ปีการศึกษา 2532

# การควบคุมการใช้ไฟฟ้าในอาคาร

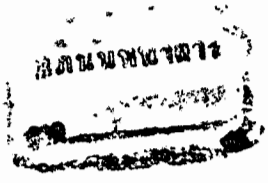
โดย

ศรัณย์ วีระวัฒนานันท์

สุกิจ คุณาธิกรกิจ

อาจารย์ที่ปรึกษา

สุเธียร เกียรติสุนทร



ปริญญานิพนธ์ปีการศึกษา 2531

ภาควิชา ระบบควบคุม

คณะวิศวะกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การควบคุมการใช้ไฟฟ้าในอาคาร

ผู้จัดทำ

1. ศรัณย์ วิระวัฒนานันท์ 28.1236
2. สุกิจ คุณาธิกรกิจ 28.1267

สุเธียร เกียรติสุนทร อาจารย์ที่ปรึกษา

( สุเธียร เกียรติสุนทร )

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

023217

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# การควบคุมการใช้ไฟฟ้าในอาคาร

ศรัณย์ วีระวัฒนานันท์

สุกิจ คุณาธิกรกิจ

อ. สุเธียร เกียรติสุนทร อาจารย์ที่ปรึกษา

ปีการศึกษา 2531

บทคัดย่อ

โครงงานนี้เป็นการศึกษาหาวิธีการในการลดพลังงานไฟฟ้าในส่วนที่ไม่จำเป็นในการใช้ออกไป ซึ่งจะทำให้เกิดความประหยัดในการใช้ไฟฟ้า โดยใช้วิธีการวัดและเก็บค่าข้อมูลที่วัดตามจุดต่าง ๆ ภายในช่วงเวลาที่ต้องการตรวจสอบ แล้วนำเอาค่าข้อมูลมาวิเคราะห์ว่า เกิดการผิดปกติในการใช้ไฟฟ้าบ้างหรือไม่ หรือเกิดการใช้กระแสไฟฟ้ามากเกินความต้องการหรือไม่ เพื่อที่จะได้นำเอาค่าข้อมูลที่วิเคราะห์ได้ไปเซ็ตอุปกรณ์ เช่น มอเตอร์ ปั๊มน้ำ ลิฟท์ เครื่องปรับอากาศ เป็นต้น ให้เหมาะสมกับความต้องการ ไม่กินกระแสมากเกินไปขอบเขตของโครงงานนี้ จะเป็นการวัดค่าข้อมูลกระแสไฟฟ้าและเก็บค่าข้อมูลนั้นลงในเครื่องไมโครคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ELECTRICAL DEMAND CONTROL IN BUILDING

Sarun Weerawattananan

Sukit Kunatikornkit

Mr. Sutiern Kietsoontorn Advisor

1988

Abstract

The objective of this project is to study for finding the way in order to reduce the unnecessary electrical power. Measuring and saving he current data in the period of time at any point of the building is one of the method. And then get the data to analize how the using of electrical occur faulty or not. In order to get the analized data to operate motor, pump, left, air condition for suitable demand. The scope of this projectos the measuring and saving the current data into microcomputer.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ


| | | หน้า |
|---|---|---|
| บทที่ 1 | บทนำ | 1 |
| บทที่ 2 | หลักการวัดและเก็บค่าข้อมูลทางไฟฟ้า | 2 |
| | - ภาคตรวจจับสัญญาณ | 3 |
| | - ภาคส่งสัญญาณ | 5 |
| | - ภาคแสดงผลและบันทึกข้อมูล | 9 |
| | : การออกแบบการเขียนโปรแกรม | 15 |
| บทที่ 3 | การออกแบบโปรแกรมควบคุม | 24 |
| บทที่ 4 | การทดลองและผลการทดลอง | 49 |
| บทที่ 5 | บทสรุปและวิจารณ์ | 51 |


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

# บทนำ

ปัจจุบันการใช้ไฟฟ้าได้เข้ามาเป็นส่วนหนึ่งของปัจจัยในการดำรงชีวิตของมนุษย์ การหาวิธีที่จะประหยัดการใช้ไฟฟ้าจึงเป็นสิ่งจำเป็นที่จะต้องศึกษา เพราะถ้าไม่หาวิธีที่จะประหยัดการใช้ไฟฟ้าแล้ว สภาวะความต้องการที่มากขึ้นก็จะก่อให้เกิดการขาดแคลนและเป็นผลเสียต่อทรัพยากรที่สูญเสียไปโดยใช่เหตุ

วิธีการประหยัดไฟฟ้าที่ดีที่สุดคือ การพยายามที่จะลดส่วนของการใช้ไฟฟ้าที่ไม่จำเป็นออกไป ซึ่งจะไม่มีผลต่อการใช้ไฟฟ้าโดยทั่วไปและไม่ก่อให้เกิดประโยชน์ต่อผู้ใช้แต่อย่างใด

โครงงานนี้จะศึกษาถึงวิธีการลดส่วนของการใช้ไฟฟ้าที่ไม่จำเป็นออกไป โดยการวัดค่าข้อมูลของกระแสไฟฟ้าที่ใช้ตามจุดต่างๆของอาคารในช่วงเวลาที่กำหนด มาแสดงเป็นมอนิเตอร์บนจอเครื่องคอมพิวเตอร์ พร้อมทั้งทำการบันทึกค่าข้อมูลเก็บไว้ เพื่อที่จะหาปริมาณการใช้ไฟฟ้าจริง ความผิดปกติของการใช้ไฟฟ้า ฯลฯ ซึ่งในการประหยัดการใช้ไฟฟ้าจำเป็นที่จะต้องรู้ค่าข้อมูลเหล่านี้เป็นพื้นฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 2

# ทฤษฎีและหลักการ

## หลักการวัดและเก็บค่าข้อมูลทางไฟฟ้า

การวัดค่าข้อมูลทางไฟฟ้าจะพิจารณาถึงตำแหน่งการใช้ไฟฟ้าตามจุดต่างๆที่สำคัญ และมีผลต่อการใช้จ่ายพลังงานไฟฟ้าส่วนมาก ซึ่งข้อมูลที่ได้ก็จะบอกถึงปริมาณการใช้ไฟฟ้าตามจุดนั้นๆ ณ เวลาจริง การเกิดความผิดปกติบางอย่าง และปริมาณการใช้ไฟฟ้ารวมทั้งหมดของอาคาร

ชนิดของค่าพารามิเตอร์(parameter)ทางไฟฟ้านั้นอาจเป็นรูปโวลท์เตจหรือกระแส แต่การใช้ไฟฟ้าส่วนใหญ่จะใช้แหล่งจ่ายที่มีโวลท์เตจคงที่ และกระแสเปลี่ยนแปลงไปตามโหลดที่ใช้ การวัดจึงทำการวัดค่ากระแสเพียงอย่างเดียวก็เพียงพอ

นอกจากค่าข้อมูลของกระแสที่วัดจะทำให้ทราบถึงปริมาณการใช้โหลดแล้ว ยังอาจ - แสดงให้เห็นถึงสถานะการปิด/เปิดของการใช้โหลดต่างๆเหล่านั้นอีกด้วย

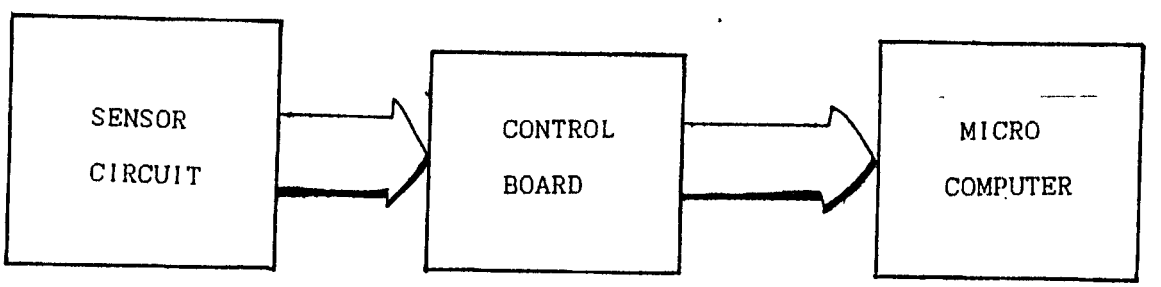
## ส่วนประกอบของการวัดและเก็บค่าข้อมูล

แบ่งได้เป็น 3 ส่วนใหญ่ๆ คือ

1: ภาคตรวจจับสัญญาณ ( เซ็นเซอร์(sensor) )

2: ภาคส่งสัญญาณ ( บอร์ดควบคุม(control board) )

3: ภาคแสดงผลและบันทึกข้อมูล ( ไมโครคอมพิวเตอร์(micro computer) )




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นว่าส่วนประกอบที่ 1 และ 2 จะมีลักษณะเป็นฮาร์ดแวร์(hardware) ส่วนส่วนประกอบที่ 3 นั้นจะมีลักษณะเป็นซอฟท์แวร์(software) คือ เป็นการเขียนโปรแกรมที่จะเข้ามาควบคุมการรับข้อมูลและทำการบันทึกลงบนแผ่นจานแม่เหล็กหรือแสดงผลออกทางจอภาพ

รายละเอียดของส่วนประกอบ

1) ภาคตรวจจับสัญญาณ

ภาคนี้จะมีหน้าที่ในการตรวจจับสัญญาณกระแสตามจุดต่างๆที่ต้องการจะวัด และนำค่าที่ได้ส่งผ่านไปยังบอร์ดควบคุม โดยได้ออกแบบโครงสร้างของภาคนี้ไว้ดังนี้

1. หม้อแปลงกระแสไฟฟ้า(current transformer)

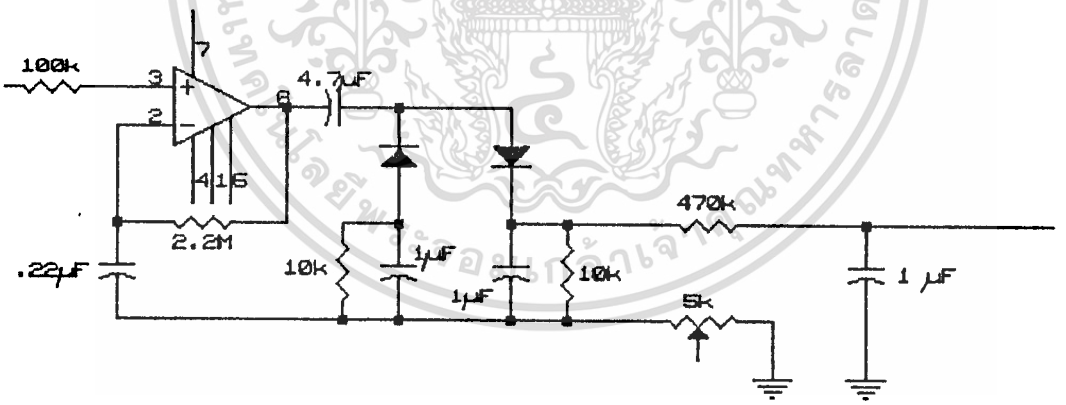
เป็นหม้อแปลงชนิดหนึ่งที่ใช้สำหรับแปลงค่าของระดับกระแสไฟฟ้า ในโครงงานนี้อุปกรณ์หม้อแปลงกระแสไฟฟ้าที่ใช้จะใช้แคลมป์มิเตอร์(clamp meter) ซึ่งเป็นหม้อแปลงกระแสไฟฟ้าที่ แปลงค่าของกระแสไฟฟ้าให้อยู่ในรูปของโวลท์เตจกระแสสลับ ซึ่งอัตราของการเปลี่ยนแปลงระดับกระแสไปเป็นโวลท์เตจกระแสสลับนั้นจะมีลักษณะเป็นเชิงเส้นกัน จากคุณลักษณะของหม้อแปลงกระแสไฟฟ้าที่ใช้ สามารถเปลี่ยนแปลงค่าระดับกระแสได้ในช่วง 0-1200 Amp ให้อยู่ในรูปของโวลท์เตจกระแสสลับคือค่าโวลท์พีก(volt peak) ได้ในช่วง 0-12 โวลท์พีก ซึ่งจากการนำเอาอุปกรณ์มาทำการทดลองพบว่าทำงานได้เป็นเชิงเส้นจริงดังตาราง

| โหลด(แอมป์) | โวลท์ (พีก) |
|---|---|
| 0 | 0 |
| 10 | 0.1 |
| 50 | 0.5 |
| 100 | 1 |
| 150 | 1.5 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าที่อ่านได้จากการวัดจะเป็นค่าของโวลท์เตจซึ่งมีค่าของกระแส=100*ค่าของโวลท์เตจ และเนื่องจากสัญญาณที่ได้จากหม้อแปลงกระแสไฟฟ้านี้จะเป็นสัญญาณแอนวลอก(analog) คือ- เป็นสัญญาณที่มีค่าไม่คงที่เป็นรูปคลื่นซายน์เวฟ(sine wave) ซึ่งยังไม่สามารถนำเอาสัญญาณไปเข้ายังบอร์ดควบคุมได้ เพราะว่าบอร์ดควบคุมนั้นจะรับสัญญาณที่เป็นค่าคงที่ได้เท่านั้น จึงต้องทำการแปลงสัญญาณโวลท์เตจที่อยู่ในรูปซายน์เวฟ ให้อยู่ในรูปของสัญญาณที่เป็นกระแสตรง ซึ่งจะกล่าวในส่วนต่อไป

2. วงจรแปลงค่าโวลท์พีกให้เป็นค่าโวลท์เตจกำลังสองเฉลี่ย(voltage root mean square)



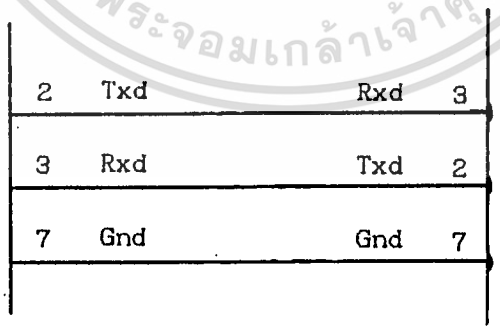

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ภาคส่งสัญญาณ

บอร์ดควบคุมที่นำมาใช้นั้น เป็นบอร์ดต้นแบบที่มีอยู่แล้ว มีหน้าที่เป็นตัวรับข้อมูลจากภาคตรวจจับสัญญาณ ซึ่งอาจเป็นข้อมูลแบบอนาล็อกหรือดิจิตอล(digital) ก็ได้ ถ้าเป็นข้อมูลเข้าแบบอนาล็อก ควรปรับภาคตรวจจับสัญญาณวัดค่าพารามิเตอร์แปลงข้อมูลเอาท์พุทให้อยู่ในช่วง 1-6 โวลท์ หรือข้อมูลแบบดิจิตอลก็สามารถต่อเข้าโดยตรง และทำการส่งข้อมูลที่ได้ ไปยังไมโครคอมพิวเตอร์ และถ้าไมโครคอมพิวเตอร์ต้องการที่จะควบคุมสถานะของอุปกรณ์ที่ทำหน้าที่จ่ายโหลด ก็จะส่งข้อมูลที่เป็นสัญญาณควบคุมมาให้ บอร์ดควบคุมนี้ก็จะรับสัญญาณเข้ามาและส่งต่อไปยังจุดควบคุมต่างๆเหล่านั้นอีกที แต่การใช้งานของบอร์ดในโครงงานนี้ เป็นเพียงส่วนรับข้อมูลเข้ามาเพื่อส่งค่าข้อมูลไปยังคอมพิวเตอร์เพียงอย่างเดียว

บอร์ดควบคุม

ลักษณะของบอร์ดควบคุมโดยทั่วๆไป จะเหมือนกับซิงเกิลบอร์ด(single board) ที่มีซีพียู 8085 ควบคุมการทำงานด้วยตัวเอง โดยมีส่วนของภาคอนาลอก-ดิจิตอล คอนเวอร์ทเตอร์(analog to digitaal converter) รวมอยู่ด้วย เพียงแต่ไม่มีคีย์บอร์ดในการป้อนคำสั่ง , ข้อมูล ที่ตัวบอร์ดโดยตรง การที่จะเขียนโปรแกรมการทำงานให้บอร์ดควบคุมทำได้โดยผ่านการป้อนข้อมูลทางไมโครคอมพิวเตอร์เชื่อมผ่านเข้าไปสู่บอร์ดควบคุม ดังรูปที่ 2.1




รูปที่ 2.1

รายละเอียดทั่วไป

- หน่วยประมวลผลกลาง (CENTRAL PROCESSING UNIT) : 8085 ที่การทำงานด้วยสัญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ญาณนาฬิกา 3 MHz

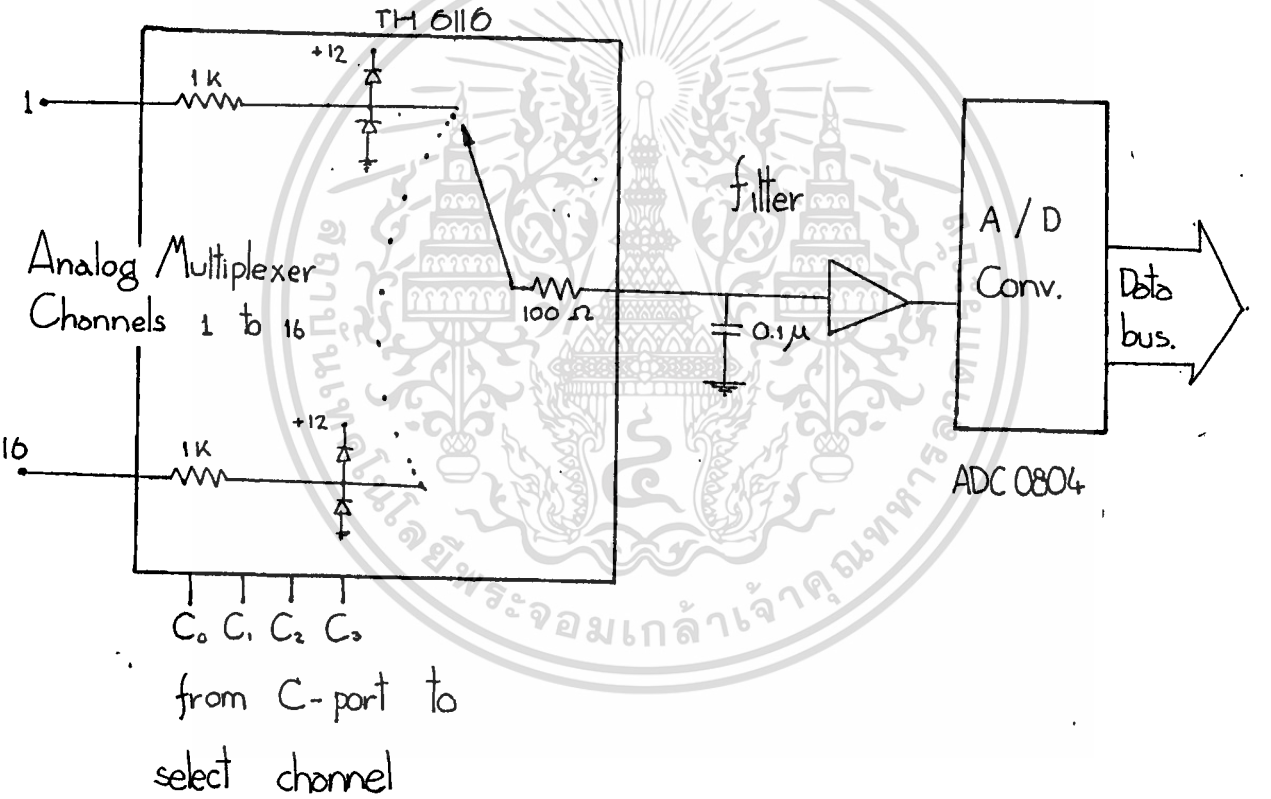
- อินพุท/เอาท์พุท : 8 อินพุทดิจิตอลชนิดออปติคอล ( ทีทีแอล ), 8 อินพุทดิจิตอลบัฟเฟอร์ (ชนิดขับกระแสสูง) , 16 อนาลอกอินพุท (1-6 โวลท์) , คอนเนคเตอร์อินพุท/เอาท์พุท 40 ขา
- หน่วยความจำ : จัดเนื้อที่ 8 K - 40 K สำหรับอีพรอม (Eprom) และ 8.25 K - 24.25 K สำหรับสแตติกแรม (Static Ram) รวมเนื้อที่หน่วยความจำทั้งหมด 40.25 ไบท์
- แหล่งจ่ายกำลังไฟ : 7-12 โวลท์ดีซี
- ไทม์เมอร์ (Timer) : วอร์ชด็อก ไทม์เมอร์ (Watch dog Timer) , โปรแกรมเอเบิ้ลไทม์เมอร์ (Programable Timer)
- การติดต่อสื่อสาร : มีพอร์ท (Port) RS 232 มาตรฐาน และสามารถแทนการทำงานด้วยพอร์ท RS 422
- ดิปสวิทช์ (Dip Switch) : 7 สวิทช์ สำหรับการเลือกและกำหนดโปรแกรม
- ความสามารถในการขยาย : มีคอนเนคเตอร์ 20 ขา สำหรับใช้งานร่วมกับบอร์ดอื่นๆ
- ซอฟท์แวร์ : มีฟอร์ทโอเปอเรทติ้งซิสเต็ม (Forth Operating System) เป็นมอนิเตอร์โปรแกรม

อนาลอกอินพุท :

ค่าอนาลอกจะถูกแปลงเป็นค่าดิจิตอล 8 บิท จากการเลือกแชนแนล(Channel) ของตัวมัลติเพล็กซ์(Multiplexer) จะใช้เวลา 104 ไมโครวินาที เวลาการตั้งค่าโวลท์เตจที่เข้ามาต้องใช้อย่างน้อย 30 ไมโครวินาที ค่าอินพุทอนาลอกที่น้อยกว่าหรือเท่ากับ 1 โวลท์ดีซีจะถูกแปลงค่าให้เป็นศูนย์ และจาก 1 โวลท์ขึ้นไปถึง 6 โวลท์จะถูกแปลงโดยมีอัตราส่วนเพิ่มขึ้นทุกๆ 20 มิลลิโวลท์ ต่อการเพิ่มหน่วยในฐานสิบหก 1 หน่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ละอนาลอกอินพุทถูกป้อนผ่านมัลติเพล็กซ์เซอร์ และวงจรแซมเปิ้ลแอนด์โฮล ( Sample and Hole Circuit ) แล้วผ่านเข้าสู่ส่วนแปลงอนาลอกเป็นดิจิตอล ( โดยพอร์ท ซี ขา C0-C3 เป็นตัวเลือกแชนแนลที่ต้องการ และเอาท์พุทของตำแหน่งแอดเดรส 60H จะเป็นตัวเปิดปิดอนาล็อกสวิทช์ทางวงจรแซมเปิ้ล ) บิท 8 ของไอซีคอนเวอร์ทเตอร์จะแสดงค่าศูนย์เมื่อการแปลงข้อมูลเสร็จสิ้น และสายสัญญาณนี้จะถูกอ่านโดยซีพียูสร้างสัญญาณอ่านข้อมูลจากไอซีตัวนี้เข้าไปจัดการทำงานต่อไป




รูปที่ 2.2

การโปรแกรมบอร์ดควบคุม

ถ้าผู้ใช้มี อี-ฟอร์ท โอเปอร์เรทติ้ง ซิสเต็ม ( E-FORTH Operating System ) ก็สามารถใช้ซอฟท์แวร์ภาษาฟอร์ท โปรแกรมลงบนแรมที่มีอยู่บนบอร์ด เพื่อจัดการการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของบอร์ดได้ในการติดต่อสื่อสารอุปกรณ์ต่างๆ แต่ถ้าผู้ใช้ไม่มีระบบนี้อยู่ ก็จำเป็นที่จะต้องเขียนโปรแกรมการควบคุมด้วยภาษาแอสแซมบลี 8085 สำหรับการทำงานต่างๆของบอร์ด เช่น การติดต่อเซ็ทการทำงานของ 8251 , การอ่านและเขียนข้อมูลระหว่างซีพียู และ 8251 , รูทีนของไดรเวอร์(driver) COMO , การควบคุมช่วงเวลาการแปลงข้อมูลของ 0808 พร้อมทั้งอ่านค่า , การสแกนแชนแนลอินพุทของ 6116 ฯลฯ

ในบอร์ดควบคุมที่ใช้นี่มี อี-ฟอร์ท โอเปอร์เรทติ้ง ซิสเต็ม อยู่ ดังนั้นจึงสามารถออกแบบโปรแกรมการทำงานของบอร์ดให้อยู่ในรูปภาษาฟอร์ทได้

โปรแกรมการทำงานของบอร์ดควบคุม

การทำงานของบอร์ดควบคุมนี้ ถูกออกแบบไว้คือ เริ่มทำการอ่านค่าอนาลอกอินพุทในแต่ละแชนแนลจนครบ 16 แชนแนล แล้วทำการส่งติดต่อกันเป็นกลุ่ม และส่งออกไปตลอดเวลา โดยจะต้องมีอักขระเพื่อที่จะบ่งบอกถึงตำแหน่งของข้อมูลในแต่ละแชนแนล ซึ่งได้กำหนดไว้คือ ให้ทำการส่ง " @ " ออกไปเป็นค่าแรก แล้วตามด้วยข้อมูลทั้ง 16 แชนแนล เรียงตั้งแต่แชนแนล 1-16 ตามลำดับ แล้วส่งอักขระปิดท้ายด้วยค่า " " เพื่อทำการบอกจุดสิ้นสุดการส่งข้อมูลในแต่ละชุด มีลักษณะดังรูปที่ 2.3

รูปที่ 2.3

จากการกำหนดรูปแบบขั้นต้น สามารถเขียนโปรแกรมควบคุมลงบนบอร์ดได้ดังนี้

```
: ASCII DUP 10 < IF 48 + ELSE 55 + THEN ;
: F11 48 DUP ROT XOR SWAP EMIT ;
: F1 64 DUP EMIT F11 F11 ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
: F21 16 /MOD ;
: F22 ASCII DUP EMIT ROT XOR SWAP ;
: F23 ASCII DUP EMIT XOR ;
: F2 17 1 DO I ADCIN F21 F22 F23 LOOP ;
: F31 ASCII EMIT ;
: F32 13 EMIT ;
: F3 F21 F31 F31 F32 ;
: BUILD BEGIN F1 F2 F3 1 0 = UNTIL ;
BUILD
```

3) ภาคแสดงผลและบันทึกข้อมูล

จากการที่ข้อมูลถูกส่งมาตลอดเวลา อีกทั้งระยะในการอ่านข้อมูลไม่จำเป็นต้องละเอียดมากนัก ทำให้ไมโครคอมพิวเตอร์มีความคล่องตัวขึ้น ผู้ใช้สามารถเรียกการทำงานต่างๆได้อย่างสะดวก และในโครงงานนี้ในส่วนนี้จะเป็นส่วนของการเขียนโปรแกรมทั้งหมด ซึ่งจะใช้ภาษาซีในการเขียนโปรแกรมดังจะกล่าวถึงต่อไป

ระบบเวลาจริง ( Real-Time System )

จากโครงสร้างของโครงงานที่กล่าวมา จะอยู่ในรูปของระบบเวลาจริง ซึ่งจะเป็นการนำเอาข้อมูล ณ เวลานั้นๆ มาเข้าสู่กระบวนการจัดการต่างๆที่ต้องการโดยมีคำจำกัดความคือ

" ลำดับของการคำนวณซึ่งถูกกำหนดโดยเวลาที่เปลี่ยนแปลงไป หรือ โดยเหตุการณ์ภายนอกไปยังคอมพิวเตอร์ และผลการทำงานในส่วนต่างๆที่อาจขึ้นกับค่าตัวแปรของเวลาที่ใช้ขณะที่ทำอยู่ "

การนำคอมพิวเตอร์ไปควบคุมระบบที่มีหลายๆกระบวนการรวมอยู่นี้ โดยในแต่ละกระบวนการจะทำงานทั้งหมดในสเกลเวลาของตัวเอง ซึ่งคอมพิวเตอร์นี้จะกล่าวได้ว่าทำงานในระบบเวลาจริง ต่อเมื่อการทำงานของคอมพิวเตอร์นั้นมีความสัมพันธ์กับสเกลเวลาของระบบภายนอกนั้น ในความสัมพันธ์นี้อาจกำหนดได้ในเทอมของการเปลี่ยนแปลงเวลาหรือตารางการทำ

023217
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งานของวัน ซึ่งเรียกระบบนี้ว่า " ฐานเวลา " ( Clock Base ) หรืออาจกำหนดในเทอมของเหตุการณ์ต่างๆ เช่น การปิด-เปิดสวิทช์ เป็นต้น ในกรณีนี้เรียกว่า " ฐานเซ็นเซอร์ หรือ ฐานเหตุการณ์ " ( sensor base or even base ) ประเภทที่สามคือ แบบ " ปฏิกิริยาต่อเนื่อง " ( Interactive ) เป็นความสัมพันธ์ระหว่างคอมพิวเตอร์กับระบบที่ถูกกำหนดไว้อย่างคร่าวๆ มีรูปแบบเป็นกลุ่มการทำงานของคอมพิวเตอร์ที่ควรเสร็จในเวลาที่กำหนด และกลุ่มงานสื่อสารก็จัดอยู่ในกลุ่มนี้ด้วย

ในส่วนของงานควบคุม ถึงแม้จะไม่มีการต่อโดยตรงกับอุปกรณ์ภายนอกก็ตาม ยังคงต้องการทำงานในรูปเวลาจริง เพราะเวลามักจะถูกรวมในการกำหนดตัวแปรต่างๆของการให้อัลกอริทึม ( Algorithm ) โดยรายละเอียดของแต่ละประเภทคือ

1. <u>ระบบฐานเวลา</u> ( Clock-Time-Base System )

จากการที่โปรเซสอาจจะถูกวัดในช่วงเวลาต่างๆ เช่น การวัดในทุกๆชั่วโมงสำหรับกระบวนการทางเคมี หรือ วัดเป็นมิลลิวินาทีสำหรับระบบในเครื่องบิน สำหรับตัวอย่างดังกล่าว การควบคุมป้อนกลับ ความต้องการอัตราสุ่มจะขึ้นกับคาบเวลาของโปรเซสที่ถูกควบคุม ระบบที่มีคาบน้อยๆ มีความต้องการอัตราสุ่มอย่างรวดเร็วเพื่อที่จะไม่ให้เกิดการสูญเสียข้อมูลที่เปลี่ยน-แปลงอย่างรวดเร็ว คอมพิวเตอร์ที่ใช้ควบคุมแพลนท์ ( Plant ) จะบรรลุผลการทำงาน ต้องเข้ากับระบบเวลาจริง หรือ เวลาธรรมชาติ ( Natural-Time ) ได้ การวัด , ควบคุม และปฏิบัติภายในช่วงเวลาของแต่ละคาบการสุ่ม การทำงานที่เสร็จสมบูรณ์ในช่วงเวลาที่เจาะจงนั้นขึ้นอยู่กับจำนวนของการกระทำที่ไปมีผลต่อการทำงานและความเร็วของคอมพิวเตอร์ การจัดการทางด้านเวลาให้สอดคล้องกันนั้นจะต้องมีสัญญาณเวลา (clock ) ที่เพิ่มในระบบคอมพิวเตอร์ ซึ่งใช้อ้างถึงสัญญาณเวลาจริง ( Real-Time clock ) และใช้สัญญาณจากสัญญาณเวลานี้ไปอินเทอรัพท์การทำงานของคอมพิวเตอร์ตามช่วงเวลาคงที่ที่กำหนด

การใช้คอมพิวเตอร์ควบคุมการทำงานโปรเซสทั่วไป การอินเทอรัพท์ของสัญญาณเวลาจะถูกตั้งไว้ให้มีอัตราเร็วกว่าอัตราการสุ่มข้อมูลเสมอ โดยการอินเทอรัพท์สัญญาณเวลาจะนับจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ห่วนการอินเทอรัพท์จนกระทั่งถึงเวลาที่ต้องการและรวมถึงระบบใหญ่ๆที่มีอัตราการสุ่มต่างกัน อีกทั้งจะถูกนำมาใช้ในการหรค่าวันและเวลาของการทำงาน

2. ระบบฐานเซ็นเซอร์ ( Sensor - Base System )

มีหลายๆระบบการทำงานที่ต้องทำไม่เพียงแต่ที่ส่วนของเวลาหรือช่วงของเวลา แต่ยังต้องการรู้ผลตอบสนองของเหตุการณ์ด้วย เช่น การปิดเปิดวาล์วหรือปั๊มเมื่อระดับของของเหลวในแทงค์ถึงค่าที่กำหนด หรือการปิดเปิดมอเตอร์เมื่อมอเตอร์ขับเคลื่อนถึงตำแหน่งที่ต้องการ ซึ่งจะถูกเช็คได้โดยไมโครสวิทช์ ฯลฯ ระบบฐานเซ็นเซอร์ นี้ใช้กันอย่างกว้างขวางในระบบที่มีการเตือนและการเริ่มแสดงสัญญาณเตือนต่างๆ เช่น อุณหภูมิสูงเกินไปหรือความดันมากไป เป็น-ต้น โดยลักษณะเฉพาะของระบบนี้มักจะรวมถึงความต้องการที่ระบบต้องตอบสนองภายในช่วงเวลาสูงสุดในแต่ละเหตุการณ์

ระบบนี้ปกติใช้การอินเทอรัพท์เพื่อบอกให้คอมพิวเตอร์ทราบว่า มีความต้องการทำงานเกิดขึ้น โดยในตัวอย่างสำหรับระบบแบบง่ายๆ อาจใช้วิธี " พูลลิ่ง " ( Polling ) (คอมพิวเตอร์จะติดต่อถามเพื่อทราบความต้องการในแต่ละตัวอุปกรณ์ต่างๆอย่างเป็นช่วงหรือคาบ)

3. ระบบปฏิกิริยาต่อเนื่อง

ระบบนี้คือระบบเวลาจริงใหญ่ๆที่มีการทำงานเป็นขั้นตอน ความต้องการเวลาจริง จะแสดงในเทอมของเวลาผลตอบสนองเฉลี่ยไม่เกินเวลาขอบเขตที่ได้ออกแบบไว้

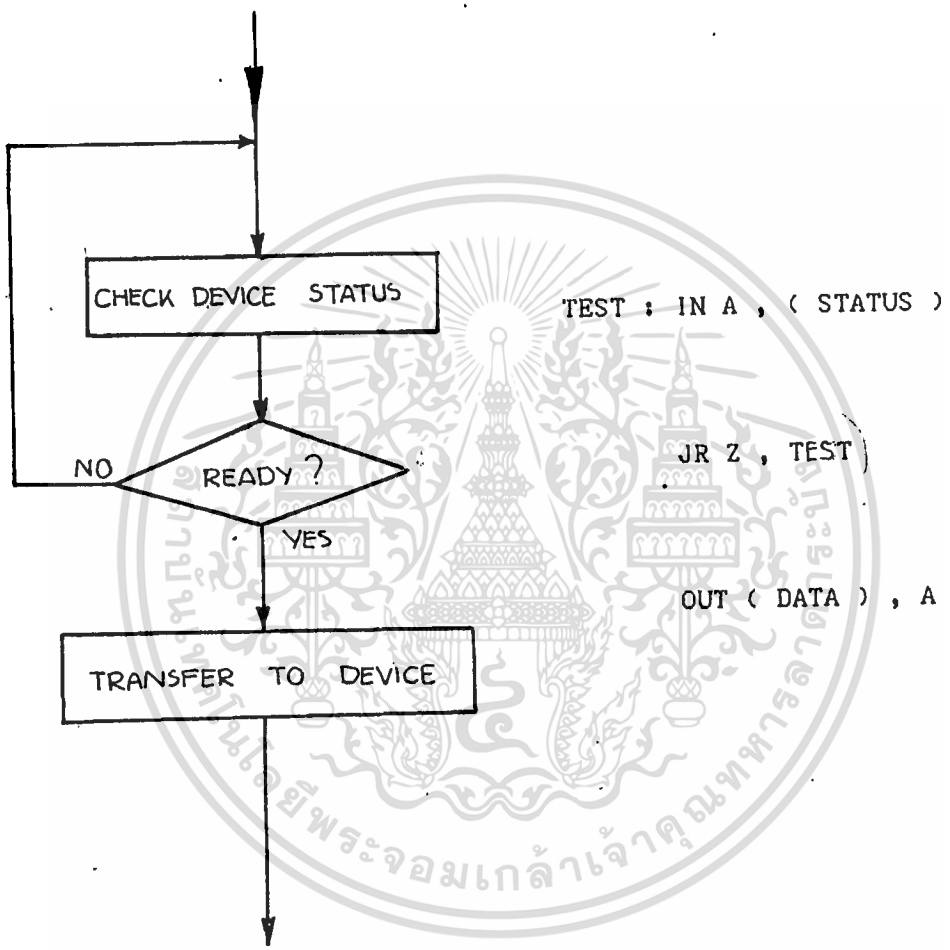
เทคนิคการเคลื่อนย้ายข้อมูล

ในการเคลื่อนย้ายข้อมูล , อ่านข้อมูลต่างๆจากโปรเซสที่มาทางอินเทอร์เฟสการ์ดสู่คอมพิวเตอร์นั้น ความแตกต่างของความเร็วในการทำงานนี้จะจำกัดความเร็วของการทำงานของคอมพิวเตอร์

ปัญหาสำคัญในการเคลื่อนย้ายข้อมูลคือเวลาในการใช้ ถ้ามีอุปกรณ์น้อยตัวติดต่อกับคอมพิวเตอร์ สามารถสร้างการเคลื่อนย้ายข้อมูลแบบไม่มีเงื่อนไขได้ สำหรับในบางระบบเอาท์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช่น เอาท์พุทสวิทช์ หลอดไฟฟ้าต่างๆ ฯลฯ สำหรับเครื่องพิมพ์และการส่งข้อมูลในช่องที่ติดต่อกับคอมพิวเตอร์ต้องแน่ใจว่าอุปกรณ์พร้อมที่จะรับส่งส่วนข้อมูลถัดไปนั้น ซึ่งจำเป็นที่จะต้องใช้การเคลื่อนย้ายแบบมีเงื่อนไขแสดงดังรูป 3.1

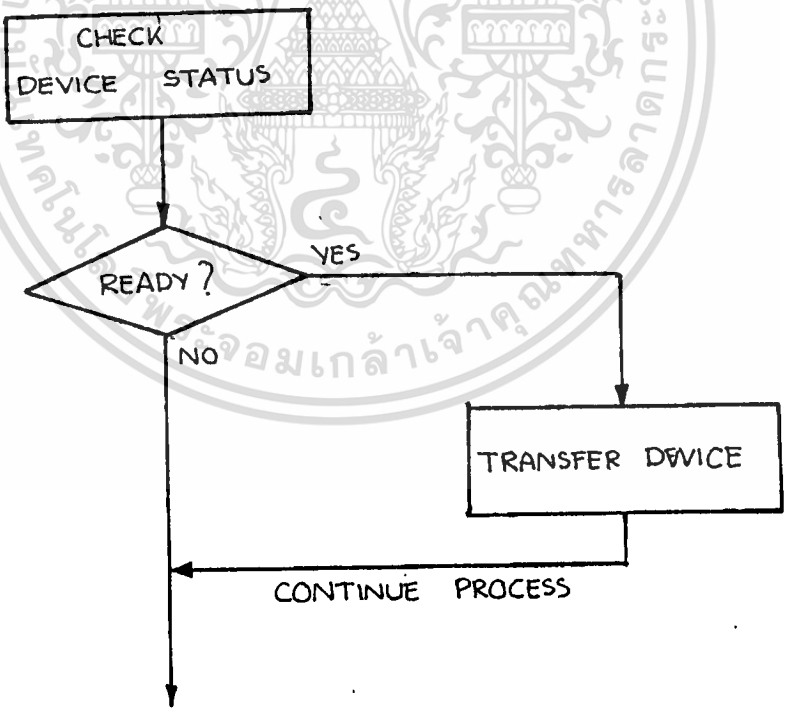



รูปที่ 3.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีนี้จะเห็นว่า จะไม่มีการเคลื่อนย้ายข้อมูลจนกว่าอุปกรณ์ติดต่อนั้นจะพร้อม ทำให้เสียเวลาในการรอการทำงานที่จุดๆเดียวมาก ไม่สามารถใช้เวลาที่รอนั้นไปทำงานอื่นได้ ทำให้มีประสิทธิภาพไม่มาก แม้ว่าการโปรแกรมและวงจรอินเตอร์เฟสจะง่ายและถูกแต่มีทางเลือกอีกทางคือ ดังตัวอย่างในรูปที่ 3.2 โดยให้คอมพิวเตอร์ทำงานอื่นต่อไปขณะที่อุปกรณ์นั้นยังยุ่งอยู่ และจะกลับมาเช็คสถานะอุปกรณ์นั้นอีกครั้งภายหลัง วิธีนี้หลีกเลี่ยงการรอในลูปสำหรับอุปกรณ์ - หนึ่งตัวที่จะเริ่มทำงาน

ถึงอย่างไรก็ตามก็ยังคงมีข้อเสียอยู่มาก ถ้าหากโปรแกรมกลับมาตรวจสอบอุปกรณ์เดิมไม่ทัน ทำให้บางข้อมูลสูญหายหรือการทำงานไม่ต่อเนื่อง เทคนิคการเคลื่อนย้ายข้อมูลแบบมีเงื่อนไขนี้รวมเรียกว่า การพูลลิ่ง เป็นการทำงานของคอมพิวเตอร์ที่ต้องเช็คว่าเมื่อไรอุปกรณ์-พร้อมที่จะเคลื่อนย้ายข้อมูล โดยปัญหานี้สามารถหลีกเลี่ยงด้วยการใช้วิธีอินเทอรัพท์

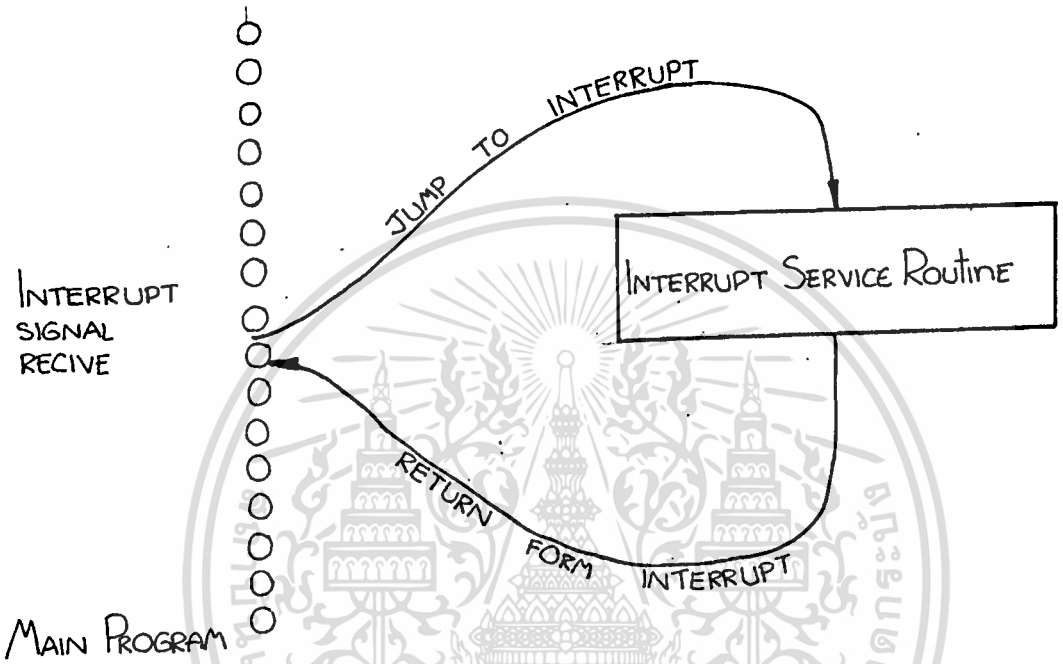



รูปที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินเทอรัพท์

การอินเทอรัพท์เป็นกลไกทำให้โปรแกรมหลักสามารถหยุดชั่วขณะ แล้วไปทำงานในส่วนของโปรแกรมเฉพาะได้ คือ ที่รูทีนอินเทอรัพท์นั้นๆ เมื่อการทำงานของรูทีนอินเทอรัพท์สิ้นสุดลง โปรแกรมหลักก็จะกลับมาทำงานอีกครั้งหนึ่ง โดยมีกระบวนการดังรูปที่ 3.3

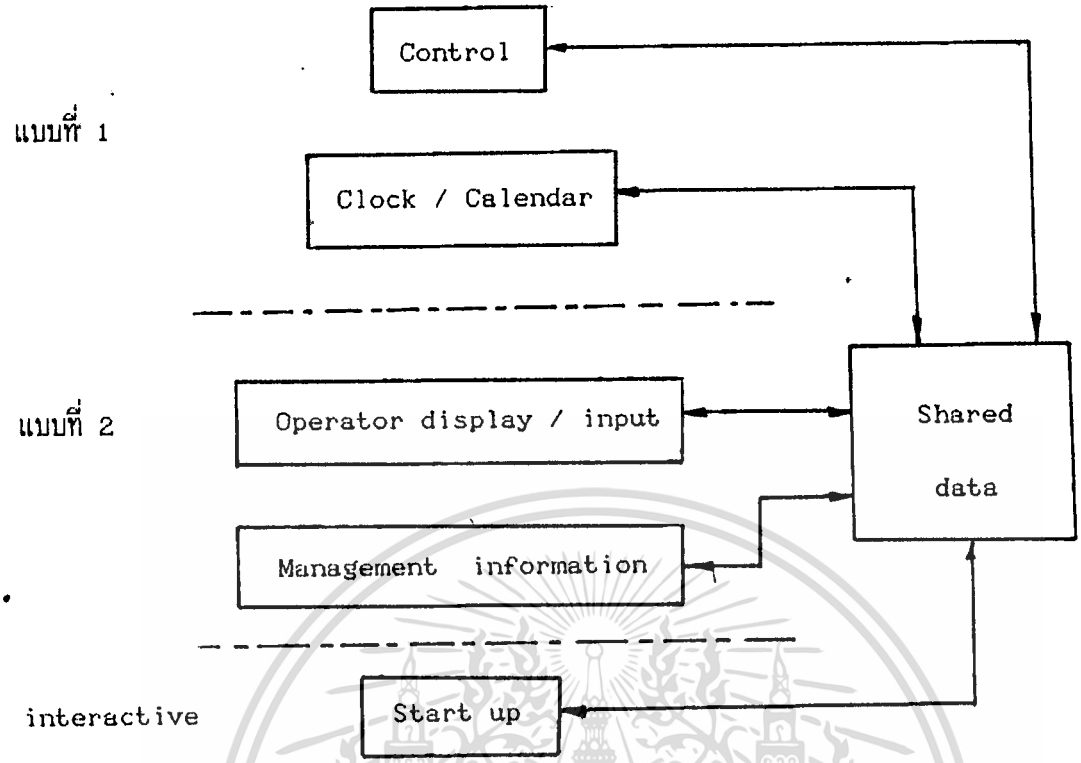



รูปที่ 3.3

เนื่องจากการอินเทอรัพท์สามารถเกิดได้ทุกๆจุดของโปรแกรม ดังนั้นจำเป็นต้องเก็บค่ารีจิสเตอร์ต่างๆในโปรแกรมไว้ก่อนที่จะถูกเขียนทับโดย อินเทอรัพท์รูทีน ( interrupt routine ) อันจะทำให้เกิดความผิดพลาดได้ การเก็บค่าต่างๆเหล่านี้ทำได้โดย

- เก็บไว้ในตำแหน่งหน่วยความจำพิเศษที่กำหนดขึ้นที่การอินเทอรัพท์ไม่สามารถรบกวนได้
- เก็บไว้ในหน่วยความจำสแตก ( memory stack ) ซึ่งเป็นที่นิยม แต่มีข้อเสียเมื่อสแตกเกิดโอเวอร์โฟลว์ ( overflow )
- เก็บไว้ในรีจิสเตอร์สำรอง รีจิสเตอร์สำรองสามารถแสดงกลไกของพื้นฐานการอินเทอรัพท์ที่ได้ดังรูปที่ 3.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 3.5 โครงสร้างโปรแกรม

แบบที่ 1 :

เป็นการตั้งค่าเวลามาตรฐานและอ่านเวลามาตรฐานเข้ามาเป็นข้อมูลโดยจะถูกใช้ในการกำหนดเวลาเก็บข้อมูล และเป็นข้อมูลเวลาของการเก็บข้อมูลที่จะถูกบันทึกลงในดิสก์เก็ตด้วยและเป็นเวลาจริงที่ถูกแสดงในมอนิเตอร์

แบบที่ 2 :

- อ่านข้อมูลที่ต้องการเก็บจากการอินเทอรัพท์เข้ามาเป็นข้อมูลร่วมใช้ในการแสดงออกมอนิเตอร์ , วาดกราฟและเก็บลงดิสก์เก็ต

- ส่วนการบันทึกข้อมูลจะบันทึกข้อมูลที่ถูกอ่านเข้ามาทุกๆ 4 วินาที โดยครบ 1 ชั่วโมงจะทำการเก็บลงดิสก์เก็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบจำลองปฏิกิริยาต่อเนื่อง (interactive module)

จากโครงสร้างที่แยกออกมานี้การสร้างการทำงานที่แตกต่างทำได้โดยสามารถแบ่งออกเป็นโปรแกรมย่อยๆต่างๆได้ โดยโปรแกรมย่อยจะต้องรวมคำสั่งต่างๆเข้าด้วยกันอย่างต่อเนื่องและถูกต้อง มี 3 วิธีด้วยกันในการออกแบบโปรแกรม ซึ่งจะต้องวิเคราะห์หารูปแบบวิธีที่ดีมาใช้โดยมีดังนี้คือ

- ซิงเกิลโปรแกรม ( Single program )
- ฟอร์กราวนด์ / แบ็คกราวนด์ ซิสเต็ม (Foreground / Background System)
- มัลติแทกส์กิ้ง ( Multi-tasking )

(1) ซิงเกิลโปรแกรม

ใช้สำหรับระบบที่มีขนาดเล็ก ซึ่งจะทำให้เส้นทางโปรแกรมชัดเจนและง่ายต่อความเข้าใจในการออกแบบได้ดี และสำหรับระบบงานที่มีทั้งจำนวนฮาร์ดแวร์ และซอฟท์แวร์น้อยๆจะง่ายต่อการการทดสอบแต่ปัญหาก็จะเพิ่มมากขึ้น โดยจะสมมติตัวอย่าง : จะเห็นว่าถ้าอินเทอรัพท์ของสัญญาณนาฬิกา เกิดขึ้นทุกๆ 20 มิลลิวินาทีแล้ว และถ้า $t_1$ , $t_2$ , t3 , t4 , และ t5 เป็นเวลาสูงสุดที่ใช้ในการคำนวณ สำหรับโมดูล Clock/Calender , Control , Generate display , input and manage output ระบบจะทำงานถูกต้องภายใต้ค่าที่ให้คือ

เมื่อ t1 + MAX ( t2 ,t3 ,t4 ,t5 ) < 20 มิลลิวินาทีนั่นเอง

(2) ระบบฟอร์กราวนด์/แบ็คกราวนด์

ข้อได้เปรียบของระบบนี้คือทางด้านเวลาจะไม่ถูกตรึงมากนัก จะมีโปรแกรมอินเทอรัพท์จากอินเทอรัพท์ฮาร์ดแวร์ ที่เป็นอิสระซึ่งเรียกว่า " ฟอร์กราวนด์ " และการทำงานในส่วนของซอฟท์แวร์หลัก เวลาจะไม่ถูกบังคับ

โดย t1 + t2 < 20 มิลลิวินาทีเท่านั้น เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

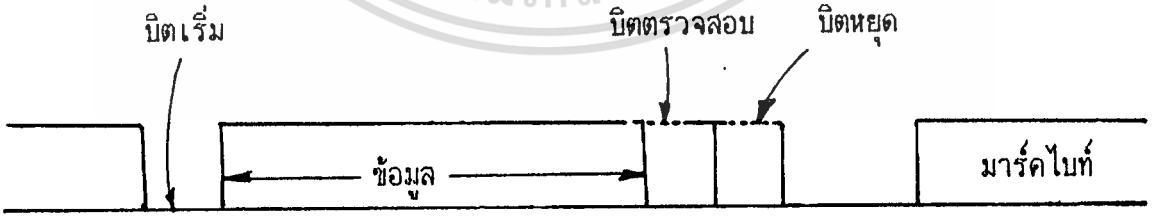
สำหรับส่วนของฟอร์กราวนด์ มักจะอยู่ในเทอมที่มีพรีออริตี้ ( priority ) สูงกว่างานแบ็คกราวนด์ และมีกลไกที่ต้องจัดการด้วยการขัดจังหวะงานแบ็คกราวนด์ จะเห็นได้ว่าการบังคับทางเวลาผ่อนคลายขึ้น การหาค่าของการกระทำของระบบเพื่อเช็คสมรรถนะที่ซับซ้อนทำได้ยากกว่ากรณี ซิงเกิลโปรแกรม

ถึงแม้โปรแกรม ฟอร์กราวนด์และแบ็คกราวนด์ จะแยกกันทำงานตามโครงสร้างของตัวเองแต่สองส่วนนี้จะเชื่อมโยงกันด้วยข้อมูลร่วมทำให้สามารถติดต่อกันได้ การทำงานของระบบนี้จะไม่ยากเกินในการควบคุมเข้าถึงตัวแปรร่วม

ในโครงงานนี้จะใช้การออกแบบโปรแกรมด้วยวิธีนี้ โดยจะกล่าวถึงต่อไป

<u>การสื่อสารข้อมูลแบบอนุกรม</u>

กำหนดให้การส่งข้อมูลจากบอร์ดควบคุมเป็นแบบ อสังวาล ( Asynchronousn ) ซึ่งอักขระที่ถูกส่งไปยังคอมพิวเตอร์นั้นไม่จำเป็นต้องต่อเนื่องกัน โดยมีสัญญาณที่ส่งออกไปนั้นเป็นตัวบอกจุดเริ่มต้นและจุดสุดท้ายรวมไปด้วย ซึ่งเรียกว่า บิตเริ่ม ( start bit ) และบิตหยุด ( stop bit ) รวมทั้งบิตตรวจสอบเพื่อให้ฝ่ายรับสามารถแยกค่าข้อมูลออกมาได้อย่างถูกต้องพร้อมกับการตรวจสอบข้อมูล โดยมีรูปแบบดังรูป




<u>รูปแบบการส่งข้อมูลแบบอสังวาล</u>

โดยการสื่อสารข้อมูลอนุกรมแบบ อสังวาลนี้ ต้องกำหนดค่าพารามิเตอร์ต่างๆ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความเร็วของการส่งข้อมูล ในที่นี้จะใช้ค่าบอร์ดเรท = 1200 ขนาดข้อมูลที่รวมเป็น 1 อักขระมีขนาด = 8 บิต , ชนิดของการตรวจสอบข้อมูล (ไม่มีการตรวจสอบ ( no parity ) ) ความยาวของบิตหยุดมีขนาด = 1 บิต

อแดปเตอร์สื่อสารแบบอนุกรมของ ไอบีเอ็ม พีซี

อแดปเตอร์สื่อสารแบบอนุกรมชนิดนี้มีระดับสัญญาณส่งออกเป็น 5 โวลท์ และ 0 โวลท์ ทำให้ต่อตรงเข้ากับบอร์ดควบคุมได้เลย มีไอซี 8250 เป็นหัวใจสำคัญโดยมีกำหนดบอร์ดเรท ที่สามารถควบคุมได้ในตัว และมีรีจิสเตอร์ใช้งานควบคุม 7 รีจิสเตอร์ด้วยกัน ไอบีเอ็ม พีซี ได้สำรองช่องของการสื่อสารแบบอนุกรมไว้ ซึ่งดอส ( DOS ) จะรับรู้ได้ 2 พอร์ท และมีการกำหนดหมายเลขของช่องทางเข้าออกของรีจิสเตอร์ควบคุมของ 8250 และความหมายการใช้งานของรีจิสเตอร์ใน 8250 มีดังนี้

| พอร์ท | รีจิสเตอร์ | การใช้งาน |
|---|---|---|
| XF8 | RBR , DLL | พักข้อมูลทั้งรับและส่ง , เป็นตัวหารความถี่ |
| XF9 | DLM , IER | เป็นตัวหารความถี่ ควบคุมการเกิดอินเทอรัพท์ |
| XFA | IIR | บอกชนิดของการอินเทอรัพท์ |
| XFB | LCR | ควบคุมพารามิเตอร์การรับส่ง บอกการใช้งานของ DLL และ DLM |
| XFC | MCR | ควบคุมโมเด็ม |
| XFD | LSR | บอกสถานะภาพการรับส่ง |
| XFE | MSR | บอกสถานะภาพของโมเด็ม |

ในการติดต่อให้ค่าแก่รีจิสเตอร์เหล่านี้ สำหรับการกำหนดข้อมูลการจัดหน้าที่การทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งานให้แก่ ไอซี 8250 นี้ จะใช้การเขียนโปรแกรม แอสแซมบลี เข้าไปติดต่อ จากตารางที่ให้มานี้ ส่วนที่สำคัญที่สุดของการทำงาน คือ ทางช่องเข้าออกหมายเลข XF9 ซึ่งมีลักษณะดังนี้

บิทที่ 0 สร้างสัญญาณอินเทอรัพท์เมื่อรับข้อมูลครบ 1 อักขระ

บิทที่ 1 สร้างสัญญาณอินเทอรัพท์เมื่อรีจิสเตอร์สำหรับส่งว่างลง

บิทที่ 2 สร้างสัญญาณอินเทอรัพท์เมื่อมีข้อผิดพลาดเกิดขึ้นจากการวัด

บิทที่ 3 สร้างสัญญาณอินเทอรัพท์เมื่อมีสัญญาณสถานะของโมเด็ม

เมื่อทำการเซ็ทบิท 0 คือการให้ 8250 สร้างสัญญาณอินเทอรัพท์เมื่อรับข้อมูลได้ 1 อักขระ คือ ความหมายเดียวกับการสร้างอินเทอรัพท์ทุกครั้ง เมื่อมีข้อมูลเข้ามานั่นเอง วิธีของโปรแกรมควบคุมการสื่อสารแบบการขัดจังหวะการทำงานของซีพียู เมื่อรับข้อมูลเข้ามาทางช่องสื่อสารนี้ จะทำให้โปรแกรมหลักมีความคล่องตัวไม่ต้องพะวงถึงข้อมูลที่จะเข้ามาทางช่องสื่อสาร เพราะมีโปรแกรมการขัดจังหวะคอยจัดการให้อยู่แล้ว ซึ่งจะเชื่อมการทำงานกับฮาร์ดแวร์อินเทอรัพท์ของไอบีเอ็มพีซี สำหรับตัวฮาร์ดแวร์อินเทอรัพท์เอง มีสัญญาณการขัดจังหวะการทำงานของซีพียูได้ 8 ช่องทาง โดยใช้ 8259 ควบคุมการขัดจังหวะซึ่งได้กำหนดการใช้งานไว้ดังนี้ คือ

IRQ0 : สำหรับการขัดจังหวะของไทม์เมอร์

IRQ1 : สำหรับการขัดจังหวะของคีย์บอร์ด

IRQ2 : สำรอง

IRQ3 : สำหรับช่องทางสื่อสารหมายเลข 1

IRQ5 : สำหรับช่องทางสื่อสารหมายเลข 0

IRQ5 : สำหรับดิสก์เก็ตคอนโทรลเลอร์

IRQ6 : สำหรับดิสก์เก็ตคอนโทรลเลอร์

IRQ7 : สำรอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และขั้นตอนการยอมให้สัญญาณการขัดจังหวะผ่านไปถึงซีพียู ซึ่งจะอธิบายการเดินทางของสัญญาณการขัดจังหวะจากช่องสื่อสารอนุกรมได้ดังนี้

1. เมื่อเหตุการณ์เกิดขึ้นที่ 8250 ตามที่เรากำหนดเหตุการณ์ที่จะทำให้เกิดสัญญาณอินเทอรัพท์ที่พอร์ท XF9 สัญญาณ Instr จะออกมาจาก 8250 เมื่อ 8250 รับข้อมูลครบ 1 อักขระ
2. สัญญาณ Instr ที่ออกมาจาก 8250 ยังถูกควบคุมด้วยเกตตัวหนึ่งซึ่งสามารถควบคุมได้โดยพอร์ท XFC บิทที่ 3 (สัญญาณ out 2)
3. เมื่อ 8259 รับสัญญาณ IRQ4จะตรวจสอบดู IMR(interrupt mask register) ว่าจะรับหรือไม่ 8259 จะรับสัญญาณ IRQ ต่อเมื่อบิทที่ตรงกับหมายเลข IRQ ในไอเอ็มอาร์รีจิสเตอร์เป็น 0 เท่านั้น และเข้าไปควบคุมไอเอ็มอาร์ได้โดยช่องทางเข้าออกหมายเลข 21H (พอร์ท = 21H)
4. ถ้า 8259 รับสัญญาณ IRQ 8259 จะตรวจสอบลำดับความสำคัญของสัญญาณ IRQ ว่าขณะนี้ซีพียูกำลังทำงานให้กับอินเทอรัพท์ไหนบ้าง โดยตรวจสอบดูที่รีจิสเตอร์ ISR (Interrupt Service Register ) ถ้าซีพียูไม่ได้บริการอินเทอรัพท์อะไรที่สำคัญกว่า 8259จะส่งสัญญาณอินเทอรัพท์ต่อไปยัง 8088 ผ่านทางอินเทอรัพท์มาสก์เอเบิ้ล ( MI : Maskable Interrupt )
5. ทุกๆคำสั่งของ 8088 หลังจากปฏิบัติแล้วจะตรวจดูสัญญาณอินเทอรัพท์ที่ขา นอนอินเทอรัพท์มาสก์เอเบิ้ล ( NMI : Non Interrupt Maskable ) ซึ่งไอบีเอ็มพีซีต่อเอาไว้กับการตรวจความผิดพลาดของหน่วยความจำ และ MI สำหรับสัญญาณ MI จะมีผลทำให้ 8088 ให้ความสนใจต่อเมื่ออินเทอรัพท์แฟลกเป็น 1
6. เมื่อ 8088 รับสัญญาณ mi จะส่งสัญญาณไปบอก 8259 ว่าตกลงรับการร้องขออินเทอรัพท์กลับไปยัง 8259
7. 8259 จะเอาหมายเลขทิศทางการขัดจังหวะ (0CH ในที่นี้) ส่งไปให้ 8088 ผ่านทางดา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้าบัสพร้อมทั้งกับตั้งค่า ISR บิทที่ตรงกับหมายเลขสัญญาณ IRQ ให้เป็น 1 เพื่อเป็นการเตือนตัวเองว่าขณะนี้ซีพียูกำลังให้บริการอินเทอรัพท์ไหนบ้าง

8. 8088 รับเอาหมายเลขทิศทางการขัดจังหวะจากดาต้าบัส แล้วเก็บค่าแฟลกและเซ็กเม้นท์ของโปรแกรม และโปรแกรมเคาท์เตอร์ขณะนั้นลงในสแตก พร้อมทั้งกระโดดไปยังอินเทอรัพท์รูทีน ที่ชี้บอกโดยตารางอินเทอรัพท์เวคเตอร์ (interrupt vector table) พร้อมทั้งเคลียค่า แฟลก F ให้เป็น 0 ไม่ยอมให้เกิดอินเทอรัพท์มาสก์เอเบิ้ลอีกต่อไป
9. เป็นหน้าที่ของอินเทอรัพท์รูทีนที่จะจัดต่อไป

<u>เทคนิคการเขียนรูทีน</u>

1. ตั้งค่าพารามิเตอร์ต่างๆสำหรับการสื่อสาร
2. กำหนดชนิดของการเกิดสัญญาณอินเทอรัพท์ใน 8250
3. กำหนดตารางเวคเตอร์( vector table ) สำหรับการอินเทอรัพท์ของการสื่อสารที่จะเข้ามา
4. ตั้งค่า IMR ใน 8250 โดยผ่านทางพอร์ท 21H
5. เปิดอินเทอรัพท์เกต ใน 8250 โดยที่บิทที่ 3 ของ MCR ให้เป็น 1
6. คืนอินเทอรัพท์เวคเตอร์เก่ากลับคืน เมื่อเลิกใช้โปรแกรมนี้

<u>การสร้างยูนิตจอภาพจากวีดีโอแรม</u>

วิธีแสดงอักขระออกจอภาพที่นิยมใช้ คือ จัดหน่วยความจำบริเวณหนึ่งให้เป็นหน่วยความจำจอภาพ เรียกว่า " วีดีโอแรม " โดยถ้านำระหัสตัวอักษรใดก็ตามมาใส่ไว้ที่หน่วยความจำบริเวณนี้ จะทำให้เกิดตัวอักษรนั้นบนจอภาพทันที

โดยปกติจอโมโนโครมมีจุดเริ่มต้นของวีดีโอแรมที่แอดเดรส : $ B000 : $ 0000 ส่วนวีดีโอแรมจอสีจะเริ่มต้นที่แอดเดรส : $ B800 : $ 0000 ขนาดของวีดีโอแรมสามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำนวณได้จากขนาดของจอภาพ เนื่องจากจอภาพโมโนโครมมีขนาด 80 * 25 = 2000 จากการมี 25 บรรทัด บรรทัดละ 80 ตัวอักษร จึงควรใช้เนื้อที่หน่วยความจำเท่ากับ 80*25 เท่ากับ 2000 ตัวอักษร ในแต่ละตัวอักษรจะใช้เนื้อที่หน่วยความจำถึง 2 ไบท์ นอกจากระหัสของตัวอักษรมันเอง 1 ไบท์แล้ว จะใช้เนื้อที่อีก 1 ไบท์ ในการเพิ่มหน่วยความสามารถในการแสดงผลตัวอักษรแบบต่างๆ เช่น ตัวกระพริบ , ขีดเส้นใต้ , ตัวรีเวอร์ส ฯลฯ ซึ่งจะเรียกค่านี้ว่า " แอททริบิวท์ " ( attribute ) โดยมีตำแหน่งติดกับระหัสตัวอักษรตัวนั้นๆ

แอททริบิวท์ไบท์ : เป็นสิ่งที่จะบ่งบอกถึงลักษณะการแสดงผลของตัวอักษร แต่ละบิทในแอททริบิวไบท์จะมีความหมายเฉพาะตัว สำหรับจอภาพโมโนโครม จะเป็นแบบการแสดงผลรีเวอร์ส , ขีดเส้นใต้ , กระพริบ , หรือความเข้ม ตามตาราง

| ชื่อแอตทริบิวต์ | ลักษณะแอตทริบิวต์ | ค่าแอตทริบิวต์ |
|---|---|---|
| Nodisplay | ไม่แสดงตัวอักษร | $00 |
| Lowdisplay | ตัวอักษรจาง | $07 |
| Highdisplay | ตัวอักษรเข้ม | $0F |
| UnderlineLow | ตัวจางขีดเส้นใต้ | $01 |
| UnderlineHigh | ตัวเข้มขีดเส้นใต้ | $09 |
| ReverseLow | ตัวจางรีเวอร์ส | $70 |
| ReverseHigh | ตัวเข้มรีเวอร์ส | $78 |
| BlinkLow | ตัวจางกะพริบ | $17 |
| BlinkHigh | ตัวเข้มกะพริบ | $1F |
| UndBlinkLow | ตัวจางขีดเส้นใต้กะพริบ | $81 |
| UndBlinkHigh | ตัวเข้มขีดเส้นใต้กะพริบ | $89 |
| RevBlinkLow | ตัวจางรีเวอร์สกะพริบ | $F0 |
| RevBlinkHigh | ตัวเข้มรีเวอร์สกะพริบ | $F8 |

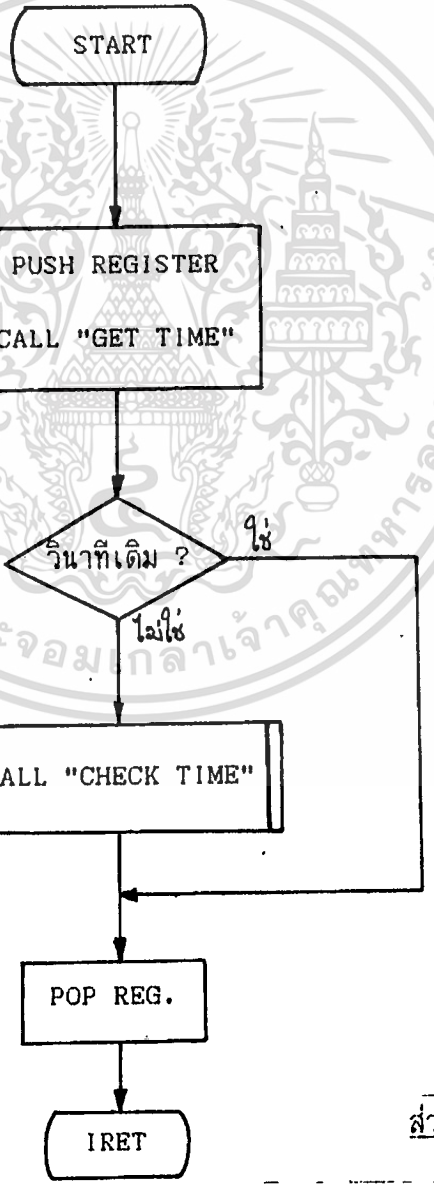
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 3

## การออกแบบโปรแกรมควบคุม

รายละเอียดของโปรแกรมควบคุม

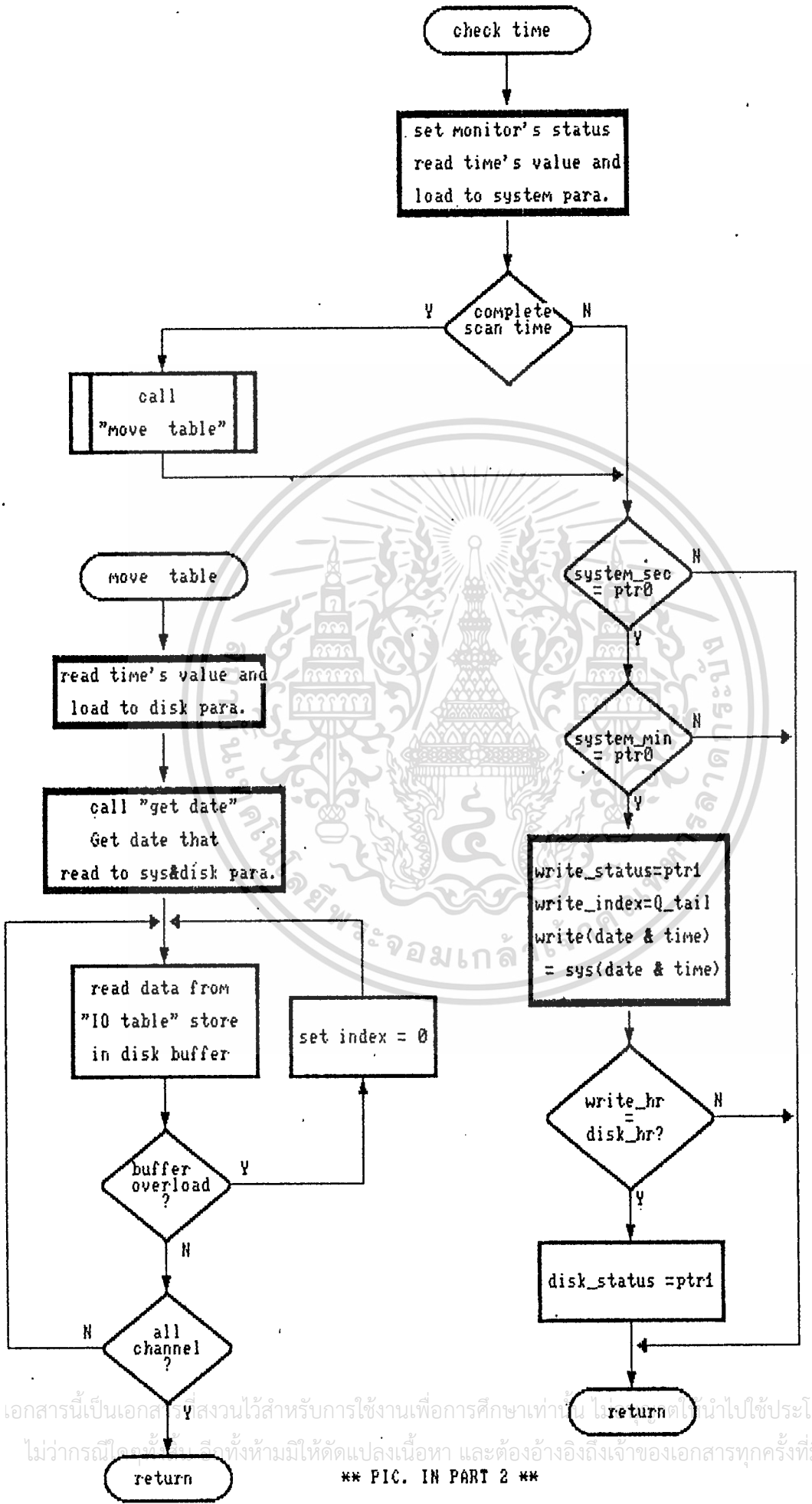
จากการที่โปรแกรมอยู่ในรูปแบบของฟอร์กราวนด์ / แบ็คกราวนด์ โปรแกรม จะอธิบายรายละเอียดของส่วนโปรแกรมก่อน คือ ฟอร์กราวนด์โปรแกรม




ส่วนที่ 1

FIG : Timer interrupt routine

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

** PIC. IN PART 2 **

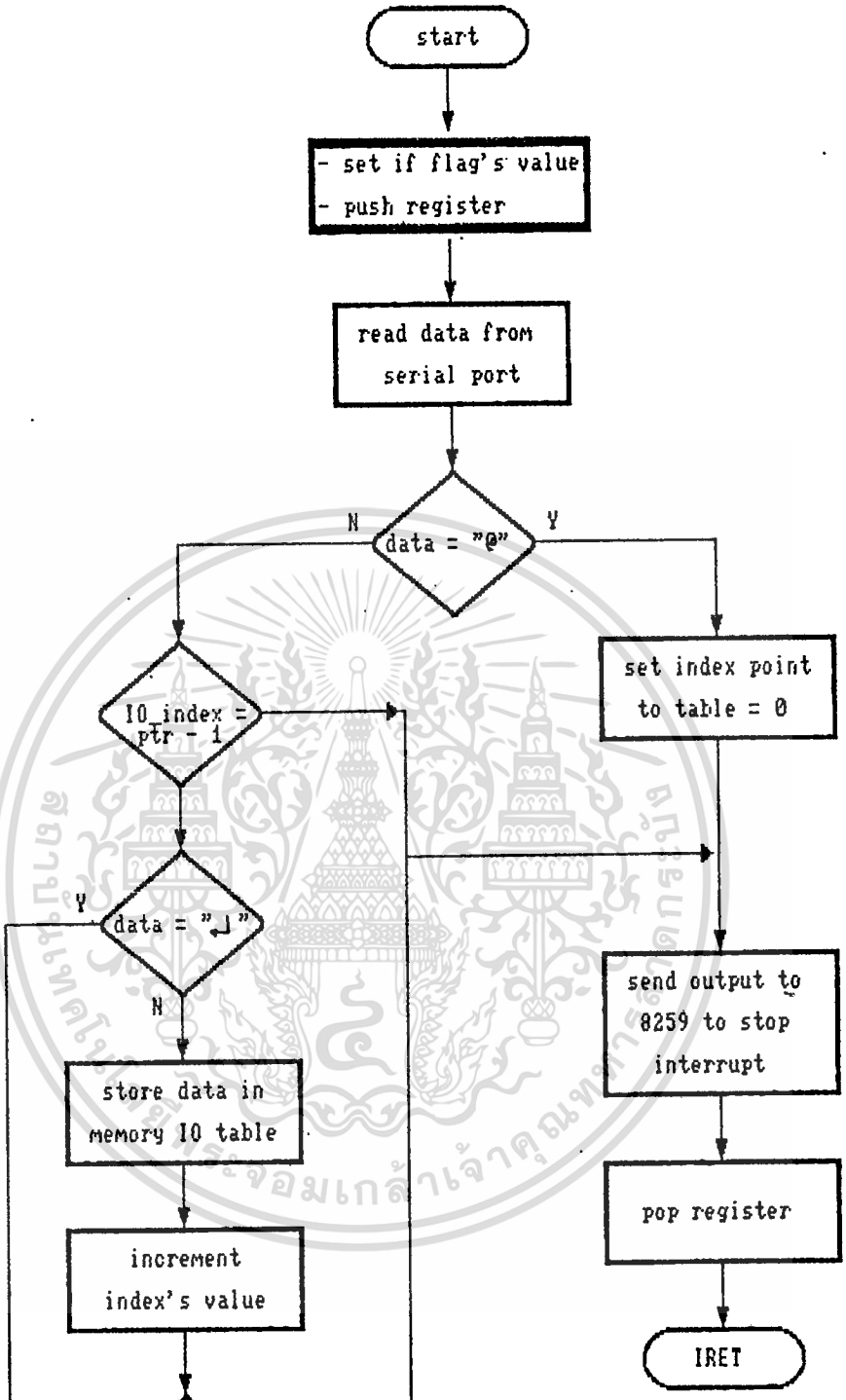
ส่วนที่ 1 : ไทม์เมอร์อินเทอรัพท์รูทีน ( timer interrupt routine )

เมื่อสัญญาณไทม์เมอร์อินเทอรัพท์เข้ามา ทำให้รูทีนส่วนนี้ทำงาน จะใช้ INT 21H FUNCTION 2CH คือ "GET TIME" อ่านค่าเวลาปัจจุบันที่กำหนด โดยออกมาในรูปของตัวเลขฐานสอง แล้วเช็คว่าวินาทีที่อ่านได้มีการเปลี่ยนแปลงหรือยัง ถ้ายังก็ไม่ต้องไปเปลี่ยนแปลงอะไร แต่ถ้าวินาทีเปลี่ยนไป การทำงานจะเข้าสู่โปรแกรมย่อย " เช็คไทม์ " ดังรูปส่วนที่ 2 คือ จะทำการเซ็ทสถานะให้แก่ตัวแปรมอนิเตอร์ที่ตั้งขึ้น นำค่าเวลาที่อ่านได้มาให้แก่ตัวแปรเวลาของระบบ จากนั้นก็ตรวจดูว่าครบรอบเวลาสแกนแล้วหรือยัง โดยเวลานี้จะเป็นค่าที่ใช้บอกถึงช่วงเวลาในการบันทึกค่าข้อมูลทั้ง 16 แชนแนลจากไอโอเทเบิ้ล ลงในหน่วยความจำส่วนหนึ่ง (ดิสก์บัฟเฟอร์) ในที่นี้ตั้งค่าเวลาสแกนไว้ทุกๆ 4 วินาที หลังจากนั้นก็จะตรวจว่า - ครบ 1 ชั่วโมงแล้วหรือยัง ถ้าครบหนึ่งชั่วโมงจะทำการเซ็ทตัวแปรเงื่อนไขเกี่ยวกับการบันทึกข้อมูล เมื่อในแบ็คกราวนด์โปรแกรมทำการตรวจสอบตัวแปรส่วนนี้ เมื่อพบว่าถูกเซ็ท ( ครบ 1 ชั่วโมง ) ก็จะทำการบันทึกข้อมูลจากหน่วยความจำที่ใช้เก็บข้อมูลนั้นลงสู่ดิสก์เก็ต

ส่วนที่ 2 : รูทีนอินเทอรัพท์สื่อสาร ( communication interrupt routine )

เมื่อมีข้อมูลเข้ามาที่ช่องสื่อสาร จะเกิดการอินเทอรัพท์ทำให้รูทีนส่วนนี้ทำงาน และจากรูปแบบของการส่งข้อมูลของบอร์ดควบคุมที่จะส่งอักขระ @ เป็นจุดบอกว่าอักขระต่อมาจะเป็นค่าของข้อมูลช่องที่ 1-16 และปิดท้ายด้วย " ↵ " ทำให้โปรแกรมส่วนรับข้อมูลเข้ามามีรูปแบบดังนี้คือ เมื่ออักขระถูกอ่านเข้ามาจากพอร์ทอนุกรม มีค่าเป็น " @ " จะทำการเซ็คค่าอินเด็กซ์ (index) ให้อยู่ในรูปที่จุดเริ่มต้นของตำแหน่งตารางข้อมูล เพื่อเป็นการกำหนดตำแหน่งในตารางให้ถูกต้องกับตำแหน่ง แชนแนลอินพุท ที่รับเข้ามา หลังจากนั้นเมื่อมีข้อมูลเข้ามาจะถูกเก็บไว้ใน ไอโอ-เทเบิ้ล ( I-O TABLE ) นี้โดย อินเด็กซ์จะเพิ่มค่าขึ้นเรื่อยๆตามจำนวนแชนแนล ดาต้า อินพุท ที่เข้ามา จนครบจำนวน อินพุทที่ตั้งไว้ 16 แชนแนล หรือพบค่า " ↵ " ก็ทำการมาร์คตัวแปร ไอโอ-อินเด็กซ์ (IO-INDEX) เพื่อที่จะแสดงถึงสถานะที่จะไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




ส่วนที่ 2 รูทีนอินเทอรัพท์สื่อสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รับค่าตัวแปรที่จะเข้ามาเก็บไว้ในตารางอินพุทอีก จนกว่าจะพบ " @ " อีกครั้งหนึ่ง

แบ็คกราวนด์โปรแกรม

เป็นส่วนของโปรแกรมหลักที่ทำหน้าที่ต่างๆ เช่น

- แสดงเป็นมอนิเตอร์เซ็นเซอร์ ซึ่งจะเป็นมอนิเตอร์หลัก แสดงค่าอินพุทที่วัดเข้ามาอยู่ตลอดเวลา
- บันทึกข้อมูลลงดิสก์เก็ต
- แสดงกราฟอินพุทที่วัดเข้ามาเทียบกับเวลา
- ส่วนของการแก้ไขและเซ็ทค่าต่างๆให้กับการทำงานของโปรแกรม เช่น
  : การตั้งและแก้ไขเวลาหรือบอกค่าเวลาให้กับเวลาของระบบ
  : การตั้งและแก้ไขเวลาเริ่มต้นการบันทึกข้อมูล
  : การตั้งและแก้ไขเวลาสิ้นสุดของการบันทึกข้อมูล
  : การกำหนดอันดับของเซ็นเซอร์แชนแนลที่ต้องการแสดงออกทางกราฟ
- ส่วนเช็คระบบและการออกจากโปรแกรม

ซึ่งจะแสดง โฟลว์ชาร์ท (FLOW CHART) อย่างคร่าวๆของแบ็คกราวนด์โปรแกรมดังรูปที่ 1 F จะแสดงส่วนประกอบต่างๆดังนี้

ส่วนที่ 1

- กำหนดค่าตัวแปรต่างๆให้ระบบ และในตัวแปรบางตัวที่อาจเปลี่ยนแปลงได้บ่อย เช่น สเกลต่างๆ เป็นต้น
- ในการออกแบบตั้งค่าบอร์ดเรทให้กับระบบ เมื่อทำการเอ็กซิคิวท์ (Execute) โปรแกรม หลังจากแปลงโปรแกรมให้เป็นออปเจ็คโคด (object code) นั้น ถ้าไม่มีการกำหนดค่าบอร์ดเรทแก่โปรแกรมที่อากิวเม้นท์ (arguement) ถัดจากชื่อไฟล์ (file) แล้ว จะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซ็ทค่าบอร์ดเรทให้เป็น 1200 โดยอัตโนมัติ

- ซีเรียลเซ็ทอัพ (serial set up)

: เซ็ทค่าบอร์ดเรท , บิทตรวจสอบ , บิทหยุด และความยาวของข้อมูลสำหรับการสื่อสาร เคลียค่าในหน่วยความจำไอโอ-เทเบิ้ล ให้เป็นศูนย์หมด

: INT 21H ฟังก์ชั่น 35H (อ่านค่าเวคเตอร์อินเทอรัพท์) ที่ OCH เก็บค่าแอดเดรสของเวคเตอร์ OCH นี้ไว้ที่ตัวแปร ซีเรียลเซ็กเม้นท์ (serisl-segment) และ ซีเรียลออฟเซ็ท (serial-offset)

: ตามด้วย INT 21H ฟังก์ชั่น 25H แล้วให้ค่าแอดเดรสของโปรแกรมอินเทอรัพท์ ที่ตำแหน่งเดียวกับตำแหน่งเริ่มต้นของ อินเทอรัพท์รูทีนสื่อสาร เป็นการเปลี่ยนตารางทิศทางการขัดจังหวะ ( interrupt vector table ) ให้ไปชี้ที่โปรซีเดอร์อินเทอรัพท์ที่ได้สร้างขึ้น

: เปิดอินเทอรัพท์เกตใน 8250 โดยให้บิทที่ 3 ของ MCR เป็น 1 (3fc)

: กำหนดการเกิดสัญญาณอินเทอรัพท์ทาง (XF8) เซ็ท บิท 0 ให้สร้างสัญญาณอินเทอรัพท์เมื่อรับข้อมูลครบ 1 อักขระ

: ตั้งค่า IMR ใน 8259 โดยผ่านพอร์ท 21H ให้ยอมรับสัญญาณอินเทอรัพท์จาก 8250

- อินนิเทียลสกรีน (initial-screen)

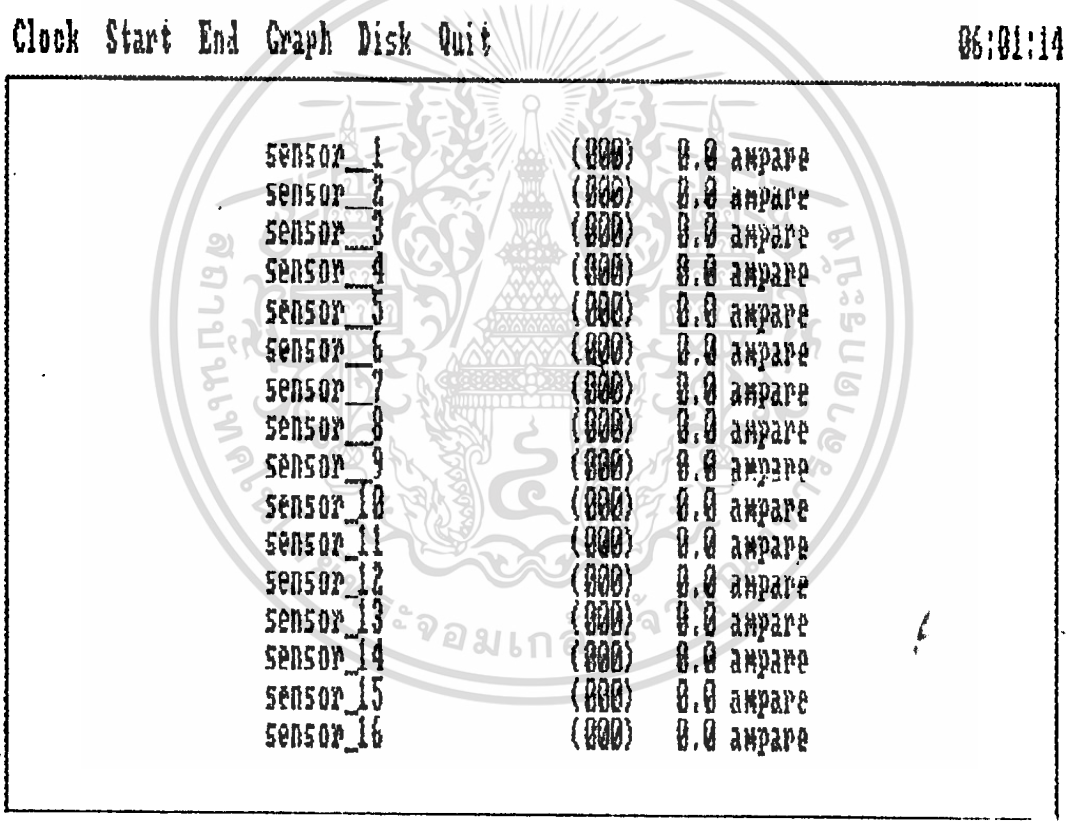
โดยมีขั้นตอนคือ

: ทำการสร้างวินโดว์ (window) สำหรับทำเป็นมอนิเตอร์ดิสเพลย์ และสร้างรูปแบบแสดงเฮดเมนู (head menu) คือ (clock,start,end,graph,disk,quit) สำหรับให้ผู้ใช้ได้เห็นและเลือกใช้งานเฉพาะอย่างตามที่ต้องการ

: แสดงชื่อเซ็นเซอร์และหน่วยเซ็นเซอร์ทั้ง 16 แชนแนล ตามลำดับ

: มอนิเตอร์เซ็นเซอร์ อ่านข้อมูลจาก ไอโอ-เทเบิ้ล แล้วแสดงออกทางจอ ตามตำแหน่งที่ได้จัดให้แต่ละ แชนแนล ไว้ก่อนแล้ว โดยข้อมูลที่อ่านจาก ไอโอ-เทเบิ้ลแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูป : มอนิเตอร์โปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงออกทางจอนี้ ต้องถูกแปลงให้เป็นข้อมูลจริงที่อ่านได้ในสเกลที่ถูกต้องก่อนที่จะส่งแล้วแสดงออกทางจอ โดยมีสูตรคือ

f = b + {[(a-b)*(f-c)]/[d-b]}

โดยที่

a = ค่าเซ็นเซอร์ที่สูงสุด ( sensor_high value )

b = ค่าเซ็นเซอร์ที่ต่ำสุด ( sensor_low value )

c = สเกลต่ำของเซ็นเซอร์ ( sensor_low scale )

d = สเกลสูงของเซ็นเซอร์ ( sensor_high scale )

e = ค่าอินพุทที่อ่านได้ ( IO value )

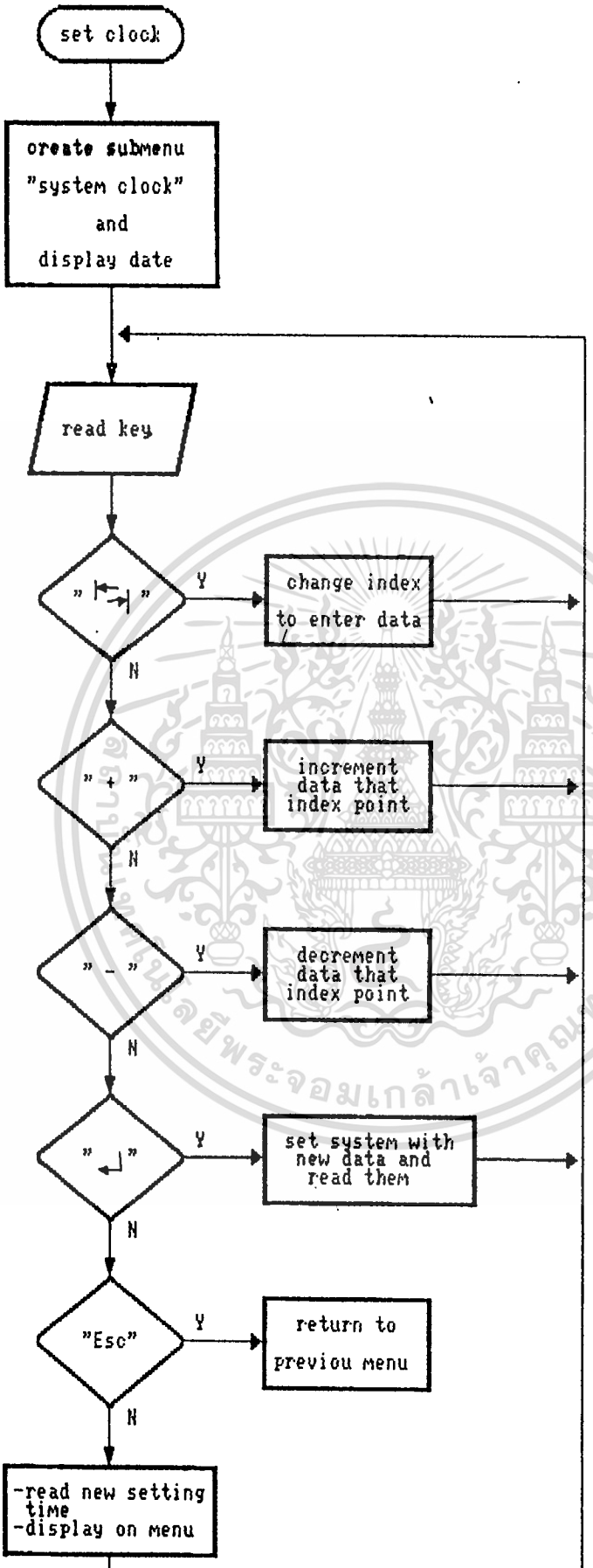
f = ค่าที่แสดงออกทางจอ

ส่วนที่ 2

เมื่อการตรวจรับคีย์พบว่าคีย์ " C " ถูกกด การทำงานของโปรแกรมจะไปทำที่ส่วนตั้งเวลา " set clock " โดยมีการทำงานของโปรแกรมย่อยตามโฟลว์ชาร์ทดังรูปที่ 2F

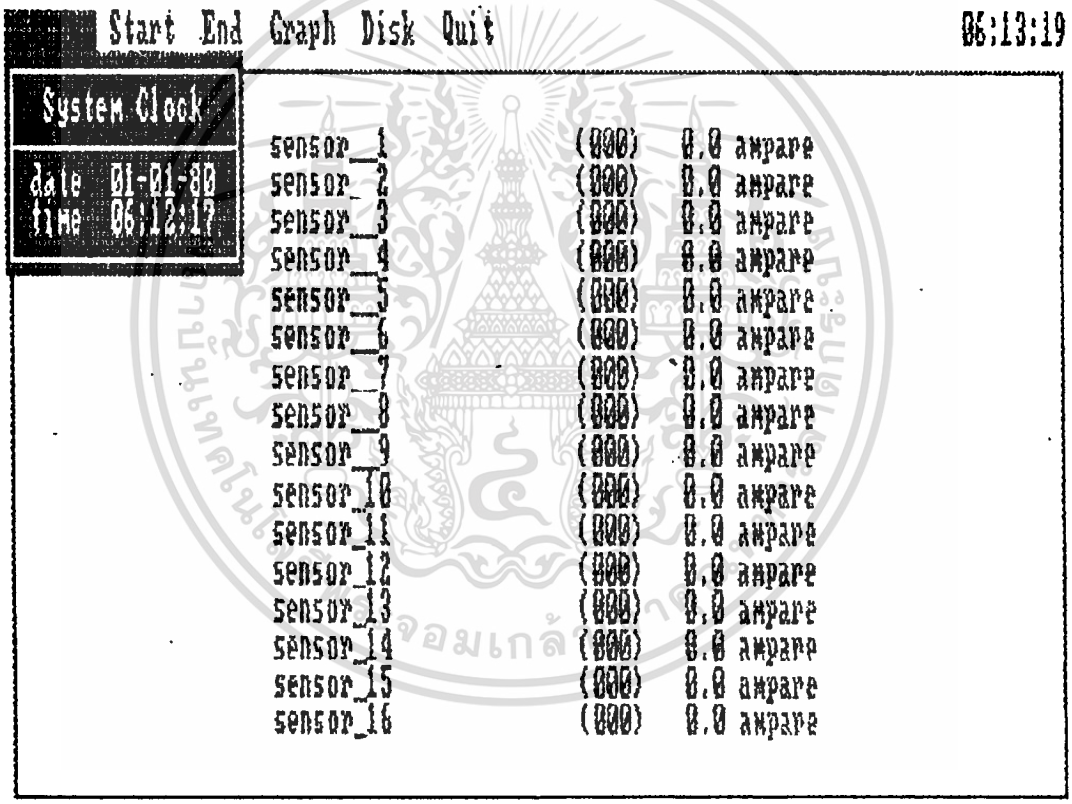
การสร้างแอททริบิวท์อักษรรีเวอร์ส โดยเริ่มต้นที่ข้อมูลแรกคือ วัน (date) เพื่อที่จะบอกถึงชนิดข้อมูลที่อาจจะทำการแก้ไขได้ในขณะนั้น และเมื่อต้องการแก้ไขข้อมูลชนิดนั้นให้มีค่าเพิ่มหรือลด ก็จะอยู่ในโหมด คีย์ "+" และ "-" ตามลำดับ เมื่อต้องการเปลี่ยนชนิดของข้อมูลเพื่อทำการเซ็ทใหม่ คีย์ " ← " ก็จะเป็นตัวเลื่อนอินเด็กซ์ที่ชี้ชนิดของข้อมูลเอง เมื่อทำการกำหนดค่า เวลาของระบบ (system clock) ใหม่เรียบร้อยแล้ว คีย์ " ↵ " เมื่อถูกกดจะมีโปรแกรมย่อยนำค่าที่เปลี่ยนแปลงแก้ไขทางเมนู เวลาของระบบ นี้ไปเซ็ทให้กับระบบ และทำการเช็คการทำงานโดยการอ่านค่ากลับเข้ามาอีกครั้ง แสดงออกทางเมนูเดิม ดูว่าเป็นค่าเดียวกับที่เปลี่ยนแปลงแก้ไขใหม่แล้วหรือไม่ เมื่อต้องการออกจากโหมดโปรแกรมเซ็ทเวลาของระบบ ก็ทำได้โดยกด คีย์ " ESC " จะนำเพลทมอนิเตอร์เดิมที่ถูกเมนูย่อยนี้ทับไว้กลับมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

** PIC. 2F **




รูป : มอนิเตอร์โปรแกรม "เซ็ท คล็อค"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงอีกครั้ง นั่นคือ การแสดงมอนิเตอร์หลักแบบเดิมอีกครั้ง

ส่วนที่ 3

ขั้นตอนการทำงานแสดงได้ดังรูปที่ 3F

เริ่มจากการสร้างเมนูสำหรับแสดงค่าเริ่มต้นที่ตั้งไว้ก่อน โดยอ่านจากตัวแปร ---- ( start_date_time ) โดยรูปแบบของคีย์อินพุท ค่อนข้างจะเหมือนกันโปรแกรมย่อย ตั้งเวลา โดยโหมด คีย์ "⇤⇥" จะเป็นตัวเลื่อนอินเด็กซ์ในการชี้ประเภทของข้อมูลในเมนู และแอททริบิวท์รีเวอร์ส ตามตำแหน่งของอินเด็กซ์ที่ชี้

โหมดคีย์ " + " , " - " จะลดหรือเพิ่มค่าข้อมูลในแต่ละชนิดนั้นๆ ตามตำแหน่งที่อินเด็กซ์ชี้อยู่

โหมดคีย์ "↵" จะโอนค่าวันที่และเวลาที่เซ็ทไว้ขั้นต้น สู่ตำแหน่ง (start_date_time) เพื่อนำไปใช้ในงานต่อไป โดยโหมดคีย์ " S " จะอ่านค่าวันและเวลาจากระบบแล้วนำค่านี้มาเซ็ทเป็นค่าเริ่มต้นของการบันทึกข้อมูล

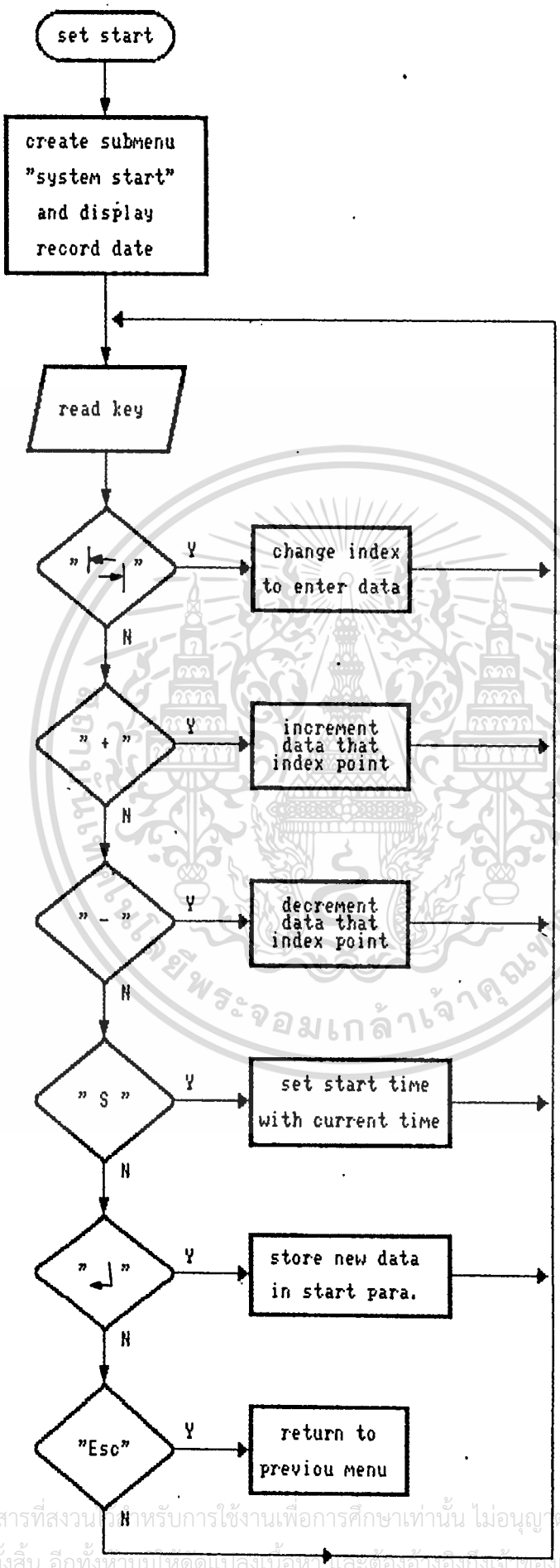
โหมดคีย์ " ESC " จะกลับสู่มอนิเตอร์เดิม

ส่วนที่ 4

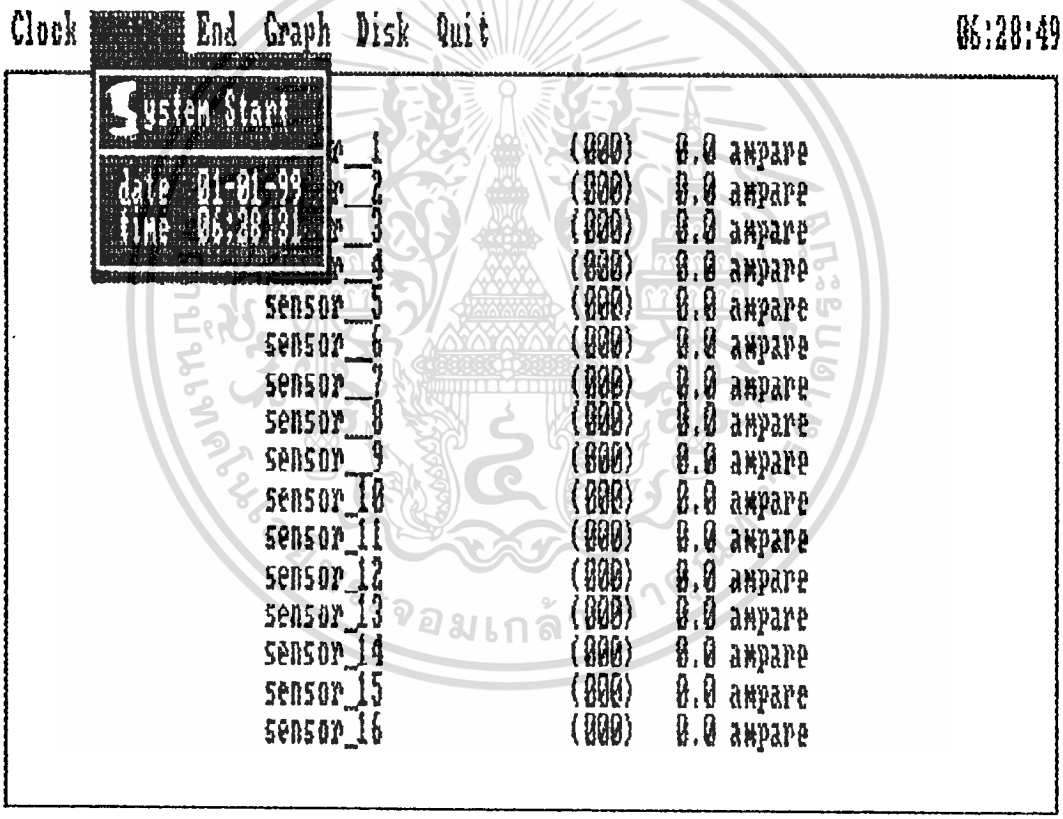
มีขั้นตอนการทำงานดังโฟลว์ชาร์ทรูปที่ 4F.1 , 4F.2 , 4F

รูปแบบโครงสร้าง โฟลว์ชาร์ท จะเหมือนโปรแกรมย่อย " set start " มาก ในช่วงของการรับคีย์ขั้นต้น จึงไม่ขออธิบายไว้ หลังจากการเช็คการกด คีย์ที่ผ่านมา จุดนี้จะเป็นจุดเริ่มต้นของการบันทึกข้อมูลลงดิสก์เก็ต ( การบันทึกข้อมูลลงดิสก์เก็ตที่กล่าวถึง จะถูกกระทำจริงเมื่อการนับเวลาจากการอินเทอรัพท์ของไทม์เมอร์ ครบรอบสแกนเวลาที่ตั้งไว้ มีผลต่อแฟลก หรือสถานะที่อนุญาติให้เกิดการบันทึกข้อมูลได้ 1 ครั้ง ) และจะมีโปรแกรมย่อยการเช็คระบบตรวจเวลาการทำงานว่า ค่าที่เซ็ทให้นั้น ( ค่าเริ่มต้นและสิ้นสุดการบันทึก ) มี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

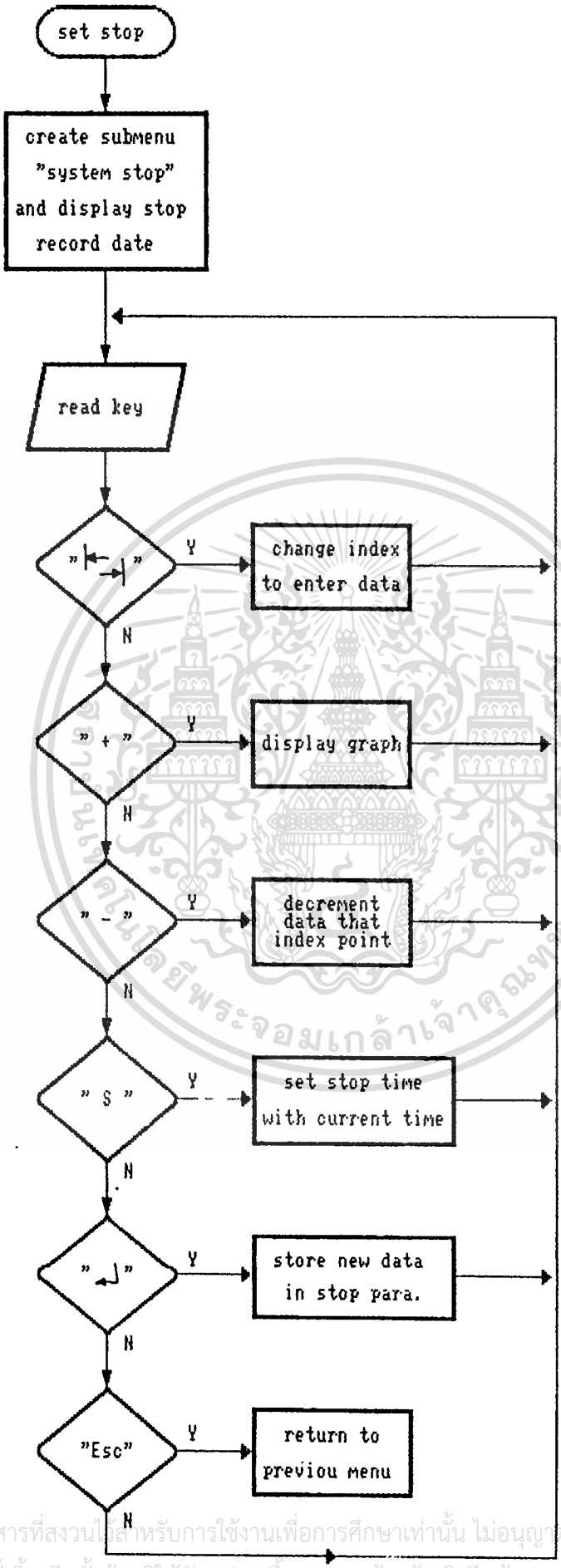



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



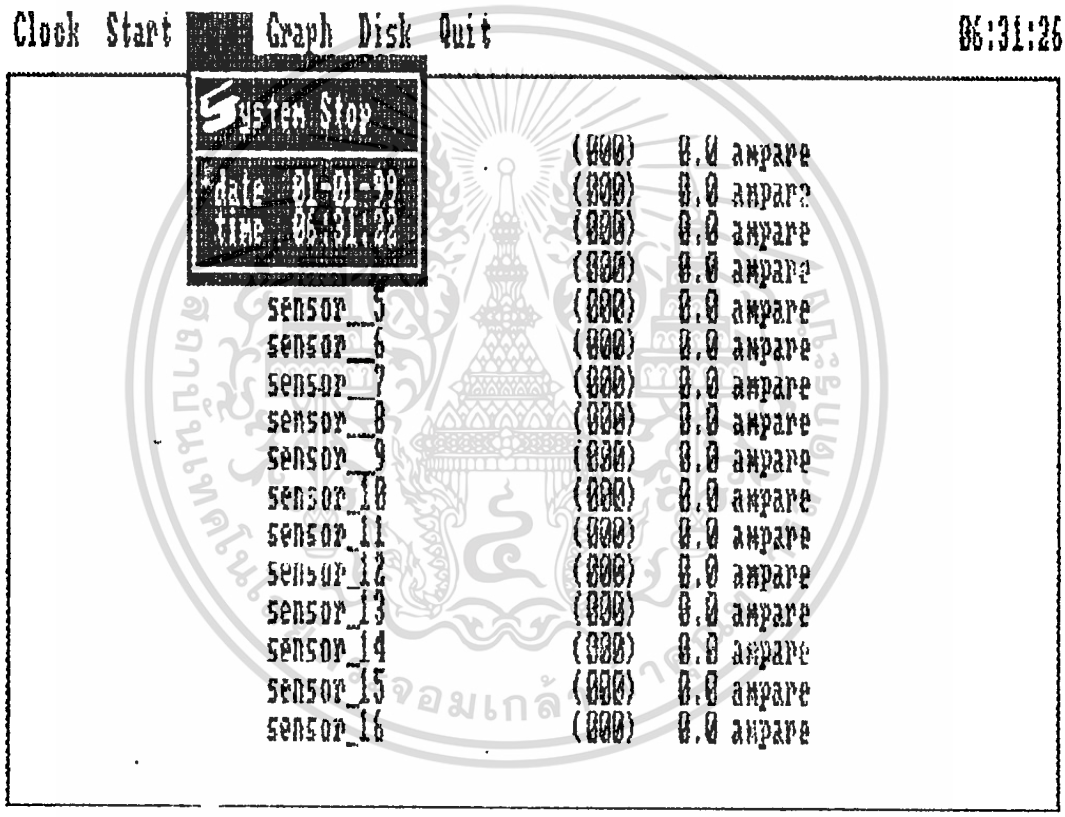

รูป : มอนิเตอร์โปรแกรม "เซ็ท สตาร์ท"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



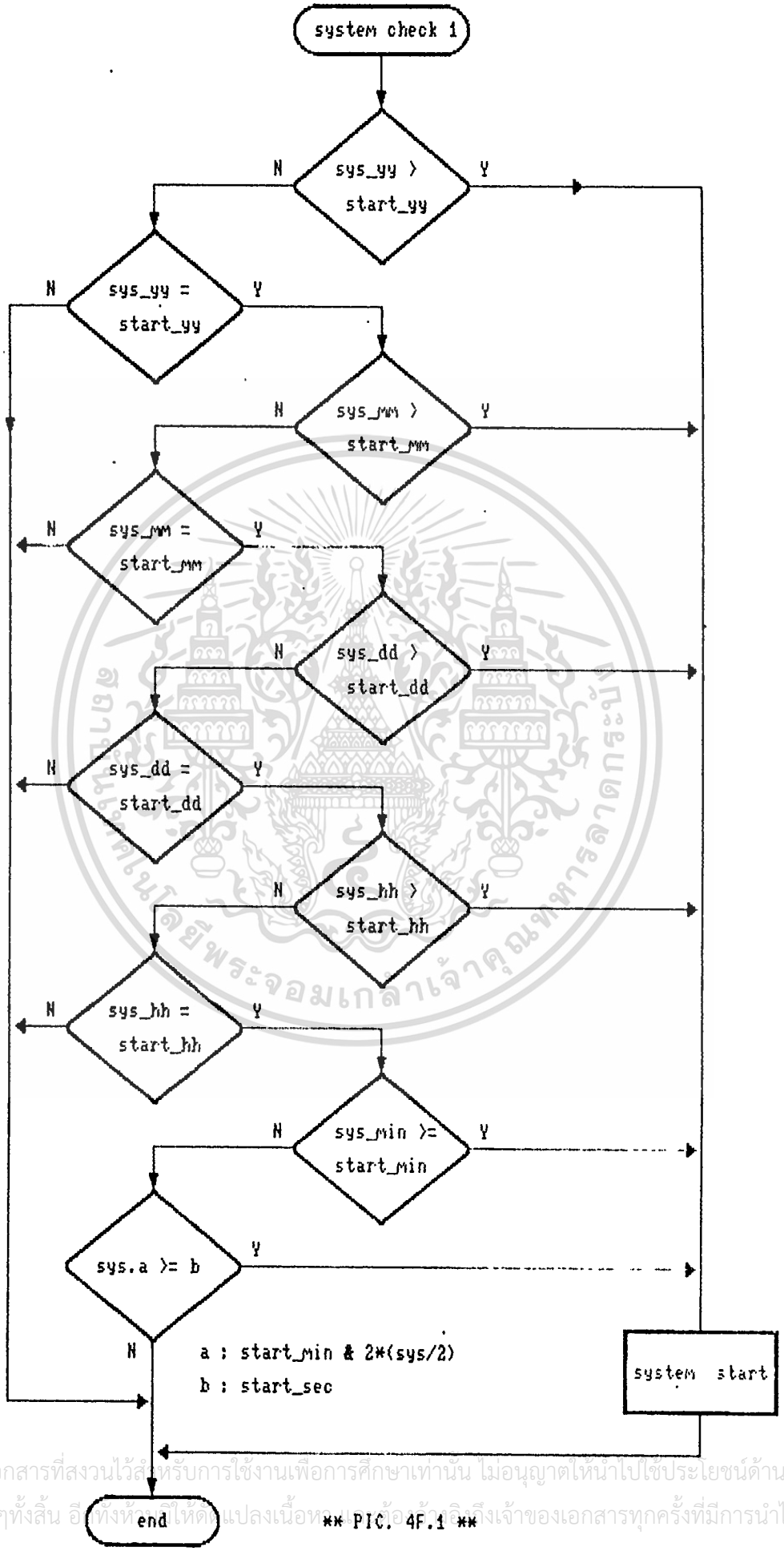

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

** PIC. 4F **



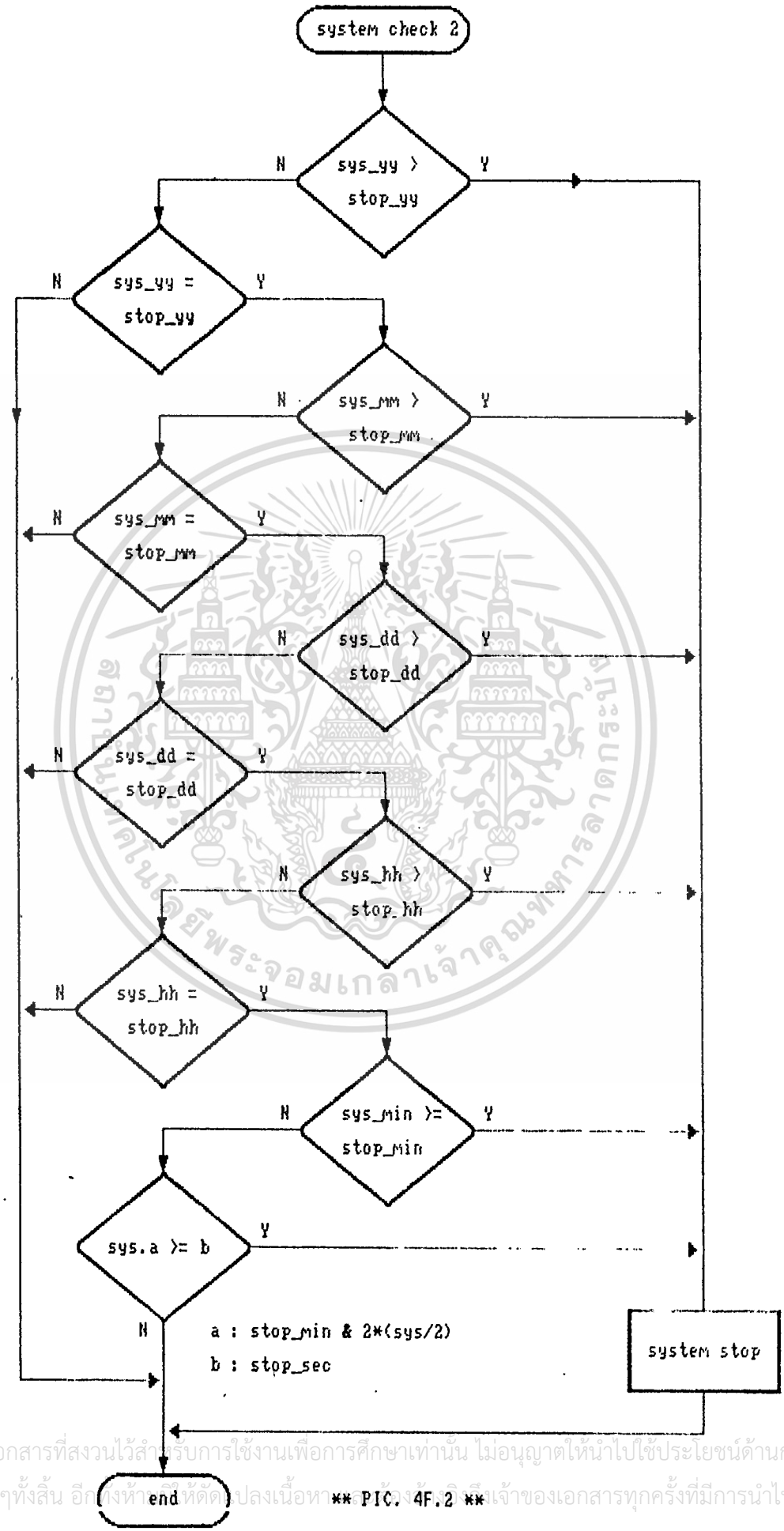

รูป : มอนิเตอร์โปรแกรม "เซ็ท สต็อป"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



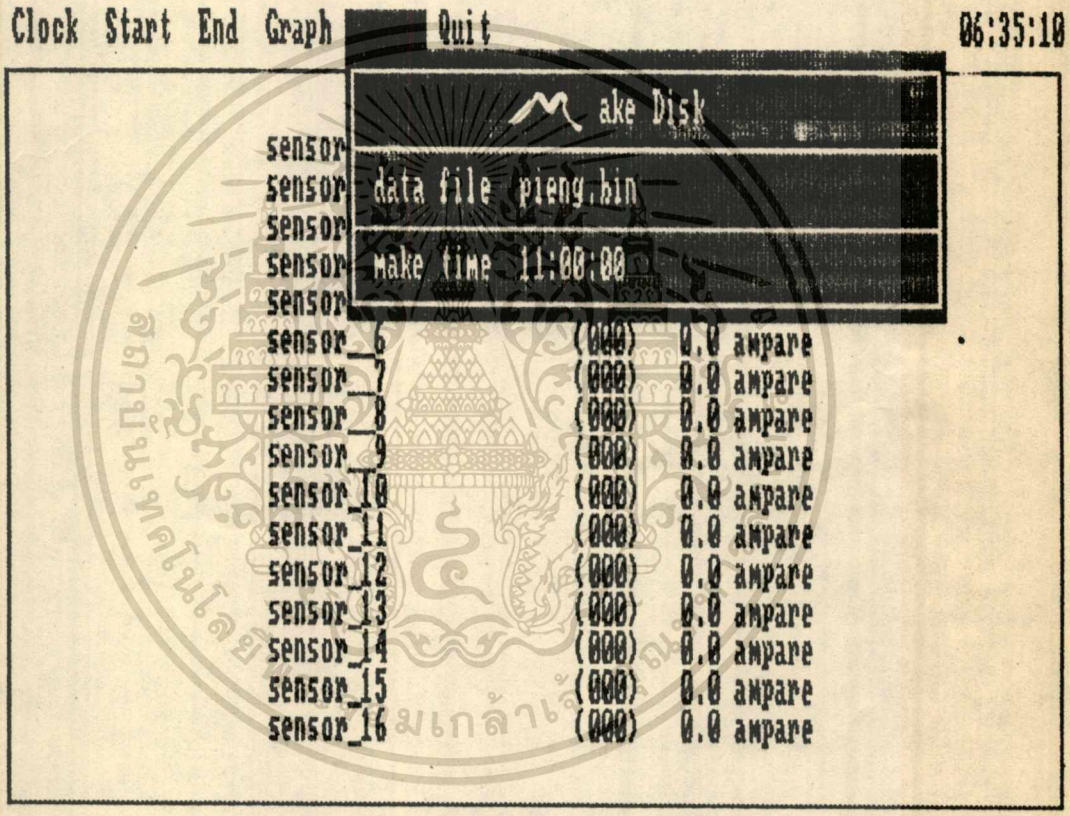

** PIC. 4F.1 **

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

** PIC. 4F.2 **




รูป : มอนิเตอร์โปรแกรม "อินนิเชียล ดิสเก็ต"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ช่วงเวลาการทำงานถูกต้องหรือไม่ โดยมีโปรแกรมโฟลว์ชาร์ทการเริ่มเช็คดังรูปที่ 4F.1 ซึ่งเป็นการเช็คว่าวันและเวลาของระบบมากกว่าวันและเวลาเริ่มต้นบันทึกค่าหรือยัง ถ้ามากกว่าแล้ว ก็ให้เริ่มต้นทำงานที่ system_start ได้โดยมีโฟลว์ชาร์ทดังรูปที่ 4.1

โปรแกรมเริ่มต้นของระบบ (system_start)

- ตั้งต้นเซ็ทค่าอินเด็กซ์ของการบันทึกข้อมูล เพื่อที่จะเริ่มต้นการบันทึกข้อมูล
- เปลี่ยนทิศทางการขัดจังหวะ จากตารางเวคเตอร์อินเทอรัพท์ ไปที่โปรซีเดอร์อินเทอรัพท์ไทม์เมอร์ที่ได้สร้างขึ้น
- อินนิเทียลดิสก์เก็ต ทำการเปิดไฟล์ แล้วทำการเขียนเวลาของระบบที่เริ่มต้นทำการบันทึก ลงในดิสก์เก็ต( ถ้าไฟล์ที่ถูกเปิดไม่มีตัวตน จะแสดงอักขระ " cannot make disk " ออกทางมอนิเตอร์) โดยรูปแบบเหมือนกับการเช็คเวลาเริ่มต้น ซึ่งจะเช็คเวลาของระบบว่ามากกว่าวันเวลาสิ้นสุดหรือไม่ ถ้าครบเวลาที่ตั้งไว้ก็ไปเริ่มต้นทำงานที่โปรแกรมหยุดการทำงาน

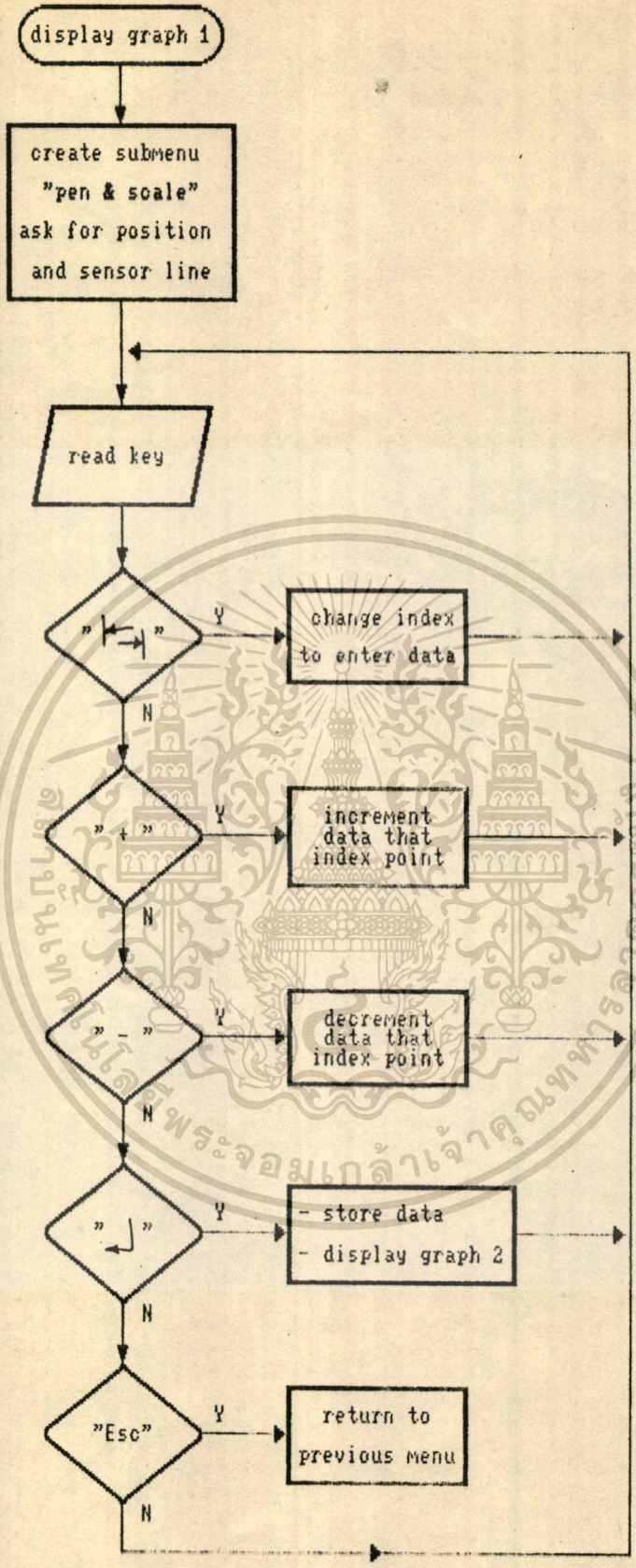
โปรแกรมหยุดการทำงาน (system_stop)

รีเซ็ทค่าตัวแปรต่างๆที่ถูกใช้ในการบันทึกข้อมูล และสร้างขึ้นจากโปรแกรมเริ่มต้นของระบบ

<u>ส่วนที่ 5</u>

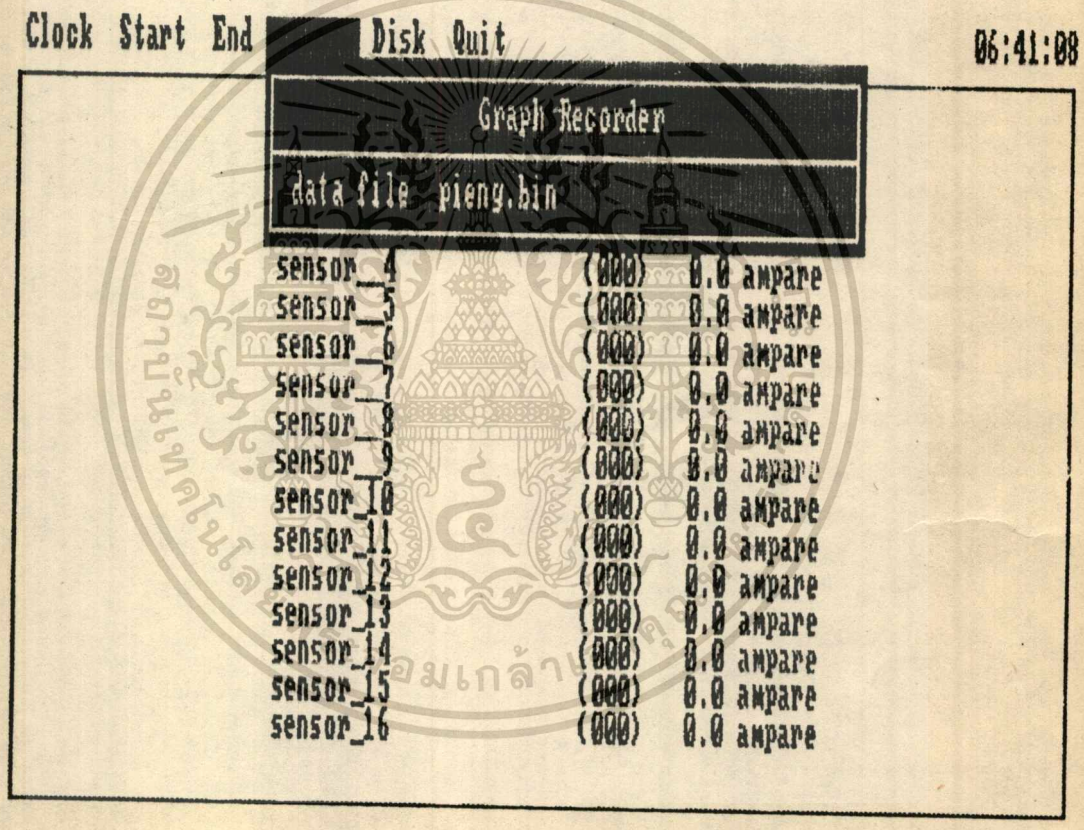
มีขั้นตอนการทำงานดังรูปที่ 5F ซึ่งมีโฟลว์ชาร์ท เป็นเมนโปรแกรมของส่วนแสดงกราฟ โดยจะมีโปรแกรมย่อยแยกไปเป็น โปรแกรม แสดงกราฟ 1 (display graph 1) โดยโปรแกรมหลักจะอ่านค่าข้อมูลของไฟล์ที่ต้องการ แล้วนำมาแสดงกราฟ โดยสามารถแสดงกราฟของข้อมูลจำนวนสูงสุดได้ไม่เกิน 4 แชนแนล พร้อมทั้งแสดงค่าเวลาและสเกลที่ต้องการเซ็ทไว้ออกทางจอภาพด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



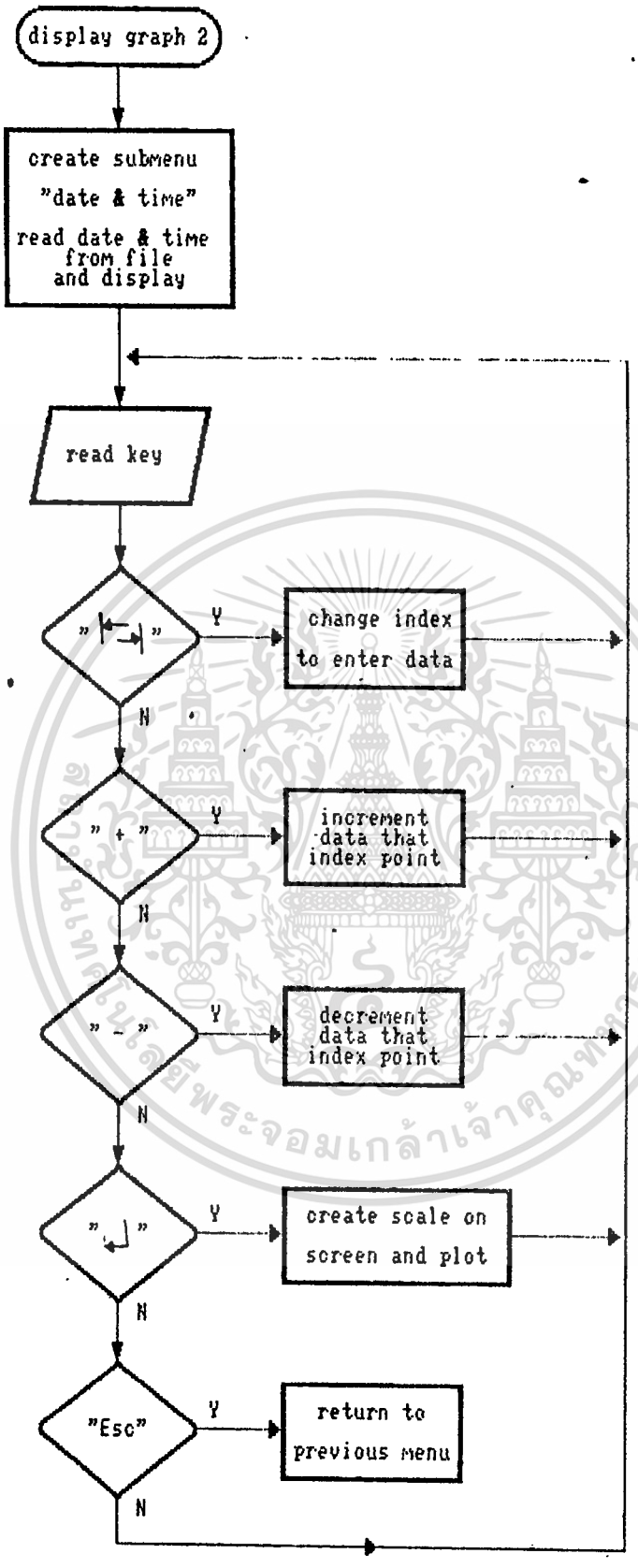

** PIC. 5F.1 **

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูป : มอนิเตอร์โปรแกรม "กราฟ รีคอร์ดเดอร์"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




** PIC. 5F.2 **

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 4

## การทดลองและผลการทดลอง

### 4.1 ภาคตรวจจับสัญญาณ

การทดลองในภาคนี้มีดังนี้

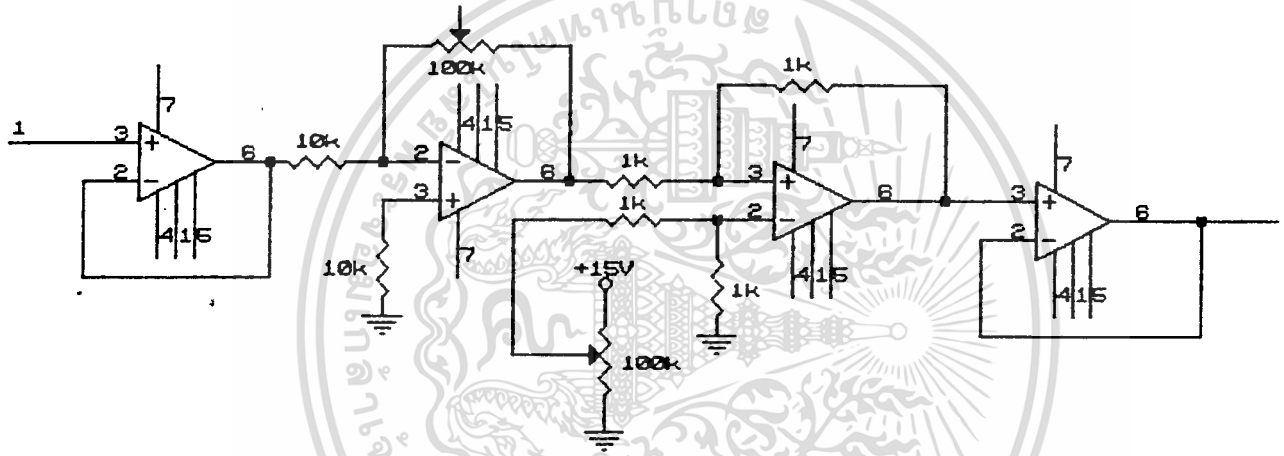
-หม้อแปลงกระแสไฟฟ้า จากการทดล่องเมื่อทำการวัดโดยนำหม้อแปลงกระแสไฟฟ้าไปคล้องกับโหลด หม้อแปลงกระแสไฟฟ้าจะทำงานมีลักษณะเป็นเชิงเส้น โดยมีอัตราส่วนคือค่ากระแสไฟฟ้าเท่ากับค่าของโวลท์เตจที่ผ่านได้คูณกับร้อย และตัวหม้อแปลงกระแสจะให้สัญญาณออกมาในรูปของโวลท์เตจซึ่งจะนำเอาไปเข้ากับส่วนของวงจรแปลงค่าโวลท์พีกให้เป็นโวลท์อาร์เอ็มเอส

-วงจรแปลงค่าโวลท์เตจกระแสสลับให้เป็นโวลท์เตจกำลังสองเฉลี่ย จากการทดลองวงจรนี้จะทำงานได้เป็นเชิงเส้นในช่วงอินพุท 0-5.6 โวลท์พีกกับเอาท์พุท 0-4 โลท์อาร์เอ็มเอส ซึ่งไม่เพียงพอกับการใช้งาน เนื่องจากบอร์ดควบคุมนั้นจะรับค่าอินพุท 1-6 โวลท์ จึงต้องมีการเพิ่มอัตราขยายและยกระดับสัญญาณให้อยู่ในช่วง 1-6 โวลท์อาร์เอ็มเอส และลดอัตราขยานทางอินพุทให้อยู่ในช่วง 0-5.6 โวลท์พีก ส่วนซีเลคเตอร์นั้นจะเป็นตัวเลือกช่วงของการวัดเพื่อให้ได้ข้อมูลที่ละเอียดขึ้น โดยการปรับสเกลทางซอฟท์แวร์บนคอมพิวเตอร์ ซึ่งวงจรทั้งหมดแสดงได้ดังรูปที่ 4.1 และจากการทดลองพบว่าสัญญาณเอาท์พุทจะได้ค่าในช่วง 1-6 โวลท์โดยมีค่าผิดพลาดประมาณ 25 มิลลิโวลท์

### 4.2 ภาคส่งสัญญาณ

ในการทดลองของภาคนี้ จะเป็นการทดลองกับบอร์ดควบคุมว่าสามารถใช้งานได้จริงหรือไม่ โดยต่อบอร์ดควบคุมเข้ากับเครื่องไมโครคอมพิวเตอร์ทางพอร์ท RS232 แล้วป้อนอินพุทให้กับบอร์ดควบคุม แล้วตรวจสอบดูว่าบอร์ดนี้จะรับค่าข้อมูลเข้าไป แล้วส่งออกไปยังคอมพิวเตอร์หรือไม่ซึ่งจากการทดลองพบว่าบอร์ดควบคุมทำงานได้จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




วงจรแปลงค่าโวลท์ผีกให้เป็นโวลท์อาร์เอ็มเอส (ต่อ)

4.3. ภาคแสดงผลและเก็บข้อมูล

เมื่อนำเอาแผ่นโปรแกรมไปรันบนคอมพิวเตอร์จะได้ผลของการแสดงส่วนเป็นมอนิเตอร์และฟังก์ชั่นการทำงานดังรูป

สรุปผลการทดลอง

เมื่อนำภาคตรวจจับสัญญาณภาคส่งสัญญาณ และภาคแสดงผลทั้งหมดมาต่อเข้าด้วยกันก็จะได้ผลการทดลองเป็นไปตามความต้องการ โดยจะทำการทดลองเพียง 1 แชนเนล ซึ่งผลที่ได้นั้น ข้อมูลที่วัดจะแสดงออกบนมอนิเตอร์ได้จริงและสามารถเก็บค่าข้อมูลลงบนดิสเก็ตโดยการตั้งเวลาเริ่มต้นและสิ้นสุดของการบันทึก พร้อมทั้งแสดงกราฟได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## TYPICAL APPLICATIONS (Contd.)

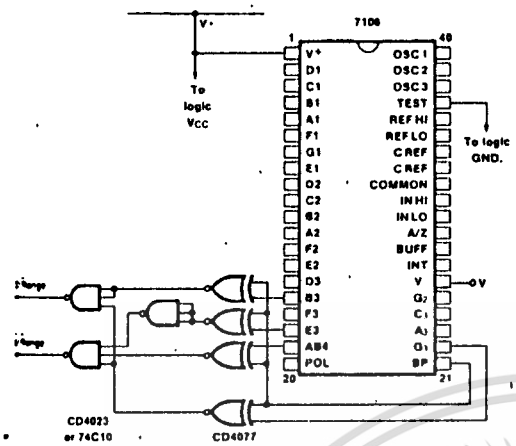



Figure 19: Circuit for developing Underrange and Overrange signals from 7106 outputs.

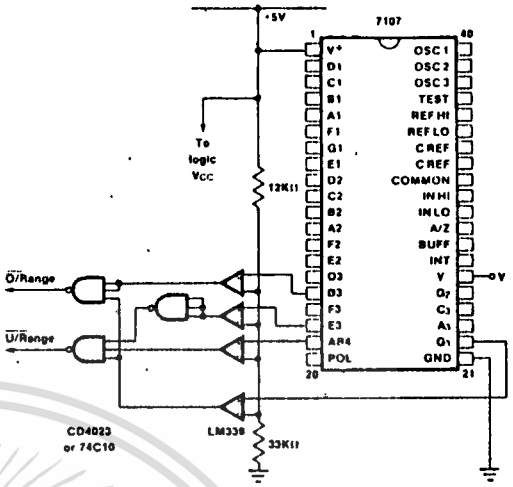



Figure 20: Circuit for developing Underrange and Overrange signals from 7107 outputs. The LM339 is required to ensure logic compatibility with heavy display loading.

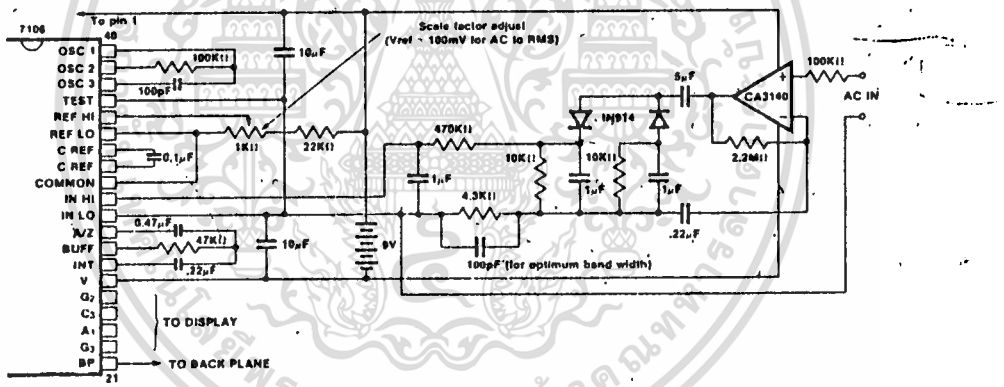



Figure 21: AC to DC Converter with 7106. Test is used as a common mode reference level to ensure compatibility with most op-amps.

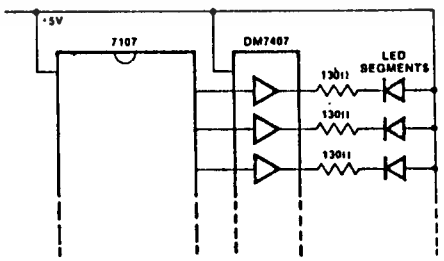
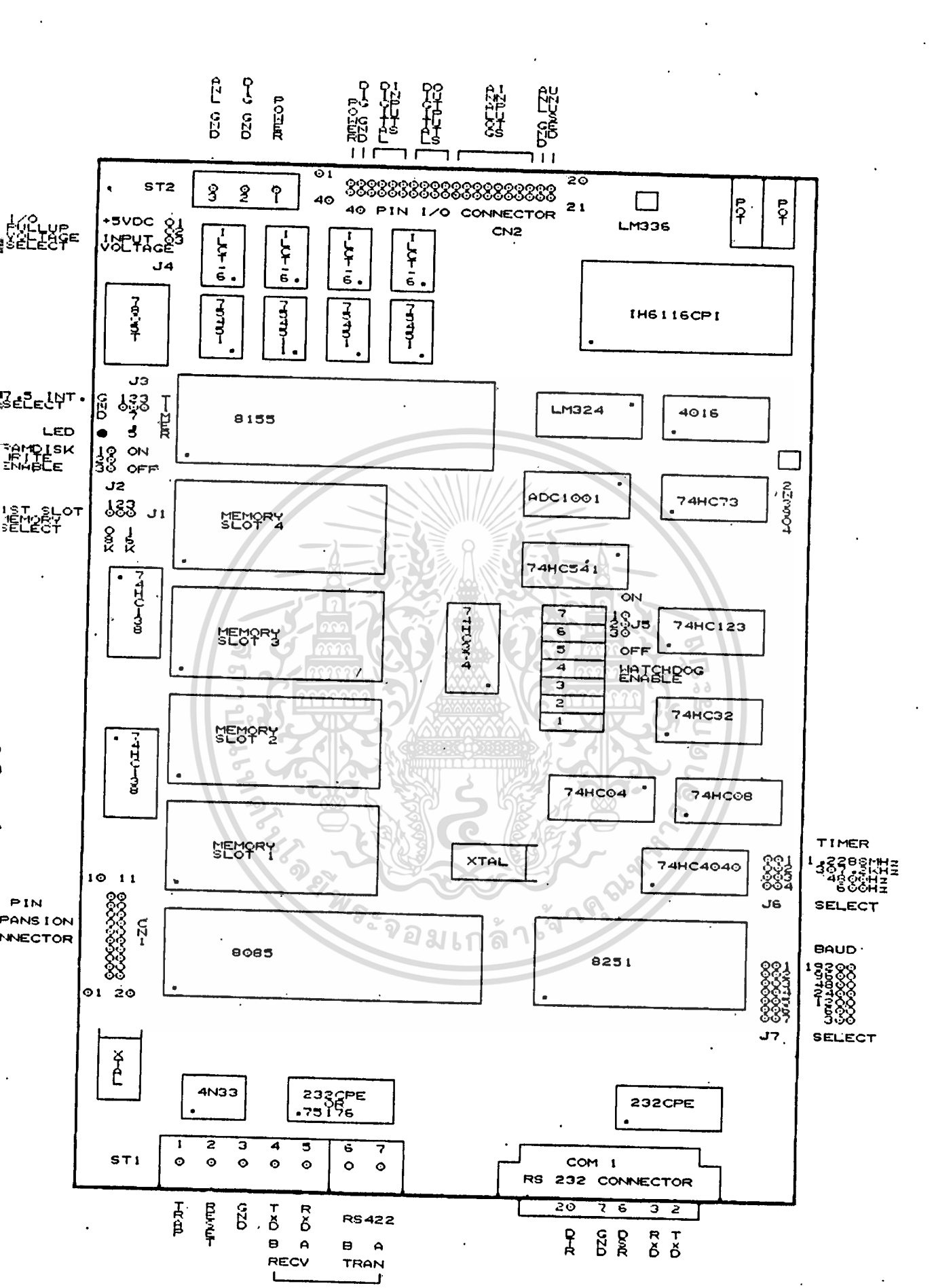
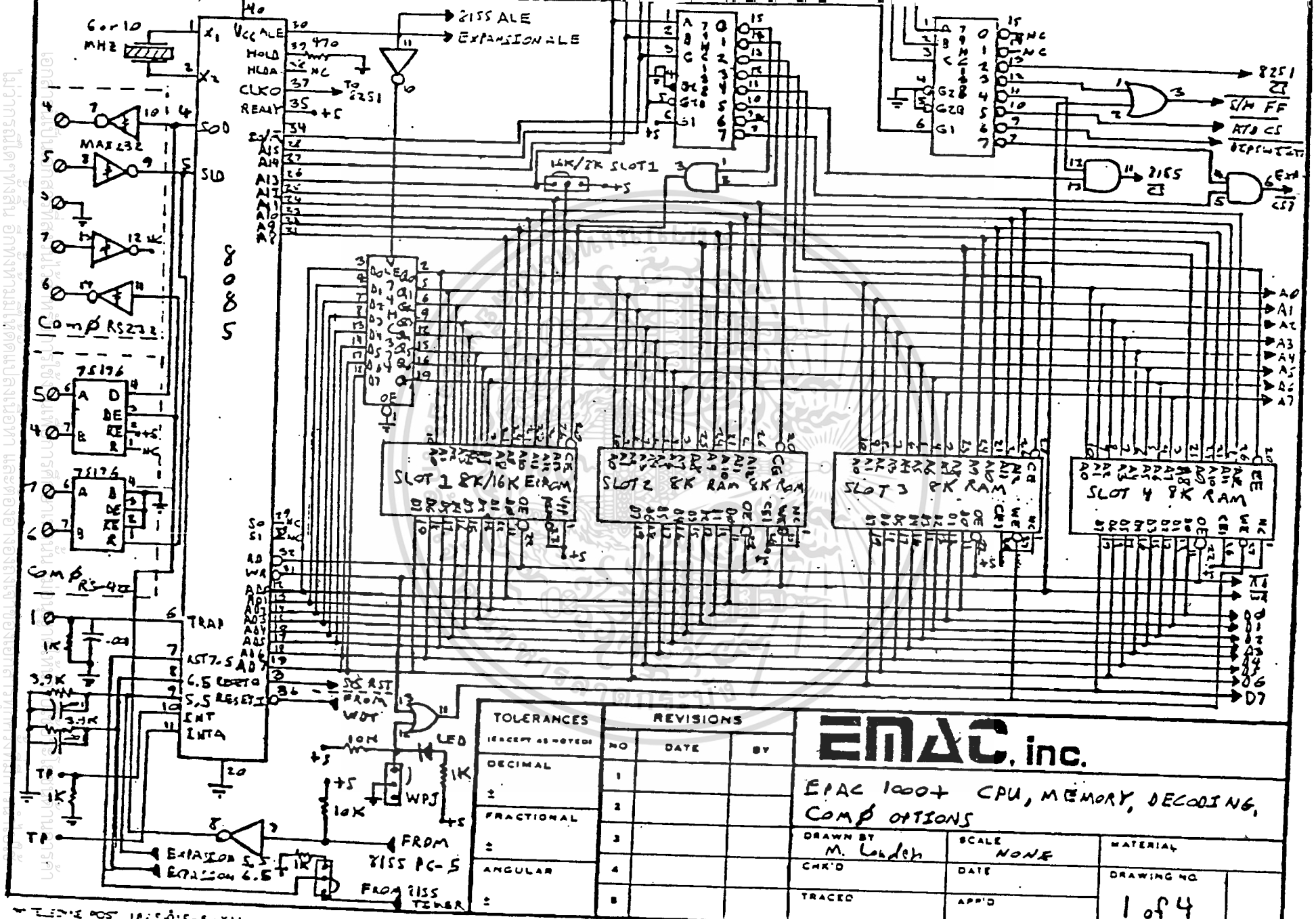



Figure 22: Display Buffering for increased drive current. Requires four DM7407 Hex Buffers. Each buffers is capable of sinking 40 mA max.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

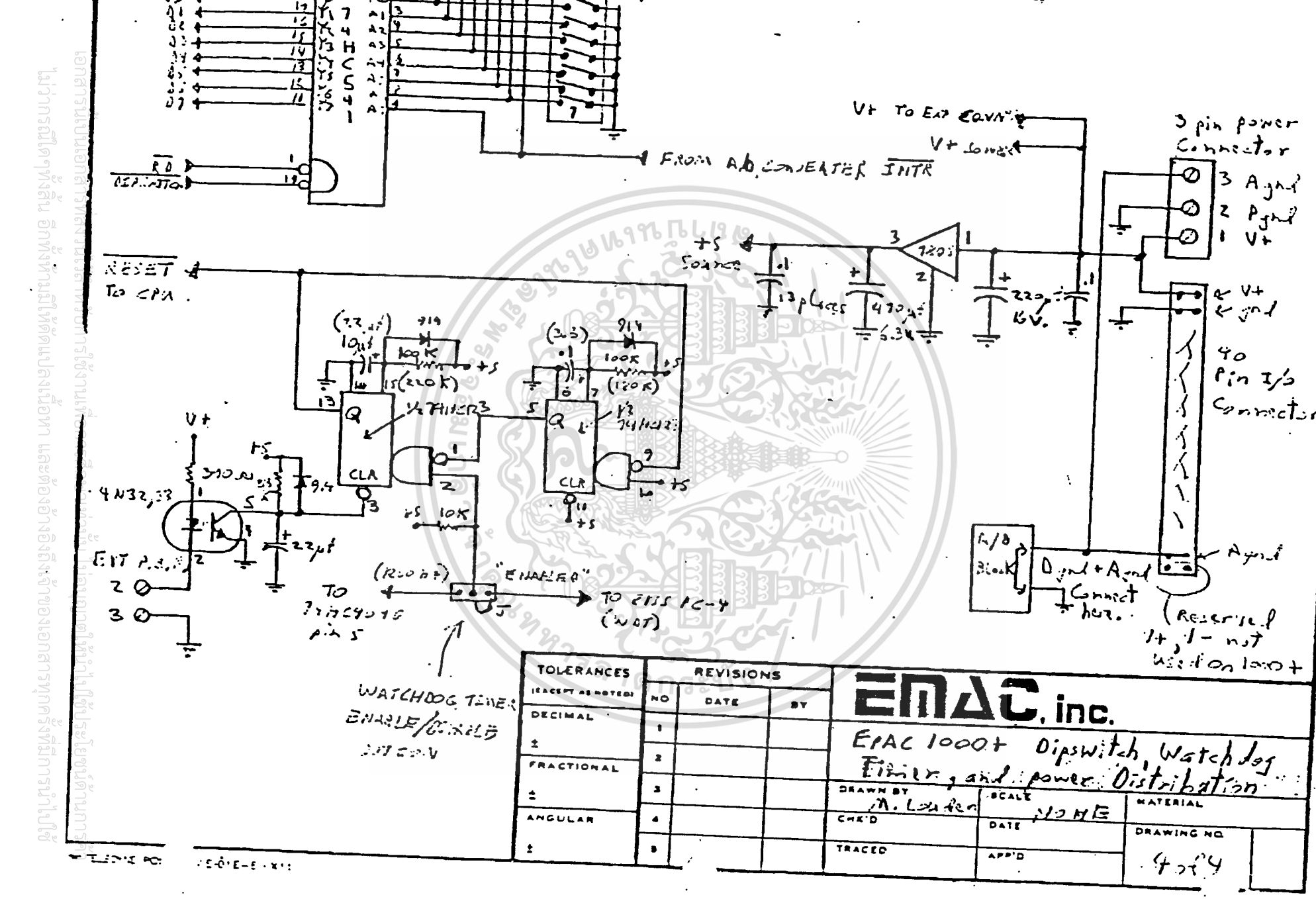



| TOLERANCES | REVISIONS | | |
|---|---|---|---|
| UNLESS OTHERWISE NOTED | NO | DATE | BY |
| DECIMAL ± | 1 | | |
| FRACTIONAL ± | 2 | | |
| ANGULAR ± | 3 | | |
| | 4 | | |
| | 5 | | |

EMAC, inc.

EPAC 1000+ CPU, MEMORY, DECODING, COMP OPTIONS

| DRAWN BY | SCALE | MATERIAL |
|---|---|---|
| M. Lodish | NONE | |
| CHK'D | DATE | DRAWING NO. |
| TRACED | APP'D | 1 of 4 |

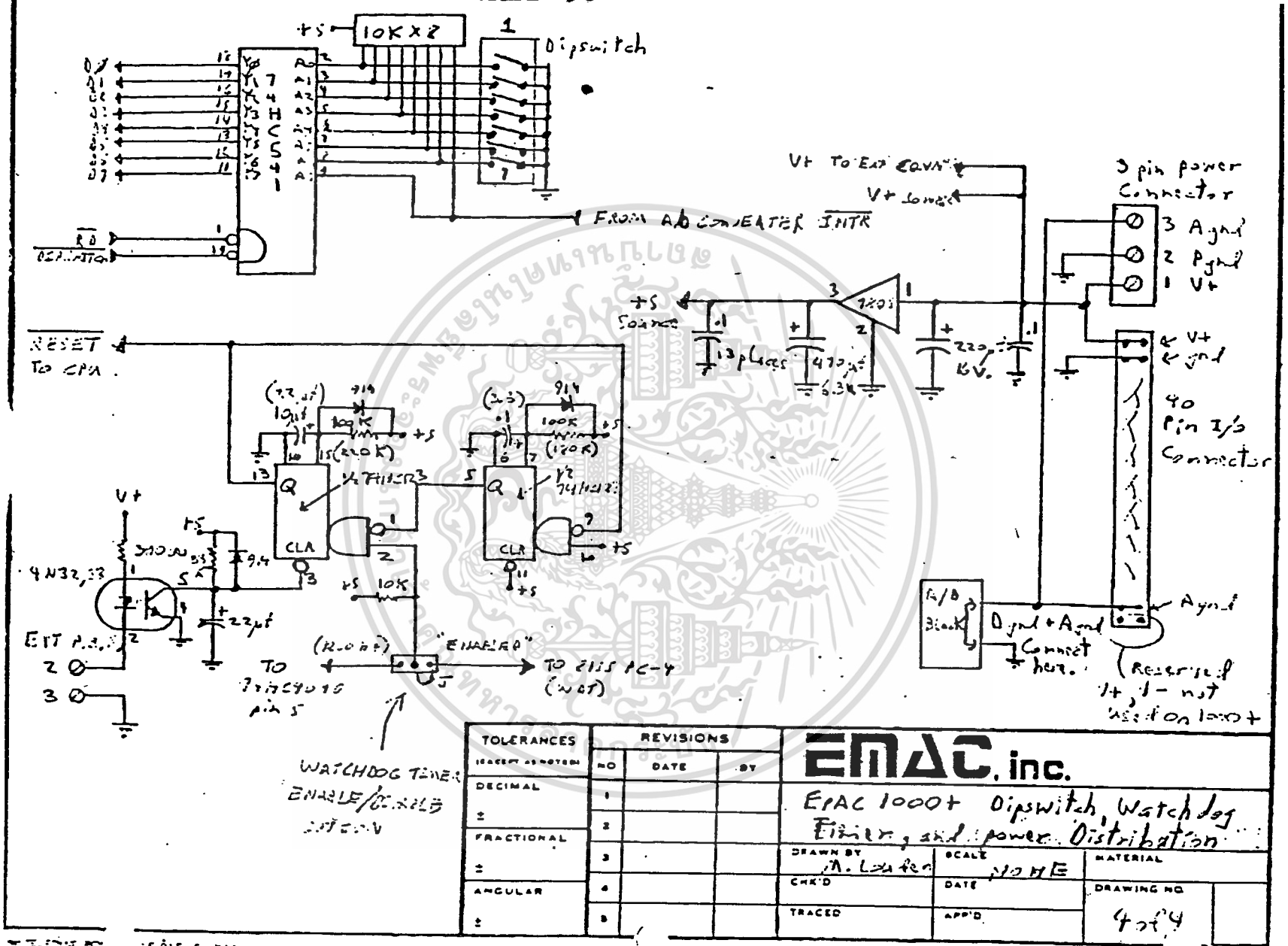



| TOLERANCES (EXCEPT AS NOTED) | REVISIONS | | |
|---|---|---|---|
| DECIMAL ± | NO | DATE | BY |
| FRACTIONAL ± | 1 | | |
| ANGULAR ± | 2 | | |
| | 3 | | |
| | 4 | | |
| | 5 | | |

EMAC, inc.

EPAC 1000+ Dipswitch, Watchdog Timer, and power Distribution

| | | |
|---|---|---|
| DRAWN BY A. Lauer | SCALE NONE | MATERIAL |
| CHK'D | DATE | DRAWING NO. 4 of 4 |
| TRACED | APP'D | |




| TOLERANCES | REVISIONS | | |
|---|---|---|---|
| | NO | DATE | BY |
| DECIMAL ± | 1 | | |
| FRACTIONAL ± | 2 | | |
| ANGULAR ± | 3 | | |
| | 4 | | |
| | 5 | | |

**EMAC, inc.**

EMAC 1000+ Dipswitch, Watchdog Timer, and Power Distribution

| | | |
|---|---|---|
| DRAWN BY A. Lawton | SCALE NONE | MATERIAL |
| CHK'D | DATE | DRAWING NO 4 of 4 |
| TRACED | APP'D | |

## กิตติกรรมประกาศ

ขอขอบพระคุณ อาจารย์ สุเชียร เกียรติสุนทร ที่ให้คำแนะนำและคำปรึกษา และขอบคุณเพื่อนๆทุกท่านที่ให้ความช่วยเหลือในการทำปริญญานิพนธ์ ทำให้ปริญญานิพนธ์ฉบับนี้สำเร็จไปได้ด้วยดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## หนังสืออ้างอิง

1 . George Alexy , " The 8086 Book including the 8088 " , Russel . . Rector , 7-1 p , 1980

2 . Stuart Bennet , " Real-time computer control "

3 . EMAC , inc. , " E-PAC control board manual " , copyright 1986

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้