6 mA		0.45	V
V_{OL1}	Output Low-voltage(1) (Port 0, ALE, $\overline{PSEN}$)	I_{OL} = 3.2 mA		0.45	V
V_{OH}	Output High-voltage (Ports 1,2,3, ALE, $\overline{PSEN}$)	I_{OH} = -60 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -25 µA	0.75 V_{CC}		V
		I_{OH} = -10 µA	0.9 V_{CC}		V
V_{OH1}	Output High-voltage (Port 0 in External Bus Mode)	I_{OH} = -800 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -300 µA	0.75 V_{CC}		V
		I_{OH} = -80 µA	0.9 V_{CC}		V
I_{IL}	Logical 0 Input Current (Ports 1,2,3)	V_{IN} = 0.45V		-50	µA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1,2,3)	V_{IN} = 2V, VCC = 5V ± 10%		-650	µA
I_{LI}	Input Leakage Current (Port 0, $\overline{EA}$)	0.45 < V_{IN} < V_{CC}		±10	µA
RRST	Reset Pull-down Resistor		50	300	KΩ
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, T_A = 25°C		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz		20	mA
		Idle Mode, 12 MHz		5	mA
	Power-down Mode(2)	V_{CC} = 6V		100	µA
		V_{CC} = 3V		40	µA

Notes:
1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
   Maximum I_{OL} per port pin: 10 mA
   Maximum I_{OL} per 8-bit port: Port 0: 26 mA
   Ports 1, 2, 3: 15 mA
   Maximum total I_{OL} for all output pins: 71 mA
   If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
2. Minimum V_{CC} for Power-down is 2V.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flash Programming and Verification Characteristics

T_A = 0°C to 70°C, V_{CC} = 5.0 ± 10%

Symbol	Parameter	Min	Max	Units
V_{PP}(1)	Programming Enable Voltage	11.5	12.5	V
I_{PP}(1)	Programming Enable Current		1.0	mA
$1/t_{CLCL}$	Oscillator Frequency	3	24	MHz
t_{AVGL}	Address Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
t_{GHAX}	Address Hold After $\overline{PROG}$	$48t_{CLCL}$		
t_{DVGL}	Data Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
t_{GHDX}	Data Hold After $\overline{PROG}$	$48t_{CLCL}$		
t_{EHSH}	P2.7 ($\overline{ENABLE}$) High to V_{PP}	$48t_{CLCL}$		
t_{SHGL}	V_{PP} Setup to $\overline{PROG}$ Low	10		µs
t_{GHSL}(1)	V_{PP} Hold After $\overline{PROG}$	10		µs
t_{GLGH}	$\overline{PROG}$ Width	1	110	µs
t_{AVQV}	Address to Data Valid		$48t_{CLCL}$	
t_{ELQV}	$\overline{ENABLE}$ Low to Data Valid		$48t_{CLCL}$	
t_{EHQZ}	Data Float After $\overline{ENABLE}$	0	$48t_{CLCL}$	
t_{GHBL}	$\overline{PROG}$ High to $\overline{BUSY}$ Low		1.0	µs
t_{WC}	Byte Write Cycle Time		2.0	ms

Note: 1. Only used in 12-volt programming mode.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC Characteristics

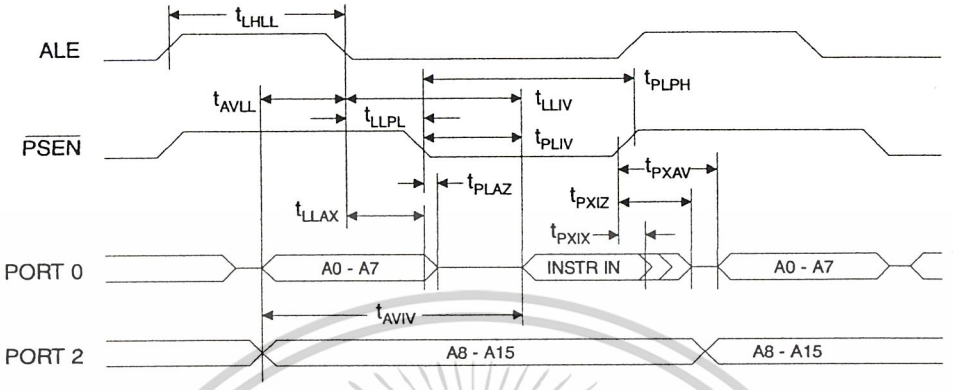
Under operating conditions, load capacitance for Port 0, ALE/$\overline{PROG}$, and $\overline{PSEN}$ = 100 pF; load capacitance for all other outputs = 80 pF.

External Program and Data Memory Characteristics

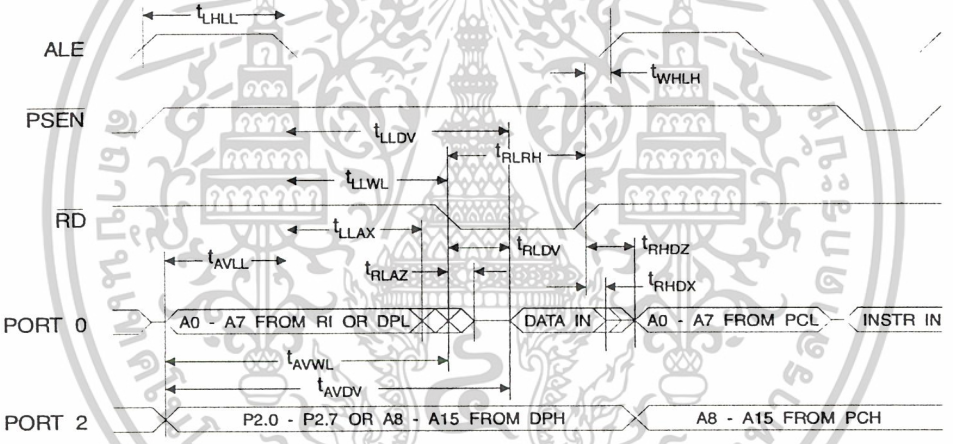
Symbol	Parameter	12 MHz Oscillator		16 to 24 MHz Oscillator		Units
		Min	Max	Min	Max	
$1/t_{CLCL}$	Oscillator Frequency			0	24	MHz
t_{LHLL}	ALE Pulse Width	127		$2t_{CLCL}$-40		ns
t_{AVLL}	Address Valid to ALE Low	43		t_{CLCL}-13		ns
t_{LLAX}	Address Hold After ALE Low	48		t_{CLCL}-20		ns
t_{LLIV}	ALE Low to Valid Instruction In		233		$4t_{CLCL}$-65	ns
t_{LLPL}	ALE Low to $\overline{PSEN}$ Low	43		t_{CLCL}-13		ns
t_{PLPH}	$\overline{PSEN}$ Pulse Width	205		$3t_{CLCL}$-20		ns
t_{PLIV}	$\overline{PSEN}$ Low to Valid Instruction In		145		$3t_{CLCL}$-45	ns
t_{PXIX}	Input Instruction Hold After $\overline{PSEN}$	0		0		ns
t_{PXIZ}	Input Instruction Float After $\overline{PSEN}$		59		t_{CLCL}-10	ns
t_{PXAV}	$\overline{PSEN}$ to Address Valid	75		t_{CLCL}-8		ns
t_{AVIV}	Address to Valid Instruction In		312		$5t_{CLCL}$-55	ns
t_{PLAZ}	$\overline{PSEN}$ Low to Address Float		10		10	ns
t_{RLRH}	$\overline{RD}$ Pulse Width	400		$6t_{CLCL}$-100		ns
t_{WLWH}	$\overline{WR}$ Pulse Width	400		$6t_{CLCL}$-100		ns
t_{RLDV}	$\overline{RD}$ Low to Valid Data In		252		$5t_{CLCL}$-90	ns
t_{RHDX}	Data Hold After $\overline{RD}$	0		0		ns
t_{RHDZ}	Data Float After $\overline{RD}$		97		$2t_{CLCL}$-28	ns
t_{LLDV}	ALE Low to Valid Data In		517		$8t_{CLCL}$-150	ns
t_{AVDV}	Address to Valid Data In		585		$9t_{CLCL}$-165	ns
t_{LLWL}	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	200	300	$3t_{CLCL}$-50	$3t_{CLCL}$+50	ns
t_{AVWL}	Address to $\overline{RD}$ or $\overline{WR}$ Low	203		$4t_{CLCL}$-75		ns
t_{QVWX}	Data Valid to $\overline{WR}$ Transition	23		t_{CLCL}-20		ns
t_{QVWH}	Data Valid to $\overline{WR}$ High	433		$7t_{CLCL}$-120		ns
t_{WHQX}	Data Hold After $\overline{WR}$	33		t_{CLCL}-20		ns
t_{RLAZ}	$\overline{RD}$ Low to Address Float		0		0	ns
t_{WHLH}	$\overline{RD}$ or $\overline{WR}$ High to ALE High	43	123	t_{CLCL}-20	t_{CLCL}+25	ns

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

External Program Memory Read Cycle

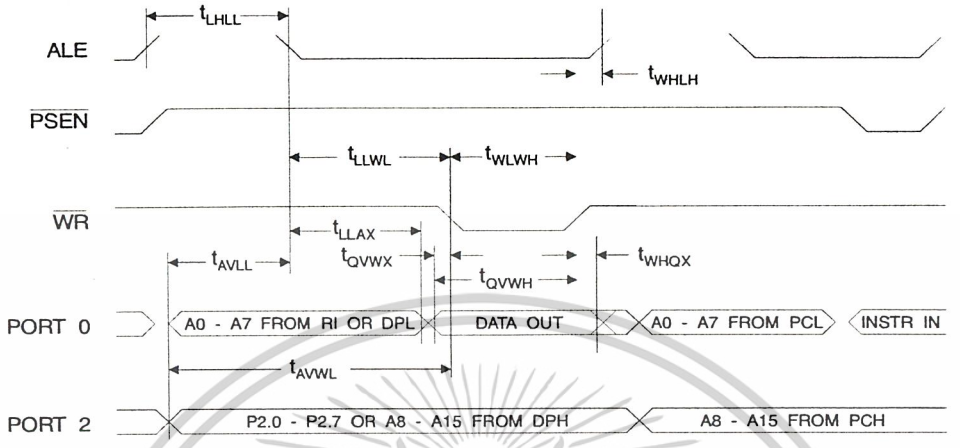


External Data Memory Read Cycle

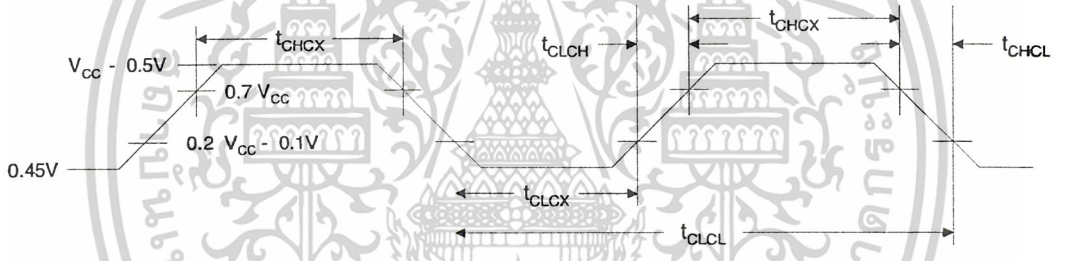


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

External Data Memory Write Cycle



External Clock Drive Waveforms



External Clock Drive

Symbol	Parameter	Min	Max	Units
$1/t_{CLCL}$	Oscillator Frequency	0	24	MHz
t_{CLCL}	Clock Period	41.6		ns
t_{CHCX}	High Time	15		ns
t_{CLCX}	Low Time	15		ns
t_{CLCH}	Rise Time		20	ns
t_{CHCL}	Fall Time		20	ns

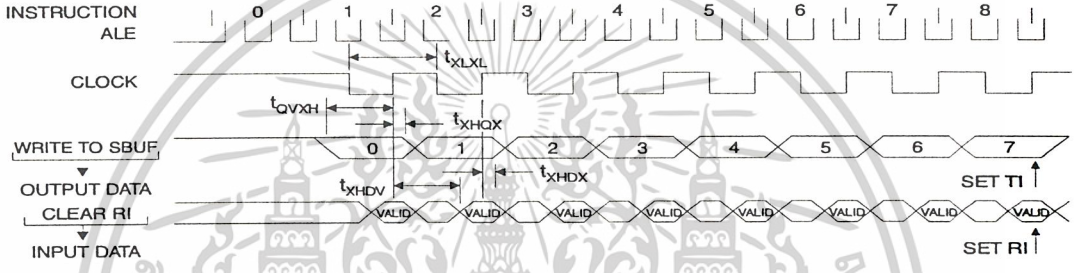
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Serial Port Timing: Shift Register Mode Test Conditions

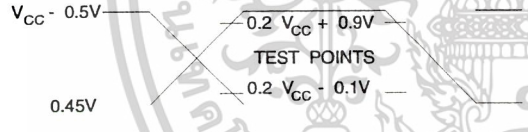
(V_{CC} = 5.0 V ± 20%; Load Capacitance = 80 pF)

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
t_{XLXL}	Serial Port Clock Cycle Time	1.0		$12t_{CLCL}$		µs
t_{QVXH}	Output Data Setup to Clock Rising Edge	700		$10t_{CLCL}$-133		ns
t_{XHQX}	Output Data Hold After Clock Rising Edge	50		$2t_{CLCL}$-117		ns
t_{XHDX}	Input Data Hold After Clock Rising Edge	0		0		ns
t_{XHDV}	Clock Rising Edge to Input Data Valid		700		$10t_{CLCL}$-133	ns

Shift Register Mode Timing Waveforms

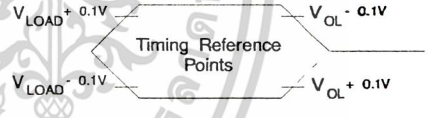


AC Testing Input/Output Waveforms[1]



Note: 1. AC Inputs during testing are driven at V_{CC} - 0.5V for a logic 1 and 0.45V for a logic 0. Timing measurements are made at V_{IH} min. for a logic 1 and V_{IL} max. for a logic 0.

Float Waveforms[1]



Note: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when 100 mV change from the loaded V_{OH}/V_{OL} level occurs.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ordering Information

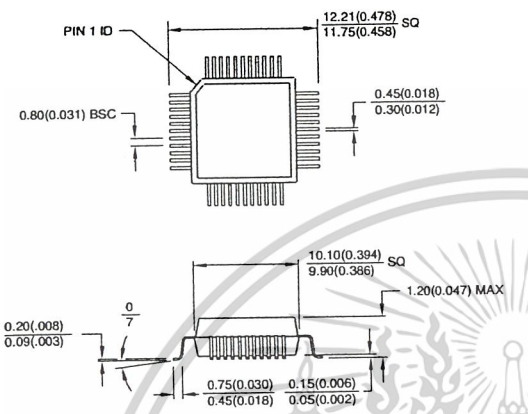
Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
12	5V ± 20%	AT89C51-12AC	44A	Commercial (0°C to 70°C)
		AT89C51-12JC	44J	
		AT89C51-12PC	40P6	
		AT89C51-12QC	44Q	
		AT89C51-12AI	44A	Industrial (-40°C to 85°C)
		AT89C51-12JI	44J	
		AT89C51-12PI	40P6	
		AT89C51-12QI	44Q	
16	5V ± 20%	AT89C51-16AC	44A	Commercial (0°C to 70°C)
		AT89C51-16JC	44J	
		AT89C51-16PC	40P6	
		AT89C51-16QC	44Q	
		AT89C51-16AI	44A	Industrial (-40°C to 85°C)
		AT89C51-16JI	44J	
		AT89C51-16PI	40P6	
		AT89C51-16QI	44Q	
20	5V ± 20%	AT89C51-20AC	44A	Commercial (0°C to 70°C)
		AT89C51-20JC	44J	
		AT89C51-20PC	40P6	
		AT89C51-20QC	44Q	
		AT89C51-20AI	44A	Industrial (-40°C to 85°C)
		AT89C51-20JI	44J	
		AT89C51-20PI	40P6	
		AT89C51-20QI	44Q	
24	5V ± 20%	AT89C51-24AC	44A	Commercial (0°C to 70°C)
		AT89C51-24JC	44J	
		AT89C51-24PC	40P6	
		AT89C51-24QC	44Q	
		AT89C51-24AI	44A	Industrial (-40°C to 85°C)
		AT89C51-24JI	44J	
		AT89C51-24PI	40P6	
		AT89C51-24QI	44Q	

Package Type	
44A	44-lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
44J	44-lead, Plastic J-leaded Chip Carrier (PLCC)
40P6	40-lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
44Q	44-lead, Plastic Gull Wing Quad Flatpack (PQFP)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

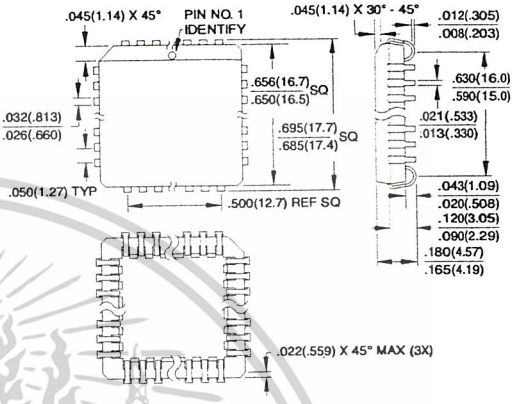
Packaging Information

44A, 44-lead, Thin (1.0 mm) Plastic Gull Wing Quad Flatpack (TQFP)
Dimensions in Millimeters and (Inches)*
JEDEC STANDARD MS-026 ACB

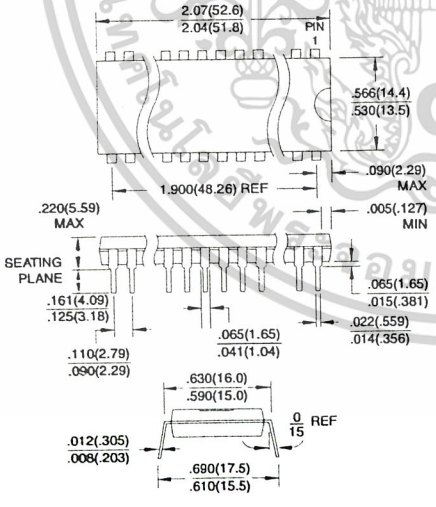


Controlling dimension: millimeters

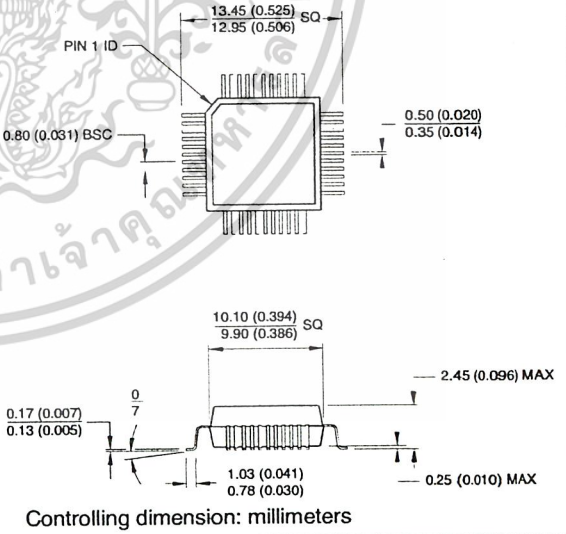
44J, 44-lead, Plastic J-leaded Chip Carrier (PLCC)
Dimensions in Inches and (Millimeters)
JEDEC STANDARD MS-018 AC



40P6, 40-lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
Dimensions in Inches and (Millimeters)



44Q, 44-lead, Plastic Quad Flat Package (PQFP)
Dimensions in Millimeters and (Inches)*
JEDEC STANDARD MS-022 AB



Controlling dimension: millimeters

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Atmel Headquarters

Corporate Headquarters
2325 Orchard Parkway
San Jose, CA 95131
TEL (408) 441-0311
FAX (408) 487-2600

Europe
Atmel U.K., Ltd.
Coliseum Business Centre
Riverside Way
Camberley, Surrey GU15 3YL
England
TEL (44) 1276-686-677
FAX (44) 1276-686-697

Asia
Atmel Asia, Ltd.
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

Japan
Atmel Japan K.K.
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
TEL (81) 3-3523-3551
FAX (81) 3-3523-7581

Atmel Operations

Atmel Colorado Springs
1150 E. Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL (719) 576-3300
FAX (719) 540-1759

Atmel Rousset
Zone Industrielle
13106 Rousset Cedex
France
TEL (33) 4-4253-6000
FAX (33) 4-4253-6001

Fax-on-Demand
North America:
1-(800) 292-8635
International:
1-(408) 441-0732

e-mail
literature@atmel.com

Web Site
http://www.atmel.com

BBS
1-(408) 436-4309

© Atmel Corporation 2000.
Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

Marks bearing ® and/or ™ are registered trademarks and trademarks of Atmel Corporation.

Terms and product names in this document may be trademarks of others.

Printed on recycled paper.

0265G–02/00/xM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้